mathematics

MDPI

Article

From Time–Frequency to Vertex–Frequency and Back

Ljubiša Stanković [1], Jonatan Lerga [2,*], Danilo Mandic [3], Miloš Brajović [1], Cédric Richard [4] and Miloš Daković [1]

1 Faculty of Electrical Engineering, University of Montenegro, 81000 Podgorica, Montenegro; ljubisa@ucg.ac.me (L.S.); milosb@ucg.ac.me (M.B.); milos@ucg.ac.me (M.D.)
2 Center for Artificial Intelligence and Cybersecurity, Faculty of Engineering, University of Rijeka, 51000 Rijeka, Croatia
3 Faculty of Electrical Engineering, Imperial College London, London SW72AZ, UK; d.mandic@imperial.ac.uk
4 Electrical Egineering, University Cote d'Azur, 06100 Nice, France; cedric.richard@unice.fr
* Correspondence: jlerga@riteh.hr

Abstract: The paper presents an analysis and overview of vertex–frequency analysis, an emerging area in graph signal processing. A strong formal link of this area to classical time–frequency analysis is provided. Vertex–frequency localization-based approaches to analyzing signals on the graph emerged as a response to challenges of analysis of big data on irregular domains. Graph signals are either localized in the vertex domain before the spectral analysis is performed or are localized in the spectral domain prior to the inverse graph Fourier transform is applied. The latter approach is the spectral form of the vertex–frequency analysis, and it will be considered in this paper since the spectral domain for signal localization is well ordered and thus simpler for application to the graph signals. The localized graph Fourier transform is defined based on its counterpart, the short-time Fourier transform, in classical signal analysis. We consider various spectral window forms based on which these transforms can tackle the localized signal behavior. Conditions for the signal reconstruction, known as the overlap-and-add (OLA) and weighted overlap-and-add (WOLA) methods, are also considered. Since the graphs can be very large, the realizations of vertex–frequency representations using polynomial form localization have a particular significance. These forms use only very localized vertex domains, and do not require either the graph Fourier transform or the inverse graph Fourier transform, are computationally efficient. These kinds of implementations are then applied to classical time–frequency analysis since their simplicity can be very attractive for the implementation in the case of large time-domain signals. Spectral varying forms of the localization functions are presented as well. These spectral varying forms are related to the wavelet transform. For completeness, the inversion and signal reconstruction are discussed as well. The presented theory is illustrated and demonstrated on numerical examples.

Keywords: time–frequency; vertex–frequency; graph; big data

check for updates

Citation: Stanković, L.; Lerga, J.; Mandic, D.; Brajović, M.; Richard, C.; Daković, M. From Time–Frequency to Vertex–Frequency and Back. *Mathematics* **2021**, *9*, 407. https://doi.org/10.3390/math9121407

Academic Editors: Paolo Crippa and Rafael Sebastian

Received: 17 May 2021
Accepted: 11 June 2021
Published: 17 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Processing of big data, whose domain is irregular and can be represented by a graph, has attracted significant research interest [1–10]. For big data, the possibility of using smaller and localized subsets of the available information is crucial for their efficient analysis and processing [11]. In addition, in practical applications when large graphs are used as the signal domain, we are commonly interested in localized analysis than in global behavior. In order to characterize the vertex-localized behavior of signals and their narrow-band spectral properties, the joint vertex–frequency domain analysis is introduced. This analysis represents a natural analogy to the time–frequency analysis, a well-established area in classical signal processing [12–14].

In classical signal analysis, the basic short-time Fourier transform approach uses window functions to localize the signal in time, while the projection of such a windowed

signal onto Fourier transform basis functions provides its spectral localization. Time localization, combined with the modulation by the basis functions, produces kernel functions for classical time–frequency analysis. The classical time–frequency analysis approach has been extended to vertex–frequency analysis for signals defined on graphs [15–22]. This generalization is not straightforward, since graph is a complex and irregular signal domain. Namely, even a time-shift operation, which is trivial in classical time-domain analysis, cannot be straightforwardly generalized to the graph signal domain. This has resulted in several approaches to define vertex–frequency kernels. One approach is based on the vertex domain windows defined using the graph spectral domain [23]. The vertex domain windows can also be fully defined in the vertex domain, using the vertex neighborhood [19].

The vertex domain approaches are based on local analysis with a vertex neighborhood and can be very efficient in the large graph analysis. This paper will focus on the vertex–frequency kernels defined in the spectral domain, with spectral shifts performed as in classical signal analysis, while the vertex shifts are implemented in an indirect way, using the basis functions. This approach produces practically very efficient forms, especially when combined with polynomial approximations of the analysis kernels. This paper's primary goal is to provide a strong link of the time–frequency analysis with vertex–frequency analysis and to indicate some new possibilities for simple methods in the time–frequency analysis of large-duration signals based on the vertex–frequency forms. Conditions for the signal reconstruction, known as overlap-and-add (OLA) method and weighted overlap-and-add (WOLA) are considered, and the window forms from the classical signal analysis are adapted to satisfy these conditions, with appropriate comments related to their application to the vertex–frequency analysis, when the eigenvalues are used instead of the frequency.

The paper is structured as follows. Basic definitions in graph theory and signals on graphs, including the graph Fourier transform, are reviewed in Section 2. A solid formal relation between the classical signal processing paradigm and graph signal processing is provided in Section 3, where benchmark graphs and signals are introduced. In Section 4, the spectral-domain localized graph Fourier transform is presented, along with a few simple basic implementation forms. The general OLA and WOLA conditions for analysis in the graph spectral domain are introduced, with illustration on several windows for each of these conditions, including the spectral domain wavelet-like transform. The polynomial approximations of the presented kernels are the topic of Section 5, where the Chebyshev polynomial series, least-squares approximation, and Legendre polynomial approximation are presented. Inversion of the local graph Fourier transform is elaborated on in Section 6, where both of the defined kernel forms are analyzed. The support uncertainty principle in the general form (such that it can be used for graph signals) is presented in Section 7, along with the discussion on the relation of the local graph Fourier transform support and the kernel function width in the spectral domain. The possibility of splitting large signals into smaller parts and simplifying the analysis of such signals is considered in Section 8. The presented theory is illustrated in numerous examples. The manuscript closes with summarized conclusions and the reference list.

2. Basic Graph Definitions

A graph consists of N vertices, $n \in \mathcal{V} = \{1, 2, \ldots, N\}$, which are connected with edges. The weight of edges are W_{mn} [24–26]. For the vertices m and n which are not connected, by definition $W_{mn} = 0$. The weights of edges are the elements of an $N \times N$ matrix, $\mathbf{W}$. The graphs can be directed and undirected. For undirected graphs it is assumed that the vertices m and n are connected by the same edge weight in both directions, resulting in a symmetric weight matrix $\mathbf{W}$, when $\mathbf{W} = \mathbf{W}^T$ holds. A graph is unweighted if all nonzero elements of its weight matrix, $\mathbf{W}$, are equal to 1. In this case the weight matrix, $\mathbf{W}$, assumes specific form and the edges are represented by a connectivity or adjacency matrix, $\mathbf{A}$. In addition to the adjacency and weight matrix, $\mathbf{A}$ or $\mathbf{W}$, in graph theory several

other matrices are used. All of them can be derived from the adjacency and weight matrix. A matrix that indicates the vertex degree in a graph is called the degree matrix. It is of diagonal form and its common notation is **D**. The elements D_{nn} of the degree matrix are obtained as a sum of all weights corresponding to the edges connected to the considered vertex, n. The diagonal elements of **D** are equal to $D_{nn} = \sum_m W_{mn}$. A combination of the weight matrix, **W**, and the degree matrix, **D**, produces one of the most commonly used matrix in the graph theory, the graph Laplacian. It is defined by

$$\mathbf{L} = \mathbf{D} - \mathbf{W}.$$

In the case of an undirected graph, the symmetric form of the weight matrix results in a symmetric graph Laplacian, $\mathbf{L} = \mathbf{L}^T$.

The eigendecomposition of the graph matrices (for example, of the graph Laplacian **L** or the adjacency matrix **A**) is used for spectral analysis of graphs and graph signals. The eigendecomposition of a graph Laplacian (or any other matrix) relates its eigenvalues, λ_k, and the corresponding eigenvectors, $\mathbf{u}_k$, by

$$\mathbf{L}\mathbf{u}_k = \lambda_k \mathbf{u}_k, \text{ for } k = 1, 2, \ldots, N.$$

where λ_1, λ_2, ..., λ_N, are not necessarily distinct. Since the graph Laplacian is a real-valued symmetric matrix, it is always diagonalizable, that is, the geometric multiplicity equals the algebraic multiplicity for every eigenvalue. The previous N equations can then be written in a compact matrix form (the eigendecomposition relation for diagonizable matrices) as

$$\mathbf{L}\mathbf{U} = \mathbf{U}\mathbf{\Lambda}.$$

The transformation matrix **U** consists of the eigenvectors, $\mathbf{u}_k$, $k = 1, 2, \ldots, N$, as its columns, while $\mathbf{\Lambda}$ is matrix of diagonal form, whose diagonal elements are $\lambda_k, k = 1, 2, \ldots, N$.

The same eigendecomposition relation can be used for the adjacency matrix $\mathbf{A}\mathbf{u}_k = \lambda_k \mathbf{u}_k, k = 1, 2, \ldots, N$.

For diagonalizable matrix there exist a set of orthonormal eigenvectors. They are used as the transformation basis functions for the definition of the graph Fourier transform (GFT),

$$\mathbf{X} = [X(1), X(2), \ldots, X(N)]^T,$$

of a graph signal, $\mathbf{x} = [x(1), x(2), \ldots, x(N)]^T$. The graph signal value at a vertex n is denoted by $x(n)$, $n = 1, 2, \ldots, N$, while the notation **x** is used for the vector of signal values at all vertices. The vector of the GFT of a graph signal **x** will be denoted by **X**, and the elements (components) of the GFT vector by $X(k)$, $k = 1, 2 \ldots, N$. The elements of a graph signal at a vertex n, $x(n)$, can then be written as a linear combination of the eigenvectors

$$x(n) = \text{IGFT}\{X(k)\} = \sum_{k=1}^{N} X(k)\ u_k(n), \tag{1}$$

where the basis function values $\mathbf{u}_k$, are the elements of the k-th eigenvector, $\mathbf{u}_k$, at the vertex n, $n = 1, 2, \ldots, N$. This is the definition of the inverse graph Fourier transform (IGFT).

Matrix form of the IGFT is $\mathbf{x} = \mathbf{U}\mathbf{X}$. For real and symmetric matrices (corresponding to undirected graphs) the transformation matrix **U** is orthogonal, $\mathbf{U}\mathbf{U}^T = \mathbf{I}$, that is $\mathbf{U}^{-1} = \mathbf{U}^T$. Then the graph Fourier transform (GFT) is defined by $\mathbf{X} = \mathbf{U}^{-1}\mathbf{x} = \mathbf{U}^T\mathbf{x}$ or in element-wise form

$$X(k) = \text{GFT}\{x(n)\} = \sum_{n=1}^{N} x(n)\ u_k(n). \tag{2}$$

For undirected graphs, both the Laplacian and the adjacency matrix are symmetric, resulting in real-valued eigenvectors and the resulting transformation matrices. However, for directed circular graphs, the eigenvalues (and eigenvectors) of the adjacency matrix are

complex-valued. Then, the elements of the inverse transformation matrix $\mathbf{U}$ should be used in the GFT definition. When $\mathbf{U}^{-1} = \mathbf{U}^H$ holds (normal matrices), the complex-conjugate basis functions, $u_k^*(n)$, are used in (2).

3. Classical Signal Processing within the Graph Signal Processing Framework

The graph signal processing will be related to classical time–frequency analysis in two ways: (1) using the directed circular graph and its adjacency matrix, or (2) using the undirected circular graphs and the graph Laplacian. These two relations are discussed next.

Directed circular graph. The signal values, $x(n)$, in classical signal processing systems, are defined in a well-ordered time domain, defined by the time instants denoted by $n = 1, 2, \ldots, N$. In the DFT-based classical analysis it has also been assumed that the signal is periodic. The domain of such signals is illustrated in Figure 1 for $N = 8$.

Consider next a classical form of a discrete-time finite impulse response (FIR) system. The input–output relation for this system is given by

$$y(n) = h_0 x(n) + h_1 x(n-1) + h_2 x(n-2) + \cdots + h_M x(n-M).$$

In order to make a connection with graphs and graph notation of the signal domain, notice that this input–output relation of the FIR system can be written in the matrix form as

$$\mathbf{y} = h_0 \mathbf{x} + h_1 \mathbf{A}\mathbf{x} + h_2 \mathbf{A}^2 \mathbf{x} + \cdots + h_M \mathbf{A}^M \mathbf{x}, \tag{3}$$

where

$$\mathbf{y} = \begin{bmatrix} y(1) \\ y(2) \\ y(3) \\ \vdots \\ y(N) \end{bmatrix}, \ \mathbf{x} = \begin{bmatrix} x(1) \\ x(2) \\ x(3) \\ \vdots \\ x(N) \end{bmatrix}, \text{ and } \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 & 1 \\ 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 1 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & 0 \end{bmatrix}.$$

For this system, the time instants are well-ordered and their connectivity matrix is given by the adjacency matrix. The instants (in graph notation vertices) relation is defined by

- $A_{mn} = 1$ if the vertex or instant m is a predecessor (connected) to the instant (vertex) n, and
- $A_{mn} = 0$ otherwise,

as shown in Figure 1 (left), for $N = 8$. Rows of the adjacency matrix indicate the corresponding vertex connectivity. In the first row, there is value 1 at the position N. It means that the vertex 1 is related to vertex N by a directed edge between these two vertices. In the second row there is value 1 at the first position, meaning that the vertex 2 is connected to vertex 1 with an edge, as shown in in Figure 1 (left).

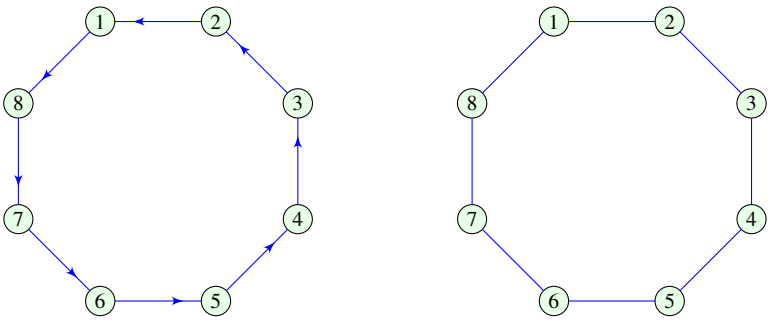


Figure 1. Time domain of periodic signals presented as: Circular unweighted directed graph (**left**) and an undirected graph (**right**), with $N = 8$ vertices (instants).

The elements of the shift matrix relation $\mathbf{y} = \mathbf{A}\mathbf{x}$ are $y(n) = x(n-1)$, as it has been expected for a simple delay operation. The delay for two instants, $y(n) = x(n-2)$, is calculated as $\mathbf{A}(\mathbf{A}\mathbf{x}) = \mathbf{A}^2\mathbf{x}$, and so on.

Now, we will perform the eigendecomposition of this adjacency matrix $\mathbf{A}$. According to the eigendecomposition relation

$$\mathbf{A}\mathbf{u}_k = \lambda_k \mathbf{u}_k,$$

or in the matrix form

$$\mathbf{A}\mathbf{U} = \mathbf{U}\mathbf{\Lambda} \text{ or } \mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}.$$

Recall that $\mathbf{U}$ is the matrix whose columns are the eigenvectors, $\mathbf{u}_k$, and the eigenvalue diagonal matrix is $\mathbf{\Lambda}$. The eigenvalues λ_k are on the diagonal of this matrix. The adjacency matrix of the directed circular graph is diagonalizable because all its eigenvalues are distinct. The adjacency matrix of a directed circular graph is a circulant matrix. As it is well-known that this kind of matrix is diagonalizable by the discrete Fourier transform [27] (as it will be shown next). In general, the adjacency matrix of a directed graph may not be diagonalizable, when the Jordan form should be used [1] and Appendix A in [3]. This kind of graphs is not considered in the paper. The input–output relation of a classical FIR system (3) can now be written as

$$\mathbf{y} = h_0\mathbf{x} + h_1\mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}\mathbf{x} + \cdots + h_M\mathbf{U}\mathbf{\Lambda}^M\mathbf{U}^{-1}\mathbf{x},$$

where the eigendecomposition property

$$\mathbf{A}^M = \mathbf{U}\mathbf{\Lambda}^M\mathbf{U}^{-1},$$

is used.

Now, by left-multiplication by $\mathbf{U}^{-1}$ we can write

$$\mathbf{U}^{-1}\mathbf{y} = h_0\mathbf{U}^{-1}\mathbf{x} + h_1\mathbf{\Lambda}\mathbf{U}^{-1}\mathbf{x} + \cdots + h_M\mathbf{\Lambda}^M\mathbf{U}^{-1}\mathbf{x},$$

or

$$\mathbf{Y} = (h_0 + h_1\mathbf{\Lambda} + h_2\mathbf{\Lambda}^2 + \cdots + h_M\mathbf{\Lambda}^M)\mathbf{X} = H(\mathbf{\Lambda})\mathbf{X},$$

where

$$\mathbf{Y} = \mathbf{U}^H\mathbf{y} \text{ and } \mathbf{X} = \mathbf{U}^H\mathbf{x},$$

are the discrete Fourier transforms (DFT) of the output signal, $\mathbf{y}$, and the input signal, $\mathbf{x}$. The diagonal transfer function is denoted by $H(\mathbf{\Lambda})$ and its elements are given by

$$H(\lambda_k) = \frac{Y(k)}{X(k)},$$

for $X(k) \neq 0$. Indeed, the presented forms represent the well-known classical DFT-based relations. In order to confirm this conclusion, we will analyze the eigenvalue relation for the presented adjacency matrix, $\mathbf{A}$,

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0.$$

The corresponding characteristic polynomial is given by

$$\det(\mathbf{A} - \lambda\mathbf{I}) = \lambda^N - 1 = 0.$$

Since $1 = e^{-j2\pi(k-1)}$, the solutions for the eigenvalues and eigenvectors are

$$\lambda_k = e^{-j2\pi(k-1)/N} \text{ with } u_k(n) = \frac{1}{\sqrt{N}}e^{j2\pi(k-1)(n-1)/N},$$

for $k = 1, 2, \ldots, N$. The eigenvectors are equal to the DFT basis functions, normalized in such a way that their energy is unity.

We can easily arrive to the element-wise form of the DFT using the GFT definition given in (2).

For implementation issues that will be addressed later, it is crucial to notice that for the calculation of $\mathbf{A}^M\mathbf{x}$ we need only the signal neighborhood M with respect to the each considered instant, n. In the time domain, it means the distance defined by $(n - M)$. The fact that $\mathbf{A}^M\mathbf{x}$ requires the signal samples within M neighborhood of the considered vertex (instant) will hold for general graphs. The local neighborhood based calculation is of key importance when large graphs are analyzed or signals, representing big data on large graphs, are processed.

