Regular and Irregular Signal Resampling

by

Andrew I. Russell

S.B. Electrical Engineering and Computer Science (1998)
Massachusetts Institute of Technology

M.Eng. Electrical Engineering and Computer Science (1999)
Massachusetts Institute of Technology

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

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2002

Author .... Andrew Russell ....
Department of Electrical Engineering and Computer Science
May 17, 2002

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

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

Regular and Irregular Signal Resampling

by

Andrew I. Russell

Submitted to the Department of Electrical Engineering and Computer Science
on May 17, 2002, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

In this thesis, we consider three main resampling problems. The first is the sampling rate conversion problem in which the input and output grids are both regularly spaced. It is known that the output signal is obtained by applying a time-varying filter to the input signal. The existing methods for finding the coefficients of this filter inherently tradeoff computational and memory requirements. Instead, we present a recursive scheme for which the computational and memory requirements are both low. In the second problem which we consider, we are given the instantaneous samples of a continuous-time (CT) signal taken on an irregular grid from which we wish to obtain samples on a regular grid. This is referred to as the nonuniform sampling problem. We present a noniterative algorithm for solving this problem, which, in contrast to the known iterative algorithms, can easily be implemented in real time. We show that each output point may be calculated by using only a finite number of input points, with an error which falls exponentially in the number of points used. Finally we look at the nonuniform lowpass reconstruction problem. In this case, we are given regular samples of a CT signal from which we wish to obtain amplitudes for a sequence of irregularly spaced impulses. These amplitudes are chosen so that the original CT signal may be recovered by lowpass filtering this sequence of impulses. We present a general solution which exhibits the same exponential localization obtained for the nonuniform sampling problem. We also consider a special case in which the irregular grid is obtained by deleting a single point from an otherwise regular grid. We refer to this as the missing pixel problem, since it may be used to model cases in which a single defective element is present in a regularly spaced array such as the pixel arrays used in flat-panel video displays. We present an optimal solution which minimizes the energy of the reconstruction error, subject to the constraint that only a given number of pixels may be adjusted.

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

to Wendy-Kaye

Acknowledgments

I would first like to thank my advisor, Al, for his support over the last four years, and especially in the last few months. Some how you knew it “would come together”, even when I didn’t. I am grateful for your encouragement to join the Ph.D. program, and for taking me on as a student. You have done a great deal for me over the years. I have particularly enjoyed our research discussions, and the atmosphere of freedom which characterizes them.

Thanks to every member of DSPG, especially Nick and Stark for their friendship. Thanks to Wade, Petros, Charles, Maya and Sourav for many fruitful discussions. I am particularly indebted to Yonina for several intellectual contributions (including initiating much of the work on nonuniform sampling), and for truly being a close friend and confidant. For the nonstudents in the group: Vanni and Greg, thank you for helping me get through difficult times, and thanks to Dianne and Darla, whose influence prevents the group from becoming an unbearably boring signal processing zone.

Pastor Brian thank you for your mentorship, encouragement and close friendship. Jung-Min, a large part of the miracle of getting this written is due directly to your prayers and support.

To my family, I love you all. Mom, thank you for not letting me settle for anything less than excellent. Your hard work has paid off. Dad, thanks for passing on your knack for engineering. It was your talent for building and fixing things that served as my inspiration. Steve and Joe, you are the best brothers in the world.

Finally I would like to publicly acknowledge the role of my wife Wendy-Kaye. Were it not for you I would have abandoned pursuing this degree a long time ago. You believed in me and supported me so selflessly over the last five years. You have always loved, and always given, and always been strong. I love you so much. There is no one else in the world as beautiful and wonderful as you are.

The fact that this thesis is finished is truly a miracle, for which I give all the credit to God, the creator of the universe, who has granted us the ability to be creative.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The field of Digital Signal Processing (DSP), relies on the fact that analog signals may be represented digitally so that they may be processed using a digital computer. In this thesis we consider two different digital representations, one which satisfies an instantaneous sampling relationship, and one which satisfies a lowpass reconstruction relationship. We discuss the problem of *resampling*, defined as the process of converting between two digital representations of the same analog signal, which have different sampling grids. These grids may be regular or irregular. A primary contribution of the thesis is the development of efficient algorithms which are suitable for real-time implementation on a DSP device.

As stated, the problem of resampling is quite broad in scope, and encompasses a variety of practical problems commonly encountered in the field of DSP. Some of these problems have already been studied in depth in the current literature while others have received comparatively little attention.

1.1 Framework

In this thesis, we consider continuous-time (CT) signals which can be modeled by the mathematical set $\mathcal{B}_\gamma$ which represents the space of all finite energy functions bandlimited to $\gamma < \infty$. Formally, $\mathcal{B}_\gamma$ is the space of all functions $x(t)$ which can be expressed in the form

$$x(t) = \frac{1}{2\pi} \int_{-\gamma}^{\gamma} X(\omega) e^{j\omega t} \, d\omega, \tag{1.1}$$

where $X(\omega)$, the Fourier transform of $x(t)$, is square integrable over $[-\gamma, \gamma]$, and is taken to be zero outside $[-\gamma, \gamma]$. We will assume $\gamma < \pi$ for notational convenience. This assumption does not restrict us since any bandlimited signal can be mapped into $\mathcal{B}_\gamma$ for $\gamma < \pi$ by rescaling the time axis.

The set of all bandlimited functions is quite large and is rich enough to model many signals of practical engineering interest. This is because most real-world signals have some effective bandwidth, *i.e.*, above some frequency they contain very little energy. Also, an analog lowpass filter can sometimes be introduced to approximately limit the bandwidth of a signal. This bandlimited assumption is often used when dealing with analog signals, though in recent times some work has been done in the area of sampling and reconstruction with spline type functions [38] and other non-bandlimited CT functions which lie in more arbitrary spaces [1, 7]. However, in this thesis, we only consider bandlimited signals.

In order to represent the CT signal digitally, two steps are required. First, a discrete sequence of real numbers must be used to represent the signal, and second, each of the real numbers must be represented by a finite number of bits. For the cases which we consider, there is no information lost in the first step, *i.e.*, the sequence of real numbers perfectly represents the CT signal. In the second step there is usually a loss of information because of the quantization involved. In this thesis, we will only establish that small errors in the sequence of real numbers, by some appropriate metric, will correspond to commensurately small errors in the corresponding CT signal. A discrete representation which has this property is called *stable*. For a stable representation the approximation error can in principle be made as small as needed by choosing an appropriate number of bits in the digital representation. Therefore we will assume infinite precision for all quantities which we consider.

The idea that a bandlimited CT signal could be perfectly represented by a discrete sequence of numbers was formalized by Shannon [35] in his seminal work on communication in the presence of noise. For a historical survey as well as a summary of the extensions and generalizations of this theorem, sometimes called the WKS sampling theorem[1], see [12, 39, 40]. The sampling theorem states that a bandlimited CT signal $x(t) \in \mathcal{B}_\gamma$ is uniquely determined by its values at the set of regularly spaced times $\{t_n = nT; n = 0, \pm1, \pm2, \dots\}$, where $T = \pi/\gamma$. Furthermore, the function $x(t)$ can be expressed as the pointwise convergent series

$$x(t) = \sum_{n=-\infty}^{\infty} a_n \phi(t - t_n), \tag{1.2}$$

where $\phi(t)$ is given as

$$\phi(t) = \frac{\sin \gamma t}{\gamma t}. \tag{1.3}$$

The coefficients in the expansion $\{a_n; n = 0, \pm1, \pm2, \dots\}$ are the required samples of the signal, given by

$$a_n = x(t_n), \quad n = 0, \pm1, \pm2, \dots \tag{1.4}$$

Henceforth we simply use the notation $\{a_n\}$ to refer to the set $\{a_n; n = 0, \pm1, \pm2, \dots\}$. We will also use the index set I to refer to the integers, so $n \in I$ means $n = 0, \pm1, \pm2, \dots$

$W = 1/T$ is the minimum sampling rate needed, referred to by Shannon as the Nyquist rate in recognition of the work done by Nyquist [24] on telegraph transmission theory.

Even though the WKS sampling theorem was formalized around the time of Shannon, the concept of representing continuous-time signals as a sequence of values predates Shannon. The concept of nonuniform sampling was even known to Cauchy, as indicated by the following statement, attributed to Cauchy by Black [3]:

[1] The theorem is referred to as the WKS sampling theorem after its originators Whittaker, Kotel'nikov and Shannon.

> "If a signal is a magnitude-time function and if time is divided into equal intervals such that each subdivision comprises an interval T seconds long where T is less than half the period of the highest significant frequency component of the signal, and if one instantaneous sample is taken from each subinterval in any manner; then a knowledge of the instantaneous magnitude of each sample plus a knowledge of the instant within each subinterval at which the sample is taken contains all the information of the original signal."

Cauchy's statement has been made more precise in three theorems by Yen [45] and two by Yao and Thomas [43], the last of which is often referred to as the nonuniform sampling theorem. In this thesis, as implied in the above quotation, we use DT representations which consist of amplitude-time pairs. However, we do not necessarily require that each amplitude value correspond to the instantaneous magnitude of the underlying CT signal. The amplitude-time pairs may be related to the underlying CT signal in some other way. We require a time-instant to be associated with each amplitude value because we wish to accommodate representations which use irregularly spaced samples. In more traditional signal processing terminology, a DT signal is only required to have an associated sampling rate. Using our terminology, a DT signal then, consists of a set of amplitudes $\{a_n\}$ and a corresponding set of time-instants $\{t_n\}$, and may be written as the pairs $\{a_n, t_n\}$. We refer to the set of times $\{t_n\}$ as the *grid*, which may be regular, as in the WKS sampling theorem, or irregular. In this thesis, we restrict ourselves to grids which have a *uniform density*. Specifically, if a grid $\{t_n\}$ has a uniform density of $W = 1/T$, then it satisfies

$$|t_n - nT| \leq L < \infty; \quad n \in I, \tag{1.5}$$

and

$$|t_n - t_m| \geq \delta > 0; \quad n \neq m, \tag{1.6}$$

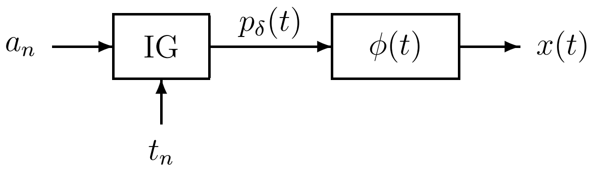

Figure 1-1: The ideal lowpass reconstruction system.

for some positive δ and finite L.

There may be other ways of representing a CT signal as a sequence of numbers for which no inherent notion of a grid exists, for example, abstract inner-product sampling described in [7]. However in the thesis, we use discrete representations which incorporate both a set of amplitudes and a set of time-instants.

The expansion (1.2) can be interpreted in terms of an ideal lowpass filter as depicted in figure 1-1. The block labeled IG is the impulse generator, and is defined by

$$p_\delta(t) = \sum_{n=-\infty}^{\infty} a_n \delta(t - t_n), \tag{1.7}$$

where $\delta(t)$ is the Dirac delta function, referred to as a CT impulse. The subscript δ in $p_\delta(t)$ is used to indicate that this function is composed of a sequence of CT impulses, and is the convention which we use in this thesis. $p_\delta(t)$, defined by (1.7), is the impulse-train associated with the DT signal $\{a_n, t_n\}$. It should be noted that two different DT signals may have the same associated impulse-train, because an amplitude value of zero contributes nothing to the sum (1.7). $p_\delta(t)$ is taken as the input to an ideal lowpass filter with impulse response $\phi(t)$ given by (1.3). The output of the filter is the bandlimited CT signal $x(t)$. If $x(t)$ satisfies the relationship given by (1.2) and depicted in figure 1-1, then we shall refer to $x(t)$ as the *lowpass reconstruction* of the DT signal $\{a_n, t_n\}$.

The lowpass reconstruction relationship can be viewed as a generalization of Shannon's reconstruction formula, where the grid is no longer restricted to be regular. Problems involving lowpass reconstruction with an irregular grid have not received a great deal of attention in the literature. More commonly, when dealing with irregular

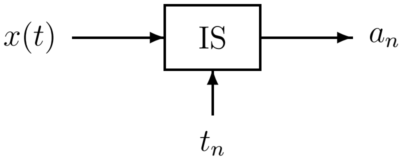

Figure 1-2: The instantaneous sampling system.

grids it is assumed that the amplitudes $\{a_n\}$ represent actual values of the underlying CT signal $x(t)$, *i.e.*, (1.4) holds. In this case, the amplitudes $\{a_n\}$ are referred to as the *instantaneous samples* of $x(t)$. This notion of instantaneous sampling corresponds to the usual meaning of the word "sampling" as used in the WKS sampling theorem and the nonuniform sampling theorem. If a DT signal, $\{a_n, t_n\}$, is related to a CT signal, $x(t)$, by instantaneous sampling, then the signals satisfy

$$a_n = x(t_n), \tag{1.8}$$

which defines our instantaneous sampler. We use a block labeled IS to denote the instantaneous sampler, as illustrated in figure 1-2.

If the grid is regular, then a CT signal, $x(t)$, and a DT signal, $\{a_n, t_n\}$, which satisfy the instantaneous sampling relationship will also satisfy the lowpass reconstruction relationship (once appropriate conditions on the sampling rate, the bandwidth of $x(t)$ and the cutoff frequency of the lowpass filter are met). This is not the case in general for an irregular grid.

1.2 Resampling

If the grid associated with a DT signal is irregular, then we may wish to resample it onto a regular grid, perhaps for easier processing. Even if the original signal has a regular grid, we may still wish to resample it so that the sampling rate will match that of another signal which we may wish to combine with the original signal. In the thesis, resampling is defined as the process of transforming a DT signal $\{a_n, t_n\}$ which

Chapter	Input	Output
2	lowpass reconstruction	instantaneous sampling
3	regular grid	regular grid
4	instantaneous sampling	regular grid
5	regular grid	lowpass reconstruction

Table 1.1: Resampling algorithms discussed in each chapter.

represents a CT signal $x(t)$, into a new DT signal $\{a'_n, t'_n\}$ which also represents $x(t)$. The output grid $\{t'_n\}$ will in general be different from the input grid $\{t_n\}$. For each of the DT signals $\{a_n, t_n\}$ and $\{a'_n, t'_n\}$, the grid may be regular or irregular. If the grid is irregular then the DT signal may be related to $x(t)$ either by instantaneous sampling, or by lowpass reconstruction. If the grid is regular, then both relationships will be satisfied. Thus each DT signal is related to the CT signal either by instantaneous sampling or by lowpass reconstruction, or both (*i.e.*, a regular grid is being used). Table 1.1 summarizes the various resampling algorithms that are discussed in each chapter of the thesis.

Using a regular grid is particularly attractive for a number of reasons. For example, linear time-invariant (LTI) filtering can be applied to the signal by means of a discrete-time convolution with the amplitude values [25]. Therefore, in most of the motivating applications which we consider, either the input or output grid is regular.

The main focus of the thesis is to develop efficient DT resampling algorithms which are suitable for real-time implementation on a DSP device. We deal with three main categories of resampling algorithms which have obvious engineering applications. The first category is that in which the input and output grids are both regular but have a different spacing. This is equivalent to the problem of sampling rate conversion and is discussed in chapter 3. In the second category, we are given a DT signal obtained by instantaneous sampling on an irregular grid, and we wish to resample this to a regular grid. This is referred to as the nonuniform sampling problem and is discussed in chapter 4. In the third category, the input grid is regular, and we wish to resample the signal onto an irregular while maintaining the lowpass reconstruction property.

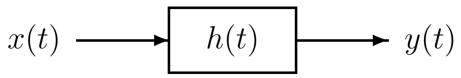

Figure 1-3: Continuous-time LTI filtering.

This problem is discussed in chapter 5 and is referred to as the nonuniform lowpass reconstruction problem. In each case, we provide motivation and highlight the current state of research presented in the literature.

Before we discuss these three categories of resampling algorithms, we will introduce a system referred to as the *elementary resampler*, which is discussed in chapter 2. This elementary resampler plays a key role in each of the resampling algorithms described in chapters 3 through 5.

1.3 Notation

This section summarizes the notational conventions used in the thesis. We typically use n or m as the discrete-time independent variable which is always an integer. The continuous time variable is typically t. CT signals are written with parentheses as in $x(t)$, and sequences are either written with square brackets as in $x[n]$ or with the subscript notation a_n. The set notation $\{a_n\}$ is also used, and is a short hand for $\{a_n; n \in I\}$. CT filters are drawn as shown in figure 1-3, with the input and output signals being a function of the same time variable. The box is labeled with a third CT signal, which represents the impulse response of the filter. The CT filter block in figure 1-3 is defined by the convolution integral,

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)\ d\tau. \tag{1.9}$$

DT LTI filters are drawn as shown in figure 1-4, with the input and output signals using the same index, in this case n. The block is labeled with the filters impulse response, which also uses the same index. The input-output relationship is defined

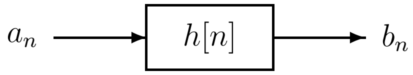

Figure 1-4: Discrete-time LTI filtering.

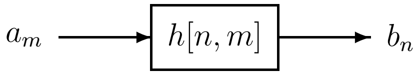

Figure 1-5: Time-varying discrete-time filtering.

by

$$b_n = a_n * h[n] = \sum_{m=-\infty}^{\infty} a_m h[n-m]. \tag{1.10}$$

We also define a more general time-varying DT filter which is drawn as shown in figure 1-5, with the input and output signals using different time variables. The block is labeled with the kernel which is a function of both variables. The input output relationship is defined by

$$b_n = \sum_{m=-\infty}^{\infty} a_m h[n, m]. \tag{1.11}$$

We use lower case letters for time domain signals, and the corresponding capital letters for the CT or DT Fourier transforms. For example, $B(\omega)$ is the DT Fourier transform of b_n, defined by

$$B(\omega) = \sum_{n=-\infty}^{\infty} b_n e^{-j\omega n}, \tag{1.12}$$

and $X(\omega)$ is the CT Fourier transform of $x(t)$, defined by

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t}\, dt. \tag{1.13}$$

We use the dot notation as in $\dot{x}(t)$ to represent differentiation, so that

$$\dot{x}(t) = \frac{d}{dt}\, x(t). \tag{1.14}$$

The notation $\delta[n]$ is used to refer to the unit sample, sometimes called a DT impulse, and is defined by

$$\delta[n] = \begin{cases} 1, & n = 0; \\ 0, & \text{otherwise.} \end{cases} \tag{1.15}$$

We use $u[n]$ to refer to the DT unit step function defined by

$$u[n] = \begin{cases} 1, & n \geq 0; \\ 0, & \text{otherwise.} \end{cases} \tag{1.16}$$

$\delta(t)$ is Dirac delta function which we call a CT impulse. $\delta(t)$ is defined by its behavior under an integral,

$$\int_{-\infty}^{\infty} x(t)\delta(t)\, dt = x(0), \tag{1.17}$$

for every function $x(t)$ which is finite and continuous at $t = 0$. Formally, $\delta(t)$ is not a function, but a *distribution.* The CT unit step function is written as $u(t)$, and is defined as

$$u(t) = \begin{cases} 1, & t \geq 0; \\ 0, & \text{otherwise.} \end{cases} \tag{1.18}$$

$r(t)$ is the CT unit height rectangular (square) window, defined by

$$r(t) = u(t) - u(t-1), \tag{1.19}$$

or

$$r(t) = \begin{cases} 1, & 0 \leq t < 1; \\ 0, & \text{otherwise.} \end{cases} \tag{1.20}$$

CT signals which are made up of a countable number of CT impulses are called impulse-train signals and are written with the subscript δ as in $p_\delta(t)$.

Chapter 2

The Elementary Resampler

In this chapter we present a system we refer to as the elementary resampler, which theoretically involves an ideal CT lowpass filter. This resampler is the simplest case which we discuss, and is used as a subsystem in the more complicated resamplers discussed in subsequent chapters. In section 2.1, we present the ideal elementary resampler. We then focus on practical implementations of the resampler, and present a general algorithm in section 2.2 which is based on known methods. In section 2.4 we evaluate the performance of this algorithm by considering a design example. In section 2.5 we consider a special case in which at least one of the grids involved in the resampling process is regular and in section 2.5.3 we present a method for designing optimal filters for this special case. Finally, in section 2.5.5 we show that there is an improvement in performance which can be achieved over the more general case.

2.1 Ideal CT System

The ideal elementary resampler is depicted in figure 2-1. If we let $h(t)$ be an ideal lowpass filter, then the first two blocks, impulse generator (IG) and filter $h(t)$, represent the lowpass reconstruction system defined in chapter 1. The third block, labeled IS represents the instantaneous sampler, also defined in chapter 1. In general, the filter $h(t)$ may be any CT filter. We will assume that the sequences $\{a_n\}, \{t_n\}$ and $\{t'_n\}$ are known and that we are required to find the sequence $\{a'_n\}$.

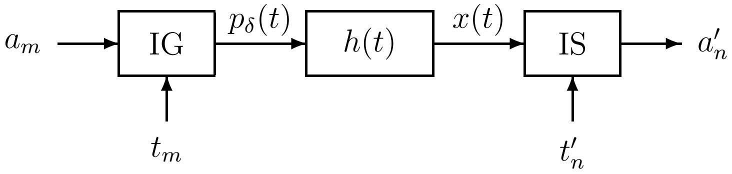

Figure 2-1: The elementary resampler.

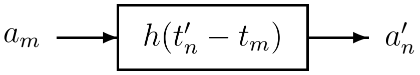

Figure 2-2: Elementary resampler as a time-varying DT filter.

From the mathematical relationships which define each block in figure 2-1, it can be shown that the corresponds to the equation

$$a'_n = \sum_{m=-\infty}^{\infty} a_m h(t'_n - t_m). \tag{2.1}$$

This is equivalent to the time-varying convolution sum where $h(t'_n - t_m)$ is the convolution kernel. This time-varying filter is depicted in figure 2-2 and is equivalent to the system of figure 2-1.

2.2 DT System

The elementary resampler described by (2.1) has been analyzed in detail by Ramstad [28] and others [17, 18, 36], though in the context of uniform grids. Here we describe an algorithm which is based on this work, and which can be implemented in discrete time.

In order to be able to evaluate (2.1) we will restrict $h(t)$ to be zero outside the interval $[0, V)$. Thus we have the finite sum

$$a'_n = \sum_{\substack{m \\ 0 \le (t'_n - t_m) < V}} a_m h(t'_n - t_m). \tag{2.2}$$

This sum only has a finite number of terms since the grids are uniformly dense, and thus cannot have infinitely many points in a finite interval. The coefficients in the sum, $\{h(t'_n - t_m)\}$, are referred to as the *filtering coefficients* because (2.2) can be thought of as a time-varying filter.

Ramstad [28] showed that generating the filtering coefficients in (2.2), could easily dominate the computation. This is especially true if a complicated, closed-form expression for $h(t)$ must be evaluated. These filtering coefficients depend only on the grids and the function $h(t)$, and are thus independent of the input and output signals. If the grids are periodic with a shared period or if there is some other regular structure to them, then it may be possible to precalculate and store all required filtering coefficients. However, in general they must be calculated in real-time. Lagadec *et al* [18] proposed a system in which a table of finely spaced samples of $h(t)$ was stored. Then when the value of $h(t)$ was required at some point, it was simply replaced by the nearest sample from the table. Smith and Gossett [36] considered performing linear interpolation of adjacent samples and Ramstad [28] additionally considered cubic spline or some other form of spline interpolation. Ramstad showed that in each of these cases, the resulting impulse response can be expressed in the form

$$h(t) = \sum_{n=0}^{N-2} g[n] \frac{b_k(t/\epsilon - n)}{\epsilon}, \tag{2.3}$$

where $g[n]$ are the samples stored in the table, ϵ is the spacing of these samples, and $b_k(t)$ is the kth order non-centered B-spline function. Ramstad concluded that for smaller values of k, less computation was required in order to achieve a given reconstruction error, but more memory must be used to store the required coefficients. Thus there is an inherent tradeoff between computational and memory requirements for the elementary resampler.

$b_k(t)$ is defined by

$$b_0(t) = \begin{cases} 1, & 0 \le t < 1; \\ 0, & \text{otherwise,} \end{cases} \tag{2.4}$$

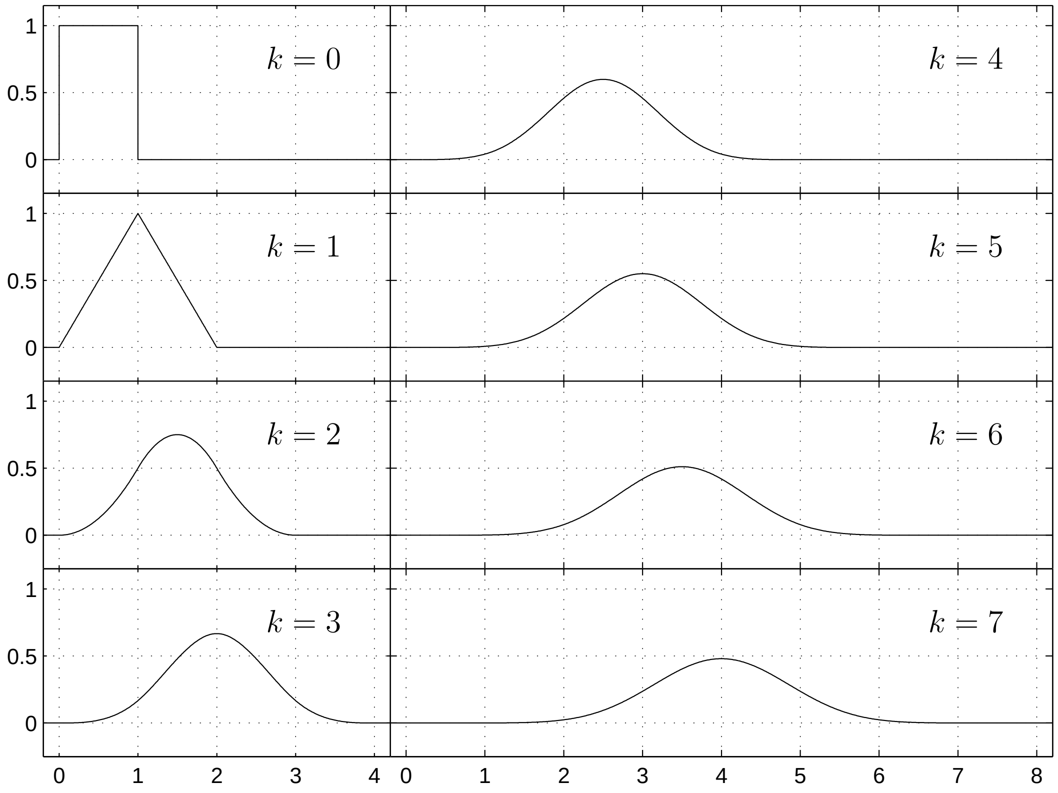

Figure 2-3: The first eight non-centered B-splines.

and

$$b_k(t) = b_0(t) * b_{k-1}(t), \quad k \geq 1. \tag{2.5}$$

Figure 2-3 shows some of these B-splines for different values of k. It can be shown [38] that these B-splines and $h(t)$ are piecewise polynomial. Under this restriction, $h(t)$ is an order k polynomial over N regularly spaced intervals, each of length ϵ. Thus $m\epsilon \leq t < (m+1)\epsilon$ defines the mth interval, where $m = 0, 1, 2, \cdots, N-1$.

