Digital Pre-Compensation for Faulty D/A Converters: The "Missing Pixel" Problem

by

Sourav Raj Dey

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2004

Author .....
Department of Electrical Engineering and Computer Science
December 14, 2003

Certified by .....
Alan V. Oppenheim
Ford Professor of Engineering
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Digital Pre-Compensation for Faulty D/A Converters:

The "Missing Pixel" Problem

by

Sourav Raj Dey

Submitted to the Department of Electrical Engineering and Computer Science
on December 14, 2003, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

In some contexts, DACs fail in such a way that specific samples are dropped. The dropped samples lead to distortion in the analog reconstruction. We refer to this as the "missing pixel" problem. Under certain conditions, it may be possible to compensate for the dropped sample by pre-processing the digital signal, thereby reducing the analog reconstruction error. We develop three such compensation strategies in this thesis.

The first strategy uses constrained minimization to calculate the optimal finite-length compensation signal. We develop a closed-form solution using the method of Lagrange multipliers. Next, we develop an approximation to the optimal solution using discrete prolate spheroidal sequences. We show that the optimal solution is a linear combination of the discrete prolates. The last compensation technique we develop is an iterative solution in class of projection-onto-convex-sets. We develop the algorithm and prove that it converges to the optimal solution found using constrained minimization. Each of the three strategies are analyzed and results from numerical simulations are presented.

Thesis Supervisor: Alan V. Oppenheim
Title: Ford Professor of Engineering

Acknowledgments

Foremost, I would like to thank my adviser, Al Oppenheim, for his mentorship and guidance. For taking me under his wing as a "green" undergraduate and shaping me into a better engineer, researcher, and person. I hope this is just the beginning of a lifelong collaboration through my doctoral studies and beyond.

I want to also thank the other professors and teachers that have helped me along the way, at MIT, Seoul International School, and Elizabethtown. Your support has brought me here.

To the DSPG members, thanks for making this place the intellectual playground that it is. My conversations with you guys, Andrew, Vijay, Petros, Charles, and Maya have gone a long way to making this thesis a reality. Thanks must also go out to my friend, Ankit, for the many long nights brainstorming on a cheap Home Depot white-board mounted on the wall of our apartment. To all my other friends, especially Sachin, Pallabi, and Arny, thank you for supporting me, both mentally and emotionally, in a very hectic year.

Lastly, to my family. You have been the greatest support of all. Wherever in the world we may live, home is always where my heart is. Piya, you're the greatest sister in the world. Thotho. To my parents, this thesis is culmination of a lifetime of love and support. Thank you for always guiding and pushing me to be all that I can be.

To my family

Contents

List of Figures

Chapter 1

Introduction

Digital to analog converters are one of the most ubiquitous components in digital systems, and, like any other component, DACs can fail. In some contexts digital-to-analog converters (DACs) fail in such a way that specific samples are dropped. There are at least two contexts in which this fault is common: flat-panel video displays and time-interleaved DACs.

Flat-panel displays, such as those found in modern personal computers, are made by placing light emitting diodes (LEDs) on a silicon wafer. Each LED corresponds to one color component of one pixel of the image. One of these LEDs can malfunction and get permanently set to a particular value. These broken LEDs are manifested as missing pixels on the screen, thus we refer to this as the "missing pixel" problem, [8].

Time-interleaved DACs are also prone to this fault. Figure 1-1 shows a model for an N-element array, [2, 15]. Each DAC operates from phase-shifted clocks. These arrays are important in high-speed communication links, since they can be clocked at lower rates, while still achieving high throughput, [2]. If one of the DACs fails, every Nth sample is dropped, leading to distortion in the analog reconstruction.

Under certain conditions, it may be possible to compensate for the dropped sample by pre-processing the digital signal. The aim of this thesis is to develop such compensation strategies. Their advent could lead to flat-panel displays that minimize the visual distortion caused by defective pixel and interleaved DAC arrays that are more robust to failure.

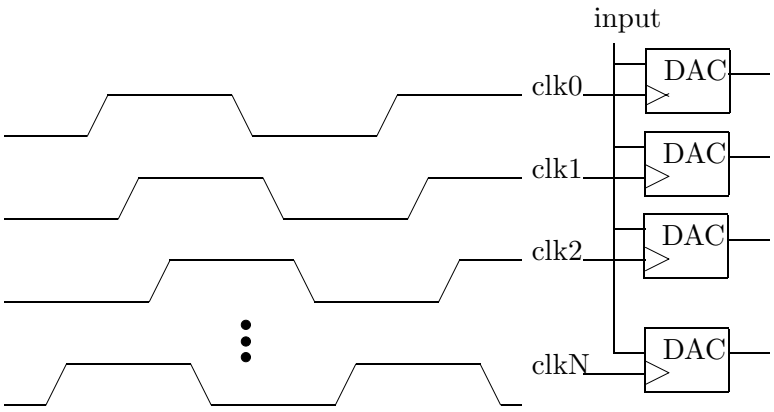

Figure 1-1: Interleaved DAC Array

1.1 Problem Statement

We adhere to the standard mathematical representation for ideal digital-to-analog conversion shown in Figure 1-2. Digital samples, $x[n] = x(nT_s)$, are converted into an impulse train, $x_p(t) = \sum_{n=-\infty}^{\infty} x[n]\delta(t - nT_s)$, through the sample-to-impulses converter (S/I). $x_p(t)$ is then filtered through an ideal low-pass filter (LPF), $H(j\Omega)$, resulting in the reconstruction, $r(t)$. Quantization issues are ignored by assuming that the digital samples, $x[n]$, can be represented with infinite precision. Furthermore, we assume that the original continuous-time (CT) signal, $x(t)$, is at least slightly oversampled. Specifically, we assume that $1/T_s = R\Omega_c/\pi$, where $x(t)$ is band-limited to Ω_c and $R > 1$ is the oversampling ratio. We denote the ratio π/R by γ. The DAC perfectly reconstructs $x(t)$ in the sinc basis,

$$r(t) = \sum_{n=-\infty}^{\infty} x[n] \frac{\sin(\pi(2\Omega_c t - n))}{\pi(2\Omega_c t - n)} = x(t) \tag{1.1}$$

In any practical application, $H(j\Omega)$ is replaced by a non-ideal filter that approximates the ideal LPF. The choice of $H(j\Omega)$ will affect our pre-compensation solution, but, for simplicity, we do not consider the effect of non-ideal $H(j\Omega)$ in this thesis. We assume $H(j\Omega)$ is always an ideal LPF, with the understanding that in practice it will be approximated accurately enough to meet design specifications.

The faulty DAC is mathematically represented as in Figure 1-3. The dropped sample is modeled as multiplication by $(1 - \delta[n])$ that sets $x[0] = 0$. Without loss of generality, we assume that the dropped sample is at index $n = 0$ and that the dropped sample is set to

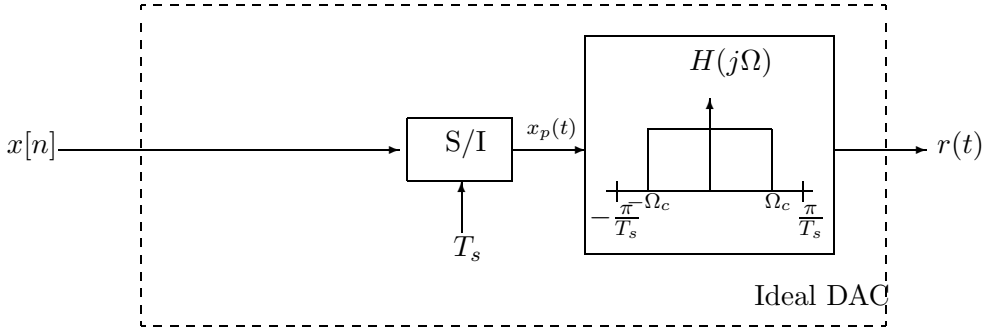

Figure 1-2: Ideal DAC

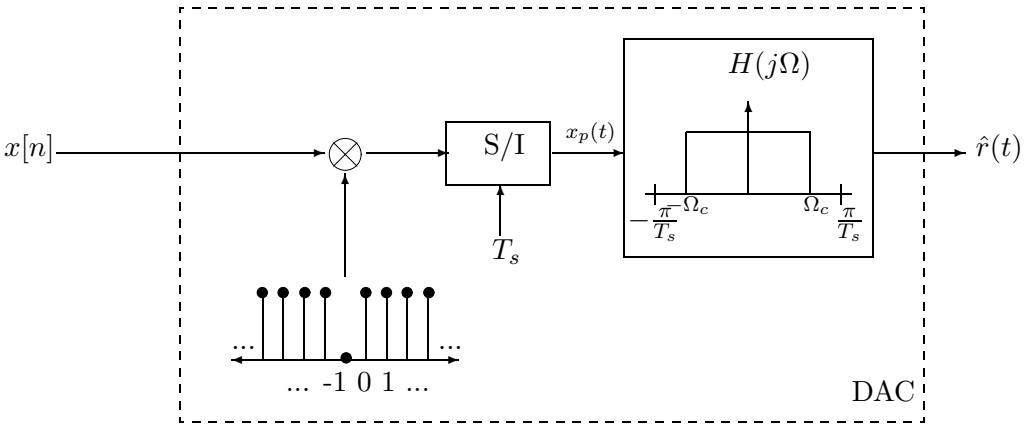

Figure 1-3: Faulty DAC

zero. Because of the dropped sample, the reconstruction, $\hat{r}(t)$, is a distorted version of the desired reconstruction, $r(t)$.

It is important to note that this problem is not one of data recovery. The sample that is dropped is known exactly inside the digital system. The problem is one of data conversion, the DAC is broken so that a specific sample cannot be used in the reconstruction. The goal is to represent the data differently in the digital domain, so we can achieve the same reconstruction. As such, interpolation and other data recovery techniques are useless.

Our goal is to pre-compensate the digital signal $x[n]$, so the distortion caused by dropping $x[0]$ is reduced. This requires altering samples that are not dropped by the DAC. Figure 1-4 illustrates the compensated DAC. Compensation is portrayed as a signal $c[n]$ that is added to $x[n]$. In general, compensation could be some complicated function of the dropped sample and neighbors. In our development, we restrict compensation to be an affine transformation of $x[n]$.

We use the squared-$\mathcal{L}_2$ energy of the error signal, $e(t) = r(t) - \hat{r}(t)$, as the error metric

$$\epsilon^2 = \int_{-\infty}^{\infty} |r(t) - \hat{r}(t)|^2 dt \tag{1.2}$$

The problem can be equivalently cast directly in the discrete-time domain as shown in Figure 1-5. In Figure 1-4, we can move the DACs to the right of the summing junction and add a ideal discrete-time LPF, $H(e^{j\omega})$, with cutoff $\gamma = \Omega_c/T_s = \pi/R$ before the DAC. The cascade of $H(e^{j\omega})$ and the DAC is equivalent to just a DAC, i.e. $H(e^{j\omega})$ does not filter any additional energy compared to the DAC. Thus we can move $H(e^{j\omega})$ left through the summing junction. Since $x[n]$ is band-limited to γ by definition, $H(e^{j\omega})$ is just a identity transformation on the lower branch, so it can be dropped.

Additionally, there are two places in Figure 1-4 where error enters the system. First from the dropped sample, $x[0]$, and secondly from the compensation signal, $c[n]$. Tracking the effects of two sources of error is difficult, so to simplify we incorporate the dropped sample into the compensation as a constraint: $c[0] = -x[0]$. This constraint ensures that $x[0]$ is zero before entering the DAC, so dropping it does not cause any additional distortion. Figure 1-5 illustrates our final, simplified representation. We equivalently use the squared-ℓ_2 energy of $e[n]$ as the error metric

$$\varepsilon^2 = \sum_{-\infty}^{\infty} |e[n]|^2 \tag{1.3}$$

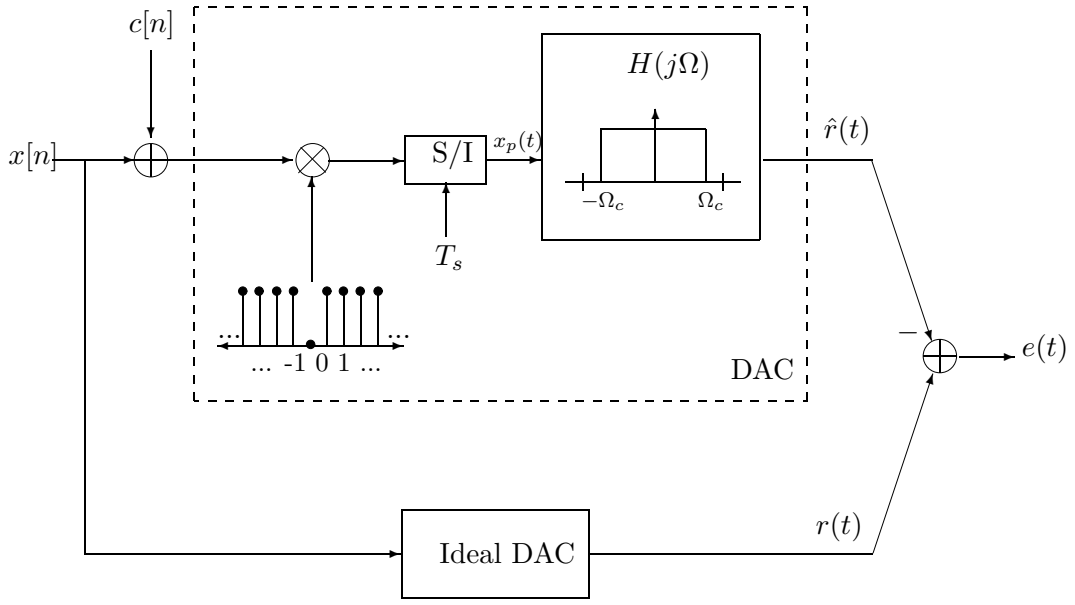

Figure 1-4: Compensated DAC

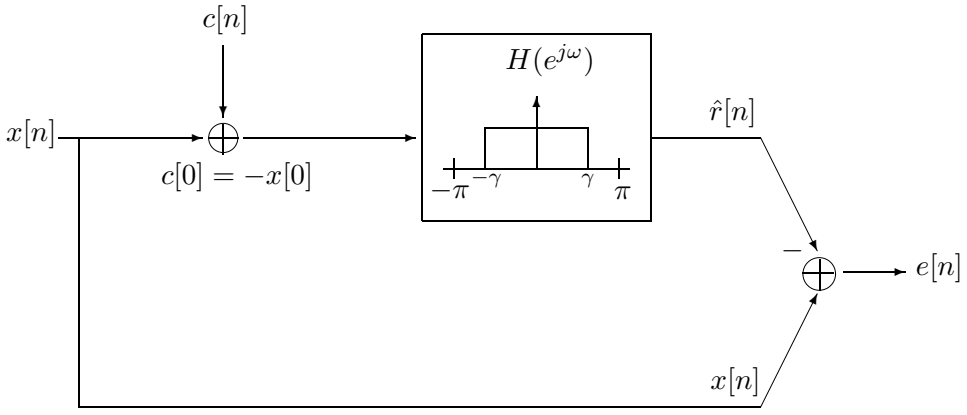

Figure 1-5: Simplified Representation

1.2 Constraints

We impose two constraints on digital pre-compensation of the dropped sample. The first, oversampling, is a direct result of the formulation. The second, a symmetric $c[n]$, is imposed for simplification.

1.2.1 Oversampling

In Figure 1-5, the error signal, $e[n]$, can be expressed as

$$\begin{aligned} e[n] &= x[n] - \hat{r}[n] && (1.4) \\ &= x[n] - h[n] * (x[n] + c[n]) && (1.5) \end{aligned}$$

In the frequency domain

$$E(e^{j\omega}) = X(e^{j\omega}) - H(e^{j\omega})X(e^{j\omega}) - H(e^{j\omega})C(e^{j\omega}) \quad (1.6)$$

Since $X(e^{j\omega})$ is band-limited inside the passband of $H(e^{j\omega})$, (1.6) reduces to

$$E(e^{j\omega}) = -H(e^{j\omega})C(e^{j\omega}) \quad (1.7)$$

Using Parseval's relation, the error ε^2 reduces to

$$\begin{aligned} \varepsilon^2 = \sum_{n=-\infty}^{\infty} |e[n]|^2 &= \int_{<2\pi>} |E(e^{j\omega})|^2 d\omega && (1.8) \\ &= \int_{<2\pi>} |H(e^{j\omega})C(e^{j\omega})|^2 d\omega && (1.9) \end{aligned}$$

Minimizing ε^2 is equivalent to minimizing the energy of $C(e^{j\omega})$ in the band $[-\gamma,\gamma]$, i.e. in the pass-band of the filter $H(e^{j\omega})$. This implies that $x(t)$ must be at least slightly oversampled for compensation.

