# Geophysical Data Analysis

# GEOPHYSICAL DATA ANALYSIS: TIME SERIES

DUNCAN CARR AGNEW
ROBERT L. PARKER

SIO 223B CLASS NOTES, SPRING 2011

# Contents

**Contents** **i**

**1 Introduction** **1**
1.1 Random Time Series . . . . . 1
1.2 Overview of the Subject . . . . . 5

**2 Linear Systems and Fourier Transforms** **8**
2.1 Linear Time-Invariant Systems . . . . . 8
2.2 Convolution . . . . . 12
2.3 The Fourier Transform . . . . . 14
2.4 A Transform Pair . . . . . 15
2.5 Fourier Theorems I: Similarity and Shift . . . . . 17
2.6 Generalized Functions . . . . . 19
2.7 Fourier Transforms of Generalized Functions . . . . . 21
2.8 Fourier Theorems II: Derivatives and Differentiability . . . . 23
2.9 Fourier Theorems III: Convolution, Power, Modulation . . . . 25
2.10 The Correlation Function . . . . . 28

**3 Fourier Theory for Discrete Time** **30**
3.1 Introduction . . . . . 30
3.2 Discrete-Time Sequences and Operations . . . . . 31
3.3 Fourier Transforms for Infinite Sequences . . . . . 32
3.4 Discrete Fourier Transform . . . . . 36
3.5 Fourier Theorems for the DFT . . . . . 39
3.6 The Dirichlet Kernel . . . . . 46
3.7 Computing the DFT: the Fast Fourier Transform . . . . . 48

**4 Sampling Theory for Time-Series** **52**
4.1 Introduction: Logging Data . . . . . 52
4.2 The Sampling Problem . . . . . 54
4.3 The Nyquist Theorem . . . . . 55
4.4 Aliasing . . . . . 58
4.5 Decimation . . . . . 61

4.6 Violating the Nyquist Criterion . . . . . 62
4.7 Quantization Error . . . . . 63

**5 Digital Filters I: Frequency-Selective Filters 65**
5.1 Introduction: Digital Filters in General . . . . . 65
5.2 FIR Filters for Frequency Selection . . . . . 67

**6 Digital Filters II: Recursive Filters and Simulating Linear Systems 79**
6.1 Introduction . . . . . 79
6.2 Lumped-Parameter Systems . . . . . 79
6.3 The Laplace Transform: System Poles and Zeros . . . . . 84
6.4 The $z$-transform for Digital Filters . . . . . 87
6.5 Recursive Digital Filters . . . . . 90

**7 Digital Filters III: Differentiators, Least-Squares, and Minimum-Phase Filters 97**
7.1 FIR Filters: Other Applications . . . . . 97
7.2 Differentiators and Digital Hilbert Transformers . . . . . 97
7.3 Least-squares Fits as Filters . . . . . 100
7.4 Minimum-Phase Filters . . . . . 102

**8 Stochastic Processes 108**
8.1 Introducing Ordered Random Data . . . . . 108
8.2 Stationary Processes and Autocovariance . . . . . 111
8.3 White Noises and their Relatives . . . . . 113
8.4 Examples from the Real World . . . . . 117

**A Computing the Discrete Fourier Transform 121**
A.1 Introduction . . . . . 121
A.2 The Basis of the Fast Fourier Transform . . . . . 121
A.3 Computing DFT's of Real Series . . . . . 123
A.4 Computing the Fourier Transform for a Few Frequencies . . 125

**B Other Reading 128**
B.1 Introduction . . . . . 128
B.2 Linear Systems and Fourier Analysis . . . . . 128
B.3 Digital Signal Processing . . . . . 129
B.4 Time Series and Spectral Estimation . . . . . 130

**Bibliography** **131**

**Index** **134**

CHAPTER 1

# INTRODUCTION

## 1.1 Random Time Series

In this course we consider datasets in which the data have a particular structure, namely that the data are **ordered**, whether in time or space (or both); we shall use the term **time series** for the general case of data ordered in one dimension. Most data are actually ordered, though we may choose to ignore this; even repeated measurements of the same thing are usually done sequentially rather than all at once.

We begin with some mathematical concepts for ordered data. In fact, we start by focusing on the "ordered" part while ignoring the "random" part: we discuss concepts and methods useful for working with ordered data regarded as conventional (rather than random) variables. Only after we have introduced these concepts, in particular Fourier theory and methods, will we return to randomness. Our ultimate aim is to develop Fourier methods for random variables, which leads to the idea of the power spectrum: this spectrum, and the procedures for estimating it, will be our focus in the latter part of this section of the course.

### 1.1.1 Two Examples of Time Series

We start with some examples of ordered data to show some of what we will be discussing in more detail. Our first examples come from very close by: sea level at the end of the Scripps pier (Figure 1.1). The ordering of these data is crucial: if we selected values at random from those shown, we would lose significant information. Given the ordering, two obvious features are a periodic variation and a long-term rise. The periodic variation turns out to be annual,[1] so that a preliminary model for the data values as a function

[1] This comes from the "steric effect", and is caused by the annual variation in water temperature at shallow depths; as it warms, water expands, and level rises. This annual change does not reflect any annual variation in the amount of water in the oceans; to

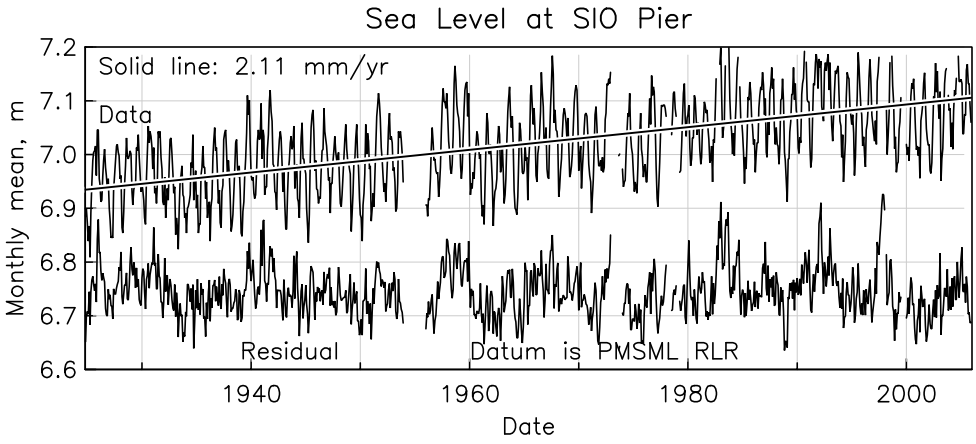



Figure 1.1: Monthly mean sea level measured at the end of the SIO pier; data from the Permanent Service for Mean Sea Level. The reference level is chosen to be well below the lowest water level.

of time $t$ (in years) is

$$x_i = a_1 + a_2 t_i + a_3 \sin(2\pi t_i) + a_4 \cos(2\pi t_i) + r_i$$

We can use the least-squares methods learned earlier to estimate the parameters $a_1$ through $a_4$; doing this, and subtracting the result from the data, gives the residuals $r_i$, also plotted in Figure 1.1. These residuals certainly cannot be called "error;" rather, these fluctuations in sea level are caused by changes in the ocean and atmosphere, and show, for example, a period of high sea level in the early 1980's, caused by an El Niño.

Our next example uses exactly same quantity (water level at the end of the Scripps pier), but measured at a much finer time resolution, to record the ocean waves. Figure 1.2 shows 10 minutes of 1-second data, 24 hours apart. The dominant behavior on both days is an oscillation with amplitude about 15 cm and period about 10 seconds: the ocean swell. More than this it is difficult to say.

Figure 1.3 shows the **power spectral density** (PSD) of these data, estimated from a longer sample (an hour of measurements). We define the PSD later; for now, a simple way of looking at it is that it shows how much of the variation is occurring at different frequencies. In this case, the PSD is large for periods of 8-15 seconds (frequencies of 0.06-0.12 Hz), as we

what extent the long-term change is steric, and to what extent driven by changes in water storage elsewhere, is much debated.

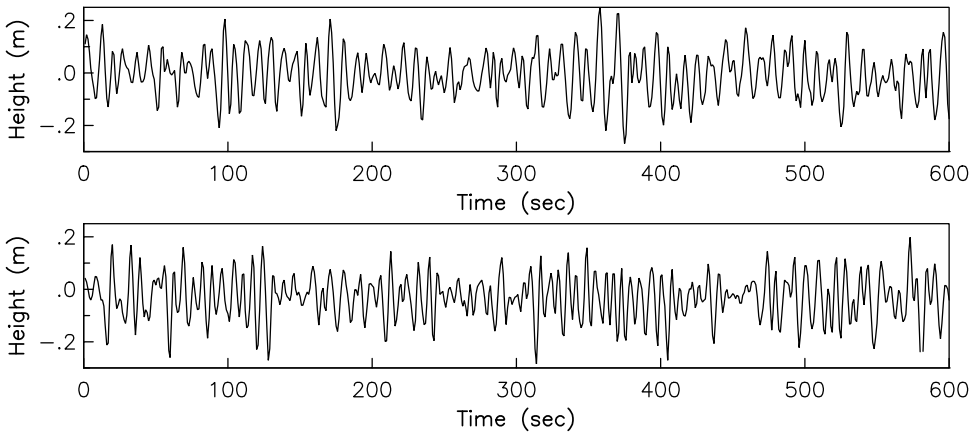



Figure 1.2: Sea level measured at the end of the SIO pier; data from the Coastal Data Information Program. Zero level is arbitrary (and not the same as in the previous figure). Each frame shows 10 minutes of data, on the top starting at 1997:240:00:00:00 and on the bottom one day later.

might expect from the plot of the time series. But we also can see that the decrease at higher and lower frequencies has very different forms (suggesting different physics). Further, we can see that the peak of the energy on day 241 has moved to a frequency 0.008 Hz higher than on day 240. If we take a series of such spectra and make a contour plot of the power spectral density over time, we see a "ridge line" of high values which shifts gradually with time. The explanation for this is that surface waves in liquids are dispersive; the wave energy was generated over a broad range of frequencies in one place (where there were high winds), but the lower-frequency energy traveled faster and arrived first.[2] Given the frequency shift over three days and the known physics of wave propagation, the distance to the source is found to be $64° \pm 1°$. We show this example to illustrate a common occurrence: characteristics of the data that are not obvious in a time series can become very obvious after we find the power spectrum of that data.

[2] See, for much more on this subject, based on an ocean-spanning data set, the classic paper by Snodgrass *et al.* (1966); a more recent summary, using satellite data, is Ardhuin *et al.* (2009).

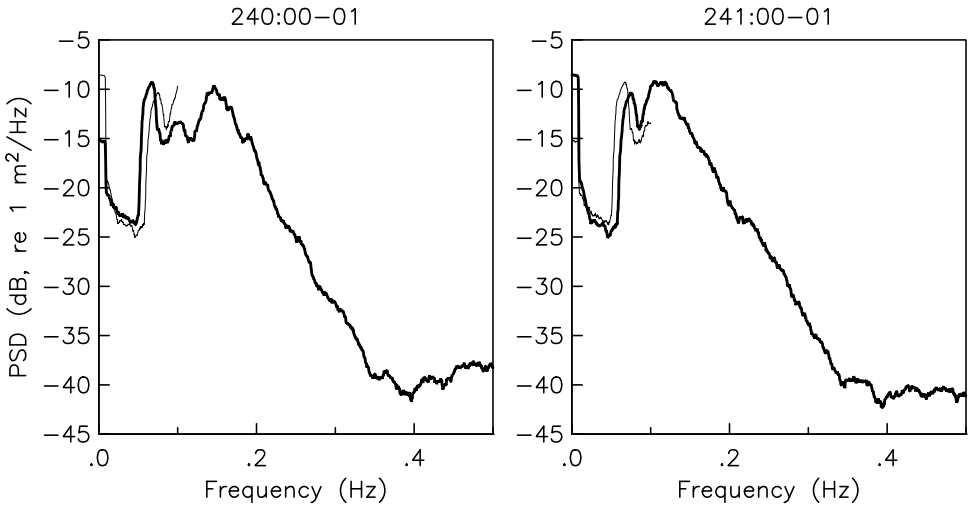



Figure 1.3: Spectrum of the wave data in Figure 1.2 (actually, of a full hour beginning at the times shown in that figure). In each frame the spectrum from the other frame is shown as a light line at the lowest frequencies, to indicate the shift in frequency of the lowest-frequency peak.

.

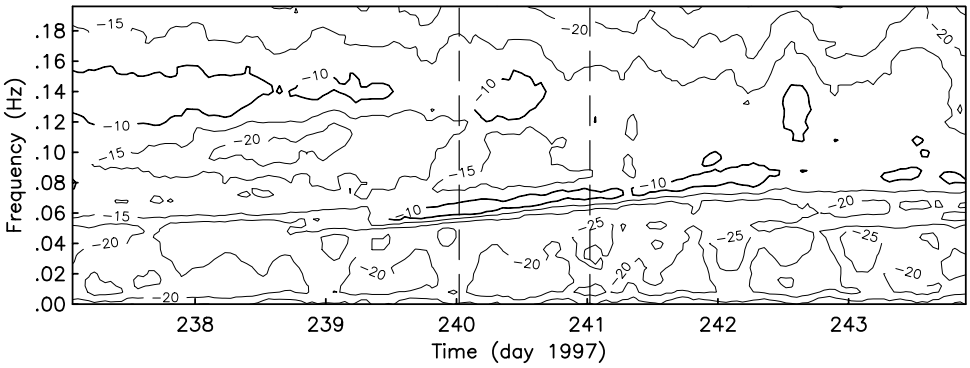



Figure 1.4: Power spectral density as a function of time for sea level at the end of the SIO pier. Contours are in dB relative to 1 $m^2$/Hz. The two dashed lines show the "slices" that give the spectra in Figure 1.3. .

### 1.1.2 A Note on Decibels

Figures 1.3 and 1.4 shows the amplitude of the spectrum plotted in units that may be new to you: **decibels (dB)**. These are another way of creating a logarithmic scale, and it is sufficiently common that you need to know what it is. The term comes, via acoustics, from telephony: hence the name, Bel, which refers to Alexander Graham Bell. The original definition is that if a signal has a power (think loudness) $P$, and there is a reference power $P_0$, the signal level in decibels is $10\log_{10}(P/P_0)$. The logarithm is used for two reasons: one is the usual reason that we can use a small range of numbers for very large changes. . The other is peculiar to acoustics, namely that the intensity of a sensation (e.g., perceived loudness) is roughly proportional to the logarithm of the power, a rule of psychophysics known as Fechner's law.

In digital signal processing, the reference level is taken to be unity. If we have something with amplitude $A$, then by definition the amplitude in dB is $10\log_{10}(A^2) = 20\log_{10}A$. The square comes from the fact that squared quantities relate to power: for example, if we have a voltage $V$ across a resistance $R$, the power dissipated in the resistor is $V^2/R$. With this definition, a factor of two in amplitude is very nearly equal to 6 dB; a factor of four to 12 dB, and a factor of eight to 18 dB. Conversely, if signal A is half as large as signal B, we would say "A is 6 dB below B"; if A had a value of one half, with the reference level being one, we would say that "A has a level of $-6$ dB". And after a little practice, you will too.

## 1.2 Overview of the Subject

A major aim of this course is to teach you to understand what a spectrum is, know how to estimate it from the data, and how to decide which of the bumps on the plotted spectrum mean something and which do not. But to get to this, we need to spend time introducing Fourier methods and some specialized parts of probability and statistics: all quite useful in themselves, but with the consequence that we will only discuss spectral methods towards the latter part of the course.

We start with mathematics applicable to ordered conventional variables, in particular the **Fourier analysis** of functions and **linear systems theory**. As we just saw, looking at data in terms of frequency rather than time can be a very powerful tool - and this is what Fourier methods are about: moving between a function defined in time, and its transform de-

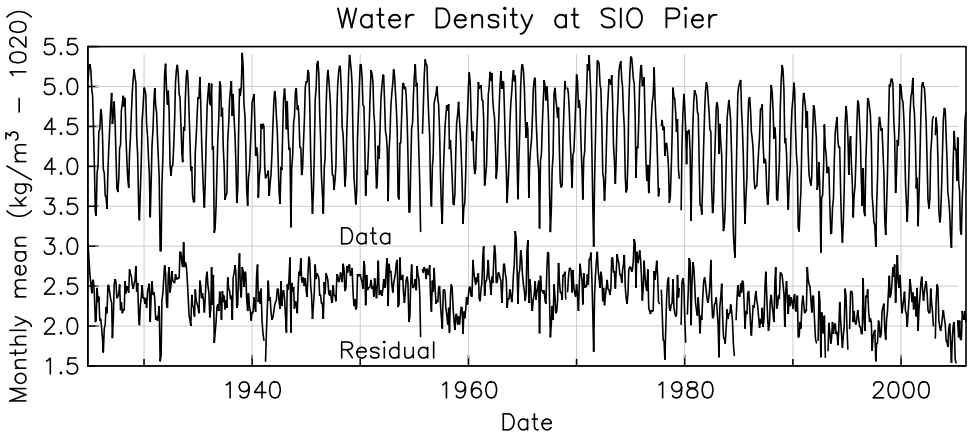



Figure 1.5: Monthly mean water density at the end of the SIO pier, found from the daily data for salinity and temperature archived at `http://shorestation.ucsd.edu/`. The upper trace is the raw data; the bottom trace is the residual (offset for clarity) after fitting and removing an annual cycle.

fined in frequency. Linear systems theory helps to explain why this procedure is so useful, and also provides a useful way of looking at many phenomena.

This way of looking at time series was developed by electrical engineers (and mathematicians who worked in collaboration with them), something also true of our next topic, **digital signal processing**. This name covers a range of concepts and methods: notably how to look at Fourier theory applied to time series in which the data occur as samples from a continuous function, and how to filter data for various purposes.

After this we move to ordered random variables. The subfield of probability that deals with random variables ordered in one or more dimensions is called the study of **stochastic processes**; we will utilize this mathematics to create descriptions that help us understand the process. For non-ordered random variables such descriptions include things like the moments of the probability density function (pdf); for time series we will develop the theory of the power spectrum, or power spectral density.

Of course, defining the mathematics of ideal models is only part of working with data: we also have to develop methods for getting a "good" estimate of this model from actual data. This, like all estimation from sets of data modeled as random variables, is a problem in statistics, not probability; within statistics it falls into the subfield termed (by statisticians)

**time-series analysis**. The concepts for estimating other parameters (e.g., bias, efficiency, consistency) apply equally to power spectrum estimation; but spectrum estimation is more complex, in part because the spectrum is a function, not just a single parameter or a set of them. Because of this complexity, this area suffers from much confusion and reliance on obsolete procedures. We will introduce you to useful and modern methods.

The power spectrum is a way of looking at a single set of ordered random variables. Often we have more than one set, and want to know how to relate them. To take an example, Figure 1.5 shows the water density at the end of the Scripps pier; if we compare this with the sea-level data in Figure 1.1, we can see that both show an annual cycle. We might ask if these two series are related at other frequencies, and if so how: to answer this question we would employ a generalization of the power spectrum known as the **cross-spectrum**, for which there are, again, good and bad estimation methods.

Another generalization of the power spectrum comes when we want to describe a time series in which the "spectrum" (in a rough sense of the amount of energy at different frequencies) changes with time. The data in Figures 1.2 through 1.4 is an example, though an "easy" case because the spectrum changed over times that were long (days) compared to the reciprocal of the frequencies involved (seconds to minutes). But there can be more rapid changes in frequency content, as, for example, in most seismograms. To describe such a process (which is termed **nonstationary**) we would need to replace the mathematical framework of Fourier analysis by what is called **wavelet analysis** – a topic we will not have time to discuss.

You should realize that there are many other areas of time series analysis that we will not touch on at all; we hope that what we do discuss will provide a useful framework for any topics you may want to learn about later.

