Algebraic Signal Processing: Modeling and Subband Analysis

Aliaksei Sandryhaila

November 2010

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy in
Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA

Abstract

Traditional linear signal processing is based on viewing signals as sequences or functions in time that flow in one direction, from past through present into future. Somewhat surprisingly, the assumption that the most basic operation that can be performed on a signal is a time shift, or "delay," is sufficient to derive many relevant signal processing concepts, including spectrum, Fourier transform, frequency response and others.

This observation has led us to search for other linear, shift-invariant signal models that are based on a different definition of a basic signal shift, and hence have different notions of filtering, spectrum, and Fourier transform. Such models can serve as alternatives to the time signal model traditionally assumed in modern linear signal processing, and provide valuable insights into signal modeling in different areas of signal processing. The platform for our work is the algebraic signal processing theory, a recently developed axiomatic approach to, as well as a generalization of linear signal processing.

In this thesis we present a new class of infinite and finite discrete signal models built on a new basic shift called the generic nearest-neighbor shift. We construct filter and signals spaces for these new models, and identify the corresponding signal processing concepts, such as frequency, spectrum, Fourier transform, frequency response, and convolution. We also derive relevant properties of these models, such as the Parseval equality and the notions of low and high frequencies.

We then consider the problem of subband analysis for the newly constructed signal models. As a corresponding subband analysis tool for infinite signals, we extend the definition of filter banks to the new models, and construct perfect-reconstruction filter banks for subband decomposition. We also construct filter banks for robust signal transmission. As a subband analysis tool for finite signals, we study the implementation of appropriate discrete Fourier transforms. We propose a mathematical approach to the factorization of general discrete Fourier transform matrices, and apply it to construct fast computational algorithms for the Fourier

transforms of interest.

Finally, we consider possible applications of the constructed signal models in different areas of signal processing. For example, we demonstrate that the use of new signal models can be beneficial in such applications as the compression of ECG signals and the efficient implementation of widely-used discrete signal transforms, such as the discrete cosine transform.

Acknowledgements

This thesis concludes my four-year journey through graduate school. During this time, I have been extremely fortunate to meet and work with many outstanding people, who, in different ways, contributed to the successful completion of this thesis.

I thank my amazing advisors Markus Püschel and Jelena Kovačević for giving me an opportunity to pursue my graduate studies under their supervision. They have been a never-ending source of wonderful research ideas and inspiration. At the same time, they always possessed tremendous understanding and flexibility to allow me pursue those research problems that seemed the most appealing to me. I will always be grateful to them for their advice and guidance that extended far beyond the realm of research.

I would like to thank my committee members José Moura and Martin Vetterli for their invaluable feedback and comments on my research. Their suggestions and advice have motivated me to consider previously unfamiliar research areas, and led to considerable improvements in my work and this thesis.

I am very grateful to Jeremy Johnson who originally introduced me to the field of signal processing and encouraged me to pursue a graduate degree. It was after attending his classes and working under his guidance as an undergraduate student at Drexel University that I became interested in the mathematical aspects of signal processing and algorithm optimization.

I would like to thank all my collaborators and colleagues, past and present, with whom I had a good fortune to work. It has been an educational and fun experience to work with Amina Chebira on frames and filter banks; with Doru Balcan on alternative algebraic signal models; and with Yevgen Voronenko, Volodymyr Arbatov, Vas Chellappa, Frédéric de Mesmay, Peter Milder, and Paolo D'Alberto on Spiral. I am grateful to Franz Franchetti and James Hoe for their help and suggestions during our work on Spiral, and to Samir Saba for his feedback and cooperation in our work on ECG signal compression.

I would like to thank my fellow students Aurora Schmidt, Charles Jackson, Divyanshu Vats, Gowri Shrinivasa, João Mota, Joel Harley, Kyle Andersen, Marek Telgarsky, Nicholas O'Donoughue, Pablo Hennings Yeomans, and Ramu Bhagavatula for their helpful feedback on my presentations and insightful discussions on numerous research topics.

Also, I would like to thank the Department of Electrical and Computer Engineering and its staff, especially Carol Patterson and Elaine Lawrence.

Finally, I wish to thank my family and my parents for their undiminished support, tremendous patience, and constant encouragement through these years. For all their limitless trust and care, I forever owe them a debt of gratitude.

This work was supported in part by NSF grant CCF-0634967.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Linear signal processing is a well-developed and comprehensive theory. At its core lie fundamental concepts such as filters and signals, spectrum and frequency, z-transform and Fourier transform. Many advanced techniques and tools for signal analysis, processing, and reconstruction are developed from these concepts.

Since a few fundamental concepts define the entire theory, signal processing operates under some implicit assumptions. Traditionally, signals are viewed as sequences in time: they go in one direction, from past through present into future. The basic signal shift is a delay and its inverse is an advance, as shown in Fig 1.1(a).

(a) Time shift (b) Space shift

Figure 1.1: Basic shifts for time and space one-dimensional signal models.

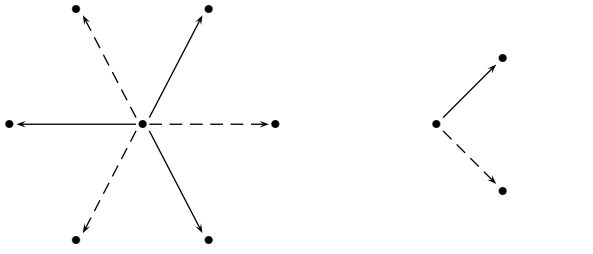
What if the underlying concepts of a signal model were different from the traditional ones? For example, there may be no inherent direction in a signal shift, or its action on a signal may somehow modify the signal contents. How much would it change the whole signal model and the corresponding signal processing theory? How can we process such signals?

Recently, some of these questions have been answered by a rigorous signal processing formalism called *algebraic signal processing* (ASP) [1–6]. It offers a structured axiomatic approach to building signal models based on fundamental general concepts. In particular, the construction of an ASP signal model starts with identifying a basic shift, which, for example, may reflect the desired relation between signal values in the model. All other concepts are defined based on this basic shift. As a result, we obtain a signal model with the desired underlying assumptions and the proper associated tools for signal analysis and processing.

For example, new *space* signal models have been constructed from the so-called space shift, shown in Fig 1.1(b), and studied in [1–3, 5, 6], as alternatives to the traditional time model. It was shown that the signal processing properties of the new models, such as convolution, Fourier transform, and others, are different from those of the standard time model. In particular, it was demonstrated that discrete cosine and sine transforms are the counterparts of the discrete Fourier transform for these models.

Both the time and space shifts are examples of one-dimensional (1-D) shifts. Multidimensional time and space signal models can be constructed from the corresponding 1-D signal models as tensor products of the underlying vector spaces of signals. Such models are called *separable*, since each dimension can be analyzed separately from others, as a 1-D signal model.

In addition, there exist *non-separable* multidimensional signal models that cannot be represented as tensor products of 1-D signal models. They are based on shifts that act in multiple dimensions. Examples include two-dimensionals hexagonal and quincunx signal models. The corresponding shifts are shown in Fig. 1.2. Such models have been studied in [7–11].



(a) Hexagonal shift (b) Quincunx shift

Figure 1.2: Basic shifts for hexagonal and quincunx non-separable two-dimensional signal models. One shift is indicated with solid lines, the other with dashed ones.

1.2 Thesis Contributions

The ASP theory introduces many questions about the existence and relevance of alternative signal models and the corresponding signal processing tools. The main goal of this thesis is to answer two of the most important of these questions:

1) Do there exist linear, shift-invariant signal models of interest, not studied previously by the traditional signal processing theory, or by the algebraic signal processing theory? How do we define signal processing concepts for the new models?

2) How do we define and construct appropriate subband analysis tools for signals for the new models, such as filter banks and finite discrete Fourier transforms?

In this thesis, we answer the above questions by identifying a new, large family of signal models, and then developing the theory of subband analysis tools for these models. We list the main contributions of the thesis below.

New signal models

We introduce a family of signal models that serve as alternatives to the traditional time model. In particular, we construct and study a family of *generic nearest-neighbor* signal models, which were originally suggested in [3]. These models are derived from the notion of the generic nearest-neighbor shift shown in Fig 1.3. In parallel to the traditional signal processing theory, we define analogs of all relevant concepts for these models, such as the filter and signal space, the operations of filtering and convolution, spectrum and frequency, z-transform and Fourier transform. Our goal is to demonstrate that generic nearest-neighbor models are a legitimate alternative to the time models traditionally assumed in linear signal processing.

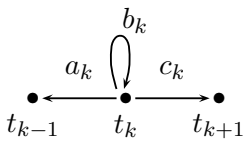


Figure 1.3: Generic nearest-neighbor shift. Coefficients $a_k, b_k, c_k \in \mathbb{R}$ are real numbers, and satisfy $a_{k+1}c_k > 0$.

Subband analysis

We construct the corresponding tools for subband analysis of the newly designed signals, and study their design and efficient implementation. For infinite discrete signals, a common tool for subband analysis and processing is a filter bank; for finite discrete signals, it is a (generalized) discrete Fourier transform.

i) Filter banks can be viewed either as arrays of band-pass filters, or as expansions of a signal into properly designed signal bases or frames. We use an alternative theoretical framework for filter bank construction based on the combination of the above methods.

   We also generalize the concepts of low and high frequencies. Since these concepts depend on the signal model, we concentrate on the above generic nearest-neighbor model and construct perfect-reconstruction filter banks for infinite generic nearest-neighbor signals.

ii) For finite discrete signals, the ASP theory establishes that the spectrum, i.e. the set of all frequencies, is a finite set. Hence, we can compute *all* frequency components of a signal. The corresponding mechanism is the generalized discrete Fourier transform, which is a so-called polynomial transform defined by the underlying signal model. The main research challenge lies in the efficient implementation of this transform.

   We study the existence and construction of fast algorithms for these generalized Fourier transforms. Specifically, we develop a novel algebraic method for the recursive decomposition of a general polynomial transform into a series of structured steps. Then, we apply this decomposition to the corresponding discrete Fourier transform associated with finite generic nearest-neighbor signals and derive efficient algorithms.

Applications

Although the main focus of this thesis is a theoretical contribution to the foundations of signal processing, we also consider potential applications of the generic nearest-neighbor models and subband analysis tools. We consider signal compression, and in particular, we study the compression of electrocardiographic signals. We also apply the developed theory for polynomial transform decomposition to the derivation of fast algorithms for different signal transforms.

Furthermore, we discuss other potential applications, including the fast computation of Karhunen-Loève transforms for Gauss-Markov random fields, and the potential use of generic nearest-neighbor models in climate modeling.

1.3 Thesis Outline

The thesis is organized as follows. We start by introducing the algebraic signal processing theory in Chapter 2, and discuss the algebraic meaning of signal processing concepts, such as a signal model, Fourier transform, filtering and convolution. We also provide examples of previously studied signal models. Then, in Chapter 3, we discuss properties and structure of filter banks, as well as existing approaches to the design and implementation of filter banks. After reviewing the background material, we construct infinite and finite generic nearest-neighbor models in Chapter 4, and define and derive the corresponding signal processing concepts for these models. In Chapter 5, we study filter bank design for infinite generic nearest-neighbor signals, and construct example filter banks for different signal models. Then we derive an algebraic approach to the derivation of fast algorithm for generalized Fourier transforms in Chapter 6, and demonstrate how it yields fast computational algorithms for discrete Fourier transforms that correspond to finite generic nearest-neighbor signal models. After that, we study various applications of the theory developed in this thesis in Chapter 7. Finally, in Chapter 8, we review the completed work and discuss future directions.

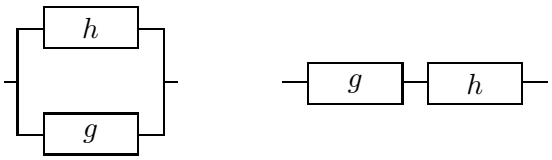
Chapter 2

Background: Algebraic Signal Processing Theory

In this chapter, we provide background on the algebraic signal processing theory. In particular, we explain the notion of an algebraic signal model and define the corresponding signal processing concepts. We then discuss infinite and finite discrete signal models and their instantiations as time and space models.

2.1 Main Concepts

The algebraic signal processing theory (ASP) [1–6] is both a generalization of and an axiomatic approach to standard linear signal processing theory. ASP is based on the concept of a signal model defined as a triple $(\mathcal{A}, \mathcal{M}, \Phi)$, where $\mathcal{A}$ is an algebra of filters, $\mathcal{M}$ is an $\mathcal{A}$-module of signals, and Φ is a linear mapping from the vector space of signals into the signal module $\mathcal{M}$ that generalizes the concept of a z-transform. Each signal model corresponds to different notions of signal and filter spaces, the z-transform, the shift, the Fourier transform, and other concepts. We start by defining the signal model and associated basic signal processing concepts.



(a) Addition: $h + g$ (b) Multiplication: $h \cdot g$

Figure 2.1: Addition and multiplication of filters.

Algebra (filter space)

A $\mathbb{C}$-*algebra* is a ring that is simultaneously a $\mathbb{C}$-vector space, such that the addition in the ring and the addition in the vector space coincide. In other words, an algebra is a vector space that is closed under the multiplication of its elements, such that the distributive law holds. Examples of algebras include the set of complex numbers $\mathbb{C}$ and the set of complex polynomials in one variable $\mathbb{C}[x]$.

Consider the vector space $\mathcal{A}$ of filters in linear signal processing. We denote filters in $\mathcal{A}$ with h. If for two filters $h_1, h_2 \in \mathcal{A}$ we associate their parallel connection with the addition $h_1 + h_2$, and their serial connection with the product $h_1 h_2$, then $\mathcal{A}$ becomes an algebra. These operations are visualized in Fig. 2.1.

Among all filters, a *shift* has a special role. The shift can be viewed as a basic non-trivial filter, and a common assumption made in linear signal processing is that (in the discrete case) all other filters can be written as linear combinations or series in multiples of the shift. In ASP, this key assumption leads to the recognition of shifts as generators of the filter algebra, as we discuss below.

Module (signal space)

Let $\mathcal{A}$ be a $\mathbb{C}$-algebra. An $\mathcal{A}$-*module* is a $\mathbb{C}$-vector space $\mathcal{M}$ on which $\mathcal{A}$ operates. This means the operation of multiplication of elements of $\mathcal{M}$ by elements of $\mathcal{A}$ is well-defined. $\mathcal{M}$ is closed under this multiplication

and the distributive law holds. Formally, for any $s, t \in \mathcal{M}$, and $h, g, 1 \in \mathcal{A}$, the following holds:

$$\begin{aligned} hs &\in \mathcal{M}, \\ h(s+t) &= hs + ht, \\ (h+g)s &= hs + gs, \\ (hg)s &= h(gs), \\ 1m &= m. \end{aligned}$$

In ASP, a signal space $\mathcal{M}$ is an $\mathcal{A}$-module, where $\mathcal{A}$ is the associated filter space. The corresponding operation of $\mathcal{A}$ on $\mathcal{M}$ is *filtering*. We denote signals in $\mathcal{M}$ with s.

Signal model

Typically, discrete signals are represented as sequences of real or complex numbers $\mathbf{s} = (s_k)_{k \in I} \in V$, where I is an index domain and V is a vector space. The purpose of a signal model is to define a notion of filtering for V. This is achieved by bijective mapping Φ that maps each signal $\mathbf{s} \in V$ to a signal $s \in \mathcal{M}$, for which filtering is defined as the operation of $\mathcal{A}$ on $\mathcal{M}$.

In summary, a *signal model* for a vector space V is a triple $(\mathcal{A}, \mathcal{M}, \Phi)$, where $\mathcal{A}$ is a chosen filter algebra, $\mathcal{M}$ is an associated signal $\mathcal{A}$-module, and Φ is a bijective mapping from V to $\mathcal{M}$. Φ generalizes the concept of the z-transform. Namely, for a fixed basis $\big(p_k\big)_{k \in I}$[1] of $\mathcal{M}$, Φ has the form

$$\Phi(\mathbf{s}) = \sum_{k \in I} s_k p_k. \tag{2.1}$$

An important example of a signal model is the one for which the signal module and the algebra of filters are equal as sets. Such a module (and the corresponding model) is called *regular*.

As shown in [3, 4] and explained next, the concept of the signal model is sufficient to define basic signal processing concepts including convolution, spectrum, and Fourier transform. These concepts take different forms for different models.

[1] Hereafter, $\big(\dots\big)$ denotes a list. We view lists as indexed sets; they do not contain duplicate elements.

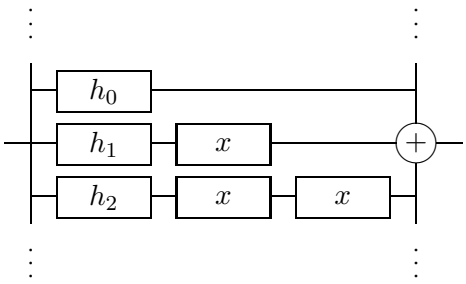


Figure 2.2: Representation of a filter $h = h(x) = \sum_\ell h_\ell x^\ell$ as a linear combination or a series of multiples of shift x.

Shift-invariance

In our research, we will be working only with *shift-invariant* signal models. In ASP this condition means that shifting any signal $s \in \mathcal{M}$ with the *basic shift* $x \in \mathcal{A}$ and then filtering it by a filter $h \in \mathcal{A}$ be equivalent to the filtering followed by the shift:

$$h \cdot (xs) = x \cdot (hs). \tag{2.2}$$

Assume that every filter $h \in \mathcal{A}$ can be represented as a linear combination or a series of multiples of shifts:

$$h = h(x) = \sum_\ell h_\ell x^\ell,$$

as shown in Fig.2.2. In mathematical terms, x acts as a *generator* of the filter algebra $\mathcal{A} = \langle x \rangle_{alg}$. Since any filters $h(x), g(x) \in \mathcal{A}$ can be written as series in x, the shift-invariance requirement (2.2) implies that

$$h(x)g(x) = g(x)h(x).$$

Hence, $\mathcal{A}$ is a *commutative* algebra.

As explained in [3, 4], commutative algebras are generated by the shift x as follows. For infinite discrete signals, $\mathcal{A} = \{\sum_\ell h_\ell x^\ell\}$ is an algebra of series in x, in which addition and multiplication are performed as usual. For finite discrete signals, $\mathcal{A} = \mathbb{C}[x]/p(x)$ is an algebra of all polynomials of degree less than $\deg p(x)$, in which addition and multiplication are performed modulo fixed polynomial $p(x)$. Such an algebra is called a *polynomial algebra*.

Filter representations

For any filter $h \in \mathcal{A}$, we can define a linear mapping on $\mathcal{M}$

$$s \mapsto h \cdot s. \tag{2.3}$$

With respect to basis $b = \left(p_k\right)_{k \in I}$ of $\mathcal{M}$, h can be associated with a $|I| \times |I|$ matrix M_h. Hence, there exists an algebra homomorphism between the filter space $\mathcal{A}$ and the space of $|I| \times |I|$ matrices $\mathbb{C}^{|I| \times |I|}$:

$$\begin{array}{rccc} \phi: & \mathcal{A} & \rightarrow & \mathbb{C}^{|I| \times |I|}, \\ & h & \mapsto & M_h. \end{array} \tag{2.4}$$

This homomorphism is called the *matrix representation of $\mathcal{A}$ afforded by the $\mathcal{A}$-module $\mathcal{M}$ with basis b.* Hence, filtering can be expressed either as a product $h(x)s(x)$ or as a matrix-vector product $\phi(h) \cdot \mathbf{s}$, where

$$\mathbf{s} = \begin{pmatrix} \dots & s_0 & s_1 & s_2 & \dots \end{pmatrix}^T \in V$$

is the vector representation of the signal $s \in \mathcal{M}$.

Spectrum

A vector subspace $\mathcal{M}' \leq \mathcal{M}$ that is itself an $\mathcal{A}$-module, is called an *$\mathcal{A}$-submodule* of $\mathcal{M}$. If $\mathcal{M}$ does not have non-trivial submodules (besides $\{0\}$ and itself), it is called *irreducible*.

In particular, every one-dimensional $\mathcal{A}$-submodule $\mathcal{M}'$ of $\mathcal{M}$ is irreducible. Furthermore, it is closed under the operations of any $h \in \mathcal{A}$, and hence is an eigenspace for any $h \in \mathcal{A}$. Namely, for any $s \in \mathcal{M}'$,

$$hs = \lambda_h s$$

for some $\lambda_h \in \mathbb{C}$.

We can write the set of all irreducible submodules of $\mathcal{M}$ as $\left(\mathcal{M}_\omega\right)_{\omega \in W}$, where W is a corresponding index domain. We call the index ω the *frequency* and W the *spectrum* of the signal model. Each $\mathcal{M}_\omega$ is called a *spectral component* of $\mathcal{M}$. A subset $W' \subseteq W$ is called a *frequency band*.

Fourier transform

It may be possible to represent the signal module $\mathcal{M}$ as a direct sum of its irreducible submodules. In this case we can define the following mapping:

$$\begin{array}{rccl} \Delta: & \mathcal{M} & \to & \bigoplus_{\omega \in W} \mathcal{M}_\omega, \\ & s & \mapsto & (s_\omega)_{\omega \in W}. \end{array} \tag{2.5}$$

This mapping is an $\mathcal{A}$-module homomorphism. Namely, for any $h \in \mathcal{A}$ and $s, t \in \mathcal{M}$, the following holds:

$$\begin{aligned} \Delta(s+t) &= \Delta(s) + \Delta(t), \\ \Delta(h \cdot s) &= h \cdot \Delta(s). \end{aligned}$$

In signal processing, Fourier analysis involves the decomposition of signals into spectral components. Since the homomorphism Δ expresses $s \in \mathcal{M}$ in terms of its spectral components s_ω, we call this homomorphism the *Fourier transform*.

Since for any signal $s \in \mathcal{M}$ and filter $h \in \mathcal{A}$, we have $\Delta(h \cdot s) = h \cdot \Delta(s)$, the general form of the convolution theorem becomes

$$h \cdot s = \Delta^{-1}(h \cdot \Delta(s)),$$

provided the inverse Δ^{-1} exists.

Frequency response

Similarly to the homomorphism (2.3), we can define a linear mapping on a spectral component $\mathcal{M}_\omega$ for each filter $h \in \mathcal{A}$:

$$s \mapsto h \cdot s. \tag{2.6}$$

If we fix a basis $\left(q_k\right)_{0\leq k<\dim M_\omega}$ in $\mathcal{M}_\omega$, then $\mathcal{M}_\omega$ affords an *irreducible* representation ϕ_ω of $\mathcal{A}$:

$$\begin{array}{rccl} \phi_\omega: & \mathcal{A} & \to & \mathbb{C}^{\dim M_\omega \times \dim M_\omega}, \\ & h & \mapsto & \phi_\omega(h). \end{array} \tag{2.7}$$

The matrix $\phi_\omega(h)$ is the *frequency response* of h at frequency ω. The collection $(\phi_\omega(h))_{\omega\in W}$ is the frequency response of h.

Suppose we choose basis $b = \left(p_k\right)_{k\in I}$ in $\mathcal{M}$, and in each $\mathcal{M}_\omega$ we choose $\left(q_k\right)_{0\leq k<\dim M_\omega}$ as a basis. Then we can express Δ in (2.5) in coordinate form as

$$\begin{array}{rccl} \mathcal{F}: & V & \to & \bigoplus_{\omega\in W} \mathbb{C}^{\dim \mathcal{M}_\omega}, \\ & \mathbf{s} & \mapsto & (\mathbf{s}_\omega)_{\omega\in W}. \end{array} \tag{2.8}$$

Filtering in the coordinate form is given by the matrix $\phi(h)$. Filtering in the decomposed module $\bigoplus_{\omega\in W} \mathcal{M}_\omega$ is represented by the direct sum of frequency responses $\bigoplus_{\omega\in W} \phi_\omega(h)$, where a direct sum of two matrices is defined as

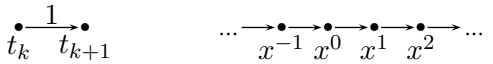
$$A \oplus B = \begin{pmatrix} A & \\ & B \end{pmatrix}.$$

The Fourier transform maps the modules $\mathcal{M}$ and $\bigoplus_{\omega\in W} \mathcal{M}_\omega$ to each other. In addition, filtering in $\mathcal{M}$ is equivalent to parallel filtering in the spectral components $\mathcal{M}_\omega$, as defined by the property $\Delta(h \cdot s) = h \cdot \Delta(s)$. Hence, we can use the coordinate version $\mathcal{F}$ of the Fourier transform to block-diagonalize the matrix representation $\phi(h)$ of $h \in \mathcal{A}$:

$$\mathcal{F} \cdot \phi(h) \cdot \mathcal{F}^{-1} = \bigoplus_{\omega\in W} \phi_\omega(h).$$

2.2 Infinite Discrete Models: Examples

As we explained before, the shift-invariant, infinite discrete signal model correspond to signals and filters that are series in multiples of the basic shift x. Depending on the definition of the shift, we obtain different signal models. The following examples of the (standard) infinite discrete time signal model and infinite



(a) Time shift (b) Time model visualization

Figure 2.3: 1-D infinite discrete time model: shift and visualization.

discrete space model illustrate the above concepts.

2.2.1 Infinite Discrete Time Model

The signal model commonly adopted for the infinite discrete time signal processing is for the space of finite-energy sequences $V = \ell^2(\mathbb{Z})$. It is given by

$$\begin{aligned} &\mathcal{A} = \{h = \textstyle\sum_{\ell\in\mathbb{Z}} h_\ell x^\ell \mid \mathbf{h} = \big(\dots, h_{-1}, h_0, h_1, \dots\big) \in \ell^1(\mathbb{Z})\}, \\ &\mathcal{M} = \{s = \textstyle\sum_{k\in\mathbb{Z}} s_k x^k \mid \mathbf{s} = \big(\dots, s_{-1}, s_0, s_1, \dots\big) \in \ell^2(\mathbb{Z})\}, \\ &\Phi : \ell^2(\mathbb{Z}) \to \mathcal{M},\ \mathbf{s} \mapsto s = \textstyle\sum_{k\in\mathbb{Z}} s_k x^k. \end{aligned} \tag{2.9}$$

Here, Φ is the standard z-transform (we substitute $x = z^{-1}$).

This model can be built on the concept of the *time* shift operator $x \in \mathcal{A}$ that imposes the direction on the basis elements $p_k(x)$ of $\mathcal{M}$:

$$\begin{aligned} p_0(x) &= 1, \\ p_{k+1}(x) &= x \cdot p_k(x). \end{aligned} \tag{2.10}$$

The unique solution to this recurrence is $p_k(x) = x^k$, which is used as the basis for $\mathcal{M}$. We also use the same basis for $\mathcal{A}$. The time shift, its action on the signals, and the visualization of the 1-D infinite discrete time model are shown in Fig. 2.3.

Eigenfunctions. The associated Fourier transform is constructed by projecting a signal onto spectral components $\mathcal{M}_\omega$. To determine the spectral components, we first need to identify the eigenfunctions of $\mathcal{M}$.

An *eigenfunction* is a signal $E_a(x) \in \mathcal{M}$, such that for any $h(x) \in \mathcal{A}$,

$$h(x) \cdot E_a(x) = c \cdot E_a(x)$$

for some constant $c \in \mathbb{C}$ that depends on filter $h(x)$ and parameter a. Since $\mathcal{A}$ is generated by x, it is equivalent to require only that

$$x \cdot E_a(x) = c \cdot E_a(x) \tag{2.11}$$

for some constant $c \in \mathbb{C}$.

Each eigenfunction $E_a(x)$ spans an irreducible $\mathcal{A}$-submodule $\mathcal{M}_a$ of dimension 1:

$$\mathcal{M}_a = \Big\{ c \cdot E_a(x) \mid c \in \mathbb{C} \Big\}.$$

For the infinite discrete time model, the eigenfunctions are of the form

$$E_a(x) = \sum_{k \in \mathbb{Z}} a^k x^k \tag{2.12}$$

for any $a \in \mathbb{C}$, since

$$xE_a(x) = x \cdot \sum_{k \in \mathbb{Z}} a^k x^k = a^{-1} \sum_{k \in \mathbb{Z}} a^k x^k = a^{-1} E_a(x).$$

Spectrum. It is not necessary to use the entire space $\mathbb{C}$ as the spectrum of the infinite time model in order to obtain an invertible Fourier transform. Instead, it is sufficient to choose the interval of orthogonality of basis functions x^k. The standard choice of spectrum W is the unit circle $|x| = 1$. If we parameterize it by $x = e^{j\omega}$, with $\omega \in [0, 2\pi)$, then

$$\int_0^{2\pi} e^{j\omega k} (e^{j\omega m})^* d\omega = \int_0^{2\pi} e^{j\omega(k-m)} d\omega = 2\pi \delta_{k-m}.$$

Fourier transform. Since we parameterized the spectrum as $e^{j\omega k}$ with $\omega \in [0, 2\pi)$, we consider only eigenfunctions (2.12) of the form

$$E_\omega(x) = \sum_{k \in \mathbb{Z}} e^{j\omega k} x^k.$$

They satisfy

$$x \cdot E_\omega(x) = e^{-j\omega} E_\omega(x). \tag{2.13}$$

Respectively, the Fourier transform and its inverse then are defined as

$$\begin{aligned} \Delta: \quad & S(\omega) = \langle s, E_\omega(x) \rangle = \sum_{k \in \mathbb{Z}} s_k e^{-j\omega k}, \\ \Delta^{-1}: \quad & s_k = \frac{1}{2\pi} \int_0^{2\pi} S(\omega) e^{j\omega k} d\omega. \end{aligned}$$

As expected, this is the standard definition of the discrete time Fourier transform.

Frequency response. From (2.13) we obtain

$$h(x) E_\omega(x) = h(e^{-j\omega}) E_\omega(x).$$

Hence, the frequency response of a filter $h = \sum_{\ell \in \mathbb{Z}} h_\ell x^\ell$ at frequency $\omega \in [0, 2\pi)$ is given by

$$h(e^{-j\omega}) = \sum_{\ell \in \mathbb{Z}} h_\ell e^{-j\omega\ell} = H(\omega).$$

Convolution. The filtering associated with the infinite discrete time model (2.9) is defined by the operation of $h(x) \in \mathcal{A}$ on $s(x) \in \mathcal{M}$:

$$hs = h(x)s(x).$$

This can be written in coordinate form as the standard time convolution of **h** and **s**:

$$h(x)s(x) = \sum_{k \in \mathbb{Z}} \hat{\mathbf{s}}_k x^k,$$

where

$$\hat{s}_k = (\mathbf{h} * \mathbf{s})_k = \sum_{\ell \in \mathbb{Z}} h_\ell s_{k-\ell}. \tag{2.14}$$

Also, it immediately follows from the definitions of the Fourier transform and the frequency response that

the convolution corresponds to the multiplication in the frequency domain:

$$\hat{S}(\omega) = H(\omega)S(\omega).$$

Parseval equality. The Parseval equality establishes the connection between the energy of the signal and the energy of its Fourier transform. For infinite discrete time signals, it has the form [12–15]

$$\sum_{k\in\mathbb{Z}} |s_k|^2 = \frac{1}{2\pi}\int_0^{2\pi} |S(\omega)|^2 d\omega.$$

Frequency domain. We call the space of the Fourier transforms $S(\omega)$ for all $s(x) \in \mathcal{M}$ the *frequency domain*. For the infinite discrete time model, the frequency domain is a Hilbert space of continuous finite-energy functions defined on $(-\pi, \pi]$, with the inner product

$$\langle u, v\rangle = \int_0^{2\pi} u(\omega)v^*(\omega)d\omega.$$

The set $\left(e^{j\omega k}\right)_{k\in\mathbb{Z}}$, where $\omega \in W = [0, 2\pi)$, is an orthogonal basis in this frequency domain.

2.2.2 Infinite Discrete Space Model

Infinite discrete space models are derived and defined in [5]. They are obtained from a different notion of a shift operator: the symmetric shift

$$x \cdot p_k(x) = \frac{1}{2}(p_{k-1}(x) + p_{k+1}(x)),$$

visualized in Fig. 2.4(a). The solution to the underlying recurrence

$$\begin{aligned} p_0(x) &= 1, \\ p_{k+1}(x) &= 2xp_k(x) - p_{k-1}(x), \end{aligned} \tag{2.15}$$

yields exactly the Chebyshev polynomials $p_k(x) = C_k(x)$ [16, 17] discussed in Appendix A.

Chebyshev polynomials are defined for $k \geq 0$. However, $C_k(x)$ for negative k can be computed

(a) Space shift (b) Space model visualization ($C = V$)

Figure 2.4: 1-D infinite discrete space model: shift and visualization.

from (2.15) as well. For example, consider the Chebyshev polynomials of the third kind $C = V$ that correspond to $C_0(x) = 1$ and $C_1(x) = 2x - 1$. For $k < 0$, we obtain $V_{-k}(x) = V_{k-1}(x)$. In particular, $V_{-1}(x) = V_0(x)$, which is called the *left boundary condition*. In the visualization of the corresponding signal model in Fig. 2.4(b) it is indicated by the loop at the left boundary.

Chebyshev polynomials $V(x)$ are orthogonal on the interval $[-1, 1]$ with respect to the weight function $\mu(x) = (1 + x)^{1/2}(1 - x)^{-1/2}$:

$$\int_{-1}^{1} V_k(x)V_m(x)\Big(\frac{1+x}{1-x}\Big)^{1/2} dx = \pi \cdot \delta_{k-m}.$$

Also, they satisfy the "symmetric shift" property

$$T_\ell(x) \cdot V_k(x) = \frac{1}{2}(V_{k-\ell}(x) + V_{k+\ell}(x)),$$

where $T_\ell(x)$ are the Chebyshev polynomial of the first kind. This property makes it convenient to choose the ℓ-fold space shifts $\{T_\ell(x)\}_{\ell \geq 0}$ as the basis of $\mathcal{A}$.

As a result we define the following infinite discrete space signal model:

$$\begin{aligned} &\mathcal{A} = \{h = \textstyle\sum_{\ell \geq 0} h_\ell T_\ell(x) \mid \mathbf{h} = \big(h_0, h_1, \dots\big) \in \ell^1(\mathbb{N}_0)\}, \\ &\mathcal{M} = \{s = \textstyle\sum_{k \geq 0} s_k V_k(x) \mid \mathbf{s} = \big(s_0, s_1, \dots\big) \in \ell^2(\mathbb{N}_0)\}, \\ &\Phi : \ \ell^2(\mathbb{N}_0) \to \mathcal{M}, \ \mathbf{s} \mapsto \textstyle\sum_{k \geq 0} s_k V_k(x). \end{aligned} \tag{2.16}$$

Eigenfunctions. Similarly to the infinite time model, we identify the eigenfunctions to define the asso-

ciated Fourier transform for the infinite space model. In this case, the eigenfunctions are of the form

$$E_a(x) = \sum_{k\geq 0} V_k(a)V_k(x), \quad a \in \mathbb{R}.$$

Namely, for any $a \in \mathbb{R}$, they satisfy

$$xE_a(x) = x \cdot \sum_{k\geq 0} V_k(a)V_k(x) = a \cdot \sum_{k\geq 0} V_k(a)V_k(x) = aE_a(x).$$

Spectrum. We choose the spectrum as a subset $W \subseteq \mathbb{R}$, such that the basis functions $V_k(x)$ are orthogonal over W. As discussed above, $[-1, 1]$ is the interval of orthogonality for Chebyshev polynomials. Hence, we set the spectrum $W = [-1, 1]$. As we show next, this choice makes the corresponding Fourier transform invertible.

Fourier transform. The considered eigenfunctions for the infinite discrete space model are

$$E_\omega(x) = \sum_{k\geq 0} V_k(\omega)V_k(x),$$

where $\omega \in [-1, 1]$. They satisfy

$$x \cdot E_\omega(x) = \omega \cdot E_\omega(x). \tag{2.17}$$

Hence, the associated Fourier transform and its inverse are defined as

$$\begin{aligned} \Delta: \quad & S(\omega) = \langle s, E_\omega(x)\rangle = \sum_{k\in\mathbb{Z}} s_k V_k(\omega) \\ \Delta^{-1}: \quad & s_k = \frac{1}{\pi}\int_{-1}^{1} S(\omega)V_k(\omega)\Big(\frac{1+\omega}{1-\omega}\Big)^{1/2} d\omega. \end{aligned}$$

Frequency response. From (2.17) we obtain

$$h(x)E_\omega(x) = h(\omega)E_\omega(x).$$

Hence, the frequency response of a filter $h = \sum_{\ell \geq 0} h_\ell T_\ell(x)$ at frequency $\omega \in [-1, 1]$ is given by

$$H(\omega) = \sum_{\ell \geq 0} h_\ell T_\ell(\omega).$$

Convolution. Similarly to the infinite discrete time model, filtering is represented by the multiplication

$$hs = h(x)s(x).$$

In the coordinate form, it is expressed as a convolution of **h** and **s** as

$$hs = \sum_{k \geq 0} \hat{s}_k V_k(x),$$

where

$$\hat{s}_k = (\mathbf{h} * \mathbf{s})_k = \sum_{\ell \in \mathbb{Z}} h'_\ell \frac{s'_{k-\ell} + s'_{k+\ell}}{2}$$

Here, $h'_0 = 2h_0$ and $h'_\ell = h_{|\ell|}$ for $\ell \neq 0$, and $s'_k = s_{|k|}$.

It follows from the definitions of the Fourier transform and the frequency response that the convolution corresponds to the multiplication in the frequency domain:

$$\hat{S}(\omega) = H(\omega)S(\omega).$$

Parseval equality. We state the Parseval equality as a theorem:

Theorem 2.2.1 *The Parseval equality for the 1-D infinite discrete space signal model for Chebyshev polynomials of the third kind is*

$$\sum_{k \geq 0} s_k^2 = \frac{1}{\pi} \int_{-1}^{1} S(\omega)^2 \Big(\frac{1+\omega}{1-\omega} \Big)^{1/2} d\omega.$$

Proof: Observe that

$$\begin{aligned}\sum_{k\geq 0} s_k^2 &= \sum_{k\geq 0} s_k \cdot \frac{1}{\pi}\int_{\omega\in W} S(\omega)V_k(\omega)\Big(\frac{1+\omega}{1-\omega}\Big)^{1/2}d\omega \\ &= \frac{1}{\pi}\int_{\omega\in W} S(\omega)\Big(\sum_{k\geq 0} s_k V_k(\omega)\Big)\Big(\frac{1+\omega}{1-\omega}\Big)^{1/2}d\omega \\ &= \frac{1}{\pi}\int_{\omega\in W} S^2(\omega)\Big(\frac{1+\omega}{1-\omega}\Big)^{1/2}d\omega.\end{aligned}$$

□

Frequency domain. The frequency domain for the infinite discrete time model is a Hilbert space of continuous finite-energy functions defined on $[-1, 1]$. The inner product is defined as

$$\langle u, v\rangle = \int_{-1}^{1} u(\omega)v(\omega)\Big(\frac{1+\omega}{1-\omega}\Big)^{1/2}d\omega.$$

The set $\big(V_k(\omega)\big)_{k\in\mathbb{Z}}$ is an orthogonal basis in this frequency domain.

Alternative infinite discrete space signal models. In addition to the signal model (2.16), space models for all four types of Chebyshev polynomials can be defined similarly [5].

2.3 Finite Discrete Models

As we mentioned in Section 2.1, a finite shift-invariant signal model $(\mathcal{A}, \mathcal{M}, \Phi)$ with one shift necessarily has $\mathcal{A} = \mathbb{C}[x]/p(x)$. Here, we first discuss the general finite discrete signal model following [3, 4], and then show examples of the standard finite discrete time model [3, 4] and the finite discrete space model [5].

2.3.1 General Finite Discrete Model

Let us fix $b = \big(p_k(x)\big)_{0 \le k < n}$ as the basis of the signal module $\mathcal{M} = \mathbb{C}[x]/p(x)$, where $n = \deg p(x)$. Then we can define the following finite discrete signal model:

$$\begin{aligned} &\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/p(x), \\ &\Phi: \ \mathbb{C}^n \to \mathcal{M}, \ \mathbf{s} \mapsto \textstyle\sum_{k=0}^{n-1} s_k p_k(x). \end{aligned} \tag{2.18}$$

Each such model is shift-invariant. The associated signal processing concepts are presented next.

Fourier transform. Assume that

$$p(x) = \prod_{k=0}^{n-1} (x - \alpha_k)$$

is a separable polynomial, i.e. its zeros are distinct: $\alpha_k \neq \alpha_\ell$ for $k \neq \ell$. Let $\alpha = \big(\alpha_0, \dots, \alpha_{n-1}\big)$. It follows from the Wedderburn theorem [18, 19] that the regular module $\mathcal{M} = \mathcal{A}$ can be decomposed into a direct sum of irreducible $\mathcal{A}$-modules. This decomposition is accomplished by the Chinese Remainder Theorem:

$$\begin{aligned} \Delta: \quad \mathbb{C}[x]/p(x) \quad &\to \quad \textstyle\bigoplus_{k=0}^{n-1} \mathbb{C}[x]/(x - \alpha_k), \\ s(x) \quad &\mapsto \quad \begin{pmatrix} s(\alpha_0) & s(\alpha_1) & \dots & s(\alpha_{n-1}) \end{pmatrix}^T. \end{aligned} \tag{2.19}$$

This is called the *generalized discrete Fourier transform*.

Let us choose the basis $\big(1\big)$ in each $\mathbb{C}[x]/(x-\alpha_k)$. Then the matrix that describes the isomorphism (2.19) is

$$\mathcal{P}_{b,\alpha} = \begin{pmatrix} p_0(\alpha_0) & p_1(\alpha_0) & \dots & p_{n-1}(\alpha_0) \\ p_0(\alpha_1) & p_1(\alpha_1) & \dots & p_{n-1}(\alpha_1) \\ \vdots & \vdots & & \vdots \\ p_0(\alpha_{n-1}) & p_1(\alpha_{n-1}) & \dots & p_{n-1}(\alpha_{n-1}) \end{pmatrix} = \Big[p_\ell(\alpha_k)\Big]_{0 \le k,\ell < n}. \tag{2.20}$$

This matrix is called a *polynomial transform* [20, 21].

The generalized discrete Fourier transform $\Delta(s(x))$ in (2.19) of a signal $s(x) = \sum_{k=0}^{n-1} s_k p_k(x) \in \mathcal{M}$

can be computed as the matrix-vector product

$$\Delta(s(x)) = \mathcal{P}_{b,\alpha} \cdot \mathbf{s}, \tag{2.21}$$

where $\mathbf{s} = \begin{pmatrix} s_0 & s_1 & \dots & s_{n-1} \end{pmatrix}^T$.

Spectrum. Each irreducible $\mathcal{A}$-submodule

$$\mathcal{M}_{\alpha_k} = \mathbb{C}[x]/(x - \alpha_k)$$

in (2.19) is a spectral component of $\mathcal{M}$. Accordingly, the set $W = \alpha$ of the zeros of $p(x)$ is the spectrum of the finite discrete model.

Frequency response. From (2.19) we observe that the projection of $s(x) \in \mathcal{M}$ on a spectral component $\mathcal{M}_{\alpha_k} = \mathbb{C}[x]/(x - \alpha_k)$ is the evaluation $s(\alpha_k)$, since

$$s(x) \equiv s(\alpha_k) \mod (x - \alpha_k).$$

Similarly,

$$h(x)s(x) \equiv h(x)s(\alpha_k) \equiv h(\alpha_k)s(\alpha_k) \mod (x - \alpha_k).$$

Hence, the frequency response of a filter $h(x) \in \mathcal{A}$ is given by

$$H(\alpha) = \begin{pmatrix} h(\alpha_0) & h(\alpha_1) & \dots & h(\alpha_{n-1}) \end{pmatrix}^T.$$

Convolution. Since $\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/p(x)$, filtering, and hence convolution, in the discrete model is represented by the product

$$hs = h(x)s(x) \mod p(x).$$

Filter representation. As explained in Section 2.1, the matrix representation of $\mathcal{A}$ afforded by $\mathcal{M}$ with basis b is a homomorphism

$$\begin{aligned} \phi: \quad \mathcal{A} &\to \mathbb{C}^{n\times n}, \\ h(x) &\mapsto \phi(h(x)) = M_h. \end{aligned} \tag{2.22}$$

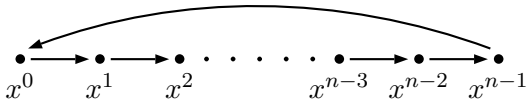


Figure 2.5: The visualization of the finite discrete time model $\mathbb{C}[x]/(x^n-1)$ with basis $1, x, x^2, \ldots, x^{n-1}$. The weight of each edge is 1.

Here, $M_h \in \mathbb{C}^{n\times n}$ is defined such that computing the polynomial product

$$\hat{s}(x) = h(x)s(x) \mod p(x)$$

is equivalent to computing the matrix-vector product

$$\hat{\mathbf{s}} = M_h \cdot \mathbf{s}$$

for any signal $s \in \mathcal{M}$ and filter $h \in \mathcal{A}$.

The corresponding polynomial transform diagonalizes the matrix representation of any $h(x) \in \mathcal{A}$:

$$\mathcal{P}_{b,\alpha} \cdot \phi(h(x)) \cdot \mathcal{P}_{b,\alpha}^{-1} = \operatorname{diag}\Big(h(\alpha_0), \ldots, h(\alpha_{n-1})\Big). \tag{2.23}$$

2.3.2 Finite Discrete Time Model

Consider the special case of (2.18) given by the signal model $\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/(x^n-1)$ with basis $b = \big(1, x, x^2, \ldots, x^{n-1}\big)$:

$$\begin{aligned} &\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/(x^n-1), \\ &\Phi:\ \mathbb{C}^n \to \mathcal{M},\ \mathbf{s} \mapsto \textstyle\sum_{k=0}^{n-1} s_k x^k. \end{aligned} \tag{2.24}$$

Observe that this model can be obtained from the infinite time model (2.9) by imposing the periodic boundary condition $x^n = 1$ [3,4]. The model is visualized in Fig. 2.5. The boundary condition is indicated by the arrow that connects the boundary points of the graph.

Fourier transform. Since the roots of $x^n - 1$ are $\alpha_k = \omega_n^k$, where $\omega_n = e^{-j\frac{2\pi}{n}}$ is a primitive n-th root of unity, the corresponding polynomial transform (2.20) is the well-known discrete Fourier transform

(DFT):

$$\mathcal{P}_{b,\alpha} = \left[\alpha_k^\ell\right]_{0\le k,\ell<n} = \left[\omega_n^{k\ell}\right]_{0\le k,\ell<n} = \mathrm{DFT}_n\,. \tag{2.25}$$

Spectrum. The spectrum of the finite discrete time model is $W = \left(\omega_n^k\right)_{0\le k<n}$.