There are some subtleties involved in this condition. Oversampling is generally defined as $x(t)$ being sampled at a rate $1/T_s > 2\Omega_c$. This is an open-set, so the limit-point, $1/T_s = 2\Omega_c$, does not exist in the set. Assuming that $x(t)$ is sampled at exactly $1/T_s = 2\Omega_c$, there is aliasing at the point $\omega = \pi$. In this limiting case, the reconstruction filter, $H(e^{j\omega})$, can be

chosen to be

$$\begin{aligned} H(e^{j\omega}) &= 1, \omega \neq \pi \\ H(e^{j\omega}) &= 0, \omega = \pi \end{aligned} \tag{1.10}$$

It eliminates the aliased point at $\omega = \pi$, and, since a point has measure zero, there is still perfect reconstruction in the ℓ_2 sense. The compensation signal, unless it is an impulse at $\omega = \pi$, will be passed unchanged to the output. Unfortunately, as Section 1.3.2 develops, choosing $c[n]$ as a properly scaled impulse at $\omega = \pi$, we can have perfect compensation. So, at least from a formal mathematical viewpoint, we can compensate the limiting case. For this thesis though, we do not consider such measure zero subtleties as they are non-physical. In the limiting case, we assume that the reconstruction filter must be the identity filter defined as

$$H(e^{j\omega}) = 1 \tag{1.11}$$

Thus, compensating with impulses will not work and from (1.8), the energy of $c[n]$ is the error.

$$\varepsilon^2 = \int_{<2\pi>} |C(e^{j\omega})|^2 d\omega \tag{1.12}$$

In this degenerate case, the optimal solution is to meet the constraint, $c[0] = -x[0]$, and set the other values of $c[n] = 0$, for $n \neq 0$. This is equivalent to having no compensation and letting the faulty DAC drop the sample $x[0]$. There is no gain in compensating. Proper compensation requires that $x(t)$ be sampled at a rate $1/T_s > 2\Omega_c + \epsilon$, for $\epsilon \neq 0$ but otherwise arbitrarily small. We use this as our definition of oversampling. As before, $\pi/R = \gamma$ where $R > 1$ is the oversampling ratio with $1/T_s = R\Omega_c/\pi$.

When oversampled according to this definition, $X(e^{j\omega})$ is band-limited to $\gamma < \pi$. As illustrated in Figure 1-6, the reconstruction filter, $H(e^{j\omega})$ can be designed with a high-frequency stop-band, $\gamma < |\omega| < \pi$. $c[n]$ can then be designed such that most of its energy is in the stop-band of $H(e^{j\omega})$, minimizing the final error.

In fact, increasing oversampling reduces error even without compensation. The faulty DAC with no compensation can be represented in our simplified model as a degenerate compensation signal

$$c[n] = -x[0]\delta[n] \tag{1.13}$$

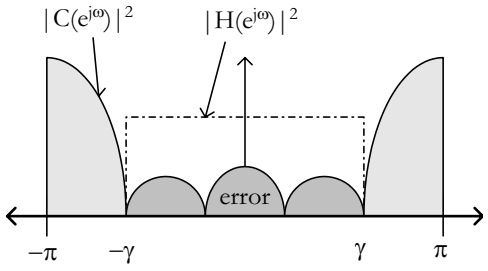

Figure 1-6: Oversampling and Compensation

In the frequency domain, $C(e^{j\omega})$ is a constant $-x[0]$. Figure 1-7 illustrates the resulting error. Mathematically, the expression for the error is

$$\varepsilon^2 = 2\gamma x^2[0] \tag{1.14}$$

As the oversampling rate increases, γ decreases, reducing the error accordingly. Intuitively, oversampling introduces redundancy into the system so dropping one sample constitutes a smaller loss of information than if the signal were sampled at the Nyquist rate.

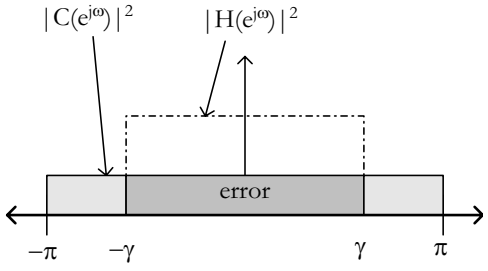

Figure 1-7: Error without Compensation

1.2.2 Symmetric Compensation

In general, affine pre-compensation can alter any arbitrary set of samples in $x[n]$. For clarity in the exposition, we focus on symmetric compensation, where $(N-1)/2$ neighboring samples on either side of the dropped sample are altered, i.e. $c[n]$ is a symmetric signal centered around $n = 0$.

In contexts where the location of the dropped sample is known a priori, such as with missing pixels on a video-display, symmetric compensation is the practical choice. In some

contexts though, symmetric compensation is not feasible. For example, for a DAC that drops samples in a streaming system the best we can do is to detect the dropped samples and compensate causally. The compensation signal will be one-sided and asymmetric. Where the extension to the more general case is obvious, we note the structure of the asymmetric solution.

1.3 Previous Work

This thesis is, in part, an extension of the work done by Russell in [8]. In this section, we review some of the results presented in [8].

1.3.1 Alternative Formulation

The "missing pixel" problem was originally formulated in [8] as a resampling problem. Resampling is the process of converting between two digital representations of an analog signal, each of which is on a different sampling grid. Specifically, in the case of one dropped sample, define an index set I to be the set of integers

$$I = \{0, \pm 1, \pm 2, \pm 3, \ldots\}$$

and a set I' to be the set of integers with zero removed

$$I' = \{\pm 1, \pm 2, \pm 3, \ldots\}$$

An analog signal $x(t)$ is digitally represented by its samples $\{x_n\}$ on I. Low-pass filtering the digital sequence $\{x_n\}$ through a filter, $h(t)$, reconstructs $x(t)$. The goal of compensation is to find coefficients $\{x'_n\}$ on I' which satisfy

$$x(t) = \sum_{n \in I'} x'_n h(t - n) \tag{1.15}$$

or equivalently

$$x(t) = \sum_{n=-\infty}^{\infty} x'_n h(t - n) \tag{1.16}$$

with the constraint

$$x'_0 = 0$$

By requiring $x'_0 = 0$ the sum may be taken over all integers. In this thesis, we take a different viewpoint and pose the problem in the context of DACs. Both formulations are equivalent when $h(t)$ is restricted to be an ideal LPF. The DAC representation is preferred in this thesis because of its practical relevance.

1.3.2 Perfect Compensation

If $c[n]$ had no frequency component outside $|\omega| > \pi - \gamma$ while meeting the constraint $c[0] = -x[0]$, it would perfectly compensate with zero error. There are an unlimited number of signals that meet this criteria. For example, we can simply choose

$$c_{\text{inf}}[n] = -x[0](-1)^n \tag{1.17}$$

This signal meets the constraint that $c_{inf}[0] = -x[0]$ and since its spectrum is

$$C(e^{j\omega}) = \frac{-x[0]}{2}(\delta(\omega - \pi) + \delta(\omega + \pi)) \tag{1.18}$$

$C(e^{j\omega})$ is completely high-pass, with zero energy in $\gamma < |\omega| < \pi$. $c_{\text{inf}}[n]$ perfectly compensates for the dropped sample. This solution only requires in theory that $R = 1 + \epsilon$, where ϵ is non-zero but otherwise arbitrarily small. Russell derives this result in [8] within a resampling framework.

All other perfect compensation solutions are signals band-limited to $|\omega| < \gamma$ multiplied by $c_{\text{inf}}[n]$. Of these choices, the minimum energy solution is

$$c_{\text{sinc}}[n] = -x[0](-1)^n \frac{\sin(\pi - \gamma)n}{(\pi - \gamma)n} \tag{1.19}$$

Unfortunately, all of these signals, although resulting in zero error, have infinite length, making them impractical to implement. Russell in [8] develops an optimal finite-length compensation strategy using constrained minimization. In that spirit, this thesis focuses exclusively on finite-length compensation using perfect compensation and optimal finite-length compensation as a starting point for the more sophisticated algorithms.

1.3.3 Missing Pixel Compensation

In the case of video-displays, the "missing pixel" problem is of practical interest and some ad hoc solutions have been proposed, [6, 4]. In these approaches neighboring pixels are brightened in order to compensate. The idea is based on the fact that the missing pixel looks dark, so making the surrounding pixels brighter reduces the visual distortion. Though several weightings are proposed, no theory is developed.

In [8] Russell implements a two-dimensional version of the optimal finite-length solution and applies it to images with missing pixels, [8]. Since the eye is not an ideal LPF, Russell's algorithm does not perfectly compensate for the missing pixels, but there is a noticeable improvement in perceived image quality. Though such extensions are not considered, the algorithms presented in this thesis can also be extended to two-dimensions for use in missing pixel compensation.

1.4 Thesis Outline

Chapter 2 extends Russell's constrained minimization approach and presents a closed-form solution, referred to as the Constrained Minimization (CM) algorithm. Results from numerical simulation are presented. Despite giving the optimal solution, CM is shown to have numerical stability problem for certain parameters.

In Chapter 3, we develop a different approach to the compensation problem by windowing the ideal infinite-length solution, $c_{\text{inf}}[n]$. The solution is related to the discrete prolate spheroidal sequences (DPSS), a class of optimally band-limited signals. We develop a DPSS-based compensation algorithm, called the discrete prolate approximation (DPAX). The DPAX solution is shown to be sub-optimal, a first-order approximation to the optimal CM solution. However, as we show, DPAX is more numerically stable than CM. As in Chapter 2, results from numerical simulation are presented. In addition, Appendix A presents an interpretation of the DPSS as a singular value decomposition. Appendix B proves duality of certain DPSS.

Chapter 4 presents an iterative algorithm in the class of projection-onto-convex-sets (POCS). The Iterative Minimization (IM) algorithm is proved to converge uniquely to the optimal CM solution. Results from numerical simulation are used to show that the IM algorithm has a slow convergence rate, a common problem for many POCS algorithms.

Chapter 2

Constrained Minimization

In this chapter, we develop constrained minimization as a technique for generating compensation signals. Extending the derivations in [8], we derive a closed-form expression for the optimal, finite-length solution called the Constrained Minimization (CM) algorithm. We evaluate the CM algorithm's performance through numerical simulation. Although optimal, CM is shown to have problems of numerical stability.

2.1 Derivation

We closely follow the derivation in [8]. However, our treatment formalizes the constrained minimization using Lagrange multipliers. We assume that the compensation signal, $c[n]$, is non-zero only for $n \in \mathcal{N}$, where $\mathcal{N}$ is the finite set of points to be adjusted. For simplicity in the presentation, we assume a symmetric form for $c[n]$, i.e. $\mathcal{N}= [-\frac{N-1}{2}, \frac{N-1}{2}]$, although the derivation is general for any set $\mathcal{N}$. Also, for notational convenience, the signals $c[n]$ and $x[n]$ are denoted as the sequences $\{c_n\}$ and $\{x_n\}$. As shown in Chapter 1,

$$e[n] = \sum_{m \in \mathcal{N}} c_m h[n-m] \tag{2.1}$$

The error is

$$\varepsilon^2 = \sum_{n=-\infty}^{\infty} |e[n]|^2 = \sum_{n=-\infty}^{\infty} \left(\sum_{m \in \mathcal{N}} c_m h[n-m] \right)^2 \tag{2.2}$$

Our desire is to minimize ε^2 subject to the constraint $g = c_0 + x_0 = 0$. We do this using the method of Lagrange multipliers. Defining $h = \varepsilon^2 - \lambda g$, we minimize h by setting the

partial derivatives with respect to c_k for $k \in \mathcal{N}$ equal to zero. For $c_k \neq c_0$,

$$\begin{aligned}
\frac{\partial}{\partial c_k} h &= \sum_{n=-\infty}^{\infty} 2\Big(\sum_{m\in\mathcal{N}} c_m h[n-m]\Big) h[n-k] = 0 && (2.3)\\
&= 2\sum_{m\in\mathcal{N}} c_m \Big(\sum_{n=-\infty}^{\infty} h[n-m]h[n-k]\Big) && (2.4)\\
&= \sum_{m\in\mathcal{N}} c_m \Theta_\gamma[k-m] = 0 && (2.5)
\end{aligned}$$

$\Theta_\gamma[n]$ is the deterministic autocorrelation function of the filter $h[n]$. The subscript γ denotes the cutoff of the LPF $h[n]$. Simplifying $\Theta_\gamma[n]$, we obtain

$$\begin{aligned}
\Theta_\gamma[n] &= \sum_{m=-\infty}^{\infty} h[m]h[n-m] && (2.6)\\
&= h[n] * h[-n] && (2.7)\\
&= \frac{\sin(\gamma n)}{\pi n} * \frac{-\sin(-\gamma n)}{\pi n} && (2.8)\\
&= \frac{\sin(\gamma n)}{\pi n} = h[n] && (2.9)
\end{aligned}$$

where $*$ denotes convolution. For clarity, we do not replace $\Theta_\gamma[n]$ with $h[n]$, despite the fact that they are equivalent. When $c_k = c_0$, the derivative has an extra term with λ, the Lagrange multiplier.

$$\begin{aligned}
\frac{\partial}{\partial c_0} h &= \sum_{m=-\infty}^{\infty} \left(\sum_{n\in\mathcal{N}} c_n h[m-n]\right)^2 - \lambda(c_0 + x_0)\Big) = 0 && (2.10)\\
&= 2\sum_{n\in\mathcal{N}} c_n \Theta_\gamma[-n] - \lambda = 0 && (2.11)
\end{aligned}$$

These derivatives produce N equations. Along with the constraint g, this results in $N+1$ equations for the N unknown sequence values of c_n and the one Lagrange multiplier λ. The system, reproduced below, has a unique solution, $c_{\text{opt}}[n]$, that is the optimal compensation signal for the given value of N.

$$\begin{aligned}
\sum_{m\in\mathcal{N}} c_m \Theta_\gamma[k-m] &= 0, \quad k \neq 0 && (2.12)\\
2\sum_{m\in\mathcal{N}} c_m \Theta_\gamma[-m] - \lambda &= 0, \quad k = 0 && (2.13)\\
c_0 - x_0 &= 0 && (2.14)
\end{aligned}$$

We chose $\mathcal{N}= [-\frac{N-1}{2}, \frac{N-1}{2}]$ to be symmetric, so the system (2.12), (2.13), (2.14) can be written in block matrix form

$$\begin{bmatrix} \mathbf{\Theta}_\gamma & -\frac{1}{2}\delta \\ \delta^{\mathbf{T}} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{c_n} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ -x_0 \end{bmatrix} \tag{2.15}$$

where δ is a vector with all zero entries except for a 1 as the center element, i.e.

$$\delta^{\mathbf{T}} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \tag{2.16}$$

$\mathbf{\Theta}_\gamma$ is the autocorrelation matrix for $h[n]$. Because $h[n]$ is a ideal low-pass filter, $\mathbf{\Theta}_\gamma$ is a symmetric, Toeplitz matrix with entries

$$\mathbf{\Theta}_\gamma(i,j) = h[i-j] = \frac{\sin \gamma[i-j]}{\pi[i-j]} \tag{2.17}$$

The symmetric, Toeplitz structure of $\mathbf{\Theta}_\gamma$ is particularly important in Chapters 3 and 4 in relation to the discrete prolate spheroidal sequences and the iterative-minimization algorithm.

To solve (2.15), the matrix $\mathbf{\Theta}_\gamma$ must be inverted. We leave a proof of invertibility for Chapter 5. For now, assuming that the inverse exists, (2.15) can be interpreted as two equations

$$\mathbf{\Theta}_\gamma \mathbf{c_n} + -\frac{1}{2}\lambda\delta = \mathbf{0} \tag{2.18}$$

$$\delta^{\mathbf{T}} \mathbf{c_n} + 0\lambda = -x_0 \tag{2.19}$$

Combining the two equations,

$$\left(\frac{1}{2}\delta^{\mathbf{T}} \mathbf{\Theta}_\gamma^{-\mathbf{1}} \delta\right) \lambda = -x_0 \tag{2.20}$$

$\delta^{\mathbf{T}} \mathbf{\Theta}_\gamma^{-\mathbf{1}} \delta = \mathbf{\Theta}_\gamma^{-\mathbf{1}} \left(\frac{(N-1)}{2}, \frac{(N-1)}{2}\right)$ is the center element of the inverse matrix. For notational convenience, we choose $\theta_c^{-1} = \mathbf{\Theta}_\gamma^{-\mathbf{1}} \left(\frac{(N-1)}{2}, \frac{(N-1)}{2}\right)$. The Lagrange multiplier is then

$$\lambda = -\frac{2x_0}{\theta_c^{-1}} \tag{2.21}$$

The optimal compensation signal is

$$c_{\text{opt}}[n] = -\frac{x_0}{\theta_c^{-1}}\mathbf{\Theta}_\gamma^{-\mathbf{1}}\delta \tag{2.22}$$

We refer to the algorithm represented by (2.22) as Constrained Minimization (CM).

2.2 Performance Analysis

2.2.1 Examples

We implemented the CM algorithm in MATLAB and calculated compensation signals for examples in which $x[0] = -1$. Figure 2-1 shows $c_{\text{opt}}[n]$ and an interpolated DFT, $C_{\text{opt}}(e^{j\omega})$, using 2048 linearly-interpolated points, for $N = 6$, $N = 10$, and $N = 20$ for $\gamma = 0.9\pi$. Figure 2-2 shows the same for $\gamma = 0.7\pi$.

There are several interesting features to note on these examples. The optimal signal is high-pass as expected. For both cases, the main lobe of the DFT is centered at π with smaller side-lobes in the interval $[-\gamma, \gamma]$. Furthermore, as N increases, the energy in this low-pass band decreases, thus decreasing the error. Also, we can see that for the same N, the solution for $\gamma = 0.7\pi$ does better than that of $\gamma = 0.9\pi$ because there is a larger high-pass band. Intuitively, for smaller γ, the system is more oversampled, so a better solution can be found using fewer samples.

