

# A Dual Domain Approach to Graph Signal Processing

Submitted in partial fulfillment of the requirements for  
the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

John Y Shi

B.S., Computer Engineering, University of Maryland, College Park

M.S., Electrical and Computer Engineering, Carnegie Mellon University

B.S., Applied Mathematics, University of Maryland, College Park

Carnegie Mellon University

Pittsburgh, PA

Aug, 2022

© John Y Shi, 2022

All Rights Reserved

# Abstract

Graph Signal Processing (GSP) extends Discrete Signal Processing (DSP) to data supported by graphs by redefining traditional DSP concepts like signals, shift, filtering, and Fourier transform among others. This thesis develops and generalizes standard DSP operations for GSP in an intuitively pleasing way: 1) new concepts in GSP are often designed as natural extensions of DSP concepts; 2) GSP is *consistent* with DSP, i.e., when the underlying graph  $G$  is a directed cyclic graph, GSP becomes DSP; and 3) GSP leads to reinterpretation of well known DSP results and sheds new light on DSP facts and assumptions that are commonly taken for granted.

We build a theoretical foundation for GSP, introducing fundamental GSP concepts such as spectral graph shift, spectral convolution, spectral graph, spectral graph filters, and spectral delta functions. This leads to a spectral graph signal processing theory ( $\text{GSP}_{\text{sp}}$ ) that is the dual of the vertex based GSP.  $\text{GSP}_{\text{sp}}$  enables us to develop a unified graph signal sampling theory with GSP vertex and spectral domain dual versions for each of the four standard sampling steps of subsampling, decimation, upsampling, and interpolation.

To define the graph  $z$ -transform,  $\text{GzT}$ , we introduce the canonical *companion* model with its canonical *companion* shift and canonical *companion* graph. The companion shift and the companion graph modifies the structure of the DSP cyclic directed shift by adding appropriate boundary conditions.

The companion GSP model and  $\text{GSP}_{\text{sp}}$  show that, in GSP, there are two distinct models: the eigenvector model from current GSP literature and the canonical model we introduce. In DSP, these two models overlap and are equivalent, obscuring which model should be used in GSP for particular data processing tasks. We illustrate the significance of this dichotomy by presenting a GSP uncertainty principle, interpolating GSP filters, and GSP modulation as natural applications of the canonical companion signal model. In doing so, we show that, while equivalent in DSP, both models are needed for the complete picture in GSP.

## Acknowledgements

It has been an incredible five year journey. It would not have been possible without the support of my colleagues, friends, and family.

First, I'd like to thank my advisor, Professor José M.F. Moura, for his support and help throughout the last five years. Despite being very busy, José was always available to discuss research. We frequently spent hours deriving equations and discussing my latest ideas. I consider myself very lucky to have José as my mentor. This thesis would not have been possible without his invaluable support and mentorship.

I'd like to thank my thesis committee: Professor José M.F. Moura (Chair and Advisor), Professor Yuejie Chi, Professor Richard Stern, Professor Pulkit Grover, all from Carnegie Mellon University and Dr. Petros Boufounos of Mitsubishi Electric Research Laboratories (MERL).

I'd also like to thank and acknowledge my fellow Porter Hall B-Level colleagues and friends. Thanks to everyone in the Moura Lab for their support and advice throughout the five years: Jonathan Mei, Egeyevy Toropov, Shanghang Zhang, Stephen Kruzick, Satwik Kottur, Yuan Chen, Lynna (Liang-yan) Gui, Yixiong Zou, Yang Li, Chaojing Duan, Shreyas Chaudhari, Kawi Kamtue, Srinivasa Pranav, and Rajshekhar Das. In particular, thanks to Dr. Jian Du for his help getting my started in my first year. Thanks to my fellow GSP co-authors: Oren Wright and Mark Cheung. Thanks to my interns: Yao (Lavender) Jiang, Chris Liu, Wendy Summer, Tina Tian, Cynthia Fu, and Austin Lin.

Lastly, thanks to all the friends I've made and all the students I've taught at CMU over the past five years. Thanks to my father, who supported me for the last five years. Thanks to my CMU badminton family for all the fun memories. Thanks to Chihying Chang, Azaan Rehman, Tyler Vuong, Anqi Yang, Carmel Fisco, who have been friends with me since day 1.

I'd also like to acknowledge the support from several institutions and grants. This research is based upon work partially funded and supported by the Department of Defense under Contract

No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. This work is also partially supported by NSF grants CCF 1837607 and CCN 1513936.



# Contents

- 1 Introduction** **1**
  - 1.1 Review of Literature . . . . . 2
  - 1.2 Thesis Contributions . . . . . 4
  - 1.3 Thesis Summary . . . . . 10
- 2 Graph Signal Processing Primer** **11**
  - 2.1 DSP as GSP . . . . . 12
  - 2.2 GSP Basics . . . . . 14
  - 2.3 Conclusion . . . . . 19
- 3 GSP Spectral Shift and Graph Impulse** **21**
  - 3.1 GSP Spectral Shift . . . . . 21
  - 3.2 Graph Impulse . . . . . 32
  - 3.3 Conclusion . . . . . 36
- 4 Dual Domain GSP Sampling** **37**
  - 4.1 GSP Sampling: Subsampling and Decimation . . . . . 38
  - 4.2 GSP Sampling: Upsampling and Interpolation . . . . . 53
  - 4.3 Conclusion . . . . . 63
- 5 Signal Representations and Companion Canonical Model** **65**
  - 5.1 Vertex and Fourier Signal Representations . . . . . 65
  - 5.2 Vertex Impulsive Representation . . . . . 69
  - 5.3 Spectral Impulsive Representation . . . . . 73
  - 5.4 Companion Model—a Canonical GSP Model . . . . . 78
  - 5.5 Conclusion . . . . . 85
- 6 The Graph  $z$ -Transform (GzT)** **87**
  - 6.1 Vertex and Spectral Graph  $z$ -transforms . . . . . 87
  - 6.2 Fast Graph Convolution with the FFT . . . . . 93
  - 6.3 Conclusion . . . . . 97

<b>7</b>	<b>GSP Uncertainty Principle</b>	<b>99</b>
7.1	Brief Literature Review . . . . .	99
7.2	Uncertainty Principle for Spectral and Vertex Impulsive Signal Representations	101
7.3	Bandlimited Signal $\hat{s}$ : Reducing the Vandermonde System . . . . .	106
7.4	Uncertainty Principle for Standard and Spectral Impulsive Signal Representations	113
7.5	Conclusion . . . . .	119
<b>8</b>	<b>GSP Interpolating Filters</b>	<b>121</b>
8.1	Finding $p$ : Lagrange Interpolation in GSP . . . . .	121
8.2	Lagrange Basis Polynomial Filters $\ell_i(A)$ . . . . .	124
8.3	Finding $q$ : Lagrange Interpolation in $\text{GSP}_{\text{sp}}$ . . . . .	129
8.4	Lagrange Basis Polynomial Filters $\ell_{i,\text{sp}}(M)$ . . . . .	131
8.5	Conclusion . . . . .	133
<b>9</b>	<b>GSP Modulation and Demodulation</b>	<b>135</b>
9.1	Vertex Division Multiplexing . . . . .	136
9.2	Spectral Division Multiplexing . . . . .	137
9.3	Spectral $z$ -transform Division Multiplexing . . . . .	142
9.4	Conclusion . . . . .	145
<b>10</b>	<b>Conclusion</b>	<b>147</b>
10.1	Summary of the Thesis . . . . .	147
10.2	Contributions of the Thesis . . . . .	153
10.3	Future Directions . . . . .	159
<b>11</b>	<b>Appendices</b>	<b>171</b>
11.1	Impulsive Graph Impulses . . . . .	171

# List of Figures

1.1	The sampling and reconstruction steps in this thesis. Each step is considered in both vertex and spectral domains. . . . .	5
2.1	Directed cycle graph with adjacency matrix $A_c$ . . . . .	12
3.1	Star graph: Shifts $A$ and $M$ for $N = 5$ . . . . .	32
4.1	The sampling and reconstruction steps in this thesis. Each step is considered in both vertex and spectral domains. . . . .	37
4.2	Circulant Matrix: sampled nodes for replicating filter are red. Blue edges connect nodes with next nodes. Orange edges connect nodes with fourth next nodes. Black edges connect nodes with sixth next nodes. . . . .	47
4.3	Kronecker Product Graphs. Red nodes are the sampled nodes for the replicating filter. . . . .	48
4.4	Values used by $Q_2$ to recover: values shown for each $Q_2$ . . . . .	62
5.1	Companion graph. Unlabelled edges have weight 1. Other edges labeled by their weights. . . . .	81
5.2	Directed ladder graph and its companion graph. . . . .	83
5.3	Eigenfrequencies of directed ladder graphs with 4 (red), 6 (green), 8 (blue), 10 (brown), 12 (black), and 14 (orange) nodes. . . . .	83
5.4	Path graph and its canonical companion graph. . . . .	84
6.1	Convolution of graphs signals $s$ and $t$ . . . . .	93
6.2	Example of circular convolution of $s$ and $t$ using both polynomial filtering (above) and the FFT (below) for the 100 node directed ladder graph in figure 5.2. Both methods produce the same result. . . . .	96
7.1	The feasibility region for the GSP uncertainty principle in result 7.3 for $N = 10$ . Red dots represent infeasible regions and blue dots represent feasible regions. . . . .	105
7.2	Erdős-Rényi graph for $N = 10$ with probability 0.25 . . . . .	111

7.3	The eigenvalues of the 10-node Erdős-Rényi graph, plotted with the unit circle (in blue). . . . .	111
7.4	$p(x), r(x)$ for the example, along with the two real coordinates at $(2.4, 0), (0, 1.75)$ . $p(x)$ interpolates and goes through both points, but $r(x)$ only goes through $(2.4, 0)$ because it is a root. . . . .	112
7.5	The feasibility region for the DSP uncertainty principle in result 7.3 for $N = 10$ . Red dots represent infeasible regions and blue dots represent feasible regions. The green line represents the boundary using the DSP Uncertainty Principle in the literature. . . . .	117
8.1	Block diagram applying the all-pass graph filter in Corollary 8.1 to $s$ . Each Lagrange polynomial filter $\ell_i(A)$ extracts weighted eigenvector of $A, \hat{s}_i v_i$ . These are then summed together to reobtain $s$ . . . . .	128
8.2	Block diagram applying the all-pass graph filter in Corollary 8.2 to $\hat{s}$ . Each Lagrange polynomial filter $\ell_{i,sp}(M)$ extracts weighted eigenvector of $M, s_i u_i$ . These are then summed together to reobtain $\hat{s}$ . . . . .	133
10.1	Graph signal domains and the transformations between them. For each domain, both the signal and shift are given. . . . .	154
10.2	DSP Signal Representations: $s = [1, 2, 3, 4]^T, \hat{s} = [5, -1 + j, -1, -1 - j]^T$ . The standard representation and impulsive representations coincide. The Fourier and spectral impulsive representations also coincide. . . . .	156
10.3	Figure 10.1 for DSP. The three colored regions all have the same signals and shift. In DSP, $A = M$ . The three colored regions are all identical to each other in DSP. . . . .	157

# List of Tables

7.1	$s$ and corresponding $\hat{s}$ . For $\hat{s}$ with exactly 1 zero, $s$ can have 0, 1, 2 trailing zeros.	106
7.2	$s$ and corresponding $\hat{s}$ , $r(x), e(x)$ , $e(x)$ interpolation points, $p(x)$ . For $\hat{s}$ with exactly 1 zero, $s$ can have 0, 1, 2 trailing zeros. In each case, $r(x) = x - j$ .	110
10.1	Comparison of the properties in GSP, DSP, and the Canonical Signal Model (CSM)	153

# Chapter 1

## Introduction

Discrete signal processing (DSP) has been successful in processing grid-based signals such as time signals and image signals. In DSP, a sample  $s_n$  of a real valued time signal  $s$  like a sample of a segment of speech or of audio is indexed by the time instant  $n$  at which the sample occurs. Similarly, the intensity or color  $s_{ij}$  of an image pixel is indexed by the location  $(i, j)$  of the pixel. Both indices, time  $n$  and pixel  $(i, j)$ , usually take values on regularly spaced one dimensional (1D) or two dimensional (2D) grids.

In contrast, the samples of graph signals such as the number of taxi pickups at street intersections of New York City, the voltages in an electrical grid, and demographic data of people in political blogs are seemingly placed in non-regular, arbitrary locations in space. This indexing structure is better described by a graph. Graph signal processing (GSP) extends traditional DSP concepts to analyze graph signals, signals indexed or defined on graphs.

References [1–3] extend several DSP concepts to GSP, including graph shift  $A$ , graph filtering, graph Fourier transform (GFT), graph frequency, and graph filter response. In GSP, the adjacency matrix  $A$  becomes the *shift* operator, and it plays in GSP the same role that the time shift  $z^{-1}$  plays in DSP. Reference [4] presents an alternative “spectral domain” development of GSP that is restricted to data indexed by nodes of *undirected* graphs. It starts from spectral decompositions of the data in terms of the eigenfunctions of a variational operator, the graph Laplacian  $L$ . For *undirected* graphs,  $L$  and  $A$  have the same eigendecomposition and, for undirected graphs, the spectral analysis is equivalent for the two approaches.

In [1–3], GSP is developed as a direct, intuitive extension and generalization of DSP. This

approach to GSP is intuitively pleasing: 1) new concepts in GSP are often designed as natural extensions of DSP concepts; 2) GSP is *consistent* with DSP, i.e., when the underlying graph  $G$  is a directed cyclic graph, GSP becomes DSP; and 3) GSP leads to reinterpretation of well known DSP results and sheds new light on DSP facts and assumptions that are commonly taken for granted. These points will become apparent in this thesis.

## 1.1 Review of Literature

The GSP literature is vast, by now covering many topics in processing graph signals. The approach in [1–3] identifies as basic building block the shift filter  $A$ , building on the Algebraic Signal Processing in [5–9]. The approach in [4] departs from a variational operator, the graph Laplacian  $L$ , motivated by for example earlier work from spectral graph theory [10–12], from work extending wavelets to data from irregularly placed sensors in sensor networks [13–15], and from research on sampling graph based data [16, 17]. A comprehensive review covering both approaches and illustrating many different applications of GSP is [18].

**Brief Review of GSP Sampling Literature:** In the last decade, there has been a robust literature on GSP sampling. With a few exceptions, these works assume *undirected* graphs. We highlight a few main points and refer to the references and their bibliography for a more comprehensive review. Most of the work has been concerned with choosing the sampling set and developing recovery methods that address issues like noise, or aliasing, or robustness to computational errors, or speed of computation. These are important issues, but are not our focus. Still, the brief review of GSP sampling literature helps put this thesis in context.

References [19, 20] consider subspaces of bandlimited signals (Paley-Wiener spaces) and show that signals supported by undirected graphs can be perfectly reconstructed from values in a *sampling* set  $S$ , termed *uniqueness* set. Critically sampled graph signals *restricted to* undirected  $k$ -regular bipartite graphs are considered in [21] that proposes two-channel wavelet filter banks for perfect reconstruction. General undirected graphs are approximated by a decomposition in terms of  $k$ -regular bipartite graphs. Papers [22–25] consider choosing  $S$  for

stable reconstruction and methods that avoid spectral decompositions of the graph Laplacian. While the previous references apply to undirected graphs, [26] shows that a necessary and sufficient condition for perfect reconstruction is for the sampling set to choose  $K$  linearly independent rows from  $K$  columns of the inverse graph Fourier transform. The paper shows that random sampling chooses with high probability for Erdős Rényi graphs a sampling set from which perfect reconstruction can be achieved. This reference also shows that sampling preserves first order differences on the sampled nodes. In [27], various sampling schemes are considered including uniform sampling, experimentally designed sampling, and active sampling. Random sampling for undirected graphs is also studied in [28] that presents a condition in terms of the invertibility of a  $K \times K$  *kernel* matrix corresponding to rows and columns selected by a determinantal point process. Reference [29] derives an uncertainty principle for graph signals and conditions for recovery of bandlimited signals from a subset of the samples. In [30], the authors propose a sampling scheme that uses as input observations taken at a single node and corresponds to sequential applications of the graph-shift operator. Beyond sampling of bandlimited signals, [31] considers piecewise constant signals on undirected graphs and [32] smooth signals, with recovery by a Lasso like procedure. Reference [33] reconstructs time signals from projections on low rank approximation subspaces, with [34] extending it to undirected graph signals. Reference [35] presents a spectral domain sampling where the spectrum of the subsampled signal replicates the bandlimited spectrum of the original signal. But the method in [35] does not respect the traditional concept of sampling, namely, discarding samples in the vertex domain and reconstructing the original signal from the samples kept. In fact, the sampled signal in [35] not only keeps all samples of the original signal, but also distorts them.

Many additional topics have been considered in GSP. A sample of these include: alternative (unitary, but not local) shift operators [36, 37]; approximating graph signals [38]; extensive work on sampling of graph signals, e.g., [22, 24, 26, 30, 35, 39], see the recent review [40]; extending classical multirate signal processing to graphs [41, 42]; an uncertainty principle for graph

signals [43]; the study of graph diffusions [44]; graph signal recovery [32]; interpolation and reconstruction of graph signals [45, 46]; stationarity of graph processes [47]; learning graphs from data [48–50]; or non-diagonalizable shifts and the graph Fourier transform [51].

## 1.2 Thesis Contributions

This thesis develops and generalizes standard DSP operations for GSP in a natural and intuitive way. We summarize the contributions of the thesis:

1. **Spectral Graph Signal Processing Theory,  $\text{GSP}_{\text{sp}}$ :** We build a theoretical foundation for GSP, introducing new fundamental GSP concepts such as *spectral* graph shift  $M$ , convolution in the *spectral* domain, and *spectral* delta functions. This leads to a *spectral* graph signal processing theory ( $\text{GSP}_{\text{sp}}$ ) that is the dual of vertex based GSP. For example,  $M$  is the spectral shift acting in the spectral domain rather than shift  $A$  that acts in the vertex domain. Also, linear shift invariant (LSI) filtering in the spectral domain is with polynomial filters  $P(M)$ , instead of the vertex domain LSI filters,  $P(A)$ .
2. **Unified GSP Sampling Theory:** GSP sampling is usually presented in GSP literature as either in the vertex (time) or in the spectral (frequency) domain with no clear connection between the two. In contrast, DSP sampling is interpretable in both domains with clear connections between the two, e.g., sampling in time with a delta train is equivalent to convolution in frequency with a different delta train. This thesis is concerned with what is lacking [40], namely, presenting an unified GSP sampling theory and the analogy (duality) between vertex and spectral GSP sampling operations, just like for DSP sampling. Reference [40] identifies the “[I]nterconnection between vertex and spectral representations of sampling ...” as an open issue worthy of further study. It further asks how “can these sampling approaches be described in a more unified way beyond a few known special cases?” stating that “This may lead to a more intuitive understanding of graph signal sampling.” This thesis deals with this open issue.

$\text{GSP}_{\text{sp}}$  enables us to develop an unified graph signal sampling theory with GSP vertex and spectral domain dual versions for each of the four standard sampling steps of subsampling, decimation, upsampling, and interpolation, see fig. 1.1. Each step is interpretable in both

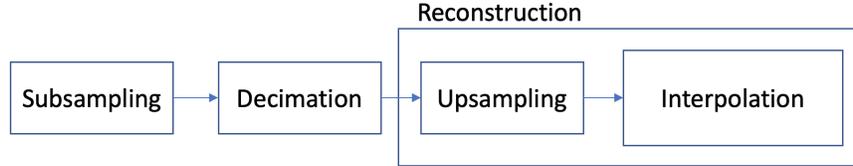


Figure 1.1: The sampling and reconstruction steps in this thesis. Each step is considered in both vertex and spectral domains.

the vertex and spectral domains. We show clear parallels between DSP and GSP sampling exist, i.e., commonly held DSP sampling facts actually remain true when generalized to GSP, but with a twist.

We summarize our contributions on GSP sampling.

- *GSP sampling—the dualism between vertex and spectral representations of sampling.* Our presentation of graph sampling parallels the traditional sampling of discrete time signals [52]. This thesis shows for graph sampling i) how it is analogous to the Shannon-Nyquist and shift-invariant sampling of time signals; and ii) how and when it deviates given the intrinsic differences between GSP and DSP constructs. Safeguarding the distinctions, we replicate the dualism between time and frequency sampling operations in DSP with a similar dualism between GSP sampling operations in the vertex and the graph spectral domains. For every sampling operation, we present its vertex domain interpretation *and* its spectral domain interpretation. When GSP becomes DSP, we point out which among several alternative choices of the sampling set leads to the Shannon-Nyquist uniform sampling of time signals.

We detail our contributions to the GSP sampling and reconstruction steps: subsampling, decimation, upsampling, and interpolation.

- *Graph subsampling: LSI spectral filtering.* Subsampling of a graph signal is multi-

plication in the vertex domain of the graph signal  $s$  by a zero-one graph sampling signal  $\delta^{\text{spl}}$ . To determine the graph spectral domain operation that is dual to multiplication of graph signals in the vertex domain, we introduce graph LSI filtering in the graph spectral domain. Namely, we introduce graph convolution in the spectral domain by a polynomial spectral filter  $P(M)$  using a new graph spectral shift  $M$  that we define. This dualizes multiplication of graph signals in the vertex domain with graph convolution or graph filtering in the spectral domain. Further, spectral shift  $M$  defines a new spectral graph whose nodes now index the graph Fourier coefficients by the graph frequencies. The interesting point is that subsampling by  $\delta^{\text{spl}}$  is in the spectral domain achieved by LSI spectral filtering (by  $P(M)$ ) just like in DSP sampling.

- *Graph subsampling: Spectral replication.* Like in DSP, in GSP we show that the spectrum of the subsampled signal is a set of (distorted) replicas of the lowpass spectrum of the original signal, with no aliasing if sampling rate is at least the “GSP Nyquist” rate.
- *Graph decimation.* Decimation downsizes the original graph from a graph of order  $N$  to a graph of order  $K$ . This is determined by i) the choice of sampling signal (and sampling set), and ii) by the  $K$  nonzero components of the graph spectrum of the signal. This follows standard procedure, for example [27]. Just like for DSP, this step expands the spectrum of the decimated graph signal to the full band (which is now reduced to a smaller number of graph frequencies). But, even though where the graph spectrum is nonzero is given or assumed, there are several choices for the sampling signal, and so for the decimated graph. We illustrate why and what choices lead to the uniform sampling in DSP. In DSP, certain choices preserve the nature of the graph, i.e., both the original and the decimated graph are cycle graphs (of different orders), while other choices lead to a decimated graph that is no longer cyclic. In GSP, this is hardly the case. For general graphs, the decimated and

original graphs are very different.

- *Reconstruction: upsampling.* The upsampling with reinsertion of zeros in the decimated graph signal, and the reconstruction of the original  $N$ th-order graph follows the same steps in GSP and DSP. Like in DSP, it leads to a contracted signal spectrum in GSP, i.e., a signal spectrum that is zero in  $N - K$  components.
- *Reconstruction: interpolation by spectral filtering and ideal LSI (vertex) filtering.* Perfect reconstruction in GSP sampling, assuming no aliasing, is obtained like in DSP by spectral filtering, but in two distinct steps. First, with a spectral filter  $Q$  that is not necessarily LSI, followed by LSI vertex ideal filtering. When the graph is cyclic and GSP becomes DSP, the reconstruction filter  $Q$  becomes trivially a gain of  $K$ , and the LSI vertex ideal sampling is the Shannon ideal filter with a sinc function as impulse response (in the vertex domain).

3. **GSP Signal Representations:** Signal representations play a crucial role in signal processing. In DSP, we have the standard representation and the frequency representation. We develop new GSP signal representations as shifted delta functions in both the vertex domain and spectral domain using the *spectral* graph shift  $M$ , convolution in the *spectral* domain, and *spectral* delta functions developed in the spectral graph signal processing theory,  $\text{GSP}_{\text{sp}}$ . Current GSP literature describes graph signals by their standard (node or vertex) representation  $s$  or their spectral representation  $\widehat{s}$ , but practically none has discussed or studied these other graph signal representations.

4. **Canonical Companion Model:** Using the new GSP signal representations, we introduce a canonical *companion* model with its canonical *companion* shift and canonical *companion* graph. Any graph  $G$  can be converted into its canonical companion graph that replicates the structure of the cyclic shift and the cyclic (time) graph in DSP, but with different boundary conditions given by the Cayley-Hamilton Theorem. This is true even for undirected graphs. We show shifts by  $A$  in the vertex domain and shifts by

$M$  in the spectral domain both become shifts by the canonical *companion* shift in the canonical *companion model*.

5. **Graph  $z$ -Transform (GzT) and Fast Linear Convolution:** The canonical *companion* model allows us to define the graph  $z$ -transform for the vertex domain. The graph  $z$ -transform provides a symbolic polynomial representation for graph signals. The graph  $z$ -transform reproduces many characteristics of DSP. We develop fast graph convolution in the vertex domain using the FFT of graph  $z$ -transform signals. We also introduce the dual of the graph  $z$ -transform, the spectral graph  $z$ -transform, that uses the spectral shift  $M$  in the spectral domain. Unlike the graph  $z$  transform, this spectral graph  $z$ -transform only exists in GSP and not DSP.
6. **GSP Uncertainty Principle:** Previous GSP Uncertainty Principles showed relationships between the spread between  $s$  and  $\hat{s}$ , but did not directly show a relationship between the bandlimitness of  $\hat{s}$  and support of  $s$  like [53] did in DSP.

In DSP, since DFT and its inverse are Vandermonde matrices,  $s$  and  $\hat{s}$  can be interpreted as interpolating polynomials and the Fourier relationship can be interpreted as polynomial interpolation:

- (a)  $s$  is the coefficients of the interpolating polynomial that goes through the points  $(\lambda_i, \sqrt{N}\hat{s}_i)$  where  $\lambda_i$  is the  $i$ th eigenvalue of  $A_c$  and  $\hat{s}_i$  is the  $i$ th entry of  $\hat{s}$ ,  $i = 0, \dots, N - 1$ ,
- (b)  $\hat{s}$  is the coefficients of the interpolating polynomial that goes through the points  $(\lambda_i^*, \sqrt{N}s_i)$  where  $\lambda_i^*$  is the conjugate of the  $i$ th eigenvalue of  $A_c$  and  $s_i$  is the  $i$ th entry of  $s$ ,  $i = 0, \dots, N - 1$ .

The DSP Uncertainty Principle in [53] relies on precisely on the DFT and its inverse being Vandermonde matrices. In GSP, since the GFT and its inverse are not Vandermonde matrices, the Fourier relationship cannot be interpreted as polynomial interpolation.

While  $s$  and  $\hat{s}$  are related by the GFT (a non-Vandermonde matrix),  $p$  and  $\hat{s}$  are related by a Vandermonde matrix. The canonical signal model allows us to interpret  $p$  and  $\hat{s}$  (similarly,  $q$  and  $s$ ) as polynomial interpolation. DSP properties that rely on the DFT being a Vandermonde matrix do not generally hold for  $s$  and  $\hat{s}$  in GSP, but can be generalized to  $p$  and  $\hat{s}$ . We explore this polynomial interpolation and its applications through the GSP Uncertainty Principle, GSP Interpolating Filters, and GSP Modulation and Demodulation.

In this thesis, we present a GSP Uncertainty Principle based on the interpolation of the graph  $z$ -transform  $p$  and spectral signal  $\hat{s}$  (similarly, spectral graph  $z$ -transform  $q$  and vertex signal  $s$ ). This GSP Uncertainty Principle directly relates the bandlimitness of  $p$  and  $\hat{s}$ . We use this uncertainty principle to show that when  $\hat{s}$  is bandlimited, we can reduce and simplify the Vandermonde system required to calculate  $p$ .

7. **GSP Interpolating Filters:** Since  $p$  and  $\hat{s}$  are related through polynomial interpolation, we provide a general way to find  $p$  using Lagrange interpolation in GSP. This leads to Lagrange basis polynomial filters in both the vertex and spectral domains,  $\ell_i(A)$  and  $\ell_{i,sp}(M)$  respectively. We explore the properties of these Lagrange basis polynomial filters, applying them to both delta functions  $\delta_0$  and signal  $s$ . Using these Lagrange basis polynomials, we develop all-pass graph filters and narrow-band filters for GSP in both the vertex and spectral domains.
8. **GSP Modulation and Demodulation:** We present three GSP multiplexing methods. In DSP, multiplexing is done by partitioning the signal bandwidth into (non-overlapping) spectral bands with each band containing one signal. This partitioning can either happen in the time domain (time division multiplexing) or the frequency domain (frequency division multiplexing). In this thesis, we present GSP multiplexing for vertex domain multiplexing, spectral domain multiplexing, and spectral  $z$ -transform division multiplexing. To perform the multiplexing, we provide methods for both GSP modulation and demodulation.

9. **Dual GSP Spaces:** DSP and GSP is presented in two domains: the vertex and spectral domains related through the GFT. In this thesis, we take this simple picture of GSP and expand it to four GSP domains: the vertex, spectral,  $z$ -transform, and spectral  $z$ -transform domains, showing their relationships and interpretations. This provides an expanded version of DSP.

Parts of this thesis have been published in papers: [54–64].

### 1.3 Thesis Summary

Chapter 2 reviews basic GSP concepts. Chapter 3 introduces  $\text{GSP}_{\text{sp}}$  including the spectral graph shift, spectral domain convolution, and graph impulses. Chapter 4 presents dual domain sampling, GSP sampling that is interpretable in both the vertex and spectral domains. Chapter 5 introduces GSP signal representations using  $\text{GSP}_{\text{sp}}$  and the canonical companion model. Chapter 6 discusses the graph  $z$ -transform and fast graph convolution with the FFT. Chapter 7 presents the GSP Uncertainty Principle. Chapter 8 presents GSP interpolating filters. Chapter 9 presents GSP modulation and demodulation. Chapter 10 concludes the thesis and discusses future directions.

# Chapter 2

## Graph Signal Processing Primer

This chapter briefly reviews GSP following [1–3]. The first section presents DSP using matrices. The second section reviews GSP basics.

Let  $G = (V, E)$  be a graph of order  $N$ , i.e., with vertex or node set  $V$  of cardinality  $|V| = N$ , and with edge set  $E$ . The graph  $G$  is arbitrary, possibly directed, undirected, or mixed with directed and undirected edges. The graph can be specified by an adjacency matrix  $A$ , where  $A_{ij} = 1$  if there is a directed edge from node  $j$  to node  $i$  or  $A_{ij} = 0$  otherwise.<sup>1</sup>

**Remark 2.1** (*A equivalence class*). *The adjacency matrix  $A$  depends on the ordering of the nodes of  $V$ . Different node orderings are related by permutations  $P$  and the corresponding adjacency matrices are conjugated by  $P$ , i.e.,  $P^{-1}AP = P^TAP$ . In other words, adjacency matrices describing the same graph are an equivalence class under the symmetric group of permutations. We assume that a representative of this class has been chosen, by fixing the labeling order of the nodes in  $V$ , which becomes now an ordered set, and identify graph  $G$  with adjacency matrix  $A$  rather than with the class of adjacency matrices.*<sup>2</sup>

A graph signal  $s$  is an (ordered)  $N$ -tuple  $s = (s_0, \dots, s_{N-1})$  that assigns to each node  $n \in V$ ,  $n = 0, \dots, N - 1$ , the graph sample  $s_n$ . In other words, graph signal samples  $s_n$  are indexed by the nodes  $n$  of the graph. In this paper, we consider the graph samples to be complex valued,  $s_n \in \mathbb{C}$ . The graph signal  $s$  is then a vector in  $\mathbb{C}^N$ , the  $N$ -dimensional vector space over the complex field  $\mathbb{C}$ .

---

<sup>1</sup> Computer Science reverses this convention and the adjacency is  $A^T$ .

<sup>2</sup> Graphs that differ by permutations  $P$  of  $V$  are isomorphic.

## 2.1 DSP as GSP

To motivate GSP, we start by casting DSP in the context of GSP, see for example [1, 3].

Consider the  $N$  node directed cycle graph  $G$  in figure 2.1 with adjacency matrix  $A_c$

$$A_c = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & \dots & 0 & 0 \\ \vdots & 1 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}. \quad (2.1)$$

The nodes of the cycle graph  $G$  represent the time ticks  $n$  and are naturally ordered. The time

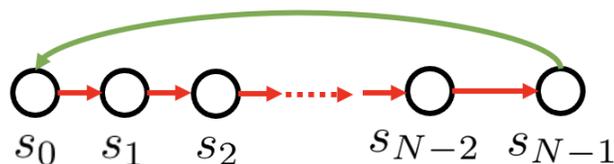


Figure 2.1: Directed cycle graph with adjacency matrix  $A_c$ .

signal samples  $s_n$  are indexed by the nodes of  $G$ . Matrix  $A_c$  is also the matrix representation of the shift  $z^{-1}$  in DSP (assuming periodic boundary conditions [1, 6, 7]).

$$A_c \begin{bmatrix} s_0 \\ s_1 \\ \dots \\ s_{N-1} \end{bmatrix} = \begin{bmatrix} s_{N-1} \\ s_0 \\ \dots \\ s_{N-2} \end{bmatrix}. \quad (2.2)$$

The eigenvalues (or a normalized version)  $\lambda_k$  and the eigenvectors  $v_k$  of the cyclic  $A_c$  in (2.1) are the discrete time frequencies and the discrete time harmonics, spectral components, or

eigenmodes of time signals

$$\lambda_k = e^{-j\frac{2\pi}{N}k}, \quad k = 0, \dots, N-1 \quad (2.3)$$

$$v_k = \frac{1}{\sqrt{N}} \left[ 1 \ e^{j\frac{2\pi}{N}k} \ \dots \ e^{j\frac{2\pi}{N}k(N-1)} \right]^T. \quad (2.4)$$

The Discrete Fourier Transform, DFT, is obtained through the diagonalization of  $A_c$

$$A_c = \text{DFT}^H \Lambda \text{DFT}, \quad (2.5)$$

where

$$\Lambda = \text{diag} [\lambda] \quad (2.6)$$

$$\lambda = [\lambda_0 \ \dots \ \lambda_{N-1}]^T = \left[ 1 \ e^{-j\frac{2\pi}{N}} \ \dots \ e^{-j\frac{2\pi}{N}(N-1)} \right]^T \quad (2.7)$$

$$\text{DFT} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & e^{-j\frac{2\pi}{N}} & \dots & e^{-j\frac{2\pi}{N}(N-1)} \\ \vdots & \vdots & & \vdots \\ 1 & e^{-j\frac{2\pi}{N}(N-1)} & \dots & e^{-j\frac{2\pi}{N}(N-1)(N-1)} \end{bmatrix} \quad (2.8)$$

$$= \frac{1}{\sqrt{N}} [\lambda^0 \ \dots \ \lambda^{N-1}] \quad (2.9)$$

$$\text{DFT}^H = [v_0 \ \dots \ v_{N-1}]. \quad (2.10)$$

The matrix  $\Lambda$  in (2.6) is the diagonal matrix of the eigenvalues, i.e., its diagonal entries are the graph frequencies. Equation (2.7) defines the *graph frequency vector*  $\lambda$  and equation (2.9) uses the notation

$$\lambda^k = \lambda \odot \lambda \ \dots \odot \lambda$$

to represent  $k$  times the Hadamard or entrywise product of the graph frequency vector  $\lambda$ . By (2.9) and (2.10), in DSP, the powers of the graph frequency vector  $\lambda$  are conjugates of the

eigenvectors,

$$\frac{1}{\sqrt{N}}\lambda^k = v_k^* \quad (2.11)$$

and, by (2.10), the columns of the  $\text{DFT}^H$  are the eigenvectors  $v_k$  of  $A$ . The DFT is symmetric,

$$\text{DFT} = \text{DFT}^T$$

and unitary,

$$\text{DFT}^{-1} = \text{DFT}^H$$

However, unlike GSP, (2.9) and (2.10) also show the DFT and its inverse are both Vandermonde matrices<sup>3</sup>. So, the columns of  $\text{DFT}^H$  have two equivalent interpretations (shown in (2.10)): either eigenvectors of  $A_c$ ,  $v_i$ , or powers of the conjugate of the eigenvalues of  $A_c$ ,  $\lambda^{*i}$ ,

$$v_i = \frac{1}{\sqrt{N}}\lambda^{*i}$$

## 2.2 GSP Basics

We now let  $A$  be the adjacency matrix of an arbitrary (directed or undirected) graph  $G$  of  $N$  nodes and  $s$  be a graph signal. As observed for DSP and following [1–3], in GSP,  $A$  is the shift operator. It captures the local dependencies of the signal sample  $s_n$  on the signal samples  $s_m$  at the in-vertex neighbors  $m \in \eta_n$  of  $n$  (given by the nonzero entries of row  $n$  of  $A$  and where  $\eta_n$  is the (in)neighborhood of  $n$ , i.e., the set of (in)neighbors of  $n$ ). The eigenvalues  $\lambda_k$  and eigenvectors  $v_k$  of  $A$  are the graph frequencies and graph spectral modes. Let graph frequency vector  $\lambda$  and matrix  $\Lambda$  be defined as before to collect the graph eigenvalues

$$\lambda = [\lambda_0, \lambda_1, \dots, \lambda_{N-1}]^T, \quad \Lambda = \text{diag}[\lambda]. \quad (2.12)$$

The following assumptions hold even when not stated. On occasion, we state them explic-

---

<sup>3</sup>See (2.27) for the definition of a Vandermonde matrix.

itly.

**Assumption 2.1** (Strongly connected graph). *The graph  $G$  is strongly connected.*

Under assumption 2.1, matrix  $A$  has no zero column or row.

**Assumption 2.2** (Distinct eigenfrequencies). *The eigenvalues of  $A$  are distinct.*

Under assumption 2.2,<sup>4</sup>  $A$  is diagonalizable and the Graph Fourier Transform (GFT) is found<sup>5</sup> by

$$A = \text{GFT}^{-1} \Lambda \text{GFT}. \quad (2.13)$$

If  $A$  is symmetric, which is the case with undirected graphs, GFT is orthogonal ( $\text{GFT}^{-1} = \text{GFT}^T$ ), and if  $A$  is normal then GFT is unitary ( $\text{GFT}^{-1} = \text{GFT}^H$ ). For general graphs,  $A$  is neither symmetric nor normal, but the GFT is full rank and invertible.<sup>6</sup> The spectral modes are the columns  $v_k$  of  $\text{GFT}^{-1}$ . In GSP, unlike DSP, the columns of  $\text{GFT}^{-1}$  can only be interpreted as the eigenvectors of  $A$ ,  $v_k$ , and are not Hadamard powers of the vector of eigenvalues of  $A$ .

The graph Fourier transform of graph signal  $s$  is

$$\hat{s} = \text{GFT}s. \quad (2.14)$$

**Remark 2.2** (Fixing the GFT). *It is well known [66] that the diagonalization of a matrix is unique up to reordering of the eigenvalues and normalization of the eigenvectors. Paralleling remark 2.1, we assume the frequencies have been ordered and the eigenvectors appropriately normalized, see [67], fixing the GFT and  $\Lambda$ .*

---

<sup>4</sup> Distinct eigenvalues are assumed for simplicity. The results can be proved in the more general setting of  $A$  non-derogatory (equal minimum and characteristic polynomials (up to a factor  $\pm 1$ ), or, equivalently, the geometric multiplicity of any eigenvalue to be 1 (single eigenvector)).

<sup>5</sup> If assumption 2.2 does not hold, see [51] for further details on the GFT.

<sup>6</sup> See [65] for numerically stable diagonalization of  $A$  for directed graphs.

## 2.2.1 Cayley-Hamilton Theorem

Let the characteristic polynomial of  $A$  be

$$\Delta(x) = c_0 + c_1x + \cdots + c_{N-1}x^{N-1} + x^N. \quad (2.15)$$

By the Cayley-Hamilton Theorem [66,68,69],  $A$  satisfies its characteristic polynomial  $\Delta(A) = 0$  and so

$$A^N = -c_0I - c_1A - \cdots - c_{N-1}A^{N-1}, \quad (2.16)$$

and  $A^k$ ,  $k \geq N$ , is reduced by modular arithmetic mod  $\Delta(A)(\cdot)$ .

## 2.2.2 Linear Shift Invariant (LSI) Filtering

Under assumption 2.2, LSI filters in the vertex domain are polynomials  $P(A)$  in the shift. By Cayley-Hamilton,<sup>7</sup>  $P(A)$  is at most of degree  $N - 1$ .

The graph Fourier theorem [1] parallels DSP's theorem (convolution in time is pointwise multiplication in frequency)

$$P(A) s \xrightarrow{\text{GFT}} P(\Lambda) \widehat{s}, \quad (2.17)$$

and, in particular, the vertex shift relation

$$A s \xrightarrow{\text{GFT}} \Lambda \widehat{s}. \quad (2.18)$$

**Filtering in the vertex domain.** In GSP, linear, shift invariant (LSI) filters are poly-

---

<sup>7</sup> Under assumption 2.2, the minimal polynomial of  $A$  equals  $\Delta(A)$ .

mials  $P(A)$  of the shift  $A$ ,

$$P(A) = p_0I + p_1A + \cdots + p_{N-1}A^{N-1}, \quad (2.19)$$

and LSI graph *filtering* is matrix-vector multiplication [1]

$$t = P(A) \cdot s = [p_0I + p_1A + \cdots + p_{N-1}A^{N-1}] \cdot s. \quad (2.20)$$

**Graph frequency response.** For graph filter  $P(A)$ ,

$$P(A) = \text{GFT}^{-1}P(\Lambda) \text{GFT} \quad (2.21)$$

$$\text{with } P(\Lambda) = \text{diag}[P(\lambda_0), \cdots, P(\lambda_{N-1})], \quad (2.22)$$

where  $P(\lambda_n)$  is  $P(x)$  evaluated at the eigenfrequency  $\lambda_n$ .

The *graph frequency response*  $p(\lambda)$  of  $P(A)$  is

$$p(\lambda) = P(\Lambda) \cdot \mathbf{1} = \left[ P(\lambda_0) \quad \cdots \quad P(\lambda_{N-1}) \right]^T. \quad (2.23)$$

**Filtering in the frequency domain.** Filtering in the spectral domain then becomes:

$$\hat{t} = P(\Lambda) \cdot \hat{s} = (P(\Lambda) \cdot \mathbf{1}) \odot \hat{s} = p(\lambda) \odot \hat{s} \quad (2.24)$$

where  $\odot$  is the Hadamard or pointwise or componentwise product of the frequency response  $p(\lambda)$  and  $\hat{s}$ .

Equations (2.20) and (2.24) are the equivalent versions of filtering in the vertex and spectral domains in GSP: graph filtering in the vertex domain multiplies the vector signal  $s$  by the matrix filter  $P(A)$ . In the spectral domain, it is the product of the *diagonal* matrix filter  $P(\Lambda)$  with the graph Fourier transformed  $\hat{s}$ , or, equivalently, it is the pointwise product  $\odot$  of the graph frequency response  $p(\lambda)$  with  $\hat{s}$ .

**LSI filter  $P(A)$  and its frequency response  $\widehat{h}$ .** Given a polynomial filter  $P(A)$  with coefficients  $p = [p_0 \dots p_{N-1}]^T$ , equation (2.23) gives the frequency response  $p(\lambda)$  of  $P(A)$ . The next result is the reverse: given a frequency response  $\widehat{h}$ , determine the filter  $P_h(A)$ .

**Result 2.1.** *The filter  $P_h(A)$  with frequency response  $\widehat{h}$  is*

$$P_h(A) = GFT^{-1} \text{diag} \left[ \widehat{h} \right] GFT. \quad (2.25)$$

*Proof.* By (2.23),  $\widehat{h} = P_h(\Lambda) \cdot 1$ . Then,  $P_h(\Lambda) = \text{diag} \left[ \widehat{h} \right]$  and the result follows. ■

We refer to  $P_h(A)$  as the LSI filter associated with  $h$  or  $\widehat{h}$ . Result 2.1 gives  $P_h(A)$  as a matrix filter. The next result gives it as a LSI or polynomial filter.

**Result 2.2** (LSI filter associated with  $\widehat{h}$ ). *Let the vector of coefficients of  $P_h(A)$  be  $p_h = [p_0 \dots p_{N-1}]^T$ . Then, under assumption 2.2 of distinct eigenvalues,  $p_h$  is the solution to*

$$\mathcal{V} p_h = \widehat{h} \quad (2.26)$$

where  $\mathcal{V}$  is the Vandermonde matrix of eigenvalues of  $A$ :

$$\mathcal{V} = [\lambda^0 \lambda^1 \dots \lambda^{N-1}] = \begin{bmatrix} 1 & \lambda_0 & \dots & \lambda_0^{N-1} \\ \vdots & & & \vdots \\ 1 & \lambda_{N-1} & \dots & \lambda_{N-1}^{N-1} \end{bmatrix} \quad (2.27)$$

Equation (2.27) uses the notation where  $\lambda$  is the vector of eigenvalues and its powers  $\lambda^n$  are the Hadamard products of  $\lambda$  with itself

$$\lambda = [1 \lambda_0 \lambda_1 \dots \lambda_{N-1}]^T, \text{ and } \lambda^n = \overbrace{\lambda \odot \dots \odot \lambda}^{n \text{ times}}. \quad (2.28)$$

*Proof.* The proof follows because  $P_h(\Lambda) \cdot 1 = \mathcal{V} \cdot p$ . ■

For DSP, the Vandermonde,  $\mathcal{V}$ , is (up to a normalizing factor) the DFT matrix, and (2.26) shows that in DSP  $p_h$  is the inverse DFT of  $\hat{h}$ , so  $p_h = h$ .

## 2.3 Conclusion

In this chapter, we reviewed GSP literature following [1–3]. Section 2.1 presented DSP using matrices in the GSP framework. The graph shift in DSP is  $A_c$ , the adjacency matrix for the directed cyclic graph. The Fourier transform is the DFT matrix, which is also a Vandermonde matrix, found through eigendecomposition of  $A_c$ . Convolution in the time domain is the matrix-vector multiplication of a polynomial of the graph shift,  $P(A_c)$ , and time signal  $s$ . The second section presented the graph shift, Fourier transform and convolution for GSP. In GSP, the graph shift is an arbitrary  $A$ . The Fourier transform is the GFT, found through eigendecomposition of  $A$ . Convolution in the vertex domain is a matrix-vector multiplication of a polynomial of the graph shift,  $P(A)$ , and graph signal  $s$ . The rest of the thesis builds on this foundation.



# Chapter 3

## GSP Spectral Shift and Graph Impulse

This chapter introduces  $\text{GSP}_{\text{sp}}$ , a spectral graph signal processing theory that is the dual of vertex based GSP. The first section introduces the spectral shift  $M$ , the spectral graph  $G_{\text{sp}}$ , LSI spectral filtering, and their properties. The second section introduces GSP graph impulses in both the vertex and spectral domains.

### 3.1 GSP Spectral Shift

In this section, we introduce graph convolution or graph filtering in the *spectral* domain. In particular, we define linear shift invariant graph filters in the graph spectral domain as polynomials  $P(M)$  of a spectral shift  $M$  (see also [70] for a different definition). The operator  $M$  shifts the graph spectrum  $\widehat{s}$  of a signal  $s$ . This section establishes such an  $M$ .

We start by recalling the DSP properties of shifting a signal in the time and frequency domains [52]:

$$s_{n-1} \xrightarrow{\mathcal{F}} e^{-j\frac{2\pi}{N}m} \widehat{s}_m \quad (3.1)$$

$$e^{j\frac{2\pi}{N}n} s_n \xrightarrow{\mathcal{F}} \widehat{s}_{m-1}. \quad (3.2)$$

These equations show that shifting in the time domain multiplies the Fourier coefficient  $\widehat{s}_m$  by the eigenvalue  $\lambda_m = e^{-j\frac{2\pi}{N}m}$  of the shift  $A$ . Likewise, shifting in the frequency domain multiplies the signal sample  $s_k$  by  $\lambda_k^* = e^{j\frac{2\pi}{N}k}$ , the complex conjugate of the eigenvalue  $\lambda_k$  of  $A$ .