In order to calculate $h(t)$ at an arbitrary point in the interval $[0, V)$, where $V = N\epsilon$, we first let

$$t = m\epsilon + \tau; \quad m \in \{0, 1, \ldots, N-1\}, \tau \in [0, \epsilon). \tag{2.6}$$

The unique m and τ which satisfy (2.6) for any given t are given by

$$m = \left\lfloor \frac{t}{\epsilon} \right\rfloor, \tag{2.7}$$

and

$$\tau = t - \left\lfloor \frac{t}{\epsilon} \right\rfloor \epsilon. \tag{2.8}$$

The notation $\lfloor \xi \rfloor$ represents the *floor* of ξ and is defined as the largest integer which is less than or equal to ξ.

For any m and τ satisfying (2.6), $h(t)$ can be expressed as

$$h(m\epsilon + \tau) = \sum_{i=0}^{k} c[i,m]\tau^i; \quad 0 \le \tau < \epsilon, \tag{2.9}$$

where the numbers $\{c[i,m]; m = 0, 1, \cdots, N-1; i = 0, 1, \cdots, k\}$, are referred to as the *polynomial coefficients.*

An equivalent form for the expression (2.9) is

$$h(m\epsilon + \tau) = c[0,m] + \tau \ \Big(c[1,m] + \tau \ \big(c[2,m] + \cdots \tau \ (c[k-1,m] + \tau c[k,m]) \cdots \big)\Big), \tag{2.10}$$

which requires fewer multiplications. It can be shown that the polynomial coefficients in (2.9) are related to $g[n]$ in (2.3) by

$$c[i,m] = g[m] * \frac{d_k[i,m]}{\epsilon^{i+1}}, \tag{2.11}$$

where the numbers $d_k[i,m]$ are the polynomial coefficients associated with the B-spline. That is

$$b_k(m + \xi) = \sum_{i=0}^{k} d_k[i,m]\xi^i; \quad m \in \{0, 1, \ldots, k\}, \xi \in [0, 1). \tag{2.12}$$

k	i	m	$d_k[i,m]$
0	0	0	1
1	0	1	1
1	1	0	1
1	1	1	-1
2	0	1	1/2
2	0	2	1/2
2	1	1	1
2	1	2	-1
2	2	1	-1
2	2	0	1/2
2	2	2	1/2
3	0	1	1/6
3	0	2	2/3
3	0	3	1/6
3	1	1	1/2
3	1	3	$-1/2$
3	2	1	1/2
3	2	2	-1
3	2	3	1/2
3	3	0	1/6
3	3	1	$-1/2$
3	3	2	1/2
3	3	3	$-1/6$
4	0	1	1/24
4	0	2	11/24
4	0	3	11/24
4	0	4	1/24
4	1	1	1/6
4	1	2	1/2
4	1	3	$-1/2$
4	1	4	$-1/6$
4	2	1	1/4
4	2	2	$-1/4$
4	2	3	$-1/4$
4	2	4	1/4
4	3	1	1/6
4	3	2	$-1/2$
4	3	3	1/2
4	3	4	$-1/6$
4	4	0	1/24
4	4	1	$-1/6$
4	4	2	1/4
4	4	3	$-1/6$
4	4	4	1/24

Table 2.1: Nonzero polynomial coefficients for the B-splines ($k \leq 4$).

The nonzero values of $d_k[i,m]$ for $k \leq 4$ are listed in table 2.1.

2.3 Filter Specification

In this section, we look at a design example in which the filter $h(t)$ has a frequency response $H(\omega)$ which is required to meet the lowpass specification described by the following two conditions;

$$1 - \delta_p \leq |H(\omega)| \leq 1; \quad \text{for } |\omega| \leq \omega_p, \tag{2.13}$$

and

$$|H(\omega)| \leq \delta_s; \quad \text{for } |\omega| \geq \omega_s. \tag{2.14}$$

Figure 2-4 shows this specification, along with an example of a frequency response which meets it. We will optionally require that $H(\omega)$ have generalized linear phase (see [26]), which imposes a symmetry constraint on $h(t)$, and typically implies that a longer filter is required.

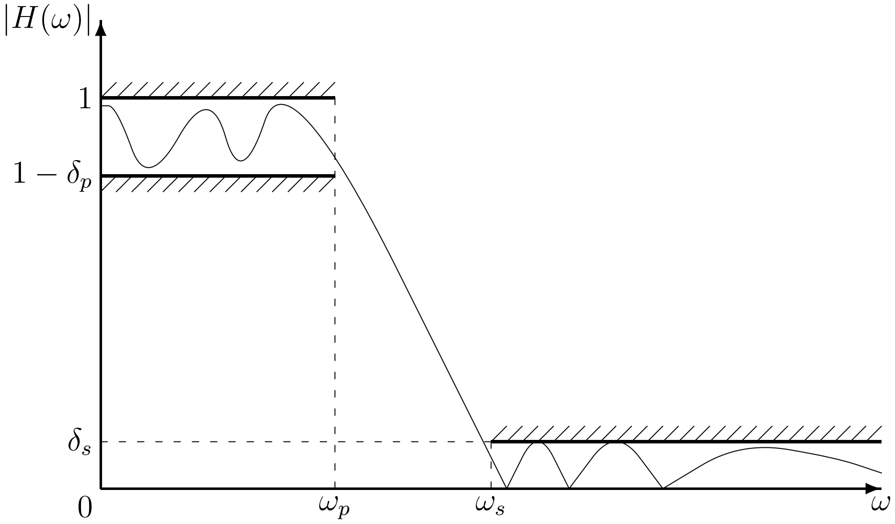

Figure 2-4: The filter specification used in the thesis.

By taking the Fourier transform of (2.3) it can be shown that

$$H(\omega) = B_k(\omega\epsilon)G(\omega\epsilon), \tag{2.15}$$

where $B_k(\omega)$ is the Fourier transform of $b_k(t)$, given by

$$B_k(\omega) = \left(\operatorname{sinc}\frac{\omega}{2\pi}\right)^{k+1}, \tag{2.16}$$

and $G(\omega)$ is the DT Fourier transform of the sequence $g[n]$.

If the parameter ϵ is not sufficiently small, then there may not exist a DT filter $g[n]$ which allows $h(t)$ to meet the given specification. Ramstad showed that for larger values of k, we may allow ϵ to be larger. His argument was based on the requirement that the error which is introduced be small compared to the step size of the quantization used in the digital representation of the signal involved. Here we allow for arbitrary precision in the digital representation used and we show that it is still possible to choose ϵ so that $h(t)$ is able to meet the specification.

$G(\omega)$ is the DT Fourier transform of a real sequence $g[n]$, and must therefore satisfy certain symmetry and periodicity properties. In fact, if we know $|G(\omega)|$ over the interval $[0, \pi]$ then we can determine it everywhere. These symmetry and periodicity

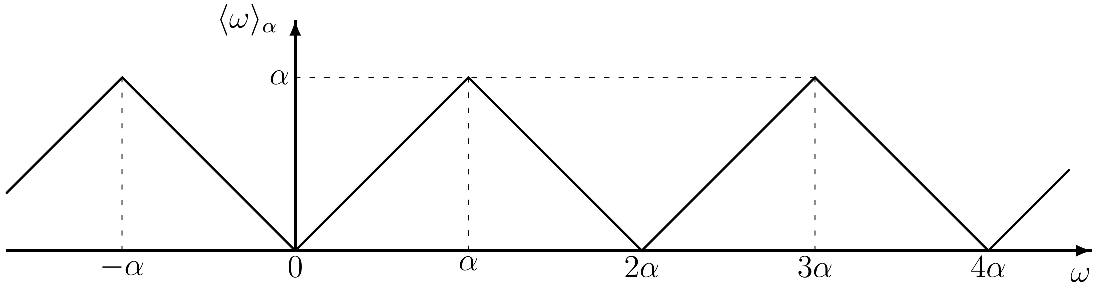

Figure 2-5: Plot of $\langle\omega\rangle_\alpha$ versus ω.

conditions are described by the relationship

$$|G(\omega)| = \left|G\big(\langle\omega\rangle_\pi\big)\right|, \tag{2.17}$$

where the operator $\langle\cdot\rangle_\alpha$ is defined by

$$\langle\omega\rangle_\alpha = \left|\omega - 2\alpha\left\lfloor\frac{\omega+\alpha}{2\alpha}\right\rfloor\right|. \tag{2.18}$$

A plot of $\langle\omega\rangle_\alpha$ as a function of ω is shown in figure 2-5.

By using (2.15), (2.16) and (2.17), it can be shown that

$$|H(\omega)| = \left|\frac{\langle\omega\rangle_{\frac{\pi}{\epsilon}}}{\omega}\right|^{k+1} |H(\langle\omega\rangle_{\frac{\pi}{\epsilon}})|. \tag{2.19}$$

This equation describes how $|H(\omega)|$ at some arbitrary frequency is related to $|H(\omega)|$ over $[0, \pi/\epsilon]$.

If we choose ϵ so that

$$\epsilon = \frac{2\pi\delta_s^{\frac{1}{k+1}}}{\omega_p\left(1+\delta_s^{\frac{1}{k+1}}\right)}, \tag{2.20}$$

and we impose the additional restriction on $|H(\omega)|$ that

$$|H(\omega)| \leq \delta_s\left|\frac{\frac{2\pi}{\epsilon}-\omega}{\omega}\right|^{k+1} \quad ; \quad \omega \in (\omega_p, \omega_s), \tag{2.21}$$

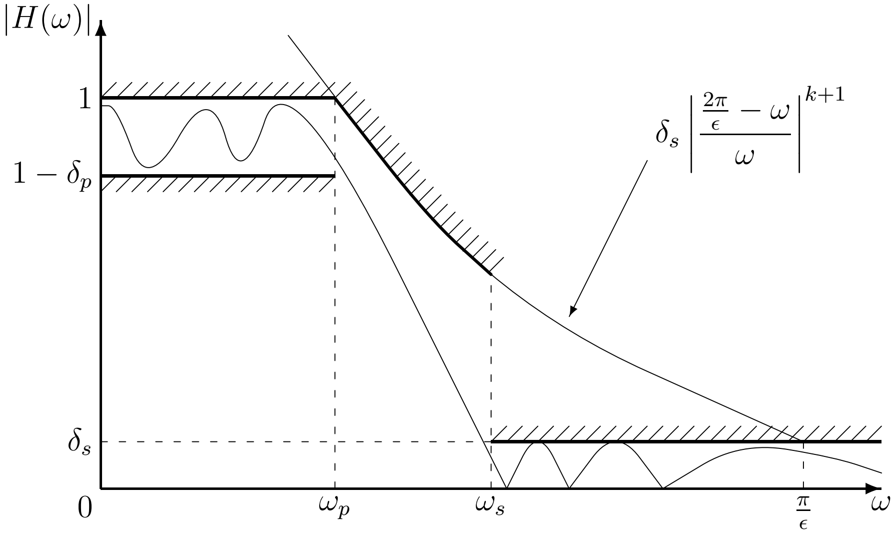

Figure 2-6: Filter specification with an additional constraint.

ω_p	0.9π	2.82734
ω_s	1.1π	3.45575
δ_p	1 dB allowed passband ripple	0.10875
δ_s	50 dB attenuation in stopband	0.00316

Table 2.2: Filter specification parameters used in our example.

then (2.13) and (2.14) will only need to be satisfied over $[0, \pi/\epsilon]$. It is then guaranteed that it will be satisfied everywhere. This can be shown directly from (2.19). The specification which includes this third constraint on $|H(\omega)|$ is shown in figure 2-6. We have found that in practice that this additional constraint (2.21) is sometimes not required, since filters designed to satisfy (2.13) and (2.14) will typically automatically satisfy (2.21).

2.4 Design Example

We now look at an example in which the parameters describing the specification are as shown in table 2.2

Standard filter design techniques such as the Parks-McClellan (PM) algorithm

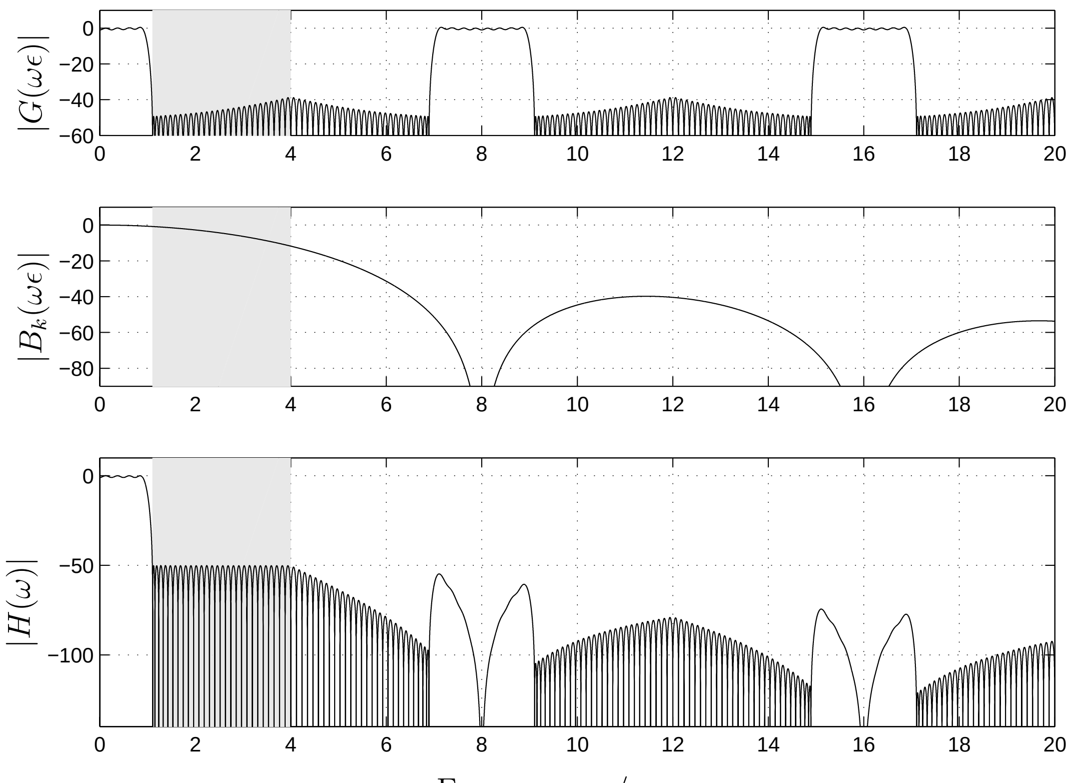

Figure 2-7: $|H(\omega)|$ and its factors $|G(\omega\epsilon)|$ and $|B_k(\omega\epsilon)|$.

may be used to design $g[n]$. This is done so that $|H(\omega)|$ will meet the given specification. When determining the target response and error weighting function which are used by the design algorithm, we must compensate for the effect of $B_k(\omega)$ in (2.15). The PM algorithm is usually used to design filters which have generalized linear phase. However we may also use the PM algorithm to design filters with an unrestricted phase response. We may require these filters to be minimum phase without loss of generality, since for any filter with an unconstrained phase response there exists a minimum phase filter of the same length which has the same magnitude response. Minimum phase filters are designed by performing spectral factorization [34] on linear phase filters. In this case, the linear phase filter is designed to have a magnitude response which satisfies the square of the specification. A minimum phase filter designed in this way will always be shorter than the equivalent linear phase filter, and thus will require less computation.

An actual design example for $k = 2$ is shown in figure 2-7. As we can see from

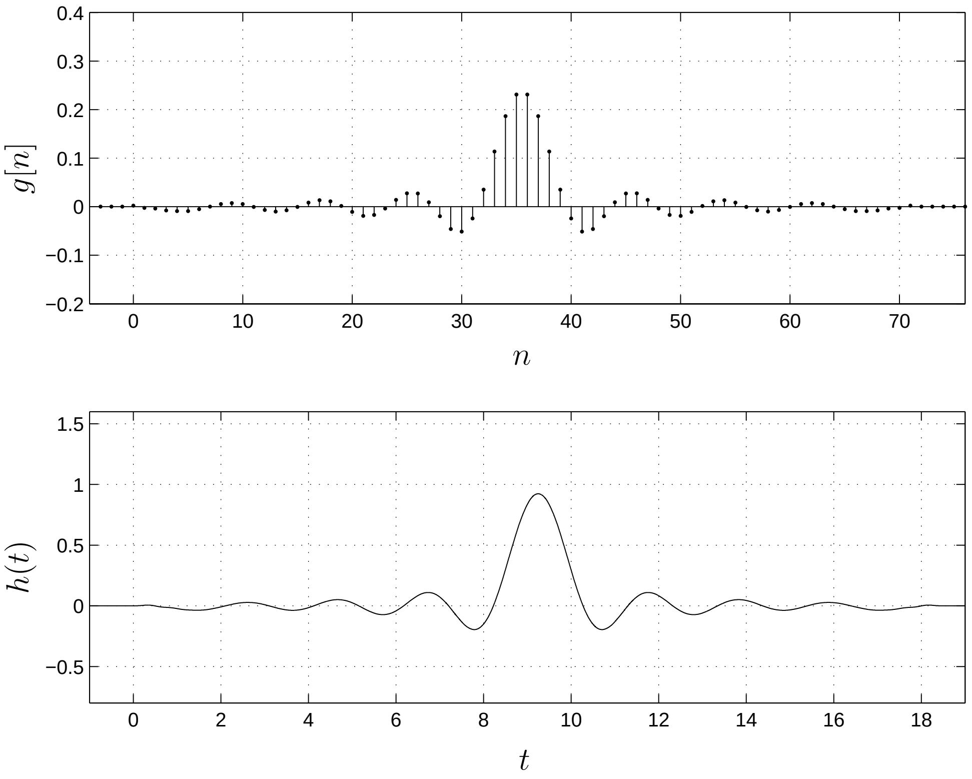

Figure 2-8: The filters $g[n]$ and $h(t)$. ($k = 2$, linear phase)

the figure, the stopband ripple for $|G(\omega\epsilon)|$ follows a curve which is exactly reciprocal to $|B_k(\omega\epsilon)|$ over the interval indicated by the shaded region. Thus, $|H(\omega)|$ will be equiripple over this interval. The corresponding filters $g[n]$ and $h(t)$ are shown in figure 2-8. An example in which $k = 1$ and $h(t)$ is minimum phase is given in figure 2-9.

2.4.1 Performance

Optimal filters were designed for different values of k. Table 2.3 shows the required length of $g[n]$ in each case. In the table, we have also given the length of the interval of support for the filter $h(t)$. We have chosen ϵ so that $1/\epsilon$ will be an integer. Though this is not necessary it greatly simplifies the implementation of our algorithm for the case in which one grid is regular with an assumed spacing of one. We will look at this special case in the next section. In table 2.3, we have given the filter lengths for both linear phase and minimum phase designs. The filter lengths for $k = 0$ are given

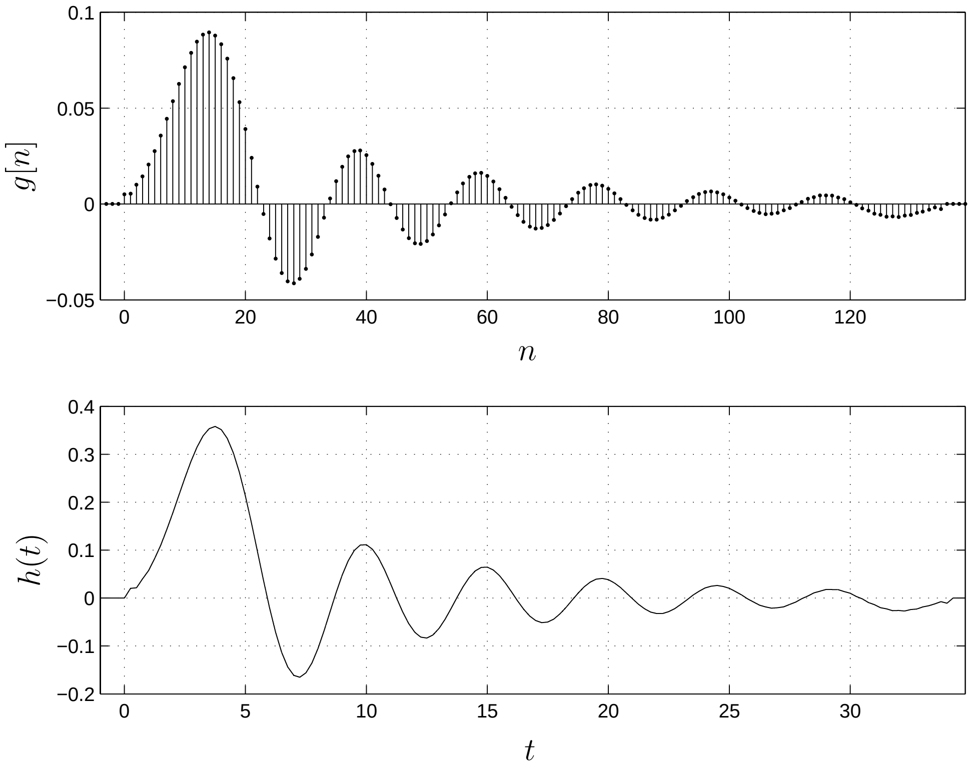

Figure 2-9: The filters $g[n]$ and $h(t)$. ($k = 1$, minimum phase)

in parentheses because these numbers are only predicted values. We were not able to design the filters in this case since the particular filter design tool which was being used was is not able to design optimal filters longer than about 1500 points.

Figure 2-10 plots the performance of our algorithm for the specification given in table 2.2. Computation is measured in multiplies per output sample (MPOS), while memory is measured in terms of the number of constants which we need to store. In this case, the constants are simply the polynomial coefficients used in the

order of polynomials, k	0	1	2	3	4	5	6
$1/\epsilon$	143	9	4	3	2	2	2
length of $g[n]$ (linear phase)	(2542)	160	72	54	36	36	36
length of $g[n]$ (minimum phase)	(2161)	136	60	45	30	29	30
length of $h(t)$ (linear phase)	(17.8)	17.9	18.5	19.0	20.0	20.5	21.0
length of $h(t)$ (minimum phase)	(15.1)	15.2	15.5	16.0	17.0	17.0	18.0

Table 2.3: Filter lengths for the general elementary resampler.

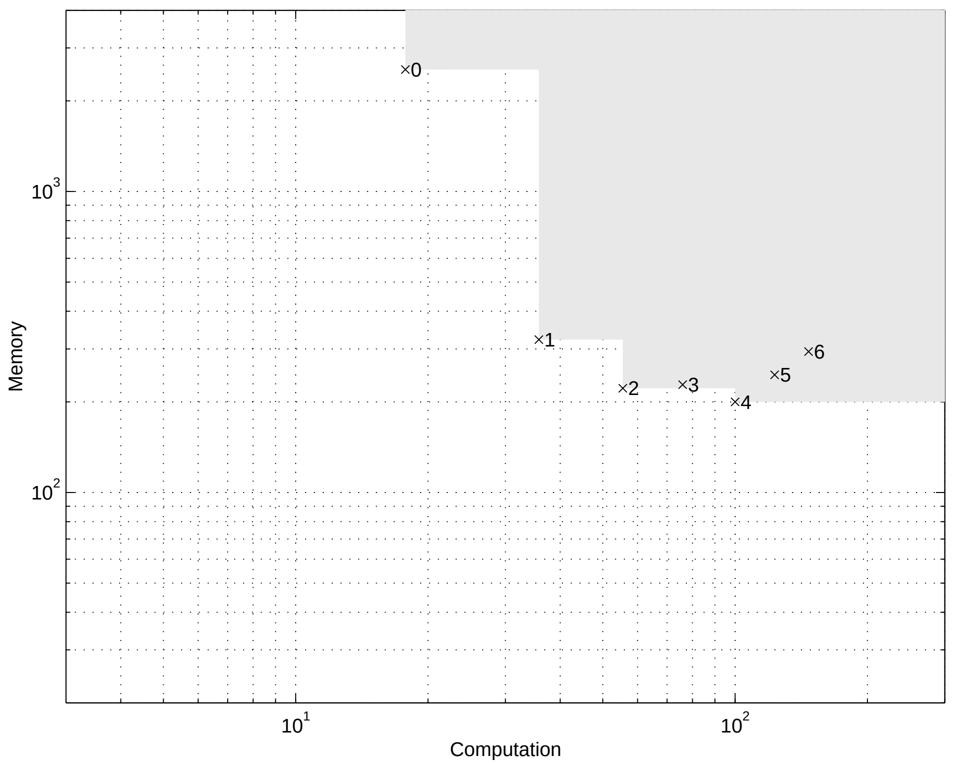

Figure 2-10: Performance plot for the elementary resampler (linear phase).

representation of $h(t)$. Each times sign ($\times$) indicates the amount of memory and computation needed by our algorithm and is labeled with the corresponding value of k. The gray shaded region represents all points in the memory-computation plane for which there is an algorithm that has strictly better performance. In other words, there is an algorithm that either uses less memory, or less computation, or less of both. In figure 2-10 we have only shown the case in which $h(t)$ is restricted to have linear phase. Notice that the examples for $k = 5$ and $k = 6$ have performance which is strictly worse than the performance for $k = 4$. In practice k is never usually chosen to be greater than 3. In figure 2-11 we show a similar plot in which $h(t)$ is minimum phase. The lighter shaded region from figure 2-10 is superimposed on this plot for ease of comparison.