Undirected circular graph. When the circular graph is not directed, as shown in Figure 1(right), then we should assume that every instant (vertex), n, is connected to both the predecessor vertex (instant), $n - 1$, and to the succeeding vertex, $n + 1$. The adjacency or weight matrix for this kind of connection, $\mathbf{A} = \mathbf{W}$, and the corresponding graph Laplacian, defined by $\mathbf{L} = \mathbf{D} - \mathbf{W}$, are given by

$$\mathbf{W} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & \ldots & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & 0 & \ldots & 1 & 0 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 2 \end{bmatrix},$$

$$\mathbf{L} = \mathbf{D} - \mathbf{W} = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & \ldots & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -1 & 0 & 0 & 0 & 0 & \ldots & -1 & 2 \end{bmatrix},$$

where $\mathbf{D}$ is the diagonal (degree) matrix with elements $D_{nn} = \sum_m W_{mn}$. The eigendecomposition relation written for the graph Laplacian, $\mathbf{L}\mathbf{u}_k = \lambda_k \mathbf{u}_k$, in element-wise form is given by

$$\mathcal{L}_{u_k}(n) = -u_k(n-1) + 2u_k(n) - u_k(n+1) = \lambda_k u_k(n), \tag{4}$$

where $\mathcal{L}_x(n)$ are the elements of the vector $\mathbf{L}\mathbf{x}$.

The solution to the difference equation of the second order, (4), can be obtained in the form

$$u_k(n) = \cos\left(\frac{2\pi(k-1)(n-1)}{N} + \phi_k\right), \tag{5}$$

with the eigenvalue

$$\lambda_k = 2\left(1 - \cos\left(\frac{2\pi(k-1)}{N}\right)\right) = 4\sin^2\left(\frac{1}{2}\frac{2\pi(k-1)}{N}\right) = 4\sin^2\left(\frac{1}{2}\omega_k\right). \tag{6}$$

For each of the eigenvalues, we can define two distinct orthogonal eigenvectors in quadrature, for example, using $\phi_k = 0$ and $\phi_k = \pi/2$ in (5). These two eigenvectors correspond to the classical sinusoidal basis functions, $\cos(2\pi(k-1)(n-1)/N)$ and $\sin(2\pi(k-1)(n-1)/N)$, in the Fourier series analysis of real-valued signals. The exceptions are the eigenvalues $\lambda_1 = 0$ and the last eigenvalue for an even N, when there is only one basis function. The sine and cosine functions should be normalized to the unit energy, to represent eigenvectors. Therefore, a definition of the graph Laplacian eigenvalues

and eigenvectors for an undirected circular graph (for an even N, for example, $N = 8$), taking into account all previous properties, is given by

$$\begin{array}{ll}
\lambda_1 = 0 & u_1(n) = \frac{1}{\sqrt{8}} \\
\lambda_2 = 4\sin^2\left(\frac{\pi}{8}\right) & u_2(n) = \frac{1}{\sqrt{4}}\cos(\frac{2\pi(n-1)}{8}), \\
\lambda_3 = 4\sin^2\left(\frac{\pi}{8}\right) & u_3(n) = \frac{1}{\sqrt{4}}\sin(\frac{2\pi(n-1)}{8}) \\
\lambda_4 = 4\sin^2\left(\frac{2\pi}{8}\right) & u_4(n) = \frac{1}{\sqrt{4}}\cos(\frac{2\pi 2(n-1)}{8}), \\
\lambda_5 = 4\sin^2\left(\frac{2\pi}{8}\right) & u_5(n) = \frac{1}{\sqrt{4}}\sin(\frac{2\pi 2(n-1)}{8}) \\
\lambda_6 = 4\sin^2\left(\frac{3\pi}{8}\right) & u_6(n) = \frac{1}{\sqrt{4}}\cos(\frac{2\pi 3(n-1)}{8}), \\
\lambda_7 = 4\sin^2\left(\frac{3\pi}{8}\right) & u_7(n) = \frac{1}{\sqrt{4}}\sin(\frac{2\pi 3(n-1)}{8}) \\
\lambda_8 = 4\sin^2\left(\frac{4\pi}{8}\right) & u_8(n) = \frac{1}{\sqrt{4}}\cos(\frac{2\pi 4(n-1)}{8}).
\end{array} \tag{7}$$

The smallest eigenvalue, $\lambda_1 = 0$, corresponds to a constant vector, $u_1(n) = 1/\sqrt{8}$, while the largest eigenvalue, $\lambda_8 = 4$, corresponds to the fastest-varying eigenvector $u_8(n) = (-1)^{(n-1)}/\sqrt{4}$.

Smoothness and local smoothness. Notice that for an undirected circular graph and small frequency, ω_k^2, the relation in (6) can be approximated by

$$\lambda_k = 4\sin^2(\frac{1}{2}\omega_k) \approx \omega_k^2. \tag{8}$$

This relation means that the graph Laplacian eigenvalue, λ_k, corresponding to the eigenvector, $\mathbf{u}_k$, can be related to the classical frequency (squared), ω_k^2, of a sinusoidal basis function in classical Fourier series analysis.

In general, it is easy to show that the eigenvalue of the graph Laplacian can be used to indicate the speed of change (called the smoothness) of an eigenvector or a graph signal, in general. Namely, if we left-multiply by $\mathbf{u}_k^T$ both sides of the eigenvalue definition relation $\mathbf{L}\mathbf{u}_k = \lambda_k\mathbf{u}_k$ we obtain $\mathbf{u}_k^T\mathbf{L}\mathbf{u}_k = \lambda_k$, since $\mathbf{u}_k^T\mathbf{u}_k = 1$. Now the quadratic form

$$\lambda_k = \mathbf{u}_k^T\mathbf{L}\mathbf{u}_k = \frac{1}{2}\sum_m\sum_n(u_k(n) - u_k(m))^2 W_{mn},$$

measures the change of neighboring values $(u_k(n) - u_k(m))^2$, weighted by W_{mn}. Fast changes of $u_k(n)$ produce large values of λ_k, while the constant $u_k(n)$ results in $\lambda_k = 0$.

The local smoothness can be defined for a vertex n. It will be denoted by $\lambda(n)$. This parameter corresponds to the classical time-varying (instantaneous) frequency, $\omega(t)$, defined at an time-instant t, in the form [28]

$$\lambda(n) = \frac{\mathcal{L}_x(n)}{x(n)}. \tag{9}$$

In this relation we used $\mathcal{L}_x(n)$ to denote the n-th element of the vector $\mathbf{Lx}$. It has been assumed that $x(n) \neq 0$. If we use $x(n) = \cos(\omega_k n)$ and the graph Laplacian of an undirected circular graph, as in (4), we obtain the value from (8). In general, if the signal $x(n)$ is equal to an eigenvector $u_k(n)$ at the vertex n and at its neighboring vertices, then $\lambda(n) = \lambda_k$.

System on a general graph. The relations presented in this section are the special cases of the general graph Fourier transform (Section 2) and systems for graph signals.

The most important difference between the classical systems and the systems for graph signals is in the fact that the standard shift operator, $x(n) = x(n-1)$, just moves a signal sample from one instant, n, to another instant, $(n-1)$, while the graph shift operator, $\mathbf{y} = \mathbf{A}\mathbf{x}$ or $\mathbf{y} = \mathbf{L}\mathbf{x}$, moves the signal sample to all neighboring vertices (in the case of graph Laplacian, in addition to the signal being moved to the neighboring vertices (with a change of sign), its sample is kept at the original vertex as well). Notice that the graph shift operator does not satisfy the isometry property since the shifted signal's energy is not the same as the energy of the original signal. In analogy with the role of time shift in standard system theory, a system for graph signals is implemented as a linear combination of a graph signal and its graph shifted versions,

$$\mathbf{y} = \left(h_0\mathbf{L}^0 + h_1\mathbf{L}^1 + \cdots + h_{M-1}\mathbf{L}^{M-1}\right)\mathbf{x} = \left(\sum_{m=0}^{M-1} h_m\mathbf{L}^m\right)\mathbf{x} = H(\mathbf{L})\,\mathbf{x}, \tag{10}$$

where, by definition $\mathbf{L}^0 = \mathbf{I}$, while $h_0, h_1, \ldots, h_{M-1}$ are the system coefficients. The spectral form of this relation is given by

$$\mathbf{Y} = \left(h_0 + h_1\mathbf{\Lambda} + h_2\mathbf{\Lambda}^2 + \cdots + h_M\mathbf{\Lambda}^M\right)\mathbf{X} = H(\mathbf{\Lambda})\mathbf{X}, \tag{11}$$

where $H(\mathbf{\Lambda})$ is a diagonal matrix representing the transfer function of the system for a graph signal. Notice that if the transfer function, in general, can be written in a form of polynomial, as in (11), then the system can be implemented using the graph-shifted forms of the signal, $\mathbf{L}\,\mathbf{x}$, $\mathbf{L}^2\,\mathbf{x}, \ldots$, up to $\mathbf{L}^{M-1}\,\mathbf{x}$, as in (10), which require only $(M-1)$ neighborhoods of each signal sample to obtain the system output, independently of the size of the considered graph.

Graph signal filtering—Graph convolution. Three approaches to filtering of a graph signal using a system whose transfer function is $G(\mathbf{\Lambda})$, with elements on the diagonal $G(\lambda_k)$, $k = 1, 2, \ldots, N$, will be presented next.

(i) The simplest approach is based on the direct employment of the GFT. It is performed by:
   (a) Calculating the GFT of the input signal, $\mathbf{X} = \mathbf{U}^{-1}\mathbf{x}$,
   (b) Finding the output signal GFT by multiplying $\mathbf{X}$ by $G(\mathbf{\Lambda})$, $\mathbf{Y} = G(\mathbf{\Lambda})\mathbf{X}$,
   (c) Calculating the output (filtered) signal as the inverse DFT of $\mathbf{Y}$, $\mathbf{y} = \mathbf{U}\mathbf{Y}$.

   The result of this operation,

$$y(n) = x(n) * g(n) = \text{IGFT}\{\text{GFT}\{x(n)\}\text{GFT}\{g(n)\}\} = \text{IGFT}\{X(k)G(\lambda_k)\},$$

   is called *a convolution of signals on a graph* [25,29].
   However, this procedure could be *computationally unacceptable for very large graphs*.

(ii) A possibility to avoid the full size transformation matrices for large graphs, is to approximate the filter transfer function, $G(\lambda)$, at the positions of the eigenvalues, $\lambda = \lambda_k$, $k = 1, 2, \ldots, N$, by a polynomial, $h_0 + h_1\lambda + h_2\lambda^2 + \cdots + h_M\lambda^M$, that is

$$h_0 + h_1\lambda_k + h_2\lambda_k^2 + \cdots + h_M\lambda_k^M = G(\lambda_k), \quad k = 1, 2, \ldots, N. \tag{12}$$

   Then the system of N equations

$$\mathbf{V}\mathbf{h} = \text{diag}\{\mathbf{G}\}, \tag{13}$$

   is solved, in the least squared sense, for $M < N$ unknown parameters of the system $\mathbf{h} = [h_0, h_1, \ldots, h_M]^T$, with a given M and

$$\text{diag}\{\mathbf{G}\} = [G(\lambda_1), G(\lambda_2), \ldots, G(\lambda_N)]^T$$

is the column vector of diagonal elements of $\mathbf{G}$. The elements of matrix $\mathbf{V}$ are $V(k,m) = \lambda_k^m$, $m = 0, 1, \ldots, M$, $k = 1, 2, \ldots, N$ (Vandermonde matrix).
This system can efficiently be solved for a relatively small M. Then, the implementation of the graph filter is performed in the vertex domain using the so obtained $h_0, h_1, \ldots, h_M$ in (10) and the M-neighborhood of a every considered vertex. Notice that the relation between the IGFT of diag$\{\mathbf{G}\}$ and the system coefficients $h_0, h_1, \ldots, h_M$ is direct in the classical DFT case only, while it is more complex in the general graph case [25].
For large M, the solution to the system of equations in (12), for the unknown parameters $h_0, h_1, \ldots, h_M$ can be *numerically unstable due to large values of the powers* λ_k^M for large M.

(iii) Another approach that allows us to avoid the direct GFT calculation in the implementation of graph filters is in approximating the given transfer function, $G(\lambda)$, by a polynomial $H(\lambda)$, using continuous variable λ.
This approximation does not guarantee that the transfer functions $G(\lambda)$ *and its polynomial approximation* $H(\lambda)$ *will be close at a discrete set of points* $\lambda = \lambda_p$, $p = 1, 2, \ldots, N$. The maximal absolute deviation of the polynomial approximation can be kept as small as possible using the so-called *min–max polynomials*. After the polynomial approximation is obtained the output of the graph system is calculated using (10), that is

$$\mathbf{y} = \Big(\sum_{m=0}^{M-1} h_m \mathbf{L}^m \Big) \mathbf{x} = H(\mathbf{L})\, \mathbf{x}.$$

This approach will be presented in Section 5.

Case study examples. In the next example we shall introduce two graphs and signals on these graphs, which will be used as benchmark models for the analysis that follows.

Example 1. *Two graphs are shown in Figure 2. A circular undirected unweighted graph represents the domain for classical signal analysis, with each of* $N = 100$ *vertices (instants) being connected to the predecessor and successor vertices (top panel). A general form of a graph, with the same number of* $N = 100$ *vertices, is shown in Figure 2 (bottom). These two graphs will be further used to demonstrate classical and graph signal processing principles and relations.*

A signal on the circular graph is shown in Figure 3 (top). We have formed this synthetic signal using parts of three graph Laplacian eigenvectors (corresponding to three harmonics in classical analysis). For the vertices in the subset $\mathcal{V}_1 = \{1, 2, 3, \ldots, 40\} \subset \mathcal{V}$, *the eigenvector (harmonic) with the spectral index* $k = 16$ *was used.*

For the subset $\mathcal{V}_2 = \{41, 42, 43, \ldots, 70\}$, *the eigenvector* $u_k(n)$, *with* $k = 84$, *is used to define the signal. The eigenvector with spectral index* $k = 29$ *was used to define the signal on the remaining set of vertices,* $\mathcal{V}_3 \subset \mathcal{V}$.

A signal on the general graph is shown in Figure 3 (bottom). It is also composed of parts of three Laplacian eigenvectors. For the vertices in $\mathcal{V}_1$, *the eigenvector with spectral index* $k = 12$ *has been used.*

For the subset of vertices $\mathcal{V}_2$, *containing the vertex indices ranging from* $n = 41$ *to* $n = 70$, *the eigenvector* $u_k(n)$ *with* $k = 84$ *was used to define the signal. Within the subset,* $\mathcal{V}_3 = \mathcal{V} \setminus (\mathcal{V}_1 \cup \mathcal{V}_2)$, *the spectral index was* $k = 29$. *Supports of these three components are designated by different vertex colors.*

The local smoothness index $\lambda(n)$, *which corresponds to the speed of change of the corresponding components,* $\lambda(n) = \lambda_k$, *is shown in Figure 4 for the presented graph signals. The local smoothness in the classical signal analysis is related to the instantaneous frequency of each signal components as* $\lambda(n) = 4 \sin^2(\omega(n)/2)$.

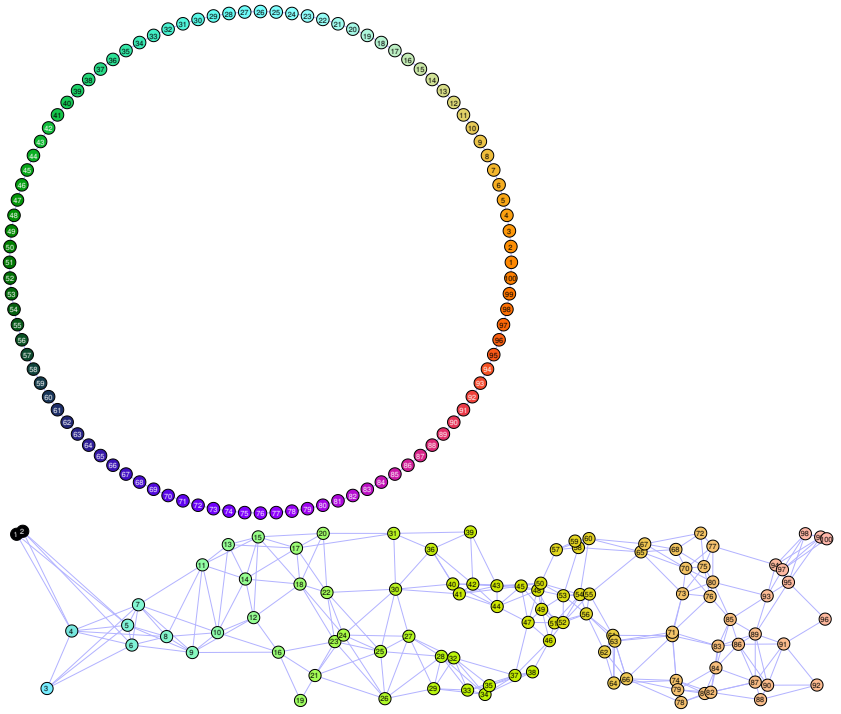


Figure 2. A circular undirected unweighted graph as the domain for classical signal analysis. Each of $N = 100$ vertices (instants) is connected to the predecessor and successor vertices (top). A general form of a graph, with $N = 100$ vertices (bottom).

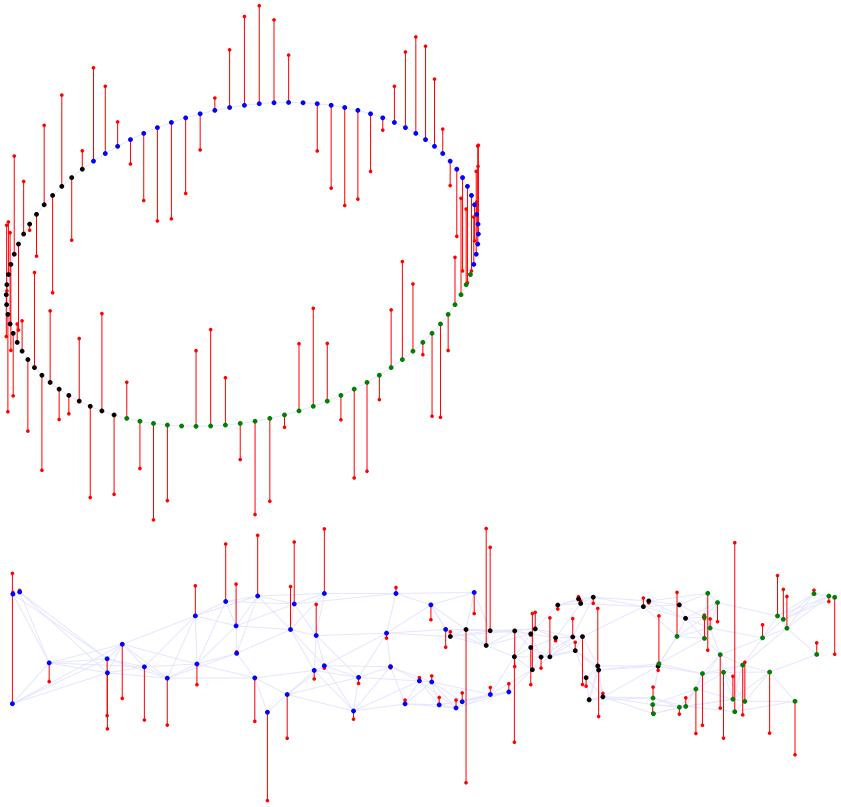


Figure 3. Graph signal on a circular undirected unweighted graph (top), and a general graph (bottom). Vertices from $\mathcal{V}_1$ are designated blue dots, vertices form $\mathcal{V}_2$ are marked by black dots, while vertices form $\mathcal{V}_3$ are given by green dots.