CHAPTER 2

# LINEAR SYSTEMS AND FOURIER TRANSFORMS

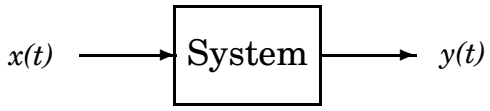
## 2.1 Linear Time-Invariant Systems

We begin with the mathematical theory – Fourier analysis – that underlies many kinds of analysis of time series, including signal processing, data filtering, and spectrum estimation. In order to make this theory usable, we will need to introduce new kinds of functions, called generalized functions.

Fourier analysis, as we will use it, is about representing a function, not as something that depends on the original variable (time) but as something that depends on a different variable, frequency. We can use this alternate representation because we can write the original function in terms of the "sum" of a series of sines and cosines in time, these sines and cosines being "indexed" by this frequency variable. We have used quotes around "sum" and "index" because these are only analogies to what actually done in Fourier transforms of functions on the real line – though, when we talk about digital data series we could remove the quote marks, since in that case these terms are correct.

We can construct functions from sums of a number of different families of functions; for example, over a finite interval we can form any function from sums of polynomials. Why use sines and cosines? There are number of ways of answering this; from the standpoint of geophysical data the answer lies in the ubiquity of **linear time-invariant systems**, and the especially simple way in which sinusoids are treated by such systems.

What do we mean when we refer to a system? It is just something into which we put an input function $x(t)$ (a function of some variable, usually thought of as time), and out of which we get an output function $y(t)$. We can write this symbolically, either as a block diagram (as shown below) or symbolically, as $x(t) \Rightarrow y(t)$.




Actual systems may well have more than one $x$ and $y$, but we avoid this complication for now. Some examples of systems are:

| **Input** | **System** | **Output** |
|---|---|---|
| Ground motion | Seismometer | Voltage |
| Magnetized sea-floor | 3-4 km of water | Surface magnetics |
| Solar-lunar gravity | Oceans | Ocean tides |
| Ice loads | Upper Mantle | Postglacial rebound |
| Wind | Oceans | Swell |
| Solar radiation | Atmosphere & ocean | Weather & climate |

These systems are given in order of increasing difficulty in studying them. The first three are examples of systems that are **linear** and **time-invariant**, which are the ones we will focus on. The last two are definitely not linear, and as a result are by far the least understood. The fourth, postglacial rebound, may or may not be linear depending on the rheology of the mantle; the advantages of linearity are shown by the almost-universal use of linear models of viscoelasticity in studying this problem.

The important point about linear time-invariant systems is that they are basically all alike – the same mathematics applies to them all – whereas nonlinear systems can be nonlinear in many different ways.

A system is linear if

$$x_1(t) \Rightarrow y_1(t) \quad \text{and} \quad x_2(t) \Rightarrow y_2(t)$$
$$\text{implies that}$$
$$x_1(t) + x_2(t) \quad \Rightarrow \quad y_1(t) + y_2(t)$$

which is often called the **principle of superposition**. Repeated application of this definition shows that

$$ax(t) \Rightarrow ay(t)$$

for any rational value of $a$; it can be shown to be true for any real value of $a$.

Time-invariance is a separate behavior, and means that for a particular input the output does not depend on the absolute time:

$$x(t+\tau) \Rightarrow y(t+\tau)$$

for all $\tau$. An example of a linear system that would not be time invariant is a pendulum whose length varies over times much longer than its period; even though a simple pendulum is linear in its response to small forces applied to the bob, gradual changes in length would keep it from being time-invariant. Time-invariance automatically holds for any system that depends only on physical laws, as in the magnetics example above; and can often be assumed as an idealization for many actual systems. One of the linear systems mentioned above, the ocean tides, will vary as the shape of the oceans changes over geologic time because of sea-level changes and continental motion; but over the past few centuries we can reasonably assume time invariance.

What do these characterizations of systems have to do with Fourier methods? The answer is that sinusoids (or indeed any exponential) are unchanged "in form" by a linear time-invariant system: the output and the input have the same functional representation. One way to see this is to note that shifting an exponential $e^{st}$ by a time $\tau$ simply multiplies it by $e^{s\tau}$, and a linear system does not change the form of the output because the amplitude of the input changes. More formally (and working with complex exponentials), we suppose that

$$x(t) = e^{2\pi i f t} \quad \Rightarrow \quad y(t) = \tilde{g}(f,t)e^{2\pi i f t}$$

which, since $\tilde{g}(f,t)$ is unspecified, allows $y$ to be an arbitrary function of time. Then applying a time shift, and time-invariance, gives

$$x(t+\tau) = e^{2\pi i f \tau} e^{2\pi i f t} \qquad \Rightarrow \qquad \tilde{g}(f,t+\tau)e^{2\pi i f \tau} e^{2\pi i f t}$$

Linearity allows us to cancel out the constant factor $e^{2\pi i f \tau}$ from both sides, with the result

$$e^{2\pi i f t} \qquad \Rightarrow \qquad \tilde{g}(f,t+\tau)e^{2\pi i f t}$$

which implies

$$\tilde{g}(f,t+\tau) = \tilde{g}(f,t)$$

But this means that $\tilde{g}$ can depend only on $f$; the time-dependence of input and output are then the same, namely a complex exponential, which is to say a sinusoid.

We call $\tilde{g}(f)$ the **frequency response** of the system, where $f$ is the **frequency** of the sinusoid. In general $\tilde{g}(f)$ will be complex (just as $x$ and $y$ are in our example), even though actual series are usually real. This use of

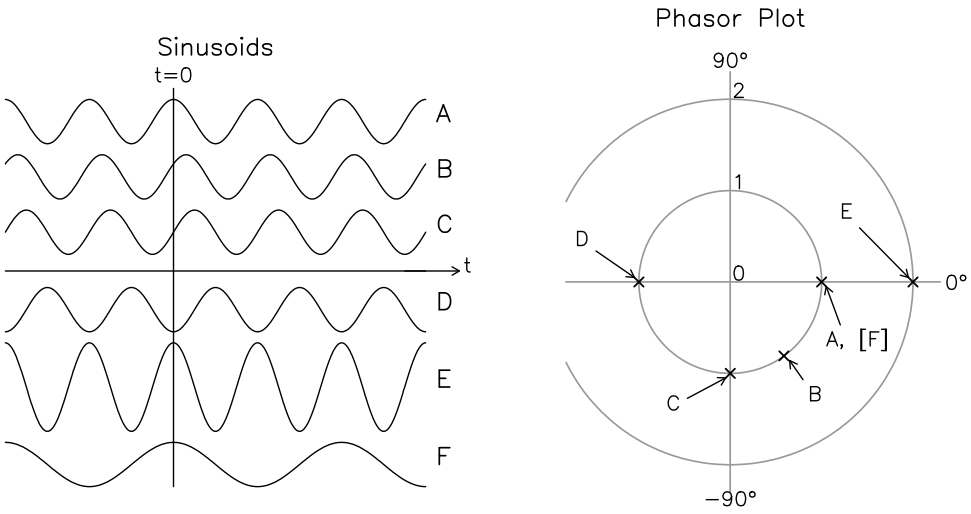



Figure 2.1: Some sinusoids, and their amplitude and phase displayed on a phasor diagram. **A** is a unit sinusoid with 0° phase shift (note that this depends on where we put the $t = 0$ point). **B** is a unit sinusoid with −30° phase shift (a delay); **C** is a unit sinusoid with −90° phase shift (we say that it is **in quadrature** with **A**); **D** is a unit sinusoid with ±180° phase shift; and **E** is a sinusoid with amplitude 2 and a 0° phase shift. **F** is another unit sinusoid with 0° phase shift, but a different frequency; in the phasor diagram, it plots in the same location as **A** though usually a single phasor diagram is taken to refer to sinusoids of the same frequency.

complex numbers simplifies the bookkeeping; what we really mean when we say that the input is $Ae^{2\pi ift}$ is that

$$x(t) = \mathscr{R}[Ae^{2\pi ift}] = \tfrac{1}{2}[Ae^{2\pi ift} + A^* e^{-2\pi ift}]$$

$$= \mathscr{R}[A]\cos 2\pi ft - \mathscr{I}[A]\sin 2\pi ft \stackrel{\text{def}}{=} a\cos(2\pi ft + \phi)$$

so that we can write the complex amplitude $A$ as $a(\cos\phi + i\sin\phi) = ae^{i\phi}$, a complex number with amplitude $a$ and phase $\phi$. We thus can represent our sinusoid on the Argand diagram often used for complex numbers; in this application it is called the **phasor diagram** (or vibration diagram) for this sinusoid. The modulus of $A$ is the amplitude and the angle $\phi$ the phase (relative to some time, which is arbitrary but must be chosen). Figure 2.1 shows some examples.

It is very important to be aware of an arbitrary part of the amplitude and phase representation, namely which sign of $\phi$ represents a lag (or lead).

Consider our sample sinusoid, $e^{2\pi i f t}$. If $\phi = 0$, a maximum will occur at $t = 0$. If $\phi$ is slightly greater than 0, this maximum will be at $t < 0$, and the waveform will be advanced (we reach the maximum sooner in time) relative to $\phi = 0$; similarly for $\phi < 0$, we reach the maximum later (a delay). Thus, $\phi < 0$ is a **phase lag**, and $\phi > 0$ a **phase lead**; we will see shortly that this convention corresponds to a particular choice of how we define the Fourier transform. Unfortunately there is no universal agreement on this sign convention; our choice is that used in electrical engineering (and hence in signal processing), but much of the older geophysical literature (for example, in tidal studies) takes phase lags to be positive.

## 2.2 Convolution

We have shown that what a linear time-invariant system does to a sinusoid is specified by its frequency response $\tilde{g}(f)$; but what about more general inputs? We shall simply state that, in general, we can relate the output $y(t)$ to the input $x(t)$ through the **convolution integral**

$$y(t) = \int_{-\infty}^{\infty} x(u)g(t-u)\,du$$

where $g(t)$ is a function characteristic of the system. We say that $y$ is the **convolution of** $x$ **and** $g$, which we write as $y(t) = x(t) * g(t)$,

It is clear that the system described by this integral is linear; to show that it is time-invariant we first show it to be commutative. Letting $u' = t - u$ we see that

$$\int_{-\infty}^{\infty} x(u')g(t-u')du' = \int_{-\infty}^{\infty} x(t-u)g(u)\,du$$

which means that $x * g = g * x$ Delaying $x(t)$ by $\tau$ is expressed in the convolution by $x(t-\tau)$, which gives

$$\int_{-\infty}^{\infty} x(t-\tau-u)g(u)du = y(t-\tau)$$

so convolution is time-invariant.

There are lots of ways of looking at convolutions; see Bracewell (1986) for a selection. The most direct way of viewing convolution, at least for those who think visually, is to imagine a plot of $x(t)$ (strictly, $x(u)$), together with one of $g$ (plotted reversed because it is $g(t-u)$). The position of $g$

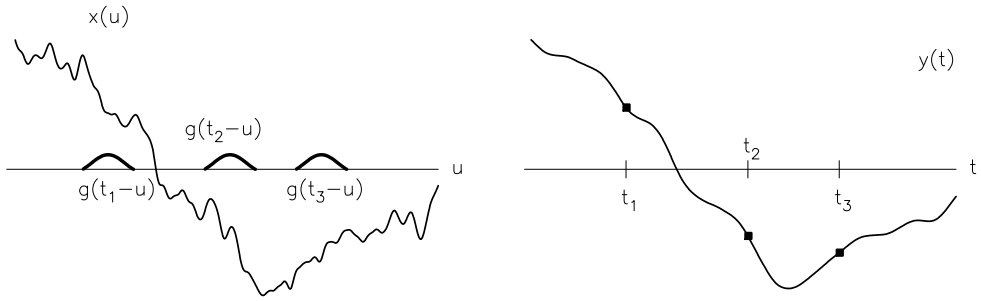


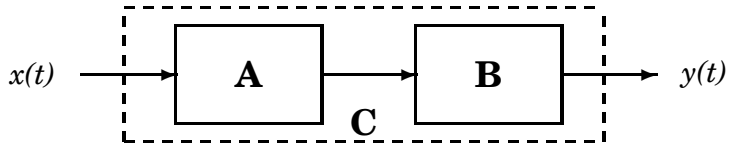

Figure 2.2: A graphical representation of convolution. The left panel shows $x(u)$, and $g(t-u)$ for three values of $t$. The dots in the right panel show the integral of the product $gx$ for these three values; when this operation is performed for all values of $t$, we get the function $y(t)$ as shown.

depends on the value of $t$; as $t$ increases, $g$ moves to the right. For any value of $t$, we form the product $gx$, and the value of $y$ is just the area under this product function. Figure 2.2 illustrates the process. To develop a better sense of how convolution works, it can be useful to practice drawing functions and sliding one past the other.

Convolution is also associative, as we can show by writing the following series of integrals, using changes of variables to get through most steps.

$$\begin{aligned}
(x*y)*z &= \int_{-\infty}^{\infty} z(t-\tau)\int_{-\infty}^{\infty} x(u)y(\tau-u)\,du\,d\tau \\
&= \int_{-\infty}^{\infty} x(u)\int_{-\infty}^{\infty} y(\tau-u)z(t-\tau)\,d\tau\,du \\
&= \int_{-\infty}^{\infty} x(u)\int_{-\infty}^{\infty} y(v)z(t-v-u)\,dv\,du \\
&= \int_{-\infty}^{\infty} x(u)\int_{-\infty}^{\infty} y(v)z[(t-u)-v]\,dv\,du \\
&= \int_{-\infty}^{\infty} x(u)w(t-u)\,du = x*(y*z)
\end{aligned}$$

What this means in terms of linear systems is that we can aggregate them together as we see fit, as suggested in the accompanying sketch, where the systems A and B can be viewed separately or as a single system C. While this result may seem a bit trivial, it is actually very important: the concept of linear systems is so useful partly because we can combine simple systems together to make complicated ones.




## 2.3 The Fourier Transform

We are now ready to discuss the Fourier transform of a function. We have seen that exponentials $e^{2\pi i f t}$ are modified in an especially simple way by linear time-invariant systems; but how can we use exponentials for a more general input? The most obvious method is to take a sum of exponentials, otherwise known as a **Fourier series**:

$$x(t) = \sum_n \tilde{x}_n e^{2\pi i f_n t}$$

where the sum is over some finite set of frequencies $f_n$ There are some geophysical signals that can be represented in this way, most notably the tides, for which this is called a harmonic development.

But the Fourier series certainly cannot represent an arbitrary function; for one thing, it cannot represent a **transient** (a function that is zero outside some range of $t$). So we generalize the weighted sum to an integral, and write

$$x(t) = \int_{-\infty}^{\infty} \tilde{x}(f) e^{2\pi i f t}\, df \tag{2.1}$$

where $f$ is now the numbers on the real line, not just a finite set of values. It can be shown that this integral holds if and only if

$$\tilde{x}(f) = \int_{-\infty}^{\infty} x(t) e^{-2\pi i f t}\, dt \tag{2.2}$$

These two equations define a function, $\tilde{x}(f)$, that is the **Fourier transform** of the function $x(t)$, which we write as $\mathscr{F}[x(t)] = \tilde{x}(f)$.[1] We say that equations (2.1) and (2.2) define a **transform pair**, (2.2) being the **forward** and (2.1) the **inverse** Fourier transform.

[1] Our notation for the Fourier transform of $x(t)$ being $\tilde{x}(f)$ (the tilde is meant to remind you of the sinusoid) is different from (Bracewell, 1986), who uses regular capital letters (e.g., $X(f)$) for the transforms of functions; unfortunately, this conflicts with the usage of capital and lower-case letters in probability and statistics. Another common notation denotes the Fourier transform of $x(t)$ by $\hat{x}(f)$ – though this, too, conflicts with statistical notation.

Any user of canned programs or other people's formulae should be aware that the definitions (2.1) and (2.2) are common but not universal. One alternative usage writes (2.2) using radian frequency $\omega$ rather than cyclic frequency $f$. The transform pair then becomes

$$\tilde{x}(\omega) = \int_{-\infty}^{\infty} x(t) e^{-i\omega t}\, dt \qquad x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \tilde{x}(f) e^{i\omega t}\, df$$

or

$$\tilde{x}(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} x(t) e^{-i\omega t}\, dt \qquad x(t) = \int_{-\infty}^{\infty} \tilde{x}(f) e^{i\omega t}\, df$$

or

$$\tilde{x}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} x(t) e^{-i\omega t}\, dt \qquad x(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \tilde{x}(f) e^{i\omega t}\, df$$

A more insidious danger is that the transform is taken to be

$$\tilde{x}(f) = \int_{-\infty}^{\infty} x(t) e^{2\pi i f t}\, dt \tag{2.3}$$

so that the inverse transform uses $e^{-2\pi i f t}$. This will give results quite different than the convention we use. A check on this, if you want to test a particular algorithm, is to give it the same signal with a time lag, and compare the phases: for our convention, a lag means a more negative phase. As we saw in the previous section, this convention comes from our choice of $e^{2\pi i f t}$ as a test function; using $e^{-2\pi i f t}$ would give the reverse Fourier convention. There are sometimes good reasons to use this other convention; for example, if we write a traveling wave as $e^{i(kx+2\pi f t)}$ we get something that propagates backwards, which is undesirable. The definition in (2.3) is therefore common when dealing with waves.

## 2.4 A Transform Pair

When we have a pair of functions related by (2.1) or (2.2), we often say that $x(t)$ is the function viewed in the time domain, and its transform $\tilde{x}(f)$ is the same function viewed in the frequency domain. If you have not encountered this phraseology before, talking about the two domains can sound rather mysterious; but in fact we do it all the time: for example, in music pitch is about frequency and rhythm about time. When we say "Doing A in the time domain has effect B in the frequency domain", you should remember that this is just another way of saying, "If we modify the function

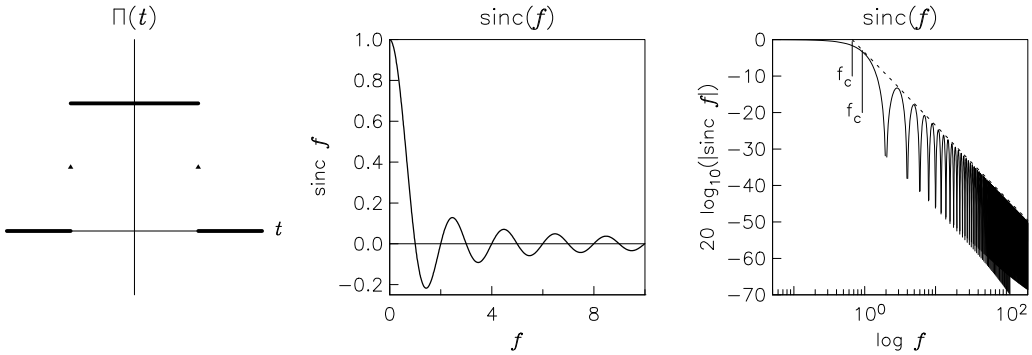



Figure 2.3: The left panel shows the Π function. The next two panels show the sinc function plotted on a linear scale (center) and on a log-log scale (right). (In the log-log plot, the zeroes in sinc should go off the bottom of the plot but do not because of the way the plot program works.) This plot also shows the "corner frequency" $f_c$, defined in two ways: one (left) as the intersection between the high-frequency and low-frequency asymptotes, and the other (right) as the frequency at which $\text{sinc}(f) = \frac{1}{2}$ (−3 dB).

in according to A, its Fourier transform will be modified according to B". And, as we illustrated in the first chapter, what is invisible in one domain is often obvious in the other.

To some extent a familiarity with transform pairs and their behavior comes only with experience; to start developing this we begin with a simple case and explore some of the behaviors that it illustrates, and then look at $\tilde{x}(f)$ for a few more functions.

Our first transform pair starts with the most basic of transients, the rectangle (really square) function (often called a boxcar, at least in American usage):

$$\Pi(t) = \begin{cases} 1 & |t| < \frac{1}{2} \\ \frac{1}{2} & \text{for} \quad |t| = \frac{1}{2} \\ 0 & |t| > \frac{1}{2} \end{cases} \tag{2.4}$$

This function is discontinuous; the value at $t = \frac{1}{2}$ is required in a fully rigorous definition. A more general form of Π is $a\Pi([t-b]/c)$, which is centered at $b$, and has amplitude $a$ and duration $c$.