Frequency response. The frequency response of a filter $h(x) \in \mathcal{A}$ is

$$H(\alpha) = \begin{pmatrix} h(\omega_n^0) & h(\omega_n^1) & \dots & h(\omega_n^{n-1}) \end{pmatrix}^T.$$

Convolution. The convolution is defined as

$$hs = h(x)s(x) \mod (x^n - 1),$$

which is equivalent to the circular convolution of **h** and **s**:

$$h(x)s(x) \mod (x^n - 1) = \sum_{k=0}^{n-1} \hat{s}_k x^k,$$

where

$$\hat{s}_k = (\mathbf{h} \circledast \mathbf{s})_k = \sum_{0\le \ell<n} h_{(k-\ell \mod n)} s_k. \tag{2.26}$$

Frequency domain. The frequency domain of the finite discrete time model can be viewed as the frequency domain of the infinite discrete time model (2.9) sampled at frequencies $\omega_k = 2\pi k/n$ [13, 22]. In particular, the basis functions $e^{j\omega m}$, $0 \le m < n$, sampled at ω_k, are the orthogonal basis of the frequency domain. Namely,

$$\left(1, e^{j2\pi m/n}, \dots, e^{j2\pi m(n-1)/n}\right)$$

is the m-th basis function.

2.3.3 Finite Discrete Space Model

Consider the special case of (2.18) given by the signal model $\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/V_n(x)$ with basis $b = \left(V_0(x), V_1(x), \dots, V_{n-1}(x)\right)$, where, as before, $V_k(x)$ denotes the k-th Chebyshev polynomial of the third

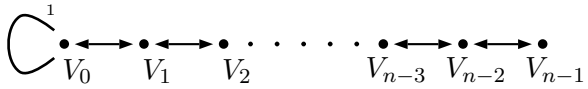


Figure 2.6: The visualization of the finite discrete space model $\mathbb{C}[x]/V_n(x)$ with basis $V_0(x), V_1(x), \ldots, V_{n-1}(x)$. The weight of each edge is $1/2$, except for the loop at the left boundary, where it is 1.

kind:

$$\begin{aligned} &\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/V_n(x), \\ &\Phi : \mathbb{C}^n \to \mathcal{M}, \ \mathbf{s} \mapsto \textstyle\sum_{k=0}^{n-1} s_k V_k(x). \end{aligned} \tag{2.27}$$

Observe that this model can be obtained from the infinite space model (2.16) by imposing the boundary condition $V_n(x) = 0$ [5]. The model is visualized in Fig. 2.6.

Fourier transform. Since the roots of $V_n(x)$ are $\alpha_k = \cos \frac{(2k+1)\pi}{2n+1}$, the corresponding Fourier transform (2.20) is a scaled discrete cosine transform of type 8 (DCT-VIII):

$$\begin{aligned} \mathcal{P}_{b,\alpha} &= \Big[V_\ell(\alpha_k)\Big]_{0\le k,\ell<n} = \Big[\cos\frac{(k+1/2)(\ell+1/2)\pi}{(n+1/2)} / \cos\frac{(k+1/2)\pi}{2n+1}\Big]_{0\le k,\ell<n} \\ &= \operatorname{diag}\Big(1/\cos\frac{(k+1/2)\pi}{2n+1}\Big)_{0\le k,\ell<n} \cdot \Big[\cos\frac{(k+1/2)(\ell+1/2)\pi}{(n+1/2)}\Big]_{0\le k,\ell<n} \\ &= \operatorname{diag}\Big(1/\cos\frac{(k+1/2)\pi}{2n+1}\Big)_{0\le k,\ell<n} \cdot \text{DCT-VIII}_n\,. \end{aligned} \tag{2.28}$$

Spectrum. The spectrum of the finite discrete space model is the list

$$W = \big(\cos\frac{(2k+1)\pi}{2n+1}\big)_{0\le k<n}$$

of roots of $V_n(x)$.

Frequency response. The frequency response of a filter $h(x) \in \mathcal{A}$ is

$$H(\alpha) = \begin{pmatrix} h(\cos\frac{\pi}{2n+1}) & h(\cos\frac{3\pi}{2n+1}) & \ldots & h(\cos\frac{(2n-1)\pi}{2n+1}) \end{pmatrix}^T.$$

Convolution. The convolution is defined as

$$hs = h(x)s(x) \mod V_n(x).$$

Frequency domain. The frequency domain of this finite discrete space model can be viewed as the frequency domain of the infinite discrete space model (2.16) sampled at frequencies $\omega_k = \cos\frac{(2k+1)\pi}{2n+1}$ [22]. In particular, the function

$$\left(\sqrt{1+\omega_0}V_m(\omega_0), \ldots, \sqrt{1+\omega_{n-1}}V_m(\omega_{n-1})\right)$$

is the m-th basis function of an orthogonal basis of the frequency domain.

Chapter 3

Background: Subband Analysis

Some of the most important techniques for signal processing are based on the analysis of a signal's frequency content. Often, a signal is split into several components occupying different subsets of the spectrum W, called *frequency bands*, which are then processed. This technique is called *subband analysis*.

For infinite discrete signal models, such as the time and space models discussed in Section 2.2, the spectrum is an uncountable set. Hence, it is not possible to perform analysis for each frequency individually. Instead, signal components that occupy continuous frequency bands are extracted. This can be performed by processing a signal with several bandpass filters, where each filter extracts a corresponding frequency band. The most common tool for such processing is a *filter bank*. In Section 3.1, we discuss the structure of filter banks, their properties and design techniques.

For finite discrete signals, such as the signal models discussed in Section 2.3, the spectrum is a finite set. Hence, we can compute each frequency component individually. This is performed by calculating the corresponding discrete Fourier transform of a signal. For a general discrete signal model (2.18), the corresponding polynomial transform (2.21) is used. The main challenge in this case is to perform the calculations efficiently. In Section 3.2, we discuss fast algorithms for polynomial transforms and approaches to the construction of such algorithms.

3.1 Filter Banks

A *filter bank* is an array of serially connected filters and down- and upsamplers; each serial connection constitutes a channel. The properties of the filters and samplers can differ significantly depending on the purpose of a filter bank. A thorough discussion of filter banks, their structure, design, and applications can be found in [23–25]. To date, only filter banks for the traditional time model have been studied. In the rest of Section 3.1, we review filter banks for time signals.

3.1.1 Filter Bank Structure

Filters

Filters used in filter banks are usually designed to attenuate the frequencies present in a processed signal. This is achieved by carefully constructing the frequency response of each filter. Depending on their frequency response, most filters can be classified as one of the following:

a) A *band-pass* filter $h(x)$ has a frequency response that is non-zero only for a specific subband $W_s \subset W$:

$$H(\omega) = 0$$

if $\omega \notin W_s$. In this case, W_s is called the filter's *passband*.

b) A *band-stop* filter $h(x)$ has a frequency response that is zero only for a specific subband $W_s \subset W$:

$$H(\omega) = 0$$

if $\omega \in W_s$. In this case, W_s is called the filter's *stopband*.

Special cases of the above include *all-pass* filters that do not remove any frequencies from a processed signal, but only attenuate them; and *notch* filters that are band-stop filters with a very small stopband.

Down- and upsamplers

A *downsampler* is a device that reduces the number of coefficients in a processed signal. The ratio of the number of coefficients in the incoming signal to the number of coefficients in the outgoing signal is called the *downsampling rate*.

Typical downsamplers with downsampling rate N are implemented by keeping every N-th coefficient of a signal and dropping others. If $\left(s_k\right)_{k\in\mathbb{Z}}$ is the incoming signal, and $\left(\hat{s}_k\right)_{k\in\mathbb{Z}}$ is the outgoing signal, then

$$\hat{s}_k = s_{kN}$$

for all $k \in \mathbb{Z}$.

Upsamplers are defined analogously. An *upsampler* is a device that increases the number of coefficients in a processed signal. The ratio of the number of coefficients in the outgoing signal to the number of coefficients in the incoming signal is called the *upsampling rate*.

Typical upsamplers with upsampling rate N are implemented by inserting $N-1$ zeros between the coefficients of a processed signal:

$$\hat{s}_k = \begin{cases} s_{k/N}, & \text{if} \quad N|k; \\ 0, & \text{otherwise}. \end{cases}$$

Analysis and synthesis parts

Typically, each channel starts with an *analysis* filter. Commonly, this is a bandpass filter that extracts a component of the processed signal that occupies a desired frequency subband. For efficient processing, the analysis filter is followed by a downsampler that reduces the overall number of coefficients to be processed. Proper downsampling does not result in information loss and does not prevent us from reconstructing the original signal from the remaining coefficients, as we explain in Section 3.1.2.

After the extracted and downsampled components are processed, the signal approximation is reconstructed by upsampling each component and filtering it with a *synthesis* filter, and then combining all reconstructed components into one signal. This is the *synthesis* part of the filter bank.

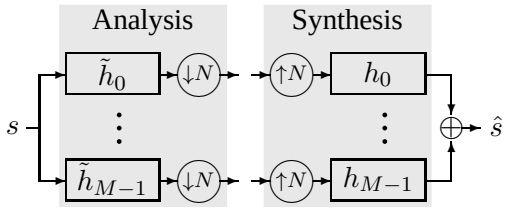


Figure 3.1: Standard M-channel filter bank sampled at rate N. $\tilde{h}_k$ and h_k denote analysis and synthesis filters.

Sampling of filter banks

Fig. 3.1 shows an M-channel filter bank with downsampling and upsampling rates N. If $N = M$, the filter bank is called *critically sampled*: the total number of coefficients computed in the analysis part of the filter bank is the same as in the original signal. If $N < M$, the filter bank is called *oversampled*, and the number of coefficients computed in the analysis part is larger than in the original signal. Finally, if $N > M$, the filter bank is called *undersampled*, and the number of coefficients computed in the analysis part is smaller than in the original signal.

In addition, the downsampling and upsampling rates in a filter bank can be different. In this case, the filter bank is said to have a *rational sampling rate* [26–29].

3.1.2 Nyquist Theorem and Downsampling

The use of downsampling and upsamping in filter banks can be explained by the Nyquist sampling theorem [30–34], which we re-state here:

Theorem 3.1.1 *Suppose an infinite continuous time signal $s(t) \in \mathcal{L}^2(\mathbb{R})$, with the Fourier transform*

$$S(\omega) = \int_{-\infty}^{\infty} s(t) e^{-j\omega t} dt,$$

is bandlimited to the bandwidth B:

$$S(\omega) = 0$$

for $|\omega| > B$.

Suppose an infinite discrete time signal $\left(s_k\right)_{k\in\mathbb{Z}}$ is obtained by sampling $s(t)$ at sampling points $t_k =$

kT :

$$s_k = s(kT).$$

Then the original signal $s(t)$ can be reconstructed from its samples $\big(s_k\big)_{k\in\mathbb{Z}}$ as

$$s(t) = \sum_{k\in\mathbb{Z}} s_k \cdot \operatorname{sinc}\Big(\frac{t-kT}{T}\Big),$$

where

$$\operatorname{sinc}(t) = \frac{\sin \pi t}{\pi t},$$

if the sampling interval T satisfies

$$T < \frac{2\pi}{2B}.$$

Suppose we have a two-channel, critically-sampled filter bank. Assume that analysis filter $\tilde{h}_0$ is an ideal half-band low-pass filter:

$$\tilde{H}_0(\omega) = \begin{cases} 1, & \text{if} \quad |\omega| < \pi/2, \\ 0, & \text{otherwise}. \end{cases}$$

Denote the incoming signal $\big(s_k\big)_{k\in\mathbb{Z}}$ filtered with $\tilde{h}_0$ with $\big(\hat{s}_k\big)_{k\in\mathbb{Z}}$. Assume that all signals have been sampled from continuous signals at points $t_k = kT$, such that sampling satisfies Nyquist theorem 3.1.1. In particular, $\big(\hat{s}_k\big)_{k\in\mathbb{Z}}$ has been sampled from $\tilde{s}(t)$.

Then $\big(\hat{s}_k\big)_{k\in\mathbb{Z}}$ can be downsampled by a factor of 2 without any loss of information, since it can be recovered from only half of its coefficients as follows:

$$\sum_{k\in\mathbb{Z}} \hat{s}_{2k} \cdot \operatorname{sinc}\Big(\frac{t-2kT}{T}\Big) = \hat{s}(t) = \sum_{k\in\mathbb{Z}} \hat{s}_k \cdot \operatorname{sinc}\Big(\frac{t-kT}{T}\Big).$$

Similarly, if $\tilde{h}_1$ is an ideal half-band high-pass filter with frequency response

$$\tilde{H}_1(\omega) = \begin{cases} 0, & \text{if} \quad |\omega| < \pi/2, \\ 1, & \text{otherwise}. \end{cases},$$

then filtering signal $\left(s_k\right)_{k\in\mathbb{Z}}$ with $\tilde{h}_1$ and downsampling the result by 2 does not result in any loss of information.

3.1.3 Filter Implementation

Difference equations

Consider a linear shift-invariant system that processes an incoming signal $s(x) = \sum_{k\in\mathbb{Z}} s_k x^k$ and outputs a signal $\hat{s}(x) = \sum_{k\in\mathbb{Z}} \hat{s}_k x^k$. Suppose, the input and output signals satisfy the relation

$$\sum_{m=0}^{M_b-1} b_m \hat{s}_{k-m} = \sum_{m=0}^{M_a-1} a_m s_{k-m}, \tag{3.1}$$

where $M_a, M_b \in \mathbb{N}_0$ are finite non-negative integers, and $a_m, b_m \in \mathbb{C}$ are constants, such that $b_0 \neq 0$. The relation (3.1) between a system's input and output is called a *linear constant-coefficient difference equation* [12, 14].

If we multiply the left- and right-hand sides of (3.1) by x^k and sum over all $k \in \mathbb{Z}$, we obtain

$$b(x)\hat{s}(x) = a(x)s(x), \tag{3.2}$$

where

$$a(x) = \sum_{m=0}^{M_a-1} a_m x^m$$

and

$$b(x) = \sum_{m=0}^{M_b-1} b_m x^m.$$

Re-writing (3.2) as

$$\hat{s}(x) = \frac{a(x)}{b(x)} s(x), \tag{3.3}$$

we obtain the *transfer function*

$$h(x) = \frac{a(x)}{b(x)},$$

of a linear shift-invariant system that is described by the difference equation 3.1 [12, 14].

FIR and IIR filters

The general form of a filter is

$$h(x) = \sum_{\ell=L_0}^{L_1} h_\ell x^\ell,$$

Depending on the values of integers $L_0, L_1 \in \mathbb{Z}$, filters can be separated into two classes:

1) *Finite impulse response* (FIR) filters have finite support: both L_0 and L_1 are finite integers.

2) *Infinite impulse response* (IIR) filters have infinite support: either L_0 or L_1 (or both) are infinity.

Any FIR filter $h(x)$ can be implemented as a linear shift-invariant system described by the difference equation 3.1, since we can set $a(x) = x^{-L_0}h(x)$ and $b(x) = x^{-L_0}$ in (3.3).

An IIR filter $h(x)$, on the other hand, can be implemented as a linear shift-invariant system described by the difference equation using a difference equation 3.1 only if it can be written in the form (3.3):

$$h(x) = \sum_{\ell\in\mathbb{Z}} h_\ell x^\ell = \frac{a(x)}{b(x)}.$$

Fast convolution

In addition to difference equations, an FIR filter $h(x) = \sum_{\ell=L_0}^{L_1} h_\ell x^\ell$ may also be implemented directly as a convolution (2.14):

$$\hat{s}_k = \sum_{\ell=L_0}^{L_1} h_\ell s_{k-\ell},$$

since the finite number of non-zero filter coefficients, or *taps*, allows us to compute the coefficients $\hat{s}_k$ in finite time. IIR filters, on the other hand, have an infinite number of taps, and cannot be implemented directly as a convolution.

A number of fast computation algorithms exist to compute the above convolution [12, 14, 20, 35]. Examples include *overlap-add* and *overlap-save* methods that are based on expressing the linear convolution (2.14) via the circular convolution (2.26) for the finite time model, and computing it using fast algorithms for DFT that we will discuss in Section 3.2.

3.1.4 Perfect-Reconstruction Filter Banks

An important class of filter banks are *perfect-reconstruction* filter banks that allow for an exact reconstruction of the input signal. Here, we review two main approaches to the design of perfect-reconstruction filter banks: a filter approach and a signal expansion approach. Both approaches were developed for the traditional time signal model. The filter approach is a more common approach to filter bank design, and can be found in most signal processing textbooks (see, for example, [12–15]). The signal expansion approach was developed as an alternative approach for the construction of perfect-reconstruction filter banks [23]. In traditional time signal processing, both approaches can be used interchangeably. In Chapter 5, we will discuss the generalization of these approaches for other signal models.

Filter approach

Since a filter bank can be characterized by its analysis filters, the original approach to the design of perfect-reconstruction filter banks was based on the properties of the analysis filters. Consider a two-channel critically-sampled filter bank in Fig. 3.1, implemented for the infinite discrete time model (2.9). Let the input signal be

$$s(x) = \sum_{k \in \mathbb{Z}} s_k x^k,$$

the analysis filters be

$$\tilde{h}_i(x) = \sum_{\ell \in \mathbb{Z}} \tilde{h}_{i,\ell} x^\ell,$$

and the synthesis filters be

$$h_i(x) = \sum_{\ell \in \mathbb{Z}} h_{i,\ell} x^\ell,$$

for $i = 0, 1$. Finally, let the output signal be

$$\hat{s}(x) = \sum_{k \in \mathbb{Z}} \hat{s}_k x^k.$$

It was shown in [36] and later generalized in [37, 38] that the output signal $\tilde{s}(x)$ satisfies the equation

$$\hat{s}(x) = s(x)\frac{\tilde{h}_0(x)h_0(x) + \tilde{h}_1(x)h_1(x)}{2} + s(-x)\frac{\tilde{h}_0(-x)h_0(x) + \tilde{h}_1(-x)h_1(x)}{2}.$$

Hence, we can obtain the perfect reconstruction $\tilde{s}(x) = s(x)$ if the following conditions hold:

$$\begin{cases} \tilde{h}_0(x)h_0(x) + \tilde{h}_1(x)h_1(x) = 2, \\ \tilde{h}_0(-x)h_0(x) + \tilde{h}_1(-x)h_1(x) = 0. \end{cases} \tag{3.4}$$

Similar conditions on the analysis and synthesis filters can be derived for filter banks with more than two channels [39, 40].

Signal expansion approach

An alternative viewpoint on the design of perfect-reconstruction filter banks is to expand a signal

$$\mathbf{s} = \big(\dots, s_{-1}, s_0, s_1, \dots\big)^T$$

into properly constructed bases

$$\big(\varphi_k^{(m)}\big)_{k\in\mathbb{Z}},$$

where $0 \le m < M$. The resulting expansion has the form

$$\hat{\mathbf{s}} = \sum_{k\in\mathbb{Z}} \langle\tilde{\varphi}_k^{(0)}, \mathbf{s}\rangle\varphi_k^{(0)} + \dots + \sum_{k\in\mathbb{Z}} \langle\tilde{\varphi}_k^{(M-1)}, \mathbf{s}\rangle\varphi_k^{(M-1)}, \tag{3.5}$$

where

$$\big(\tilde{\varphi}_k^{(m)}\big)_{k\in\mathbb{Z}},$$

for $0 \le m < M$, are the corresponding dual bases[1]. This leads to the filter bank interpretation shown in Fig. 3.2.

[1] A *dual basis* $\big(\tilde{\varphi}_k\big)$ is a basis in the dual space of a vector space spanned by the basis $\big(\varphi_k\big)$. The basis and its dual are mutually orthogonal: $\langle\varphi_k, \tilde{\varphi}_m\rangle = \delta_{k-m}$. Orthogonal bases are self-dual.

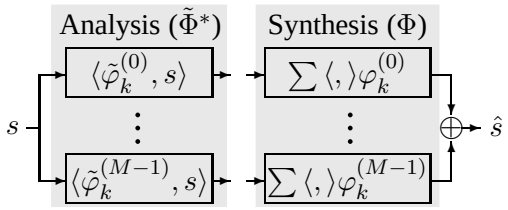


Figure 3.2: Signal expansion interpretations of an M-channel filter bank sampled by N. $\tilde{\Phi}^*$ and Φ are the analysis and synthesis matrices.

The design criteria for the bases $\left(\varphi_k^{(m)}\right)_{k\in\mathbb{Z}}$ include their frequency content and structure. In particular, we may view them as signals occupying a particular frequency band. Also, for practical purposes, the bases are often constructed as periodically shifted sequences of a few prototype signals. All these assumptions lead to the filter bank structure shown in Fig. 3.1, since the inner product and time convolution are interchangeable — one only needs to reverse the order of the coefficients [23]. The expansion approach can be further generalized by viewing $\left(\varphi_k\right)_{k\in\mathbb{Z}}$ and $\left(\tilde{\varphi}_k\right)_{k\in\mathbb{Z}}$ as frames[2], which results in (3.5) being an overcomplete expansion.

Such signal expansions have been studied in various areas of signal processing, including blocked transforms, such as windowed Fourier transform, lapped orthogonal transforms [42] and later lapped tight frame transforms [43–45]; signal compression and multiresolution analysis [46]; and robust transmission [44, 45, 47, 48].

Connection between the approaches

To illustrate the connection between the filter approach and the signal expansion approach, consider the M-channel filter bank in Fig. 3.1. Assume that all analysis and synthesis filters

$$\tilde{h}_m = (\tilde{h}_{m,0}, \ldots, \tilde{h}_{m,L-1})^T$$

and

$$h_m = (h_{m,0}, \ldots, h_{m,L-1})^T$$

[2]A *frame* $\left(\varphi_k\right)$ is a redundant set of vectors that span a vector space V [41]. Unlike basis vectors, frame vector can be linearly dependent. A *dual frame* $\left(\tilde{\varphi}_k\right)$ is another frame such that any vector $v \in V$ can be expanded as $v = \sum_k \langle \tilde{\varphi}_k, v \rangle \varphi_k$. A self-dual frame is called *tight*.

are FIR filters, and have the same length $L = qN$ for some $q \in \mathbb{N}_0$. For a signal $\mathbf{s}$, the operation of the filter bank can be described via matrix-vector products: the transform

$$\mathbf{S} = \tilde{\Phi}^* \mathbf{s}$$

is filtering followed by downsampling (the analysis part), and the inverse transform

$$\hat{\mathbf{s}} = \Phi \mathbf{S}$$

is upsampling followed by filtering (the synthesis part). Here, Φ has the form

$$\Phi = \begin{pmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ \dots & \Phi_0 & 0 & \dots & 0 & 0 & \dots \\ \dots & \Phi_1 & \Phi_0 & \dots & 0 & 0 & \dots \\ \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots \\ \dots & \Phi_{q-1} & \Phi_{q-2} & \dots & \Phi_0 & 0 & \dots \\ \dots & 0 & \Phi_{q-1} & \dots & \Phi_1 & \Phi_0 & \dots \\ \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \tag{3.6}$$

where each block $\Phi_r,\ 0 \le r \le q-1$, is the $N \times M$ matrix

$$\Phi_r = \begin{pmatrix} h_{0,rN} & \dots & h_{M-1,rN} \\ \vdots & \ddots & \vdots \\ h_{0,rN+N-1} & \dots & h_{M-1,rN+N-1} \end{pmatrix}. \tag{3.7}$$

The analysis matrix $\tilde{\Phi}$ is constructed similarly from the reverses of filters $\tilde{h}_k$; the order of the coefficients of $\tilde{h}_k$ in the columns of blocks $\tilde{\Phi}_r$ is reversed.

Then, $\varphi_k^{(m)}$ in the expansion (3.5) is the $(kM+m)$-th column of Φ; and $\tilde{\varphi}_k^{(m)}$ is the $(kM+m)$-th column of $\tilde{\Phi}$. If $M = N$, then the columns of Φ and $\tilde{\Phi}$ are bases, dual to each other; if $M > N$, then the columns of Φ and $\tilde{\Phi}$ are mutually dual frames.

Orthogonal and biorthogonal filter banks

The perfect reconstruction $\mathbf{s} = \hat{\mathbf{s}}$ corresponds to the condition

$$\Phi\tilde{\Phi}^* = I. \tag{3.8}$$

Consider a perfect-reconstruction, critically sampled filter bank in Fig. 3.1 that satisfies (3.8). If the corresponding matrix Φ is orthogonal, that is $\Phi = \tilde{\Phi}$, then the filter banks is called *orthogonal*, since the columns of Φ form an orthonormal basis. Otherwise, it is called *biorthogonal*.

Example 3.1.2 *Consider a two-channel, critically-sampled filter bank with analysis filters*

$$\tilde{h}_0(x) = \frac{1+x^{-1}}{\sqrt{2}}, \qquad \tilde{h}_1(x) = \frac{1-x^{-1}}{\sqrt{2}}$$

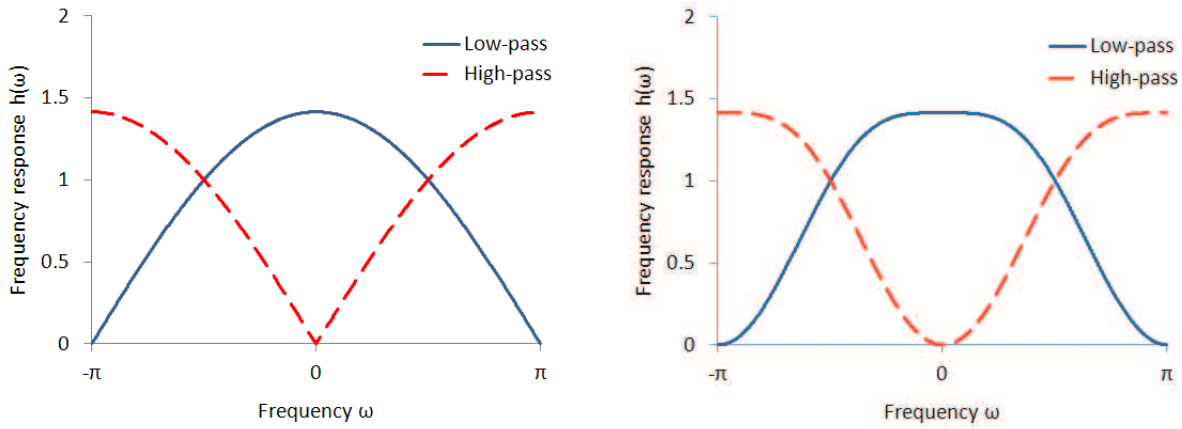
and synthesis filters

$$h_0(x) = \frac{1+x}{\sqrt{2}}, \qquad h_1(x) = \frac{1-x}{\sqrt{2}}.$$

The frequency responses of analysis low-pass and high-pass filters $\tilde{h}_0(x)$ and $\tilde{h}_1(x)$ are shown in Fig. 3.3(a). This filter bank is known as Haar *filter bank [23, 49].*

We can verify that the analysis and synthesis filters satisfy the perfect reconstruction condition (3.4)*:*

$$\begin{aligned}
\tilde{h}_0(x)h_0(x) + \tilde{h}_1(x)h_1(x) &= \frac{1+x^{-1}}{\sqrt{2}} \cdot \frac{1+x}{\sqrt{2}} + \frac{1-x^{-1}}{\sqrt{2}} \cdot \frac{1-x}{\sqrt{2}} = 2, \\
\tilde{h}_0(-x)h_0(x) + \tilde{h}_1(-x)h_1(x) &= \frac{1-x^{-1}}{\sqrt{2}} \cdot \frac{1+x}{\sqrt{2}} + \frac{1+x^{-1}}{\sqrt{2}} \cdot \frac{1-x}{\sqrt{2}} = 0.
\end{aligned}$$



(a) Haar (flatness degree 1).

(b) Daubechies D_4 (flatness degree 2).

Figure 3.3: Frequency responses of low-pass and high-pass filters for Haar and Daubechies D_4 filter banks.

Alternatively, we can verify that the matrix Φ *in* (3.6) *is orthogonal, since*

$$\Phi = \begin{pmatrix} \ddots & & & \\ & \Phi_0 & & \\ & & \Phi_0 & \\ & & & \ddots \end{pmatrix},$$

where

$$\Phi_0 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Hence, Haar filter bank is an orthogonal filter bank.

Example 3.1.3 *Consider a two-channel, critically-sampled filter bank with analysis filters*

$$\begin{aligned} \tilde{h}_0(x) &= \frac{1+\sqrt{3}}{4\sqrt{2}} + \frac{3+\sqrt{3}}{4\sqrt{2}}x^{-1} + \frac{3-\sqrt{3}}{4\sqrt{2}}x^{-2} + \frac{1-\sqrt{3}}{4\sqrt{2}}x^{-3}, \\ \tilde{h}_1(x) &= \frac{1-\sqrt{3}}{4\sqrt{2}} + \frac{\sqrt{3}-3}{4\sqrt{2}}x^{-1} + \frac{3+\sqrt{3}}{4\sqrt{2}}x^{-2} + \frac{-1-\sqrt{3}}{4\sqrt{2}}x^{-3} \end{aligned}$$

and synthesis filters

$$\begin{aligned} h_0(x) &= \frac{1+\sqrt{3}}{4\sqrt{2}} + \frac{3+\sqrt{3}}{4\sqrt{2}}x + \frac{3-\sqrt{3}}{4\sqrt{2}}x^2 + \frac{1-\sqrt{3}}{4\sqrt{2}}x^3, \\ h_1(x) &= \frac{1-\sqrt{3}}{4\sqrt{2}} + \frac{\sqrt{3}-3}{4\sqrt{2}}x + \frac{3+\sqrt{3}}{4\sqrt{2}}x^2 + \frac{-1-\sqrt{3}}{4\sqrt{2}}x^3. \end{aligned}$$

The frequency responses of analysis low-pass and high-pass filters $\tilde{h}_0(x)$ and $\tilde{h}_1(x)$ are shown in Fig. 3.3(b). This filter bank is known as Daubechies D_4 *filter bank [50,51].*

We can verify that the analysis and synthesis filters satisfy the perfect reconstruction condition (3.4)*:*

$$\begin{aligned} \tilde{h}_0(x)h_0(x) + \tilde{h}_1(x)h_1(x) &= 2, \\ \tilde{h}_0(-x)h_0(x) + \tilde{h}_1(-x)h_1(x) &= 0. \end{aligned}$$

Alternatively, we can verify that the matrix Φ in (3.6) *is orthogonal, since*

$$\Phi = \begin{pmatrix} \ddots & & & & \\ \ddots & \Phi_0 & & & \\ & \Phi_1 & \Phi_0 & & \\ & & \Phi_1 & \Phi_0 & \\ & & & \ddots & \ddots \end{pmatrix},$$

where

$$\Phi_0 = \frac{1}{4\sqrt{2}} \begin{pmatrix} 1+\sqrt{3} & 1-\sqrt{3} \\ 3+\sqrt{3} & \sqrt{3}-3 \end{pmatrix}$$

and

$$\Phi_1 = \frac{1}{4\sqrt{2}} \begin{pmatrix} 3-\sqrt{3} & 3+\sqrt{3} \\ 1-\sqrt{3} & -1-\sqrt{3} \end{pmatrix}.$$

Hence, Daubechies D_4 filter bank is an orthogonal filter bank.

3.1.5 Applications of Filter Banks

The characteristics and design of perfect-reconstruction filter banks can vary significantly depending on the application for which a filter bank is constructed. In this thesis, we consider two potential applications of perfect-reconstruction filter banks—low/high-frequency separation, and robust transmission of signals.

Frequency separation

Consider a two-channel critically-sampled filter bank that is designed to separate low and high frequencies of input signals. Ideally, the low-pass analysis filter $\tilde{h}_0(x)$ would have the frequency response

$$\tilde{H}_0(\omega) = \begin{cases} 1, & \text{if} \quad |\omega| \leq \pi/2, \\ 0, & \text{if} \quad \pi/2 < |\omega| < \pi. \end{cases}$$

Similarly, the frequency response of an ideal high-pass filter $\tilde{h}_1(x)$ would be

$$\tilde{H}_1(\omega) = \begin{cases} 0, & \text{if} \quad |\omega| \leq \pi/2, \\ 1, & \text{if} \quad \pi/2 < |\omega| < \pi. \end{cases}$$

Unfortunately, in this case the analysis filters $\tilde{h}_0(x)$ and $\tilde{h}_1(x)$ would have to be IIR filters, which makes their implementation both impractical and impossible, since they cannot be implemented recursively. For practical reasons, FIR filters can be used to approximate the behavior of the ideal IIR filters. One of the most widely used characteristics of such FIR filters is the degree of flatness.

A *degree of flatness* of a low-pass filter is specified by the number of derivatives of its frequency response that vanish at the highest frequency (respectively, lowest frequency for a high-pass filter) [24, 46]. For example, a low-pass filter $h(x)$ with M degrees of flatness satisfies the condition

$$\frac{d^m}{d\omega^m} H(\omega)\Big|_{\omega=\pi} = 0$$

for $0 \leq m < M$, since $\omega = \pi$ corresponds to the highest frequency in the infinite time model. Equivalently, $(1 + e^{-j\omega})^M$ must divide $H(\omega)$. A low-pass filter with a greater degree of flatness is a better approximation

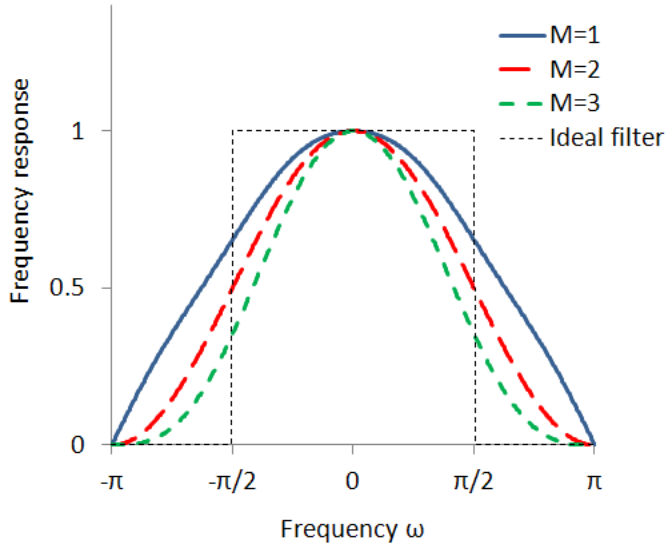


Figure 3.4: Frequency response of a low-pass filter $((1+e^{-j\omega})/2)^M$ for $M = 1, 2, 3$.

of an ideal low-pass filter, as illustrated in Fig. 3.4.

Among all filters with the same degree of flatness, the one with the shortest support length are called *maximally flat* or *maflat*. Both orthogonal and biorthogonal filter banks with maxflat filters have been constructed for the infinite discrete time signals in [50–54].

Example 3.1.4 *Consider the Haar filter bank introduced in Example 3.1.2. It uses maxflat low- and high-pass filters of order 1, since*

$$\begin{aligned} \tilde{H}_0(\pi) &= \frac{1}{\sqrt{2}}(1+e^{j\pi}) = 0, \\ \tilde{H}_1(0) &= \frac{1}{\sqrt{2}}(1-e^{j0}) = 0, \end{aligned}$$

and they have the shortest possible support length 2.

Similarly, the Daubechies D_4 filter bank introduced in Example 3.1.3 uses maxflat low- and high-pass

filters of order 2, since

$$\tilde{H}_0(\pi) = 0, \qquad \frac{d}{d\omega}\tilde{H}_0(\omega)\Big|_{\omega=\pi} = 0,$$
$$\tilde{H}_1(0) = 0, \qquad \frac{d}{d\omega}\tilde{H}_1(\omega)|_{\omega=0} = 0,$$

and they have the shortest possible support length 4.

Robust signal transmission

Consider an M-channel oversampled perfect-reconstruction filter bank. As we explained in Section 3.1.4, in this case the columns of matrices Φ and $\tilde{\Phi}$ form mutually dual frames; or in the case $\Phi = \tilde{\Phi}$, a tight frame.

An advantage of frames in comparison to bases is their redundancy. After a signal is expanded into a frame, a loss of several expansion coefficients $\langle \varphi_k^{(m)}, \mathbf{s} \rangle$ in (3.5) may not prevent us from reconstructing the original signal from the remaining coefficients. In the most extreme case, for an M-channel oversampled perfect-reconstruction filter bank that is sampled at rate $N < M$, we should be able to reconstruct the original signal from any N/M fraction of its coefficients. Frames that satisfy this requirement are called *maximally robust to erasures*. They have been studied, in particular, in [44, 45, 47, 48]. Filter banks that correspond to maximally robust to erasures frames can be used for robust storage and transmission of signals over lossy channels.

3.2 Polynomial Transforms

As we discussed in Section 2.3, the generalized discrete Fourier transform can be computed with the corresponding polynomial transform (2.20). If $n = \deg p(x)$ in (2.18), then $\mathcal{P}_{b,\alpha} \in \mathbb{C}^{n\times n}$ is an $n \times n$ matrix. The computation of the generalized discrete Fourier transform as the matrix-vector product (2.21) requires, in general, $O(n^2)$ operations. This cost may become prohibitive for large n in some practical applications. It brings up the main challenge in the implementation of polynomial transforms: how to construct fast algorithms for the transforms of interest.

Here, we briefly review the developments in the theory of fast algorithms for polynomial transforms. We discuss the construction of fast algorithms for the most well-known and widely used transforms, such as DFT and discrete cosine and sine transforms (DCTs and DSTs). We also review existing fast algorithms for other, less well-known polynomial transforms.

3.2.1 Known Fast Algorithms

Fast Fourier transforms

Over the last decades, fast algorithms have been studied only for a small number of polynomial transforms. Among them, the DFT is arguably the most famous and well-studied transform. The discovery of the Cooley-Tukey fast Fourier transform (FFT) algorithm [55], which reduced the computation cost of DFT_n to $O(n \log n)$ operations, led to decades of research and multiple new FFT algorithms (see [6, 20, 35, 56–59] and references therein).

The majority of FFT algorithms can be interpreted as factorizations of the DFT matrix into a product of sparse matrices. The goal is to find such a factorization that the combined computational costs of these matrices is lower than the cost of a straightforward computation of DFT. The following example demonstrates this.

Example 3.2.1 *The Cooley-Tukey FFT algorithm for* DFT_4 *corresponds to the following matrix decomposition:*

$$\begin{aligned}
\mathrm{DFT}_4 &= \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix} \\
&= \begin{pmatrix} 1 & & & \\ & & 1 & \\ & 1 & & \\ & & & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & & \\ 1 & -1 & & \\ & & 1 & 1 \\ & & 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & -j \end{pmatrix} \begin{pmatrix} 1 & & 1 & \\ & 1 & & 1 \\ 1 & & -1 & \\ & 1 & & -1 \end{pmatrix}.
\end{aligned}$$

The direct computation of DFT_4 *requires 12 operations (multiplications by* ± 1 *and* $\pm i$ *are not considered). In comparison, the computation of the four matrices in the factorization requires only 8 operations.*

In addition, fast algorithms exist for DFT-related signal transforms, including the real DFT and the disrete Hartley transform, for example [6, 60, 61].

Fast trigonometric transforms

Another well-known class of signal transforms is the class of discrete trigonometric transforms—discrete cosine and sine transforms (DCT and DST) of types 1 through 8. Similarly to DFT, fast algorithms for trigonometric transforms reduce the computational cost from $O(n^2)$ to $O(n \log n)$ operations (for example, see [2, 6, 62–71], as well as references therein). Most of these algorithms can also be interpreted as factorizations of a transform matrix into a product of sparse matrices, as demonstrated by the following example.

Example 3.2.2 *The Wang fast algorithm for* $\mathrm{DCT\text{-}IV}_4$ *[66] corresponds to the following matrix decomposition:*

$$
\begin{aligned}
\mathrm{DCT\text{-}IV}_4 &= \begin{pmatrix} 0.9808 & 0.8315 & 0.5556 & 0.1951 \\ 0.8315 & -0.1951 & -0.9808 & -0.5556 \\ 0.5556 & -0.9808 & 0.1951 & 0.8315 \\ 0.1951 & -0.5556 & 0.8315 & -0.9808 \end{pmatrix} \\
&= \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \\ & 1 & & \end{pmatrix} \begin{pmatrix} 0.6935 & 0.1379 & & \\ 0.1379 & -0.6935 & & \\ & & 0.5879 & -0.3928 \\ & & 0.3928 & 0.5879 \end{pmatrix} \\
&\quad \times \begin{pmatrix} 1 & & 1 & \\ & 1 & & 1 \\ 1 & & -1 & \\ & 1 & & -1 \end{pmatrix} \begin{pmatrix} \sqrt{2} & & & \\ & & & \sqrt{2} \\ & 1 & -1 & \\ & 1 & -1 & \end{pmatrix}.
\end{aligned}
$$

The direct computation of $\mathrm{DCT\text{-}IV}_4$ *requires 28 operations. In comparison, the computation of the four matrices in the factorization requires only 20 operations.*

Polynomial transforms for orthogonal polynomials

Another class of transforms that have been studied are polynomial transforms that are based on orthogonal polynomials [72–75]. We introduce orthogonal polynomials and discuss their properties in Appendix A.

With the exception of DCTs and DSTs, which belong to this group of transforms as well, the fast algorithms reported in the literature require $O(n \log^2 n)$ operations. They include:

a) Algorithms for transforms evaluated at arbitrary nodes α_k [74]. However, the exact cost of these algorithms is greater than $36n \log_2^2 n$, which renders them impractical. Moreover, these algorithms are not numerically stable.

b) Algorithms for transforms evaluated at the roots α_k of Chebyshev polynomials $T_n(x)$ [75]. These algorithms require greater than $4n \log_2^2 n$ operations, and are not stable in general.

c) An approximation algorithm for the polynomial transform for Legendre polynomials evaluated at the roots α_k of Legendre polynomial $P_n(x)$ [72]. Evaluation to the precision ε requires $O(n \log \frac{1}{\varepsilon})$ operations.

In general, reported fast algorithms use the recursion (A.1) for orthogonal polynomials to iteratively compute the value $P_\ell(\alpha_k)$ from $P_{\ell-1}(\alpha_k)$ and $P_{\ell-2}(\alpha_k)$. They do not use any information about the points α_k, which, understandably, leads to slower algorithms. As we demonstrate in Section 6, incorporating information about sampling points allows us to construct fast algorithms for polynomial transforms based on orthogonal polynomials that require $O(n \log n)$ operations.

3.2.2 Algebraic Approach

Most fast algorithms for polynomial transforms, including those discussed in Section 3.2.1, are derived by clever and often complicated manipulations of matrix coefficients. They provide little insight into why such decompositions exist and how to extend and generalize them.

Several algebraic approaches have been developed to explain the derivation of known fast algorithms and construct new ones. Originally, group algebras were utilized to explain the Cooley-Tukey FFT. For example, DFT_n can be interpreted as a decomposition matrix for the group algebra $\mathbb{C}[\mathbb{Z}_n]$, where $\mathbb{Z}_n$ is a cyclic group of order n [76, 77]. Since $\mathbb{C}[\mathbb{Z}_n]$ is isomorphic to the polynomial algebra $\mathbb{C}[x]/(x^n - 1)$, it can be decomposed as

$$\mathbb{C}[\mathbb{Z}_n] \cong C[x]/(x^n - 1) \to \mathbb{C}[x]/(x - \omega_n^0) \oplus \cdots \oplus \mathbb{C}[x]/(x - \omega_n^{n-1}), \tag{3.9}$$

as we showed in Section 2.3. This decomposition can be performed stepwise: first, we decompose $\mathbb{C}[x]/(x^n - 1)$ into a direct sum of subalgebras, not necessarily simple ones; then, we further decompose each subalgebra. Performed in a very particular way, such stepwise decomposition yields the Cooley-Tukey FFT.

The group point of view was generalized to derive fast Fourier transforms for group algebras $\mathbb{C}[G]$ for noncyclic finite groups G [78–84]. Some of the constructed algorithms were based on the induction for group algebras, an algebraic construction that is analogous to the method we use in Chapter 6.

In parallel to group algebras, polynomial algebras were also used in [20, 85–89] to construct fast algorithms for the DFT. Extension of this approach led to a comprehensive theory of fast algorithms for all 16 types of DCTs and DSTs [2, 6], as well as for complex and real DFT [61]. This generalization led to the development of the ASP theory [3–5] discussed in Chapter 2.

3.2.3 Orthogonality

Orthogonality is an important and desired property of a polynomial transform, since, in general, the inverse of a polynomial transform is not a polynomial transform itself. In this case, constructing a fast algorithm for the inverse becomes a challenging task.

The well-known polynomial transforms (DFT, DCT, and DST) can be made orthogonal by multiplying them with diagonal matrices. Trivially, the fast implementation of their inverses reduces to transposing the original algorithm. As we discuss in Section 4.3, the polynomial transforms for finite GNN models can also be orthogonalized this way.

Chapter 4

Generic Nearest-Neighbor Models

In this chapter we construct a class of new linear shift-invariant signal models called the *generic nearest-neighbor* (GNN) models. Originally, these models were introduced in [3]. For the finite model, the associated discrete Fourier transform and spectrum were identified.

Here, we formally define both infinite and finite discrete GNN models. In both cases, we identify the corresponding signal processing concepts including spectrum, Fourier transform, convolution, and Parseval equality.

The infinite and finite space models discussed in Chapter 2 are particular cases of the infinite and finite GNN models that use Chebyshev polynomials as basis functions. Nevertheless, we sometimes refer to these models for comparison, since they have already been extensively studied in [1–3, 5, 6, 22].

4.1 Normalized Orthogonal Polynomials

Before we define the infinite and finite discrete GNN models, we introduce the concept of normalized orthogonal polynomials.

Consider the shift defined by the recurrence

$$x \cdot P_k(x) = a_{k-1}P_{k-1}(x) + b_k P_k(x) + c_k P_{k+1}(x). \tag{4.1}$$

We call it the *generic nearest-neighbor* shift. If we assume that the coefficients $a_k, b_k, c_k \in \mathbb{R}$ satisfy the

condition $a_k c_k > 0$ for $k \geq 0$, and $P_0(x) = 1$ and $P_{-1}(x) = 0$, then the solution to the recurrence (4.1) is a family $\big(P_k(x)\big)_{k\geq 0}$ of *orthogonal polynomials* [16, 17]. An overview of orthogonal polynomials and their properties is provided in Appendix A.

We assume that polynomials $\big(P_k(x)\big)_{k\geq 0}$ are orthogonal over the interval $W \subseteq \mathbb{R}$ with the weight function function $\mu(x)$:

$$\int_{x\in W} P_k(x) P_m(x) \mu(x) dx = \mu_k \delta_{k-m}.$$

Furthermore, we assume that the $\mathcal{L}^2_\mu$-norm of $P_k(x)$ is

$$||P_k(x)||_{2,\mu} = \Big(\langle P_k(x), P_k(x)\rangle_\mu\Big)^{1/2} = \mu_k^{1/2},$$

where the norm is induced by the inner product

$$\langle f(x), g(x)\rangle_\mu = \int_{x\in W} f(x) g(x) \mu(x) dx.$$

Norm calculation

In order to calculate the norm $||P_k(x)||_{2,\mu} = \mu_k^{1/2}$, we need to know the weight function $\mu(x)$ and the orthogonality interval W. However, as we discuss in Appendix A, it may not be feasible to obtain $\mu(x)$ and W directly from the recursion (4.1).

Nevertheless, we can bypass the necessity to determine $\mu(x)$, and determine the norms of $P_k(x)$ from the coefficients a_k, b_k, and c_k in recursion (4.1). This is a known result (see, for example, [90, 91]), and we only provide it here for completeness.