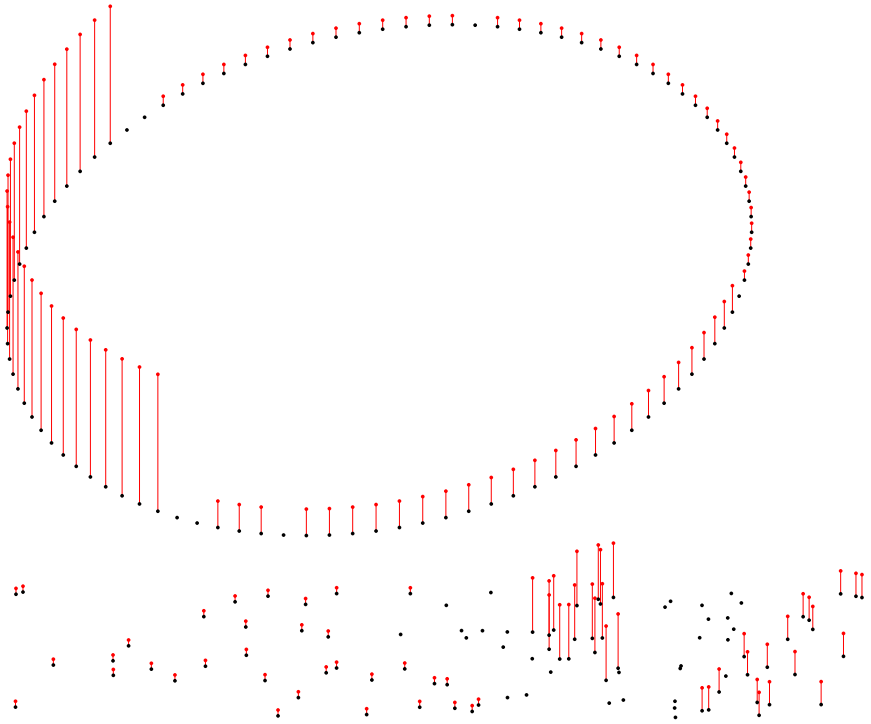


Figure 4. Local smoothness of the signals from Figure 3. The values are shown for nonzero signal samples. The local smoothness in classical signal analysis is related to the instantaneous frequency as $\lambda(n) = 4\sin^2(\omega(n)/2)$.

Other graph shift operators. Finally, notice that in relation (10), we used the graph Laplacian, $\mathbf{L}$, as the shift operator. In addition to the adjacency matrix, $\mathbf{A}$, as another common choice for the shift operation, the normalized version of the adjacency matrix, $(\mathbf{A}/\lambda_{\max})$, normalized graph Laplacian $(\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2})$, or the random walk (also called diffusion) matrix, $(\mathbf{D}^{-1}\mathbf{W})$, may be used as graph shift operators, producing corresponding spectral forms of the systems for graph signals [30].

Remark 1. *The normalized graph Laplacian,*

$$\mathbf{L}_N = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{D}^{-1/2}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$$

is used as a shift operator in the first-order system, to define the convolution operation and the convolution layer in the graph convolutional neural networks (GCNN). Its form is

$$\mathbf{y} = \left(h_0\mathbf{L}_N^0 + h_1\mathbf{L}_N^1\right)\mathbf{x} = (h_0+h_1)\mathbf{x} - h_1\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}\mathbf{x}. \tag{14}$$

Using this relation, the input, $\mathbf{x}_c^{(l-1)}$ and the output, $\mathbf{x}_c^{(l)}$ of the c-th channel of the l-th convolution layer in the GCNN are implemented as

$$\mathbf{x}_c^{(l)} = w_{0c}^{(l-1)}\mathbf{x}_c^{(l-1)} + w_{1c}^{(l-1)}\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}\mathbf{x}_c^{(l-1)}. \tag{15}$$

where the weight $w_{0c}^{(l-1)}$, in the c-th channel of the l-th convolution layer, corresponds to the weight (h_0+h_1) in (14) and $w_{1c}^{(l-1)}$ corresponds to $(-h_1)$ in (14).

4. Spectral Domain Localized Graph Fourier Transform (LGFT)

Classical short-time Fourier transform (STFT) admits time–frequency localization of the analyzed signal using the Fourier transform of the windowed and shifted versions of the signal. This principle is possible in the graph signal processing [29,31]. However, since this approach requires sophisticated approaches to the vertex shift operation on the signals, the spectral domain localization is more commonly used in vertex–frequency analysis. Although, the spectral domain is possible and well-defined in classical analysis, it has been rarely used for time–frequency analysis of signals. The time–frequency localization of a signal in the spectral domain is obtained using a spectral domain localization window, which is shifted in frequency, while the time shift is achieved by the modulation of the windowed Fourier transform of the signal.

We shall use the spectral approach to perform vertex–frequency localization. The graph Fourier transform localized in the spectral domain (LGFT) is defined as an inverse graph Fourier transform of the graph Fourier transform, $X(p)$, multiplied by a spectral domain window, $H(k-p)$. The spectral domain window is nonzero at and around the spectral index k. Therefore, the element-wise LGFT is calculated using

$$S(m,k) = \sum_{p=1}^{N} X(p)H(k-p)\ u_p(m). \tag{16}$$

The shift is here performed in the well-ordered spectral domain, along the spectral index k, instead of the more complex signal shift in the vertex domain. As it will be shown, this form of the vertex–frequency analysis, will also allow vertex localized implementations of the vertex–frequency analysis, even without calculation of the graph Fourier transform of the signal, which is of crucial importance in the case of very large graphs.

Remark 2. *The counterpart of (16) in the classical time–frequency analysis is well-known short-time Fourier transform (STFT) [12]*

$$S(m,k) = \frac{1}{\sqrt{N}} \sum_{p=1}^{N} X(p)H(k-p)e^{j\frac{2\pi}{N}(m-1)(p-1)},$$

where $H(k)$ is a frequency domain localization window.

The LGFT defined in the spectral domain by (16) can be realized by using bandpass transfer functions, denoted by $H_k(\lambda_p) = H(k-p)$. Then the LGFT definition is given by

$$S(m,k) = \sum_{p=1}^{N} X(p)H_k(\lambda_p)\ u_p(m). \tag{17}$$

The transfer function in (17), $H_k(\lambda_p)$, is centered (shifted) at a spectral index, k, by definition. The vertex–frequency domain kernel, $\mathcal{H}_{m,k}(n)$, of the form

$$\mathcal{H}_{m,k}(n) = \sum_{p=1}^{N} H_k(\lambda_p)u_p(m)u_p(n), \tag{18}$$

is obtained from

$$S(m,k) = \sum_{p=1}^{N} \Big(\sum_{n=1}^{N} x(n)\ u_p(n) \Big) H_k(\lambda_p)\ u_p(m) = \sum_{n=1}^{N} x(n)\mathcal{H}_{m,k}(n) = \langle x(n), \mathcal{H}_{m,k}(n) \rangle. \tag{19}$$

Remark 3. *In classical time–frequency analysis the elements of the inverse DFT matrix* $\mathbf{U}$ *are equal to* $u_k(n) = \exp(j2\pi(n-1)(k-1)/N)/\sqrt{N}$ *and* $H_k(\lambda_p) = H_k(e^{j\omega_p})$ *are the bandpass transfer functions, with the kernel*

$$\mathcal{H}_{m,k}(n) = \frac{1}{N}\sum_{p=1}^{N} H_k(e^{j\omega_p})e^{j2\pi(n-1)(p-1)/N}e^{j2\pi(m-1)(k-1)/N}. \tag{20}$$

The STFT is then defined as

$$S(m,k) = \langle x(n), \mathcal{H}_{m,k}(n)\rangle. \tag{21}$$

The matrix form of the vertex–frequency spectrum (17) is

$$\begin{bmatrix} S(1,k) \\ S(2,k) \\ \vdots \\ S(N,k) \end{bmatrix} = \begin{bmatrix} u_1(1) & u_2(1) & \cdots & u_N(1) \\ u_1(2) & u_2(2) & \cdots & u_N(2) \\ \vdots & & & \\ u_1(N) & u_2(N) & \cdots & u_N(N) \end{bmatrix} \begin{bmatrix} H_k(\lambda_1) & 0 & \cdots & 0 \\ 0 & H_k(\lambda_2) & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & H_k(\lambda_N) \end{bmatrix} \begin{bmatrix} X(1) \\ X(2) \\ \vdots \\ X(N) \end{bmatrix}$$

or using vector/matrix notation

$$\mathbf{s}_k = \mathbf{U}H_k(\mathbf{\Lambda})\mathbf{X} = \mathbf{U}H_k(\mathbf{\Lambda})\,\mathbf{U}^T\mathbf{x}, \tag{22}$$

where the column vector whose elements are $S(m,k)$, $m = 1,2,\ldots,N$, is denoted by $\mathbf{s}_k$.

4.1. Binomial Decomposition

Consider the simplest decomposition when the total spectral domain of graph signal is divided into $K = 2$ bands. These two bands, indexed by $k = 0$ and $k = 1$, cover the low-pass part and high-pass part of spectral content of signal, respectively. First we will use the linear functions of eigenvalue λ, to achieve these properties

$$H_0(\lambda_p) = (1 - \frac{\lambda_p}{\lambda_{\max}}), \quad H_1(\lambda_p) = \frac{\lambda_p}{\lambda_{\max}}. \tag{23}$$

Using the relation between (10) and (11) we can conclude that the vertex-domain implementation of this kind of LGFT analysis is very simple

$$\mathbf{s}_0 = (\mathbf{I} - \frac{1}{\lambda_{\max}}\mathbf{L})\,\mathbf{x}, \quad \mathbf{s}_1 = \frac{1}{\lambda_{\max}}\mathbf{L}\,\mathbf{x},$$

and for each vertex, m, the calculation of $S(m,0) = s_0(m)$ and $S(m,1) = s_1(m)$, requires only the combination of the signal at this vertex and its neighboring vertices, to calculate the elements of $\mathbf{L}\,\mathbf{x}$.

Remark 4. *The classical time–frequency analysis counterpart of (23) is obtained using the eigenvalue to frequency relation for the circular undirected graph* $\lambda = 2\sin^2(\omega/2)$ *to produce low-pass and high-pass type transfer functions*

$$H_0(\omega) = 1 - \sin^2(\omega/2) = \cos^2(\omega/2),$$
$$H_1(\omega) = \sin^2(\omega/2),$$

as shown in Figure 5 (top). These spectral transfer functions are dual to the classical Hann (raised cosine) window forms, used for signal localization in the time domain.

To improve the spectral resolution and to divide the spectral range into more than two bands. we can use the same transfer function forms by applying them to the low-pass part of the signal and dividing the spectral content of this part of the signal into its low-pass

part and its high-pass part. In classical signal processing, two common approaches are applied:

(a) The high-pass part is kept unchanged, while the low-pass part is split. This approach corresponds to the wavelet transform or the frequency-varying classical analysis.

(b) The high-pass part is also split into its low-pass and high-pass parts to keep the frequency resolution constant for all frequency bands.

Next, we consider these two approaches for the division of frequency bands.

(a) In a two-scale wavelet-like analysis we keep the high-pass part $\mathbf{s}_1$, while the low-pass part, $\mathbf{s}_0$, is split in its low-pass part, $\mathbf{s}_{00}$ and high-pass part, $\mathbf{s}_{01}$, using the same transfer function, as

$$\mathbf{s}_1 = \frac{1}{\lambda_{\max}}\mathbf{L}\,\mathbf{x},$$
$$\mathbf{s}_{00} = \left(\mathbf{I} - \frac{\mathbf{L}}{\lambda_{\max}}\right)^2\mathbf{x}, \quad \mathbf{s}_{01} = \left(\mathbf{I} - \frac{\mathbf{L}}{\lambda_{\max}}\right)\frac{\mathbf{L}}{\lambda_{\max}}\mathbf{x}.$$

For the third scale step we would keep $\mathbf{s}_1$ and the high-pass part of the scale two step, $\mathbf{s}_{01}$, and then split the low-pass part in scale two, $\mathbf{s}_{00}$, into its low-pass part, $\mathbf{s}_{000}$, and high-pass part, $\mathbf{s}_{001}$, using

$$\mathbf{s}_{000} = \left(\mathbf{I} - \frac{\mathbf{L}}{\lambda_{\max}}\right)^3\mathbf{x}, \quad \mathbf{s}_{001} = \left(\mathbf{I} - \frac{\mathbf{L}}{\lambda_{\max}}\right)^2\frac{\mathbf{L}}{\lambda_{\max}}\mathbf{x}.$$

This process could be continued until the desired scale (frequency resolution) is reached.

(b) For the uniform frequency bands both the low-pass and the high-pass bands are split in the same way, to obtain

$$\mathbf{s}_{00} = \left(\mathbf{I} - \frac{\mathbf{L}}{\lambda_{\max}}\right)^2\mathbf{x}, \quad \mathbf{s}_{01} = 2\left(\mathbf{I} - \frac{\mathbf{L}}{\lambda_{\max}}\right)\frac{\mathbf{L}}{\lambda_{\max}}\mathbf{x}, \quad \mathbf{s}_{11} = \frac{\mathbf{L}^2}{\lambda_{\max}^2}\mathbf{x}. \tag{24}$$

Notice that this kind of spectral band division will produce two times the same result. Once when the original low-pass part is multiplied by the high-pass function, and then again when the original high-pass part is multiplied by the low-pass function. This is the reason why the constant value of 2 has appeared in the new middle pass-band, $\mathbf{s}_{01}$.

The bands in relation (24) can be obtained as the terms of the binomial expression $\left((\mathbf{I} - \mathbf{L}/\lambda_{\max}) + \mathbf{L}/\lambda_{\max}\right)^2\mathbf{x}$. If we continue to the next level, by multiplying all the elements in (24) by the low-pass part, $(\mathbf{I} - \mathbf{L}/\lambda_{\max})$, and then by the high-part part, $\mathbf{L}/\lambda_{\max}$, after grouping the same terms, we would obtain the signal bands of the same form as the terms of the binomial $\left((\mathbf{I} - \mathbf{L}/\lambda_{\max}) + \mathbf{L}/\lambda_{\max}\right)^3\mathbf{x}$. We can conclude that the division can be performed into K bands corresponding to the terms of a binomial form

$$\left((\mathbf{I} - \mathbf{L}/\lambda_{\max}) + \mathbf{L}/\lambda_{\max}\right)^K\mathbf{x}. \tag{25}$$

The transfer function of the k-th, $k = 0, 1, 2, \ldots, K$, term, has the vertex domain form

$$H_k(\mathbf{L}) = \binom{K}{k}\left(\mathbf{I} - \frac{1}{\lambda_{\max}}\mathbf{L}\right)^{K-k}\left(\frac{1}{\lambda_{\max}}\mathbf{L}\right)^k.$$

Of course, the sum of all parts of signal, filtered by $H_k(\mathbf{L})$, produces the reconstruction relation, $\sum_{k=0}^{K} H_k(\mathbf{L})\mathbf{x} = \mathbf{x}$, what is obvious from the identity in (25), that is, from $(\mathbf{I} - \mathbf{L}/\lambda_{\max}) + \mathbf{L}/\lambda_{\max} = \mathbf{I}$.

Example 2. *The spectral domain transfer functions $H_k(\lambda_p)$, $p = 1, 2, \ldots, N$, $k = 0, 1, \ldots, K-1$, which correspond to classical time–frequency processing and the binomial form terms for $K = 2$, $K = 3$, and $K = 26$ are shown in Figure 5. The last two panels (the third and fourth panel) show the case with $K = 26$. In the third panel, the amplitudes of every transfer function is normalized. In the fourth panel, all transfer functions for $K = 26$ are shown, without the amplitude normalization.*

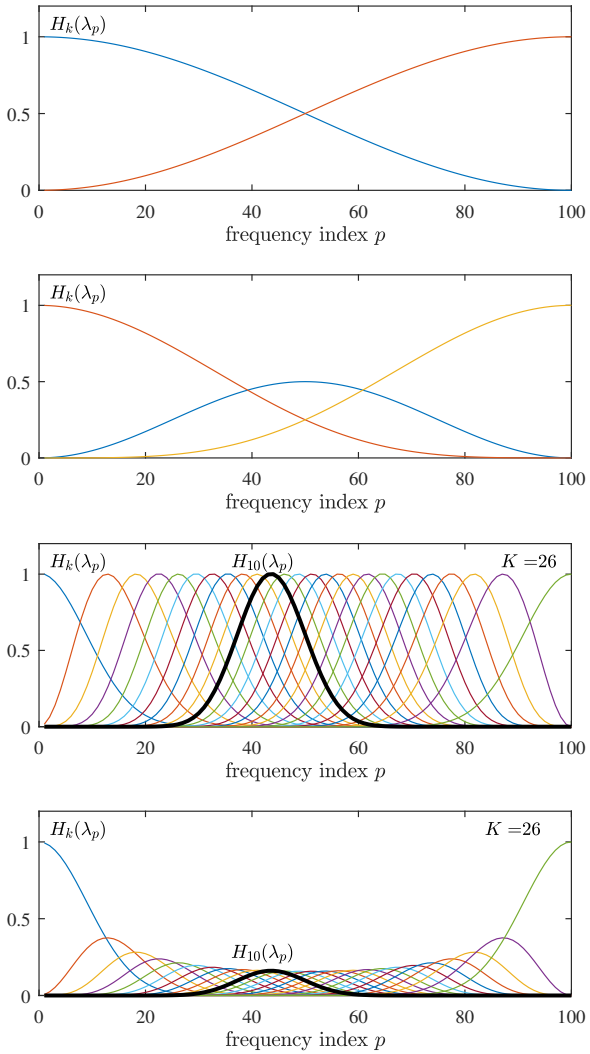


Figure 5. The spectral domain transfer functions $H_k(\lambda_p)$, for a circular undirected and unweighted graph (classical analysis), $p = 1, 2, \ldots, N$, $k = 0, 1, \ldots, K-1$, that correspond to the terms of the binomial form for $K = 26$.

Example 3. *For a general graph, the spectral domain transfer functions $H_k(\lambda_p)$, $p = 1, 2, \ldots, N$, $k = 0, 1, \ldots, K-1$, that can be obtained as the terms of the binomial form for $K = 2$, $K = 3$, and $K = 26$ are shown in Figure 6. The last two panels (the third and fourth panel) again show the case with $K = 26$. In the third panel, the amplitudes of every transfer function is normalized. In the fourth panel, all transfer functions for $K = 26$ are shown, without the amplitude normalization.*

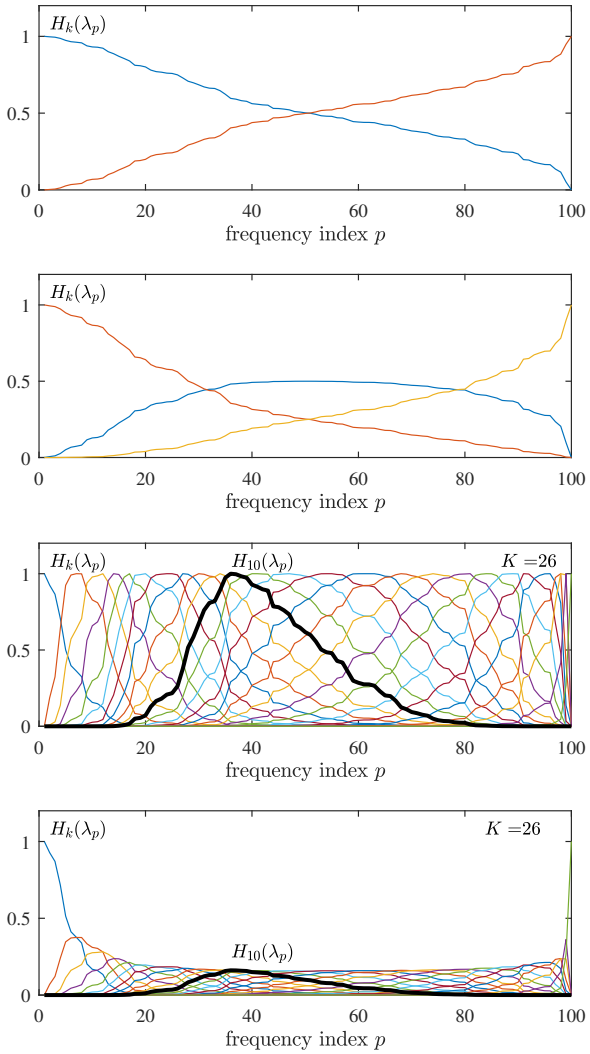


Figure 6. The spectral domain transfer functions $H_k(\lambda_p)$, for a general graph $p = 1, 2, \ldots, N$, $k = 0, 1, \ldots, K-1$, that correspond to the terms of the binomial form for $K = 26$.