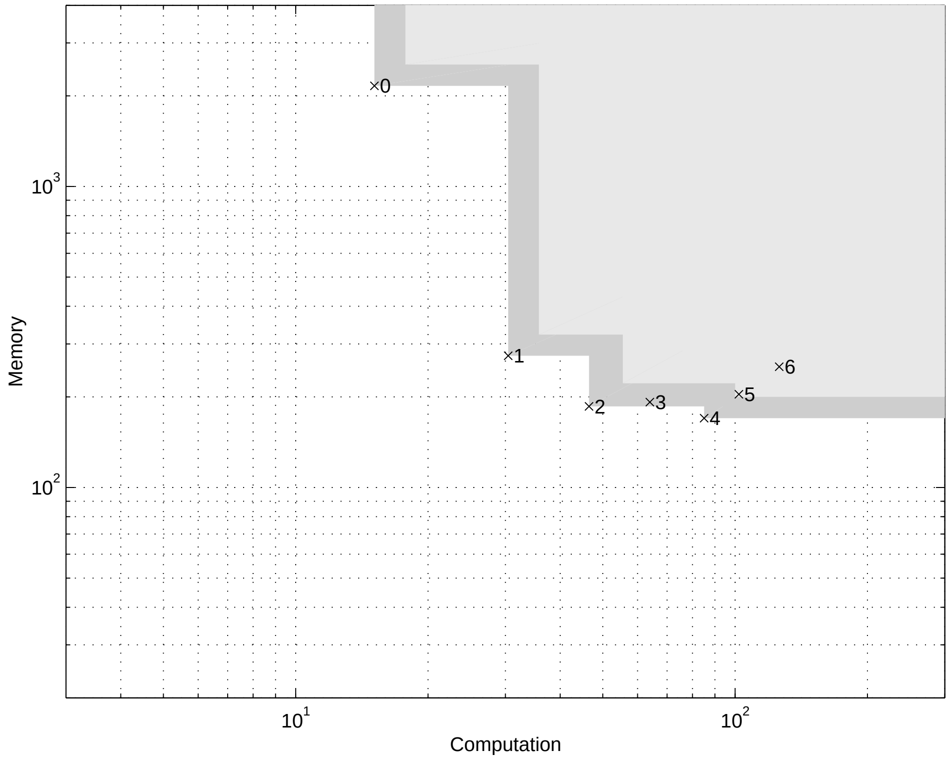

Figure 2-11: Performance plot for the elementary resampler (minimum phase).

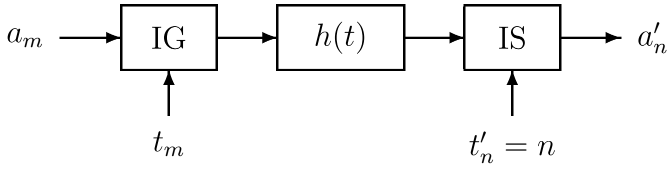

Figure 2-12: Elementary resampler with regular output grid.

2.5 System With One Regular Grid

In this section we consider a specialization of the elementary resampler in which at least one of the grids is regular. This case is particularly interesting because such a system is needed in chapters 4 and 5 where we discuss certain problems related to nonuniform sampling and reconstruction. We will assume that the output grid is regular, while the input grid remains unconstrained, as shown in figure 2-12. (The results which we derive here are equally applicable for the case in which the input grid is instead regular.) For notational convenience, we assume for the remainder of

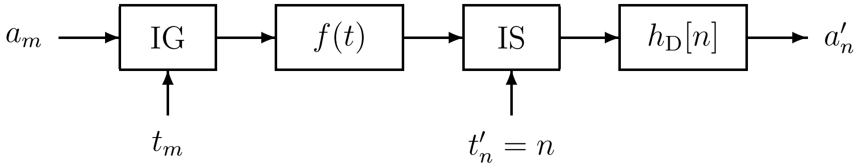

Figure 2-13: Elementary resampler with a fixed DT filter.

this chapter that the regular grid has an associated spacing of $T = 1$.

2.5.1 System With a Fixed DT Filter

In the context of sampling rate conversion, Martinez and Parks [20] and Ramstad [27, 28] have suggested introducing a fixed LTI filter which operates either on the input signal or the output signal. It has also been shown [28, 32] that if $h(t)$ in figure 2-12 is a proper rational CT filter, then an equivalent system may be derived in which a fixed (time-invariant) filter is introduced at the input or the output which allows the CT filter to be FIR. In the context of rational ratio sampling rate conversion using an IIR filter, fixed filter(s) at the input or output may be introduced in order to increase efficiency [30]. In almost all the cases considered, the DT filter which is introduced is an all-pole IIR filter. In this section, we do not restrict the DT filter to be IIR, and as we show, this still leads to an improvement in performance.

With reference to figure 2-12, we have assumed that the grid associated with the output signal is regular, and therefore we may introduce an LTI filter at the output as depicted in figure 2-13. This LTI filter can be implemented directly in discrete time using the convolution sum, or an appropriate fast convolution algorithm. We denote the CT filter (which must be FIR) as $f(t)$, and the DT filter as $h_{\mathrm{D}}[n]$. For any choice of $f(t)$ and $h_{\mathrm{D}}[n]$, there is a filter, $h(t)$ such that the systems of figures 2-12 and 2-13 will be equivalent. The reason we introduce this DT filter is to allow the CT filter to have a shorter interval of support. That way, fewer coefficients will need to be calculated in real time, and thus we will be able to improve the performance of the resampler.

We will first focus our attention on minimizing the length of the CT filter $f(t)$.

By so doing, we may use the computation and memory requirements associated with implementing $f(t)$ as a lower bound on the overall performance of our algorithm.

2.5.2 An Implicit Constraint on $f(t)$

The frequency response specification is given in terms of the over all filter $h(t)$, and implies certain constraints on $f(t)$ and $h_{\mathrm{D}}[n]$. The constraints on one filter can be expressed in terms of the other filter. In this section we derive an implicit constraint on the frequency response of $f(t)$ which is independent of the filter $h_{\mathrm{D}}[n]$. Assume that we are given a lowpass specification as depicted in figure 2-4, with $\omega_p < \pi$.

It can be shown by writing out the convolution sum for $h_{\mathrm{D}}[n]$, that the systems in figures 2-12 and 2-13 are equivalent if

$$h(t) = \sum_{n=-\infty}^{\infty} h_{\mathrm{D}}[n] f(t-n). \tag{2.22}$$

By taking the Fourier transform, we get that

$$|H(\omega)| = |F(\omega)| \cdot |H_{\mathrm{D}}(\omega)|, \tag{2.23}$$

where $H_{\mathrm{D}}(\omega)$ is the DT Fourier transform of the sequence $h_{\mathrm{D}}[n]$, and is therefore periodic in ω. We will use the same angle bracket notation defined by (2.18) to get

$$|H_{\mathrm{D}}(\omega)| = |H_{\mathrm{D}}(\langle\omega\rangle_\pi)|. \tag{2.24}$$

If the overall magnitude response $|H(\omega)|$ satisfies the conditions implied by the given specification, *i.e.*, (2.13) and (2.14), then it can be shown from (2.14) and (2.23) that

$$|F(\omega)| \leq \frac{\delta_s}{|H_{\mathrm{D}}(\langle\omega\rangle_\pi)|}; \quad \text{for } \omega \geq \omega_s \text{ and } \langle\omega\rangle_\pi \leq \omega_p. \tag{2.25}$$

Since we wish to obtain a constraint on $|F(\omega)|$, which does not depend on $|H_{\mathrm{D}}(\omega)|$, we use the fact that

$$|F(\omega)H_{\mathrm{D}}(\omega)| \leq 1; \quad 0 \leq \omega \leq \omega_p \tag{2.26}$$

to impose the tighter constraint

$$|F(\omega)| \leq \delta_s |F(\langle\omega\rangle_\pi)|; \quad \text{for } \omega \geq \omega_s \text{ and } \langle\omega\rangle_\pi \leq \omega_p. \tag{2.27}$$

This allows us to separate the design of $f(t)$ and $h_{\mathrm{D}}[n]$, *i.e.*, $f(t)$ may be designed with out regard for $h_{\mathrm{D}}[n]$. There is no great loss in using (2.27) instead of (2.25), since $\delta_s|F(\langle\omega\rangle_\pi)|$ is quite close to $\delta_s/|H_{\mathrm{D}}(\langle\omega\rangle_\pi)|$. In fact, the former is never smaller than $1-\delta_p$ times the latter over the specified range.

2.5.3 Design of $f(t)$

In this section, we discuss the filter design algorithm which we use to design $f(t)$ so that it satisfies (2.27). The design algorithm is inspired by algorithms presented by Martinez and Parks [19, 20], Ramstad [27] and Göckler [9] which are used in designing optimal filters for rational ratio sampling rate conversion or straightforward lowpass filtering. These algorithms design equiripple filters which are composed of two DT filters in cascade, possibly operating at different rates. One of these filters is FIR and the other is all-pole IIR. These algorithms work by alternating between optimizing the second filter while the first is fixed, and optimizing the first filter while the second is fixed. In all cases tested, the algorithms were found to converge, though no proof of convergence or optimality was given[1]. Our algorithm is of a similar nature, but is used to design a single filter which obeys the implicit constraint (2.27).

We start with a filter $f_1(t)$ which is designed so that

$$|F_1(\omega)| \leq \delta_1; \quad \text{for } \omega \geq \omega_s \text{ and } \langle\omega\rangle_\pi \leq \omega_p, \tag{2.28}$$

[1] It was however found in all cases tested that if the two filters operate at the same rate and are of the same order, then the algorithm converges to an elliptic filter, which is known to be optimal.

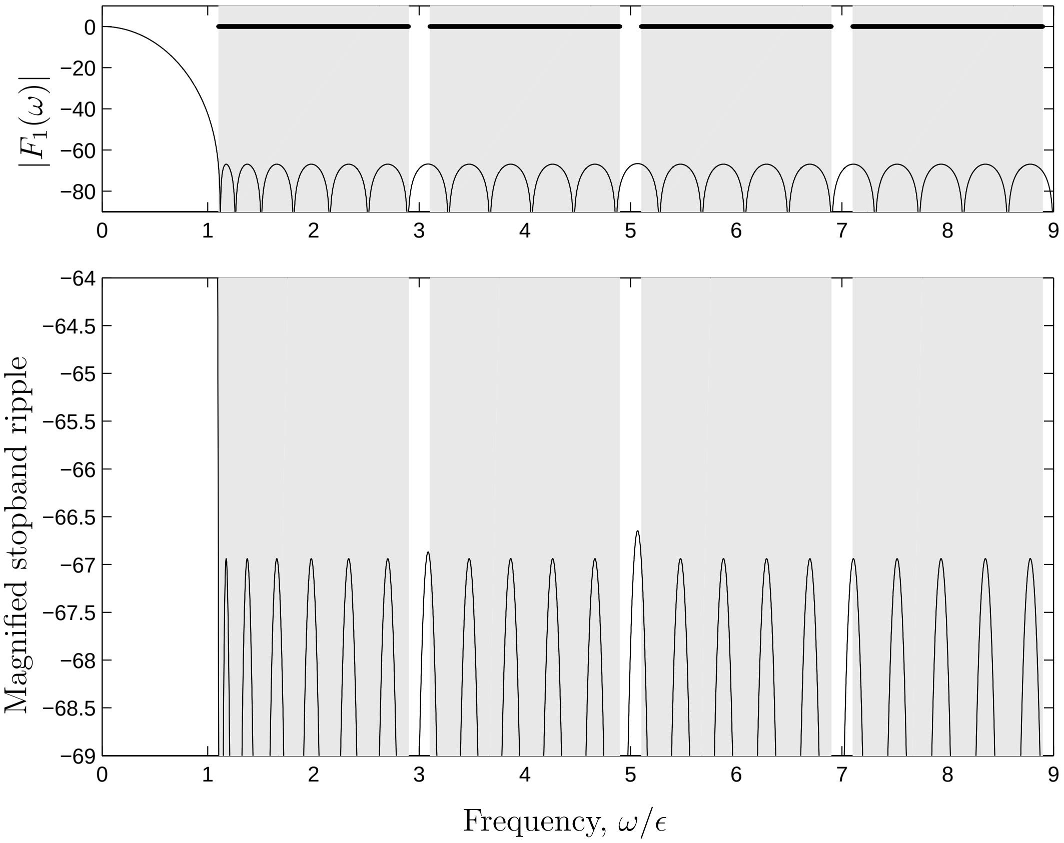

Figure 2-14: Frequency response $|F_1(\omega)|$, and magnified stopband.

where δ_1 is minimized subject to the constraint that $|F_1(0)| = 1$. This results in a filter which is equiripple over certain portions of the stopband, as shown in figure 2-14. The filter is only equiripple over the bands that are shaded as we can see in the magnified lower plot. The thick horizontal lines in the upper plot are used to indicate the shape of the desired stopband ripple (in this case, flat).

Note that we make the same assumptions about the structure of $f(t)$ as we did for $h(t)$ in section 2.2, and are thus able to use DT filter design algorithms to design $f(t)$ as described in section 2.3.

The next step is to design a new filter $f_2(t)$ so that

$$|F_2(\omega)| \leq \delta_2 |F_1(\langle\omega\rangle_\pi)|; \quad \text{for } \omega \geq \omega_s \text{ and } \langle\omega\rangle_\pi \leq \omega_p, \tag{2.29}$$

where δ_2 is minimized subject to $|F_2(0)| = 1$. This new filter has more attenuation where $|F_1(\langle\omega\rangle_\pi)|$ is smaller and less where it is bigger, as shown in figure 2-15. The

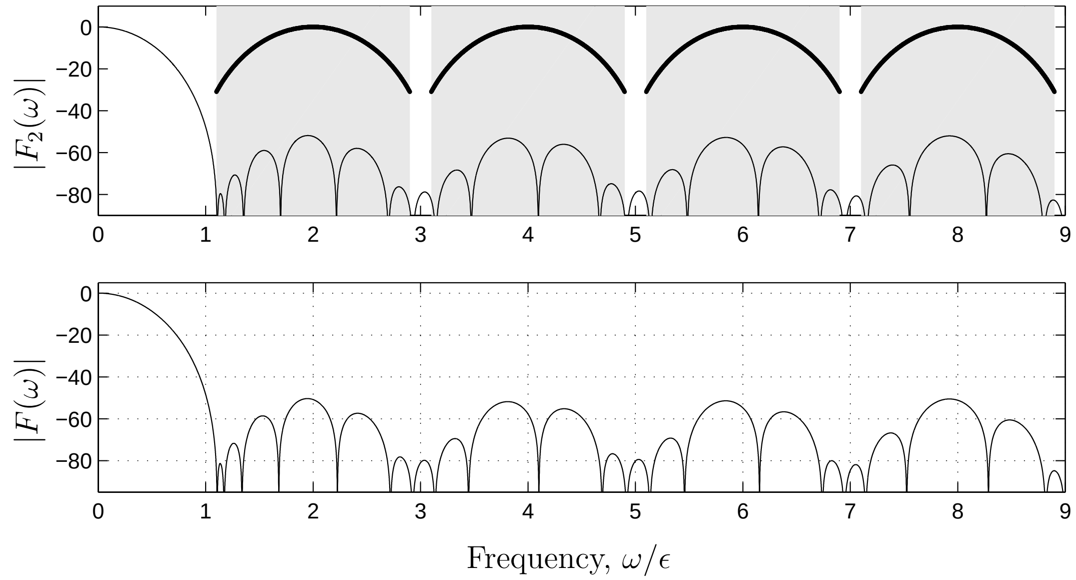

Figure 2-15: Frequency responses $|F_2(\omega)|$, and $|F(\omega)|$.

thick lines indicate the desired shape of the stopband ripple and are obtained from the passband of $|F_1(\omega)|$.

We then repeat the process, with each successive filter $f_n(t)$ being designed so that

$$|F_n(\omega)| \leq \delta_n |F_{n-1}(\langle\omega\rangle_\pi)|; \quad \text{for } \omega \geq \omega_s \text{ and } \langle\omega\rangle_\pi \leq \omega_p, \tag{2.30}$$

where δ_n is minimized subject to $|F_n(0)| = 1$. For the cases which were tested, the algorithm required no more than ten iterations in order to converge to within machine precision. In fact, the time taken to design each filter in the sequence decreases at each step since the previous filter can be used as an initial guess for our optimization. Thus the time required for the algorithm to converge is not significantly greater than the time required to design the first filter $f_1(t)$. The frequency response of the filter $f(t)$ obtained in this example is shown in figure 2-15, and we see that it is almost identical to the response of $f_2(t)$. We have not proven that the resulting filter is optimal, but the filters which we have obtained using this method were found to have very good performance.

2.5.4 Specification and design of $h_{\mathrm{D}}[n]$

After we have found $f(t)$, which must be FIR, we need to design $h_{\mathrm{D}}[n]$ so that the overall specification on the combined filter $h(t)$ is satisfied. We assume at this point that $f(t)$ is fixed, and thus it can be incorporated into the requirement on $h_{\mathrm{D}}[n]$.

The constraint on $f(t)$ given by (2.27) guarantees that if $|H(\omega)|$ satisfies the specification for $|\omega| \leq \omega_p$, then it will also satisfy the specification wherever $\langle\omega\rangle_\pi \leq \omega_p$. Thus we will design $h_{\mathrm{D}}[n]$ so that the overall response $|H(\omega)|$ will satisfy the specification in two cases. Firstly where $|\omega| \leq \omega_p$; and secondly, where $\langle\omega\rangle_\pi > \omega_p$. These two conditions are satisfied respectively by the conditions

$$\frac{1-\delta_p}{|F(\omega)|} \leq |H_{\mathrm{D}}(\omega)| \leq \frac{1}{|F(\omega)|}; \quad \text{for } |\omega| \leq \omega_p, \tag{2.31}$$

and

$$|H_{\mathrm{D}}(\omega)| \leq \min_{\substack{\alpha \\ \langle\alpha\rangle_\pi = \omega, \alpha \geq \omega_s}} \frac{\delta_s}{|F(\alpha)|}; \quad \text{for } \omega_p < \omega \leq \pi. \tag{2.32}$$

We will now verify that if $|F(\omega)|$ satisfies (2.27) and $|H_{\mathrm{D}}(\omega)|$ satisfies (2.31) and (2.32), then $|H(\omega)|$ is guaranteed to satisfy (2.13) and (2.14). We can see that $|H(\omega)|$ satisfies (2.13) by substituting (2.31) into (2.23). In order to confirm that (2.14) is satisfied for any frequency $\omega_0 \geq \omega_s$, we consider two cases:

1. $\langle\omega_0\rangle_\pi \leq \omega_p$. In this case, (2.27) applies. Also we have that

$$|H_{\mathrm{D}}(\omega_0)| \leq \frac{1}{|F(\langle\omega_0\rangle_\pi)|}; \quad \text{for } \langle\omega_0\rangle_\pi \leq \omega_p \tag{2.33}$$

by substituting (2.24) into (2.31). By combining (2.33) with (2.27) and (2.23), we get that

$$|H(\omega_0)| = |F(\omega_0)| \cdot |H_{\mathrm{D}}(\omega_0)| \leq \frac{|F(\omega_0)|}{|F(\langle\omega_0\rangle_\pi)|} \leq \delta_s; \quad \text{for } \langle\omega_0\rangle_\pi \leq \omega_p. \tag{2.34}$$

2. $\langle\omega_0\rangle_\pi > \omega_p$. In this case, by (2.32) and (2.24) we get that

$$|H_{\mathrm{D}}(\omega_0)| \leq \inf_{\substack{\alpha \\ \langle\alpha\rangle_\pi = \langle\omega_0\rangle_\pi, \alpha \geq \omega_s}} \frac{\delta_s}{|F(\alpha)|}; \quad \text{for } \langle\omega_0\rangle_\pi > \omega_p. \tag{2.35}$$

Now, since $\omega_0 = \alpha$ implies that $\langle\omega_0\rangle_\pi = \langle\alpha\rangle_\pi$, we have that

$$|H_{\mathrm{D}}(\omega_0)| \leq \frac{\delta_s}{|F(\omega_0)|}; \quad \text{for } \langle\omega_0\rangle_\pi > \omega_p, \tag{2.36}$$

or

$$|H(\omega_0)| \leq \delta_s; \quad \text{for } \langle\omega_0\rangle_\pi > \omega_p. \tag{2.37}$$

Once $f(t)$ has been found, $h_{\mathrm{D}}[n]$ can be designed to meet (2.31) and (2.32) by using the PM algorithm with an appropriate target response and error weighting function. For the $|F(\omega)|$ shown in figure 2-15, we plot the $|H_{\mathrm{D}}(\omega)|$ which was found, and the combined response $|H(\omega)|$ in figure 2-16. The corresponding time-domain signals are shown in figure 2-17, where we have required that the filters have linear phase, and $k = 1$. A second example for $k = 3$ is shown in figure 2-18, where the minimum phase solution is given.

2.5.5 Performance Plots

$f(t)$ is implemented as a time-varying DT filter, as shown in figure 2-19. As we discussed in section 2.4, there is an inherent tradeoff between computation and memory requirements for implementing this time-varying filter. This tradeoff is controlled by the parameter k, the order of the polynomials used to construct $f(t)$. By varying k we were able to design several filters for $f(t)$ which meet the specification given by table 2.2. A plot showing memory versus computation is given in figure 2-20. We have superimposed the shaded regions from figure 2-11 here for the purposes of comparison. Here we can see that for a given amount of memory, we can achieve an algorithm which uses less computation, and for a given amount of computation, we

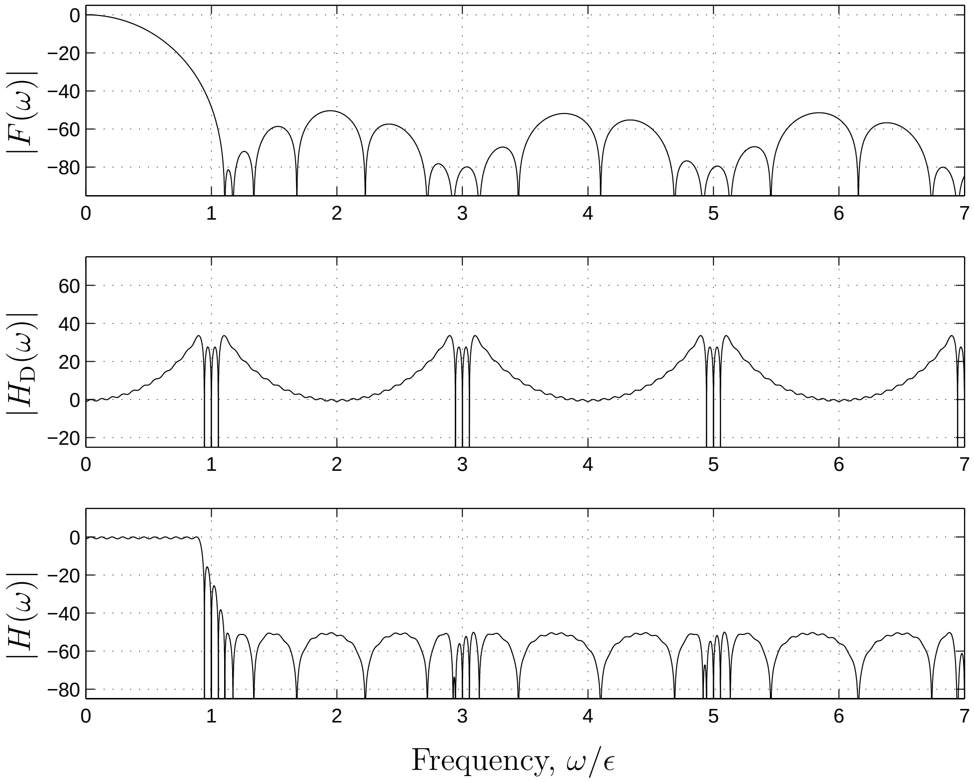

Figure 2-16: Frequency responses $|F(\omega)|$, $|H_{\mathrm{D}}(\omega)|$ and $|H(\omega)|$.

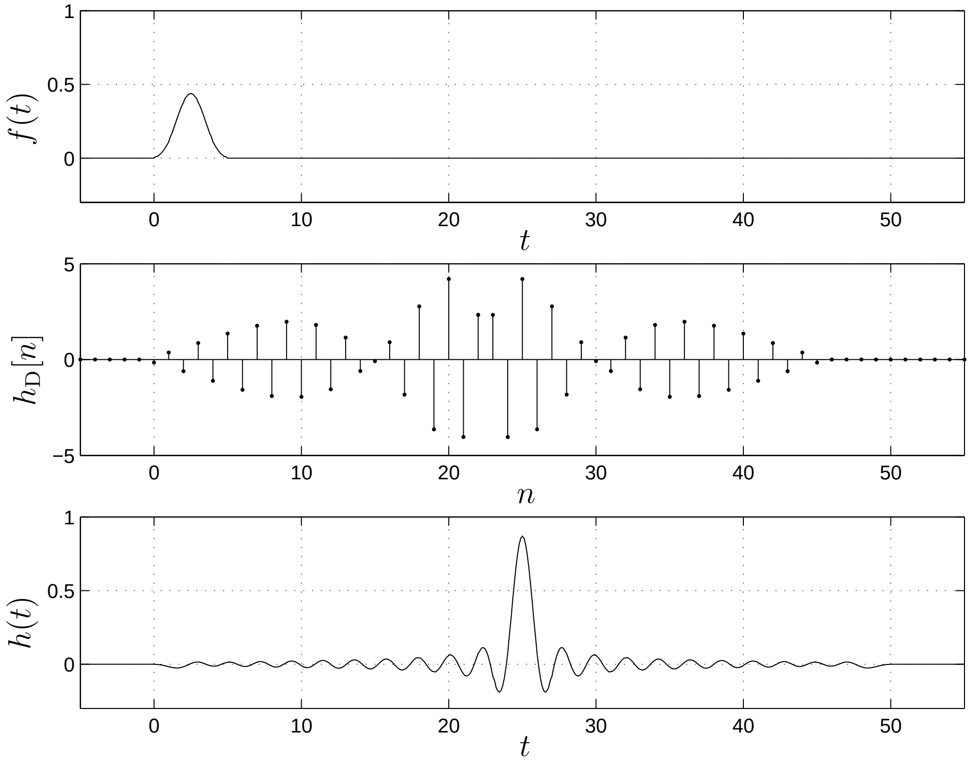

Figure 2-17: Filters $f(t)$, $h_{\mathrm{D}}[n]$ and $h(t)$ ($k = 1$, linear phase).

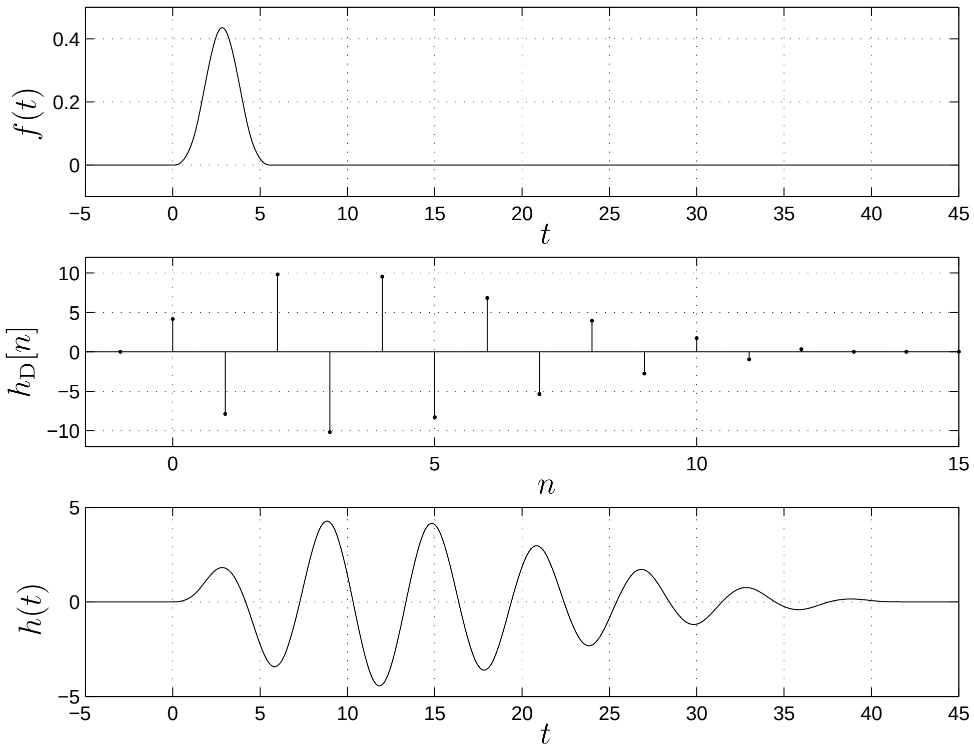

Figure 2-18: Filters $f(t)$, $h_{\mathrm{D}}[n]$ and $h(t)$ ($k = 3$, minimum phase).