The Fourier transform of $\Pi(t)$ is easy to find and gives another impor-

tant function:

$$\mathscr{F}[\Pi(t)] = \int_{-\infty}^{\infty} \Pi(t) e^{-2\pi i f t}\, dt = \int_{-1/2}^{1/2} e^{-2\pi i f t}\, dt = \int_{-1/2}^{1/2} \cos 2\pi f t\, dt$$
$$= \frac{sin\pi f}{\pi f} \stackrel{\text{def}}{=} \text{sinc}(f)$$

Figure 2.3 shows what the sinc function looks like. We can then write the inverse transform

$$\Pi(t) = \int_{-\infty}^{\infty} \text{sinc} f e^{2\pi i f t}\, df$$

but if we interchange $f$ and $t$, and substitute $-f$ for $f$, which is allowable because sinc is an even function, we get

$$\Pi(f) = \int_{-\infty}^{\infty} \text{sinc}(t) e^{-2\pi i f t}\, dt$$

another Fourier transform. We see that while sinc($t$) is nonzero over an infinite range, and very smooth, its distribution with frequency is $\Pi(f)$, which is nonzero only over a finite range ("bandlimited") and not smooth at all.

## 2.5 Fourier Theorems I: Similarity and Shift

Now, suppose we make the rectangle function broader or narrower, so it is $\Pi(at)$. The transform is (defining $t' = at$):

$$\int_{-1/2a}^{1/2a} e^{-2\pi i f t}\, dt = \frac{1}{a} \int_{-1/2}^{1/2} e^{-2\pi (f/a) t'}\, dt' = \frac{1}{a} \text{sinc}\left(\frac{f}{a}\right)$$

If $a$ becomes large, $\Pi(at)$ becomes narrower, but its Fourier transform sinc($f/a$) becomes wider. This duality of width between the two domains is in fact a general result, called the **similarity theorem**; again letting $t' = at$

$$\int_{-\infty}^{\infty} x(at) e^{-2\pi i f t}\, dt = \frac{1}{|a|} \int_{-\infty}^{\infty} x(t') e^{-2\pi i f t'/a} dt'$$

which means that if $\mathscr{F}[x(t)] = \tilde{x}(f)$,

$$\mathscr{F}[x(at)] = \frac{1}{|a|} \tilde{x}(f/a)$$

where the $|a|$ comes from the exchange of limits for $a < 0$. Shortening in the time domain thus implies broadening in the frequency domain; a generalization of the obvious case of a sine wave, for which less time between peaks means a higher frequency.

We can be more precise about this relationship between broadening and narrowing by noting that, by the definition of the transform and its inverse,

$$\tilde{x}(0) = \int_{-\infty}^{\infty} x(t)dt \qquad \text{and} \qquad x(0) = \int_{-\infty}^{\infty} \tilde{x}(f)\,df$$

which means that

$$\frac{\int_{-\infty}^{\infty} x(t)dt}{x(0)} = \left[\frac{\int_{-\infty}^{\infty} \tilde{x}(f)df}{\tilde{x}(0)}\right]^{-1}$$

Since $\int_{-\infty}^{\infty} x(t)\,dt$ is just the area under the function, $\frac{1}{x(0)}\int_{-\infty}^{\infty} x(t)\,dt$ is a kind of measure of the width of the function, specifically of the width of a rectangle of height $x(0)$ with the same area. This width, and the width of the transform, are thus reciprocals. Also note that we have now seen the first example of another duality: global aspects in one domain (the area) are connected to local aspects in the other (the value at zero).

Rather than changing the scale of the $t$ axis, and thus the "width" of a function we can ask what happens if we shift it in time. The Fourier transform of a function $x(t)$ that has been shifted in time by an amount $\tau$ is:

$$\mathscr{F}[x(t-\tau)] = \int_{-\infty}^{\infty} x(t-\tau)e^{-2\pi i f t}\,dt = \int_{-\infty}^{\infty} x(t')e^{-2\pi i f(t'+\tau)}\,dt' = \tilde{x}(f)e^{-2\pi i f\tau}$$

Looking at the amplitude and phase of $\tilde{x}(f)e^{-2\pi i f\tau}$, we see that the first, which is $|\tilde{x}(f)|$, is unchanged; but the phase has $-2\pi f\tau$ added to it: a time shift thus causes a phase that varies linearly with frequency. If (as is usual) we use the term **amplitude spectrum** to refer to the amplitude of the Fourier transform, we can say that a time shift leaves the amplitude spectrum unaltered, but changes the phase spectrum. For our sign convention for the Fourier transform, a time delay will cause the phase to be more negative at a given frequency. In any situation in which uncontrollable, or irrelevant, delays are present, the amplitude spectrum will be the part of the Fourier transform to look at. For example, this is the spectrum usually taken of seismic wave arrivals, unless the travel time is important.

## 2.6 Generalized Functions

So far we have proceeded as though, given a function, there would in fact be another function that is its Fourier transform. But this may easily not be true; we can easily come up with functions, for example $x(t) = \cos t$, or $x(t) = 1$, for which the transform integral (2.2) does not exist. We could limit ourselves to functions for which (2.2) does exist, but it turns out to be possible, to extend, without loss of rigor, the idea of what $x(t)$ can be, to include what are called **generalized functions**; once we introduce these we can assume that a much wider class of functions has Fourier transforms.

Adding to the kinds of entities we can consider, rather than declaring results to be meaningless, is often a fruitful approach. To take the simplest examples, if we start with the integers, we have to introduce rational numbers for division of any integer by any other to make sense. Similarly, for subtraction of any two to be possible requires that we introduce negative numbers; and allowing square roots requires irrational and imaginary numbers. As the names of the last two attest, some of these extensions have not been easily accepted.

A full and rigorous treatment of generalized functions goes well beyond the scope of this course. We give a brief, and very nonrigorous, version, though our strategy for developing generalized functions will be the same as the one used in a rigorous approach: we treat such functions as the limit of a sequence of ordinary functions.

As a start, consider the convolution of the rectangle function with another function $x(t)$. Evaluated for $t = 0$, this is

$$x(t) * \Pi(t)|_{t=0} = \int_{-\infty}^{\infty} x(u)\Pi(-u)du = \int_{-1/2}^{1/2} x(t)\,dt$$

Next, do the same thing for $a\Pi(at) * x(t)$; the $t = 0$ value is

$$a\int_{-1/2a}^{1/2a} x(t)\,dt$$

What happens as $a \to \infty$, making the rectangle function higher and narrower? Provided that $x(t)$ is itself well-behaved near zero, what we get is the area under a rectangle with height close to the average of $x(t)$, and with base $a^{-1}$; we then multiply this by $a$. The $a^{-1}$ and $a$ cancel, so that this product is the average of $x(t)$ over the range from $-\frac{1}{2}a$ to $\frac{1}{2}a$. As $a$ gets larger, this average of $x$ approaches $x(0)$, so that

$$\lim_{a\to\infty} [a\Pi(at) * x(t)]_{t=0} = x(0)$$

If we take the limiting operation "outside" the convolution, we may define a generalized function

$$\delta(t) = \lim_{a\to\infty} [a\Pi(at)]$$

such that

$$\int_{-\infty}^{\infty} x(u)\delta(u)\,du = x(0)$$

This is the **Dirac $\delta$-function**, which can be pictured as an infinitely narrow, infinitely high spike at $x = 0$. By using a shifted version of $\Pi(t)$ we can equally well show that

$$\int_{-\infty}^{\infty} x(u)\delta(u-t)\,du = x(t)$$

which means that convolution with $\delta(t-a)$ selects the value of the function at $t = a$. The $\delta$-function thus is the mathematical link between continuous time and sampled data.

Note that from the above we can conclude

$$\delta(t) = 0 \qquad \text{for} \qquad t \neq 0 \qquad \text{but} \qquad \int_{-\infty}^{\infty} \delta(t)\,dt = 1$$

making the $\delta$ function one that vanishes everywhere but zero, but still has a finite integral. Note that the value of $\delta$ at $t = 0$ is undefined: this function (like other generalized functions) only has meaningful properties when integrated, not when standing alone.

A function closely related to the $\delta$ function (but not itself a generalized function) is the **Heaviside step function**

$$H(t) = \begin{cases} 1 & t > 0 \\ \text{for} & \\ 0 & t < 0 \end{cases}$$

which is the mathematical way of saying, "Throw the switch."[2] If we perform the convolution $x * H$ we get

$$\int_{-\infty}^{\infty} x(u)H(t-u)du = \int_{-\infty}^{t} x(u)\,du$$

[2] Entirely appropriately, Oliver Heaviside, the inventor of this and much of the other mathematical machinery in this field, worked as a telegrapher.

which is just the integral of $x$ evaluated at $t$. Taking the derivative of this, by the theorems of calculus, gives $x(t)$ again. We may, very nonrigorously, write:

$$\frac{d}{dt}[H * x] = x$$

and applying the derivative operator to the step function (interchanging differentiation and integration) gives $H' * x = x$, which means that $H(t)' = \delta(t)$. That the derivative of $H$ is the $\delta$-function, while graphically appealing, does considerable violence to the usual notion of derivative, but again can be made rigorous by considering sequences of functions which approach $H(t)$; if these are differentiable, it can be shown that the derivatives approach $\delta(t)$. Indeed, it is possible to keep "taking derivatives" in this fashion to get a function $\delta'(t)$, which when convolved with a function $x(t)$ produces the derivative $x'(t)$.

## 2.7 Fourier Transforms of Generalized Functions

So, what additional Fourier transforms can we do now? If we take Fourier transforms of the sequence of rectangle functions, $\mathscr{F}[a\Pi(at)]$, the transform is

$$a\int_{-1/2a}^{+1/2a} e^{-2\pi i f t}\,dt = \int_{-1/2}^{1/2} e^{-2\pi(f/a)t'}\,dt' = \mathrm{sinc}(f/a)$$

If we now let $a$ go to $\infty$, $f/a \to 0$ and since $\mathrm{sinc}(0) = 1$, we have (interchanging integration and the limiting operation)

$$\mathscr{F}[\delta(t)] = 1$$

which says that the $\delta$-function contains all frequencies equally. Of course just taking the transform, and using the properties of the $\delta$-function, gives the same result:

$$\mathscr{F}[\delta(t)] = \int_{-\infty}^{\infty} \delta(t) e^{-2\pi i f t}\,dt = e^{-2\pi i 0} = 1$$

The inverse transform then implies that

$$\int_{-\infty}^{\infty} e^{2\pi i f t}\,df = \delta(t)$$

One way to "see" this is to note that for $t \neq 0$ the integrand oscillates and so gives a zero integral, but for $t = 0$ the integral is infinite. Again, in ordinary function theory this equation would be a complete nonsense; the left side is nonintegrable and the right side is meaningless. The extension to generalized functions allows us to deal with such problems in a consistent and rigorous way.

If we now swap $f$ and $t$, we get

$$\mathscr{F}[e^{2\pi i f_0 t}] = \int_{-\infty}^{\infty} e^{2\pi i t(f_0 - f)} dt = \delta(f_0 - f) = \delta(f - f_0)$$

with, as might be expected, an "infinite" peak at frequency $f_0$. A $\delta$ function away from zero is thus the Fourier transform of a complex sinusoid.

This is the transform of a complex sinusoids; what about sines and cosines? Their transforms are:

$$\mathscr{F}[\cos 2\pi f_0 t] = \mathscr{F}[\tfrac{1}{2}(e^{-2\pi i f_0 t} + e^{-2\pi i f_0 t})] = \frac{1}{2}[\delta(f - f_0) + \delta(f + f_0)]$$

which is two $\delta$-functions at $\pm f_0$. Similarly,

$$\mathscr{F}[\sin 2\pi f_0 t] = \mathscr{F}[\frac{-i}{2}(e^{2\pi i f_0 t} - e^{-2\pi i f_0 t})] = \frac{-i}{2}[\delta(f - f_0) - \delta(f + f_0)]$$

which is purely imaginary, and also has $\delta$-functions at $+-f_0$, though they are oppositely directed.

These two results cases illustrate some important **symmetry relations** for the Fourier transform: $\cos 2\pi f_0 t$ is real and even, and so is its transform; $\sin 2\pi f_0 t$ is odd, and its transform is also, but is purely imaginary. Since any real function can be split into even and odd parts, its transform must have a real part which is even, and an odd part which is imaginary: for $x(t)$ real,

$$\mathscr{R}[\tilde{x}(f)] = \mathscr{R}[\tilde{x}(-f)] \qquad \mathscr{I}[\tilde{x}(f)] = -\mathscr{I}[\tilde{x}(-f)]$$

which is to say $\tilde{x}(f) = \tilde{x}^*(-f)$. Functions with this property are known as **Hermitian**. Because real functions have Hermitian transforms, the transform only needs to be specified for $f \geq 0$.[3] Table 2.7 summarizes most of the useful symmetry relations between functions and their transforms.

Since we have just established that negative frequencies are superfluous when we are dealing with real-valued functions, and knowing that all

[3] Or $f \leq 0$, but nobody does that.

| Time Domain | Frequency Domain | |
|---|---|---|
| Complex | Complex | |
| Imaginary | anti-Hermitian | $\tilde{x}(f) = -\tilde{x}^*(-f)$ |
| Real | Hermitian | $\tilde{x}(f) = -\tilde{x}^*(-f)$ |
| Real, even | Real, even | $\tilde{x}(f) = \tilde{x}(-f)$ |
| Real, odd | Imaginary, odd | $\tilde{x}(f) = -\tilde{x}(-f)$ |

data are real-valued, you may wonder if there is any meaning, outside the formalism of the mathematics, for negative frequencies. The answer is that they are useful in certain cases, namely, in dealing with 2-dimensional vector processes: for example, horizontal currents in the ocean. We can represent such data as complex numbers using the Argand-diagram representation: given a North velocity $v_N$ and East velocity $v_E$, we can write the vector velocity as $v = v_E + iv_N$, where $v$ is complex-valued. The transform of data $v$ then has no symmetry about $f = 0$: we need to look at it for both positive and negative frequencies. What the sign of the frequency tells us, is whether the vector *rotates* in one sense or the other (clockwise vs anticlockwise). For systems in which gyroscopic forces are important, this can be a useful way of looking at the data. It would be very useful if there were something like this for vectors in three dimensions; unfortunately there isn't.

## 2.8 Fourier Theorems II: Derivatives and Differentiability

We now return to general theorems on Fourier transform pairs, concerning ourselves now with derivatives. A (nonrigorous) application of integration by parts gives

$$\mathscr{F}[x'(t)] = \int_{-\infty}^{\infty} x'(t) e^{-2\pi i f t} dt = \frac{e^{-2\pi i f t} x(t)}{2\pi i f}\Big|_{-\infty}^{\infty} + 2\pi i f \int_{-\infty}^{\infty} x(t) e^{-2\pi i f t} dt \tag{2.5}$$

so that $\mathscr{F}[x'(t)] = 2\pi i f \tilde{x}(f)$, which also holds in reverse, with, $\mathscr{F}^{-1}[\tilde{x}'(f)] = -2\pi i t x(t)$.

These results are the basis for using Fourier transforms to solve some classes of linear differential equations, something we discuss later on in

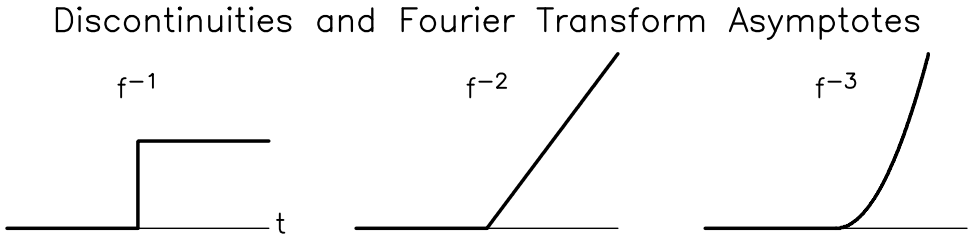



Figure 2.4: Sketches to show what level of discontinuity (left to right: in function, first derivative, and second derivative) provides what level of asymptotic decay of the Fourier transform.

designing some kinds of digital filters. In the terms we are developing, of looking at the function in two domains, what this result is important for is showing what taking a derivative or integral in the time domain does to the function in the frequency domain. For example, consider ground motion from an earthquake: for given displacements, the velocities (the first derivative) will have a transform scaled by $f$, and will thus be much rougher (richer in high frequencies); and the accelerations will be rougher still, with a Fourier transform scaled by $f^2$.

This result leads to other ones, which like the earlier results on widths connect local properties in one domain with global ones in another. For second derivatives

$$\mathscr{F}^{-1}[\tilde{x}''(f)] = -4\pi^2 t^2 x(t)$$

Taking the transform of both sides gives

$$\int_{-\infty}^{\infty} t^2 x(t) e^{-2\pi i f_o t} dt = \frac{-\tilde{x}''(f)}{4\pi^2}$$

which means that the second moment of the function $x(t)$ is

$$\int_{-\infty}^{\infty} t^2 x(t) dt = \frac{-\tilde{x}''(0)}{4\pi^2}$$

that is, the second moment of the function is proportional to the second derivative of its Fourier transform, evaluated at the origin. Once again we see that a global property (the second moment) in the time domain is connected to a very local property (the second derivative at the origin) in the frequency domain.

Still another example of the local/global relation between the original and the transform is a theorem which relates the differentiability of $x(t)$ to

the asymptotic behavior of $\tilde{x}(f)$ at large $f$; that is, to the bounds on $|\tilde{x}(f)|$ as $f \to \infty$. Intuitively one might suppose these to be connected; the "rougher" a function is, the more high frequencies we would expect it to contain. We can motivate (not prove!) the result we seek by supposing $x(t)$ to contain steps; then $x'(t)$ (using our extended notion of differentiation) contains $\delta$-functions, and so $|\mathscr{F}[x'(t)]| \propto 1$ for $f$ large. But then we have, from equation (2.5):

$$|\mathscr{F}[x(t)]| = |\tilde{x}(f)| = \left|\mathscr{F}[x'(t)]2\pi i f\right| \propto f^{-1}$$

for $f \to \infty$. Similarly, a function with steps is the derivative of one that is continuous but has corners (discontinuous derivatives) and so by the same argument the Fourier transform of such a function will be proportional to $f^{-2}$ as $f \to \infty$. Extended, this shows that if $x(t)$ has a discontinuous $n$-th derivative, $|\tilde{x}(f)|$ will fall off as $f^{-(n+1)}$ at high frequencies; Figure 2.4 gives a graphical summary of the first few cases.

Two extreme examples are $\Pi(t)$, for which the zeroth derivative is discontinuous; and indeed

$$|\mathscr{F}[\Pi(t)]| = |\mathrm{sinc} f| < k/f \qquad \text{as} \quad f \to \infty$$

(here $k = 1/\pi$, but it is the rate of the falloff that matters). At the other extreme, suppose $x(t) = e^{-\pi t^2}$, which is infinitely differentiable. For this,

$$\begin{aligned}\tilde{x}(f) = \mathscr{F}[x(t)] &= \int_{-\infty}^{\infty} e^{-(\pi t^2 + 2\pi i f t)}\, de \\ &= e^{-\pi f^2} \int_{-\infty}^{\infty} e^{-\pi(t+if)^2}\, dt \\ &= e^{-\pi f^2} \int_{-\infty}^{\infty} e^{-\pi u^2}\, du = e^{-\pi f^2}\end{aligned}$$

which for large $f$ falls off (goes to 0) more rapidly than any power of $f$. We also see that the Gaussian is its own Fourier transform.