Theorem 4.1.1 *The $\mathcal{L}^2_\mu$-norm of the polynomials $P_k(x)$ that satisfy* (4.1) *and are orthogonal on W with respect to the inner product* (A.3)*, is*

$$||P_n(x)||_{2,\mu} = \mu_n^{1/2} = \mu_0^{1/2} \cdot \sqrt{\prod_{i=0}^{n-1} \frac{a_i}{c_i}}. \tag{4.2}$$

Proof: Let

$$D = \operatorname{diag}\left(\sqrt{\prod_{i=0}^{k-1} \frac{a_i}{c_i}}\right)_{0 \le k < n}$$

be an $n \times n$ diagonal matrix, and

$$S = \begin{pmatrix} b_0 & a_0 & & & \\ c_0 & b_1 & a_1 & & \\ & c_1 & b_2 & \ddots & \\ & & \ddots & \ddots & a_{n-2} \\ & & & c_{n-2} & b_{n-1} \end{pmatrix}$$

be the tridiagonal matrix defined in (A.4). Then D conjugates S to the symmetric tridiagonal matrix

$$DSD^{-1} = \begin{pmatrix} b_0 & \sqrt{a_0 c_0} & & & \\ \sqrt{a_0 c_0} & b_1 & \sqrt{a_1 c_1} & & \\ & \sqrt{a_1 c_1} & b_2 & \ddots & \\ & & \ddots & \ddots & \sqrt{a_{n-2} c_{n-2}} \\ & & & \sqrt{a_{n-2} c_{n-2}} & b_{n-1} \end{pmatrix}.$$

On the other hand, using the Christoffel-Darboux formula it was shown in [3] that the diagonal matrix

$$E = \operatorname{diag}\left(\mu_0^{1/2}, \mu_1^{1/2}, \ldots, \mu_{n-1}^{1/2}\right)$$

also conjugates S to a symmetric tridiagonal matrix.

Since there exists a unique (up to a constant factor) diagonal matrix that conjugates a tridiagonal matrix to a symmetric tridiagonal matrix, we conclude that $D = cE$, and hence

$$\mu_k^{1/2} = c \cdot \sqrt{\prod_{i=0}^{k-1} \frac{a_i}{c_i}}$$

for some non-zero constant $c \in \mathbb{R}$. In particular, for $k = 0$ we obtain

$$c = \mu_0^{1/2},$$

where

$$\mu_0 = \int_{x \in W} \mu(x) dx.$$

□

The following result is an immediate consequence of Theorem 4.1.1:

Corollary 4.1.2 *If $a_k = c_k$ for $k \geq 0$, then $\mu_k = \mu_0$, and all $P_k(x)$ have the same norm*

$$||P_k(x)||_{2,\mu} = \sqrt{\mu_0}.$$

Normalization

In general, orthogonal polynomials $P_k(x)$ have different norms: $\mu_k \neq \mu_m$ for $k \neq m$. They can be normalized as $\mu_k^{-1/2} P_k(x)$ to have the same norm 1 for all $k \geq 0$.

To simplify the construction of the infinite and finite discrete GNN signal models in Sections 4.2 and 4.3, we use the normalized polynomials $\mu_k^{-1/2} P_k(x)$ as the basis of the signal module $\mathcal{M}$. Since any family of orthogonal polynomials can be orthogonalized, hereafter we only consider the families of orthogonal polynomials $\big(P_k(x)\big)_{k \geq 0}$ that have equal norms.

The following theorem establishes which families $\big(P_k(x)\big)_{k \geq 0}$ have equal norms, and shows how to construct normalized polynomials for an arbitrary family.

Theorem 4.1.3 *The orthogonal polynomials $P_k(x)$ have the same norm $||P_k(x)||_{2,\mu} = ||P_0(x)||_{2,\mu}$, if they satisfy the recursion of the form*

$$x \cdot P_k(x) = a_{k-1} P_{k-1}(x) + b_k P_k(x) + a_k P_{k+1}(x), \tag{4.3}$$

with $P_0(x) = 1$ and $P_{-1}(x) = 0$. That is, the recurrence coefficients in (4.1) *satisfy the symmetric condition*

$$a_k = c_k$$

for all $k \geq 0$.

Furthermore, if the family $\big(P_k(x)\big)_{k\geq 0}$ is a solution to (4.1)*, then the corresponding normalized polynomials $Q_k(x) = \mu_k^{-1/2} P_k(x)$ are a solution to the recurrence*

$$x \cdot p_k(x) = \sqrt{a_{k-1}c_{k-1}}Q_{k-1}(x) + b_k Q_k(x) + \sqrt{a_k c_k}Q_{k+1}(x). \tag{4.4}$$

Proof: Recall from Appendix A that

$$\mu_k = \mu_0 \cdot \prod_{i=0}^{k-1} \frac{a_i}{c_i}.$$

If $a_k = c_k$ for all $k \geq 0$, then $\mu_k = \mu_0$, and hence $||P_k(x)||_{2,\mu} = ||P_0(x)||_{2,\mu}$.

Next, observe that

$$\mu_{k+1}/\mu_k = a_k/c_k.$$

Then the family of normalized polynomials $Q_k(x) = \mu_k^{-1/2} P_k(x)$ satisfies the recurrence

$$\begin{aligned} x \cdot Q_k(x) &= x \cdot \mu_k^{-1/2} P_k(x) \\ &= \mu_k^{-1/2}\Big(a_{k-1}P_{k-1}(x) + b_k P_k(x) + c_k P_{k+1}(x)\Big) \\ &= a_{k-1}\frac{\mu_{k-1}}{\mu_k}Q_{k-1}(x) + b_k Q_k(x) + c_k \frac{\mu_{k+1}}{\mu_k}Q_{k+1}(x) \\ &= \sqrt{a_{k-1}c_{k-1}}Q_{k-1}(x) + b_k Q_k(x) + \sqrt{a_k c_k}Q_{k+1}(x). \end{aligned}$$

□

(a) GNN shift (b) GNN model visualization

Figure 4.1: Infinite discrete GNN signal model: shift and visualization.

4.2 Infinite Discrete Model

Signal model

Hereafter, we only consider the GNN models that are based on the *symmetric* GNN shift (4.3). As discussed in Section 4.1, the basis of the signal module $\mathcal{M}$ is defined by the corresponding equal-norm orthogonal polynomials $P_k(x)$.

The basis of the filter algebra, ideally, would be defined by ℓ-fold shifts, as in the case of time (2.9) or space (2.16) models. Unfortunately, for a general family of orthogonal polynomials there is no obvious notion of an ℓ-fold shift, so it is unclear which basis is appropriate for the filter algebra $\mathcal{A}$. We choose the same basis as in the signal module $\mathcal{M}$.

Hence, we obtain the following *infinite discrete GNN signal model*:

$$\begin{aligned} &\mathcal{A} = \{h = \textstyle\sum_{\ell\geq 0} h_\ell P_\ell(x) \mid \mathbf{h} = \big(h_0, h_1, \dots\big) \in \ell^1(\mathbb{N}_0)\}, \\ &\mathcal{M} = \{s = \textstyle\sum_{k\geq 0} s_k P_k(x) \mid \mathbf{s} = \big(s_0, s_1, \dots\big) \in \ell^2(\mathbb{N}_0)\}, \\ &\Phi:\ \ell^2(\mathbb{N}_0) \to \mathcal{M},\ \mathbf{s} \mapsto \textstyle\sum_{k\geq 0} s_k P_k(x). \end{aligned} \tag{4.5}$$

The symmetric GNN shift (4.3) and the visualization of the infinite discrete GNN signal model (4.5) are shown in Fig. 4.1.

Eigenfunctions

As discussed in Section 2.2, to construct the associated Fourier transform for the GNN model, we need to find the eigenfunctions of $\mathcal{M}$, identify the spectrum W, and project signals on the spectral components

spanned by the eigenfunctions.

The following theorem identifies the structure of eigenfunctions:

Theorem 4.2.1 *The eigenfunctions for the infinite discrete GNN model have the form*

$$E_a(x) = \sum_{k\geq 0} P_k(a)P_k(x).$$

In particular, they satisfy

$$x \cdot E_a(x) = a \cdot E_a(x) \tag{4.6}$$

for any $a \in \mathbb{R}$.

Proof: As follows from (4.3), any function of the form $\sum_{k\geq 0} s_k P_k(x) \in \mathcal{M}$ satisfies the condition

$$x \cdot \sum_{k\geq 0} s_k P_k(x) = \sum_{k\geq 0} \Big(a_{k-1}s_{k-1} + b_k s_k + a_k s_{k+1}\Big) P_k(x). \tag{4.7}$$

Consider the function

$$E_a(x) = \sum_{k\geq 0} P_k(a)P_k(x).$$

From (4.7) we obtain

$$\begin{aligned} x \cdot \sum_{k\geq 0} P_k(a)P_k(x) &= \sum_{k\geq 0} \Big(a_{k-1}P_{k-1}(a) + b_k P_k(a) + a_k P_{k+1}(a)\Big) P_k(x) \\ &= \sum_{k\geq 0} aP_k(a)P_k(x) \\ &= a \cdot \sum_{k\geq 0} P_k(a)P_k(x). \end{aligned}$$

□

Spectrum

The eigenfunctions in Theorem 4.2.1 satisfy (4.6) for all $a \in \mathbb{R}$. However, we restrict the values of a to the interval of orthogonality W only. This restriction is sufficient to make the Fourier transform invertible, as

we will show later.

Hence, the spectrum is defined precisely as the interval of orthogonality W.

Fourier transform

Since the Fourier transform is a projection onto the eigenfunctions $E_\omega(x)$, where $\omega \in W$, we define the *GNN Fourier transform* as

$$\Delta: \qquad S(\omega) = \langle s(x), E_\omega(x) \rangle = \sum_{k \geq 0} s_k P_k(\omega). \tag{4.8}$$

Given that we have restricted the spectrum to the interval of orthogonality W, the corresponding *inverse GNN Fourier transform* is

$$\Delta^{-1}: \qquad s_k = \frac{1}{\mu_0} \int_{\omega \in W} S(\omega) P_k(\omega) \mu(\omega) d\omega, \tag{4.9}$$

since

$$\begin{aligned}
\frac{1}{\mu_0} \int_{\omega \in W} S(\omega) P_k(\omega) \mu(\omega) d\omega &= \frac{1}{\mu_0} \int_{\omega \in W} \Big(\sum_{m \geq 0} s_m P_m(\omega) \Big) P_k(\omega) \mu(\omega) d\omega \\
&= \frac{1}{\mu_0} \sum_{m \geq 0} s_m \Big(\int_{\omega \in W} P_m(\omega) P_k(\omega) \mu(\omega) d\omega \Big) \\
&= \frac{1}{\mu_0} s_k \mu_0 \\
&= s_k.
\end{aligned}$$

Frequency response

From (4.6) we obtain

$$h(x) E_\omega(x) = h(\omega) E_\omega(x)$$

for any polynomial $h(x) \in \mathcal{A}$. Hence, the frequency response $H(\omega)$ of a filter $h(x)$ at frequency ω is simply its evaluation at this frequency, similarly to the time and space signal models:

$$H(\omega) = \sum_{\ell \geq 0} h_\ell P_\ell(\omega). \tag{4.10}$$

Convolution

The convolution in the signal domain is defined by the action of a filter $h = h(x) \in \mathcal{A}$ on a signal $s = s(x) \in \mathcal{M}$. ASP defines the convolution as the product

$$hs = h(x)s(x).$$

Similarly to the infinite time and space models (2.9) and (2.16), it follows from the definitions of the Fourier transform (4.8) and the frequency response (4.10) that the convolution corresponds to the multiplication in the frequency domain. If $\hat{s} = \hat{s}(s) = h(x)s(x)$, then

$$\hat{S}(\omega) = H(\omega) \cdot S(\omega).$$

Expressing the convolution directly via the coefficients h_ℓ and s_k of **h** and **s** is a tedious task that yields a complicated formula. For computational purposes, we simply express the product $h(x)s(x)$ in the basis $\big(P_k(x)\big)_{k\geq 0}$; the coefficients of the expansion are precisely $\hat{s}_k = (\mathbf{h} * \mathbf{s})_k$:

$$\sum_{\ell \geq 0} h_\ell P_\ell(x) \cdot \sum_{k \geq 0} s_k P_k(x) = \sum_{k \geq 0} (\mathbf{h} * \mathbf{s})_k P_k(x).$$

Respectively, the convolution in the frequency domain corresponds to the pointwise multiplication in the signal domain. Hence,

$$H(\omega) * S(\omega) = \frac{1}{\mu_0} \int_{\theta \in W} S(\theta) G_\omega(\theta) \mu(\theta) d\theta, \tag{4.11}$$

since

$$\begin{aligned} H(\omega) * S(\omega) &= \sum_{k\geq 0} h_k s_k P_k(\omega) \\ &= \sum_{k\geq 0} h_k P_k(\omega) \cdot \frac{1}{\mu_0} \int_{\theta\in W} S(\theta) P_k(\theta) \mu(\theta) d\theta \\ &= \frac{1}{\mu_0} \int_{\theta\in W} S(\theta) \Big(\sum_{k\geq 0} h_k P_k(\omega) P_k(\theta) \Big) \mu(\theta) d\theta \\ &= \Big[\text{let } g_{\omega,k} = h_k P_k(\omega) \Big] \\ &= \frac{1}{\mu_0} \int_{\theta\in W} S(\theta) G_\omega(\theta) \mu(\theta) d\theta. \end{aligned}$$

Parseval equality

The following theorem establishes the Parseval equality for infinite discrete GNN signal models.

Theorem 4.2.2 *The Parseval equality for the 1-D infinite discrete GNN signal model is*

$$\sum_{k\geq 0} s_k^2 = \frac{1}{\mu_0} \int_{\omega\in W} S^2(\omega) \mu(\omega) d\omega. \tag{4.12}$$

Proof: Observe that

$$\begin{aligned} \sum_{k\geq 0} s_k^2 &= \sum_{k\geq 0} s_k \frac{1}{\mu_0} \int_{\omega\in W} S(\omega) P_k(\omega) \mu(\omega) d\omega \\ &= \frac{1}{\mu_0} \int_{\omega\in W} S(\omega) \Big(\sum_{k\geq 0} s_k P_k(\omega) \Big) \mu(\omega) d\omega \\ &= \frac{1}{\mu_0} \int_{\omega\in W} S^2(\omega) \mu(\omega) d\omega. \end{aligned}$$

□

Frequency domain

The frequency domain for the infinite discrete GNN model is the Hilbert space of all polynomials defined on interval W [92, 93]. The inner product is defined as

$$\langle u, v\rangle = \int_{\omega\in W} u(\omega)v(\omega)\mu(\omega)d\omega. \tag{4.13}$$

The set of polynomials $\big(P_k(\omega)\big)_{k\geq 0}$ is an orthogonal basis of this space.

Observe that the GNN Fourier transform is an isomorphism between the signal space

$$\ell^2(\mathbb{N}_0) = \{\big(s_0, s_1, \dots\big) | \sum_{k\geq 0} s_k^2 < \infty\}$$

of semi-infinite sequences with finite energy, and the Hilbert space of polynomials defined on interval W with inner product (4.13). In cases when $W = [0, \infty)$ or $W = (-\infty, \infty)$ is an unbounded interval, the existence of such isomorphism may seem counter-intuitive. It becomes clear, however, if we recall from Appendix A that in such cases weight function $\mu(\omega)$ decays very quickly. In particular, the decay rate of $\mu(\omega)$ is higher than polynomial. Hence, for practical purposes, the interval of orthogonality can be viewed as finite.

Other infinite GNN models

The infinite discrete GNN model (4.5) can be generalized by allowing other left boundary conditions. All corresponding signal processing concepts derived in this section can be easily generalized for these models.

Consider the recurrence (4.1). If instead of the zero boundary condition $P_{-1}(x) = 0$, we assume

$$P_{-1}(x) = bP_0(x)$$

for any $b \in \mathbb{R}$, then the recurrence remains the same except we use the coefficient $b_0 + b$ instead of b_0. The solution to the new recurrence is a family of orthogonal polynomials as well. Hence, we can construct an infinite discrete GNN model based on the GNN shift defined by the new recurrence. We then normalize it to obtain a new GNN model (4.5).

We can also assume

$$P_{-1}(x) = cP_1(x)$$

for any $c \in \mathbb{R}$, such that $a_0(c_0 + c) > 0$. The recurrence (4.1) will now have the coefficient $c_0 + c$ instead of c_0. The solution to the new recurrence is also a family of orthogonal polynomials, and we can construct an infinite discrete GNN model based on the GNN shift defined by the new recurrence.

Example 4.2.3 *Consider Hermite polynomials $H_k(x)$ introduced in Appendix A. They satisfy the following recursion:*

$$x \cdot H_k(x) = kH_{k-1}(x) + \frac{1}{2}H_{k+1}(x),$$

with $H_0(x) = 1$ and $H_1(x) = 2x$. They are orthogonal over the entire real line $\mathbb{R}$ with the weight function e^{-x^2}:

$$\int_{-\infty}^{\infty} H_k(x)H_m(x)e^{-x^2}dx = k!2^k\sqrt{\pi}.$$

Consider the normalized Hermite polynomials $\widehat{H}_k(x)$. As follows from Theorem 4.1.3, they have the form

$$\widehat{H}_k(x) = \frac{1}{\sqrt{k!2^k}}H_k(x),$$

and satisfy the recursion

$$x\widehat{H}_k(x) = \sqrt{\frac{k}{2}}\widehat{H}_{k-1}(x) + \sqrt{\frac{k+1}{2}}\widehat{H}_{k+1}(x), \tag{4.14}$$

with $\widehat{H}_0 = 1$ and $\widehat{H}_1 = \sqrt{2}x$. They are orthogonal over the entire real line $\mathbb{R}$ with respect to the weight function $\mu(x) = e^{-x^2}$:

$$\langle \widehat{H}_k(x), \widehat{H}_m(x) \rangle = \int_{\mathbb{R}} \widehat{H}_k(x)\widehat{H}_m(x)e^{-x^2}dx = \pi^{1/4}\delta_{k-m}.$$

The infinite discrete GNN signal model that corresponds to the normalized Hermite polynomials is

(a) Shift (b) Model visualization

Figure 4.2: Infinite discrete normalized Hermite signal model: shift and visualization.

$$\begin{aligned}
&\mathcal{A} = \{h = \textstyle\sum_{\ell\geq 0} h_\ell \widehat{H}_\ell(x) \mid \boldsymbol{h} = \big(h_0, h_1, \dots\big) \in \ell^1(\mathbb{N}_0)\},\\
&\mathcal{M} = \{s = \textstyle\sum_{k\geq 0} s_k \widehat{H}_k(x) \mid \boldsymbol{s} = \big(s_0, s_1, \dots\big) \in \ell^2(\mathbb{N}_0)\},\\
&\Phi:\ \ell^2(\mathbb{N}_0) \to \mathcal{M},\ \boldsymbol{s} \mapsto \textstyle\sum_{k\geq 0} s_k \widehat{H}_k(x).
\end{aligned} \tag{4.15}$$

The shift and model visualization are shown in Fig. 4.2.

The Fourier transform and its inverse are defined as

$$\begin{aligned}
\Delta:\quad & S(\omega) = \sum_{k\geq 0} s_k \widehat{H}_k(\omega),\\
\Delta^{-1}:\quad & s_k = \frac{1}{\pi^{1/4}} \int_{\omega\in\mathbb{R}} S(\omega)\widehat{H}_k(\omega) e^{-\omega^2} d\omega.
\end{aligned}$$

The frequency response is

$$H(\omega) = \sum_{\ell\geq 0} h_\ell \widehat{H}_\ell(\omega).$$

The Parseval equality is

$$\sum_{k\geq 0} s_k^2 = \frac{1}{\pi^{1/4}} \int_{\omega\in\mathbb{R}} S^2(\omega) e^{-\omega^2} d\omega.$$

4.3 Finite Discrete Model

Consider the signal model $\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/P_n(x)$ with basis $b = \big(P_0(x), P_1(x), \dots, P_{n-1}(x)\big)$, where the basis polynomials $P_k(x)$ satisfy the recursion (4.3). Recall from Appendix A that $P_n(x)$ has exactly n distinct real zeros $\alpha = \big(\alpha_0, \dots, \alpha_{n-1}\big)$, $0 \leq k < n$, and they all lie inside the interval W.

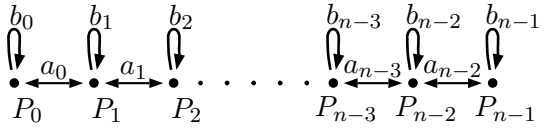


Figure 4.3: The visualization of the finite discrete GNN model $\mathbb{C}[x]/P_n(x)$ with basis $P_0(x), P_1(x), \ldots, P_{n-1}(x)$.

Signal model

We define the following *finite discrete GNN signal model*:

$$\begin{aligned} &\mathcal{A} = \{h = h(x) = \textstyle\sum_{0\le \ell<n} h_\ell P_\ell(x)\}, \\ &\mathcal{M} = \{s = s(x) = \textstyle\sum_{0\le k<n} s_k P_k(x)\}, \\ &\Phi:\ \mathbb{C}^n \to \mathcal{M},\ \mathbf{s} \mapsto s(x) = \textstyle\sum_{0\le k<n} s_k P_k(x). \end{aligned} \tag{4.16}$$

Observe that this model can be obtained from the corresponding infinite discrete model by imposing the boundary condition $P_n(x) = 0$. Fig. 4.3 visualizes the finite discrete GNN signal model. The boundary condition $P_n(x) = 0$ is indicated by the absence of an edge at the right boundary point.

Fourier transform

As we explained in Section 2.3, the Fourier transform for the model (4.16) has the form (2.19):

$$\begin{aligned} \Delta:\quad \mathbb{C}[x]/p(x) &\to \textstyle\bigoplus_{k=0}^{n-1} \mathbb{C}[x]/(x-\alpha_k), \\ s(x) &\mapsto \begin{pmatrix} s(\alpha_0) & s(\alpha_1) & \ldots & s(\alpha_{n-1}) \end{pmatrix}^T. \end{aligned}$$

It can be computed in matrix-vector form as

$$\Delta(s(x)) = \mathcal{P}_{b,\alpha} \cdot \mathbf{s},$$

where polynomial transform

$$\mathcal{P}_{b,\alpha} = \Big[P_\ell(\alpha_k)\Big]_{0\le k,\ell<n}. \tag{4.17}$$

is called the *discrete GNN Fourier transform*.

The transform $\mathcal{P}_{b,\alpha}$ in (4.17) can be easily orthogonalized. Namely,

$$\mathcal{P}_{b,\alpha}\mathcal{P}_{b,\alpha}^T = D,$$

where

$$D = a_{n-1} \cdot \operatorname{diag}\Big(P_{n-1}(\alpha_k)P_n'(\alpha_k)\Big)_{0\le k<n}. \tag{4.18}$$

Hence, the matrix

$$D^{-1/2}\mathcal{P}_{b,\alpha} \tag{4.19}$$

is orthogonal. It follows that the inverse of the discrete GNN Fourier transform (4.17) is

$$\mathcal{P}_{b,\alpha}^{-1} = \mathcal{P}_{b,\alpha}^T D^{-1}.$$

Spectrum

The spectrum of the finite discrete GNN model is the set

$$W = \alpha = \big(\alpha_0, \alpha_1, \dots, \alpha_{n-1}\big)$$

of zeros of $P_n(x)$.

Frequency response

The frequency response of a filter $h(x) \in \mathcal{A}$ is

$$H(\alpha) = \Big(h(\alpha_0) \quad h(\alpha_1) \quad \dots \quad h(\alpha_{n-1})\Big)^T.$$

Convolution

The convolution is defined as

$$hs = h(x)s(x) \mod P_n(x).$$

Frequency domain

The frequency domain of the finite discrete GNN model can be viewed as the frequency domain of the infinite discrete space model (2.16) sampled at frequencies $\omega_k = \alpha_k$ [93]. In particular, from (4.19) we observe that the function

$$\big(P_{n-1}(\omega_k)P'_n(\omega_k)P_m(\omega_k)\big)_{0\le k<n}$$

is the m-th basis function of an orthogonal basis in the frequency domain.

Basic shift representation matrix

The matrix representation $\phi(x)$ of the basic shift for the model (4.16) is

$$\phi(x) = \begin{pmatrix} b_0 & a_0 & & & \\ a_0 & b_1 & a_1 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-3} & b_{n-2} & a_{n-2} \\ & & & a_{n-2} & b_{n-1} \end{pmatrix}. \tag{4.20}$$

It follows from Section 2.3 and Appendix A that

$$\mathcal{P}_{b,\alpha}\phi(x)\mathcal{P}_{b,\alpha}^{-1} = \operatorname{diag}\Big(\alpha_0, \ldots, \alpha_{n-1}\Big).$$

Other finite GNN models

The finite discrete GNN model (4.16) can be generalized to the model $\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/P_n(x) - cP_{n-1}(x)$ for any $c \in \mathbb{R}$. This is possible since the polynomial $P_n(x) - cP_{n-1}(x)$ has n distinct zeros for any value of $c \in \mathbb{R}$ [94]. All concepts for the model (4.16) described above apply to this model as well.

Example 4.3.1 *The finite discrete GNN signal model that corresponds to the normalized Hermite polynomials $\widehat{H}_k(x)$, discussed in Example 4.2.3, is*

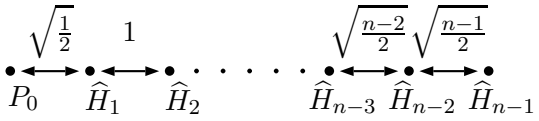


Figure 4.4: The visualization of the finite discrete normalized Hermite model $\mathbb{C}[x]/\widehat{H}_n(x)$ with basis $\widehat{H}_0(x), \widehat{H}_1(x), \ldots, \widehat{H}_{n-1}(x)$.

$$\begin{aligned} &\mathcal{A} = \{h = \textstyle\sum_{0\le\ell<n} h_\ell \widehat{H}_\ell(x)\}, \\ &\mathcal{M} = \{s = \textstyle\sum_{0\le k<n} s_k \widehat{H}_k(x)\}, \\ &\Phi:\ \mathbb{C}^n \to \mathcal{M},\ \boldsymbol{s} \mapsto \textstyle\sum_{0\le k<n} s_k \widehat{H}_k(x). \end{aligned} \tag{4.21}$$

The visualization of this model is shown in Fig. 4.4.

The associated Fourier transform for this model is specified by the polynomial transform

$$\mathcal{P}_{b,\alpha} = \left[\widehat{H}_\ell(\alpha_k)\right]_{0\le k,\ell n}, \tag{4.22}$$

where $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}$ *are the zeros of* $\widehat{H}_n(x)$. *For example, for* $n = 6$, *it has the form*

$$\mathcal{P}_{b,\alpha} = \begin{pmatrix} 1 & -3.3243 & 7.1069 & -10.9258 & 12.0053 & -8.0754 \\ 1 & -1.8892 & 1.8165 & -0.4388 & -1.1587 & 1.3714 \\ 1 & -0.6167 & -0.4382 & 0.6596 & 0.1761 & -0.6385 \\ 1 & 0.6167 & -0.4382 & -0.6596 & 0.1761 & 0.6385 \\ 1 & 1.8892 & 1.8165 & 0.4388 & -1.1587 & -1.3714 \\ 1 & 3.3243 & 7.1069 & 10.9258 & 12.0053 & 8.0754 \end{pmatrix}.$$

The orthogonalized polynomial transform then is

$$D^{-1/2}\mathcal{P}_{b,\alpha} = \begin{pmatrix} 0.0506 & -0.1681 & 0.3593 & -0.5523 & 0.6069 & -0.4082 \\ 0.2977 & -0.5624 & 0.5408 & -0.1306 & -0.3449 & 0.4082 \\ 0.6394 & -0.3943 & -0.2802 & 0.4217 & 0.1126 & -0.4082 \\ 0.6394 & 0.3943 & -0.2802 & -0.4217 & 0.1126 & 0.4082 \\ 0.2977 & 0.5624 & 0.5408 & 0.1306 & -0.3449 & -0.4082 \\ 0.0506 & 0.1681 & 0.3593 & 0.5523 & 0.6069 & 0.4082 \end{pmatrix}.$$

Chapter 5

Perfect-Reconstruction Filter Banks

In this chapter we introduce perfect-reconstruction filter banks for infinite discrete GNN signal models. We generalize the notions of low and high frequencies for GNN signals. Then we compare advantages and disadvantages of filter and signal expansions approaches to the construction of filter banks. Finally, we construct two classes of perfect-reconstruction filter banks for infinite discrete GNN signals.

5.1 Flatness

In Section 3.1, we introduced the concept of flatness for filters. Namely, a low-pass filter $h(x)$ is said to have degree of flatness M, if

$$\frac{d^m}{d\omega^m} H(\omega)\Big|_{\omega=\omega_H} = 0$$

for $0 \leq m < M$, where ω_H is the highest frequency. Respectively, a high-pass filter $h(x)$ is said to have degree of flatness M, if

$$\frac{d^m}{d\omega^m} H(\omega)\Big|_{\omega=\omega_L} = 0$$

for $0 \leq m < M$, where ω_L is the lowest frequency.

Naturally, in order to introduce the concept of flatness for filters in infinite discrete GNN models, we must first properly define the low and high frequencies for such models.

5.1.1 Low and High Frequencies

The concept of low and high frequencies, strictly speaking, introduces a partial (weak) order $\preceq$ on the spectrum W for infinite discrete GNN signal models. We choose the following physical interpretation.

Definition 5.1.1 *Let v be a vector or an infinite sequence. The* number of oscillations $N_T(v)$ *of v over interval $T \in \mathbb{N}_0$ is equal to the number of times the sequence starts decreasing or increasing, i.e.*

$$N_T(v) = \Big|\{k \mid 0 \le k < T, (v_{k-1} > v_k \wedge v_k < v_{k+1}) \vee (v_{k-1} < v_k \wedge v_k > v_{k+1})\}\Big|.$$

We define the order $\preceq$ on the spectrum W based on the number of oscillations of corresponding eigenfunctions $E_\omega(x)$.

Definition 5.1.2 *Frequencies $\omega_1, \omega_2 \in W$ are said to be in the order*

$$\omega_1 \preceq \omega_2,$$

if the corresponding eigenfunction $E_{\omega_1}(x)$ has fewer oscillations than $E_{\omega_2}(x)$ over the same interval T :

$$\omega_1 \preceq \omega_2 \quad \Leftrightarrow \quad N_T(E_{\omega_1}(x)) \le N_T(E_{\omega_2}(x)).$$

The interval T in Definition 5.1.2 can be chosen empirically. The frequency ω_L, such that $\omega_L \preceq \omega$ for any $\omega \in W$, is called the *lowest frequency*. Respectively, ω_H, such that $\omega \preceq \omega_H$ for any $\omega \in W$, is called the *highest frequency*.

Example 5.1.3 *Let us identify the lowest and highest frequencies for several GNN models.*

1) Consider the infinite discrete GNN model based on Chebyshev polynomials of the third type (2.16). *The frequency spectrum of this model is $W = [-1, 1]$. It follows from Chapter 4 that the eigenfunctions for this model have the form*

$$E_\omega(x) = \sum_{k \ge 0} V_k(\omega) V_k(x).$$

The corresponding coordinate form is

$$\boldsymbol{E}_\omega = \big(V_0(\omega), V_1(\omega), V_2(\omega), \dots\big).$$

As shown in Fig. 5.1(a), the eigenfunction

$$\boldsymbol{E}_1 = \big(1, 1, 1, \dots\big)$$

is a constant. The number of oscillations of $\boldsymbol{E}_\omega$ increases as ω becomes closer to -1. Hence, the lowest frequency for the infinite space model based on Chebyshev polynomials of the third kind is $\omega_L = 1$ and the highest frequency is $\omega_H = -1$.

2) *Next, consider the infinite discrete GNN model based on Laguerre polynomials. The frequency spectrum of this model is $W = [0, \infty)$. It follows from Chapter 4 that the eigenfunctions for this model have the coordinate form*

$$\boldsymbol{E}_\omega = \big(L_0(\omega), L_1(\omega), L_2(\omega), \dots\big).$$

As shown in Fig. 5.1(b), eigenfunction

$$\boldsymbol{E}_0 = \big(1, 1, 1, \dots\big)$$

is a constant. The number of oscillations of $\boldsymbol{E}_\omega$ increases as ω increases. Hence, the lowest frequency for the infinite space model based on Laguerre polynomials is $\omega_L = 0$ and the highest frequency is $\omega_H = \infty$.

3) *Finally, consider the infinite discrete GNN model based on normalized Hermite polynomials introduced in Example 4.2.3. The frequency spectrum of this model is $W = (-\infty, \infty)$. The eigenfunctions for this model have the coordinate form*

$$\boldsymbol{E}_\omega = \big(\widehat{H}_0(\omega), \widehat{H}_1(\omega), \widehat{H}_2(\omega), \dots\big).$$

As shown in Fig. 5.1(c), the number of oscillations of $\boldsymbol{E}_\omega$ decreases as ω increases. Hence, the lowest

frequency for the infinite space model based on normalized Hermite polynomials is $\omega_L = \infty$ *and the highest frequency is* $\omega_H = -\infty$.

Observe that in all examples above the order function $\preceq$ is monotonous with ω. Namely, if any frequencies $\omega_1, \omega_2, \omega_3 \in W$ satisfy

$$\omega_1 < \omega_2 < \omega_3,$$

then their frequency ordering is either

$$\omega_1 \preceq \omega_2 \preceq \omega_3$$

(for the second model), or

$$\omega_3 \preceq \omega_2 \preceq \omega_1$$

(for the first and third models).

5.1.2 Flatness of Filters and Signals

We have introduced the notion of frequency order for infinite discrete GNN models. Now we can extend the definition of flatness to these models.

Definition 5.1.4 *Consider an infinite discrete GNN signal model* (4.5)*, with filters of the form*

$$h(x) = \sum_{\ell=L_0}^{L_1-1} h_\ell P_\ell(x).$$

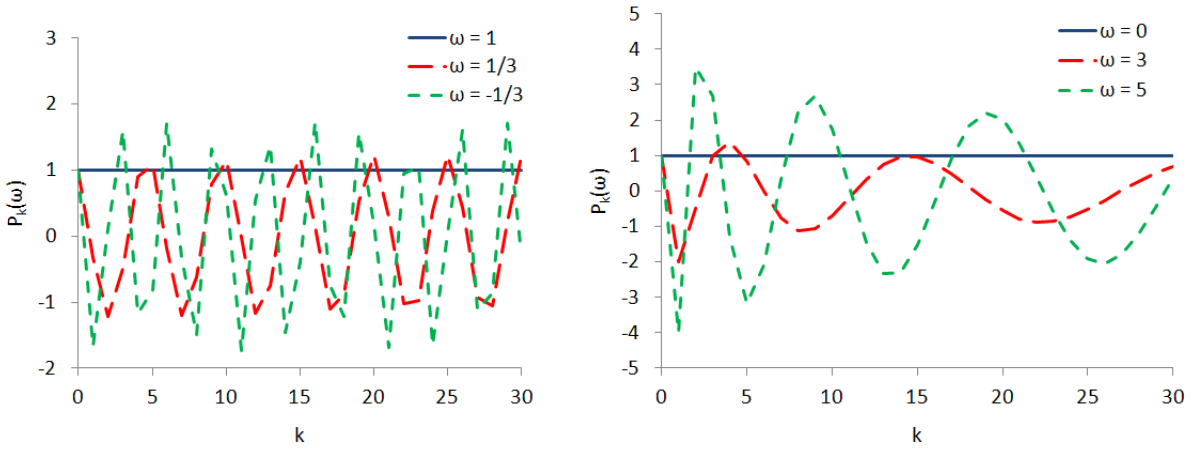
Filter $h(x)$ *is called a low-pass filter of degree* M *with* degree of flatness M*, if it satisfies*

$$\frac{d^m}{d\omega^m} H(\omega)\Big|_{\omega=\omega_H} = 0$$

for $0 \leq m < M$.

The definition of a high-pass filter with a specific degree of flatness is analogous.

Similarly to filters, we can talk about a degree of flatness for signals:



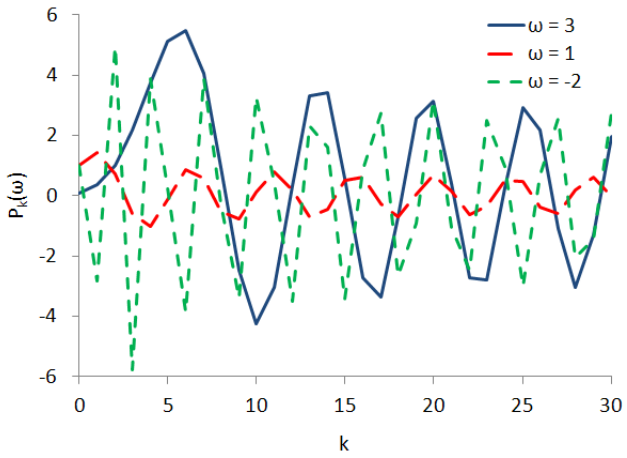


Figure 5.1: Eigenfunctions corresponding to different frequencies ω for infinite discrete GNN signal models based on (a) Chebyshev polynomials of the third kind ($P = V$); (b) Laguerre polynomials ($P = L$); and (c) normalized Hermite polynomials ($P = \tilde{H}$).

Definition 5.1.5 *Consider an infinite discrete GNN signal model* (4.5)*, with signals of the form*

$$s(x) = \sum_{k=K_0}^{K_1-1} s_k P_k(x).$$

Signal $s(x)$ is called a low-frequency signal with degree of flatness M*, if it satisfies*

$$\frac{d^m}{d\omega^m} S(\omega)\Big|_{\omega=\omega_H} = 0$$

for $0 \leq m < M$.

The definition of a high-frequency signal with a specific degree of flatness is analogous.

Observe that there can exist multiple (in fact, infinitely many) filters and signals with a given degree of flatness.

5.2 Filter Approach vs. Expansion Approach

In Section 3.1, we discussed two approaches to the design of perfect reconstruction filter banks. One approach was based on filtering followed downsampling. Another approach was based on expanding signals into bases (or frames) with desired characteristics.

The equivalence of these approaches is possible for two reasons. First, we assume a particular structure of the basis/frame elements. Second, the inner product and time convolution are interchangeable; one only needs to reverse the order of the coefficients [23, 46].

Unfortunately, this interchangeability between the convolution and the scalar product may not extend to other signal models. The convolution operation depends on the underlying signal model and is defined by the product $h(x)s(x)$ of a filter $h(x) \in \mathcal{A}$ and a signal $s(x) \in \mathcal{M}$ as follows. Consider an arbitrary infinite discrete signal model. Let the basis of $\mathcal{M}$ be $\big(\dots, P_0(x), P_1(x), \dots\big)$; and the basis of $\mathcal{A}$ be

$\big(\dots, R_0(x), R_1(x), \dots\big)$. Then the k-th element of the convolution $\mathbf{h} * \mathbf{s}$ is determined from the product

$$\begin{aligned} h(x)s(x) &= \sum_{\ell} h_\ell R_\ell(x) \sum_{k} s_k P_k(x) \\ &= \sum_{k} (h * s)_k P_k(x). \end{aligned}$$

On the other hand, the definition of a scalar product remains the same. Hence, we have to choose a preferred approach to the design of filter banks for other signal models, or somehow combine the two approaches together.

In this section, we investigate the construction of Haar-like filter banks for infinite discrete space models (2.16). We construct the filter banks using both approaches. We then analyze their advantages and disadvantages, and identify the most suitable approach for the construction of filter banks for infinite discrete GNN models.

5.2.1 Signal Expansion Approach

In [95], we use the signal expansion approach to construct Haar-like filter banks for infinite discrete space signals. Similarly to the standard Haar filter banks for time signals, our goal is to split a signal into a "coarse" and a "detailed" components by averaging the signal coefficients and computing the remaining details.

Since we have $M = N = 2$ channels, we slightly change the notation from Section 3.1. Instead of denoting bases with $\big(\varphi_k^{(0)}\big)$ and $\big(\varphi_k^{(1)}\big)$, we use $\varphi = \big(\varphi_k\big)$ and $\psi = \big(\psi_k\big)$, respectively. The support length of φ_k and ψ_k is $L = 1 \cdot N = 2$.

Consider the infinite discrete space model (2.16) based on Chebyshev polynomials of the third kind. We seek to construct low- and a high-frequency bases $\varphi = \big(\varphi_k\big)_{k\geq 0}$ and $\psi = \big(\psi_k\big)_{k\geq 0}$ that have only two non-zero coefficients in the $2k$-th and $(2k+1)$-th positions:

$$\begin{aligned} \varphi_k &= \Big(\dots, 0, a_k, b_k, 0, \dots\Big), \\ \psi_k &= \Big(\dots, 0, c_k, d_k, 0, \dots\Big). \end{aligned}$$

We write them as

$$\begin{aligned}\varphi_k(x) &= a_k V_{2k}(x) + b_k V_{2k+1}(x)\\ \psi_k(x) &= c_k V_{2k}(x) + d_k V_{2k+1}(x).\end{aligned}$$

To ensure the frequency conditions, we require that the spectra of basis functions φ_k and ψ_k disappear at the highest ($\omega_H = -1$) and lowest ($\omega_L = 1$) frequencies, respectively:

$$\begin{cases} a_k V_{2k}(-1) + b_k V_{2k+1}(-1) = 0,\\ c_k V_{2k}(1) + d_k V_{2k+1}(1) = 0.\end{cases} \tag{5.1}$$

Using the properties of Chebyshev polynomials, we solve (5.1) to obtain $(4k+3)b_k = (4k+1)a_k$ and $d_k = -c_k$. Hence,

$$\varphi_k(x) = a_k V_{2k}(x) + \frac{4k+1}{4k+3} a_k V_{2k+1}, \tag{5.2}$$

$$\psi_k(x) = c_k V_{2k}(x) - c_k V_{2k+1}(x). \tag{5.3}$$

Observe that the bases φ and ψ span independent subspaces of $\mathcal{M}$, i.e. $\langle\varphi\rangle \cap \langle\psi\rangle = \{0\}$. Moreover, the original basis $\big(V_k(x)\big)_{k\le 0}$ can be expressed in terms of φ and ψ: assuming $a_k = 1$ and $c_k = 1$ for all k,

$$\begin{aligned}V_{2k}(x) &= \frac{4k+3}{8k+4}\varphi_k(x) + \frac{4k+1}{8k+4}\psi_k(x),\\ V_{2k+1}(x) &= \frac{4k+3}{8k+4}\varphi_k(x) - \frac{4k+3}{8k+4}\psi_k(x).\end{aligned}$$

Hence, $\varphi \cup \psi$ is a basis for the signal space $\mathcal{M}$, and we obtain a critically-sampled perfect reconstruction filter bank.

To compute the projections of the signal s onto φ and ψ, we construct dual bases $\tilde{\varphi}$ and $\tilde{\psi}$ that satisfy

$$\begin{cases} \langle\varphi_k, \tilde{\varphi}_m\rangle = \langle\psi_k, \tilde{\psi}_m\rangle = \delta_{k-m},\\ \langle\varphi_k, \tilde{\psi}_m\rangle = \langle\tilde{\varphi}_k, \psi_m\rangle = 0.\end{cases} \tag{5.4}$$

Then we compute scalar products $\langle\tilde{\varphi}_k, s\rangle$ and $\langle\tilde{\psi}_k, s\rangle$ to find the projection coefficients. From (5.2)-(5.4) we derive the dual bases

$$\begin{aligned}\tilde{\varphi}_k(x) &= \frac{4k+3}{(8k+4)a_k}V_{2k}(x) + \frac{4k+3}{(8k+4)a_k}V_{2k+1}(x),\\ \tilde{\psi}_k(x) &= \frac{4k+1}{(8k+4)c_k}V_{2k}(x) - \frac{4k+3}{(8k+4)c_k}V_{2k+1}(x).\end{aligned}$$

Assuming for simplicity that $a_k = c_k = 1/\sqrt{2}$, we obtain the analysis and synthesis matrices

$$\tilde{\Phi} = \begin{pmatrix} \tilde{\Phi}_0 & & & \\ & \tilde{\Phi}_1 & & \\ & & \tilde{\Phi}_2 & \\ & & & \ddots \end{pmatrix}, \qquad \Phi = \begin{pmatrix} \Phi_0 & & & \\ & \Phi_1 & & \\ & & \Phi_2 & \\ & & & \ddots \end{pmatrix},$$

where

$$\tilde{\Phi}_k = \sqrt{2}\begin{pmatrix} \frac{4k+3}{8k+4} & \frac{4k+1}{8k+4} \\ \frac{4k+3}{8k+4} & -\frac{4k+3}{8k+4} \end{pmatrix}, \qquad \Phi_k = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ \frac{4k+1}{4k+3} & -1 \end{pmatrix}.$$

Observe that, as $k \to \infty$,

$$\tilde{\Phi}_k \to \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad \Phi_k \to \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Hence, we can approximate this filter bank with the standard Haar filter bank for the time signal model constructed in Example 3.1.2.

Similarly, we have derived Haar filter banks for other space signal models:

$C = T$:

$$\begin{aligned}
\varphi_k(x) &= a_k T_{2k}(x) + a_k T_{2k+1}(x) \\
\psi_k(x) &= c_k T_{2k}(x) - c_k T_{2k+1}(x) \\
\tilde{\varphi}_k(x) &= \frac{1}{2a_k} T_{2k}(x) + \frac{1}{2a_k} T_{2k+1}(x) \\
\tilde{\psi}_k(x) &= \frac{1}{2c_k} T_{2k}(x) - \frac{1}{2c_k} T_{2k+1}(x)
\end{aligned}$$

$C = U$:

$$\begin{aligned}
\varphi_k(x) &= a_k U_{2k}(x) + \frac{2k+1}{2k+2} a_k U_{2k+1}(x) \\
\psi_k(x) &= c_k U_{2k}(x) - \frac{2k+1}{2k+2} c_k U_{2k+1}(x) \\
\tilde{\varphi}_k(x) &= \frac{1}{2a_k} U_{2k}(x) + \frac{k+1}{(2k+1)a_k} U_{2k+1}(x) \\
\tilde{\psi}_k(x) &= \frac{1}{2c_k} U_{2k}(x) - \frac{k+1}{(2k+1)c_k} U_{2k+1}(x)
\end{aligned}$$

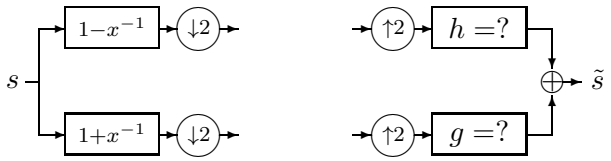
$C = W$:

$$\begin{aligned}
\varphi_k(x) &= a_k W_{2k}(x) + a_k W_{2k+1}(x) \\
\psi_k(x) &= c_k W_{2k}(x) - \frac{4k+1}{4k+3} c_k W_{2k+1}(x) \\
\tilde{\varphi}_k(x) &= \frac{4k+1}{(8k+4)a_k} W_{2k}(x) + \frac{4k+3}{(8k+4)a_k} W_{2k+1}(x) \\
\tilde{\psi}_k(x) &= \frac{4k+3}{(8k+4)c_k} W_{2k}(x) - \frac{4k+3}{(8k+4)c_k} W_{2k+1}(x)
\end{aligned}$$

5.2.2 Filter Approach

In [22], sampling theorems have been formulated for the four infinite discrete space models. As follows from them, the proper downsampling for space signals after filtering with a low-pass half-band filter is the same as the downsampling for time signals: omitting every other coefficient. Hence, we can attempt to construct a Haar-like filter bank for space signals using the filter approach.