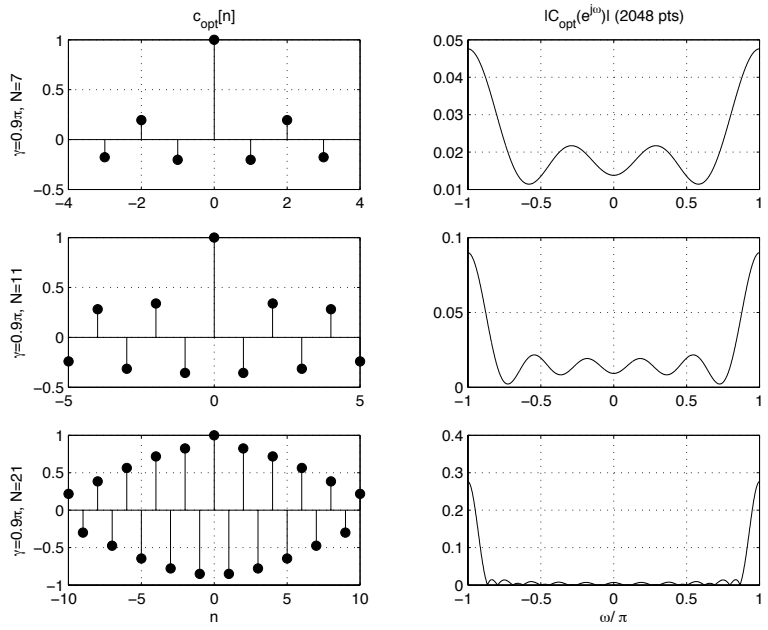

Figure 2-1: Optimal sequences $c[n]$ for $\gamma = 0.9\pi$ and N=7, 11, 21

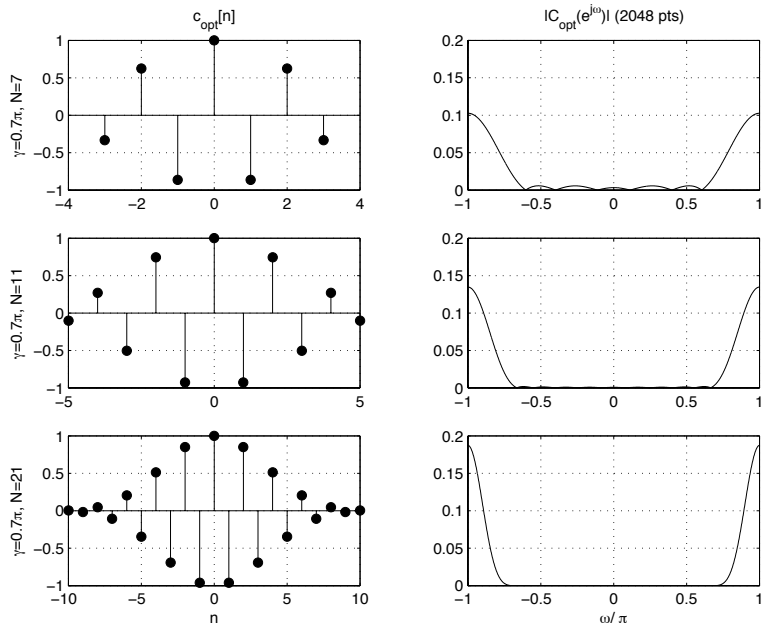

Figure 2-2: Optimal sequences $c[n]$ for $\gamma = 0.7\pi$ and N=7, 11, 21

2.2.2 Error Performance

Figure 2-3 illustrates ε^2 as a function of N for $\gamma = 0.1\pi$, 0.3π, 0.5π, 0.7π, and 0.9π. The graph shows that ε^2 decreases approximately exponentially in N. Since the CM algorithm generates the optimal solution, the error curves shown in Figure 2-3 serves as a baseline for performance. There is a limited set of parameters γ and N for which the problem is well-conditioned. Beyond $\varepsilon^2 = 10^{-9}$, the solution becomes numerically unstable beyond the precision of MATLAB.

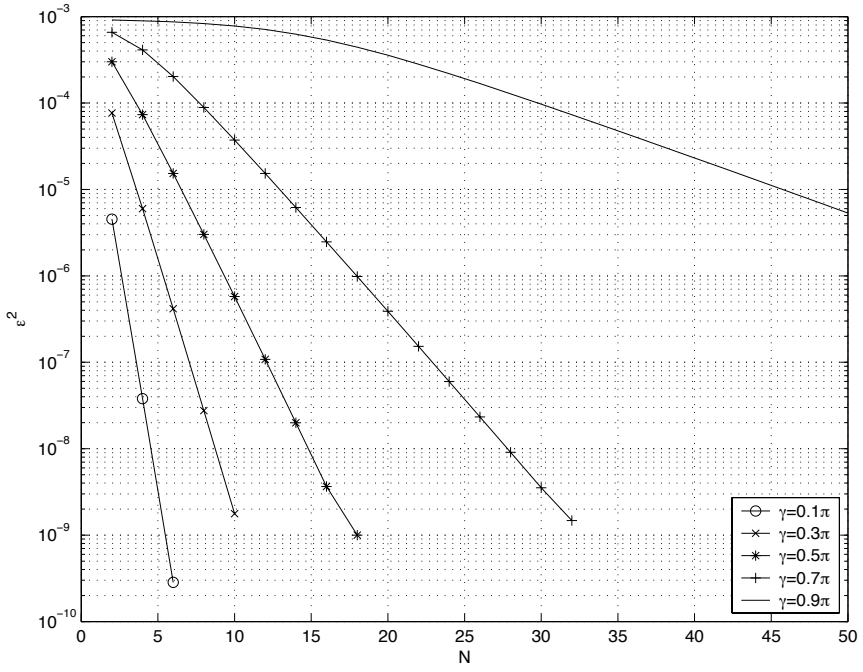

Figure 2-3: ε^2 as a function of N

2.2.3 Computational Complexity

With a direct implementation, using Gaussian elimination, the $N \times N$ inversion of $\mathbf{\Theta}_\gamma$ requires $O(N^3)$ multiplications, and $O(N^2)$ memory, [13]. Exploiting the Toeplitz, symmetric structure, we can use a Levinson recursion to find the solution. This does better, using $O(N^2)$ multiplications and $O(N)$ space, [7].

2.3 Numerical Stability

As N increases and γ decreases, the condition number of $\mathbf{\Theta}_\gamma$ increases until inversion becomes unstable. Figure 2-4 plots the condition number of $\mathbf{\Theta}_\gamma$ as a function of N for various values of γ. The plot shows that the condition number increases approximately exponentially. As γ decreases the condition number also increases approximately exponentially.

Figure 2-5 shows the condition number as a function in the two-dimensional (N, γ) plane. The black region in the lower right-hand corner is where the condition number is reasonable, and the solution is numerically stable. The large semi-circular gray region centered at the upper left-hand corner is where MATLAB returns inaccurate solutions because the problem is too ill-conditioned. The whitish crescent region defines a meta-stable boundary between well-conditioned and ill-conditioned solutions. MATLAB can find accurate solutions for some points in this region, and not for others.

There is only a small region in the plane where the problem is well-conditioned enough for a floating-point implementation of MATLAB to find a solution. Conditioning problems would be even more pronounced in fixed-point DSP systems. Fortunately, the inversion can be done off line on a computer with arbitrarily high precision, since once $c_{\text{opt}}[n]$ is found it can be stored and retrieved when the algorithm needs to be implemented. Also, in most contexts, an error of $10^{-9} = -180\text{dB}$, compared to the signal, is more than sufficient.

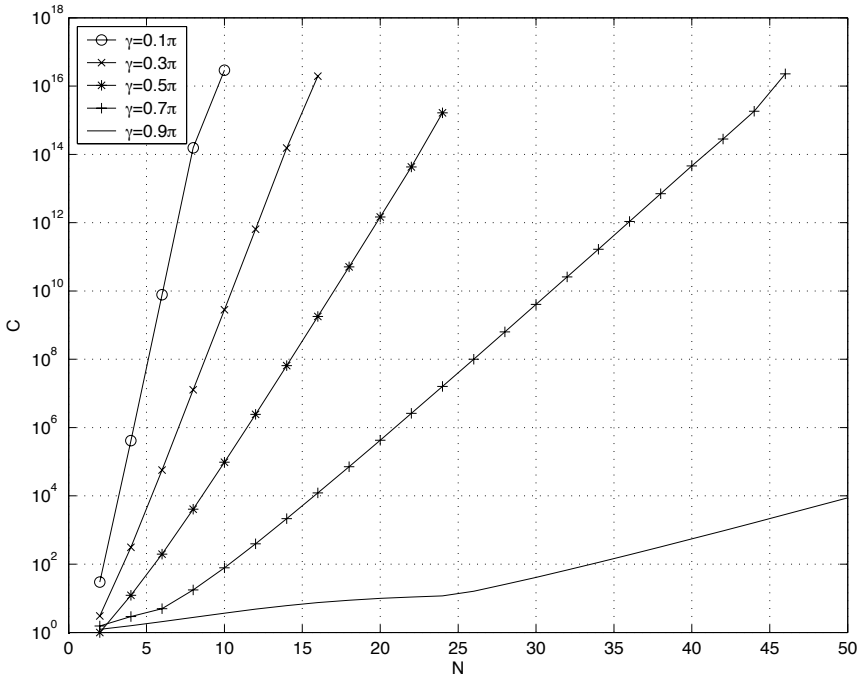

Figure 2-4: Condition Number as function of N for $\gamma = 0.1\pi, 0.3\pi, 0.5\pi, 0.7\pi, 0.9\pi$

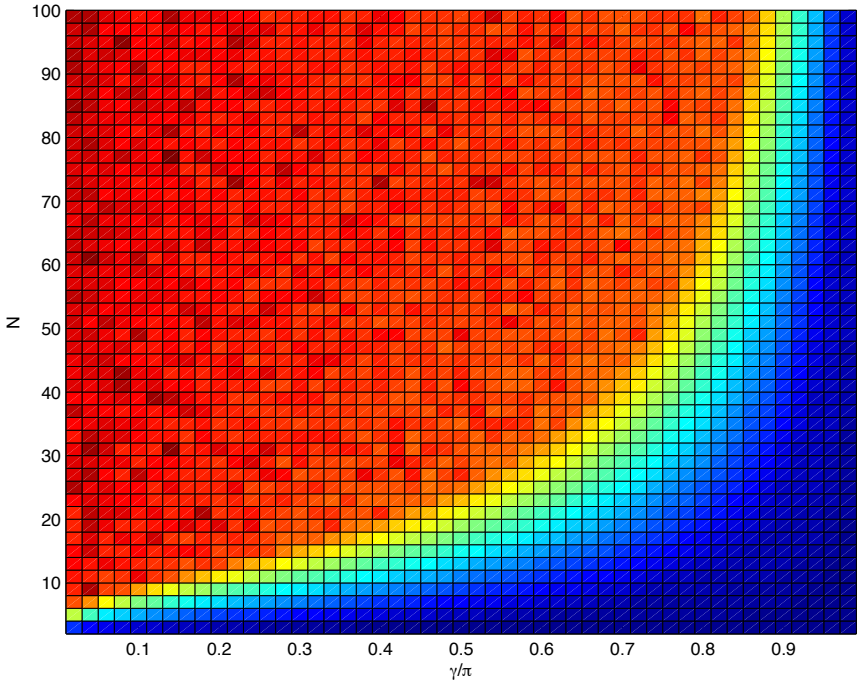

Figure 2-5: Condition Number as function of N and γ

Chapter 3

Discrete Prolate Approximation

In this chapter, we formulate an alternate approach to the design of finite-length compensation signals based on applying a finite-length window to the infinite length solution, $c_{\text{inf}}[n]$. The problem formulation leads us to consider a class of optimally band-limited signals, called discrete prolate spheroidal sequences (DPSS), as windows. An algorithm, which we refer to as the Discrete Prolate Approximation (DPAX) is presented. As the name implies, this solution is an approximation to the optimal CM solution using discrete prolate spheroidal sequences. We explore the relationship between DPAX and the CM solution, concluding that the DPAX solution is a first-order approximation of CM in its eigenvector basis. In addition, we evaluate the algorithms performance through numerical simulation. DPAX is shown to be more numerically stable than CM.

3.1 Derivation

3.1.1 Windowing and Discrete Prolate Spheroidal Sequences

With CM, we construct a finite-length compensation signal directly from the imposed constraints. Alternatively, we can start with the infinite-length signal, $c_{\text{inf}}[n] = -x[0](-1)^n$, and truncate it through appropriate windowing. From this perspective, the problem then becomes one of designing a finite-length window, $w[n]$, such that

$$c[n] = w[n]c_{\text{inf}}[n] \tag{3.1}$$

has minimum energy in the frequency band $|\omega| < \gamma$. For the symmetric case, we pose the

window design as an optimization problem. $C(e^{j\omega}) = W(e^{j\omega}) * C_{\inf}(e^{j\omega})$, so we design $w[n] \in \ell_2(-\frac{N-1}{2}, \frac{N-1}{2})$, to maximize the concentration ratio

$$\alpha(N,W) = \frac{\int_{-W=-\pi+\gamma}^{W=\pi-\gamma} |W(e^{j\omega})|^2 d\omega}{\int_{-\pi}^{\pi} |W(e^{j\omega})|^2 d\omega} \tag{3.2}$$

Slepian, Landau, and Pollak solve this problem in [9, 5, 10] through the development of discrete prolate spheroidal sequences (DPSS). Using variational methods they show that the sequence $w[n]$ that maximizes the concentration satisfies the equation

$$\sum_{m=-\frac{N-1}{2}}^{\frac{N-1}{2}} \frac{\sin 2W\pi[n-m]}{\pi[n-m]} w[m] = \lambda w[n] \tag{3.3}$$

Expressed in matrix form, the solutions to (3.3) are eigenvectors of the $N \times N$ symmetric, positive-definite, Toeplitz matrix, $\mathbf{\Theta_W}$, with elements

$$\mathbf{\Theta_W}[n,m] = \frac{\sin 2W(m-n)}{\pi(m-n)} \tag{3.4}$$

$$m, n = -(N-1)/2, ... -1, 0, 1, ...(N-1)/2$$

If $W = \gamma$, we obtain $\mathbf{\Theta}_\gamma$, the same matrix as in Section 3. By the spectral theorem, the eigenvectors, $v_i^W[n]$, are real and orthogonal with associated real, positive eigenvalues, λ_i^W. In addition, in [10] it is shown that these particular eigenvalues are always distinct and can be ordered

$$\lambda_1 > \lambda_2 > ... > \lambda_N$$

These eigenvectors, $v_i^W[n]$, are time-limited versions of discrete prolate spheroidal sequences (DPSS). They form a finite orthonormal basis for $\ell_2(-\frac{N-1}{2}, \frac{N-1}{2})$, [10].

Note that the DPSS are parametrized by N and W. In cases where there may be confusion, the DPSS are denoted as $v_i^{(N,W)}[n]$. This specifies the i-th DPSS for the interval $|n| < \frac{N-1}{2}$ and the band $|\omega| < W$.

The original concentration problem only requires that (3.3) hold for $|n| < \frac{N-1}{2}$. By low-pass filtering the time-limited DPSS $v^{(i)}[n]$, we can extend the range of definition and

thus define the full DPSS

$$u_i[n] = \sum_{m=-\infty}^{\infty} \frac{\sin 2W\pi[m-n]}{\pi[m-n]} v_i[m] \tag{3.5}$$

The $u_i[n]$ are defined for all $n = (-\infty, \infty)$. They are band-limited functions that can be shown to be orthogonal on $(-\infty, \infty)$ as well as on $[-\frac{N-1}{2}, \frac{N-1}{2}]$, [10]. They form an orthonormal basis for the set of band-limited functions $\ell_2(-W, W)$, [10]. References [10, 9, 5] all comment on this "remarkable" double-orthogonality of the DPSS and its practical importance in applications. In Appendix A we show that this double-orthogonality, and many other properties of the DPSS can be interpreted as the result of the DPSS being a singular value decomposition (SVD) basis.