Collecting the time samples  $s_n$  in the vector signal  $s$  and the Fourier coefficients  $\hat{s}_m$  in vector  $\hat{s}$ , equations (3.1) and (3.2) lead to

$$A \cdot s \xrightarrow{\mathcal{F}} \Lambda \cdot \hat{s} \quad (3.3)$$

$$\Lambda^* \cdot s \xrightarrow{\mathcal{F}} \begin{bmatrix} \hat{s}_{N-1} \\ \hat{s}_0 \\ \vdots \\ \hat{s}_{(N-1)-1} \end{bmatrix} = A \cdot \hat{s}. \quad (3.4)$$

Equation (3.3) shows that shifting signal  $s$  in the vertex (time) domain multiplies  $\hat{s}$  by the diagonal matrix  $\Lambda$  of the eigenvalues of  $A$ . Similarly, equation (3.4) shows that shifting vector  $\hat{s}$  in the frequency domain multiplies  $s$  by  $\Lambda^*$ , the diagonal matrix of the conjugate eigenvalues of  $A$ .

The DFT of the left side of (3.4) is its right-hand side. Inserting  $\text{DFT}^H \cdot \text{DFT}$  as below, get

$$\underbrace{\text{DFT} \cdot \Lambda^* \cdot \text{DFT}^H}_M \cdot \underbrace{\text{DFT} \cdot s}_{\hat{s}} = \begin{bmatrix} \hat{s}_{N-1} \\ \hat{s}_0 \\ \vdots \\ \hat{s}_{(N-1)-1} \end{bmatrix} = A \cdot \hat{s}. \quad (3.5)$$

The middle vector in (3.5) is of course the shifted (by one)  $\hat{s}$  and the right-hand side equation in (3.5) is the same as the right-hand side equation in (3.4). The ‘surprise’ here is that the left-hand side of (3.5) indicates that the shifted (by one)  $\hat{s}$  is also obtained by multiplying  $\hat{s}$  by a new ‘spectral shift’  $M$ . We readily recognize that in this case  $M = A^*$  and since  $A$  is real the DSP spectral shift is  $M = A$ .

Because it will be important in the sequel, we write together the dual pairs, equation (3.3) that shifts in time, and the equation that shifts in frequency, resulting from combining equa-

tion (3.4) and the left-hand side of (3.5):

$$A \cdot s \xrightarrow{\mathcal{F}} \Lambda \cdot \widehat{s} \quad (3.6)$$

$$\Lambda^* s \xrightarrow{\mathcal{F}} M \cdot \widehat{s}. \quad (3.7)$$

### 3.1.1 Spectral Shift $M$

We define the graph spectral shift  $M$  in GSP so that (3.6) and (3.7) are preserved and remain invariant in GSP.

**Definition 3.1** (GSP: Spectral shift  $M$  [54, 71]). *Let the vertex graph shift  $A$  be diagonalized in (2.13), with  $\Lambda$  the diagonal matrix of eigenvalues, and  $s$ ,  $\widehat{s}$ , and  $y = \Lambda^* \widehat{s}$  be given. Then, the graph spectral shift  $M$  is the operator defined by*

$$\Lambda^* s \xrightarrow{\mathcal{F}} M \cdot \widehat{s}. \quad (3.8)$$

By this definition, the DSP duality (3.6) and (3.7) holds for GSP shifting in the vertex and spectral graph domains.

The next result gives an explicit expression for  $M$ .

**Result 3.1** (GSP: Spectral shift  $M$  [54, 71]). *The shift  $M$  is*

$$M = GFT \cdot \Lambda^* \cdot GFT^{-1}. \quad (3.9)$$

*Proof.* The proof mimics the steps going from (3.4) to (3.5).

**If:** Multiply on the left by  $GFT^{-1}$  the right-hand side of (3.8) and insert  $GFT \cdot GFT^{-1}$  between  $M$  and  $\widehat{s}$

$$GFT^{-1} \cdot M \cdot GFT \cdot GFT^{-1} \cdot \widehat{s}. \quad (3.10)$$

Now replacing  $M$  in (3.10) by its expression in (3.9), canceling terms, and recognizing that

$\text{GFT}^{-1} \cdot \hat{s} = s$ , we get the left-hand side of (3.8) as desired.

**Only if:** Start from definition 3.1. Multiply the left-hand side of (3.8) by  $\text{GFT}$  and insert between  $\Lambda^*$  and  $s$  the product  $\text{GFT}^{-1} \cdot \text{GFT}$ . Get

$$\underbrace{\text{GFT} \cdot \Lambda^* \cdot \text{GFT}^{-1}}_Q \cdot \underbrace{\text{GFT} \cdot s}_{\hat{s}} = M \cdot \hat{s} \quad (3.11)$$

$$Q \cdot \hat{s} = M \cdot \hat{s}. \quad (3.12)$$

Since (3.11) holds for every  $s$ , and so for every  $\hat{s}$ , conclude from (3.12)  $M = Q = \text{GFT} \cdot \Lambda^* \cdot \text{GFT}^{-1}$ , proving the result. ■

Definition 3.1 generalizes  $M$  in DSP as presented in (3.5) since the GFT is the DFT in DSP and the DFT is unitary. While for DSP,  $M = A$ , in GSP  $M$  may not equal  $A$  [54, 71]. The next result addresses when  $M = A$  for GSP.

**Result 3.2** (GSP:  $A = M$ ). *Let  $A$  be real and normal. Then  $A = M$  if  $\text{GFT}^T = \text{GFT}$ .*

*Proof.* For  $A$  normal, real, and GFT symmetric,

$$A = \text{GFT}^* \cdot \Lambda \cdot \text{GFT} = A^* = \text{GFT} \cdot \Lambda^* \cdot \text{GFT}^* = M, \quad (3.13)$$

since  $\text{GFT}^{-1} = \text{GFT}^H$ . This proves the result. ■

Result 3.2 is sufficient for  $A = M$ . If in addition the eigenvalues of  $A$  are nonzero, then we get a necessary condition.

**Result 3.3** (GSP:  $A = M$ ). *Let  $A$  be real, normal, and have nonzero eigenvalues. Then  $A = M$  only if  $\text{GFT}^T = \text{GFT}$ .*

*Proof.* Now,  $M = A = A^*$  only if

$$\text{GFT} \cdot \Lambda^* \cdot \text{GFT}^H = \text{GFT}^H \cdot \Lambda \cdot \text{GFT} = \text{GFT}^T \cdot \Lambda^* \cdot \text{GFT}^* \quad (3.14)$$

Multiply the leftmost and rightmost sides of the equation by  $\text{GFT}^H$  on the left and by  $\text{GFT}$  on the right:

$$\Lambda^* = \text{GFT}^H \cdot \text{GFT}^T \cdot \Lambda^* \cdot \text{GFT}^* \cdot \text{GFT}. \quad (3.15)$$

For all the eigenvalues nonzero, this holds only if

$$\text{GFT}^H \cdot \text{GFT}^T = I = \text{GFT}^* \cdot \text{GFT}. \quad (3.16)$$

Taking the transpose on the left, since  $I^T = I$ , get

$$\text{GFT} \cdot \text{GFT}^* = I = \text{GFT}^* \cdot \text{GFT}, \quad (3.17)$$

which is true only if  $\text{GFT}^{-1} = \text{GFT}^*$ . Since the inverse is unique,  $\text{GFT}^H = \text{GFT}^*$ , implying  $\text{GFT}^T = \text{GFT}$ . ■

We define LSI filtering in the spectral domain.

**Product of signals and LSI spectral filtering.** LSI polynomial spectral filtering in the *spectral* domain is matrix-vector multiplication of a polynomial filter  $P(M)$

$$P(M) = p_0I + p_1M + \cdots + p_{N-1}M^{N-1}, \quad (3.18)$$

with vector  $\hat{s}$

$$\hat{t} = P(M) \cdot \hat{s} = [p_0I + p_1M + \cdots + p_{N-1}M^{N-1}] \cdot \hat{s}. \quad (3.19)$$

**Vertex domain product–spectral convolution.** Consider the duality between product and convolution.

**Result 3.4** (Vertex domain product–spectral convolution).

$$p(\lambda^*) \odot s = P(\Lambda^*) \cdot s \xrightarrow{\mathcal{F}} P(M) \cdot \hat{s}. \quad (3.20)$$

The proof of result 3.4 follows from the eigendecomposition of  $P(M)$  given in (3.9) in result 3.1.

**Graph ‘vertex’ response.** In analogy to the frequency response of a LSI filter  $P(A)$  given in (2.23), we let the “vertex response” of the filter  $P(M)$  to be  $p(\lambda^*)$  defined in (3.20).

**LSI filter  $P_s(M)$  and its vertex response  $s$ .** Similarly to result 2.1 and result 2.2, we determine LSI filter  $P_s(M)$  from “vertex response”  $s$ . Let the coefficients of  $P_s(M)$  be  $p_s = [p_0 \dots p_{N-1}]^T$ .

**Result 3.5** (Matrix  $P_s(M)$ ). *The LSI filter  $P_s(M)$  with vertex response  $s$  is*

$$P_s(M) = GFT \text{diag}[s] GFT^{-1}. \quad (3.21)$$

*Proof.* The result follows from realizing that  $P_s(\Lambda^*) = GFT^{-1}P_s(M)GFT = \text{diag}[s]$ . ■

**Result 3.6** (LSI spectral filter with vertex response  $s$ ). *Under assumption 2.2, the vector  $p_s$  of coefficients of the LSI spectral filter  $P_s(M)$  with vertex response  $s$  is*

$$\mathcal{V}^* \cdot p_s = s. \quad (3.22)$$

where the Vandermonde matrix  $\mathcal{V}$  is given in (2.27).

**Spectral convolution.** To interpret steps in sampling in the spectral domain, we define convolution of two spectral signals. Let  $\otimes$  represent (circular) convolution. The next result tells us how to compute convolution.

**Result 3.7** (Spectral convolution of two signals). *Consider spectral signals  $\hat{s}$  and  $\hat{t}$  and their*

corresponding LSI filters  $P_s(M)$  and  $P_t(M)$ , where  $M$  is the spectral shift. Then

$$\hat{u} = \hat{s} \circledast \hat{t} = P_s(M) \cdot \hat{t} = P_t(M) \cdot \hat{s} \quad (3.23)$$

$$\hat{u} = \hat{s} \circledast \hat{t} = P_s(M)P_t(M) \cdot \hat{\delta}_{sp,0}^{flat} \quad (3.24)$$

$$\hat{u} = \hat{s} \circledast \hat{t} \xleftarrow{\mathcal{F}} u = s \odot t \quad (3.25)$$

where we define the spectral domain impulse  $\hat{\delta}_{sp,0}^{flat} \xleftrightarrow{\mathcal{F}} \delta_{sp,0}^{flat} = \frac{1}{\sqrt{N}}1$ , to be flat (constant) in the vertex domain (See next Section 3.2 for a detailed discussion on impulse functions).

Equation (3.23) shows spectral convolution of  $\hat{s}$  and  $\hat{t}$  as filtering of  $\hat{s}$  (or  $\hat{t}$ ) with LSI filter  $P_t(M)$  (or  $P_s(M)$ ), while (3.24) shows spectral convolution of the two signals as the *impulse response* of the LSI filter  $P_s(M)P_t(M)$ . Equation (3.25) shows spectral convolution in the spectral domain as pointwise multiplication in the vertex domain.

### 3.1.2 Vertex Shift $A$ and Graph Spectral Shift $M$ : Equivariance

For graph  $G = (V, E)$ , adjacency  $A$  is defined up to a relabelling of the vertices by a permutation  $\Pi_1$ . Denote a quantity with respect to a new relabeling by  $(\cdot)'$ . Then

$$A' = \Pi_1 \cdot A \cdot \Pi_1^T \quad (3.26)$$

$$s' = \Pi_1 \cdot s. \quad (3.27)$$

To see how  $\Pi_1$  affects the graph spectral shift  $M$ , we first consider how  $\Pi_1$  impacts the GFT and  $\Lambda$ .

Eigendecomposing  $A'$ , from (2.13), get two forms

$$A' = \Pi_1 \cdot \text{GFT}^{-1} \cdot \Pi_2^H \cdot \Pi_2 \cdot \Lambda \cdot \Pi_2^H \cdot \Pi_2 \cdot \text{GFT} \cdot \Pi_1^T \quad (3.28)$$

$$= \Pi_1 \cdot \text{GFT}^{-1} \cdot \Lambda \cdot \text{GFT} \cdot \Pi_1^T. \quad (3.29)$$

where  $\Pi_2$  is unitary. These lead to two alternatives

$$\text{GFT}' = \Pi_2 \cdot \text{GFT} \cdot \Pi_1^T \quad (3.30)$$

$$\text{GFT}'' = \text{GFT} \cdot \Pi_1^T, \quad (3.31)$$

with  $\Lambda$  given by either one of the two:

$$\Lambda' = \Pi_2 \cdot \Lambda \cdot \Pi_2^H \quad (3.32)$$

$$\Lambda'' = \Lambda. \quad (3.33)$$

In (3.32),  $\Pi_2$  must be a permutation matrix to keep  $\Lambda'$  diagonal with the same eigenvalues as  $\Lambda$ . For each of the two definitions of the GFT, the graph Fourier transform  $\hat{s}$  of  $s$  after relabeling with  $\Pi_1$

$$\hat{s}' = \text{GFT}' \cdot s' = \Pi_2 \cdot \text{GFT} \cdot \Pi_1^T \cdot \Pi_1 \cdot s = \Pi_2 \cdot \hat{s} \quad (3.34)$$

$$\hat{s}'' = \text{GFT}'' \cdot s' = \text{GFT} \cdot \Pi_1^T \cdot \Pi_1 \cdot s = \hat{s}. \quad (3.35)$$

The first definition permutes the graph Fourier transform  $\hat{s}$  by  $\Pi_2$ . The second leaves  $\hat{s}$  invariant. The question is which GFT should be adopted: (3.30) or (3.31).

To resolve this, we look at a simple DSP example. Consider, for example,  $N = 3$ ,  $s = [s_0 \ s_1 \ s_2]^T$ , and a circular shift of the nodes to get  $s' = [s_2 \ s_0 \ s_1]^T$ . If we use  $\text{GFT}''$  and  $\Lambda'' = \Lambda$  from (3.31) and (3.33),

$$\Lambda^* \cdot \Pi_1 \cdot s = \begin{bmatrix} 1 & & \\ & e^{-j\frac{2\pi}{3}} & \\ & & e^{-j\frac{2\pi}{3}2} \end{bmatrix} \cdot \begin{bmatrix} s_2 \\ s_0 \\ s_1 \end{bmatrix} = \begin{bmatrix} s_2 \\ e^{-j\frac{2\pi}{3}} s_0 \\ e^{-j\frac{2\pi}{3}2} s_1 \end{bmatrix}. \quad (3.36)$$

In (3.36), the time samples are multiplied by the wrong phase shift, for example, time sample  $s_2$  is multiplied by 1 instead of  $e^{-j\frac{2\pi}{3}2}$ . We now consider computing (3.2), using  $s'$  from (3.27),

GFT' from (3.30) and  $\Lambda'$  from (3.32). We get

$$\Pi_2 \cdot \Lambda^* \cdot \Pi_2^T \cdot \Pi_1 \cdot s = \begin{bmatrix} e^{-j\frac{2\pi}{3}2} & & \\ & 1 & \\ & & e^{-j\frac{2\pi}{3}} \end{bmatrix} \cdot \begin{bmatrix} s_2 \\ s_0 \\ s_1 \end{bmatrix} = \begin{bmatrix} e^{-j\frac{2\pi}{3}2} s_2 \\ s_0 \\ e^{-j\frac{2\pi}{3}} s_1 \end{bmatrix}^T. \quad (3.37)$$

For the right hand side in (3.37) to be the correct shifts as shown,  $\Pi_2^T \Pi_1$  needs to cancel and  $\Pi_2 = \Pi_1$ . We conclude that, after relabeling the vertices of the graph by  $\Pi_1$ , the GFT should be given by (3.30) and not by (3.31), with  $\Pi_1 = \Pi_2$ . The eigenvalue matrix  $\Lambda$  should also then be permuted as in (3.32), with  $\Pi_1 = \Pi_2$ . Also, when  $\Pi_1 = \Pi_2 = I$ , we obtain  $A' = A$  and the original eigendecomposition of  $A$ .

We can now determine how relabeling nodes impacts  $M$ .

**Result 3.8** (Equivariance to permutation). *When nodes of  $G$  are permuted by  $\Pi$ , shifts  $A$  and  $M$  are conjugated*

$$A' = \Pi \cdot A \cdot \Pi^T \text{ and } M' = \Pi \cdot M \cdot \Pi^T. \quad (3.38)$$

*Proof.* The equivariance of  $A$  to permutation was already proven in [1]. We now consider the equivariance of  $M$ .

From definition 3.1, the action of  $M$  on  $\hat{s}$  is the vector  $y = \Lambda^* \cdot s$  in the vertex domain. This spectral shifting property multiplies the vertex domain component  $s_n$  of  $s$  by the conjugate of the graph frequency  $\lambda_n$ . If we reshuffle the labeling of the nodes by  $\Pi$ , then  $\Lambda^*$  is conjugated by  $\Pi$ , i.e.,  $\Lambda^*$  is given by (3.32), in order to preserve the spectral shifting property. This forces the graph Fourier transform GFT to also be conjugated by  $\Pi$  as given by (3.30). Similarly, we can conclude that  $\text{GFT}^{-1}$  is conjugated by  $\Pi$ . Putting these together leads to the equivariance of  $M$  to permutation  $\Pi$  as asserted by the result. ■

Result 3.8 is pleasing, it shows that  $A$  and  $M$  are impacted similarly: both are equivariant to  $\Pi$ .

**Remark 3.1** (Scrambling vertex and spectral domains). *In DSP, time and frequencies are usually implicitly ordered. This is so natural that DSP seldom explicitly discusses the indexing of the time samples or the indexing of the Fourier coefficients: the signal  $(s_0, s_1, \dots, s_{N-1})$  is an ordered  $N$ -tuple, exactly like  $(\widehat{s}_0, \widehat{s}_1, \dots, \widehat{s}_{N-1})$  is an ordered  $N$ -tuple. So, it may seem strange that in GSP one needs to share the ordering adopted for the vertices of the graph and/or for the graph frequencies. But actually this should not surprise us. Even in DSP, there are applications where it is useful to permute signal samples either in time or frequency. One such early technique for securing voice communication used scrambling [72, 73]. In simple terms, speech samples are scrambled to change their order. At the receiver, a descrambling block is required to reorder the speech samples. Other secure communications use scrambling in the frequency domain, or in other transform domains. This is to illustrate that although time signals and their spectra are naturally ordered by the time and frequency indices, reordering or permuting the samples in time or frequency have found applications in DSP. The important point is that scrambling requires then a descrambling block. In other words, the transmitter and receiver have a way to share their labeling scheme of the signal samples or of the spectral samples. Likewise, in GSP, different researchers working with the same graph need to share their node labeling to make sense of the graph signal. Likewise, if they share the graph spectrum, they must also share their labeling of graph frequencies to know the  $GFT^{-1}$  that inverts the spectrum<sup>1</sup>.*

### 3.1.3 GSP<sub>sp</sub>: Dual Graph Signal Processing

The shifts  $A$  and  $M$  play twin or dual roles; just as GSP is built from  $A$ , we build a dual GSP<sub>sp</sub> from  $M$ .

*Data and spectral graphs.* As adjacency matrices,  $A$  and  $M$  define graphs: shift  $A$  determines the (data) graph  $G$  whose node  $n$  indexes the data sample  $s_n$ , while the spectral shift  $M$

---

<sup>1</sup>Note that the graph and values defined on the graph do not depend on indexing. These entities exist independently of the indexing. It is only when researchers use matrices (e.g., the adjacency matrix) to represent the graph and vectors (e.g., the graph signal) to represent graph values that they need to choose an indexing.

defines a new graph, the *spectral graph*  $G_{\text{sp}} = (V_{\text{sp}}, E_{\text{sp}})$ , whose node  $m$  (the graph frequency  $\lambda_m$ ,  $m = 0 \cdots N - 1$ ) indexes the graph Fourier coefficient  $\hat{s}_m$  of the data. As shown at the beginning of this section 3.1, in DSP,  $G$  and  $G_{\text{sp}}$  are cycle graphs—the time samples are indexed by the time ticks (vertices of  $G$ ), the spectral coefficients are indexed by the frequencies (vertices of  $G_{\text{sp}}$ ).

$GFT_{\text{sp}}$ : *Spectral GFT*. Since  $M$  is diagonalized in (3.9),

$$GFT_{\text{sp}} = GFT^{-1}. \quad (3.39)$$

The spectral  $GFT_{\text{sp}}$  of the spectral graph signal  $\hat{s}$  is

$$\hat{\hat{s}} = GFT_{\text{sp}} \cdot \hat{s} = GFT^{-1} \cdot \hat{s} = s. \quad (3.40)$$

*LSI (linear shift invariant) spectral filters*: Since  $M$  is diagonalizable, LSI spectral filters  $P_{\text{sp}}(M)$  are polynomials<sup>2</sup>, see equation (3.18) and result 3.4.

**Example 3.1** (Star graph). Consider  $A$  for the undirected star graph

$$A = \begin{bmatrix} 0 & 1_{N-1} \\ 1_{N-1}^T & 0_{(N-1)(N-1)} \end{bmatrix} \quad (3.41)$$

Its eigenvalues are  $\pm\sqrt{N-1}$  with multiplicity 1 and 0 with algebraic and geometric multiplicities  $N-2$ . The  $GFT$

$$GFT = \frac{1}{\sqrt{N-1}} \begin{bmatrix} \frac{\sqrt{N-1}}{\sqrt{2}} & -\frac{\sqrt{N-1}}{\sqrt{2}} & 0 & \dots & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 1 & \dots & 1 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & e^{-j\frac{2\pi}{N-1}} & \dots & e^{-j\frac{2\pi(N-2)}{N-1}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & e^{-j\frac{2\pi(N-2)}{N-1}} & \dots & e^{-j\frac{2\pi(N-2)^2}{N-1}} \end{bmatrix} \quad (3.42)$$

$$GFT_{\text{sp}} = GFT^{-1} = GFT^H \quad (3.43)$$

---

<sup>2</sup>In the sequel, we will usually ignore the subindexing sp.

This GFT diagonalizes  $A$ . To develop it, note that the characteristic polynomial of  $A$  is  $\Delta(\lambda) = \lambda^{N-2}(\lambda - (N - 1))^2$  and that the eigenvectors of  $A$  are the same as  $A^2$ . Matrix  $A^2$  is block diagonal, with a scalar minor  $N - 1$  and a matrix minor  $1_{N-1} \cdot 1_{N-1}^T$ . The latter is circulant and diagonalized by  $DFT_{N-1}$ . The spectral shift<sup>3</sup> for the star graph follows, shown for  $N = 5$  in Figure 3.1:

$$M = \frac{1}{\sqrt{N-1}} \begin{bmatrix} \frac{N}{2} \begin{bmatrix} 1 & \\ & -1 \end{bmatrix} \begin{bmatrix} 1 & -1 \end{bmatrix} - I_2 & -\frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes 1_{N-2}^T \\ -\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \end{bmatrix} \otimes 1_{N-2} & -1_{N-2} \cdot 1_{N-2}^T \end{bmatrix} \quad (3.44)$$

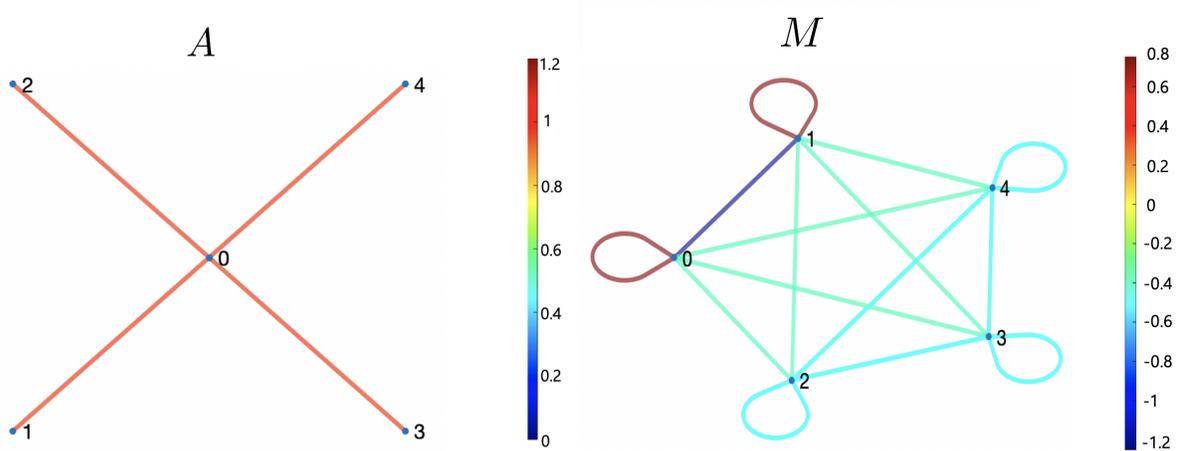


Figure 3.1: Star graph: Shifts  $A$  and  $M$  for  $N = 5$ .

## 3.2 Graph Impulse

When studying graph signal representations, we need the concept of graph delta or graph impulse. In DSP, the impulse in the time domain and its Fourier transform are

$$\delta_{t,0} = e_0, \quad \mathcal{F} \rightarrow \widehat{\delta}_{t,0} = \text{DFT } \delta_{t,0} = \frac{1}{\sqrt{N}} \mathbf{1}, \quad (3.45)$$

<sup>3</sup>Since the ordering of the nodes is specified, the GFT and the spectral shift  $M$  are fixed.

where  $\mathbf{1}$  is the vector of ones and  $e_0 = [1, 0, \dots, 0]^T$ . In DSP, the time impulse  $\delta_{t,0}$  is impulsive in the vertex domain (nonzero only at 0) and flat in the frequency domain. Further, the delayed time impulses

$$\delta_{t,n} = A_c^n \delta_{t,0} = e_n$$

are centered at  $n$  and impulsive. Likewise, in DSP, the impulse in the frequency domain  $\widehat{\delta}_{f,0}$  is impulsive now in frequency and flat in time. In other words, in DSP, the definition of impulse in time and frequency are symmetric—the time and frequency impulses are impulsive at  $t = 0$  and at  $f = 0$ , respectively.

In GSP, in general, we either get impulsivity in one domain or flatness in the other, but not both. We have then two possible definitions for the vertex impulse and two possible definitions for the spectral impulse. We choose to preserve flatness and define delta graph signals that are flat in one domain in this thesis (See Appendix 11.1 for a discussion on impulsive delta graph signals). We discuss next how to define 1) graph impulse signal in the vertex domain as the inverse GFT of a flat signal in the spectral domain; and 2) graph impulse signal in the spectral domain as the GFT of a flat signal in the vertex domain.

### 3.2.1 Vertex Graph Impulse

In the vertex domain, define the vertex impulse or delta  $\delta_0$  as the inverse GFT of a flat graph spectrum

$$\delta_0 \xrightarrow{\mathcal{F}} \widehat{\delta}_0 = \frac{1}{\sqrt{N}} \mathbf{1} \implies \delta_0 \triangleq \text{GFT}^{-1} \left[ \frac{1}{\sqrt{N}} \mathbf{1} \right]. \quad (3.46)$$

The shifted replicas of the vertex graph impulse  $\delta_0$  are

$$\delta_n = A^n \delta_0 \xrightarrow{\mathcal{F}} \widehat{\delta}_n = \Lambda^n \frac{1}{\sqrt{N}} \mathbf{1} = \frac{1}{\sqrt{N}} \lambda^n. \quad (3.47)$$

In GSP, the  $\delta_n$ 's, delayed  $\delta_0$  by  $A^n$ , are not impulsive.

**Theorem 3.1.** (*Synthesis of signal  $s$  using the vertex graph impulse*) Under assumption 2.2

of unique eigenvalues, any signal  $s$  can be written as a matrix vector product,  $P(A)\delta_0$ .

*Proof.* Take the GFT of  $P(A)\delta_0$ .

$$P(A)\delta_0 \xrightarrow{\text{GFT}} \text{GFT } P(A)\delta_0 = P(\Lambda)\widehat{\delta}_0 = P(\Lambda)\frac{1}{\sqrt{N}}\mathbf{1} = \frac{1}{\sqrt{N}}P(\Lambda)\odot\mathbf{1} = \frac{1}{\sqrt{N}}P(\Lambda).$$

We need to find the polynomial that satisfies

$$\frac{1}{\sqrt{N}}P(\Lambda) = \text{GFT } s = \widehat{s}.$$

Absorb the  $\frac{1}{\sqrt{N}}$  into  $P(\Lambda)$ . Let the coefficients of the polynomial be  $p$ . Then,  $p$  is the solution to the system

$$\mathcal{V}p = \widehat{s}$$

where  $\mathcal{V}$  is the Vandermonde matrix in (2.27). Under assumption 2.2 of unique eigenvalues,  $\mathcal{V}$  is invertible and thus,  $p$  always exists. Thus, every signal  $s$  can be written as a matrix vector product,

$$s = P(A)\delta_0.$$

■

### 3.2.2 Spectral Graph Impulse

We now consider the spectral graph impulse  $\widehat{\delta}_{\text{sp},0}$  in the spectral domain. We define it as the GFT of a flat signal in the vertex domain

$$\delta_{\text{sp},0} = \frac{1}{\sqrt{N}}\mathbf{1} \xrightarrow{\mathcal{F}} \widehat{\delta}_{\text{sp},0} \implies \widehat{\delta}_{\text{sp},0} \triangleq \text{GFT} \left[ \frac{1}{\sqrt{N}}\mathbf{1} \right]. \quad (3.48)$$

The shifts of  $\widehat{\delta}_{\text{sp},0}$  in the spectral domain are obtained with the spectral shift  $M$ . Replicating (3.47), get

$$\delta_{\text{sp},n} = \Lambda^{*n} \frac{1}{\sqrt{N}} \mathbf{1} = \frac{1}{\sqrt{N}} \lambda^{*n} \xrightarrow{\mathcal{F}} \widehat{\delta}_{\text{sp},n} = M^n \widehat{\delta}_{\text{sp},0}. \quad (3.49)$$

**Theorem 3.2.** (*Synthesis of signal  $\widehat{s}$  using the spectral graph impulse*) Under assumption 2.2 of unique eigenvalues, any signal  $\widehat{s}$  can be written as a matrix vector product,  $\widehat{s} = P(M)\widehat{\delta}_{\text{sp},0}$ .

*Proof.* This is a dual proof to the proof of theorem 3.1. Take the  $\text{GFT}^{-1}$  of  $P(M)\widehat{\delta}_{\text{sp},0}$ .

$$P(M)\widehat{\delta}_{\text{sp},0} \xrightarrow{\text{GFT}^{-1}} \text{GFT}^{-1} P(M)\widehat{\delta}_{\text{sp},0} = P(\Lambda^*)\widehat{\delta}_0 = P(\Lambda^*)\frac{1}{\sqrt{N}}\mathbf{1} = \frac{1}{\sqrt{N}}P(\lambda^*) \odot \mathbf{1} = \frac{1}{\sqrt{N}}P(\lambda^*).$$

We need to find the polynomial that satisfies

$$\frac{1}{\sqrt{N}}P(\lambda^*) = \text{GFT}^{-1}\widehat{s} = s.$$

Absorb the  $\frac{1}{\sqrt{N}}$  into  $P(\lambda^*)$ . Let the coefficients of the polynomial be  $q$ .  $q$  is the solution to the system

$$\mathcal{V}^*q = s$$

where  $\mathcal{V}$  is the Vandermonde matrix in (2.27). Under assumption 2.2 of unique eigenvalues,  $\mathcal{V}^*$  is invertible and thus,  $q$  always exists. Thus, every signal  $\widehat{s}$  can be written as a matrix vector product,

$$\widehat{s} = P(M)\widehat{\delta}_{\text{sp},0}.$$

■

**Remark 3.2** (Notation on vertex and spectral quantities). *When referring to quantities using the shift  $A$  or the vertex impulse  $\delta_0$  we will often not qualify them with the word ‘‘vertex.’’ In contrast, we will consistently qualify by ‘‘spectral’’ quantities related to the spectral shift  $M$  or the spectral impulse  $\delta_{\text{sp},n}$  using the subscript ‘sp’ as a reminder.*

**Remark 3.3** (Vertex and spectral impulses). *We emphasize that we have two graph impulses, the vertex impulse  $\delta_0$  (that is flat in the spectral domain, see (3.46)) and the spectral impulse  $\delta_{sp,n}$  (that is flat in the vertex domain, see (3.48)). In general, in GSP, neither is actually “impulsive” in either domain.*

### 3.3 Conclusion

In this chapter, we presented a spectral graph signal processing theory,  $\text{GSP}_{\text{sp}}$ , the dual of vertex based GSP. Using the DSP properties of shifting a signal in frequency domain, we defined a spectral shift  $M = \text{GFT} \Lambda^* \text{GFT}^{-1}$ , dual to the vertex shift  $A$ . From there, we developed convolution in the spectral domain as the matrix-vector multiplication between a polynomial of the spectral shift,  $P(M)$ , and the spectral graph signal  $\hat{s}$ . In the second section, we introduced GSP graph impulses in both the vertex and spectral domain. In DSP, delta functions are impulsive in one domain and flat in the other. In GSP, delta functions are either impulsive in one domain or flat in the other, but not both. In this thesis, we define delta functions in vertex domain as flat in the spectral domain. Simialrly, spectral delta functions are flat in the vertex domain.

# Chapter 4

## Dual Domain GSP Sampling

This chapter presents a unified graph signal sampling theory with GSP vertex and spectral domain dual interpretations. Recently, [40] reviews comprehensively the existing methods for graph sampling. According to [40], “two definitions [of graph sampling] can be possible . . .” and current theories are either in the vertex or in the graph spectral domain, with no simple analogy between them. This stands in contrast with the *dualism* between time and frequency domain approaches to DSP. This chapter deals with this open issue.

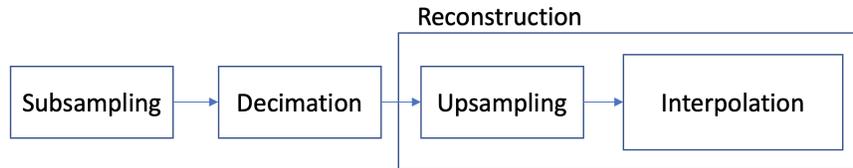


Figure 4.1: The sampling and reconstruction steps in this thesis. Each step is considered in both vertex and spectral domains.

We assume that the sampling set  $S$  is given. In DSP, this is similar to assuming a particular sampling scheme, say, uniform sampling (keeping every  $K$ th sample). In DSP, the following *duality* holds: 1) subsampling in the frequency domain is linear shift invariant filtering leading to spectral replication, and 2) reconstruction is by ideal lowpass filtering. Likewise, extending the DSP sampling framework to GSP sampling in a natural way, we show similar dualism: 1) GSP subsampling in the vertex domain is LSI filtering in the spectral domain, leading to spectral replication, and 2) when is GSP reconstruction achieved by LSI filtering. To develop this, we use the spectral shift  $M$ , the spectral graph  $G_{\text{sp}}$ , spectral GSP filtering, and other concepts from Chapter 3, and we show which choices among alternatives in GSP replicate

Shannon-Nyquist sampling in DSP.

## 4.1 GSP Sampling: Subsampling and Decimation

In DSP, sampling of bandlimited signals  $s$  include: 1) *Subsampling* that zeroes samples of  $s$  (e.g., every other sample) to get a subsampled signal  $s_\delta$ ; 2) *decimating* or *downsampling* that discards zeroed samples in  $s_\delta$  to get the decimated signal  $s_d$ ; and 3) *reconstruction* that 3.i) *up-samples*  $s_d$  by reinserting the zeros discarded to get back the upsampled signal  $s_\delta$ , and 3.ii) *interpolates* by ideal *lowpass filtering*  $s_\delta$  to get<sup>1</sup>  $s_r = s$ . All these steps have interpretations in both time and frequency.

Likewise, we consider the equivalent sequence of steps in GSP sampling—*subsampling*, *decimation*, and *reconstruction*—and develop for each step dual interpretations in the vertex and graph spectral domains. To achieve these dual interpretations, we use  $\text{GSP}_{\text{sp}}$  from Chapter 3. Further, we show explicitly which choices among alternatives have to be made in GSP sampling to obtain DSP sampling when the graph is the cyclic graph. These choices are implicit and taken for granted in DSP. We provide further insights and interpretations into DSP.

In this and the next section,  $A$  is the adjacency matrix of a  $N$  node arbitrary graph. Let  $\|\widehat{s}\|_0$  be the  $\ell_0$  pseudo-norm, i.e., the number of nonzero entries of  $\widehat{s}$ . Let  $s$  be bandlimited with bandwidth  $K$ , i.e.,  $\|\widehat{s}\|_0 \leq K$ ,  $K \leq N$ . For ease of notation, we assume the last  $N - K$  entries of  $\widehat{s}$  are zero, i.e.,  $\widehat{s} = [\widehat{s}_K^T \ \widehat{s}_{N-K}^T]^T$ , where  $\widehat{s}_{N-K} = 0$ ,<sup>2</sup> and that  $K$  divides  $N$ ,  $K|N$ .

This section considers first the sampling set in subsection 4.1.1, subsampling in subsection 4.1.2, and decimation in subsection 4.1.3. In all these sections, we provide vertex and spectral domain GSP interpretations that parallel DSP.

---

<sup>1</sup> If sampling below the Nyquist rate,  $s_r$  in 3.ii) is an aliased version of  $s$ .

<sup>2</sup> In actuality, the zero entries can occur anywhere in  $\widehat{s}$ . In this case, we are assuming that we reorder  $\widehat{s}$  such that its last  $N - K$  entries are 0.

### 4.1.1 Sampling Set $S$

We refer to section 1.1 that reviews the significant work in defining the sampling set  $S$ . In our context, with  $K|N$  and finite graphs, the sampling set  $S$  [19, 23, 74] is the minimum set of vertices indexing the signal samples that enables perfect reconstruction of bandlimited signals from the corresponding decimated signal  $s_d$ .

**Choice of sampling set  $S$ :** We assume that the graph signal  $s$  is lowpass and bandlimited to  $K$  and  $K|N$ . To fix notation, we briefly describe one method to determine the sampling set  $S$ , or its characteristic graph signal  $\delta^{(spl)}$ , a vector of zeros and ones. When  $s$  is bandlimited, knowing  $s$  at the vertices in  $S$  allows for perfect reconstruction of  $s$ . This is the decimated or downsampled version  $s_d$  of  $s$ .

**Result 4.1** (Sampling set). *With the notation and assumptions above, let graph  $G = (V, E)$ ,  $|V| = N$  whose nodes index bandlimited graph signals  $s$  with bandwidth  $K$ ,  $K|N$ . Then, there is a sampling set  $S$  with cardinality  $K$  and indicator signal  $\delta^{(spl)}$  such that  $s$  is perfectly reconstructed from its samples indexed by vertices in  $S$ .*

*Proof.* The proof can be found in [19, 20]. In this chapter, we consider one method of finding a sampling set and comment on its nonuniqueness.

We start with the Fourier relation between the signal  $s$  and its graph Fourier transform  $\widehat{s}$  and block partition rowwise the GFT matrix as indicated below:

$$\text{GFT } s = \widehat{s} \implies \boxed{\begin{bmatrix} \text{GFT}_K \\ \text{GFT}_{N-K} \end{bmatrix} s = \begin{bmatrix} \widehat{s}_K \\ \widehat{s}_{N-K} \end{bmatrix}} \quad (4.1)$$

with the top  $K$  rows of the GFT in  $\text{GFT}_K : K \times N$  and the bottom  $N - K$  rows in  $\text{GFT}_{N-K} : (N - K) \times N$ . Given that GFT is full rank,  $\text{GFT}_K$  and  $\text{GFT}_{N-K}$  are full rank.

Taking  $\widehat{s}_{N-K} = 0$  in (4.1), we get

$$\text{GFT}_{N-K} s = \widehat{s}_{N-K} = 0. \quad (4.2)$$

Solving (4.2) determines  $N - K$  components of  $s$  (so called pivot variables) in terms of the other  $K$  components (so called free variables). There are many alternative possible sets of  $N - K$  pivots and  $K$  free variables, i.e., this split is not unique. There are also different ways to determine these sets. We illustrate with Gauss Elimination (GE) as just one method.

GE determines  $N - K$  linearly independent rows and columns of  $\text{GFT}_{N-K}$ , reducing it to row echelon form:

$$\text{GFT}_{N-K}s = \widehat{s}_{N-K} \xrightarrow{\text{GE}} E \cdot \text{GFT}_{N-K} \Pi_{\text{col}}^T \cdot \Pi_{\text{col}} s = 0. \quad (4.3)$$

In (4.3),  $E$  represents the row operations that reduce the  $\text{GFT}_{N-K}$  to row echelon form. Matrix  $E : (N - K) \times (N - K)$  is the product of elementary matrices and so it is full rank. Partition the row echelon form of  $\text{GFT}_{N-K}$  as

$$E \cdot \text{GFT}_{N-K} \Pi_{\text{col}}^T = [B_{11} \ B_{12}], \quad (4.4)$$

with  $B_{11} : (N - K) \times (N - K)$  upper triangular with ones on the diagonal, and  $B_{12} : (N - K) \times K$ . The matrix  $\Pi_{\text{col}}$  is a permutation representing possible column swapping.

Let

$$\Pi_{\text{col}} s = \begin{bmatrix} s_{N-K} \\ s_K \end{bmatrix}. \quad (4.5)$$

Replacing (4.4) in (4.3) and using the partitioning of  $\Pi_{\text{col}} s$  in (4.5), equation (4.3) becomes:

$$[B_{11} \ B_{12}] \begin{bmatrix} s_{N-K} \\ s_K \end{bmatrix} = 0. \quad (4.6)$$

Since  $B_{11}$  is invertible, (4.6) leads to

$$s_{N-K} = -B_{11}^{-1} \cdot B_{12} \cdot s_K. \quad (4.7)$$

This determines  $s_{N-K}$  from  $s_K$ . The vector  $s_{N-K}$  collects the  $N - K$  pivot entries and  $s_K$  collects the  $K$  free variables. This shows that, given the free variables  $s_K$ , we recover:

$$\Pi_{\text{col}S} = \begin{bmatrix} -B_{11}^{-1} \cdot B_{12} \\ I_K \end{bmatrix} s_K. \quad (4.8)$$

With  $S$  the set of indices of the free variables  $s_K$  ( $|S| = K$ ) and  $\delta^{(\text{spl})}$  its indicator signal, the result follows. ■

**Remark 4.1** ( $S$  not unique). *Applying GE to (4.2), we can permute rows and columns of  $GFT_{N-K}$ , leading to different choices of pivots and free variables. Hence,  $S$  and  $\delta^{(\text{spl})}$  are not unique. Regardless of the choice for  $S$ ,  $|S| = \|\delta^{(\text{spl})}\|_0 = K$  equals the number of degrees of freedom in (4.2) and the bandwidth of  $s$ .*

**Result 4.2** ( $S$  and  $\delta^{(\text{spl})}$ ). *Under the set-up of result 4.1, given the sampling set  $S$ , the sampling signal  $\delta^{(\text{spl})}$  is unique and the signal samples indexed by  $S$  uniquely determine  $s$ .*

This result is of course tautologic since  $\delta^{(\text{spl})}$  is the characteristic signal of  $S$  and (4.8) shows how to recover  $s$  from  $s_K$ . We make it explicit for easy future reference that the degrees of freedom are in choosing  $S$ . Once chosen, the sampling signal is fixed and  $s$  is uniquely determined.

### 4.1.2 GSP Subsampling by LSI Filtering

This section shows that GSP and DSP subsampling have equivalent vertex and spectral domain dual interpretations.

Assume a sampling set  $S$  and its sampling graph signal  $\delta^{(\text{spl})}$  have been chosen. In DSP, uniform ideal subsampling  $s$  in the vertex domain is multiplication of  $s$  by a train of pulses

$\delta^{(\text{spl})}$ . In the spectral domain, it is convolution (or LSI filtering) by a periodic train of pulses. We now discuss subsampling in GSP.

Let  $G = (V, E)$ , with shift  $A$  and spectral shift  $M$ . Let  $s_\delta$  be the subsampled graph signal obtained from  $s$ . Its  $K$  nonzero entries are the entries indexed by vertices in  $S$ .

**Result 4.3** (GSP subsampling as LSI filter). *Under assumption 2.2, GSP subsampling in the spectral domain is LSI filtering*

$$s_\delta = \delta^{(\text{spl})} \odot s \xrightarrow{\mathcal{F}} P_{\delta^{(\text{spl})}}(M) \cdot \widehat{s} = \widehat{s}_\delta \quad (4.9)$$

*Proof.* Subsampling is pointwise multiplication

$$s_\delta = \delta^{(\text{spl})} \odot s \quad (4.10)$$

in the vertex domain, which is the left-hand side of (4.9).

To show it is LSI filtering in the spectral domain, we need to show that the GFT of  $s_\delta$  is obtained by polynomial filtering  $\widehat{s}$ . By result 3.7 and equation (3.25), pointwise multiplication of  $\delta^{(\text{spl})}$  and  $s$  in the vertex domain is convolution in the graph spectral domain. By (3.23), the convolution is filtering  $\widehat{s}$  with the LSI polynomial filter  $P_{\delta^{(\text{spl})}}(M)$ . We only need to show that, given the sampling signal  $\delta^{(\text{spl})}$ ,  $P_{\delta^{(\text{spl})}}(M)$  is well defined.

By result 3.5 and equation (3.21),  $P_{\delta^{(\text{spl})}}(M)$  is given by

$$P_{\delta^{(\text{spl})}}(M) = \text{GFT} \text{diag} [\delta^{(\text{spl})}] \text{GFT}^{-1}, \quad (4.11)$$

with coefficients  $p_{\delta^{(\text{spl})}}$  given by equation (3.22):

$$\mathcal{V}^* \cdot p_{\delta^{(\text{spl})}} = \delta^{(\text{spl})}. \quad (4.12)$$

By assumption 2.2 of distinct eigenvalues, the Vandermonde matrix  $\mathcal{V}$  is full rank and (4.12)

has a unique solution:

$$p_{\delta^{(\text{spl})}} = \mathcal{V}^{*-1} \delta^{(\text{spl})} \quad (4.13)$$

Hence,  $p_{\delta^{(\text{spl})}}$  and  $P_{\delta^{(\text{spl})}}(M)$  are well defined, proving the result.  $\blacksquare$

In DSP, it is well known that the spectrum of the subsampled signal  $s_\delta$  is the nonzero spectrum  $\widehat{s}_K$  of  $s$  replicated  $N - K$  times. In other words, in the spectral domain DSP subsampling is convolution with a train of equispaced spectral pulses. We now wish to show that the graph spectrum  $\widehat{s}_\delta$  of the subsampled signal  $s_\delta = \delta^{(\text{spl})} \odot s$  is given by (possibly filtered) copies of the nonzero spectrum of  $s$ , i.e.,  $\widehat{s}_K$ . At first sight, this is not obvious. In fact, the spectral domain LSI filter  $P_{\delta^{(\text{spl})}}(M)$  with coefficients  $p_{\delta^{(\text{spl})}} = [p_0 \ p_1 \ \cdots \ p_{N-1}]^T$  given by (4.12) is

$$P_{\delta^{(\text{spl})}}(M) = p_0 I + p_1 M + \cdots + p_{N-1} M^{N-1}. \quad (4.14)$$

Then,

$$\widehat{s}_\delta = (p_0 I + p_1 M + \cdots + p_{N-1} M^{N-1}) \begin{bmatrix} \widehat{s}_K \\ 0_{N-K} \end{bmatrix}. \quad (4.15)$$

This shows  $\widehat{s}_\delta$  is a superposition of replicas of  $\widehat{s}_K$ , which could overlap. We show this is not the case, if the bandlimited graph signal is sampled at graph Nyquist rate  $K$ .