## 2.9 Fourier Theorems III: Convolution, Power, Modulation

So far, we have looked only at properties of a single function and its Fourier transform. We now consider combinations of functions. Linearity means that if we add two functions the transform of the result is the sum of the two transforms. A much more interesting result appears if we consider the

function formed by convolution of two functions. The result is the **convolution theorem**, which states that

$$\mathscr{F}[x(t) * y(t)] = \mathscr{F}[x(t)]\mathscr{F}[y(t)]$$

which is to say, the transform of the function formed by convolving two functions is the product of the transform of the individual functions. This is not difficult to show; again, using change of variables freely, we have:

$$\begin{aligned}\int_{-\infty}^{\infty} e^{-2\pi i f t} \int_{-\infty}^{\infty} x(u)y(t-u)\,du\,dt &= \int_{-\infty}^{\infty} x(u) \int_{-\infty}^{\infty} y(t-u)e^{-2\pi i f t}\,dt\,du \\ &= \int_{-\infty}^{\infty} x(u) \int_{-\infty}^{\infty} y(t')e^{-2\pi i f t'} e^{-2\pi i f u}\,dt'\,du \\ &= \mathscr{F}[y(t)] \int_{-\infty}^{\infty} x(u)e^{-2\pi i f u}\,du = \mathscr{F}[y(t)]\mathscr{F}[x(t)]\end{aligned}$$

Writing $\mathscr{F}[x] = \tilde{x}$, $\mathscr{F}[y] = \tilde{y}$, we can state the related results

$$x * y = \mathscr{F}^{-1}[\tilde{x}\tilde{y}] \qquad \mathscr{F}^{-1}[\tilde{x} * \tilde{y}] = xy \qquad \mathscr{F}[xy] = \tilde{x} * \tilde{y}$$

This theorem is *the* main reason for our use of Fourier transforms: while the action of a linear system is a convolution in the time domain, it is a multiplication in the frequency domain – and multiplication is much easier to grasp than convolution. While the frequency domain may initially seem less natural than the time domain, operations in it are often much simpler. Our earlier result, that a sinusoid $e^{2\pi i f t}$ put into a linear time-invariant system yields another sinusoid $\tilde{g}(f)e^{2\pi i f t}$ out, now appears as a special case of the convolution theorem. We write the effect of the system as $y = g * x$, which implies $\tilde{y}(f) = \tilde{g}(f)\tilde{x}(f)$, and if $\tilde{x}(f) = \delta(f - f_0)$, $\tilde{y}(f)$ will be

$$\tilde{g}(f)\delta(f - f_0) = \tilde{g}(f_0)\delta(f - f_0)$$

since $\delta(f - f_0) = 0$ for $f \neq f_0$. Again, and now perhaps more clearly than in our earlier discussion, we see that $\tilde{g}(f)$ is the frequency response of the system.

If we make the input to the system $x(t) = \delta(t)$, then the output is

$$y(t) = \int_{-\infty}^{\infty} \delta(t-u)g(u)du = g(t)$$

which is called the **impulse response** of the system; albeit this is not easy to produce in practice, it is still a useful characterization. Then the frequency response is seen to be $\tilde{g}(f) = \mathscr{F}[g(t)]$

A function related to the impulse response is its integral, the **step response**:

$$h(t) = \int_{-\infty}^{t} g(u)\,du$$

which is the response of the system to a Heaviside step function $H(t)$ applied at $t = 0$. For any actual system this is much easier to produce than the impulse response is.

As an example of a system's frequency response, consider differentiation, which is a linear time-invariant system – though not usually thought of as such. We have seen that $\mathcal{F}[x'(t)] = 2\pi i f \tilde{x}(f)$ and so the frequency response for this system is $\tilde{g}(f) = 2\pi i f$, with amplitude and phase spectra

$$|\tilde{g}(f)| = 2\pi f \qquad \text{Ph}[\tilde{g}(f)] = \pi/4 = 90^\circ$$

As might be expected, the response rises with increasing frequency; the 90° phase response means that sines are turned into cosines, and vice versa.

The frequency response of a system is often much simpler to describe than its impulse response (and more indicative of the underlying physics); we will (eventually) get to ways of estimating it given $x(t)$ and $y(t)$.

An interesting consequence of the convolution theorem relates to the total variance, also called energy (or, sometimes, power) in $x(t)$, which is

$$\int_{-\infty}^{\infty} |x(t)|^2\,dt$$

We first consider the Fourier transform of the conjugate of $x$, which is

$$\mathcal{F}[x^*(t)] = \int_{-\infty}^{\infty} x^*(t) e^{-2\pi i f t}\,dt = \left[\int_{-\infty}^{\infty} x(t) e^{-2\pi i(-f)t}\,dt\right]^* = \tilde{x}^*(-f)$$

and so, using one version of the convolution theorem,

$$\mathcal{F}[x(t)|^2] = \mathcal{F}[xx^*] = \tilde{x}(f) * \tilde{x}^*(-f)$$

Remembering that the area under a function is the value of its Fourier transform at $f = 0$, this means that

$$\int_{-\infty}^{\infty} |x(t)|^2\,dt = \tilde{x}(f) * \tilde{x}^*(-f)\Big|_{f=0} = \left[\int_{-\infty}^{\infty} \tilde{x}(u)\tilde{x}^*(u-f)\,du\right]_{f=0} = \int_{-\infty}^{\infty} |\tilde{x}(f)|^2\,df$$

so that the total energy is the same in the function as in its transform. This result goes by several names; Bracewell (1986) calls it **Rayleigh's theorem**, but says that the name Plancherel's theorem is also in use; this result

is also sometimes called **Parseval's theorem**, especially for the analogous case of discrete-time Fourier transforms, which we will meet later on.

As another example of the convolution theorem, we derive the transform of a modulated waveform, in which the data looks like a "slowly varying" sinusoid, where "slow" means "over many cycles of the sinusoid". The slow variation, termed a modulation, can be in amplitude, frequency, or phase; all of these are used in different forms of radiowave communication.

Suppose the sinusoid ("carrier" in radio parlance) is the cosine $\cos 2\pi f_0 t$; this has the Fourier transform

$$\mathscr{F}[\cos 2\pi f_0 t] = \tfrac{1}{2}[\delta(f - f_0) + \delta(f + f_0)]$$

Suppose also that the modulation is purely in amplitude, and the modulating function is $x(t)$. Then the modulated waveform is $x(t)\cos 2\pi f_0 t$. We can now apply the inverse of the convolution theorem: the transform of a product of two functions is the convolution of the individual transforms. Remembering that convolution with a $\delta$-function recovers the function being convolved, this means that

$$\begin{aligned}\mathscr{F}[x(t)\cos 2\pi f_0 t] &= \tfrac{1}{2}[\delta(f - f_0) * \tilde{x}(f) + \delta(f + f_0) * \tilde{x}(f)] \\ &= \tfrac{1}{2}[\tilde{x}(f - f_0) + \tilde{x}(f + f_0)]\end{aligned}$$

where $\tilde{x}(f) = \mathscr{F}[x(t)]$.

The effect of multiplying a function by a sine wave is thus to replicate the original transform at $\pm f_0$ (Figure 2.5); alternatively, we can say that the effect of modulating a sine wave is to spread its original $\delta$-function transform out over a broader band, though if $x(t)$ contains only frequencies much less than $f_0$ (as is true in the case of radio), the band will be narrow. This is also true for other classes of modulation, though the relation between $\tilde{x}(f)$ and the transform of the modulated function is less straightforward.

## 2.10 The Correlation Function

We can modify the convolution integral by not time-reversing one function. The result is called the **cross-correlation**, because that is the form taken by the correlation between two random time series, as we will see later on. For now we look at some of the properties of this expression when we use the same function for both parts; this is called the **autocorrelation**. For a complex-valued $x$, this is:

$$C(\tau) = \int_{-\infty}^{\infty} x^*(t)x(t+\tau)\,dt$$

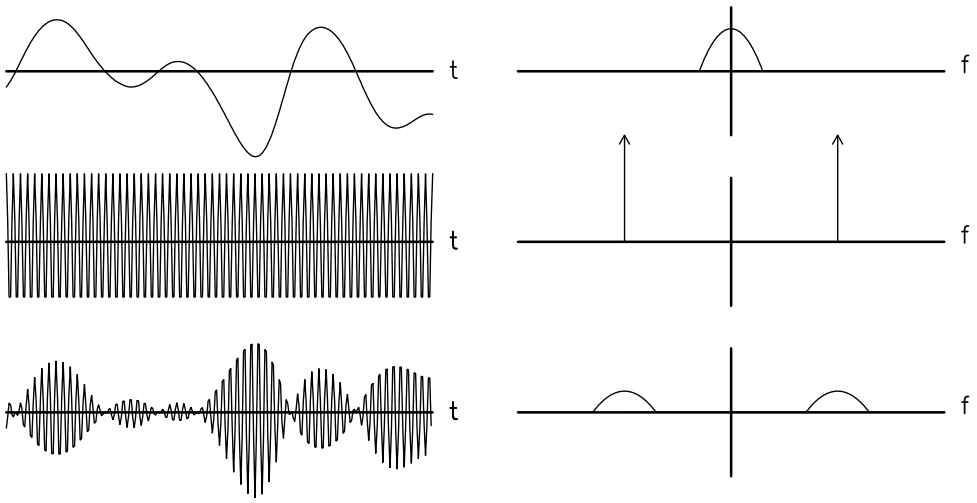



Figure 2.5: From top to bottom, the time series and spectra for a signal, a carrier waves, and the amplitude-modulated carrier. The spectra are schematic.

The location of the conjugate part matters; if we conjugate the other $x$ in the integral, we have, since $yz^* = (y^* z)^*$ in general, that

$$\int_{-\infty}^{\infty} x(t)x^*(t+\tau)dt = \left(\int_{-\infty}^{\infty} x^*(t)x(t+\tau)\,dt\right)^* = C^*(\tau)$$

which is not the same as $C(\tau)$. $C$ has some symmetry properties; if we change variables, we get the result:

$$C(-\tau) = \int_{-\infty}^{\infty} x^*(t)x(t-\tau)dt = \int_{-\infty}^{\infty} x^*(u+\tau)x(u)\,du$$

which from the previous result is $C^*(\tau)$. Hence, $C(-\tau) = C^*(\tau)$, which is to say that $C$ is Hermitian; this implies that its Fourier transform is always real. If $x(t)$ is real, $C$ is also; which means that $C$, and its transform, must be even. All of these properties will become relevant when we define the power spectral density, since the most general form of this is the Fourier transform of a Hermitian function, and the most usual form is the Fourier transform of a real symmetric function.

CHAPTER 3

# FOURIER THEORY FOR DISCRETE TIME

## 3.1 Introduction

We now turn from Fourier theory for functions to the same theory for ordered sets of numbers: this is part of **digital signal processing**. As we will see, much of what has been covered in the Fourier theory discussion will have parallels here – though also significant differences, which come about because the theory of the Fourier transform assumes that what is being transformed (and the transforms themselves) are functions on the real line (or a higher-dimensional equivalent). If we specifically take these functions as functions of time, we say that they are defined "in continuous time". However, digital signals are, intrinsically, not so defined: rather, they are collections of numbers, representing (usually) a continuous time signal sampled at regular intervals. Such functions are called **sampled data**[1] and are be said to be defined in **discrete time**.

We begin by describing how Fourier theory works applied to such data, for two cases:

- Discrete-time data defined, like that in continuous time, over an infinite range, so the data "go on forever." The Fourier transform of such data turns out to be a function on the real line; but unlike the Fourier transform of a function, which is another function, the transform of an infinite amount of discrete data is a function defined on only a part of the real time. A discrete-time series and its transform are very different.

[1] This terminology is more commonly (in statistics) used for the act of getting data on a subset of some population, ideally without biasing the results of an statistical investigation. We will not deal with this topic.

- Discrete-time data defined over a finite range – which is what we actually have to deal with. The Fourier transform of this turns out to be a finite set of numbers at discrete frequencies: in this case, as with functions on the real line, a function and its transform are the same kind of thing.

The Fourier transform of a finite amount of discrete-time data is called (what else?) the **Discrete Fourier Transform** or **DFT**. We will spend some time in this chapter exploring the properties of the DFT, including an overview of how to compute it efficiently, though we relegate the details of this to Appendix A. In Chapter 4 we will use the theory to discuss how to go from continuous time to discrete time without losing information: a question best approached, as it turns out, by looking at Fourier transforms. The following chapters describe the commonest operations we perform on discrete-time data: filtering them to remove certain frequencies (Chapter 5), to emulate how systems behave in continuous time (Chapter 6), or for other reasons (Chapter 7).

## 3.2 Discrete-Time Sequences and Operations

In discrete time there are no functions, only sequences of numbers; we denote such a sequence by $x_n$ or $x(n)$ or $x_n$, where in all cases $n$ is an integer-valued index. To start with we assume that we have an infinite amount of data, with $n$ running from $-\infty$ to $\infty$.

We seek as many parallels with the continuous-time case as we can. Some things are much simpler in discrete time. For example, the distinction between generalized functions and other functions disappears: all are just sequences. The $\delta$-function is just another the sequence:

$$\delta_n = \delta_{n0} = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$

where $\delta_{ij}$ is the Kronecker $\delta$-function. The discrete-time version of the Heaviside step function is

$$\delta_n = \delta_{n0} = \begin{cases} 1 & n > 0 \\ 0 & n \leq 0 \end{cases}$$

so that

$$H_n = \sum_{k=-\infty}^{n} \delta_k$$

In discrete time summation replaces integration, and we face none of the difficulties in making things rigorous that we had to consider when dealing with functions on the real line.

Likewise, convolution is a summation rather than an integral; the convolution of two sequences (also called the **serial product**) is still written as $z = x * y$, but this now denotes

$$z_n = \sum_{k=-\infty}^{\infty} x_k y_{n-k}$$

(again one series is reversed). This definition satisfies the requirements for convolution of being linear in $x$ and $y$, and also invariant with shifts in time (sequence number); if we shift $x$ by $m$ terms, the convolution $x * y$ is

$$\sum_{k=-\infty}^{\infty} x_{k-m} y_{n-k} = \sum_{l=-\infty}^{\infty} x_l y_{n-m-l} = z_{n-m} \tag{3.1}$$

so $z$ is also shifted.

Convolution is just as important in discrete time as in continuous time; indeed, some types of sequences are defined by what they do in convolution. One example is that a sequence $\{g_k\}$ is **causal** if and only if $g_k = 0$ for $k < 0$; this means that if $\{x_k\}$ is causal, $g * x$ will be also: the output will not precede the input. If you are processing data in real time, or trying to emulate a system that is causal (such as an instrument, or wave propagation) this is an important distinction; more generally, however, it is not: usually we analyze data long after they are collected, so that for most on a computer the actual time is usually far behind real time.

## 3.3 Fourier Transforms for Infinite Sequences

We started our discussion of Fourier theory by looking at what linear systems do to sinusoids; similarly, we develop Fourier transforms for infinite sequences by considering the convolution of a sinusoidal sequence with the sequence of interest. In continuous time we saw that a sinusoid input into a linear system produced an output sinusoid, scaled by a function (the frequency response) that is the Fourier transform of the convolving function.

The same thing holds in discrete time; if we convolve $x_n = e^{2\pi i f n}$ with $g_n$, we get, starting by commuting the series,

$$\sum_{k=-\infty}^{\infty} g_k x_{n-k} = \sum_{k=-\infty}^{\infty} g_k e^{2\pi i f(n-k)} = e^{2\pi i f n} \sum_{k=-\infty}^{\infty} g_k e^{-2\pi i f k} \stackrel{\text{def}}{=} \tilde{g}(f) e^{2\pi i f n}$$

$\tilde{g}(f)$ is again the frequency response, and regarding it as the Fourier transform of the sequence $g_k$ gives the definition for the Fourier transform of an infinite sequence $x_n$

$$\tilde{x}(f) = \sum_{k=-\infty}^{\infty} x_k e^{-2\pi i f k} \tag{3.2}$$

Another way to get this definition from the continuous-time Fourier transform is to introduce the **sampled-data function** $x_s(t)$, defined by

$$x_s(t) = \sum_{k=-\infty}^{\infty} \delta(t-k) x(t)$$

While this is a continuous-time function (actually a generalized function), it is like a sequence in that it contains information about the value of $x(t)$ only at the sample points. Just as a single $\delta$ function samples $x$ at a single time, an infinite array of them samples $x$ at the times of the sequence $x_n$. We will explore the effects of such sampling in the next chapter; for now we simply note that the Fourier transform of the sampled-data function is

$$\mathcal{F}[x_s(t)] = \int_{-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \delta(t-k) x(t) e^{-2\pi i f t} dt = \sum_{k=-\infty}^{\infty} x(k) e^{-2\pi i f k}$$

which is equivalent to how we defined the Fourier transform for a sequence. Notice that, as is conventional in digital signal processing, we are setting the sample interval to unity; for a sequence indexed by integer values there is no clue to this interval, and one is a convenient choice.

$\tilde{x}(f)$ is a function of $f$, that is, a function on the real line; but it is a function that is periodic, and with period one:

$$\tilde{x}(f+m) = \sum_{k=-\infty}^{\infty} x_k e^{-2\pi i f k} e^{-2\pi i m k} = \tilde{x}(f) \quad \text{for m an integer}$$

It is often more useful, and certainly sufficient, to regard $\tilde{x}(f)$ as defined only over $[-\frac{1}{2}, \frac{1}{2}]$. (This is the interval for $f$; for $\omega = 2\pi f$, the interval would be $[-\pi, \pi]$.) When we discuss sampling, we will see that these limits

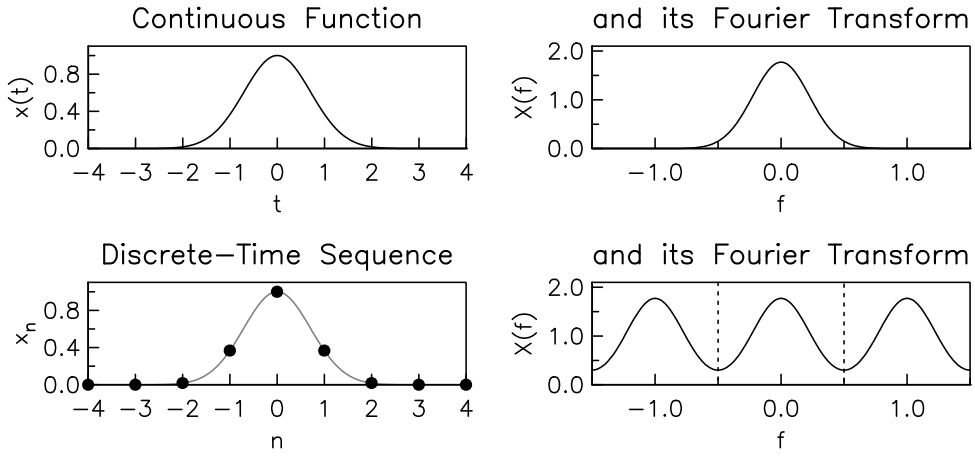



Figure 3.1: An example of the relationships between a function, a discrete-time sequence related to it, and their Fourier transforms. The function (top left) is a Gaussian, $e^{-x^2}$; its Fourier transform is shown top right. The bottom left panel shows part of a discrete-time sequence that matches this (with the original function in gray). The bottom right shows the Fourier transform of the sequence; the region of $f$ between the two dashed lines is sufficient to describe it. Note that the transform of the sequence is close to the Fourier transform of the function, except near to these two lines – the effect of aliasing, which we will discuss in Chapter 4.

are what is called the Nyquist frequency for a sample interval ($\Delta$) of one. The result for $\mathcal{F}[x_s(t)]$ can be generalized to $\Delta \neq 1$ by replacing $k$ by $k\Delta$, in which case the Fourier transform becomes

$$\tilde{x}(f) = \sum_{k=-\infty}^{\infty} x_k e^{-2\pi i f k \Delta}$$

which is periodic with period $1/\Delta$, and so defined over $-1/(2\Delta)$ to $1/(2\Delta)$; again, this will turn out to be the range between Nyquist frequencies. This result also shows that formulas for arbitrary $\Delta$ can be gotten from those for $\Delta = 1$ by replacing $f$ with $f\Delta$.