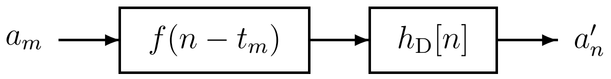

Figure 2-19: $f(t)$ as a time-varying DT filter.

can achieve an algorithm which requires less memory.

This comparison is somewhat unfair because the performance of the new elementary resampler using only $f(t)$ is not the same as the performance of the old one which uses $h(t)$, unless the effect of the extra filter $h_{\mathrm{D}}[n]$ is taken into account. $h_{\mathrm{D}}[n]$ is implemented in discrete time, and it is therefore difficult to determine the exact computational and memory requirements, since there are several algorithms for implementing a DT filter. For example, there is the direct implementation which involves evaluating the convolution sum. There are also the so called fast FIR filtering algorithms some of which are presented in [22], and there are FFT based filtering techniques, such as the overlap-add and overlap-save methods (see [26]). There are even hybrid techniques which use a combination of FFT based methods, and direct

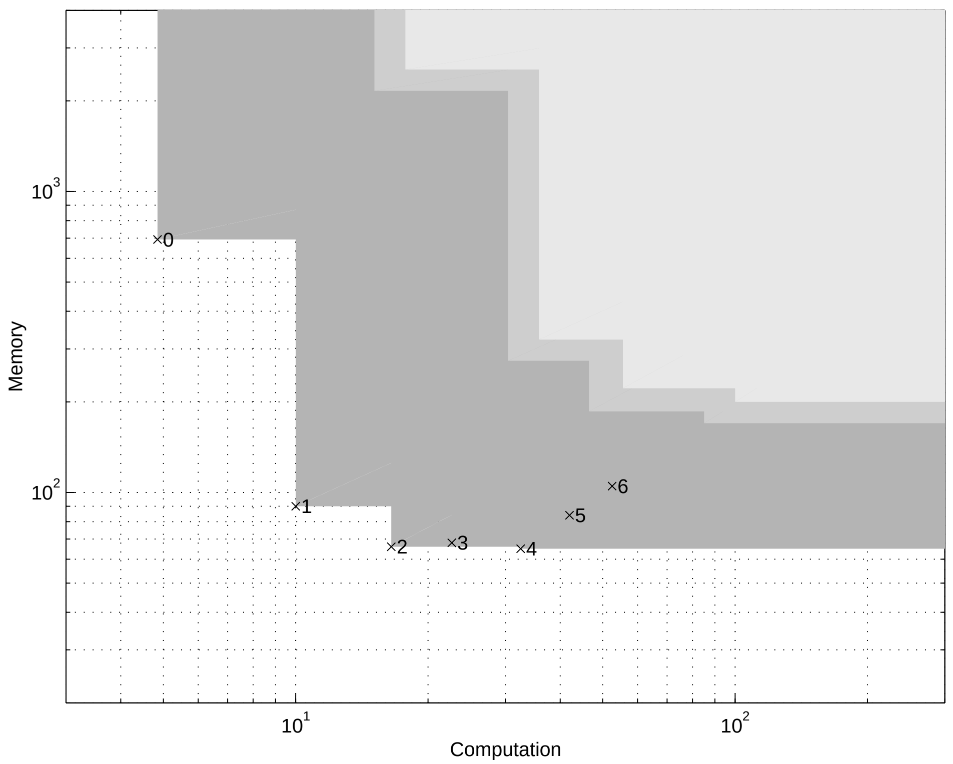

Figure 2-20: Performance plot for $f(t)$.

filtering. These implementations each have different computational and memory requirements and the best choice of which implementation to use is not always obvious. Furthermore, $h_\mathrm{D}[n]$ is no longer required to be FIR, because rational IIR DT filters can be implemented by means of a recursive finite difference equation. Even in the IIR case, there are various methods of implementing the filter. If the poles of the filter are implemented using a difference equation, the zeros may be implemented by any of the FIR techniques described above. Since any implementation of $h_\mathrm{D}[n]$ will require some computation and some memory, the performance plot shown shown in figure 2-20 essentially gives a lower bound on the actual performance of our algorithm. In figure 2-21 some points have been added to the plot which account for the computation and memory requirements of $h_\mathrm{D}[n]$. These points are marked with a times sign (×), and labeled with the corresponding value of k. In this example, $h_\mathrm{D}[n]$ was designed as the optimal linear-phase FIR filter. It is assumed that $h_\mathrm{D}[n]$ was implemented by directly evaluating the convolution sum. As seen from the plot, there can be a performance savings over the general elementary resampler even in this

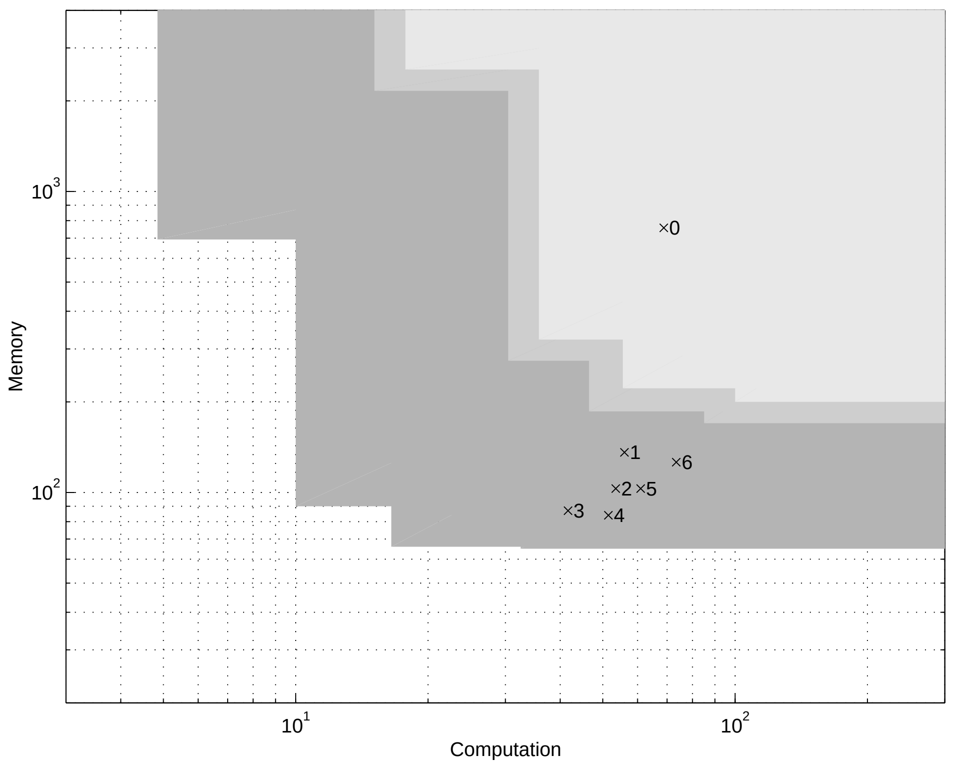

Figure 2-21: Achieved performance when $h_{\mathrm{D}}[n]$ is accounted for (linear phase).

case. This does not represent the best performance achievable, since choosing $h_{\mathrm{D}}[n]$ to be IIR or using some more efficient implementation could potentially improve the performance. However, in each case, we have an example of an achievable point on the memory-computation plane. In figure 2-22, we have shown a similar plot. The only difference being that $h_{\mathrm{D}}[n]$ is designed to be minimum phase. Table 2.4 gives the length of the filters which were designed for different values of k.

order of polynomials, k	0	1	2	3	4	5	6
length of $g[n]$	693	44	20	14	9	9	9
length of $f(t)$	4.8	5.0	5.5	5.7	6.5	7.0	7.5
length of $h_{\mathrm{D}}[n]$ (linear phase)	64	46	37	19	19	19	21
length of $h_{\mathrm{D}}[n]$ (minimum phase)	46	36	29	13	14	14	15

Table 2.4: Filter lengths for the elementary resampler with one regular grid.

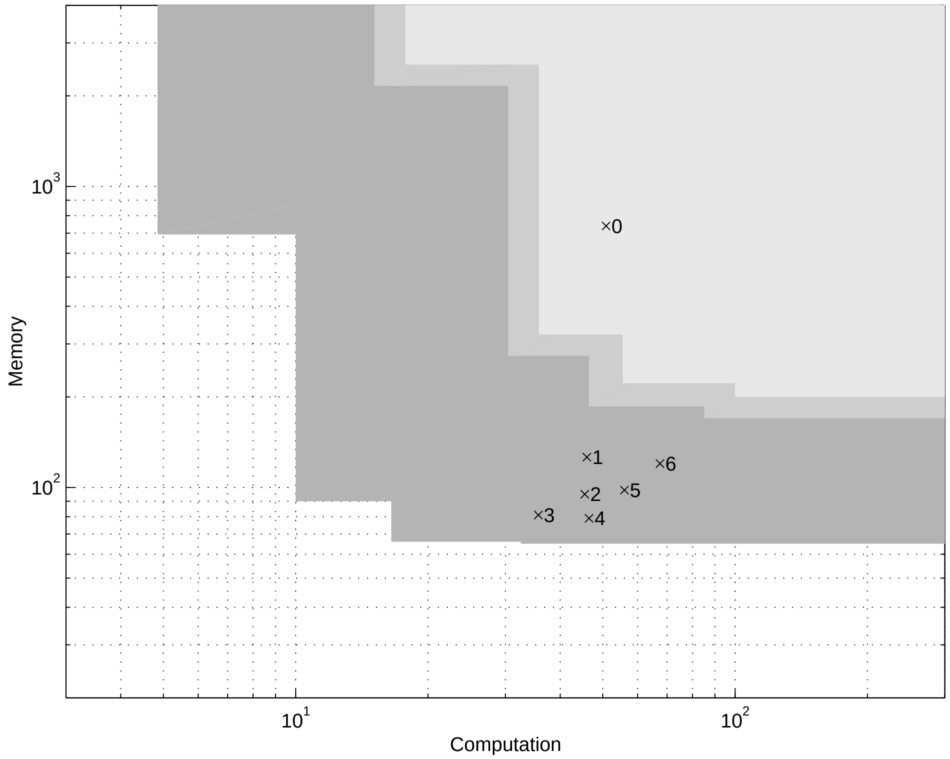

Figure 2-22: Achieved performance when $h_{\mathrm{D}}[n]$ is accounted for (minimum phase).

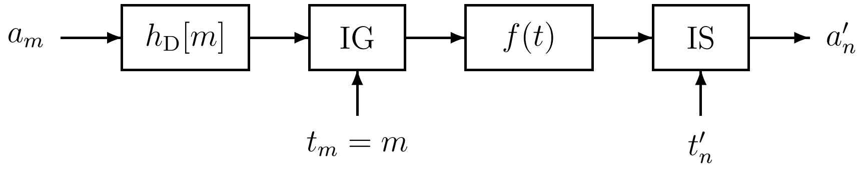

Figure 2-23: Elementary resampler with regular input grid.

2.5.6 System With a Regular Input Grid

So far when considering the elementary resampler with one regular grid, we have assumed that it is the output grid is regular. However, if the input grid is instead assumed to be regular, then it can be shown that the results derived are equally valid.

Assume that the input grid is regular and has a spacing of one second. In this case, the DT filter $h_{\mathrm{D}}[n]$ is introduced at the input, and operates directly on the input signal in discrete time. Thus we have the system shown in figure 2-23. By writing out the relationship between the input and the output, it is not difficult to show that the equivalent overall filter $h(t)$ can be expressed as

$$h(t) = \sum_{n=-\infty}^{\infty} h_{\mathrm{D}}[n] f(t-n), \tag{2.38}$$

or in the frequency domain as

$$|H(\omega)| = |F(\omega)| \cdot |H_{\mathrm{D}}(\omega)|. \tag{2.39}$$

This means that the filter design techniques presented in sections 2.5.3 and 2.5.4, and the performance plots given in section 2.5.5 all apply equally well in this case.

Chapter 3

Sampling Rate Conversion

3.1 Introduction

Digital audio is typically recorded at several different sampling rates. For example, compact discs (CDs) use a sampling rate of 44.1 kHz and digital audio tapes (DATs) use a sampling rate of 48 kHz. Sometimes, audio signals in these different formats may need to be combined and processed jointly. In that case, sampling rate conversion must be performed. A common method which is used involves a combination of up-sampling, filtering, and down-sampling [41]. This works well if the ratio of sampling rates is a rational number. In many cases, however, the sampling rates are not related by a rational number. This occurs frequently when trying to synchronize data streams which are arriving on separate digital networks. For example, in audio applications we may have a DT system which must output data at 44.1 kHz while the input data stream is arriving at a frequency which is slightly lower or higher because the clocks are not synchronized. In fact, the input data rate may be slowly varying in which case the receiving system must accommodate these variations. In some cases, it is possible to synchronize the processing system to the incoming data rate. In other applications, there may be multiple data streams at arbitrary sampling rates. In these cases, the conversion ratio cannot be assumed to be rational.

The problem of performing sampling rate conversion by a rational ratio has been studied in detail, *eg.*, in [5, 41, 30] (further references are given in [28]). Methods

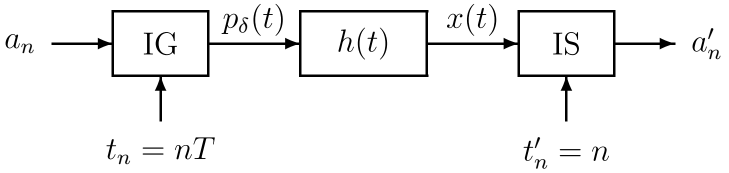

Figure 3-1: Generic sampling rate converter.

have also been proposed which allow for conversion by irrational ratios [33, 28]. These methods are based on the observation made in chapter 2, that a system which includes a continuous-time (CT) filter is equivalent to a time-varying DT filter. In these papers, various choices for the CT filter are proposed. In all cases, the authors conclude that there is a tradeoff between computational and memory requirements for generating the coefficients of the time-varying DT filter. The tradeoff can be seen in the design examples given in sections 2.4 and 2.5.5, and is characterized by the performance plots in figures 2-10, 2-11 and 2-20. In contrast, the algorithm presented in this chapter is computationally efficient while requiring very little memory. This is possible because the coefficients of the time-varying DT filter are calculated recursively[1].

We denote the input DT signal as $\{a_n, t_n\}$, and the output DT signal as $\{a'_n, t'_n\}$. The input and output grids are both regular and are given by

$$t_n = nT, \tag{3.1}$$

and

$$t'_n = nT' = n. \tag{3.2}$$

We have chosen $T' = 1$ without loss of generality.

The sample rate conversion system is depicted in figure figure 3-1, and can be viewed as a specialization of the elementary resampler with both input and output

[1] This work was done in collaboration with Paul Beckmann at Bose Corporation [32], and its use may be restricted by United States patent law [31]. The algorithm presented has since been implemented in real time by Bose Corporation, and is being used on one of their DSP-based audio products.

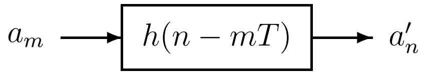

Figure 3-2: Sampling rate converter as a time-varying DT filter.

grids being regular. Thus a_n and a'_n are related by

$$a'_n = \sum_{m=-\infty}^{\infty} a_m h(n - mT). \tag{3.3}$$

This relationship can also be viewed as a time-varying DT filter, depicted in figure 3-2. The algorithms and filter design techniques described in chapter 2 may be directly applied to the sampling rate conversion problem.

In this chapter we choose the system function $H(s)$ to be rational. For example an elliptic, Butterworth, or Chebyshev lowpass filter. Ramstad [28] showed that in this case, the CT filter in figure 3-1 can be made to have finite support if a fixed recursive DT filter is introduced which operates at either the input or output sampling rates. Ramstad then showed that this finite-support filter can be implemented as described in chapter 2, by storing finely spaced samples of the impulse response. In this chapter, we show that by choosing the filter to be rational, we no longer need to store a table of samples of the impulse response. The required samples at any time-step can be calculated efficiently from the samples at the previous time-step.

If $H(s)$ is rational and proper, then $h(t)$ can be decomposed by means of a partial fraction expansion so that it can be expressed as the sum of first and second order filters. Thus we first consider the case in which $h(t)$ is either a first- or second-order filter (sections 3.2 and 3.3 respectively), and then we use these results to derive the general algorithm.

3.2 First-order Filter Approximation

Consider the case where $h(t)$ in figure 3-1 is a first-order filter. The impulse response is a decaying exponential which can be expressed in the form

$$h(t) = \alpha^t u(t). \tag{3.4}$$

where $u(t)$ is the CT unit-step function. By exploiting the structure of an exponential, we can express $h(t)$ as

$$h(t) = \sum_{n=0}^{\infty} h_{\mathrm{D}}[n] f(t-n), \tag{3.5}$$

where $f(t)$ is a finite duration function given by

$$f(t) = \alpha^t \big[u(t) - u(t-1)\big] = \alpha^t r(t), \tag{3.6}$$

and $h_{\mathrm{D}}[n]$ is the decaying exponential sequence

$$h_{\mathrm{D}}[n] = \alpha^n u[n]. \tag{3.7}$$

$u[n]$ is the DT unit step function and $r(t)$ is a unit-height rectangular pulse over the interval $[0, 1)$ and is defined as

$$r(t) = u(t) - u(t-1). \tag{3.8}$$

The decomposition of $h(t)$ according to (3.5) is illustrated in figure 3-3. As we showed in chapter 2, if $h(t)$ is expressed in this form, then we may represent the resampler with the block diagram of figure 3-4.

For any value of m, the function $f(n - mT)$ is non zero for exactly one value of n since $f(t)$ is nonzero only over the interval $[0, 1)$. The value of n for which $f(n - mT)$

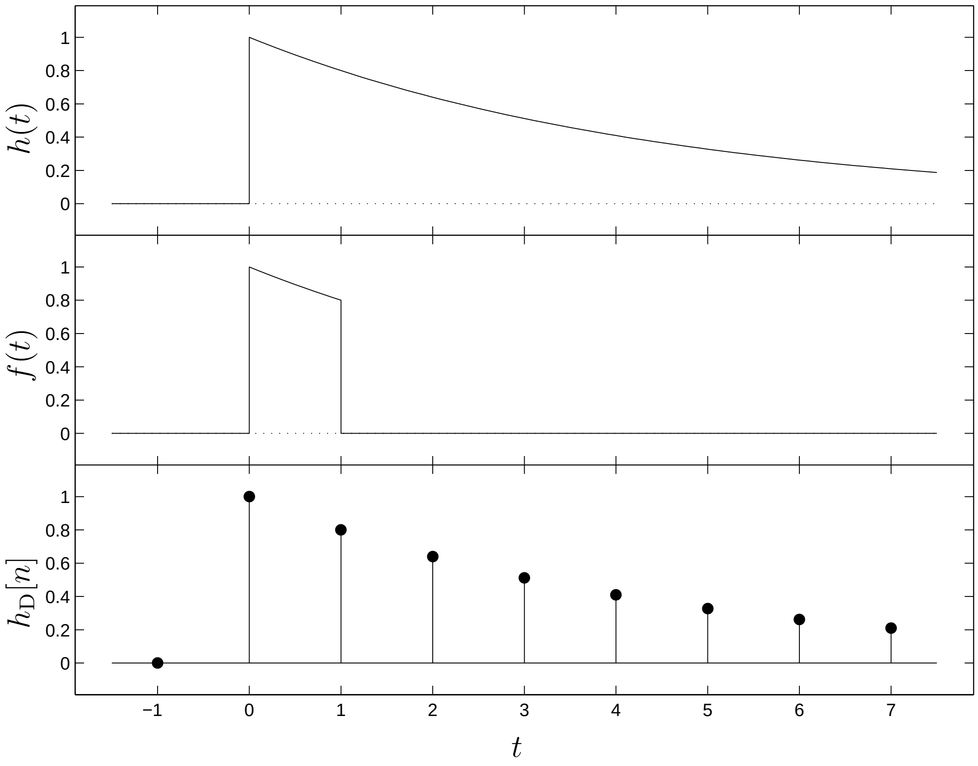

Figure 3-3: $f(t)$, $h_{\mathrm{D}}[n]$ and the equivalent filter $h(t)$ (first order case).

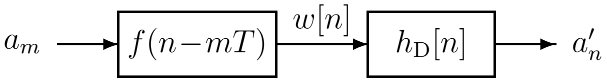

Figure 3-4: Sampling rate converter with FIR time-varying filter.

nonzero is

$$n = \lceil mT \rceil, \tag{3.9}$$

where $\lceil mT \rceil$ is the smallest integer greater than or equal to mT. Thus $f(n - mT)$ can then be rewritten as

$$f(n - mT) = f(\lceil mT \rceil - mT)\delta[n - \lceil mT \rceil]. \tag{3.10}$$

We can then express $w[n]$ as

$$w[n] = \sum_{m=-\infty}^{\infty} a_m e[m] \delta \left[n - \lceil mT \rceil \right], \tag{3.11}$$

or equivalently

$$w[n] = \sum_{\substack{m \\ n = \lceil mT \rceil}} a_m e[m], \tag{3.12}$$

where the coefficient $e[m]$ is given by

$$e[m] = f(\tau[m]) = \alpha^{\tau[m]}, \tag{3.13}$$

and

$$\tau[m] = \lceil mT \rceil - mT. \tag{3.14}$$

The summation in (3.11) can then be evaluated directly. For each term, we need to calculate $e[m]$ and perform a single multiplication. Since $e[n]$ is obtained by evaluating $f(t)$, an exponential function, this can be computationally demanding. However, a more efficient scheme exists which takes advantage of the specific shape of $f(t)$ and the fact that the points at which $f(t)$ is evaluated have a regular structure.

In order to show how to calculate $e[n]$ efficiently, we define

$$\begin{aligned} \Delta[m] &= \tau[m] - \tau[m-1] \\ &= \begin{cases} \lceil T \rceil - T, & \tau[m-1] < T - \lceil T \rceil + 1; \\ \lceil T \rceil - T - 1, & \tau[m-1] \geq T - \lceil T \rceil + 1. \end{cases} \end{aligned} \tag{3.15}$$

$\Delta[m]$ can take on only two possible values. We denote the non-negative value by Δ^+,

$$\Delta^+ = \lceil T \rceil - T \tag{3.16}$$

and the negative value by Δ^-,

$$\Delta^- = \lceil T \rceil - T - 1. \tag{3.17}$$

We use Δ to denote either Δ^+ or Δ^-. Since

$$\tau[m] = \tau[m-1] + \Delta, \tag{3.18}$$

at each time-step $\tau[m]$ either increases by a known quantity, Δ^+, or decreases by a known quantity, $|\Delta^-|$. In either case $0 \leq \tau[m] < 1$, for all m.

Comparing (3.13) and (3.18) it can be shown that

$$e[m] = E \cdot e[m-1], \tag{3.19}$$

where

$$E = \alpha^{\Delta}. \tag{3.20}$$

Since there are two possible values of Δ, there are two constants, E^+ and E^- corresponding to $\Delta = \Delta^+$ and $\Delta = \Delta^-$, which need to be calculated ahead of time and stored.

The final step in computing the output is to filter $w[n]$ by $h[n]$. $h[n]$ is given in (3.7) and can be implemented by the difference equation

$$a'_n = w[n] + \alpha a'_{n-1}. \tag{3.21}$$

In this section, we have described a very efficient algorithm for performing sampling rate conversion where $h(t)$, in figure 3-1, is a first-order filter. The equivalent DT system is shown in figure 3-4. The implementation of this system is summarized by (3.11), (3.19) and (3.21).

3.3 Second-order Filter Approximation

Assume that $h(t)$ in figure 3-1 is a second-order CT IIR filter. The impulse response is a decaying sinusoid which can be expressed in the form

$$h(t) = \gamma^t \sin(\theta t + \phi) u(t). \tag{3.22}$$

It can be shown that this is equivalent to

$$h(t) = \sum_{n=0}^{\infty} h_{\mathrm{D}}[n] f(t-n), \tag{3.23}$$

where $f(t)$ is a finite duration function given by

$$f(t) = \gamma^t \big\{ \sin(\theta t + \phi) r(t) - \sin(\theta(t-2) + \phi) r(t-1) \big\}, \tag{3.24}$$

and $h_{\mathrm{D}}[n]$ given by

$$h_{\mathrm{D}}[n] = \gamma^n \frac{\sin(\theta(n+1))}{\sin\theta} u[n] \tag{3.25}$$

is the impulse response of a second-order DT IIR filter. The decomposition of $h(t)$ in (3.23) is illustrated in figure 3-5. The derivation of this result is somewhat lengthy and not particularly interesting, and is thus omitted. With this choice for $f(t)$ and $h_{\mathrm{D}}[n]$, the system shown in figure 3-4 is still valid.