**Result 4.4** (Replication: GSP spectrum of subsampled  $s_\delta$ ). *The spectrum  $\widehat{s}_\delta$  of the subsampled  $s_\delta$  corresponds to  $\frac{N}{K}$  (filtered, possibly distorted) copies of the nonzero spectrum  $\widehat{s}_K$  of  $s$ . Sampling at the “graph” Nyquist rate  $K$ , aliasing does not occur.*

We start with preliminary notation before the proof. Assume the sampling set  $S$  has been chosen with given sampling graph signal  $\delta^{(\text{spl})}$ . Without loss of generality, to make the

presentation easier, assume reordering the vertices of the graph by permutation  $\Pi$  so that

$$\delta^{(\text{spl})} = [1_K^T 0_{N-K}^T]^T. \quad (4.16)$$

Permutation  $\Pi$  conjugates  $A$ , GFT, and  $\text{GFT}^{-1}$ . We will ignore  $\Pi$ . Partition GFT,  $\text{GFT}^{-1}$ , and  $P_{\delta^{(\text{spl})}}(M)$ :

$$\text{GFT} = \begin{bmatrix} \text{GFT}_K & \text{GFT}_{N-K} \end{bmatrix} \quad (4.17)$$

$$= \begin{bmatrix} \text{GFT}_{KK} & \text{GFT}_{K(N-K)} \\ \text{GFT}_{(N-K)K} & \text{GFT}_{(N-K)(N-K)} \end{bmatrix} \quad (4.18)$$

$$\text{GFT}^{-1} = \begin{bmatrix} \text{GFT}_K^{-1} & \text{GFT}_{N-K}^{-1} \end{bmatrix} \quad (4.19)$$

$$= \begin{bmatrix} \text{GFT}_{KK}^{-1} & \text{GFT}_{K(N-K)}^{-1} \\ \text{GFT}_{(N-K)K}^{-1} & \text{GFT}_{(N-K)(N-K)}^{-1} \end{bmatrix} \quad (4.20)$$

$$P_{\delta^{(\text{spl})}}(M) = \begin{bmatrix} P_{\delta^{(\text{spl})}}(M)_K & P_{\delta^{(\text{spl})}}(M)_{N-K} \end{bmatrix}. \quad (4.21)$$

In (4.17), (4.19), and (4.21), the partitions are columnwise, not rowwise as in (4.1) and the left blocks are  $N \times K$  and the right blocks  $N \times (N - K)$ . In (4.18) and (4.20), the subindices give the dimensions of each subblock of GFT and  $\text{GFT}^{-1}$ . E.g.,  $\text{GFT}_{KK}$  is the top left  $K \times K$  subblock of the GFT.

*Proof.* Recall the filter  $P(M)_{\delta^{(\text{spl})}}$  given by (4.11). Using (4.17) and (4.19) in (4.11), get

$$P_{\delta^{(\text{spl})}}(M) = \begin{bmatrix} \text{GFT}_K & \text{GFT}_{N-K} \end{bmatrix} \overbrace{\begin{bmatrix} I_K \\ 0_{N-K} \end{bmatrix}}^{\text{diag}[\delta^{(\text{spl})}]} \begin{bmatrix} \text{GFT}_K^{-1} & \text{GFT}_{N-K}^{-1} \end{bmatrix} \quad (4.22)$$

$$= \begin{bmatrix} \text{GFT}_K \text{GFT}_{KK}^{-1} & \text{GFT}_K \text{GFT}_{K(N-K)}^{-1} \end{bmatrix}. \quad (4.23)$$

Now, using the bandlimitedness of  $\hat{s}$  and  $\delta^{(\text{spl})}$  as in (4.16) in equation (4.9) of result 4.3, we

get

$$\delta^{(\text{spl})} \odot s = \begin{bmatrix} s_K \\ 0 \end{bmatrix} \xrightarrow{\mathcal{F}} P_{\delta^{(\text{spl})}}(M) \cdot \widehat{s} = \text{GFT}_K \text{GFT}_{KK}^{-1} \widehat{s}_K. \quad (4.24)$$

Taking the GFT of the left-hand side of (4.24), we get

$$\widehat{s}_\delta = \text{GFT} \begin{bmatrix} s_K \\ 0 \end{bmatrix} = P_{\delta^{(\text{spl})}}(M) \cdot \widehat{s} \quad (4.25)$$

$$= \text{GFT}_K \text{GFT}_{KK}^{-1} \widehat{s}_K \quad (4.26)$$

$$= \begin{bmatrix} [\text{GFT}_K]_{0K} \text{GFT}_{KK}^{-1} \\ \dots \\ [\text{GFT}_K]_{iK} \text{GFT}_{KK}^{-1} \\ \dots \\ [\text{GFT}_K]_{(\frac{N}{K}-1)K} \text{GFT}_{KK}^{-1} \end{bmatrix} \widehat{s}_K, \quad (4.27)$$

where we partitioned the  $N \times K$  matrix  $\text{GFT}_K$  into  $\frac{N}{K}$  blocks  $[\text{GFT}_K]_{iK}$ ,  $i = 0, \dots, \frac{N}{K} - 1$ , where block  $[\text{GFT}_K]_{iK}$  collects the  $K$  rows  $iK, i\frac{N}{K} + 1, \dots, (i+1)K - 1$ . Then (4.27) shows that  $\widehat{s}_\delta$  has  $\frac{N}{K}$  copies  $[\text{GFT}_K]_{iK} \text{GFT}_{KK}^{-1} \widehat{s}_K$  as we wanted to show. Since each of the  $\frac{N}{K}$  blocks of  $\widehat{s}_\delta$  is a (filtered) replica of  $\widehat{s}_K$  obtained by multiplying it with a  $K \times K$  matrix block, no aliasing occurs.  $\blacksquare$

Result 4.3 and equation (4.9), as well as result 4.4 and equation (4.27), show that, just like for DSP, GSP graph *subsampling* has the dual interpretation of 1) pointwise multiplication (modulation)  $\delta^{(\text{spl})} \odot s$  in the vertex domain; and 2) LSI filtering in the spectral frequency domain. But further and very interestingly 3) result 4.4 and equation (4.27) show the spectrum *replication* effect, with the spectrum of the sampled signal  $\widehat{s}_\delta$  given by  $\frac{N}{K}$  (filtered, possibly distorted) copies of the nonzero spectrum  $\widehat{s}_K$  of  $s$ . We note that we have assumed that we are sampling at rate  $K$  with no aliasing.

Although we have  $\frac{N}{K}$  (filtered, possibly distorted) copies of  $\widehat{s}_K$ , we are not guaranteed that any of the blocks  $\text{GFT}_{iK}$ ,  $i = 0, \dots, \frac{N}{K} - 1$  is full rank. This question will be taken care of when we consider decimation in section 4.1.3.

The next Theorem shows that  $P(M)$  in result 4.3 is the replicating filter equivalent to a train of frequency deltas when the GSP graph  $G$  is the directed cycle graph of DSP.

**Theorem 4.1** (DSP  $P(M)$ ). *Let:  $G$  be a directed cycle graph of  $N$  nodes;  $s$  a lowpass signal with cutoff frequency  $K$ ;  $I_K$  the  $K \times K$  identity matrix. Sampling with period  $\frac{N}{K}$ ,*<sup>3</sup>

$$P_{\delta^{(spl)}}(M) = \frac{K}{N} \begin{bmatrix} I_K & I_K & \dots & I_K \\ I_K & I_K & \dots & I_K \\ \vdots & \vdots & \ddots & \vdots \\ I_K & I_K & \dots & I_K \end{bmatrix}. \quad (4.28)$$

A proof of Theorem 4.1 can be found in the DSP literature [75]. Equation (4.11) using  $\text{GFT} = \text{DFT}$  and  $\delta^{(spl)}$  as the uniform sampling also yields  $P_{\delta^{(spl)}}(M)$ .

From Theorem 4.1, we see that uniformly sampling in DSP produces a  $P(M)$  that replicates exactly the band  $\widehat{s}$ . The spectrum of  $\widehat{s}_\delta$  shows exact replications of  $\widehat{s}_K$  in DSP. For GSP and arbitrary graphs, by result 4.4 and equation (4.27),  $\widehat{s}_\delta$  is also multiple replicated copies of  $\widehat{s}_K$ , but the replicas may be distorted.

We may ask for which other graphs, besides the cycle graph, is the replicating filter LSI leading to  $\widehat{s}_\delta$  to be  $\frac{N}{K}$  exact replicas of  $\widehat{s}_K$ . We provide two classes of graphs that, with specific choices of  $\delta^{(spl)}$ , also lead to exact replicating filters that are LSI filters, i.e., polynomials in  $M$ .

Let  $P_{\text{repl}}$  be the replicating LSI filter in Theorem 4.1.

**Example 4.1** (Circulant graphs). *Interpret  $P_{\text{repl}}$  as a circulant matrix. Its eigendecomposition is*

$$P_{\text{repl}} = \text{DFT} \text{diag} [\delta^{(spl)}] \text{DFT}^{-1}, \quad (4.29)$$

---

<sup>3</sup>This samples uniformly every  $\frac{N}{K}$ , keeping  $K$  samples and zeroing  $\frac{N}{K} - 1$  samples in between.

where  $\delta^{(spl)}$  is uniformly sampling every  $\frac{N}{K}$  values.

Consider that the graph is also given by circulant adjacency matrices but now with distinct eigenvalues. The graph Fourier transform is again  $GFT = DFT$ . Then, pre- and post-multiplying  $P_{repl}$  in (4.29) by  $GFT^{-1}$  and  $GFT$

$$GFT^{-1}P_{repl}GFT = \text{diag}(\delta^{(spl)}), \quad (4.30)$$

which is diagonal. The replicating filter  $P_{repl}$  can then be written as a polynomial of  $M$  and is thus, LSI.

We illustrate with the specific graph in figure 4.2. Its adjacency matrix  $A$  is circulant. In this example, every node of the graph is connected to its next node, its fourth next node and its sixth next node. By result 3.2,  $A = A_c + A_c^4 + A_c^6 = DFT^H \Lambda DFT = M$  where  $A_c$  is the adjacency matrix of the  $8 \times 8$  directed cyclic graph in (2.1),  $\Lambda = \text{diag}(\lambda)$ ,  $\lambda = [3, -.29j, .29j, -j, -1.7 - 1.7j, 1, -1.7, 1.7j, j, -.29, -.29j]^T$ .

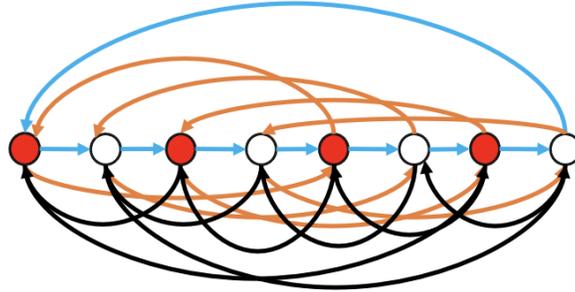


Figure 4.2: Circulant Matrix: sampled nodes for replicating filter are red. Blue edges connect nodes with next nodes. Orange edges connect nodes with fourth next nodes. Black edges connect nodes with sixth next nodes.

Let  $K = 4$ . Then, we compute  $\delta^{(spl)}$ . As in equation (4.30):  $\text{diag}(\delta^{(spl)}) = DFT^H P_{repl}(M)DFT = [1, 0, 1, 0, 1, 0, 1, 0]^T$ .

We now compute the coefficients  $p_{repl}$  of  $P_{repl}(M)$  as a polynomial in  $M$ . Using (4.13) yields  $P_{repl}(M) = -.01I_8 - .08M - .1M^2 + .18M^3 + .88M^4 + .21M^5 - .03M^6 - .05M^7$ .

**Example 4.2** (Kronecker Product). Write  $P_{repl}$  as

$$P_{repl} = [11^T]_{\frac{N}{K}} \otimes I_K \quad (4.31)$$

where  $\otimes$  is the Kronecker product and  $[11^T]_{\frac{N}{K}}$  is a  $\frac{N}{K} \times \frac{N}{K}$  matrix of all 1s. This yields the eigendecomposition

$$P_{repl} = \left( DFT_{\frac{N}{K}} \otimes V_K \right) \text{diag} \left( [I_K^T, 0_{N-K}^T]^T \right) \left( DFT_{\frac{N}{K}}^{-1} \otimes V_K^{-1} \right), \quad (4.32)$$

where  $V_K$  is any invertible matrix.

Consider graphs with adjacency matrix  $A = A_{\frac{N}{K}} \otimes B$  with unique eigenvalues where  $A_{\frac{N}{K}}$  is the  $\frac{N}{K}$  node cycle graph and  $B$  is any  $K$  node graph. The GFT of this graph is  $DFT_{\frac{N}{K}} \otimes V_K$  where  $V_K$  is the GFT of  $B$ . So:

$$GFT^{-1} P_{repl} GFT = \text{diag}(\delta^{(spl)}) = \text{diag} \left( [I_K^T, 0_{N-K}^T]^T \right) \quad (4.33)$$

is diagonal. The replicating filter  $P_{repl}$  can be written as a polynomial of  $M$  and is thus, LSI.

We illustrate this example with the graph in figure 4.3 with adjacency matrix,  $A = A_4 \otimes B$

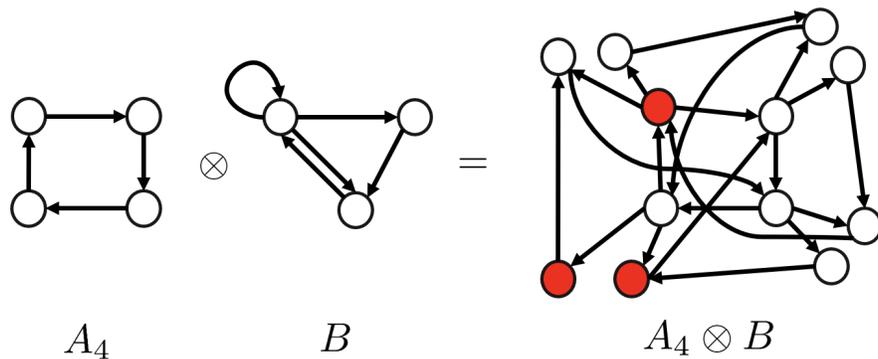


Figure 4.3: Kronecker Product Graphs. Red nodes are the sampled nodes for the replicating filter.

where  $A_4$  is the four node cycle graph and  $B = [b_0 \ b_1 \ b_2]$  with  $b_0 = [0 \ 1 \ 0]^T$ ,  $b_1 = [1 \ 1 \ 1]^T$ , and

$b_2 = [1\ 0\ 0]^T$ . The eigendecompositions of  $B$  and  $A$  are

$$B = GFT_B^{-1} \Lambda_B GFT_B$$

$$\Lambda_B = \text{diag}(\lambda_B), \lambda_B = [1.893, -.419 + .606j, -.419 - .606j]^T$$

$$GFT_B^{-1} = \begin{bmatrix} -.594 & .676 & .676 \\ -.707 & -.402 - .172j & -.402 + .172j \\ -.384 & .119 + .581j & .119 - .581j \end{bmatrix}$$

$$A = GFT_A^{-1} \Lambda_A GFT_A, GFT_A = DFT_4 \otimes GFT_B$$

$$\Lambda_A = \text{diag}[\lambda_B^T, -j\lambda_B^T, -\lambda_B^T, j\lambda_B^T].$$

Then,

$$M = (DFT_4 \otimes GFT_B) \Lambda_A^* (DFT_4^H \otimes GFT_B^{-1}).$$

Let  $P_{\text{repl}}(M)$  be the replicating filter in Theorem 4.1 with  $K = 3$ . We have

$$\text{diag}(\delta^{(\text{spl})}) = GFT_A^{-1} P_{\text{repl}}(M) GFT_A = [1, 1, 1, 0, 0, \dots, 0]^T.$$

We compute LSI  $P_{\text{repl}}(M)$ . Using (4.13), obtain

$$\begin{aligned} P_{\text{repl}}(M) &= .25I_{12} - .5M + .31M^2 + .46M^3 - 1.38M^5 \\ &\quad + 2.13M^6 - .75M^7 + .13M^9 - .19M^{10} + .06M^{11}. \end{aligned}$$

### 4.1.3 Decimation

This section shows that vertex and spectral GSP decimation parallel DSP decimation. In DSP, decimation keeps the  $K$  sampled values and removes the  $N - K$  zeros from the subsampled signal  $s_\delta$ . The  $N$  node cycle graph shrinks to the  $K$  node cycle graph with the  $K \times K$  DFT

in (2.8). The signal spectrum “stretches” in frequency. Likewise, in GSP, the decimated signal  $s_d$  is the downsampled signal that keeps the  $K$  sampled values and removes the  $N - K$  zeros. While the DSP “decimated” graph is the  $K$  node cycle graph, the GSP “decimated” graph  $A_d$  is not as straightforward. We consider here the “decimated” signal  $s_d$ , confirm the stretching of its graph spectrum  $\widehat{s}_d$ , and present the “decimated”  $\text{GFT}_d$  and the “decimated” graph  $A_d$ .

**Result 4.5** (Decimated  $s_d$  and  $\text{GFT}_d$ ). *Let bandlimited  $\widehat{s} = [\widehat{s}_K^T \ 0_{N-k}^T]^T$  and  $\delta^{(spl)}$  as in (4.16).*

*Then,*

$$s_d = s_K, \widehat{s}_d = \widehat{s}_K, \text{ and } \text{GFT}_d = \text{GFT}_{KK}, \quad (4.34)$$

where  $\text{GFT}_{KK}$  is the top left  $K \times K$  subblock of the  $\text{GFT}$  (see equation (4.18)) and  $\text{GFT}_d$  is the “decimated”  $\text{GFT}$ , the  $\text{GFT}$  of the “decimated” graph  $A_d$ .

*Proof.* From (4.24), we have

$$s_\delta = \begin{bmatrix} s_K \\ 0 \end{bmatrix} = \text{GFT}^{-1} \text{GFT}_K \text{GFT}_{KK}^{-1} \widehat{s}_K. \quad (4.35)$$

But

$$\text{GFT}^{-1} \text{GFT}_K = \begin{bmatrix} I_{KK} \\ 0_{(N-K)K} \end{bmatrix}, \quad (4.36)$$

since multiplication of the first  $K$  rows of  $\text{GFT}^{-1}$  by  $\text{GFT}_K$  gives  $I_{KK}$  and the last  $N - K$  rows of  $\text{GFT}^{-1}$  are orthogonal to the columns in  $\text{GFT}_K$ . Substituting (4.36) in (4.35), get

$$s_d = s_K = \text{GFT}_{KK}^{-1} \widehat{s}_K. \quad (4.37)$$

Finally, we prove that  $\text{GFT}_{KK}^{-1}$  is full rank and hence invertible. This follows because, by choice of the sampling set  $S$  (and sampling graph signal), by result 4.2, or equation (4.8),  $s_K$  uniquely

determines signal  $s$ . By uniqueness of the GFT,  $\widehat{s} = [\widehat{s}_K^T \ 0_{N-k}^T]^T$  is uniquely determined from  $s$  and hence from  $s_K$ . This also determines  $\widehat{s}_K$  uniquely from  $s_K$ . Since (4.37) is a  $K \times K$  linear relation between  $s_K$  and  $\widehat{s}_K$ , we conclude that  $\text{GFT}_{KK}^{-1}$  is full rank and thus invertible. Since  $s_d = s_K$  and  $\text{GFT}_d = \text{GFT}_{KK}$ , we also get from (4.37) that  $\widehat{s}_d = \widehat{s}_K$ . The proof is complete.  $\blacksquare$

Consider graph  $G_d$  with adjacency  $A_d$  indexing  $s_d$ .

**Result 4.6** (Decimated graph  $A_d$ ). *Let bandlimited  $\widehat{s} = [\widehat{s}_K^T \ 0_{N-k}^T]^T$  and the sampling signal  $\delta^{(\text{spl})}$  as in (4.16), and  $\text{GFT}_d = \text{GFT}_{KK}$ . Then the decimated graph  $A_d$  is*

$$A_d = \text{GFT}_d^{-1} \cdot \Lambda_d \cdot \text{GFT}_d = \text{GFT}_{KK}^{-1} \cdot \Lambda_d \cdot \text{GFT}_{KK} \quad (4.38)$$

where  $\Lambda_d = \text{diag}[\lambda_0 \dots \lambda_{K-1}]$ .

*Proof.* We first determine the eigenvalues of the decimated graph. From (3.22), the signal  $s$  is a linear combination of the powers of the vector  $\lambda^* = [\lambda_0^* \ \lambda_1^* \ \dots \ \lambda_{N-1}^*]^T$

$$s = p_0 \mathbf{1} + p_1 \lambda^* + \dots + p_{N-1} \lambda^{*(N-1)}. \quad (4.39)$$

Then, the sampled signal is

$$\delta^{(\text{spl})} \odot s = \delta^{(\text{spl})} \odot (p_0 \mathbf{1} + p_1 \lambda^* + \dots + p_{N-1} \lambda^{*(N-1)}) \quad (4.40)$$

$$= p_0 (\delta^{(\text{spl})} \odot \mathbf{1}) + \dots + p_{N-1} (\delta^{(\text{spl})} \odot \lambda^{*(N-1)}). \quad (4.41)$$

In (4.41), powers of vector  $\lambda^n$  of eigenvalues are sampled by  $\delta^{(\text{spl})}$ , zeroing out  $N-K$  eigenvalues. Let  $\lambda_d$  be the vector of non zeroed  $K$  eigenvalues (same ordering). They are the eigenvalues of  $A_d$ . Then  $A_d$  follows as in (4.38).  $\blacksquare$

**Remark 4.2** (Sampling eigenvalues). *When sampling  $s$  using  $\delta^{(\text{spl})}$ , the eigenvalues are sampled the same way. The chosen eigenvalues do not depend on which components of  $\widehat{s}$  are zero, only*

on the choice of the sampling set.

We finally consider the stretched spectrum of  $s_d$ .

**Result 4.7** (GSP: Stretching). *Given the set-up of the previous result, with  $s$  bandlimited,  $\widehat{s} = [\widehat{s}_K^T \ 0_{N-K}^T]^T$ , the spectrum of the decimated signal  $s_d$  is stretched over the full range of frequencies of the decimated graph  $A_d$ .*

*Proof.* By result 4.6, we see that the spectrum of the decimated signal  $s_d$  is  $\widehat{s}_d = \widehat{s}_K$ , and it occupies the full band  $\Lambda_d$  of eigenvalues of the decimated graph  $A_d$ . ■

Note that  $\Lambda_d$  is a subset of eigenvalues of  $A$ , so stretching has the same interpretation in both DSP and GSP.

The next example illustrates how to derive the DSP vertex (time) and frequency interpretations for  $s_d$  when the graph is the time directed cycle graph.

**Example 4.3** (DSP Example). *Sample uniformly  $s$  defined on cycle graph with adjacency  $A$ , taking every  $\frac{N}{K}$  samples. Let decimated signal be  $s_d = s_K$  and  $\widehat{s} = [\widehat{s}_K^T, 0^T]^T$ .*

*The eigenvalues of  $A$  are  $\lambda_k = e^{-j\frac{2\pi}{N}k}$ . Sampling uniformly produces  $\lambda_{d,r} = e^{-j\frac{2\pi}{N}(\frac{N}{K})r} = e^{-j\frac{2\pi}{K}r}$ ,  $r = 0, 1, \dots, K-1$ . These are the eigenvalues of the  $K$  node cycle graph  $C_K$  illustrating the need to sample uniformly. Not sampling uniformly does not choose the eigenvalues of  $C_K$ . Finally, the decimated graph is  $A_d = DFT_d^{-1} \cdot \Lambda_d \cdot DFT_d$ , the  $K$  node cycle graph  $C_K$  where  $DFT_d$  is the  $K \times K$  DFT in (2.8).*

*This shows again for DSP that the spectrum of the subsampled signal  $s_\delta$  is  $\frac{N}{K}$  replicas of the low pass spectrum  $\widehat{s}_K$ . Using (4.37) and Theorem 4.1*

$$P_{\delta(\text{spl})}(M)\widehat{s} = GFT_K s_K = \begin{bmatrix} DFT_d \\ \vdots \\ DFT_d \end{bmatrix} s_K = \begin{bmatrix} \widehat{s}_K \\ \vdots \\ \widehat{s}_K \end{bmatrix}, \quad (4.42)$$

**Remark 4.3** (DSP: Stretching). *The original signal is at  $N$  frequencies  $2\pi l/N$ ,  $l = 0, 1, \dots, N-1$ , while the decimated signal is at  $K$  frequencies,  $2\pi l/K$ ,  $l = 0, 1, \dots, K-1$ . This “stretches” the signal spectrum to fill the  $2\pi$  range.*

## 4.2 GSP Sampling: Upsampling and Interpolation

We assume the same set-up as described in the introduction to section 4.1. We start in subsection 4.2.1 with upsampling. Then in subsection 4.2.2 we address the conditions of when does a given sampling signal  $\delta^{(\text{spl})}$  lead to perfect reconstruction, and finally in subsection 4.2.3 we explore GSP interpolation as filtering.

### 4.2.1 Reconstruction: Upsampling

Upsampling in DSP reintroduces the zeros into the  $K \times 1$  signal,  $s_d$ , producing the sampled  $N \times 1$ ,  $s_\delta$ . The  $K$  node cycle graph becomes the  $N$  node cycle graph. We emphasize that in DSP we know: 1) the larger and decimated graphs ( $N$  and  $K$  node cycle graphs) and their adjacency matrices  $A$  and  $A_d$ ; 2) the positions of the zeros when adding the zeros back into  $s_d$ ; 3) the eigenvalues of  $A$  and  $A_d$ ; and 4) the  $\text{DFT}_N$  and  $\text{DFT}_K$ .

In GSP, to upsample, we also need to know 1) both the original and downsampled graphs  $G$  and  $G_d$  and their adjacency matrices  $A$  and  $A_d$ ; 2) the positions of the zeros when adding the zeros back into  $s_d$ ; 3)  $\Lambda$  and  $\Lambda_d$ ; and 4)  $\text{GFT}$  and  $\text{GFT}_d$ . Then, upsampling starts with padding zeros to  $s_d$  to produce  $s_\delta$  in the vertex domain. In the spectral domain, we obtain  $P(M)\hat{s}$ . Now,  $\hat{s}$  no longer extends over the frequency range of  $A$  since it is bandlimited.

### 4.2.2 Reconstruction: Perfect Reconstruction

Let  $s \xleftrightarrow{\mathcal{F}} \hat{s}$ . Assume  $s$  has bandwidth  $K$ , i.e.,  $\|\hat{s}\|_0 \leq K$ , and that after possible permutation  $\hat{s} = [\hat{s}_K^T \hat{s}_{N-K}^T]^T$  with  $\hat{s}_{N-K} = 0$ . By result 4.2 there is a sampling set  $S$  with characteristic function  $\delta^{(\text{spl})}$  such that perfect reconstruction is possible from the samples  $s_d$  in  $S$ , i.e.,  $s = s_r$ ,

with  $s_r$  reconstructed from  $s_d$ , for example, using (4.8).

Now we address a different question for  $s$  bandlimited with bandwidth =  $K$ . Let the sampling signal  $\delta^{(\text{spl})}$  have  $\|\delta^{(\text{spl})}\| = K$ . The question is when does  $\delta^{(\text{spl})}$  lead to perfect reconstruction of  $s$ . In other words, when can we recover  $s$  from  $s_d = s_K$  where  $s_K$  is obtained by discarding the zeros of the subsampled  $s_\delta = \delta^{(\text{spl})} \odot s$ .

Let  $\delta^{(\text{spl})} = [1_K^T \ 0_{N-K}^T]^T$  and recall  $P_{\delta^{(\text{spl})}}(M)$  in (4.21). Let  $P_{\delta^{(\text{spl})}}(M)_K$  be its first  $K$  columns in its partition (4.23).

**Result 4.8** (Rank of  $P_{\delta^{(\text{spl})}}(M)_K$ ). *Then*

$$\text{rank}[P_{\delta^{(\text{spl})}}(M)_K] = \text{rank}(GFT_K GFT_{KK}^{-1}) = K \quad (4.43)$$

*iff the  $K \times K$  square matrix  $GFT_{KK}^{-1}$  is invertible.*

*Proof. Only if:*

We have  $\text{rank}(GFT_K) = K$  and  $\text{rank}(P_{\delta^{(\text{spl})}}(M)_K) \leq \min(\text{rank}(GFT_K), \text{rank}(GFT_{KK}^{-1}))$ .

Then, if  $GFT_{KK}^{-1}$  not invertible, its rank  $< K$ , and  $\text{rank}(P_{\delta^{(\text{spl})}}(M)_K) < K$ .

*If:* If  $\text{rank}(GFT_{KK}^{-1}) = K \implies \text{rank}(P_{\delta^{(\text{spl})}}(M)_K) = \text{rank}(GFT_K) = K$ . ■

**Result 4.9** (Perfect reconstruction sampling condition). *Without loss of generality, let  $\delta^{(\text{spl})} = [1_K^T \ 0_{N-K}^T]^T$ . Assume  $\hat{s}_{N-K} = 0$ . The signal  $s$  can be perfectly reconstructed from  $s_d = s_K$  iff  $\text{rank}(GFT_{KK}^{-1}) = K$ .*

*Proof.* With  $s$  lowpass,

$$\delta^{(\text{spl})} \odot s = \delta^{(\text{spl})} \odot \begin{bmatrix} s_K \\ s_{N-K} \end{bmatrix} = \begin{bmatrix} s_K \\ 0_{N-K} \end{bmatrix} \quad (4.44)$$

$$= GFT^{-1} P_{\delta^{(\text{spl})}}(M)_K \hat{s}_K \quad (4.45)$$

$$= GFT^{-1} GFT_K GFT_{KK}^{-1} \hat{s}_K \quad (4.46)$$

$$= \begin{bmatrix} GFT_{KK}^{-1} \\ 0_{N-K} \end{bmatrix} \hat{s}_K. \quad (4.47)$$

Given  $s_d = s_K$ ,  $\widehat{s}_K$  is determined from (4.47) iff  $\text{GFT}_{KK}^{-1}$  is invertible, from which  $s$  is perfectly reconstructed. ■

This result seems repetitive when contrasted with result 4.5 and equation (4.37). The difference is that in result 4.5 and equation (4.37) we assume that  $\delta^{(\text{spl})}$  corresponds to a sampling set  $S$  for which we know we can reconstruct perfectly  $s$  from  $s_d$ , while here we are given a  $\delta^{(\text{spl})}$  and have to find conditions for perfect reconstruction of  $s$  from  $s_d$ .

Result 4.9 provides how to reconstruct in the spectral domain, shown in result 4.10.

**Result 4.10** (Reconstruction in spectral domain). *Under result 4.9 assumptions, let bandlimited  $s$  with bandwidth  $K$  be decimated to  $s_d = s_K$ . Then  $s$  is reconstructed by*

$$\widehat{s}_K = \left[ \text{GFT}_{(KK)}^{-1} \right]^{-1} s_K \implies s = \text{GFT}^{-1} \begin{bmatrix} \widehat{s}_K \\ 0_{N-K} \end{bmatrix}. \quad (4.48)$$

The proof follows from result 4.9 and (4.47).

Partitioning  $\text{GFT}^{-1}$  as in (4.20), get from result 4.10

$$s = \begin{bmatrix} I_K \\ \text{GFT}_{(N-K)K}^{-1} \left[ \text{GFT}_{(KK)}^{-1} \right]^{-1} \end{bmatrix} s_K. \quad (4.49)$$

This shows it is possible to recover  $s$  from  $s_d = s_K$ . But, like for DSP and Shannon reconstruction, it is important to find equivalent filtering interpretations for reconstruction, in both the vertex and the spectral domains. The next subsection explores this.

### 4.2.3 Reconstruction: Interpolation as Filtering

Result 4.10 shows one way to reconstruct  $s$  from  $s_d = s_K$ . In DSP, Shannon's Sampling Theorem reconstructs the signal by ideal lowpass *filtering* the upsampled signal. Likewise, we consider a GSP *filtering* approach to reconstruct  $s$  from the upsampled  $s_d$ . This parallels section 4.1.2, where downsampling is by *LSI spectral* filtering, see (4.9), result 4.3. We show

that reconstruction from an upsampled signal can be achieved by spectral domain filtering, but, in contrast with section 4.1.2, the reconstruction filter is not in general LSI.

**Result 4.11** (Reconstruction by filtering). *Let  $s$  be bandlimited with bandwidth  $K$ ,  $\hat{s} = [\hat{s}_K^T \ 0_{N-K}^T]^T$ . Let  $\delta^{(spl)} = [1_K^T \ 0_{N-K}^T]^T$  be the sampling signal for a sampling set  $S$ , and  $s$  be decimated to  $s_d = s_K$  by  $\delta^{(spl)}$ . Then reconstruct  $s$  by filtering upsampled  $s_d$  as follows:*

$$s = P_{\delta^{(spl)}}(A) \cdot F \cdot s_d \xrightarrow{\mathcal{F}} \begin{bmatrix} 1_K \\ 0_{N-K} \end{bmatrix} \odot Q \cdot \hat{s}_d \quad (4.50)$$

where

$$Q = \begin{bmatrix} Q_{KK} & Q_{K(N-K)} \\ \hline & Q_{(N-K)N} \end{bmatrix}, \quad F = GFT^{-1} \cdot Q \cdot GFT \quad (4.51)$$

$$Q \cdot P_{\delta^{(spl)}}(M) = \begin{bmatrix} I_{KK} & B_{K(N-K)} \\ \hline & B_{(N-K)N} \end{bmatrix} \quad (4.52)$$

$$P(A) = GFT^{-1} \begin{bmatrix} I_{KK} & \\ & 0_{(N-K)(N-K)} \end{bmatrix} GFT \quad (4.53)$$

where  $B_{K(N-K)}$  and  $B_{(N-K)N}$  are non prescribed.

We interpret the result before proving it. The left side of (4.50) reconstructs  $s$  by filtering in the vertex domain the upsampled  $s_d$ . The right-hand side, reconstructs  $\hat{s}$  by filtering in the spectral domain the upsampled  $\hat{s}_d$ . Equation (4.50) gives GSP reconstruction in the spectral domain as filtering with filter  $Q$  followed by lowpass LSI ideal filtering (Hadamard product with  $[1_K^T \ 0_{N-K}^T]^T$ ). In the vertex domain (left side of (4.50)), we have first the non LSI filtering by  $F$ , followed by the ideal lowpass LSI filter  $P(A)$ , whose frequency response is  $[1_K^T \ 0_{N-K}^T]^T$ . In DSP, as we will show below,  $Q$  and  $F$  are trivial filters, and reconstruction is limited to ideal lowpass  $P(A)$  (impulse response is the discrete sinc) with flat frequency response over the signal band. In (4.50), the reconstruction filter  $Q$  is not necessarily LSI, i.e., not necessarily

polynomial in the spectral shift  $M$ . Certain blocks like  $Q_{(N-K)N}$  are not constrained.

Its equivalent in the vertex domain is filter  $F$  that again is not, in general, LSI, i.e., not polynomial in  $A$ . On the other hand,  $\begin{bmatrix} 1_K \\ 0_{N-K} \end{bmatrix}$  is an ideal lowpass LSI filter whose vertex equivalent is the LSI polynomial filter  $P(A)$ .

*Proof.* By result 3.5 and (3.21) and graph signal  $\delta^{(\text{spl})}$

$$P_{\delta^{(\text{spl})}}(M) = \text{GFTdiag}[\delta^{(\text{spl})}] \text{GFT}^{-1}. \quad (4.54)$$

Since  $\|\delta^{(\text{spl})}\|_0 = K$ ,  $\text{rank}(P_{\delta^{(\text{spl})}}) = K$ . From (4.9), result 4.3,

$$P_{\delta^{(\text{spl})}}(M) \begin{bmatrix} \widehat{s}_K \\ 0_{N-K} \end{bmatrix} = \widehat{s}_\delta. \quad (4.55)$$

Gauss Jordan elimination (GJ-E) determines a  $Q$  such that  $P_{\delta^{(\text{spl})}}(M)$  is in reduced row echelon form. After GJ-E, (4.55) becomes

$$Q \cdot P_{\delta^{(\text{spl})}}(M) \begin{bmatrix} \widehat{s}_K \\ 0_{N-K} \end{bmatrix} = Q \cdot \widehat{s}_\delta \implies \begin{bmatrix} I_{KK} & \widetilde{P}_{K(N-K)} \\ \dots & \dots \\ 0_{(N-K)N} & \dots \end{bmatrix} \begin{bmatrix} \widehat{s}_K \\ 0_{N-K} \end{bmatrix} = \begin{bmatrix} \widehat{s}_{\delta_K} \\ 0_{N-K} \end{bmatrix} \quad (4.56)$$

from which  $\widehat{s}_K = \widehat{s}_{\delta_K}$ . The ideal lowpass filter  $\begin{bmatrix} 1_K^T & 0_{N-K}^T \end{bmatrix}^T$  still recovers  $\begin{bmatrix} \widehat{s}_K^T & 0_{N-K}^T \end{bmatrix}^T$  whose  $\text{GFT}^{-1}$  reconstructs  $s$ .

The left-hand side in (4.50) follows since spectral filtering by  $Q$  is in the vertex domain filtering by  $F$  given in (4.51), and ideal lowpass filtering in the spectral domain is in the vertex domain LSI filtering by  $P(A)$  given in (4.53).  $\blacksquare$

**Remark 4.4.** 1) The  $Q$  obtained by GJ-E in the proof of result 4.11 is not unique, and there are other methods to design a spectral filter  $Q$  such that  $Q \cdot P_{\delta^{(\text{spl})}}(M)$  has the block form in (4.52).  
2) If there is column swapping in GJ-E, then one needs to account for a permutation  $\Pi$ .

3) In the block form of (4.52), only the  $I_{KK}$  block matters, since the zero block in the lowpass signal  $\hat{s}$  multiplies  $\tilde{P}_{K(N-K)}$  and the ideal lowpass filter filters out  $\tilde{P}_{(N-K)N}$ . This provides degrees of freedom in designing spectral filter  $Q$ . 4) By row and column permutation, we can rearrange  $P_{\delta^{(spl)}}$  so that its block  $P_{\delta^{(spl)}}(M)_{KK}$  is invertible. Then,

$$Q = \left[ \begin{array}{c|c} [P_{\delta^{(spl)}}(M)_{KK}]^{-1} & 0_{K(N-K)} \\ \hline & Q_{(N-K)N} \end{array} \right] \quad (4.57)$$

can be used, with  $Q_{(N-K)N}$  designed to possibly achieve other design considerations. 5) Result 4.11 reconstructs  $s$  by filtering from its decimated  $s_d$  and upsampled  $s_\delta$  versions, paralleling the DSP uniform sampling reconstruction by ideal lowpass filtering. In general  $Q$  is not LSI, i.e., a polynomial in  $M$ . Given the degrees of freedom in  $Q$ , one can in some cases find a LSI version of  $Q$ , in which case reconstruction in GSP is, like in DSP, achieved by LSI filtering, see next example.

**Example 4.4.** Consider the five node star graph in Example 3.1 with bandlimited ( $K = 2$ )  $s = [-2\ 3\ 3\ 3\ 3]^T$ ,  $\hat{s} = [1\ 2\ 0\ 0\ 0]^T$ . Let  $\delta^{(spl)} = [1\ 1\ 0\ 0\ 0]^T$ . By (4.55),

$$P_{\delta^{(spl)}}(M) = \frac{1}{4}M^2 = \left[ \begin{array}{cc} B_{11} & B_{12} \\ B_{12}^T & B_{22} \end{array} \right] \quad (4.58)$$

$$B_{11} = \begin{bmatrix} .625 & -.375 \\ -.375 & .675 \end{bmatrix}, B_{12} = .177 \begin{bmatrix} 1_3^T \\ \\ 1_3^T \end{bmatrix}, B_{22} = .25I_3. \quad (4.59)$$

Since  $B_{11}$  is invertible, a possible filter is  $Q = \text{blockdiag}[Q_{11}, 4I_3]$ , with  $Q_{11} = [q_1\ q_2]$ ,  $q_1 = [2.5\ 1.5]^T$  and  $q_2 = [1.5\ 2.5]^T$ . This  $Q$  is LSI,  $Q = 4I_5 - .75M - .375M^2$ .

The ideal lowpass filter  $[1_2^T\ 0_3^T]$  is in the vertex domain  $P(A) = \frac{1}{4}A^2$ .

Reconstruction: Signal  $s$  is reconstructed from  $s_\delta$  and  $\hat{s}_\delta$  in the vertex and spectral domains,

both by **LSI filtering**:

$$\begin{array}{c}
 \underbrace{\frac{1}{4}A^2}_{\substack{\text{l.p.f.} \\ P(A) \\ \text{in} \\ \text{vertex}}} \underbrace{\begin{bmatrix} 1 \\ 4 \\ 4 \\ 4 \\ 4 \end{bmatrix}}_{\substack{F=Q \\ \text{in vertex}}} \odot \underbrace{\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\substack{\delta^{(spl)} \\ s_\delta}} \odot \underbrace{\begin{bmatrix} -2 \\ 3 \\ 3 \\ 3 \\ 3 \end{bmatrix}}_s \xrightarrow{\mathcal{F}} \underbrace{\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\text{l.p.f.}} \odot \underbrace{(4I - .75M - .375M^2)}_Q \odot \underbrace{\begin{bmatrix} M^2 \\ 4 \end{bmatrix}}_{P(M)} \underbrace{\begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\hat{s}_s}
 \end{array} \tag{4.60}$$

### 4.2.4 DSP from GSP: Nyquist-Shannon Sampling

To illustrate the impact of different GSP choices, we consider a simple example when  $G$  is the cycle graph and with DSP Nyquist-Shannon sampling. Let  $N = 4$ , signal  $s$  with four values, bandlimited with  $K = 2$ , and sampled uniformly.  $A$  is the directed cycle graph of 4 nodes. As shown in section 3.1,  $A = M$ . Also, in DSP, since the eigenvalues are all unique, multiplication in one domain is filtering in the other with a polynomial filter, either  $P(A)$  or  $P(M)$ .

Nyquist-Shannon sampling is illustrated by:

$$\underbrace{\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{\substack{P(A) \\ \text{sinc}}} \underbrace{\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{\delta^{(spl)}} \odot \underbrace{\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix}}_s \xrightarrow{\mathcal{F}} \underbrace{\begin{bmatrix} 2 \\ 2 \\ 0 \\ 0 \end{bmatrix}}_{\text{l.p.f.}} \odot \underbrace{\frac{1}{2} \begin{bmatrix} I_2 & I_2 \\ I_2 & I_2 \end{bmatrix}}_{P(M)} \underbrace{\begin{bmatrix} \hat{s}_0 \\ \hat{s}_1 \\ 0 \\ 0 \end{bmatrix}}_{\hat{s}_s}
 \end{array} \tag{4.61}$$

where  $I_2$  is the 2 – dimensional identity. Colors indicate same operation in both domains.

We show how Nyquist-Shannon recovery derives from GSP sampling. Let  $Q_2$  be the upper

half of  $Q$ . The condition on filter  $Q$  is for its upper half to satisfy

$$Q_2 P(M)_2 = Q_2 \frac{1}{2} \begin{bmatrix} I_2 \\ I_2 \end{bmatrix} = I_2 \quad (4.62)$$

where filters are indexed by their dimension. We consider three different GSP choices for  $Q_2$  that satisfy (4.62).

1.  $Q_2 = 2 \begin{bmatrix} I_2 & 0 \end{bmatrix}$ . We can fill in the bottom  $N - K$  rows of  $Q$  to produce a  $Q$  that is LSI:<sup>4</sup>  $Q = 2 \text{blockdiag} [I_2, I_2] = 2I_4$ . This  $Q$  is equivalent to multiplying  $s_\delta$  by  $2[1, 1, 1, 1]^T$  in the time domain. Thus,  $Q$  yields the following recovery:

$$\begin{array}{c}
 \underbrace{\begin{bmatrix} 2 \\ 0 \\ 2 \\ 0 \end{bmatrix}}_{\substack{P(A) \\ \text{sinc}}} \odot \underbrace{\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{\substack{\delta^{(\text{spl})} \\ Q_{\text{in time}}}} \odot \underbrace{\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix}}_{\substack{s_\delta \\ s}} \xrightarrow{\mathcal{F}} \underbrace{\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}}_{\text{l.p.f.}} \odot \underbrace{\begin{bmatrix} 2I_4 \\ \frac{1}{2} \begin{bmatrix} I_2 & I_2 \\ I_2 & I_2 \end{bmatrix} \end{bmatrix}}_{\substack{Q \\ P(M)}} \odot \underbrace{\begin{bmatrix} \hat{s}_0 \\ \hat{s}_1 \\ 0 \\ 0 \end{bmatrix}}_{\hat{s}}
 \end{array} \quad (4.63)$$

By moving the factor of 2 in (4.63) from  $Q$  to the low-pass filter, and similarly moving the factor of 2 from  $Q$  in time into the sinc, we achieve the traditional Nyquist-Shannon sampling recovery in (4.61). Equation (4.61) is a simplified version of (4.63), removing the filter  $Q$  since it is the identity.

2.  $Q_2 = \begin{bmatrix} I_2 & I_2 \end{bmatrix}$ . Choose the remaining  $N - K$  rows of  $Q$  to produce LSI filter  $Q = 1 \cdot 1^T \otimes I_2$ , a matrix with four  $I_2$  blocks. This  $Q$  is equivalent to multiplying  $s_\delta$  by  $[2, 0, 2, 0]^T = 2\delta^{(\text{spl})}$

<sup>4</sup>An LSI filter in the time domain is  $P(A)$  and in the frequency domain is  $P(M)$ . Since  $A$  is the directed cycle graph adjacency matrix and  $M = A$ , all LSI filters in DSP are circulant matrices.

in the time domain.

$$\begin{array}{c}
 \underbrace{\begin{bmatrix} 2 \\ 0 \\ 2 \\ 0 \end{bmatrix}}_{\text{sinc}} \odot \underbrace{\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{\delta^{(\text{spl})}} \odot \underbrace{\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix}}_{s} \xrightarrow{\mathcal{F}} \underbrace{\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}}_{\text{l.p.f.}} \odot \underbrace{\begin{bmatrix} I_2 & I_2 \\ I_2 & I_2 \end{bmatrix}}_Q \odot \underbrace{\frac{1}{2} \begin{bmatrix} I_2 & I_2 \\ I_2 & I_2 \end{bmatrix}}_{P(M)} \odot \underbrace{\begin{bmatrix} \hat{s}_0 \\ \hat{s}_1 \\ 0 \\ 0 \end{bmatrix}}_{\hat{s}}
 \end{array} \quad (4.64)$$

Since  $QP(M) = 2P(M)$ , move the factor of 2 from  $Q$  in time into the sinc and move the 2 in  $2P(M)$  into the low-pass filter in (4.64). By doing so, achieve traditional Nyquist-Shannon sampling recovery in (4.61). Equation (4.61) is a simplified version of (4.64) by replacing  $QP(M)$  with  $2P(M)$ .