Example 4. *Vertex-domain implementation is based on the multiplication of signal,* $\mathbf{x}$*, by the graph Laplacian* $\mathbf{L}$*. For each vertex n it is localized to its neighborhood one. After the signal* $\mathbf{Lx}$ *is calculated, then the new signal* $\mathbf{L}^2\mathbf{x}$ *is easily obtained as the graph Laplacian multiplication with the calculated signal,* $\mathbf{Lx}$*, that is* $\mathbf{L}^2\mathbf{x} = \mathbf{L}\Big(\mathbf{Lx}\Big)$*. This procedure is continued up to the any order,* $\mathbf{L}^k\mathbf{x} = \mathbf{L}\Big(\mathbf{L}^{k-1}\mathbf{x}\Big)$.

In classical time–frequency analysis the multiplication by the graph Laplacian of an undirected circular graph, $\frac{1}{\lambda_{\max}}\mathbf{Lx}$ *with* $\lambda_{\max} = 4$*, is equivalent to the convolution of the signal* $\mathbf{x}$ *with the impulse response of the finite impulse response filter*

$$\mathbf{h} = [-1/4, 1/2, -1/4]^T,$$

that corresponds to the transfer function $H_1(\omega) = \sin^2(\omega/2) = \Big(1 - \cos(\omega)\Big)/2$*. It means that the high-pass and low-pass part of the signal are obtained as (the element-wise form of the Laplacian operator applied to the signal is given by (4)*

$$\mathbf{s}_1 = \mathbf{x} * [-1/4, 1/2, -1/4]^T,$$
$$s_1(k) = -\frac{1}{4}x(k-1) + \frac{1}{2}x(k) - \frac{1}{4}x(k+1),$$

and

$$\mathbf{s}_0 = \mathbf{x} - \mathbf{x} * \mathbf{h} = \mathbf{x} * [1/4, 1/2, 1/4]^T,$$
$$s_0(k) = \frac{1}{4}x(k-1) + \frac{1}{2}x(k) + \frac{1}{4}x(k+1),$$

where $*$ *denotes convolution operation. These convolutions can be repeated to produce wavelet-like band distribution or uniform distribution of frequency bands.*

If no downsampling is used, then the redundant representation of signal is obtained with each of these components containing the same number of samples as the original signal. However, it is possible to form nonredundant form of this representation. Using downsampling with factor of 2*, the values*

$$s_1(2n) = -\frac{1}{4}x(2n-1) + \frac{1}{2}x(2n) - \frac{1}{4}x(2n+1),$$
$$s_0(2n) = \frac{1}{4}x(2n-1) + \frac{1}{2}x(2n) + \frac{1}{4}x(2n+1),$$

are kept. The signal samples at the even indexed instants, $2n$*, are easily obtained as*

$$x(2n) = s_0(2n) + s_1(2n),$$

while for the samples at the odd indexed instants, $2n+1$*, we have*

$$x(2n+1) = 2s_0(2n) - 2s_1(2n) - x(2n-1).$$

Using the initial condition $x(-1)$*, we can reconstruct all odd-indexed samples. This reconstruction can be noise sensitive for large* N*, due to repeated recursions in the last relation.*

Example 5. *Time–frequency and vertex–frequency analysis based on the binomial decomposition of the signals from Example 1 is performed in this example. The corresponding transfer functions for the time–frequency analysis (circular undirected graph, Figure 2 (top)) and vertex–frequency analysis (general graph, Figure 2 (bottom)) are shown in Figures 5 and 6, respectively. The time–frequency representation of the three-harmonic signal from Figure 3 (top) is shown in Figure 7a, (left panel). Its reassigned version to the position of the maximum distribution value is given in Figure 7a (right panel). The same analysis for the general graph signal from Figure 3 (bottom) is shown in Figure 7b. Finally, in order to present the common complex-valued harmonic form, the signal is composed by adding two corresponding sine and cosine components (as in (7)) and forming the complex-valued components* $u_{16}(n) + ju_{17}(n)$*, within* $\mathcal{V}_1$*,* $u_{84}(n) + ju_{85}(n)$*, within* $\mathcal{V}_2$*, and* $u_{28}(n) + ju_{29}(n)$*, within* $\mathcal{V}_3$*. Time–frequency representation of this signal is given in Figure 7c.*

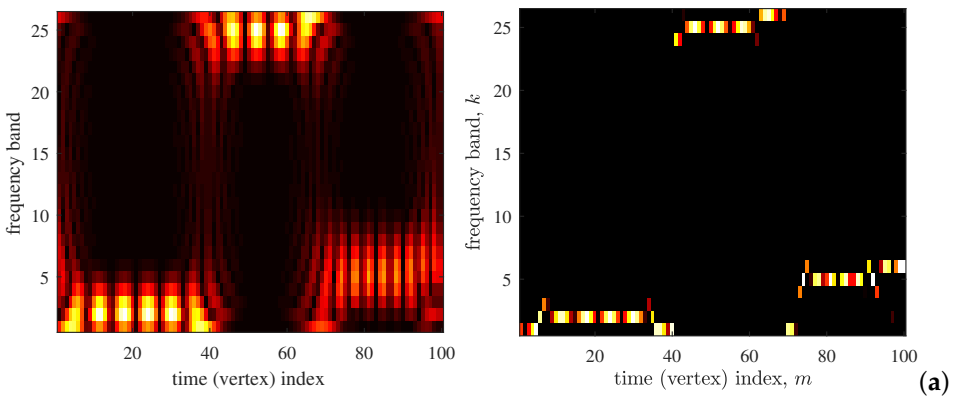


Figure 7. *Cont.*

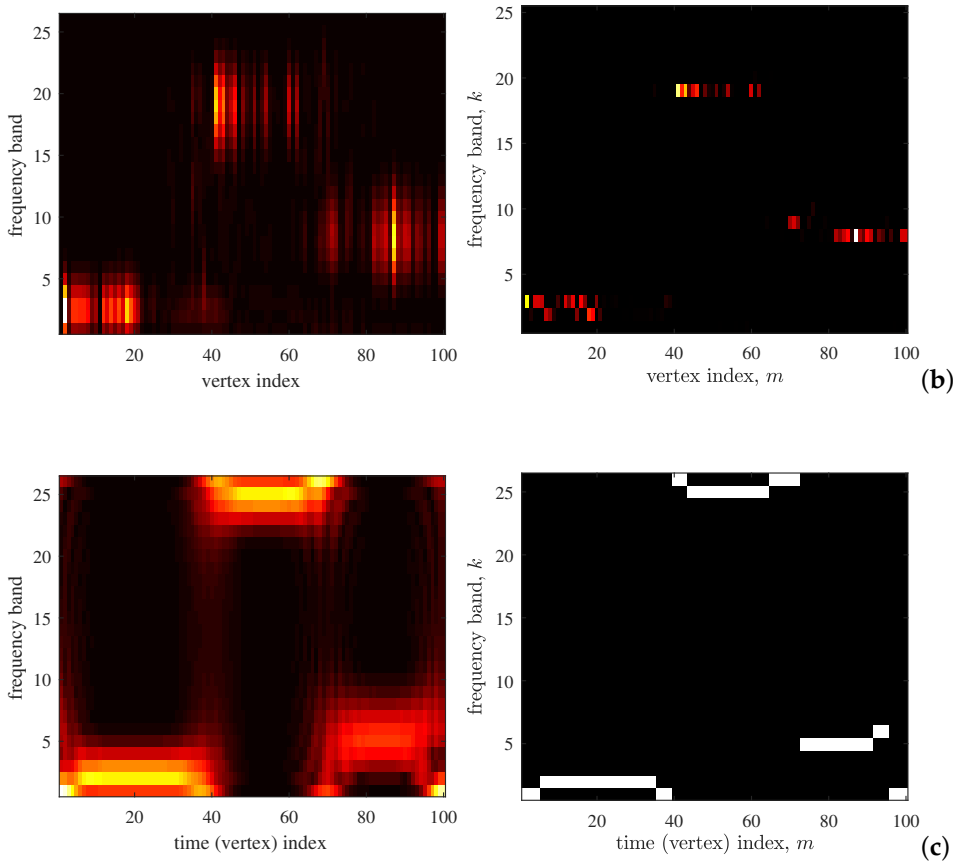


Figure 7. Time–frequency and vertex–frequency representations of the signals from Example 1: (**a**) Time–frequency analysis of the harmonic signal from Example 1, shown in Figure 3 (top), using the transfer functions from Figure 5 (bottom). (**b**) Vertex–frequency analysis of the general graph signal from Example 1, shown in Figure 3 (bottom), using the transfer functions from Figure 6 (bottom). (**c**) Time–frequency analysis of the harmonic complex signal from Example 1, using the transfer functions from Figure 5 (bottom). The complex signal is formed by adding two corresponding sine and cosine components. In all cases, the original representation is given on the left panel, while the reassigned value to the position of the distribution maximum is given on the right panel.

Selectivity of the transfer functions can be improved using higher order polynomials, instead of the linear functions in (23). Assuming that the high-pass part should satisfy $H_1(0) = 0$ and $H_1(\lambda_{\max}) = 1$, and that its derivative is zero at the initial interval point, ($\lambda_p = 0$), and the ending interval point ($\lambda_p = \lambda_{\max}$), as well as that $H_0(\lambda) + H_1(\lambda) = 1$, we can use the following polynomial forms

$$H_0(\lambda_p) = (1 - H_1(\lambda_p)), \quad H_1(\lambda_p) = 3\Big(\frac{\lambda_p}{\lambda_{\max}}\Big)^2 - 2\Big(\frac{\lambda_p}{\lambda_{\max}}\Big)^3. \tag{26}$$

The vertex-domain implementation is performed according to

$$\mathbf{s}_1 = \frac{3}{\lambda_{\max}^2}\mathbf{L}^2\mathbf{x} - \frac{2}{\lambda_{\max}^3}\mathbf{L}^3\mathbf{x}, \quad \mathbf{s}_0 = (\mathbf{x} - \mathbf{s}_1).$$

The same analysis can be now repeated as for (23). These polynomial forms will be revisited later in this paper.

4.2. Hann (Raised Cosine) Window Decomposition

We have presented the simplest decomposition to the low-pass and high-part of a signal. However, the LGFT of the form (17) can be calculated using any other set of bandpass functions, $H_k(\mathbf{\Lambda})$, $k = 0, 1, \ldots, K-1$, as

$$\mathbf{s}_k = H_k(\mathbf{L})\mathbf{x}, \quad k = 0, 1, \ldots, K-1. \tag{27}$$

The spline or raised cosine (Hann window) functions are commonly used as bandpass functions. To further illustrate the concepts, we will consider next transfer functions in general form of the shifted raised cosine functions. They are given by

$$H_k(\lambda) = \begin{cases} \sin^2\left(\frac{\pi}{2}\frac{a_k}{b_k - a_k}\left(\frac{\lambda}{a_k} - 1\right)\right), & \text{for } a_k < \lambda \leq b_k \\ \cos^2\left(\frac{\pi}{2}\frac{b_k}{c_k - b_k}\left(\frac{\lambda}{b_k} - 1\right)\right), & \text{for } b_k < \lambda \leq c_k \\ 0, & \text{elsewhere,} \end{cases} \tag{28}$$

where the spectral bands for $H_k(\mathbf{\Lambda})$ are defined with $(a_k, b_k]$ and $(b_k, c_k]$, $k = 0, 1, \ldots, K-1$. If spectral bands were uniform within $0 \leq \lambda \leq \lambda_{\max}$, the corresponding intervals are based on

$$a_k = a_{k-1} + \frac{\lambda_{\max}}{K-1}, \quad b_k = a_k + \frac{\lambda_{\max}}{K-1}, \quad c_k = a_k + 2\frac{\lambda_{\max}}{K-1}, \tag{29}$$

with $a_1 = 0$ and $\lim_{\lambda \to 0}(a_1/\lambda) = 1$. Here, only $0 = b_0 \leq \lambda \leq c_0 = \lambda_{\max}/K$ is used to define the initial transfer function, $H_0(\lambda)$, while the interval $a_{K-1} < \lambda \leq b_{K-1} = \lambda_{\max}$ in (28) is used for the last transfer function, $H_{K-1}(\lambda)$. The transfer functions with $K = 15$ uniform bands and $\lambda_{\max} = 4$ are shown in Figure 8 (top).

Example 6. *The transfer functions (28) in the eigenvalue (smoothness index) spectral domain, $H_k(\lambda_p)$, $p = 1, 2, \ldots, N$, $k = 0, 1, \ldots, K-1$, and the frequency domain for the classical analysis (graph analysis on circular undirected unweighted graphs) are shown in Figure 8 (top) and (bottom), respectively, for $K = 15$, $0 \leq \lambda \leq \lambda_{\max} = 4$ and $0 \leq \omega \leq \pi$. Notice that the relation between these two domains is nonlinear through $\lambda = 4\sin(\omega^2/2)$.*

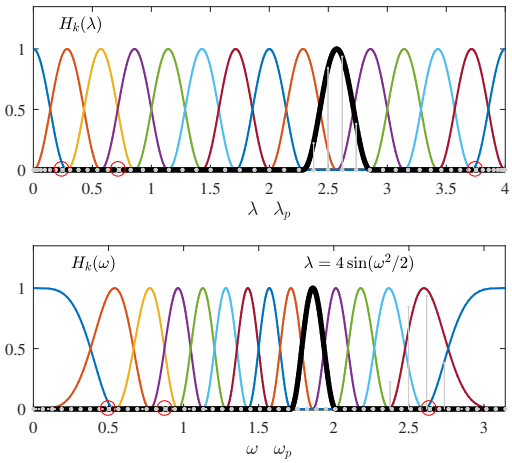


Figure 8. Transfer functions in the spectral eigenvalue and frequency domains for classical analysis: (top) The eigenvalue spectral domain transfer functions $H_k(\lambda_p)$, $p = 1, 2, \ldots, N$, $k = 0, 1, \ldots, K-1$, for $K = 15$ and $0 \leq \lambda \leq \lambda_{\max} = 4$. (down) The frequency spectral domain transfer functions $H_k(\omega_p)$, $p = 1, 2, \ldots, N$, $k = 0, 1, \ldots, K-1$, for $K = 15$ and $0 \leq \omega \leq \pi$. Horizontal axis represents the continuous variable λ and discrete values λ_p, $p = 1, 2, \ldots, N$, corresponding to the eigenvalues and denoted by gray dots along the axis. The same notation is used for frequency ω and its discrete values ω_p that correspond to λ_p.

Example 7. *The transfer function for various widths of the Hann window are shown in Figure 9. The most common case with uniform division of the spectral domain, as defined by (29), is given in Figure 9a. Two forms of the spectral dependent widths are shown in Figure 9b,c. While the widths, defined by the constants a_k in (29) increase as in the wavelet transform case, the widths of the transfer functions in Figure 9c are kept narrow around the spectral indices of signal components, in order to make finer spectral resolution at these regions (signal adaptive approach). Finally, Figure 9d shows polynomial approximations of the transfer functions form Figure 9a, which will be discussed later.*

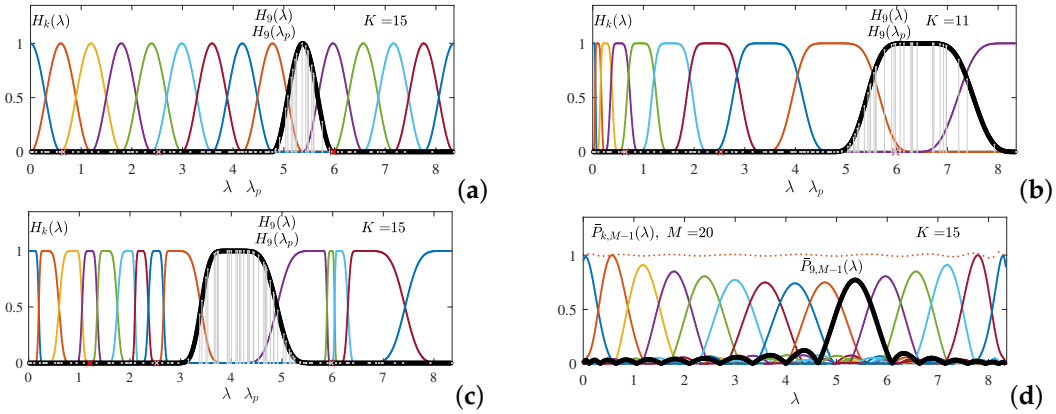


Figure 9. Transfer functions in the spectral domain. (**a**) The transfer functions corresponding to the Hann form terms for $K = 15$. (**b**) The spectral index-varying (wavelet-like) transfer functions whose terms are of half-cosine form, with $K = 11$. (**c**) The spectral domain signal adaptive transfer functions with $K = 17$. (**d**) Approximations of transfer functions from panel (**a**) using Chebyshev polynomials, with $H_9(\lambda)$ being designated by the thick black lines, whereas gray markers indicate the corresponding discrete values.