If we swap time and frequency in 3.2 we get an expression for a function of time as a sum of an infinite set of sinusoids in time:

$$x(t) = \sum_{k=-\infty}^{\infty} \tilde{x}_k e^{2\pi i t k}$$

which is just the **Fourier series** expansion of a periodic function of time in terms of sinusoids: this is the result that Fourier first obtained, and the

way that Fourier theory is often introduced. Of course, we need a way to find the coefficients for the sinusoids; these are given by Fourier's result that

$$\tilde{x}_n = \int_{-1/2}^{1/2} x(t) e^{-2\pi i t k}\, dt$$

So, swapping $t$ and $f$ again, we get the inverse transform for the Fourier transform $\tilde{x}(f)$ of a sequence $x_n$; the $x_k$'s can be regarded as the Fourier series coefficients, which are given by

$$x_n = \int_{-1/2}^{1/2} \tilde{x}(f) e^{2\pi i f n}\, df$$

So we have a transform pair – albeit not a symmetric one, since the forward transform is an infinite sum and the inverse transform a definite integral.

Many of the theorems already outlined for Fourier transforms carry over to this new transform pair, although others (e.g., the derivative theorems) do not. Some examples of ones that do involve integrals that become sums. Using $\tilde{x}(f)$ for the Fourier transform of both the function $x(t)$ and the sequence $x_n$ (remember that these are different $\tilde{x}$'s), we have for continuous time:

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

and in discrete time, from 3.2

$$\sum_{n=-\infty}^{\infty} x_n = \tilde{x}(0)$$

For Rayleigh's theorem we take complex conjugates of the inverse Fourier transform relation,

$$x_n^* = \int_{-1/2}^{1/2} \tilde{x}^*(f) e^{-2\pi i f n}\, df$$

which means that

$$\begin{aligned}\sum_{n=-\infty}^{\infty} |x_n|^2 &= \sum_{n=-\infty}^{\infty} x_n x_n^* = \sum_{n=-\infty}^{\infty} x_n \int_{-1/2}^{1/2} \tilde{x}^*(f) e^{-2\pi i f n}\, df \\ &= \int_{-1/2}^{1/2} \tilde{x}^*(f) \sum_{n=-\infty}^{\infty} x_n e^{-2\pi i f n}\, df = \int_{-1/2}^{1/2} \tilde{x}^*(f)\tilde{x}(f)\, df = \int_{-1/2}^{1/2} |\tilde{x}(f)|^2\, df\end{aligned}$$

The convolution theorem becomes one of sums rather than integrals: if we have a sequence $z = x * y$ then

$$\begin{aligned}\tilde{z}(f) &= \sum_{k=-\infty}^{\infty} z_k e^{-2\pi i f k} = \sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} x_l y_{k-l} e^{-2\pi i f k} \\ &= \sum_{l=-\infty}^{\infty} x_l \sum_{k=-\infty}^{\infty} y_{k-l} e^{-2\pi i f k} = \sum_{l=-\infty}^{\infty} x_l e^{-2\pi i f l} \sum_{k=-\infty}^{\infty} y_{k-l} e^{-2\pi i f (k-l)} \\ &= \sum_{l=-\infty}^{\infty} x_l e^{-2\pi i f l} \sum_{m=-\infty}^{\infty} y_m e^{-2\pi i f m} = \tilde{x}(f)\tilde{y}(f)\end{aligned}$$

## 3.4 Discrete Fourier Transform

Now we suppose that our sequence $x_n$ is, not infinite, but finite in length, with terms $x_0, x_1 \ldots x_{N-1}$: $N$ terms in all. How do we define the Fourier transform now?

We start in what is a somewhat unusual way, or at least one that is different from most treatments in signal processing: we consider the general problem of fitting sine waves to the data. We can always represent the data as a sum of sine waves plus a residual:

$$x_n = \sum_{l=0}^{L-1} C_l e^{2\pi i n f_l} + \epsilon_n \quad n = 0, \ldots N-1$$

where the frequencies $f_0, f_1 \ldots$ are assumed to be specified. If we choose the coefficients $C_l$ to minimize the sum of squares of the residuals, $\sum_{n=0}^{N-1} |\epsilon_n|^2$, we do so using least-squares theory. To apply this we write the sequences as vectors:

$$x^T = (x_0, x_1 \ldots x_{N-1}) \qquad C^T = (C_0, C_1 \ldots C_{L-1})$$

and the array of exponentials (in $l$ and $n$) as a matrix $A$, which is most easily written in terms of its adjoint (the complex conjugate of the transpose) $A^\dagger$:

$$A^\dagger = \begin{pmatrix} 1 & e^{-2\pi i f_1} & e^{-4\pi i f_1} & \ldots & e^{-2\pi i (N-1) f_1} \\ 1 & e^{-2\pi i f_2} & e^{-4\pi i f_2} & \ldots & e^{-2\pi i (N-1) f_2} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & e^{-2\pi i f_{L-1}} & e^{-4\pi i f_{L-1}} & \ldots & e^{-2\pi i (N-1) f_{L-1}} \end{pmatrix}$$

The coefficients $C$ that minimize the sum of squares of the residual is then given by the normal equations

$$(A^\dagger A) C = A^\dagger x \qquad \text{or} \qquad BC = A^\dagger x$$

where $B$ is an $L \times L$ matrix with elements

$$B_{pq} = \sum_{n=0}^{N-1} e^{-2\pi i n f_p} (e^{-2\pi i n f_q})^* = \sum_{n=0}^{N-1} e^{-2\pi i n (f_p - f_q)} \tag{3.3}$$

The solution for $C$ is then

$$C = B^{-1}(A^\dagger x)$$

In a general least-squares problem solving this will involve inverting an $L \times L$ matrix: a good-sized computational task if $L$ is large.

However, for the special form of $B$ given in equation (3.3), and with proper choice of the frequencies, we can simplify the form of $B$ substantially. First, we can compute the elements of $B$ analytically rather than by actually computing the sum 3.3, since there is a closed-form expression for finite sums of the exponential function $e^{-2i\pi fn}$. To get this expression, we first note that for any $z$, we have

$$\sum_{n=0}^{L-1} z^n = (1 - z^L) \sum_{n=0}^{\infty} z^n$$

as can be verified through term-by-term expansion; then setting $L = 1$ we get

$$\frac{1}{(1-z)} = \sum_{n=0}^{\infty} z^n$$

Combining these two results gives us a closed-form expression for a finite sum; then if we set $z = e^{-2i\pi f}$, we get

$$\sum_{n=0}^{N-1} e^{-2i\pi fn} = \frac{1 - e^{-2i\pi fN}}{1 - e^{-2i\pi f}} = \frac{e^{-i\pi fN}}{e^{-i\pi f}} \left( \frac{e^{i\pi fN} - e^{-i\pi fN}}{e^{i\pi f} - e^{-i\pi f}} \right) = e^{-i\pi f(N-1)} \frac{sin\pi Nf}{sin\pi f} \tag{3.4}$$

which gives, for the elements of $B$,

$$B_{pq} = e^{-i\pi(N-1)(f_p - f_q)} \frac{sin\pi N(f_p - f_q)}{sin\pi(f_p - f_q)}$$

Next, we choose a particular set of frequencies:

$$f_l = \frac{l}{N} \quad l = 0, 1, \ldots N-1 \tag{3.5}$$

so that there are as many frequencies as there are data values. Given this choice of frequencies, the matrix elements become:

$$B_{pq} = e^{-i\pi(N-1)(p-q)/N} \frac{sin\pi(p-q)}{sin(\pi(p-q)/N)} = \begin{cases} N & p = q \\ 0 & p \neq q \end{cases} = N\delta_{pq} \tag{3.6}$$

where we get the result for $p = q$ by treating $p$ and $q$ as real variables and taking the limit as $p \to q$. Note that this relationship, as given, is only true if the integers $p$ and $q$ are in the range from 0 through $N-1$; but this is true for the matrix elements. A more general expression is

$$\sum_{n=0}^{N-1} e^{-2\pi iln/N} = e^{-i\pi l(N-1)/N} \frac{sin\pi l}{sin(\pi l/N)} = \begin{cases} N & l = 0, \pm N, \pm 2N \ldots \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

which we will use a little bit later.

Using 3.6, we find that $B = NI$, where $I$ is the identity matrix. This could not be easier to invert; the inverse $M^{-1}$ is just $N^{-1}I$, which means that the coefficients for our fit are

$$C = \frac{1}{N} A^{\dagger} x$$

which we can write out explicitly as

$$C_l = \frac{1}{N} \sum_{n=0}^{N-1} x_n e^{-2\pi in(l/N)}$$

Getting the matrix $B$ to be the identity matrix has required two conditions and a mathematical result. The first condition is that an increment in sequence number is a constant increment in the fitting function; this is a consequence of equal spacing in time. The second condition is that we choose our frequencies according to equation (3.5). Given these specifications, the result we use is that the imaginary exponentials are orthogonal in summation over the finite interval. A major reason for assuming equispaced samples is that without this, we lose orthogonality in the fitting functions.

We define the **Discrete Fourier Transform (DFT)** as our fit, though with a different normalization:

$$\tilde{x}_k = \sum_{n=0}^{N-1} x_n e^{-2\pi ink/N} \quad k = 0, \ldots . N-1 \tag{3.8}$$

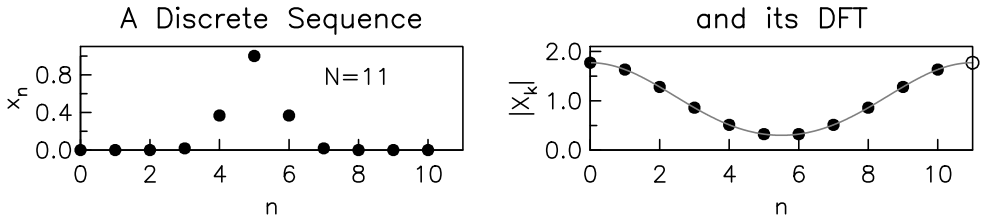



Figure 3.2: An example of a DFT pair, using a finite length of the same discrete-time sequence shown in Figure 3.1. The right side shows the DFT values (filled circles), with the gray line being the Fourier Transform of the infinite sequence; in this case these look the same because the infinite sequence is very small outside the range shown. The unfilled circle is an "extra" DFT point to show the periodicity of the DFT.

which, if we compare it with the definition of the Fourier transform of an infinite sequence (3.2), shows that

$$\tilde{x}_k = \tilde{x}(f) \qquad \text{for} \qquad f = \frac{k}{N} \quad \text{with} \quad k = 0, 1, \ldots N-1$$

The DFT coefficients can thus be considered as samples from the continuous-frequency transform.

To obtain the inverse transform, we use the orthogonality relations 3.7 again, writing

$$\begin{aligned}\frac{1}{N}\sum_{k=0}^{N-1}\tilde{x}_k e^{2\pi ink/N} &= \frac{1}{N}\sum_{k=0}^{N-1}\sum_{m=0}^{N-1} x_m e^{2\pi i(nk-mk)/N} = \frac{1}{N}\sum_{k=0}^{N-1}\sum_{m=0}^{N-1} x_m e^{2\pi ik(n-m)/N} \\ &= \frac{1}{N}\sum_{m=0}^{N-1} x_m N\delta_{nm} = x_n\end{aligned} \tag{3.9}$$

where the next-to-last step makes use of the orthogonality relationship (3.7). We thus have a transform pair – the DFT (equation 3.8) and the inverse DFT (equation 3.9) – between finite-length sequences of numbers.

## 3.5 Fourier Theorems for the DFT

While the DFT and inverse DFT are completely consistent, there are notable pitfalls in using them, mostly arising from trying to apply lessons in

continuous and infinite time, some of which do not carry over into the world of finite-length sequences. The basic confusion lies in supposing that our finite sequence is just a part of a longer sequence, so that we can apply continuous-time (and infinite-range) Fourier theory simply by multiplying the series by a function that is zero over most of the range, and one over some part of it: that is, by a scaled version of the boxcar $\Pi(t)$).

This kind of multiplication by a function that is nonzero only over some range is called **windowing**. It is tempting to take the view that the DFT of a finite sequence is the FT of a windowed version of an infinite series; after all, the data we analyze are usually obtained from a longer series in just this way. However, this view is *wrong*. Once the data have been obtained, the process of getting them ceases to matter: we have to regard them as a finite sequence, not part of some longer one, and use the mathematics appropriate to such sequences.

As an introductory example, suppose we compute the value of the inverse DFT of the DFT of a sequence for term numbers outside the original range from 0 to $N-1$. If we do this for the regular Fourier transform of a windowed function, we of course get the original function back; having been windowed, it is indeed zero except over a limited range. But this is not what happens for the DFT. From the definitions of the transforms, we can write the "recovered" series $x^{rec}$ as

$$x_n^{rec} = \sum_{m=0}^{N-1} x_m \left[ \frac{1}{N} \sum_{k=0}^{N-1} e^{2\pi i k(n-m)/N} \right] \tag{3.10}$$

Now, by 3.7, the part in square brackets is nonzero (and indeed equal to one) for $n-m$ a multiple of $N$; that is, for $n-m = jN$, which we can rewrite as $n = m + jN$, where $j$ is any integer. The usual way to express this relationship between $n$ and $m$ is to say that "$n$ equals $m$, modulo $N$", the notation for which is

$$m = n_{\mathrm{mod}N}$$

which, if we apply it to (3.10), gives us the result

$$x_n^{rec} = x_{n\mathrm{mod}N}$$

In this expression $n$ can be any integer, but $n_{\mathrm{mod}N}$ ranges only from 0 through $N-1$: this is just the range over which we actually know the original sequence. The infinite sequence we have thus "recovered," and which is in some sense equivalent to the finite sequence, is therefore the finite

sequence replicated over and over; because it is periodic, it contains no information not in the original sequence – and after all, how could it?. If we want to apply our earlier Fourier theory to an infinite sequence, it has to be to one that is periodic, not to a finite sequence surrounded by zeros on both ends.

Another interesting example comes from supposing that our finite periodic sequence is just a sinusoid

$$x_n = e^{2\pi i f_0 n}$$

with $f_0$ real; the discrete Fourier transform of this is

$$\tilde{x}_k = \sum_{n=0}^{N-1} e^{2\pi i n(f_0 - k/N)} \quad k = 0 \ldots N-1$$

If, but only if, $f_0 = l/N$, with $l$ an integer, we get

$$\tilde{x}_k = \begin{cases} N & k = l \\ 0 & k \neq l \end{cases}$$

so that this finite sequence can have a (discrete) Fourier transform that is a discrete-frequency $\delta$-function, with only one nonzero value. In continuous time we get a $\delta$-function as the transform only if the time series is an infinitely long sinusoid; the transform of a section of a windowed sinusoid is

$$\mathscr{F}[e^{2\pi i f_0 t}\Pi(t/T)] = T\delta(f - f_0) * \mathrm{sinc}(fT) = T\mathrm{sinc}[(f - f_0)T]$$

which is not a $\delta$-function for any choice of $f_0$.

This all fits with the concept of replicating the sequence over an infinite interval: $f_0 = l/N$ means that exactly $l$ cycles occur over the finite segment, and repeating the function then gives an infinite, untruncated sinusoid – with a $\delta$-function transform. But if $f_0 = l/N$, the replicas of the sequence do not join smoothly, so the Fourier transform is not a $\delta$-function – and neither is the DFT. Figure 3.3 shows an example.

### 3.5.1 Shift Theorem

We next consider what is the effect in the time domain of some operation carried out in the frequency domain, with the "time domain" being a finite sequence. For conventional Fourier transforms, the shift theorem

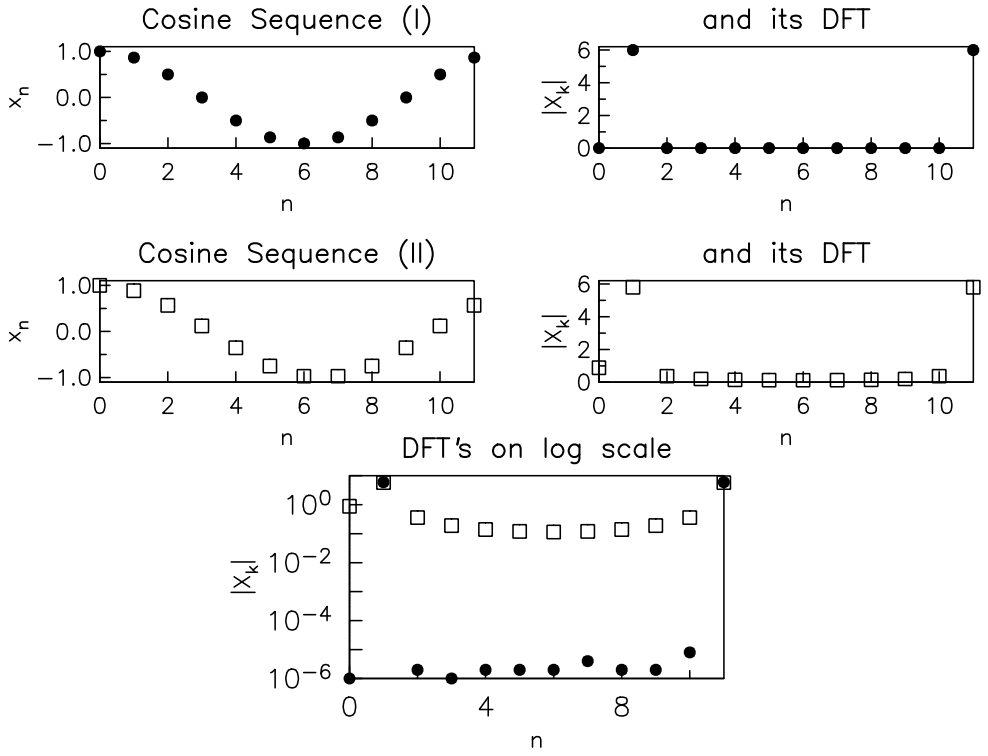



Figure 3.3: The top two pairs of plots show a 12-point sequence and its DFT, where the sequence is $x_n = cos(2\pi f n)$, with $n = 1/6$ in the top left plot and $n = 1/6.5$ in the middle left plot. For each the right-hand plot shows the corresponding DFT. Especially when plotted on a log scale (lowest plot), a very large difference is apparent between the DFT for a cosine with an integral number of cycles, (for which the DFT sequence is a $\delta$ sequence with roundoff error), and the DFT for one which is merely close to an integral number of cycles.

states that, if we take the Fourier transform of a function $x(t)$ shifted by an amount $\tau$, we get

$$\begin{aligned}\mathscr{F}[x(t-\tau)] &= \int_{-\infty}^{\infty} x(t-\tau)e^{-2\pi ift}\,dt = \int_{-\infty}^{\infty} x(t')e^{-2\pi if(t'+\tau)}dt' \\ &= \tilde{x}(f)e^{-2\pi if\tau}\end{aligned}$$

and so if we multiplied a Fourier transform $\tilde{x}(f)$ by $e^{-2\pi if\tau}$, and took the inverse transform, we would get $x(t-\tau)$.

Now we consider the parallel case for finite sequences. Call the DFT of $x_n$, $\tilde{x}_k$; and multiply $\tilde{x}_k$ by $e^{2\pi imk/N}$. What does the inverse DFT produce? We use $x^s_n$ for the series whose transform is $\tilde{x}_k e^{2\pi imk/N}$; then we write down the definitions for the DFT and inverse DFT, to get

$$\begin{aligned}x^s_n &= \frac{1}{N}\sum_{k=0}^{N-1}\tilde{x}_k e^{2\pi imk/N}e^{2\pi ink/N} = \frac{1}{N}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1}x_l e^{-2\pi ilk/N}e^{2\pi imk/N}e^{2\pi ink/N} \\ &= \sum_{l=0}^{N-1}x_l\left[\frac{1}{N}\sum_{k=0}^{N-1}e^{2\pi ik(m+n-l)/N}\right]\end{aligned}$$

The sum in square brackets is nonzero for $n = (l-m)_{\mathrm{mod}N}$, since then $n - (l-m)$ is a multiple of $N$. Thus

$$x^s_n = x_{(l-m)_{\mathrm{mod}N}}$$

To see what this means, it helps to take a specific value for $m$; if we take $m = 3$, then taking $l = 3$ gives $n = 0$, $l = 4$ gives $n = 1$, and so on:

$$x^s_0 = x_3 \quad x^s_1 = x_4 \quad \ldots \quad x^s_{N-4} = x_{N-1}$$

This is all as we expect, but the "right end" of the back-transformed series is

$$x^s_{N-3} = x_{N\mathrm{mod}N} = x_0 \quad x^s_{N-2} = x_{N+1\mathrm{mod}N} = x_1 \quad x^s_{N-1} = x_2$$

which is to say, the start of the original series. We can look at this in two ways. One way is to view this as a linear shift applied to a replicated, periodic sequence. The other way is to view it as a **circular shift**, where we imagine the finite sequence to be on a circle, so that any shift moves the end to the beginning.