3.  $Q_2 = 2 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ . Now  $Q$  cannot be LSI since the main diagonal must be constant. Filter  $Q$  is equivalent to multiplying by a filter  $\text{DFT}^{-1}Q\text{DFT}$  in the time domain. Since  $Q$  is not LSI, the time domain filter is not diagonal and not pointwise multiplication.

$$\begin{array}{c}
 \underbrace{\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{\text{sinc}} \odot \underbrace{\text{DFT}^{-1}Q\text{DFT}}_{Q \text{ in time}} \odot \underbrace{\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix}}_s \xrightarrow{\mathcal{F}} \underbrace{\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}}_{\text{l.p.f.}} \odot \underbrace{Q}_{\frac{1}{2} \begin{bmatrix} I_2 & I_2 \\ I_2 & I_2 \end{bmatrix}} \odot \underbrace{\begin{bmatrix} \hat{s}_0 \\ \hat{s}_1 \\ 0 \\ 0 \end{bmatrix}}_{\hat{x}_\delta}
 \end{array} \quad (4.65)$$

This example looked at three different possibilities for  $Q_2$ . The first two choices for the upper part of  $Q$ ,  $Q_2$ , lead to LSI  $Q$  and equation (4.61) leads to Nyquist-Shannon recovery. The third  $Q_2$  does not lead to an LSI filter, but can be used to recover the signal.

All  $Q_2$  use some values of  $\hat{s}_\delta$  to recover the signal. The first  $Q_2$  uses the first and second values. The second  $Q_2$  uses all four values. The third  $Q_2$  uses the first and fourth values. This

is shown in figure 4.4. Also, one might consider using the pseudoinverse of  $P(M)_2$  as in [34].

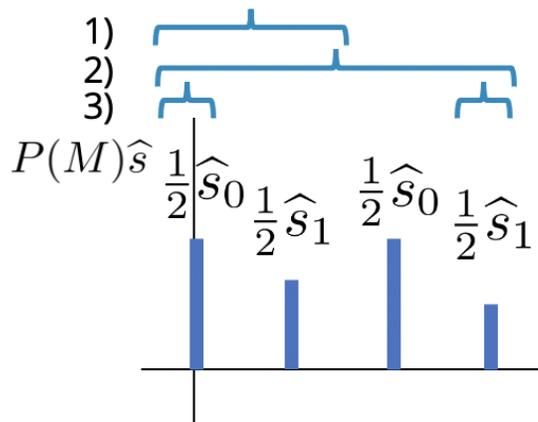


Figure 4.4: Values used by  $Q_2$  to recover: values shown for each  $Q_2$ .

The pseudoinverse of  $P(M)_2$  in (4.62) is the second  $Q_2$  considered above. It does indeed solve (4.62), but it is only one of possible choices as illustrated by the theory above.

In DSP Nyquist-Shannon Sampling,  $Q$  can be chosen as  $\frac{N}{K}I_N$ . The upper block of  $Q$  is  $Q_K = I_K$  followed by  $0_K$  matrices, scaled by  $\frac{N}{K}$ , as shown in (4.63). This factor can be merged into the lowpass filter and  $Q$  removed because it is the identity matrix. Thus, Nyquist-Shannon sampling recovery is a special, simplified case, where the  $Q$  is the identity and is removed. However, the example shows that GSP allows for other upper blocks of  $Q$ ,  $Q_K$ , including non-LSI ones that recover  $s$  and that are not considered by traditional Nyquist-Shannon sampling.

#### 4.2.5 Connection to Frequency Domain Sampling in [34, 35]

The frequency domain sampling proposed in [34, 35] does not correspond to the traditional concept of sampling as we explain now using our spectral filtering approach. In [35], let  $U_0^*$  be the GFT for the original undirected graph and  $U_1$  be  $\text{GFT}^{-1}$  for the sampled graph. Bandlimited graph signal  $s$  with band  $K$  is sampled in the frequency domain by:

$$f = U_1 Q_1 U_0^* s, \text{ with } Q_1 = \begin{bmatrix} I_{N/K} & I_{N/K} & \dots & I_{N/K} \end{bmatrix} \quad (4.66)$$

where  $f$  is the sampled signal. Matrix  $Q_1$  produces  $f$  whose GFT spectrum is replicated like in DSP Nyquist-Shannon sampling. Matrix  $Q_1$  is a decimated version of filter  $Q = [Q_1^T \cdots Q_1^T]^T$  with  $K \times K$  blocks. Filtering with  $Q$  in the frequency domain multiplies by  $\text{GFT}^{-1}Q\text{GFT}$  in the vertex domain. Since  $Q$  is circular, real, and symmetric  $Q = \text{DFT}^H \cdot \Lambda^* \cdot \text{DFT} = \text{DFT} \cdot \Lambda \cdot \text{DFT}^H$ . This is equivalent to multiplying in the vertex domain by  $\tilde{Q} = \text{GFT}^{-1} \cdot \text{DFT} \cdot \Lambda \cdot \text{DFT}^H \cdot \text{GFT}$ . In general  $\text{GFT}^{-1} \neq \text{DFT}^*$  and  $\text{GFT}^{-1} \cdot \text{DFT} \neq I$ . So,  $\tilde{Q}$  is not diagonal, and it cannot subsample in the vertex domain (keep some and discard other signal values). Because of this, filtering in the spectral domain with the approach in [35] requires knowledge and distorts **all** of the signal's samples in the vertex domain. This is different from recoverability in traditional sampling where only a decimated signal is used.

### 4.3 Conclusion

The literature on graph sampling is quite robust. Several approaches address the design of the sampling set  $S$  and develop alternative recovery methods. Some are developed in the vertex domain, others in the spectral domain. But the vertex and spectral domain sampling methods are not related lacking the dualism in DSP sampling. This chapter used the graph spectral shift  $M$ , the spectral graph signal processing theory (GSP<sub>sp</sub>), and the spectral delta functions introduced in Chapter 3 to present a unifying theory for GSP sampling showing the analogy and dualism between the vertex and spectral domain versions of all standard sampling steps. We show that GSP vertex subsampling is LSI filtering in the spectral domain with polynomials  $P(M)$ , decimation replicates the spectrum of the decimated signal, and interpolation is achieved by filtering operations in both vertex and spectral domains. Examples illustrate the impact of choices that can be made in GSP and show how GSP sampling becomes DSP sampling when the graph is the directed cycle time graph.



## Chapter 5

# Signal Representations and Companion Canonical Model

Chapter 3 introduced a spectral graph theory ( $\text{GSP}_{\text{sp}}$ ) including a spectral shift  $M$  and spectral delta functions. This chapter uses these to introduce new GSP signal representations as shifted delta functions in both the vertex domain ( $\delta_n = A^n \delta_0$ ,  $n = 0, 1, \dots, N-1$ ) and spectral domain ( $\widehat{\delta}_{n,\text{sp}} = M^n \delta_{\text{sp},0}$ ,  $n = 0, 1, \dots, N-1$ ). These new GSP signal representations lead to a *canonical* graph signal model defined by a *canonical* graph and a *canonical* shift, the *companion* graph and the *companion* shift. These are canonical because, under standard conditions, we show that any graph signal processing (GSP) model can be transformed into the canonical model.

Current GSP literature describes graph signals by their standard (node or vertex) representation  $s$  or their spectral representation  $\widehat{s}$ , but practically none has discussed or studied other graph signal representations or the issues related to signal representations that we pursue here.

### 5.1 Vertex and Fourier Signal Representations

At an abstract level, graph signals are vectors in an  $N$  dimensional graph signal vector space  $\mathbb{V}$  over field  $\mathbb{F}$ . Amplifying, attenuating, adding, filtering, or processing signals is simplified by first expressing them as linear combinations of  $N$  basic signals. As a prelude to the novel representations in the next sections, here, we discuss first in subsection 5.1.1 a generic representation, and then in subsections 5.1.2 and 5.1.3 consider the vertex and spectral representations, respectively.

The vertex signal  $s$  and the graph spectrum  $\widehat{s}$ , are two ways of describing the same graph signal but with respect to two different bases, the vertex standard Euclidean basis and the graph Fourier basis. Each of these signal descriptions has its own advantages. The vertex basis is the natural one, since the data is often collected at each node. The graph Fourier basis decomposes the signal model space into invariant subspaces (for diagonalizable shifts, these are the  $N$  one-dimensional eigenvector spaces) and signals aligned with these invariant subspaces are invariant to linear graph filtering.

### 5.1.1 Graph Signal Representations

In the  $N$ -dimensional signal vector space  $\mathbb{V}$  over the field  $\mathbb{F}$ , let  $B_U = \{u_0, \dots, u_{N-1}\}$  be a basis. Recall that the basis vectors  $\{u_n\}_{0 \leq n \leq N-1}$  are all nonzero and linearly independent. Mathematically, for any  $s \in \mathbb{V}$ :

$$s = (s_U)_0 u_0 + \dots + (s_U)_{N-1} u_{N-1} \quad (5.1)$$

$$= \underbrace{[u_0 \dots u_{N-1}]}_U \begin{bmatrix} (s_U)_0 \\ \vdots \\ (s_U)_{N-1} \end{bmatrix}. \quad (5.2)$$

$\underbrace{\hspace{10em}}_{s_U}$

**Remark 5.1** ( $\mathbb{V} \approx \mathbb{C}^N$ ). We assume the field  $\mathbb{F} = \mathbb{C}$ , so,  $s_U \in \mathbb{C}^N$ . By (5.1),  $\mathbb{V}$  is isomorphic to  $\mathbb{C}^N$ . In the sequel, we use this isomorphism and assume the signal space is the  $N$ -dimensional vector space  $\mathbb{C}^N$  over the field  $\mathbb{C}$ .

**Remark 5.2** (Ordered basis). For  $s_U$  to be well defined, the basis  $B_U$  is ordered. If we reorder the basis by a permutation  $P$ , the coordinate vector  $s_U$  is itself reshuffled by  $P$ :

$$s_U^P = P s_U. \quad (5.3)$$

By equation (5.2), processing signals is equivalent to computing using their coordinate

vectors  $s_U$ . Because of its significance, this coordinatization of signals receives a special designation.

**Definition 5.1** (Representation). *The representation of  $s$  with respect to the **ordered** basis  $B_U$  is its coordinate vector  $s_U$ . The  $n$ th component  $(s_U)_n$  of  $s_U$  is the coefficient of the basis vector  $u_n$  in the linear combination (5.1).*

Choosing a signal representation corresponds to choosing a basis  $B_U$ . There are infinitely many, with some particularly useful. DSP is essentially built around two representations (see section 2.1), discussed in the following sections 5.1.2 and 5.1.3 for GSP. For GSP, we consider six representations and discuss their specific advantages in the following sections in Chapter 5 and in Chapter 6.

### 5.1.2 Vertex, Standard, or Euclidean Representation

The graph signal  $s$  is an indexed collection of samples  $s = \{s_n\}_{n \in V}$ , one at each vertex of the graph. The vertex graph signal representation is the natural one where the  $n$ th-component of the coordinate vector is the graph sample  $s_n$  at indexing vertex  $n \in V$  of the graph. This representation corresponds to the standard or Euclidean ordered basis  $B_E = \{e_0, \dots, e_{N-1}\}$ . Clearly,  $\{e_n \neq 0\}_{0 \leq n \leq N-1}$  are linearly independent. For easy reference, we formally present the vertex or Euclidean representation.

**Definition 5.2** (Vertex, standard, or Euclidean representation). *The vertex, standard, or Euclidean representation of graph signal  $s \in \mathbb{V} \approx \mathbb{C}^N$  is the coordinate vector of  $s$  with respect to the standard basis  $B_E$ .*

$$s = s_0 e_0 + \dots + s_{N-1} e_{N-1} \tag{5.4}$$

$$= \underbrace{[e_0 \dots e_{N-1}]}_{I_N} \begin{bmatrix} s_0 \\ \vdots \\ s_{N-1} \end{bmatrix} = \begin{bmatrix} s_0 \\ \vdots \\ s_{N-1} \end{bmatrix} = s_E. \tag{5.5}$$

Component  $n$  of the coordinate vector  $s_E$  corresponds to  $n \in V$  and to  $e_n \in B_E$ . Ordering  $B_E$ , orders nodes and  $s_E$  is well defined. Because the matrix with columns  $e_n$  is the identity, we usually omit the subindex  $E$  and use the same symbol, e.g.,  $s$ , for the graph signal  $s$  and its vertex representation  $s_E$ .

Reordering  $B_E$  or the vertices permutes  $s_E$  as in (5.3). In DSP, time is ordered and this issue is taken for granted. In GSP, to process signals, the ordering should be fixed and shared.

### 5.1.3 Graph Fourier Representation

Fourier analysis, frequency components, bandlimited, low pass come naturally from the spectral or Fourier transform domain description  $\hat{s}$  of the signal  $s$ . This can also be interpreted as a representation of  $s$  where the Fourier basis is:

$$B_{\text{Fourier}} = \{v_0, \dots, v_{N-1}\}, \quad (5.6)$$

where the eigenvectors  $v_n$  of  $A$  are spectral modes and the columns of  $\text{GFT}^{-1}$ . We order the Fourier basis  $B_{\text{Fourier}}$ , ordering the spectral components  $v_k$  and the graph frequencies  $\lambda_k$ . The graph Fourier representation is formally presented next, again, for easy reference.

**Definition 5.3** (Graph Fourier representation). *The graph Fourier representation of  $s$  is its graph spectrum  $\hat{s}$ .*

$$s = \hat{s}_0 v_0 + \dots + \hat{s}_{N-1} v_{N-1} \quad (5.7)$$

$$= \underbrace{\begin{bmatrix} v_0 & \dots & v_{N-1} \end{bmatrix}}_{\text{GFT}^{-1}} \underbrace{\begin{bmatrix} \hat{s}_0 \\ \dots \\ \hat{s}_{N-1} \end{bmatrix}}_{\hat{s}} \quad (5.8)$$

## 5.2 Vertex Impulsive Representation

In this section, we consider several representations for the graph signal: 1) as a linear combination of graph vertex impulses; 2) as the impulse response of a graph filter; 3) as the inverse GFT of a linear combination of powers of the eigenvalues.

### 5.2.1 Vertex Impulsive Representation

Consider the (ordered) set of the graph vertex impulse and its delayed replicas in (3.46) and (3.47):

$$B_{\text{imp}} = \{\delta_0, \delta_1, \dots, \delta_{N-1}\} = \{I\delta_0, A\delta_0, \dots, A^{N-1}\delta_0\}. \quad (5.9)$$

To prove  $B_{\text{imp}}$  is a basis, introduce the *vertex impulsive* matrix  $D_{\text{imp}}$  with columns the vectors in  $B_{\text{imp}}$ :

$$D_{\text{imp}} \triangleq [\delta_0 \ \delta_1 \ \dots \ \delta_{N-1}] = [A^0\delta_0 \ A\delta_0 \ \dots \ A^{N-1}\delta_0]. \quad (5.10)$$

We relate  $D_{\text{imp}}$  to a Vandermonde matrix  $\mathcal{V}$ .

**Result 5.1** (Vertex impulsive and Vandermonde matrices).

$$D_{\text{imp}} \xrightarrow{\mathcal{F}} \frac{1}{\sqrt{N}} \mathcal{V}, \quad (5.11)$$

where  $\mathcal{V}$  is the Vandermonde matrix

$$\mathcal{V} = \begin{bmatrix} \lambda^0 & \dots & \lambda^{N-1} \end{bmatrix} = \begin{bmatrix} 1 & \lambda_0 & \lambda_0^2 & \dots & \lambda_0^{N-1} \\ 1 & \lambda_1 & \lambda_1^2 & \dots & \lambda_1^{N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_{N-1} & \lambda_{N-1}^2 & \dots & \lambda_{N-1}^{N-1} \end{bmatrix}. \quad (5.12)$$

*Proof.* This result follows by using (3.47) for  $\delta_n$  in  $D_{\text{imp}}$ . ■

**Result 5.2** (Full rank of vertex impulsive matrix). *Under assumption 2.2,  $D_{\text{imp}}$  is full rank.*

*Proof.* By result 5.1 and equation (5.11),  $D_{\text{imp}}$  is the GFT<sup>-1</sup> of the Vandermonde matrix  $\mathcal{V}$ . Under assumption 2.2,  $\mathcal{V}$  is full rank [66, 68, 69]. Hence,  $D_{\text{imp}}$  is full rank. ■

**Result 5.3** (Vertex impulsive basis). *Under assumption 2.2,  $B_{\text{imp}}$  is a basis—the vertex impulsive basis.*

*Proof.* The vectors in  $B_{\text{imp}}$  are the columns of  $D_{\text{imp}}$ , which by result 5.2 is full rank. Hence,  $B_{\text{imp}}$  is a basis. ■

**Definition 5.4** (Vertex impulsive representation  $p$ ). *The vertex impulsive representation of graph signal  $s$  is its coordinate vector  $p$  with respect to basis  $B_{\text{imp}}$ :*

$$s = p_0\delta_0 + p_1\delta_1 + \cdots + p_{N-1}\delta_{N-1} \quad (5.13)$$

$$= \underbrace{\begin{bmatrix} \delta_0 & \delta_1 & \cdots & \delta_{N-1} \end{bmatrix}}_{D_{\text{imp}}} \underbrace{\begin{bmatrix} p_0 \\ \cdots \\ p_{N-1} \end{bmatrix}}_p \quad (5.14)$$

**Computing  $p$ .** To find the coordinate vector  $p$  of  $s$  with respect to  $B_{\text{imp}}$ , in general, we solve the linear system (5.14). In practice, a sparse approximation may suffice by minimizing  $\|D_{\text{imp}}p - s\|_2^2 + \|p\|_1$ .

### 5.2.2 Polynomial Transform Filter

Next, we interpret the vertex impulsive representation  $p$  as the coefficients of a linear shift invariant (LSI) graph filter.

**Result 5.4** ( $s$  as impulse response of  $p(A)$ ). *Let assumption 2.2 hold. Then the graph signal  $s$  is the impulse response*

$$s = p(A)\delta_0 \quad (5.15)$$

*of the LSI polynomial filter*

$$p(A) = p_0I + p_1A + \cdots + p_{N-1}A^{N-1} \quad (5.16)$$

*iff the vector of coefficients  $p_{\text{coef}}$  of  $p(A)$  is the vertex impulsive representation  $p$  in (5.14):*

$$p_{\text{coef}} = \left[ p_0 \ p_1 \ \cdots \ p_{N-1} \right]^T = p. \quad (5.17)$$

*Proof.* The impulse response of the LSI  $p(A)$  is

$$s = p(A)\delta_0 \quad (5.18)$$

$$= [p_0I + p_1A + \cdots + p_{N-1}A^{N-1}] \delta_0 \quad (5.19)$$

$$= p_0I\delta_0 + p_1A\delta_0 + \cdots + p_{N-1}A^{N-1}\delta_0 \quad (5.20)$$

$$= \underbrace{\left[ \delta_0 \ \delta_1 \ \cdots \ \delta_{N-1} \right]}_{D_{\text{imp}}} p_{\text{coef}}. \quad (5.21)$$

Under assumption 2.2, the impulse response of  $p(A)$  is the graph signal  $s$  iff  $p_{\text{coef}}$  in (5.21) equals  $p$  in (5.14). ■

**Definition 5.5** (Polynomial transform filter). *The LSI polynomial filter  $p(A)$  in (5.16) is the polynomial transform filter of  $s$ .*

The polynomial transform filter  $p(A)$  in (5.16) is in powers of  $A$ . We provide an alternative description.

**Result 5.5** (Graph signal  $s$  and  $p(A)$ ). *Given  $s \xleftrightarrow{\mathcal{F}} \hat{s}$ , its LSI polynomial transform filter  $p(A)$*

is alternatively given by

$$p(A) = GFT^{-1} \text{diag} \left[ \sqrt{N} \hat{s} \right] GFT \quad (5.22)$$

*Proof.* From  $s$  given as impulse response of  $p(A)$  in (5.19), using the diagonalization of  $p(A)$ , it successively follows

$$p(A)\delta_0 = GFT^{-1} \cdot p(\Lambda) \cdot GFT \cdot GFT^{-1} \cdot \frac{1}{\sqrt{N}} \mathbf{1} = s \quad (5.23)$$

$$\implies p(\Lambda) \frac{1}{\sqrt{N}} \mathbf{1} = \hat{s} \implies \frac{1}{\sqrt{N}} p(\Lambda) = \text{diag} [\hat{s}] \quad (5.24)$$

$$\implies p(A) = GFT^{-1} \text{diag} \left[ \sqrt{N} \hat{s} \right] GFT, \quad (5.25)$$

where we used the definition of  $\delta_0 = GFT^{-1} \cdot \frac{1}{\sqrt{N}} \mathbf{1}$ . ■

### 5.2.3 Spectrum Vertex Impulse Representation

Take the GFT of both sides of (5.13). Using result 5.1 and equation (3.47), obtain the representation of  $\hat{s}$  with respect to

$$B_\lambda = \left\{ \frac{1}{\sqrt{N}} \mathbf{1}, \frac{1}{\sqrt{N}} \lambda, \dots, \frac{1}{\sqrt{N}} \lambda^{N-1} \right\}. \quad (5.26)$$

The set  $B_\lambda$  is a basis. In fact, its vectors, apart the scaling factor  $\frac{1}{\sqrt{N}}$ , are the columns of the Vandermonde matrix  $\mathcal{V}$ , and  $\mathcal{V}$  is full rank under assumption 2.2.

**Definition 5.6** (Spectrum vertex impulsive representation). *The spectrum vertex impulsive*

representation of  $\widehat{s} = \text{GFT } s$  is the coordinate vector  $p$  with respect to basis  $B_\lambda$ .

$$\widehat{s} = p_0 \widehat{\delta}_0 + p_1 \widehat{\delta}_1 + \cdots + p_{N-1} \widehat{\delta}_{N-1} \quad (5.27)$$

$$= \underbrace{\begin{bmatrix} \frac{1}{\sqrt{N}} \lambda^0 & \frac{1}{\sqrt{N}} \lambda & \cdots & \frac{1}{\sqrt{N}} \lambda^{N-1} \end{bmatrix}}_{\frac{1}{\sqrt{N}} \mathcal{V}} \underbrace{\begin{bmatrix} p_0 \\ \cdots \\ p_{N-1} \end{bmatrix}}_p \quad (5.28)$$

The spectrum vertex impulsive representation for  $\widehat{s}$  has the same coordinate vector  $p$  as the vertex impulsive representation for  $s$ . It is the basis that is different. Now, it is the frequency vector  $\lambda$  and its powers that are the basis vectors for this representation of  $\widehat{s}$ .

From (5.28), take the  $\text{GFT}^{-1}$  of both sides to obtain  $s$ .

$$s = \text{GFT}^{-1} \left( \underbrace{\begin{bmatrix} \frac{1}{\sqrt{N}} \lambda^0 & \frac{1}{\sqrt{N}} \lambda & \cdots & \frac{1}{\sqrt{N}} \lambda^{N-1} \end{bmatrix}}_{\frac{1}{\sqrt{N}} \mathcal{V}} \underbrace{\begin{bmatrix} p_0 \\ \cdots \\ p_{N-1} \end{bmatrix}}_p \right) \quad (5.29)$$

Equations (5.13) and (5.29) both provide expressions for  $s$  using  $p$ . The difference is (5.13) is a signal representation of  $s$  using  $p$  and shifted delta functions as the basis and (5.29) is the inverse GFT of a signal representation for  $\widehat{s}$  using  $p$  and the powers of the eigenvalues as the basis.

### 5.3 Spectral Impulsive Representation

In Chapter 3, we introduced  $\text{GSP}_{\text{sp}}$ , the dual of GSP, starting from the spectral domain instead of the vertex domain. Section 5.2 presents the representation of graph signals with respect to the basis  $B_{\text{imp}}$  whose basis vectors are the vertex impulse  $\delta_0$  and its delayed replicas defined in section 3.2.1. In this section, we dualize the vertex impulsive representation from previous

section 5.2, using spectral shift  $M$  and the spectral delta functions ( $\widehat{\delta}_{sp,0}$ ) instead of vertex shift  $A$  and  $\delta_0$ . We consider several representations for the *spectral* graph signal,  $\widehat{s}$ : 1) as a linear combination of graph *spectral* impulses; 2) as the impulse response of a *spectral* graph filter; 3) as the GFT of a linear combination of the conjugate powers of the eigenvalues.

### 5.3.1 Spectral Impulsive Representation

This section considers the representation of graph signal  $\widehat{s}$  with respect to  $\widehat{\delta}_{sp,0}$  and its delayed replicas. Consider the set of  $\widehat{\delta}_{sp,0}$  and its spectral shifts:

$$\widehat{B}_{sp,imp} = \left\{ \widehat{\delta}_{sp,0}, \widehat{\delta}_{sp,1}, \dots, \widehat{\delta}_{sp,N-1} \right\} \quad (5.30)$$

$$= \left\{ M^0 \widehat{\delta}_{sp,0}, M \widehat{\delta}_{sp,0}, \dots, M^{N-1} \widehat{\delta}_{sp,0} \right\}. \quad (5.31)$$

Collect the vectors in  $B_{sp,imp}$  in the spectral impulse matrix<sup>1</sup>

$$D_{sp,imp} = \begin{bmatrix} \delta_{sp,0} & \delta_{sp,1} & \dots & \delta_{sp,N-1} \end{bmatrix}. \quad (5.32)$$

The set  $\widehat{B}_{sp,imp}$  is a basis, because its vectors are the GFT of the vectors of the basis  $B_{sp,imp}$ .

**Definition 5.7** (Spectral impulsive representation). *The spectral impulsive representation of  $\widehat{s} = GFT s$  is the coordinate vector  $q$  with respect to basis  $\widehat{B}_{sp,imp}$ .*

$$\widehat{s} = q_0 \widehat{\delta}_{sp,0} + q_1 \widehat{\delta}_{sp,1} + \dots + q_{N-1} \widehat{\delta}_{sp,N-1} \quad (5.33)$$

$$= \underbrace{\begin{bmatrix} \widehat{\delta}_{sp,0} & \widehat{\delta}_{sp,1} & \dots & \widehat{\delta}_{sp,N-1} \end{bmatrix}}_{\widehat{D}_{sp,imp}} \underbrace{\begin{bmatrix} q_0 \\ \dots \\ q_{N-1} \end{bmatrix}}_q \quad (5.34)$$

<sup>1</sup> Note that the columns of  $D_{sp,imp}$  are flat, not impulsive.

### 5.3.2 Spectral Polynomial Transform Filter

We proceed to obtain results similar to sections 5.2.2 for  $q$ . We start by associating with it a linear shift invariant (LSI) spectral polynomial transfer filter  $q(M)$ , now in the spectral shift  $M$ .

**Result 5.6** ( $\widehat{s}$  as impulse response of  $q(M)$ ). *Let assumption 2.2 hold. Then  $\widehat{s}$  is the impulse response of LSI filter  $q(M)$*

$$\widehat{s} = q(M)\widehat{\delta}_{sp,0} \quad (5.35)$$

$$q(M) = q_0I + q_1M + \cdots + q_{N-1}M^{N-1}, \quad (5.36)$$

iff the vector of coefficients  $q_{\text{coef}}$  of  $q(M)$  is the spectral impulsive representation  $q$  in (5.34):

$$q_{\text{coef}} = \begin{bmatrix} q_0 & q_1 & \cdots & q_{N-1} \end{bmatrix}^T = q. \quad (5.37)$$

Filter  $q(M)$  is the *spectral* polynomial transform filter.

Note that equation (5.43) can be rewritten as

$$s = q(\Lambda^*)\delta_{sp,0} = \left( q_0I + q_1\Lambda^* + \cdots + q_{N-1}(\Lambda^*)^{N-1} \right) \frac{1}{\sqrt{N}}\mathbf{1}. \quad (5.38)$$

In other words, (5.38) interprets the original signal  $s$  as the impulse response of the diagonal filter  $q(\Lambda^*)$ . From this, the next result follows.

**Result 5.7** (Graph signal  $\widehat{s}$  and  $q(M)$ ). *The LSI polynomial transform filter  $q(M)$  is alternatively given by*

$$q(M) = GFT \text{diag} \left[ \sqrt{N}s \right] GFT^{-1} \quad (5.39)$$

Results 5.6 and 5.7 parallel results 5.4 and 5.5.

### 5.3.3 Spectrum Spectral Impulse Representation

Consider the set of  $\delta_{sp,0}$  and its spectral shifts:

$$B_{sp,imp} = \{\delta_{sp,0}, \delta_{sp,1}, \dots, \delta_{sp,N-1}\}. \quad (5.40)$$

**Result 5.8** ( $D_{sp,imp}$  and  $\mathcal{V}^*$ ).

$$D_{sp,imp} = \frac{1}{\sqrt{N}} \mathcal{V}^* = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & \lambda^* & \dots & \lambda^{*N-1} \end{bmatrix}. \quad (5.41)$$

*Proof.* Result follows from equation (3.49). ■

Equation (5.41) shows that the vectors of the set  $B_{sp,imp}$  are, apart a scaling, the columns of  $\mathcal{V}^*$ .

**Result 5.9** (Spectral impulse basis  $B_{sp,imp}$ ). *Under assumption 2.2,  $B_{sp,imp}$  is a basis.*

*Proof.* By assumption 2.2,  $\mathcal{V}^*$  is full rank. ■

**Definition 5.8** (Spectrum spectral impulse representation). *The spectral impulsive representation of graph signal  $s$  is its coordinate vector  $q_{sp,imp}$  with respect to basis  $B_{sp,imp}$ :*

$$s = q_0 \delta_{sp,0} + q_1 \delta_{sp,1} + \dots + q_{N-1} \delta_{sp,N-1} \quad (5.42)$$

$$= D_{sp,imp} \underbrace{\begin{bmatrix} q_0 \\ q_1 \\ \dots \\ q_{N-1} \end{bmatrix}}_q = \frac{1}{\sqrt{N}} \mathcal{V}^* q \quad (5.43)$$

The spectral impulsive representation for  $\hat{s}$  has the same coordinate vector  $q$  as the spectral impulsive representation for  $s$  (see (5.34) and (5.43)). It is the basis that is different. Now,

it is the spectral impulse  $\widehat{\delta}_{\text{sp},0}$  and its powers that are the basis vectors for this representation of  $\widehat{s}$ .

Taking the GFT of both sides of (5.43) yields

$$\widehat{s} = GFT \left( \frac{1}{\sqrt{N}} \mathcal{V}^* q \right) \quad (5.44)$$

Equations (5.33) and (5.44) both provide expressions for  $\widehat{s}$  using  $q$ . The difference is (5.33) is a signal representation of  $\widehat{s}$  using  $q$  and shifted spectral delta functions as the basis and (5.44) is the GFT of a signal representation for  $s$  using  $q$  and the powers of the eigenvalues as the basis.

**Result 5.10** (Relation between  $p$  and  $q$ ).

$$q = (\mathcal{V}^*)^{-1} GFT^{-1} \mathcal{V} p \text{ and } p = \mathcal{V}^{-1} GFT \mathcal{V}^* q. \quad (5.45)$$

*Proof.* It follows from (5.28) and (5.34), using (5.41) in result 5.8. ■

**Remark 5.3.** *A note on notation:  $p$  are the coefficients of  $p(A)$  and  $q$  are the coefficients of  $q(M)$ . The thesis uses  $p(\cdot)$  as a polynomial with coefficients  $p$ . Similarly, the thesis uses  $q(\cdot)$  as a polynomial with coefficients  $q$ . For example,  $p(x)$ ,  $p(A)$  have the same coefficients  $p$ , but  $p(x)$  is in terms of  $x$  and  $p(A)$  is in terms of  $A$ . The conversion between  $p(\cdot)$  and  $p$  (similarly with  $q(\cdot)$  and  $q$ ) is used frequently in the thesis and is not explicitly stated each time.*

*Also, unless otherwise specified, the signal  $p$  is related to  $s$  through (5.13) and the signal  $q$  is related to  $\widehat{s}$  through (5.33). This conversion also is used frequently and is not explicitly stated.*

## 5.4 Companion Model—a Canonical GSP Model

In DSP, the DSP cyclic shift  $A$  in (2.1) acts on graph signal  $s$  as given in (2.2). Decompose the cyclic shift as

$$A_c = \underbrace{\begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \\ \vdots & 1 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}}_{A_{c,\text{line shift}}} + \underbrace{\begin{bmatrix} 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & 0 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix}}_{A_{c,\text{periodic bc}}}. \quad (5.46)$$

Then the shifted time signal  $As$  is delayed (moved downwards) by the *line shift* (left block in (5.46)) and the signal extension  $s_N$  is determined by the periodic boundary condition (right block in (5.46)) [6–8] that wraps around the time signal so that sample  $s_{N-1}$  reappears as the first component of  $As$ .

In this section, we look for a GSP signal model where the graph shift acts in similar fashion to (5.46). We accomplish it with the *impulsive* GSP signal representation. The resulting GSP model leads to the *companion* shift and the *companion* graph. These are *canonical* shift and *canonical* graph representations to which, under assumptions 2.1 and 2.2, every other generic GSP model can be reduced to.

### 5.4.1 Canonical Companion Shift

To obtain the representation of  $A$  with respect to  $B_{\text{imp}}$ , we apply the shift to each vector  $\delta_n \in B_{\text{imp}}$ . Get

$$A\delta_0 = \delta_1, \dots, A\delta_n = A^{n+1}\delta_0 = \delta_{n+1}, \dots, A\delta_{N-2} = \delta_{N-1}. \quad (5.47)$$

We need a “signal extension” or “boundary condition” for

$$A\delta_{N-1} = A^N\delta_0. \quad (5.48)$$

This boundary condition is embedded in the matrix  $A$  and is obtained by reducing it by Cayley-Hamilton Theorem. Applying then this theorem through equation (2.16),

$$A\delta_{N-1} = -c_0I\delta_0 - c_1A\delta_0 - c_2A^2\delta_0 - \cdots - c_{N-1}A^{N-1}\delta_0. \quad (5.49)$$

The boundary condition in (5.49) for  $A\delta_{N-1}$  is a linear combination of the basis vectors  $A\delta_n \in B_{\text{imp}}$ . The coefficients of the linear combination are the negative of the coefficients  $c_n$  of the characteristic polynomial  $\Delta_A(x)$  of  $A$  given in (2.15).

Putting together the  $N$  equations (5.47)-(5.48) and using the boundary condition (5.49),

$$A \begin{bmatrix} \delta_0 & \delta_1 & \cdots & \delta_{N-1} \end{bmatrix} = \begin{bmatrix} \delta_1 & \delta_2 & \cdots & \delta_N \end{bmatrix} \quad (5.50)$$

$$= \begin{bmatrix} \delta_0 & \delta_1 & \cdots & \delta_{N-1} \end{bmatrix} \underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 & -c_0 \\ 1 & 0 & \cdots & 0 & -c_1 \\ 0 & 1 & \ddots & 0 & -c_2 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & -c_{N-1} \end{bmatrix}}_{C_{\text{comp}}}. \quad (5.51)$$

Equation (5.51) shows that the representation of the shift with respect to  $B_{\text{imp}}$  is the *companion matrix*  $C_{\text{comp}}$ . It is the *companion matrix* [66, 68, 69] of the characteristic polynomial  $\Delta_A(x)$  of

the graph shift  $A$ . We refer to  $C_{\text{comp}}$  as the *companion shift*. We can rewrite it as:

$$C_{\text{comp}} = \underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}}_{A_c} + \underbrace{\begin{bmatrix} -1 - c_0 \\ -c_1 \\ \cdots \\ -c_{N-1} \end{bmatrix}}_{\text{rank 1}} [0 \ 0 \ \cdots \ 0 \ 1]. \quad (5.52)$$

Equation (5.52) gives  $C_{\text{comp}}$  as the sum of a unitary matrix, the DSP cyclic shift  $A_c$ , plus a rank one matrix. On the other hand, we may decompose  $C_{\text{comp}}$  as:

$$C_{\text{comp}} = \underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}}_{C_{\text{line shift}}} + \underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 & -c_0 \\ 0 & 0 & \cdots & 0 & -c_1 \\ 0 & 0 & \cdots & 0 & -c_2 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & -c_{N-1} \end{bmatrix}}_{C_{\text{linear bc}}}. \quad (5.53)$$

Equation (5.53) resolves  $C_{\text{comp}}$  as a ‘line shift’  $C_{\text{line shift}}$  corrected by a ‘boundary condition’  $C_{\text{linear bc}}$ . It replicates the structure of the DSP cyclic shift given in (5.46). Like  $A_{c,\text{line shift}}$  in (5.46),  $C_{\text{line shift}}$  moves the graph signal downwards, while  $C_{\text{linear bc}}$  retains the coefficients  $\{-c_n\}_{0 \leq n \leq N-1}$  of the boundary condition. This is a more general boundary condition than for the cyclic shift, since for  $A_{c,\text{periodic bc}}$  all  $c_n = 0$ , except  $c_0 = -1$ , see (5.46). This agrees with the characteristic polynomial of the DSP cyclic shift of for which  $\Delta_{A_c}(x) = x^N - 1$ .

Since  $C_{\text{comp}}$  is determined by the characteristic polynomial  $\Delta_A(x)$ , it only depends on the graph frequencies or eigenvalues of  $A$ , not on the spectral modes or eigenvectors of  $A$ . And this shows that, under diagonalization of  $A$ , we can associate to arbitrary adjacency matrices a *canonical* weighted adjacency matrix, its *companion* shift.

**Result 5.11** (Diagonalization of  $C_{\text{comp}}$ ). *Under assumption 2.2,  $C_{\text{comp}}$  is diagonalized by the Vandermonde matrix*

$$C_{\text{comp}} = \mathcal{V}^{-1} \Lambda \mathcal{V}. \quad (5.54)$$

*Proof.* This is a well known result. It can be verified by direct substitution that  $[1 \ \lambda_i \ \cdots \ \lambda_i^{N-1}]$  is a left eigenvector of  $C_{\text{comp}}$  for eigenvalue  $\lambda_i$ , from which the result follows. ■

*Companion graph Fourier transform.* Given (5.54), the Vandermonde matrix  $\mathcal{V}$  is the graph Fourier transform for signals in impulsive representation, replicating the DSP result where the DFT is the Vandermonde matrix of the eigenfrequencies (apart a normalizing factor), see (2.8).

This shows that the impulsive representation replicates for GSP another dimension of DSP. In fact, just like for DSP, the eigenvalues (frequencies) provide the whole picture, since the companion graph Fourier transform is defined by the frequency vectors  $\lambda$  and its powers.

Next, we associate a weighted *companion* graph  $G_{\text{comp}}$  to  $C_{\text{comp}}$ . Both of these,  $G_{\text{comp}}$  and  $C_{\text{comp}}$ , are *canonical* graph representations connected with any GSP graph.

### 5.4.2 Canonical Companion Graph

The companion matrix  $C_{\text{comp}}$  defines the (weighted) companion graph  $G_{\text{comp}} = (V_{\text{comp}}, E_{\text{comp}})$  displayed in figure 5.1. Under assumption 2.2, any directed or undirected signal graph  $G$  has a corresponding weighted *companion* graph.

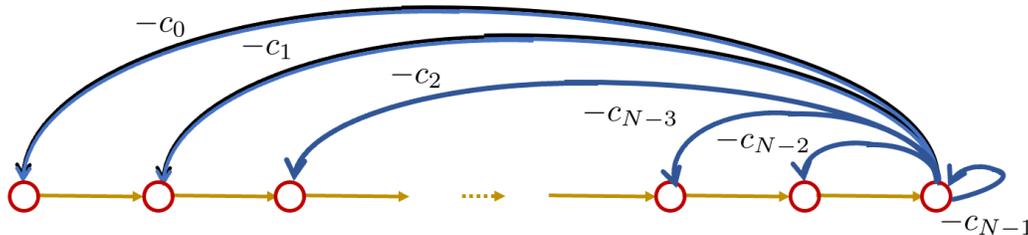


Figure 5.1: Companion graph. Unlabelled edges have weight 1. Other edges labeled by their weights.

The structure of the companion graph in figure 5.1 extends the structure of the DSP cyclic graph in figure 2.1. The DSP cyclic graph follows from the companion graph of figure 5.1 by taking  $c_0 = -1$  and eliminating the self-loop and all the remaining backward pointing edges.

The *companion* graph  $G_{\text{comp}}$  has a canonical structure: 1) its node set  $V_{\text{comp}}$  has  $N$  nodes, node  $n$  is associated with basis vector  $\delta_n \in B_{\text{imp}}$  (or power  $A^n$ ). In other words, these nodes are not the nodes of the original graph  $G$  associated with  $A$ ; 2) it is directed; 3) the edge set  $E_{\text{comp}}$  combines a directed path graph with possibly a self-loop at node  $N - 1$  and up to  $N - 1$  directed backward edges pointing from node  $N - 1$  to the previous nodes; 4) these directed edges are weighted by the negative of the coefficient  $c_n$  of  $\Delta_A(x)$ ; 5) iff  $c_0 \neq 0$ , the companion graph is strongly connected. This is the case if zero is not an eigenvalue of  $A$ .

### 5.4.3 Example

Figure 5.2 shows on top a “directed” ladder graph with 12 nodes and below it the corresponding canonical graph. The characteristic polynomial of the adjacency matrix of a ladder graph like shown in the figure but with  $2k$  nodes is

$$\Delta_A(x) = -1 - x^2 - x^4 - x^8 - \dots - x^{2(k-2)} + x^{2k}. \quad (5.55)$$

The polynomial  $\Delta_A(x)$  explains why the edge weights of the companion graph of the directed ladder graph are all ones (the coefficients of  $\Delta_A(x)$  are  $c_n \equiv -1$ ). The eigenfrequencies of this directed ladder graph are illustrated for  $k=4, 6, 8, 10, 12$ , and 14 nodes in figure 5.3. They distribute close to the unit circle.

As another example, consider the undirected  $N$  node path. Its characteristic polynomial is the 3-term recursion

$$\Delta_N(x) = x\Delta_{N-1}(x) - \Delta_{N-2}(x), \quad \Delta_0(x) = 1, \Delta_1(x) = \frac{x}{2}. \quad (5.56)$$

This gives  $\Delta_N(x) = U\left(\frac{x}{2}\right)$  where  $U(x)$  is the Chebyshev polynomial of the second kind [76].

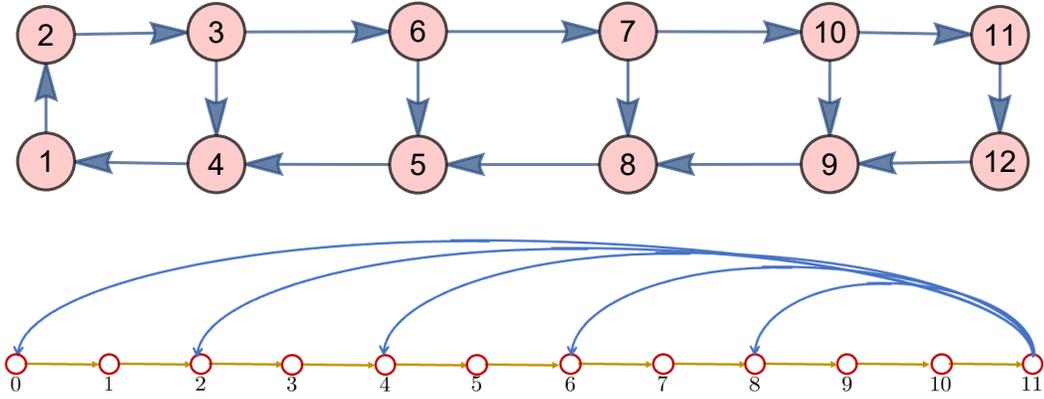


Figure 5.2: Directed ladder graph and its companion graph.

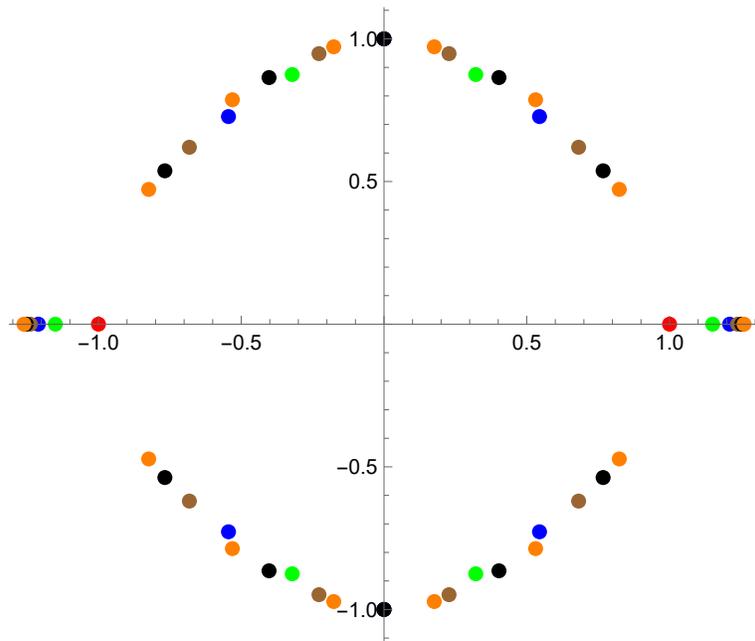


Figure 5.3: Eigenfrequencies of directed ladder graphs with 4 (red), 6 (green), 8 (blue), 10 (brown), 12 (black), and 14 (orange) nodes.

For example, for  $N = 8$

$$\Delta_8(x) = x^8 - 7x^6 + 15x^4 - 10x^2 + 1 \quad (5.57)$$

The path and its companion graph are in figure 5.4.

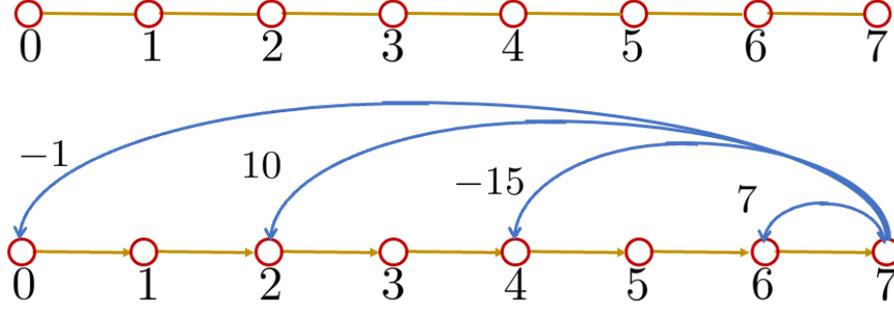


Figure 5.4: Path graph and its canonical companion graph.

#### 5.4.4 Spectral Companion Model: Canonical Companion Matrix and Graph

We now consider the dual to the vertex companion model developed in the previous section 5.4.1, the spectral companion model. Following  $\text{GSP}_{\text{sp}}$ , we start from the spectral domain and develop the spectral companion model using spectral shift  $M$ .