Example 8. *The transfer functions with various widths of the Hann window form Figure 9 are used for time–frequency representation of the signal on the circular graph from Example 1. The results are shown in Figure 10.*

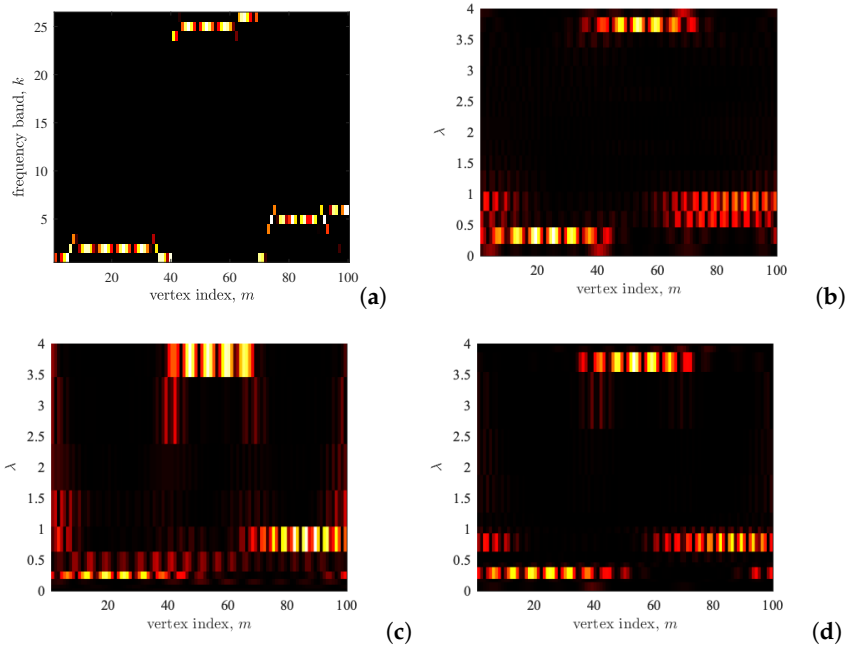


Figure 10. *Cont.*

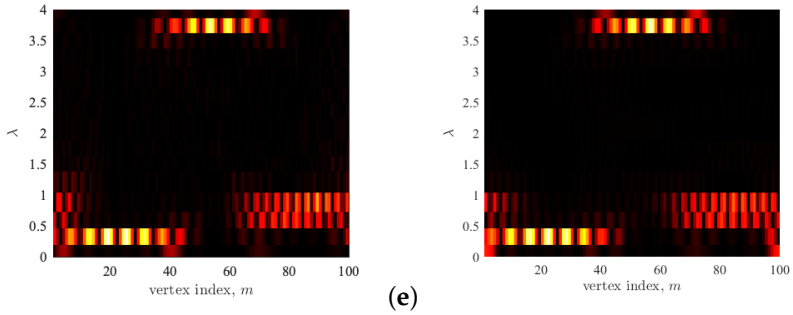


Figure 10. Time–frequency representation of a three-component time-domain signal from Example 1, shown in Figure 3 (top), based on various transfer functions from Figure 5. The LGFT is calculated based on: (**a**) the transfer functions in Figure 5, (**b**) the transfer functions in Figure 9a, (**c**) the wavelet-like spectral transfer functions in Figure 9b, (**d**) the signal adaptive transfer functions from Figure 9b, (**e**) Chebyshev polynomial-based approximations from Figure 9d, with $M = 20$ and (**f**) Chebyshev polynomial-based approximations from Figure 9d, with $M = 50$.

Example 9. *In this example, the same transfer functions form Figure 5 are used for vertex–frequency representation of the signal on the general graph from Example 1. The results are shown in Figure 11.*

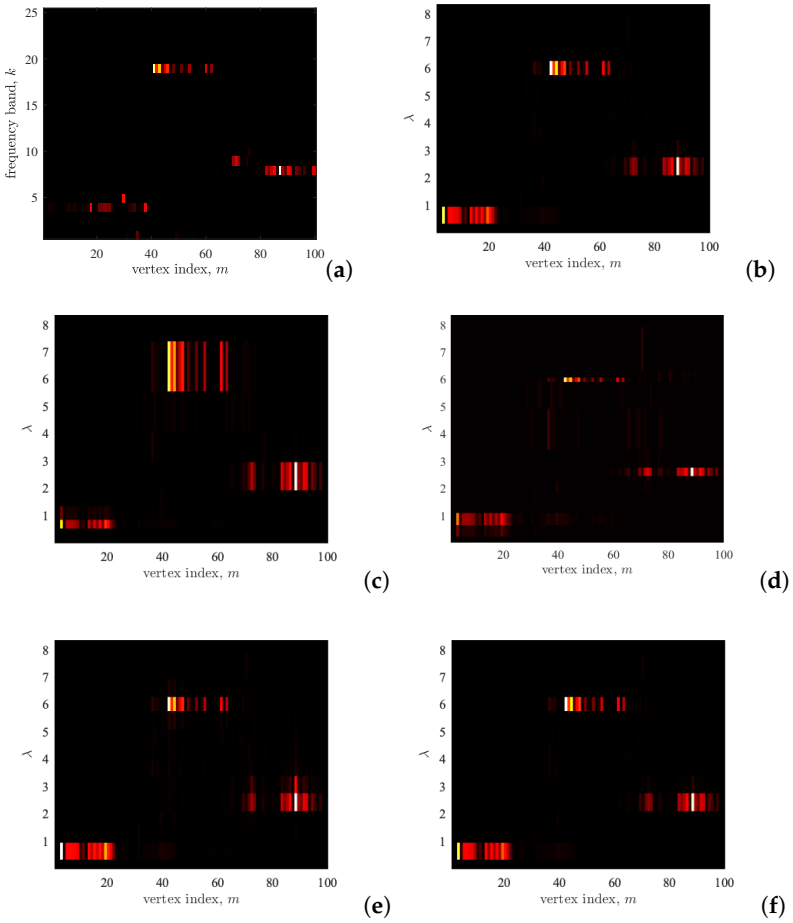


Figure 11. vertex–frequency representation of a three-component general graph signal from Example 1, shown in Figure 3 (bottom). The LGFT is calculated based on: (**a**) the transfer functions in Figure 5, (**b**) the transfer functions in Figure 9a, (**c**) the wavelet-like spectral transfer functions in Figure 9b, (**d**) the signal adaptive transfer functions from Figure 9b, (**e**) Chebyshev polynomial-based approximations from Figure 9d, with $M = 20$ and (**f**) Chebyshev polynomial-based approximations from Figure 9d, with $M = 50$.

4.3. General Window Form Decomposition—OLA condition

The spectral transfer functions in the form of the raised cosine transfer function (28) are characterized by

$$\sum_{k=0}^{K-1} H_k(\lambda_p) = 1. \tag{30}$$

We may use any common window for the decomposition which satisfies this relation. Next, we will list some of these windows:

- A combination of the raised cosine windows. After one set of the raised cosine windows is defined, we may use another set with different constants a_k, b_k, c_k and overlap it with the existing set. If the window values are divided by $1/2$ then the resulting window satisfies (30). In this way, we can increase the number of different **overlapping windows**.
- **Hamming window** can be used in the same way as in (28). The only difference is that the Hamming windows sum-up to 1.08 in the overlapping interval, meaning that the result should be divided by this constant.
- **Bartlett (triangular) window** with the same constants a_k, b_k, c_k as in (28) satisfies the condition (30), along with combinations with different sets of a_k, b_k, c_k to increase overlapping.
- **Tukey window** has a flat part in the middle and the cosine form in the transition interval. It can also be used with appropriately defined a_k, b_k, c_k to take into account the flat (constant) window range.

4.4. Frame Decomposition—WOLA Condition

For the signal reconstruction, using the kernel orthogonality and the frames concept, the windows should satisfy the condition

$$\sum_{k=0}^{K-1} H_k^2(\lambda_p) = 1. \tag{31}$$

The graph signal reconstruction can be performed based on (30) and (31), as discussed with more details in Section 6.

Several windows that satisfy the condition in (31) will be presented next:

- **Sine window** is obtained as the square root of the raised cosine window in (28). Obviously, this window will satisfy (31). Its form is

$$H_k(\lambda) = \begin{cases} \sin\left(\frac{\pi}{2}\frac{a_k}{b_k-a_k}\left(\frac{\lambda}{a_k}-1\right)\right), & \text{for } a_k < \lambda \leq b_k \\ \cos\left(\frac{\pi}{2}\frac{b_k}{c_k-b_k}\left(\frac{\lambda}{b_k}-1\right)\right), & \text{for } b_k < \lambda \leq c_k \\ 0, & \text{elsewhere.} \end{cases} \tag{32}$$

- A window that satisfies (31) can be formed for **any window** in the previous section, by taking its square root.

Example 10. *For the case of the Hann window and the triangular (Bartlett) window, their corresponding squared root forms, that will produce* $\sum_{k=0}^{K-1} H_k^2(\lambda_p) = 1$, *are shown in Figure 12, for a uniform splitting of the spectral domain and a signal-dependent (wavelet-like) form. Notice that the squared root of the Hann window is the sine window form.*
It is obvious that the windows are not differentiable at the ending interval points, meaning that their transforms will be very spread (slow-converging).

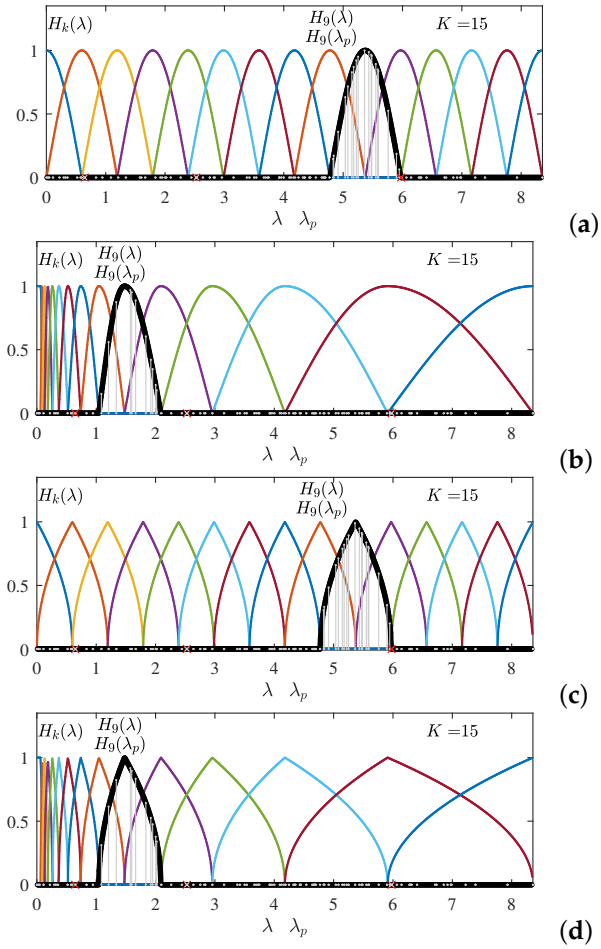


Figure 12. Transfer functions formed using the Hann window (**a**,**b**) and the Bartlett window (**c**,**d**) square root so that the reconstruction condition $\sum_{k=0}^{K-1} H_k^2(\lambda_p) = 1$ is satisfied, for a uniform splitting (**a**,**c**) of the spectral domain and a wavelet-like splitting (**b**,**d**).

The windows defined as square roots of the presented windows (which originally satisfy the OLA condition), do not satisfy **the first derivative continuity property** at the ending interval points. For example, the raised cosine window satisfied that property, but its square root (sine) window loses this desirable property Figure 12. To restore this property, we may either define new windows or just use the same windows, such as the raised cosine window, and change argument so that the window derivative is continuous at the ending point. This technique is used to define the following window form.

- **Mayer's window form** modifies the square root of the raised cosine window (sine window) by adding the function $v_x(x)$ in the argument, x, which will make the first derivative continuous at the ending points. In this case, the window functions become [32]

$$H_k(\lambda) = \begin{cases} \sin\left(\frac{\pi}{2} v_x\left(\frac{a_k}{b_k - a_k}\left(\frac{\lambda}{a_k} - 1\right)\right)\right), & \text{for } a_k < \lambda \leq b_k \\ \cos\left(\frac{\pi}{2} v_x\left(\frac{b_k}{c_k - bk}\left(\frac{\lambda}{b_k} - 1\right)\right)\right), & \text{for } b_k < \lambda \leq c_k \\ 0, \text{ elsewhere,} \end{cases} \tag{33}$$

with $a_{k+1} = b_k$, $b_{k+1} = c_k$, while the initial and the last intervals are defined as in (29). In order to overcome the non-differentiability of the sine and cosine functions at the interval-end points, the previous argument from (28), of the form

$$v_x(x) = x = \frac{a_k}{b_k - a_k}\left(\frac{\lambda}{a_k} - 1\right),$$

is mapped as

$$v_x(x) = x^4(35 - 84x + 70x^2 - 20x^3),$$

for $0 \leq x \leq 1$ with $x = \frac{a_k}{b_k - a_k}\left(\frac{\lambda}{a_k} - 1\right)$, producing Meyer's wavelet-like transfer functions.

If we now check the derivative of a transfer function, $dH_k(\lambda)/d\lambda$, at the ending interval points, we will find that it is zero-valued. This was the reason for introducing the nonlinear (polynomial) argument form instead x or λ, having in mind the relation between the arguments x and λ.

Example 11. *The transfer functions from the previous example, for the case of the Hann window and the triangular (Bartlett) window of forms that will produce* $\sum_{k=0}^{K-1} H_k^2(\lambda_p) = 1$, *whose argument is modified in order to achieve differentiability at the ending points, are shown in Figure 13. Due to differentiability, these transfer functions have a faster convergence than the forms in the previous example, and are appropriate for vertex–frequency and time–frequency analysis. The results of this analysis would be similar to those presented in Figures 10 and 11. The difference exists is in the reconstruction procedure as well.*

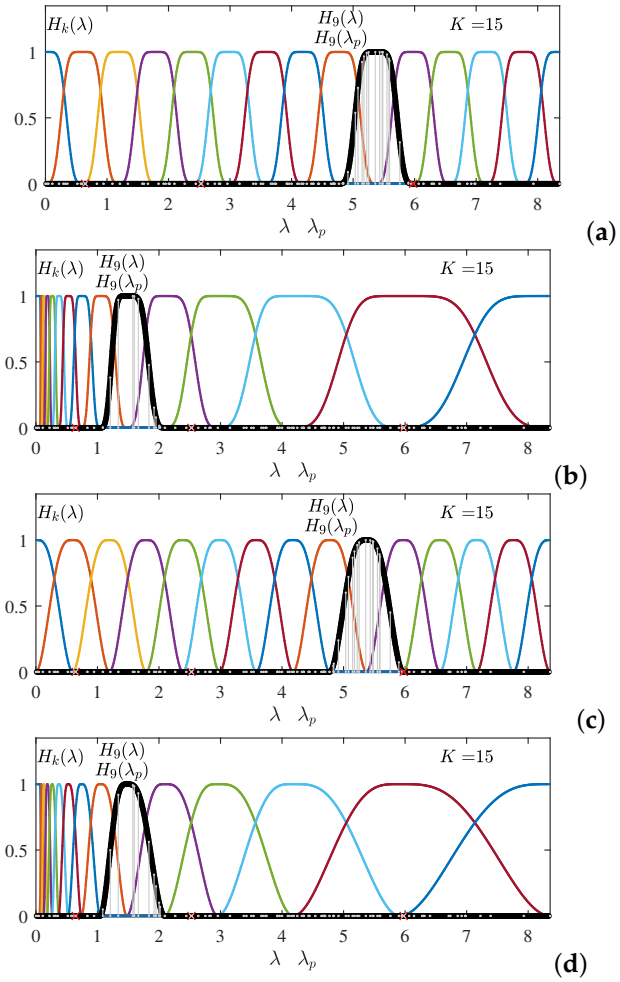


Figure 13. Transfer functions formed using the Hann window (**a**,**b**) and the Bartlett window square root with modified argument (**c**,**d**), using the argument x mapping $v_x(x) = x^4(35 - 84x + 70x^2 - 20x^3)$. The WOLA reconstruction condition $\sum_{k=0}^{K-1} H_k^2(\lambda_p) = 1$ is satisfied.

- **Polynomial windows** are obtained if the function $v_x(x) = x^4(35 - 84x + 70x^2 - 20x^3)$ is applied to the triangular window. Their form is

$$H_k(\lambda) = \begin{cases} v_x\left(\frac{a_k}{b_k - a_k}(\frac{\lambda}{a_k} - 1)\right), \text{ for } a_k < \lambda \leq b_k \\ 1 - v_x\left(\frac{b_k}{c_k - bk}(\frac{\lambda}{b_k} - 1)\right), \text{ for } b_k < \lambda \leq c_k \\ 0, \text{ elsewhere.} \end{cases} \tag{34}$$

The simplest polynomial for that would satisfy the conditions $v_x(0) = 0$, $v_x(1) = 1$, $v'_x(0) = v'_x(1) = 0$ is $v_x(x) = ax^3 + bx^2$ with $a + b = 1$, $3a + 2b = 0$, that is

$$v_x(x) = -2x^3 + 3x^2.$$

In general, the conditions:

$$v_x(0) = 0, \qquad v_x(1) = 1, \qquad \frac{dv_x(x)}{dx}_{|x=0} = 0, \qquad \frac{dv_x(x)}{dx}_{|x=1} = 0,$$

are satisfied by

$$v_x(x) = ax^n + bx^{n-1},$$

for $n \geq 3$, if $na + (n-1)b = 0$ and $a + b = 1$, $a = -(n-1)$, $b = n$. With $n = 5$ we obtain

$$v_x(x) = -4x^5 + 5x^4.$$

These transfer functions are the extension of the linear forms presented in (23) and could be very convenient for the vertex (time) implementation. The polynomial of the third order in λ will require only neighborhood 3 in the vertex (time) domain implementation.

- **Spectral graph wavelet transform.** In the same way as the LGFT can be defined as a *projection of a graph signal onto the corresponding kernel functions*, the spectral graph wavelet transform can be calculated as the projections of the signal onto the wavelet transform kernels. The basic form of the wavelet transfer function in the spectral domain is denoted by $H(\lambda_p)$. Then, the other transfer functions of the wavelet transform are obtained as the scaled versions of the basic function $H(\lambda_p)$ using the scales s_i, $i = 1, 2, \ldots, K-1$. The scaled transform functions are $H_{s_i}(\lambda_p) = H(s_i\lambda_p)$ [21,22,33–36].
The father wavelet is a low-pass scale function denoted by $G(\lambda_p)$. It is a low-pass function, in the same way as in the LGFT was the function $H_0(\lambda_p)$. The set of scales for the calculation of the wavelet transform is $s \in \{s_1, s_2, \ldots, s_{K-1}\}$/ The scaled transform functions obtained in these scales are $H_{s_i}(\lambda_p)$ and $G(\lambda_p)$. Next, the spectral wavelet transform is calculated as a projection of the signal onto the bandpass (and scaled) wavelet kernel, $\psi_{m,s_i}(n)$, in the same way as the kernel $\mathcal{H}_{m,k}(n)$ was used in the LGFT in (18). It means that the wavelet transform elements are

$$\psi_{m,s_i}(n) = \sum_{p=1}^{N} H(s_i\lambda_p)u_p(m)u_p(n), \tag{35}$$

with the wavelet coefficients given by

$$W(m, s_i) = \sum_{n=1}^{N} \psi_{m,s_i}(n)x(n)$$
$$= \sum_{n=1}^{N}\sum_{p=1}^{N} H(s_i\lambda_p)x(n)u_p(m)u_p(n) = \sum_{p=1}^{N} H(s_i\lambda_p)X(p)u_p(m).$$

The Meyer approach to the transfer functions is defined in (33) with the argument $v_x(q(s_i\lambda - 1))$. The same form can be applied to the wavelet transform using $H(s_i\lambda_p)$ and the intervals of the support for this function given by:

- $1/s_1 < \lambda \leq M/s_1$ for $i = 1$ (sine function in (33)).
- $1/s_i < \lambda \leq M/s_i$ (sine functions in (33)), $i = 2, 3, \ldots, K-1$.
- $M/s_i < \lambda \leq M^2/s_i$ (cosine functions in (33)), $i = 2, 3, \ldots, K-1$.
- where the scales are defined by $s_i = s_{i-1}M = M^i/\lambda_{\max}$.
- The interval for the low-pass function, $G(\lambda)$, is $0 \leq \lambda \leq M^2/s_{K-1}$ (cosine function within $M/s_{K-1} < \lambda \leq M^2/s_{K-1}$ and the value $G(\lambda) = 1$ as $\lambda \to 0$).

Notice that the wavelet transform is just a special case of the varying transfer function, when the narrow transfer functions are used for low spectral indices and wide transfer functions are used for high spectral indices, as shown in Figure 13b or Figure 10b,d. In the implementations, we can use the vertex domain localized polynomial approximations of the spectral wavelet functions in the same way as described in Section 5.