### 3.5.2 Convolution Theorem for the DFT

A much more important result, with parallels to the one just given, is provided by the extension of the convolution theorem to discrete, finite-length series. We know that the Fourier transform of the convolution of two functions is the product of the two Fourier transforms; so what is the sequence produced by multiplying two sets of DFT coefficients and taking the inverse DFT? More precisely, suppose $x_n$ and $y_n$ to have DFT's $\tilde{x}_k$ and $\tilde{y}_k$; what series $z_n$ has the DFT $\tilde{z}_k = \tilde{x}_k \tilde{y}_k$? As before, the derivation just involves writing the inverse transform, replacing $\tilde{z}$ by the product $\tilde{x}\tilde{y}$, substituting in the DFT expressions for these, and finally collecting all the exponentials together:

$$\begin{aligned} z_n &= \frac{1}{N}\sum_{k=0}^{N-1} \tilde{z}_k e^{2\pi i n k/N} = \frac{1}{N}\sum_{k=0}^{N-1} \tilde{x}_k \tilde{y}_k e^{2\pi i n k/N} \\ &= \frac{1}{N}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1}\sum_{m=0}^{N-1} x_l e^{-2\pi i l k/N} y_m e^{-2\pi i m k/N} e^{2\pi i n k/N} \\ &= \sum_{l=0}^{N-1} x_l \sum_{m=0}^{N-1} y_m \frac{1}{N}\sum_{k=0}^{N-1} e^{2\pi i k(n-m-l)/N} \end{aligned}$$

Once again, we need the sum over $k$ to be nonzero; the orthogonality relation (3.7) then implies that

$$(n-m-l)_{\mathrm{mod}N} = 0 \quad \text{or} \quad m = (n-l)_{\mathrm{mod}N}$$

giving the final result that the sequence $z$ is given by

$$z_n = \sum_{l=0}^{N-1} x_l y_{(n-l)\mathrm{mod}N}$$

This is somewhat like the convolutions we have seen before, though not exactly. Again, an example may help; taking $n = 5$ gives

$$z_5 = x_0 y_5 + x_1 y_4 + \ldots x_5 y_0 + x_6 y_{N-1} + x_7 y_{N-2} + \ldots x_{N-1} y_6$$

meaning that as the convolution sum goes off the beginning of the $y_n$ sequence it goes around to the end. One way to visualize this is to imagine both sequences mapped onto circles: a shift is then merely a rotation of one circle relative to another, and the kind of convolution described by the above equation is therefore called **circular convolution**.

If we attempt to convolve two series in the time domain by multiplying their DFT's, we must be careful to ensure that such **"wraparound effects"** do not create any problems. These effects complicate such seemingly simple techniques as removing certain frequencies from data by taking the DFT, setting some of the DFT values to zero, and taking the inverse DFT. This can work – but we must remember that the series we get back can show the effects of data at one end influencing data at the other, since using the DFT makes the two ends contiguous.

It is indeed possible (and indeed often most efficient) to convolve two sequences by multiplying their transforms. To avoid wraparound problems, we need to pad both time series with zeroes before taking the DFT's; with enough zeros, the circularity of the convolution has no effect. Specifically, if we have an $M$-term series, $x_n$ and an $L$-term series $y_n$, we will get the correct result if we pad both out to length $M+L-1$ before doing the DFT. The circular convolution is then

$$z_n = \sum_{l=0}^{M+L-2} x_l y_{(n-l) \bmod M+L-1} = \sum_{l=0}^{M-1} x_l y_{(n-l) \bmod M+L-1}$$

giving (as examples from the two ends of $z_n$)

$$z_0 = x_0 y_0 + x_1 y_{M+L-2} + \ldots + x_{M-1} y_{M+L-M}$$
$$z_{M+L-1} = x_0 y_{M+L-2} + \ldots + x_{M-2} y_{M+L-M} + x_{M-1} y_{L-1}$$

For the first sum, all terms but the first are zero; for the second sum, all terms but the last, so we get the same series $z_n$ as we would from a time-domain convolution in which both sequences were padded with infinite numbers of zeroes on both ends.

### 3.5.3 Symmetry Relations

The DFT output runs from 0 to $N-1$, whereas we have been plotting the Fourier transform from negative to positive frequency. The connection is given by the result

$$\tilde{x}_{N-k} = \sum_{n=0}^{N-1} x_n e^{-2\pi i n(N-k)/N} = \sum_{n=0}^{N-1} x_n e^{-2\pi i n(-k)/N} = \tilde{x}_{-k}$$

This result is consistent with the notions of periodicity that we have been discussing; but it also means that the "upper half" of the DFT sequence

($k$ from $N/2$ to $N-1$) can also be thought of as the negative frequencies $k = -N/2$ to $-1$.

As with the Fourier transform in continuous time, we have symmetry rules for the transforms of special classes of sequences: for example, if the sequence $x_n$ is real, the transform $\tilde{x}$ is Hermitian. Combining this with the previous result, we see that, for a real sequence, half of the spectrum is redundant: usually we say this is the top half, or equivalently the negative frequencies. Formally, we have that if $x_n$ is real, then

$$\tilde{x}_{N-k} = \sum_{n=0}^{N-1} x_n e^{2\pi i n k/N} = \sum_{n=0}^{N-1} x_n^* \left(e^{-2\pi i n k/N}\right)^* = \tilde{x}_k^*$$

so the top half is just the complex conjugate of the bottom half. This makes sense, since if we put in $N$ real numbers, we expect to get $N$ independent numbers out – and we do, namely, $N/2$ independent complex DFT coefficients. Actually, for $x_n$ real, two $\tilde{x}$'s are real, namely

$$\tilde{x}_0 = \sum_{n=0}^{N-1} x_n \qquad \text{and} \qquad \tilde{x}_{N/2} = \sum_{n=0}^{N-1} (-1)^n x_n$$

Our final theorem for the DFT is Parseval's relation:

$$\begin{aligned}\sum_{n=0}^{N-1} |x_n|^2 &= \sum_{n=0}^{N-1} x_n x_n^* = \frac{1}{N^2} \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} \tilde{x}_k e^{2\pi i k n/N} \sum_{l=0}^{N-1} \tilde{x}_l^* e^{-2\pi i l n/N} \\ &= \frac{1}{N^2} \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} \tilde{x}_k \tilde{x}_l^* e^{2\pi i n (k-l)/N} = \frac{1}{N^2} \sum_{k=0}^{N-1} \tilde{x}_k \tilde{x}_k^* \sum_{n=0}^{N-1} 1 \\ &= \frac{1}{N} \sum_{k=0}^{N-1} |\tilde{x}_k|^2\end{aligned}$$

which is useful for checking the normalization (and accuracy) of any actual DFT algorithm.

## 3.6 The Dirichlet Kernel

We now use the result (3.4) to find the Fourier transform of the discrete-time version of the rectangle (or boxcar) function; remember that in continuous time the Fourier transform is $\mathrm{sinc}(f)$. In discrete time the Fourier transform is called the **Dirichlet kernel**; this is worth a section because

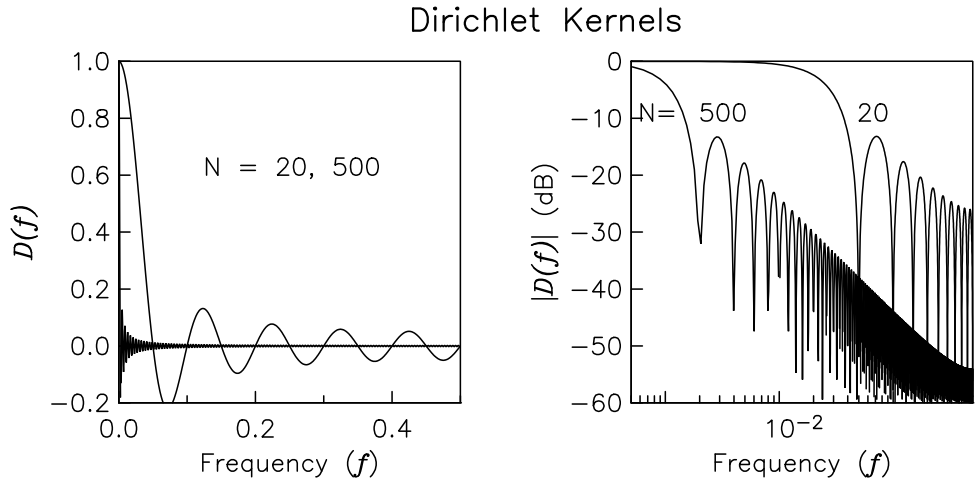



Figure 3.4: The Dirichlet kernel $D_N$ plotted for $N$ equal to 20 and 200, on the left on linear scales and on the right on log scales.

it plays a fundamental role in many topics in Fourier analysis and signal processing.

We consider a rectangle function in discrete time that is centered on the origin and runs from $-M$ to $M$; the total length $N = M + 1$ is then odd, but the derivation may be extended to an even length. The sequence is:

$$w_n = \begin{cases} 1 & |n| \le M \\ 0 & |n| > M \end{cases}$$

If we now form the Fourier transform of this (in continuous frequency, since this is an infinite sequence), we find

$$W(f) = \sum_{n=-\infty}^{n=\infty} w_n e^{-2i\pi fn} = \sum_{n=-M}^{M} (e^{-2i\pi f})^n = e^{2i\pi fM} \left[ \sum_{n=0}^{n=2M} e^{-2i\pi fn} \right]$$

We evaluate the sum in square brackets using equation (3.4), with $2M = N - 1$; the result is

$$W_M(f) = \frac{\sin \pi(2M+1)f}{\sin \pi f}$$

The Dirichlet kernel is normalized slightly differently:

$$D_N(f) = \frac{\sin \pi N f}{N \sin \pi f}$$

which means that $W_M(f) = (2M+1)D_{2M+1}(f)$, and also that at $f = 0$ the value of the Dirichlet kernel is one. Sometimes the Dirichlet kernel is normalized so that $D_{2M+1}(f) = W_M(f)$, in which case the value at zero depends on $N$ but the total area under all the lobes is invariant. Either way, the zeroes are at $f = l/N$, with $|l|$ ranging from 1 through $N/2$. The first minimum, which is close to $f = 3/2N$, is about $-N/5$, the next extremum is about $N/7$, and so on. Away from zero the function thus decays rather slowly with increasing $f$.

To see the connection between the Dirichlet kernel and the sinc function, put $y = Nf$ and let $N$ go to infinity while keeping $y$ fixed: this is analogous to finer and finer sampling. Then $D_N(f)$ behaves like

$$\frac{\sin \pi y}{N \sin \pi y/N} \to \frac{\sin \pi y}{\pi y} = \mathrm{sinc}(y)$$

## 3.7 Computing the DFT: the Fast Fourier Transform

Many methods of signal processing and data analysis require that we compute the DFT – sometimes many DFT's. This computation can be done rapidly, though this is not immediately obvious; the DFT formula, equation (3.8), appears at first sight to be the multiplication of an $N$-length vector (the sequence $x_n$) and an $N$-by-$N$ matrix (the $e$'s) to produce another $N$-vector (the $\tilde{x}_k$'s). Computing such a matrix multiplication in general requires a multiple of $N^2$ operations (additions and multiplications), an algorithm that we say takes **polynomial time**: the number of operations is a polynomial (above a linear one) of the number of data. It might also appear that we have to compute $N^2$ sines and cosines (for the matrix of $e$'s), but this is not so; because of the periodicity of $e^{-2\pi im/N}$, we have $e^{2\pi ink/N} = e^{-2\pi i(nk)_{\mathrm{mod}N}/N}$, so that while the matrix of $e$'s has $N^2$ elements there are only $N$ distinct values.

In fact, the periodic nature of $e^{2\pi ink/N}$ allows us to compute the DFT with many fewer than $N^2$ operations. For certain $N$ we can compute all the $\tilde{x}_k$'s with only $N \log_2^N$ operations, so that the computation can be done $N/\log_2 N$ times faster: for example, 100 times faster for $N = 1024$. The algorithm that does this is called the **Fast Fourier Transform** (or FFT).

You should realize that the FFT is just an algorithm for computing the DFT quickly, not a definition of the DFT. A common error is to suppose that

any restrictions that apply to an FFT algorithm (notably restrictions on $N$) also apply to the DFT; but this is not so: the DFT is defined for any $N$.

The FFT algorithm is not completely obvious, as is shown by how many people failed to develop it: at the time the FFT was first announced, in 1965, many groups were computing DFT's using matrix multiplication, and it was with some amazement that they learned how much time they could save just by using a different procedure.[2] At the same time, the FFT was obvious enough that by 1965 it had already been invented independently several times: but either the inventors dismissed it as too small an improvement to be worth publishing (true for $N$ very small), or the invention was simply ignored.[3]

The publication, and popularization, of an FFT algorithm for $N = 2^M$, by Cooley and Tukey (1965) led to a complete change, and since then, devising faster transforms, and fast techniques for other transforms, has become a subarea of signal processing.

The FFT is something you are very unlikely to program yourself – indeed, you shouldn't, since other people have spent lots of time doing so. We have therefore relegated our derivation of it to Appendix A. But it is worth discussing the FFT because it illustrates several points:

1. The FFT shows that considerable increases in speed can come from using a good algorithm; for whatever problem you will be doing, you should become familiar with how the time needed (the operations count) scales with problem size. There is a big difference between scaling as $N \log N$, which is not much worse than scaling as $N$ (**linear time**), and scaling as a power of $N$ (even if the power is only $N^2$). Even worse is when a computation scales not in polynomial time but in **exponential time**: that is, the operations count scales as $e^N$. If the only algorithms available have this scaling, computational methods are impracticable unless $N$ is quite small.

2. A related lesson from the FFT is that the scaling with $N$ is much

[2] For example, Alsop and Nowroozi (1966), computing spectra of seismic data for free oscillations, reported that their computation time shrank from 23 minutes to 2.4 seconds.

[3] The first (and uninfluential) discovery of an FFT algorithm was by Gauss in 1805 (Heideman *et al.*, 1985). A notable example of the use and non-use of the FFT in the early 1960's was here at SIO: Philip Rudnick, at MPL, had been using his own version of an FFT for a number of years, but only published it after Cooley and Tukey published theirs. At the same time, Walter Munk's group, at IGPP, was computing transforms in the less-efficient way.

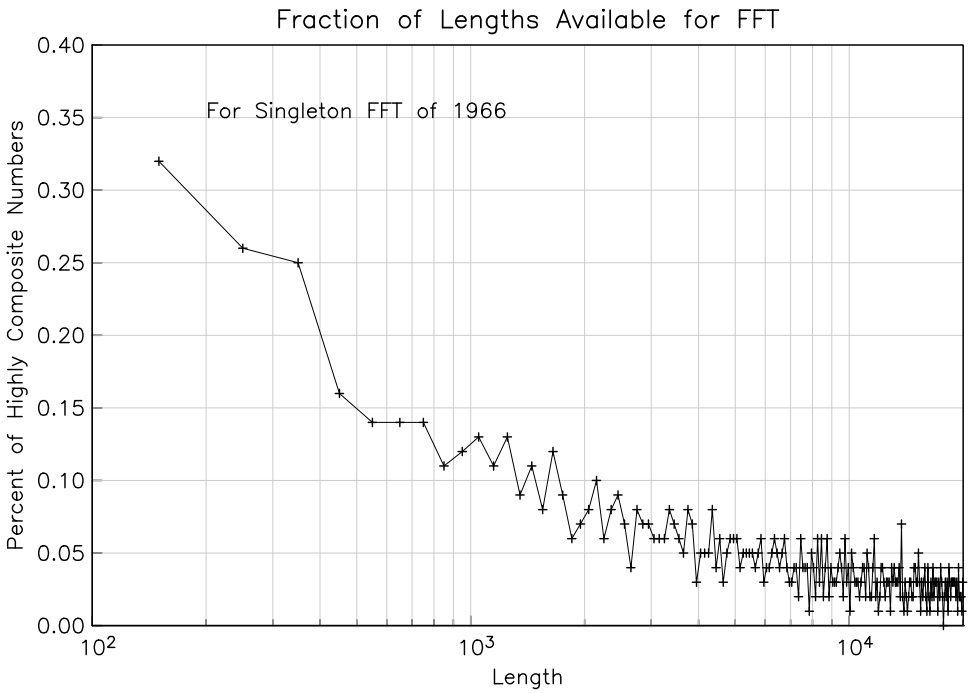



Figure 3.5: Fraction of highly-composite numbers for all values less than $N$, as a function of $N$. "Highly-composite" means having no prime factors larger than 23, a not-uncommon restriction in some implementations of the FFT.

more important than the details of implementation. The FFT at once gave at least two orders of magnitude improvement in speed for even moderate values of $N$. Thirty-five more years of effort have produced about a factor of four improvement between a general-purpose FFT from 1969 (Singleton, 1969) and the best one available in 2004.[4]

3. The shortest and simplest FFT algorithms are for $N$ a power of 2. But this is not a requirement: there are algorithms that are nearly as fast if $N$ is **highly composite**: that is, can be factored into a product of small primes. While some special cases ($N$ prime) also can be done quickly, you should if possible avoid values of $N$ whose

[4] FFTW, currently the fastest general-purpose program (hence the name: Fastest Fourier Transform in the West). For information on FFTW, and links to lots of other FFT programs and results, see `http://www.fftw.org`. Note that FFTW only gives its best performance if you will be doing many transforms of the same length within a single execution of the program – which may well not be true.

factors include large primes, Fortunately this is not difficult. Figure 3.5 shows the percent of lengths available, as a function of length; while this decreases towards zero, it does so slowly: up to lengths of 10000, about 5% of the lengths are feasible. Another way to put this is that for lengths greater than 1000, the largest change in series length needed to be at an acceptable value is almost always less than 1%. You should approach with care any DFT program that will handle all $N$; if $N$ is not highly composite, then what the program actually does may be a slow Fourier transform.

4. If, as is usually the case, your sequence $x_n$ is real, you should not simply treat it as complex with all imaginary parts zero; you can get a factor of two improvement in speed by instead treating alternate terms as real and imaginary parts of a sequence of length $N/2$. The DFT result then needs some auxiliary processing, which we describe in Appendix A.

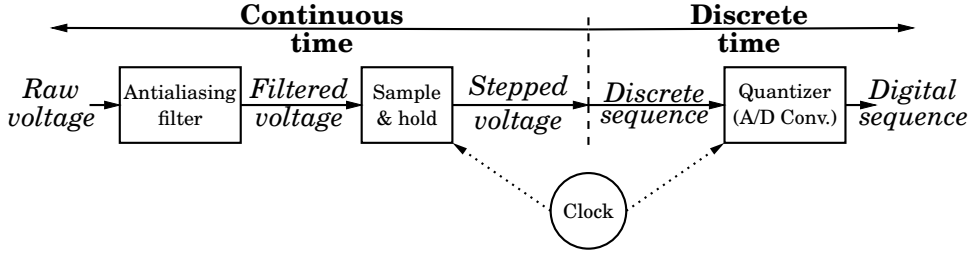
5. If you only need the transform $\tilde{x}$ at a few frequencies, or at those that are not automatically produced by the DFT relationship, there are special algorithms for rapid computation. We describe one, the Goertzel algorithm, in Appendix A.