To obtain the representation of  $M$  with respect to  $\widehat{B}_{\text{sp,imp}}$ , we apply the shift to each vector  $\widehat{\delta}_{\text{sp},n} \in \widehat{B}_{\text{sp,imp}}$ . Get

$$M\widehat{\delta}_{\text{sp},0} = \widehat{\delta}_{\text{sp},1}, \dots, M\widehat{\delta}_{\text{sp},n} = M^{n+1}\widehat{\delta}_{\text{sp},0} = \widehat{\delta}_{\text{sp},n+1}, \dots, M\widehat{\delta}_{\text{sp},N-2} = \widehat{\delta}_{\text{sp},N-1}. \quad (5.58)$$

We use a similar “signal extension” or “boundary condition” as (5.48):

$$M\widehat{\delta}_{\text{sp},N-1} = M^N\widehat{\delta}_{\text{sp},0} \quad (5.59)$$

This boundary condition is embedded in the matrix  $M$  and is obtained by reducing it by Cayley-Hamilton Theorem. Applying then this theorem through equation (2.16),

$$M\widehat{\delta}_{\text{sp},N-1} = -c_0 I\widehat{\delta}_{\text{sp},0} - c_1 M\widehat{\delta}_{\text{sp},0} - c_2 M^2\widehat{\delta}_{\text{sp},0} - \dots - c_{N-1} M^{N-1}\widehat{\delta}_{\text{sp},0}. \quad (5.60)$$

The boundary condition in (5.60) for  $M\widehat{\delta}_{\text{sp},N-1}$  is a linear combination of the basis vectors

$M\widehat{\delta}_{\text{sp},n} \in \widehat{B}_{\text{sp,imp}}$ . The coefficients of the linear combination are the negative of the coefficients  $c_n$  of the characteristic polynomial  $\Delta_M(x)$  of  $M$  given in (2.15).

Putting together the  $N$  equations (5.58)-(5.59) and using the boundary condition (5.60),

$$\begin{aligned}
 M \begin{bmatrix} \widehat{\delta}_{\text{sp},0} & \widehat{\delta}_{\text{sp},1} & \cdots & \widehat{\delta}_{\text{sp},N-1} \end{bmatrix} &= \begin{bmatrix} \widehat{\delta}_{\text{sp},1} & \widehat{\delta}_{\text{sp},2} & \cdots & \widehat{\delta}_{\text{sp},N} \end{bmatrix} & (5.61) \\
 &= \begin{bmatrix} \widehat{\delta}_{\text{sp},0} & \widehat{\delta}_{\text{sp},1} & \cdots & \widehat{\delta}_{\text{sp},N-1} \end{bmatrix} \underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 & -c_0 \\ 1 & 0 & \cdots & 0 & -c_1 \\ 0 & 1 & \ddots & 0 & -c_2 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & -c_{N-1} \end{bmatrix}}_{C_{\text{comp}}} \cdot & (5.62)
 \end{aligned}$$

Equation (5.62) shows that the representation of the shift with respect to  $\widehat{B}_{\text{sp,imp}}$  is the *companion matrix*  $C_{\text{comp}}$ . It is the *companion matrix* [66, 68, 69] of the characteristic polynomial  $\Delta_M(x)$  of the graph shift  $M$ .

Since  $A$  and  $M$  are co-spectral (share the same spectrum), their characteristic polynomials  $\Delta_A(\lambda)$  and  $\Delta_M(\lambda)$  are equal. This means that we can associate with the spectral impulse representation the same companion matrix  $C_{\text{comp}}$  and the same companion graph  $G_{\text{comp}}$  as in sections 5.4.1, equation (5.51), and 5.4.2, respectively.

## 5.5 Conclusion

This chapter introduces six signal representations and the canonical companion model. The six representations are the vertex, Fourier, vertex impulsive, spectrum vertex impulsive, spectral impulsive, and spectrum spectral impulsive representations. The vertex, vertex impulsive, and spectrum spectral impulsive are all signal representations for vertex domain signal  $s$  with different bases. The Fourier, spectral impulsive, and spectrum vertex impulsive are all signal

representations for spectral domain signal  $\widehat{s}$  (dual to the signal representations for  $s$ ) with different bases. These signal representations lead to the canonical companion model defined by a *canonical* graph and *canonical* shift, the *companion* graph and the *companion* shift. The companion shift can be viewed as the path graph with a boundary condition, determined by the Cayley-Hamilton Theorem. Shifting in the vertex domain by  $A$  or shifting in the spectral domain by  $M$  are both equivalent to shifting by  $C_{\text{comp}}$  in the canonical companion model.

# Chapter 6

## The Graph $z$ -Transform (GzT)

In Chapter 5, we introduced several signal representations and the canonical companion signal model. In this chapter, we introduce the graph  $z$ -transform (GzT); and through the GzT provide a symbolic polynomial representation for graph signals. We show the graph  $z$ -transform and its dual, the spectral graph  $z$ -transform, and their properties. We show that using the graph  $z$ -transform can lead to a fast graph convolution using the FFT.

### 6.1 Vertex and Spectral Graph $z$ -transforms

#### 6.1.1 Graph $z$ -transform (GzT)

The powers of the shift  $A$  of the polynomial transform filter  $p(A)$  in (5.16) represent the GSP equivalent of the powers of the DSP shift  $z^{-1}$ . This motivates the following definition.

**Definition 6.1** (Graph  $z$ -transform (GzT)). *The GzT is*

$$\text{GzT} = D_{\text{imp}}^{-1} = \left[ \begin{array}{cccc} \delta_0 & \delta_1 & \cdots & \delta_{N-1} \end{array} \right]^{-1}. \quad (6.1)$$

In diagram form, the GzT of  $s$  and its reconstruction are:

$$s \xrightarrow{\text{GzT}} p = \text{GzT } s \xrightarrow{\text{GzT}^{-1}} s \quad (6.2)$$

**Result 6.1** ( $GzT^{-1}$  and  $\mathcal{V}$ ). *The  $GzT^{-1}$  and  $\mathcal{V}$  are GFT pairs:*

$$GzT^{-1} \xrightarrow{GFT} \frac{1}{\sqrt{N}} \mathcal{V} = GFT \cdot GzT^{-1}. \quad (6.3)$$

*Proof.* This follows from result 5.1 and definition 6.1. ■

$GzT$  maps vertex signals  $s$  into  $z$ -transformed signals  $p$ .

The  $GzT$  of  $s$  is the polynomial coefficient vector  $p$  in (5.17) that defines  $p(A)$  in (5.16). To simplify notation, we introduce a symbolic polynomial representation  $p(x)$ .

**Definition 6.2** (Graph  $z$ -transform representation  $p(x)$ ). *The graph  $z$ -transform representation of graph signal  $s$  is its coordinate vector  $p$  with respect to the monomial basis  $B_{\text{monomial}} = \{1, x, \dots, x^{N-1}\}$ :*

$$GzTs = p(x) = p_0 + p_1 x + p_2 x^2 + \dots + p_{N-1} x^{N-1} \quad (6.4)$$

$$= \begin{bmatrix} 1 & x & x^2 & \dots & x^{N-1} \end{bmatrix} p. \quad (6.5)$$

**Result 6.2.** (*Shifting in vertex domain and graph  $z$ -transform*) *The  $z$ -transform of shifted signal  $As$  is the shifted signal  $C_{\text{comp}} p$ .*

$$As \xrightarrow{GzT} C_{\text{comp}} p \quad (6.6)$$

*Proof.* From (5.19),

$$s = [p_0 I + p_1 A + \dots + p_{N-1} A^{N-1}] \delta_0.$$

Shifting by  $s$  using  $A$  yields

$$As = [p_0 A + p_1 A^2 + \dots + p_{N-2} A^{N-1} + p_{N-1} A^N] \delta_0.$$

We apply the boundary condition from (5.49) to  $A_N$  and simplify.

$$As = \left( [p_0A + p_1A^2 + \dots + p_{N-2}A^{N-1}] + p_{N-1} [-c_0I - c_1A - c_2A^2 - \dots - c_{N-1}A^{N-1}] \right) \delta_0.$$

Taking the  $z$ -transform of both sides yields  $C_{\text{comp}} p$ . ■

**Remark 6.1** (Various meanings for  $p$ ). *We have multiple interpretations for  $p$ : 1) as coordinate vector of  $s$  with respect to basis  $B_{\text{imp}}$  in (5.14); 2) determining the polynomial transform filter  $p(A)$  in (5.17); 3) as  $GzT$  of  $s$  in (6.2); and 4) defining  $p(x)$  in (6.5). We take advantage of these several understandings in the different sections and chapters.*

**Remark 6.2.** (Connection between  $z$ -transform and Fourier transform). *The graph  $z$ -transform of  $s$  is  $p(x)$ . From (5.28),  $\sqrt{N}\hat{s} = \mathcal{V}p$ . Since  $\mathcal{V}$  is a Vandermonde matrix of  $\lambda$ ,  $p(\lambda) = \sqrt{N}\hat{s}$ . This means, plugging in  $\lambda$  into  $p(x)$  yields  $\sqrt{N}\hat{s}$ , the  $GFT$  of  $s$  scaled by  $\sqrt{N}$ . This is also true in  $DSP$ . Taking the  $z$ -transform of  $s$  and plugging in the roots of unity in  $DSP$  yields  $\sqrt{N}\hat{s}$ , the  $DFT$  of  $s$ .*

**Result 6.3.** (Permutation invariance of  $z$ -transform) *Let  $A = GFT^{-1}\Lambda GFT$  be an adjacency matrix. Let  $s$  be the graph signal on  $A$ .*

*Let  $A' = \Pi_1 A \Pi_1^T = \Pi_1 GFT^{-1} \Pi_2^T \Pi_2 \Lambda \Pi_2^T \Pi_2 GFT \Pi_1^T$  be the adjacency matrix with a relabelling of the vertices of  $A$  by permutation  $\Pi_1$  and a reordering of the eigenpairs by  $\Pi_2$ . Let  $s' = \Pi_1 s$  be the graph signal on  $A'$ .*

*Then, under assumption 2.2 of unique eigenvalues,  $s$  and  $s'$  have the same  $z$ -transform,  $p(x)$ .*

*Proof.* By (5.28),

$$p = \mathcal{V}^{-1}\sqrt{N}\hat{s}.$$

We need to show that  $p' = p$ .

We group the terms in  $A'$ :

$$A' = \underbrace{(\Pi_1 \text{GFT}^{-1} \Pi_2^T)}_{\text{GFT}'^{-1}} \underbrace{(\Pi_2 \Lambda \Pi_2^T)}_{\Lambda'} \underbrace{(\Pi_2 \text{GFT} \Pi_1^T)}_{\text{GFT}'}$$

Taking the GFT' of  $s'$  yields:

$$\widehat{s}' = \Pi_2 \text{GFT} \Pi_1^T s' = \Pi_2 \text{GFT} \Pi_1^T \Pi_1 s = \Pi_2 \widehat{s}.$$

Note that the reordering of the vertices  $\Pi_1$  has no effect on  $\widehat{s}'$  and the reordering of the eigenpairs  $\Pi_2$  has only reordered the entries of  $\widehat{s}$  (due to the eigenvectors in basis  $B_{\text{Fourier}}$  being reordered).

The Vandermonde matrix of  $A'$  is formed using the reordered eigenvalues  $\Lambda'$ . So,

$$\mathcal{V}' = \Pi_2 \mathcal{V}.$$

Note this is a reordering of the rows of the Vandermonde because the eigenvalues were reordered. Observe that the rows of the Vandermonde and the entries of  $\widehat{s}$  were reordered the same way. Because of assumption 2.2 of unique eigenvalues, there is no ambiguity from repeated eigenvalues.

So,

$$\mathcal{V}' p' = \sqrt{N} \widehat{s}'$$

and

$$p' = \mathcal{V}'^{-1} \Pi_2^T \sqrt{N} \Pi_2 \widehat{s} = \mathcal{V}^{-1} \sqrt{N} \widehat{s} = p.$$

Thus,  $s$  and  $s'$  have the same  $z$ -transform. ■

**Remark 6.3.** *In GSP, two researchers may use different vertex orderings and may have a different ordering of the eigenpairs, e.g.,  $A$  and  $A'$ . This leads to their graph signals  $s$  and  $s'$  (and  $\widehat{s}$  and  $\widehat{s}'$ ) being off by a permutation, making it difficult to directly compare. Result 6.3*

shows that the  $z$ -transform is permutation-invariant. This means that the two researchers can take their graph signals ( $s$  and  $s'$ ) and instead of direct comparison, compare their  $z$ -transforms ( $p$  and  $p'$ ). If  $p = p'$ , then the  $s$  and  $s'$  are permutations of the same signal. Also, given  $p$ , each researcher can plug in their shift  $p(A)\delta_0$  and  $p(A')\delta'_0$  to obtain the graph signal  $s$  and  $s'$  respectively. This allows researchers to share the signal  $p$  instead of  $s$  or  $s'$  and convert it back to the vertex domain to re-obtain the signals for their (permuted) graphs without knowing permutation  $\Pi_1$  or  $\Pi_2$ .

### 6.1.2 Spectral Graph $z$ -transform

Like  $p(A)$  in (5.16) led us to the GzT, we associate with  $q(M)$  a spectral graph  $z$ -transform ( $\widehat{\text{GzT}}_{\text{sp}}$ ). This is the dual of the graph  $z$ -transform, the graph  $z$ -transform in  $\text{GSP}_{\text{sp}}$ . We state the definition and corresponding results (without proof) that parallel those in section 6.1.1.

**Definition 6.3** (Spectral graph  $z$ -transform ( $\widehat{\text{GzT}}_{\text{sp}}$ )). *Define*

$$\widehat{\text{GzT}}_{\text{sp}} = \widehat{D}_{\text{sp,imp}}^{-1} = \left[ \begin{array}{cccc} \widehat{\delta}_{\text{sp},0} & \widehat{\delta}_{\text{sp},1} & \cdots & \widehat{\delta}_{\text{sp},N-1} \end{array} \right]^{-1}. \quad (6.7)$$

**Result 6.4** (Fourier pairs  $\widehat{\text{GzT}}_{\text{sp}}^{-1}$  and  $\mathcal{V}^*$ ).

$$\frac{1}{\sqrt{N}}\mathcal{V}^* \xrightarrow{\text{GFT}} \widehat{\text{GzT}}_{\text{sp}}^{-1} = \text{GFT} \frac{1}{\sqrt{N}}\mathcal{V}^*. \quad (6.8)$$

*Proof.* From (5.43),  $D_{\text{sp,imp}} = \frac{1}{\sqrt{N}}\mathcal{V}^*$ , so, by definition 6.3

$$\widehat{\text{GzT}}_{\text{sp}}^{-1} = \widehat{D}_{\text{sp,imp}} = \text{GFT} D_{\text{sp,imp}} = \text{GFT} \frac{1}{\sqrt{N}}\mathcal{V}^*. \quad (6.9)$$

■

$\widehat{\text{GzT}}_{\text{sp}}$  maps  $\widehat{s}$  into spectral  $z$ -transformed signals  $q$ :

$$\widehat{s} \xrightarrow{\widehat{\text{GzT}}_{\text{sp}}} q = \widehat{\text{GzT}}_{\text{sp}} \widehat{s} \xrightarrow{\widehat{\text{GzT}}_{\text{sp}}^{-1}} \widehat{s} \quad (6.10)$$

The  $\widehat{\text{GzT}}_{\text{sp}}$  of  $\widehat{s}$  is the polynomial coefficient vector  $q$  in (5.34) that defines  $q(M)$  in (5.36) in result 5.6. To simplify notation, we use also a symbolic polynomial representation  $q(x)$  with coefficients given by  $q = \widehat{\text{GzT}}_{\text{sp}} s$

$$\widehat{\text{GzT}}_{\text{sp}} \widehat{s} \approx q(x) = \begin{bmatrix} 1 & x & x^2 & \cdots & x^{N-1} \end{bmatrix} q \quad (6.11)$$

$$= q_0 + q_1 x + q_2 x^2 + \cdots + q_{N-1} x^{N-1}. \quad (6.12)$$

The polynomial  $q(x)$  expresses the  $\widehat{\text{GzT}}_{\text{sp}}$  of  $\widehat{s}$  in terms of the monomial basis  $B_{\text{monomial}} = \{1, x, \dots, x^{N-1}\}$ .

**Result 6.5.** (*Shifting in spectral domain and graph  $z$ -transform*) The spectral graph  $z$ -transform of shifted signal  $M \widehat{s}$  is the shifted signal  $C_{\text{comp}} q$ . This is the dual of result 6.2.

$$M \widehat{s} \xrightarrow{\widehat{\text{GzT}}_{\text{sp}}} C_{\text{comp}} q \quad (6.13)$$

*Proof.* From (5.35),

$$\widehat{s} = [q_0 I + q_1 M + \cdots + q_{N-1} M^{N-1}] \widehat{\delta}_{\text{sp},0}.$$

Shifting by  $\widehat{s}$  using  $M$  yields

$$M \widehat{s} = [q_0 M + q_1 M^2 + \cdots + q_{N-2} M^{N-1} + q_{N-1} M^N] \widehat{\delta}_{\text{sp},0}.$$

We apply the boundary condition from (5.60) to  $M_N$  and simplify.

$$M \widehat{s} = ([q_0 M + q_1 M^2 + \cdots + q_{N-2} M^{N-1}] + q_{N-1} [-c_0 I - c_1 M - c_2 M^2 - \cdots - c_{N-1} M^{N-1}]) \delta_0.$$

Taking the spectral graph  $z$ -transform of both sides yields  $C_{\text{comp}} q$ . ■

**Remark 6.4.** (*Spectral  $z$ -transform in DSP*) The spectral  $z$ -transform is rarely mentioned or considered in DSP. Usually, in DSP, we start with a signal  $s$  in time and consider its  $z$ -transform  $p(z^{-1})$ , writing  $s$  as a linear combination of time shifts. The spectral  $z$ -transform starts with a frequency signal  $\hat{s}$  and considers its  $z$ -transform  $q(z^{-1})$ , writing  $\hat{s}$  as a linear combination of frequency shifts. It is the dual of the  $z$ -transform. Similar to how  $s$  is the coefficients of  $p(z^{-1})$ ,  $\hat{s}$  is the coefficients of  $q(z^{-1})$ .

## 6.2 Fast Graph Convolution with the FFT

Filtering in the *vertex* domain is defined in [1] as the product of *matrix* graph filter  $F$  with graph *vector* signal  $s$ . If the filter is linear shift invariant, it is a polynomial filter  $p(A)$ . We now consider convolution of two graph signals.

**Definition 6.4** (Convolution of vertex domain graph signals). *The (vertex domain) convolution of graph signals  $s$  and  $t$  is*

$$t \circledast s = P_t(A) \cdot P_s(A) \delta_0, \tag{6.14}$$

where  $P_s(A)$  and  $P_t(A)$  are the LSI polynomial transform filters for  $s$  and  $t$ .

Definition 6.4 and equation (6.14) define convolution of graph signals  $s$  and  $t$  as the impulse response of the serial concatenation of the polynomial transform filters  $P_s(A)$  of  $s$  and  $P_t(A)$  of  $t$ . Figure 6.1 illustrates this convolution. Since polynomial filters commute, the convolution

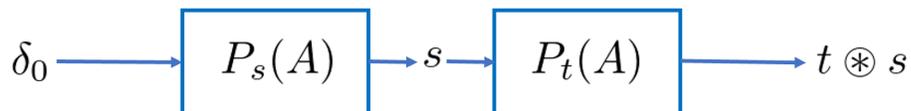


Figure 6.1: Convolution of graphs signals  $s$  and  $t$ .

in (6.14) commutes. The next result provides alternative ways of computing the convolution.

**Result 6.6** (Vertex convolution). *Consider graph signals  $s$  and  $t$  and their polynomial transform filters  $P_s(A)$  and  $P_t(A)$ . Then*

$$t \circledast s = P_t(A) \cdot s = P_s(A) \cdot t \quad (6.15)$$

$$s \circledast t \xleftarrow{\mathcal{F}^{-1}} \sqrt{N} \hat{t} \odot \hat{s} \quad (6.16)$$

*Proof.* Equation (6.15) follows from result 5.4 and equation (5.15).

Equation (6.16) follows by taking the GFT of both sides of (6.14) and using the diagonalization of the transform filters in (5.22). ■

Equation (6.15) interprets convolution of  $s$  and  $t$  as filtering the graph signal  $s$  by a filter whose impulse response is  $t$ . In the *spectral* domain, equation (6.16) shows that convolution of the two signals is in the spectral domain the pointwise multiplication of the GFTs of the signals. This replicates the graph Fourier filtering theorem (see equation (27) in [1]).

### 6.2.1 Convolution of Graph Signals with the FFT

Equation (6.16) shows that, as in DSP, we can compute convolution by finding the two GFTs  $\hat{s}$  and  $\hat{t}$  of the two signals, then pointwise multiplying these, and finally taking the inverse GFT of the pointwise product. Even though this replicates the DSP result, GFTs and inverse GFTs are matrix vector products that are order  $N^2$ , not fast operations. We show that in “companion space,” i.e., working with the impulsive representations, the GzT of graph signals or their polynomial representation, and the polynomial transform filters of the signals, graph vertex convolution can be obtained by FFT.

*Fast convolution.* Consider the  $z$ -transform representations  $s(x)$ ,  $t(x)$ , and  $u(x)$  of  $s$ ,  $t$ , and  $u = s \circledast t$ .

**Result 6.7** (GSP convolution and linear convolution). *We have*

$$u(x) = (s(x) \cdot t(x)) \bmod \Delta_A(x) \quad (6.17)$$

where  $\Delta_A(x)$  is the characteristic polynomial of  $A$ , and  $u = s \otimes t$  is the vector of coefficients of  $u(x)$ .

*Proof.* The product of polynomials (in  $x$  or in  $A$ ) is the polynomial whose coefficients are the linear convolution of the sequences of coefficients of the polynomials. Powers larger than  $N - 1$  are reduced by Cayley-Hamilton achieved by  $\text{mod } \Delta_A(x)$  reduction. ■

**Remark 6.5.** In DSP,  $A^n = A^{n \bmod N}$ . This is the wrap-around effect, or “time-aliasing” in DSP. In DSP, the coefficient of the power of  $N$  is added to the coefficient of the power of  $n \bmod N = 0$ , the coefficient of  $N + 1$  is added to the power of  $n \bmod N = 1$ , and so on. For a generic graph in GSP, there is also “vertex-aliasing,” but it is not one-to-one like in DSP. The coefficient of the power of  $N$  (and higher powers) is scaled differently and added to lower powers from 0 to  $N - 1$  as per Cayley-Hamilton.

Result 6.7 and equation (6.17) are a fast convolution of the two graph signals when given their  $z$ -transforms. The linear convolution of the sequences of coefficients of  $P_s(A)$  and  $P_t(A)$  is computed by fast Fourier transform (FFT). The mod reduction is computed by (fast) polynomial division,  $O(N)$  operations [77]. Result 6.7 is very pleasing. It evaluates GSP LSI convolution using the FFT, an intrinsically DSP algorithm.

**Remark 6.6.** An interesting question is when are linear and circular (vertex) convolution equivalent in GSP. From (6.17), we see that, in GSP, if the degree of the product polynomial  $s(x)t(x)$  is not greater than  $N - 1$ , then  $u(x) = (s(x) \cdot t(x)) \bmod \Delta_A(x) = s(x) \cdot t(x)$ . Linear and circular convolution are equivalent and the reduction by  $\text{mod } \Delta_A(x)$  produces no effect. This is the same condition for when linear and circular convolution are equivalent in DSP. In practice, one may want to pad with zeros either or both of  $s(x)$  and  $t(x)$  to get faster processing.

An example of the convolution of graph signals using both polynomial filtering (figure 6.1) and the FFT for an expanded version (100 nodes) of the directed ladder graph in figure 5.2 is shown in figure 6.2. The steps in the figure illustrate several signal representations and transforms. On top, we compute the convolution by equation (6.14) in definition 6.4. From

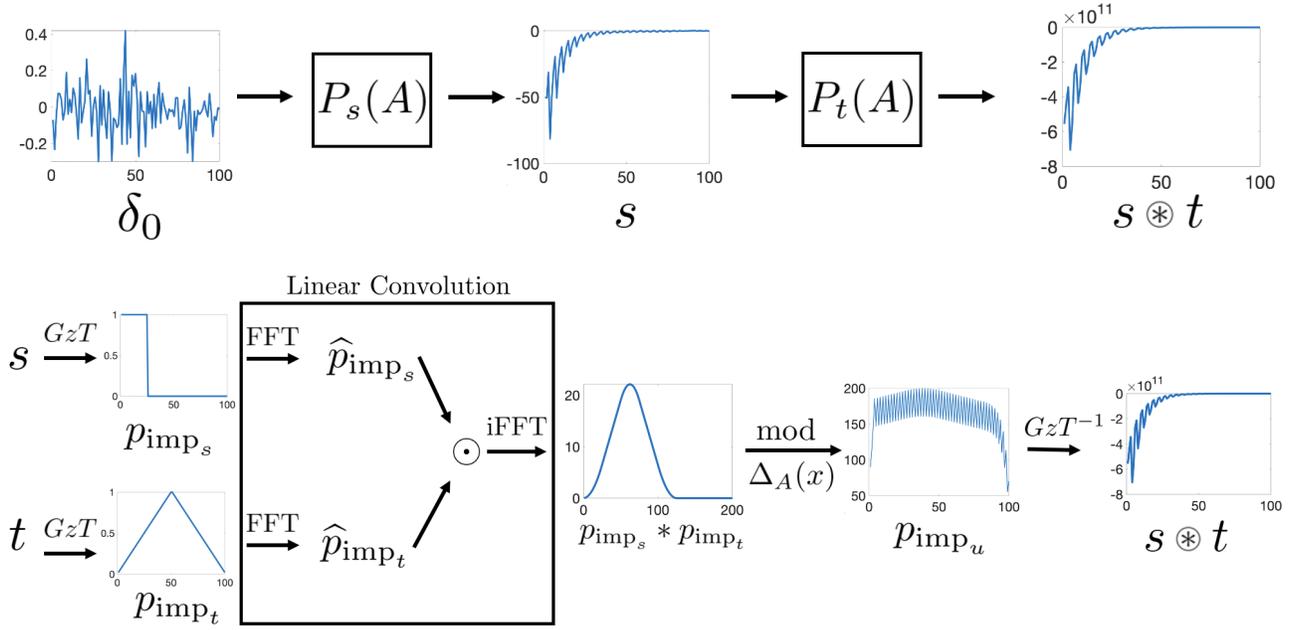


Figure 6.2: Example of circular convolution of  $s$  and  $t$  using both polynomial filtering (above) and the FFT (below) for the 100 node directed ladder graph in figure 5.2. Both methods produce the same result.

left to right, we start with the vertex impulse  $\delta_0$  (obtained by  $\text{GFT}^{-1}$  of a flat impulse in the spectral domain), going through the polynomial transform filters  $P_s(A)$  and  $P_t(A)$  to get  $t \circledast s$ . At the bottom, we illustrate the fast convolution in equation (6.17). We compute the  $\text{GzT}$  of  $s$  and  $t$  to obtain  $p_{\text{imp}_s}$  and  $p_{\text{imp}_t}$ . These are linearly convolved and reduced by  $\text{mod } \Delta_A(x)$  to obtain  $p_{\text{imp}_u}$ . A final  $\text{GzT}^{-1}$  gets the circular convolution  $s \circledast t$ .

The polynomial coefficient vectors for signals  $s$  and  $t$  are: 1) the first 25 entries of  $p_{\text{imp}_s}$  are 1 and the remaining 75 entries are 0; and 2) the entries of  $p_{\text{imp}_t}$  were chosen as a triangle signal, with the first half going from  $\frac{1}{50}$  to 1 with a step of  $\frac{1}{50}$  and then the remaining to back down to  $\frac{1}{50}$  with a step of  $-\frac{1}{50}$ . Comparing the two plots on the right of figure 6.2, we conclude that different methods lead to the same result for  $s \circledast t$ , with a maximum pointwise magnitude difference (due to roundoff errors) between them of 0.15 (with convolution result values of order  $10^{11}$ ).

## 6.3 Conclusion

In this chapter, we present the graph  $z$ -transform (GzT) and the spectral graph  $z$ -transform. Using the graph  $z$ -transform, we provide a symbolic polynomial representation for graph signals. We show when plugging in the graph frequencies  $\lambda$  into the graph  $z$ -transform polynomial, similar to the  $z$ -transform in DSP, we obtain a scaled version of the GFT of  $s$ ,  $\sqrt{N}\hat{s}$ . We also show the graph  $z$ -transform is permutation invariant. Using the graph  $z$ -transform, we provide an algorithm for fast graph convolution with the FFT.



# Chapter 7

## GSP Uncertainty Principle

In this chapter, we present a GSP Uncertainty Principle based on the interpolation of the graph  $z$ -transform  $p$  and spectral signal  $\hat{s}$  (similarly, spectral graph  $z$ -transform  $q$  and vertex signal  $s$ ). This GSP Uncertainty Principle directly relates the bandlimitness of  $p$  and  $\hat{s}$ , unlike current GSP uncertainty principles which relate the spread of  $s$  and  $\hat{s}$ . We use this uncertainty principle to show that when  $\hat{s}$  is bandlimited, we can reduce and simplify the Vandermonde system required to calculate  $p$ .

### 7.1 Brief Literature Review

We briefly review literature on the uncertainty principle in GSP. The literature on the uncertainty principle in GSP [78, 79] draws inspiration from the time spread and frequency spread of a continuous time signal  $x(t)$  and its Fourier transform  $X(f)$  to define a notion of spread for graph signals. In continuous time, the time spread is

$$\Delta_t^2 = \frac{\int_{-\infty}^{\infty} (t - t_0)^2 |x(t)|^2 dt}{\int_{-\infty}^{\infty} |x(t)|^2 dt}, t_0 = \frac{\int_{-\infty}^{\infty} t |x(t)|^2 dt}{\int_{-\infty}^{\infty} |x(t)|^2 dt} \quad (7.1)$$

The frequency spread is

$$\Delta_f^2 = \frac{\int_{-\infty}^{\infty} (f - f_0)^2 |X(f)|^2 df}{\int_{-\infty}^{\infty} |X(f)|^2 df}, f_0 = \frac{\int_{-\infty}^{\infty} f |X(f)|^2 df}{\int_{-\infty}^{\infty} |X(f)|^2 df} \quad (7.2)$$

Using these two definitions of spread yields the famous uncertainty principle,

$$\Delta_t^2 \Delta_f^2 \geq \frac{1}{(4\pi)^2} \quad (7.3)$$

From there, [78] defines graph spread and spectral spread (similar to time and frequency spread above). In particular, they define graph spread about a vertex  $u_0$  and signal  $x$  as

$$\Delta_{g,u_0}^2(x) = \frac{1}{\|x\|^2} x^T P_{u_0}^2 x \quad (7.4)$$

where  $P_{u_0} = \text{diag}\{d(u_0, v_1), d(u_0, v_2), \dots, d(u_0, v_N)\}$ , the distance between  $u_0$  and the nodes  $v_i$  of the graph and  $d$  is a distance matrix, e.g., geodesic distance. They define spectral spread as

$$\Delta_s^2(x) = \frac{1}{\|x\|^2} \sum_{n=1}^N \lambda_n |\hat{x}_n|^2 \quad (7.5)$$

Using these two quantities, they discuss the tradeoff between  $\Delta_{g,u_0}^2(x)$  and  $\Delta_s^2(x)$ .

Reference [79] uses a new definition of vertex and spectral spread to avoid potential shortcomings of choosing a distance metric  $d$  to define spread like [78] in (7.4). Instead of using a distance metric, [79] define vertex spread ( $\alpha^2$ ) and spectral spread ( $\beta^2$ ) for a signal  $x$  as

$$\alpha^2 = \frac{\|Dx\|_2^2}{\|x\|_2^2}, \beta^2 = \frac{\|Bx\|_2^2}{\|x\|_2^2}, \quad (7.6)$$

where  $Dx$  samples in the vertex domain and  $Bx$  is the vertex domain filter corresponding to sampling in the spectral domain. Using these quantities, [79] explores an uncertainty principle and feasibility region for all possible  $\alpha$  and  $\beta$ .

In both [78, 79], uncertainty principles and feasibility regions are explored in the context of vertex and spectral spread. These works do not directly relate the bandlimit of signals like  $s$ ,  $\hat{s}$ ,  $p$ ,  $q$ , but instead relate the energy and spread of signals. In [79], they give a sufficient, but not necessary condition for the existence of a perfectly localized signal  $s$  (i.e.,  $\|s\|_0 = |\mathcal{S}|$  for a

vertex subset  $\mathcal{S}$ ,  $\|\widehat{s}\|_0 = |\mathcal{F}|$  for a frequency subset  $\mathcal{F}$  where  $\|\cdot\|_0$  is the  $\ell_0$  pseudo-norm) in both the vertex and spectral domains:

$$|\mathcal{S}| + |\mathcal{F}| > N \tag{7.7}$$

where  $N$  is the number of nodes in the graph.

Reference [53] establishes the DSP inequality

$$N_t \cdot N_w \geq N \tag{7.8}$$

where  $N_t$  is the number of non-zero entries in the time signal  $s$ ,  $N_w$  is the number of non-zero entries in the corresponding frequency signal  $\widehat{s}$  and  $N$  is the length of the signal. The argument presented in [53] directly relies on the DFT being a Vandermonde matrix. Since the GFT is not a Vandermonde matrix, it is difficult to use a similar argument to relate  $s$  and  $\widehat{s}$ . However,  $\widehat{s}$  and  $p$  (as well as  $s$  and  $q$ ) are related by a Vandermonde matrix. This enables us to develop a GSP Uncertainty Principle (relating these quantities), drawing inspiration from [53].

In this chapter, we assume the eigenvalues of  $A$  are unique (assumption 2.2).

## 7.2 Uncertainty Principle for Spectral and Vertex Impulsive Signal Representations

In this section, we develop an uncertainty principle relating the bandlimits of the spectral signal,  $\widehat{s}$  and vertex impulsive signal,  $p$ . We begin by defining bandlimits for each signal, then discuss constraints of each signal if the other is bandlimited. Lastly, we present the uncertainty principle.

## 7.2.1 Definitions

A common consideration in signal processing are signals that are bandlimited in the spectral (frequency) domain.

**Definition 7.1.** (*Spectral Bandlimitedness*) A signal  $\hat{s}$  is  $N_{\hat{s}}$ -**bandlimited** in the spectral domain if  $\hat{s}$  contains exactly  $N - N_{\hat{s}}$  zeros.

Without loss of generality, we assume the  $N - N_{\hat{s}}$  zeros in  $\hat{s}$  are the last  $N - N_{\hat{s}}$  entries.<sup>1</sup>

We also assume that  $\hat{s}$  is not the zero vector, i.e., all zeros.

We will focus on  $\hat{s}$  that are  $(N_{\hat{s}})$ -bandlimited:

$$\hat{s} = \begin{bmatrix} \hat{s}_{N_{\hat{s}}} \\ 0_{N-N_{\hat{s}}} \end{bmatrix} \quad (7.9)$$

We now define bandlimited  $p$ .

**Definition 7.2.** (*Polynomial Bandlimitedness of  $p$* ) A signal  $p$  is  $N_p$ -**bandlimited** if  $p$  contains exactly  $N - N_p$  zeros **that are the bottom entries of  $p$** .

Definition 7.2 is slightly different than definition 7.1. A  $N_{\hat{s}}$ -bandlimited  $\hat{s}$  can have the zeros anywhere in  $\hat{s}$  and must have exactly  $N - N_{\hat{s}}$  zeros. On the other hand, a  $N_p$ -bandlimited  $p$  must have exactly  $N - N_p$  zeros at the end of the signal  $p$  with a nonzero value at the  $N_p$  entry of  $p$ , but may have zeros in the first  $N_p - 1$  entries of  $p$ .

**Remark 7.1.** *The reason for this distinction is that the ordering of  $p$  has mathematical and graph significance. Given a signal  $s = p(A)\delta_0$ ,  $p$  is the polynomial coefficients of  $p(A)$  with entry,  $p_i$ , being the coefficient of  $A^i$ . Definition 7.2 is equivalent to  $p(A)$  is a polynomial of degree  $N_p - 1$ . It only states  $p_{N_p} \neq 0$  and  $p_i = 0$  for  $i > N_p$ . It does not say anything about the first  $N_p - 1$  indices of  $p$ . Limiting the degree of the polynomial to  $N_p - 1$  also means that the*

---

<sup>1</sup>This is a notational convenience. This is not a necessary condition. The eigenvalues and eigenvectors of  $A$  can be permuted so that this is true.

filter only considers powers of the adjacency matrix up to  $N_p - 1$ , representing up to  $N_p - 1$ -hop neighbors of nodes in the graph. This is a desirable property because we want to consider local neighborhoods of nodes when filtering. If  $N_p$  is close to  $N$ , then the filter considers the non-local nodes (close to  $N - 1$  hop neighbors) in  $A$  for every node.

## 7.2.2 Tradeoff between Bandlimited Signals in Spectral and Vertex Impulsive Representations

We begin by assuming  $\hat{s}$  is  $N_{\hat{s}}$ -bandlimited as given in (7.9). From (5.28),

$$\hat{s} = \mathcal{V}p \tag{7.10}$$

Since  $\mathcal{V}$  is a Vandermonde matrix of the eigenvalues  $\lambda$ , (7.10) is equivalent to interpolating  $N$  points<sup>2</sup>,  $(\lambda_i, \hat{s}_i)$ , using a  $N - 1$  degree polynomial with (ordered) polynomial coefficients,  $p$ . Let  $p(x)$  be the interpolating polynomial.

**Result 7.1.** *If  $\hat{s}$  is  $N_{\hat{s}}$ -bandlimited,  $p(x)$  has at least degree  $N - N_{\hat{s}}$ .*

*Proof.* Since  $\hat{s}$  is  $N_{\hat{s}}$ -bandlimited, it contains  $N - N_{\hat{s}}$  zeros by definition 7.1. From (7.10), we are interpolating with  $N$  points that include the  $N - N_{\hat{s}}$  points,  $(\lambda_i, 0)$  for  $i = N_{\hat{s}}, \dots, N - 1$ . These  $\lambda_i$  are  $N - N_{\hat{s}}$  roots of polynomial  $p(x)$ . Since there are  $N_{\hat{s}}$  other points ( $i = 0, \dots, N_{\hat{s}} - 1$ ), we only know that there are at least  $N - N_{\hat{s}}$  roots. Since  $p(x)$  has at least  $N - N_{\hat{s}}$  roots,  $p(x)$  has at least degree  $N - N_{\hat{s}}$ . ■

**Corollary 7.1.** *If  $\hat{s}$  is  $N_{\hat{s}}$ -bandlimited,  $p$  is at least  $(N - N_{\hat{s}} + 1)$ -bandlimited.*

*Proof.* The proof follows from the previous result. If  $p(x)$  has degree  $N - N_{\hat{s}}$ , then  $p_{N - N_{\hat{s}} + 1} \neq 0$  and  $p_K = 0$  for  $K > N - N_{\hat{s}} + 1$ . So,  $p$  is  $(N - N_{\hat{s}} + 1)$ -bandlimited and has  $N_{\hat{s}} - 1$  trailing zeros. Since  $p(x)$  has at least degree  $N - N_{\hat{s}}$ , by result 7.1,  $p$  is at least  $(N - N_{\hat{s}} + 1)$ -bandlimited, i.e.,  $p$  can only contain at most  $N_{\hat{s}} - 1$  trailing zeros. ■

---

<sup>2</sup>By assumption 2.2, the points are unique because the eigenvalues are unique.

Corollary 7.1 suggests a tradeoff between the bandlimitedness of  $\hat{s}$  and  $p$ . As  $N_{\hat{s}}$  increases, the number of zeros in  $\hat{s}$  decreases. This increases the band of  $\hat{s}$ , but decreases the minimum degree of  $p(x)$  (potentially, less roots). This decreases the lower bound on the bandlimitedness of  $p$ .

Similarly, if  $N_{\hat{s}}$  decreases, the number of zeros in  $\hat{s}$  increases. This decreases the band of  $\hat{s}$ , but increases the minimum degree of  $p(x)$  (guaranteed more roots). This increases the lower bound on the bandlimitedness of  $p$ .

An interesting question is if  $p$  is bandlimited what does that say about  $\hat{s}$ ? The next result addresses this.

**Result 7.2.** *If  $p$  is  $N_p$ -bandlimited, then  $\hat{s}$  has at most  $N - N_p$  zero entries, and is at least  $(N - N_p + 1)$ -bandlimited.*

*Proof.* Since  $p$  is  $N_p$ -bandlimited,  $p(x)$  is a  $N_p - 1$  degree polynomial with exactly  $N_p - 1$  zeros. The only way to get zeros in  $\hat{s}$  is if the eigenvalues,  $\lambda$ , of  $A$  contain the roots of  $p(x)$ . Since there are exactly  $N_p - 1$  roots and no repeated eigenvalues (assumption 2.2), there are maximally  $N_p - 1$  roots in the interpolation points. If the eigenvalues of  $A$  and the roots of  $p(x)$  do not coincide, then there will be no zeros in  $\hat{s}$ . Thus, there are at most  $N_p - 1$  zeros and using Definition 7.1,  $\hat{s}$  is at least  $(N - N_p + 1)$ -bandlimited. ■

Corollary 7.1 and Result 7.2 yield the Uncertainty Principle.

**Result 7.3.** *(GSP Uncertainty Principle:  $\hat{s}, p$ ) Let  $N_{\hat{s}}$  be the bandlimit of  $\hat{s}$ . Let  $N_p$  be the bandlimit of  $p$  in Definition 7.2. Then,*

$$N_{\hat{s}} + N_p \geq N + 1 \tag{7.11}$$

A visual representation of the feasibility region Uncertainty Principle for  $N = 10$  is in Fig. 7.1.

An example of when the above bound is tight is when  $\hat{s} = \hat{\delta}_0 = \frac{1}{\sqrt{N}}\mathbf{1}$  as defined in (3.46). In this case,  $p = \mathcal{V}^{-1}\hat{s} = \frac{1}{\sqrt{N}}e_0 = [\frac{1}{\sqrt{N}}, 0, \dots, 0]^T$ . This is because the first column of  $\mathcal{V}$  is 1,

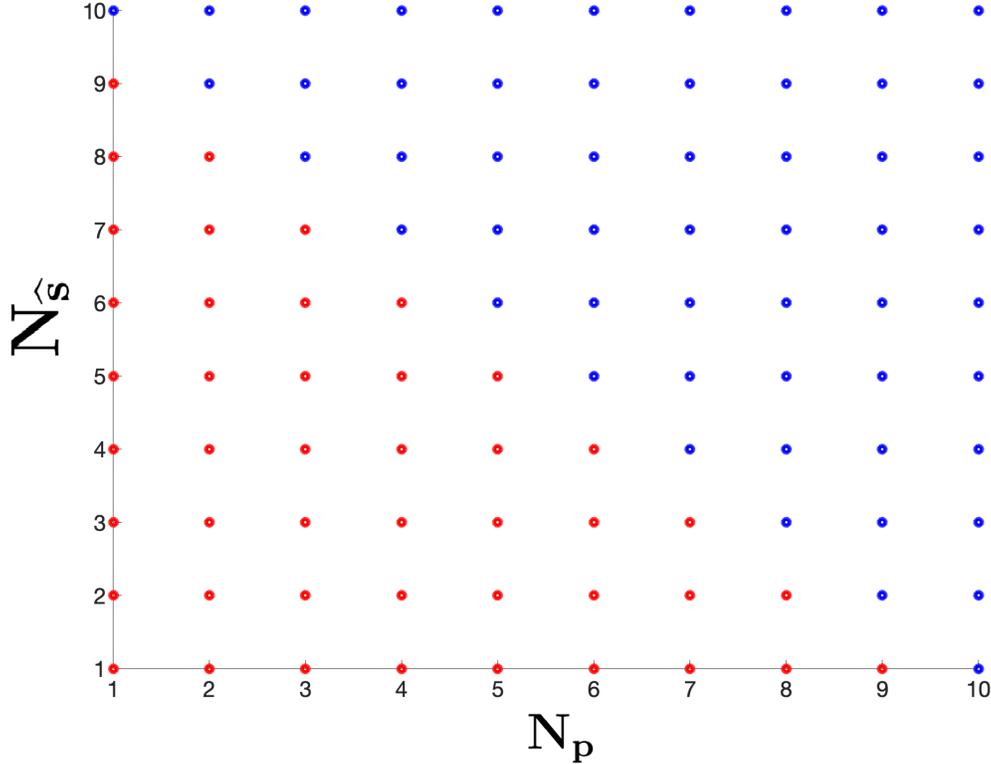


Figure 7.1: The feasibility region for the GSP uncertainty principle in result 7.3 for  $N = 10$ . Red dots represent infeasible regions and blue dots represent feasible regions.

the vector of all 1s. The bandlimit of  $\hat{s}$  is  $N_{\hat{s}} = N$ . The bandlimit of the corresponding  $p$  is 1. So, in this case,  $N_{\hat{s}} + N_p = N + 1$ .

The above results and corollary hold in GSP. We now briefly consider the Uncertainty Principle in DSP. In DSP, we have  $p = s$ . This means the interpolating polynomial  $p(x)$  coefficients is the same as the time signal  $s$ . We can rewrite result 7.3 using this fact.

**Result 7.4.** (*DSP Uncertainty Principle:  $s, \hat{s}$* ) Let  $N_{\hat{s}}$  be the bandlimit of  $\hat{s}$  ( $L = N_{\hat{s}}$  in Definition 7.1). Let  $N_s$  be the bandlimit of  $s$  ( $L = N_s$  in Definition 7.2). Then,

$$N_s + N_{\hat{s}} \geq N + 1 \tag{7.12}$$

These results in DSP suggest the classical relationship / tradeoff between time and frequency in DSP. Namely, that signals that are “wide” in one domain are typically “narrow” in the other. For example, take  $N = 4, N_{\hat{s}} = 3$  in DSP. The above result says if  $N_{\hat{s}} = 3$  then

$s$	$\hat{s}$
$[-j, 1, 0, 0]^T$	$[1 - j, -2j, -1 - j, 0]^T$
$[-2j, 2 - j, 1, 0]^T$	$[3 - 3j, -2 - 4j, -1 - j, 0]^T$
$[1.5, .5j, .5, -.5j]^T$	$[1, 1, 1, 0]^T$

Table 7.1:  $s$  and corresponding  $\hat{s}$ . For  $\hat{s}$  with exactly 1 zero,  $s$  can have 0, 1, 2 trailing zeros.

$N_s \geq 2$ , so  $s$  has at most 2 trailing zeros. Also, if  $s$  has 3 trailing zeros ( $N_s = 1$ ), then  $\hat{s}$  has 4 nonzero values ( $N_{\hat{s}} = 4$ ).

## 7.3 Bandlimited Signal $\hat{s}$ : Reducing the Vandermonde System

In this section, we consider a bandlimited  $\hat{s}$  and show how this can reduce the linear system in (7.10) when converting from  $\hat{s}$  to  $p$  using (7.10). Let  $\hat{s}$  be a  $N_{\hat{s}}$ -bandlimited signal given in (7.9). Let  $K = N - N_{\hat{s}}$  be the number of zeros.

From (7.10), we are interpolating  $N$  points,  $(\lambda_i, \sqrt{N}\hat{s}_i)$ , using a  $N - 1$  degree polynomial with (ordered) polynomial coefficients,  $p$ .<sup>3</sup> Let  $p(x)$  be the interpolating polynomial.

Since  $\hat{s}$  contains  $K$  zeros, we are interpolating with  $N$  points that include  $K$  roots. Thus, we can break the interpolating polynomial  $p(x)$  into two parts:

$$p(x) = r(x)e(x) \tag{7.13}$$

where  $r(x)$  is the  $K$  degree polynomial with the  $K$  roots at  $\lambda_i$  for  $i = N - K, \dots, N - 1$  corresponding to the points  $(\lambda_i, \sqrt{N}\hat{s}_i = 0)$  and leading coefficient 1 and  $e(x)$  is a polynomial of at most degree  $N - K - 1$ .

---

<sup>3</sup>An alternative interpolation is to interpolate the points  $(\lambda_i, \hat{s}_i)$ . Then, scale the interpolating polynomial by  $\sqrt{N}$ . This will produce the same  $p$ .