The following example illustrates that the convolution and scalar product are not interchangeable for



(a) Analysis part (b) Synthesis part

Figure 5.2: Haar filter bank for the infinite discrete space model (2.16) constructed using the filter approach.

signal models other than the standard time model. Again, consider the infinite discrete space model (2.16) based on Chebyshev polynomials of the third kind. Let $\tilde{h}(x)$ be a low-pass maxflat filter of degree 1. Its frequency response $\tilde{H}(\omega)$ must vanish at the highest frequency $\omega = -1$, such that

$$\tilde{H}(-1) = 0.$$

We choose the low-pass filter with the shortest support

$$\tilde{h}(x) = 1 + x = 1 + T_1(x).$$

Similarly, the maxflat high-pass filter $\tilde{g}(x)$ of degree 1 must satisfy

$$\tilde{G}(1) = 0.$$

We choose

$$\tilde{g}(x) = 1 - x = 1 - T_1(x).$$

Hence, the analysis part of the Haar filter bank has the structure as shown in Fig. 5.2(a).

The constructed analysis part *looks* just like the analysis part of the traditional Haar filter bank for the

infinite time signals. However, notice that the filtered signals are

$$\begin{aligned}(1+x)\cdot\sum_{k\geq 0}s_kV_k(x) &= \frac{3s_0+s_1}{2}V_0(x)+\sum_{k\geq 1}\frac{s_{k-1}+2s_k+s_{k+1}}{2}V_k(x),\\ (1-x)\cdot\sum_{k\geq 0}s_kV_k(x) &= \frac{s_0-s_1}{2}V_0(x)-\sum_{k\geq 1}\frac{s_{k-1}-2s_k+s_{k+1}}{2}V_k(x).\end{aligned}$$

After downsampling and upsampling by 2, we obtain the signals

$$\begin{aligned}s'(x) &= \frac{3s_0+s_1}{2}V_0(x)+\sum_{k\geq 1}\frac{s_{2k-1}+2s_{2k}+s_{2k+1}}{2}V_{2k}(x),\\ s''(x) &= \frac{s_0-s_1}{2}V_0(x)-\sum_{k\geq 1}\frac{s_{2k-1}-2s_{2k}+s_{2k+1}}{2}V_{2k}(x).\end{aligned}$$

The original signal $s(x)$ can be reconstructed from $s'(x)$ and $s''(x)$ as follows:

$$\begin{aligned}s_{2k} &= \frac{s'_k+s''_k}{2},\\ s_1 &= \frac{s'_0-3s''_0}{2},\\ s_{2k+1} &= s'_k-s''_k-s_{2k-1}=\sum_{i=0}^{k}(-1)^{k-i}(s'_i-s''_i)-2(-1)^k s''_0.\end{aligned} \tag{5.5}$$

This reconstruction can be implemented recursively using basic building blocks for filter banks, such as multipliers, adders, and delays. However, there are no filters $h(x), g(x)\in\mathcal{A}$ such that

$$h(x)s'(x)+g(x)s''(x)=s(x).$$

Hence, we cannot design a synthesis part of the filter bank that has the structure similar to the standard filter bank, as shown in Fig. 5.2(b).

We can, however, interpret the reconstruction equations (5.5) through the basis point of view. Consider

the bases $\big(\varphi_k(x)\big)_{k\geq 0}$ and $\big(\psi_k(x)\big)_{k\geq 0}$, where

$$\begin{aligned}\varphi_0(x) &= \frac{1}{2}V_0(x) + \frac{1}{2}V_1(x) + \sum_{m\geq 1}(-1)^m V_{2m+1}(x),\\ \varphi_k(x) &= \frac{1}{2}V_{2k}(x) + \sum_{m\geq 1}(-1)^{m-1} V_{2m+1}(x),\ \ k\geq 1,\end{aligned}$$

and

$$\begin{aligned}\psi_0(x) &= \frac{1}{2}V_0(x) - \frac{3}{2}V_1(x) + 3\cdot\sum_{m\geq 1}(-1)^{m-1} V_{2m+1}(x),\\ \psi_k(x) &= \frac{1}{2}V_{2k}(x) + \sum_{m\geq 1}(-1)^{m} V_{2m+1}(x),\ \ k\geq 1.\end{aligned}$$

Then, the original signal $s(x)$ can be reconstructed as

$$s(x) = \sum_{k\geq 0} s'_k\varphi_k(x) + \sum_{k\geq 0} s''_k\psi_k(x).$$

5.2.3 Combined Approach

As we mentioned, we cannot construct synthesis parts of filter banks for infinite discrete GNN models that consist of upsamplers followed by filters. This is the main disadvantage of the filter approach to filter bank construction for infinite discrete GNN models.

The signal expansion approach to filter bank construction, on the other hand, suffers from an opposite disadvantage—the construction of signal bases with a desired degree of flatness that define the analysis part of a filter bank. Consider, for example, a filter bank for an infinite discrete GNN model with the analysis part corresponding to the bases with flatness degree 3. The search for bases of low- and high-frequency signals with flatness degree 3, whose union is a basis for the entire signal space $\mathcal{M}$, is highly complicated. In particular, there is no general closed form expression for derivatives of orthogonal polynomials evaluated at the lowest and highest frequencies ω_L and ω_H, respectively.

As a result, we propose to use a combined approach for the construction of two-channel filter banks for infinite discrete GNN models that have a desired degree of flatness.

a) The analysis part of each channel in an M-channel filter bank is constructed as a low-pass or high-pass filter $\tilde{h}_m$ with a desired flatness degree, followed by a downsampler.

b) At this point we do not have a general theory of continuous infinite GNN models. As a result, we cannot formulate an analog of Nyquist sampling theorem for infinite discrete GNN models that would establish how a low-pass or a high-pass filtered signal must be downsampled[1] . Thus, we choose to downsample the same way it is done in case of infinite time signals: downsampling at the rate $N \leq M$ means keeping every N-th signal coefficient and dropping other ones.

c) The synthesis part of each channel is constructed as a linear combination of basis signals that span a subspace $\mathcal{M}_m$ of the entire signal space $\mathcal{M}$. The example of Haar filter bank for infinite space model constructed in Section 5.2.2 is an illustration of this combined approach.

Depending on the purpose of a filter bank, the proposed combined approach may or may not be the most suitable one. It is centered around constructing analysis filters first, and is appropriate if the properties of the analysis filters are the main characteristics of a filter bank. However, if the synthesis part of a desired filter bank is the main design criteria, our approach may not be optimal, since it requires the construction of signal bases with infinitely many vectors that must satisfy given criteria.

5.3 Two-Channel Filter Banks

In this section, we study the construction of two-channel filter banks for infinite discrete GNN models based on different families of orthogonal polynomials. We select representatives from different classes depending on the spectrum (see Appendix A for more details), and construct example filter banks of order 1 (Haar-like) or 2. We also identify certain limitations in the filter bank construction that arise for infinite discrete GNN models based on Laguerre-like and Hermite-like polynomials.

[1] In fact, previous research (see, for example, the discussion in [93]) suggests that there may be *no* continuous GNN signal models. However, we may try to determine the proper sampling technique without direct construction of continuous GNN models. For example, we can utilize such mathematical techniques as interpolation, similarly to the way Lagrange interpolation was used to re-derive the Nyquist sampling theorem in [96] .

5.3.1 GNN Models Based on Jacobi-Like Polynomials

Infinite discrete GNN models based on Jacobi-like polynomials have the finite spectrum $W = [-1, 1]$. As a representative of this class of orthogonal polynomials, we continue to use Chebyshev polynomials of the third kind.

We have constructed the Haar-like filter bank, which is a filter bank with flatness degree 1, in Section 5.2.2. Here, we proceed to construct a two-channel filter bank with flatness degree 2.

In this case, low-pass filter $\tilde{h}_0(x)$ and high-pass filter $\tilde{h}_1(x)$ must satisfy

$$\tilde{H}_0(-1) = \frac{d}{d\omega}\tilde{H}_0(\omega)\Big|_{\omega=-1} = 0,$$
$$\tilde{H}_1(1) = \frac{d}{d\omega}\tilde{H}_1(\omega)\Big|_{\omega=1} = 0,$$

As the shortest filters with these properties, we choose

$$\begin{aligned}\tilde{h}_0(x) &= 6T_0(x) + 8T_1(x) + 2T_2(x),\\ \tilde{h}_1(x) &= 6T_0(x) - 8T_1(x) + 2T_2(x).\end{aligned}$$

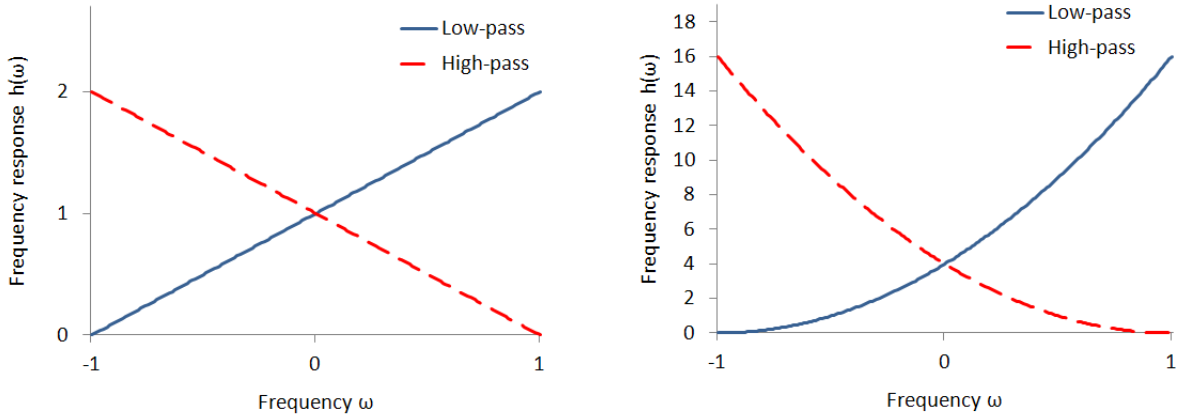
Their frequency responses are shown in Fig. 5.3(b). If we compare them with the frequency responses of the Haar-like filter bank that are shown in Fig. 5.3(a), we observe that the low-pass and high-pass filters with flatness degree 2 attenuate, respectively, high and low frequencies more than the corresponding filters in the Haar-like filter bank.

After filtering input signal $s(x) = \sum_{k\geq 0} s_k V_k(x)$ with filter $\tilde{h}_0(x)$ and downsampling by 2, we obtain new signal

$$s'(x) = (10s_0 + 5s_1 + s_2)V_0(x) + \sum_{k\geq 1}(s_{2k-2} + 4s_{2k-1} + 6s_{2k} + 4s_{2k+1} + s_{2k+2})V_k(x)$$

as the output of the low-pass channel. Similarly, we get

$$s''(x) = (2s_0 - 3s_1 + s_2)V_0(x) + \sum_{k\geq 1}(s_{2k-2} - 4s_{2k-1} + 6s_{2k} - 4s_{2k+1} + s_{2k+2})V_k(x)$$



(a) Filters with flatness degree 1 (Haar-like).

(b) Filters with flatness degree 2.

Figure 5.3: Frequency responses of low-pass and high-pass filters for the infinite discrete GNN model based on Chebyshev polynomials of the third kind.

as the output of the high-pass channel.

To construct the synthesis part of this filter bank, we must find linearly independent sets $\big(\varphi_k\big)_{k\geq 0}$ and $\big(\psi_k\big)_{k\geq 0}$, such that their union is a basis of the entire signal space $\mathcal{M}$, and

$$s(x) = \sum_{k\geq 0} s'_k \varphi_k + \sum_{k\geq 0} s''_k \psi_k.$$

To simplify the construction of the basis, we can assume that the first four signal coefficients are zero: $s_0 = s_1 = s_2 = s_3 = 0$. Alternatively, we can attach four zeros to input signal $s(x)$. Then, the required basis functions are

$$\begin{aligned}
\varphi_k &= \frac{1}{8\sqrt{2}} \sum_{m\geq 0} \lambda_m V_{2k+2m+1}(x) + \frac{1}{8} \sum_{m\geq 0} V_{2k+2m+1}(x), \\
\psi_k &= \frac{1}{8\sqrt{2}} \sum_{m\geq 0} \lambda_m V_{2k+2m+1}(x) - \frac{1}{8} \sum_{m\geq 0} V_{2k+2m+1}(x),
\end{aligned}$$

where

$$\lambda_m = (2\sqrt{2}-3)^{m+1} + (-1)^m (2\sqrt{2}+3)^{m+1}.$$

For practical purposes, this reconstruction can be implemented recursively.

Using this approach we can construct filter banks with arbitrary flatness degrees.

5.3.2 GNN Models Based on Laguerre-Like and Hermite-Like Polynomials

Infinite discrete GNN models based on Laguerre-like polynomials have the semifinite spectrum $W = [0, \infty)$. Infinite discrete GNN models based on Hermite-like polynomials have the infinite spectrum $W = (-\infty, \infty)$. In both cases we run into the following problem. Assume

$$h(x) = \sum_{\ell=L_0}^{L_1-1} h_\ell P_\ell(x)$$

is an FIR filter. Since its frequency response $H(\omega) = h(\omega)$ is a polynomial in variable ω, it is unbounded:

$$\lim_{\omega\to\infty} |H(\omega)| = \infty.$$

Thus, all FIR filters for infinite discrete GNN models based on Laguerre-like or Hermite-like polynomials have unbounded frequency responses. For this reason, FIR filters are impractical to use in two-channel filter banks with a low-pass and a high-pass channels—the attenuation of large frequencies is unbounded.

Instead, we can use IIR filters to construct filter banks. As an example, we construct a filter bank with flatness degree 1 for the infinite discrete GNN model based on Laguerre polynomials.

As we discussed in Section 5.1, in this case the spectrum is $W = [0, \infty)$, and the lowest and highest frequencies are $\omega_L = 0$ and $\omega_H = \infty$, respectively. As the low-pass and high-pass analysis filters $\tilde{h}_0(x)$ and $\tilde{h}_1(x)$ that satisfy

$$\begin{aligned} \tilde{H}_0(\infty) &= 0, \\ \tilde{H}_1(0) &= 0, \end{aligned}$$

we choose

$$\begin{aligned} \tilde{h}_0(x) &= \sum_{\ell\geq 0} (-1)^\ell x^\ell = \frac{1}{1+x}, \\ \tilde{h}_1(x) &= \sum_{\ell\geq 0} (-1)^\ell x^{\ell+1} = \frac{x}{1+x}. \end{aligned}$$

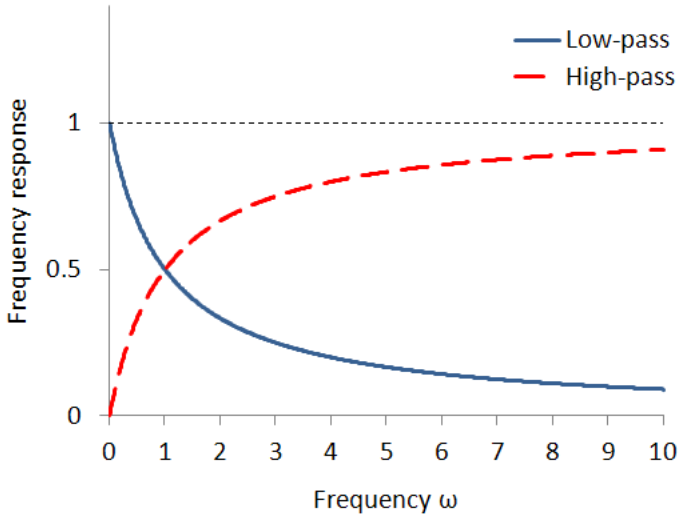


Figure 5.4: Frequency responses of the low-pass and high-pass filters for the infinite discrete GNN model based on Laguerre polynomials.

Their frequency responses are shown in Fig. 5.4.

First, we must determine the result $s'(x) = \sum_{k\geq 0} s'_k L_k(x)$ of filtering input signal $s(x) = \sum_{k\geq 0} s_k L_k(x)$ with filter $\tilde{h}_0(x)$. As we discussed in Section 3.1.3, we cannot compute the convolution directly, since the number of taps in the filter is infinite. Instead, we use the differential equations.

Since

$$s'(x) = \tilde{h}_0(x) \cdot s(x) = \frac{1}{1+x} s(x),$$

we obtain

$$\begin{aligned}
& (1+x)s'(x) = s(x) \\
\Rightarrow \quad & \sum_{k\geq 0} s'_k L_k(x) + x \cdot \sum_{k\geq 0} s'_k L_k(x) = \sum_{k\geq 0} s_k L_k(x) \\
\Rightarrow \quad & \sum_{k\geq 0} s'_k L_k(x) + \sum_{k\geq 0} s'_k(-kL_{k-1}(x) + (2k+1)L_k(x) - (k+1)L_{k+1}(x)) = \sum_{k\geq 0} s_k L_k(x).
\end{aligned}$$

After comparing coefficients at each polynomial $L_k(x)$, we obtain the relation

$$-ks'_{k-1} + (2k+2)s'_k - (k+1)s'_{k+1} = s_k$$

for $k \geq 0$. Assuming, for example, $s'_1 = 0$, we derive the recurrence

$$\begin{aligned} s'_0 &= \frac{1}{2}s_0, \\ s'_1 &= 0, \\ s'_k &= 2s'_{k-1} - \frac{k-1}{k}s'_{k-2} - \frac{1}{k}s_{k-1}, \quad k \geq 2. \end{aligned}$$

Similarly, the result $s''(x) = \sum_{k\geq 0} s'_k L_k(x)$ of filtering input signal $s(x) = \sum_{k\geq 0} s_k L_k(x)$ with filter $\tilde{h}_1(x)$ can be calculated recursively as

$$\begin{aligned} s''_0 &= \frac{s_0 - s_1}{2}, \\ s''_1 &= 0, \\ s''_k &= 2s''_{k-1} - \frac{k-1}{k}s''_{k-2} + \frac{k-1}{k}s_{k-2} - \frac{2k-1}{k}s_{k-1} + s_k, \quad k \geq 2. \end{aligned}$$

After that, we can downsample $s'(x)$ and $s''(x)$ at rate 2, thus keeping only coefficients s'_{2k} and s''_{2k}. These are the outputs of the analysis part of the filter bank.

Similarly to the derivation in Section 5.3.1, we construct the synthesis part of this filter bank by finding linearly independent sets (φ_k) and (ψ_k), such that their union is a basis of the entire signal space $\mathcal{M}$, and

$$s(x) = \sum_{k\geq 0} s'_{2k}\varphi_k + \sum_{k\geq 0} s''_{2k}\psi_k.$$

Due to the complexity of this construction, we omit it here.

5.4 Filter Banks for Robust Transmission

In this section, we study the construction of filter banks for robust signal transmission. For this construction, we employ the signal expansion approach. We construct maximally robust frames that allow for a redundant signal representation, as discussed in Section 3.1.5.

5.4.1 Maximally Robust Frames from DFT

The construction of maximally robust frames from polynomial transforms was studied in details in [48]. Building on this research, we have designed large families of critically-sampled and oversampled perfect reconstruction filter banks using the signal expansion approach [44, 45]. Such filter banks implement signal expansions that are robust to coefficient loss, and, as an additional benefit, also eliminate blocking effect. In addition, they are computationally efficient and allow for straightforward signal reconstruction.

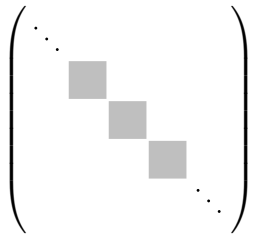
To simplify reconstruction, we consider orthonormal bases and tight frames to obtain critically-sampled and oversampled filter banks, respectively. Recall from Section 3.1 that both these cases imply self-duality $\Phi = \tilde{\Phi}$, and the perfect reconstruction condition (3.8) becomes

$$\Phi\Phi^* = I. \tag{5.6}$$

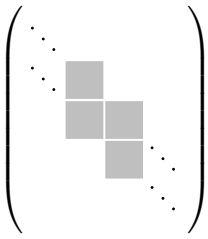
To eliminate the blocking effect, we require that the basis and frame functions have overlapping support. As we indicate in Section 3.1, each basis/frame function has support of length $L = qN$ for some integer $q \geq 1$. Then, depending on the value of q, Φ processes the signal $\mathbf{s}$ either in nonoverlapping ($q = 1$) or overlapping ($q \geq 2$) blocks, thus leading to either *blocked* or *lapped* transforms Φ^*. These cases are illustrated in Fig. 5.5. In this work, we assume $q = 2$.

Critically-sampled, perfect reconstruction filter banks with bases of overlapping support are known as *lapped orthogonal transforms (LOTs)* [42]. In [43], the frame counterparts of LOTs, called the *lapped tight frame transforms (LTFTs)*, have been constructed from LOTs using a special form of submatrix extraction. In this work, we systematically construct a large class of real LOTs from specific submatrices of DFT matrices. We then construct real LTFTs as properly selected submatrices LOTs. We prove that the corresponding frames are equal-norm, tight, and that many of them are maximally robust to erasures.

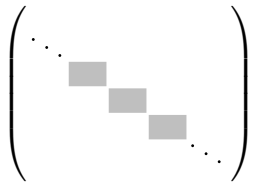
We defer the derivations and proofs to Appendix B, and only state here the main results.



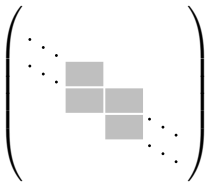
(a) Φ for basis expansion with blocked transform



(b) Φ for basis expansion with lapped transform ($q = 2$)



(c) Φ for frame expansion with blocked transform



(d) Φ for frame expansion with lapped transform ($q = 2$)

Figure 5.5: The infinite basis/frame matrix Φ in four different scenarios. The columns of Φ are the basis/frame vectors.

Lapped orthogonal transforms

Consider the $K \times K$ matrix

$$\mathrm{DFT}_{p,K}(z) = \frac{1}{\sqrt{K}} \left[\cos \frac{2k\ell\pi}{K} + z^{-1} \sin \frac{2k\ell\pi}{K} \right]_{0 \le k,\ell \le K-1}. \tag{5.7}$$

Let

$$\Psi_p(z) = \Psi_0 + z^{-1}\Psi_1$$

be an $M \times M$ submatrix of

$$\sqrt{K/M} \cdot \mathrm{DFT}_{p,K}(z),$$

where $K \ge M \ge 2$, constructed by selecting the following row and column sets:

$$\begin{aligned} \text{rows:} \quad & \{r + kR \bmod K \mid 0 \le k \le M-1\} \\ \text{columns:} \quad & \{c + \ell C \bmod K \mid 0 \le \ell \le M-1\} \end{aligned}$$

for some constants $0 \leq r, c, R, C < K$.

Assume that $K = M \gcd(K, RC)$. Also, assume that one of the following conditions is satisfied:

(i) K divides $2rC$, $4rc$, and $2MRc$;

(ii) K does not divide $2rC$, and K divides both $2r(2c + CM - C)$ and $R(2c + CM - C)$.

Then the matrix

$$\Phi = \begin{pmatrix} \ddots & & & \\ \ddots & \Psi_0 & & \\ & \Psi_1 & \Psi_0 & \\ & & \Psi_1 & \ddots \\ & & & \ddots \end{pmatrix} \tag{5.8}$$

corresponds to an LOT. Its columns form an orthonormal basis in $\ell^2(\mathbb{Z})$. The support length of $\varphi_k^{(m)}$ is $L = 2N$ and hence overlaps by N with the supports of $\varphi_{k-1}^{(m)}$ and $\varphi_{k+1}^{(m)}$. This LOT can be implemented with a critically-sampled M-channel ($M = N$) perfect-reconstruction filter bank.

Class of LOTs. Next we give one example of how to construct an entire class of LOTs for any size M. In Theorem B.3.2, we choose $K = aM$ with any $a \in \mathbb{N}_0$, $r = c = 0$, $R = 1$ and $C = a$ to satisfy condition (i). Then, for any $a \in \mathbb{N}_0$,

$$\Psi_{p,a}(z) = \frac{1}{\sqrt{M}} \left[\cos \frac{2k\ell a\pi}{K} + \sin \frac{2k\ell a\pi}{K} z^{-1} \right]_{0 \leq k,\ell < M} \tag{5.9}$$

is paraunitary, that is, Ψ^* is an LOT.

Number of new LOTs. Let us investigate how many $M \times M$ $\Psi_p(z)$ can be derived from $\mathrm{DFT}_{p,K}(z)$. Necessarily, $M \mid K$, which implies that K is not prime. This in mind, Table 5.1 shows the number of new LOTs generated using our method. For example, there are 28 3×3 paraunitary submatrices of $\mathrm{DFT}_{p,6}(z)$ and all are found with the theorem. Note that every submatrix is specified by a row subset and column subset of $\mathrm{DFT}_{p,6}(z)$; the ordering does not matter.

Further, there are 40 5×5 paraunitary submatrices of $\mathrm{DFT}_{p,10}$ that do not arise from Theorem B.3.2. One such example is the row set $\{0, 1, 3, 7, 9\}$, and the column set $\{0, 2, 4, 6, 8\}$. However, we speculate

	M						
K	2	3	4	5	6	7	8
4	16						
6	17	28					
8	128	—	64				
9	—	66					
10	49	—	—	84 *40*			
12	304	384 *128*	53 *116*	—	96 *176*		
14	97	—	—	—	—	172 *336*	
15	—	161 *120*	—	141 *376*			
16	896	—	1216 *1088*	—	—	—	256 *768*

Table 5.1: Number of paraunitary $M \times M$ $\Psi_p(z)$ generated from $\mathrm{DFT}_{p,K}$ using Theorem B.3.2. The numbers of paraunitary submatrices $\Psi_p(z)$ that do not satisfy Theorem B.3.2 are shown in italic.

that these matrices are up to permutations the same as other submatrices that are derived from the theorem. In fact, Theorem B.3.2 could be extended based on the permutation symmetries of the DFT [2, 97] and may then cover all paraunitary submatrices.

Finally, we must note that empirical tests show that there are no $M \times M$ paraunitary submatrices of $\mathrm{DFT}_{p,K}(z)$ for M not dividing K, for $K \leq 16$.

Lapped tight frame transforms

Let $\Psi_p(z)$ be constructed as above. Assume that M and MRC/K are co-prime. Further, let

$$\Phi_p(z) = \Phi_0 + z^{-1}\Phi_1$$

be constructed from $\Psi_p(z)$ by retaining $N < M$ rows with indices in set $\mathcal{I} \subseteq \{0, \ldots, M-1\}$, such that $|\mathcal{I}| = N$. If (as sets)

$$\mathcal{I} = \{d + Dk \bmod M \mid 0 \leq k < N\}$$

for some $0 \leq d < M$ and $D \equiv (MRC/K)^{-1} \bmod M$, then

$$\Phi = \begin{pmatrix} \ddots & & & \\ \ddots & \Phi_0 & & \\ & \Phi_1 & \Phi_0 & \\ & & \Phi_1 & \ddots \\ & & & \ddots \end{pmatrix} \tag{5.10}$$

corresponds to a lapped tight frame transform. Its columns form a lapped, tight, equal-norm, maximally robust to erasures frame in $\ell^2(\mathbb{Z})$ in $\ell^2(\mathbb{Z})$. The support length of $\varphi_k^{(m)}$ is $L = 2N$ and hence overlaps by N with the supports of $\varphi_{k-1}^{(m)}$ and $\varphi_{k+1}^{(m)}$. This LTFT can be implemented with a oversampled M-channel perfect reconstruction filter bank with sampling rate N.

Class of LTFTs. Given any $N < M$, an $N \times M$ $\Phi_p(z)$ can be constructed by seeding the $M \times M$ $\Psi_p(z)$ in (5.9), retaining N rows. Any such frame will be tight and equal norm. Since the construction parameters satisfy Theorem B.4.2, and $(MRC/K)^{-1} \equiv 1 \bmod M$, $\Phi_p(z)$ also is maximally robust if it results from consecutive seeding.

5.4.2 Maximally Robust Frames from GNN Transforms

As discussed in Section 5.4.1 and Appendix B, as well as in [44,45,48], tight frames can be constructed from orthonormal bases by the process called seeding. Hence, from each orthogonalized discrete GNN Fourier transform we can seed a tight frame. Moreover, a clever seeding procedure may ensure additional desirable properties of these frames, such as maximal robustness to erasures and equal norm.

Seeding by rows

In the rest of this section, we use a seeding procedure that is different from the one used in previous sections. Namely, instead of removing columns from a matrix, we remove *rows*. To distinguish this seeding, we write it as follows:

Definition 5.4.1 *Let A be an $n \times n$ matrix. Let $0 \leq i_0, i_1, \ldots, i_{m-1} < n$ be m distinct integers. Then matrix*

is said to be seeded by rows *from matrix A, if it is an $m \times n$ matrix constructed from rows $i_0, i_1, \ldots, i_{m-1}$ of matrix A. We write it as*

$$\Phi = A(i_0, i_1, \ldots, i_{m-1}).$$

Tight frames

In the following theorem we establish a seeding of orthogonalized discrete GNN transforms that yields tight frames in $\mathbb{R}^m$.

Theorem 5.4.2 *Consider discrete GNN transform $\mathcal{P}_{b,\alpha}$ that corresponds to the finite discrete GNN model* (4.16). *Let*

$$\mathcal{P}' = D^{-1/2} \cdot \mathcal{P}_{b,\alpha}$$

denote the orthogonalized polynomial transform, where D is defined in (4.18).

Let matrix Φ be obtained by seeding $\mathcal{P}'_{b,\alpha}$ by rows:

$$\Phi = \mathcal{P}'(i_0, i_1, \ldots, i_{m-1}).$$

Then the columns of Φ correspond to a real, tight frame of size in $\mathbb{R}^m$.

Proof: Without loss of generality, assume that $i_k = k$ for each $0 \leq k < m$. Then we can write

$$\mathcal{P}' = \begin{pmatrix} \Phi \\ \Psi \end{pmatrix},$$

where

$$\Psi = \mathcal{P}'(m, m+1, \ldots, n-1).$$

Since

$$I_n \;=\; \mathcal{P}'(\mathcal{P}')^T = \begin{pmatrix} \Phi\Phi^T & \Phi\Psi^T \\ \Psi\Phi^T & \Psi\Psi^T \end{pmatrix},$$

we obtain

$$\Phi\Phi^T = I_m.$$

Since Φ is an $m \times n$ matrix, the columns of Φ correspond to a tight frame in $\mathbb{R}^m$. □

Equal-norm frames

In the following theorem we establish a seeding of orthogonalized discrete GNN transforms that yields equal-norm frames in $\mathbb{R}^m$.

Theorem 5.4.3 *Consider discrete GNN transform $\mathcal{P}_{b,\alpha}$ that corresponds to the finite discrete GNN model* (4.16)*, for which all coefficients $b_k = 0$ in recursion* (4.3)*. Let*

$$\mathcal{P}' = D^{-1/2} \cdot \mathcal{P}_{b,\alpha}$$

denote the orthogonalized polynomial transform, where D is defined in (4.18)*. Define*

$$m = \lfloor \frac{n}{2} \rfloor.$$

Let matrix Φ be obtained by consecutively seeding $\mathcal{P}'_{b,\alpha}$ by rows:

$$\Phi = \mathcal{P}'(0, 1, \ldots, m-1)$$

Then the columns of Φ all have the same norm.

Proof: First, recall from Appendix A that orthogonal polynomials that satisfy recursion (4.3) with all coefficients $b_k = 0$, are either even or odd. In particular, roots $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}$ of $P_n(x)$ satisfy

$$\alpha_k = -\alpha_{n-1-k}$$

for $0 \leq k < n$.

Assume that n is even, and hence $m = n/2$. It follows from the definition (4.18) of D that the elements

of $\mathcal{P}'$ are either symmetric or antisymmetric around the middle row of the matrix. In other words, they satisfy

$$\begin{aligned} \mathcal{P}'_{k,2\ell} &= \mathcal{P}'_{n-1-k,2\ell}, \\ \mathcal{P}'_{k,2\ell+1} &= -\mathcal{P}'_{n-1-k,2\ell+1}. \end{aligned}$$

Let

$$\Phi = \mathcal{P}'(0, 1, \ldots, m-1)$$

be constructed from the first $m = n/2$ rows of $\mathcal{P}'$. Then the ℓ-th column of $\mathcal{P}'$ can be written as

$$\begin{pmatrix} \mathcal{P}'_{0,\ell} \\ \mathcal{P}'_{1,\ell} \\ \vdots \\ \mathcal{P}'_{m-1,\ell} \\ \mathcal{P}'_{m,\ell} \\ \vdots \\ \mathcal{P}'_{n-1,\ell} \end{pmatrix} = \begin{pmatrix} \Phi_{0,\ell} \\ \Phi_{1,\ell} \\ \vdots \\ \Phi_{m-1,\ell} \\ (-1)^\ell \Phi_{m-1,\ell} \\ \vdots \\ (-1)^\ell \Phi_{0,\ell} \end{pmatrix}.$$

Since $\mathcal{P}'_{b,\alpha}$ is an orthogonal matrix, the norm of each of its columns is 1. From this, we can compute the norm of the ℓ-th column of Φ as follows:

$$\begin{aligned} \sqrt{\sum_{i=0}^{m-1} \Phi_{i,\ell}^2} &= \sqrt{\frac{\sum_{i=0}^{m-1} \Phi_{i,\ell}^2 + \sum_{i=0}^{m-1} \Phi_{i,\ell}^2}{2}} \\ &= \sqrt{\frac{\sum_{i=0}^{m-1} (\mathcal{P}'_{i,\ell})^2 + \sum_{i=m}^{n-1} (\mathcal{P}'_{i,\ell})^2}{2}} \\ &= \frac{1}{\sqrt{2}}. \end{aligned}$$

Hence, all columns of Φ have the same norm $1/\sqrt{2}$.

The proof for $n = 2m + 1$ is analogous. □

Maximally robust frames

In the following theorem we establish a seeding of orthogonalized discrete GNN transforms that yields maximally robust to erasures frames in $\mathbb{R}^m$.

Theorem 5.4.4 *Consider discrete GNN transform $\mathcal{P}_{b,\alpha}$ that corresponds to the finite discrete GNN model* (4.16)*, for which all coefficients $b_k = 0$ in recursion* (4.3)*. Let*

$$\mathcal{P}' = D^{-1/2} \cdot \mathcal{P}_{b,\alpha}$$

denote the orthogonalized polynomial transform, where D is defined in (4.18)*. Define*

$$m = \lfloor \frac{n}{2} \rfloor.$$

Let matrix Φ be obtained by consecutively seeding $\mathcal{P}'_{b,\alpha}$ by rows:

$$\Phi = \mathcal{P}'(0, 1, \ldots, m-1)$$

Then the columns of Φ correspond to a maximally robust to erasures frame in $\mathbb{R}^m$.

Proof: The proof is by contradiction.

Let $n = 2m$. Let $\mathcal{P}'(:, \ell)$ denote the ℓ-th column of $\mathcal{P}'$, and $\Phi(:, \ell)$ denote the ℓ-th column of Φ. As follows from the proof of Theorem 5.4.3, we can write

$$\mathcal{P}'(:, \ell) = \begin{pmatrix} \Phi(:, \ell) \\ (-1)^\ell \cdot J_m \cdot \Phi(:, \ell) \end{pmatrix}, \tag{5.11}$$

where

$$J_m = \begin{pmatrix} & & & 1 \\ & & \iddots & \\ & 1 & & \\ 1 & & & \end{pmatrix} \in \mathbb{R}^{m \times m}$$

is a complementary diagonal matrix.

Assume that the columns of Φ do not correspond to a maximally robust frame. Then there must exist an $m \times m$ submatrix of Φ that is singular. Without loss of generality, assume that it is the submatrix constructed from first m columns $\Phi(:,0), \Phi(:,1), \ldots, \Phi(:,m-1)$. Then $\Phi(:,m-1)$ can be expressed as a linear combination of $\Phi(:,0), \Phi(:,1), \ldots, \Phi(:,m-2)$ as

$$\Phi(:,m-1) = d_0\Phi(:,0) + d_1\Phi(:,1) + d_{m-2}\Phi(:,m-2).$$

In this case, it follows immediately from (5.11) that

$$\mathcal{P}'(:,m-1) = d_0\mathcal{P}'(:,0) + d_1\mathcal{P}'(:,1) + d_{m-2}\mathcal{P}'(:,m-2),$$

which implies that $\mathcal{P}'$ is singular. However, it contradicts the fact that $\mathcal{P}'$ is an orthogonal matrix. Hence, every $m \times m$ submatrix of Φ must be invertible.

The proof for $n = 2m + 1$ is analogous. □

Filter banks

As follows from Theorems 5.4.2 through 5.4.4, the columns of matrix

$$\Phi = \mathcal{P}'(0, 1, \ldots, m-1),$$

where $m = \lfloor \frac{n}{2} \rfloor$, correspond to a tight, equal-norm maximally robust frame in $\mathbb{R}^m$. Examples of orthogonal polynomials that satisfy the requirements of these theorems include Chebyshev polynomials of the first and second kinds, Legendre polynomials, and normalized Hermite polynomials.

Example 5.4.5 *Consider the discrete GNN transform of the finite GNN model based on normalized Hermite polynomials that we discussed in Example 4.3.1. Since in this example* $n = 2m = 6$, *we select the first*

$m = 3$ rows from the orthogonalized transform, and obtain matrix

$$\Phi = \mathcal{P}'(0,1,2) = \begin{pmatrix} 0.0506 & -0.1681 & 0.3593 & -0.5523 & 0.6069 & -0.4082 \\ 0.2977 & -0.5624 & 0.5408 & -0.1306 & -0.3449 & 0.4082 \\ 0.6394 & -0.3943 & -0.2802 & 0.4217 & 0.1126 & -0.4082 \end{pmatrix}.$$

Each column in this matrix has norm $1/\sqrt{2}$.

The $n = 6$ columns of the above matrix are an equal-norm, tight, maximally robust to erasures frame in $\mathbb{R}^m = \mathbb{R}^3$.

The filter banks that correspond to the constructed frames are n-channel filter banks pictured in Fig. 3.2 (there, we set $M = n$). Since the frames are tight, the analysis and synthesis basis functions are the same:

$$\varphi_k^{(j)} = \tilde{\varphi}_k^{(j)}$$

for $0 \le j < n$. Hence, the expansion of a signal $s(x)$ in the coordinate form is

$$\mathbf{s} = \big(\dots, s_0, s_1, s_2, \dots\big) = \sum_k \langle \varphi_k^{(0)}, \mathbf{s} \rangle \varphi_k^{(0)} + \dots + \sum_k \langle \varphi_k^{(n-1)}, \mathbf{s} \rangle \varphi_k^{(n-1)}. \tag{5.12}$$

The support of $\varphi_k^{(j)}$ is

$$k + jm, k + jm + 1, \dots, k + (j+1)m - 1,$$

and has length m. The corresponding m coefficients are determined by the j-th column of matrix Φ.

Chapter 6

Fast Discrete GNN Transforms

In this section, we introduce a general approach to decomposing *any* polynomial transform into a product of structured matrices involving other polynomial transforms of smaller size. As we show, in some cases the proposed approach produces novel fast algorithms for discrete GNN transforms, as well as for well-known signal transforms, such as the standard DFT and DCT of type 4. This work has been submitted for publication in [98].

We derive fast algorithms using the algebraic construction called *module induction*. Induction is based on the notion of a *subalgebra* and an associated decomposition that is similar to the coset decomposition in group theory. Our approach generalizes the decomposition algorithms in [6].

6.1 Subalgebras and Their Structure

In this section we define and discuss the structure of subalgebras of $\mathcal{A} = \mathbb{C}[x]/p(x)$.

6.1.1 Definition

Choose a polynomial $r(x) \in \mathcal{A}$, and consider the space of polynomials in $r(x)$ with addition and multiplication performed modulo $p(x)$:

$$\mathcal{B} = \Big\{ \sum_{k \geq 0} c_k r^k(x) \mod p(x) \mid c_k \in \mathbb{C} \Big\}, \tag{6.1}$$

where all sums are finite. We call $\mathcal{B}$ the *subalgebra* of $\mathcal{A}$ *generated* by $r(x)$ and write this as $\mathcal{B} = \langle r(x) \rangle \leq \mathcal{A}$.

6.1.2 Structure

Given $r(x) \in \mathcal{A}$, we first determine the dimension of $\mathcal{B} = \langle r(x) \rangle$. Then we identify $\mathcal{B}$ with a polynomial algebra of the form $\mathbb{C}[y]/q(y)$ with a suitably chosen polynomial $q(y)$.

Let $\alpha = (\alpha_0, \ldots, \alpha_{n-1})$ be the list of roots of $p(x)$. The generator $r(x)$ maps α to the list $\beta = (\beta_0, \ldots, \beta_{m-1})$, such that for each $\alpha_k \in \alpha$ there is a $\beta_j \in \beta$, for which $r(\alpha_k) = \beta_j$. Hence, $m \leq n$, since for some k and ℓ we may have $r(\alpha_k) = r(\alpha_\ell)$.

Theorem 6.1.1 *The dimension of* $\mathcal{B} = \langle r(x) \rangle$ *is* $\dim \mathcal{B} = m = |\beta|$.

Proof: Let $d = \dim \mathcal{B}$. Since $\mathcal{B} \leq \mathcal{A}$, then $\dim \mathcal{B} \leq \dim \mathcal{A}$ and the polynomials $(1, r(x), \ldots, r^{n-1}(x))$ span the entire $\mathcal{B}$. From the isomorphism (2.19) we obtain

$$\begin{aligned} d &= \text{rank} \left(\Delta(1), \Delta(r(x)), \ldots, \Delta(r^{n-1}(x)) \right) \\ &= \text{rank} \left[r^\ell(\alpha_k) \right]_{0 \leq k, \ell < n} . \end{aligned}$$

Since $r(\alpha_k) \in \beta$ and $|\beta| = m$, the above matrix has only m different rows; hence, $d \leq m$. On the other hand, it contains the full-rank $m \times m$ Vandermonde matrix

$$\left[\beta_i^\ell \right]_{0 \leq i, \ell < m}$$

as a submatrix; hence, $d \geq m$. Thus, we conclude that $d = \dim \mathcal{B} = m$. □

Next, we identify $\mathcal{B}$ with a polynomial algebra.

Theorem 6.1.2 *The subalgebra* $\mathcal{B} = \langle r(x) \rangle$ *can be identified with the polynomial algebra* $\mathbb{C}[y]/q(y)$*, where*

$q(y) = \prod_{j=0}^{m-1}(y - \beta_j)$, *via the following canonical isomorphism of algebras:*

$$\begin{array}{rccl} \kappa: & \mathcal{B} & \rightarrow & \mathbb{C}[y]/q(y), \\ & r(x) & \mapsto & y. \end{array} \tag{6.2}$$

We indicate this canonical isomorphism as $\mathcal{B} \cong \mathbb{C}[y]/q(y)$.

Proof: Observe that $\mathcal{B}$ and $\mathbb{C}[y]/q(y)$ have the same dimension m, and κ maps the generator $r(x)$ of $\mathcal{B}$ to the generator y of $\mathbb{C}[y]/q(y)$. Hence, it suffices to show that $q(r(x)) \equiv 0 \mod p(x)$ in $\mathcal{B}$. From (2.19) we obtain

$$\begin{array}{rcl} \Delta(q(r(x))) & = & \begin{pmatrix} q(r(\alpha_0)) & \ldots & q(r(\alpha_{n-1})) \end{pmatrix}^T \\ & = & \begin{pmatrix} 0 & \ldots & 0 \end{pmatrix}^T, \end{array}$$

which implies that $q(r(x)) \equiv 0 \mod p(x)$ in $\mathcal{A}$, and hence in $\mathcal{B}$. □

Let $c = \big(q_0(y), \ldots, q_{m-1}(y)\big)$ be a basis of $\mathbb{C}[y]/q(y)$. The polynomial transform (2.20) that decomposes the regular module $\mathbb{C}[y]/q(y)$ (and hence the regular $\mathcal{B}$-module $\mathcal{B}$) is given by (2.19) as

$$\mathcal{P}_{c,\beta} = [q_\ell(\beta_i)]_{0 \le i,\ell < m}\,.$$

Example 6.1.3 *Consider the polynomial algebra* $\mathcal{A} = \mathbb{C}[x]/(x^4 - 1)$ *with* $\alpha = \big(1, -j, -1, j\big)$. *The polynomial* $r_1(x) = x^2$ *generates the subalgebra* $\mathcal{B}_1 = \big\langle r_1(x) \big\rangle \cong \mathbb{C}[y]/(y^2 - 1)$ *of dimension 2, since* $r_1(x)$ *maps* α *to* $\beta = \big(1, -1\big)$.

The polynomial $r_2(x) = (x + x^{-1})/2 = (x + x^3)/2$ *generates the subalgebra* $\mathcal{B}_2 = \big\langle r_2(x) \big\rangle \cong \mathbb{C}[y]/(y^3 - y)$ *of dimension 3, since* $r_2(x)$ *maps* α *to* $\beta = \big(1, 0, -1\big)$.

6.2 Module Induction

In this section we introduce the concept of *module induction*, which constructs an $\mathcal{A}$-module $\mathcal{M}$ from a $\mathcal{B}$-module $\mathcal{N}$ for a subalgebra $\mathcal{B} \le \mathcal{A}$. We will show that every regular $\mathcal{A}$-module is an induction, which is

the basis of our technique for polynomial transform decomposition.

6.2.1 Induction

Similar to the coset decomposition in group theory [18, 19], we can decompose a polynomial algebra $\mathcal{A} = \mathbb{C}[x]/p(x)$ using a subalgebra $\mathcal{B}$ and associated *transversal*:

Definition 6.2.1 (Transversal) *Let $\mathcal{B} \leq \mathcal{A}$ be a subalgebra of $\mathcal{A}$. A* transversal *of $\mathcal{B}$ in $\mathcal{A}$ is a list of polynomials $T = \big(t_0(x), \dots, t_{L-1}(x)\big) \subset \mathcal{A}$, such that, as vector spaces,*

$$\mathcal{A} = \bigoplus_{\ell=0}^{L-1} t_\ell(x)\mathcal{B} = t_0(x)\mathcal{B} \oplus \cdots \oplus t_{L-1}(x)\mathcal{B}. \tag{6.3}$$

Later, in Theorem 6.2.6, we establish necessary and sufficient conditions for a list of polynomials to be a transversal of $\mathcal{B}$ in $\mathcal{A}$. In particular, for any $\mathcal{B} \leq \mathcal{A}$ there always exists a transversal.

Given a transversal of $\mathcal{B}$ in $\mathcal{A}$, we define the module induction, which is analogous to the induction for group algebras in [19].