CHAPTER 4

# SAMPLING THEORY FOR TIME-SERIES

## 4.1 Introduction: Logging Data

We now consider the transition between continuous time and discrete time, namely the **sampling** of a function at specified times (or, for spatial data, specified places) to get the numbers that we later process. This sampling is what takes place in a digital datalogger, which converts an analog function of continuous time (almost always a voltage) $x(t)$, into a sequence of numbers $x_n$, each number being the value of $x(t)$ at some instant: $x_n = x(t_n)$. For this chapter we use $x_n$ for a single sample and $x_n$ for a sequence of samples.




The block diagram above shows the steps of converting an analog signal to a digital sequence; Figure 4.1 shows the corresponding time series. The first step is to to remove high frequencies from the continuous function; this is done with an analog filter (we will encounter a simple analog filter in the next chapter). This is called an **anti-aliasing filter**. for reasons we will see below. The filtered signal now varies smoothly; it goes to a **sample-and-hold circuit**; this produces a continuously-varying voltage that is held at fixed values for a time determined by a clock; each time a clock signal goes to the sample and hold, the value of the output changes to the value of the input at that time ($t_n$). We call this the "stepped voltage".

This voltage is still a function of time $x(t)$, but it has the same information as, and so can be be regarded as equivalent to, a sequence of values $x_n$.

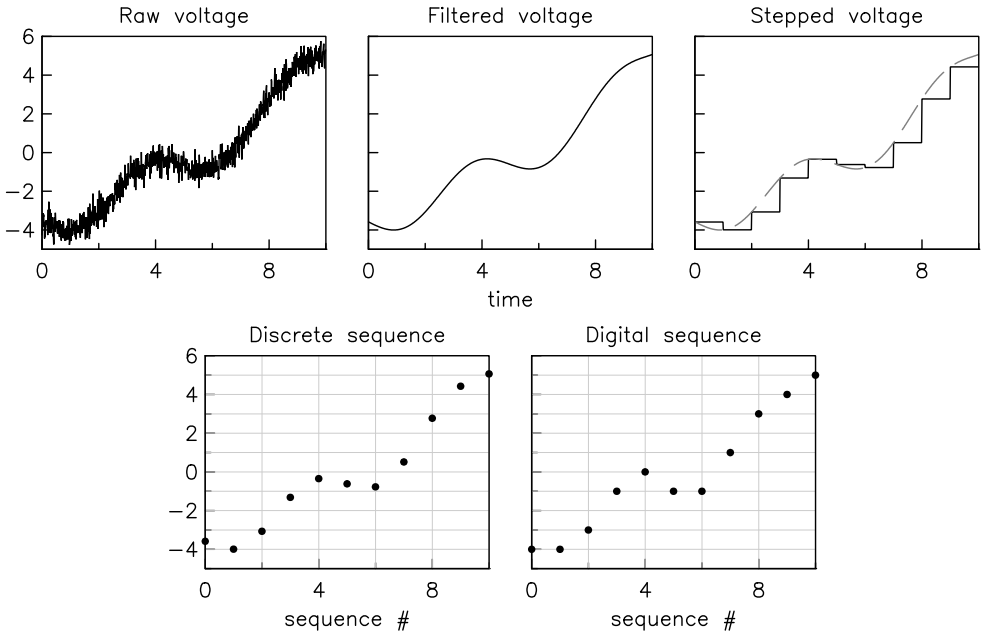



Figure 4.1: The upper block diagram shows the steps included in recording an analog signal and converting it to a digital sequence; the lower plots are cartoons of the different forms of the signal at each step. See text for details.

In the block diagram this point is the division between continuous and discrete time, though in the datalogger there is no explicit division. The next stage in the datalogger is a quantizer, or **analog-to-digital converter** (usually called "an A to D"). This device is a nonlinear system which, given a real number $x$, produces a value $\breve{x}$ that can be represented in a finite number of digits; the reason for the sample-and-hold is to give a constant input voltage to the A-to-D for the amount of time needed for it to compute its output value. We naturally want $\breve{x}$ to be as close to $x$ as possible, and it can be an interesting problem to choose a representation for $\breve{x}$ and then design the quantizer to make the difference $\breve{x} - x$ unimportant. However, in practice this is usually not too difficult, as is suggested by the bottom two plots, which show the discrete sequence $x_n$ and the digital sequence $\breve{x}_n$, quantized by rounding to the nearest integer. For most of this chapter we assume that we have the real-valued $x_n$'s available; we discuss quantization error very briefly at the end.

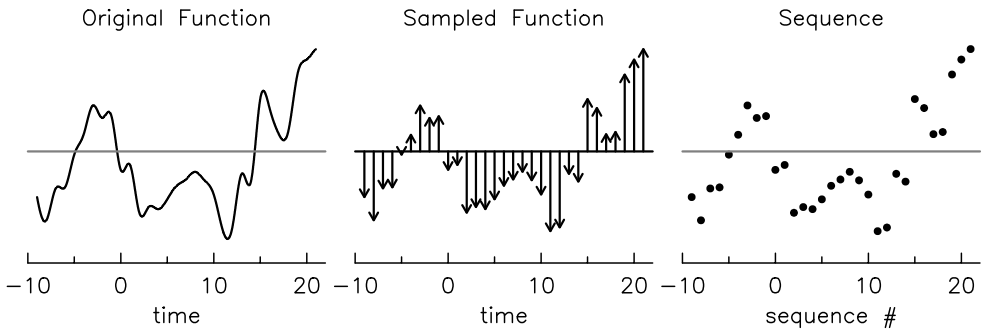



Figure 4.2: A mathematical model for sampling, shown in pictures. We start with a function (on the real line), then create another function whose values depend only on those of the sampled data. The arrows are schematic for differently-weighted $\delta$-functions. If we can reconstruct the original function from the sampled-data function, we have lost no information by sampling. The gray lines in the left and right panels are the zero of the y-axis.

## 4.2 The Sampling Problem

What we shall discuss for most of the chapter is how to choose the $t_n$'s such that, given $\{x_n\}$ we could (in principle) reconstruct $x(t)$ – and what errors we incur if we cannot. Classical interpolation theory would be one way of studying this; but a different approach, working in the frequency domain, turns out to be much more useful.

To avoid what would be substantial complications, we require the samples to be equispaced in time; the spacing is the **sample interval** $\Delta$, given in whatever time units we are using, so the times are:

$$t_n = n\Delta \qquad n = -\infty, \ldots -2, -1, 0, 1, 2, \ldots \infty$$

In the last chapter we implicitly set $\Delta$ to be unity; for now we do not do this, though we will assume it to be unity later on.

Slight departures of $t_n$ from absolutely equal spacing do not much matter, since they can be viewed as adding some error (noise) to the data; for a timing error $\epsilon$, the sample value $x(n\Delta + \epsilon)$ will be approximately $x(n\Delta) + \epsilon\dot{x}(n\Delta)$, and so long as the second term is usually smaller than the first there will not be a problem. As we saw in our earlier discussion of the Discrete Fourier Transform, larger irregularities complicate analysis greatly and are best avoided if at all possible, if necessary by filling in missing data.

To get from functions to sequences we need an intermediate step. We start with $x(t)$, a function in continuous time; mathematically we say that it is a function on the real line, which we would express pictorially as the left-hand panel of Figure 4.2. The sequence $x_n$, on the other hand, is just an array of numbers, which we would express pictorially as the right-hand panel of Figure 4.2. We connect these two using the delta-function $\delta(t-t_n)$; the (generalized) function

$$x(t)\delta(t-t_n)$$

is zero at $t \neq t_n$, and contains the value $x(t_n)$ in the weighting of the delta-function. It thus is equivalent to the sampled value $x_n$: not equal to it, since one is a function and the other a number, but equivalent in the sense that either could be constructed given the other. Such a link between functions and numbers is just what we seek.[1]

For equispaced sampling the sequence $x_n$ is thus equivalent to the function

$$x(t)\sum_{n=-\infty}^{\infty}\delta(t-n\Delta)=\sum_{n=-\infty}^{\infty}x(n\Delta)\delta(t-n\Delta)=x_s(t)$$

which is shown in the middle panel of Figure 4.2 (using arrows for delta-functions). We can view this function as formed by multiplying $x(t)$ by an infinite "comb" of delta-functions. The normalized version of such a comb is the III function (so called by Bracewell because it looks like the Hebrew letter *shah*):

$$\mathrm{III}(t)=\sum_{n=-\infty}^{\infty}\delta(t-n)$$

Changing the spacing to $\Delta$ rescales the III, making our sampled-data function, for a sample interval $\Delta$,

$$x_s(t)=\frac{1}{\Delta}\mathrm{III}(t/\Delta)x(t)$$

## 4.3 The Nyquist Theorem

To return to our original question, we can, given $x_n$ easily get $x_s(t)$ as a weighted array of delta-functions, but how do we get (and can we get) $x(t)$

[1] It might seem more in keeping with the action of a sampling circuit to multiply $x(t)$ by a function that is 1 at $t=t_n$ and zero elsewhere. But such a function, even multiplied by any finite value, has an integral of 0. This makes any integral transform, such as the Fourier transform, also 0: not very useful.

from $x_s(t)$? We approach this problem by comparing $\tilde{x}_s(f) = \mathscr{F}[x_s(t)]$ with $\tilde{x}(f) = \mathscr{F}[x(t)]$. By the convolution theorem,

$$\tilde{x}_s(f) = \mathscr{F}\left[\frac{1}{\Delta x(t)}\mathrm{III}(t/\Delta)\right] = \frac{1}{\Delta\tilde{x}(f)} * \mathscr{F}[\mathrm{III}(t/\Delta)]$$

To proceed further we need $\mathscr{F}[\mathrm{III}(t)]$, which is

$$\mathscr{F}[\mathrm{III}(t)] = \int_{-\infty}^{\infty} e^{-2\pi i f t} \sum_{n=-\infty}^{\infty} \delta(t-n)dt = \sum_{n=-\infty}^{\infty} e^{2\pi i f n}$$

An extremely nonrigorous way to see what this might give is to observe that for $f$ an integer this is $\sum 1$, and hence infinite; while for $f$ noninteger $2\pi i f n$ assumes all possible values (modulo $2\pi$) and so the sum will be zero. This suggests what is the actual result:

$$\mathscr{F}[\mathrm{III}(t)] = \mathrm{III}(f)$$

that is, that the III function is its own Fourier transform. This can in fact be proved rigorously.

We then have

$$\tilde{x}_s(f) = \tilde{x}(f) * \frac{1}{\Delta}\mathscr{F}[\mathrm{III}(t/\Delta)] = \tilde{x}(f) * \mathrm{III}(\Delta f)$$

Convolving any function with a $\delta$-function simply recovers the original, and if we convolve with $\delta(f - f_0)$ we recover $\tilde{x}(f)$ shifted by an amount $f_0$. Thus

$$\begin{aligned}\tilde{x}_s(f) &= \tilde{x}(f) * \mathrm{III}(\Delta f) = \Delta \sum_{n=-\infty}^{\infty} \delta(f - n/\Delta) * \tilde{x}(f) \\ &= \Delta \sum_{n=-\infty}^{\infty} \tilde{x}(f - n/\Delta)\end{aligned}$$

Our sampled-data series $x_s$ thus has a Fourier transform that can be gotten by replicating that of the original series at frequency interval $\Delta^{-1}$, and adding up the replicas. Figure 4.3 shows the whole process.

We can now show that if $\tilde{x}_s$ has a certain form, we can recover $x(t)$ from $x_s(t)$. The requirement for this is called the **Nyquist criterion**, and is given in the frequency domain:

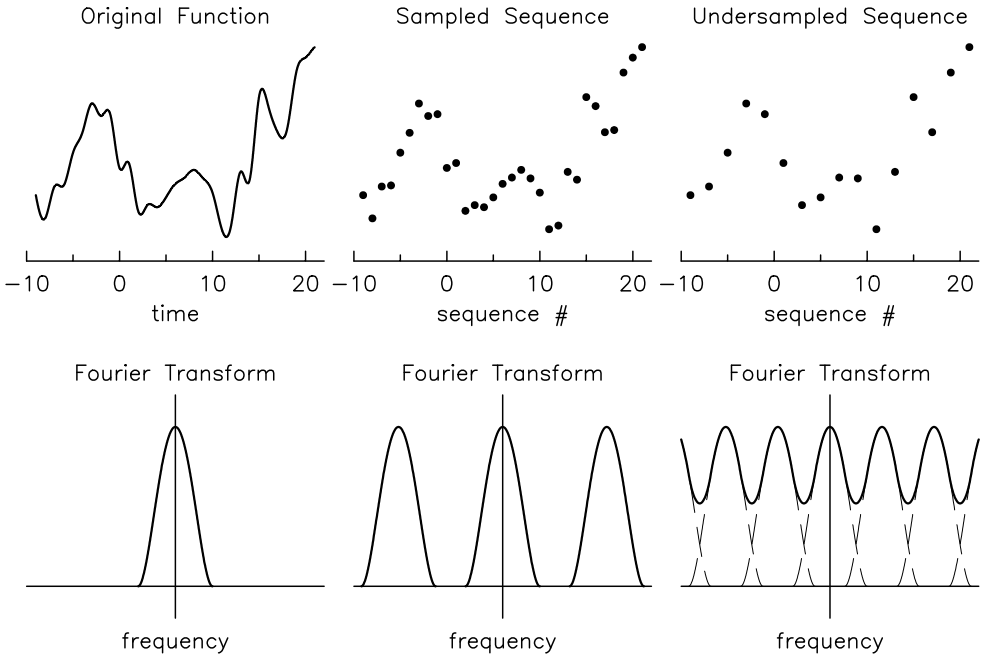



Figure 4.3: The upper panels show a function, then a sampled version of it, then a version sampled half as often. The lower panels show the corresponding Fourier transform: the original transform (this is actually a made-up function), and the replicated-and-summed version that results from sampling. In the middle, the transform replicas are separated; on the right, they are not. The right plot shows the replicas (aliases) with dashed lines.

$\tilde{x}(f)$ can be recovered from $\tilde{x}_s(f)$ and hence $x(t)$ recovered from $x_s(t)$, if and only if

$$\tilde{x}(f) = 0 \qquad \text{for} \qquad |f| > \frac{1}{2\Delta} \stackrel{\text{def}}{=} f_N \tag{4.1}$$

where $f_N$ is called the **Nyquist frequency**.

This requirement for $\tilde{x}(f)$ comes from its replicated nature. The replicas are spaced $1/\Delta$ apart; if the Nyquist criterion holds, each replica will remain separate, as in the lower middle panel of Figure 4.3. However, if $\tilde{x}(f)$ is nonzero above $f_N$, the replicas will overlap, and when summed will blend together, as in the lower right panel of Figure 4.3. There is then no way of disentangling them.

If the Nyquist criterion holds, we also have a way of getting $\tilde{x}(f)$ from $\tilde{x}_s(f)$: we simply set all replicas but the center one to zero. by multiplying $\tilde{x}_s(f)$ by a suitable boxcar function, scaled to be zero outside $|f| < f_N$:

$$\tilde{x}(f) = \tilde{x}_s(f)\Delta\Pi(f\Delta)$$

Taking inverse transforms we have

$$\begin{aligned} x(t) = x_s(t) * \mathrm{sinc}(t/\Delta) &= \mathrm{sinc}(t/\Delta) * \sum_{n=-\infty}^{\infty} x(n\Delta)\delta(t-n\Delta) \\ &= \sum_{n=-\infty}^{\infty} x(n\Delta)\mathrm{sinc}\left(\frac{t-n\Delta}{\Delta}\right) \end{aligned} \tag{4.2}$$

The final sum is the expression for getting $x(t)$ from $x_n$; note that if we put $t = m\Delta$ the zeros of the sinc function fall on all the other values than the one of interest, so we recover only one, $x(m\Delta)$, as we should.

It is useful to look at a pure sinusoid to see how the Nyquist criterion works. Suppose we have a sine (or cosine) with $f = f_N$, so there are two points per cycle. The cosine then seems to be recoverable, but the sine is zero at all points sampled. Since

$$\mathscr{F}[\sin(2\pi t/T)] = \frac{1}{2i}[\delta(f-1/T) - \delta(f+1/T)]$$

we see that for $\Delta = T/2$, the replicas just meet, causing the opposing delta-functions to cancel, so that $\tilde{x}_s(f) = 0$, consistant with the sampling. But for a cosine

$$\mathscr{F}[\cos(2\pi t/T)] = \tfrac{1}{2}[\delta(f-1/T) + \delta(f+1/T)]$$

and for $\Delta = T/2$, the two $\delta$-functions meet and reinforce, so that $\tilde{x}(f)$ and thus $x(t)$ can be recovered.

If $\tilde{x}(f) = 0$ for $|f - f_c| < f_b$, we say that $x(t)$ is **bandlimited**, with a **bandwidth** $2f_b$ and **center frequency** $f_c$. A sine wave modulated in amplitude would be an example of such a function. For the Nyquist criterion to hold, $x(t)$ must be bandlimited about zero with a bandwidth $2f_N$; when we sample data we ideally want $\Delta$ to have a value that will make this true.

## 4.4 Aliasing

But what if $x(t)$ is not bandlimited? And it never will be, since no transient function is.[2] The best way to understand the resulting errors is again to

[2] The apparent paradox that no continuous function can be limited in both time and frequency (which is what we expect of all signals we record) is discussed by Slepian (1976).

look at replication of the Fourier transform in the frequency domain.

To start with, we suppose that $x(t)$ is nearly bandlimited and falls to near zero at large frequency (otherwise we might as well give up), and that $f_N$ is large enough that only the overlap with the two nearest neighbors matters much:

$$\tilde{x}_s(f) \approx \tilde{x}(f) + \tilde{x}(f - 1/\Delta) + \tilde{x}(f + 1/\Delta)$$

For frequencies near $f_N$, the major overlap is with $\tilde{x}(f_N - 1/\Delta) \approx \tilde{x}(-f_N)$; the other overlap is with $\tilde{x}(3f_N)$ which we assume is smaller. Thus for "large" positive frequency, the major contamination is from energy at "large" negative frequency.

We have used the term "replica" to emphasize what the convolution with III does, but this is not the usual terminology – usually the replicas are called **aliases** of the transform of the continuous-time function, the contaminating energy being termed **aliased energy**, and the whole process **aliasing**. The expression (4.4) shows that for a complex-valued sequence the effect of sampling is to alias energy below the negative Nyquist frequency onto positive frequencies just below the positive Nyquist frequency.

In the more usual case that $x(t)$ is real, you may remember that $\tilde{x}(-f) = \tilde{x}^*(f)$, so we can write the "nearest-neighbor" aliases as

$$\tilde{x}_s(f) \approx \tilde{x}(f) + \tilde{x}^*(2f_N - f) + \tilde{x}(2f_N + f)$$

It is useful to just consider $|\tilde{x}|$ and ignore the complex conjugates. Consider the positive frequencies from 0 to $f_N$, and consider what the aliased frequencies are as $f$ goes from 0 to $f_N$ (which we write as $0 \to f \to f_N$:

$$\begin{array}{ccccccc} 0 & \to & f & \to & f_N & \text{(main)} & \\ 2f_N & \to & 2f_N - f & \to & f_N & \text{(first alias)} & (4.3) \\ 2f_N & \to & 2f_N + f & \to & 3f_N & \text{(second alias)} & \end{array}$$

We could continue this on indefinitely, but it serves to show that, for $x(t)$ real, we can regard sampling as mapping the real line of positive frequencies, from zero to infinity, into the finite interval 0 to $f_N$; if $x(t)$ is complex, sampling maps the entire real line into $-f_N$ to $f_N$.

Another way of seeing this is to note that our result (4.2) for the reconstructed $x(t)$ is a sum of sinc functions, and hence will be, as they are, bandlimited. Given a sequence of samples, we can construct an infinity

This paper is well worth reading for its commentary on the relationship between mathematical idealizations and the real world.