3.1.2 Extremal Properties of the DPSS

Finally, normalizing the full DPSS to unit energy

$$\sum_{n=-\infty}^{\infty} u_i[n]u_j[n] = \delta_{ij} \tag{3.6}$$

It follows that

$$\sum_{n=-N/2}^{N/2} u_i[n]u_j[n] = \lambda_i\delta_{ij} \tag{3.7}$$

So the eigenvalue, λ_i, is the fraction of energy that lies in the interval $[-\frac{N-1}{2}, \frac{N-1}{2}]$, [10]. Conversely, normalizing the time-limited DPSS to unit energy

$$\sum_{n=-N/2}^{N/2} v_i[n]v_j[n] = \delta_{ij} \tag{3.8}$$

It follows that

$$\int_{-W}^{W} V_i(e^{j\omega})V_j^*(e^{j\omega})d\omega = \lambda_i\delta_{ij} \tag{3.9}$$

The eigenvalue, λ_i, in this case, is the fraction of energy in the band $|\omega| < W$, [10]. We can now use the DPSS orthonormal basis to find the answer to our concentration problem. The signal $w[n] \in \ell_2$, so it has some finite energy E. Expressed in the DPSS basis

$$w[n] = \alpha_1 v_1[n] + \alpha_2 v_2[n] + \cdots + \alpha_N v_N[n] \tag{3.10}$$

with the constraint

$$\alpha_1^2 + \alpha_2^2 + \cdots + \alpha_N^2 = E \tag{3.11}$$

Our goal is to choose the α_i such that the energy in the band $|\omega| < \pi - \gamma$ is maximized. By the orthogonality of $\{v_i[n]\}$, the energy of $w[n]$ in $|\omega| < \pi - \gamma$ is

$$\alpha_1^2\lambda_1 + \alpha_2^2\lambda_2 + \cdots + \alpha_N^2\lambda_N \tag{3.12}$$

Since there is a constraint on E, in order to maximize the ratio of energy in-band all of the energy should be put onto the first eigenvector, i.e. $\alpha_1 = \sqrt{E}$ and $\alpha_i = 0$ for $i \neq 1$. Thus the first DPSS, $v_1[n]$, solves the concentration problem. The maximum concentration is the eigenvalue, λ_1. Consequently, the optimal window in our formulation is $v_1^{\pi-\gamma}[n]$. Modulating $v_1^{\pi-\gamma}[n]$ up to π and scaling it to meet the constraint, $c[0] = -x[0]$, provides a potential compensation signal

$$c_{\text{dpax}}[n] = -\frac{x[0]}{v_1^{\pi-\gamma}[0]}(-1)^n v_1^{\pi-\gamma}[n] \tag{3.13}$$

3.1.3 Duality of DPSS

Every DPSS has a dual symmetric partner. In particular,

$$v_{N+1-i}^{W}[n] = (-1)^n v_i^{\pi-W}[n] \tag{3.14}$$

The eigenvalues are related

$$\lambda_{N+1-i}^{W} = \lambda_i^{\pi-W} \tag{3.15}$$

[10] states this property without a detailed proof. We provide a comprehensive proof in Appendix B. Duality implies that the compensation signal, $c_{\text{dpax}}[n]$ in (3.13), can also be expressed

$$c_{\text{dpax}}[n] = -\frac{x[0]}{v_N^{\gamma}[0]}v_N^{\gamma}[n] \tag{3.16}$$

Independent of which DPSS is used to express it, we refer to this solution as the Discrete Prolate Approximation (DPAX). The algorithm above is specific for symmetric compensation. For asymmetric compensation, the solution is also the first DPSS, scaled relative to the dropped sample, $v_N^{\gamma}[k]$, for $k \neq 0$.

3.2 Relationship to Constrained Minimization Solution

It should be clear that $c_{\text{dpax}}[n]$ is not equivalent to the CM solution, $c_{\text{opt}}[n]$. In this section, we illustrate how the window formulation constrains the problem differently, leading to a sub-optimal solution. We also show that the optimal solution is a linear combination of the DPSS. In this context, the DPAX solution is a first-order approximation of the optimal solution.

3.2.1 Different Constraints

The window formulation starts with a finite-energy signal, optimizes for that energy, and then scales to meet the $c[0] = -x[0]$ constraint. CM does not begin with an energy constraint, thus it finds the optimal solution. Specifically, in the CM algorithm, we have a finite set $\mathcal{N}$ on which we design N samples of the signal $c[n]$. Assume $\mathcal{N}= \{0, 1\}$. Two sample values, $c[0]$ and $c[1]$, must be determined. Graphically, as illustrated in Figure 3-1, there exists the $(c[0], c[1])$ plane on which the error ε^2 is defined. The constraint, $c[0] = -x[0]$, defines a vertical line in the space. The CM algorithm finds the unique point on this line that minimizes ε^2. This point, $c_{\text{opt}}[n]$, is the optimal solution given the constraints.

The DPAX solution is fundamentally different. $v_N^\gamma[n]$, is a signal of unit energy that has minimum error, ε^2. As illustrated in Figure 3-2, it is the minimum on a circle of radius $E = 1$ in the $(c[0], c[1])$ plane. The DPAX algorithm scales $v_N^\gamma[n]$ to meet the constraint $c[0] = -x[0]$. Graphically, this amounts to scaling the unit circle until the minimum error point intersects the constraint line $c[0] = -x[0]$. This point, which is both on the scaled circle and the line, is $c_{\text{dpax}}[n]$. This point is not the same as $c_{\text{opt}}[n]$. The DPAX solution is thus sub-optimal for the constraints on the problem.

3.2.2 Constrained Mimimization Solution in the DPSS Basis

The exact relationship between $c_{\text{dpax}}[n]$ and $c_{\text{opt}}[n]$ can be found by decomposing $c_{\text{opt}}[n]$ in the DPSS basis, $\{v_i^\gamma[n]\}$. Since $\mathbf{\Theta}_\gamma$ is real and symmetric, it can be diagonalized into its real, orthonormal eigenvector basis which are the time-limited DPSS.

$$\mathbf{\Theta}_\gamma = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\mathbf{T} \tag{3.17}$$

$\mathbf{V}$ is a orthogonal matrix with columns that are the DPSS. $\mathbf{\Lambda}$ is a diagonal matrix of

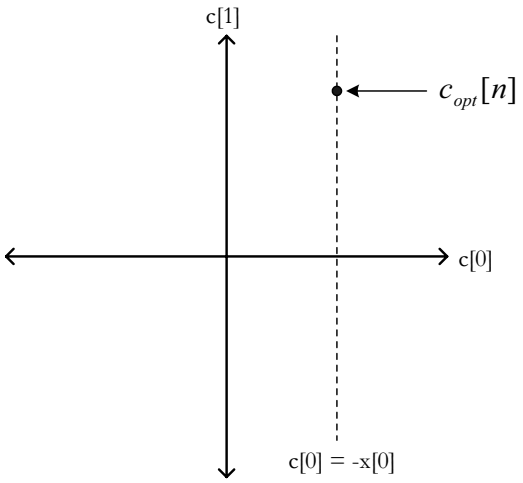

Figure 3-1: Constrained Minimization, $N = 2$

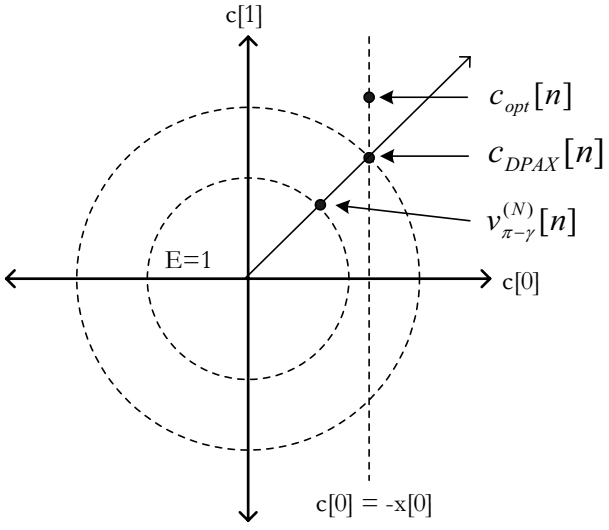

Figure 3-2: Discrete Prolate Approximation, $N = 2$

eigenvalues. The inverse, $\mathbf{\Theta_\gamma^{-1}}$ is also symmetric and can be diagonalized in the same basis.

$$\mathbf{\Theta_\gamma^{-1}} = \mathbf{V\Lambda^{-1}V^T} \tag{3.18}$$

As proved in [10], the eigenvalues associated with the DPSS are all real, positive, and distinct. Thus, $\mathbf{\Theta}_\gamma$ can be diagonalized with non-zero, distinct eigenvalues. The matrix is thus non-singular and can be inverted. $c_{\text{opt}}[n]$ exists, and it can be expressed as

$$c_{\text{opt}}[n] = -\frac{x_0}{\theta_c^{-1}}\mathbf{\Theta_\gamma^{-1}}\delta = -\frac{x_0}{\theta_c^{-1}}\mathbf{V\Lambda^{-1}V^T}\delta \tag{3.19}$$

θ_c^{-1} is the middle element of $\mathbf{\Theta_\gamma^{-1}}$, which can be expressed in the DPSS basis as

$$\theta_c^{-1} = \sum_{i=1}^{N} \lambda_i^{-1}(v_i^\gamma[0])^2 \tag{3.20}$$

Without matrices, $c_{\text{opt}}[n]$ can be expressed as

$$c_{\text{opt}}[n] = -\frac{x[0]}{\theta_c^{-1}}\left(\lambda_1^{-1}\beta_1 v_1^\gamma[n] + \cdots + \lambda_N^{-1}\beta_N v_N^\gamma[n]\right) \tag{3.21}$$

where $\beta_i = v_i^\gamma[0]$. The eigenvalues, λ_i, are distributed between 0 and 1. The expression for the optimal solution depends on the reciprocals $1/\lambda_i$, so the eigenvector with the smallest eigenvalue, $v_N^\gamma[n]$, will dominate. Since scaling this vector produces $c_{\text{dpax}}[n]$, DPAX can be interpreted as a first-order approximation to $c_{\text{opt}}[n]$.

3.3 Performance Analysis

3.3.1 Examples

We implemented the DPAX algorithm in MATLAB and calculated compensation signals in which $x[0] = -1$. The DPSS were found using the `dpss()` function in the MATLAB Signal Processing Toolbox. This function computes the eigenvectors of $\mathbf{\Theta}_\gamma$ using a similar, tri-diagonal matrix ρ_γ. The next section, describes the particulars of the implementation. Figure 3-3 shows $c_{\text{dpax}}[n]$ and an interpolated DFT, $C_{\text{dpax}}(e^{j\omega})$, using 2048 linearly-interpolated points, for $N = 7$, $N = 11$, and $N = 21$ for $\gamma = 0.9\pi$. Figure 3-4 shows the same for $\gamma = 0.7\pi$.

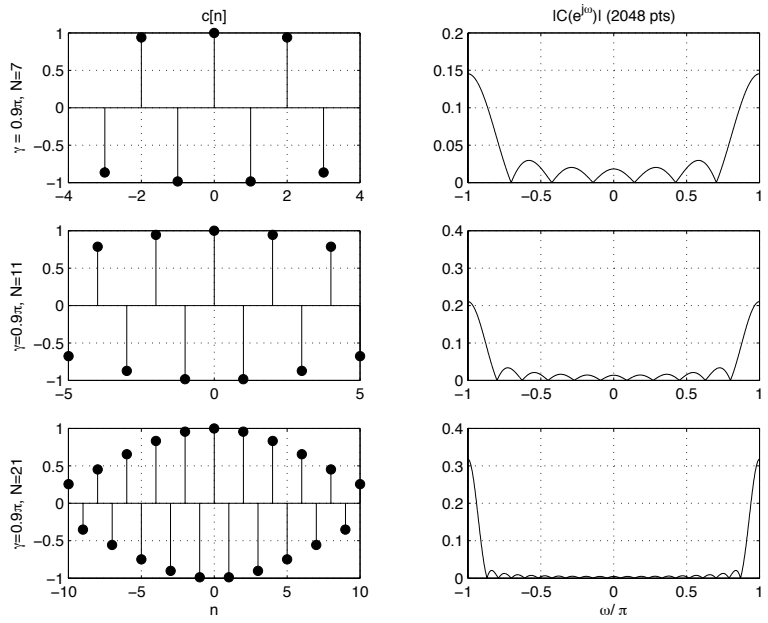

Figure 3-3: DPAX solution $c[n]$ for $\gamma = 0.9\pi$ and N=7, 11, 21

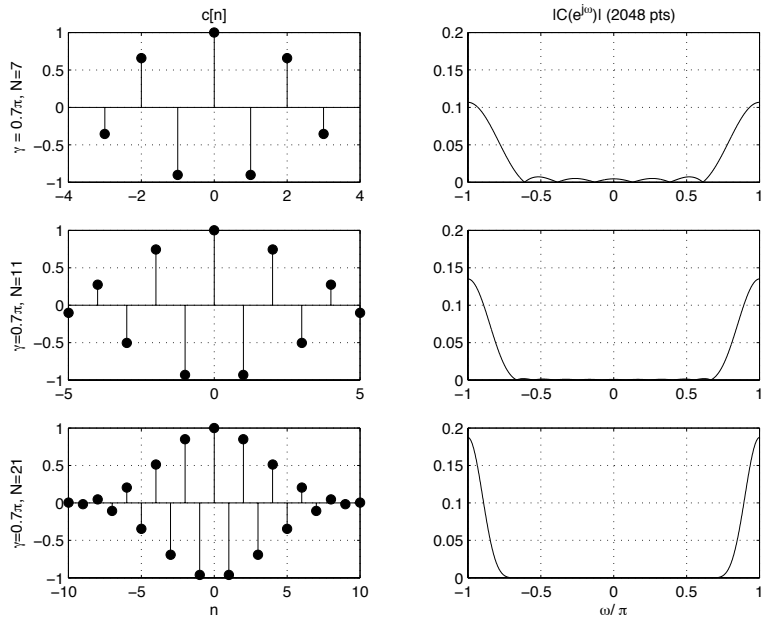

Figure 3-4: DPAX solution $c[n]$ for $\gamma = 0.7\pi$ and N=7, 11, 21

Comparing these plots to Figure 2-1 and Figure 2-2 in Chapter 2, we observe that $c_{\text{dpax}}[n]$ looks similar to $c_{\text{opt}}[n]$. Like $c_{\text{opt}}[n]$, $c_{\text{dpax}}[n]$ is harder to band-limit when there is only a small high-pass band. One noticeable difference is that the zeroth sample is not discontinuous from the rest of the samples like in $c_{\text{opt}}[n]$. All of the samples for $c_{\text{dpax}}[n]$ can be connected by a smooth continuous envelope.

3.3.2 Error performance

Figure 3-5 illustrates ε^2 as a function of N for various values of γ. DPAX is not as ill-conditioned as CM, so by increasing N it can achieve values of ε^2 in the range of $\varepsilon^2 = 10^{-20}$, i.e. about ten orders of magnitude smaller than that using CM.

The DPAX solution is suboptimal compared to the CM solution for the same parameters γ and N. Figure 3.22 plots the gain

$$G_\varepsilon = \frac{\varepsilon^2_{\text{dpss}}}{\varepsilon^2_{\text{opt}}} \tag{3.22}$$

in error due to DPAX. As the figure illustrates, the gain becomes negligible as N increases and γ decreases. The near-optimal performance of the DPAX solution is explained by the eigenvalue distribution of $\mathbf{\Theta}_\gamma$. As N increases and γ decreases, the reciprocal of the smallest eigenvalue, $1/\lambda_N$, increasingly dominates the reciprocals of the other eigenvalues. In (3.21), $v^\gamma_N[n]$ dominates the other terms, making $c_{\text{dpax}}[n]$ a tighter approximation.

Figure 3-7 shows the ratio λ_{N-1}/λ_N as a function of γ for $N = 7$, $N = 11$, and $N = 21$. It illustrates empirically that λ_{N-1} becomes significantly larger than λ_N as N increases and γ decreases. The other eigenvalues λ_1, λ_2, etc., are even larger in size. For example, when $N = 21$, λ_{N-1} is 250 times larger than λ_N at $\gamma = 0.5\pi$. Thus in (3.21) the $v^{(N)}_\gamma[n]$ term is 250 times more significant than the next order term.