Since  $r(x)$  has  $K$  roots at  $\lambda_i$ , we can determine  $r(x)$ .

$$r(x) = \prod_{i=N-K}^{N-1} (x - \lambda_i) \quad (7.14)$$

An alternative way is to take the characteristic polynomial  $\Delta_A(x)$  of  $A$  and divide by the polynomial with leading coefficient 1 and roots  $\lambda_i$ ,  $i = 0, 1, \dots, N - K - 1$ .

$$r(x) = \frac{\Delta_A(x)}{\prod_{i=0}^{N-K-1} (x - \lambda_i)} \quad (7.15)$$

This is because  $\Delta_A(x)$  has  $N$  roots,  $\lambda_i$ . By dividing by the polynomial in (7.15), we obtain a polynomial with the  $K$  roots.

From (7.13), (7.14), we obtain:

$$p(x) = r(x)e(x) = e(x) \prod_{i=N-K}^{N-1} (x - \lambda_i) \quad (7.16)$$

To find  $e(x)$ , we must use the remaining  $N - K$  points,  $(\lambda_i, \sqrt{N}\widehat{s}_i)$  for  $i = 0, \dots, N - K - 1$ . Since we know  $r(x)$ , we can divide each  $\sqrt{N}\widehat{s}_i$  by  $r(\lambda_i)$ <sup>4</sup>, producing  $N - K$  points,  $(\lambda_i, \frac{\sqrt{N}\widehat{s}_i}{r(\lambda_i)})$ ,  $i = 0, \dots, N - K - 1$ . We know that  $e(x)$  must go through these points, so we can interpolate these points to obtain  $e(x)$ .

Interpolation of  $e(x)$  can be done using a Vandermonde matrix. Let  $\lambda_s$  be a vector of the first  $\lambda$  values,  $\lambda_s = [\lambda_0, \lambda_1, \dots, \lambda_{N-K-1}]^T$ . Let  $e$  be the coefficient vector of  $e(x)$  with  $e_0$  being the coefficient of the constant term,  $e_1$  being the coefficient of the  $x$  term, etc. Let  $V_s$  be the  $N - K \times N - K$  Vandermonde matrix of  $\lambda_s$ ,  $V_s = [\lambda_s^0, \lambda_s^1, \dots, \lambda_s^{N-K-1}]$ . Let  $d$  be the vector of  $\frac{\sqrt{N}\widehat{s}_i}{r(\lambda_i)}$ ,  $d = \left[ \frac{\sqrt{N}\widehat{s}_0}{r(\lambda_0)}, \frac{\sqrt{N}\widehat{s}_1}{r(\lambda_1)}, \dots, \frac{\sqrt{N}\widehat{s}_{N-K-1}}{r(\lambda_{N-K-1})} \right]^T$ . Then,

$$d = V_s e \quad (7.17)$$

and  $e = V_s^{-1}d$ . Using the coefficients  $e$ , we can determine  $e(x)$ .

---

<sup>4</sup>Note that since the roots of  $r(x)$  are  $\lambda_i$ ,  $i = N - K, \dots, N - 1$  by (7.14),  $r(\lambda_i) \neq 0$  for  $i = 0, \dots, N - K - 1$ .

**Result 7.5.**  $e(x)$  has degree 0 to  $N - K - 1$  and  $p(x)$  has degree  $K$  to  $N - 1$ .

*Proof.* We know  $e(x)$  must go through the  $N - K$  points,  $\left(\lambda_i, \frac{\sqrt{N}\widehat{s}_i}{r(\lambda_i)}\right)$ ,  $i = 0, \dots, N - K - 1$ . Observe that  $\frac{\sqrt{N}\widehat{s}_i}{r(\lambda_i)} \neq 0$  because otherwise,  $\widehat{s}_i = 0$ , making  $\lambda_i$  a root of  $r(x)$ . So,  $\frac{\sqrt{N}\widehat{s}_i}{r(\lambda_i)}$  are not roots of  $e(x)$ . In general, interpolating  $N - K$  points of  $e(x)$  yields a polynomial of degree  $N - K - 1$ . However, the points could also be interpolated of a polynomial of lesser degree, e.g., a constant polynomial would fit the points if all  $\frac{\sqrt{N}\widehat{s}_i}{r(\lambda_i)}$  were equal. Thus,  $e(x)$  has degree 0 to  $N - K - 1$ . Since  $r(x)$  is degree  $K$  and  $p(x) = r(x)e(x)$ ,  $p(x)$  has degree  $K$  to  $N - 1$ . ■

**Remark 7.2.** From result 7.5, we see that  $e(x)$  controls how many trailing zeros  $p$  has. If  $e(x)$  is constant, then there are  $N - K - 1$  trailing zeros in  $p$ . If  $e(x)$  has degree  $N - K - 1$ , then there are no trailing zeros in  $p$ . So,  $r(x)$  establishes the minimum degree of  $p(x)$  and maximum number of trailing zeros in  $p$ . While  $e(x)$  determines the exact number of trailing zeros. Result 7.5 supports result 7.1.

**Remark 7.3.** (Simpler Way to find  $p$  using (7.10) for  $\widehat{s}$  bandlimited signals)

Normally, to find  $p$ , we would have to solve (7.10). This involves solving a linear system with  $N$  unknowns. However, if  $\widehat{s}$  is bandlimited, then we can simplify the calculation. We can find  $r(x)$  from (7.14) using the  $K$  roots. Then, interpolate the  $N - K$  points  $(\lambda_i, \frac{\sqrt{N}\widehat{s}_i}{r(\lambda_i)})$ ,  $i = 0, \dots, N - K - 1$  to find  $e(x)$ . This can be done by solving a system with a smaller  $N - K \times N - K$  matrix (see (7.17)). From there,  $p(x) = r(x)e(x)$ .

We provide the algorithm block below.

---

**Algorithm 1:** Find  $p$  using (7.10) for  $(N - K)$ -bandlimited  $\widehat{s}$

---

- 1 **Given:** Eigenvalue vector  $\lambda$ ,  $(N - K)$ -bandlimited  $\widehat{s}$  from (7.9).
  - 2 Produce  $N$  points from  $\lambda$  and  $\widehat{s}$ :  $(\lambda_i, \sqrt{N}\widehat{s}_i)$  ( $i = 0, \dots, N - K - 1$ ,  $N - K$  non-roots),  $(\lambda_i, \widehat{s}_i = 0)$  ( $i = N - K, \dots, N - 1$ ,  $K$  roots)
  - 3 Using the  $K$  roots, find  $r(x)$  using (7.14) or (7.15). From  $N - K$  non-roots, form  $N - K$  points,  $\left(\lambda_i, \frac{\sqrt{N}\widehat{s}_i}{r(\lambda_i)}\right)$ ,  $i = 0, \dots, N - K - 1$ .
  - 4 Using (7.17), find  $e$ . Using  $e$ , find  $e(x)$ .
  - 5 Multiply  $r(x)$  and  $e(x)$  to find  $p(x)$ .
  - 6 Take the coefficients from  $p(x)$  to find  $p$ .
-

This algorithm shows that we can reduce the size of the linear system from  $N \times N$  to  $(N - K) \times (N - K)$  if  $\hat{s}$  is  $(N - K)$ -bandlimited. For  $K$  large and close to  $N$  ( $\hat{s}$  contains  $K$  zeros), this results in solving a significantly smaller linear system. Note that  $r(x)$  can be calculated using linear convolution and FFTs.  $p(x)$  can be calculated from  $r(x)$  and  $e(x)$  using linear convolution and FFTs.

### 7.3.1 Producing a Bandlimited Signal $\hat{s}$ with Zeros in Specific Locations

So far, we have discussed how to convert from a bandlimited  $\hat{s}$  to  $p$ . In this section, we discuss an algorithm on how to produce a bandlimited signal  $\hat{s}$  with zeros in specific entries. Without loss of generality, assume we want zeros in the last  $K$  entries of  $\hat{s}$ . Thus, the polynomial  $p(x)$  has roots at  $\lambda_i$ ,  $i = N - K, \dots, N - 1$ . Using these roots and (7.14), we can find  $r(x)$ .  $r(x)$  guarantees the zeros in the specified locations.

Now, we can choose an arbitrary  $e(x)$  polynomial of maximum degree  $N - 1 - K$ . Some design considerations are if we want additional zeros in  $\hat{s}$  at other locations besides the last  $K$  entries and what we want the bandlimit,  $N_p$  to be. If we want additional zeros in  $\hat{s}$  at other locations, we must choose  $e(x)$  so it has roots at the  $\lambda_i$  corresponding to those locations. If we want to fix  $N_p$ , we must choose  $e(x)$  to be degree  $N_p - 1 - K$ , following the Uncertainty Principle in result 7.3.

Once we choose  $e(x)$ , we can use (7.13) to find  $p(x)$ , whose coefficients gives  $p$ . From there, we use (7.10) to find the  $\hat{s}$  with the zeros in specific locations.

### 7.3.2 DSP Example

We continue the example in Table 7.1 where  $N = 4$ ,  $K = 2$  in DSP. We provide  $r(x)$ ,  $e(x)$  and  $p(x)$  as well as the  $e(x)$  interpolating points in Table 7.2. Note that  $\lambda = [1, -j, -1, j]^T$  in DSP.

For all three cases,  $r(x) = x - j$ , a root at  $x = j$ . However, depending on  $\hat{s}$ , we obtain

$s$	$\hat{s}$	$e(x)$	$p(x)$
$[-2j, 2, 0, 0]^T$	$[1 - j, -2j, -1 - j, 0]^T$	1	$2x - 2j$
$[-4j, 4 - 2j, 2, 0]^T$	$[3 - 3j, -2 - 4j, -1 - j, 0]^T$	$x + 2$	$2x^2 + (4 - 2j)x - 4j$
$[3, j, 1, -j]^T$	$[2, 2, 2, 0]^T$	$-.5jx^2 + x + 1.5j$	$-jx^3 + x^2 + jx + 3$

Table 7.2:  $s$  and corresponding  $\hat{s}$ ,  $r(x), e(x)$ ,  $e(x)$  interpolation points,  $p(x)$ . For  $\hat{s}$  with exactly 1 zero,  $s$  can have 0, 1, 2 trailing zeros. In each case,  $r(x) = x - j$ .

different  $e(x)$  with different degrees. The degree of the  $p(x)$  is 1, 2, 3, corresponding to the degree of  $e(x)$  being 0, 1, 2 respectively.

### 7.3.3 GSP Example

We show an example of the specific calculations applying Algorithm 1 to a 10-node Erdős-Rényi graph.

Consider the 10-node directed Erdős-Rényi graph (formed using probability 0.25), shown in figure 7.2, with adjacency matrix:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \quad (7.18)$$

Its eigenvalues are  $\lambda = [2.39, -.61 + 1.7j, -.61 - 1.7j, .38 + .64j, .38 - .64j, -.54 + .2j, -.54 - .2j, -.42 + .72j, -.42 - .72j, 0]^T$ . A plot of the eigenvalues and the unit circle are shown in figure 7.3.

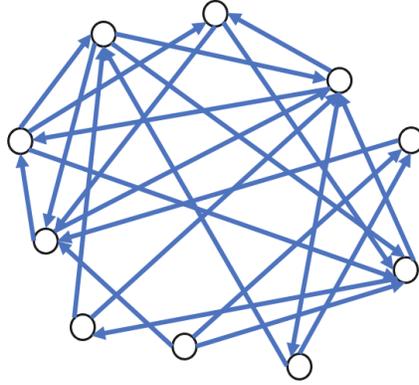


Figure 7.2: Erdős-Rényi graph for  $N = 10$  with probability 0.25

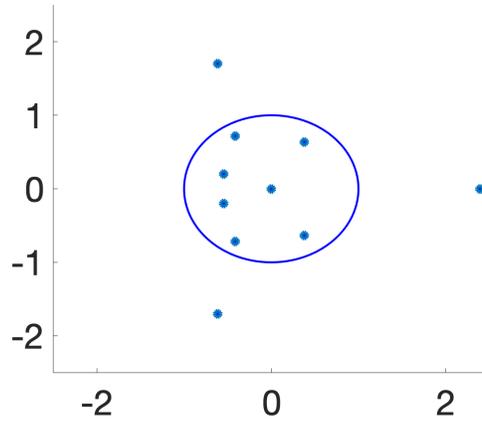


Figure 7.3: The eigenvalues of the 10-node Erdős-Rényi graph, plotted with the unit circle (in blue).

Consider  $\hat{s} = \frac{1}{\sqrt{10}}[0, 0, 0, 0, 0, 0, 0, 7.09 + 8.22j, 7.09 - 8.22j, 1.75]^T$ , a 3-bandlimited signal. We want to find  $p$  and  $s$  using Algorithm 1. The roots of  $r(x)$  are  $2.39, -.61 + 1.7j, -.61 - 1.7j, .38 + .64j, .38 - .64j, -.54 + .2j, -.54 - .2j$ , so  $r(x)$  is a polynomial of degree 7. Using (7.14),  $r(x) = x^7 - .83x^6 + .01x^5 - 7.43x^4 - 2.82x^3 - .55x^2 - 2.61x - 1.46$ .

Using (7.17) with  $\lambda_s = [-.42 + .72j, -.42 - .72j, 0]^T$  and  $r(\lambda_s) = [-.42 - 4.35j, -.42 + 4.35j, -1.46]^T$  yields:

$$\begin{bmatrix} 1 & -.42 + .72j & -.34 - .6j \\ 1 & -.42 - .72j & -.34 + .6j \\ 1 & 0 & 0 \end{bmatrix} e = \begin{bmatrix} \frac{7.1+8.22j}{-.42-4.35j} \\ \frac{7.09-8.22j}{-.42+4.35j} \\ \frac{1.75}{-1.46} \end{bmatrix} \quad (7.19)$$

So,  $e = [-1.2, 2, 0]^T$  and  $e(x) = 2x - 1.2$ .

Thus,  $p(x) = e(x)r(x) = 2x^8 - 2.86x^7 + 1.01x^6 - 14.87x^5 + 3.28x^4 + 2.29x^3 - 4.55x^2 + .21x + 1.75$  and  $p = [1.75, .21, -4.55, 2.29, 3.28, -14.87, 1.01, -2.86, 2, 0]^T$

We observe that  $Vp = \hat{s} = \frac{1}{\sqrt{10}}[0, 0, 0, 0, 0, 0, 7.09 + 8.22j, 7.09 - 8.22j, 1.75]^T$ . Also, we observe that the example obeys the Uncertainty Principle in Result 7.3 with  $N_{\hat{s}} = 3$ ,  $N_p = 9$  and  $N_{\hat{s}} + N_p = 12 > 11 = N + 1$ . Note that this result does not depend on the GFT and eigenvectors, only on the eigenvalues. Only when we calculate  $s$  does it depend on the GFT and eigenvectors: Using  $p$  as the coefficients of  $p(A)$  and calculating  $s = p(A)\delta_0$  yields  $s = [.98, -4.54, 1.98, 10.02, 2.95, .87, -7.16, 4.89, 1.93, -6.6]^T$  This is equal to  $s = \text{GFT}^{-1}\hat{s}$ .

$r(x)$  and  $p(x)$  are plotted in figure 7.4 with the two real coordinates at  $(2.4, 0)$ ,  $(0, 1.75)$ .  $r(x)$  and  $p(x)$  both go through the root at 2.4, but only  $p(x)$  goes through the point  $(0, 1.75)$  because it is not a root.

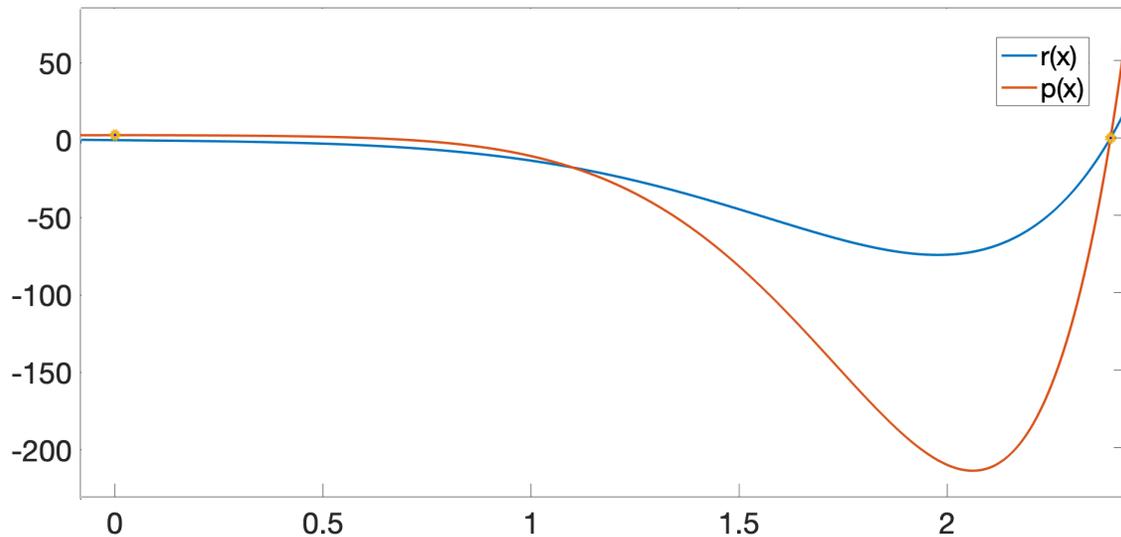


Figure 7.4:  $p(x), r(x)$  for the example, along with the two real coordinates at  $(2.4, 0)$ ,  $(0, 1.75)$ .  $p(x)$  interpolates and goes through both points, but  $r(x)$  only goes through  $(2.4, 0)$  because it is a root.

## 7.4 Uncertainty Principle for Standard and Spectral Impulsive Signal Representations

In this section, we dualize the results of the previous section to develop an uncertainty principle relating the bandlimits of the standard signal,  $s$ , and spectral impulsive signal,  $q$ . We proceed in a similar fashion as the last section by defining bandlimit for  $s$  and  $q$ , discussing constraints of each signal if the other is bandlimited. Then, we present the uncertainty principle.

### 7.4.1 Definitions

**Definition 7.3.** (*Finite Support of  $s$* ) A signal  $s$  is  $N_s$ -**supported** in the vertex domain if  $s$  contains exactly  $N - N_s$  zeros.

This definition is the dual of definition 7.3. Without loss of generality, we can assume the  $N - N_s$  zeros in  $s$  are the last  $N - N_s$  entries of  $s$ .<sup>5</sup> We also assume that  $s$  is not the zero vector, i.e., all zeros.

We will focus on  $s$  that are  $(N_s)$ -supported:

$$s = \begin{bmatrix} s_{N_s} \\ 0_{N-N_s} \end{bmatrix} \quad (7.20)$$

We now define bandlimited  $q$  in a similar way to definition 7.2.

**Definition 7.4.** (*Polynomial bandlimitedness of  $q$* ) A signal  $q$  is  $N_q$ -**bandlimited** if  $q$  contains exactly  $N - N_q$  zeros **at the bottom of  $q$** .

---

<sup>5</sup>Like with  $\widehat{s}$ , this is a notational convenience. This is not a necessary condition. The vertices of  $A$  can be reordered so that this is true.

## 7.4.2 Tradeoff between Reduced Supported Signals in Vertex and Bandlimited Spectral Impulsive Representations

This section parallels section 7.2.2 except it relates  $s$  and  $q$  instead of  $\hat{s}$  and  $p$ . We begin by assuming  $s$  is  $N_s$ -supported as given in (7.20). From (5.33),

$$s = \mathcal{V}^* q \tag{7.21}$$

Since  $\mathcal{V}^*$  is a Vandermonde matrix of the eigenvalues  $\lambda$ , (7.21) is equivalent to interpolating  $N$  points,  $(\lambda_i^*, s_i)$ , using a  $N - 1$  degree polynomial with (ordered) polynomial coefficients,  $q$ . Let  $q(x)$  be the interpolating polynomial. Note that the coefficients of  $q(x)$  are the entries of  $q$ .

**Result 7.6.** *If  $s$  is  $N_s$ -supported,  $q(x)$  has at least degree  $N - N_s$ .*

*Proof.* This is a similar argument to result 7.1. Since  $s$  is  $N_s$ -supported, it contains  $N - N_s$  zeros by definition 7.3. From (7.21), we are interpolating with  $N$  points that include the  $N - N_s$  points,  $(\lambda_i^*, 0)$  for  $i = N_s, \dots, N - 1$ . These  $\lambda_i^*$  are  $N - N_s$  roots of polynomial  $q(x)$ . Since there are  $N_s$  other points ( $i = 0, \dots, N_s - 1$ ), we only know that there are at least  $N - N_s$  roots. Since  $p(x)$  has at least  $N - N_s$  roots,  $p(x)$  has at least degree  $N - N_s$ . ■

**Corollary 7.2.** *If  $s$  is  $N_s$ -supported,  $q$  is at least  $(N - N_s + 1)$ -bandlimited.*

*Proof.* The proof follows from result 7.6. If  $q(x)$  has degree  $N - N_s$ , then  $q_{N-N_s+1} \neq 0$  and  $q_K = 0$  for  $K > N - N_s + 1$ . So,  $q$  is  $(N - N_s + 1)$ -bandlimited and has  $N_s - 1$  trailing zeros. Since  $q(x)$  has at least degree  $N - N_s$ , by result 7.6, then  $q$  is at least  $(N - N_s + 1)$ -bandlimited, i.e.,  $q$  can only contain at most  $N_s - 1$  trailing zeros. ■

Corollary 7.2 suggests a tradeoff between the reduced support of  $s$  and bandlimitedness of  $q$ . As  $N_s$  increases, the number of zeros in  $s$  decreases. This increases the support of  $s$ , but decreases the minimum degree of  $q(x)$  (potentially, less roots). This decreases the lower bound on the bandlimitedness of  $q$ .

Similarly, if  $N_s$  decreases, the number of zeros in  $s$  increases. This decreases the support of  $s$ , but increases the minimum degree of  $q(x)$  (guaranteed more roots). This increases the lower bound on the bandlimitedness of  $q$ .

An interesting question is if  $q$  is bandlimited, what does that say about  $s$ . The next result addresses this.

**Result 7.7.** *If  $q$  is  $N_q$ -bandlimited, then  $s$  has at most  $N - N_q$  zero entries and is at least  $(N - N_q + 1)$ -supported.*

*Proof.* Since  $q$  is  $N_q$ -bandlimited,  $q(x)$  is a  $N_q - 1$  degree polynomial with exactly  $N_q - 1$  zeros. The only way to get zeros in  $s$  is if the eigenvalues,  $\lambda^*$ , of  $A$  contain the roots of  $q(x)$ . Since there are exactly  $N_q - 1$  roots and no repeated eigenvalues, there are maximally  $N_q - 1$  roots in the interpolation points. If the eigenvalues of  $A$  and the roots of  $q(x)$  do not coincide, then there will be no zeros in  $\hat{s}$ . Thus, there are at most  $N_q - 1$  zeros and using Definition 7.3,  $s$  is at least  $(N - N_q + 1)$ -supported. ■

Corollary 7.2 and Result 7.7 yield the Uncertainty Principle.

**Result 7.8.** *(GSP Uncertainty Principle:  $s, q$ ) Let  $N_s$  be the support of  $s$  ( $L = N_s$  in Definition 7.3). Let  $N_q$  be the bandlimit of  $q$  ( $L = N_q$  in Definition 7.4). Then,*

$$N_s + N_q \geq N + 1 \tag{7.22}$$

The above bound is tight when  $s = \delta_{sp,0} = \frac{1}{\sqrt{N}}\mathbf{1}$  as defined in (3.48). In this case,  $q = \mathcal{V}^{*-1}s = \frac{1}{\sqrt{N}}e_0 = [\frac{1}{\sqrt{N}}, 0, \dots, 0]^T$ . This is because the first column of  $\mathcal{V}^*$  is  $\mathbf{1}$ , the vector of all 1s. The support of  $s$  is  $N_s = N$ . The bandlimit of the corresponding  $q$  is 1. So, in this case,  $N_s + N_q = N + 1$ .

In DSP, we have  $q = \hat{s}$ . This means the interpolating polynomial  $q(x)$  coefficients is the same as the frequency signal  $\hat{s}$ . We can rewrite result 7.8 using this fact.

**Result 7.9.** (*DSP Uncertainty Principle v2:  $s, \hat{s}$* ) Let  $N_s$  be the support of  $s$  ( $L = N_s$  in Definition 7.3). Let  $N_{\hat{s}}$  be the bandlimit of  $\hat{s}$  ( $L = N_{\hat{s}}$  in Definition 7.4). Then,

$$N_s + N_{\hat{s}} \geq N + 1 \quad (7.23)$$

This result may seem the same as Result 7.4. However, there is a key difference. In result 7.8, we assume  $s$  has  $N_s$  non-zero entries and  $\hat{s}$  contains  $N - N_{\hat{s}}$  trailing zeros. In 7.3, we assume  $s$  has  $N - N_s$  trailing zeros and  $\hat{s}$  contains  $N_{\hat{s}}$  non-zero entries.

**Remark 7.4.** (*Comparison with Uncertainty Principle in [53]*) The DSP Uncertainty Principle in [53] considers signals with a certain number of zeros and non-zeros in any location. The results rely on the DFT matrix being both a Vandermonde matrix in terms of its rows and its columns. In GSP,  $p$  and  $\hat{s}$  are related using only a Vandermonde matrix in terms of its columns, not its rows. So, in GSP, we assume the zeros of  $p$  or  $q$  are at the end and cannot assume they are anywhere. Because of this, applying our GSP Uncertainty Principle to DSP (Results 7.4 and 7.9) yields a different Uncertainty Principle than [53] (see fig. 7.5). This is because we assume either  $s$  or  $\hat{s}$  has zeros at the end of the signal while [53] does not, producing a tighter bound than [53].

We stated that if  $\hat{s}$  contains 1 zero, then  $s$  has at most 2 trailing zeros in DSP. We can find signals for  $\hat{s}$  with exactly one zero where  $s$  has 0, 1, 2 trailing zeros (see Table 7.1). But what determines whether  $s$  has 0, 1 or 2 trailing zeros? This is addressed in general for GSP in the next section 7.3.

### 7.4.3 Reduced Support $s$ : Reducing the Vandermonde System

This section presents the dual of Algorithm 1 in  $\text{GSP}_{\text{sp}}$ , starting with a reduced support vertex domain signal  $s$  (instead of  $\hat{s}$ ) and finding the spectral  $z$ -domain signal,  $q$  (instead of  $p$ ). Equation (7.21) is the dual of (7.10). Since  $\mathcal{V}^*$  is a Vandermonde matrix of the eigenvalues  $\lambda^*$ , (7.21) is equivalent to interpolating  $N$  points,  $(\lambda_i^*, \sqrt{N}s_i)$ , using a  $N - 1$  degree polynomial

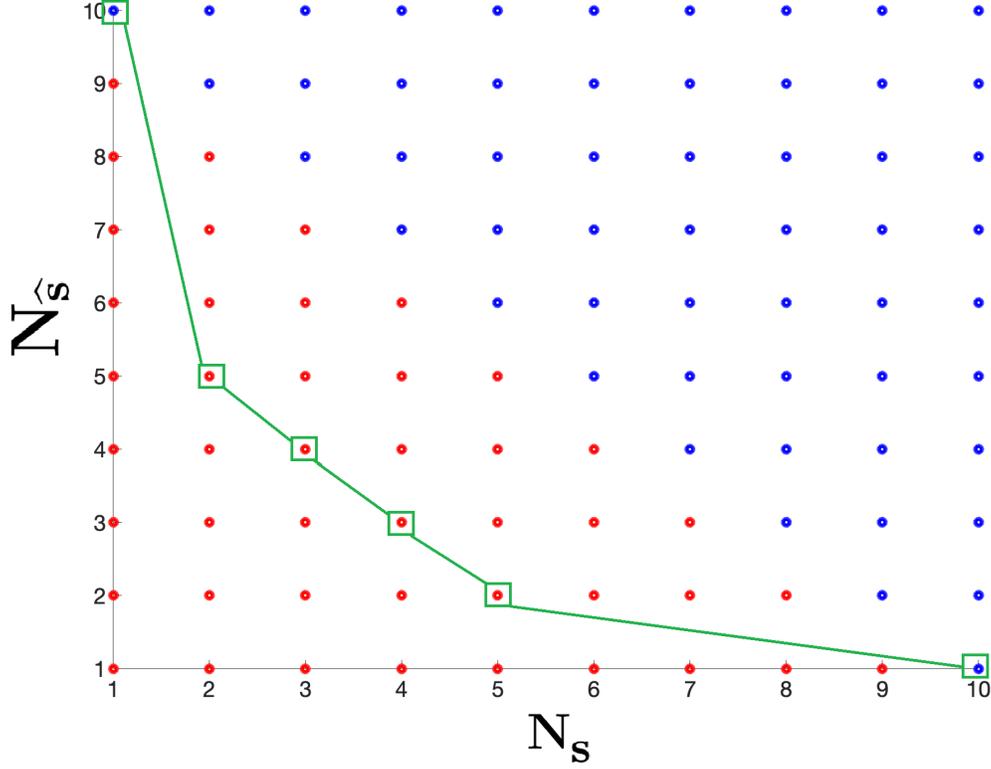


Figure 7.5: The feasibility region for the DSP uncertainty principle in result 7.3 for  $N = 10$ . Red dots represent infeasible regions and blue dots represent feasible regions. The green line represents the boundary using the DSP Uncertainty Principle in the literature.

with (ordered) polynomial coefficients,  $q$ . Let the interpolating polynomial be  $q(x)$ .

Let  $s$  be a  $N_s$ -bandlimited signal in (7.20). Let  $K = N - N_s$  be the number of zeros in  $s$ . We split  $q(x)$  into the product of  $r(x)$  and  $e(x)$ :

$$q(x) = r(x)e(x)$$

where  $r(x)$  is the  $K$  degree polynomial with the  $K$  roots at  $\lambda_i^*$  for  $i = N - K, \dots, N - 1$  corresponding to the points  $(\lambda_i^*, \sqrt{N}\widehat{s}_i = 0)$  and leading coefficient 1 and  $e(x)$  is a polynomial of at most degree  $N - K - 1$ .

Since  $r(x)$  has  $K$  roots at  $\lambda_i$ , we can determine  $r(x)$ .

$$r(x) = \prod_{i=N-K}^{N-1} (x - \lambda_i^*) \quad (7.24)$$

An alternative way is to take the characteristic polynomial  $\Delta_M(x)$  of  $M$  and divide by the polynomial with leading coefficient 1 and roots  $\lambda_i^*$ ,  $i = 0, 1, \dots, N - K - 1$ .

$$r(x) = \frac{\Delta_M(x)}{\prod_{i=0}^{N-K-1} (x - \lambda_i^*)} \quad (7.25)$$

Interpolation of  $e(x)$  can be done using a Vandermonde matrix. Let  $\lambda_s^*$  be a vector of the conjugate of the first  $\lambda$  values,  $\lambda_s^* = [\lambda_0^*, \lambda_1^*, \dots, \lambda_{N-K-1}^*]^T$ . Let  $e$  be the coefficient vector of  $e(x)$  with  $e_0$  being the coefficient of the constant term,  $e_1$  being the coefficient of the  $x$  term, etc. Let  $V_s^*$  be the  $(N - K) \times (N - K)$  Vandermonde matrix of  $\lambda_s^*$ ,  $V_s^* = [\lambda_s^{*0}, \lambda_s^{*1}, \dots, \lambda_s^{*(N-K-1)}]$ . Let  $d$  be the vector of  $\frac{\sqrt{N}s_i}{r(\lambda_i^*)}$ ,  $d = \left[ \frac{\sqrt{N}s_0}{r(\lambda_0^*)}, \frac{\sqrt{N}s_1}{r(\lambda_1^*)}, \dots, \frac{\sqrt{N}s_{N-K-1}}{r(\lambda_{N-K-1}^*)} \right]^T$ . Then,

$$d = V_s^* e \quad (7.26)$$

and  $e = V_s^{*-1} d$ . Using the coefficients  $e$ , we can determine  $e(x)$ .

Algorithm 1 can also be used to find  $q$  for a bandlimited  $s$  by replacing  $p$  with  $q$ ,  $\hat{s}$  for  $s$ ,  $\lambda_i$  with  $\lambda_i^*$  and  $p(x)$  with  $q(x)$ .

---

**Algorithm 2:** Find  $q$  using (7.21) for  $(N - K)$ -bandlimited  $s$

---

- 1 **Given:** Eigenvalue vector  $\lambda^*$ ,  $(N - K)$ -bandlimited  $s$  from (7.20).
  - 2 Produce  $N$  points from  $\lambda^*$  and  $s$ :
  - 3  $(\lambda_i^*, \sqrt{N}s_i)$  ( $i = 0, \dots, N - K - 1$ ,  $N - K$  non-roots),  $(\lambda_i^*, s_i = 0)$  ( $i = N - K, \dots, N - 1$ ,  $K$  roots)
  - 4 Using the  $K$  roots, find  $r(x)$  using (7.24) or (7.25). From  $N - K$  non-roots, form  $N - K$  points,  $\left( \lambda_i^*, \frac{\sqrt{N}s_i}{r(\lambda_i^*)} \right)$ ,  $i = 0, \dots, N - K - 1$ .
  - 5 Using (7.26), find  $e$ . Using  $e$ , find  $e(x)$ .
  - 6 Multiply  $r(x)$  and  $e(x)$  to find  $q(x)$ .
  - 7 Take the coefficients from  $q(x)$  to find  $q$ .
-

## 7.5 Conclusion

In this chapter, we provided a GSP Uncertainty Principle for  $p$  and  $\hat{s}$ , as well as its dual, a GSP Uncertainty Principle for  $q$  and  $s$ . Previous GSP uncertainty principles relate the spread of vertex domain signal  $s$  and spectral domain signal  $\hat{s}$ . Unlike previous GSP uncertainty principles, we directly relate the bandlimitedness of  $p$  and bandlimitedness of  $\hat{s}$ . We show a tradeoff between the bandlimits of  $p$  and  $\hat{s}$ . As the number of non-zeros in  $\hat{s}$  increases, the lower bound on the bandlimitedness of  $p$  also decreases. Similarly, if the number of non-zeros in  $\hat{s}$  decreases, the lower bound on the bandlimitedness of  $p$  increases. Using these uncertainty principles, we give algorithms for reducing the Vandermonde system for finding  $p$  and  $q$  when  $\hat{s}$  is bandlimited or when  $s$  has reduced support respectively.



# Chapter 8

## GSP Interpolating Filters

This chapter performs Lagrange interpolation using Lagrange basis polynomials in GSP to find  $p$ , given  $\hat{s}$ . Then, we explore properties of Lagrange basis polynomial filters. These are the Lagrange basis polynomials evaluated using the graph shift  $A$  and spectral graph shift  $M$ .

### 8.1 Finding $p$ : Lagrange Interpolation in GSP

In this section, we derive a general method for finding  $p$  given  $\hat{s}$  that avoids solving the linear system in (7.10). This general method is based on Algorithm 1 presented in the last section. We then show this method yields an interesting relationship between the eigenvalues and eigenvectors of  $A$ .

Consider the unit vector  $e_i$ , a vector with a 1 in the  $i$ th position and 0s elsewhere. We want to find the  $p(x)$  corresponding to  $\hat{s} = \frac{1}{\sqrt{N}}e_i$ . From (7.10), we observe this is interpolation with the  $N - 1$  points  $(\lambda_j, 0)$ ,  $j \neq i$  and the point  $(\lambda_i, \frac{1}{\sqrt{N}})$ . Following Algorithm 1, obtain:

$$r(x) = \prod_{j=0, j \neq i}^{N-1} (x - \lambda_j) \quad (8.1)$$

$$e = V_s^{-1}d = [1]^{-1} \begin{bmatrix} 1 \\ r(\lambda_i) \end{bmatrix} = \frac{1}{r(\lambda_i)} = \frac{1}{\prod_{j=0, j \neq i}^{N-1} (\lambda_i - \lambda_j)} \quad (8.2)$$

$$e(x) = \frac{1}{\prod_{j=0, j \neq i}^{N-1} (\lambda_i - \lambda_j)} \quad (8.3)$$

$$p(x) = \frac{r(x)}{r(\lambda_i)} = \frac{\prod_{j=0, j \neq i}^{N-1} (x - \lambda_j)}{\prod_{j=0, j \neq i}^{N-1} (\lambda_i - \lambda_j)} \quad (8.4)$$

where  $e(x)$  is degree 0 (constant),  $r(x)$  and  $p(x)$  are degree  $N$ . For each unit vector,  $e_i$ ,  $i = 0, \dots, N - 1$ , we obtain a different interpolating polynomial, defined below.

**Definition 8.1.** Let  $\ell_i(x)$  be the interpolating polynomial for the unit vector  $\frac{1}{\sqrt{N}}e_i$ ,  $i = 0, \dots, N - 1$ . Let the  $N$  interpolating polynomials,  $\ell_i(x)$ ,  $i = 0, \dots, N - 1$ , be the **Lagrange basis polynomials**. From (8.4),

$$\ell_i(x) = \frac{\prod_{j=0, j \neq i}^{N-1} (x - \lambda_j)}{\prod_{j=0, j \neq i}^{N-1} (\lambda_i - \lambda_j)}, \quad i = 0, \dots, N - 1 \quad (8.5)$$

**Result 8.1.** The Lagrange basis polynomials,  $\ell_i(x)$  form a basis for  $p(x)$  with coefficients  $\sqrt{N}\hat{s}_i$ .

*Proof.* Rewrite  $\hat{s}$  using the standard basis:

$$\hat{s} = \sum_{i=0}^{N-1} \hat{s}_i e_i = \sum_{i=0}^{N-1} \sqrt{N}\hat{s}_i \frac{1}{\sqrt{N}} e_i$$

. Using (7.10), obtain:

$$p = \mathcal{V}^{-1}\hat{s} = \mathcal{V}^{-1} \left( \sum_{i=0}^{N-1} \sqrt{N}\hat{s}_i \frac{1}{\sqrt{N}} e_i \right) = \sum_{i=0}^{N-1} \sqrt{N}\hat{s}_i \left( \mathcal{V}^{-1} \frac{1}{\sqrt{N}} e_i \right)$$

$\mathcal{V}^{-1} \frac{1}{\sqrt{N}} e_i$  is the polynomial interpolation of the unit vector  $\frac{1}{\sqrt{N}} e_i$  with  $\lambda_j$ ,  $j = 0, \dots, N - 1$ .

From (8.4), we know the interpolating polynomial for this is the Lagrange basis polynomial,

$\ell_i(x)$ . So,

$$p = \sum_{i=0}^{N-1} \sqrt{N}\hat{s}_i \ell_i$$

Thus,

$$p(x) = \sum_{i=0}^{N-1} \sqrt{N}\hat{s}_i \ell_i(x) = \sum_{i=0}^{N-1} \sqrt{N}\hat{s}_i \frac{\prod_{j=0, j \neq i}^{N-1} (x - \lambda_j)}{\prod_{j=0, j \neq i}^{N-1} (\lambda_i - \lambda_j)} \quad (8.6)$$

■

Equation (8.6) provides a general way to solve the system in (7.10) through interpolation.

**Remark 8.1.** Equation (8.6) is an example of Lagrange interpolation, a well-known method

for finding  $s$  given  $\hat{s}$  in DSP. In GSP, one cannot use (Lagrange) interpolation to find  $s$  given  $\hat{s}$ . However, (8.6) shows that Lagrange interpolation can be used to find  $p$  given  $\hat{s}$  in GSP.

We verify that  $p(x)$  in (8.6) is the interpolating polynomial of points  $(\lambda_j, \sqrt{N}\hat{s}_j)$ ,  $j = 0, \dots, N-1$ . For each Lagrange basis polynomial,

$$\ell_i(\lambda_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \quad (8.7)$$

When  $i = j$ , the numerator and denominator are the same, so  $\ell_i(\lambda_j) = 1$ . When  $i \neq j$ , the numerator of  $\ell_i(\lambda_j)$  contains  $x - \lambda_j = \lambda_j - \lambda_j = 0$ . So,  $\ell_i(\lambda_j) = 0$ . Using (8.6),  $p(\lambda_j) = \sum_{i=0}^{N-1} \sqrt{N}\hat{s}_i \ell_i(\lambda_j) = \sqrt{N}\hat{s}_j$ ,  $j = 0, \dots, N-1$ . So,  $p(x)$  is the interpolating polynomial of  $(\lambda_j, \sqrt{N}\hat{s}_j)$ .

**Remark 8.2.** (*Signal representations for  $p(x)$* ) We present two different signal representations of  $p(x)$ . There is the standard basis: using the powers of  $x$  as the basis and  $p$  as the coefficients. Equation (8.6) gives a second signal representation of  $p(x)$ : using the Lagrange basis polynomials and  $\sqrt{N}\hat{s}$  as the coefficients.

$$p(x) = \sum_{i=0}^{N-1} p_i x^i = \sum_{i=0}^{N-1} \sqrt{N} \ell_i(x) \hat{s}_i \quad (8.8)$$

**Result 8.2.** *Let  $s$  be an eigenvector of  $A$ ,  $s = v_i$ . Then,  $p(x) = \sqrt{N}\ell_i(x)$ .*

*Proof.* The result follows from (8.4). Since  $\hat{s} = \text{GFT } s = e_i$ ,  $p(x) = \sqrt{N}\ell_i(x)$ . ■

Result 8.2 establishes the relationship between the Lagrange polynomials and the eigenvectors of  $A$  in the vertex domain. Equation (8.7) shows that the Lagrange polynomials are a set of polynomials ( $\mathbb{R} \rightarrow \mathbb{R}$ ) that become impulse (delta) functions (1 at a single frequency and 0 otherwise) when evaluated at the graph frequencies,  $\lambda_i$ . This is consistent with DSP, where the polynomial is the  $z$ -transform and the Fourier transform is found by evaluating the

polynomial at the graph frequencies (roots of unity in DSP) to produce  $\widehat{s}$ . We explore  $\ell_i(x)$  in DSP next.

**Remark 8.3.** *In DSP,  $s = p$ . So, the coefficients of  $\ell_i(x)$  are equal to the signal  $s$ . Result 8.2 gives that  $s = v_i$  when  $p(x) = \sqrt{N}\ell_i(x)$ . So,  $s = p = v_i$  and the coefficients of  $\sqrt{N}\ell_i(x)$  are equal to the values of  $v_i$ .*

## 8.2 Lagrange Basis Polynomial Filters $\ell_i(A)$

In the previous section, we introduced Lagrange basis polynomials,  $\ell_i(x)$ , to solve (7.10). Polynomial  $p(x)$  has the same polynomial coefficients as  $P(A)$ , the LSI polynomial filter in the vertex domain. In this section, we explore the significance and properties of Lagrange basis polynomials of  $A$ ,  $M$  and  $C_{\text{comp}}$ . These are LSI GSP filters in the vertex domain, spectral domain and vertex impulsive domains respectively.

We find  $\ell_i(A)$  using (8.5).

$$\ell_i(A) = \frac{\prod_{j=0, j \neq i}^{N-1} (A - \lambda_j I)}{\prod_{j=0, j \neq i}^{N-1} (\lambda_i - \lambda_j)}, \quad i = 0, \dots, N - 1 \quad (8.9)$$

In GSP,

$$P(A)\delta_0 = s.$$

Since  $P(A)$  contains a weighted sum of Lagrange basis polynomials of  $A$  (see (8.6)), we consider the effect of  $\ell_i(A)$  on  $\delta_0$  in result 8.3.

**Result 8.3.** *Applying the  $\ell_i(A)$  filter to the delta function  $\delta_0$  yields the corresponding eigenvector for all  $i = 0, \dots, N - 1$ :*

$$\ell_i(A)\delta_0 = \frac{1}{\sqrt{N}}v_i \quad (8.10)$$

*Proof.* By 3.46,

$$\delta_0 = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} v_k.$$

By linearity,

$$\ell_i(A)\delta_0 = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \ell_i(A)v_k = \sum_{k=0}^{N-1} \frac{1}{\sqrt{N}} \frac{\prod_{j=0, j \neq i}^{N-1} (A - \lambda_j I)}{\prod_{j=0, j \neq i}^{N-1} (\lambda_i - \lambda_j)} v_k.$$

Since  $v_k$  is an eigenvector,

$$(A - \lambda_j I)v_k = (\lambda_k - \lambda_j)v_k$$

for all  $j, k = 0, \dots, N - 1$ . So, we can rewrite  $\ell_i(A)\delta_0$  as

$$\sum_{k=0}^{N-1} \frac{1}{\sqrt{N}} \frac{\prod_{j=0, j \neq i}^{N-1} (\lambda_k - \lambda_j)}{\prod_{j=0, j \neq i}^{N-1} (\lambda_i - \lambda_j)} v_k$$

Consider two cases:

1) If  $k = i$ , then

$$\frac{1}{\sqrt{N}} \frac{\prod_{j=0, j \neq i}^{N-1} (\lambda_k - \lambda_j)}{\prod_{j=0, j \neq i}^{N-1} (\lambda_i - \lambda_j)} v_k = 1 \cdot v_k = v_i.$$

2) If  $k \neq i$ , then the numerator of

$$\frac{1}{\sqrt{N}} \frac{\prod_{j=0, j \neq i}^{N-1} (\lambda_k - \lambda_j)}{\prod_{j=0, j \neq i}^{N-1} (\lambda_i - \lambda_j)} v_k$$

contains

$$\lambda_j - \lambda_j = 0.$$

So, the product is

$$0 \cdot v_k = 0.$$

Thus, the term in the summation is only non-zero when  $k = i$ . So,

$$\ell_i(A)\delta_0 = \frac{1}{\sqrt{N}} v_i$$

for all  $i = 0, \dots, N - 1$ . ■

A simple corollary of result 8.3 is

$$\ell_i(A)\widehat{s}_i\delta_0 = \frac{1}{\sqrt{N}}\widehat{s}_iv_i,$$

obtained since  $\ell_i(A)$  is LSI. Next, we generalize result 8.3 to any signal  $s$ .

**Result 8.4.**

$$\ell_i(A)s = \widehat{s}_iv_i$$

*Proof.* Rewrite  $s$  as weighted sum of eigenvectors:

$$s = \sum_{k=0}^{N-1} \widehat{s}_k v_k.$$

So,

$$\ell_i(A)s = \sum_{k=0}^{N-1} \widehat{s}_k \ell_i(A)v_k$$

Following a similar argument as the proof of result 8.3,

$$\ell_i(A)v_k = 0$$

when  $k \neq i$ . When  $k = i$ ,

$$\ell_i(A)v_k = v_i$$

Thus,

$$\ell_i(A)s = \widehat{s}_iv_i$$

■

Result 8.4 shows that applying filter  $\ell_i(A)$  to any signal  $s$  extracts the eigenvector  $v_i$  component of  $s$  (with the same scaling  $\widehat{s}$ ).

**Result 8.5.**

$$\sum_{i=0}^{N-1} \ell_i(A)s = s.$$

*Proof.* From result 8.4,

$$\ell_i(A)s = \widehat{s}_i v_i.$$

So,

$$\sum_{i=0}^{N-1} \ell_i(A)s = \sum_{i=0}^{N-1} \widehat{s}_i v_i = s.$$

■

Result 8.5 shows that we can re-obtain  $s$  by summing  $\ell_i(A)s$  together. This is intuitively pleasing because each filter extracts the  $v_i$  component of  $s$  and then the summation puts the components back together to form  $s$ .

**Corollary 8.1.** (*All-pass graph filter*)

$$\sum_{i=0}^{N-1} \ell_i(A) = I.$$

*Proof.* This is a well known fact of Lagrange basis polynomials. It can be seen from result 8.5:

$$\sum_{i=0}^{N-1} \ell_i(A)s = \left( \sum_{i=0}^{N-1} \ell_i(A) \right) s = s.$$

■

Corollary 8.1 shows the effect of applying the combined filter  $\sum_{i=0}^{N-1} \ell_i(A)$  on signal  $s$ . It is equivalent to applying the identity (all-pass) filter. This is shown in fig. 8.1.