We see that $w[n]$ is related to a_m by

$$w[n] = \sum_{m=-\infty}^{\infty} a_m f(n - mT). \tag{3.26}$$

Since $f(t)$ is nonzero only over the interval $[0, 2)$, the function $f(n - mT)$ is non zero for exactly two values of n. These are $n = \lceil mT \rceil$ and $n = \lceil mT \rceil + 1$. We may

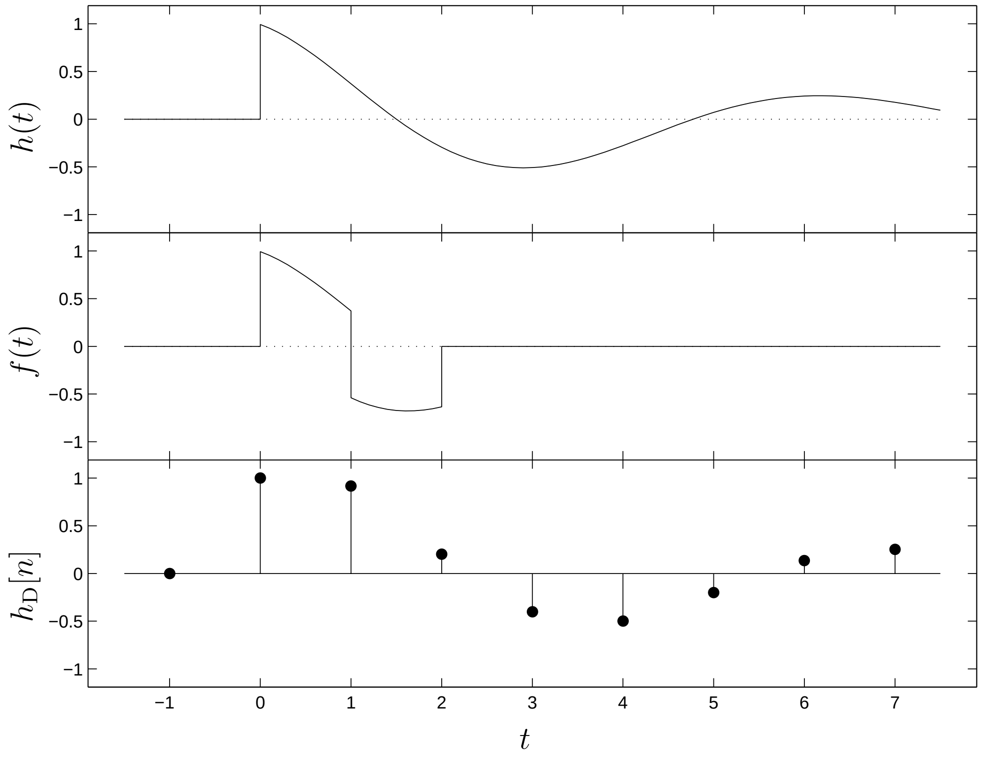

Figure 3-5: $f(t)$, $h_{\mathrm{D}}[n]$ and the equivalent filter $h(t)$ (second order case).

therefore express $w[n]$ in a form similar to (3.11), giving

$$w[n] = \sum_{m=-\infty}^{\infty} a_m \Big\{ c[m]\delta[n - \lceil mT \rceil] + d[m]\delta[n - \lceil mT \rceil - 1] \Big\}, \tag{3.27}$$

where

$$c[m] = f(\tau[m]) = \gamma^{\tau[m]} \sin(\theta\tau[m] + \phi), \tag{3.28}$$

and

$$d[m] = f(\tau[m] + 1) = -\gamma^{(\tau[m]+1)} \sin(\theta(\tau[m] - 1) + \phi). \tag{3.29}$$

$\tau[n]$ is defined by (3.14).

For the first-order case, we saw that the coefficient at each step could be calculated efficiently from the coefficient at the previous step by using (3.19). The corresponding equations for the second-order case are:

$$\begin{aligned} c[n] &= A \cdot c[n-1] + B \cdot d[n-1]; & (3.30) \\ d[n] &= C \cdot c[n-1] + D \cdot d[n-1], & (3.31) \end{aligned}$$

where

$$\begin{aligned} A &= \gamma^{\Delta} \left(\cos \Delta\theta + \sin \Delta\theta \cot \theta \right); & (3.32) \\ B &= \gamma^{(\Delta-1)} \frac{\sin \Delta\theta}{\sin \theta}; & (3.33) \\ C &= -\gamma^{(\Delta+1)} \frac{\sin \Delta\theta}{\sin \theta}; & (3.34) \\ D &= \gamma^{\Delta} \left(\cos \Delta\theta - \sin \Delta\theta \cot \theta \right), & (3.35) \end{aligned}$$

and Δ is given by (3.15).

Since A, B, C and D can be predetermined and stored, the coefficients at each step can be calculated from the coefficients at the previous step by performing only four multiplications. Since there are two possible values for Δ, two different values for the constants A, B, C and D need to be calculated and stored. These are denoted A^+, A^- *etc.*

It can be shown by using z-transform techniques that if $h_{\mathrm{D}}[n]$ is given by (3.25), then a'_n and $w[n]$ are related by the difference equation

$$a'_n = w[n] + (2\gamma \cos \theta) a'_{n-1} - \gamma^2 a'_{n-2}. \qquad (3.36)$$

The coefficients in this equation, $2\gamma \cos \theta$ and $-\gamma^2$ are constants which can be predetermined.

As in the first-order case, the second-order case can be implemented very efficiently by using the system shown in figure 3-4. The time-varying filter block represents our efficient evaluation of (3.27) which results from using (3.30) and (3.31) to determine the coefficients $c[n]$ and $d[n]$.

3.4 Nth-order Filter Approximation

In section 3.2 we saw how to efficiently simulate the system shown in figure 3-1 where $h(t)$ was chosen to be a first-order filter. In section 3.3, we saw how to do the same thing, with $h(t)$ chosen to be a second-order filter. In this section, we show that these two cases are sufficient to implement any filter with a rational, proper system function, provided all the poles are distinct.

$H(s)$ is assumed to have N distinct poles and up to $N-1$ zeros in the finite s-plane. A partial-fraction expansion can be performed in order to decompose this Nth-order filter into the sum of several first- or second-order filters. If there are N_r poles on the real axis, then there are N_p complex conjugate pole pairs, where $N_r + 2N_p = N$. In performing the partial-fraction expansion, we keep the complex conjugate pole pairs together, so that we end up with N_r first-order terms and N_p second-order terms. Thus we can obtain an impulse response of the form

$$h(t) = \sum_{i=1}^{N_r} \beta_i \alpha_i^t u(t) + \sum_{i=1}^{N_p} \zeta_i \gamma_i^t \sin(\theta_i t + \phi_i) u(t). \tag{3.37}$$

$h(t)$ is the sum of N_r first-order terms and N_p second-order terms for a total of $M = N_r + N_p$ terms. We refer to each of these terms as $h_k(t)$ for $k = 1, 2, \ldots, M$. Therefore,

$$h(t) = \sum_{k=1}^{M} h_k(t). \tag{3.38}$$

This decomposition is illustrated in figure 3-6 for a sixth-order IIR filter. By expressing $h(t)$ in this form, we can implement the filter $h(t)$ in figure 3-1 using the parallel structure shown in figure 3-7. By distributing the IS and IG blocks, each branch then has the same overall structure as the original system shown in figure 3-1. Since each of the filters is now either a first- or second-order filter, each branch can be implemented using the methods described in the previous two sections. This results in the final system shown in figure 3-8.

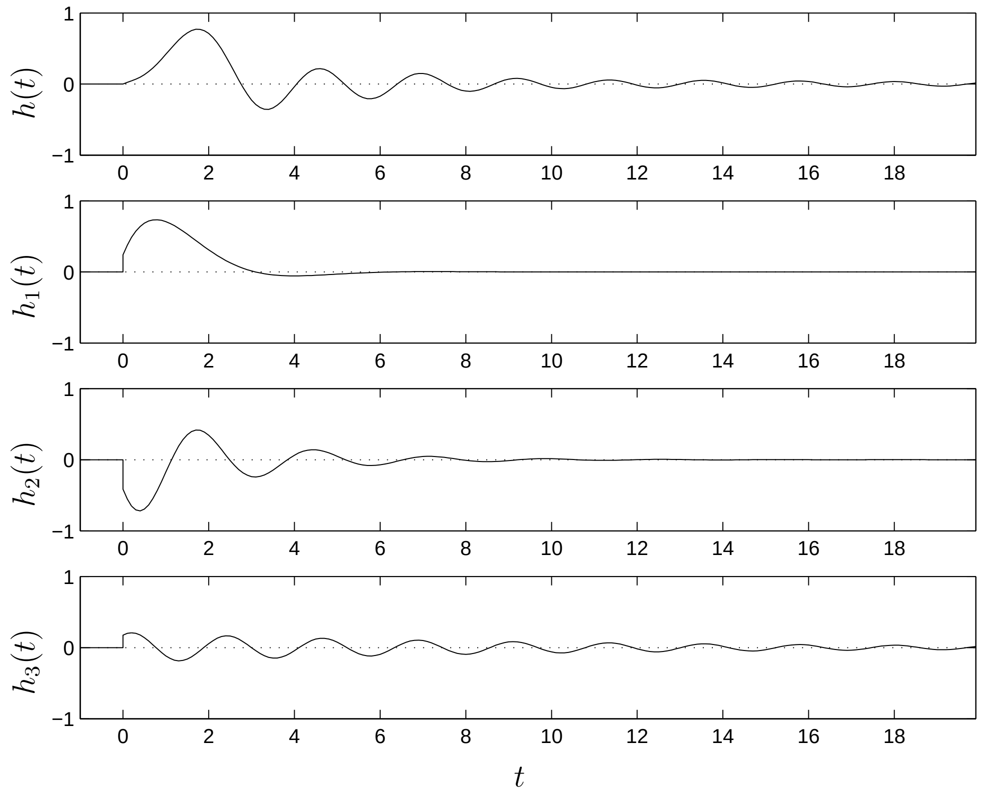

Figure 3-6: Sixth-order IIR filter and its three second-order components.

3.5 Summary of Algorithm

In this section, we give an overview of the algorithm, summarizing the implementation steps. With reference to figure 3-1, we assume that $T' = 1$ and that we are given T. We also assume that a specification on the frequency response of the filter has been determined.

3.5.1 Design and Preparation

Here we describe the setup stage of the algorithm, which must be performed before we start processing the input signal. The following is a list of the setup steps.

1. **Design the filter.** The filter $h(t)$ must be designed so that its frequency response meets the given specification. The system function $H(s)$ should be rational with distinct poles, and must be proper. That is, the order of the numerator must be less than the order of the denominator. Standard elliptic,

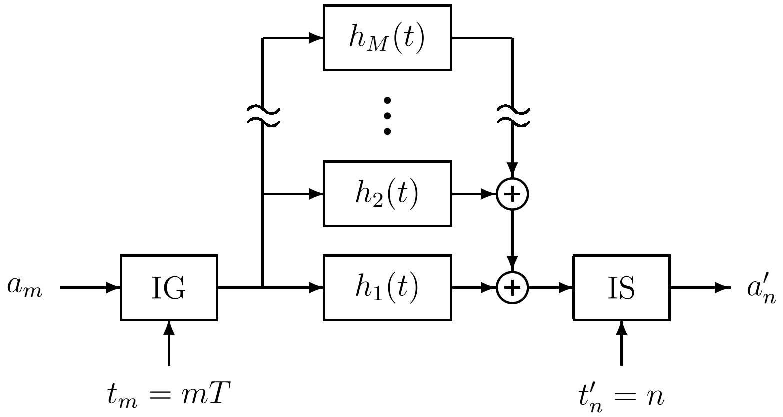

Figure 3-7: $h(t)$ implemented with a parallel structure.

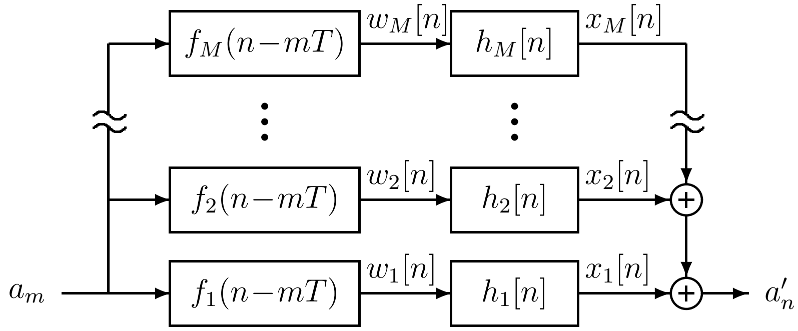

Figure 3-8: Final system for high-quality sampling rate conversion.

Butterworth, or Chebyshev filters will work. If the filter is elliptic or Chebyshev type II, then the order should be chosen to be odd so that there is a zero at infinity, forcing the system function to be proper.

2. **Decompose the filter.** The filter's impulse response $h(t)$ should be decomposed into terms as explained in section 3.4 so that it can be expressed in the form given by (3.37). For each first-order term, the parameters α and β must be found. For each second-order term, the parameters γ, θ, ϕ and ζ must be found.

3. **Determine the constants.** From the filter parameters, the values of the constants can be precomputed and stored. For each first-order term, E^+ and E^- must be determined using (3.20). For each second-order term, A^+, A^-, B^+,

B^-, C^+, C^-, D^+ and D^- must be determined using (3.32), (3.33), (3.34) and (3.35). These constants must be stored and will be used when the algorithm is running. The plus and minus versions are found by using $\Delta = \Delta^+$ and $\Delta = \Delta^-$ respectively, given by (3.16) and (3.17). Finally, the difference equation coefficients should be determined and stored. There is one coefficient, α, for each first-order term as seen in (3.21), and two coefficients, $2\gamma\cos\theta$ and $-\gamma^2$, for each second-order term as seen in (3.36). Only the constants, A^+, A^-, *etc.*., and the difference equation coefficients need to be stored. The other parameters may be discarded.

4. **Initialize the coefficients.** We need to initialize the values of the coefficient $e[n]$ for each first-order term and the two coefficients $c[n]$ and $d[n]$ for each second-order term. If $n = 0$, then $\tau[n] = \tau[0] = 0$. Therefore, for each first-order term, $e[0] = \beta$ and for each second-order term, $c[0] = \zeta \sin\phi$ and $d[0] = \zeta \sin(\phi - \theta)$. We obtain these values by first letting $\tau[0] = 0$ in (3.13), (3.28) and (3.29), and then performing the appropriate scaling.

3.5.2 Runtime Execution

We now describe the runtime stage of algorithm, which is performed while we are processing the input signal. Note that this part of the algorithm can be executed in real time. Since the filter has been decomposed into first- and second-order components we implement our algorithm in a parallel structure as shown in figure 3-8. There are two sets of tasks which need to be accomplished. The first set of tasks occur at the input sampling rate. These are performed whenever the next sample of our input, a_n, becomes available. The second set of tasks occur at the output sampling rate, and are performed whenever the next sample of the output, a'_n, is required. We now describe these tasks in detail.

(i) Input becomes available.

Whenever the next input sample becomes available we perform the following steps for each of the branches. The steps are described for the ith branch and the kth input sample.

1. **Determine the sign of Δ** We must determine whether $\Delta[k] = \Delta^+$ or $\Delta[k] = \Delta^-$ will cause $\tau[k]$ in (3.14) to satisfy $0 \leq \tau[k] < 1$. If $\Delta[k] = \Delta^+$, then we use the plus constants $A = A^+, B = B^+$, *etc.* If $\Delta[k] = \Delta^-$, then we use the minus constants $A = A^-, B = B^-$, *etc.*

2. **Update the coefficients.** If the ith branch is first-order, there is only one coefficient $e[k]$ which is updated using (3.19). If the branch is second-order, then there are two coefficients $c[k]$ and $d[k]$ which are updated using (3.30) and (3.31).

3. **Multiply by the input.** We then multiply the input $x[k]$ by the coefficient(s) and add the result(s) to the correct location(s) in $w_i[n]$. For the first-order case, the result is added to $w_i[\lceil kT \rceil]$, in accordance with (3.11). For the second-order case, there are two results which are added to $w_i[\lceil kT \rceil]$ and $w_i[\lceil kT \rceil + 1]$ respectively, in accordance with (3.27).

(ii) Output is required.

Whenever it is time to output the next sample of a'_n, the following steps are performed. Assume we need the kth output sample.

1. **Apply the difference equation.** We generate the next sample of the output for the ith branch, $x_i[k]$, from the intermediate variable, $w_i[k]$, using the appropriate difference equation. This must be done for all M branches. If the branch is first-order, then the difference equation used is (3.21). If the branch is second-order, then the difference equation used is (3.36).

2. **Sum over the branches.** We then add all of the outputs from each branch to

produce the final output. $a'_k = \sum_{i=1}^{M} x_i[k]$, where M is the number of branches. We then output this final value a'_k.

The two sets of tasks which run at two different rates can be implemented on a digital signal processing (DSP) chip, with each set of tasks programmed as an interrupt service routine (ISR). The first set of tasks can be performed by the input ISR, while the second set can be performed by the output ISR.

3.6 Computational Complexity

In this section, we analyze the computational complexity of our algorithm, measured in MPOS (*i.e.*, the average number of multiplications needed to produce one sample of a'_n).

For simplicity, we assume that $h(t)$ in figure 3-1 is chosen as a Nth-order elliptic, Butterworth, or Chebyshev lowpass filter, where N is odd. This implies that $H(s)$ has one real pole and $N_p = (N-1)/2$ complex conjugate pole pairs. Consequently our implementation requires $N_p + 1$ branches. We also assume that our input consists of Q channels which use the same time axis, and which all need to be converted. For example, a stereo audio signal has two channels, the left channel and the right channel. This allows us to perform the parts of the algorithm which do not need to be repeated for each channel only once.

With reference to section 3.5.2(i), step 1 requires no multiplications. Step 2 requires one multiplication for the first-order branch and four multiplications for each second-order branch. All together we have $2N-1$ multiplications for step 2, independent of the number of channels. Step 3 requires one multiplication for the first-order branch and two multiplications for each second-order branch for each channel. Thus, the total number of MPOS required for this part of the algorithm is $(NQ + 2N - 1)/T$ MPOS.

With reference to section 3.5.2(ii), step 1 requires one multiplication for the first order branch and two multiplications for each of the second-order branches per channel. Since step 2 does not require any multiplications, we require NQ MPOS to

Poles	$-0.10178 \pm 2.82183j$
	$-0.40252 \pm 2.32412j$
	$-0.79570 \pm 0.97295j$
Zeros	$\pm 3.52955j$
	$\pm 4.46260j$

Table 3.1: Pole and zero locations for the sixth-order IIR filter.

implement the steps in section 3.5.2(ii).

Therefore, for an Nth-order filter with one real pole, our algorithm requires $NQ + (NQ + 2N - 1)/T$ MPOS to convert Q channels of data. Here N is taken to be odd. Similarly, we can show that if N were even such that $H(s)$ had no poles on the real axis, the computation required is $NQ + N(Q + 2)/T$ MPOS.

3.7 Design Example

We now look at a specific example in order to compare the performance of our algorithm with conventional methods which were described in chapter 2. We will use the same lowpass filter specification which we used in the design examples from chapter 2, given by table 2.2.

In order to meet the specification, a sixth-order filter is required. This filter was designed using the algorithm developed by Martinez and Parks [19], and Göckler [9]. The impulse response of this filter and the three second-order partial fraction components were shown in figure 3-6, while the frequency response is shown in figure 3-9. Table 3.1 lists the s-plane locations of the poles and zeros of the system function.

By using the complexity formula given in section 3.6, we find that $12{+}24/T$ MPOS or 1.59 million multiplications per second are required. Here we assume that $Q = 2$ and that our sampling rate, $1/T$, is 44.1 kHz, which corresponds to the CD audio standard. We also assume that the input sampling rate is approximately the same as the output sampling rate.

In order to match the same specification using conventional FIR filter techniques, a time varying impulse response of length 15 is required (18 if the filter is linear

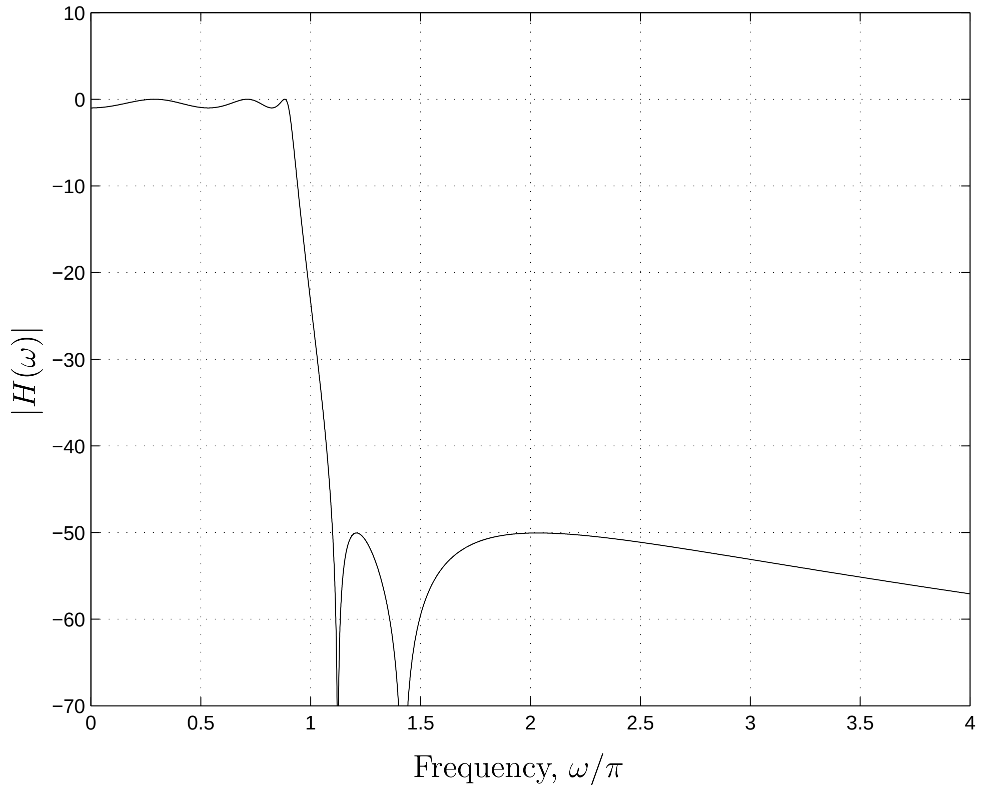

Figure 3-9: The magnitude response of the sixth-order IIR filter.

phase). So 15 multiplications per channel are needed for each output sample. That is 1.32 million multiplications per second—which is actually slightly less than the proposed method. The most significant difference between the two methods is the amount of memory required.

If the taps of the time varying filter are generated using nearest-neighbor interpolation of a stored impulse response, then no computation is needed for generating them. This is why only 1.32 million multiplications per second are required. The disadvantage of using this method of interpolation is that a large table of the CT impulse response must be stored; the table would need to contain about 2,200 points. This number can be even larger if the transition band is very narrow.

More typically, linear interpolation is used to generate the taps of the time varying filter. In that case, one multiplication per tap would also need to be performed in order to do the linear interpolation. This would increase the total amount of computation required to 1.98 million multiplications per second. This drastically reduces the amount of memory needed. Now only about 270 points need to be stored. The memory requirements for our algorithm are very small. Only about 35 numbers need to be stored for the sixth-order IIR filter used in this example.

In order to emphasize the computational efficiency of our algorithm, suppose that five channels need to be converted ($Q = 5$), *eg.*, in a digital surround-sound audio system. In this case, our algorithm actually requires slightly less computation than the nearest-neighbor FIR technique, even though memory requirements are smaller by almost a factor of 70. When linear interpolation is used, our algorithm only requires about 80% of the computation and about 12% of the memory required by the conventional FIR technique.

If we use the method presented by Ramstad [28] for implementing a rational filter, then the savings are not as dramatic. Assume that we wish to implement the same sixth-order IIR filter which we have used in this section. Then a time-varying filter with six taps must be implemented. Let us also assume that the computation allowed for calculating the coefficients is the same as for the proposed method, with two multiplications used for updating each coefficient. This corresponds to a quadratic interpolation of the stored impulse response. Then about 72 points need to be stored which is about twice that required by the proposed method.

We present a single plot in figure 3-10 which we use to compare the performance of the proposed algorithm with that of the algorithms presented in chapter 2. Because our algorithm requires the use of a causal IIR filter which cannot have a linear phase, we make the comparison with the minimum phase versions of the filters from the previous chapter. Each plus sign ($+$) represents our discrete-time implementation of the system shown in figure 2-1 and each times sign ($\times$) represents our implementation of the system of figure 2-13. In each case, we have given several points corresponding to a different values of k. Our algorithm calculates the required filter recursively and does not require a table of impulse response values to be stored. Consequently, there is no inherent trade-off between computation and memory. The performance of our algorithm is therefore represented as a single point, marked with an asterisk ($\ast$). We have superimposed the gray shaded regions from figure 2-20 for ease of comparison.

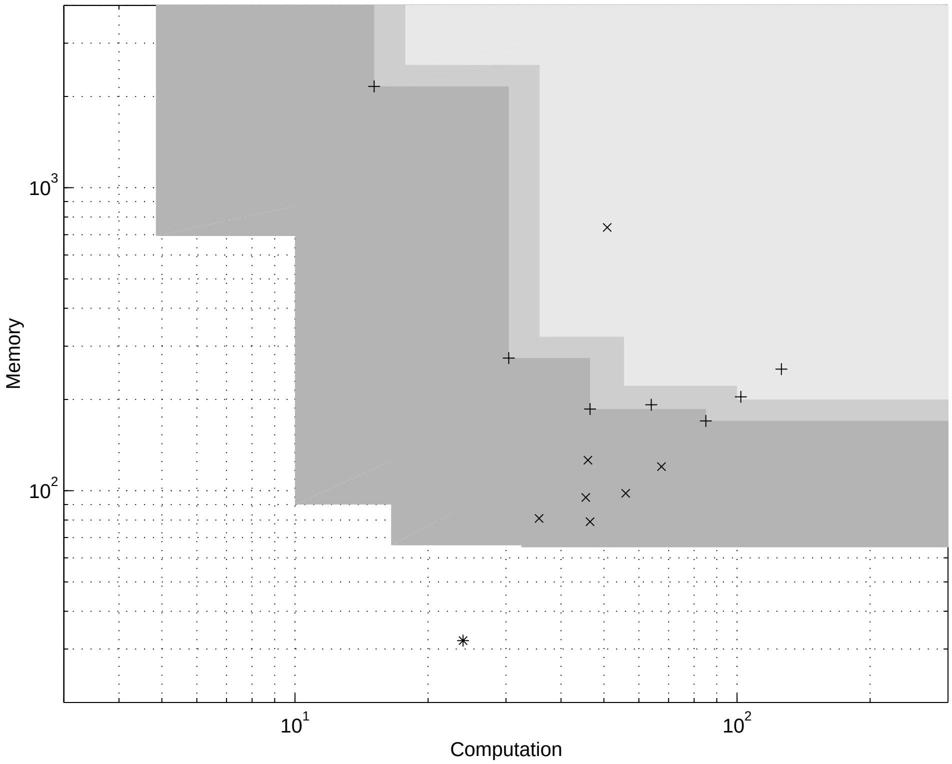

Figure 3-10: Performance plot for the proposed algorithm.

As a final note, we have presented an implementation in which the fixed recursive part of the filter operates at the output sampling rate, while the time-varying FIR part operates at the input sampling rate. A different implementation can be derived in which the fixed recursive part operates at the input sampling rate, while the time-varying FIR part operates at the output sampling rate. In either case, the coefficients of the time-varying filter can be calculated recursively. Depending on whether sampling rate is being increased or decreased, one of these implementations may offer some computational advantage over the other.

Chapter 4

Nonuniform Sampling

4.1 Introduction

In this chapter, we discus a particular resampling algorithm in which we are given a DT signal representing the instantaneous samples of a CT signal on an irregular grid. We then wish to resample this DT signal onto a regular grid, in order to facilitate subsequent processing, or to allow for a more straightforward reconstruction procedure.