3.3.3 Computational Complexity

The DPAX solution can be computed directly as the last eigenvector of $\mathbf{\Theta}_\gamma$. Due to the ill-conditioning of $\mathbf{\Theta}_\gamma$, the DPAX solution is better found as the last eigenvector of a similar matrix, ρ_γ. In either case, DPAX requires finding the last eigenvector of a $N \times N$ matrix. Many numerical routines exists for the computation of eigenvectors. For small matrices ($N < 45$), the standard LAPACK routines, like those used by MATLAB's `eig()` function,

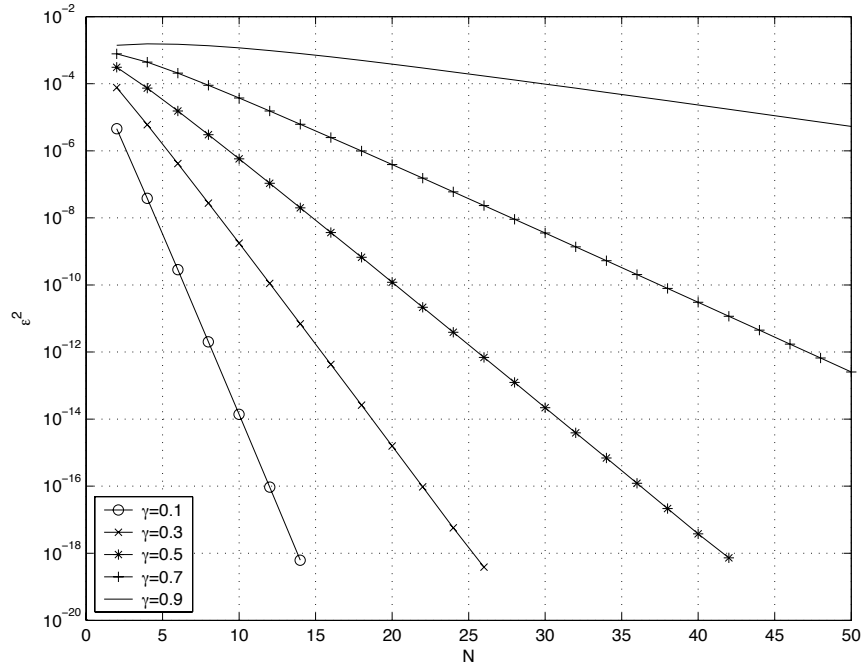

Figure 3-5: ε^2 as a function of N

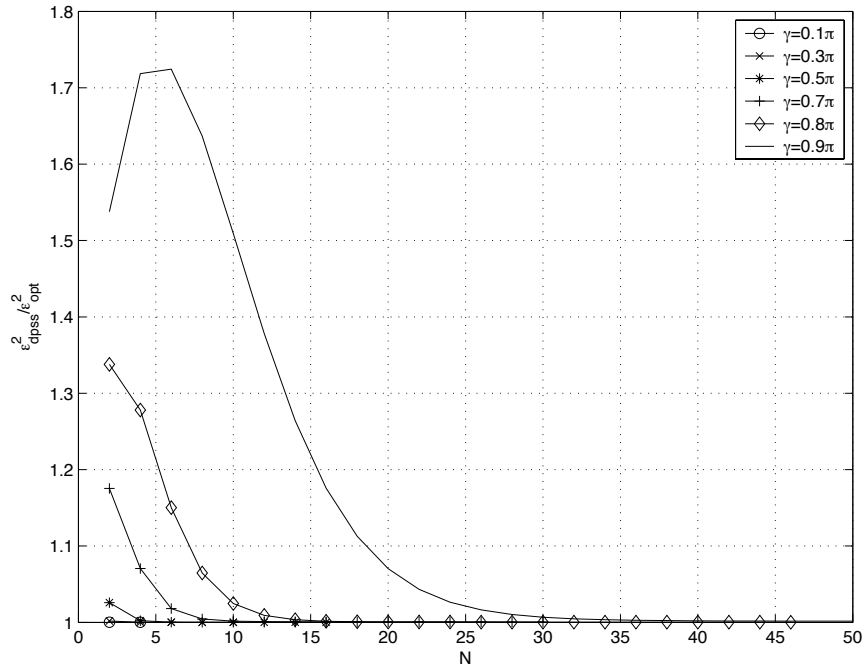

Figure 3-6: $G_\varepsilon = \varepsilon^2_{\text{dpss}}/\varepsilon^2_{\text{opt}}$ as a function of N

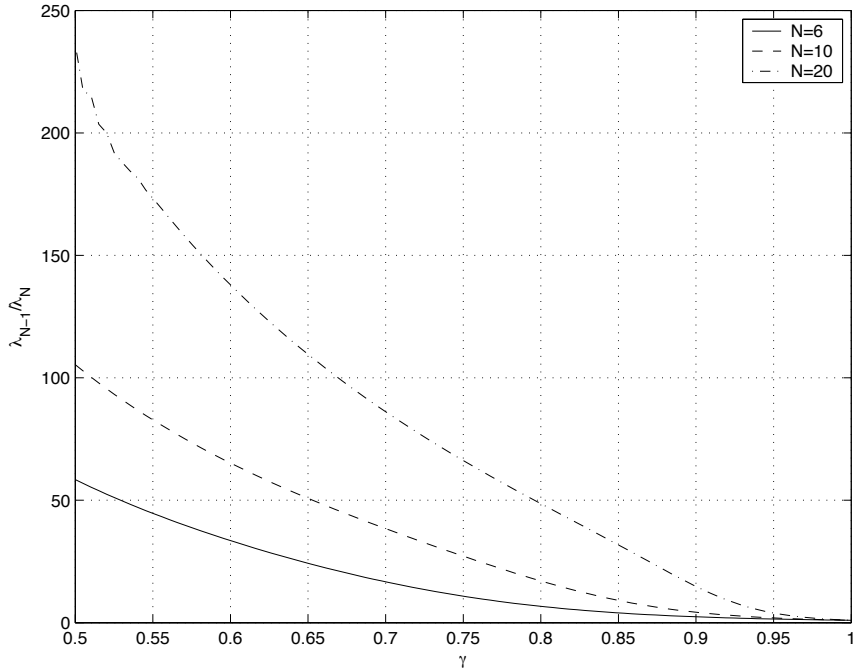

Figure 3-7: Ratio of two smallest eigenvalues: $\frac{\lambda_{N-1}}{\lambda_N}$

find all the eigenvectors of a matrix in $O(N^2)$ time, [14].

The DPAX solution only requires calculation of one eigenvector though. As N increases, finding all the eigenvectors of a matrix is wasteful in terms of time and computation. For such cases, the power method for calculating the first eigenvector of a symmetric matrix is well suited for this problem [14]. In particular, the power method can be applied to $\mathbf{\Theta}_{\pi-\gamma}$, the dual matrix, to find the first eigenvector. By dual symmetry the first DPSS of $\mathbf{\Theta}_{\pi-\gamma}$ multiplied by $(-1)^n$ is equivalent to the last DPSS of $\mathbf{\Theta}_\gamma$. Using the power method, the DPAX algorithm can be implemented more efficiently for larger N.

3.4 Numerical Stability

The ill-conditioning of $\mathbf{\Theta}_\gamma$ can be better understood by looking at its eigenstructure. The eigenvalues of $\mathbf{\Theta}_\gamma$ correspond to the energy ratio of each of DPSS in band, thus they have values between 0 and 1. Although these eigenvalues can be proved to be distinct, they are usually so clustered around 0 or 1 that they are effectively degenerate when finite machine arithmetic is used. Figure 3-8 shows the distribution of the eigenvalues for $N = 11$ as a function of γ.

This degeneracy is the cause of the ill-conditioning of $\mathbf{\Theta}_\gamma$. This was a problem for the CM algorithm, and is also a problem for the DPAX algorithm. Fortunately, the time-limited DPSS are also solutions of the second order difference equation

$$\frac{1}{2}n(N-n)v_i[n-1]+[(\frac{N-1}{2}-n)^2\cos 2\pi W-\chi_i]v_i[n]+\frac{1}{2}(n+1)(N-1-n)v_i[n+1]=0 \quad (3.23)$$

$$k,n=-\frac{N-1}{2},...,-1,0,1,...\frac{N-1}{2}$$

In matrix form, the solutions to (3.23) are the normalized eigenvectors of a symmetric, tridiagonal matrix ρ_γ of the form

$$\rho_{\pi-\gamma}[i,j]=\begin{cases}\frac{1}{2}i(N-i) & j=i-1\\ \left(\frac{N-1}{2}-i\right)^2\cos 2(\pi-\gamma) & j=i\\ \frac{1}{2}(i+1)(N-1-i) & j=i+1\\ 0 & \text{otherwise}\end{cases} \quad (3.24)$$

$$i,j=-\frac{N-1}{2},...-1,0,1,...\frac{N-1}{2}$$

The χ_i's are the eigenvalues. This equation arises in quantum mechanics when trying to separate the three-dimensional scalar wave equation on a prolate spheroidal coordinate system. This is where the name "prolate spheroidal sequences" originated, [9, 10].

[10] shows that if the eigenvectors of ρ_γ are sorted according to their respective eigenvalues, then they are the same as the ordered eigenvectors of $\mathbf{\Theta}_\gamma$. The eigenvalues, χ_i, are completely different though. Figure 3-9 shows a plot of the eigenvalues as a function of γ for $N=11$. The eigenvalues of ρ_γ are well spread. In fact, the eigenvalues can be proved to be differ at least by 1, [14]. Accordingly, the DPSS can be computed without any conditioning problems. MATLAB's `dpss()` function uses this method to find the DPSS.

DPAX is a powerful alternative algorithm to CM. Its performance is nearly optimal, it is less complex, and it has fewer stability problems. The only drawback of DPAX is that for smaller N and large γ, there is a performance loss associated with it. Fortunately, this regime is exactly where the CM algorithm is well-conditioned and feasible to implement.

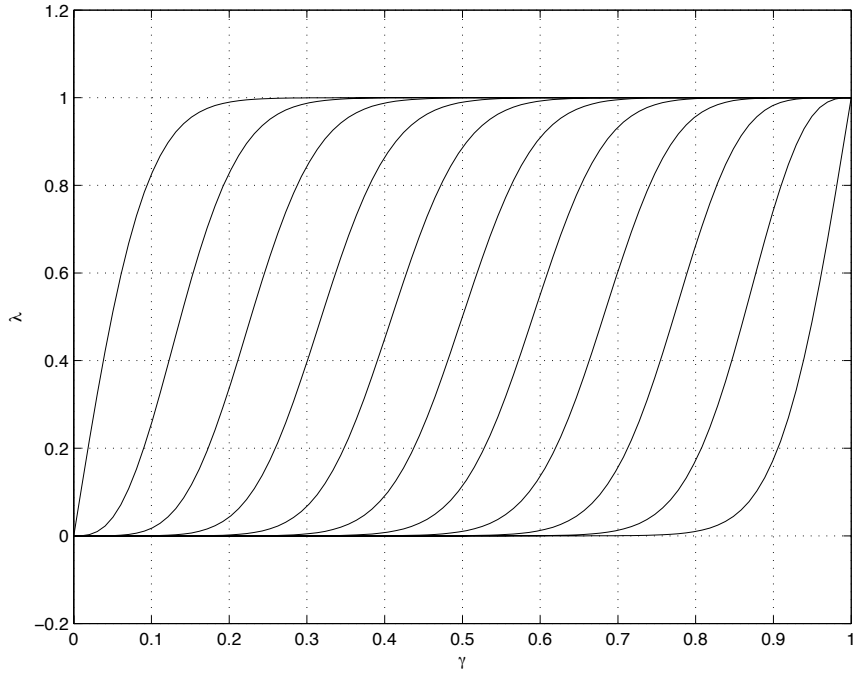

Figure 3-8: Eigenvalues of $\mathbf{\Theta}_\gamma$ versus γ for $N = 11$

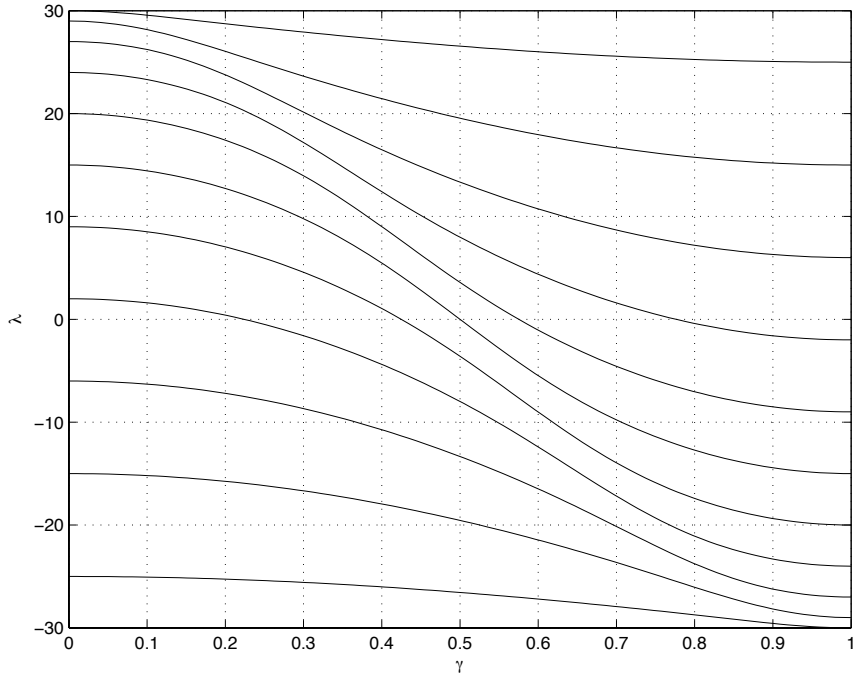

Figure 3-9: Eigenvalues of ρ_γ versus γ for $N = 11$

Chapter 4

Iterative Minimization

In this chapter, as an alternative to the two closed-form algorithms, we develop an iterative solution in the class of projection-onto-convex sets (POCS). The algorithm, which we refer to as Iterative Minimization (IM) is proved to uniquely converge to the optimal solution, $c_{opt}[n]$. Results from numerical simulation are presented. Empirical evaluation shows that the IM has a slow convergence rate.

4.1 Derivation

Iterative algorithms are often preferred to direct computation because they are simpler to implement. One common framework for iterative algorithms is projection onto convex sets (POCS). In the POCS framework, iterations are continued projections that eventually converge to a solution that is either a common point in all the sets projected into or, if there are no such points, then points that are the closest between the sets. Detailed background on POCS can be found in [12].

We formulate a POCS algorithm, called Iterative Minimization (IM), that converges to an optimal window, $w^*[n]$, that when multiplied by $(-1)^n$ gives $c_{\text{opt}}[n]$, the optimal finite-length solution. It is shown in block diagram form as Figure 4-1. There are three projections in each iteration. Each projection is onto a convex set. A set C is convex if

$$\mathbf{w} = \mu\mathbf{w_1} + (1-\mu)\mathbf{w_2} \in C \tag{4.1}$$

for all $\mathbf{w_1}, \mathbf{w_2} \in C$ and $0 \leq \mu \leq 1$, i.e. for any two points in a convex set, the line connecting

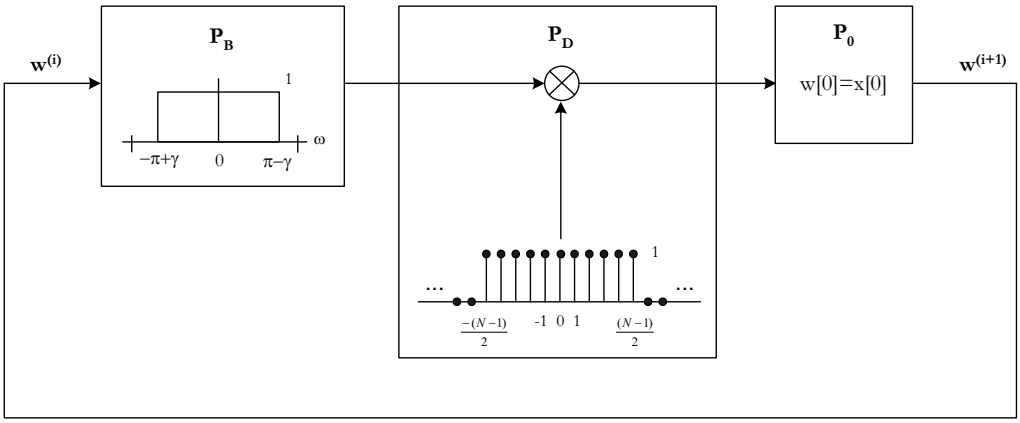

Figure 4-1: POCS Representation

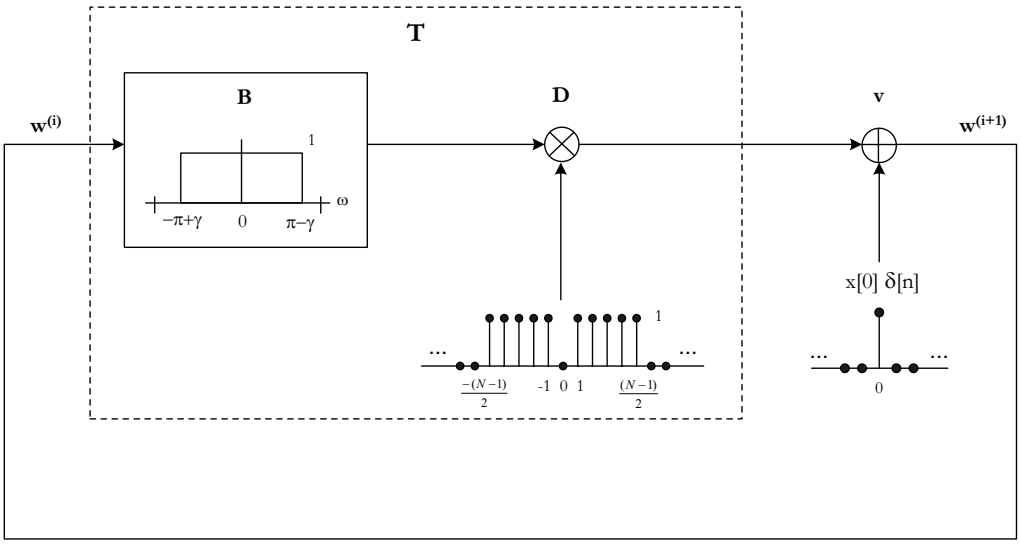

Figure 4-2: Affine Representation

the two points is also inside the set, [12].

The first projection, P_B, is an ideal low-pass filter. This projects the signal onto the set of band-limited signals $\ell_2(\pi - \gamma)$. As proved in [3], $\ell_2(\pi - \gamma)$ is a convex set in ℓ_2. The second projection, P_D, is a truncation. This projects the input onto the set of time-limited signals $\ell_2(-\frac{N-1}{2}, \frac{N-1}{2})$. [3] proves that $\ell_2(-\frac{N-1}{2}, \frac{N-1}{2})$ is also a convex set in ℓ_2. The last projection, P_0, replaces the zeroth sample with the value $x[0]$. It projects the signal onto a set of signals, C_0, that have $w[0] = x[0]$. In ℓ_2, C_0 is a hyperplane, which is by definition a convex set.