$\ell_i(A)$  is a  $\mathbb{C}^N \rightarrow \mathbb{C}^N$  mapping of signal  $s \in \mathbb{C}^N$  to the line  $kv_i$ ,  $k \in \mathbb{C}$ . They are a set of narrow-band filters (tuned to spectral component  $v_i$ ), each extracting a single eigenvector (spectral) component  $v_i$ .

**Result 8.6.** *We list some general properties of  $\ell_i(A)$ :*

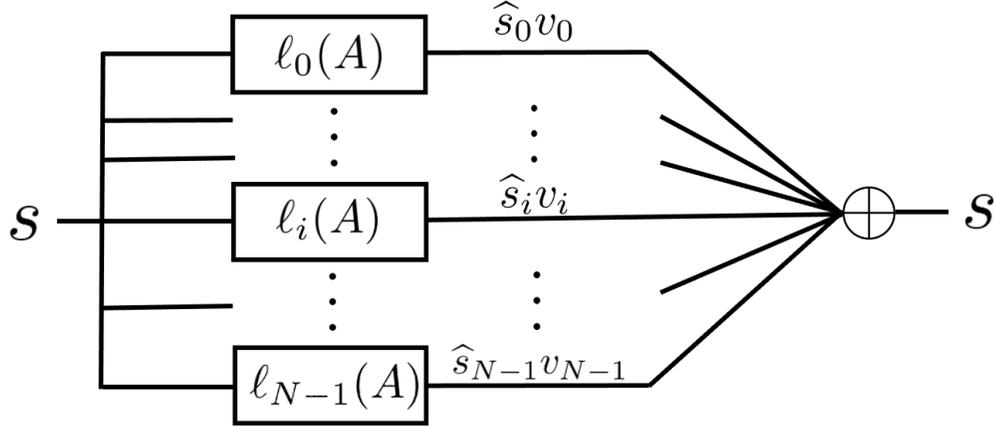


Figure 8.1: Block diagram applying the all-pass graph filter in Corollary 8.1 to  $s$ . Each Lagrange polynomial filter  $\ell_i(A)$  extracts weighted eigenvector of  $A$ ,  $\widehat{s}_i v_i$ . These are then summed together to reobtain  $s$ .

- 1)  $\ell_i(A)$  is a polynomial of  $A$  with degree  $N - 1$ .
- 2)  $\ell_i(A)s = \text{proj}_{v_i} s$ .
- 3)  $\ell_i(A)$  has rank 1.<sup>1</sup>

Property 2 says that  $\ell_i(A)s$  is generally, an oblique projection of  $s$  onto the eigenvector  $v_i$  because the eigenvectors are not orthogonal. If  $A$  is symmetric (such as with undirected graphs), then the eigenvectors of  $A$  are orthogonal and  $\ell_i(A)s$  is an orthogonal projection.

**Remark 8.4.** We explore  $\ell_i(A_c)$  in DSP. By property 1 of result 8.6,  $\ell_i(A_c)$  is a polynomial of  $A_c$ , so it is a circulant matrix in DSP. By result 8.3,  $\ell_i(A_c)\delta_0 = \frac{1}{\sqrt{N}}v_i$ . However,  $\delta_0 = e_0 = [1, 0, \dots, 0]^T$ . So, the first column of  $\ell_i(A_c)$  is  $\frac{1}{\sqrt{N}}v_i$ . Since the matrix is circulant, we obtain the entire matrix  $\ell_i(A_c)$  from the first column.  $\ell_i(A)$  is a circulant matrix whose first column is the eigenvector  $v_i$ :  $\ell_i(A_c) = \frac{1}{\sqrt{N}} [v_i \ A_c v_i \ A_c^2 v_i \ \dots \ A_c^{N-1} v_i]$ .

<sup>1</sup>This is due to property 2. The matrix projects a vector  $s$  onto a space with dimension 1, spanned by  $v_i$  vector. So, it has rank 1.

### 8.3 Finding $q$ : Lagrange Interpolation in $\text{GSP}_{\text{sp}}$

This section presents a general method for finding  $q$  using Lagrange interpolation. We obtain the dual results of previous section 8.1 in  $\text{GSP}_{\text{sp}}$ , using  $s$  and  $q$  instead of  $\hat{s}$  and  $p$ .

Consider the unit vector  $e_i$ , a vector with a 1 in the  $i$ th position and 0s elsewhere. We want to find the  $q(x)$  corresponding to  $s = \frac{1}{\sqrt{N}}e_i$ . From (7.21), we observe this is interpolation with the  $N - 1$  points  $(\lambda_j^*, 0)$ ,  $j \neq i$  and the point  $(\lambda_i^*, \frac{1}{\sqrt{N}})$ . Following Algorithm 2, obtain:

$$r(x) = \prod_{j=0, j \neq i}^{N-1} (x - \lambda_j^*) \quad (8.11)$$

$$e = V_s^{*-1}d = [1]^{-1} \left[ \frac{1}{r(\lambda_i^*)} \right] = \frac{1}{r(\lambda_i^*)} = \frac{1}{\prod_{j=0, j \neq i}^{N-1} (\lambda_i^* - \lambda_j^*)} \quad (8.12)$$

$$e(x) = \frac{1}{\prod_{j=0, j \neq i}^{N-1} (\lambda_i^* - \lambda_j^*)} \quad (8.13)$$

$$p(x) = \frac{r(x)}{r(\lambda_i^*)} = \frac{\prod_{j=0, j \neq i}^{N-1} (x - \lambda_j^*)}{\prod_{j=0, j \neq i}^{N-1} (\lambda_i^* - \lambda_j^*)} \quad (8.14)$$

where  $e(x)$  is degree 0 (constant),  $r(x)$  and  $p(x)$  are degree  $N$ . For each unit vector,  $e_i$ ,  $i = 0, \dots, N - 1$ , we obtain a different interpolating polynomial, defined below.

**Definition 8.2.** Let  $\ell_{i,\text{sp}}(x)$  be the interpolating polynomial for the unit vector  $\frac{1}{\sqrt{N}}e_i$ ,  $i = 0, \dots, N - 1$ . The  $N$  interpolating polynomials,  $\ell_{i,\text{sp}}(x)$ ,  $i = 0, \dots, N - 1$ , be the **spectral Lagrange basis polynomials**. From (8.4),

$$\ell_{i,\text{sp}}(x) = \frac{\prod_{j=0, j \neq i}^{N-1} (x - \lambda_j^*)}{\prod_{j=0, j \neq i}^{N-1} (\lambda_i^* - \lambda_j^*)}, \quad i = 0, \dots, N - 1 \quad (8.15)$$

**Result 8.7.** The spectral Lagrange basis polynomials,  $\ell_{i,\text{sp}}(x)$  form a basis for  $q(x)$  with coefficients  $\sqrt{N}s_i$ .

*Proof.* Rewrite  $s$  using the standard basis:

$$s = \sum_{i=0}^{N-1} s_i e_i = \sum_{i=0}^{N-1} \sqrt{N} s_i \frac{1}{\sqrt{N}} e_i$$

Using (7.21), obtain:

$$q = \mathcal{V}^{*-1} s = \mathcal{V}^{*-1} \left( \sum_{i=0}^{N-1} \sqrt{N} s_i \frac{1}{\sqrt{N}} e_i \right) = \sum_{i=0}^{N-1} \sqrt{N} s_i \left( \mathcal{V}^{*-1} \frac{1}{\sqrt{N}} e_i \right)$$

$\mathcal{V}^{*-1} \frac{1}{\sqrt{N}} e_i$  is the polynomial interpolation of the unit vector  $\frac{1}{\sqrt{N}} e_i$  with  $\lambda_j^*$ ,  $j = 0, \dots, N - 1$ . From (8.4), we know the interpolating polynomial for this is the spectral Lagrange basis polynomial,  $\ell_{i,sp}(x)$ . So,

$$q = \sum_{i=0}^{N-1} \sqrt{N} s_i \ell_{i,sp}$$

Thus,

$$q(x) = \sum_{i=0}^{N-1} \sqrt{N} s_i \ell_{i,sp}(x) = \sum_{i=0}^{N-1} \sqrt{N} s_i \frac{\prod_{j=0, j \neq i}^{N-1} (x - \lambda_j^*)}{\prod_{j=0, j \neq i}^{N-1} (\lambda_i^* - \lambda_j^*)} \quad (8.16)$$

■

Equation (8.16) provides a general way to solve the system in (7.21) through interpolation.

**Remark 8.5.** Equation (8.6) is an example of Lagrange interpolation, a well-known method for finding  $\hat{s}$  given  $s$  in DSP. In GSP, one cannot use (Lagrange) interpolation to find  $\hat{s}$  given  $s$ . However, (8.16) shows that Lagrange interpolation can be used to find  $q$  given  $s$  in GSP.

**Remark 8.6.** (Signal representations for  $q(x)$ ) We present two different signal representations of  $q(x)$ . There is the standard basis: using the powers of  $x$  as the basis and  $q$  as the coefficients. Equation (8.6) gives a second signal representation of  $q(x)$ : using the Lagrange basis polynomials and  $\sqrt{N}s$  as the coefficients.

$$q(x) = \sum_{i=0}^{N-1} q_i x^i = \sum_{i=0}^{N-1} \sqrt{N} \ell_{i,sp}(x) s_i \quad (8.17)$$

In  $\text{GSP}_{\text{sp}}$ , the GFT matrix plays the role that  $\text{GFT}^{-1}$  plays in GSP. Let the columns of the GFT be  $u_i$ . Similar to how the columns of the  $\text{GFT}^{-1}$  are the eigenvectors of  $A$ , the columns of the GFT are the eigenvectors of  $M$ .

$$\text{GFT} = [u_0, u_1, u_2, \dots, u_{N-1}] \quad (8.18)$$

**Result 8.8.** *Let  $\hat{s}$  be an eigenvector of  $M$ ,  $\hat{s} = u_i$ . Then,  $q(x) = \sqrt{N}\ell_{i,\text{sp}}(x)$ .*

*Proof.* The result follows from (8.14). Since  $s = \text{GFT}^{-1}\hat{s} = e_i$ ,  $q(x) = \sqrt{N}\ell_{i,\text{sp}}(x)$ . ■

Result 8.8 establishes the relationship between the spectral Lagrange polynomials and the eigenvectors of  $M$  in the *spectral* domain. The spectral Lagrange polynomials are a set of polynomials ( $\mathbb{R} \rightarrow \mathbb{R}$ ) that become impulse (delta) functions (1 at a single frequency and 0 otherwise) when evaluated at the graph frequencies,  $\lambda_i^*$ . We explore  $\ell_{i,\text{sp}}(x)$  in DSP next.

**Remark 8.7.** *In DSP,  $\hat{s} = q$ . So, the coefficients of  $\ell_{i,\text{sp}}(x)$  are equal to the signal  $\hat{s}$ . Result 8.8 gives that  $\hat{s} = u_i$  when  $q(x) = \sqrt{N}\ell_{i,\text{sp}}(x)$ . So,  $\hat{s} = q = u_i$  and the coefficients of  $\sqrt{N}\ell_{i,\text{sp}}(x)$  are equal to the values of  $u_i$ .*

## 8.4 Lagrange Basis Polynomial Filters $\ell_{i,\text{sp}}(M)$

In this section, we obtain the dual results of section 8.2 (without proof), working with  $\ell_{i,\text{sp}}(M)$  and  $\hat{s}$  instead of  $\ell_i(A)$  and  $s$ . We obtain similar properties for the spectral Lagrange basis polynomial filters,  $\ell_{i,\text{sp}}(M)$ , in  $\text{GSP}_{\text{sp}}$ .

We use the spectral Lagrange basis polynomials in (8.15).

$$\ell_{i,\text{sp}}(M) = \frac{\prod_{j=0, j \neq i}^{N-1} (M - \lambda_j^* I)}{\prod_{j=0, j \neq i}^{N-1} (\lambda_i^* - \lambda_j^*)}, \quad i = 0, \dots, N-1 \quad (8.19)$$

**Result 8.9.** *Applying the  $\ell_{i,\text{sp}}(M)$  filter to the spectral delta function  $\widehat{\delta}_{\text{sp},0}$  yields the corre-*

spending eigenvector for all  $i = 0, \dots, N - 1$ :

$$\ell_{i,sp}(M)\widehat{\delta}_{sp,0} = \frac{1}{\sqrt{N}}u_i \quad (8.20)$$

with  $u_i$  given in (8.18).

A simple corollary of result 8.9 is

$$\ell_{i,sp}(M)s_i\widehat{\delta}_{sp,0} = \frac{1}{\sqrt{N}}s_iu_i.$$

**Result 8.10.**

$$\ell_{i,sp}(M)\widehat{s} = s_iu_i.$$

**Result 8.11.**

$$\sum_{i=0}^{N-1} \ell_{i,sp}(M)\widehat{s} = \widehat{s}.$$

*Proof.* From result 8.10,

$$\ell_{i,sp}(M)\widehat{s} = s_iu_i.$$

So,

$$\sum_{i=0}^{N-1} \ell_{i,sp}(M)\widehat{s} = \sum_{i=0}^{N-1} s_iu_i = \widehat{s}.$$

■

Result 8.11 shows that we can re-obtain  $\widehat{s}$  by summing  $\ell_{i,sp}(M)\widehat{s}$  together. This is intuitively pleasing because each filter extracts the  $v_i$  component of  $s$  and then the summation puts the components back together to form  $\widehat{s}$ .

**Corollary 8.2.** (*All-pass spectral graph filter*)

$$\sum_{i=0}^{N-1} \ell_{i,sp}(M) = I.$$

*Proof.* This is a well known fact of Lagrange basis polynomials. It can be seen from result

8.11:

$$\sum_{i=0}^{N-1} \ell_i(A)s = \left( \sum_{i=0}^{N-1} \ell_{i,sp}(M) \right) \hat{s} = \hat{s}.$$

■

Corollary 8.2 shows the effect of applying the combined filter  $\sum_{i=0}^{N-1} \ell_{i,sp}(M)$  on signal  $\hat{s}$ . It is equivalent to applying the identity (all-pass) filter. This is shown in fig. 8.2.

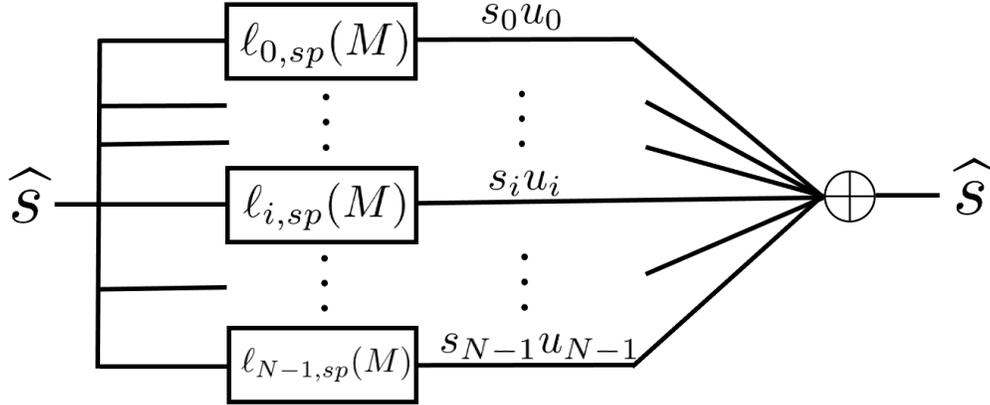


Figure 8.2: Block diagram applying the all-pass graph filter in Corollary 8.2 to  $\hat{s}$ . Each Lagrange polynomial filter  $\ell_{i,sp}(M)$  extracts weighted eigenvector of  $M$ ,  $s_i u_i$ . These are then summed together to reobtain  $\hat{s}$ .

$\ell_{i,sp}(M)$  is a  $\mathbb{C}^N \rightarrow \mathbb{C}^N$  mapping of signal  $s \in \mathbb{C}^N$  to the line  $ku_i$ ,  $k \in \mathbb{C}$ . They are a set of narrow-band filters (tuned to vertex component  $u_i$ ), each extracting a single eigenvector (vertex) component  $u_i$ .

**Result 8.12.** *We list some general properties of  $\ell_{i,sp}(M)$ :*

- 1)  $\ell_{i,sp}(M)$  is a polynomial of  $M$  with degree  $N - 1$ .
- 2)  $\ell_{i,sp}(M)\hat{s} = \text{proj}_{u_i}\hat{s}$ .
- 3)  $\ell_{i,sp}(M)$  has rank 1.

## 8.5 Conclusion

This chapter performs Lagrange interpolation on the points  $(\lambda_i, \sqrt{N}\hat{s}_i)$  to find  $p$  given  $\hat{s}$ . In doing so, we show that the Lagrange basis polynomials,  $\ell_i(x)$ , form a basis for  $p$  with coefficients

$\sqrt{N}\widehat{s}_i$ . Then, we explore Lagrange basis polynomial filters,  $\ell_i(A)$ . We show various properties of  $\ell_i(A)$ . They are a set of narrow-band filters (tuned to spectral component  $v_i$ ), each extracting a single eigenvector  $v_i$ . Summing the polynomial filters together yields an all-pass filter. We also show the dual by substituting  $p$  with  $q$  and  $\widehat{s}$  with  $s$ . We perform Lagrange interpolation to find  $q$  given  $s$ . We obtain similar results for the spectral Lagrange basis polynomial filters,  $\ell_{i,sp}(M)$ .

## Chapter 9

# GSP Modulation and Demodulation

Multiplexing is the process of combining  $D$  signals,  $s^{(1)}, s^{(2)}, \dots, s^{(D)}$  into a single signal,  $r$ . This signal is then sent to a receiver that demodulates and recovers the original  $D$  signals,  $s^{(1)}, s^{(2)}, \dots, s^{(D)}$  from the received signal  $r$ .<sup>1</sup> In DSP, multiplexing is done by partitioning the signal bandwidth into  $D$  (non-overlapping) spectral bands with each band containing one signal  $s^{(d)}, d = 1, \dots, D$ . This partitioning can either happen in the time domain (time division multiplexing) or the frequency domain (frequency division multiplexing). This chapter develops GSP modulation and demodulation by exploring GSP multiplexing: vertex division multiplexing, spectral division multiplexing, and spectral  $z$ -transform division multiplexing. Vertex domain multiplexing is done by partitioning the vertex domain using sampling, described in Chapter 4. Spectral division multiplexing is done by partitioning the graph spectral domain. Spectral  $z$ -transform division multiplexing is done by partitioning the spectral  $z$ -transform domain, described in section 6.1.2.

In this chapter, we assume the signals  $s^{(1)}, s^{(2)}, \dots, s^{(D)}$  are  $K$ -bandlimited where the last  $N - K$  entries of  $\hat{s}$  are 0.<sup>2</sup>

$$\hat{s}^{(i)} = \begin{bmatrix} \hat{s}_K^{(i)} \\ 0_{N-K} \end{bmatrix} \quad (9.1)$$

We assume  $DK \leq N$ , i.e., the  $D$   $K$ -bandlimited signals all fit in the bandwidth of  $r$ ,  $N$ .

---

<sup>1</sup>On notation:  $s_i^{(d)}, d = 1, \dots, D, k = 0, 1, \dots, N-1$  refers to the  $i$ th entry of the  $d$ th signal being modulated. The number of signals is  $D$ . Each signal is dimension  $N \times 1$ .

<sup>2</sup>This is a notational convenience. This is not a necessary condition. The eigenvalues and eigenvectors of  $A$  can be permuted so that this is true.

## 9.1 Vertex Division Multiplexing

This section considers vertex (time) division multiplexing in DSP and GSP, which partitions the vertex (time) domain.

### 9.1.1 Time Division Multiplexing

In DSP, time division multiplexing is done by sampling each of the  $D$  signals  $s^{(1)}, s^{(2)}, \dots, s^{(D)}$  and summing them together to form  $r$ .

$$r = \sum_{d=1}^N (\delta_{s^{(d)}} \odot s^{(d)}) \quad (9.2)$$

where  $\delta_{s^{(d)}}$  is a zero-one sampling signal for  $s^{(d)}$ . Since the signals are  $K$ -bandlimited,  $\|\delta_{s^{(d)}}\|_0 = K$ .

To demodulate, each of the sampled signals  $\delta_{s^{(d)}} \odot s^{(d)}$  is obtained from  $r$ . Using a low-pass filter, the original signals  $s^{(d)}$  are recovered from their sampled signals,  $\delta_{s^{(d)}} \odot s^{(d)}$ .

### 9.1.2 Vertex Division Multiplexing

This method is similar to time division multiplexing. We partition the  $N$  nodes of  $A$  into  $D$  disjoint sets of  $K$  nodes. Each set of  $K$  nodes must be a sampling set  $S$  described in section 4.1.1. Let the sampling set associated with signal  $s^{(d)}$  be  $S_d$ .

We do vertex division multiplexing by sampling each by sampling each of the  $D$  signals  $s^{(1)}, s^{(2)}, \dots, s^{(D)}$  using  $\delta_{S_d}$  and summing them together to form  $r$ .

$$r = \sum_{d=1}^N (\delta_{S_d} \odot s^{(d)}) \quad (9.3)$$

where  $\delta_{S_d}$  is a zero-one sampling signal for  $s^{(d)}$  using sampling set  $S_d$ .

To demodulate  $s^{(d)}$ , we pointwise multiply  $r$  by  $\delta_{S_d}$ . Since the sampling sets are disjoint,

this yields:

$$\delta_{S_d} \odot r = \delta_{S_d} \odot \left( \sum_{d=1}^N (\delta_{S_d} \odot s^{(d)}) \right) = \delta_{S_d} \odot s^{(d)} \quad (9.4)$$

To obtain  $s^{(d)}$  from  $\delta_{S_d} \odot s^{(d)}$ , we use the sampling interpolation methods described in 4.2.

**Remark 9.1.** *The dual of vertex division multiplexing is sampling  $\hat{s}$  in the spectral domain instead of  $s$  in the vertex domain. Instead of dividing the nodes  $N$  of  $A$  into  $K$  groups, we partition the  $N$  nodes of  $M$  into  $K$  groups. Demodulation is done in a similar fashion; inverting a small block of the GFT instead of the  $GFT^{-1}$  to recover the signal. While this does partition the spectral bandwidth, we do not consider this the spectral division multiplexing described in the next section 9.2, but rather the dual of vertex division multiplexing.*

## 9.2 Spectral Division Multiplexing

This section considers spectral (frequency) division multiplexing in DSP and GSP, which partitions the spectral (frequency) domain.

### 9.2.1 Frequency Division Multiplexing

In DSP, given  $K$ -bandlimited signals  $s^{(1)}[n], s^{(2)}[n], \dots, s^{(D)}[n]$ , Frequency Division Multiplexing is done by multiplying each signal  $s^{(d)}[n]$  by a different complex exponential  $e^{-\frac{2\pi}{N}j(d-1)Kn}$ ,  $d = 0, \dots, D - 1$  and then summing the products to form  $r$ .

$$r[n] = \sum_{d=1}^N s^{(d)}[n] e^{-\frac{2\pi}{N}j(d-1)Kn} \quad (9.5)$$

Each complex exponential  $e^{-\frac{2\pi}{N}j(d-1)Kn}$  shifts the band of  $s^{(d)}[n]$  into a separate, non-overlapping part of  $\hat{r}$ .

To recover  $s^{(d)}[n]$  from  $r$ , multiply  $r$  by  $e^{\frac{2\pi}{N}j(d-1)Kn}$ . This shifts the band  $\hat{s}_K^{(d)}$  back to the original band in the frequency domain. A low pass filter with bandwidth  $K$  is then applied in the frequency domain to recover  $s^{(d)}[n]$ .

$$\widehat{r} = \begin{bmatrix} \widehat{s}_K^{(1)} \\ \widehat{s}_K^{(2)} \\ \vdots \\ \widehat{s}_K^{(D)} \end{bmatrix} \xrightarrow{\text{Multiply } r[n] \text{ by } e^{\frac{2\pi}{N}j(d-1)Kn}} \begin{bmatrix} \widehat{s}_K^{(d)} \\ \widehat{s}_K^{(d+1) \bmod D} \\ \vdots \\ \widehat{s}_K^{(d-1) \bmod D} \end{bmatrix} \xrightarrow{\text{l.p.f in frequency}} \begin{bmatrix} \widehat{s}_K^{(d)} \\ 0_{N-K} \end{bmatrix} = \widehat{s}^{(d)} \quad (9.6)$$

**Remark 9.2.** An alternative to multiplying  $r[n]$  by  $e^{\frac{2\pi}{N}j(d-1)Kn}$  is to multiply by  $e^{-\frac{2\pi}{N}j(N-(d-1)K)n}$ . This will shift the signal  $\widehat{r}[m]$  in the spectral domain  $N - (d-1)K$  times, producing the original signal as in (9.6) due to  $M = A_c$ , the directed cyclic graph.

### 9.2.2 Three Equivalent Interpretations

The multiplication operation of  $e^{-\frac{2\pi}{N}j(d-1)Kn}$  in (9.5) can be interpreted in DSP three different, but equivalent ways.

1. Multiplication of  $r[n]$  by the complex exponential,  $e^{-\frac{2\pi}{N}j(d-1)Kn}$
2. Pointwise multiplication of  $r[n] = [r[0], r[1], \dots, r[N-1]]^T$  by an eigenvector of  $A$   
From (2.10), multiplying  $s^{(d)}[n]$  by  $e^{-\frac{2\pi}{N}j(d-1)Kn}$  is equivalent to  $v_{(d-1)K} \odot s^{(d)}[n]$ .

3. Pointwise multiplication by a power of the vector of eigenvalues of  $A$   
From (2.11), multiplying  $s^{(d)}[n]$  by the complex exponential  $e^{-\frac{2\pi}{N}j(d-1)Kn}$  is equivalent to the pointwise multiplication of  $\lambda^{*(N-(d-1)K)} \odot s^{(d)}[n]$ .

### 9.2.3 Spectral Division Multiplexing - Choosing an Interpretation

In Section 9.2.2, there were three interpretations of frequency division multiplexing, equivalent in DSP, but *not* equivalent in GSP. We explore these three interpretations as potential ways for modulation in GSP.

**Case 1:** This applies directly the DSP modulation process to GSP. We multiply by the

complex exponential,  $e^{-\frac{2\pi}{N}jk_in}$ . From (3.21),

$$P(M) = \text{GFT} \text{diag}(e^{-\frac{2\pi}{N}j^{(d-1)Kn}}) \text{GFT}^{-1}$$

$P(M)$  is only a power of  $M$  (a spectral shift) when

$$\text{GFT} = \text{DFT}$$

This does not generalize well to an arbitrary graph.

**Case 2:** This generalizes modulation in GSP as multiplying by an eigenvector  $v_i$  of  $A$ . From (3.21),

$$P(M) = \text{GFT} \text{diag}(v_i) \text{GFT}^{-1}$$

Consider the  $k$ th column of  $P(M)$ , denoted  $P(M)_k$ .

$$P(M)_k = \text{GFT} \text{diag}(v_i) \text{GFT}_k^{-1}$$

where  $\text{GFT}_k^{-1}$  is the  $k$ th column of the  $\text{GFT}^{-1}$ . The  $k$ th column of the  $\text{GFT}^{-1}$  is the  $k$ th eigenvector of  $A$ ,  $v_k$ .

Thus,

$$P(M)_k = \text{GFT} \text{diag}(v_i) v_k = \text{GFT} (v_i \odot v_k) = \widehat{v}_i * \widehat{v}_k = e_i * e_k = \text{GFT} \text{diag}(v_i) \text{GFT}^{-1} e_k$$

The  $k$ th column of  $P(M)$  is the GFT of the pointwise product of eigenvector  $v_i$  and  $v_k$ . In general, this means that, to form all the columns of  $P(M)$ ,  $v_i$  will be multiplied by every eigenvector of  $A$ . In GSP, the multiplication of any two eigenvectors is not another eigenvector in general. It has no interpretable meaning for an arbitrary graph. Thus, multiplication by an eigenvector does not generalize well to an arbitrary graph.

**Case 3:** We generalize modulation in GSP as multiplying by a power of the vector of

eigenvalues, i.e.,  $\lambda^{*k}$ . From (3.21),

$$P(M) = \text{GFT} \text{diag}(\lambda^{*k}) \text{GFT}^{-1} = \text{GFT} \Lambda^{*k} \text{GFT}^{-1} = M^k$$

Thus,

$$\lambda^{*k} \odot s \xrightarrow{\mathcal{F}} M^k \hat{s}.$$

In this case, we see multiplication by a power of the vector of eigenvalues of  $A$  yields a power of the spectral shift  $M$  in the spectral domain. This generalizes well for an arbitrary graph in GSP.

Thus, we interpret modulation in GSP in case 3, multiplying signals  $s^{(1)}, s^{(2)}, \dots, s^{(D)}$  by  $\lambda^{*(d-1)K}$  and then summing the products to obtain  $r$ .

**GSP Spectral Division Multiplexing (based on DSP (9.5)):**

$$r = \sum_{d=1}^N \left( \lambda^{*(d-1)K} \odot s^{(d)} \right) \quad (9.7)$$

$$\hat{r} = \sum_{d=1}^N M^{(d-1)K} \hat{s}^{(d)} \quad (9.8)$$

In order to be able to perform spectral division multiplexing demodulation,  $\hat{r}$  must contain each band of  $\hat{s}^{(d)}$  with no aliasing between them (similar to how  $\hat{r}$  in (9.6) contains each band  $\hat{s}^{(d)}$  with no overlap). To ensure this, each product

$$M^{(d-1)K} \hat{s}^{(d)}$$

must widen the band of  $\hat{s}^{(d)}$  by  $K$  values more than the previous product's band. In other words, ordering the bandwidth of each product from least to greatest must be the sequence  $K, 2K, \dots, dK$  in order to be able to recover each signal  $s^{(d)}$ .

This is difficult to guarantee for an arbitrary graph, so instead of multiplying by  $\lambda^{*(d-1)K}$ , we multiply each signal by a polynomial of  $\lambda^*$ ,  $q_d(\lambda^*)$ , instead of  $\lambda^{*(d-1)K}$ .

## GSP Spectral Division Multiplexing: (Generalized)

$$r = \sum_{d=1}^N (q_d(\lambda^*) \odot s^{(d)}) \quad (9.9)$$

$$\hat{r} = \sum_{d=1}^N q_d(M) \hat{s}^{(d)} \quad (9.10)$$

### 9.2.4 GSP Spectral Division Multiplexing - Demodulation

This section illustrates demodulation of  $r$  in (9.9).

**Assumption 9.1** (Choice of  $q_d(\lambda^*)$ ). *The polynomial  $q_d(\lambda^*)$  is chosen for each signal  $\hat{s}^{(d)}$  such that the bandwidth of  $q_d(M)\hat{s}^{(d)}$  is increased to  $dK$ . In other words, ordering the bandwidth of each product from least to greatest is the sequence  $K, 2K, \dots, dK$ .*

Assumption 9.1 allows for the demodulation of signals  $s^{(d)}$  from  $r$  in (9.9).

We show demodulation for GSP Spectral Division Multiplexing for two signals  $\hat{s}^{(1)}$  and  $\hat{s}^{(2)}$ . This process can be generalized to any number of signals. Without loss of generality, suppose that  $P_1(M)\hat{s}^{(1)}$  produces a  $K$  bandlimited signal and  $P_2(M)\hat{s}^{(2)}$  produces a  $2K$  bandlimited signal.

$$\hat{r} = q_1(M)\hat{s}^{(1)} + q_2(M)\hat{x}_2 = q_1(M) \begin{bmatrix} \hat{s}_K^{(1)} \\ \mathbf{0} \end{bmatrix} + q_2(M) \begin{bmatrix} \hat{s}_K^{(2)} \\ \mathbf{0} \end{bmatrix} = q_1(M)_K \hat{s}_K^{(1)} + q_2(M)_K \hat{s}_K^{(2)} \quad (9.11)$$

where  $q_i(M)_K$  are the first  $K$  columns of  $q_i(M)$ . We partition the vectors into the first  $K$  rows and the next  $K$  rows (with subscript 1 for the first  $K$  rows and subscript 2 for the next  $K$

rows).<sup>3</sup>

$$\begin{bmatrix} \widehat{r}_{K1} \\ \widehat{r}_{K2} \end{bmatrix} = \begin{bmatrix} q_1(M)_{K1} \widehat{s}_K^{(1)} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} q_2(M)_{K1} \widehat{s}_K^{(2)} \\ q_2(M)_{K2} \widehat{s}_K^{(2)} \end{bmatrix} = \begin{bmatrix} q_1(M)_{K1} \widehat{s}_K^{(1)} + q_2(M)_{K1} \widehat{s}_K^{(2)} \\ q_2(M)_{K2} \widehat{s}_K^{(2)} \end{bmatrix} \quad (9.12)$$

From the bottom  $K$  rows of (9.12), we obtain  $\widehat{r}_{K2} = q_2(M)_{K2} \widehat{s}_K^{(2)}$ . We can solve this for  $\widehat{s}_K^{(2)}$ .

$$\widehat{s}_K^{(2)} = (q_2(M)_{K2})^{-1} \widehat{r}_{K2} \quad (9.13)$$

Knowing  $\widehat{s}_K^{(2)}$  and using the top  $K$  rows of (9.12), we can solve for  $\widehat{s}_K^{(1)}$ .

$$\widehat{s}_K^{(1)} = (q_1(M)_{K1})^{-1} \left( \widehat{r}_{K1} - q_2(M)_{K1} \widehat{s}_K^{(2)} \right) \quad (9.14)$$

Using  $\widehat{s}_K^{(1)}$  and  $\widehat{s}_K^{(2)}$ , we can pad 0s and take the GFT<sup>-1</sup> to recover the original signal,  $\widehat{s}^{(1)}$  and  $\widehat{s}^{(2)}$ .

We make the following assumption:

**Assumption 9.2** (Invertibility of blocks of  $q_i(M)$ ).  $q_i(M)_{Ki}$  is invertible, i.e., the  $K \times K$  block of  $q_i(M)$  formed using the first  $K$  columns of  $q_i(M)$  and the  $i$ th set of  $K$  rows of  $q_i(M)$  is invertible.

This allows for the demodulation of  $\widehat{s}_K^{(1)}$  and  $\widehat{s}_K^{(2)}$  in (9.13) and (9.14).

**Remark 9.3.** Assumptions 9.1 and 9.2 must be satisfied for spectral division multiplexing. Finding a set of  $q_i(M)$  that satisfies the assumptions is specific to the spectral graph  $M$ .

### 9.3 Spectral $z$ -transform Division Multiplexing

In DSP, the spectral  $z$ -transform domain and frequency domain are the same with  $\widehat{s} = q$ . So, in DSP, spectral  $z$ -transform division multiplexing is the same as the frequency division

---

<sup>3</sup>For ease of notation, we write only the first  $2K$  rows. The other values are not written. They can be included, but are not directly used in the calculation.

multiplexing in section 9.2.1.

### 9.3.1 Spectral $z$ -transform Division Multiplexing

Since spectral  $z$ -transform division multiplexing is the same as the frequency division multiplexing in DSP, we use the same equations for spectral  $z$ -transform division multiplexing as we do for spectral division multiplexing ((9.7) and (9.8)).

Instead of assuming the signals  $s^{(d)}$  are  $K$ -bandlimited in the *spectral* domain (shown in (9.1)), we assume the signals are  $K$ -bandlimited in the *spectral  $z$ -transform* domain.

$$q^{(d)} = \begin{bmatrix} q_K^{(d)} \\ 0_{N-K} \end{bmatrix} \quad (9.15)$$

where  $q^{(i)}$  is the spectral  $z$  transform of  $s^{(d)}$ .

In DSP, since  $\hat{s} = q$ , (9.1) and (9.15) are equivalent, both assuming that the signals are  $K$ -bandlimited in the *spectral* domain.

#### GSP Spectral $z$ -transform Division Multiplexing (based on DSP (9.5)):

$$r = \sum_{d=1}^N \left( \lambda^{*(d-1)K} \odot s^{(d)} \right) \quad (9.16)$$

$$q_r = \sum_{d=1}^N C_{\text{comp}}^{(d-1)K} q^{(d)} \quad (9.17)$$

where  $q_r$  is the spectral  $z$ -transform of  $r$ .

From (5.53),  $C_{\text{comp}}$  is a path graph with a boundary condition. Since  $(d-1)K < N$ ,  $d = 1, \dots, D$ , and  $q^{(d)}$  is  $K$ -bandlimited, shifting by  $C_{\text{comp}}^{(d-1)K}$  will shift the signal down the path graph  $(d-1)K$  steps and the boundary condition will not be used.

Thus,

$$q_r = \begin{bmatrix} q_K^{(1)} \\ q_K^{(2)} \\ \vdots \\ q_K^{(D)} \end{bmatrix}$$

This is the same form as  $\hat{r}$  in frequency division multiplexing in DSP in (9.6).

### 9.3.2 Spectral $z$ -Transform Division Multiplexing - Demodulation

In order to recover  $s^{(d)}$  from  $r$ , we use a similar process as demodulation in frequency domain multiplexing in DSP (given by (9.6)). We first bandpass filter in the spectral  $z$ -domain to isolate the desired  $q_K^{(d)}$ . Then, we multiply by  $\lambda^{*(d-1)K}$  in the vertex domain to shift  $q_K^{(d)}$  back to the first  $K$  entries.<sup>4</sup>

$$q_r = \begin{bmatrix} q_K^{(1)} \\ q_K^{(2)} \\ \vdots \\ q_K^{(D)} \end{bmatrix} \xrightarrow{\text{b.p.f in spectral } z\text{-transform}} \begin{bmatrix} 0_K \\ \vdots \\ 0_K \\ q_K^{(d)} \\ 0_K \\ \vdots \\ 0_K \end{bmatrix} \xrightarrow{\text{Multiply by } \lambda^{*(d-1)K} \text{ in vertex domain}} \begin{bmatrix} q_K^{(d)} \\ 0_{N-K} \end{bmatrix} = q^{(d)} \quad (9.18)$$

Taking the inverse spectral  $z$ -transform of  $q^{(d)}$  yields  $s^{(d)}$ .

**Remark 9.4.** Remark 9.2, (9.6), and (9.18) are all valid choices for demodulation of frequency domain multiplexing in DSP. However, only (9.18) works in GSP. This shows that when designing GSP algorithms based on DSP, one must make a choice. Certain DSP methods rely on DSP-specific properties that are not generally true in GSP. For example, remark 9.2 and

---

<sup>4</sup>We assume that the eigenvalues of  $A$  are not 0.

(9.6) both rely on the DSP directed cyclic shift  $A_c$ , which has boundary condition  $A^N = I$ , which is not true in GSP. Choosing these methods will not generalize well to GSP. The method presented in (9.18) works in GSP because it does not rely on DSP-specific properties.

## 9.4 Conclusion

This chapter explores GSP multiplexing by developing GSP modulation and demodulation. DSP multiplexing is done by partitioning signal bandwidth into  $D$  spectral bands with each band containing one of the signals to be combined. In DSP, this partitioning can happen in either the time or frequency domain. We develop GSP multiplexing in three different ways: 1) vertex division multiplexing: by partitioning the vertex domain, 2) spectral division multiplexing: by partitioning the spectral domain, and 3) spectral  $z$ -transform division multiplexing: by partitioning the spectral  $z$ -transform domain. For modulation and demodulation, there are many equivalent interpretations in DSP, obscuring which interpretation to use for GSP modulation. In this chapter, we show which interpretations to use for GSP modulation, applying it to spectral division multiplexing and spectral  $z$ -transform division multiplexing.



# Chapter 10

## Conclusion

### 10.1 Summary of the Thesis

In this thesis, we presented new GSP theory and concepts including:

1. Spectral graph signal processing theory,  $\text{GSP}_{\text{sp}}$ , including spectral shift  $M$ , spectral graph  $G_{\text{sp}}$ , spectral delta functions, and convolution in the spectral domain.
2. New signal representations including the vertex impulsive (p) and spectral impulsive representations (q)
3. The canonical companion model with the canonical companion shift and canonical companion graph.
4. The graph  $z$ -transform and fast graph convolution using the FFT.

We summarize concepts 1-4 by comparing the structure, Fourier transform, filtering, and impulse (delta) functions for GSP, DSP, and the canonical companion model in section [10.1.1](#).

We then applied this theory to develop novel GSP applications:

1. Dual Domain Sampling with interpretations in both the vertex and spectral domains for the four sampling steps: subsampling, decimation, upsampling, interpolation.
2. GSP Uncertainty Principle, providing a way to reduce the Vandermonde system used to calculate  $p$  if the signal  $\hat{s}$  is bandlimited

3. GSP Lagrange Interpolation with all-pass and narrowband Lagrange basis polynomial filters
4. GSP Modulation and Demodulation with vertex division, spectral division, and spectral  $z$ -transform division multiplexing.
5. Expansion of the simple picture of GSP with two domains (vertex and spectral) to four domains (vertex, spectral,  $z$ -transform, and spectral  $z$ -transform), providing an expanded view of DSP.

### 10.1.1 GSP and Canonical Companion Model: An Expanded View of DSP

In this section, we briefly compare GSP, DSP, and the Canonical Companion Model using results from Chapters 2 through 6. We focus on four aspects: structure, Fourier transform, filtering, and impulse (delta) functions.

#### 1) GSP:

**Structure:** In GSP,  $G$  is an arbitrary graph.

**Fourier transform:** The Graph Fourier transform (GFT) is found using the eigendecomposition of  $A$ .  $s$  is written as a linear combination of the eigenvectors of  $A$ ,  $v_i$ , using  $\hat{s}$  as the weights.

**Filtering:** Filters in the vertex domain are polynomials of the adjacency matrix  $A$ :  $p(A)$ .  $p(A)$  is at most degree  $N - 1$  with powers greater than  $N - 1$  being reduced by the Cayley-Hamilton Theorem.

Filters in the spectral domain are polynomials of the spectral shift  $M$ ,  $q(M)$ .

$$M = \text{GFT } \Lambda^* \text{ GFT}^{-1} \tag{10.1}$$

where  $\Lambda^*$  is the complex conjugate of the values in Similar to  $p(A)$ ,  $q(M)$  is at most degree  $N - 1$  with powers greater than  $N - 1$  being reduced by the Cayley Hamilton Theorem.

**Impulse (Delta) Functions:** In GSP, the vertex domain graph impulse,  $\delta_0$  is wide and flat (all 1s) in the spectral domain.

A vertex domain signal  $s$  can be written as a filtering of  $\delta_0$  by a polynomial of  $A$ .

$$s = p(A)\delta_0 = \sum_{i=0}^{N-1} p_i A^i \delta_0 \quad (10.2)$$

Assumption 2.2 guarantees that every signal  $s$  can be written in this form.

Similarly, the spectral domain graph impulse  $\widehat{\delta}_{\text{sp},0}$  is wide and flat (all 1s) in the vertex domain.

$$\delta_{\text{sp},0} = \frac{1}{\sqrt{N}} \mathbf{1} \xrightarrow{\mathcal{F}} \widehat{\delta}_{\text{sp},0} = \text{GFT} \left( \frac{1}{\sqrt{N}} \mathbf{1} \right) \quad (10.3)$$

A spectral domain signal  $\widehat{s}$  can be written as a filtering of  $\widehat{\delta}_{\text{sp},0}$  by a polynomial of  $M$ .

$$\widehat{s} = q(M)\widehat{\delta}_{\text{sp},0} = \sum_{i=0}^{N-1} q_i M^i \widehat{\delta}_{\text{sp},0} \quad (10.4)$$

Assumption 2.2 guarantees that every signal  $\widehat{s}$  can be written in this form.

In GSP, the vertex domain graph impulse is not generally narrow or impulsive in the vertex domain. Similarly, the spectral domain graph impulse is not generally narrow or impulsive in the spectral domain.

## 2) GSP Consistency with DSP:

GSP theory as we present is consistent with existing DSP theory, reducing to DSP when a directed cycle graph is used. When restricting the underlying graph  $G$  to the directed cycle graph,  $A_c$ , we recover DSP from GSP. Doing this, we also show what holds in DSP that does not hold in GSP. For example:

**Structure:** The GSP generic graph  $G$  is, in DSP, the  $N$  node directed cycle graph with adjacency matrix  $A_c$ , shown in Figure 2.1. The cycle graph represents periodic time values with each node representing a time value at  $0, \dots, N - 1$ .

**Fourier transform:** Similar to GSP, the DFT is also found through the eigendecompo-

sition of  $A_c$ . The DFT is unitary ( $\text{DFT}^{-1} = \text{DFT}^H$ ). Like in GSP, the  $\text{DFT}^H$  is the matrix whose columns are the eigenvectors of  $A_c$  (see (2.9)). However, unlike GSP, (2.9) and (2.10) also show the DFT and its inverse are both Vandermonde matrices. So, the columns of  $\text{DFT}^H$  have two equivalent interpretations (shown in (2.10)): either eigenvectors of  $A_c$ ,  $v_i$  or powers of the conjugate of the eigenvalues  $\lambda^{*i}$ . In GSP, the columns can only be interpreted as the eigenvectors, not vectors of powers of eigenvalues.

**Spectral Shift  $M$ :** The spectral shift  $M$  in GSP is  $M = A$  in DSP. So, in DSP, linear shift invariant filters in the time and frequency domain are both polynomials of the adjacency matrix  $A$ ,  $p(A)$  and  $q(A)$  respectively. In GSP, since  $A \neq M$ , this is not the same and we have  $p(A)$  for vertex domain filtering and  $q(M)$  for spectral domain filtering.

**Impulse (Delta) Functions:** In DSP, the time domain and frequency domain impulses can be defined the same way as in GSP. Similar to GSP, the Fourier transform of the time domain impulse,  $\widehat{\delta}_0$  is wide and flat in the frequency domain. However, unlike GSP, the time domain impulse,  $\delta_0$  is narrow and impulsive.

$$\delta_0 = e_0, \quad \xrightarrow{\mathcal{F}} \quad \widehat{\delta}_0 = \frac{1}{\sqrt{N}}\mathbf{1}, \quad (10.5)$$

where  $e_0$  is the standard unit vector with 1 in the first entry and  $\mathbf{1}$  is the vector of ones. The frequency domain impulse is wide and flat in the time domain (like in GSP), but it is narrow and impulsive in the frequency domain (unlike GSP).

$$\delta_{\text{sp},0} = \frac{1}{\sqrt{N}}\mathbf{1} \quad \xrightarrow{\mathcal{F}} \quad \widehat{\delta}_{\text{sp},0} = e_0 \quad (10.6)$$

A time domain signal  $s$  can be written as

$$s = p(A)\delta_0$$

in DSP. Since

$$\delta_0 = e_0$$

in DSP, the coefficients of  $p(A)$  are equal to  $s$ ,

$$p(A) = s_{N-1}A^{N-1} + \dots + s_1A + s_0I$$

where  $s_i$  is the  $i$ th entry of  $s$ .

Similarly, a frequency domain signal  $\hat{s}$  can be written as

$$\hat{s} = q(A)\delta_{\text{sp},0}$$

in DSP. Since

$$\hat{\delta}_{\text{sp},0} = e_0$$

in DSP, the coefficients of  $q(A)$  are equal to  $\hat{s}$ ,

$$p(A) = \hat{s}_{N-1}A^{N-1} + \dots + \hat{s}_1A + \hat{s}_0I$$

where  $\hat{s}_i$  is the  $i$ th entry of  $\hat{s}$ .

### 3) Canonical Companion Model:

The companion canonical model has properties that are hold in DSP, but not generally true in GSP. We discuss the structure, Fourier transform, filtering and impulse (delta) functions for the companion canonical model.