This resampling problem has motivation in various engineering applications, including problems relating to communication, medical and underwater imaging and analog to digital (A/D) conversion. References for these and other examples can be found in [1]. In each case, the samples obtained are instantaneous values of some underlying continuous function and the goal is to reconstruct the original function. By resampling onto a regular grid, we allow the reconstruction to be accomplished by lowpass filtering.

The Shannon sampling theorem has been extended by Yen [45], Yao and Thomas [43] and others (see [1] for a comprehensive review) to allow for sampling times which are not uniformly spaced. Several slightly different versions of the nonuniform sampling theorem have arisen. The differences lie in the spaces of functions being considered and the different classes of sampling times which are permitted. The theorem essentially says that a bandlimited signal $x(t)$ is uniquely determined by knowledge of its samples $\{a_n = x(t_n)\}$ as long as the sampling times $\{t_n\}$ occur at a rate which is on average higher than the Nyquist rate. Yao and Thomas [43] showed that even though a signal $x(t)$ is uniquely determined by is samples $\{x(t_n)\}$, the corresponding sampling expansion may not be stable. In order to be stable, Yao and Thomas required that small errors in the sample values correspond to commensurately small errors in the signal. Examples of sets of sampling times which lead to stable and unstable sampling expansions were given.

The reconstruction formula, taken from [45] and [43], is presented in section 4.2 where we discuss some of the associated implementational difficulties. Because of these difficulties, many of the reconstruction algorithms for the nonuniform sampling problem have been based on iterative methods [10, 1, 2, 4, 21, 42] (with the notable exception of [8]). There are some disadvantages to using theses iterative methods. For example, the algorithms are proven to converge when ideal filters are used, but convergence is not guaranteed for approximate filters. Also, these iterative algorithms are not well suited to real-time implementation on a DSP device. These algorithms are usually implemented off-line so that at each step of the iteration, the entire signal may be processed. It may be possible to restructure these iterative algorithms so that each iteration can be implemented as a real-time system, with a number of these systems in cascade. In this case, the overall system can be decomposed into several identical subsystems. We therefore expect that a system which is not constrained in this way, but which uses the same amount of computation, will be able to give a smaller reconstruction error.

The main contribution of this chapter is to derive a possible alternate reconstruction system which is not iterative. This algorithm is based on the reconstruction formulae given by Yen [45] and Yao and Thomas [43], and preliminary results suggest that it may be able to outperform traditional iterative methods. Eldar and Oppenheim [8] have presented a noniterative reconstruction algorithm based on a filterbank implementation, which is valid only for sampling grids which are periodic. Furthermore, the structure of their algorithm is significantly different for grids with different periods. The algorithm presented in this chapter is the same for any periodic grid, and is even valid for certain nonperiodic grids.

In section 4.2 we present an ideal CT system which is equivalent to the reconstruction formula given in [45] and [43]. There are several practical issues which prevent this ideal system from being implemented directly.

4.2 Ideal CT System

Throughout this chapter, we will consider a particular class of sampling grids which have a *uniform density*, as defined by Duffin and Schafer [6]. A grid $\{t_n\}$ has a uniform density of D if

$$\left|t_n - \frac{n}{D}\right| \leq L < \infty, \quad n = 0, \pm 1, \pm 2, \cdots \tag{4.1}$$

and

$$|t_n - t_m| \geq \delta > 0, \quad n \neq m. \tag{4.2}$$

In this chapter, we assume that $D = 1$, without loss of generality. We will also require that for any function $x(t) \in \mathcal{B}_\gamma, \gamma < \pi$, the summation

$$x(t) = \sum_{n=-\infty}^{\infty} x(t_n)\psi_n(t), \quad x(t) \in \mathcal{B}_\gamma \tag{4.3}$$

must converge to $x(t)$ for all t and must be stable. The composing functions $\{\psi_n(t)\}$ depend only on the grid $\{t_n\}$, and are given by

$$\psi_n(t) = \frac{g(t)}{(t - t_n)\dot{g}(t_n)}, \tag{4.4}$$

where

$$g(t) = (t - t_0) \prod_{n=1}^{\infty} \left(1 - \frac{t}{t_n}\right)\left(1 - \frac{t}{t_{-n}}\right), \tag{4.5}$$

and $\dot{g}(t_n)$ is the derivative of $g(t)$ evaluated at $t = t_n$. In order for the expansion (4.3) to be stable, we require that small errors in the sample values $x(t_n)$ must result in commensurately small errors in the reconstructed signal[1]. Yao and Thomas [44] (Theorem 2) imply that the conditions (4.1) and (4.2) guarantee convergence and stability of (4.3) for all $x(t) \in \mathcal{B}_\gamma; \gamma < \pi$. However this is not the case. It is known [6] that when (4.1) and (4.2) are satisfied, there exist functions $\{\psi_n(t)\}$ for which (4.3) will give a convergent, stable reconstruction. However these functions are not given by (4.4). In this thesis we will assume that (4.3) converges and is stable. Here are a few examples of when this occurs:

1. The grid is periodic with some finite period [45] (Theorem III). That is there exists some finite integer M for which $t_m - t_n = MD$, independent of n.

2. The grid is close to uniform [43] (Theorem 1). Specifically, L in (4.1) is less than $D/4$.

3. The grid is obtained by migrating a finite number of points in a uniform grid [45] (Theorem I). Specifically, only a finite number of points in the grid do not satisfy $t_n = nD$.

In each of these cases, the condition that $\gamma < \pi$ may be relaxed to allow $\gamma = \pi$.

By substituting (4.4) into (4.3) we obtain the reconstruction formula

$$x(t) = \sum_{n=-\infty}^{\infty} \frac{x(t_n)g(t)}{(t - t_n)\dot{g}(t_n)}. \tag{4.6}$$

Assuming that this converges at every t, we let

$$r(t) = \sum_{n=-\infty}^{\infty} \frac{x(t_n)}{\dot{g}(t_n)} \frac{1}{(t - t_n)}, \tag{4.7}$$

[1] The l_2-norm is used to measure the error in the sampling values, while the L_2-norm is used to measure the error in the reconstructed signal.

which still converges at every t excluding the points $t \in \{t_n\}$. In this case we have

$$x(t) = g(t)r(t). \tag{4.8}$$

$r(t)$ may also be written as

$$r(t) = q_\delta(t) * h(t), \tag{4.9}$$

where $q_\delta(t)$ is the impulse-train signal given by

$$q_\delta(t) = \sum_{n=-\infty}^{\infty} \frac{x(t_n)}{\dot{g}(t_n)} \delta(t - t_n), \tag{4.10}$$

and $h(t)$ is given by

$$h(t) = \frac{1}{t}. \tag{4.11}$$

We may also write $q_\delta(t)$ as

$$q_\delta(t) = \frac{p_\delta(t)}{\dot{g}(t)}, \tag{4.12}$$

where $p_\delta(t)$ is given by

$$p_\delta(t) = \sum_{n=-\infty}^{\infty} x(t_n)\delta(t - t_n). \tag{4.13}$$

Thus, from (4.8), (4.9) and (4.12), the nonuniform reconstruction formula (4.6) can be represented by the block diagram[2] shown in figure 4-1. To within a scale factor, $h(t)$, given by (4.11), is the impulse response of an ideal Hilbert transformer.

This ideal CT system is difficult to implement for a number of reasons. Specifically, from (4.7), $r(t)$ becomes infinite at every $\{t_n\}$, since $h(t)$ has a singularity at $t = 0$. In section 4.3, we show how this singularity may be removed. The second problem with

[2] We wish to acknowledge the contribution of Yonina Eldar who first pointed out this block diagram representation of the reconstruction formula.

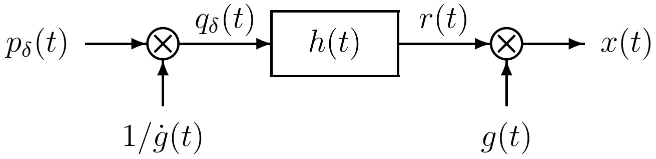

Figure 4-1: Ideal CT reconstruction system.

implementing this system is that $h(t)$ is difficult to approximate since it decays very slowly. This slow decay is related to the fact that the frequency response has a sharp discontinuity. In section 4.4, we show that a transition band may be introduced in the frequency response, thus allowing $h(t)$ to have a faster rate of decay. Since $h(t)$ is required to have linear phase, it would be difficult to implement this system using analog filters. Thus in section 4.5 we present a DT system which implements the system, and produces uniform samples of $x(t)$. The final difficulty with implementing this system is that it requires knowledge of the function $g(t)$. In section 4.6 we present an ideal system for calculating $g(t)$, based on the work of Logan [15, 14].

4.3 System With Nonsingular Filters

A filter with impulse response given by $1/t$ is referred to as a Hilbert transformer[3] [37, 11]. The frequency response this Hilbert transformer is known [37], and is given by

$$H(\omega) = -j\pi \operatorname{sgn} \omega, \tag{4.14}$$

where

$$\operatorname{sgn} \omega = \begin{cases} -1, & \omega < 0, \\ 0, & \omega = 0, \\ 1, & \omega > 0. \end{cases} \tag{4.15}$$

[3] This is true to within a scale factor since the ideal Hilbert transformer has an impulse response $1/\pi t$.

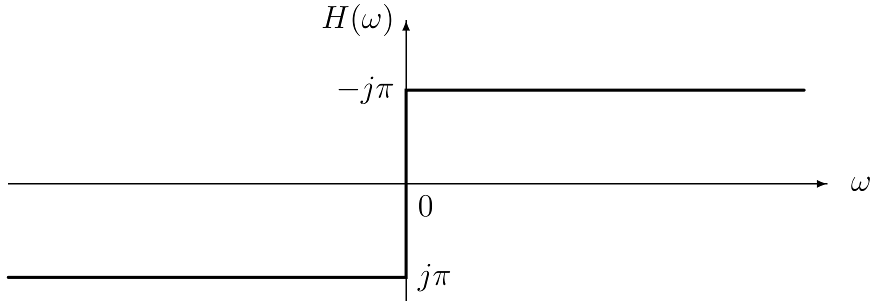

Figure 4-2: Frequency response $H(\omega)$.

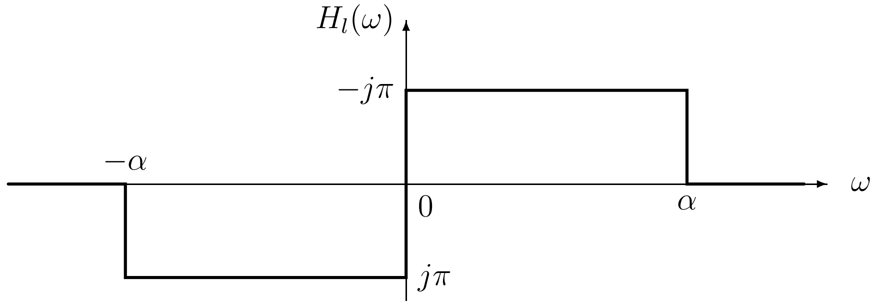

Figure 4-3: Frequency response $H_l(\omega)$.

$H(\omega)$ is shown in figure 4-2.

In order to remove the singularity, we construct a new filter $h_l(t)$ whose frequency response is given by

$$H_l(\omega) = H(\omega) - \frac{1}{2}H(\omega - \alpha) - \frac{1}{2}H(\omega + \alpha), \tag{4.16}$$

and is shown in figure 4-3. $H_l(\omega)$ is zero outside $[-\alpha, \alpha]$, and thus the impulse response $h_l(t)$ is a lowpass function, which is therefore free of singularities. By the modulation property of the Fourier transform, $h_l(t)$ is given by

$$\begin{aligned} h_l(t) &= h(t)\left(1 - \frac{1}{2}e^{-j\alpha t} - \frac{1}{2}e^{j\alpha t}\right) \\ &= \frac{1 - \frac{1}{2}e^{-j\alpha t} - \frac{1}{2}e^{j\alpha t}}{t}. \end{aligned} \tag{4.17}$$

Now, if we let

$$r_l(t) = q_\delta(t) * h_l(t), \tag{4.18}$$

then we have

$$r_l(t) = \sum_{n=-\infty}^{\infty} \frac{x(t_n)\left(1 - \frac{1}{2}e^{-j\alpha(t-t_n)} - \frac{1}{2}e^{j\alpha(t-t_n)}\right)}{\dot{g}(t_n)(t-t_n)}. \tag{4.19}$$

Letting $x_l(t) = g(t)r_l(t)$ and substituting (4.4) we get

$$x_l(t) = \sum_{n=-\infty}^{\infty} x(t_n)\psi_n(t)\left(1 - \frac{1}{2}e^{-j\alpha(t-t_n)} - \frac{1}{2}e^{j\alpha(t-t_n)}\right). \tag{4.20}$$

We can rearrange this sum to give

$$\begin{aligned} x_l(t) \;\; = & \sum_{n=-\infty}^{\infty} x(t_n)\psi_n(t) \\ & -\frac{1}{2}e^{-j\alpha t} \sum_{n=-\infty}^{\infty} x(t_n)e^{j\alpha t_n}\psi_n(t) \\ & -\frac{1}{2}e^{j\alpha t} \sum_{n=-\infty}^{\infty} x(t_n)e^{-j\alpha t_n}\psi_n(t). \end{aligned} \tag{4.21}$$

The first sum reduces to $x(t)$ and the second and third sums give functions in $\mathcal{B}_\pi$, which are then modulated by $e^{j\alpha t}$ and $e^{-j\alpha t}$ respectively[4]. Thus these last two terms provide no contribution to the spectrum of $x_l(t)$ between $-\zeta$ and ζ, where

$$\zeta = \alpha - \pi. \tag{4.22}$$

Therefore $x(t)$ can be recovered by lowpass filtering $x_l(t)$, provided that $\zeta > \gamma$. That

[4] The fact that the second and third sums give functions in $\mathcal{B}_\pi$ results from the assumed stability of (4.3), and the fact that the sequence $\{x(t_n)\}$ is in l_2 [6].

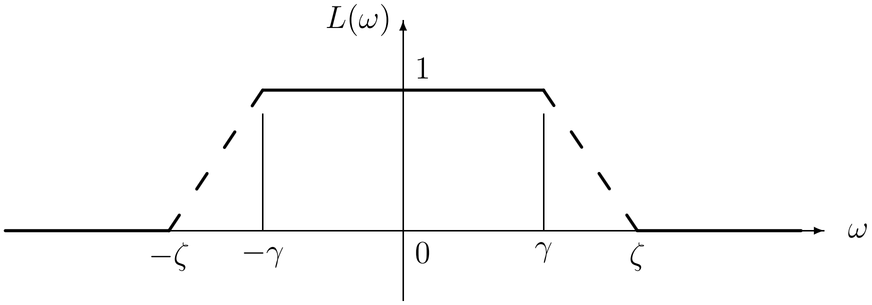

Figure 4-4: Frequency response $L(\omega)$.

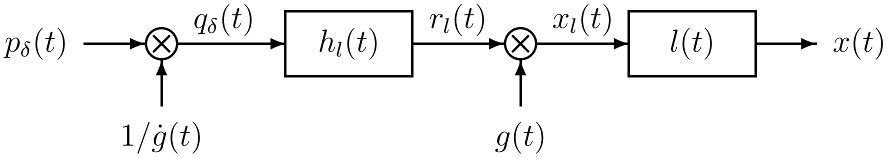

Figure 4-5: Reconstruction system with nonsingular filters.

is

$$x(t) = x_l * l(t), \tag{4.23}$$

for any filter $l(t)$ whose frequency response $L(\omega)$ satisfies

$$L(\omega) = \begin{cases} 1, & |\omega| \leq \gamma, \\ 0, & |\omega| \geq \zeta. \end{cases} \tag{4.24}$$

$L(\omega)$ is drawn in figure 4-4, where the dashed lines represent the transition bands, *i.e.*, regions in which the response is unconstrained.

The modified system for recovering $x(t)$ is shown in figure 4-5. If the filters $l(t)$ and $h_l(t)$ satisfy (4.24) and (4.16) respectively, then the systems in figures 4-5 and 4-1 will give identical outputs.

4.4 System With Transition Bands Allowed

Kaiser [16] showed that a finite support lowpass filter could be designed by a windowing method which allows the frequency-domain approximation error to fall exponentially in the length of the filter. The error goes as $e^{-\Delta\Theta/4}$, where Δ is the width of the transition band and Θ is the length of the filter. Logan [15] considered FIR Hilbert transformers and achieved the same exponential convergence in the filter length, and transition band width. Logan's result was given in terms of the time-domain approximation error.

The filter $l(t)$, whose frequency response is given in figure 4-4, allows transition bands of width $\Delta_l = \zeta - \gamma$. We may therefore replace $l(t)$ with an FIR approximation of length Θ_l and achieve an approximation error which decays exponentially in the quantity $\Delta_l\Theta_l$. Since the length of the FIR filter is directly proportional to the amount of computation required to implement it, this result is encouraging because it suggests that we will be able to implement $l(t)$ efficiently.

In this section we will show that the filter $h_l(t)$ can also be allowed to have transition bands and we can then achieve the same exponential convergence in the length of the approximation filter.

In order to facilitate the discussion, we will make use of terminology introduced by Logan [13]. In particular, Logan suggested that it is not necessary to define the Fourier transform of a function $f(t)$ in order to give precise meaning to the statement that "the Fourier transform of $f(t)$ vanishes on a certain set E." In particular, we say that a bounded function $f(t)$ has a Fourier transform which vanishes on E, if

$$\int_{-\infty}^{\infty} f(t)\eta(-t)\,dt = 0 \tag{4.25}$$

for all $\eta(t)$ in L_1 whose Fourier transform is zero outside of E. This definition is consistent in that if $f(t)$ does have a Fourier transform, then that Fourier transform will be zero for all frequencies in E. Logan also said that two functions, $f_1(t)$ and $f_2(t)$ could be said to "have Fourier transforms which agree on E" if their difference $f_1(t) - f_2(t)$ has a Fourier transform which vanishes on E. The mathematical details of this definition are not essential in order to follow the discussion, we only need to recognize that our intuition regarding finite energy signals can be applied to infinite energy signals as long as they are bounded.

By Theorem 1 in [13] and the theory of entire functions [29], any function $f(t)$ with only real zeros which occur with a uniform density of ξ will have a Fourier transform which vanishes outside $[-\pi\xi, \pi\xi]$. Thus the function $g(t)$, whose zeros are the points in the grid $\{t_n\}$, will have a Fourier transform which vanishes outside $[-\pi, \pi]$. It can be shown that this fact implies that two functions $x(t)$ and $x'(t)$, where

$$x(t) = y(t)g(t) \tag{4.26}$$

and

$$x'(t) = y'(t)g(t), \tag{4.27}$$

will have Fourier transforms which agree over $[-\zeta, \zeta]$ provided that $y(t)$ and $y'(t)$ have Fourier transforms which agree over $[-\alpha, \alpha]$. ζ is given by (4.22).

The statement that transition bands may be introduced for the Hilbert transformer relies on the fact that the impulse-train signal $q_\delta(t)$ in figure 4-5 is highpass, as we will demonstrate in section 4.4.1. Specifically, $q_\delta(t)$ has a Fourier transform which vanishes over $(-\lambda, \lambda)$, where

$$\lambda = \pi - \gamma. \tag{4.28}$$

If $q_\delta(t)$ has no spectrum over $(-\lambda, \lambda)$, the frequency response of $h_l(t)$ is irrelevant in that region, in which case, we can introduce a transition band at $\omega = 0$ of width 2λ. We replace $h_l(t)$ with a filter $h_t(t)$ whose frequency response $H_t(\omega)$ satisfies

$$H_t(\omega) = \begin{cases} j\pi, & \lambda \leq \omega \leq \alpha \\ -j\pi, & -\lambda \geq \omega \geq -\alpha \\ 0, & |\omega| \geq \beta. \end{cases} \tag{4.29}$$

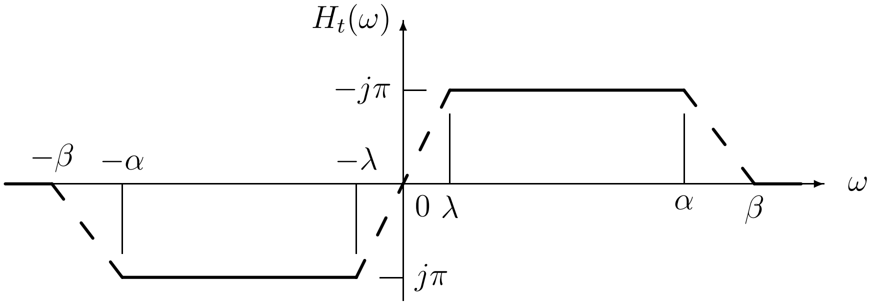

Figure 4-6: Frequency response $H_t(\omega)$.

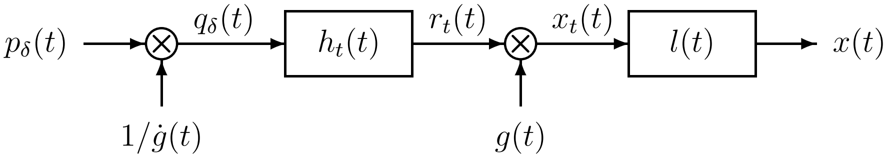

Figure 4-7: Reconstruction system which allows transition bands.

The frequency response is unspecified over the intervals $(-\beta, -\alpha)$, $(-\lambda, \lambda)$ and (α, β), and is depicted in figure 4-6. We now have the system shown in figure 4-7. $r_t(t)$ and $r_l(t)$ will have Fourier transforms which agree over $[-\alpha, \alpha]$ because the Fourier transforms of $h_t(t)$ and $h_l(t)$ agree over that interval except in $(-\lambda, \lambda)$, where the Fourier transforms of $r_l(t)$ and $r_t(t)$ both vanish.

Since the Fourier transforms of $r_l(t)$ and $r_t(t)$ agree over $[-\alpha, \alpha]$, Fourier transforms of $x_t(t)$ and $x_l(t)$ will agree over $[-\zeta, \zeta]$; ζ given by (4.22). Now, since the frequency response of $l(t)$ satisfies (4.24), the output of the system in figure 4-7 will be identical to the output of the system in figure 4-5.

For convenience, we can let $\beta = \alpha + 2\lambda$ so that all three transition bands of $h_t(t)$ are of the same width, $\Delta_h = 2\lambda$. We can then approximate $h_t(t)$ with an FIR filter of length Θ_h and achieve an approximation error which decays exponentially in the quantity $\Delta_h \Theta_h$.

4.4.1 Proof that $q_\delta(t)$ is Highpass

Assume that each point in the grid, $\{t_n\}$, is an integer multiple of some small time increment ϵ, *i.e.*,

$$t_n = k_n \epsilon, \tag{4.30}$$

where each k_n is an integer[5]. In this case, we have that

$$g(t) \overset{\propto}{=} \prod_{n=-\infty}^{\infty} \left(1 - \frac{t}{t_n}\right), \tag{4.31}$$

which converges on the condition that the terms of the product are ordered by increasing $|t_n|$ (see [15]). The symbol $\overset{\propto}{=}$ is used to indicate equality to within a multiplicative constant factor.

The set of times $\{t_n\}$ form a subset of the regular grid $\{n\epsilon\}$. We construct a function $\sigma(t)$ which has a zero at every point in this regular grid, *i.e.*,

$$\sigma(t) = \sin(\pi t/\epsilon). \tag{4.32}$$

The sine function can be expressed as the infinite product

$$\sigma(t) \overset{\propto}{=} t \prod_{\substack{n=-\infty \\ n\neq 0}}^{\infty} \left(1 - \frac{t}{n\epsilon}\right), \tag{4.33}$$

where the terms are ordered increasing $|n|$. The set of zeros of $\sigma(t)$ has a uniform density of $1/\epsilon$, while the set of zeros of $g(t)$ has a uniform density of 1. Now consider constructing a new grid $\{\tau_n\}$ consisting of all the zeros of $\sigma(t)$ which are not also zeros of $g(t)$. This new grid $\{\tau_n\}$ will have a uniform density of $1/\epsilon - 1$. The set $\{n\epsilon\}$ is the union of the disjoint sets $\{t_n\}$ and $\{\tau_n\}$. We define the function $\rho(t)$ to be the

[5] We also assume that zero is not a member of the grid, $\{t_n\}$. If zero is a point in the grid $\{t_n\}$, then we may shift the time origin so that this is not the case.

product

$$\rho(t) \overset{\propto}{=} t \prod_{\substack{n=-\infty \\ n\neq 0}}^{\infty} \left(1 - \frac{t}{\tau_n}\right), \tag{4.34}$$

where we let $\tau_0 = 0$ and the terms in the product are ordered by increasing $|\tau_n|$. It follows directly from the definitions (4.31), (4.33) and (4.34) that

$$\sigma(t) \overset{\propto}{=} g(t)\rho(t). \tag{4.35}$$

Since $\rho(t)$ becomes zero at every point in $\{\tau_n\}$, it follows that

$$\rho(\tau_n) = 0; \quad n = 0, \pm 1, \pm 2, \ldots \tag{4.36}$$

$\rho(t)$ is also related to the derivative of $g(t)$ by the following equation;

$$\rho(t_n) \overset{\propto}{=} \frac{\cos(\pi t_n/\epsilon)}{\dot{g}(t_n)}, \tag{4.37}$$

where the relationship only holds on the grid $\{t_n\}$. To show this, we differentiate (4.35) to give

$$\dot{\sigma}(t) \overset{\propto}{=} \dot{g}(t)\rho(t) + g(t)\dot{\rho}(t). \tag{4.38}$$

The second term on the right hand side goes to zero when evaluated at any t_n since $g(t)$ has a zero there. By rearranging (4.38), substituting (4.32) and letting $t = t_n$, we get (4.37). An additional important property of the function $\rho(t)$ is that it is lowpass, with its Fourier transform vanishing outside $[-\mu, \mu]$, where

$$\mu = \pi\left(\frac{1}{\epsilon} - 1\right). \tag{4.39}$$

To see this, note that the set of zeros of $\rho(t)$ has a uniform density of $1/\epsilon - 1$.

Let the impulse-train function, $z_\delta(t)$, be defined by

$$z_\delta(t) = \rho(t) \sum_{n=-\infty}^{\infty} \frac{1}{\cos \pi n} \delta(t - n\epsilon). \tag{4.40}$$

The summation in (4.40) has impulses at the points in $\{t_n\}$ and at the points in $\{\tau_n\}$, *i.e.*, at all integer multiples of ϵ. The impulses that occur at the points in $\{\tau_n\}$ are "zeroed out" when we multiply by $\rho(t)$ because of (4.36). Thus $z_\delta(t)$ will have impulses only at the points in the grid $\{t_n\}$. By substituting for $\rho(t_n)$ given by (4.37) we have that

$$z_\delta(t) = \sum_{n=-\infty}^{\infty} \frac{\delta(t - t_n)}{\dot{g}(t_n)}. \tag{4.41}$$

The impulse-train function

$$\sum_{n=-\infty}^{\infty} \frac{1}{\cos \pi n} \delta(t - n\epsilon) = \sum_{n=-\infty}^{\infty} (-1)^n \delta(t - n\epsilon) \tag{4.42}$$

from (4.40) is highpass, with its Fourier transform vanishing over $(-\pi/\epsilon, \pi/\epsilon)$. Since $\rho(t)$ has a Fourier transform that vanishes outside $[-\mu, \mu]$, with μ given by (4.39), by a result given earlier, $z_\delta(t)$ will be highpass with its Fourier transform vanishing over $(-\pi, \pi)$.