Proving that each projection is onto a convex set would normally enable us to use the fundamental theorem of POCS to prove convergence. In general, for convex sets $C_1, C_2, ..., C_m$ and their associated projection operators $P_1, P_2, ..., P_m$, we can define a total projection operation

$$P = P_m P_{m-1} ... P_1 \tag{4.2}$$

The fundamental theorem of POCS states that if the intersection of all the convex sets, $C = \bigcap_{i=1}^{m} C_i$ is non-empty, then the sequence $\{P^n \mathbf{x}\}$ converges weakly to a point of C. It converges strongly in the norm if each C_i is also a closed subspace, [12].

Define $C = \ell_2(\pi - \gamma) \bigcap \ell_2(-N/2, N/2) \bigcap C_0$. C is *empty* because the only point in that is both time-limited and band-limited is the trivial zero-signal. But the zero signal is not in the set C_0.

$$C = \phi \tag{4.3}$$

Consequently, the fundamental theorem cannot be used to prove convergence. Alternatively, in POCS theory, there is a sequential projection theorem for non-intersecting sets. It states that the sequence $\{\mathbf{w_n}\}$ generated by

$$\mathbf{w}^{(n+1)} = P_m ... P_2 P_1 \mathbf{w_n} \tag{4.4}$$

converges weakly if one of the sets $C_1, C_2, ..., C_m$ is bounded,[12]. A set C is bounded if there exists $\mu < \infty$ such that $||\mathbf{w}|| < \mu$ for every $\mathbf{w}$ in C, [12]. By this definition, $\ell_2(\pi - \gamma)$, $\ell_2(-\frac{N-1}{2}, \frac{N-1}{2})$, or C_0 are not bounded. We cannot use the sequential projection theorem to prove convergence.

4.1.1 Convergence

To facilitate proofs, we represent the projections in Figure 4-1 in terms of the affine transformation of Figure 4-2. The two representations are isomorphic; i.e. they give the same solution after *each iteration*. In this representation, there are three steps in each iteration. The first step is B, a band-limiting operator, i.e. an ideal low-pass filter with cut-off $\pi - \gamma$. The second step is D, a truncation operator. The support of the truncation is $[-\frac{N-1}{2}, -1]$ and $[1, \frac{N-1}{2}]$, i.e. D time-limits to $N/2$ and additionally removes the value at index $n = 0$. B and D can be conglomerated together into one linear operator T. The third and last step is the addition of an impulse $v = x[0]\delta[n]$.

Assuming that $\mathbf{w}^{(0)} \in \ell_2$, the iteration defines a sequence in ℓ_2

$$\mathbf{w}^{(n+1)} = T\mathbf{w}^{(n)} + v \tag{4.5}$$

Assume the algorithm has fixed-point $\mathbf{w}^*$. Define the error signal after each iteration, $\mathbf{e}^{(n)}$, as the Euclidean distance from the present signal $\mathbf{w}^{(n)}$ to the fixed-point

$$\mathbf{e}^{(n)} = \mathbf{w}^{(n)} - \mathbf{w}^* \tag{4.6}$$

Applying T to both sides, adding v, and rearranging the expression,

$$T\mathbf{e}^{(n)} + v = T\mathbf{w}^{(n)} - T\mathbf{w}^* + v \tag{4.7}$$

$$T\mathbf{e}^{(n)} = (T\mathbf{w}^{(n)} + v) - (T\mathbf{w}^* + v) \tag{4.8}$$

$\mathbf{w}^*$ is a fixed-point, so $T\mathbf{w}^* + v = \mathbf{w}^*$. In addition, $T\mathbf{w}^{(n)} + v = \mathbf{w}^{(n+1)}$ by definition.

$$T\mathbf{e}^{(n)} = \mathbf{w}^{(n+1)} - \mathbf{w}^* \tag{4.9}$$

$$\mathbf{e}^{(n+1)} = T\mathbf{e}^{(n)} \tag{4.10}$$

Convergence implies that $||\mathbf{e}^{(n)}||$ approaches zero as $n \to \infty$. Thus a sufficient condition for convergence is that T is a strictly non-expansive operator. This means if $\mathbf{w_1}$ and $\mathbf{w_2}$ are two-points in ℓ_2, the normed distance between them must strictly decrease under T. Mathematically,

$$||T(\mathbf{w_1} - \mathbf{w_2})|| < ||\mathbf{w_1} - \mathbf{w_2}|| \tag{4.11}$$

In our case, T has two components: band-limiting and truncation.

$$T = DB \tag{4.12}$$

We focus on the band-limiting operation. The input to T, $\mathbf{w}^{(n)}$, is a signal time-limited to $[-N/2, N/2]$ that has $w[0] > 0$. The only time-limited signal that is also band-limited is the trivial zero-signal. $\mathbf{w}^{(n)}$ is not the zero-signal, so it has energy in the stop-band of the low-pass filter B. Removing this energy will reduce the energy in the input $\mathbf{w}^{(n)}$. Thus,

$$||B\mathbf{w}^{(n)}|| < ||\mathbf{w}^{(n)}|| \tag{4.13}$$

Band-limiting strictly reduces the energy in $\mathbf{w}^{(n)}$. In total, T reduces the energy,

$$||T\mathbf{w}^{(n)}|| < ||\mathbf{w}^{(n)}|| \tag{4.14}$$

This inequality fails only when $\mathbf{w} = \mathbf{0}$. In other words, $||T(\mathbf{w_1} - \mathbf{w_2})|| < ||(\mathbf{w_1} - \mathbf{w_2})$ unless $\mathbf{w_1} = \mathbf{w_2}$. Thus, T is a strictly-non-expansive operator and IM converges strongly to some set of fixed points.

4.1.2 Uniqueness

Suppose that there are two fixed-points, $\mathbf{w_1^*}$ and $\mathbf{w_2^*}$, that are linearly independent such that $\mathbf{w_2^*} \neq \beta\mathbf{w_1^*}$, for any $\beta \in \Re$. Since they are both fixed-points,

$$T\mathbf{w_1^*} + v = \mathbf{w_1^*} \tag{4.15}$$

$$T\mathbf{w_2^*} + v = \mathbf{w_2^*} \tag{4.16}$$

Subtracting the two expressions, and using the linearity of T, implies

$$T(\mathbf{w_1^*} - \mathbf{w_2^*}) = \mathbf{w_1^*} - \mathbf{w_2^*} \tag{4.17}$$

Since we know T is a strictly non-expansive operator, this implies that $\mathbf{w_1^*} = \mathbf{w_2^*}$, a contradiction. Thus, if it exists, the fixed-point $\mathbf{w}^*$ is unique.

4.1.3 Existence

The unique fixed-point, $\mathbf{w}^*$, of the IM algorithm is $(-1)^n c_{\text{opt}}[n]$. We outline a proof using direct substitution. First, we express $\mathbf{w}^*$ by combining the scaling factor $-x_0/\theta_c^{-1}$ into a constant A.

$$\begin{aligned} \mathbf{w}^* &= (-1)^n c_{\text{opt}}[n] \\ &= -\frac{x_0}{\theta_c^{-1}}(-1)^n \boldsymbol{\Theta}_\gamma^{-\mathbf{1}}\delta \\ &= A(-1)^n \boldsymbol{\Theta}_\gamma^{-\mathbf{1}}\delta \end{aligned} \tag{4.18}$$

The POCS representation is used for this proof, since the substitutions are easier. The first two projections, P_B and P_D, can be represented as a Toeplitz matrix, $\boldsymbol{\Theta}_{\pi-\gamma}$. Our goal is to show that $\mathbf{w}^*$, as defined in (4.18), is a fixed-point.

$$P_0 P_B P_D \mathbf{w}^* = \mathbf{w}^* \tag{4.19}$$

$$P_0 \boldsymbol{\Theta}_{\pi-\gamma} A\left((-1)^n \boldsymbol{\Theta}_\gamma^{-\mathbf{1}}\delta\right) = A(-1)^n \boldsymbol{\Theta}_\gamma^{-\mathbf{1}}\delta \tag{4.20}$$

As shown in Chapter 3, $c_{\text{opt}}[n]$ can be expanded into its DPSS basis,

$$c_{\text{opt}}[n] = A\left(\beta_1 \lambda_1^{-1} v_1^\gamma[n] + \cdots + \beta_N \lambda_N^{-1} v_N^\gamma[n]\right) \tag{4.21}$$

The coefficients in the DPSS expansion of $c_{\text{opt}}[n]$ are denoted $\beta_i = v_i^\gamma[0]$. Substituting into (4.20), the next step is to multiply $c_{\text{opt}}[n]$ by $(-1)^n$ and $\boldsymbol{\Theta}_{\pi-\gamma}$. By the dual symmetry of the DPSS, the eigenvectors of $\boldsymbol{\Theta}_{\pi-\gamma}$ are exactly the eigenvectors of $\boldsymbol{\Theta}_\gamma$ modulated by $(-1)^n$. Thus the expression,

$$A\boldsymbol{\Theta}_{\pi-\gamma}\left(\beta_1 \lambda_1^{-1}((-1)^n v_1^\gamma[n]) + \cdots + \beta_N \lambda_N^{-1}((-1)^n v_N^\gamma[n])\right) \tag{4.22}$$

becomes,

$$A[\beta_1 \lambda_1^{-1}(1-\lambda_1)((-1)^n v_1^\gamma[n]) + \cdots + \beta_N \lambda_N^{-1}(1-\lambda_N)((-1)^n v_N^\gamma[n])] \tag{4.23}$$

We can split this expression into two terms, one that is the fixed point, $\mathbf{w}^*$, and the

other composed of residual terms without factors of λ.

$$\begin{aligned} A \quad & \left(\beta_1\lambda_1^{-1}((-1)^n v_1^\gamma[n]) + \cdots + \beta_N\lambda_N^{-1}((-1)^n v_N^\gamma[n])\right) && (4.24)\\ & +A\left(\beta_1((-1)^n v_1^\gamma[n]) + \cdots + \beta_N((-1)^n v_N^\gamma[n])\right) \\ = \quad & (-1)^n c_{\text{opt}}[n] + A(-1)^n\left(\beta_1(v_1^\gamma[n]) + \cdots + \beta_N(v_N^\gamma[n])\right) && (4.25)\end{aligned}$$

To be a fixed point, the second term must be a scaled impulse. Then the projection operator, P_0, will return $A(-1)^n c_{\text{opt}}[n]$. Substitute $\beta_i = v_i^\gamma[0]$ in the second expression. Since the DPSS $v_i^\gamma[n]$ form an orthonormal basis of $\ell_2(-\frac{N-1}{2}, \frac{N-1}{2})$, equation 4.26 is the decomposition of $\delta[n]$ into the orthonormal basis $\{v_i^\gamma[n]\}$.

$$\begin{aligned} = \quad & v_1^\gamma[0](v_1^\gamma[n]) + \cdots + v_N^\gamma[0](v_N^\gamma[n]) && (4.26)\\ = \quad & \langle v_1^\gamma[n], \delta[n]\rangle v_1^\gamma[n] + \cdots + \langle v_N^\gamma[n], \delta[n]\rangle v_N^\gamma[n] && (4.27)\\ = \quad & \delta[\mathbf{n}] && (4.28)\end{aligned}$$

(4.25) can thus be simplified,

$$\mathbf{\Theta}_{\pi-\gamma}\mathbf{w}^* = (-1)^n c_{\text{opt}}[n] + A(-1)^n delta[n] \tag{4.29}$$

Projection with P_0 returns the fixed-point.

$$\begin{aligned} P_0\mathbf{\Theta}_{\pi-\gamma}\mathbf{w}^* \quad & = \quad P_0\left((-1)^n c_{\text{opt}}[n] + A(-1)^n\delta[\mathbf{n}]\right) && (4.30)\\ & = \quad (-1)^n c_{\text{opt}}[n] && (4.31)\end{aligned}$$

We can thus use IM to compute the optimal, finite-length compensation signal, $c_{\text{opt}}[n]$, that was originally found using CM.

4.2 Performance Analysis

4.2.1 Examples

We implemented IM in MATLAB and calculated compensation signals in which $x[0] = -1$. Each iteration was started from an impulse initial condition.

$$\mathbf{w}^{(0)} = \delta[n] \tag{4.32}$$

Figure 4-3 illustrates the solution after 1, 20, 400, and 8000 iterations for $N = 21$ and $\gamma = 0.9\pi$. Figure 4-4 illustrates the same for $N = 21$ and $\gamma = 0.7\pi$. The IM algorithm was allowed to run for up to $N^3 = 8000$ iterations because, in the worst case, using Gaussian elimination, this is the computational complexity of the CM algorithm.

For the first N iterations there is a period of rapid convergence where the solution exponentially converges toward $(-1)^n c_{\text{opt}}[n]$. After this initial period, the rate of convergence slows down considerably. This property is especially clear for the case of $N = 21$, $\gamma = 0.7\pi$. After 20 iterations the solution takes its basic form, the next 7980 iterations augment the solution minimally. As Figure 4-5 illustrates, the solution is far from optimal, even after 8000 iterations.

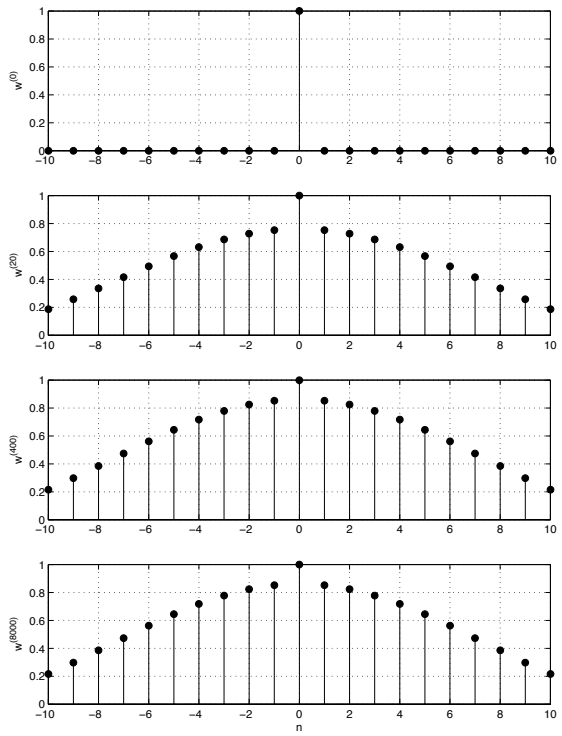

Figure 4-3: Converging to Window, N=21, $\gamma = 0.9\pi$

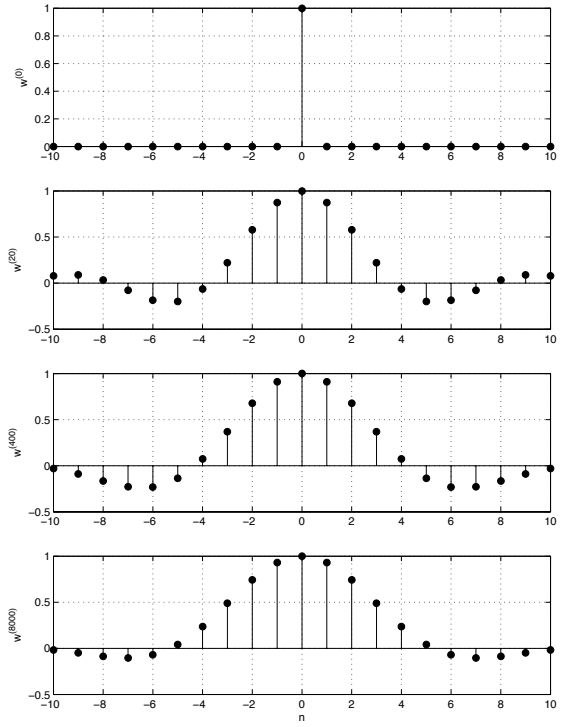

Figure 4-4: Converging to Window, N=21, $\gamma = 0.7\pi$

4.2.2 Convergence Rate

Figures 4-5 and 4-6 are the convergence curves for $\mathbf{w}^{(n)}$ as functions of N and γ, respectively. The long-scale convergence, after 8000 iterations, is shown on the upper plot. The bottom plot is an inset of the convergence curve for the first 100 iterations. Figure 4-5 illustrates the convergence rate for a family of solutions with constant $\gamma = 0.7\pi$, indexed by various values of N. As N increases, the convergence rate decreases. Note the distinctive bend in the convergence curve. It separates the initial, rapid stage of convergence from the later, slower stage of convergence. Figure 4-6 illustrates similar curves for a constant $N = 21$, indexed by various values of γ. This family of curves illustrates the same properties as in Figure 4-5. The curves cross because for larger γ there is a larger initial error for our particular choice of initial conditions, $\mathbf{w}^{(0)} = \delta[n]$. Only the long-time convergence rate is important, so these initial condition effects can be ignored.

The IM algorithm exhibits a historical problem for many POCS algorithms: slow convergence. Although very elegant and theoretically feasible, the slow convergence rate makes IM impractical. Even after $8,000$ iterations, the algorithm does not converge acceptably close to the $(-1)^n c_{\text{opt}}[n]$. Note that IM is slow to converge precisely where CM is ill-conditioned. When CM is ill-conditioned, most of the eigenvalues of $\mathbf{\Theta}_{\pi-\gamma}$ are clustered close to 1. Each iteration scales each DPSS component of $\mathbf{w}^{(n)}$ by λ_i. After N iterations this scaling is λ_i^N. When λ_i are close to 1, there is not much change between each iteration, leading to slow convergence. Though not studied in this treatment, POCS relaxation techniques could potentially be used to speed up the convergence rate.