**Structure:** The graph  $G$  for the companion signal model is the  $N$  node path graph with a boundary condition. The adjacency matrix is a companion matrix  $C_{\text{comp}}$ . This replicates exactly the structure of the DSP (time) graph, this with the appropriate (time) cycle boundary condition that holds in DSP. In fact, the cycle graph in DSP,  $A_c$ , is a specific case of  $C_{\text{comp}}$  where  $c_0 = -1$  and  $c_{i \neq 0} = 0$ . The companion matrix represents the shift  $z^{-1}$  with arbitrary boundary conditions in Algebraic Signal Processing [6, 7].

**Fourier Transform:**  $p$  is the vector of coefficients of  $p(A)$ ,  $p = [p_0, \dots, p_{N-1}]^T$ .  $p$  is related to  $\hat{s}$  by a Vandermonde matrix.

$$\hat{s} = \frac{1}{\sqrt{N}} \mathcal{V} p \quad (10.7)$$

where  $\mathcal{V}$  is the Vandermonde matrix produced using the eigenvalues of  $A$  and  $\lambda$ .

By assumption 2.2,  $\mathcal{V}$  is invertible. In DSP,  $\mathcal{V} = \text{DFT}$  and  $s = p$ .

Similarly,  $q$  is the vector of coefficients of  $q(M)$ ,  $q = [q_0, \dots, q_{N-1}]^T$ .

$$s = \mathcal{V}^* q \quad (10.8)$$

where  $\mathcal{V}$  is the Vandermonde from (5.12). In DSP,  $\mathcal{V}^* = \text{DFT}^{-1}$  and  $\hat{s} = q$ .

In DSP, the  $\text{DFT}^{-1}$  is both the matrix whose columns are eigenvectors and a Vandermonde matrix. In GSP, the  $\text{GFT}^{-1}$  is a matrix whose columns are eigenvectors, but it is not a Vandermonde matrix. Here,  $\hat{s}$  and  $p$  (and  $s$  and  $q$ ) are related by a Vandermonde matrix, which is not directly related to the eigenvectors. So, the companion signal model captures the Vandermonde properties in DSP, while GSP captures the eigenvector properties in DSP.

**Filtering:** In the companion signal model, filtering in the vertex domain  $p(A)$  in GSP becomes  $p(C_{\text{comp}})$ . Similarly, filtering in the spectral domain  $q(M)$  in GSP becomes  $q(C_{\text{comp}})$ . In GSP, we need two shifts,  $A$  and  $M$ , to filter in the vertex and spectral domain. In DSP, we only need one shift  $A$ , since  $A = M$  to filter in the time and frequency domain. With the companion signal model, like DSP, we only need one shift  $C_{\text{comp}}$  to filter in both domains.

**Impulse (Delta) Functions:** Let

$$\hat{s} = \hat{\delta}_0 = \frac{1}{\sqrt{N}} \mathbf{1}.$$

Since the first column of  $\mathcal{V}$  is all 1s,  $p = e_0$ . Similarly, let

$$s = \delta_{\text{sp},0} = \frac{1}{\sqrt{N}} \mathbf{1}.$$

Since the first column of  $\mathcal{V}^*$  is all 1s,  $q = e_0$ .

Property	GSP	DSP	CSM
Structure	$A$	$A_c$	$C_{\text{comp}}$
FT columns	$v_i$	both $v_i, \lambda^k$	$\lambda^k$
Filtering (Vertex)	$p(A)$	$p(A_c)$	$p(C_{\text{comp}})$
Filtering (Spectral)	$q(M)$	$q(A_c)$	$q(C_{\text{comp}})$
$\delta_0$	$\text{GFT}^{-1} \left( \frac{1}{\sqrt{N}} \mathbf{1} \right)$	$e_0$	$e_0$
$\widehat{\delta}_0$	$\frac{1}{\sqrt{N}} \mathbf{1}$	$\frac{1}{\sqrt{N}} \mathbf{1}$	$\frac{1}{\sqrt{N}} \mathbf{1}$
$\delta_{\text{sp},0}$	$\frac{1}{\sqrt{N}} \mathbf{1}$	$\frac{1}{\sqrt{N}} \mathbf{1}$	$\frac{1}{\sqrt{N}} \mathbf{1}$
$\widehat{\delta}_{\text{sp},0}$	$\text{GFT} \left( \frac{1}{\sqrt{N}} \mathbf{1} \right)$	$e_0$	$e_0$

Table 10.1: Comparison of the properties in GSP, DSP, and the Canonical Signal Model (CSM)

The relationship between  $\widehat{s} = \widehat{\delta}_0$  and  $p = e_0$  is the same as DSP. Similarly, the relationship between  $s = \delta_{\text{sp},0}$  and  $q = e_0$  is also the same as DSP.

This shows that with the companion model, GSP impulse (delta) functions in one domain are flat and wide in the other domain (like in DSP). They are only narrow and impulsive as a consequence of the Vandermonde matrix. This is why we see the narrow and impulsive delta in DSP and the canonical signal model, but not in GSP (which uses the eigenvectors instead of the Vandermonde matrix).

#### 4) Summary:

In this section, we explored the structure, Fourier transform, filtering and delta functions in GSP, DSP and the canonical signal model. GSP uses a GFT that is based on the eigenvectors of  $A$ . The canonical signal model uses a Vandermonde matrix that uses the eigenvalues of  $A$ . The DFT is both a Vandermonde matrix of eigenvalues and contains the eigenvectors of  $A$ . As a result, certain properties taken for granted in DSP are true in either GSP or the canonical signal model, but not both. Table 10.1 summarizes this section and the properties.

## 10.2 Contributions of the Thesis

We summarize the contributions of the thesis:

1. **The expanded GSP picture:  $\text{GSP}_{\text{sp}}$ , GSP Canonical Model, Signal Representations, and  $z$ -transforms:** In Chapter 3, 5, 6, we expanded the current GSP theory by introducing and discussing  $\text{GSP}_{\text{sp}}$ , the GSP Canonical Model, new signal representations, and  $z$ -transforms. This provides the full GSP picture, allowing for interpretations of operations in both the vertex and spectral domains, as well as the graph  $z$ -transform and spectral  $z$ -transform domains.

We summarize these in figure 10.1 that illustrates the corresponding signal domains and

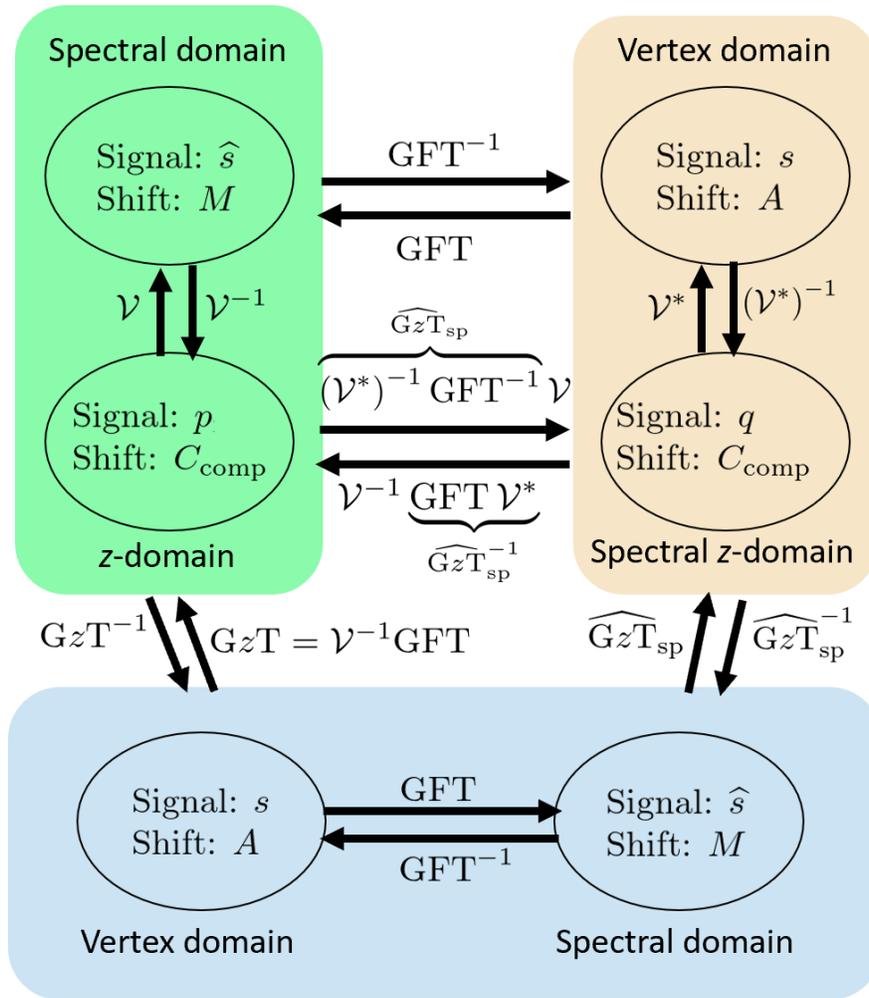


Figure 10.1: Graph signal domains and the transformations between them. For each domain, both the signal and shift are given.

the transforms relating them, summarizing the main results from these chapters. At the bottom, we have the standard Euclidean vertex domain signals  $s$  with its shift  $A$

and the spectral domain signals  $\hat{s}$  with its shift  $M$ . The relation between these two domains is the GFT and its inverse  $\text{GFT}^{-1}$ . At the intermediate level, we have the two  $z$ -transform domains corresponding to the two impulsive representations, the vertex impulsive  $z$ -transformed signals  $p$  and the spectral impulsive  $z$ -transformed signals  $q$ . The graph  $z$ -transform  $\text{GzT}$  obtains  $p$  from  $s$ , while  $\widehat{\text{GzT}}_{\text{sp}}$  obtains  $q$  from  $\hat{s}$ . These two  $z$ -transformed signal domains reflect a number of interesting and surprising facts. Their shift is the same, the companion matrix  $C_{\text{comp}}$ , to which we associate a “companion graph”  $G_{\text{comp}}$ , see figure 5.1. In these domains, the graph eigenvalues  $\{\lambda_n\}_{0 \leq n \leq N-1}$  contain all needed information, since the eigenvectors derive from the graph frequency vector  $\lambda$  and its powers. At the top, we indicate that the Vandermonde matrix  $\mathcal{V}$  and its conjugate relate the  $z$ -transformed signal domains back to the spectral and vertex domains.

With DSP, the picture is much simpler. Although not usually presented this way [52, 80], by reinterpreting the above GSP representations in DSP, we cast four common DSP signal representations as vector coordinatizations of the signal  $s \in \mathbb{C}^N$  with respect to choices of basis  $B$  in  $\mathbb{C}^N$ .

- (a) Standard:  $B_E = \{e_0, e_1, \dots, e_{N-1}\}$  is the standard or Euclidean basis and the signal representation is the vector of signal samples  $s = [e_0, e_1, \dots, e_{N-1}]_{B_E} = I_N s_{B_E} = s_{B_E}$ .
- (b) Impulsive:  $B_{\text{imp}} = \{\delta_0, \delta_1, \dots, \delta_{N-1}\}$  is the basis of the impulse and its delayed replicas. Since in this case,  $D_{\text{imp}} = I_N$ , the signal representation is  $s = [\delta_0, \delta_1, \dots, \delta_{N-1}] p = I_N p$ , and  $p = s$ .
- (c) Spectral:  $B_{\text{Fourier}} = \{v_0, v_1, \dots, v_{N-1}\}$  is the basis of the eigenmodes or harmonics and the signal representation is the Fourier transform of the signal  $s = [v_0, v_1, \dots, v_{N-1}] \hat{s} = \text{DFT}^H \hat{s}$ .
- (d) Spectral impulsive:  $B_{\text{sp,imp}} = \frac{1}{\sqrt{N}} \{\lambda^{*0}, \lambda^{*1}, \dots, \lambda^{*(N-1)}\}$ . In this case,  $D_{\text{sp,imp}} = \text{DFT}^H$ ,

and the signal representation is  $s = \frac{1}{\sqrt{N}} [\lambda^{*0}, \lambda^{*1}, \dots, \lambda^{*N-1}] q = DFT^H q$ , and  $q = \hat{s}$ .

In DSP, the above four signal representations reduce to two distinct ones, see figure 10.2 that illustrates this for a  $N = 4$  signal. Since  $\delta_n = e_n, n = 0, \dots, N - 1$ , the standard

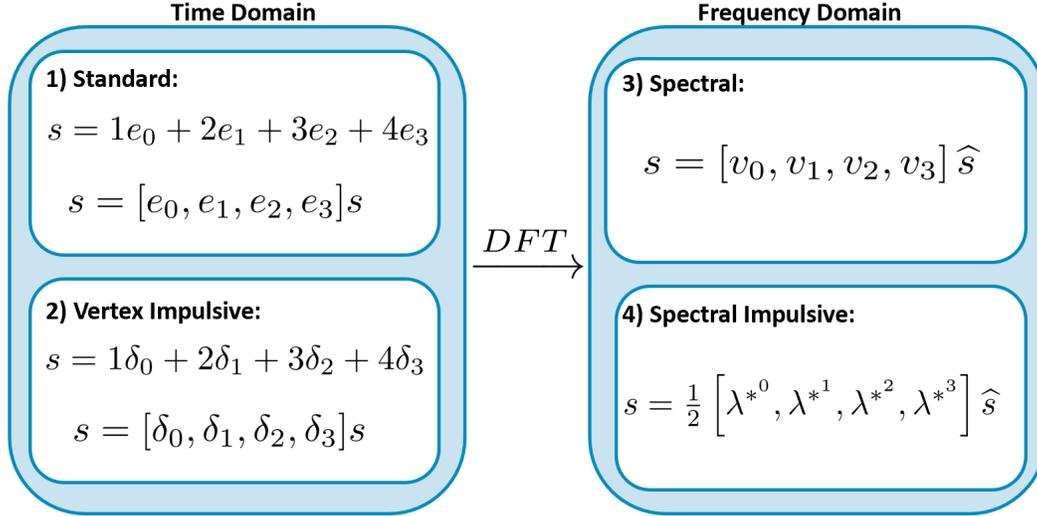


Figure 10.2: DSP Signal Representations:  $s = [1, 2, 3, 4]^T$ ,  $\hat{s} = [5, -1 + j, -1, -1 - j]^T$ . The standard representation and impulsive representations coincide. The Fourier and spectral impulsive representations also coincide.

and impulsive bases and corresponding signal representations coincide,  $B_E = B_{\text{imp}}$  and  $s_E = p = s$ . Similarly, since  $v_n = \frac{1}{\sqrt{N}} \lambda^{*n}$ , the Fourier and spectral impulsive bases and corresponding signal representations coincide,  $B_{\text{Fourier}} = B_{\text{sp,imp}}$  and  $\hat{s} = q = DFT s$ .

So, in DSP, the standard and impulsive representations can be used interchangeably as the time domain signal,  $s$ , and, similarly, the eigenvalue and spectral representations can be used interchangeably as the frequency domain signal,  $\hat{s}$ . Also, in DSP,  $A = M = C_{\text{comp}}$  [54, 71]. In DSP, the directed cyclic shift is already a companion matrix. Thus, the canonical companion model in DSP reduces to DSP. In contrast, GSP does not start with a companion matrix for the graph, so we have differences between GSP and the canonical companion model.

Having this in mind, figure 10.1 is much simpler with DSP as illustrated in figure 10.3.

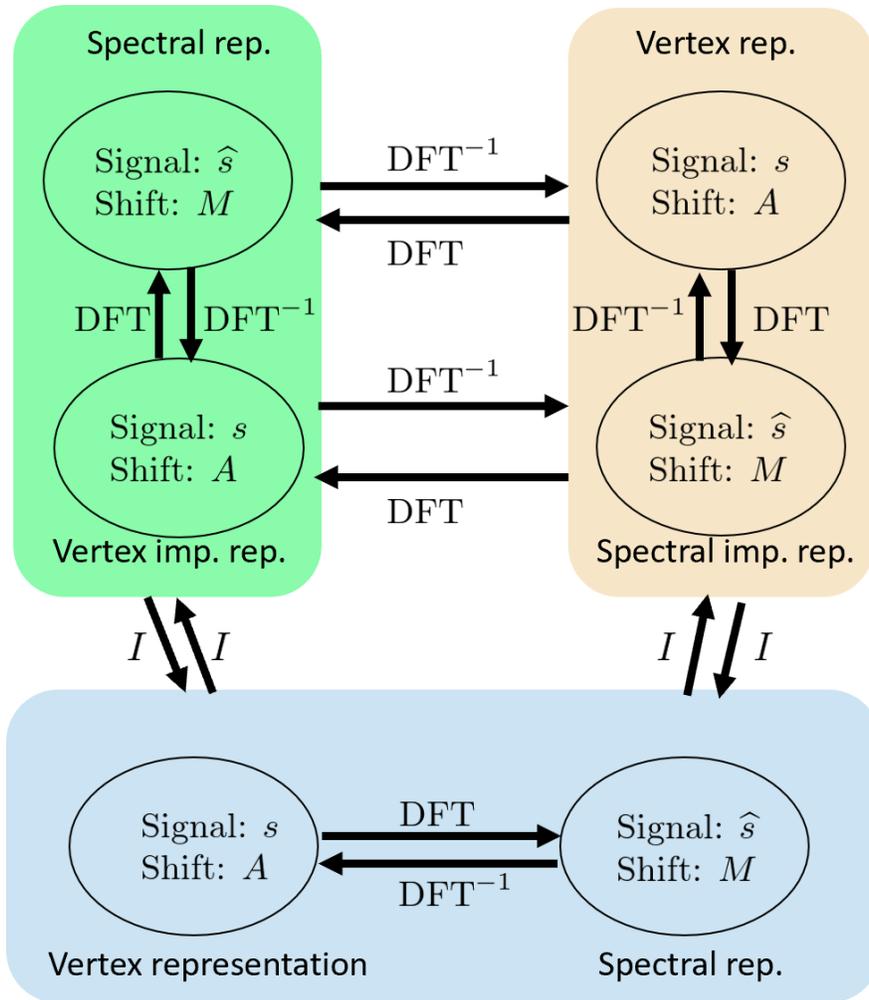


Figure 10.3: Figure 10.1 for DSP. The three colored regions all have the same signals and shift. In DSP,  $A = M$ . The three colored regions are all identical to each other in DSP.

2. **Two Distinct GSP Models:** The companion GSP model and  $\text{GSP}_{\text{sp}}$  show that there are two distinct models in GSP: the eigenvector model from current GSP literature and the canonical model that we introduce that only uses the eigenvalues. In DSP, (pointwise) powers of a vector of the eigenvalues and the eigenvectors are the same, so, these two models overlap and are equivalent, obscuring which model should be used in GSP for particular data processing tasks. Many DSP concepts can be explained in either model, using either the powers of the vector of eigenvalues or the eigenvectors, which can be used interchangeably in DSP. This does not hold in GSP, where the eigenvectors are not powers of the vector of eigenvalues.

The canonical graph signal model has nice properties: 1) new concepts are natural, intuitive extensions of existing DSP concepts, 2) it is *consistent* with existing DSP theory when applied to a cycle graph ( $s = p$  and  $\hat{s} = q$  in DSP), 3) it reveals many choices and assumptions taken for granted in DSP, and 4) since  $s$  and  $\hat{s}$  are related to  $p$  and  $q$  in GSP, results in the canonical graph signal model can lead to new results with  $s$  and  $\hat{s}$  in GSP.

We illustrate the significance of this dichotomy in GSP by presenting a GSP uncertainty principle, interpolating GSP filters, and GSP modulation as natural applications of the canonical companion signal model. We show that, while equivalent in DSP, both models are essential for the complete picture in GSP.

3. **Dualized operations in GSP for  $\text{GSP}_{\text{sp}}$ :** In this thesis, we dualized GSP operations to the spectral domain. Instead of starting with a vertex domain signal  $s$  and shift  $A$ , we start with a spectral domain signal  $\hat{s}$  and spectral shift  $M$ . From there, we develop  $\text{GSP}_{\text{sp}}$ . We dualized everything in this thesis to  $\text{GSP}_{\text{sp}}$ : spectral graph shift, spectral graph, spectral delta functions, spectral convolution, spectral impulsive signal representations, sampling in the spectral domain, spectral  $z$ -transform, spectral Uncertainty Principle, spectral Lagrange interpolating polynomial filters, and spectral domain multiplexing.
4. **Dualized GSP Sampling:** Using  $\text{GSP}_{\text{sp}}$ , we provide interpretations for the four standard sampling steps: subsampling, decimation, upsampling, interpolation, in both the vertex and spectral domains.
5. **Graph  $z$ -Transform (GzT) and Fast Graph Convolution:** We present the graph  $z$ -transform and its dual, the spectral graph  $z$ -transform. The graph  $z$ -transform provides a symbolic polynomial representation for graph signals and reproduces many DSP characteristics. We present an algorithm for fast vertex domain convolution using the FFT of graph  $z$ -transform signals.
6. **GSP Uncertainty Principle:** We present a GSP Uncertainty Principle relating the  $z$ -

transform  $p$  and spectral signal  $\hat{s}$ . We also present its dual, replacing  $p$  with the spectral  $z$ -transform  $q$  and  $\hat{s}$  with vertex signal  $s$ . Using these, we can reduce the Vandermonde system if  $\hat{s}$  is bandlimited (or  $s$  has reduced support).

7. **Lagrange Interpolation in GSP:** We present Lagrange basis polynomial filters and their properties. We find that the filters individually are narrow-band GSP filters, and the sum of the filters is an all-pass GSP filter.
8. **GSP Modulation and Demodulation:** We explore three interpretations of GSP modulation, equivalent in DSP. Using modulation, we present three methods for GSP multiplexing: vertex division, spectral division, and spectral  $z$ -transform multiplexing. These three methods differ by what domain they partition the signal bandwidth. Vertex division multiplexing partitions the vertex domain. Spectral division multiplexing partitions the spectral domain. Spectral  $z$ -transform multiplexing partitions the spectral  $z$ -transform domain.

### 10.3 Future Directions

In the future, we want to explore applications of the graph  $z$ -transform and canonical model on real world datasets. We also want to use dual domain sampling, GSP uncertainty principle, and GSP modulation on real world applications and problems. Using the theory developed in this thesis, we want to extend other important DSP concepts such as filter design, short term Fourier transform, and source detection to GSP and  $\text{GSP}_{\text{sp}}$ .

We want to explore relationships between this theory and graph convolutional neural networks (GCNNs). We want to be able to interpret GCNN operations in GSP. A first step is [64]. Also, we want to design new algorithms for GCNNs based on this GSP theory. For example, some preliminary work [61, 63] involves replacing the GCNN vertex domain convolution using  $A$  with spectral domain convolution using spectral shift  $M$ .

For spectral division multiplexing, we want to explore algorithms for finding  $q_d(\lambda^*)$  that

satisfy assumption 9.1, allowing for the demodulation of signals in GSP spectral division multiplexing. Lastly, since the graph  $z$ -transform  $p$  and  $\hat{s}$  (similarly,  $q$  and  $s$ ) are related by the Vandermonde matrix, we want to explore fast stable methods of solving the Vandermonde system for large graphs.

# Bibliography

- [1] A. Sandryhaila and J. M. F. Moura, “Discrete signal processing on graphs,” *IEEE Trans. Signal Proc.*, vol. 61, pp. 1644–1656, April 2013.
- [2] A. Sandryhaila and J. M. F. Moura, “Discrete signal processing on graphs: Frequency analysis,” *IEEE Trans. Signal Proc.*, vol. 62, pp. 3042–3054, June 2014.
- [3] A. Sandryhaila and J. M. F. Moura, “Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure,” *IEEE Signal Processing Magazine*, vol. 31, pp. 80–90, September 2014.
- [4] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” *IEEE Signal Proc. Magazine*, vol. 30, pp. 83–98, May 2013.
- [5] M. Püschel and J. M. F. Moura, “The algebraic approach to the discrete cosine and sine transforms and their fast algorithms,” *SIAM J. Comp.*, vol. 32, no. 5, pp. 1280–1316, 2003.
- [6] M. Püschel and J. M. F. Moura, “Algebraic signal processing theory.” 67 pages., December 2006.
- [7] M. Püschel and J. M. F. Moura, “Algebraic signal processing theory: Foundation and 1-D time,” *IEEE Trans. Signal Proc.*, vol. 56, no. 8, pp. 3572–3585, 2008.
- [8] M. Püschel and J. M. F. Moura, “Algebraic signal processing theory: 1-D space,” *IEEE Trans. Signal Proc.*, vol. 56, no. 8, pp. 3586–3599, 2008.

- [9] M. Püschel and J. M. F. Moura, “Algebraic signal processing theory: Cooley-Tukey type algorithms for DCTs and DSTs,” *IEEE Trans. Signal Proc.*, vol. 56, no. 4, pp. 1502–1521, 2008.
- [10] F. R. K. Chung, *Spectral Graph Theory*. AMS, 1996.
- [11] M. Belkin and P. Niyogi, “Using manifold structure for partially labeled classification,” in *Neural Information Processing Symposium (NIPS)*, 2002.
- [12] R. R. Coifman, S. Lafon, A. Lee, M. Maggioni, B. Nadler, F. J. Warner, and S. W. Zucker, “Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps,” *Proc. Nat. Acad. Sci.*, vol. 102, no. 21, pp. 7426–7431, 2005.
- [13] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden, “Distributed regression: an efficient framework for modeling sensor network data,” in *IPSN*, pp. 1–10, 2004.
- [14] R. Wagner, H. Choi, R. G. Baraniuk, and V. Delouille, “Distributed wavelet transform for irregular sensor network grids,” in *IEEE SSP Workshop*, pp. 1196–1201, 2005.
- [15] D. K. Hammond, P. Vandergheynst, and R. Gribonval, “Wavelets on graphs via spectral graph theory,” *J. Appl. Comp. Harm. Anal.*, vol. 30, no. 2, pp. 129–150, 2011.
- [16] S. K. Narang and A. Ortega, “Local two-channel critically sampled filter-banks on graphs,” in *ICIP*, pp. 333–336, 2010.
- [17] S. K. Narang and A. Ortega, “Downsampling graphs using spectral theory,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4208–4211, 2011.
- [18] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, “Graph signal processing: Overview, challenges, and applications,” *Proceedings of the IEEE*, vol. 106, pp. 808–828, May 2018.

- [19] I. Pesenson, “Sampling in Paley-Wiener spaces on combinatorial graphs,” *Transactions of the American Mathematical Society*, vol. 360, no. 10, pp. 5603–5627, 2008.
- [20] I. Z. Pesenson and M. Z. Pesenson, “Sampling, filtering and sparse approximations on combinatorial graphs,” *Journal of Fourier Analysis and Applications*, vol. 16, no. 6, pp. 921–942, 2010.
- [21] S. K. Narang and A. Ortega, “Perfect reconstruction two-channel wavelet filter banks for graph structured data,” *IEEE Trans. Signal Proc.*, vol. 60, no. 6, pp. 2786–2799, 2012.
- [22] A. Anis, A. Gadde, and A. Ortega, “Towards a sampling theorem for signals on arbitrary graphs,” in *2014 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3864–3868, IEEE, 2014.
- [23] A. Gadde and A. Ortega, “A probabilistic interpretation of sampling theory of graph signals,” in *2015 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3257–3261, IEEE, 2015.
- [24] A. Anis, A. Gadde, and A. Ortega, “Efficient sampling set selection for bandlimited graph signals using graph spectral proxies,” *IEEE Trans. Signal Proc.*, vol. 64, no. 14, pp. 3775–3789, 2016.
- [25] A. Anis and A. Ortega, “Critical sampling for wavelet filterbanks on arbitrary graphs,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3889–3893, IEEE, 2017.
- [26] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, “Discrete signal processing on graphs: Sampling theory,” *IEEE Trans. Signal Proc.*, vol. 63, no. 24, pp. 6510 – 6523, 2015.
- [27] S. Chen, R. Varma, A. Singh, and J. Kovacević, “Signal recovery on graphs: Fundamental limits of sampling strategies,” *IEEE Tr. on Sig. and Inform. Proc. over Networks*, vol. 2, no. 4, pp. 539–554, 2016.

- [28] N. Tremblay, P.-O. Amblard, and S. Barthelmé, “Graph sampling with determinantal processes,” in *2017 25th European Signal Processing Conference (EUSIPCO)*, pp. 1674–1678, IEEE, 2017.
- [29] M. Tsitsvero, S. Barbarossa, and P. Di Lorenzo, “Signals on graphs: Uncertainty principle and sampling,” *IEEE Transactions on Signal Processing*, vol. 64, no. 18, pp. 4845–4860, 2016.
- [30] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, “Sampling of graph signals with successive local aggregations,” *IEEE Transactions on Signal Processing*, vol. 64, no. 7, pp. 1832–1843, 2015.
- [31] A. Jung, A. O. Hero, III, A. C. Mara, S. Jahromi, A. Heimowitz, and Y. C. Eldar, “Semi-supervised learning in network-structured data via total variation minimization,” *IEEE Transactions on Signal Processing*, vol. 67, no. 24, pp. 6256–6269, 2019.
- [32] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovačević, “Signal recovery on graphs: Variation minimization,” *IEEE Transactions on Signal Processing*, vol. 63, no. 17, pp. 4609–4624, 2015.
- [33] Y. C. Eldar, *Sampling Theory: Beyond Bandlimited Systems*. USA: Cambridge University Press, 1st ed., 2015.
- [34] Y. Tanaka and Y. Eldar, “Generalized sampling on graphs with subspace and smoothness priors,” *IEEE Tr. on Signal Proc.*, vol. 68, 2020.
- [35] Y. Tanaka, “Spectral domain sampling of graph signals,” *IEEE Transactions on Signal Processing*, vol. 66, no. 14, pp. 3752–3767, 2018.
- [36] B. Girault, P. Gonçalves, and É. Fleury, “Translation on graphs: An isometric shift operator,” *IEEE Signal Processing Letters*, vol. 22, pp. 2416–2420, Dec 2015.

- [37] A. Gavili and X. P. Zhang, “On the shift operator, graph frequency, and optimal filtering in graph signal processing,” *IEEE Transactions on Signal Processing*, vol. 65, pp. 6303–6318, Dec 2017.
- [38] X. Zhu and M. Rabbat, “Approximating signals supported on graphs,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3921–3924, 2012.
- [39] L. F. Chamon and A. Ribeiro, “Greedy sampling of graph signals,” *IEEE Transactions on Signal Processing*, vol. 66, no. 1, pp. 34–47, 2017.
- [40] Y. Tanaka, Y. C. Eldar, A. Ortega, and G. Cheung, “Sampling signals on graphs: From theory to applications,” *IEEE Signal Processing Magazine*, vol. 37, no. 6, pp. 14–30, 2020.
- [41] O. Teke and P. P. Vaidyanathan, “Extending classical multirate signal processing theory to graphs—part I fundamentals,” *IEEE Transactions on Signal Processing*, vol. 65, no. 2, pp. 409–422, 2017.
- [42] O. Teke and P. P. Vaidyanathan, “Extending classical multirate signal processing theory to graphs—part II: M-channel filter banks,” *IEEE Transactions on Signal Processing*, vol. 65, no. 2, pp. 423–437, 2017.
- [43] A. Agaskar and Y. M. Lu, “A spectral graph uncertainty principle,” *IEEE Transactions on Information Theory*, vol. 59, no. 7, pp. 4338–4356, 2013.
- [44] B. Padeloup, V. Gripon, G. Mercier, D. Pastor, and M. G. Rabbat, “Characterization and inference of graph diffusion processes from observations of stationary signals,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 3, pp. 481–496, 2018.
- [45] S. Segarra, A. G. Marques, G. Leus, and A. Ribeiro, “Interpolation of graph signals using shift-invariant graph filters,” in *2015 23rd European Signal Processing Conference (EUSIPCO)*, pp. 210–214, IEEE, 2015.

- [46] S. Segarra, A. G. Marques, G. Leus, and A. Ribeiro, “Reconstruction of graph signals through percolation from seeding nodes,” *IEEE Transactions on Signal Processing*, vol. 64, no. 16, pp. 4363–4378, 2016.
- [47] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, “Stationary graph processes and spectral estimation,” *IEEE Transactions on Signal Processing*, vol. 65, no. 22, pp. 5911–5926, 2017.
- [48] J. Mei and J. M. F. Moura, “Signal processing on graphs: Causal modeling of unstructured data,” *IEEE Transactions on Signal Processing*, vol. 65, no. 8, pp. 2077–2092, 2017.
- [49] J. Mei and J. M. F. Moura, “Silvar: Single index latent variable models,” *IEEE Transactions on Signal Processing*, vol. 66, no. 11, pp. 2790–2803, 2018.
- [50] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, “Learning graphs from data: A signal representation perspective,” *IEEE Signal Processing Magazine*, vol. 36, no. 3, pp. 44–63, 2019.
- [51] J. A. Deri and J. M. F. Moura, “Spectral projector-based graph Fourier transforms,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, pp. 785–795, Sept 2017.
- [52] A. V. Oppenheim and A. S. Willsky, *Signals and Systems*. Englewood Cliffs, New Jersey: Prentice-Hall, 1983.
- [53] D. Donoho and P. Stack, “Uncertainty principles and signal recovery,” *SIAM J. Appl Math*, vol. 49, no. 3, pp. 906–931, 1989.
- [54] J. Shi and J. M. F. Moura, “Topics in graph signal processing: Convolution and modulation,” in *(ACSSC) Asilomar Conference on Signals, Systems, and Computers*, IEEE, 2019.

- [55] J. Shi and J. M. F. Moura, “Graph signal processing: Dualizing gsp sampling in the vertex and spectral domains,” *IEEE Transactions on Signal Processing*, vol. 70, pp. 2883–2898, 2022.
- [56] J. Shi and J. M. F. Moura, “The companion model—a canonical model in graph signal processing,” March 2021.
- [57] J. Du, J. Shi, S. Kar, and J. M. F. Moura, “On graph convolution for graph cnns,” in *IEEE Data Science Workshop*, IEEE, 2018.
- [58] J. Shi, M. Cheung, J. Du, and J. M. F. Moura, “Classification with vertex-based graph convolutional neural networks,” in *(ACSSC) Asilomar Conference on Signals, Systems, and Computers*, IEEE, 2018.
- [59] M. Cheung, J. Shi, O. Wright, Y. Jiang, and J. M. F. Moura, “Pooling in graph convolutional neural networks,” in *4th GSP Workshop*, IEEE, 2019.
- [60] M. Cheung, J. Shi, O. Wright, Y. Jiang, and J. M. F. Moura, “Pooling in graph convolutional neural networks,” in *(ACSSC) Asilomar Conference on Signals, Systems, and Computers*, IEEE, 2019.
- [61] J. Shi, W. Summer, and J. M. F. Moura, “A dual approach to graph cnns,” in *(ACSSC) Asilomar Conference on Signals, Systems, and Computers*, IEEE, 2020.
- [62] Y. Jiang, J. Shi, M. Cheung, O. Wright, and J. M. F. Moura, “Evaluating effectiveness of graph structures,” in *(ACSSC) Asilomar Conference on Signals, Systems, and Computers*, IEEE, 2020.
- [63] A. Lin, W. Summer, J. Shi, M. Cheung, and J. M. F. Moura, “Using sparse spectral shifts in graph cnns,” in *(ACSSC) Asilomar Conference on Signals, Systems, and Computers*, IEEE, 2021.

- [64] M. Cheung, J. Shi, O. Wright, L. Y. Jiang, X. Liu, and J. M. F. Moura, “Graph signal processing and deep learning: Convolution, pooling, and topology,” *IEEE Signal Processing Magazine*, vol. 37, no. 6, pp. 139–149, 2020.
- [65] J. Domingos and J. M. Moura, “Graph fourier transform: A stable approximation,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 4422–4437, July 2020.
- [66] F. R. Gantmacher, “Matrix theory,” *Chelsea, New York*, vol. 21, 1959.
- [67] J. Shi and J. M. F. Moura, “Graph signal processing: Dualizing gsp sampling in the vertex and spectral domains,” March 2021. Updated version of [71]. Submitted for publication.
- [68] P. Lancaster and M. Tismenetsky, *The Theory of Matrices With Applications*. Elsevier, 1985.
- [69] R. A. Horn and C. R. Johnson, *Matrix analysis, 2nd edition*. Cambridge University Press, 2012.
- [70] G. Leus, S. Segarra, A. Ribeiro, and A. G. Marques, “The dual graph shift operator: Identifying the support of the frequency domain,” 2017.
- [71] J. Shi and J. M. F. Moura, “Graph signal processing: Modulation, convolution, and sampling,” December 2019.
- [72] N. S. Jayant, “Analog scramblers for speech privacy,” *Computers & Security*, vol. 1, no. 3, pp. 275–289, 1982.
- [73] K. Sakurai, K. Koga, and T. Muratani, “A speech scrambler using the fast Fourier transform technique,” *IEEE Journal on Selected Areas in Comm.*, vol. 2, no. 3, pp. 434–442, 1984.
- [74] I. Pesenson, “Sampling of band-limited vectors,” *Journal of Fourier Analysis and Applications*, vol. 7, no. 1, pp. 93–100, 2001.

- [75] P. Vaidyanathan, *Multirate Systems and Filter Banks*. Prentice Hall, 1993.
- [76] J. Mason and D. Handscomb, *Chebyshev Polynomials*. CRC Press, 2002.
- [77] D. E. Knuth, *The Art of Computer Programming*, vol. 2: Seminumerical Algorithms. Addison-Wesley, 1981.
- [78] A. Agaska and Y. M. Lu, “A spectral graph uncertainty principle,” *IEEE Transactions on Information Theory*, vol. 59, no. 7, pp. 4338–4356, 2013.
- [79] M. Tsitsvero, S. Barbarossa, and P. Di Lorenzo, “Signals on graphs: Uncertainty principle and sampling,” *IEEE Transactions on Signal Processing*, vol. 64, no. 18, pp. 4845–4860, 2016.
- [80] W. M. Siebert, *Circuits, Signals, and Systems*. Cambridge, MA: The MIT Press, 1986.



# Chapter 11

## Appendices

### 11.1 Impulsive Graph Impulses

We briefly consider impulsive graph impulses in both the vertex and spectral domains. These are impulsive in one domain and not necessarily flat in the other. We show bases using shifted impulsive graph impulses in both the vertex and spectral domains are complete. So, they can be used as bases for graph signal representations of  $s$  and  $\hat{s}$  respectively (see section 5.1.1).

#### 11.1.1 Impulsive in Vertex Domain: $\delta_0^{\text{imp}}$

We define  $\delta_0^{\text{imp}}$  to be impulsive in the vertex domain. We take its GFT to get  $\hat{\delta}_0^{\text{imp}}$ .

$$\delta_0^{\text{imp}} = e_0 \xrightarrow{\mathcal{F}} \hat{\delta}_0^{\text{imp}} = \text{GFT} \cdot e_0 = \frac{1}{\sqrt{N}} y_0 \quad (11.1)$$

The GFT of the impulse  $\delta_0^{\text{imp}}$  is the first column  $y_0$  of the GFT (in general, not flat), giving  $y_0$  special significance.

Consider the signal representation of graph signal  $s$  with respect to the basis formed using the shifted  $\delta_0^{\text{imp}}$ ,

$$B_{\delta^{\text{imp}}} = \{ \delta_n^{\text{imp}} \}.$$

**Result 11.1** (Shifts  $\delta_n^{\text{imp}}$ ). *With  $\delta_0^{\text{imp}} = e_0$ , the following holds:*

$$\delta_n^{\text{imp}} = a_0^{(n)} \xrightarrow{\mathcal{F}} \widehat{\delta}_n^{\text{imp}} = \frac{1}{\sqrt{N}} y_0 \odot \lambda^n = \mathcal{Y}_0 \cdot \lambda^n \quad (11.2)$$

$$B_{\delta^{\text{imp}}} = \{\delta_0^{\text{imp}}, \delta_1^{\text{imp}}, \dots, \delta_{N-1}^{\text{imp}}\} = \{a_0^{(0)}, a_0^{(1)}, \dots, a_0^{(N-1)}\} \quad (11.3)$$

$$D^{\text{imp}} = \begin{bmatrix} a_0^{(0)} & a_0^{(1)} & \dots & a_0^{(N-1)} \end{bmatrix} \xrightarrow{\mathcal{F}} \widehat{D}^{\text{imp}} = \mathcal{Y}_0 \mathcal{V} \quad (11.4)$$

where  $a_0^{(n)}$  is column 0 of  $A^n$ , in general, not impulsive,

$$\mathcal{Y}_0 = \frac{1}{\sqrt{N}} \text{diag}[y_0]$$

and  $\mathcal{V}$  is the Vandermonde matrix in (2.27).

*Proof.* To show (11.2), start with the GFT of the shifted impulse and then take the inverse GFT to get

$$\widehat{\delta}_n^{\text{imp}} = \widehat{A^n \cdot \delta_0^{\text{imp}}} = \Lambda^n \cdot \frac{1}{\sqrt{N}} y_0 = \frac{1}{\sqrt{N}} y_0 \odot \lambda^n = \mathcal{Y}_0 \cdot \lambda^n \quad (11.5)$$

$$\delta_n^{\text{imp}} = a_0^{(n)} = \text{GFT}^{-1} \left( \frac{1}{\sqrt{N}} y_0 \odot \lambda^n \right) = \text{GFT}^{-1} (\mathcal{Y}_0 \cdot \lambda^n) \quad (11.6)$$

Using  $\delta_n^{\text{imp}}$  and its GFT from (11.2), (11.4) follows. ■

We state the following assumption.

**Assumption 11.1** (Nonzero entries of first column  $y_0$  of GFT). *The entries of  $y_0$  are nonzero.*

Clearly, this assumption holds in DSP where  $y_0 = 1_N$ .

**Result 11.2** (Complete  $B_{\delta^{\text{imp}}}$ ). *Under assumptions 2.2 and 11.1,  $D^{\text{imp}}$  is invertible and impulse basis  $B_{\delta^{\text{imp}}} = \{\delta_n^{\text{imp}}\}$  complete.*

This follows because under assumption 2.2, the Vandermonde matrix  $\mathcal{V}$  is invertible, and under assumption 11.1,  $\mathcal{Y}_0$  is invertible, so  $\widehat{D}^{\text{imp}}$  and  $D^{\text{imp}}$  are invertible and  $B_{\delta^{\text{imp}}}$  is complete.

### 11.1.2 Impulsive in Spectral Domain: $\widehat{\delta}_{\text{sp},0}^{\text{imp}}$

We consider  $\widehat{\delta}_{\text{sp},0}^{\text{imp}}$ , impulsive in the *spectral* domain and not necessarily flat in the vertex domain.

This is the dual of the impulsive in vertex domain delta signal  $\delta_0^{\text{imp}}$ .

$$\widehat{\delta}_{\text{sp},0}^{\text{imp}} = e_0 \xleftrightarrow{\mathcal{F}} \delta_{\text{sp},0}^{\text{imp}} = \frac{1}{\sqrt{N}} y_0 \quad (11.7)$$

where  $y_0$  is the first column of  $\text{GFT}^{-1}$  and  $\delta_{\text{sp},0}^{\text{imp}}$  is not flat. Consider the signal representation of graph signal  $\widehat{s}$  with respect to the basis formed using the shifted  $\widehat{\delta}_{\text{sp},0}^{\text{imp}}$ ,

$$B_{\widehat{\delta}_{\text{sp}}^{\text{imp}}} = \left\{ \widehat{\delta}_{\text{sp},n}^{\text{imp}} \right\}.$$

**Result 11.3.** *The shifts in the spectral domain of  $\widehat{\delta}_{\text{sp},0}^{\text{imp}}$  are*

$$\widehat{\delta}_{\text{sp},n}^{\text{imp}} = M^n \cdot \widehat{\delta}_{\text{sp},0}^{\text{imp}} = m_0^{(n)} \xrightarrow{\mathcal{F}^{-1}} \delta_{\text{sp},0}^{\text{imp}} = \frac{1}{\sqrt{N}} y_0 \odot \lambda^{*n} = \mathcal{Y}_0 \cdot \lambda^{*n} \quad (11.8)$$

$$B_{\widehat{\delta}_{\text{sp}}^{\text{imp}}} = \left\{ \widehat{\delta}_{\text{sp},0}^{\text{imp}}, \widehat{\delta}_{\text{sp},1}^{\text{imp}}, \dots, \widehat{\delta}_{\text{sp},N-1}^{\text{imp}} \right\} = \left\{ m_0^{(0)}, m_0^{(1)}, \dots, m_0^{(N-1)} \right\} \quad (11.9)$$

$$\widehat{C}^{\text{imp}} = \begin{bmatrix} m_0^{(0)} & m_0^{(1)} & \dots & m_0^{(N-1)} \end{bmatrix} \xrightarrow{\mathcal{F}^{-1}} C^{\text{imp}} = \mathcal{Y}_0 \mathcal{V}^* \quad (11.10)$$

where  $m_0^{(n)}$  is column 0 of  $M^n$ ,

$$\mathcal{Y}_0 = \frac{1}{\sqrt{N}} \text{diag}[y_0],$$

and  $\mathcal{V}$  is the Vandermonde matrix in (2.27).

In general,  $\widehat{\delta}_{\text{sp},n}^{\text{imp}}$  are not impulsive. To prove result 11.3 and equation (11.8), note that the lhs follows from the impulsive definition of  $\widehat{\delta}_{\text{sp},n}^{\text{imp}}$ . The rhs follows as

$$\delta_{\text{sp},0}^{\text{imp}} = \widehat{M^n \cdot \widehat{\delta}_{\text{sp},n}^{\text{imp}}} = \Lambda^{*n} \cdot \frac{1}{\sqrt{N}} y_0 = \frac{1}{\sqrt{N}} y_0 \odot \lambda^{*n} = \mathcal{Y}_0 \cdot \lambda^{*n} \quad (11.11)$$

We state the following assumption.

**Assumption 11.2** (Nonzero entries of first column  $y_0$  of  $\text{GFT}^{-1}$ ). *The entries of  $y_0$  are nonzero.*

Clearly, this assumption holds in DSP where  $y_0 = 1_N$ .

**Result 11.4** (Complete  $B_{\widehat{\delta}_{\text{sp}}^{\text{imp}}}$ ). *Under assumptions 2.2 and 11.2,  $\widehat{C}^{\text{imp}}$  is invertible and impulse basis  $B_{\widehat{\delta}_{\text{sp}}^{\text{imp}}} = \{\widehat{\delta}_{\text{sp},n}^{\text{imp}}\}$  complete.*

This follows because under assumption 2.2, the Vandermonde matrix  $\mathcal{V}^*$  is invertible, and under assumption 11.2,  $\mathcal{Y}_0$  is invertible, so  $C^{\text{imp}}$  and  $\widehat{C}^{\text{imp}}$  are invertible and  $B_{\widehat{\delta}_{\text{sp}}^{\text{imp}}}$  is complete.

### 11.1.3 Conclusion

Impulsive delta signals in both the vertex and spectral domain and bases,  $B_{\delta^{\text{imp}}}$  and  $B_{\widehat{\delta}_{\text{sp}}^{\text{imp}}}$ , rely on column 0 of  $A^n$  and  $M^n$  respectively. This is not ideal. One needs to choose at which node where the impulse occurs (i.e., at which node the 1 in  $e_0$  occurs). In DSP, this is intuitive, the impulse for  $\delta_0$  and  $\widehat{\delta}_{\text{sp},0}$  occurs at  $t = 0$  and  $f = 0$  respectively. However, in GSP, with an arbitrary graph, the choice is not intuitive as there is no given or natural ordering of the nodes. This makes using impulsive delta signals less ideal than using the flat delta signals, described in previous sections 3.2.1 and 3.2.2.