Recall that we have restricted the grid $\{t_n\}$ so that each point is an integer multiple of ϵ, but the result which we have derived does not depend on ϵ. In other words, for any value of ϵ, $z_\delta(t)$ will have a Fourier transform which vanishes over $(-\pi, \pi)$. We therefore conjecture that this restriction may be relaxed, since any grid satisfying (4.2) can be approximated arbitrarily closely by letting ϵ go to zero. Furthermore, uncertainty in the sampling times may be translated into uncertainty in the sample amplitudes by bounding the derivative of $x(t)$, and small amplitude errors will lead to small errors in the reconstructed signal since the reconstruction process is stable. No formal proof for the more general case will be given in the thesis, and so we will assume that the grid $\{t_n\}$ satisfies (4.30).

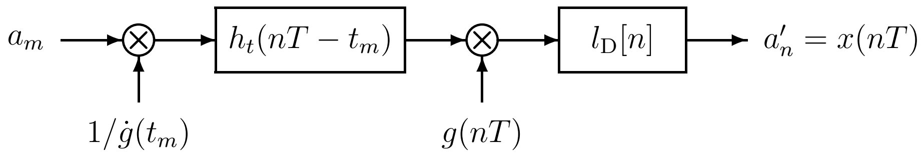

Figure 4-8: Ideal DT resampling system.

With reference to figure 4-7, the signal $q_\delta(t)$ given by (4.10), can also be expressed as

$$q_\delta(t) = x(t) z_\delta(t). \tag{4.43}$$

Now, since $z_\delta(t)$ is highpass, with a Fourier transform which vanishes over $(-\pi, \pi)$, and $x(t)$ is lowpass with a Fourier transform which vanishes outside of $[-\gamma, \gamma]; \gamma < \pi$, we can conclude that $q_\delta(t)$ is highpass, with a Fourier transform which vanishes over $(-\lambda, \lambda)$, where λ is given by (4.28). This proves the result.

4.5 DT System

The ideal CT system given in figure 4-7 can be used to derive the ideal DT system shown in figure 4-8, with T chosen so that

$$T \leq \frac{2\pi}{\alpha + \beta}, \tag{4.44}$$

and the filters $h_t(t)$ and $l_{\mathrm{D}}[n]$ satisfying the conditions shown in figures 4-6 and 4-9. In this case, the samples $x(nT)$ at the output of the system are exact. We may then consider replacing the filters with approximate FIR filters where the lengths of the filters may be chosen so that any desired degree of accuracy for the output samples may be achieved. Since the output grid is regular, the elementary resampler can be replaced with two filters $f(t)$ and $h_{\mathrm{D}}[n]$ as described in section 2.5, where $f(t)$ is allowed to be very short. In this case, $h_{\mathrm{D}}[n]$ must not only equalize for the passband of $f(t)$, but must also incorporate the Hilbert transforming property of the filter $h(t)$.

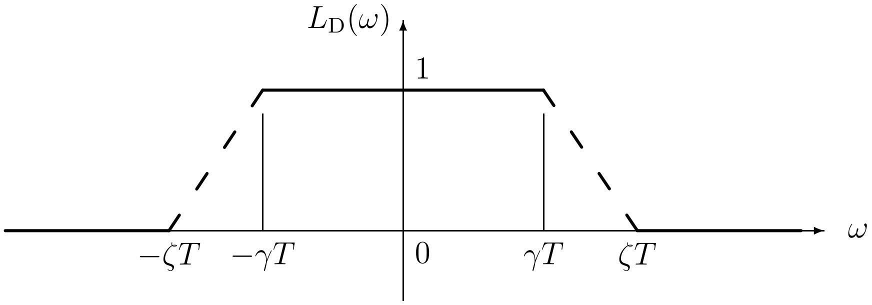

Figure 4-9: Frequency response $L_D(\omega)$.

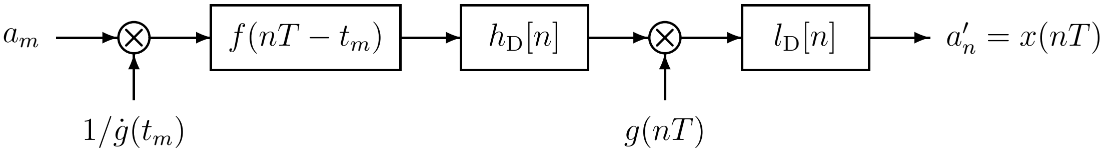

Figure 4-10: DT system which allows for a short time-varying filter.

This results in the final system shown in figure 4-10.

4.6 Ideal System for Calculating $g(t)$

So far we have assumed that the function $g(t)$ is known since it depends only on the grid $\{t_n\}$. However, in real-time applications, it may be necessary to calculate the required samples of $g(t)$ on the fly. This can be quite difficult since $g(t)$ is given as an infinite product (4.5). Logan [15] has investigated the problem of localizing the infinite product, and presented a system in [14] which may lead to a real-time algorithm for evaluating $g(t)$. Figure 4-11 shows a slightly modified version of the ideal system presented by Logan. It remains to be shown that this ideal system can be implemented in discrete time. In figure 4-11, $\Re\{\cdot\}$ takes the real part of its input,

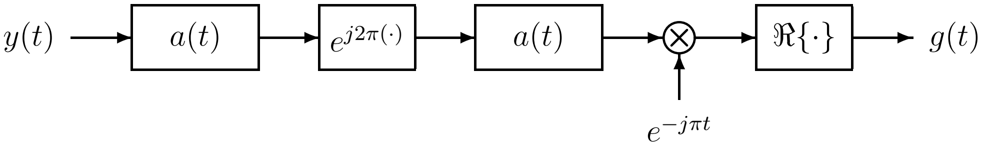

Figure 4-11: Ideal CT system for calculating $g(t)$.

and the frequency response of filter $a(t)$ is

$$A(\omega) = \begin{cases} 1, & 0 \leq \omega \leq \pi, \\ 0, & \text{otherwise.} \end{cases} \tag{4.45}$$

The function $y(t)$ depends only on the grid, and is given by

$$y(t) = t + \sum_{n=-\infty}^{\infty} u(-t_n) - u(t - t_n). \tag{4.46}$$

$y(t)$ has a uniform slope of 1, but has discontinuities of magnitude -1 at each point in the grid $\{t_n\}$.

Chapter 5

Nonuniform Reconstruction

5.1 Introduction

In chapter 4 we saw that applications of the nonuniform sampling problem tend to arise in data acquisition systems or A/D converters. This is because the sampling process may be corrupted by errors or constrained by the physical or geometrical limitations of these systems so that the resulting samples are irregularly spaced. In this chapter, we consider the *nonuniform lowpass reconstruction problem*, which has applications that tend to arise in the context of signal generation or D/A conversion. The nonuniformity is introduced either by a defect in the D/A converter or may be part of the design.

Consider a form of D/A conversion which can be modeled as follows. First, an impulse-train signal is generated with impulses whose locations are fixed, but whose amplitudes are adjustable. This impulse train is then processed with a lowpass filter to recover the CT signal. The impulses are usually uniformly spaced, but in some cases may not be. Although the mathematical tools exist, this problem has received surprisingly little attention in the engineering literature.

A similar problem referred to as click modulation was solved by Logan [14] in which the amplitudes of the impulses are fixed, and the times are allowed to vary.

In section 5.2, we introduce an ideal solution to the nonuniform lowpass reconstruction problem, and we present an ideal CT system corresponding to this solution.

We briefly discuss an equivalent DT system in section 5.3. We then turn our attention to a special case in which the irregular grid my be obtained by deletion of a single point from an otherwise regular grid. In section 5.4 we present a solution which is based on the theoretical results derived earlier. Then in section 5.4.1 we generalize this solution, and finally in section 5.4.2 we present an optimal FIR approximation.

5.2 Ideal CT System

In this section, we derive a solution to the nonuniform lowpass reconstruction problem by using the nonuniform sampling reconstruction formula presented in chapter 4.

Let $x(t)$ be a bandlimited signal in $\mathcal{B}_\gamma$, where $\gamma < \pi$, and let $\{t'_n\}$ be an irregular grid with a uniform density of one. We wish to find amplitudes $\{a'_n\}$ so that

$$x(t) = \sum_{n=-\infty}^{\infty} a'_n \phi(t - t'_n), \tag{5.1}$$

where $\phi(t)$ is given as

$$\phi(t) = \frac{\sin \gamma t}{\pi t}. \tag{5.2}$$

Equivalently, we wish to find a impulse-train signal $p_\delta(t)$ given by

$$p_\delta(t) = \sum_{n=-\infty}^{\infty} a'_n \delta(t - t'_n), \tag{5.3}$$

so that $x(t)$ may be recovered by lowpass filtering $p_\delta(t)$. That is, $x(t)$ is related to $p_\delta(t)$ by

$$x(t) = p_\delta(t) * \phi(t). \tag{5.4}$$

Figure 5-1 depicts this lowpass reconstruction relationship, where the block labeled IG is the impulse generator defined by (5.3). The frequency response of the filter $\phi(t)$

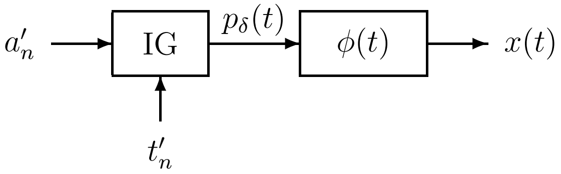

Figure 5-1: Lowpass reconstruction system.

is given by

$$\Phi(\omega) = \begin{cases} 1, & |\omega| \leq \gamma, \\ 0, & \text{otherwise.} \end{cases} \tag{5.5}$$

In order to derive the solution, we will impose some of the same restrictions on the grid which we imposed in chapter 4. Specifically, we require that the grid $\{t'_n\}$ have the following property: for any function $f(t) \in \mathcal{B}_\gamma$, there must exist a stable expansion of the form

$$f(t) = \sum_{n=-\infty}^{\infty} f(t'_n)\psi_n(t), \tag{5.6}$$

where

$$\psi_n(t) = \frac{g(t)}{(t - t'_n)\dot{g}(t'_n)}, \tag{5.7}$$

and

$$g(t) = (t - t'_0) \prod_{n=1}^{\infty} \left(1 - \frac{t}{t'_n}\right) \left(1 - \frac{t}{t'_{-n}}\right), \tag{5.8}$$

and $\dot{g}(t'_n)$ is the derivative of $g(t)$ evaluated at $t = t'_n$. A few examples of grids satisfying this condition were given in section 4.2.

Since, $x(t)$ is in $\mathcal{B}_\gamma$, we know that

$$\begin{aligned} x(t) &= x(t) * \phi(t) \\ &= \int_{-\infty}^{\infty} x(\tau)\phi(t - \tau)\, d\tau, \end{aligned}$$

assuming $\phi(t)$ is even symmetric. Since $\phi(t-\tau)$, treated as a function of τ with t fixed, is in $\mathcal{B}_\gamma$, we can replace $\phi(t-\tau)$ with an expansion of the form (5.6). This gives

$$\begin{aligned} x(t) &= \int_{-\infty}^{\infty} x(\tau)\left[\sum_{n=-\infty}^{\infty}\phi(t-t'_n)\psi_n(\tau)\right]\,d\tau \\ &= \int_{-\infty}^{\infty}\sum_{n=-\infty}^{\infty} x(\tau)\psi_n(\tau)\phi(t-t'_n)\,d\tau \\ &= \sum_{n=-\infty}^{\infty}\left[\int_{-\infty}^{\infty} x(\tau)\psi_n(\tau)\,d\tau\right]\phi(t-t'_n). \end{aligned} \tag{5.9}$$

Swapping the order of the summation and the integral is justified by Fubini's theorem since $x(t) \in \mathcal{B}_\gamma$, and is therefore finite. By comparing (5.9) and (5.1) we see that the coefficients $\{a'_n\}$ are given by

$$a'_n = \int_{-\infty}^{\infty} x(t)\psi_n(t)\,dt, \tag{5.10}$$

where the functions $\{\psi_n(t)\}$ are defined by (5.7). Note that this solution is not necessarily unique. In other words, for a particular value of γ there may be a class of sequences $\{a'_n\}$ which satisfy (5.1). However the solution given by (5.10) is valid for every $\gamma < \pi$.

By substituting (5.7) into (5.10) we get

$$\begin{aligned} a_n &= \int_{-\infty}^{\infty} x(t)\frac{g(t)}{(t-t'_n)\dot{g}(t'_n)}\,dt, \\ &= \frac{-1}{\dot{g}(t'_n)}\int_{-\infty}^{\infty} x(t)g(t)\frac{1}{(t'_n-t)}\,dt. \end{aligned} \tag{5.11}$$

By letting

$$v(t) = x(t)g(t), \tag{5.12}$$

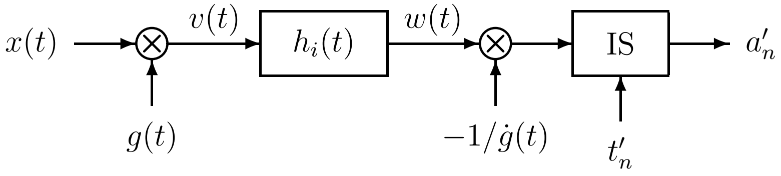

Figure 5-2: Ideal CT system.

and

$$h_i(t) = \frac{1}{t}, \tag{5.13}$$

(5.11) becomes

$$a_n = \frac{-1}{\dot{g}(t'_n)} \int_{-\infty}^{\infty} v(t) h_i(t'_n - t)\ dt, \tag{5.14}$$

or

$$a'_n = \frac{-1}{\dot{g}(t'_n)} w(t'_n), \tag{5.15}$$

where

$$w(t) = v(t) * h_i(t). \tag{5.16}$$

By combining (5.15), (5.16) and (5.12), we get the block diagram shown in figure 5-2, where the block labeled IS is the instantaneous sampler defined in chapter 1.

We may express $p_\delta(t)$, given by (5.3) as

$$\begin{aligned} p_\delta(t) &= \sum_{n=-\infty}^{\infty} \frac{-1}{\dot{g}(t'_n)} w(t'_n) \delta(t - t'_n) \\ &= -w(t) z_\delta(t), \end{aligned} \tag{5.17}$$

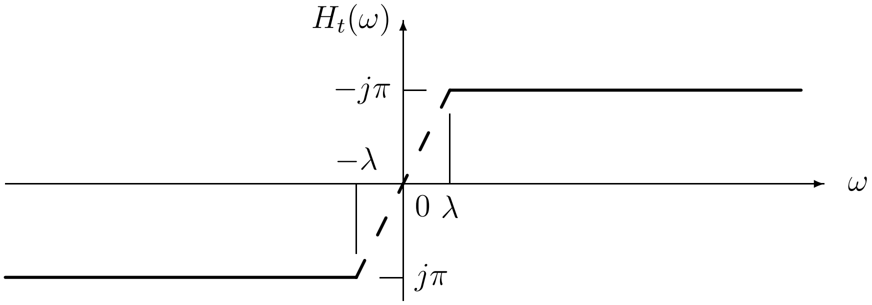

Figure 5-3: Frequency response $H_t(\omega)$.

where $z_\delta(t)$ is given by

$$z_\delta(t) = \sum_{n=-\infty}^{\infty} \frac{1}{\dot{g}(t'_n)} \delta(t - t'_n). \tag{5.18}$$

From figure 5-1 we see that $x(t)$ is recovered from $p_\delta(t)$ by filtering with $\phi(t)$, which we assume to be ideal, with frequency response given by (5.5). Thus, frequency components of $p_\delta(t)$ outside the range $[-\gamma, \gamma]$ are irrelevant. In other words, any sequence $\{a'_n\}$ for which $p_\delta(t)$ satisfies (5.4), is an equally valid solution. We will use this fact to show that the Hilbert transformer is allowed a transition band over $(-\lambda, \lambda)$, where $\lambda = \pi - \gamma$. Consider what happens if we replace the ideal Hilbert transformer $h_i(t)$ in figure 5-2 with $h_t(t)$, which is allowed a transition band as shown in figure 5-3. The dashed line in the figure indicates the region for which the frequency response is unspecified, *i.e.*, the transition band. If we let

$$w_t(t) = v(t) * h_t(t), \tag{5.19}$$

then $w_t(t)$ and $w(t)$ will have Fourier transforms which differ only over $(-\lambda, \lambda)$. We now utilize the result derived in section 4.4 that $z_\delta(t)$ is highpass, having a Fourier transform which vanishes over $(-\pi, \pi)$. If we let

$$p_{t\delta}(t) = -w_t(t) z_\delta(t), \tag{5.20}$$

then $p_{t\delta}(t)$ and $p_\delta(t)$ will have Fourier transforms which differ only outside $[-\gamma, \gamma]$.

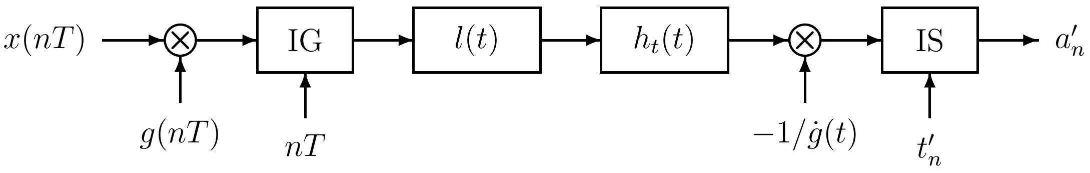

Figure 5-4: System with discrete input and output.

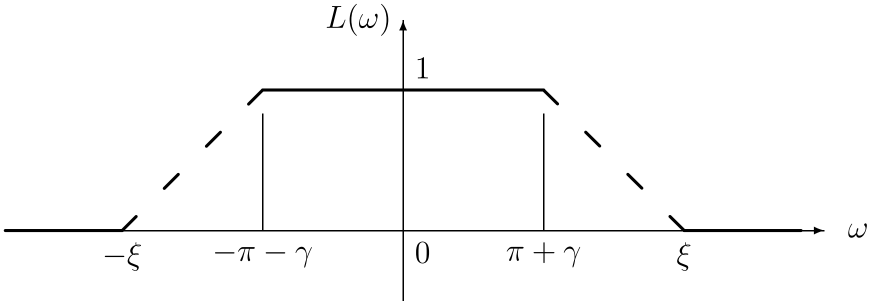

Figure 5-5: Frequency response $L(\omega)$.

By allowing a transition band for the Hilbert transformer, we may use an FIR approximation, which gives an approximate output which approaches the ideal output exponentially in the length of the filter (see [15]).

5.3 DT System

$x(t)$ and $g(t)$ are both lowpass and therefore $v(t)$ will also be lowpass. In fact, it can be shown that $v(t)$ has a Fourier transform which vanishes outside $[-\pi-\gamma, \pi+\gamma]$. Thus we may represent $v(t)$ by its uniform samples $v(nT)$, for a sufficiently small T. This results in the system in figure 5-4, where $l(t)$ is the lowpass reconstruction filter whose frequency response is shown in figure 5-5. ξ must be chosen so

$$\xi \leq \frac{2\pi}{T} - \pi - \gamma. \tag{5.21}$$

If we choose $T = 1/2$, then we may allow the transition bands to be of width 2λ. We then combine the lowpass filter, $l(t)$, and the Hilbert transformer, $h_t(t)$, into a single filter $h(t)$ whose frequency response is shown in figure 5-6. Figure 5-7 shows the

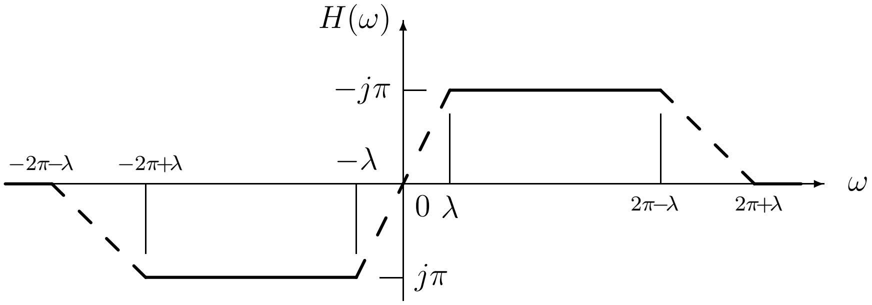

Figure 5-6: Frequency response $H(\omega)$.

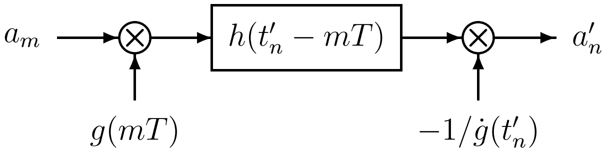

Figure 5-7: Ideal DT resampling system.

final version of the ideal DT algorithm. The block labeled $h(t'_n - mT)$ represents an elementary resampler and is described in chapter 2. For a practical implementation of this system, we replace the filter $h(t)$ with an FIR approximation. For efficiency, we use two filters in cascade; $h_{\mathrm{D}}[m]$ and $f(t)$ (see section 2.5). We may do this because the input grid is regular. This results in the system shown in figure 5-8.

So far we have assumed that $g(t)$ is known exactly. Since $g(t)$ depends only on the irregular grid $\{t'_n\}$, it may be possible to calculate it ahead of time and store it in memory. If we need to calculate $g(t)$ in real-time, the system presented in section 4.6 may be used.

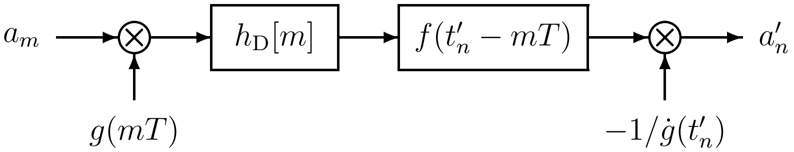

Figure 5-8: DT system which allows for a short time-varying filter.

5.4 The Missing Pixel Problem

In this section, we look at a specific example of lowpass reconstruction in which the irregular grid is obtained by deleting a single point from an otherwise regular grid. Specifically, consider flat-panel displays such as those used in modern television sets or in personal computers, which are made by placing thousands of light emitting diodes (LEDs) on a silicon wafer. Each LED corresponds to one color component of one pixel of the image. These displays are particularly expensive to manufacture because if one LED is defective, the entire wafer must be discarded. However, it may be possible to compensate for a defective LED by adjusting the intensity of neighboring LEDs. In this case, the non-zero pixel size along with the blurring introduced by the eye acts as the lowpass filter. The defective LED causes the reconstruction grid to be irregular. We refer to this as the missing pixel problem[1], and we present a solution which could lead to flat-panel displays which are less expensive to manufacture since the defective pixels could be corrected for by performing some processing on the signal.

In the previous sections we have developed a solution for the nonuniform lowpass reconstruction problem with a general irregular grid. However, in our motivating example the irregular grid has a very specific structure. In particular, it can be obtained by deleting a single point from a regular grid. In this section, we will consider this specific irregular grid for which there is a closed-form expression for $g(t)$. This leads to an algorithm which requires very little computation.

For notational convenience, we define the index set I to be the set of integers;

$$I = 0, \pm 1, \pm 2, \pm 3, \dots\ , \tag{5.22}$$

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

$$I' = \pm 1, \pm 2, \pm 3, \dots\ . \tag{5.23}$$

[1] We wish to acknowledge the contribution of Petros Boufounos in suggesting the missing pixel problem as an application of nonuniform reconstruction.

Let the grid $\{t'_n\}$ be equal to I', and let $x(t)$ be a signal in $\mathcal{B}_\gamma$, where $\gamma < \pi$. We wish to find coefficients $\{a'_n; n \in I'\}$ which satisfy

$$x(t) = \sum_{n \in I'} a'_n \phi(t-n), \tag{5.24}$$

or equivalently,

$$x(t) = \sum_{n=-\infty}^{\infty} a'_n \phi(t-n), \tag{5.25}$$

with

$$a'_0 = 0. \tag{5.26}$$

By requiring $a'_0 = 0$ we may take the sum over all integers.

In order to use the solution given by (5.10), we now derive closed form expressions for the functions $\{\psi_n(t); n \in I'\}$. It can be shown that $g(t)$ defined in (5.8) is given by

$$g(t) = \frac{\sin \pi t}{\pi t} = \operatorname{sinc} t. \tag{5.27}$$

In order to see this, recall that $g(t)$ is the function[2] whose zeros are given by the set $\{t'_n; n \in I'\}$, *i.e.*, the integers, with zero removed. We may therefore obtain $g(t)$ by dividing $\sin \pi t$ (which has a zero at every integer) by πt (which has a single zero at $t = 0$). By taking the derivative of (5.27), we get that

$$\dot{g}(t) = \frac{\pi(\cos \pi t)\pi t - \pi \sin \pi t}{(\pi t)^2}. \tag{5.28}$$

[2] Since we only deal with real-valued sampling times, $g(t)$ is a real-zero function [29], *i.e.*, it is the restriction to the real line of an entire function whose zeros all lie on the real axis. We know that $g(t) = \operatorname{sinc} t$ since this class of functions is uniquely determined by its zero crossings (to within a scale factor).

If we evaluate this derivative at each time instant $\{t'_n; n \in I'\}$, we get

$$\dot{g}(t'_n) = \dot{g}(n) = \frac{\cos \pi n}{n} = \frac{(-1)^n}{n}, \quad n \in I'. \tag{5.29}$$

We do not need to be concerned with the division by zero since zero is not in I'. From (5.7) we get that for $n \in I'$,

$$\begin{aligned} \psi_n(t) &= \frac{g(t)}{(t-n)\dot{g}(n)} \\ &= (-1)^n \frac{n \sin \pi t}{\pi t (t-n)} \\ &= \operatorname{sinc}(t-n) - (-1)^n \operatorname{sinc} t. \end{aligned} \tag{5.30}$$

Now we can compute each of the coefficients $\{a'_n; n \in I'\}$, by substituting (5.30) into (5.10), to give

$$a'_n = \int_{-\infty}^{\infty} x(t) \left[\operatorname{sinc}(t-n) - (-1)^n \operatorname{sinc} t\right] \, dt; \quad n \in I'. \tag{5.31}$$

In order to compute this integral, we observe that $x(t)$ is in $\mathcal{B}_\gamma$ and is therefore also in $\mathcal{B}_\pi$. Thus $x(t)$ satisfies

$$x(\tau) = \int_{-\infty}^{\infty} x(t) \operatorname{sinc}(t-\tau) \, dt. \tag{5.32}$$

By using this fact, we may evaluate (5.31) to give

$$a'_n = x(n) - (-1)^n x(0); \quad n \in I'. \tag{5.33}$$

Thus we have

$$x(t) = \sum_{n \in I'} \left[x(n) - (-1)^n x(0)\right] \phi(t-n). \tag{5.34}$$

(5.33) gives us the desired values $\{a'_n; n \in I'\}$ in terms of the CT signal $x(t)$. However, in the context of resampling, we do not have access to $x(t)$, but only to its samples $\{a_n\}$ on a uniform grid $\{t_n\}$, where

$$a_n = x(t_n) = x(n); \quad n \in I. \tag{5.35}$$

The input grid $\{t_n\}$ is just the regular grid defined by $\{t_n = n; n \in I\}$. Thus (5.33) can be written in terms of our input sequence $\{a_n\}$ as

$$a'_n = a_n - (-1)^n a_0; \quad n \in I', \tag{5.36}$$

The resulting resampling algorithm is remarkably simple, with each output point depending only on two input points. It is also very efficient computationally, requiring no multiplications, only a single addition and possibly a change of sign for each output sample. In the context of the LED flat-panel display example which we consider, this solution is not very attractive since it requires us to adjust every pixel in the image. We would prefer a solution in which effect of the missing pixel is localized in some sense.