Definition 6.2.2 (Induction) *Let $\mathcal{B} \leq \mathcal{A}$ be a subalgebra of $\mathcal{A}$ with a transversal T as in* (6.3)*, and let $\mathcal{N}$ be a $\mathcal{B}$-module. Then the following construction is an $\mathcal{A}$-module:*

$$\mathcal{M} = \bigoplus_{\ell=0}^{L-1} t_\ell(x)\mathcal{N}, \tag{6.4}$$

where the direct sum is again of vector spaces. It is called the induction *of the $\mathcal{B}$-module $\mathcal{N}$ with the transversal T to an $\mathcal{A}$-module. We write this as $\mathcal{M} = \mathcal{N} \uparrow_T \mathcal{A}$.*

Here, we are primarily interested in regular modules. These are always inductions, as follows directly from (6.3) and (6.4):

Lemma 6.2.3 *Let $\mathcal{B} \leq \mathcal{A}$ with a transversal T. Then the regular module $\mathcal{A}$ is an induction of the regular module $\mathcal{B}$:*

$$\mathcal{A} = \mathcal{B} \uparrow_T \mathcal{A}. \tag{6.5}$$

6.2.2 Structure of Cosets

We have established in (6.2) that the subalgebra $\mathcal{B} \leq \mathcal{A}$, generated by $r(x) \in \mathcal{A}$, can be identified with a polynomial algebra $\mathbb{C}[y]/q(y)$. Next, we investigate the structure of each $\mathcal{B}$-module $t_\ell(x)\mathcal{B}$ in the induction (6.5).

Consider a polynomial $t(x) \in \mathcal{A}$. As in Theorem 6.1.2, let $r(x)$ map α to β, and let $q(y) = \prod_{j=0}^{m-1}(y - \beta_j)$. Further, let $\alpha' = \big(\alpha_k \mid t(\alpha_k) \neq 0\big) \subseteq \alpha$ be the sublist of α that consists of those α_k that are not roots of $t(x)$. Finally, let $r(x)$ map α' to $\beta' \subseteq \beta$, and denote $|\beta'| = m'$.

Theorem 6.2.4 *The dimension of $t(x)\mathcal{B}$ is* $\dim t(x)\mathcal{B} = |\beta'| = m'$.

Proof: The proof is similar to that of Theorem 6.1.1. The list of polynomials $\big(t(x), t(x)r(x), \ldots, t(x)r^{n-1}(x)\big)$ generates $t(x)\mathcal{B}$ as a vector space. Using the isomorphism Δ in (2.19) we obtain

$$\begin{aligned} \dim\Big(t(x)\mathcal{B}\Big) &= \text{rank}\,\Big(\Delta(t(x)), \Delta(t(x)r(x)), \ldots, \Delta(t(x)r^{n-1}(x))\Big) \\ &= \text{rank}\,\Big[t(\alpha_k)r^\ell(\alpha_k)\Big]_{0\leq k,\ell<n} \\ &= \text{rank}\Big(\,\text{diag}\,\Big(t(\alpha_k)\Big)_{0\leq k<n} \cdot \Big[r^\ell(\alpha_k)\Big]_{0\leq k,\ell<n}\Big). \end{aligned} \tag{6.6}$$

Theorem 6.1.2 shows that $\Big[r^\ell(\alpha_k)\Big]_{0\leq k,\ell<n}$ has exactly $m = |\beta|$ linearly independent rows of the form

$$\begin{pmatrix} 1 & \beta_i & \beta_i^2 & \ldots & \beta_i^{n-1} \end{pmatrix}.$$

For each β_i, the above row contributes exactly 1 to the rank of the matrix (6.6) if and only if there exists α_k such that $t(\alpha_k) \neq 0$ and $r(\alpha_k) = \beta_i$. Since there are exactly $|\beta'| = m'$ such values of β_i, we conclude that $\dim\big(t(x)\mathcal{B}\big) = m'$. □

Next, we identify the $\mathcal{B}$-module $t(x)\mathcal{B}$ with a $\mathbb{C}[y]/q(y)$-module.

Theorem 6.2.5 *The $\mathcal{B}$-module $t(x)\mathcal{B}$ can be identified with the $\mathbb{C}[y]/q(y)$-module $\mathbb{C}[y]/q'(y)$, where $q'(y) =$*

$\prod_{\beta_j \in \beta'} (y - \beta_j)$, *via the module isomorphism*

$$\begin{array}{rccl} \eta: & t(x)\mathcal{B} & \to & \mathbb{C}[y]/q'(y), \\ & t(x)r^k(x) & \mapsto & y^k. \end{array} \tag{6.7}$$

By a slight abuse of notation, we write $t(x)\mathcal{B} \cong \mathbb{C}[y]/q'(y)$. This is an isomorphism of modules and should not be confused with the isomorphism of algebras in Theorem 6.1.2.

Proof: It follows from Theorem 6.2.4 that $\big(t(x), t(x)r(x), \ldots, t(x)r^{m'-1}(x)\big)$ is a basis of $t(x)\mathcal{B}$, viewed as a vector space. On the other hand, $\big(1, y, \ldots, y^{m'-1}\big)$ is obviously a basis of $\mathbb{C}[y]/q'(y)$, also viewed as a vector space. Hence, η in (6.7) is a bijective linear mapping between $t(x)\mathcal{B}$ and $\mathbb{C}[y]/q'(y)$.

In order for η to be an isomorphism of modules, it must also be a module homomorphism—it must preserve the addition and multiplication in $t(x)\mathcal{B}$ and $\mathbb{C}[y]/q'(y)$. Namely, for $h(x) \in \mathcal{B}$ and $u(x), v(x) \in t(x)\mathcal{B}$, the following conditions must hold:

$$\begin{aligned} \eta\big(u(x) + v(x)\big) &= \eta(u(x)) + \eta(v(x)), \\ \eta\big(h(x)v(x)\big) &= \kappa(h(x)) \cdot \eta(v(x)). \end{aligned}$$

The first condition is trivial. To show that the second condition holds, let $h(x) = \sum_{k=0}^{m-1} h_k r^k(x) \in \mathcal{B}$ and $v(x) = \sum_{i=0}^{m'-1} v_i t(x) r^i(x) \in t(x)\mathcal{B}$. Then

$$\begin{aligned} \eta\big(h(x)v(x)\big) &= \eta\Big(\sum_{i=0}^{m+m'-2} \sum_{k=0}^{i} h_k v_{i-k} t(x) r^i(x)\Big) = \sum_{i=0}^{m+m'-2} \sum_{k=0}^{i} h_k v_{i-k} y^i \\ &= \sum_{k=0}^{m-1} h_k y^k \cdot \sum_{i=0}^{m'-1} v_i y^i = \kappa(h(x)) \cdot \eta(v(x)). \end{aligned}$$

Hence, η is a module isomorphism. □

Note that, depending on $t(x)$, the dimension of $t(x)\mathcal{B}$ may be smaller than the dimension of $\mathcal{B}$: $m' \leq m$. This effect is called *annihilation*.

Also, the definition of η in (6.7) assumes the standard basis $\big(1, y, \ldots, y^{m'-1}\big)$ in $\mathbb{C}[y]/q'(y)$. If another basis $\big(b_0(y), \ldots, b_{m'-1}(y)\big)$ were desired, the corresponding basis in $t(x)\mathcal{B}$ would be $\big(t(x)b_0(r(x)), \ldots, t(x)b_{m'-1}(r(x))\big)$.

As a consequence of Theorem 6.2.5 and the above discussion, decomposing the $\mathcal{B}$-module $t(x)\mathcal{B}$ with basis $\big(t(x)q_0(r(x)),\ldots,t(x)q_{m'-1}(r(x))\big)$ is the same as decomposing the $\mathbb{C}[y]/q(y)$-module $\mathbb{C}[y]/q'(y)$ with basis $c=\big(q_0(y),\ldots,q_{m'-1}(y)\big)$. The decomposition matrix is the same as for the regular module $\mathbb{C}[y]/q'(y)$ with the same basis, namely

$$\mathcal{P}_{c,\beta'}=[q_\ell(\beta_i)]_{0\le i,\ell<m'}\,. \tag{6.8}$$

6.2.3 Existence of a Transversal

Consider $T=\big(t_0(x),\ldots,t_{L-1}(x)\big)\subset\mathcal{A}$, and let $\dim\big(t_\ell(x)\mathcal{B}\big)=m_\ell$ for $0\le\ell<L$. Then $\big(t_\ell(x),t_\ell(x)r(x),\ldots,t_\ell(x)r^{m_\ell-}$ is a basis of $t_\ell(x)\mathcal{B}$, as follows from Theorem 6.2.4. Hence, T satisfies (6.3) if and only if $m_0+\cdots+m_{L-1}=n$ and the concatenation of bases

$$b'=\bigcup_{\ell=0}^{L-1}\Big(t_\ell(x),\ldots,t_\ell(x)r^{m_\ell-1}(x)\Big) \tag{6.9}$$

is a basis in $\mathcal{A}$. The following theorem states this condition in a matrix form.

Theorem 6.2.6 *Using previous notation, T is a transversal if and only if the following is a full-rank $n\times n$ matrix:*

$$M'=\Big(D_0B_0\quad|\quad D_1B_1\quad|\quad\ldots\quad|\quad D_{L-1}B_{L-1}\Big)\,, \tag{6.10}$$

where

$$D_\ell=\operatorname{diag}\Big(t_\ell(\alpha_k)\Big)_{0\le k<n},$$

and

$$B_\ell=\big[r^i(\alpha_k)\big]_{0\le k<n,0\le i<m_\ell}\,.$$

Proof: The proof is similar to the proofs of Theorems 6.1.1 and 6.2.4. Observe that the k-th element of b' in (6.9) is mapped to the k-th column of M' in (6.10) by the isomorphism Δ in (2.19). Hence, b' is a basis in $\mathcal{A}$ if and only if M' has exactly n columns and $\operatorname{rank}(M')=n$. □

It follows from Theorem 6.2.6 that for any algebra $\mathcal{A}$ and its subalgebra $\mathcal{B}$ there always exists a transversal. For example, we can choose $T = \big(t_0(x), \dots, t_{n-1}(x)\big)$, where $t_\ell(\alpha_k) = 0$ for $\ell \neq k$ and $t_\ell(\alpha_\ell) \neq 0$. In this case

$$M' = \operatorname{diag}\Big(t_\ell(\alpha_\ell)\Big)_{0 \leq \ell < n}$$

in (6.10) is a full-rank diagonal matrix.

Example 6.2.7

Consider the subalgebras constructed in Example 6.1.3.

For $\mathcal{B}_1 = \langle x^2 \rangle$ of dimension 2, we can choose the transversal $T = \big(1, x\big)$, since $\big(1, x^2\big) \cup \big(x, x^3\big)$ is a basis for $\mathcal{A}$. Since x maps α to $\big(1, -j, -1, j\big)$, we have $\alpha' = \big(1, -j, -1, j\big)$ and $\beta' = \big(1, -1\big)$. Hence, $q'(y) = (y-1)(y+1)$ and $x\mathcal{B}_1 \cong \mathbb{C}[y]/(y^2-1)$ is of dimension 2.

For $\mathcal{B}_2 = \big\langle (x + x^{-1})/2 \big\rangle$ of dimension 3, we can choose the transversal $T = \big(1, (x - x^{-1})/2\big)$, since the corresponding matrix

$$M' = \begin{pmatrix} 1 & 1 & 1 & \\ 1 & & & -j \\ 1 & -1 & 1 & \\ 1 & & & j \end{pmatrix}$$

from (6.10) has full rank. Since $(x - x^{-1})/2$ maps α to $\big(0, -j, 0, j\big)$, we obtain $\alpha' = \big(-j, j\big)$, $\beta' = \big(0\big)$, and thus $q'(y) = y$. Hence, $(x - x^{-1})/2 \cdot \mathcal{B}_2 \cong \mathbb{C}[y]/y$ is of dimension 1.

6.3 Decomposition of Polynomial Transforms Using Induction

In this section we use the induction (6.5) to express the polynomial transform of $\mathcal{A}$ via the polynomial transforms of each $t_\ell(x)\mathcal{B} \cong \mathbb{C}[y]/q'_\ell(y)$ in (6.3).

6.3.1 Decomposition

As before, we consider $\mathcal{A} = \mathbb{C}[x]/p(x)$, where $p(x) = \prod_{k=0}^{n-1}(x - \alpha_k)$. We view it as a regular $\mathcal{A}$-module with the chosen basis $b = \big(p_0(x), \dots, p_{n-1}(x)\big)$.

Let $\mathcal{B} = \langle r(x) \rangle \leq \mathcal{A}$ be a subalgebra generated by $r(x) \in \mathcal{A}$, and $\mathcal{B} \cong \mathbb{C}[y]/q(y)$ according to Theorem 6.1.2, where $q(y) = \prod_{j=0}^{m-1} (y - \beta_j)$ and $\beta = \big(\beta_0, \ldots, \beta_{m-1}\big)$.

Suppose $T = \big(t_0(x), \ldots, t_{L-1}(x)\big)$ is a transversal of $\mathcal{B}$ in $\mathcal{A}$. Let each $t_\ell(x)\mathcal{B}$ in (6.3) be identified with a $\mathbb{C}[y]/q(y)$-module $\mathbb{C}[y]/q^{(\ell)}(y)$ according to Theorem 6.2.5, where $q^{(\ell)}(y) = \prod_{\beta_j \in \beta^{(\ell)}} (y - \beta_j)$ and $m_\ell = |\beta^{(\ell)}|$. The basis $b^{(\ell)} = \big(b_0^{(\ell)}(y), \ldots, b_{m_\ell - 1}^{(\ell)}(y)\big)$ of $\mathbb{C}[y]/q^{(\ell)}(y)$ corresponds to the basis $\big(t_\ell(x) b_0^{(\ell)}(r(x)), \ldots, t_\ell(x) b_{m_\ell-1}^{(\ell)}(r(x))\big)$ of $t_\ell(x)\mathcal{B}$. Hence, the corresponding polynomial transform (6.8) is $\mathcal{P}_{b^{(\ell)}, \beta^{(\ell)}}$.

Theorem 6.3.1 *Given the induction* (6.5)*, the polynomial transform $\mathcal{P}_{b,\alpha}$ can be decomposed as*

$$\mathcal{P}_{b,\alpha} = \Big(D_0 M_0 \quad | \quad D_1 M_1 \quad | \quad \ldots \quad | \quad D_{L-1} M_{L-1} \Big) \Big(\bigoplus_{\ell=0}^{L-1} \mathcal{P}_{b^{(\ell)}, \beta^{(\ell)}} \Big) B. \tag{6.11}$$

Here, B is the base change matrix from the basis b to the concatenation of bases

$$\bigcup_{\ell=0}^{L-1} \big(t_\ell(x) b_0^{(\ell)}(r(x)), \ldots, t_\ell(x) b_{m_\ell-1}^{(\ell)}(r(x)) \big).$$

Each

$$D_\ell = \operatorname{diag}\Big(t_\ell(\alpha_k) \Big)_{0 \leq k < n}$$

is a diagonal matrix. Each M_ℓ is an $n \times m_\ell$ matrix whose (k,i)-th element is 1 if $r(\alpha_k)$ is equal to the i-th element

$$\beta_j^{(\ell)}$$

of $\beta^{(\ell)}$, and 0 otherwise. $\oplus$ denotes the direct sum of matrices:

$$\bigoplus_{\ell=0}^{L-1} \mathcal{P}_{b^{(\ell)}, \beta^{(\ell)}} = \begin{pmatrix} \mathcal{P}_{b^{(0)}, \beta^{(0)}} & & & \\ & \mathcal{P}_{b^{(1)}, \beta^{(1)}} & & \\ & & \ddots & \\ & & & \mathcal{P}_{b^{(L-1)}, \beta^{(L-1)}} \end{pmatrix}.$$

Proof: We prove the theorem for $L = 2$; that is, for $\mathcal{A} = t_0(x)\mathcal{B} \oplus t_1(x)\mathcal{B}$. The proof for arbitrary L is

analogous.

Let $\mathcal{B} \cong \mathbb{C}[y]/q(y)$ according to Theorem 6.1.2, where $q(y) = \prod_{i=0}^{m-1}(y - \beta_i)$ and $\beta = \big(\beta_0, \ldots, \beta_{m-1}\big)$. For $\ell \in \{0, 1\}$, let $t_\ell(x)\mathcal{B} \cong \mathbb{C}[y]/q^{(\ell)}(y)$ according to Theorem 6.2.5, where $q^{(\ell)}(y) = \prod_{\beta_i \in \beta^{(\ell)}}(y - \beta_i)$ and $m_\ell = |\beta^{(\ell)}|$. Let $b^{(\ell)} = \big(b_0^{(\ell)}(y), \ldots, b_{m_0-1}^{(\ell)}(y)\big)$ be a basis of $\mathbb{C}[y]/q^{(\ell)}(y)$.

Let $t_\ell(x)b^{(\ell)}(r(x)) = \big(t_\ell(x)b_0^{(\ell)}(r(x)), \ldots, t_\ell(x)b_{m_0-1}^{(\ell)}(r(x))\big)$. As we established in Theorem 6.2.6, $b' = t_0(x)b^{(0)}(r(x)) \bigcup t_1(x)b^{(1)}(r(x))$ is a basis of $\mathcal{A}$. The original basis b can be expressed in the new basis b' as $p_k(x) = \sum_{\ell=0}^{m_0-1} B_{k,\ell}t_0(x)b_\ell^{(0)}(r(x)) + \sum_{\ell=0}^{m_1-1} C_{k,\ell}t_1(x)b_\ell^{(1)}(r(x))$. Hence, if B is the base change matrix from b to b', then

$$\mathcal{P}_{b,\alpha} = \mathcal{P}_{b',\alpha} \cdot B. \tag{6.12}$$

The ℓ-th column of B is $(B_{0,\ell}, \ldots, B_{m_0-1,\ell}, C_{0,\ell}, \ldots, C_{m_1-1,\ell})^T$.

Next, observe that

$$\mathcal{P}_{b',\alpha} = \Big(\mathcal{P}_{t_0(x)b^{(0)}(r(x)),\alpha} \mid \mathcal{P}_{t_1(x)b^{(1)}(r(x)),\alpha}\Big). \tag{6.13}$$

For each ℓ, the (k, i)-th element of $\mathcal{P}_{t_\ell(x)b^{(\ell)}(r(x)),\alpha}$ is $t_\ell(\alpha_k)b_i^{(\ell)}(r(\alpha_k))$. Hence,

$$\mathcal{P}_{t_\ell(x)b^{(\ell)}(r(x)),\alpha} = D_\ell \cdot M_\ell \cdot \mathcal{P}_{b^{(\ell)},\beta^{(\ell)}}, \tag{6.14}$$

where M_ℓ is an $n \times m_\ell$ matrix whose (k, i)-th element is 1 if $r(\alpha_k)$ equals to the i-th element of $\beta^{(\ell)}$, and 0 otherwise; and

$$D_\ell = \operatorname{diag}\Big(t_\ell(\alpha_k)\Big)_{0 \le k \le n-1}.$$

Hence, from (6.12-6.14) we obtain the desired decomposition:

$$\mathcal{P}_{b,\alpha} = \Big(D_0M_0 \mid D_1M_1\Big) \cdot \Big(\mathcal{P}_{b^{(0)},\beta^{(0)}} \oplus \mathcal{P}_{b^{(1)},\beta^{(1)}}\Big) \cdot B. \tag{6.15}$$

□

Corollary 6.3.2 *Consider the $n \times m$ matrix M whose (k, i)-th element is 1 if $r(\alpha_k) = \beta_i$ and 0 otherwise. Then*

1. *M contains exactly n 1s and $n(m-1)$ 0s .*

2. *Each matrix M_ℓ in Theorem 6.3.1 is a submatrix of M. It contains the i-th column of M if and only if $\beta_i \in \beta^{(\ell)}$.*

3. *If the number of non-zero elements in the i-th column of M is c_i, then there are precisely c_i matrices among $M_0, \dots, M_{L-1}$ that contain this column.*

6.3.2 Discussion

The three factors in (6.11) correspond to the decomposition (2.19) of the regular module $\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/p(x)$ in three steps:

Step 1. $\mathcal{A}$ is represented as an induction (6.5) by changing the basis in $\mathcal{A}$ to the concatenation of bases $b^{(\ell)}$ of $t_\ell(x)\mathcal{B}$, using the base change matrix B.

Step 2. Each $t_\ell(x)\mathcal{B}$ is decomposed into a direct sum of irreducible $\mathcal{B}$-submodules, using the corresponding polynomial transform $\mathcal{P}_{b^{(\ell)},\beta^{(\ell)}}$.

Step 3. The resulting direct sum of irreducible $\mathcal{B}$-modules is decomposed into a direct sum of irreducible $\mathcal{A}$-modules, using the matrix M.

The factorization (6.11) is a fast algorithm for $\mathcal{P}_{b,\alpha}$ if the matrices B and M have sufficiently low costs, since the recursive nature of the second step allows for repeated application of Theorem 6.3.1. We illustrate this with two examples of novel algorithms derived using this theorem in Section 6.4.

6.3.3 Special Case: Factorization of p(x)

A special case of Theorem 6.3.1 has been derived in [2, 5]. Namely, assume that $\mathcal{A} = \mathbb{C}[x]/p(x)$, and we can decompose $p(x) = q(r(x))$. Then $\mathcal{B} = \langle r(x) \rangle \cong \mathbb{C}[y]/q(y)$, and any basis $t = \big(1, t_1(x), \dots, t_{k-1}(x)\big)$ of $\mathbb{C}[x]/r(x)$ is a transversal of $\mathcal{B}$ in $\mathcal{A}$. This leads to the following result.

Corollary 6.3.3 *Choose $c = \big(c_0(y), \dots, c_{m-1}(y)\big)$ as the basis of $\mathbb{C}[y]/q(y)$. Denote the roots of $r(x) - \beta_i$ as $\gamma^{(i)} = \big(\gamma_0^{(i)}, \dots, \gamma_{k-1}^{(i)}\big)$. Notice that $\bigcup_{i=0}^{m-1} \big(\gamma_0^{(i)}, \dots, \gamma_{k-1}^{(i)}\big)$ is simply a permutation of $\big(\alpha_0, \dots, \alpha_{n-1}\big)$, and denote the corresponding permutation matrix as P. Then, the polynomial transform decomposition* (6.11) *has the form*

$$\mathcal{P}_{b,\alpha} = P^{-1}\Big(\bigoplus_{i=0}^{m-1} \mathcal{P}_{t,\gamma^{(i)}}\Big) L_m^n \Big(I_k \otimes \mathcal{P}_{c,\beta}\Big) B. \tag{6.16}$$

Here, $\otimes$ denotes the tensor product of matrices.

Corollary 6.3.3 has been used to derive a large class of fast algorithms for real and complex DFTs, and DCTs and DSTs [2,5,61]. Theorem 6.3.1 further generalizes this approach, and, as we show in the following example and in Section 6.4, also yields fast algorithms not based on Corollary 6.3.3.

Example 6.3.4

Consider the polynomial algebra $\mathcal{A} = \mathbb{C}[x]/(x^4 - 1)$ with basis $b = \big(1, x, x^2, x^3\big)$. As we showed in Section 2.3, the corresponding polynomial transform is $\mathcal{P}_{b,\alpha} = \mathrm{DFT}_4$.

We continue from Example 6.2.7. First, consider $\mathcal{B}_1 = \langle x^2 \rangle$ and the induction $\mathcal{A} = \mathcal{B}_1 \oplus x\mathcal{B}_1$. Let us choose $b^{(0)} = \big(1, y\big)$ as the basis of $\mathbb{C}[y]/(y^2 - 1) \cong \mathcal{B}_1$; it corresponds to the basis $\big(1, x^2\big)$ of $\mathcal{B}_1$. We then choose $b^{(1)} = \big(1, y\big)$ as the basis of $\mathbb{C}[y]/(y^2 - 1) \cong x\mathcal{B}_1$; it corresponds to the basis $\big(x, x^3\big)$ of $x\mathcal{B}_1$. According to Theorem 6.3.1,

$$D_0 = \mathrm{diag}\Big(1, 1, 1, 1\Big), \quad D_1 = \mathrm{diag}\Big(1, -j, -1, j\Big),$$

and

$$M_0 = M_1 = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix}, \quad \mathcal{P}_{b^{(0)},\beta^{(0)}} = \mathcal{P}_{b^{(1)},\beta^{(1)}} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \mathrm{DFT}_2,$$

and B is the base change matrix from $\big(1, x, x^2, x^3\big)$ to $\big(1, x^2\big) \cup \big(x, x^3\big)$. Hence,

$$\mathrm{DFT}_4 = \begin{pmatrix} 1 & & 1 & \\ & 1 & & -j \\ 1 & & -1 & \\ & 1 & & j \end{pmatrix} \begin{pmatrix} \mathrm{DFT}_2 & \\ & \mathrm{DFT}_2 \end{pmatrix} \begin{pmatrix} 1 & & & \\ & & 1 & \\ & 1 & & \\ & & & 1 \end{pmatrix}. \tag{6.17}$$

As we show in Section 6.4.2, (6.17) is exactly the Cooley-Tukey FFT for DFT_4 [55].

Next, consider $\mathcal{B}_2 = \langle (x + x^{-1})/2 \rangle$ and the induction $\mathcal{A} = \mathcal{B}_2 \oplus (x - x^{-1})/2 \cdot \mathcal{B}_2$. Let us choose $b^{(0)} = \big(T_0(y), T_1(y), T_2(y)\big) = \big(1, y, 2y^2 - 1\big)$ as the basis of $\mathbb{C}[y]/(y^3 - y) \cong \mathcal{B}_2$; it corresponds to the basis $\big(1, (x + x^{-1})/2, (x^2 + x^{-2})/2\big)$ of $\mathcal{B}_2$. We then choose $b^{(1)} = \big(1\big)$ as the basis of $\mathbb{C}[y]/y \cong (x - x^{-1})/2 \cdot \mathcal{B}_2$; it corresponds to the basis $\big((x - x^{-1})/2\big)$ of $(x - x^{-1})/2 \cdot \mathcal{B}_2$. According to Theorem 6.3.1,

$$D_0 = \operatorname{diag}\Big(1, 1, 1, 1\Big), \quad D_1 = \operatorname{diag}\Big(0, -j, 0, j\Big), \quad \mathcal{P}_{b^{(1)},\beta^{(1)}} = (1) = \text{DST-I}_1,$$

and

$$M_0 = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & & 1 \\ & & 1 & \end{pmatrix}, M_1 = \begin{pmatrix} & 1 & & \\ & & & \\ & & & \\ & & 1 & \end{pmatrix}, \mathcal{P}_{b^{(0)},\beta^{(0)}} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & & -1 \\ 1 & -1 & 1 \end{pmatrix} = \text{DCT-I}_3,$$

and B is the base change matrix from $\big(1, x, x^2, x^3\big)$ to $\big(1, (x + x^{-1})/2, (x^2 + x^{-2})/2\big) \cup \big((x - x^{-1})/2\big)$. Hence,

$$\text{DFT}_4 = \begin{pmatrix} 1 & & & \\ & 1 & & -j \\ & & 1 & \\ & 1 & & j \end{pmatrix} \begin{pmatrix} \text{DCT-I}_3 & \\ & \text{DST-I}_1 \end{pmatrix} \begin{pmatrix} 1 & & & \\ & 1 & & 1 \\ & & 1 & \\ & 1 & & -1 \end{pmatrix}. \tag{6.18}$$

As we show in Section 7.1.1, (6.18) is the Britanak-Rao algorithm for DFT_4 [59].

6.4 Examples

In this section we derive fast algorithms for the discrete Fourier and cosine transforms. We apply Theorem 6.3.1 to express the transform as a product (6.11). The transforms in the resulting decomposition all have $O(n \log n)$ cost, and all other matrices have $O(n)$ cost. Hence, the overall algorithm cost is $O(n \log n)$.

6.4.1 Notation

Hereafter, we use the following special matrices:

I_n is the identity matrix of size n.

J_n is the complementary identity matrix of size n: its $(k, n-1-k)$-th element is 1 for $0 \leq k < n$, and 0 otherwise.

$\mathbf{1}_n = \begin{pmatrix} 1 & 1 & \dots & 1 \end{pmatrix}^T$ is a column vector of n ones.

Z_n is the $n \times n$ circular shift matrix:

$$Z_n = \begin{pmatrix} & & & 1 \\ & & & \\ & I_{n-1} & & \\ & & & \end{pmatrix}.$$

L^n_k, where k divides n, is an $n \times n$ permutation matrix that selects elements of $0, 1, \dots, n-1$ at the stride k; the corresponding permutation is $ik + j \mapsto jm + i$, where $0 \leq i < m$ and $0 \leq j < k$. The (i, j)-th element of L^n_k is 1 if $j = \lfloor \frac{ik(n+1)}{n} \rfloor \mod n$, and 0 otherwise.

$K^n_k = (I_m \oplus J_m \oplus I_m \oplus \dots) L^n_k$, where k divides n, is another permutation matrix.

$T^n_k = \operatorname{diag}\Big(\Big(w_n^{i\ell} \mid 0 \leq i < k, 0 \leq \ell < m\Big)\Big)$, where the index i runs faster, and $n = km$, is a twiddle factor matrix used in the Cooley-Tukey FFT.

Complementary direct sum:

$$\oslash_{i=0}^{m-1} A_i = \begin{pmatrix} & & & A_0 \\ & & \iddots & \\ & & & \\ A_{m-1} & & & \end{pmatrix}.$$

6.4.2 Cooley-Tukey FFT

In this section, we derive the general-radix Cooley-Tukey FFT using Theorem 6.3.1. As was shown in [2], Corollary 6.3.3 is sufficient in this case.

Consider $\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/(x^n - 1)$. Let $b = \big(1, x, \dots, x^{n-1}\big)$ be the basis of $\mathcal{M}$. As we showed in Section 2.3, the corresponding polynomial transform is DFT_n. Assume $n = km$. Let $r(x) = x^k$, and $\mathcal{B} = \big\langle r(x) \big\rangle$. Then $x^\ell \mathcal{B} \cong \mathbb{C}[y]/(y^m - 1)$, for $\ell = 0 \dots k-1$, and $\mathcal{A} = \oplus_{\ell=0}^{k-1} x^\ell \mathcal{B}$. Choosing the same basis $b^{(\ell)} = \{1, y, \dots, y^{m-1}\}$ in each $\mathbb{C}[y]/(y^m - 1) \cong x^\ell \mathcal{B}$ yields $\mathcal{P}_{b^{(\ell)}, \beta^{(\ell)}} = \mathrm{DFT}_m$. By Theorem 6.3.1, we

obtain

$$\mathrm{DFT}_{km} \quad = \quad M \cdot (I_k \otimes \mathrm{DFT}_m) \cdot B.$$

Here, $B = L^{km}_k$ and $M = (D_0 M_0 | \ldots | D_{k-1} M_0)$, where $M_0 = \mathbf{1}_k \otimes I_m$, and $D_\ell = \mathrm{diag}\left(\omega^{\ell i}_{km}\right)_{0 \le i < km}$ for $0 \le \ell < k$. Hence, we can rewrite

$$M = L^{km}_k \left(I_m \otimes \mathrm{DFT}_k\right) T^{km}_k L^{km}_m.$$

to obtain the well-known general-radix Cooley-Tukey FFT algorithm [5, 55]:

$$\begin{aligned} \mathrm{DFT}_{km} &= L^{km}_k \left(I_m \otimes \mathrm{DFT}_k\right) T^{km}_k L^{km}_m \left(I_k \otimes \mathrm{DFT}_m\right) L^{km}_k \\ &= L^{km}_k \left(I_m \otimes \mathrm{DFT}_k\right) T^{km}_k \left(\mathrm{DFT}_m \otimes I_k\right). \end{aligned} \tag{6.19}$$

6.4.3 Good-Thomas FFT

In this section, we derive the general-radix Good-Thomas FFT using Theorem 6.3.1. Similarly to Section 6.4.2, Corollary 6.3.3 is sufficient in this case.

Consider $\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/(x^n - 1)$. Let $b = \left(1, x, \ldots, x^{n-1}\right)$ be the basis of $\mathcal{M}$. As we showed in Section 2.3, the corresponding polynomial transform is DFT_n. Assume $n = km$, such that $\gcd(k, m) = 1$. Let $r(x) = x^k$ and $\mathcal{B} = \langle r(x) \rangle$. Then $x^{\ell m}\mathcal{B} \cong \mathbb{C}[y]/(y^m - 1)$, for $\ell = 0 \ldots k-1$, and $\mathcal{A} = \oplus_{\ell=0}^{k-1} x^{\ell m}\mathcal{B}$. Choosing the same basis $b^{(\ell)} = \{1, y, \ldots, y^{m-1}\}$ in each $\mathbb{C}[y]/(y^m - 1) \cong x^{\ell m}\mathcal{B}$ yields $\mathcal{P}_{b^{(\ell)}, \beta^{(\ell)}} = \mathrm{DFT}_m$. By Theorem 6.3.1, we obtain

$$\mathrm{DFT}_{km} \quad = \quad M \cdot (I_k \otimes \mathrm{DFT}_m) \cdot B.$$

B is the permutation matrix that maps the list $\left(0, 1, \ldots, km-1\right)$ into $\left(i_k k + i_m m\right)$, where indices $0 \le i_k < m$, $0 \le i_m < k$, and i_k runs faster. This mapping is known as *Ruritanian* or *Good's* mapping [99]. Further,

$$M = (D_0 M_0 | \ldots | D_{k-1} M_0),$$

where $M_0 = \mathbf{1}_k \otimes I_m$, and $D_\ell = \mathrm{diag}\left(\omega_k^{\ell i} \mid 0 \leq i < km\right)$ for $0 \leq \ell < k$. We can rewrite

$$M = PL_m^{km}\left(\mathrm{DFT}_k \otimes I_m\right),$$

where P is the permutation matrix for the *CRT* mapping [57] that maps the list $\big(0, 1, \ldots, km-1\big)$ into $\big(i_k k^{-1} k + i_m m^{-1} m\big)$; here, indices $0 \leq i_k < m$, $0 \leq i_m < k$, and i_k runs faster. k^{-1} is the multiplicative inverse of k modulo m; and m^{-1} is the multiplicative inverse of m modulo k.

Hence, we obtain the Good-Thomas (or prime-factor) FFT algorithm [99]:

$$\begin{aligned} \mathrm{DFT}_{km} &= PL_m^{km}\left(\mathrm{DFT}_k \otimes I_m\right)\left(I_k \otimes \mathrm{DFT}_m\right) B \\ &= PL_m^{km}\left(\mathrm{DFT}_k \otimes \mathrm{DFT}_m\right) B. \end{aligned}$$

6.5 Fast Discrete GNN Transforms

In this section, we apply Theorem 6.3.1 to construct decompositions of discrete GNN Fourier transforms. In particular, we identify classes of finite discrete GNN models of the form $\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/p(x)$ that can be decomposed into the induction of the form $\mathcal{A} = \mathcal{B} \oplus t(x)\mathcal{B}$, where $\mathcal{B}$ itself is a polynomial algebra that corresponds to a finite discrete GNN model. We then use this induction to decompose the corresponding polynomial transform $\mathcal{P}_{b,\alpha}$ of $\mathcal{A}$ into the direct sum of two polynomial transform of $\mathcal{B}$, which are discrete GNN transforms as well. We demonstrate that if $\mathcal{A}$ satisfies specific conditions, then the cost of such decomposition is linear.

We then derive the conditions that the polynomial algebra $\mathcal{B}$ has to satisfy such that Theorem 6.3.1 can then be applied recursively. As a result, we identify a class of finite discrete GNN models that possess a fast $O(n \log_2 n)$ computational algorithm for their corresponding polynomial transforms.

6.5.1 Preliminaries

Let $\big(P_k(x)\big)_{k \geq 0}$ be a family of orthogonal polynomials that satisfy the recursion

$$xP_k(x) = a_{k-1}P_{k-1}(x) + b_k P_k(x) + a_k P_{k+1}(x), \tag{6.20}$$

with $P_0(x) = 1$ and $P_1(x) = (x - b_0)/a_0$.

Consider the finite discrete GNN model

$$\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/P_n(x), \tag{6.21}$$

and let $b = \big(P_k(x)\big)_{0 \le k < n}$ be the basis of $\mathcal{M}$. The corresponding polynomial transform (4.17) is $\mathcal{P}_{b,\alpha}$. Let us denote the corresponding shift matrix (4.20) as

$$S = \phi(x) = \begin{pmatrix} b_0 & a_0 & & & \\ a_0 & b_1 & a_1 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-3} & b_{n-2} & a_{n-2} \\ & & & a_{n-2} & b_{n-1} \end{pmatrix}. \tag{6.22}$$

Lemma 6.5.1 *Let $b_k = 0$ for all $0 \le k < K$ in* (6.20)*. Then for any constant $d \in \mathbb{R}$ polynomials $P_2k(x)$, $0 \le k < \lfloor K/2 \rfloor$, can be expressed as*

$$P_{2k}(x) = Q_k(x^2 - d), \tag{6.23}$$

where polynomials $Q_k(y)$ are also orthogonal polynomials that satisfy the recursion

$$yQ_k(y) = a_{2k-2}a_{2k-1}Q_{k-1}(y) + (a_{2k-1}^2 + a_{2k}^2 - d)Q_k(y) + a_{2k}a_{2k+1}Q_{k+1}(y), \tag{6.24}$$

with $Q_0(y) = 1$ and $Q_1(y) = (y - a_0^2 + d)/a_0a_1$.

Proof: The proof is by induction. By definition,

$$\begin{aligned} P_0(x) &= 1 = Q_0(x^2 - d, \\ P_2(x) &= (x^2 - a_0^2)/a_0a_1 = Q_1(x^2 - d). \end{aligned}$$

Assume that (6.23) hold for all $k \leq k_0$. Then, from (6.20) we obtain

$$(x^2-d)P_{2k_0}(x) = a_{2k_0-2}a_{2k_0-1}P_{2k_0-2}(x) + (a_{2k_0-1}^2 + a_{2k_0}^2 - d)P_{2k_0}(x) + a_{2k_0}a_{2k_0+1}P_{2k_0+2}(x)$$

and hence,

$$\begin{aligned} P_{2k_0+2}(x) &= \frac{(x^2-d)-(a_{2k_0-1}^2+a_{2k_0}^2-d)}{a_{2k_0}a_{2k_0+1}}P_{2k_0}(x) - \frac{a_{2k_0-2}a_{2k_0-1}}{a_{2k_0}a_{2k_0+1}}P_{2k_0-2}(x) \\ &= \frac{(x^2-d)-(a_{2k_0-1}^2+a_{2k_0}^2-d)}{a_{2k_0}a_{2k_0+1}}Q_{k_0}(x^2-d) - \frac{a_{2k_0-2}a_{2k_0-1}}{a_{2k_0}a_{2k_0+1}}Q_{k_0-1}(x^2-d) \\ &= Q_{k_0+1}(x^2-d), \end{aligned}$$

since from (6.24) we have

$$Q_{k+1}(y) = \frac{y-(a_{2k-1}^2+a_{2k}^2-d)}{a_{2k}a_{2k+1}}Q_k(y) - \frac{a_{2k-2}a_{2k-1}}{a_{2k}a_{2k+1}}Q_{k-q}(y).$$

□

6.5.2 Decomposition of Discrete GNN Fourier Transforms of Even Sizes

We identify a class of finite discrete GNN models of the form $\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/p(x)$, where $\deg p(x) = 2m$ is even, that can be decomposed into the induction of the form $\mathcal{A} = \mathcal{B} \oplus t(x)\mathcal{B}$. Here, $\mathcal{B}$ itself is a polynomial algebra that corresponds to a finite discrete GNN model; and either $t(x) = x$ or $t(x) = x^{-1}$.

Decomposition using transversal $\left(1, x^{-1}\right)$

Theorem 6.5.2 *Let $b_k = 0$ for $0 \leq k < n$ in* (6.20)*. Consider the corresponding signal model* (6.21)*, and assume that $n = 2m$.*

Let $\mathcal{P}_{q,\beta}$ be the polynomial transform corresponding to the signal model with the polynomial algebra $\mathbb{C}[y]/Q_m(y)$ with basis $q = \left(Q_k(y)\right)_{0 \leq k < m}$.

Then the polynomial transform $\mathcal{P}_{b,\alpha}$ can be decomposed as follows:

$$\mathcal{P}_{b,\alpha} = \begin{pmatrix} J_m & -J_m \\ I_m & I_m \end{pmatrix} \cdot \left(I_m \oplus \operatorname{diag}\left(\alpha_k^{-1} \right)_{m \le k < n} \right) \cdot \left(I_2 \otimes \mathcal{P}_{q,\beta} \right) \cdot B. \tag{6.25}$$

Here,

$$B = \left(I_m \oplus S' \right) \cdot L_2^n,$$

where S' is the submatrix of S in (6.22) *constructed from its even rows (with indices $0, 2, \ldots, n-2$) and odd columns (with indices $1, 3, \ldots, n-1$).*

Proof: It follows from Lemma 6.5.1 that, for $d = 0$, $P_n(x) = Q_m(x^2)$, where $Q_k(x)$ satisfy the recursion

$$yQ_k(y) = a_{2k-2}a_{2k-1}Q_{k-1}(y) + (a_{2k-1}^2 + a_{2k}^2)Q_k(y) + a_{2k}a_{2k+1}Q_{k+1}(y),$$

with $Q_0(y) = 1$ and $Q_1(y) = (y - a_0^2)/a_0a_1$. In particular, $P_n(x)$ contains only even powers of x.

Let $\alpha_0 < \alpha_1 < \ldots < \alpha_{n-1}$ be the n distinct roots of $P_n(x)$ (since it is a separable polynomial). Then $\alpha_k = -\alpha_{n-1-k}$ for $0 \le k < n$. Moreover, 0 is not a root of $P_n(x)$: $P_n(0) \neq 0$.

Consider the polynomial $r(x) = x^2 \in \mathcal{A}$ and the subalgebra $\mathcal{B} = \langle r(x) \rangle$ generated by this polynomial. $r(x)$ maps $\alpha = \left(\alpha_k\right)_{0 \le k < n}$ to $\beta = \left(\beta_i\right)_{0 \le i < m}$, where $\beta_i = \alpha_i^2 = \alpha_{n-1-i}^2$, By Theorem 6.1.2, we obtain $\mathcal{B} \cong \mathbb{C}[y]/Q_m(y)$. Let $b^{(0)} = \left(Q_k(y)\right)_{0 \le k < m}$ be the basis of $\mathbb{C}[y]/Q_m(y)$. It corresponds to the basis $\left(P_{2k}(x)\right)_{0 \le k < m}$ of $\mathcal{B}$.

Furthermore, consider the polynomial $t(x) = (P_n(0) - P_n(x))/xP_n(0) \in \mathcal{A}$. Observe that $t(x)$ is indeed a polynomial, and $t(\alpha_k) = \alpha_k^{-1}$ for any α_k. Thus,

$$t(x) \equiv x^{-1} \mod P_n(x).$$

In the remainder of the proof, we will use $t(x) = x^{-1}$.

We can use Theorem 6.2.6 to verify that $T = \left(1, t(x)\right)$ is a transversal of $\mathcal{B}$ in $\mathcal{A}$, such that $\mathcal{A} = \mathcal{B} \oplus t(x) \cdot \mathcal{B}$. From Theorem 6.2.5 we obtain $t(x)\mathcal{B} \cong \mathbb{C}[y]/Q_m(y)$. Let $b^{(1)} = \left(t(x)Q_k(r(x))\right)_{0 \le k < m}$ be the basis of $\mathbb{C}[y]/Q_m(y)$. It corresponds to the basis $\left(x^{-1}P_{2k}(x)\right)_{0 \le k < m}$ of $t(x)\mathcal{B}$.

Hence, from Theorem 6.3.1 we obtain the following decomposition:

$$\mathcal{P}_{b,\alpha} = M \cdot \mathcal{P}_{b(0),\beta} \oplus \mathcal{P}_{b(1),\beta} \cdot B.$$

Here, B is the base change matrix from b to $b^{(0)} \cup b^{(1)}$. Observe that

$$P_{2k+1} = a_{2k} x^{-1} P_{2k} + a_{2k+1} x^{-1} P_{2k+2}.$$

Hence,

$$B = \Big(I_m \oplus S' \Big) \cdot L_2^n,$$

where S' is the submatrix of S in (6.22) constructed from its even rows (with indices $0, 2, \ldots, n-2$) and odd columns (with indices $1, 3, \ldots, n-1$).

Since $\beta_i = \alpha_i^2 = \alpha_{n-1-i}^2$, it follows from the construction of the matrix M in Theorem 6.3.1 that

$$M = \begin{pmatrix} J_m & -J_m \\ I_m & I_m \end{pmatrix} \cdot \Big(I_m \oplus \operatorname{diag}\Big(\alpha_k^{-1} \Big)_{m \le k < n} \Big).$$

Finally, by construction, $\mathcal{P}_{b^{(0)},\beta} = \mathcal{P}_{b^{(1)},\beta} = \mathcal{P}_{q,\beta}$. □

Decomposition using transversal $(1, x)$

Alternatively, we can use a different transversal $(1, x)$ for the subalgebra $\mathcal{B} < \mathcal{A}$ constructed in Theorem 6.5.2. This approach leads to a different decomposition of $\mathcal{P}_{b,\alpha}$, as we demonstrate in the following theorem.

Theorem 6.5.3 *Let $b_k = 0$ for $0 \le k < n$ in* (6.20)*. Consider the corresponding signal model* (6.21)*, and assume that $n = 2m$.*

Let $\mathcal{P}_{q,\beta}$ be the polynomial transform corresponding to the signal model with the polynomial algebra $\mathbb{C}[y]/Q_m(y)$ with basis $q = \big(Q_k(y)\big)_{0 \le k < m}$.

Then the polynomial transform $\mathcal{P}_{b,\alpha}$ can be decomposed as follows:

$$\mathcal{P}_{b,\alpha} = \begin{pmatrix} J_m & -J_m \\ I_m & I_m \end{pmatrix} \cdot \Big(I_m \oplus \operatorname{diag}\Big(\alpha_k\Big)_{m \le k < n}\Big) \cdot \Big(I_2 \otimes \mathcal{P}_{q,\beta}\Big) \cdot B. \tag{6.26}$$

Here,

$$B = \Big(\prod_{i=0}^{m-1} B_i \Big) \cdot L_2^n,$$

Here, B_i is an identity matrix except its $(m+1+i, m+1+i)$-th and $(m+i, m+1+i)$-th elements are equal to $1/a_{2k}$ and $-a_{2k-1}/a_{2k}$, respectively.

Proof: The proof is identical to the proof of Theorem 6.5.2. The only difference is that for the construction of $\mathcal{B}$ we use the property

$$P_{2k+1}(x) = \frac{x}{a_{2k}} P_{2k}(x) - \frac{a_{2k-1}}{a_{2k}} P_{2k-1}(x).$$

We use this property to compute the bases change matrix $\mathcal{B}$ in steps that correspond to the matrices $B_0, B_1, \ldots, B_{m-1}$. □

Generalization to other finite discrete GNN models

The factorizations of $\mathcal{P}_{b,\alpha}$ constructed in Theorems 6.5.2 and 6.5.3 can be generalized to a larger class of finite discrete GNN models.