- **Optimization of the vertex–frequency representations.** As in classical time–frequency analysis, various measures can be used to compare and optimize joint vertex–frequency representations. An overview of these measures may be found in [37]. Here, we shall suggest the one-norm (in the vector norm sense), introduced to the time–frequency optimization problems in [37], in the form

$$\mathcal{M} = \frac{1}{F}\sum_{m=1}^{N}\sum_{k=0}^{K-1}|S(m,k)| = \frac{1}{F}\|\mathbf{S}\|_1, \tag{36}$$

where $F = \|\mathbf{S}\|_F = \sqrt{\sum_{m=1}^{N}\sum_{k=0}^{K-1}|S(m,k)|^2}$ is the Frobenius norm of matrix $\mathbf{S}$, used for the energy normalization. The normalization factor can be omitted if $S(m,k)$ is a tight frame. Here we will just underline that the functions, $S(m,k)$, are referred to as *a frame*. In the case of a graph signal, $\mathbf{x}$, the set functions $S(m,k)$, is a frame is [22]

$$a||\mathbf{x}||_2^2 \leq \sum_{k=0}^{K-1}\sum_{m=1}^{N}|S(m,k)|^2 \leq b||\mathbf{x}||_2^2,$$

holds , with a and b being positive constants. This constants determine the stability of reconstructing the signal from the values $S(m,k)$. A frame is called *Parseval's tight frame* if $a = b$. The LGFT, as given by in (17), represents Parseval's tight frame when

$$\sum_{k=0}^{K-1}\sum_{m=1}^{N}|S(m,k)|^2 = \sum_{k=0}^{K-1}\sum_{p=1}^{N}|X(p)H_k(\lambda_p)|^2 = E_x = constant. \tag{37}$$

Notice that Parseval's theorem is used for the LGFT, $S(m,k)$, as it is the GFT of the spectral windowed signal, $X(p)H_k(\lambda_p)$. With this fact in mind we obtain

$$\sum_{m=1}^{N}|S(m,k)|^2 = \sum_{p=1}^{N}|X(p)H_k(\lambda_p)|^2.$$

The LGFT defined by (17) is a tight frame if the condition in (31) or (49) holds. This is the condition that is used to define transfer functions shown in Figure 5b,c.

5. Polynomial LGFT Approximation

Let us assume that the spectral domain localization window in the LGFT corresponds to a transfer function of a bandpass graph system, $H_k(\lambda_p)$. In the case of very large graphs, for the vertex domain realization of the LGFT, it is of crucial importance to define or approximate this transfer function by a polynomial

$$H_k(\lambda_p) = h_{0,k} + h_{1,k}\lambda_p + \cdots + h_{M-1,k}\lambda_p^{M-1}, \tag{38}$$

where $k = 0, 1, \ldots, K-1$, K is the number of spectral bands and M is the polynomial order. It is assumed that transfer function $H_k(\lambda_p)$ is centered at an eigenvalue, λ_k, and is of a bandpass type around it (as in (17)). The vector form of the LGFT, $S(m,k)$, defined for the vertex index m and spectral index k by (18) is given as follows

$$\mathbf{s}_k = \mathbf{U}H_k(\mathbf{\Lambda})\mathbf{U}^T\mathbf{x} = H_k(\mathbf{L})\mathbf{x} = \sum_{p=0}^{M-1} h_{p,k}\mathbf{L}^p\,\mathbf{x}, \tag{39}$$

In vector notation, $\mathbf{s}_k$ is a column vector, whose elements are equal to $S(m,k)$, $m = 1, 2, \ldots, N$. The property of the eigendecomposition of a power matrix is used to obtain this result. *The number of shifted transfer functions, K, does not depend on the number of indices N. The realization of the LGFT is based on the linear combination of the graph signal shifts* $\mathbf{L}^p\,\mathbf{x}$, *and does not require any graph Fourier transform or other operation on the entire graph.*

For this reason the bandpass LGFT functions, $H_k(\lambda)$, $k = 0, 1, \ldots, K-1$, in the form given by (28) or (33) should be realized using their approximations by polynomials whose order is $(M-1)$. Although the approximation based on the Chebyshev polynomial is most commonly used for this purpose [25,31], we will revisit alternative approaches [38] as well.

5.1. Chebyshev Polynomial

The transfer functions in the graph relations, denoted by $H_k(\lambda)$, are defined at discrete set of eigenvalues $\lambda = \lambda_p$. The polynomial approximation is obtained by using a function that is continuous. Its argument is within the range $0 \le \lambda \le \lambda_{\max}$. Then the optimal choice for the polynomial approximation type are the so-called "min–max" Chebyshev polynomials. They have the property that the maximal possible deviation from the desired function is minimal. This property is of crucial importance since we approximate the transfer functions in continuous λ, which will be used as transfer functions at a discrete set of eigenvalues λ_p of the LGFT.

The Chebyshev polynomials are defined by

$$T_0(z) = 1, \;\; T_1(z) = z, \ldots, T_m(z) = 2zT_{m-1}(z) - T_{m-1}(z),$$

for $m \ge 2$ and $-1 \le z \le 1$.

The mapping $\bar{T}_m(\lambda) = T_m(2\lambda/\lambda_{\max} - 1)$ is introduced to transform the argument from $0 \le \lambda \le \lambda_{\max}$ to -1 to 1. Then, the Chebyshev polynomials of the finite $(M-1)$-order can be written as follows

$$\bar{P}_{k,M-1}(\lambda) = \frac{c_{k,0}}{2} + \sum_{m=1}^{M-1} c_{k,m}\bar{T}_m(\lambda), \tag{40}$$

where the polynomial coefficients are calculated using the Chebyshev polynomial inversion property as

$$c_{k,m} = \frac{2}{\pi}\int_{-1}^{1} H_k((z+1)\lambda_{\max}/2)T_m(z)\frac{dz}{\sqrt{1-z^2}}.$$

Based on the previous definitions, the vertex-domain implementation (39) of the spectral LGFT form, can be now written as follows

$$\mathbf{s}_k = \bar{P}_{k,M-1}(\mathbf{L})\mathbf{x}, \quad k = 0, 1, 2, \ldots, K-1,$$

with

$$\begin{aligned} \bar{P}_{k,M-1}(\mathbf{L}) &= \frac{c_{k,0}}{2} + \sum_{m=1}^{M-1} c_{k,m}\bar{T}_m(\mathbf{L}), \\ &= h_{0,k}\mathbf{I} + h_{1,k}\mathbf{L} + h_{2,k}\mathbf{L}^2 + \cdots + h_{(M-1),k}\mathbf{L}^{M-1}. \end{aligned} \tag{41}$$

In the calculation of the polynomial form of the transfer functions in (41), the $(M-1)$—neighborhood is only used to obtain the LGFT for every vertex, n. This form does not employ the eigendecomposition analysis over the whole graph, in any way. Therefore, the computational complexity for large graphs is feasible.

Example 12. *The Chebyshev polynomial approximation approach will be illustrated on a set of the transfer functions, $H_k(\lambda)$, defined in (28) and (29). For $K = 15$, these transfer function are shown in Figure 9a. The transfer functions $H_k(\lambda)$ satisfy OLA condition, $\sum_{k=0}^{K-1} H_k(\lambda) = 1$. We used the Chebyshev polynomial, $\bar{P}_{k,M-1}, k = 0, 1, \ldots, K-1$, given by (40), to approximate each individual transfer function, $H_k(\lambda)$. Two polynomial orders are considered for the approximation, $M = 20$, $M = 50$. The resulting Chebyshev polynomial approximations of the transfer functions shown in Figure 9a are given in Figure 9d when the polynomial order was $M = 20$. In order to show the compliance of the obtained approximation with the imposed OLA condition, the value of $\sum_{k=0}^{K-1} \bar{P}_{k,M-1}(\lambda)$ is presented in the figure. This summation value is depicted by the dotted line in Figure 9d. As it can be seen, these values are close to unity. This means that the signal reconstruction from the LGFT calculated using the presented polynomial approximation will be stable and accurate.*

The Chebyshev polynomial approximations of $H_k(\lambda)$, calculated in the way presented here, are applied to obtain the vertex–frequency analysis the signal from Example 1, using the LGFT. Time–frequency representations of the harmonic signal from Example 1 are shown for both polynomial orders, for $M = 20$ in Figure 10e and for $M = 50$ in Figure 10f. As it can be seen from Figure 10d, the representations with a lower order polynomial approximation, when $M = 20$, is less concentrated than representation in Figure 10e obtained for $M = 50$. However, using higher orders, $(M-1)$, of the polynomial approximation increases calculation complexity, since wider neighborhoods are required in the LGFT calculation. The experiment is repeated for the graph signal from Example 1. The two considered sets of Chebyshev polynomial-based approximations of bandpass transfer functions $H_k(\lambda)$, $k = 0, 1, \ldots, K-1$, from Figure 9a are now used in the calculation of vertex–frequency representations from Figure 11e,f, for $M = 20$ and $M = 40$, respectively.

Example 13. *In order to present the Chebyshev polynomial approximation in more detail, and give the exact values of the approximation coefficients we further reduced the approximation order to $(M-1) = 5$. Then we used this order to calculate the approximations of the bandpass functions, $H_k(\lambda)$, for every k in the case of the raised cosine form, given in (28), with $K = 10$ bands. The resulting approximation coefficients, $h_{i,k}$, which are used in the vertex-domain implementation, defined by (39), are shown in Table 1.*

Table 1. Coefficients, $h_{p,k}$, $p = 0, 1, \ldots, M-1$, $k = 0, 1, \ldots, K-1$, for the polynomial calculation of the LGFT, $\mathbf{s}_k$, of a signal $\mathbf{x}$, in various spectral bands, k, for $(M-1) = 5$ and $K = 10$.

$\mathbf{s}_k = (h_{0,k}\mathbf{I} + h_{1,k}\mathbf{L} + h_{2,k}\mathbf{L}^2 + h_{3,k}\mathbf{L}^3 + h_{4,k}\mathbf{L}^4 + h_{5,k}\mathbf{L}^5)\mathbf{x}$						
k	$h_{0,k}$	$h_{1,k}$	$h_{2,k}$	$h_{3,k}$	$h_{4,k}$	$h_{5,k}$
0	1.079	−1.867	1.101	−0.2885	0.03458	−0.001548
1	−0.053	1.983	−1.798	0.5744	−0.07722	0.003723
2	−0.134	0.763	−0.310	0.0222	0.00422	−0.000460
3	0.050	−0.608	0.900	−0.3551	0.05348	−0.002762
4	0.096	−0.726	0.768	−0.2475	0.03172	−0.001424
5	0.016	−0.013	−0.128	0.1047	−0.02231	0.001424
6	−0.073	0.616	−0.779	0.3228	−0.05135	0.002762
7	−0.051	0.351	−0.356	0.1146	−0.01323	0.000460
8	0.084	−0.687	0.871	−0.3751	0.06409	−0.003723
9	−0.021	0.183	−0.251	0.1172	−0.02196	0.001419

5.2. Least Squares Approximation

Bandpass transfer functions $H_k(\lambda)$, used in the calculation of vertex–frequency (time–frequency) representations, can be approximated using polynomial

$$P^{LS}_{k,M-1}(\lambda) = \bar{\alpha}_{0,k} + \bar{\alpha}_{1,k}\lambda + \cdots + \bar{\alpha}_{M-1,k}\lambda^{M-1},$$

such that squared error

$$\int_0^{\lambda_{\max}} |H_k(\lambda) - P^{LS}_{k,M-1}(\lambda)|^2 d\lambda$$

is minimized. This approximation will be referred to as *Least Squares (LS) approximation*. As in the case of Chebyshev approximation, the interval $0 \leq \lambda \leq \lambda_{\max}$ is normalized to $[-1, 1]$, to ensure the standard calculation procedure. This is achieved using the substitution $z = (2\lambda - \lambda_{\max})/\lambda_{\max}$. Upon introducing the following variables

$$s_m = \int_{-1}^{1} z^m dz\,, m = 0, 1, \ldots, 2M-2,$$

and

$$b_m = \int_{-1}^{1} z^m H_k((z+1)\lambda_{\max}/2) dz,$$

$m = 0, 1, \ldots, M-1$, we obtain

$$\begin{bmatrix} s_0 & s_1 & \ldots & s_{M-1} \\ s_1 & s_2 & \ldots & s_M \\ \vdots & \vdots & \ldots & \vdots \\ s_{M-1} & s_M & \ldots & s_{2M-2} \end{bmatrix} \begin{bmatrix} \alpha_{0,k} \\ \alpha_{1,k} \\ \vdots \\ \alpha_{M-1,k} \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{M-1} \end{bmatrix}.$$

The matrix form of the previous relation is $\mathbf{Sa} = \mathbf{b}$. When this linear system of equations is solved, the approximation coefficients $\alpha_{0,k}, \alpha_{1,k}, \ldots, \alpha_{M-1,k}$ are obtained. With $\lambda = 0.5(z+1)\lambda_{\max}$ we further have

$$\sum_{m=0}^{M-1} \alpha_{m,k} z^m = \sum_{m=0}^{M-1} \bar{\alpha}_{i,k} \lambda^m.$$

The approximation is then

$$P^{LS}_{k,M-1}(\mathbf{L}) = \bar{\alpha}_{0,k}\mathbf{I} + \bar{\alpha}_{1,k}\mathbf{L} + \bar{\alpha}_{2,k}\mathbf{L}^2 + \cdots + \bar{\alpha}_{(M-1),k}\mathbf{L}^{M-1}. \quad (42)$$

The vertex-domain implementation (39) of the spectral LGFT form, based on this approximation is performed according to

$$\mathbf{s}_k = \bar{P}^{LS}_{k,M-1}(\mathbf{L})\mathbf{x}, \text{ for every } k = 0, 1, \ldots, K-1.$$

5.3. Legendre Polynomial

The *least squares approximation using Legendre polynomials* assumes minimization of

$$\int_0^{\lambda_{\max}} |H_k(\lambda) - P^{Leg}_{k,M-1}(\lambda)|^2 d\lambda,$$

where

$$P^{Leg}_{k,M-1}(\lambda) = \bar{\beta}_{0,k}\phi_0(\lambda) + \bar{\beta}_{1,k}\phi_1(\lambda) + \cdots + \bar{\beta}_{M-1,k}\phi_{M-1}(\lambda).$$

Polynomials

$$\phi_0(z) = 1,\ \phi_1(z) = z,\ \phi_2(z) = z^2 - 1/3, \ldots$$

are referred to as Legendre polynomials. These polynomials satisfy the so-called Bonnet's recursive relation

$$(m+1)\phi_{m+1}(z) = (2m+1)z\phi_m(z) - m\phi_{m-1}(z). \tag{43}$$

This case also assumes the normalization and shift of interval $0 \leq \lambda \leq \lambda_{\max}$, to achieve the mapping to $[-1, 1]$. This is performed with $z = (2\lambda - \lambda_{\max})/\lambda_{\max}$ to obtain

$$\bar{\phi}_m(\lambda) = \phi_m(2\lambda/\lambda_{\max} - 1).$$

For each $m = 0, 1, \ldots, M-1$ coefficients of the form

$$C_m = \int_{-1}^{1} \bar{\phi}^2_m(z) dz,$$

are calculated, and are further used to obtain the polynomial coefficients of the form

$$\beta_m = \frac{1}{C_m} \int_{-1}^{1} H_k((z+1)\lambda_{\max}/2)\bar{\phi}_m(z) dz.$$

As with $\lambda = 0.5(z+1)\lambda_{\max}$, relation

$$\sum_{m=0}^{M-1} \beta_{m,k}\phi_m(z) = \sum_{m=0}^{M-1} \bar{\beta}_{i,k}\lambda^m,$$

holds, in analogy with previous cases, we obtain the following approximation

$$P^{Leg}_{k,M-1}(\mathbf{L}) = \bar{\beta}_{0,k}\mathbf{I} + \bar{\beta}_{1,k}\mathbf{L} + \bar{\beta}_{2,k}\mathbf{L}^2 + \cdots + \bar{\beta}_{(M-1),k}\mathbf{L}^{M-1}, \tag{44}$$

serving as a basis for the implementation of vertex–frequency analysis using the spectral LGFT form in (39) according to

$$\mathbf{s}_k = \bar{P}^{Leg}_{k,M-1}(\mathbf{L})\mathbf{x}, \text{ for every } k = 0, 1, \ldots, K-1.$$

Example 14. *The Legendre polynomial approximation will be illustrated on the transfer functions, $H_k(\lambda)$, defined by (28) and (29) for every $k = 0, 1, \ldots, K-1$ with $K = 15$. Recall that functions $H_k(\lambda)$ satisfy $\sum_{k=0}^{K-1} H_k(\lambda) = 1$. In this example, we consider approximations of these functions using LS approximation (42), as well as the approximations based on Legendre polynomial (44). For convenience, we also consider the approximation based on Chebyshev polynomial.*

To illustrate how the polynomial order influences the convergence of the approximations, we consider three orders of polynomials: $M = 12$, $M = 20$, and $M = 40$. Shifted spectral transfer functions

$H_k(\lambda)$, $k = 0, 1, \ldots, K-1$, *which are being approximated, are shown in Figures 14a, 15a, and 16a. The approximations based on the Chebyshev polynomial are shown in Figures 14b, 15b, and 16b, for the considered polynomial orders. The approximations using the Legendre polynomial are shown in Figures 14c, 15c, and 16c, while the LS approximations are shown in Figures 14d, 15d and 16d, also for the three considered polynomial orders. It can be seen that even with* $M = 12$, *the Chebyshev and LS based approximations are sufficiently narrow to enable clear distinction between the various spectral bands. The approximations using Legendre polynomials are shown to be less convenient for this purpose.*

The fact that the order of polynomials as low as $M = 12$ *can be used in the calculation of time–frequency representations and vertex–frequency representations is indicated in Figure 17. Polynomial approximations from Figure 14b–d are used in the calculation of time–frequency representations for the harmonic signal from Example 1. As shown in Figure 17a,c,e, signal components are clearly distinguishable in representations obtained based on approximated band functions. The experiment was repeated for the graph signal from Example 1, and the obtained representations are presented in Figure 17b,d,f, for the spectral bandpass functions approximated using the considered polynomials: Chebyshev, Legendre, and LS. We can conclude that when a higher polynomial order M is used in the approximation, it increases the LGFT calculation complexity since it uses a wider neighborhood of the considered vertex.*

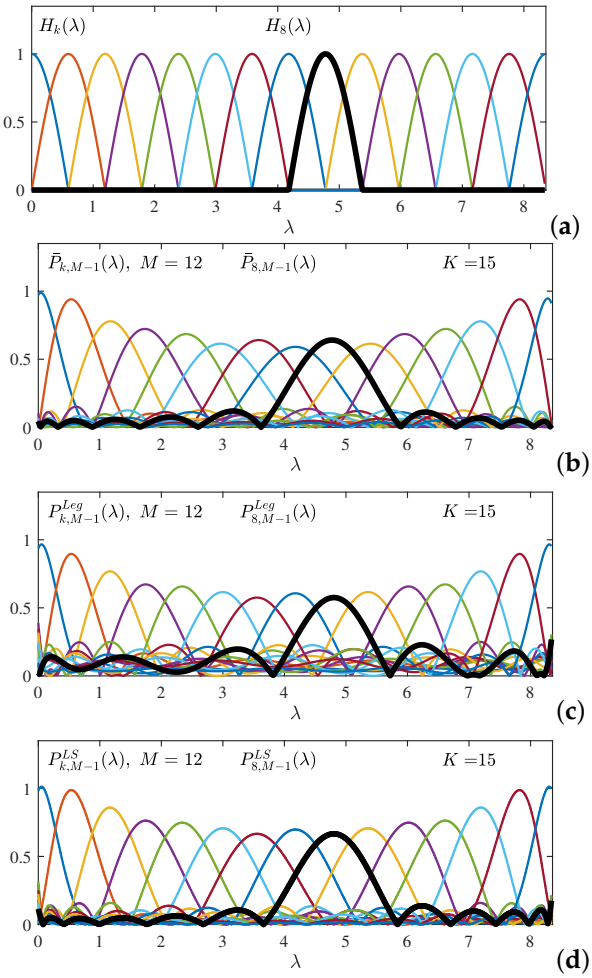


Figure 14. Spectral bandpass transfer functions used for calculation of the LGFT and their polynomial approximations. (**a**) Spectral functions of the Hann form with $K = 15$. (**b**) Approximation of spectral transfer functions based on Chebyshev polynomial with $M = 12$. (**c**) Legendre-polynomial-based approximations of spectral transfer functions with $M = 12$. (**d**) Least squares approximation of spectral transfer functions with $M = 12$. For convenience, function $H_8(\lambda)$ is designated with a thick black line on each panel.