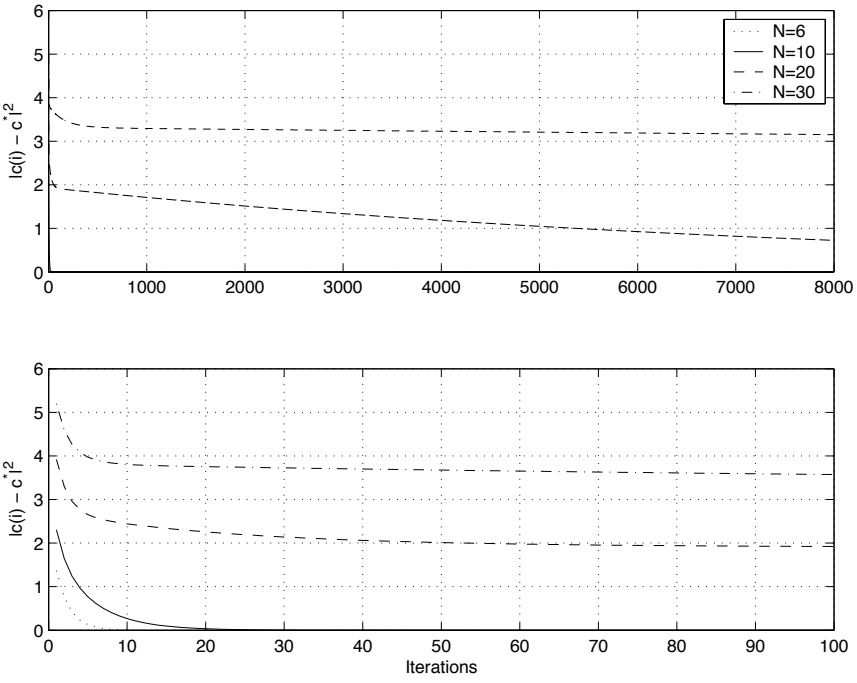

Figure 4-5: Convergence to Optimal Solution for $\gamma = 0.7\pi$ and $N = 7$, $N = 11$, $N = 15$, $N = 21$, $N = 31$

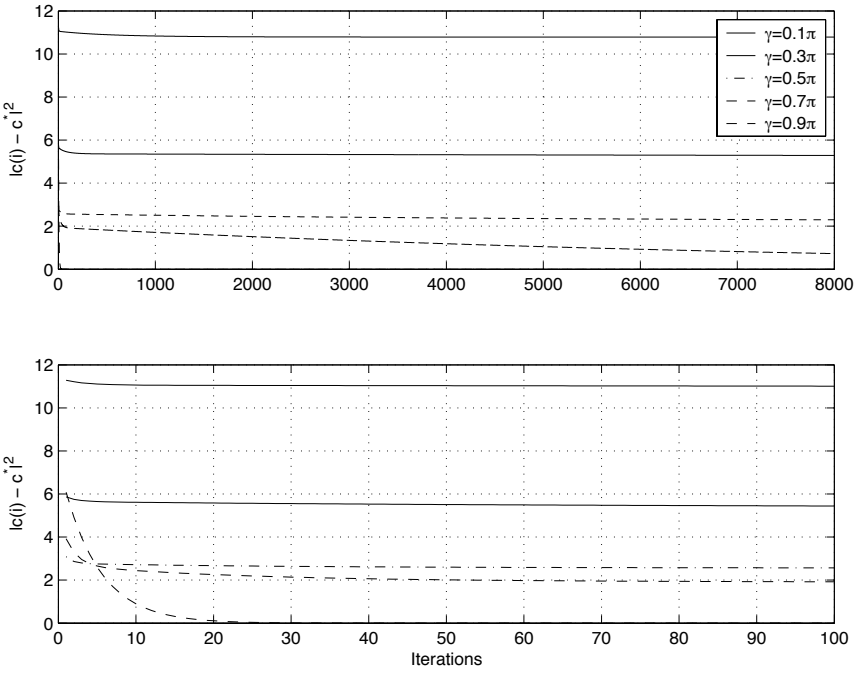

Figure 4-6: Convergence to Optimal Solution for $N = 21$ and $\gamma = 0.1\pi$, 0.3π, 0.5π, 0.7π, 0.9π

Appendix A

DPSS and the Singular Value Decomposition

The discrete prolate spheroidal sequences (DPSS) figure prominently in this thesis. The many extremal properties of the DPSS, developed in a series of papers [5, 9, 10, 11] by Slepian, Landau, and Pollak, are of primary importance. Many of these properties, like "double orthogonality", hint at a deeper mathematical structure. In this appendix, we present the singular value decomposition (SVD) as a unifying framework in which the DPSS can be interpreted.

In particular, we show that the DPSS are an SVD basis for the space of time-limited and band-limited signals. This result may be known to some, but since it is relevant to results presented in the thesis, we include a development here for the sake of completeness. In addition, while mathematical, the development presented below is not rigorous. Where necessary, we have outlined the proof, but the details of the verification are left to the reader.

A.1 Singular Value Decomposition

The singular value decomposition (SVD) is a decomposition of a linear map T into elementary transformations. Generally, the SVD focuses on cases where $T = \mathbf{T}$ is a matrix, but in this development we focus on the case where T is an arbitrary linear map in a coordinate-free Hilbert space. The development closely follows [1], which offers a more rigorous definition of the SVD.

Let T be a linear map from $\mathbf{U}$ to $\mathbf{V}$ as in Figure A-1. Let the inner product in each of these spaces be denoted by $\langle x, y\rangle$ and the squared-norm as $||x||^2 = \langle x, x\rangle$. T^* is the adjoint operator of T. It is a linear map from $\mathbf{V}$ to $\mathbf{U}$ defined such that given an unique vector $u \in \mathbf{U}$ and an unique vector $v \in \mathbf{V}$,

$$\langle Tu, v\rangle = \langle u, T^*v\rangle \tag{A.1}$$

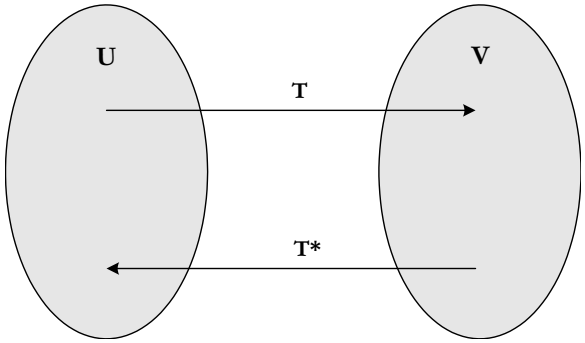

Figure A-1: Singular Value Decomposition

In matrix representation, the adjoint of a matrix $\mathbf{T}$ is its Hermitian $\mathbf{T^H}$. While $\mathbf{T}$ maps its row space into its column space, $\mathbf{T^H}$ maps the column space into the row space, [13]. In our coordinate-free development, the column-space is analogous to $\mathbf{U}$ and the row-space is analogous to $\mathbf{V}$.

The SVD is a decomposition of T such that for an arbitrary $u \in \mathbf{U}$,

$$Tu = \sum_{i=1}^{r} \langle u_i, u\rangle v_i \tag{A.2}$$

where r is the rank of the operator and $\{u_i\}$ and $\{v_i\}$ are complete orthonormal bases of $\mathbf{U}$ and $\mathbf{V}$ respectively. σ_i are the singular values of T. T can thus be interpreted at a basis rotation and scaling, [1]. $\{u_i\}$ and $\{v_i\}$ are the eigenvectors of TT^* and T^*T, respectively. TT^* is a transformation that maps $\mathbf{U}$ into $\mathbf{U}$. It is self-adjoint, since for $u \in \mathbf{U}$

$$\begin{aligned} \langle u, TT^*u\rangle &= \langle T^*u, T^*u\rangle & \text{(A.3)} \\ &= \langle TT^*u, u\rangle & \text{(A.4)} \end{aligned}$$

In addition, TT^* is a positive operator, [1]. By the spectral theorem, its eigenvectors

can be chosen to form a full orthonormal basis, $\{u_i\}$, of $\mathbf{U}$. The eigenvalues are σ_i^2.

$$TT^* u_i = \lambda_i u_i = \sigma_i^2 u_i \tag{A.5}$$

T^*T is also a self-adjoint, positive operator and its eigenvectors are $\{v_i\}$ form an orthonormal basis for $\mathbf{V}$. The eigenvalues are also σ_i^2.

$$T^*T v_i = \lambda_i v_i = \sigma_i^2 v_i \tag{A.6}$$

Intuitively, the SVD is a diagonalization of T using two bases, $\{u_i\}$ and $\{v_i\}$. [1, 13] offer a more detailed development of the SVD.

A.2 Discrete Prolate Spheroidal Sequences

The space of finite-energy signals, ℓ_2, is a Hilbert space under the inner-product

$$\langle x, y \rangle = \sum_{n=-\infty}^{\infty} x[n] y^*[n] \tag{A.7}$$

The set of finite-energy signals band-limited to W, $\ell_2(W)$, forms a subspace of ℓ_2, [12]. $\ell_2(-\frac{(N-1)}{2}, \frac{(N-1)}{2})$, the set of finite-energy signals time-limited to the interval $[-\frac{(N-1)}{2}, \frac{(N-1)}{2}]$, also forms a subspace of ℓ_2. Consequently, both of these subspaces are induced Hilbert spaces under the inner-product (A.7). For clarity, let $\mathbf{U} = \ell_2(W)$ and $\mathbf{V} = \ell_2(-\frac{(N-1)}{2}, \frac{(N-1)}{2})$. In this remainder of this section, we make two claims and outline a proof for each of them.

A.2.1 Band-limiting and Time-limiting as Adjoint Operations

We define a transformation, $T^* : \mathbf{V} \rightarrow \mathbf{U}$, an ideal low-pass filter that band-limits signals in $\ell_2(-\frac{(N-1)}{2}, \frac{(N-1)}{2})$ to W. We will show that the adjoint transformation, $T : \mathbf{U} \rightarrow \mathbf{V}$, is the truncation of band-limited sequences to $[-\frac{(N-1)}{2}, \frac{(N-1)}{2}]$.

We want to show, given $u \in \mathbf{U}$ and $v \in \mathbf{V}$, that

$$\langle Tu, v \rangle = \langle u, T^* v \rangle \tag{A.8}$$

In the right-hand expression, Tu and $v \in \ell_2(-\frac{(N-1)}{2}, \frac{(N-1)}{2})$. The inner product between

these two time-limited signals is

$$\langle Tu, v\rangle = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} u[n]v^*[n] \tag{A.9}$$

In the left-hand expression of (A.8), u and $T^*v \in \ell_2(W)$. Since they are band-limited, they are of infinite length in the time-domain. Let $T^*v = v_{\mathrm{BL}}[n]$. The inner product is

$$\langle u, T^*v\rangle = \sum_{n=-\infty}^{\infty} u[n]v^*_{\mathrm{BL}}[n] \tag{A.10}$$

Using Plancherel's Theorem which states that for two signals $x[n], y[n] \in \ell_2$,

$$\sum_{n=-\infty}^{\infty} x[n]y^*[n] = \int_{-\pi}^{\pi} X(e^{j\omega})Y^*(e^{j\omega})d\omega \tag{A.11}$$

we can express (A.10) in the frequency domain. Both $u[n]$ and $v_{\mathrm{BL}}[n]$ are band-limited to W so

$$\langle u, T^*v\rangle = \int_{-W}^{W} U(e^{j\omega})V^*_{\mathrm{BL}}(e^{j\omega})d\omega \tag{A.12}$$

By definition, $V_{\mathrm{BL}}(e^{j\omega}) = V(e^{j\omega})$ on $[-W, W]$, so

$$\langle u, T^*v\rangle = \int_{-W}^{W} U(e^{j\omega})V^*(e^{j\omega})d\omega \tag{A.13}$$

Our goal is to show that (A.9) and (A.13) are equal. In (A.9), $u[n] \in \ell_2(-\frac{(N-1)}{2}, \frac{(N-1)}{2})$. So, the product $u[n]v^*[n]$ will be zero outside of the interval $[-\frac{(N-1)}{2}, \frac{(N-1)}{2}]$. The summation can be re-indexed from $-\infty$ to ∞.

$$\sum_{n=-N/2}^{N/2} u[n]v^*[n] = \sum_{n=-\infty}^{\infty} u[n]v^*[n] \tag{A.14}$$

Analogously, in (A.13), $V^*(e^{j\omega})$ is band-limited to γ, so the limits can be extended to $-\pi$ and π without changing the value of the integral.

$$\int_{-W}^{W} U(e^{j\omega})V^*(e^{j\omega})d\omega = \int_{-\pi}^{\pi} U(e^{j\omega})V^*(e^{j\omega})d\omega \tag{A.15}$$

By Plancherel's Theorem

$$\sum_{n=-\infty}^{\infty} u[n]v^*[n] = \int_{-\pi}^{\pi} U(e^{j\omega})V^*(e^{j\omega})d\omega \tag{A.16}$$

Therefore, T and T^* are adjoint transformations for $\mathbf{U} = \ell_2(W)$ and $\mathbf{V} = \ell_2(-\frac{(N-1)}{2}, \frac{(N-1)}{2})$.

A.2.2 DPSS as an SVD Basis

We show in this section that the DPSS construct the orthonormal SVD basis for T^*, an ideal low-pass filter. The SVD basis vectors $\{u_i\}$ for $\mathbf{U}$ are the eigenvectors of TT^*. Figure A-2 shows the operation TT^* in block diagram form. The eigenvectors are band-limited signals that after truncation and low-pass filtering are just scaled versions of themselves.

Mathematically, the transformation TT^* can be expressed as an eigenvalue equation

$$\sum_{m=-\frac{N-1}{2}}^{\frac{N-1}{2}} \frac{\sin 2\pi W[n-m]}{\pi[n-m]} u_i[m] = \lambda_i u_i[n] \tag{A.17}$$

where the sequence $u_i[n]$ is band-limited to W, i.e. $u_i[n] \in \ell_2(W)$. (A.17) is identical to (3.3), the eigenvector equation that generates the DPSS. The eigenvectors, $\{u_i\}$, are precisely the DPSS.

Analogously, the basis vectors $\{v_i\}$ of $\mathbf{V}$ can be found as the eigenvectors of T^*T. Figure A-3 shows T^*T in block diagram form. The eigenvectors are time-limited signals that are scaled versions of themselves after low-pass filtering and truncation. Mathematically, the cascade simplifies to an eigenvalue equation

$$\sum_{m=-\frac{N-1}{2}}^{\frac{N-1}{2}} \frac{\sin 2\pi W[n-m]}{\pi[n-m]} v_i[m] = \lambda_i v_i[n] \tag{A.18}$$

where the sequence $v_i[n]$ is time-limited to $[-\frac{(N-1)}{2}, \frac{(N-1)}{2}]$, i.e. $v_i[n] \in \ell_2(-\frac{(N-1)}{2}, \frac{(N-1)}{2})$. Again, (A.18) is identical to equation (3.3). The $\{v_i\}$ are precisely the time-limited DPSS.