Theorem 6.5.4 *Let $b_k = d$ for $0 \le k < n$ in* (6.20)*, where $d \in \mathbb{R}$ is an arbitrary constant. Consider the corresponding signal model* (6.21)*, and assume that $n = 2m$.*

In addition, consider another family of orthogonal polynomials $\big(\tilde{P}_k(x)\big)_{k \ge 0}$ that satisfy (6.20) *with $b_k = 0$ for $0 \le k < n$, and the same a_k as for $P_k(x)$ above. Denote the corresponding signal model* (6.21) *as $\tilde{\mathcal{A}} = \tilde{\mathcal{M}} = \mathbb{C}[x]/\tilde{P}_n(x)$, and fix the basis $\tilde{b} = \big(\tilde{P}_k(x)\big)_0 \le k < n$. Denote the zeros of $\tilde{P}_n(x)$ as $\tilde{\alpha} = \big(\tilde{\alpha_0}, \ldots, \tilde{\alpha_{n-1}}\big)$.*

Then the polynomial transform $\mathcal{P}_{b,\alpha}$ can be computed as follows:

$$\mathcal{P}_{b,\alpha} = \operatorname{diag}\Big(\sqrt{\frac{P_{n-1}(\alpha_k) P_n'(\alpha_k)}{\tilde{P}_{n-1}(\alpha_k - d) \tilde{P}_n'(\alpha_k - d)}} \Big) \cdot \mathcal{P}_{\tilde{b},\tilde{\alpha}}. \tag{6.27}$$

Proof: Observe that the shift matrix S in (6.22) for the signal model $\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/P_n(x)$ and the shift matrix $\tilde{S}$ for the signal model $\tilde{\mathcal{A}} = \tilde{\mathcal{M}} = \mathbb{C}[x]/\tilde{P}_n(x)$ are related as

$$S = \tilde{S} + dI_n.$$

Next, observe that S is symmetric. Hence, it has an orthogonal eigenvector matrix V, and can be factored as

$$S = V \cdot \mathrm{diag}\left(\alpha_0, \ldots, \alpha_{n-1}\right) \cdot V^T.$$

It follows from Section 4.3 that

$$V = \mathcal{P}_{b,\alpha}^T D^{1/2},$$

where

$$D = \mathrm{diag}\left(\frac{1}{P_{n-1}(\alpha_k)P_n'(\alpha_k)}\right)_{0 \le k < n}.$$

The same applies to $\tilde{S}$. It also has an orthogonal eigenvector matrix $\tilde{V}$, and can be factored as

$$\tilde{S} = \tilde{V} \cdot \mathrm{diag}\left(\tilde{\alpha}_0, \ldots, \tilde{\alpha}_{n-1}\right) \cdot \tilde{V}^T,$$

where

$$\tilde{V} = \mathcal{P}_{\tilde{b},\tilde{\alpha}}^T \tilde{D}^{1/2},$$

and

$$\tilde{D} = \mathrm{diag}\left(\frac{1}{\tilde{P}_{n-1}(\tilde{\alpha}_k)\tilde{P}_n'(\tilde{\alpha}_k)}\right)_{0 \le k < n}.$$

It follows from $S = \tilde{S} + dI_n$ that $V = \tilde{V}$ and $\alpha_k = \tilde{\alpha_k} + d$ for all k [100, 101]. Hence

$$\mathcal{P}_{b,\alpha}^T D^{1/2} = \mathcal{P}_{\tilde{b},\tilde{\alpha}}^T \tilde{D}^{1/2},$$

from which we immediately derive (6.27). □

Operational cost analysis. The computation of the matrices M and B in Theorems 6.5.2 and 6.5.3 requires $3n/2$ operations each. In addition, the computation of the diagonal matrix D in Theorem 6.5.4 requires n operations. Hence, the factorization (6.27) allows us to compute $\mathcal{P}_{b,\alpha}$ in $4n+2C(n/2)$ operations (additions and multiplications), where $C(n/2)$ denotes the number of operations required to compute the polynomial transform $\mathcal{P}_{q,\beta}$ of size $n/2$.

In general, we cannot apply Theorem 6.5.4 recursively to decompose $\mathcal{P}_{q,\beta}$, since, in general, the corresponding orthogonal polynomials $\big(Q_m(y)\big)_{m\geq 0}$ may not satisfy the conditions of the theorem anymore. However, we can identify a class of finite GNN models, for which Theorem 6.5.4 can be applied recursively all the way to $n = 2$, thus yielding a fast algorithm. We specify these models later in Section 6.5.4.

Example 6.5.5 *Consider the normalized Hermite polynomials $\widehat{H}_k(x)$ discussed in Example 4.3.1. Let the model* (6.21) *be $\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/\widehat{H}_n(x)$ with basis $b = \big(\widehat{H}_0, \ldots, \widehat{H}_{n-1}\big)$, and assume $n = 2m$.*

Since the normalized Hermite polynomials satisfy the recursion

$$x\widehat{H}_k(x) = \sqrt{\frac{k}{2}}\widehat{H}_{k-1}(x) + \sqrt{\frac{k+1}{2}}\widehat{H}_{k+1}(x),$$

and $n = 2m$ is even, we can apply either Theorem 6.5.2 or Theorem 6.5.3.

Let us apply the former, for example, in the case when $n = 6$. We obtain the decomposition

$$\mathcal{P}_{b,\alpha} = \begin{pmatrix} 1 & -3.3243 & 7.1069 & -10.9258 & 12.0053 & -8.0754 \\ 1 & -1.8892 & 1.8165 & -0.4388 & -1.1587 & 1.3714 \\ 1 & -0.6167 & -0.4382 & 0.6596 & 0.1761 & -0.6385 \\ 1 & 0.6167 & -0.4382 & -0.6596 & 0.1761 & 0.6385 \\ 1 & 1.8892 & 1.8165 & 0.4388 & -1.1587 & -1.3714 \\ 1 & 3.3243 & 7.1069 & 10.9258 & 12.0053 & 8.0754 \end{pmatrix}$$

$$= \begin{pmatrix} & 1 & & & & -0.4254 \\ 1 & & & & -0.7486 & \\ 1 & & -2.2932 & & & \\ 1 & & 2.2932 & & & \\ & 1 & & & 0.7486 & \\ & & 1 & & & 0.4254 \end{pmatrix} \cdot \left[I_2 \otimes \begin{pmatrix} 1 & -0.4382 & 0.1761 \\ 1 & 1.8165 & -1.1587 \\ 1 & 7.1069 & 12.0053 \end{pmatrix} \right]$$

$$\times \begin{pmatrix} 1 & & & & & \\ & & 1 & & & \\ & & & & 1 & \\ & 0.7071 & & & & \\ & & 1 & 1.2247 & & \\ & & & 1.4142 & & 1.5811 \end{pmatrix}.$$

6.5.3 Decomposition of Discrete GNN Fourier Transforms of Odd Sizes

We now identify a class of finite discrete GNN models of the form $\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/p(x)$, where $\deg p(x) = 2m + 1$ is odd, that can be decomposed into the induction of the form $\mathcal{A} = \mathcal{B} \oplus t(x)\mathcal{B}$. Here, $\mathcal{B}$ itself is a polynomial algebra that corresponds to a finite discrete GNN model; and $t(x) = x$. The results and decompositions are analogous to those in Section 6.5.2.

Decomposition using transversal $(1, x)$

Theorem 6.5.6 *Let $b_k = 0$ for $0 \leq k < n$ in* (6.20)*. Consider the corresponding signal model* (6.21)*, and assume that $n = 2m + 1$.*

Let $\mathcal{P}_{q,\beta}$ be the polynomial transform corresponding to the signal model with the polynomial algebra $\mathbb{C}[y]/Q_{m+1}(y)$ with basis $q = \big(Q_k(y)\big)_{0 \leq k < m+1}$.

Let $\mathcal{P}_{q',\beta'}$ be the polynomial transform corresponding to the signal model with the polynomial algebra $\mathbb{C}[y]/x^{-1}Q_{m+1}(y)$ with basis $q' = \big(Q_k(y)\big)_{0 \leq k < m}$. Observe that $\beta' = \beta \setminus \{0\}$.

Then the polynomial transform $\mathcal{P}_{b,\alpha}$ can be decomposed as follows:

$$\mathcal{P}_{p,\alpha} = \begin{pmatrix} & J_m & -J_m \\ 1 & & \\ & I_m & I_m \end{pmatrix} \cdot \Big(I_{m+1} \oplus \operatorname{diag}\Big(\alpha_k\Big)_{m+1 \leq k < n}\Big) \cdot \Big(\mathcal{P}_{q,\beta} \oplus \mathcal{P}_{q',\beta'}\Big) \cdot B. \tag{6.28}$$

Here,

$$B = \Big(\prod_{i=0}^{m-1} B_i \Big) \cdot L_2^n,$$

Here, B_i is an identity matrix except its $(m+1+i, m+1+i)$-th and $(m+i, m+1+i)$-th elements are equal to $1/a_{2k}$ and $-a_{2k-1}/a_{2k}$, respectively.

Proof: The proof is identical to the proof of Theorem 6.5.3. The only difference is that now $x\mathcal{B} \cong \mathbb{C}[y]/x^{-1}Q_{m+1}(y)$, since $\alpha_m = 0$ is a zero of $P_n(x)$, and $t(0) = 0$. Hence, by Theorem 6.2.5, the dimension of $x\mathcal{B}$ is m. □

Generalization to other finite discrete GNN models

Similarly to the generalization in Section 6.5.2, we can factorize a larger class of finite discrete GNN models.

Theorem 6.5.7 *Let $b_k = d$ for $0 \leq k < n$ in* (6.20)*, where $d \in \mathbb{R}$ is an arbitrary constant. Consider the corresponding signal model* (6.21)*, and assume that $n = 2m + 1$.*

In addition, consider another family of orthogonal polynomials $\big(\tilde{P}_k(x)\big)_{k \geq 0}$ that satisfy (6.20) *with $b_k = 0$ for all $k \geq 0$, and the same a_k as for $P_k(x)$ above. Denote the corresponding signal model* (6.21)

as $\tilde{\mathcal{A}} = \tilde{\mathcal{M}} = \mathbb{C}[x]/\tilde{P}_n(x)$, *and fix the basis* $\tilde{b} = \big(\tilde{P}_k(x)\big)_0 \leq k < n$. *Denote the zeros of* $\tilde{P}_n(x)$ *as* $\tilde{\alpha} = \big(\tilde{\alpha_0}, \ldots, \tilde{\alpha_{n-1}}\big)$.

Then the polynomial transform $\mathcal{P}_{b,\alpha}$ *can be computed as follows:*

$$\mathcal{P}_{b,\alpha} = \operatorname{diag}\left(\sqrt{\frac{P_{n-1}(\alpha_k)P'_n(\alpha_k)}{\tilde{P}_{n-1}(\alpha_k - d)\tilde{P}'_n(\alpha_k - d)}}\right) \cdot \mathcal{P}_{\tilde{b},\tilde{\alpha}}. \tag{6.29}$$

Proof: The proof is identical to the proof of Theorem 6.5.7. □

Operational cost analysis. Similarly to the discussion in Section 6.5.2, the computational cost of $\mathcal{P}_{b,\alpha}$ consists of the costs of matrices M and B, which add up to $3n$ operations; the cost of diagonal matrix D, which is n operations; the cost of $\mathcal{P}_{q,\beta}$, which is $C((n+1)/2)$; and the cost of $\mathcal{P}_{q',\beta'}$.

Now, observe that the signal model $\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/x^{-1}Q_{m+1}(x)$ is not a finite GNN model of the form (6.21). In this case, it may not be clear how to compute $\mathcal{P}_{q',\beta'}$. However, observe that

$$\mathcal{P}_{q,\beta} = \begin{pmatrix} Q_0(0) & Q_1(0) & Q_2(0) & \ldots & Q_m(0) \\ & & & & Q_m(\beta_1) \\ & \mathcal{P}_{q',\beta'} & & & \vdots \\ & & & & Q_m(\beta_m) \end{pmatrix}.$$

Hence, we can compute a matrix-vector product $\mathcal{P}_{q',\beta'} \cdot \mathbf{s}$ by appending a 0 to the input vector $\mathbf{s}$, multiplying it with $\mathcal{P}_{q,\beta}$, and then dropping the first output:

$$\begin{pmatrix} \mathcal{P}_{q',\beta'} \cdot \mathbf{s} \\ \cdot \end{pmatrix} = \mathcal{P}_{q,\beta} \cdot \begin{pmatrix} \mathbf{s} \\ 0 \end{pmatrix}.$$

Since we can use $\mathcal{P}_{q,\beta}$ instead of $\mathcal{P}_{q',\beta'}$, the total operational cost of computing $\mathcal{P}_{b,\alpha}$ is $4n + 2C((n+1)/2)$ operations.

Similarly to the discussion in Section 6.5.2, we should note that, in general, we cannot apply Theorem 6.5.7 recursively to decompose $\mathcal{P}_{q,\beta}$, since, in general, the corresponding orthogonal polynomials $\big(Q_m(y)\big)_{m\geq 0}$ may not satisfy the conditions of the theorem anymore. However, we can identify a class of

finite GNN models, for which Theorem 6.5.7 can be applied recursively all the way to $n = 2$, thus yielding a fast algorithm.

Example 6.5.8 *Consider the normalized Hermite polynomials $\widehat{H}_k(x)$ discussed in Example 4.3.1. Let the model* (6.21) *be $\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/\widehat{H}_n(x)$ with basis $b = \big(\widehat{H}_0, \ldots, \widehat{H}_{n-1}\big)$, and assume $n = 2m + 1$.*

Since the normalized Hermite polynomials satisfy the recursion

$$x\widehat{H}_k(x) = \sqrt{\frac{k}{2}}\widehat{H}_{k-1}(x) + \sqrt{\frac{k+1}{2}}\widehat{H}_{k+1}(x),$$

and $n = 2m + 1$ is odd, we can apply Theorem 6.5.6. In the case of, for example, $n = 5$, we obtain the decomposition

$$\mathcal{P}_{b,\alpha} = \begin{pmatrix} 1 & -2.8570 & 5.0645 & -6.0210 & 4.2150 \\ 1 & -1.3556 & 0.5924 & 0.6432 & -0.9490 \\ 1 & 0 & -0.7071 & 0 & 0.6124 \\ 1 & 1.3556 & 0.5924 & -0.6432 & -0.9490 \\ 1 & 2.8570 & 5.0645 & 6.0210 & 4.2150 \end{pmatrix}$$

$$= \begin{pmatrix} & 1 & & & -2.0202 \\ 1 & & -0.9586 & & \\ 1 & & & & \\ & 1 & & 0.9586 & \\ & & 1 & & 2.0202 \end{pmatrix} \cdot \left[\begin{pmatrix} 1 & -0.7071 & 0.6124 \\ 1 & 0.5924 & -0.9490 \\ 1 & 5.0645 & 4.2150 \end{pmatrix} \oplus \begin{pmatrix} 1 & 0.5924 \\ 1 & 5.0645 \end{pmatrix} \right]$$

$$\times \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1.4142 & \\ & & & & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & -0.8165 \\ & & & & 0.8165 \end{pmatrix} \cdot$$

6.5.4 Fast Algorithm for Discrete GNN Fourier Transforms

As we demonstrated in Sections 6.5.2 and 6.5.3, if $b_k = d$ for $0 \leq k < n$ in (6.20), then we can factorize polynomial transform $\mathcal{P}_{b,\alpha}$ for the finite discrete GNN model (6.21) using Theorems 6.5.4 and 6.5.7. The factorization allows us to compute $\mathcal{P}_{b,\alpha}$ using at most

$$4n + 2C(\lfloor \frac{n+1}{2} \rfloor)$$

operations, where $C(\lfloor \frac{n+1}{2} \rfloor)$ is the operational cost of computing $\mathcal{P}_{q,\beta}$ of size $n/2$ or $(n+1)/2$ (hence, $\lfloor \frac{n+1}{2} \rfloor$).

Recursive decomposition

If we can recursively apply the theorems from Sections 6.5.2 and 6.5.3 to the resulting two polynomial transforms $\mathcal{P}_{q,\beta}$ of size $\lfloor \frac{n+1}{2} \rfloor$, we can further reduce the computational cost of $\mathcal{P}_{b,\alpha}$ to, at most,

$$8n + 4C(\lfloor \frac{n+1}{4} \rfloor)$$

operations. Now, $C(\lfloor \frac{n+1}{4} \rfloor)$ indicates the cost of discrete GNN transforms of size $\lfloor \frac{n+1}{4} \rfloor$.

The question is what the necessary conditions are that allow us to decompose $\mathcal{P}_{q,\beta}$ recursively into two smaller discrete GNN transforms. The following lemmas provides such conditions.

Lemma 6.5.9 *Let $n = 2m$. Consider the model* (6.21)*, and assume that Theorem 6.5.4 applies to the corresponding $\mathcal{P}_{b,\alpha}$, such that*

$$\mathcal{P}_{b,\alpha} = M \cdot \Big(I_2 \otimes \mathcal{P}_{q,\beta} \Big) \cdot B.$$

Then we can apply Theorem 6.5.4 to $\mathcal{P}_{q,\beta}$, if the coefficients a_k in (6.20) *satisfy the conditions*

$$a_0^2 = a_{2k-1}^2 + a_{2k}^2 \tag{6.30}$$

for $1 \leq k \leq \lfloor (n-1)/2 \rfloor$.

Proof: Consider the polynomials $\tilde{P}_k$ constructed in Theorem 6.5.4, and the corresponding model $\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/\tilde{P}_n(x)$. The subalgebra $\mathcal{B} \cong \mathbb{C}[x]/Q_m(y)$ has the basis $q = \big(Q_0(y), \ldots, Q_{m-1}(y)\big)$ of orthogonal polynomials that satisfy

$$yQ_k(y) = a_{2k-2}a_{2k-1}Q_{k-1}(y) + (a_{2k-1}^2 + a_{2k}^2)Q_k(y) + a_{2k}a_{2k+1}Q_{k+1}(y),$$

with $Q_0(y) = 1$ and $Q_1(y) = (y - a_0^2)/a_0a_1$ (we set $d = 0$ in Lemma 6.5.1).

If we now re-define a_k and b_k in (6.20) as

$$\begin{aligned} a_k &:= a_{2k}a_{2k+1}, \\ b_0 &:= a_0^2, \\ b_k &:= a_{2k-1}^2 + a_{2k}^2. \end{aligned}$$

and assume that the condition (6.30) holds, we immediately observe that Theorem 6.5.4 applies to the model $\mathcal{B} \cong \mathbb{C}[x]/Q_m(y)$ with basis $q = \big(Q_0(y), \ldots, Q_{m-1}(y)\big)$. □

An identical lemma applies in the case $n = 2m + 1$. We state it here without a proof.

Lemma 6.5.10 *Let $n = 2m + 1$. Consider the model* (6.21)*, and assume that Theorem 6.5.7 applies to the corresponding $\mathcal{P}_{b,\alpha}$, such that*

$$\mathcal{P}_{b,\alpha} = M \cdot \Big(\mathcal{P}_{q,\beta} \oplus \mathcal{P}_{q',\beta'}\Big) \cdot B.$$

Then we can apply Theorem 6.5.7 to $\mathcal{P}_{q,\beta}$, if the coefficients a_k in (6.20) *satisfy the conditions*

$$a_0^2 = a_{2k-1}^2 + a_{2k}^2 \tag{6.31}$$

for $1 \leq k \leq \lfloor (n-1)/2 \rfloor$.

Fast algorithms

Suppose we can recursively construct such factorizations for every polynomial transform of ever decreasing sizes, until we reach polynomial transforms of size 1. Let us indicate the operational cost of a discrete

GNN transform obtained at the i-th decomposition step with $C_i(n)$, where n is the size of this transform. Then, by constructing the factorization of $\mathcal{P}_{p,\alpha}$ until we reach polynomial transforms of size 1, we obtain a computational algorithm that requires approximately

$$\begin{aligned} C_0(n) &= 4n + 2C_1(\lfloor \frac{n+1}{2} \rfloor) \\ &= 8n + 4C_2(\lfloor \frac{n+1}{4} \rfloor) \\ &\vdots \\ &\approx 4n \log_2 n \end{aligned}$$

operations.

It is now a straightforward task to identify the finite GNN models, for which such factorizations can be performed. Here, we provide the simplest scenario when n is a power of 2. This example gives an idea of how these conditions are derived. Similar conditions can be derived for other sizes n, since at each decomposition step we can apply Theorem 6.5.4 or 6.5.7, depending on the parity of the size of polynomial transforms in the factorization.

The following theorem identifies a class of finite discrete GNN models, for which we can compute $\mathcal{P}_{b,\alpha}$ in $4n \log_2 n$ operations.

Theorem 6.5.11 *Consider the finite discrete GNN model* (6.21)*. Let n be a power of 2, i.e. $\log_2 n \in \mathbb{N}_0$. Assume that*

$$P_n(x) = (x^2 - d_0) \circ \cdots \circ (x^2 - d_{\log_2 n - 1}), \tag{6.32}$$

where $\circ$ denotes the composition of polynomials:

$$f(x) \circ g(x) = f(g(x)).$$

Then the corresponding discrete GNN transform $\mathcal{P}_{b,\alpha}$ can be computed using $4n \log_2 n$ operations.

Proof: The proof follows immediately from the recursive application of Theorem 6.5.4 that is justified by Lemma 6.5.9. □

Observe that the GNN models described in Theorem 6.5.11 include (but are not limited to) the models for the discrete cosine and sine transforms. Moreover, the algorithms can be made numerically stable by orthogonalizing the corresponding polynomial transforms, as explained in Section 4.3.

Finally, notice that polynomial transforms associated with finite discrete GNN models described in Theorem 6.5.11, can also be decomposed using other methods [6]. However, this is only possible in the case when n is a power of 2. As mentioned above, our approach allows us to construct fast algorithms for finite GNN transforms of other sizes as well.

Sufficient condition. The necessary condition in Theorem 6.5.11 is also a sufficient condition. Namely, for any polynomial $P_n(x)$ that can satisfies (6.32), there exists a list of orthogonal polynomials $\big(P_k(x)\big)_{0\leq k<n}$ that satisfy the recursion (6.20) with $b_k = 0$. To find the recursion coefficients, we solve the following system of equations for the given values of $d_0, \ldots, d_{\log_2 n-1}$:

$$\bigcup_{\ell=0}^{\log_2 n-1} \Big\{ \prod_{i=0}^{2^\ell-1} a_i c_i = d_\ell, \big\{ \prod_{i=0}^{2^\ell-1} a_{2^{\ell+1}m-2^\ell+i} c_{2^{\ell+1}m-2^\ell+i} + \prod_{i=0}^{2^\ell-1} a_{2^{\ell+1}m+i} c_{2^{\ell+1}m+i} = d_\ell \big\}_{1\leq m<n/2^{\ell+1}} \Big\}.$$

Chapter 7

Applications

In this chapter, we discuss potential applications of the theory of GNN signal models developed in Chapters 4 through 6.

7.1 Fast Signal Transforms

We can apply the module induction approach to construct novel fast algorithms for various discrete signal transform that correspond to different signal models. Notice that the choice of models is not limited to only the time model and various GNN models.

7.1.1 General-Radix Britanak-Rao FFT

We derive a novel fast general-radix algorithms for DFT_n. It requires $O(n \log n)$ operations. To the best of our knowledge, this algorithm has not been reported in the literature.

In [59], Britanak and Rao derived a fast algorithm for DFT_{2m} that can be written as the factorization

$$\mathrm{DFT}_{2m} = X_m^{2m} \Big(I_m \oplus Z_m^{-1} \Big) D_m^{2m} \Big(\mathrm{DCT\text{-}I}_{m+1} \oplus \mathrm{DST\text{-}I}_{m-1} \Big) B_m^{2m}.$$

Matrices D_m^{2m}, B_m^{2m}, and X_m^{2m} are specified in (C.4-C.6) by setting $k = 1$.

In Appendix C.1, we derive the following general-radix version of this algorithm:

Theorem 7.1.1

$$\begin{aligned}\mathrm{DFT}_{2km} &= L^{2km}_k\Big(I_{2m}\otimes \mathrm{DFT}_k\Big)X^{2km}_m L^{2km}_{2m}\Big(I_m\oplus Z_m^{-1}\oplus I_{2(k-1)m}\Big)D^{2km}_m \\ &\quad\times\Big(\mathrm{DCT\text{-}I}_{m+1}\oplus \mathrm{DST\text{-}I}_{m-1}\oplus I_{k-1}\otimes(\mathrm{DCT\text{-}II}_m\oplus \mathrm{DST\text{-}II}_m)\Big)B^{2km}_m.\end{aligned}$$

Here, D^{2km}_m is a diagonal matrix, and B^{2km}_m and X^{2km}_m are 2-sparse matrices (that is, with each row containing only two non-zero entries) specified in (C.4-C.6).

This factorization is obtained by inducing a subalgebra $\mathcal{B} = \langle (x^k + x^{-k})/2\rangle$ of an algebra $\mathcal{A} = \mathbb{C}[x]/(x^{2km}-1)$ with transversal $t_0(x) = 1, t_1(x) = (x^k - x^{-k})/2, t_{2j}(x) = x^j(x^k+1)/2$, and $t_{2j+1}(x) = x^j(x^k-1)/2$ for $1 \le j < k$.

DFT_k requires $O(k\log k)$ operations; $\mathrm{DCT\text{-}I}_{m+1}$, $\mathrm{DST\text{-}I}_{m-1}$, $\mathrm{DCT\text{-}II}_m$, and $\mathrm{DST\text{-}II}_m$ require $O(m\log m)$ operations each [2,5]. D^{2km}_m requires $n = 2km$ operations and B^{2km}_m and X^{2km}_m each require $3n$ operations. Hence, the algorithm for DFT_n in Theorem 7.1.1 requires $O(n\log n)$ operations.

7.1.2 General-Radix Wang Algorithm for DCT-4

We also derive a novel fast general-radix algorithms for $\mathrm{DCT\text{-}IV}_n$. It requires $O(n\log n)$ operations. To the best of our knowledge, this algorithm has not been reported in the literature.

In [66], Wang derived a fast algorithm for $\mathrm{DCT\text{-}IV}_{2m}$ that can be written as the factorization

$$\begin{aligned}\mathrm{DCT\text{-}IV}_{2m} &= K^{2m}_2\cdot\bigoplus_{j=0}^{m-1}\begin{pmatrix}\cos\frac{2m-2j-1}{8m}\pi & (-1)^j\cos\frac{2j+1-2m}{8m}\pi\\ \cos\frac{2j+1-2m}{8m}\pi & (-1)^{j+1}\cos\frac{2m-2j-1}{8m}\pi\end{pmatrix}\\ &\quad\times(\mathrm{DCT\text{-}III}_m\otimes I_2)(K^{2m}_2)^T\cdot\begin{pmatrix}1 & & & \\ & L_2^{2(m-1)}\cdot I_{m-1}\otimes\mathrm{DFT}_2 & & \\ & & & 1\end{pmatrix}.\end{aligned}$$

In Appendix C.2, we derive the following general-radix version of this algorithm:

Theorem 7.1.2

$$\begin{aligned}\text{DCT-IV}_{2km} &= K_k^{2km}(K_2^{2m} \otimes \text{DCT-IV}_k)Y_m^{2km} \cdot (\text{DCT-III}_m \otimes L_2^{2k})(K_{2k}^n)^T \\ &\times I_k \otimes \begin{pmatrix} 1 & & \\ & L_2^{2(m-1)} \cdot I_{m-1} \otimes \text{DFT}_2 & \\ & & 1 \end{pmatrix} (K_{2m}^{2km})^T.\end{aligned}$$

Here, Y_m^{2km} is a 2-sparse matrix specified in (C.10).

This factorization is obtained by inducing a subalgebra $\mathcal{B} = \langle T_{2k}(x) \rangle$ of an algebra $\mathcal{A} = \mathbb{C}[x]/T_{2km}(x)$ with transversal $t_{2j}(x) = V_j(x)$ and $t_{2j+1}(x) = W_j(x)(V_{2k-1}(x) - V_{2k}(x))/2$ for $0 \leq j < k$.

DCT-IV_k requires $O(k \log k)$ operations, and DCT-III_m requires $O(m \log m)$ operations [2, 5]. Y_m^{2km} requires $3n$ operations, where $n = 2km$. Hence, the algorithm for DCT-IV_n in Theorem 7.1.2 requires $O(n \log n)$ operations.

7.2 Compression of ECG Signals

In [102], we use the finite GNN model based on normalized Hermite polynomials to efficiently compress electrocardiographic signals.

7.2.1 Processing of ECG Signals

Many signals encountered in electrophysiology often have (or can be assumed to have) a compact support. These signals usually represent the impulse response of a system (organ) to an electrical stimulation recorded on the body surface. Examples include electrocardiographic (ECG), electroencephalographic, and myoelectric signals.

Visual analysis of long-term repetitive electrophysiological signals, especially in real time, is a tedious task that requires the presence of a human operator. Computer-based systems have been developed to facilitate this process. For efficient storage, automatic analysis and interpretation, electrophysiological signals are usually represented by a set of features, either heuristic, such as duration and amplitude, or formal, such as coefficients of the expansion in an orthogonal basis. In the latter case, a continuous basis can be used,

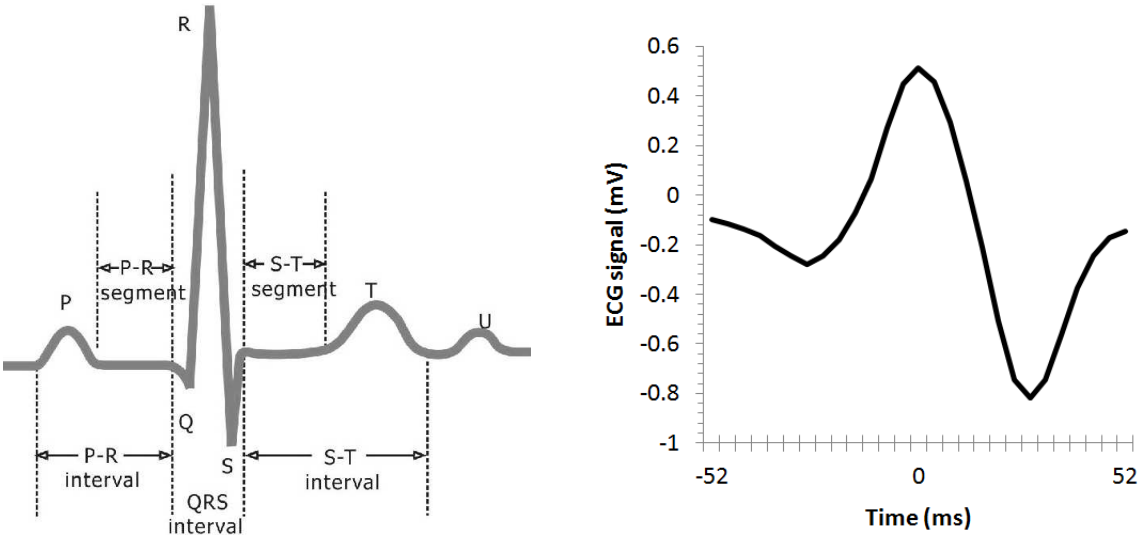


Figure 7.1: (a) ECG structure. Reprinted from LabVIEW for ECG Signal Processing, National Instruments, http://zone.ni.com/devzone/cda/tut/p/id/6349. (b) Example of a QRS complex (centered around the peak).

and the projection and reconstruction of a compact-support signal are computed using numerical methods for integral approximation, such as a numerical quadrature. Alternatively, a discrete basis can be used, and a discrete signal transform, such as DFT or DCT, can be applied to a digitized signal—obtained by sampling a continuous one.

In both continuous and discrete cases, usually only a few projection coefficients are used for the storage and reconstruction of a signal, leading to a reconstruction error. The goal of the compression optimization is to minimize the error while achieving the greatest compression (for example, by using the fewest coefficients possible).

We study the compression of QRS complexes, which are the most characteristic waves of ECG signals [103]. The structure of an ECG signal and an example QRS complex are shown in Fig. 7.1. In particular, we examine the expansion of QRS complexes into the basis of Hermite functions. Such functions, in their continuous form, provide a highly suitable basis for the representation and compression of QRS complexes [103–106]. However, as we discuss in Section 7.2.3, the reported computer implementations of such expansion suffer from certain limitations, such as the inability to obtain an exact reconstruction of a signal, large computational cost, and an a priori selection of coefficients for reconstruction.

We propose an improved compression algorithm for QRS complexes that expands digitized signals into the basis of *discrete* Hermite functions, obtained by sampling the continuous Hermite functions at specific points, not necessarily on a uniform grid. This approach is based on the results obtained Chapters 4 and 6. The proposed algorithm achieves the perfect reconstruction of signals, has a lower computational cost, and allows us to choose coefficients for reconstruction from a larger pool of coefficients. Experiments comparing the approximation accuracy demonstrate that the new algorithm performs on par with other algorithms for low compression ratios (less than 4.5), and outperforms them for higher compression ratios.

7.2.2 Expansion into Hermite Functions

Hermite functions

Consider the family of normalized Hermite polynomials $\widehat{H}_\ell(t)$, $\ell \geq 0$, discussed in Example 4.3.1. Recall that they satisfy recursion (4.14), and hence can be constructed for $\ell \geq 2$ as

$$H_\ell(t) = \sqrt{\frac{2}{\ell}} t H_{\ell-1}(t) - \sqrt{\frac{\ell-1}{\ell}} H_{\ell-2}(t),$$

with $H_0(t) = 1$ and $H_1(t) = \sqrt{2}t$. These polynomials are orthogonal on the real line $\mathbb{R}$ with respect to the weight function e^{-t^2} :

$$\int_{\mathbb{R}} \widehat{H}_\ell(t) \widehat{H}_m(t) e^{-t^2} dt = \sqrt{\pi} \cdot \delta_{\ell-m}. \tag{7.1}$$

It immediately follows that functions

$$\varphi_\ell(t, \sigma) = \pi^{-1/4} e^{-t^2/2\sigma^2} \widehat{H}_\ell(t/\sigma) \tag{7.2}$$

are orthonormal on $\mathbb{R}$ with respect to the inner product

$$\langle \varphi_\ell(t, \sigma), \varphi_m(t, \sigma) \rangle = \int_{\mathbb{R}} \varphi_\ell(t, \sigma) \varphi_m(t, \sigma) dt = \delta_{\ell-m}. \tag{7.3}$$

These functions are called *Hermite functions*. The set of Hermite functions $\{\varphi_\ell(t, \sigma)\}_{\ell \geq 0}$ is an orthonormal basis in the Hilbert space of continuous functions defined on $\mathbb{R}$ [92, 93]. Any such function $s(t)$ can be

represented as

$$s(t) = \sum_{\ell \geq 0} c_\ell \varphi_\ell(t, \sigma), \tag{7.4}$$

where

$$c_\ell = \langle s(t), \varphi_\ell(t, \sigma) \rangle = \int_{\mathbb{R}} s(t) \varphi_\ell(t, \sigma) dt.$$

The first four Hermite functions are shown in Fig. 7.2. Notice that each $\varphi_\ell(t, \sigma)$ quickly approaches zero as the value of $|t|$ increases, since $H_\ell(t/\sigma)$ is a polynomial of degree ℓ, and, as $|t| \to \infty$, $e^{-t^2/2\sigma^2} H_\ell(t/\sigma) \to 0$. Hence, we can assume that each Hermite function has a compact support. In particular, we assume that first L Hermite functions have the same compact support $[-T_\sigma, T_\sigma]$, such that $\varphi_\ell(t, \sigma) = 0$ for $t \notin [-T_\sigma, T_\sigma]$, where $0 \leq \ell < L$, and T_σ is a suitably chosen constant that depends on σ and L. If $s(t)$ also has a compact support of $[-T_\sigma, T_\sigma]$, then we can compute the coefficients c_ℓ with a finite integral:

$$c_\ell = \int_{\mathbb{R}} s(t) \varphi_\ell(t, \sigma) dt = \int_{-T_\sigma}^{T_\sigma} s(t) \varphi_\ell(t, \sigma) dt. \tag{7.5}$$

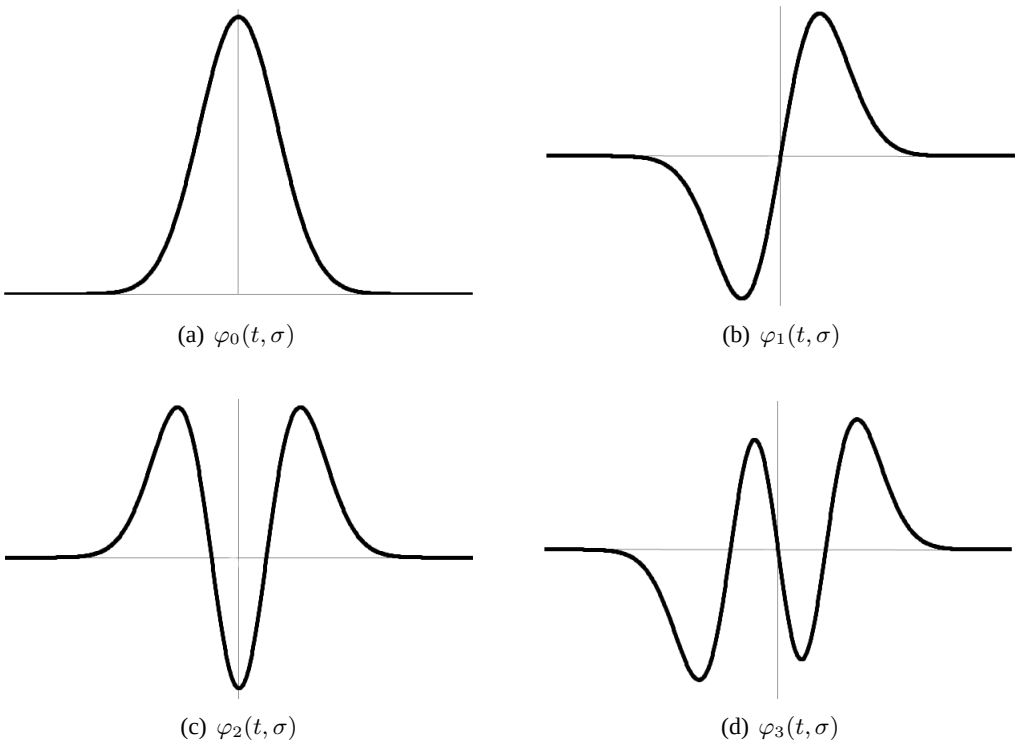
Compression

In practical applications, only a finite number M of Hermite functions are used to represent the signal $s(t)$ in (7.4). Accordingly, only a few *a priori* selected coefficients $c_{\ell_0}, \ldots, c_{\ell_{M-1}}$ are computed. Here, c_{ℓ_k} corresponds to $\varphi_{\ell_k}(t, \sigma)$ in (7.5). Alternatively, a larger pool of coefficients can be computed, from which M ones are selected. It is well-known that for an orthonormal basis selecting coefficients with the largest magnitude minimizes the approximation error computed as the energy of the difference between the signal $s(t)$ and its approximation with M basis functions.

Digital implementation

A computer-based computation of the coefficient (7.5) and the Hermite expansion (7.4) has to be performed in the discrete form. The integral in (7.5) can be computed with a numerical quadrature using, for example, a rectangle rule:

$$c_\ell = \int_{-T_\sigma}^{T_\sigma} s(t) \varphi_\ell(t, \sigma) dt \approx \sum_{k=-K}^{K} s(\tau_k) \varphi_\ell(\tau_k, \sigma)(t_k - t_{k-1}). \tag{7.6}$$



(a) $\varphi_0(t,\sigma)$ (b) $\varphi_1(t,\sigma)$

(c) $\varphi_2(t,\sigma)$ (d) $\varphi_3(t,\sigma)$

Figure 7.2: First four Hermite functions (plotted for the same scale σ).

Here, $-T = t_{-K-1} < t_{-K} < \ldots < t_{K-1} < t_K = T$, and each $t_{k-1} \leq \tau_k \leq t_k$. The signal is then approximated with M Hermite functions as

$$\hat{s}(\tau_k) = \sum_{m=0}^{M-1} c_{\ell_m} \varphi_{\ell_m}(\tau_k, \sigma). \tag{7.7}$$

Let t_k be such that $t_k - t_{k-1} = \Delta$ for all k. Then (7.6) and (7.7) can be expressed in the matrix-vector notation. Let

$$\mathbf{s} = \begin{pmatrix} s(\tau_{-K}) \\ \vdots \\ s(\tau_K) \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} c_0 \\ \vdots \\ c_{M-1} \end{pmatrix}, \quad \hat{\mathbf{s}} = \begin{pmatrix} \hat{s}(\tau_{-K}) \\ \vdots \\ \hat{s}(\tau_K) \end{pmatrix}.$$

Then

$$\mathbf{c} = \Delta \Phi^T \mathbf{s} \quad \text{and} \quad \hat{\mathbf{s}} = \Phi \mathbf{c}, \tag{7.8}$$

where $\Phi \in \mathbb{R}^{(2K+1)\times M}$, such that its m-th column is the ℓ_m-th Hermite function sampled at the points

$\tau_{-K}, \tau_{-K+1}, \ldots, \tau_K$:

$$\Phi_{k,m} = \varphi_{\ell_m}(\tau_k, \sigma)$$

for $-K \leq k \leq K, 0 \leq m < M$.

Observe that for perfect reconstruction $\hat{\mathbf{s}} = \mathbf{s}$, Φ must be an orthogonal matrix: $\Phi\Phi^T = I_{2K+1}$.

Compression of QRS complexes: Previous work

The compression of QRS complexes using the expansion into continuous Hermite functions has been studied in [103–106]. It was originally motivated by the visual similarity of QRS complexes, centered around their peaks, and Hermite functions, as can be observed from Figs. 7.1 and 7.2. Varying the value of σ allows us to "stretch" or "compress" the Hermite functions $\varphi_\ell(t, \sigma)$ to optimally match a given QRS complex.

Since ECG signals are usually available as discrete signals equidistantly sampled at $\tau_k = k\Delta$, previous works used $t_k = \tau_k = k\Delta$ in (7.8). In addition, they proposed to use only the *first* M Hermite functions $\varphi_0(t, \sigma), \ldots, \varphi_{M-1}(t, \sigma)$ for the approximation of QRS complexes.

In Section 7.2.3, we propose an improved compression algorithm that re-samples ECG signals at non-equidistant points, and uses M Hermite functions that have the *largest* coefficients c_ℓ.

Hermite polynomial transforms

Recall from Example 4.3.1 that the polynomial transform (4.22) for normalized Hermite polynomials satisfies the condition

$$\mathcal{P}_{P,\alpha}^{-1} = \mathcal{P}_{P,\alpha}^T D, \tag{7.9}$$

where $D \in \mathbb{R}^{n \times n}$ is a diagonal matrix whose k-th diagonal element is $\sqrt{2/n}/P_{n-1}(\alpha_k)P_n'(\alpha_k)$.

Using the decomposition algorithms for polynomial transforms derived in Section 6.5, a matrix-vector product with $\mathcal{P}_{P,\alpha}$ can be computed with approximately $3n + n^2/4$ operations for small values of n, and $3n + 21.5n \log_2^2(n/2)$ operations for large n, instead of n^2 and $43n \log_2^2 n$, respectively [74]. As a result, the cost is reduced approximately by a factor of 2. Similarly, we can use (7.9) to compute a matrix-vector product with $\mathcal{P}_{P,\alpha}^{-1}$ with only $4n + n^2/4$ and $4n + 21.5n \log_2^2(n/2)$ operations instead of n^2. This reduction of the computational cost is especially significant for large values of n.

7.2.3 Proposed Algorithm

The compression algorithm based on the expansion into continuous Hermite functions has several important limitations. It does not achieve the perfect reconstruction of a signal $\mathbf{s}$, since $\Phi\Phi^T \neq I_{2K+1}$ for $\tau_k = k\Delta$. As a result, $\hat{\mathbf{s}}$ will not converge to $\mathbf{s}$, regardless of the number M of Hermite functions used for the construction of an approximation. This problem could be solved by setting $M = 2K + 1$ and using Φ^{-1} instead of Φ^T to compute $\mathbf{c}$ in (7.8). However, the matrix-vector product $\Phi^{-1}\mathbf{s}$ requires $O((2K + 1)^2)$ operations. This cost can be prohibitive for large K, and makes this approach impractical. Finally, the solution suggested in previous works, that uses the *first* M Hermite functions, may not be the optimal choice for the construction of $\hat{\mathbf{s}}$ with M basis functions.

Algorithm modifications. In Section 7.2.2 the parameter σ was used to "stretch" and "compress" the Hermite functions $\varphi_k(t, \sigma)$ relatively to the signal $s(t)$. Alternatively, we can fix $\sigma = 1$, and introduce a parameter λ to "stretch" and "compress" signal $s(t\lambda)$. In this case the numerical quadrature (7.6) becomes

$$c_\ell = \int_{-T_\lambda}^{T_\lambda} s(t\lambda)\varphi_\ell(t, 1)dt \approx \sum_{k=-K}^{K} s(\tau_k\lambda)\varphi_\ell(\tau_k, 1)(t_k - t_{k-1}).$$

Furthermore, we use different, non-equispaced sampling points. Let $\tau_k = \alpha_{k+K}$, $-K \leq k \leq K$, be the roots of the Hermite polynomial $H_{2K+1}(t)$, and define polynomials $P_\ell(t) = \frac{1}{\sqrt{2^\ell \ell!}} H_\ell(t)$. Then Φ in (7.8) has the form

$$\Phi = \pi^{-1/4} W \mathcal{P}_{P,\alpha}, \tag{7.10}$$

where $W = \operatorname{diag}\left(e^{-\alpha_k^2/2}\right)_{0 \leq k < 2K+1}$ is a diagonal matrix, and $\mathcal{P}_{P,\alpha}$ is given in (2.20).

Finally, if $M = 2K + 1$, then it follows from (7.9) that the columns of Φ form an orthogonal basis:

$$\Phi\Phi^T = \pi^{-1/2} W^2 D^{-1} = \Lambda. \tag{7.11}$$

Thus, to account for the vector norms, we must pre-multiply the input signal $\mathbf{s}$ with the weight matrix Λ^{-1}.

Proposed algorithm. The proposed compression algorithm operates as follows. First, we sample ECG

signal $s(t)$ at sampling points $\alpha_{k+K}\lambda$, $-K \leq k \leq K$, to obtain a vector of samples

$$\mathbf{s} = \Big(s(\alpha_0\lambda), s(\alpha_1\lambda), \ldots, s(\alpha_{2K}\lambda) \Big)^T.$$

Then we compute vector of expansion coefficients

$$\mathbf{c} = \Delta \Phi^T \Lambda^{-1} \mathbf{s},$$

where Φ and Λ are given in (7.10) and (7.11). Finally, we construct vector $\hat{\mathbf{c}}$ by keeping only L coefficients with the largest magnitudes in $\mathbf{c}$ and setting others to zero. Then we use $\hat{\mathbf{c}}$ to obtain signal approximation

$$\hat{\mathbf{s}} = \Delta^{-1} \Phi \hat{\mathbf{c}}.$$

Advantages. The proposed algorithm addresses all limitations of the original compression algorithms based on continuous Hermite functions. The exact reconstruction of signals can be achieved by using all $L = 2K + 1$ coefficients to obtain $\hat{\mathbf{c}}$. Further, to minimize the approximation error, we can compute all coefficients c_ℓ for $0 \leq \ell < 2K + 1$, and only after that pick a few to obtain $\hat{\mathbf{c}}$. This is a practical approach, since the computational cost of both Φ and Φ^T is now smaller, as explained in Section 7.2.2,

7.2.4 Experiments

Setup

In order to analyze the performance of the proposed compression algorithm, we study the compression of QRS complexes extracted from ECG signals obtained from the MIT-BIH ECG Compression Test Database [107]. A total of $N = 29$ QRS complexes are used. Each complex is available as a discrete signal of length $2K + 1 = 27$, and represents a continuous signal of duration 104 milliseconds sampled at 250 Hz.