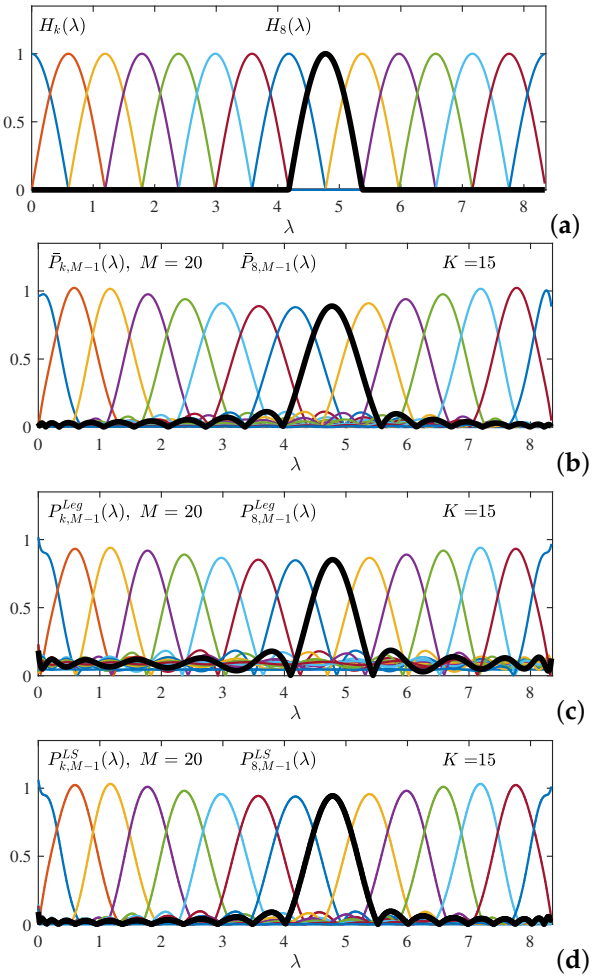


Figure 15. Spectral bandpass transfer functions used for calculation of the LGFT and their polynomial approximations. (**a**) Spectral functions of the Hann form with $K = 15$. (**b**) Approximation of spectral transfer functions based on Chebyshev polynomial with $M = 20$. (**c**) Legendre-polynomial-based approximations of spectral transfer functions with $M = 20$. (**d**) Least squares approximation of spectral transfer functions with $M = 20$. For convenience, function $H_8(\lambda)$ is designated with a thick black line on each panel.

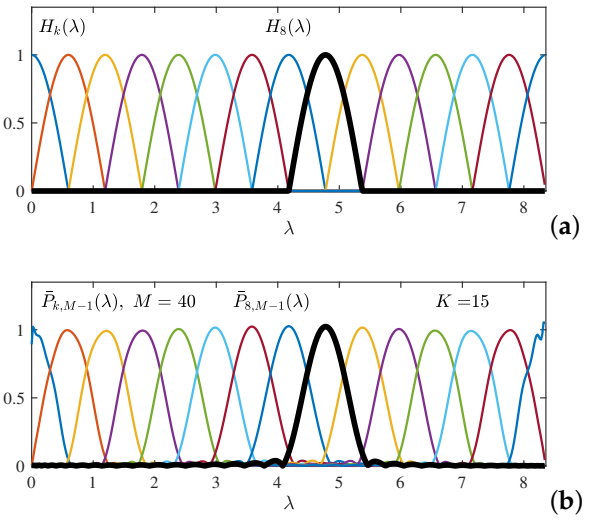


Figure 16. *Cont.*

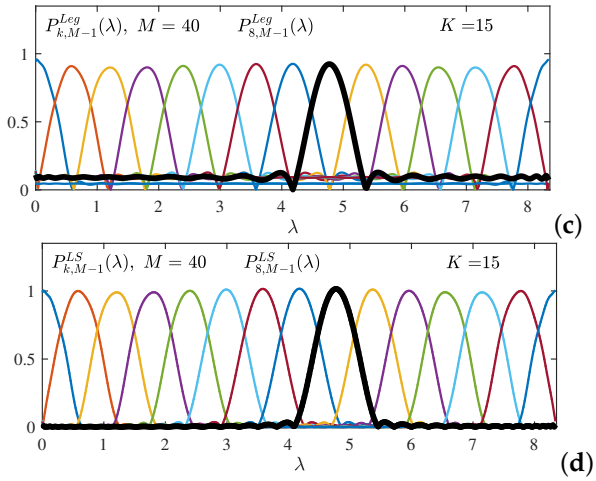


Figure 16. Spectral bandpass transfer functions used for calculation of the LGFT and their polynomial approximations. (**a**) Spectral functions of the Hann form with $K = 15$. (**b**) Approximation of spectral transfer functions based on Chebyshev polynomial with $M = 40$. (**c**) Legendre-polynomial-based approximations of spectral transfer functions with $M = 40$. (**d**) Least squares approximation of spectral transfer functions with $M = 40$. For convenience, function $H_8(\lambda)$ is designated with a thick black line on each panel.

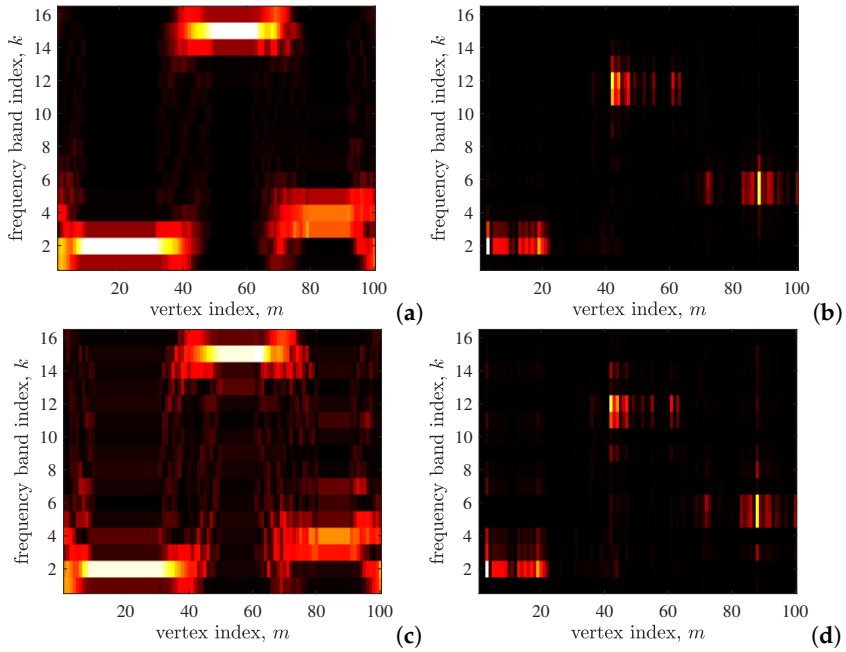


Figure 17. *Cont.*

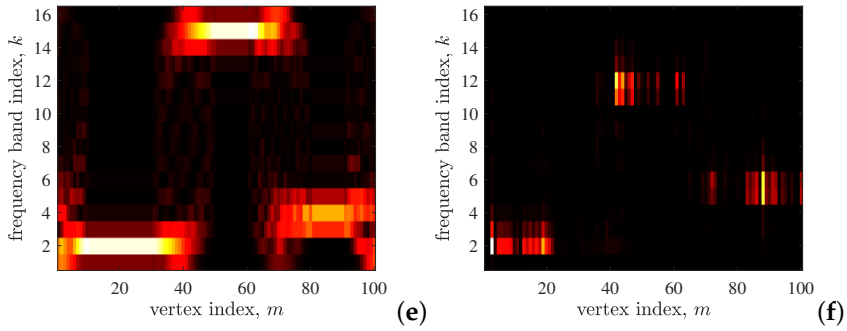


Figure 17. (**a**) Time–frequency analysis of the harmonic signal from Example 1, shown in Figure 3 (top), using the polynomial approximations of transfer functions from Figure 14b. (**b**) Vertex–frequency analysis of the general graph signal from Example 1, shown in Figure 3 (bottom), using the transfer functions from Figure 14b. (**c**) Time–frequency analysis of the harmonic complex signal from Example 1, using the transfer functions from Figure 14c. (**d**) Vertex–frequency analysis of the general graph signal from Example 1, using the transfer functions from Figure 14c. (**e**) Time–frequency analysis of the harmonic complex signal from Example 1, using the transfer functions from Figure 14d. (**f**) Vertex–frequency analysis of the general graph signal from Example 1, using the transfer functions from Figure 14d. A complex signal is formed by adding two corresponding sine and cosine components.

6. Inversion of the LGFT

Two approaches to the inversion of the classical STFT are used. One is based on the summation of the STFT values (overlap-and-add approach) and the other uses the weighted STFT values for the reconstruction (weighted overlap-and-add approach). These two approaches will be used in the vertex–frequency analysis as well.

6.1. Inversion by Summation (OLA Method)

For the LGFT, defined by (27) or in a polynomial form by (39), as

$$\mathbf{s}_k = H_k(\mathbf{L})\mathbf{x}, \quad \text{or} \quad \mathbf{s}_k = \sum_{p=0}^{M-1} h_{p,k}\mathbf{L}^p\mathbf{x}, \tag{45}$$

the signal can be reconstructed by a summation over all spectral shifts

$$\sum_{k=0}^{K-1} \mathbf{s}_k = \sum_{k=0}^{K-1} \left(\sum_{p=0}^{M-1} h_{p,k}\mathbf{L}^p\mathbf{x} \right) = \sum_{k=0}^{K-1} H_k(\mathbf{L})\mathbf{x} = \mathbf{x}. \tag{46}$$

This relation holds when the OLA condition $\sum_{k=0}^{K-1} H_k(\mathbf{L}) = \mathbf{I}$ holds.
The spectral-domain form of this condition is given by

$$\sum_{k=0}^{K-1} H_k(\mathbf{\Lambda}) = \mathbf{I}$$

since we may write $\sum_{k=0}^{K-1} H_k(\mathbf{L}) = \mathbf{U}\sum_{k=0}^{K-1} H_k(\mathbf{\Lambda})\mathbf{U}^T = \mathbf{I}$ and the fact that $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ holds for a symmetric matrix $\mathbf{L}$. This condition used when the transfer functions in Figure 5a are defined.

The element-wise form of the inversion relation (46) is

$$x(n) = \sum_{k=0}^{K-1} S(n,k). \tag{47}$$

6.2. Kernel-Based LGFT Inversion (WOLA Method—Frames)

Another common approach to the inversion in classical time–frequency analysis and wavelet transforms is based on the Gabor (frames) expansions [12]. When applied to the vertex–frequency analysis, it means that the signal is recovered from the LGFT, $S(m,k)$, $S(m,k)$, by projected back the LGFT values to the vertex–frequency kernels, $\mathcal{H}_{m,k}(n)$. It can be considered as the WOLA reconstruction.

If the LGFT calculated by employing the spectral shifted transfer functions, defined by (17) and (18), the Gabor framework based inversion is obtained as

$$x(n) = \sum_{m=1}^{N} \sum_{k=0}^{K-1} S(m,k)\mathcal{H}_{m,k}(n), \tag{48}$$

if the following condition

$$\sum_{k=0}^{K-1} H_k^2(\lambda_p) = 1, \tag{49}$$

is satisfied for all λ_p, $p = 1, 2, \ldots, N$.

The formula for the signal reconstruction (inversion formula) in (48) follows, when the condition in (49) is satisfied. This conclusion is obtained from the analysis

$$\sum_{m=1}^{N} \sum_{k=0}^{K-1} S(m,k)\mathcal{H}_{m,k}(n) = \sum_{m=1}^{N} \sum_{k=0}^{K-1} \sum_{p=1}^{N} X(p)H_k(\lambda_p)u_p(m) \sum_{l=1}^{N} H_k(\lambda_l)u_l(m)u_l(n).$$

Having in mind that the eigenvalues are orthonormal, $\sum_{m=1}^{N} u_p(m)u_l(m) = \delta(p-l)$, we obtain the graph signal, $x(n)$, from

$$\sum_{k=0}^{K-1} \sum_{p=1}^{N} X(p)H_k(\lambda_p)H_k(\lambda_p)u_p(n) = x(n). \tag{50}$$

The condition is that the transfer functions, $H_k(\lambda_p)$, satisfy the WOLA condition in (49) for every λ_p.

The inversion formula for the spectral form of the wavelet transform is just a special case of the varying LGFT analysis. It follows from (48), with the WOLA condition given by (49), taking into account the wavelet transform notation of the transfer functions

$$G^2(\lambda_p) + \sum_{i=1}^{K-1} H^2(s_i\lambda_p) = 1, \tag{51}$$

The set of discrete scales, denoted by $s \in \{s_1, s_2, \ldots, s_{K-1}\}$, is used for the wavelet transform calculation. The corresponding spectral transfer functions are $H(s_i\lambda_p)$, $i = 1, 2, \ldots, K-1$. The father wavelet (low-pass scale function) is denoted by $G(\lambda_p)$. It plays the role of the low-pass function in the LGFT, $H_0(\lambda_p)$.

Vertex-Varying Filtering. For the vertex-varying filtering of the graph signals using the vertex–frequency representation we can use a support function $B(m,k)$ in the vertex–frequency domain. Then, the signal that is filtered in the vertex–frequency using the LGFT is obtained as $S_f(m,k) = S(m,k)B(m,k)$.

The signal that is filtered by the support function $B(m,k)$ and denoted by $x_f(n)$, is obtained using the inversion of $S_f(m,k)$. In the inversion we can use either OLA or WOLA inversion method depending on the condition satisfied by the transfer functions.

The simplest way of the filtering support function, $B(m,k)$, follows by hard-thresholding the noisy values of the calculated vertex–frequency representation, $S(m,k)$.

7. Support Uncertainty Principle in the LGFT

In the classical time–frequency analysis, the window function is used to localize signal in the joint time–frequency domain. As it is known, the uncertainty principle prevents an ideal simultaneous localization in the time and the frequency domains.

The uncertainty principle is defined in various forms. For a survey see [39–42]. Concentration measures, reviewed in [37], are closely related to all forms of uncertainty principle. The product of the effective signal widths in the time and the frequency domain is the basis for the common uncertainty principle form used in the time–frequency signal analysis [13,43]. In quantum mechanics, this form is also known as the Robertson–Schrödinger inequality. In general signal theory, the most commonly used form is the support uncertainty principle, closely related to the sparsity support measure [37,42]. In the classical signal analysis, the support uncertainty principle relates the discrete signal $\mathbf{x}$, and its DFT, $\mathbf{X}$, as follows

$$\frac{1}{4}\Big(\|\mathbf{x}\|_0 + \|\mathbf{X}\|_0\Big)^2 \geq \|\mathbf{x}\|_0\|\mathbf{X}\|_0 \geq N. \tag{52}$$

In other words, the product of the number of nonzero signal values, $\|\mathbf{x}\|_0$, and the number of its nonzero DFT coefficients, $\|\mathbf{X}\|_0$, is greater or equal than the total number of signal samples N. This form of uncertainty principle will be generalized to the LGFT. The LGFT form shall produce the classical support uncertainty principle as a special case.

The support uncertainty principle in the LGFT can be derived in the same way as in the case of the graph Fourier transform. A simple derivation procedure, as in [44,45], will be followed. If a function

$$F_p(n,k) = X(k)H_p(\lambda_k)S(n,p)u_k(n), \tag{53}$$

is formed, then its sum over indices n and k,

$$\sum_n\sum_k F_p(n,k) = \sum_n\sum_k X(k)H_p(\lambda_k)S(n,p)u_k(n) = \sum_n S^2(n,p) = \sum_k X^2(k)H_p^2(\lambda_k) = E_p, \tag{54}$$

is equal to the energy of the LGFT, $S(n,p)$, for the given frequency band indexed by p. Assume next that the number of nonzero values of $S(n,p)$, for a given p, is equal to $||\mathbf{s}_p||_0$, while the number of nonzero values of the spectral localized signal $X(k)H_p(\lambda_k)$ is $||\mathbf{X}\mathbf{H}_p||_0$. Using the Schwartz inequality

$$E_p^2 = \Big(\sum_n\sum_k X(k)H_p(\lambda_k)S(n,p)u_k(n)\Big)^2 \leq \sum_n\sum_k \Big(X(k)H_p(\lambda_k)(S(n,p)\Big)^2 \sum_n\sum_k u_k^2(n), \tag{55}$$

we obtain

$$1 \leq \sum_n\sum_k u_k^2(n) \leq ||\mathbf{s}_p||_0 \;\; ||\mathbf{X}\mathbf{H}_p||_0 \;\; \mu^2, \tag{56}$$

since $\sum_n\sum_k \Big(X(k)H_p(\lambda_k)(S(n,p)\Big)^2 = \sum_k \Big(X(k)H_p(\lambda_k)\Big)^2 \sum_n S^2(n,p) = E_p^2$, and

$$\mu = \max_{n,k}\{|u_k(n)|\},$$

while the summation in $\sum_n\sum_k u_k^2(n)$ is performed over nonzero values of $X(k)H_p(\lambda_k)S(n,p)$, meaning that $\sum_n\sum_k u_k^2(n) \leq ||\mathbf{s}_p||_0 \;\; ||\mathbf{X}\mathbf{H}_p||_0 \;\; \mu^2$.

Therefore, the support uncertainty principle assumes the following form

$$\frac{1}{4}\Big(||\mathbf{s}_p||_0 + ||\mathbf{X}\mathbf{H}_p||_0\Big)^2 \geq ||\mathbf{s}_p||_0 \;\; ||\mathbf{X}\mathbf{H}_p||_0 \geq \frac{1}{\mu^2}. \tag{57}$$

The first inequality is written based on the general property that the arithmetic mean is greater than geometric mean of positive numbers.

For the standard DFT, with $\mu = |u_k(n)| = 1/\sqrt{N}$ and $H_k(\lambda_p) = 1$, when $S(n,p) = x(n)$, we easily obtain the classical uncertainty principle (52).

Having in mind that $H_p(\lambda_k)$ is band-limited to Q_p nonzero samples, then

$$||\mathbf{X}\mathbf{H}_p||_0 \leq Q_p,$$

meaning that the support of the LGFT satisfies

$$||\mathbf{s}_p||_0 \geq \frac{1}{Q_p \mu^2}. \tag{58}$$

The smallest possible number of nonzero samples in the LGFT is defined by

$$||\mathbf{s}_p||_0 \geq \frac{1}{Q\mu^2}, \tag{59}$$

where $Q = \max_p\{Q_p\}$.