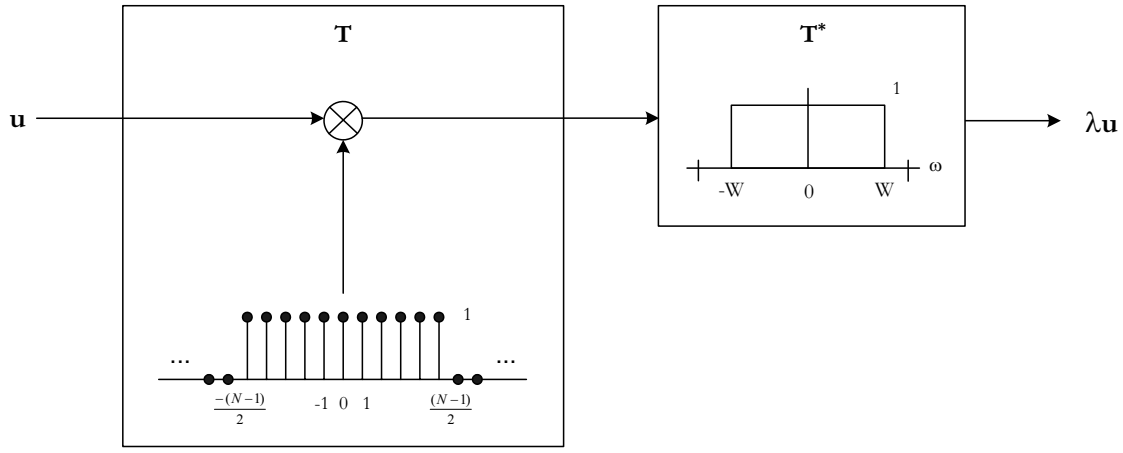

Figure A-2: TT^* Block Diagram

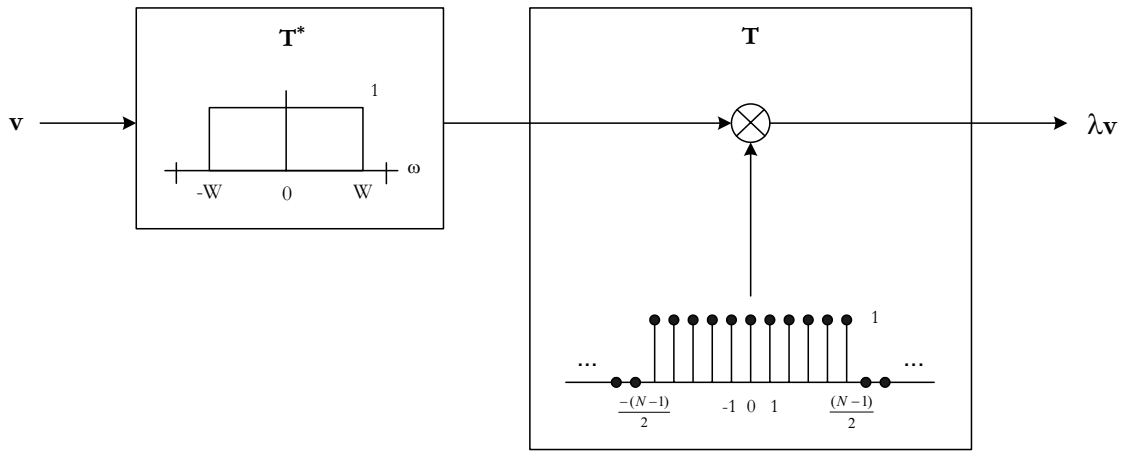

Figure A-3: T^*T Block Diagram

A.3 Historical Results in the SVD Framework

With the SVD the "double-orthogonality" of the DPSS that was historically considered a curious property, follows directly. Truncating a DPSS, $u_i[n]$, results in a scaled, time-limited DPSS, $v_i[n]$.

$$Tu_i[n] = \sigma_i v_i[n] \tag{A.19}$$

The energy preserved under truncation is

$$\begin{aligned} ||Tu_i[n]||^2 &= ||\sigma_i v_i[n]||^2 \\ &= \sigma_i^2 \sum_{n=-\infty}^{\infty} |v_i[n]|^2 \\ &= \sigma_i^2 = \lambda_i \end{aligned} \tag{A.20}$$

Since $\lambda_i < 1$, this implies $\sigma_i < 1$. Low-pass filtering a time-limited DPSS, $v_i[n]$, gives a scaled DPSS, $u_i[n]$.

$$T^* v_i[n] = \sigma_i u_i[n] \tag{A.21}$$

The energy preserved under band-limiting is

$$\begin{aligned} ||T^* v_i[n]||^2 &= ||\sigma_i u_i[n]||^2 \\ &= \sigma_i^2 \sum_{n=-N/2}^{N/2} |u_i[n]|^2 \\ &= \sigma_i^2 = \lambda_i \end{aligned} \tag{A.22}$$

Other extremal properties of the DPSS can also be developed within the SVD framework. For example, in Chapter 3, our desire is to design a window in $\ell_2(-\frac{(N-1)}{2}, \frac{(N-1)}{2})$ that maximizes the concentration ratio

$$\alpha = \frac{\int_{-\pi+\gamma}^{\pi-\gamma} |W(e^{j\omega})|^2 d\omega}{\int_{-\pi}^{\pi} |W(e^{j\omega})|^2 d\omega} \tag{A.23}$$

Define $\mathbf{U} = \ell_2(\pi - \gamma)$ and $\mathbf{V} = \ell_2(-\frac{(N-1)}{2}, \frac{(N-1)}{2})$. The SVD orthonormal bases for $\mathbf{U}$ and $\mathbf{V}$ are the DPSS. We can pose the concentration problem in terms of the DPSS basis.

$w[n] \in \ell_2$, so it has some finite energy E. Our goal is to choose the proper coefficients β_i in

$$w[n] = \beta_1 v_1[n] + \beta_2 v_2[n] + \cdots + \beta_N v_N[n] \tag{A.24}$$

given the constraint

$$\beta_1^2 + \beta_2^2 + \cdots + \beta_N^2 = E \tag{A.25}$$

such that the energy preserved under T^*, a low-pass filter with cut-off $\pi - \gamma$, is maximized. We can write $T^* w[n]$ in terms of $v_i[n]$ and apply the SVD relation

$$\begin{aligned} T^* w[n] &= \beta_1 (T^* v_1[n]) + \beta_2 (T^* v_2[n]) + \cdots + \beta_N (T^* v_N[n]) \\ &= \beta_1 (\sigma_1 u_1[n]) + \beta_2 (\sigma_1 u_2[n]) + \cdots + \beta_N (\sigma_1 u_N[n]) \end{aligned} \tag{A.26}$$

The energy preserved under the transformation is

$$||T^* w[n]||^2 = \beta_1^2 \sigma_1^2 + \beta_2^2 \sigma_2^2 + \cdots + \beta_N^2 \sigma_N^2 \tag{A.27}$$

This is a canonical problem in principal component analysis. Since there is a constraint on the energy E, to maximize $||T^* w[n]||^2$, all of the energy should be put onto the first eigenvector, i.e. $a_1 = \sqrt{E}$ and $a_i = 0$ for $i \neq 1$. Any energy put on the other $v_i[n]$ would be wasted because it could have been put on $v_1[n]$ and more of it would have been preserved. The signal with maximum concentration is thus the first eigenvector, $v_1^{\pi-\gamma}[n]$, the time-limited DPSS with the largest singular value, [10].

Appendix B

Dual Symmetry of DPSS

We rely on the dual symmetry of the DPSS for many of the results in this thesis. This symmetry property is presented in [10], but the proof is left to the reader. For the sake of completeness, we outline a basic proof of dual symmetry in this appendix.

As mentioned in Chapter 3, every DPSS has a dual symmetric partner. In particular,

$$v^{W}_{N+1-i}[n] = (-1)^{n} v^{\pi-W}_{i}[n] \tag{B.1}$$

The eigenvalues are related

$$\lambda^{W}_{N+1-i} = \lambda^{\pi-W}_{i} \tag{B.2}$$

Figure B-1 represents a filter-bank that operates on a time-limited DPSS, $v^{\gamma}[n]$. For notational simplicity we denote $v^{\gamma}[n] = v[n]$. The upper-branch filter, $h[n]$, is an ideal LPF with cut-off γ. The lower-branch filter, $g[n]e^{j\pi n}$, is an ideal high-pass filter with cut-off γ. We denote the output of the upper branch as $v_h[n]$ and the output of the lower branch as $v_g[n]$. The sum of these two signals is $v[n]$ because the two filters $h[n]$ and $g[n]e^{j\pi n}$ are mutually exclusive and together pass $V(e^{j\omega})$ unchanged.

$$v_h[n] + v_g[n] = v[n] \tag{B.3}$$

Multiplying $v[n]$ by a rectangular window, $w[n]$, leaves $v[n]$ unchanged because $v[n]$ is time-limited to $[-\frac{(N-1)}{2}, \frac{(N-1)}{2}]$. Multiplication is distributive,

$$w[n]v[n] \quad = \quad w[n](v_h[n] + v_g[n]) \tag{B.4}$$

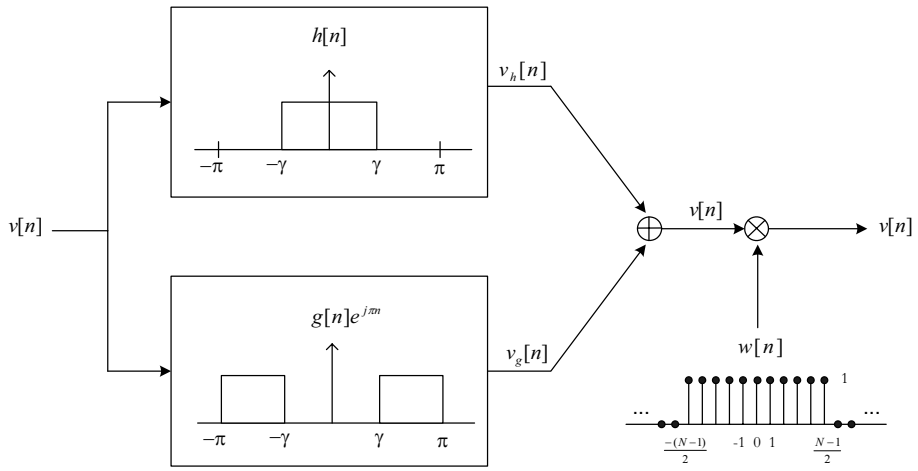

Figure B-1: Identity Filter-bank

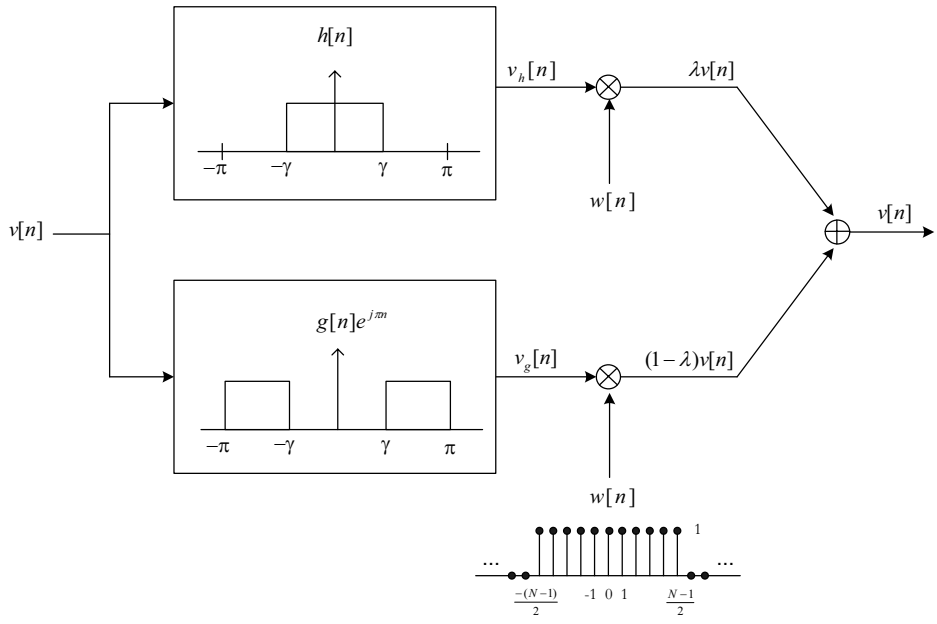

Figure B-2: Identity Filter-bank with multiplication distributed into the branches

$$= \quad w[n]v_h[n] + w[n]v_g[n] \tag{B.5}$$

Thus, we can shift the multiplication by $w[n]$ to the left of the summation node, into the branches of the filter-bank. Figure B-2 represents the resulting filter-bank. The upper-branch is a system for which the eigenfunctions are the DPSS, $v[n]$. Consequently, the output of the upper-branch is $\lambda v[n]$. This implies that the output of the lower-branch must be $(1-\lambda)x[n]$.

Thus, $v[n]$, is also an eigenfunction of the lower branch with eigenvalue $(1-\lambda)$. We can convert the high-pass system of the lower branch into low-pass form by modulating $v_g[n]$ by $e^{-j\pi n} = (-1)^n$. As illustrated in Figure B-3,

$$e^{-j\pi n}v_g[n] = \left(\sum v[m]g[n-m]e^{j\pi[n-m]}\right)e^{-j\pi n} \tag{B.6}$$

$$= \sum (v[m]e^{j\pi m})g[n-m] \tag{B.7}$$

The output $v_g[n](-1)^n$ can be represented as the convolution of $v[n](-1)^n$ with the low-pass filter $g[n]$. Since truncating $v_g[n]$ gives $(1-\lambda)v[n]$, truncating $v_g[n](-1)^n$ gives $(1-\lambda)v[n](-1)^n$. Thus, $v[n](-1)^n$ is an eigenvector of the low-pass system, $\mathbf{Q}$, illustrated in the lower part of Figure B-3. Consequently, $v[n](-1)^n = v^{\gamma}[n](-1)^n$ is a DPSS, $v^{\pi-\gamma}[n]$, with eigenvalue $(1-\lambda)$. Since the eigenvalues of the DPSS are unique and can be ordered, [10]

$$\lambda_1 > \lambda_2 > \cdots > \lambda_N \tag{B.8}$$

the DPSS for $h[n]$, parametrized by γ, fully determine the DPSS for $g[n]$, parametrized by $\pi - \gamma$. Matching the equivalent eigenfunctions proves the dual symmetry of the DPSS.

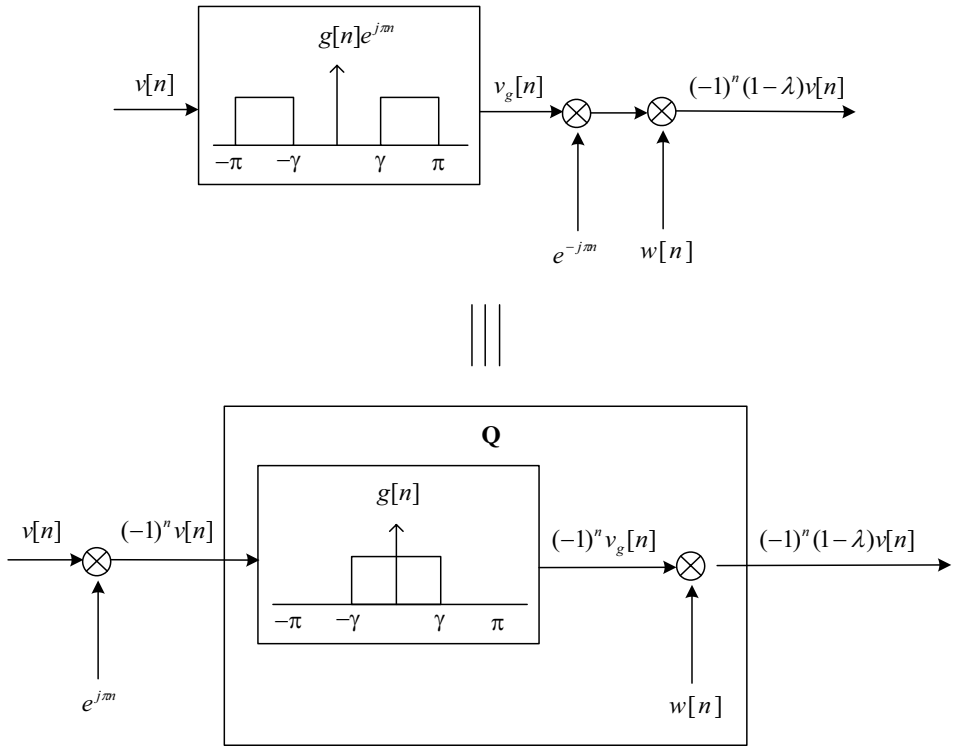

Figure B-3: Equivalence between high-pass system of lower branch and low-pass system

Bibliography

[1] Sheldon Axler. *Linear Algebra Done Right*. Springer-Verlag, New York, 2 edition, 1997.

[2] William F. Ellersick. *Data Converters for High Speed CMOS Links*. PhD thesis, Stanford University, August 2001.

[3] P. J. S. G. Ferreira. Iterative and Noniterative Recovery of Missing Samples for 1-D Bandlimited Signals. In Marvasti, editor, *Nonuniform Sampling: Theory and Practice*, chapter 5, pages 235–281. Kluwer Academic/Plenum Publishers, New York, 2001.

[4] R. G. Fielding and Rank Brimar Ltd. Video display systems. International Patent WO 91/15843, 1991.

[5] H.J. Landau and H. O. Pollak. Prolate Spheroidal Wave Functions, Fourier Analysis and Uncertainty – II. *Bell System Technical Journal*, 40(1):65–84, January 1961.

[6] V. Markandey, R. J. Gove, and Texas Instruments Inc. Method of reducing the visual impact of defects present in a spatial light modulator display. U.S. Patent 5,504,504, 1996.

[7] Alan V. Oppenheim, Ronald W. Schafer, and John R. Buck. *Discrete-Time Signal Processing*. Prentice Hall Signal Processing Series. Prentice Hall, Upper Saddle River, New Jersey, 1999.

[8] Andrew I. Russell. *Regular and Irregular Signal Resampling*. PhD thesis, Massachusetts Institute of Technology, May 2002.

[9] D. Slepian and H. O. Pollak. Prolate Spheroidal Wave Functions, Fourier Analysis and Uncertainty – I. *Bell System Technical Journal*, 40:43–63, January 1961.

[10] David Slepian. Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty –V: The Discrete Case. *Bell System Technical Journal*, 57(5):1371–1430, June 1978.

[11] David Slepian. Some Comments on Fourier Analysis, Uncertainty and Modeling. *SIAM Review*, 25(3):379–393, July 1983.

[12] Henry Stark and Yongyi Yang. *Vector Space Projections*. Wiley Series in Telecommunications and Signal Processing. John Wiley and Sons, New York, 1998.

[13] Gilbert Strang. *Linear Algebra and Its Applications*. Harcourt Brace College Publishers, 3 edition, 1988.

[14] Tony Verma, Stefan Bilbao, and Teresa H. Y. Meng. The Digital Prolate Spheroidal Window. In *1996 IEEE International Conference of Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 3, pages 1351–1354, May 1996.

[15] Chih-Kong Ken Yang, Vladimir Stojanovic, Siamak Jodjtahedi, Mark A Horowitz, and William F. Ellersick. A Serial-Link Tranciever Based on 8-GSamples/s A/D and D/A converters in 0.25-μm CMOS. *IEEE Journal of Solid-State Circuits*, 36(11):1684–1692, November 2001.