For the original compression algorithm that uses continuous Hermite functions, we compute $2K + 1$ coefficients $c_0, \ldots, c_{26}$. Among them, we select $1 \leq L \leq 27$ coefficients with the largest magnitude,

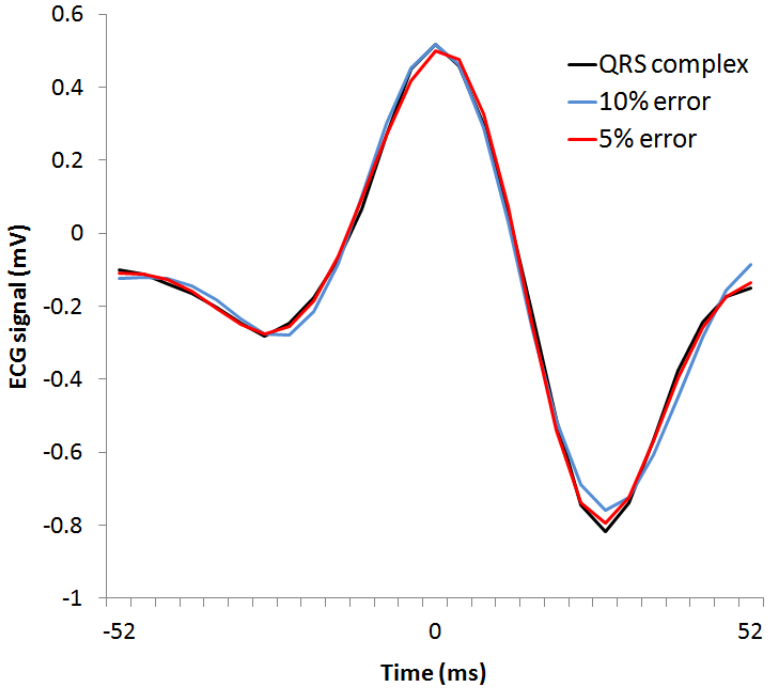


Figure 7.3: A QRS complex and its approximations with 10% and 5% errors.

construct the approximation $\hat{\mathbf{s}}$ using the transpose of Φ, and compute the approximation error

$$E_L = \frac{||\hat{\mathbf{s}} - \mathbf{s}||_2}{||\mathbf{s}||_2}.$$

For the new compression algorithm, we have to re-sample the QRS complexes at points $\tau_k\lambda$ proportional to the roots τ_k of $P_{2K+1}(t)$. To do so, we interpolate the available discrete signals with *sinc* functions, and sample it at points $\tau_k\sigma$. Then we compute $2K+1$ coefficients, select L ones with the largest magnitude, construct the approximation using the inverse transform, and compute the approximation error.

In addition, we study the accuracy of compression algorithms based on two widely-used orthogonal discrete signal transforms—DFT and DCT. As above, we apply the transforms to $\mathbf{s}$, select L largest coefficients, and compute the approximation error of the reconstruction $\hat{\mathbf{s}}$.

The purpose of the experiment is to obtain average approximation errors of 10% and 5% with the fewest coefficients possible. We assume that approximations that capture 90% or 95% of the energy of a QRS complex is sufficient to represent its important features for correct analysis and interpretation. Fig. 7.3 gives

an example of such approximations.

Results

The average approximation errors that were computed during the experiments are plotted in Fig.7.4. Here, Fig. 7.4(a) shows all approximation errors, and Fig. 7.4(b) shows only the ones less than 10%. The x-axis shows the number of coefficients used for reconstruction, and the y-axis shows the errors.

To obtain the average reconstruction error of 10%, our algorithm requires only $L = 5$ coefficients out of $2K + 1 = 27$ (compression ratio 5.4), while the original Hermite algorithm requires 6 coefficients, and DFT and DCT-based algorithms require 7 coefficients (4.5 and 3.86, respectively). To obtain the error of 5%, our algorithm, as well as the ones based on DFT and DCT, requires 8 coefficients (compression ratio 3.5), while the original Hermite algorithm requires 17 coefficients (1.6).

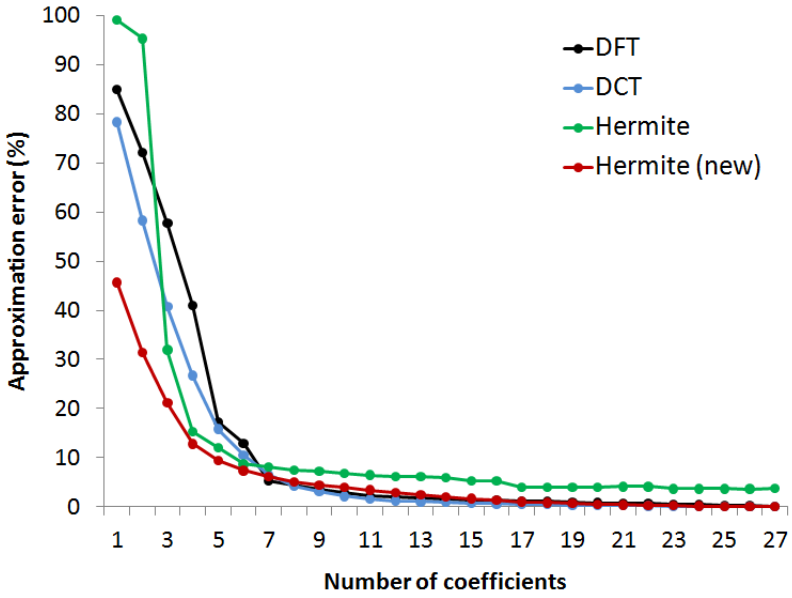
Discussion

As we observe from Fig.7.4, the new compression algorithm has the lowest approximation error among all algorithms if the compression ratio is 4.5 or higher; i.e. if we use up to 6 out of 27 coefficients for reconstruction. For lower compression ratios, it performs on par with the algorithms based on DFT and DCT, and significantly outperforms the original Hermite algorithm.

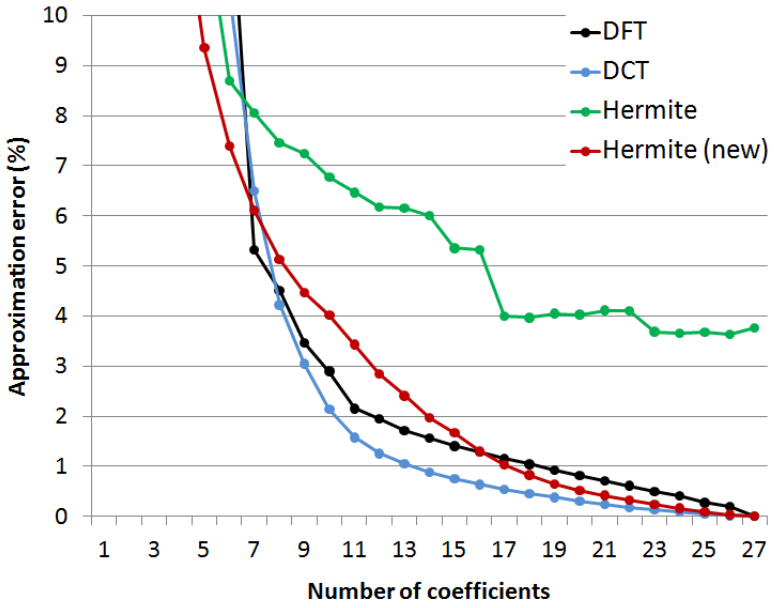
The choice of the values for parameters σ and λ is crucial for optimal representation of signals. We have obtained the best results using $\sigma = \lambda = 0.017$ for all $N = 29$ test signals (these values are for variables t and τ_k measured in seconds). However, in computer-based systems these parameters can be adjusted automatically for each ECG signal to achieve a yet higher accuracy of compression and approximation.

7.3 Gauss-Markov Random Fields

In this section, we discuss the connection of finite discrete GNN models and Gauss-Markov random fields that was originally suggested in [3]. In particular, we demonstrate that generalized discrete Fourier transform for the finite GNN model is precisely the Karhunen-Loève transform matrix for a suitably defined first-order Gauss-Markov random fields. This link provides an educational insight into the similarities of the



(a) All errors.



(b) Errors less than 10%.

Figure 7.4: Average approximation errors for different compression algorithms.

deterministic and stochastic signal processing points of view.

Consider n random variables $\xi_0, \ldots, \xi_{n-1}$ that satisfy the difference equation

$$\xi_k = a_{k-1}\xi_{k-1} + b_k\xi_k + a_k\xi_{k+1} + \nu_k, \tag{7.12}$$

where ν_k is a zero-mean Gaussian noise. $\big(\xi_k\big)_{0\le k<n}$ are called a *first-order Gauss-Markov random field* defined on the finite lattice $0 \le k < n$. We assume zero (Dirichlet) boundary conditions $\xi_{-1} = 0$ and $\xi_n = 0$.

The Karhunen-Loève transform diagonalizes the covariance matrix Σ of the above Gauss-Markov random fields, as well as its inverse. As demonstrated in [108, 109], the inverse of the covariance matrix Σ, called the *potential matrix*, is

$$\Sigma^{-1} = \begin{pmatrix} b_0 & a_0 & & & \\ a_0 & b_1 & a_1 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-3} & b_{n-2} & a_{n-2} \\ & & & a_{n-2} & b_{n-1} \end{pmatrix}. \tag{7.13}$$

Observe that it corresponds to the matrix representation of the basic shift (4.20) of the finite GNN model (4.16). As immediately follows from Section 4.3, the Karhunen-Loève transform of the Gauss-Markov random fields (7.12) is exactly the orthogonalized discrete GNN Fourier transform $D^{-1/2}\mathcal{P}_{b,\alpha}$ defined in (4.19).

7.4 Climate Modeling

In this section, we briefly discuss the need for fast computational algorithms for discrete GNN transforms that arises in the climate modeling. For more detailed description of climate modeling we refer readers to [110, 111].

Many computational climate models include integrals of the following form:

$$\int_0^{\pi} f(\cos\omega) P_\ell(\cos\omega) d\omega,$$

where $0 \leq \ell < n$, and $P_\ell(x)$ are Legendre polynomials discussed in Appendix A. Parameter ω corresponds to the lattitude of a location.

It has been shown that for the functions $f(x)$ of interest to the climate modeling, such integrals can be computed *exactly* using the Gaussian quadrature

$$\int_{-1}^{1} f(x) P_\ell(x) dx = \sum_{k=0}^{n-1} a_k P_\ell(\alpha_k),$$

where α_k are roots of $P_n(x)$ and a_k are properly selected coefficients that depend on $f(x)$.

Hence, we need to calculate the following matrix-vector product:

$$\begin{pmatrix} \int_{-1}^{1} f(x)P_0(x)dx \\ \int_{-1}^{1} f(x)P_1(x)dx \\ \vdots \\ \int_{-1}^{1} f(x)P_{n-1}(x)dx \end{pmatrix} = \begin{pmatrix} P_0(\alpha_0) & P_0(\alpha_1) & \dots & P_0(\alpha_{n-1}) \\ P_1(\alpha_0) & P_1(\alpha_1) & \dots & P_1(\alpha_{n-1}) \\ \vdots & \vdots & & \vdots \\ P_{n-1}(\alpha_0) & P_{n-1}(\alpha_1) & \dots & P_{n-1}(\alpha_{n-1}) \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix}.$$

The matrix above is precisely the transpose of the discrete GNN transform of the finite discrete GNN model $\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/P_n(x)$ with basis $\big(P_0(x), P_1(x), \dots, P_{n-1}(x)\big)$. Using the module induction approach developed in Chapter 6, we can reduce the computational cost of this transform.

7.5 Other Application

In addition, the infinite and finite GNN signal models can be applied in other areas. We discuss two examples below.

Signal compression

Efficient signal representation and compression may require a suitable signal expansion basis. In addition to the examples and applications discussed throughout this thesis, other applications include the use of Hermite polynomials for image coding and processing [112–114]; Laguerre polynomials for signal compression of exponentially decaying signals [115] and speech coding [116].

Birth-death processes

Consider a discrete-state system or process with states $t_0, t_1, t_2, \ldots$. Suppose that a state can change from t_k only to t_{k-1}, t_k, or t_{k+1}. Let the change from state t_k to t_{k-1} occur at rate a_{k-1}; the change from state t_k to t_{k+1} occur at rate c_k; and staying in state t_k occur at rate b_k.

For example, in population modeling, states t_k are associated with the population quantity, and can be set to $t_k = k$. Then, if the current population count is t_k, a death occurs at rate a_{k-1}, and a birth occurs at rate c_k. For this reasons, such processes are often called *birth-death processes*.

Another example can be drawn from the queuing theory. Consider a server with a buffer that accepts and processes data packets every discrete moment of time t_k. Then at each moment t_k we assume that the rate at which packets arrive is c_k, and the rate at which packets are processed is a_{k-1}.

We can associate a family of orthogonal polynomials $\big(P_0(x), P_1(x), \ldots\big)$ with such processes. Namely, we require that they satisfy the recursion

$$x \cdot P_k(x) = a_{k-1}P_{k-1}(x) + b_k P_k(x) + c_k P_{k+1}(x).$$

Recall that this is precisely the recursion (A.1) for orthogonal polynomials.

Such polynomial are sometimes called *birth-death polynomials*. They have been used as convenient tools for modeling birth-death processes [117–119]. Since GNN signal models are based on the same recursion (A.1), they can also be interpreted as models that describe birth-death processes.

Chapter 8

Conclusion

8.1 Summary

In this thesis, we have answered important questions that contribute to the fundamentals of the signal processing theory:

1) Do there exist linear, shift-invariant signal models of interest, not studied previously by the traditional signal processing theory, or by the algebraic signal processing theory? How do we define signal processing concepts for the new models?

2) How do we define and construct the appropriate tools for subband analysis of signals from the new models, such as filter banks and analogs of the discrete Fourier transform?

We constructed a new, large family of signal models called generic nearest-neighbor models, and defined and derived all relevant signal processing concepts for these models. Furthermore, we developed the theory of subband analysis tools for these models that describes the design and implementation of filter banks for infinite discrete generic nearest-neighbor signals, and efficient implementation of associated discrete Fourier transforms for finite discrete generic nearest-neighbor signals.

8.2 Main Contributions

The main contributions of this thesis include:

- **Generic nearest-neighbor signal models.**

  We constructed the infinite and finite discrete generic nearest-neighbor signal models. These models are based on the notion of the generic nearest-neighbor shift that significantly differs from the traditional time shift. We identified the relevant signal processing concepts for these models, including the filter and signal spaces, z-transform, the spectrum, the corresponding Fourier transform, the frequency response, the convolution, and the frequency domain.

  We also generalized the notions of low and high frequencies, as well as the degree of flatness, beyond the domain of traditional time signals to the generic nearest-neighbor signals. The proposed generalization extends to other models as well.

  The proposed infinite and finite discrete generic nearest-neighbor signal models can serve as legitimate alternatives to the infinite and finite discrete time models that are traditionally assumed in modern linear signal processing.

- **Filter banks.**

  We extended the theory of perfect-reconstruction filter banks to the infinite discrete generic nearest-neighbor signal models. Prior to this thesis, design and implementation of filter banks was only studied in the context of traditional time signal model.

  We also introduced a combined approach to the design of filter banks that combines two different approaches that consider filter banks either as arrays of band-pass filters, or as expansions of a signal into properly designed signal bases or frames. The proposed approach can be further extended to future signal models.

  As a demonstration of the developed theory, we constructed two classes of perfect-reconstruction filter banks for infinite discrete generic nearest-neighbor signals. First, we designed two-channel filter banks that extract low-frequency and high-frequency components of an input signal. Second, we designed multichannel filter banks for robust signal transmission that can tolerate partial coefficient loss and allow for exact signal reconstruction from a subset of all coefficients.

  The contribution of this work is two-fold. The more obvious goal is to construct perfect-reconstruction fil-

ter banks for the infinite discrete generic nearest-neighbor signals. However, a more general contribution is the extension of the ASP theory with new, advanced concepts that have not been considered previously. These concepts include the notions of low and high frequency, degree of flatness, and, most importantly, the notion of filter banks for alternative signal models.

- **Fast signal transforms.**

  We introduced a generalized framework for the factorization of arbitrary polynomial transforms into products of sparse matrices. The approach is based on module induction—an algebraic structure that allows us to decompose a signal module into smaller signal modules, hence expressing the original polynomial transform via polynomial transforms of smaller sizes. Such decomposition, applied recursively to all polynomial transforms in the factorization of an original polynomial transform, can lead to a significant reduction of the computational cost of the polynomial transform.

  We applied the developed theory to the factorization of discrete generic nearest-neighbor Fourier transforms. We identified conditions, under which such factorization lead to a reduction in the computational cost of the transforms. As an ultimate result, we identified a class of discrete generic nearest-neighbor Fourier transforms of size $n \times n$ that can be computed only in $O(n \log_2 n)$ operations, rather than in $O(n^2)$.

  We also demonstrated that the developed approach can be used to re-derive existing and discover novel fast algorithms for signal transforms that have already been thoroughly studied, such as discrete Fourier transform and discrete cosine transform.

- **Applications.**

  We studied the applications of generic nearest-neighbor signal models in several areas of signal processing. In particular, the use of finite signal models in the compression of electrocardiographic signals resulted in improvements in the compression accuracy compared to traditional compression methods. Also, we used the developed theory for polynomial transform decomposition to discover fast algorithms for widely-used signal transforms, including DCT and DFT. Finally, we discussed other potential applications, including the fast computation of Karhunen-Loéve transforms for Gauss-Markov random fields, and the use of generic nearest-neighbor models in climate modeling.

8.3 Future Work

Throughout the thesis, we identified several outstanding questions that still remain to be answered. They can serve as indicators of potential directions for future research on generic nearest-neighbor signal models.

Continuous generic nearest-neighbor models. One of the most challenging questions is whether we can construct infinite and finite *continuous* generic nearest-neighbor signal models. These models, on the one hand, must be linear and shift-invariant. On the other hand, they must yield infinite and finite discrete generic nearest-neighbor models after sampling. Unfortunately, the results reported in the literature to date hint that, unlike for time and space models, there may not exist shift-invariant continuous generic nearest-neighbor models [93].

Downsampling. Another intriguing question is connected to the existence of continuous models: What is a proper way to downsampled discrete generic nearest-neighbor signals that have been band-pass filtered? Certainly, a connection between the continuous and discrete models could have lead to the identification of proper sampling techniques and sampling theorems. However, the absence of continuous models requires us to search for other approach. For example, we may try to determine a proper sampling technique without direct construction of continuous GNN models by using interpolation theory. This approach, was used to re-derive the Nyquist sampling theorem with the help of Lagrange interpolation in [96]

Filter bank construction. In this thesis, we introduced an approach for filter bank construction for infinite generic nearest-neighbor signals. However, at its present state, this approach requires laborious calculations to construct even a simple filter bank, such as a Haar-like one. A better approach, such as one that avoids the need to derive equations for each coefficient separately, is very much needed.

Fast algorithms. In this thesis, we identified a class of discrete generic nearest-neighbor Fourier transforms, for which a fast algorithm can be constructed using the module induction technique. The resulting algorithms require $O(n \log_2 n)$ operations. A discovery of fast algorithms for other transforms that are not included in the reported class, will be an important and valuable extension of the developed theory.

Applications. As we discussed in Chapter 7, there exist numerous applications of generic nearest-neighbor signal models in different areas of signal processing. For example, we can further explore the connection

between finite discrete generic nearest-neighbor models and Gauss-Markov random fields. In particular, we can construct fast algorithms for the Karhunen-Loève transform of certain Gauss-Markov random fields. It would be of interest to investigate how to approximate a Gauss-Markov random fields, for which we do not have a fast algorithms, with another Gauss-Markov random fields, for which we have one. Another example is to study classes of signals, which can be efficiently represented and compressed using various generic nearest-neighbor signal models.

Each of the above questions represents a considerable research challenge. Answering it will be a valuable contribution to the fundamental signal processing theory, as well as a deep insight into signal modeling.

Appendix A

Orthogonal Polynomials

A.1 Definition and Properties

There is a large body of literature dedicated to orthogonal polynomials and their properties. A thorough discussion can be found in [90, 91, 94, 120, 121]. Here, we discuss and, if necessary, derive only those properties that are later used in this thesis.

Definition

Polynomials $P = \{P_k(x)\}_{k\geq 0}$ that satisfy the three-term recursion

$$\begin{aligned} x \cdot P_k(x) &= a_{k-1}P_{k-1}(x) + b_k P_k(x) + c_k P_{k+1}(x), \\ & P_0(x) = 1, \;\; P_{-1}(x) = 0, \end{aligned} \tag{A.1}$$

where $a_k, b_, c_k \in \mathbb{R}$ satisfy the condition $a_k c_k > 0$ for $k \geq 0$, are called *orthogonal polynomials*.

Orthogonality

By Favard theorem, there exists an interval $W \subseteq \mathbb{R}$ and a weight function $\mu(x)$, non-negative on W, such that P_n are orthogonal over W with respect to $\mu(x)$:

$$\int_{x \in W} P_k(x) P_m(x) \mu(x) dx = \mu_k \delta_{k-m}. \tag{A.2}$$

Here,

$$||P_k(x)||_{2,\mu} = \Big(\langle P_k(x), P_k(x) \rangle_\mu \Big)^{1/2} = \mu_k^{1/2}$$

is the $\mathcal{L}^2_\mu$-norm of $P_k(x)$ induced by the inner product

$$\langle f(x), g(x) \rangle_\mu = \int_{x \in W} f(x) g(x) \mu(x) dx. \tag{A.3}$$

Note that all μ_k are finite, which holds if $\int_{x \in W} p(x)\mu(x)dx < \infty$ is finite for any polynomial $p(x)$ of an arbitrary degree.

Basis of orthogonal polynomials

The set of orthogonal polynomials $\{P_k(x)\}_{k \geq 0}$ is an orthogonal basis in the Hilbert space of all polynomials defined on the interval W, with the inner product (A.3). Respectively,

$$\{\mu_k^{-1/2} P_k(x)\}_{k \geq 0}$$

is an orthonormal basis in the above space .

Roots of $P_n(x)$

Orthogonal polynomial $P_n(x)$ has exactly n real, distinct roots $\alpha_0 < ... < \alpha_{n-1}$ that lie within the interval of orthogonality W :

$$\alpha_k \in W$$

for all $0 \leq k < n$. Hence, $P_n(x)$ is a separable polynomial of degree n.

Polynomial $P_n(x)$ is a characteristic polynomial (up to a scalar factor) of the following tridiagonal matrix:

$$S = \begin{pmatrix} b_0 & a_0 & & & \\ c_0 & b_1 & a_1 & & \\ & c_1 & b_2 & \ddots & \\ & & \ddots & \ddots & a_{n-2} \\ & & & c_{n-2} & b_{n-1} \end{pmatrix}. \tag{A.4}$$

This means $\alpha_0, \ldots, \alpha_{n-1}$ are exactly the eigenvalues of S. This property allows us to straightforwardly compute the roots of $P_n(x)$.

Even and odd polynomials

In the special case, when all $b_k = 0$ in the recursion (A.1), each polynomial $P_k(x)$ is an even polynomial for even $k = 2m$: $P_{2m}(-x) = P_{2m}(x)$, and odd for odd $k = 2m + 1$: $P_{2m+1}(-x) = -P_{2m+1}(x)$.

Interval of orthogonality

The orthogonality interval $W \subseteq \mathbb{R}$ can be finite $W = [A, B]$, semi-infinite $W = [A, \infty)$, or infinite $W = \mathbb{R}$. Hereafter, we assume that W is one of the following:

- $W = [-1, 1]$;

- $W = [0, \infty)$;

- $W = \mathbb{R}$.

Any family of orthogonal polynomials $\big(P_k(x)\big)_{k\geq 0}$ can be scaled and shifted to be orthogonal on one of the above intervals using a linear transformation

$$x \to ax + b,$$

where a and b are appropriately selected constants.

The orthogonality interval W can be determined by computing lower and upper bounds on the roots of polynomials $P_k(x)$ for increasing k. These bounds can be estimated using the recursion coefficients a_k, b_k, and c_k in recursion (A.1). If both lower and upper bounds are finite, then orthogonality interval W is finite. If only one bound is finite, the interval is semi-infinite. If both bounds are infinite, the interval is the entire real line $\mathbb{R}$.

This is a valid technique, since the roots of orthogonal polynomials satisfy the following "expansion" property: if $\alpha_{n,min}$ and $\alpha_{n,max}$ are, respectively, the smallest and largest roots of $P_n(x)$, then $\alpha_{n+1,min} < \alpha_{n,min}$ and $\alpha_{n+1,max} > \alpha_{n,max}$.

Decay of $\mu(x)$

In general, it is non-trivial to determine the weight function $\mu(x)$ solely from the recursion (A.1). Nevertheless, we can still estimate its behavior in certain cases. Namely, if W is a semi-infinite or infinite interval $[0, \infty)$ or $\mathbb{R}$, respectively, then $\mu(x)$ decreases rapidly for large x. It follows from the requirement that each μ_k must be finite:

$$\int_{x \in W} P_k^2(x)\mu(x)dx = \mu_k < \infty$$

even though

$$\lim_{x \to \infty} |P_k(x)P_m(x)| = \infty.$$

Hence, the decrease rate of weight function $\mu(x)$ is faster than polynomial:

$$\mu(x) = o(x^{-k}) \tag{A.5}$$

for any $k \geq 0$.

A.2 Chebyshev Polynomials

Among all orthogonal polynomials, *Chebyshev* polynomials are arguably the most well-studied ones [16, 17].

	C_1	Closed form for C_n	Weight function $\mu(x)$	Symmetry
T	x	$\cos(n\theta)$	$(1-x^2)^{-1/2}$	$T_{-n} = T_n$
U	$2x$	$\frac{\sin(n+1)\theta}{\sin\theta}$	$(1-x^2)^{1/2}$	$U_{-n} = -U_{n-2}$
V	$2x-1$	$\frac{\cos(n+\frac{1}{2})\theta}{\cos\frac{\theta}{2}}$	$(1+x)^{1/2}(1-x)^{-1/2}$	$V_{-n} = V_{n-1}$
W	$2x+1$	$\frac{\sin(n+\frac{1}{2})\theta}{\sin\frac{\theta}{2}}$	$(1-x)^{1/2}(1+x)^{-1/2}$	$W_{-n} = -W_{n-1}$

Table A.1: Chebyshev polynomials and their properties.

Chebyshev polynomials $C_k(x)$ satisfy the recursion

$$xC_{k+1}(x) = \frac{1}{2}C_{k-1}(x) + \frac{1}{2}C_{k+1}(x)$$

for $k \geq 1$. They are orthogonal on interval $W = [-1, 1]$.

Polynomial $C_0(x) = 1$ is fixed. However, depending on the choice of $C_1(x)$, different orthogonal polynomials may be constructed. They are known as Chebyshev polynomials of first, second, third, and fourth kind; they are denotes as $T_k(x)$, $U_k(x)$, $V_k(x)$, and $W_k(x)$, respectively.

A crucial distinction of Chebyshev polynomials from other orthogonal polynomials is that there exist closed-form expression for their roots. They have been derived using the following property of Chebyshev polinomials: setting $x = \cos\theta$ allows us to express Chebyshev polynomials in their trigonometric closed form $C_n(\cos\theta)$ as functions of cosines and sines of θ.

Main properties of Chebyshev polynomials are listed in Table A.1.

A.3 Other Orthogonal Polynomials

As we discussed above, all orthogonal polynomials have either closed, semi-infinite, or infinite intervals of orthogonality. Furthermore, these intervals can be scaled and shifted to be $[-1, 1]$, $[0, \infty)$, or $\mathbb{R}$.

This property of orthogonal polynomials allows researchers to organize families of orthogonal polynomials into three large classes based on the interval of orthogonality. Each of these classes are named after one of their well-known representatives.

1) *Jacobi-like polynomials* are orthogonal polynomials with orthogonality interval that can be scaled and

shifted to $[-1, 1]$. These polynomials have the general form

$$P_n^{(a,b)}(x) = \frac{\Gamma(a+n+1)}{n!\Gamma(a+b+n+1)} \sum_{k=0}^{n} \binom{n}{k} \frac{\Gamma(a+b+n++k+1)}{\Gamma(a+k+1)} \Big(\frac{x-1}{2}\Big)^k.$$

They are orthogonal over $[-1, 1]$ with respect to weight function

$$\mu^{(a,b)}(x) = (1-x)^a(1+x)^b$$

as follows:

$$\int_{-1}^{1} P_k^{(a,b)}(x) P_m^{(a,b)}(x) \mu^{(a,b)}(x) dx = \frac{2^{a+b+1}}{2k+a+b+1} \frac{\Gamma(k+a+1)\Gamma(k+b+1)}{k!\Gamma(k+a+b+1)} \delta_{k-m}.$$

Besides *Jacobi* polynomials, other well-known polynomials of this class are *Gegenbauer* polynomials, *Legendre* polynomials, and four kinds types of Chebyshev polynomials.

2) *Laguerre-like polynomials* are orthogonal polynomials with orthogonality interval that can be scaled and shifted to $[0, \infty)$. These polynomials have the general form

$$L_n^{(a)}(x) = \sum_{k=0}^{n} (-1)^k \binom{n+a}{n-k} \frac{x^k}{k!}.$$

They are orthogonal over $[0, \infty)$ with respect to weight function

$$\mu^{(a)}(x) = x^a e^{-x}$$

as follows:

$$\int_0^{\infty} L_k^{(a)}(x) L_m^{(a)}(x) \mu^{(a)}(x) dx = \frac{\Gamma(k+a+1)}{k!} \delta_{k-m}.$$

Polynomials $L_n^{(0)}(x)$, for $a = 0$, are known as *Laguerre* polynomials. They are usually denoted simply as $L_n(x)$.

3) *Hermite-like polynomials* are orthogonal polynomials with orthogonality interval that can be scaled and

Polynomials	Recursion	Weight $\mu(x)$	Norm $\|\|P_k(x)\|\|_{2,\mu}$
Legendre	$xP_k = \frac{k}{2k+1}P_{k-1} + \frac{k+1}{2k+1}P_{k+1}$	1	$\sqrt{\frac{2}{2k+1}}$
Laguerre	$xL_k = -kL_{k-1} + (2k+1)L_k - (k+1)L_{k+1}$	e^{-x}	1
Hermite	$xH_k = kH_{k-1} + \frac{1}{2}H_{k+1}$	e^{-x^2}	$\sqrt{k!2^k\sqrt{\pi}}$

Table A.2: Orthogonal polynomials and their properties.

shifted to $\mathbb{R}$. These polynomials have the general form

$$H_n^{(a)}(x) = (2a)^{-n/2} n! \sum_{k=0}^{\lfloor n/2 \rfloor} \frac{(-1)^k}{k!(n-2k)!} \Big(\frac{\sqrt{2}x}{\sqrt{a}}\Big)^{n-2k}.$$

They are orthogonal over $\mathbb{R}$ with respect to weight function

$$\mu^{(a)}(x) = e^{-x^2/2\alpha}$$

as follows:

$$\int_{\mathbb{R}} H_k^{(a)}(x) H_m^{(a)}(x) \mu^{(a)}(x) dx = a^{-(k+m+1)/2} k! \sqrt{2\pi} \delta_{k-m}.$$

Polynomials $H_n^{(1/2)}(x)$, for $a = 1/2$, are known as *Hermite* polynomials. They are usually denoted simply as $H_n(x)$.

In Table A.2, we list the properties of several families of orthogonal polynomials that are used or mentioned in this thesis. Recall that, in general, there are no closed-form expressions for orthogonal polynomials and their roots.

Appendix B

Lapped Tight Frame Transforms

First, observe that we can rewrite (5.6) in the z-domain using polyphase analysis. Namely, we define the $N \times M$ polyphase matrix $\Phi_p(z)$ as[1]

$$\Phi_p(z) = \sum_{r=0}^{q-1} \Phi_r z^{-r}, \tag{B.1}$$

with Φ_r as defined in (3.7). We say $\Phi_p(z)$ has degree $q-1$, since any polynomial in $\Phi_p(z)$ has degree at most $q-1$. Using (B.1), (5.6) is equivalent to $\Phi_p(z)$ being paraunitary:

$$\Phi_p(z)\Phi_p^*(z) = I. \tag{B.2}$$

Here, $\Phi_p^*(z)$ represents the Hermitian transpose of a polyphase matrix of $\Phi(z)$, in which coefficients are complex-conjugated, z^{-1} is replaced by z, and the matrix is transposed. A paraunitary square matrix is unitary on the unit circle.

As mentioned in Section 3.1, oversampled filter banks correspond to frames in $\ell^2(\mathbb{Z})$, whose elements form the columns of Φ in (3.6). The converse is also true. This class of frames is called *filter bank frames*.

We have three equivalent representations of filter bank frames, and, by slight abuse of notation, we will use them interchangeably as convenient and refer to all of them as frames:

- a set of vectors $\{\varphi_i\}_{i\in\mathbb{Z}}$ spanning $\ell^2(\mathbb{Z})$;

[1]The subscript p will always denote a polyphase matrix and should not be confused with subscripts denoting submatrices as in (3.6).

- an infinite matrix Φ as in (3.6);
- a polyphase matrix $\Phi_p(z)$ as in (B.1).

We will also encounter finite frames, that is, spanning sets of $\mathbb{C}^N$ or $\mathbb{R}^N$, and will view them equivalently as $N \times M$ matrices, $M \geq N$. A finite basis hence corresponds to a square matrix.

Hereafter, we often emphasize the special case of a basis by denoting Φ with Ψ. Correspondingly, the base vectors are denoted with φ for frames or ψ for bases.

B.1 Basis Expansions

Basis Expansions with Blocked Transforms. In a critically-sampled filter bank ($M = N$) with filters of length equal to the sampling factor $L = N = M$ ($q = 1$),

$$\Psi = \mathrm{diag}\Big(\ldots, \Psi_0, \Psi_0, \ldots\Big) \tag{B.3}$$

is a block-diagonal matrix with copies of Ψ_0 on the diagonal, as visualized in Fig. 5.5(a). In this case, (5.6) is equivalent to $\Psi_0\Psi_0^* = I_M$, that is, Ψ_0 is an orthonormal basis in $\mathbb{C}^M$. The filter bank processes an infinite signal $x \in \ell^2(\mathbb{Z})$ by applying Ψ_0 to successive nonoverlapping blocks of M signal elements. Since signal blocks are processed as independent signals, and the results are then concatenated, *blocking effects* occur due to boundary discontinuities. A well-known example of a blocked transform uses $\Psi_0^* = \mathrm{DFT}_M$; others include the use of discrete cosine and sine transforms or the discrete Hartley transform.

In the case of the DFT,

$$\Psi_0^* = \mathrm{DFT}_M = \frac{1}{\sqrt{M}}[\omega_M^{mk}]_{0 \leq m,k < M}, \quad \omega_M = e^{-2\pi j/M}.$$

Basis Expansions with Lapped Transforms. To avoid blocking artifacts, basis vectors with longer support can be used, as is the case with LOTs. They can be viewed as a class of M-channel critically-sampled filter banks, originally developed for filters of length $L = 2N = 2M$ and later generalized to arbitrary integer multiples of N [42].

In this work, we focus on LOTs Ψ^* with basis vector support $L = 2N = 2M$ ($q = 2$) whose bases Ψ are visualized in Fig. 5.5(b). The only nonzero blocks in (3.6) are Ψ_0 and Ψ_1; hence, (B.1) yields a polyphase matrix of degree $q - 1 = 1$:

$$\Psi_p(z) = \Psi_0 + z^{-1}\Psi_1. \tag{B.4}$$

Since $\Psi_p(z)$ is square, (B.2) is equivalent to

$$\Psi_0\Psi_0^* + \Psi_1\Psi_1^* = I, \tag{B.5a}$$

$$\Psi_0\Psi_1^* = \Psi_1\Psi_0^* = 0. \tag{B.5b}$$

We use these conditions later to show that the new transforms we construct are indeed LOTs.

B.2 Frame Expansions

In the previous section we explained how critically-sampled filter banks compute basis expansions. Similarly, oversampled filter banks compute frame expansions.

For frames, the property (5.6), $\Phi\Phi^* = I$, is called *tightness* [122].[2] Tight frames can be constructed from orthonormal bases using the Naimark theorem [123, 124]:

Theorem B.2.1 *A set $\{\varphi_i\}_{i\in\mathcal{I}}$ is a tight frame for a Hilbert space $\mathbb{H}$ if and only if there exists another Hilbert space $\mathbb{K} \supset \mathbb{H}$ with an orthonormal basis $\{\psi_i\}_{i\in\mathcal{I}}$, so that the orthogonal projection P of $\mathbb{K}$ onto $\mathbb{H}$ satisfies: $P\psi_i = \varphi_i$, for all $i \in \mathcal{I}$.*

One example of an orthogonal projection is the canonical projection that simply omits coordinates and is called *seeding* [48].

In the finite case, seeding yields a frame ($N \times M$ matrix) Φ for $\mathbb{C}^N$ by omitting rows from a basis ($M \times M$ matrix) Ψ of $\mathbb{C}^M$. Conversely, every finite frame can be obtained this way.[3]

To seed in the infinite case considered here, we extend this approach to polyphase matrices $\Psi_p(z)$.

[2] Note that in general, a tight frame is also one for which $\Phi\Phi^* = cI$; however, since c can be pulled into Φ, we consider only $c = 1$ here.

[3] Just extend Φ with rows to an invertible square matrix.

Definition B.2.2 *A frame $\Phi_p(z)$ is obtained by* seeding *from a basis $\Psi_p(z)$, if it is constructed from $\Psi_p(z)$ by preserving only a subset of the rows of $\Psi_p(z)$. This is written as $\Phi_p(z) = \Psi_p(z)[\mathcal{I}]$, where $\mathcal{I}$ is the set of indices of the retained rows.*

In particular, for $q = 1$, seeding constructs frames of the form in Fig. 5.5(c) from bases of the form in Fig. 5.5(a). Conversely, every such frame can be constructed this way.

For $q > 1$, seeding constructs frames of the form in Fig. 5.5(d) from bases of the form in Fig. 5.5(b) (the example in the figure is for $q = 2$). However, in this case, it is unclear whether the converse is true.

The following result is a special case of Theorem B.2.1:

Lemma B.2.3 *Seeding an orthonormal basis (paraunitary) $\Psi_p(z)$ yields a tight frame $\Phi_p(z)$.*

Next, we discuss the blocked and lapped frame expansions in Figs. 5.5(c) and (d) in greater detail.

Frame Expansions with Blocked Transforms. If $q = 1$, then, as visualized in Fig. 5.5(c),

$$\Phi = \mathrm{diag}\Big(\dots, \Phi_0, \Phi_0, \dots\Big). \tag{B.6}$$

The difference from (B.3) is that Φ_0 is now rectangular: $\Phi_0 \in \mathbb{C}^{N\times M}$, and can be viewed as an M-element frame in $\mathbb{C}^N$. Hence, if it is tight, it can be constructed from an orthogonal basis in $\mathbb{C}^M$ by seeding.

Frame Expansions with Lapped Transforms. Projecting signals onto frame vectors with nonoverlapping support leads to similar blocking artifacts as for orthonormal bases. We thus use the same approach as for orthonormal bases in Section B.1 and consider frames in $\ell^2(\mathbb{Z})$ with vector support $L = 2N$, visualized in Fig. 5.5(d).

As in (B.4), the resulting polyphase matrix $\Phi_p(z)$ has degree 1:

$$\Phi_p(z) = \Phi_0 + z^{-1}\Phi_1,$$

and the tightness condition $\Phi\Phi^* = I$ is equivalent to $\Phi_p(z)$ being paraunitary (B.2).

In [43], LTFTs were constructed by seeding the polyphase matrix $\Psi_p(z)$ of an LOT basis:

$$\Phi_p(z) = \Psi_p(z)[\mathcal{I}]. \tag{B.7}$$

By the Naimark theorem, the constructed frames are tight; this is why we named them lapped tight frame transforms. We will follow later the same procedure here to derive LTFTs from LOT bases. First, we introduce the frame properties we consider.

B.3 Construction of New Real LOTs

In Section B.1 we showed that a real LOT basis corresponds to a real square paraunitary polyphase matrix $\Psi_p(z)$ of degree $q-1$. Although in general $\Psi_p(z)$ is paraunitary if and only if it is unitary on the entire unit circle $|z|=1$, for a real $\Psi_p(z)$ of degree $q-1=1$, it suffices to check only two conditions:

Lemma B.3.1 *Let $\Psi_p(z)$ be a real $M \times M$ polyphase matrix of degree 1, that is, $\Psi_p(z) = \Psi_0 + z^{-1}\Psi_1$, where $\Psi_0, \Psi_1 \in \mathbb{R}^{M\times M}$. Then, $\Psi_p(z)$ is paraunitary if and only if $\Psi_p(1)$ and $\Psi_p(j)$ are unitary.*

Proof: "$\Rightarrow$" is immediate. To prove "$\Leftarrow$", let $\Psi_p(1) = \Psi_0 + \Psi_1$ and $\Psi_p(j) = \Psi_0 - j\Psi_1$ be unitary, that is,

$$\begin{aligned} & \begin{cases} (\Psi_0+\Psi_1)(\Psi_0^T+\Psi_1^T) = I_M \\ (\Psi_0-j\Psi_1)(\Psi_0^T+j\Psi_1^T) = I_M \end{cases} \\ \Leftrightarrow \quad & \begin{cases} \Psi_0\Psi_0^T + \Psi_1\Psi_1^T + \Psi_0\Psi_1^T + \Psi_1\Psi_0^T = I_M \\ \Psi_0\Psi_0^T + \Psi_1\Psi_1^T + j(\Psi_0\Psi_1^T - \Psi_1\Psi_0^T) = I_M \end{cases} \end{aligned} \tag{B.8}$$

Subtracting the two equations yields

$$\begin{aligned} & \Psi_0\Psi_1^T + \Psi_1\Psi_0^T - j(\Psi_0\Psi_1^T - \Psi_1\Psi_0^T) = 0_M \\ \Leftrightarrow \quad & \Psi_0\Psi_1^T + \Psi_1\Psi_0^T = 0_M \text{ and } \Psi_0\Psi_1^T - \Psi_1\Psi_0^T = 0_M \\ \Leftrightarrow \quad & \Psi_0\Psi_1^T = 0_M \text{ and } \Psi_1\Psi_0^T = 0_M \end{aligned}$$

Inserting into (B.8) yields $\Psi_0\Psi_0^T + \Psi_1\Psi_1^T = I_M$; all requirements (B.5a)-(B.5b) for a paraunitary $\Psi_p(z)$ are satisfied. □

$$\mathrm{DFT}_{p,K}(z) \xrightarrow{\text{submatrix}} \Psi_p(z) \xrightarrow{\text{seed}} \Phi_p(z)$$

Figure B.1: Construction of frames for $q = 2$.

Lemma B.3.1 chooses 1 and j as evaluation points. Using a very similar proof, we can generalize to arbitrary roots of unity ω_1 and ω_2, provided $\omega_2 \neq \pm\omega_1, \pm\omega_1^*$.

As an example application of Lemma B.3.1, consider the $K \times K$ polyphase matrix

$$\mathrm{DFT}_{p,K}(z) = \frac{1}{\sqrt{K}} \left[\cos \frac{2k\ell\pi}{K} + z^{-1} \sin \frac{2k\ell\pi}{K} \right]_{0 \leq k,\ell \leq K-1}. \tag{B.9}$$

Both $\mathrm{DFT}_{p,K}(j) = \mathrm{DFT}_K$ and $\mathrm{DFT}_{p,K}(1) = \mathrm{DHT}_K$ (the discrete Hartley transform [125]) are unitary; hence, by Lemma B.3.1, $\mathrm{DFT}_{p,K}(z)$ is paraunitary.

In Theorem B.3.2, we show that specific submatrices of $\mathrm{DFT}_{p,K}(z)$ are paraunitary, and thus correspond to LOTs. In Section B.4 we will seed these matrices to obtain LTFTs (this algorithm is depicted in Fig. B.1).

Theorem B.3.2 *Let $\Psi_p(z)$ be an $M \times M$ submatrix of $\sqrt{K/M} \cdot \mathrm{DFT}_{p,K}(z)$, $K \geq M \geq 2$, constructed by selecting the following row and column sets:*

$$\begin{aligned} \textit{rows:} \quad & \{r + kR \bmod K \mid 0 \leq k \leq M-1\} \\ \textit{columns:} \quad & \{c + \ell C \bmod K \mid 0 \leq \ell \leq M-1\} \end{aligned}$$

for some constants $0 \leq r, c, R, C < K$.

Then, $\Psi_p(z)$ is paraunitary if $K = M \gcd(K, RC)$ (in particular, M divides K) and one of the following is satisfied:

(i) K divides $2rC$, $4rc$, and $2MRc$;

(ii) K does not divide $2rC$, and K divides both $2r(2c + CM - C)$ and $R(2c + CM - C)$.

Proof: According to Lemma B.3.1, to show that $\Psi_p(z)$ is paraunitary, it is enough to show that $\Psi_p(j)$ and $\Psi_p(1)$ are unitary.

The elements of the matrix $\Psi_p(z)$ are

$$\begin{aligned}\psi_{k,\ell}(z) &= \frac{1}{\sqrt{M}}\Big(\cos\big(\frac{2\pi(r+kR)(c+\ell C)}{K}\big) \\ &+ \sin\big(\frac{2\pi(r+kR)(c+\ell C)}{K}\big)z^{-1}\Big),\end{aligned}$$

$0 \le k, \ell \le M-1$ and $0 \le r, c, R, C \le M-1$.

We first find the conditions for $\Psi_p(j)$ to be unitary. The (k,ℓ)th element of $\Psi_p(j)\Psi_p^*(j)$ is given by

$$\begin{aligned}\big(\Psi_p(j)\Psi_p^*(j)\big)_{k,\ell} &= \frac{1}{M}\sum_{m=0}^{M-1}\omega_K^{(r+kR)(c+mC)-(r+\ell R)(c+mC)} \\ &= \frac{1}{M}\omega_K^{(k-\ell)Rc}\sum_{m=0}^{M-1}\omega_K^{(k-\ell)RCm} \\ &= \begin{cases}1, & k=\ell; \\ \frac{1}{M}\omega_K^{(k-\ell)Rc}\frac{1-\omega_K^{(k-\ell)RCM}}{1-\omega_K^{(k-\ell)RC}}, & k\neq\ell.\end{cases}\end{aligned}$$

$\Psi_p(j)$ is unitary if and only if $(\Psi_p(j)\Psi_p^*(j))_{k,\ell} = 0$ for any $k \neq \ell$, or, equivalently, if and only if K is divisible by the product RCM, but not divisible by $(k-\ell)RC$ for any $k-\ell \neq 0$ such that $1 \le |k-\ell| \le M-1$. This is possible if and only if $K = M\gcd(K, RC)$. Thus, $\Psi_p(j)\Psi_p^*(j) = I_M$, and $\Psi_p(j)$ is unitary if and only if $K = M\gcd(K, RC)$.