For example, if the classical Fourier analysis is considered then the maximal absolute eigenvector value is $\mu = \max_{n,k}\{|u_k(n)|\} = \max_{n,k}\{|\exp(j2\pi nk/N)/\sqrt{N}|\} = \sqrt{N}$ and

$$||\mathbf{s}_p||_0 \geq N/Q.$$

If we select just one spectral frequency by a bandpass filter, with $Q = 1$, then the duration of the $S(n, p)$ must be N. If half of the spectral band is selected by the bandpass function, $Q = N/2$, then $||\mathbf{s}_p||_0 \geq 2$. Finally, if all spectral components are used, then a delta pulse is possible in the time domain, that is, for $Q = N$, we can have $||\mathbf{s}_p||_0 \geq 1$.

8. Analysis Based on Splitting Large Signals

The graph analysis suggested a polynomial approximation of the transfer functions and the implementation of vertex–frequency analysis using powers of the Laplacian applied to signal. This means that the neighborhood of the considered sample, defined by the power of the Laplacian, is used. In classical signal analysis, this problem was approached by windowing a large signal and then by splitting the analysis into smaller nonoverlapping or overlapping time segments. This idea will be now generalized to graph signals, which may then be used to define more general forms of signal splitting in the classical analysis.

Assume that the signal, $x(n)$, and its domain of N vertices (with even N), is split into two nonoverlapping segments (subsets) with $N/2$ vertices in each subdomain. Without loss of generality assume that the vertex indices for the first half of the signal samples are from 1 to $N/2$ and that the vertices for the second half of the signal samples are indexed with remaining values, $N/2+1$ to N. Then we can write

$$\begin{gathered}\mathbf{X}_0 = \mathbf{U}^{-1}[x(1),\, x(2),\, \ldots, x(N/2), 0,\, 0, \ldots, 0]^T = \mathbf{U}^{-1}\mathbf{x}_0,\\ \mathbf{X}_1 = \mathbf{U}^{-1}[0,\, 0,\, \ldots, 0,\, x(N/2+1),\, x(N/2+2), \ldots, x(N)]^T = \mathbf{U}^{-1}\mathbf{x}_1,\\ \mathbf{X} = \mathbf{X}_0 + \mathbf{X}_1.\end{gathered}$$

For undirected graphs $\mathbf{U}^{-1} = \mathbf{U}^T$ holds. From the inverse GFT, $\mathbf{x}_0 = \mathbf{U}\mathbf{X}_0$ and $\mathbf{x}_1 = \mathbf{U}\mathbf{X}_1$, having in mind the positions of the zero values in $\mathbf{x}_0$ and $\mathbf{x}_1$, we obtain

$$\begin{aligned}\mathbf{0} &= \mathbf{U}_{Lo}\mathbf{X}_0,\\ \mathbf{0} &= \mathbf{U}_{Up}\mathbf{X}_1,\end{aligned}$$

where $\mathbf{U}_{Lo}$ and $\mathbf{U}_{Up}$ are the lower and upper parts of the matrix $\mathbf{U}$. They consist of the rows of $\mathbf{U}$ corresponding to the zero-value positions of the signal $\mathbf{x}_0$ and $\mathbf{x}_1$, respectively.

Splitting now the transform vector $\mathbf{X}_0$ in its even-indexed part, $(\mathbf{X}_0)_{Even}$, and odd-indexed part, $(\mathbf{X}_0)_{Odd}$, and doing the same for the transform vector $\mathbf{X}_1$, we obtain

$$\begin{aligned} \mathbf{0} &= \mathbf{U}_{Lo,Even}(\mathbf{X}_0)_{Even} + \mathbf{U}_{Lo,Odd}(\mathbf{X}_0)_{Odd}, \\ \mathbf{0} &= \mathbf{U}_{Up,Even}(\mathbf{X}_1)_{Even} + \mathbf{U}_{Up,Odd}(\mathbf{X}_1)_{Odd}. \end{aligned} \tag{60}$$

This means that there is no need to calculate the full GFT. It is sufficient to calculate the GFT of order $N/2$, and its values $(\mathbf{X}_0)_{Even}$ and $(\mathbf{X}_1)_{Even}$ or $(\mathbf{X}_0)_{Odd}$ and $(\mathbf{X}_1)_{Odd}$. The remaining parts of the transform vectors could be obtained from (60). This can reduce the problem dimensionality.

The algorithm steps for this approach to the GFT calculation are as follows:

- Calculate even-indexed elements of $\mathbf{X}_0$ and $\mathbf{X}_1$, corresponding to two halves of the signal samples, as

$$\begin{aligned} (\mathbf{X}_0)_{Even} &= (\mathbf{U}^{-1})_{Up,Even}[x(1),\ x(2),\ \ldots, x(N/2)]^T = (\mathbf{U}^{-1})_{Up,Even}\mathbf{x}_{Up}, \\ (\mathbf{X}_1)_{Even} &= (\mathbf{U}^{-1})_{Lo,Even}[x(N/2+1),\ x(N/2+2), \ldots, x(N)]^T = (\mathbf{U}^{-1})_{Lo,Even}\mathbf{x}_{Lo}. \end{aligned}$$

- Find the odd-indexed elements of the GFT using (60) as

$$\begin{aligned} (\mathbf{X}_0)_{Odd} &= (\mathbf{U}_{Lo,Odd})^{-1}\mathbf{U}_{Lo,Even}(\mathbf{X}_0)_{Even}, \\ (\mathbf{X}_1)_{Odd} &= (\mathbf{U}_{Up,Odd})^{-1}\mathbf{U}_{Up,Even}(\mathbf{X}_1)_{Even}. \end{aligned}$$

- Reconstruct the GFT elements of the whole signal,

$$\begin{aligned} (\mathbf{X})_{Even} &= (\mathbf{X}_0)_{Even} + (\mathbf{X}_1)_{Even} \\ (\mathbf{X})_{Odd} &= (\mathbf{X}_0)_{Odd} + (\mathbf{X}_1)_{Odd}. \end{aligned}$$

  Notice that all matrices used in this relation are of size $N/2 \times N/2$, while all vectors are of size $N/2 \times 1$.

This approach can be applied to different splitting schemes (for example, we can split the signal into even and odd indexed samples, and then split the transform elements into upper and lower part). The same procedure can be used for splitting the signal into not equal sets of samples. The case with the overlapping windowed signal can easily be split into nonoverlapping problems. For example, if the window in the classical domain overlaps for half of the window width, then the problem can be separated into two sets of nonoverlapping windows since every other window does not overlap [46–51].

9. Conclusions

Time–frequency analysis is a basis for extending the classical concepts to the vertex-varying spectral analysis of signals on graphs. Attention has been paid to linear signal transformations as the most important forms in classical signal analysis and graph signal processing. The spectral domain of these representations has been considered in detail since it provides an opportunity for a direct generalization of the well-developed time–frequency approaches to vertex–frequency analysis. Various polynomial forms are used in the implementation since they can be computationally very efficient in the case of very large graphs. The polynomial forms are developed in detail in graph signal processing, and can then be used in classical time–frequency analysis, with their simplicity being attractive for the implementation in the case of large time-domain signals. Reconstruction of the graph signals from the vertex–frequency representation has been reviewed, with some practical notes on the filtering, optimal parameter selection, uncertainty principle, and schemes for large signal division into smaller parts. All results are illustrated by numerous numerical examples.

Author Contributions: Conceptualization, L.S. and M.D.; methodology, L.S., J.L., D.M., M.B., C.R. and M.D.; software, L.S., M.B. and M.D.; validation, L.S., J.L., D.M., M.B., C.R. and M.D.; formal analysis, L.S., J.L., D.M., C.R. and. M.D.; investigation, L.S., J.L., D.M., M.B. and M.D.; resources, D.M., M.B. and C.R.; data curation, L.S., J.L., D.M, M.B. and M.D.; writing—original draft preparation, L.S., J.L., D.M., M.B. and M.D.; writing—review and editing, L.S., J.L., D.M., M.B., C.R. and M.D.; visualization, L.S., D.M., M.B. and M.D.; supervision, L.S., J.L., C.R. and M.D.; project administration, L.S. and J.L; funding acquisition, L.S., J.L. and M.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Croatian Science Foundation under the project IP-2018-01-3739, IRI2 project "ABsistemDCiCloud" (KK.01.2.1.02.0179), and the University of Rijeka under the projects uniri-tehnic-18-17 and uniri-tehnic-18-15.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sandryhaila, A.; Moura, J.M. Discrete signal processing on graphs. *IEEE Trans. Signal Process.* **2013**, *61*, 1644–1656. [CrossRef]
2. Chen, S.; Varma, R.; Sandryhaila, A.; Kovačević, J. Discrete Signal Processing on Graphs: Sampling Theory. *IEEE Trans. Signal Process.* **2015**, *63*, 6510–6523. [CrossRef]
3. Sandryhaila, A.; Moura, J.M. Discrete Signal Processing on Graphs: Frequency Analysis. *IEEE Trans. Signal Process.* **2014**, *62*, 3042–3054. [CrossRef]
4. Ortega, A.; Frossard, P.; Kovačević, J.; Moura, J.M.; Vandergheynst, P. Graph signal processing: Overview, challenges, and applications. *Proc. IEEE* **2018**, *106*, 808–828. [CrossRef]
5. Djuric, P.; Richard, C. (Eds.) *Cooperative and Graph Signal Processing: Principles and Applications*; Academic Press: Cambridge, MA, USA, 2018.
6. Hamon, R.; Borgnat, P.; Flandrin, P.; Robardet, C. Extraction of temporal network structures from graph-based signals. *IEEE Trans. Signal Inf. Process. Netw.* **2016**, *2*, 215–226. [CrossRef]
7. Marques, A.; Ribeiro, A.; Segarra, S. Graph Signal Processing: Fundamentals and Applications to Diffusion Processes. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017.
8. Quinn, C.J.; Kiyavash, N.; Coleman, T.P. Directed information graphs. *IEEE Trans. Inf. Theory* **2015**, *61*, 6887–6909. [CrossRef]
9. Raginsky, M.; Jafarpour, S.; Harmany, Z.T.; Marcia, R.F.; Willett, R.M.; Calderbank, R. Performance bounds for expander-based compressed sensing in Poisson noise. *IEEE Trans. Signal Process.* **2011**, *59*, 4139–4153. [CrossRef]
10. Hamon, R.; Borgnat, P.; Flandrin, P.; Robardet, C. Transformation from Graphs to Signals and Back. In *Vertex-Frequency Analysis of Graph Signals*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 111–139.
11. Sandryhaila, A.; Moura, J.M. Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure. *IEEE Signal Process. Mag.* **2014**, *31*, 80–90. [CrossRef]
12. Stanković, L.; Daković, M.; Thayaparan, T. *Time-Frequency Signal Analysis with Applications*; Artech House: London, UK, 2014.
13. Cohen, L. *Time-Frequency Analysis*; Prentice Hall PTR: Englewood Cliffs, NJ, USA, 1995.
14. Boashash, B. *Time-Frequency Signal Analysis and Processing: A Comprehensive Reference*; Academic Press: Cambridge, MA, USA, 2015.
15. Shuman, D.I.; Ricaud, B.; Vandergheynst, P. Vertex-frequency analysis on graphs. *Appl. Comput. Harmon. Anal.* **2016**, *40*, 260–291. [CrossRef]
16. Shuman, D.I.; Ricaud, B.; Vandergheynst, P. A windowed graph Fourier transform. In Proceedings of the IEEE Statistical Signal Processing Workshop (SSP), Ann Arbor, MI, USA, 5–8 August, 2012; pp. 133–136.
17. Zheng, X.W.; Tang, Y.Y.; Zhou, J.T.; Yuan, H.L.; Wang, Y.L.; Yang, L.N.; Pan, J.J. Multi-windowed graph Fourier frames. In Proceedings of the IEEE International Conference on Machine Learning and Cybernetics (ICMLC), Ningbo, China, 9–12 July, 2016; Volume 2, pp. 1042–1048.
18. Tepper, M.; Sapiro, G. A short-graph Fourier transform via personalized pagerank vectors. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March, 2016; pp. 4806–4810.
19. Stanković, L.; Daković, M.; Sejdić, E. Vertex-Frequency Analysis: A Way to Localize Graph Spectral Components [Lecture Notes]. *IEEE Signal Process. Mag.* **2017**, *34*, 176–182. [CrossRef]
20. Cioacă, T.; Dumitrescu, B.; Stupariu, M.S. Graph-Based Wavelet Multiresolution Modeling of Multivariate Terrain Data. In *Vertex-Frequency Analysis of Graph Signals*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 479–507.
21. Hammond, D.K.; Vandergheynst, P.; Gribonval, R. The Spectral Graph Wavelet Transform: Fundamental Theory and Fast Computation. In *Vertex-Frequency Analysis of Graph Signals*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 141–175.

22. Behjat, H.; Van De Ville, D. Spectral Design of Signal-Adapted Tight Frames on Graphs. In *Vertex-Frequency Analysis of Graph Signals*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 177–206.
23. Shuman, D.I.; Narang, S.K.; Frossard, P.; Ortega, A.; Vandergheynst, P. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Process. Mag.* **2013**, *30*, 83–98. [CrossRef]
24. Stanković, L.; Mandic, D.; Daković, M.; Brajović, M.; Scalzo, B.; Li, S.; Constantinides, A.G. Data Analytics on Graphs Part I: Graphs and Spectra on Graphs. *Found. Trends Mach. Learn.* **2020**, *13*, 1–157. [CrossRef]
25. Stanković, L.; Mandic, D.; Daković, M.; Brajović, M.; Scalzo, B.; Li, S.; Constantinides, A.G. Data Analytics on Graphs Part II: Signals on Graphs. *Found. Trends Mach. Learn.* **2020**, *13*, 158–331. [CrossRef]
26. Stanković, L.; Mandic, D.; Daković, M.; Brajović, M.; Scalzo, B.; Li, S.; Constantinides, A.G. Data Analytics on Graphs Part III, Machine Learning on Graphs, from Graph Topology to Applications. *Found. Trends Mach. Learn.* **2020**, *13*, 332–530. [CrossRef]
27. Gray, R.M. *Toeplitz and Circulant Matrices: A Review*; NOW Publishers: Delft, The Netherlands, 2006.
28. Dakovic, M.; Stankovic, L.J.; Sejdic, E. Local Smoothness of Graph Signals. *Math. Probl. Eng.* **2019**, *2019*, 3208569. [CrossRef]
29. Hammond, D.K.; Vandergheynst, P.; Gribonval, R. Wavelets on graphs via spectral graph theory. *Appl. Comput. Harmon.* **2011**, *30*, 129–150. [CrossRef]
30. Stanković, L.; Mandic, D.; Daković, M.; Kisil, I.; Sejdić, E.; Constantinides, A.G. Understanding the Basis of Graph Signal Processing via an Intuitive Example-Driven Approach. *IEEE Signal Process. Mag.* **2019**, *36*, 133–145. [CrossRef]
31. Stanković, L.; Mandic, D.; Daković, M.; Scalzo, B.; Brajović, M.; Sejdić, E.; Constantinides, A.G. Vertex-frequency graph signal processing: A comprehensive review. *Digit. Signal Process.* **2020**, *107*, 102802. [CrossRef]
32. Leonardi, N.; Van De Ville, D. Tight wavelet frames on multislice graphs. *IEEE Trans. Signal Process.* **2013**, *61*, 3357–3367. [CrossRef]
33. Behjat, H.; Leonardi, N.; Sörnmo, L.; Van De Ville, D. Anatomically-adapted Graph Wavelets for Improved Group-level fMRI Activation Mapping. *NeuroImage* **2015**, *123*, 185–199. [CrossRef] [PubMed]
34. Rustamov, R.; Guibas, L.J. Wavelets on graphs via deep learning. In Proceedings of the 27th Annual Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–8 December 2013; pp. 998–1006.
35. Jestrović, I.; Coyle, J.L.; Sejdić, E. A fast algorithm for vertex-frequency representations of signals on graphs. *Signal Process.* **2017**, *131*, 483–491. [CrossRef]
36. Masoumi, M.; Rezaei, M.; Hamza, A.B. Shape Analysis of Carpal Bones Using Spectral Graph Wavelets. In *Vertex-Frequency Analysis of Graph Signals*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 419–436.
37. Stanković, L. A measure of some time-frequency distributions concentration. *Signal Process.* **2001**, *81*, 621–631. [CrossRef]
38. Brajović, M; Stanković, L.; Daković, M. On Polynomial Approximations of Spectral Windows in Vertex-Frequency Representations. In Proceedings of the 24th International Conference on Information Technology, Žabljak, Montenegro, 18–22 February 2020.
39. Pasdeloup, B.; Gripon, V.; Alami, R.; Rabbat, M.G. Uncertainty principle on graphs. In *Vertex-Frequency Analysis of Graph Signals*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 317–340.
40. Perraudin, N.; Ricaud, B.; Shuman, D.I.; Vandergheynst, P. Global and local uncertainty principles for signals on graphs. *APSIPA Trans. Signal Inf. Process.* **2018**, *7*, 1–26. [CrossRef]
41. Tsitsvero, M.; Barbarossa, S.; Di Lorenzo, P. Signals on graphs: Uncertainty principle and sampling. *IEEE Trans. Signal Process.* **2016**, *64*, 539–554. [CrossRef]
42. Ricaud, B.; Tottesani, B. A survey of uncertainty principles and some signal processing applications. *Adv. Comput. Math.* **2014**, *40*, 629–650. [CrossRef]
43. Stankovic, L. Highly concentrated time-frequency distributions: Pseudo quantum signal representation. *IEEE Trans. Signal Process.* **1997**, *45*, 543–551. [CrossRef]
44. Stanković, L. The Support Uncertainty Principle and the Graph Rihaczek Distribution: Revisited and Improved. *IEEE Signal Process. Lett.* **2020**, *27*, 1030–1034. [CrossRef]
45. Elad, M.; Bruckstein, A.M. Generalized uncertainty prin- ciple and sparse representation in pairs of bases. *IEEE Trans. Inf. Theory* **2002**, *48*, 2558–2567. [CrossRef]
46. Stanković, L. *Digital Signal Processing with Selected Topics*; CreateSpace Independent Publishing Platform: Scotts Valley, CA, USA, 2015; ISBN 978-1514179987.
47. Stanković. L.; Sejdić, E; Daković, M. Vertex-frequency energy distributions. *IEEE Signal Process. Lett.* **2018**, *25*, 358–362. [CrossRef]
48. Stanković. L.; Sejdić, E; Daković, M. Reduced interference vertex-frequency distributions. *IEEE Signal Process. Lett.* **2018**, *25*, 1393–1397. [CrossRef]
49. Agaskar, A.; Lu, Y.M. A spectral graph uncertainty principle. *IEEE Trans. Inf. Theory* **2013**, *59*, 4338–4356. [CrossRef]
50. Sakiyama, A.; Tanaka, Y. Oversampled graph Laplacian matrix for graph filter banks. *IEEE Trans. Signal Process.* **2014**, *62*, 6425–6437. [CrossRef]
51. Girault, B. Stationary graph signals using an isometric graph translation. In Proceedings of the 2015 23rd IEEE European Signal Processing Conference (EUSIPCO), Nice, France, 31 August–4 September 2015; pp. 1516–1520.