5.4.1 Generalization of the Solution

The solution given by (5.36) is not unique, in that it is not the only solution which satisfies (5.24). However, the form of (5.36) suggests a more general solution. We will consider sequences $\{a'_n\}$ which can be expressed as

$$a'_n = a_n - b_n; \quad n \in I. \tag{5.37}$$

The problem then reduces to finding an appropriate sequence $\{b_n\}$ so that $\{a'_n\}$ will satisfy (5.25) and (5.26). Clearly we have not excluded any solutions by considering only those of the form (5.37), because any solution can be expressed in this form by letting $b_n = a_n - a'_n$. We now determine the precise conditions on $\{b_n\}$ such that (5.37) is a valid solution.

From (5.26) and (5.37), we see that

$$b_0 = a_0. \tag{5.38}$$

This is the first condition that $\{b_n\}$ must satisfy. Now we substitute (5.37) into (5.25) to get

$$\begin{aligned} x(t) &= \sum_{n=-\infty}^{\infty} (a_n - b_n)\phi(t-n) \\ &= \sum_{n=-\infty}^{\infty} a_n\phi(t-n) - \sum_{n=-\infty}^{\infty} b_n\phi(t-n). \end{aligned} \tag{5.39}$$

Since $x(t) \in B_\gamma$, it can be expressed as the sum

$$x(t) = \sum_{n=-\infty}^{\infty} x(n)\phi(t-n). \tag{5.40}$$

Note that this is not the usual sinc interpolation formula because the functions $\{\phi(t-n)\}$ are shifted by integer amounts, while the zeros these functions do not necessarily occur at integer locations. However, because $x(t)$ is actually bandlimited to γ, (5.40) is still valid. By comparing (5.40) and (5.39), we get that

$$\sum_{n=-\infty}^{\infty} b_n\phi(t-n) = 0, \tag{5.41}$$

which is the second condition on $\{b_n\}$. Any sequence $\{b_n\}$ which satisfies (5.38) and (5.41) will give a valid solution.

It can be shown that (5.41) is equivalent to the condition that $B(\omega)$, the DT Fourier transform of $\{b_n\}$ satisfies

$$B(\omega) = 0; \quad |\omega| \le \gamma. \tag{5.42}$$

In other words, $\{b_n\}$ must be highpass[3].

[3] Note that $\{b_n\}$ is not allowed to have a frequency component of infinite-energy at $\omega = \gamma$.

We will now use these two conditions to verify that the solution given by (5.36) is valid. In this case, the sequence $\{b_n\}$ is given by

$$b_n = (-1)^n a_0, \tag{5.43}$$

which clearly satisfies (5.38). If we substitute (5.43) into (5.41), the left hand side becomes

$$a_0 \sum_{n=-\infty}^{\infty} (-1)^n \phi(t-n). \tag{5.44}$$

We recognize this sum as the DT Fourier transform of $\phi(t-n)$ evaluated at $\omega = \pi$. Here t is fixed, and n is the time variable. It can be shown that the DT Fourier transform of the sequence $\{\phi(n-t); n \in I\}$ is

$$\Phi(\omega) = \begin{cases} e^{-j\omega t}, & |\omega| < \gamma; \\ 0, & \gamma \leq |\omega| \leq \pi, \end{cases} \tag{5.45}$$

which is zero at $\omega = \pi$. Thus condition (5.41) is satisfied and we have verified that (5.43) is a valid solution.

The two conditions which $\{b_n\}$ must satisfy, (5.38) and (5.41), do not specify a unique solution. Thus we may attempt to find an optimal solution by choosing an appropriate optimality criterion. For example in our video display problem , we wish to localize the effects of the missing pixel. It may therefore be reasonable to minimize the l_2-norm of $\{b_n\}$ (subject to the two constraints mentioned earlier). If we solve this optimization problem, we obtain

$$b_n = (-1)^n \operatorname{sinc}\left(\frac{\pi - \gamma}{\pi} n\right). \tag{5.46}$$

This solution is more appealing than the one given by (5.43) because b_n now decays as we move further away from the missing pixel. However it is still infinite in extent.

5.4.2 Optimal FIR Approximation

In this section, we only allow a finite number of pixels to be adjusted. We derive a solution which is optimal in that it minimizes the L_2-norm of the reconstruction error. We also allow for an arbitrary reconstruction filter, $\phi(t)$, which may not be the sinc function given by (5.2)

Consider a signal $x(t)$, not necessarily bandlimited, which can be expressed as

$$x(t) = \sum_{n=-\infty}^{\infty} a_n \phi(t-n), \tag{5.47}$$

where $\phi(t)$ is some arbitrary reconstruction filter. The approximate reconstruction $x'(t)$ is given by

$$x'(t) = \sum_{n=-\infty}^{\infty} a'_n \phi(t-n), \tag{5.48}$$

where $\{a'_n\}$ is given by (5.37). The sequence $\{b_n\}$, which must satisfy condition (5.38), is now assumed to be finite in extent, *i.e.*, it is nonzero only for $n \in \mathcal{N}$, where $\mathcal{N}$ is the set of points which may be adjusted, in addition to the point at the origin. If we only allow only N points to be adjusted, $\mathcal{N}$ will contain $N+1$ points. Let $e(t)$ be the difference between the ideal and approximate reconstructed signals;

$$e(t) = x(t) - x'(t). \tag{5.49}$$

By substituting (5.47), (5.48) and (5.37), we get

$$e(t) = \sum_{n \in \mathcal{N}} b_n \phi(t-n). \tag{5.50}$$

Let $\mathcal{E}$ be the L_2-norm of this reconstruction error $e(t)$, *i.e.*,

$$\mathcal{E} = \left(\int_{-\infty}^{\infty} |e(t)|^2 \, dt \right)^{\frac{1}{2}}. \tag{5.51}$$

We wish to minimize $\mathcal{E}$, but since it is positive, we may instead minimize $\mathcal{E}^2$. By substituting (5.50) we get

$$\mathcal{E}^2 = \int_{-\infty}^{\infty} \left(\sum_{n \in \mathcal{N}} b_n \phi(t-n) \right)^2 \, dt. \tag{5.52}$$

In order to minimize this, we find the partial derivative with respect to b_k for some $k; k \neq 0; k \in \mathcal{N}$.

$$\begin{aligned} \frac{\partial}{\partial b_k} \mathcal{E}^2 &= \int_{-\infty}^{\infty} 2 \left(\sum_{n \in \mathcal{N}} b_n \phi(t-n) \right) \phi(t-k) \, dt \\ &= 2 \sum_{n \in \mathcal{N}} b_n \int_{-\infty}^{\infty} \phi(t-n)\phi(t-k) \, dt \\ &= 2 \sum_{n \in \mathcal{N}} b_n \theta(k-n), \end{aligned} \tag{5.53}$$

where $\theta(t)$ is the deterministic autocorrelation function of $\phi(t)$, and is defined as

$$\theta(t) = \int_{-\infty}^{\infty} \phi(\tau)\phi(\tau - t) \, d\tau. \tag{5.54}$$

By setting the derivative to zero and substituting $b_0 = a_0$, we get

$$\sum_{\substack{n \in \mathcal{N} \\ n \neq 0}} b_n \theta(k-n) = -a_0 \theta(k); \quad k \in \mathcal{N}, k \neq 0. \tag{5.55}$$

This gives a system of N equations and N unknowns, which can be solved by performing a $N \times N$ matrix inversion.

We have performed the matrix inversion numerically and calculated the solution for a number of examples in which $\phi(t)$ is an ideal lowpass filter given by (5.2), and $a_0 = 1$. Figure 5-9 shows b_n for $N = 6$ and $N = 20$ and for $\gamma = 0.7\pi$ and $\gamma = 0.9\pi$. We have assumed that an equal number of points are to be adjusted on either side of the missing pixel. Figure 5-10 shows how $\mathcal{E}$ varies with N for $\gamma = 0.7\pi, 0.8\pi, 0.9\pi$ (solid, dashed and dotted lines respectively). From this graph, we see that $\mathcal{E}$ falls exponentially in N as expected.

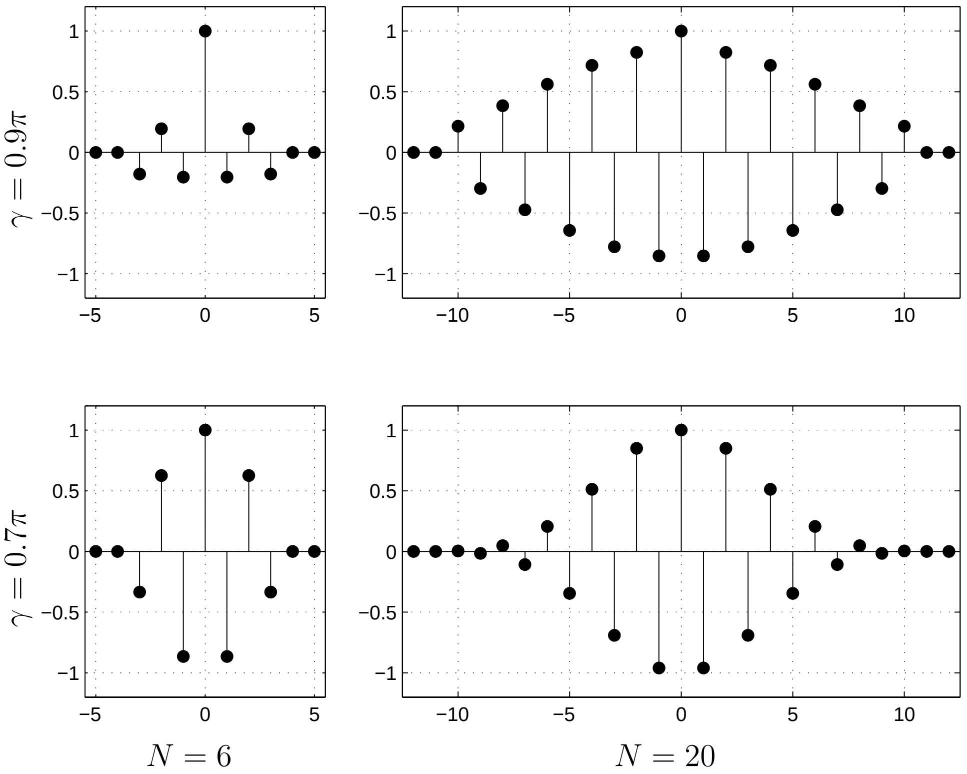

Figure 5-9: Examples of optimal sequences $\{b_n\}$.

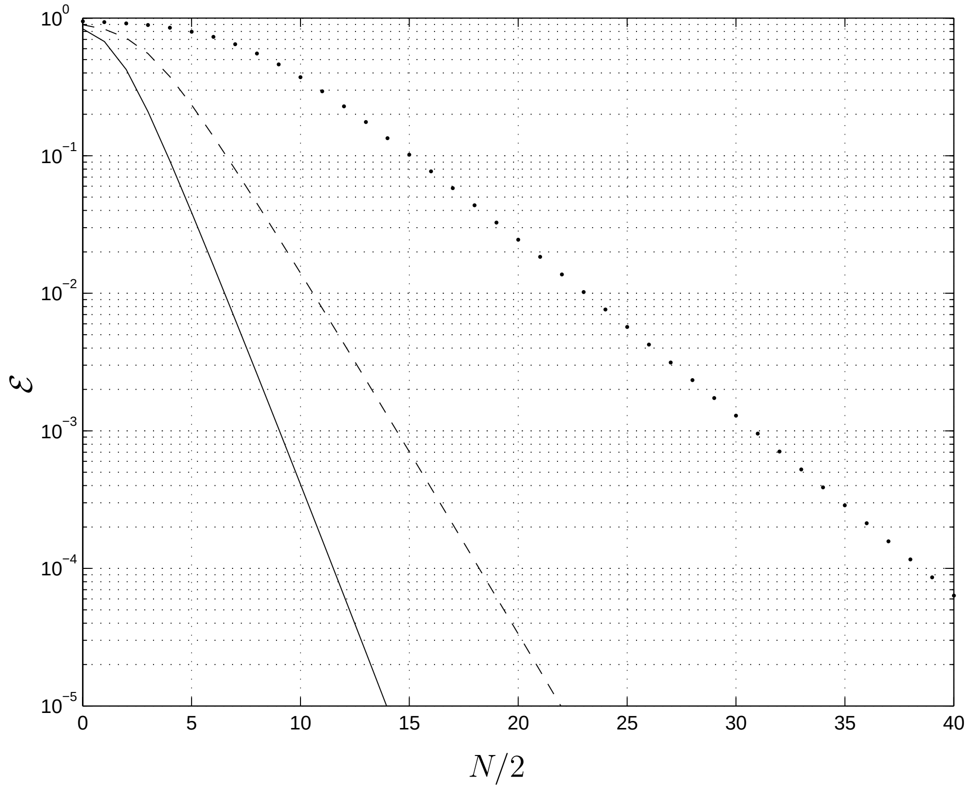

Figure 5-10: $\mathcal{E}$ as a function of N for different γ.

The system of equations given by (5.55) can easily be extended to the higher dimensional case by allowing k and n to be vectors. Figure 5-11 shows an example for a two dimensional image. In this example, 0.3% of the pixels have been randomly chosen as “missing”. The upper plot shows what a video display with broken elements might look like. The lower plot shows the result when our algorithm has been used to adjust the values of neighboring pixels. The image should be viewed from a distance of about five feet. In this example, we have allowed a 5×5 region of pixels to be adjusted, centered on the missing pixel. Thus, for each missing pixel, $N = 24$ other pixels are adjusted. Note that the matrix inversion was only be performed once to obtain the sequence b_n, which was then scaled and added to the image at different locations, corresponding to the missing pixels. Only N multiplies and adds need to be performed for each missing pixel. In order to obtain b_n, we assumed that the spatial frequency response of the eye is a radially symmetric decaying exponential per [23]. While our solution does not perfectly compensate for the missing pixels, there does seem to be some improvement in perceived image quality.

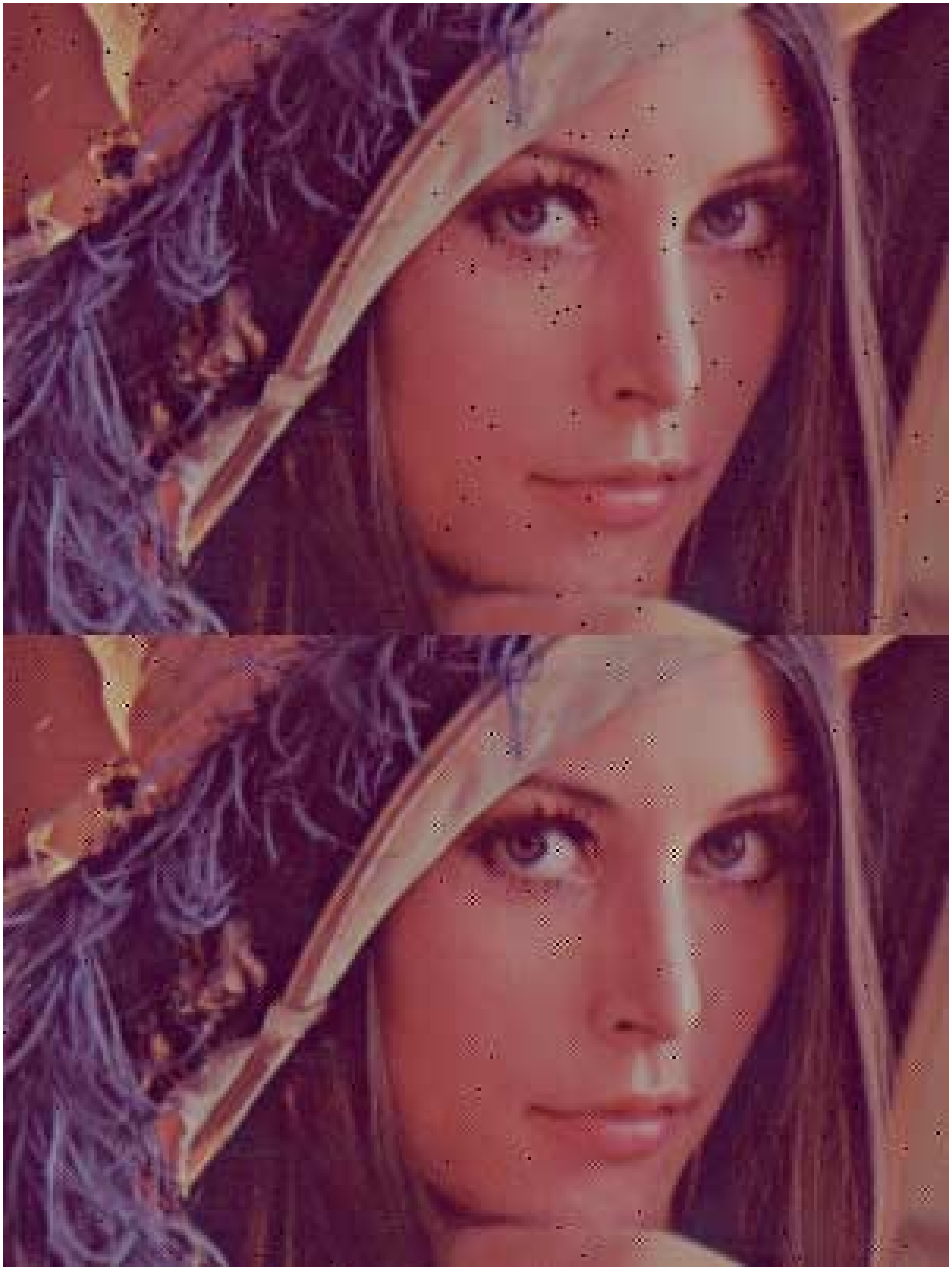

Figure 5-11: Simulated video display with missing pixels; uncorrected (top), and corrected using proposed algorithm (bottom). A viewing distance of five feet is recommended.

Bibliography

[1] A. Aldroubi and K. Gröchenig, "Nonuniform sampling and reconstruction in shift-invariant spaces," *SIAM Review*, vol. 43, no. 4, pp. 585–620, 2002. http://www.siam.org.journals/sirev/43-4/38698.html.

[2] F. J. Beutler, "Error-free recovery of signals from irregularly spaced samples," *SIAM Rev.*, vol. 8, pp. 328–335, July 1966.

[3] H. S. Black, *Modulation Theory.* New York, NY: D. Van Nostrand Co., Inc., 1953.

[4] C. Cenker, H. G. Feichtinger, and M. Herrmann, "Iterative algorithms in irregular sampling a first comparison of methods," in *Proc. ICCCP*, (Phoenix, AZ), Mar. 1991.

[5] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing.* Prentice Hall Signal Processing Series, Einglewood Cliffs, New Jersey: Prentice Hall, 1983.

[6] R. J. Duffin and A. C. Schaeffer, "A class of nonharmonic Fourier series," *Trans. Am. Math. Soc.*, vol. 72, pp. 341–366, Mar. 1952.

[7] Y. C. Eldar, "Sampling and reconstruction in arbitrary spaces and oblique dual frame vectors," *J. Fourier Analys. Appl.* to appear.

[8] Y. C. Eldar and A. V. Oppenheim, "Filterbank reconstruction of bandlimited signals from nonuniform and generalized samples," *IEEE Trans. on Signal Processing*, vol. 48, pp. 2864–2875, Oct. 2000.

[9] H. Göckler, "Design of recursive polyphase networks with optimum magnitude and minimum phase," *Signal Processing*, vol. 3, pp. 365–376, Oct. 1981. North-Holland Publishing Company.

[10] K. Gröchenig, "Reconstruction algorithms in irregular sampling," *Mathematics of Computation*, vol. 59, pp. 181–194, July 1992.

[11] E. A. Guillemin, *Theory of Linear Physical Systems*. New York: John Wiley & Sons, 1963. Chapter XVIII.

[12] A. J. Jerri, "The Shannon sampling theorem—Its various extensions and applications: A tutorial review," *Proceedings of the IEEE*, vol. 65, pp. 1565–1596, Nov. 1977.

[13] B. F. L. Jr., "Theory of analytic modulation systems," *AT&T Bell Labs Tech. Jour.*, vol. 57, pp. 491–576, Mar. 1978.

[14] B. F. L. Jr., "Click modulation," *AT&T Bell Labs Tech. Jour.*, vol. 63, pp. 401–423, Mar. 1984.

[15] B. F. L. Jr., "Signals designed for recovery after clipping—I. Localization of infinite products," *AT&T Bell Labs Tech. Jour.*, vol. 63, pp. 261–285, Feb. 1984.

[16] J. F. Kaiser, "Nonrecursive digital filter design using the I_0-sinh window function," in *Proc. 1974 IEEE International Symp. on Circuits and Systems*, (San Francisco, CA), pp. 20–23, Apr. 1974.

[17] R. Lagadec and H. O. Kunz, "A universal, digital sampling frequency converter for digital audio," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, ((Atlanta, GA)), pp. 595–598, 1981.

[18] R. Lagadec, D. Pelloni, and D. Weiss, "A 2-channel, 16-bit digital sampling frequency converter for professional digital audio," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, ((Paris, France)), pp. 93–96, 1982.

[19] H. G. Martinez and T. W. Parks, “Design of recursive digital filters with optimum magnitude and attenuation poles on the unit circle,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-26, pp. 150–156, Apr. 1978.

[20] H. G. Martinez and T. W. Parks, “A class of infinite-duration impulse response digital filters for sampling rate reduction,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, pp. 154–162, Apr. 1979.

[21] F. Marvasti, M. Analoui, and M. Gamshadzahi, “Recovery of signals from nonuniform samples using iterative methods,” *IEEE Trans. Signal Processing*, vol. 39, pp. 872–877, Apr. 1991.

[22] Z. Mou and P. Duhamel, “Fast FIR filtering: Algorithms and implementations,” *Signal Processing*, vol. 13, pp. 377–384, Dec. 1987. North-Holland Publishing Company.

[23] R. Näsänen, “Visibility of halftone dot textures,” *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-14, pp. 920–924, Nov.–Dec. 1984.

[24] H. Nyquist, “Certain topics in telegraph transmission theory,” *AIEE Transactions*, vol. 35, pp. 617–644, 1928.

[25] A. Oppenheim and D. Johnson, “Discrete representation of signals,” *Proceedings of the IEEE*, vol. 60, pp. 681–691, June 1972.

[26] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing.* Prentice Hall Signal Processing Series, Einglewood Cliffs, New Jersey: Prentice Hall, 1983.

[27] T. A. Ramstad, “Digital two-rate IIR and hybrid IIR/FIR filters for sampling rate conversion,” *IEEE Transactions on Communications*, vol. 30, pp. 1466–1476, July 1982.

[28] T. A. Ramstad, “Digital methods for conversion between arbitrary sampling frequencies,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 577–591, June 1984.

[29] A. Requicha, "The zeros of entire functions: Theory and engineering applications," *Proceedings of the IEEE*, vol. 68, pp. 308–328, Mar. 1980.

[30] A. I. Russell, "Efficient rational sampling rate alteration using IIR filters," *IEEE Signal Processing Lett.*, vol. 7, pp. 6–7, Jan. 2000.

[31] A. I. Russell and P. E. Beckmann, "Sampling rate conversion." U.S. Patent (Pending). Bose Corporation, The Mountain, Framingham, MA.

[32] A. I. Russell and P. E. Beckmann, "Efficient arbitrary sampling rate conversion with recursive calculation of coefficients," *IEEE Trans. on Signal Processing*, vol. 50, pp. 854–865, Apr. 2002.

[33] T. Saramäki and T. Ritoniemi, "An efficient approach for conversion between arbitrary sampling frequencies," in *Proc. IEEE Int. Symp. Circuits and Syst.*, vol. 2, pp. 285–288, 1996.

[34] H. W. Schüssler and P. Steffen, *Some Advanced Topics in Filter Design* in Advanced Topics in Signal Processing, J. S. Lim and A. V. Oppenheim Eds. Einglewood Cliffs, New Jersey: Prentice Hall, 1988.

[35] C. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, pp. 10–21, 1949.

[36] J. O. Smith and P. Gossett, "A flexible sampling-rate conversion method," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 2, (New York), pp. 19.4.1–19.4.2, IEEE Press, March 1984.

[37] E. C. Titchmarsh, *Introduction to the Theory of Fourier Integrals, Second Edition.* London: Oxford University Press, 1948. Chapter V.

[38] M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Processing Magazine*, vol. 16, November 1999.

[39] M. Unser, "Sampling—50 years after Shannon," *Proceedings of the IEEE*, vol. 88, pp. 569–587, Apr. 2000.

[40] P. P. Vaidyanathan, “Generalizations of the sampling theorem: Seven decades after Nyquist,” *IEEE Trans. Circuit Sys.* to appear.

[41] P. P. Vaidyanathan, *Multirate Systems and Filter Banks.* Prentice Hall Signal Processing Series, Einglewood Cliffs, New Jersey: Prentice Hall, 1993.

[42] R. G. Willey, “Recovery of bandlimited signals from unequally spaced samples,” *IEEE Trans. Commun.*, vol. COMM-26, pp. 135–137, Jan. 1978.

[43] K. Yao and J. Thomas, “On some stability and interpolatory properties of nonuniform sampling expansions,” *IEEE Transactions on Circuit Theory*, vol. CT-14, pp. 404–408, Dec. 1967.

[44] K. Yao and J. Thomas, “On a class of nonuniform sampling representation,” in *Symp. Signal Trans. Processing (sponsored by Columbia University)*, pp. 69–75, May 1968.

[45] J. L. Yen, “On nonuniform sampling of bandwidth-limited signals,” *IEEE Trans. Circuit Theory*, vol. CT-3, pp. 251–257, Dec. 1956.