We next investigate conditions for $\Psi_p(1)$ to be unitary. The (m,ℓ)th element of $\Psi_p(1)$ is

$$\begin{aligned}\psi_{k,\ell}(1) &= \frac{1}{\sqrt{M}}\Big(\cos\big(\frac{2\pi(r+kR)(c+\ell C)}{K}\big) \\ &\quad + \sin\big(\frac{2\pi(r+kR)(c+\ell C)}{K}\big)\Big) \\ &= \frac{1}{\sqrt{M}}\Big(\frac{1+j}{2}\omega_K^{(r+kR)(c+\ell C)} \\ &\quad + \frac{1-j}{2}\omega_K^{-(r+kR)(c+\ell C)}\Big).\end{aligned}$$

The (k,ℓ)-th element of $\Psi_p(1)\Psi_p^*(1)$ is

$$\begin{aligned}
(\Psi_p(1)\Psi_p^*(1))_{k,\ell} &= \frac{1}{M}\sum_{m=0}^{M-1}\Big[\big(\tfrac{1+j}{2}\omega_K^{(r+kR)(c+mC)} \\
&\quad +\tfrac{1-j}{2}\omega_K^{-(r+kR)(c+mC)}\big) \\
&\quad \times\big(\tfrac{1-j}{2}\omega_K^{-(r+\ell R)(c+mC)} + \tfrac{1+j}{2}\omega_K^{(r+\ell R)(c+mC)}\big)\Big] \\
&= \tfrac{1}{2M}\sum_{m=0}^{M-1}\big(\omega_K^{(k-\ell)R(c+kC)} + \omega_K^{(\ell-k)R(c+mC)}\big) \\
&\quad +\tfrac{j}{2M}\sum_{m=0}^{M-1}\big(\omega_K^{(2r+(k+\ell)R)(c+mC)} \\
&\quad -\omega_K^{-(2r+(k+\ell)R)(c+mC)}\big) \\
&= \tfrac{1}{2M}\Sigma_{k,\ell}^{(1)} + \tfrac{j}{2M}\Sigma_{k,\ell}^{(2)}.
\end{aligned}$$

Since $K = M\gcd(K,RC)$, then for any $0 \le k,\ell \le M-1$ with $k \ne \ell$, K is not divisible by $(k-\ell)RC$. Thus

$$\begin{aligned}
\Sigma_{k,\ell}^{(1)} &= \begin{cases} \sum_{m=0}^{M-1} 2, & k=\ell; \\ \omega_K^{(k-\ell)Rc}\frac{1-\omega_K^{(k-\ell)RCM}}{1-\omega_K^{(k-\ell)RC}} & \\ +\omega_K^{(\ell-k)Rc}\frac{1-\omega_K^{(\ell-k)RCM}}{1-\omega_K^{(\ell-k)RC}}, & k\ne\ell; \end{cases} \\
&= \begin{cases} 2M, & k=\ell; \\ 0, & k\ne\ell. \end{cases}
\end{aligned}$$

To make $\Psi_p(1)$ a unitary matrix, we choose to impose the condition $\Sigma_{k,\ell}^{(2)} = 0$ for any $0 \le k,\ell \le M-1$. Here, we consider the two cases specified by the theorem:

Case (i). If K divides $2rC$, $4rc$, and $2MRc$, then for any k, ℓ

$$\begin{aligned}\Sigma_{k,\ell}^{(2)} &= \omega_K^{(2r+(k+\ell)R)c} \sum_{m=0}^{M-1} \omega_K^{(2r+(k+\ell)R)Cm} \\ &\quad -\omega_K^{-(2r+(k+\ell)R)c} \sum_{m=0}^{M-1} \omega_K^{-(2r+(k+\ell)R)Cm}\end{aligned}$$

$$\begin{aligned}&= \begin{cases} M(\omega_K^{2rc} - \omega_K^{-2rc}), & k+\ell = 0; \\ M(\omega_K^{2rc+MRc} - \omega_K^{-2rc-MRc}), & k+\ell = M; \\ \omega_K^{2rc+(k+\ell)Rc} \frac{1-\omega_K^{(k+\ell)RCM}}{1-\omega_K^{(k+\ell)RC}} & \\ -\omega_K^{-2rc-(k+\ell)Rc_0} \frac{1-\omega_K^{-(k+\ell)RCM}}{1-\omega_K^{-(k+\ell)RC}}, & \text{otherwise;} \end{cases} \\ &= 0.\end{aligned}$$

Case (ii). If K does not divide $2rC$, then $\Sigma_{k,\ell}^{(2)} = 0$ is equivalent to

$$\omega_K^{(2r+(k+\ell)R)(2c+CM-C)} = 1$$

for any k, ℓ. This is possible if K divides both $2r(2c + CM - C)$ and $R(2c + CM - C)$.

Thus, in either of the two cases $\Psi_p(1)\Psi_p^*(1) = I_M$, and $\Psi_p(1)$ is unitary.

Since the above conditions make $\Psi_p(j)$ and $\Psi_p(1)$ unitary, Lemma B.3.1 implies that $\Psi_p(z)$ is paraunitary. □

Since $\mathrm{DFT}_{p,K}(z)$ is symmetric, we can interchange the row and column index sets in the theorem:

Corollary B.3.3 *$\Psi_p(z)$ constructed as in Theorem B.3.2 is paraunitary if and only if $\Psi_p(z)^T$ is paraunitary.*

Note that in Theorem B.3.2 we work with index sets instead of lists since permutations of rows and columns preserve paraunitarity.

Each paraunitary matrix $\Psi_p(z)$ obtained with Theorem B.3.2 defines a basis Ψ; the associated LOT is Ψ^*. Next, we complete the theory and discuss the seeding of LTFTs from the above LOTs Ψ.

B.4 Construction of New LTFTs from LOTs

In this section we seed $M \times M$ LOT matrices $\Psi_p(z)$, constructed as in Theorem B.3.2, to obtain $N \times M$ frames $\Phi_p(z)$ and establish their properties.

Tightness. Any seeding of a $\Psi_p(z)$ obtained with Theorem B.3.2 yields a tight frame $\Phi_p(z)$ by Lemma B.2.3.

Equal Norm. Every element of $\Psi_p(z)$ constructed with Theorem B.3.2 has the norm $1/\sqrt{M}$. Hence, the columns of any seeded $N \times M$ matrix $\Phi_p(z)$ have the same norm $\sqrt{N/M}$.

Maximally Robust Frames. In general, maximal robustness for frames is a property difficult to prove since one has to check that every $N \times N$ submatrix of $\Phi_p(z)$ is invertible. The good news is that it is sufficient to ensure that each such submatrix is nonsingular for at least one value [126]:

Lemma B.4.1 *A square polyphase matrix $A_p(z)$ is nonsingular if and only if there exists $z_0 \in \mathbb{C}$ such that $\det A_p(z_0) \neq 0$.*

We will use this fact in the proof of the following theorem.

Theorem B.4.2 *Let $\Psi_p(z)$ be a paraunitary polyphase matrix constructed using Theorem B.3.2 such that M and MRC/K are co-prime. Further, we seed a frame,*

$$\Phi_p(z) = \Psi_p(z)[\mathcal{I}],$$

by retaining $N < M$ rows. Then $\Phi_p(z)$ is maximally robust to erasures if (as sets)

$$\mathcal{I} = \{d + Dk \bmod M \mid 0 \leq k < N\}$$

for some $0 \leq d < M$ and $D \equiv (MRC/K)^{-1} \bmod M$.

Proof: We use Lemma B.4.1 with $z_0 = j$, which makes $\Psi_p(j)$ a submatrix of $\sqrt{K/M}\,\mathrm{DFT}_K$. We fix the

order of rows and columns and get

$$\begin{aligned}\Psi_p(j) &= \frac{1}{\sqrt{M}}\left[\omega_K^{(r+kR)(c+\ell C)}\right]_{0\le k,\ell<M} \\ &= \frac{1}{\sqrt{M}}\left[\omega_K^{rc}\cdot\omega_K^{ckR}\cdot\omega_K^{r\ell C}\cdot\omega_K^{k\ell RC}\right]_{0\le k,\ell<M} \\ &= \frac{1}{\sqrt{M}}\omega_K^{rc}\cdot\Omega_1\cdot\left[\omega_K^{k\ell RC}\right]_{0\le k,\ell<M}\cdot\Omega_2.\end{aligned}$$

Here, $\Omega_1 = \mathrm{diag}\left(\omega_K^{ckR}\right)_{0\le k<M}$ and $\Omega_2 = \mathrm{diag}\left(\omega_K^{r\ell C}\right)_{0\le \ell<M}$ are full-rank diagonal matrices, and $\omega_K^{rc} \neq 0$. Hence, we can omit them in studying the seeding of MR frames.

Setting $MRC/K = A$ yields

$$\omega_K^{k\ell RC} = \omega_K^{k\ell\frac{AK}{M}} = (\omega_M^A)^{k\ell}. \tag{B.10}$$

Since $\gcd(M, A) = 1$, ω_M^A is a primitive Mth root of unity, and thus

$$\frac{1}{\sqrt{M}}\left[\omega_K^{k\ell RC}\right]_{0\le k,\ell\le M-1} = P\cdot\mathrm{DFT}_M\cdot P^T, \tag{B.11}$$

where P is the $M\times M$ permutation matrix:

$$P_{k\ell} = \begin{cases} 1, & \text{if } \ell = Ak \mod M \\ 0, & \text{otherwise}\end{cases}. \tag{B.12}$$

Further, let $D \equiv (MRC/K)^{-1} \bmod M$, $1 \le D < M$, and consider an $N\times M$ submatrix of (B.11), constructed by selecting rows $\mathcal{I} = \{d + Dk \bmod M \mid 0 \le k < N\}$. Then

$$\left(P\cdot\mathrm{DFT}_M\cdot P^T\right)[\mathcal{I}] = \mathrm{DFT}_M[\mathcal{J}]\cdot \mathrm{P}^T, \tag{B.13}$$

where $\mathcal{J} = \{dA + k \bmod M \mid 0 \le k < N\}$. Since $\mathrm{DFT}_M[\mathcal{J}]$ is an $N\times M$ submatrix of DFT_M constructed from adjacent rows (possibly looping around the bottom of the matrix), each $N\times N$ submatrix of it is invertible [48]. It follows that each $N\times N$ submatrix of $\Phi_p(j)$ is also invertible.

Hence, by Lemma B.4.1, every $N\times N$ submatrix of $\Phi_p(z)$ is nonsingular, and $\Phi_p(z)$ is maximally

robust to erasures. □

As an example, consider the following family of maximally robust LTFTs:

Corollary B.4.3 *If $\Psi_p(z)$ is constructed as in Theorem B.3.2 with $R = 1, C = K/M$, and $r = c = 0$, then any consecutive seeding (retaining of consecutive rows) of $\Psi_p(z)$ yields a maximally robust LTFT $\Phi_p(z)$.*

Note that the LTFTs constructed as in Corollary B.4.3 and seeded starting with the first row (i.e. $\mathcal{I} = \{0, 1, \ldots, M-1\}$) are Weyl-Heisenberg frames [127].

Appendix C

Proofs of Theorems in Chapter 7

C.1 Proof of Theorem 7.1.1

Consider $\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/(x^{2km} - 1)$, with basis $\big(1, x, \ldots, x^{2km-1}\big)$ and $\alpha_k = \omega_{2km}^k$. The corresponding polynomial transform is DFT_{2km}.

By Theorem 6.1.2, the polynomial $r(x) = (x^k + x^{-k})/2$ generates the subalgebra

$$\mathcal{B} = \big\langle r(x) \big\rangle \cong \mathbb{C}[y]/2(y^2 - 1)U_{m-1}(y).$$

If we choose $\big(T_\ell(y)\big)_{0 \le \ell < m+1}$ as the basis , the polynomial transform is

$$\Big[T_\ell(\cos \frac{k\pi}{m})\Big]_{0 \le k, \ell < m+1} = \text{DCT-I}_{m+1}\,.$$

By Theorem 6.2.5, the $\mathcal{B}$-module $(x^k - x^{-k})/2 \cdot \mathcal{B} \cong \mathbb{C}[y]/U_{m-1}(y)$. If we choose the basis $\big(U_\ell(y)\big)_{0 \le \ell < m-1}$, then the polynomial transform is

$$\Big[U_\ell(\cos \frac{k\pi}{m})\Big]_{0 \le k, \ell < m-1} = \mathrm{diag}\left(1/\sin \frac{(k+1)\pi}{m}\right)_{0 \le k < m-2} \cdot \text{DST-I}_{m-1} = \overline{\text{DST-I}}_{m-1}. \tag{C.1}$$

Similarly, the $\mathcal{B}$-module $x^j(x^k + 1)/2 \cdot \mathcal{B} \cong \mathbb{C}[y]/2(y - 1)U_{m-1}(y)$ for any $1 \le j < k$. If we choose

the basis $\big(V_\ell(y)\big)_{0\le\ell<m}$, then the polynomial transform is

$$\Big[V_\ell(\cos\frac{k\pi}{m})\Big]_{0\le k,\ell<m} = \operatorname{diag}\Big(1/\cos\frac{k\pi}{2m}\Big)_{0\le k<m}\cdot \text{DCT-II}_m = \overline{\text{DCT-II}}_m. \tag{C.2}$$

Finally, the $\mathcal{B}$-module $x^j(x^k-1)/2\cdot\mathcal{B}\cong\mathbb{C}[y]/2(y+1)U_{m-1}(y)$ for any $1\le j<k$. If we choose the basis $\big(W_\ell(y)\big)_{0\le\ell<m}$, then the polynomial transform is

$$\Big[W_\ell(\cos\frac{(k+1)\pi}{m})\Big]_{0\le k,\ell<m} = \operatorname{diag}\Big(1/\sin\frac{(k+1)\pi}{2m}\Big)_{0\le k<m}\cdot \text{DST-II}_m = \overline{\text{DST-II}}_m. \tag{C.3}$$

Using Theorem 6.2.6, we can verify that $t_0(x)=1$, $t_1(x)=(x^k-x^{-k})/2$, $t_{2j}(x)=x^j(x^k+1)/2$, and $t_{2j+1}(x)=x^j(x^k-1)/2$ for $1\le j<k$, is a transversal of $\mathcal{B}$ in $\mathcal{A}$. Hence, by Theorem 6.3.1, we obtain the factorization

$$\text{DFT}_n \;=\; M\Big(\text{DCT-I}_{m+1}\oplus\overline{\text{DST-I}}_{m-1}\oplus I_{k-1}\otimes(\overline{\text{DCT-II}}_m\oplus\overline{\text{DST-II}}_m)\Big)B_m^{2km}.$$

Here, B_m^{2km} is the base change matrix from $\big(x^\ell\big)_{0\le\ell\le n-1}$ to the concatenation of bases of $t_j(x)\mathcal{B}$, $0\le j<2k$, and by construction

$$B_m^{2km} = \begin{pmatrix} 1 & & & \\ & I_{m-1} & & J_{m-1} \\ & & 1 & \\ & I_{m-1} & & -J_{m-1} \end{pmatrix} \oplus I_{k-1}\otimes \begin{pmatrix} 1 & 1 & & \\ & & I_{m-1} & J_{m-1} \\ -1 & 1 & & \\ & & I_{m-1} & -J_{m-1} \end{pmatrix}\cdot L_k^{2km}. \tag{C.4}$$

M is constructed as follows. Let

$$M_0 = \mathbf{1}_k\otimes\begin{pmatrix} 1 & & & \\ & I_{m-1} & & \\ & & & 1 \\ & J_{m-1} & & \end{pmatrix}.$$

Let $M_0(j_0,\dots,j_\ell)$ be the subset of columns of M_0 with indices $j_0,\dots,j_\ell$; and let $D_j=\operatorname{diag}\Big(t_j(\alpha_i)\Big)_{0\le i<n}$,

for $0 \leq j < 2k$. Then

$$M = \Big(D_0 M_0 \mid D_1 M_1 \mid D_2 M_2 \mid \cdots \mid D_{2k-1} M_{2k-1} \Big),$$

where $M_1 = M_0(1, \ldots, m-1)$; $M_{2j} = M_0(0, \ldots, m-1)$ and $M_{2j+1} = M_0(1, \ldots, m)$ for $1 \leq j < k$. We can further rewrite M as

$$M = L_k^{2km}(I_{2m} \otimes \mathrm{DFT}_k) X_m^{2km} L_{2m}^{2km}(I_m \oplus Z_m^{-1} \oplus I_{2(k-1)m}).$$

Here, matrix X_m^{2km} has the structure

$$X_m^{2km} = \begin{pmatrix} I_k & & & \\ & \oplus_{j=1}^{m-1} C_j & \oplus_{j=1}^{m-1} D_j & \\ & & & F \\ & \oslash_{j=m+1}^{2m-1} C_j & \oslash_{j=1}^{m-1} D_j & \end{pmatrix}, \tag{C.5}$$

where

$$\begin{aligned} C_j &= 1 \oplus \mathrm{diag}\left(\omega_{2km}^{j\ell}(\omega_{2m}^j + 1)/2 \right)_{1 \leq \ell < k}, \\ D_j &= \left((\omega_{2m}^j - \omega_{2m}^{-j})/2 \right) \oplus \mathrm{diag}\left(\omega_{2km}^{j\ell}(\omega_{2m}^j - 1)/2 \right)_{1 \leq \ell < k}, \\ F &= 1 \oplus \mathrm{diag}\left(-\omega_{2k}^j \right)_{1 \leq j < k}. \end{aligned}$$

After the substitution of $\overline{\text{DST-I}}_{m-1}$, $\overline{\text{DCT-II}}_m$, and $\overline{\text{DST-II}}_m$ with DST-I_{m-1}, DCT-II_m, and DST-II_m using (C.1-C.3), and simplification, we obtain the factorization

$$\begin{aligned} \mathrm{DFT}_{2km} &= L_k^{2km}(I_{2m} \otimes \mathrm{DFT}_k) X_m^{2km} L_{2m}^{2km}(I_m \oplus Z_m^{-1} \oplus I_{2(k-1)m}) D_m^{2km} \\ &\quad \cdot \Big(\text{DCT-I}_{m+1} \oplus \text{DST-I}_{m-1} \oplus I_{k-1} \otimes (\text{DCT-II}_m \oplus \text{DST-II}_m) \Big) B_m^{2km}, \end{aligned}$$

where B_m^{2km} and X_m^{2km} are defined in (C.4) and (C.5), and

$$\begin{aligned} D_m^{2km} &= I_{m+1} \oplus \operatorname{diag}\Big(1/\sin\frac{(j+1)\pi}{m}\Big)_{0\le j<m-1} \\ &\quad \oplus I_{k-1} \otimes \Big(\operatorname{diag}\Big(1/\cos\frac{j\pi}{2m}\Big)_{0\le j<m} \oplus \operatorname{diag}\Big(1/\sin\frac{(j+1)\pi}{2m}\Big)_{0\le j<m}\Big). \end{aligned} \tag{C.6}$$

C.2 Proof of Theorem 7.1.2.

Consider $\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/2T_{2km}(x)$ with basis $\big(V_0(x), V_1(x), \ldots, V_{2km-1}(x)\big)$. The corresponding polynomial transform is

$$\operatorname{diag}\Big(1/\cos\frac{(k+1/2)\pi}{4km}\Big)_{0\le k<2km} \cdot \text{DCT-IV}_{2km} = \overline{\text{DCT-IV}}_{2km}. \tag{C.7}$$

By Theorem 6.1.2, the polynomial $r(x) = T_{2k}(x)$ generates the subalgebra

$$\mathcal{B} = \big\langle r(x) \big\rangle \cong \mathbb{C}[y]/2T_m(y).$$

By Theorem 6.2.5, the $\mathcal{B}$-module $V_j(x)\mathcal{B} \cong \mathbb{C}[y]/2T_m(y)$ for any $0 \le j < k$. If we choose the basis $\big(V_\ell(y)\big)_{0\le\ell<m}$, then the polynomial transform is

$$\Big[T_\ell(\cos\frac{(k+1/2)\pi}{m})\Big]_{0\le k,\ell<m} = \text{DCT-III}_m\,.$$

Similarly, the $\mathcal{B}$-module $W_j(x)(V_{2k-1}(x) - V_{2k}(x))/2 \cdot \mathcal{B} \cong \mathbb{C}[y]/2T_m(y)$ for any $0 \le j < k$. If we choose the basis If we choose the basis $\big(U_\ell(y)\big)_{0\le\ell<m}$, then the polynomial transform is

$$\Big[U_\ell(\cos\frac{(k+1/2)\pi}{m})\Big]_{0\le k,\ell<m-1} = \operatorname{diag}\Big(1/\sin\frac{(k+1/2)\pi}{m}\Big)_{0\le k<m} \cdot \text{DST-III}_m = \overline{\text{DST-III}}_m. \tag{C.8}$$

We can verify using Theorem 6.2.6 that $t_{2j} = V_j(x)$ and $t_{2j+1} = W_j(x)(V_{2k-1}(x) - V_{2k}(x))/2$ for $0 \le j < k$, is a transversal of $\mathcal{B}$ in $\mathcal{A}$. Hence, by Theorem 6.3.1, we obtain the decomposition

$$\overline{\text{DCT-IV}}_{2km} = M\,\big(I_k \otimes (\overline{\text{DCT-III}}_m \oplus \overline{\text{DST-III}}_m)\big)\,B.$$

Here, B is the base change matrix from $\left(x^\ell\right)_{0\le\ell\le n-1}$ to the concatenation of bases of $t_j(x)\mathcal{B}$, $0 \le j < 2k$, and by construction

$$B = I_k \otimes \begin{pmatrix} 1 & & \\ & L_2^{2(m-1)} \cdot I_{m-1} \otimes \mathrm{DFT}_2 & \\ & & 1 \end{pmatrix} (K_{2m}^{2km})^T.$$

M is constructed as follows. Let

$$M_0 = \mathbf{1}_k \otimes \begin{pmatrix} I_m \\ J_m \end{pmatrix}.$$

Let $D_j = \mathrm{diag}\left(t_j(\alpha_i)\right)_{0\le i<n}$ for $0 \le j < 2k$. Then

$$M = \left(D_0M_0 \mid D_1M_0 \mid D_2M_0 \mid \cdots \mid D_{2k-1}M_0 \right).$$

We can simplify matrix M. Let us introduce matrices

$$X_k^{(C4)}(r) = \begin{pmatrix} c_0 & & & s_{k-1} \\ & \ddots & \iddots & \\ & \iddots & \ddots & \\ s_0 & & & c_{k-1} \end{pmatrix}, \quad X_k^{(S4)}(r) = \begin{pmatrix} c_0 & & & -s_{k-1} \\ & \ddots & \iddots & \\ & \iddots & \ddots & \\ -s_0 & & & c_{k-1} \end{pmatrix}. \tag{C.9}$$

Here, $c_\ell = \cos\frac{(1-2r)(2\ell+1)\pi}{4k}$ and $s_\ell = \sin\frac{(1-2r)(2\ell+1)\pi}{4k}$. These matrices are used for the so-called *skew* DCT and DST [5]. Further, let us define $r^{(i)} = (2i+1)/(4m)$ and

$$r_j^{(i)} = \begin{cases} \frac{r^{(i)}+2j}{k}, & \text{if } j \text{ is even} \\ \frac{2-r^{(i)}+2j}{k}, & \text{if } j \text{ is odd} \end{cases}$$

for $0 \le j < \lfloor \frac{k}{2} \rfloor$. In case k is odd, we also define $r_{k-1}^{(i)} = \frac{r^{(i)}-1}{k} + 1$. Finally, let us define diagonal matrices

$$\begin{aligned} D_k^{(C4)}(r^{(i)}) &= \operatorname{diag}\left(1/\cos\left(r_j^{(i)}\pi/2\right)\right)_{0\le j<k}, \\ D_k^{(S4)}(r^{(i)}) &= \operatorname{diag}\left(\sin\left(2kr_j^{(i)}\pi\right)/\cos\left(r_j^{(i)}\pi/2\right)\right)_{0\le j<k}. \end{aligned}$$

Then $M = K_k^n \widehat{M} L_{2m}^n$, where $\widehat{M} =$

$$\begin{pmatrix} \oplus_{i=0}^{m-1} D_k^{(C4)}(r^{(i)})\,\text{DCT-IV}_k(r^{(i)})X_k^{(C4)}(r^{(i)}) & \oplus_{i=0}^{m-1} D_k^{(S4)}(r^{(i)})\,\text{DST-IV}_k(r^{(i)})X_k^{(S4)}(r^{(i)}) \\ \oslash_{i=m}^{2m-1} D_k^{(C4)}(r^{(i)})\,\text{DCT-IV}_k(r^{(i)})X_k^{(C4)}(r^{(i)}) & \oslash_{i=m}^{2m-1} D_k^{(S4)}(r^{(i)})\,\text{DST-IV}_k(r^{(i)})X_k^{(S4)}(r^{(i)}) \end{pmatrix}.$$

We can further simplify (C.7) by substituting $\overline{\text{DCT-IV}}_{2km}$ and $\overline{\text{DST-III}}_m$ with DCT-IV_{2km} DST-III_m using (C.7) and (C.8). Then we use the equalities

$$\begin{aligned} X_k^{(C4)}(r) &= X_k^{(S4)}(1-r), \\ \text{DST-III}_m &= \operatorname{diag}\left((-1)^j\right)_{0\le j<m} \cdot \text{DCT-III}_m \cdot J_m, \\ \text{DST-IV}_k &= \operatorname{diag}\left((-1)^j\right)_{0\le j<k} \cdot \text{DCT-IV}_k \cdot J_k, \end{aligned}$$

to obtain the decomposition

$$\begin{aligned} \text{DCT-IV}_{2km} &= K_k^{2km}(K_2^{2m} \otimes \text{DCT-IV}_k)Y_m^{2km}(\text{DCT-III}_m \otimes L_2^{2k})(K_{2k}^{2km})^T \\ &\quad \cdot I_k \otimes \begin{pmatrix} 1 & & \\ & L_2^{2(m-1)} \cdot I_{m-1} \otimes \text{DFT}_2 & \\ & & 1 \end{pmatrix} (K_{2m}^{2km})^T, \end{aligned}$$

where

$$Y_m^{2km} = \bigoplus_{j=0}^{m-1} \begin{pmatrix} X_k^{(C4)}(r^{(j)}) & (-1)^j \cdot J_k \cdot X_k^{(C4)}(1-r^{(j)}) \\ X_k^{(C4)}(1-r^{(j)}) & (-1)^{j+1} \cdot J_k \cdot X_k^{(C4)}(r^{(j)}) \end{pmatrix} \tag{C.10}$$

and $X_k^{(C4)}(r)$ is defined in (C.9).

Bibliography

[1] M. Püschel, "Cooley-Tukey FFT like algorithms for the DCT," in *Proc. ICASSP*, 2003, vol. 2, pp. 501–504.

[2] M. Püschel and J. M. F. Moura, "The algebraic approach to the discrete cosine and sine transforms and their fast algorithms," *SIAM Journal of Computing*, vol. 32, no. 5, pp. 1280–1316, 2003.

[3] M. Püschel and J. M. F. Moura, "Algebraic signal processing theory," available at http://arxiv.org/abs/cs.IT/0612077, parts of this manuscript are published as [4] and [5].

[4] M. Püschel and J. M. F. Moura, "Algebraic signal processing theory: Foundation and 1-D time," *IEEE Trans. on Signal Proc.*, vol. 56, no. 8, pp. 3572–3585, 2008.

[5] M. Püschel and J. M. F. Moura, "Algebraic signal processing theory: 1-D space," *IEEE Trans. on Signal Proc.*, vol. 56, no. 8, pp. 3586–3599, 2008.

[6] M. Püschel and J. M. F. Moura, "Algebraic signal processing theory: Cooley-Tukey type algorithms for DCTs and DSTs," *IEEE Trans. on Signal Proc.*, vol. 56, no. 4, pp. 1502–1521, 2008.

[7] M. Püschel and M. Rötteler, "The Discrete Triangle Transform," in *Proc. ICASSP*, 2004.

[8] M. Püschel and M. Rötteler, "Cooley-Tukey FFT like fast algorithms for the discrete triangle transform," in *Proc. 11th IEEE DSP Workshop*, 2004.

[9] M. Püschel and M. Rötteler, "Fourier transform for the directed quincunx lattice," in *Proc. ICASSP*, 2005.

[10] M. Püschel and M. Rötteler, "Fourier transform for the spatial quincunx lattice," in *Proc. ICIP*, 2005.

[11] M. Püschel and M. Rötteler, "Algebraic signal processing theory: 2-D hexagonal spatial lattice," *IEEE Transactions on Image Processing*, vol. 16, no. 6, pp. 1506–1521, 2007.

[12] H. S. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*, Prentice Hall, 1975.

[13] A. V. Oppenheim, A. S. Willsky, and S. Hamid, *Signals and Systems*, Prentice Hall, 2nd edition, 1996.

[14] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, Prentice Hall, 2nd edition, 1999.

[15] J. G. Proakis and D. K. Manolakis, *Digital Signal Processing*, Prentice Hall, 4th edition, 2006.

[16] T. J. Rivlin, *The Chebyshev Polynomials*, Wiley Interscience, 1974.

[17] J. C. Mason and D. C. Handscomb, *Chebyshev Polynomials*, Chapman and Hall/CRC, 2002.

[18] D. S. Dummit and R. M. Foote, *Abstract Algebra*, Wiley, 3rd edition, 2003.

[19] W. C. Curtis and I. Reiner, *Representation Theory of Finite Groups*, Interscience, 1962.

[20] H. J. Nussbaumer, *Fast Fourier Transformation and Convolution Algorithms*, Springer, 2nd edition, 1982.

[21] P. A. Fuhrman, *A Polynomial Approach to Linear Algebra*, Springer Verlag, New York, 1996.

[22] J. Kovačević and M. Püschel, "Algebraic signal processing theory: Sampling for infinite and finite 1-D space," *IEEE Transactions on Signal Processing*, vol. 58, no. 1, pp. 242–257, 2010.

[23] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding*, Signal Processing, Prentice Hall, Englewood Cliffs, NJ, 1995.

[24] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge University, revised edition, 1997.

[25] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice Hall, 1993.

[26] J. Kovačević and M. Vetterli, "Perfect reconstruction filter banks with rational sampling rate changes," in *Proc. ICASSP*, 1991, pp. 1785–1788.

[27] J. Kovačević, *Filter Banks and Wavelets: Extensions and Applications*, Ph.D. thesis, Columbia University, 1991.

[28] J. Kovačević and M. Vetterli, "Perfect reconstruction filter banks with rational sampling factors," *IEEE Trans. on Signal Proc.*, vol. 41, no. 6, pp. 2047–2060, 1993.

[29] I. Bayram and I. W. Selesnik, "Orthonormal FBs with rational sampling factors and oversampled DFT-modulated FBs: a connection and filter design," *IEEE Trans. on Signal Proc.*, vol. 57, no. 7, pp. 2515–2526, 2009.

[30] E. T. Whittaker, "On the functions which are represented by the expansions of the interpolation theory," *Proc. Royal Soc. Edinburgh*, vol. 35, pp. 181–194, 1915.

[31] H. Nyquist, "Certain topics in telegraph transmission theory," *Trans. AIEE*, vol. 47, pp. 617–644, 1928.

[32] V. A. Kotelnikov, "On the carrying capacity of the ether and wire in telecommunications," *First All-Union Conference on Questions of Communication, Moscow*, 1933.

[33] C. E. Shannon, "Communication in the presence of noise," *Proc. Institute of Radio Engineers*, vol. 37, no. 1, pp. 10–21, 1949.

[34] M. Unser, "Sampling—50 years after Shannon," *Proc. IEEE*, vol. 88, no. 3, pp. 569–587, 2000.

[35] R. Tolimieri, M. An, and C. Lu, *Algorithms for Discrete Fourier Transforms and Convolution*, Springer, 2nd edition, 1997.

[36] A. Croisier, D. Esteban, and C. Galand, "Perfect channel splitting by use of interpolation/decimation-/tree decomposition techniques," in *Proc. Int. Conf. Inform. Sciences and Syst.*, Aug. 1976, pp. 443–446.

[37] M. Vetterli, "Filter banks allowing perfect reconstruction," *Signal Processing*, vol. 10, no. 3, pp. 219–244, 1986.

[38] M. J. Smith and T. P. Barnwell III, "Exact reconstruction for tree-structured subband coders," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 34, no. 3, pp. 431–441, 1986.

[39] P. P. Vaidyanathan, "Quadrature mirror filter banks, M-band extensions and perfect reconstruction techniques," *IEEE ASSP Mag.*, vol. 4, no. 3, pp. 4–20, 1987.

[40] P. P. Vaidyanathan, "Theory and design of M-channel maximally decimated quadrature mirror filters with arbitrary m, having the perfect reconstruction property," *IEEE Trans. on Signal Processing*, vol. 35, no. 4, pp. 476–482, 1987.

[41] J. Kovacevic and A. Chebira, "Life beyond bases: the advent of frames (part I)," *IEEE Signal Proc.*, vol. 24, no. 4, pp. 86–104, 2007.

[42] H. S. Malvar, *Signal Processing with Lapped Transforms*, Artech House, Norwood, MA, 1992.

[43] A. Chebira and J. Kovačević, "Lapped tight frame transforms," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Proc.*, 2007, pp. 857–860.

[44] A. Sandryhaila, A. Chebira, J. Kovacevic, and M. Püschel, "A new class of seeded real lapped tight frame transforms," in *Proc. SPIE Conf. on Wavelet Appl. in Signal and Image Proc. (Wavelets XIII)*, 2009, vol. 7446, pp. 74460M–74460M–8.

[45] A. Sandryhaila, A. Chebira, C. Milo, J. Kovacevic, and M. Püschel, "Systematic construction of real lapped tight frame transforms," *IEEE Transactions on Signal Processing*, vol. 58, no. 5, pp. 2556–2567, 2010.

[46] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 1999.

[47] V. K. Goyal, J. Kovačević, and J. A. Kelner, "Quantized frame expansions with erasures," *Journ. Appl. and Comput. Harmonic Analysis*, vol. 10, no. 3, pp. 203–233, 2001.

[48] M. Püschel and J. Kovacevic, "Real, tight frames maximally robust to erasures," in *Data Compression Conference (DCC)*, 2005, pp. 63–72.

[49] A. Haar, "Zur theorie der orthogonalen funktionensysteme," *Mathematische Annalen*, vol. 69, pp. 331–371, 1910.

[50] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Commun. Pure Appl. Math.*, vol. XLI, pp. 909–996, 1988.

[51] I. Daubechies, *Ten Lectures on Wavelets*, SIAM, 1992.

[52] A. Cohen, I. Daubechies, and J. C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Commun. Pure Appl. Math.*, vol. XLV, pp. 485–560, 1992.

[53] I. Daubechies, "Orthonormal bases of compactly supported wavelets II. variations on a theme," *SIAM J. Math. Anal.*, vol. 24, no. 2, 1993.

[54] A. Cohen and I. Daubechies, "Orthonormal bases of compactly supported wavelets III. better frequency localization," *SIAM J. Math. Anal.*, vol. 24, no. 2, 1993.

[55] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. of Computation*, vol. 19, pp. 297–301, 1965.

[56] M. Vetterli and H. J. Nussbaumer, "Simple FFT and DCT algorithms with reduced number of operations," *Signal Processing*, vol. 6, pp. 267–278, 1984.

[57] P. Duhamel and M. Vetterli, "Fast fourier transforms: a tutorial review and a state of the art," *Signal Processing*, vol. 4, no. 19, pp. 259–299, 1990.

[58] C. Van Loan, *Computational Framework of the Fast Fourier Transform*, Siam, 1992.

[59] V. Britanak and K. R. Rao, "The fast generalized discrete Fourier transforms: A unified approach to the discrete sinusoidal transforms computation," *Signal Processing*, vol. 79, pp. 135–150, 1999.

[60] R. N. Bracewell, "The fast Hartley transform," *Proc. IEEE*, vol. 72, no. 8, pp. 1010–1018, 1984.

[61] Y. Voronenko and M. Püschel, "Algebraic signal processing theory: Cooley-Tukey type algorithms for real DFTs," *IEEE Trans. Signal Proc.*, vol. 57, no. 1, pp. 205–222, 2009.

[62] P. Yip and K. R. Rao, "A fast computational algorithm for the discrete sine transform," *IEEE Trans. on Communications*, vol. COM-28, no. 2, pp. 304–307, 1980.

[63] P. Yip and K. R. Rao, "Fast decimation-in-time algorithms for a family of discrete sine and cosine transforms," *Circuits, Systems, and Signal Processing*, vol. 3, no. 4, pp. 387–408, 1984.

[64] P. Yip and K. R. Rao, "The decimation-in-frequency algorithms for a family of discrete sine and cosine transforms," *Circuits, Systems, and Signal Processing*, vol. 7, no. 1, pp. 3–19, 1988.

[65] Z. Wang, "A fast algorithm for the discrete sine transform implemented by the fast cosine transform," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-30, no. 5, pp. 814–815, 1982.

[66] Z. Wang, "Fast algorithms for the discrete W transform and for the discrete Fourier transform," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-32, no. 4, pp. 803–816, 1984.

[67] Z. Wang, "On computing the discrete Fourier and cosine transforms," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-33, no. 4, pp. 1341–1344, 1985.

[68] Z. Wang, "Fast discrete sine transform algorithms," *Signal Processing*, vol. 19, pp. 91–102, 1990.

[69] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*, Academic Press, 1990.

[70] E. Feig and S. Winograd, "Fast algorithms for the discrete cosine transform," *IEEE Trans. on Signal Processing*, vol. 40, no. 9, pp. 2174–2193, 1992.

[71] T. Kailath and V. Olshevsky, "Displacement structure approach to discrete trigonometric transform based preconditioners of G. Strang and T. Chan type," *SIAM J. on Matrix Analysis and Appl.*, vol. 26, no. 3, pp. 706–734, 2005.

[72] B. K. Alpert and V. Rokhlin, "A fast algorithm for the evalutation of Legendre expansions," *SIAM J. Sci. Stat. Comput.*, vol. 12, no. 1, pp. 158–179, 1991.

[73] Th. Kailath and V. Olshevsky, "Displacement structure approach to polynomial Vandermonde and related matrices," *Linear Algebra and Applications*, vol. 261, pp. 49–90, 1997.

[74] J. R. Driscoll, D. M. Healy Jr., and D. Rockmore, "Fast discrete polynomial transforms with applications to data analysis for distance transitive graphs," *SIAM Journal Comput.*, vol. 26, pp. 1066–1099, 1997.

[75] D. Potts, G. Steidl, and M. Tasche, "Fast algorithms for discrete polynomial transforms," *Mathematics of Computation*, vol. 67, no. 224, pp. 1577–1590, 1998.

[76] L. Auslander, E. Feig, and S. Winograd, "Abelian semi-simple algebras and algorithms for the discrete Fourier transform," *Advances in Applied Mathematics*, vol. 5, pp. 31–55, 1984.

[77] Th. Beth, *Verfahren der Schnellen Fouriertransformation [Methods for the Fast Fourier Transform]*, Teubner, 1984.

[78] Th. Beth, "On the computational complexity of the general discrete Fourier transform," *Theoretical Computer Science*, vol. 51, pp. 331–339, 1987.

[79] M. Clausen, *Beiträge zum Entwurf schneller Spektraltransformationen (Habilitationsschrift)*, Univ. Karlsruhe, 1988.

[80] P. Diaconis and D. Rockmore, "Efficient computation of the Fourier transform on finite groups," *Amer. Math. Soc.*, vol. 3(2), pp. 297–332, 1990.

[81] D. Rockmore, "Efficient computation of Fourier inversion for finite groups," *Assoc. Comp. Mach.*, vol. 41, no. 1, pp. 31–66, 1994.

[82] M. Clausen and U. Baum, *Fast Fourier Transforms*, BI-Wiss.-Verl., 1993.

[83] M. Püschel, *Konstruktive Darstellungstheorie und Algorithmengenerierung*, Ph.D. thesis, Universität Karlsruhe, Informatik, 1998, translated in [84].

[84] M. Püschel, "Constructive representation theory and fast signal transforms," Tech. Rep. Drexel-MCS-1999-1, Drexel University, Philadelphia, 1999, translation of [83].

[85] Henri J. Nussbaumer and Philippe Quandalle, "Fast computation of discrete Fourier transforms using polynomial transforms," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-27, no. 2, pp. 169–181, 1979.

[86] S. Winograd, "On the multiplicative complexity of the discrete Fourier transform," *Advances in Mathematics*, vol. 32, pp. 83–117, 1979.

[87] H. W. Johnson and C. S. Burrus, "On the structure of efficient DFT algorithmss," *IEEE Trans. Acoust., Speech, Signal Proc.*, , no. 1.

[88] L. Auslander, E. Feig, and S. Winograd, "The multiplicative complexity of the discrete Fourier transform," *Advances in Applied Mathematics*, vol. 5, pp. 87–109, 1984.

[89] M. T. Heideman and C. S. Burrus, "On the number of multiplications necessary to compute a length-2^n DFT," *IEEE Trans. Acoust., Speech, Signal Proc.*, , no. 1, pp. 91–95, 1986.

[90] G. Szegö, *Orthogonal Polynomials*, Amer. Math. Soc. Colloq. Publ., 3rd edition, 1967.

[91] R. Askey, *Orthogonal Polynomials and Special Functions*, SIAM, 1966.

[92] B. S. Kashin and A. A. Saakyan, *Orthogonal Series*, American Mathematical Society, 1989.

[93] A. Jerri, *Integral and Discrete Transforms with Applications and Error Analysis*, CRC Press, 1992.

[94] W. Gautschi, *Orthogonal Polynomials: Computation and Approximation*, Oxford Univ. Press, 2004.

[95] A. Sandryhaila, J. Kovacevic, and M. Püschel, "Haar filter banks for 1-D space signals," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Proc.*, 2008, pp. 3505–3508.

[96] P. Prandoni and M. Vetterli, "From Lagrange to Shannon... and back: Another look at sampling," *IEEE Signal Proc.*, vol. 26, no. 5, pp. 138–144, 2009.

[97] S. Egner and M. Püschel, "Automatic generation of fast discrete signal transforms," *IEEE Trans. on Signal Processing*, vol. 49, no. 9, pp. 1992–2002, 2001.

[98] A. Sandryhaila, J. Kovacevic, and M. Püschel, "Algebraic signal processing theory: Cooley-Tukey type algorithms for polynomial transforms based on induction," submitted for publication.

[99] I. J. Good, "The interaction algorithm and practical Fourier analysis," *Journal Royal Statist. Soc.*, vol. B20, pp. 361–375, 1958.

[100] P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, Academic Press, 2nd edition, 1985.

[101] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge Univ. Press, 1990.

[102] A. Sandryhaila, J. Kovacevic, and M. Püschel, "Compression of QRS complexes using Hermite expansion," submitted for publication.

[103] P. Laguna, R. Jané, S. Olmos, N. V. Thakor, H. Rix, and P. Caminal, "Adaptive estimation of QRS complex wave features of ECG signal by the Hermite model," *J. of Med. and Biol. Eng. and Comput.*, vol. 34, no. 1, pp. 58–68, 1996.

[104] L. Sörnmo, P. O. Börjesson, P. Nygards, and O. Pahlm, "A method for evaluation of QRS shape features using a mathematical model for the ECG," *IEEE Trans. on Biomed. Eng.*, vol. BME-28, no. 10, pp. 713–717, 1981.

[105] L. R. Lo Conte, R. Merletti, and G. V. Sandri, "Hermite expansion of compact support waveforms: Applications to myoelectric signals," *IEEE Trans. on Biomed. Eng.*, vol. 41, no. 12, pp. 1147–1159, 1994.

[106] M. Lagerholm, C. Peterson, G. Braccini, L. Edenbranhdt, and L. Sörnmo, "Clustering ECG complexes using Hermite functions and self-organizing maps," *IEEE Trans. on Biomed. Eng.*, vol. 47, no. 7, pp. 838–848, 2000.

[107] "MIT-BIH ECG Compression Test Database," http://www.physionet.org/physiobank/database/cdb.

[108] J. M. F. Moura and N. Balram, "Recursive structure of noncausal Gauss Markov random fields," *IEEE Trans. Information Theory*, vol. 38, no. 2, pp. 334–354, March 1992.

[109] J. M. F. Moura and M. G. S. Bruno, "DCT/DST and Gauss-Markov fields: Conditions for equivalence," *IEEE Trans. on Signal Processing*, vol. 46, no. 9, pp. 2571–2574, 1998.

[110] W. A. Robinson, *Modeling Dynamic Climate Systems*, Springer, 2001.

[111] W. M. Washington and C. L. Parkinson, *Introduction To Three-dimensional Climate Modeling*, University Science Books, 2nd edition, 2005.

[112] J.-B. Martens, "The Hermite transform—theory," *IEEE Trans. on Acoustics, Speech, and Signal Proc.*, vol. 38, no. 9, pp. 1595–1605, 1990.

[113] J.-B. Martens, "The Hermite transform—applications," *IEEE Trans. on Acoustics, Speech, and Signal Proc.*, vol. 38, no. 9, pp. 1607–1618, 1990.

[114] J.-B. Martens, "Local orientation analysis in images by means of the Hermite transform," *IEEE Trans. on Image Proc.*, vol. 6, no. 8, pp. 1103–1116, 1997.

[115] G. Mandyam and N. Ahmed, "The discrete Laguerre transform: Derivation and applications," *IEEE Trans. on Signal Processing*, vol. 44, no. 12, pp. 2925–2931, 1996.

[116] G. Mandyam, N. Ahmed, and N. Magotra, "Application of the discrete Laguerre transform to speech coding," in *Proc. Asilomar*, 1995, vol. 2, pp. 1225–1228.

[117] M. E. H. Ismail, J. Letessier, and G. Valent, "Linear birth and death models and associated Laguerre and Meixner polynomials," *J. of Approx. Theory*, vol. 55, pp. 337–348, 1988.

[118] H. Dette, "First return probabilities of birth and death chains and associated orthogonal polynomials," *Proc. Amer. Math. Soc.*, vol. 129, pp. 1805–1815, 2000.

[119] E. A. Van Doorn, "Birth-death processes and associated polynomials," *J. of Comput. and Applied Math.*, vol. 153, no. 1-2, pp. 497–506, 2003.

[120] P. Beckmann, *Orthogonal Polynomials for Engineers and Physicists*, Golem Press, 1973.

[121] T. S. Chihara, *An Introduction to Orthogonal Polynomials*, Gordon and Breach, 1978.

[122] J. J. Benedetto and M. C. Fickus, "Finite normalized tight frames," *Adv. Comp. Math., sp. iss. Frames*, vol. 18, pp. 357–385, 2003.

[123] N. I. Akhiezer and I. M. Glazman, *Theory of Linear Operators in Hilbert Spaces*, vol. 1, Frederick Ungar, 1966.

[124] D. Han and D. R. Larson, *Frames, bases and group representations*, Number 697 in Memoirs AMS. Amer. Math. Soc., 2000.

[125] R. N. Bracewell, "Discrete Hartley transform," *J. Optical Society America*, vol. 73, no. 12, pp. 1832–1835, 1983.

[126] T. Kailath, *Linear Systems*, Prentice Hall, Englewood Cliffs, NJ, 1980.

[127] Z. Cvetković and M. Vetterli, "Tight Weyl-Heisenberg frames in $\ell^2(\mathbb{Z})$," *IEEE Trans. Signal Proc.*, vol. 46, no. 5, pp. 1256–1259, 1998.