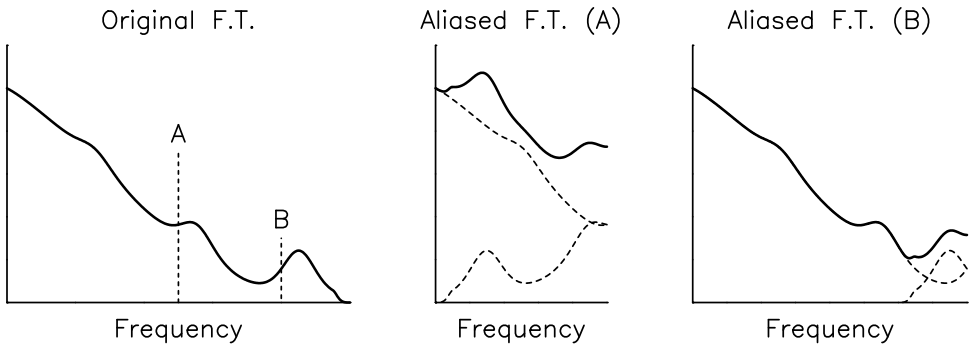



Figure 4.4: The left panel shows a made-up Fourier transform (amplitude), with two possible locations for the Nyquist frequency. The next two panels show what the transform would be for these two choices, with the aliases dashed and their sum solid. Choice A gives a badly-aliased result, choice B one that is probably acceptable, since the only aliasing is near the Nyquist frequency.

of functions on the real line – but only one of these will be bandlimited. For real functions and sequences, that bandlimited one has the transform given by mapping $[0,\infty]$ into $[0, f_N]$ according to the rule (4.3), continued to infinite frequency. One way to visualize this rule is as an accordion folding of the frequency axis (a fold at $f_N$ brings $2f_N$ to 0; then another fold at $2f_N$ brings $3f_N$ to $f_N$, and so on). We can write a formula for this: for $f > f_N$, let $N$ be the integer part of $f/f_N$; then $f$ maps to

$$f' = \begin{cases} f - Nf_n & N \text{ even} \\ & \text{for} \\ (N+1)f_n - f & N \text{ odd} \end{cases}$$

In practice, we can do two things to minimize the effect of aliasing:

- Before we sample the data, shape its Fourier transform using an analog filter (physical or electronic) to remove high frequencies: this is the "antialias" filter in the block diagram at the start of the chapter.
- Choose $\Delta$ so that what aliased energy there is will be much less than the energy in the main transform in the frequency band of interest. We do not want "significant" contamination, but what level is significant depends on the problem in hand. Figure 4.4 shows an example; while choice B allows some aliasing, the amount is small except near $f_N$, which is probably acceptable. Near $f_n$, $\tilde{x}(f)$ is *always* slightly

aliased; our concern is usually with lower frequencies. In actual data an indication of aliasing is successive values with alternating sign (perhaps around a trend); this behavior means that there is substantial energy near $f_N$, most likely because of significant aliasing.

For a somewhat specialized example of what aliasing might do, consider the effect of 60-Hz noise on data sampled less often (unless it is filtered out, 60-Hz hum is common on most electronic signals). If $\Delta$ is 1 second, $N$ (see above) is 120 ($f_N = ½$ Hz) and $f'$ is 0. This would be acceptable, except that "60-Hz" line noise is really 60 Hz $\pm 700\mu$ Hz, which maps into 0 to 700 $\mu$Hz; this would interfere with a signal with a period of 12 hours. We can avoid this peblem by sampling at some different interval; if we sampled at 2048 times per hour, $f_N$ = 0.2844 Hz, $N$ = 210, and $f'$ = 0.2667 Hz, so that any 60-Hz energy aliases to near the Nyquist, out of the way of lower-frequency signals.

## 4.5 Decimation

A very common activity in digital signal processing is **decimation**. We say that we have decimated a discrete-time series by $n$ if we take every $n$-th value to create a new and shorter series with sample interval $n\Delta$. (Another term for this is **downsampling**.) The new series is related to the one it came from by exactly the same aliasing rules that apply to sampling of a continuous-time series; the only difference is that the folding of the frequency axis extends only up to the Nyquist frequency for the original series, instead of to infinity.

For example, if we decimate by two, there is a single folding, so that frequencies from the old Nyquist $f_N$ to the new one ($½f_N$) fold into the new interval from 0 to $½f_N$. Just as in the continuous case, the degree to which the decimated series is aliased depends on the relative levels of the Fourier transform above and below the new Nyquist frequency; before decimation we may need to digitally filter the data, as we describe in the next chapter, to avoid aliasing of the output.

The opposite of decimation is putting additional values between each value in a sequence to give a series sampled more often; this is called either interpolation, or, what else, **upsampling**. Although it seems counterintutive, we can do this just by setting the new values to zero and then digitally filtering the series to remove energy above the old Nyquist frequency. One reason for doing such an interpolation would be to create a smooth series

for conversion back into analog form: standard practice in digital music players.

Decimation and interpolation can be combined to change sample rates; we might, for example, decimate by 5 and then interpolate by 3 to get a new rate that is 0.6 of the old. But whatever we do, we need to worry about, and correct for, any possible aliasing.

## 4.6 Violating the Nyquist Criterion

We have discussed the importance of not aliasing data – but there are situations in which the data can violate the Nyquist criterion without loss of information. We might, for example, have additional information at the sample times, such as the values of both $x(t)$ and its derivative $\dot{x}(t)$), though this is rarely the case. More often, the signal has a special structure that allows us to sample less often:

- A periodic signal can, under certain conditions, be measured adequately while being sampled at random times: one practical area where this occurs is in measurements of regularly variable stars, since for the longest-period ones the observations may occur at random times over many decades.

- If the signal can be described by a few parameters, these may be determined even though there is aliasing. To take an example from some work done here at IGPP, the signal produced by the interferometer in an absolute gravimeter is what is called a chirp, has a time-varying frequency:

  $$x(t) = x_0 + cos[((f_0 + f_1 t)t + \phi]$$

  where the parameter $f_1$ is related to the local acceleration of gravity. The frequency rapidly increases to a value that cannot be sampled fast enough to satisfy the Nyquist criterion, but it is still possible to process the signal to determine its parameters (Parker *et al.*, 1995).

- Another case is a real signal that is bandlimited between frequencies $f_l$ and $f_h$. To not lose information, we need to sample it so that the sampling "folds" the frequency axis [0, $\infty$) into the interval $[0, f_n]$, none of the folds fall within the frequency band $f_l$ to $f_h$. This is

equivalent to requiring that, for for some integer $n$ (not initially determined) the Nyquist frequency $f_n$ satisfies

$$(n-1)f_n \le f_l \quad \text{and} \quad nf_n \ge f_h$$

which combine to give

$$\frac{f_h}{n} \le f_n \le \frac{f_l}{n-1}$$

This in turn means that $n$ must be such that $f_h/n \le f_l/(n-1)$, equivalent to requiring that

$$n \le \frac{f_h}{f_h - f_l}$$

Taking $n$ to be the largest integer that satisfies this inequality, we then have that $f_n \ge f_h/n$, which means that the sample frequency $f_s = 2f_n$ must satisfy

$$f_s > 2f_h/n$$

and, from the other inequality involving $f_l$,

$$f_s < \frac{2f_l}{n-1}$$

## 4.7 Quantization Error

We close this chapter by discussing briefly the effects of quantization of a signal; we consider the simple (and common) case in which an A-to-D converts a real-valued input voltage to an output integer. A-to-D's are specified by the number of bits in this integer: for example, a 16-bit system outputs values from 32767 to 32768 (plus and minus $2^{15}$). over the full range of input voltage. So the more bits, the smaller the voltage corresponding to a unit change in the output, and the larger the ratio of the maximum signal to the smallest one that can be detected. This ratio is called the **dynamic range**, for an $N$-bit A-to-D, it is $2^N$, though it is usually expressed in dB.

The voltage range and number of bits determine the size of the **least count** in volts, which may be scaled by the instrument sensitivity to find the least count in the units of whatever is being recorded; we denote this value by $\epsilon$. If (as is usual) the A-to-D rounds the real value to the nearest integer, the maximum error is $\epsilon/2$. Provided the signal moves by more than $\epsilon$ between samples, it is easy to see that this error will be uniformly distributed, provided that where the signal "lands", relative to the integer

boundaries, is reasonably random. The error thus has a uniform distribution from $-\epsilon/2$ to $\epsilon/2$, which has a variance of one-twelfth, equivalent to a standard deviation of about 0.3 least counts.

This result suggests that quantization error will not be large, which in fact is the case: in general, it is not difficult to make quantization error small enough to ignore. You should however always estimate it and compare it to the size of what signal you might wish to record. This is often best done, again, in the frequency domain: though this comparison requires use of the power spectral density – which we will get to.

CHAPTER 5

# DIGITAL FILTERS I: FREQUENCY-SELECTIVE FILTERS

> Engineering is the art of doing that thing well with one dollar, which any bungler can do with two after a fashion.
>
> Arthur M. Wellington, *The Economic Theory of the Location of Railways* (1887).

## 5.1 Introduction: Digital Filters in General

Given a discrete-time sequence $\{x_n\}$ we often wish to perform some kind of operation on it. The DFT is one such operation; many others go by the name of filters. There are many different things that filters may be used to do; and for each of these actions there are many different ways to design filters that are "good". You should remember that filter design is engineering, not science; our aim is to get a good enough, and economical, answer, for the situation we face.

The list of tasks that filters can be used for includes:

A. Select out certain frequencies. This is the most common operation in the kind of analysis done in geophysics: especially removing high frequencies, to allow decimation of a sequence to make it smaller; or low frequencies, to reduce large low-frequency changes that hide smaller fluctuations.

B. Differentiate or integrate, at least approximately; since these are continuous-time operations, they only make sense for sequences properly sampled from a continuous-time function. The design of filters for these operations is quite similar to the design of frequency-selective ones.

C. Simulate what a continuous-time linear system would do to the series, again supposing the sequence to have been sampled from a continuous-time function. This goal leads to another way of designing filters, through simulating a linear system represented by an ordinary differential equation. In the signal processing literature, these methods are used to design frequency-selective filters by simulating filter designs developed for analog systems. For most geophysical data, such simulation filters are more important for reproducing a synthetic instrument through which one wants to pass synthetic data, for comparison with a real dataset recorded by a real instrument.

D. Predict future values, which is an major subfield all by itself: it can be economically very valuable, but it is also used a lot in such matters as autopilots for airplanes, where knowledge of where the plane is about to be is very important. Many methods exist, in part because prediction depends on what model we take to characterize the sequence. Quite a few methods model the sequence probabilistically: for example, the **Kalman filter** operates in the time domain to infer future values from past ones. A rapidly-growing part of this field connects to the burgeoning subject of nonlinear dynamics, since if a time series that appears to be random can actually be described as a deterministic but chaotic system, better predictions may be possible.

E. Detect a signal, defined as some kind of anomalous behavior. This goal is closely related to the previous one: if the series is not as predicted, then there might be an anomaly. Anomaly detection can be described using the methods of statistical hypothesis testing, and has developed a large literature of its own, again because of its technological importance in radar and sonar.

F. Assuming that the sequence has been produced as the result of a convolution of two other sequences $w_n$ and $y_n$, determine something about one or both of these other two. This is known as **deconvolution**, and covers a wide variety of techniques, which depend on what we think we know about the convolving sequence $w_n$. One example would be the reverse of (C): we have the result of a sequence (or a function) being passed through a known linear system, and want to determine the original. Two geophysical examples of this are correcting data for instrument response, and downward continuation of magnetic or gravity data measured above its source (the height acts as a filter). Another type

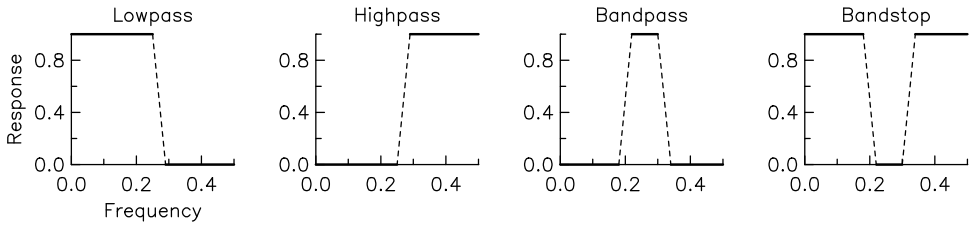



Figure 5.1: Ideal frequency-selective filters: the four commonest types. The parts of the frequency band for which the response is 1 are called the **passband**(s); where the response is 0 is the **stopband**(s). The dashed lines show the region in which (for most design methods) the response goes from 1 to 0 (not necessarily linearly as shown here), which is called the **transition band**. Another name for the bandstop filter is **notch filter**.

of deconvolution (used in exploration geophysics and image processing) occurs when we have only partial knowledge about the two sequences (e.g., some of their statistical properties).

In this chapter we discuss filters for purpose (A): certainly the commonest application in geophysics. In the next chapter we discuss (C), partly so we can introduce some new ways of looking at digital filters. Purpose (B), along with some other filtering topics, will be placed in a third chapter. Purposes (D), (E), and (F), all of which combine filtering and statistics, will not be covered in this course.

## 5.2 FIR Filters for Frequency Selection

Even just for purpose (A), we have many choices in how to design filters. We start by introducing the ideal filter response $W_d(f)$; for frequency-selective filters $W_d$ is either 1 or 0 over particular frequency bands. Figure 5.1 shows some typical designs, and their names.

In this chapter we restrict ourselves to filters that consist of a finite sequence $\{w_n\}$, usually called the **weights**. These filters are used by convolving the weight sequence $\{w_n\}$ with the data sequence $\{x_n\}$ to produce a filtered sequence $\{y_n\} = \{w_n\} * \{x_n\}$. In the signal-processing literature this type of filter is called a **finite impulse response (FIR)** filter; in the statistics literature the usual name is **moving average**, usually abbreviated as MA. Another term is **nonrecursive filter**.

The frequency response of a FIR filter is just the Fourier transform of the weight sequence:

$$W(f) = \sum_{n=0}^{N-1} w_n e^{-2\pi i f n} = \sum_{n=-\infty}^{\infty} w_n e^{-2\pi i f n} \qquad \text{for } 0 \le f \le 1 \quad (\text{or } -\tfrac{1}{2} \le f \le \tfrac{1}{2}) \tag{5.1}$$

where $N$ is the number of weights. It is useful to consider the weights to be an infinite sequence, though one that happens to be zero outside the finite range from 0 to $N-1$.

We next impose three more restrictions:

- The weights are real.
- The number of weights, $N$, is odd.
- The weights are "symmetric": that is, for $N = 2M+1$,

$$w_{M-k} = w_{M+k} \quad k = 0, \ldots M$$

The consequence of all this is that $w_n = w_{2M-n}$, making the frequency response

$$\begin{aligned} W(f) &= \sum_{n=0}^{2M} w_n e^{-2\pi i f n} = \sum_{n=0}^{M-1} w_n (e^{-2\pi i f n} + e^{-2\pi i f (2M-n)}) + w_M e^{-2\pi i f M} \\ &= e^{-2\pi i f M} \left[ w_M + \sum_{n=0}^{M-1} w_n (e^{-2\pi i f (n-M)} + e^{-2\pi i f (M-n)}) \right] \\ &= e^{-2\pi i f M} \left[ w_M + 2 \sum_{n=0}^{M-1} w_n \cos 2\pi (M-n) f \right] \end{aligned} \tag{5.2}$$

The part in square brackets is purely real; like any real quantity viewed using the amplitude and phase form for complex numbers, its phase is zero. The shift theorem shows that the initial exponential is equivalent to an $M$ point shift, which is to say a time delay. Ignoring this, a symmetric FIR filter thus does not shift the phases of different frequencies: this minimizes distortion of the input, other than by removing energy at certain frequencies. If the part in square brackets were one, we would have just an $M$-term delay, which obviously creates no distortion at all. Because the phase shift is linear in frequency, symmetric FIR filters are usually termed **linear phase**.

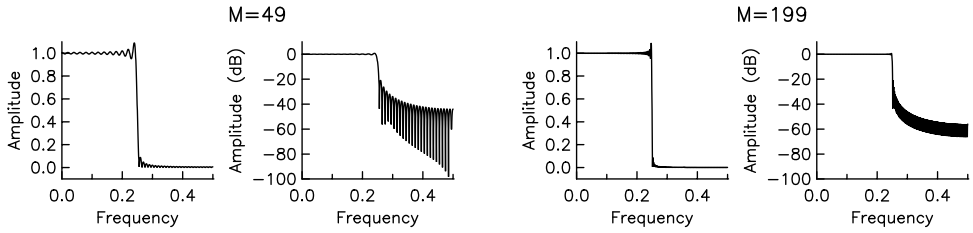



Figure 5.2: Frequency response (shown in both linear and log amplitude) for lowpass filters designed by simple truncation of the sequence of ideal weights. The ideal response $W_d(f)$ has a passband ($W_d = 1$) from 0 to 0.25, and a stopband ($W_d = 0$) from 0.25 to 0.5. The left-hand pair of plots show the response for $M = 49$ (99 weights total), and the right-hand pair for $M = 99$ (199 weights total).

If we do not mind having the filter be acausal (the output for a step input precedes the step), we can make it symmetrical around $n = 0$, defining $w_n = w_{-n}$ for $n = -M, \ldots M$. The frequency response is then

$$W(f) = w_0 + 2\sum_{n=1}^{M} w_n \cos 2\pi n f \tag{5.3}$$

which has zero phase shift; in this form the FIR filter is often called a **zero-phase filter**. Because such a filter is acausal it can be implemented only after the data have been collected, but in digital systems this can always be the case – if only because we are free to relabel the absolute time of the terms in the series as we see fit. In general a lack of causality is not a problem, though seismic data can be an exception to this: if our interest is in the exact time of arrival of bursts of energy, we do not want the filtering to put any energy before it actually arrives.[1] In Chapter 7 we will see how to make FIR filters that are better suited to this application.

Given all the restrictions so far applied, the filter design problem becomes how to choose the $M + 1$ weights $w_0, w_1, \ldots w_{M+1}$, to best approximate a given frequency response $W_d(f)$. As we will see, the meaning of "best" is not unique. Indeed it is characteristic of filter design that the best answer is not always the same: what we want from a filter will depend on what we are trying to do.

A naive approach would be to just take the inverse Fourier transform of the ideal filter response: since $W_d(f)$ is real and symmetric, it might appear

[1] A good example of exactly this problem confounding the interpretation of the initial rupture in earthquakes is given by Scherbaum and Bouin (1997).

that we can use the inverse transform for sequences to get

$$w_n = \int_{-½}^{½} W_d(f) e^{2\pi i f n} df = 2\int_{0}^{½} W_d(f) \cos 2\pi f n \, df$$

However, this will not in general give a sequence of finite length – something we certainly need for a practical filter. We need to be more sophisticated.

### 5.2.1 Designs Using Tapers

One method of filter design is to construct, from the ideal response, the (infinite) sequence of ideal weights

$$w_n^d = 2\int_{0}^{½} W_d(f) \cos 2\pi f n \, df$$

and then create a finite sequence from this by multiplying by a **taper** (or **window**) sequence $\{a_n\}$ to create the final weights

$$w_n = a_n w_n^d$$

The taper sequence must have two properties. The first is that $a_n = 0$ for $n > M$, so that all but a finite section of $\{w_n\}$ is zero, to give a finite-length filter. The second is that the taper must also be symmetric about $n = 0$, to keep the final filter weights symmetric.

We might expect that this time-domain multiplication would be equivalent to a convolution in the frequency domain; and so it is:

$$\begin{aligned} W(f) &= \sum_{n=-\infty}^{\infty} a_n w_n^d e^{-2\pi i f n} = \sum_{n=-\infty}^{\infty} a_n e^{-2\pi i f n} \int_{-½}^{½} W_d(u) e^{2\pi i f n u} \, du \\ &= \int_{-½}^{½} W_d(u) \sum_{n=-\infty}^{\infty} a_n e^{-2\pi i n (f-u)} \, du = \int_{-½}^{½} W_d(u) A(f-u) \, du \end{aligned}$$

where in the final convolution integral both functions are assumed to be periodic outside the range $[-½, ½]$.

This means that the closer $A(f)$, the transform of the taper sequence, is to a delta-function, the closer the filter response will be to the ideal $W_d(f)$ Of course, what we mean by "close to" will, again, depend on what we think is important: in thinking about closeness to a delta-function, is it more

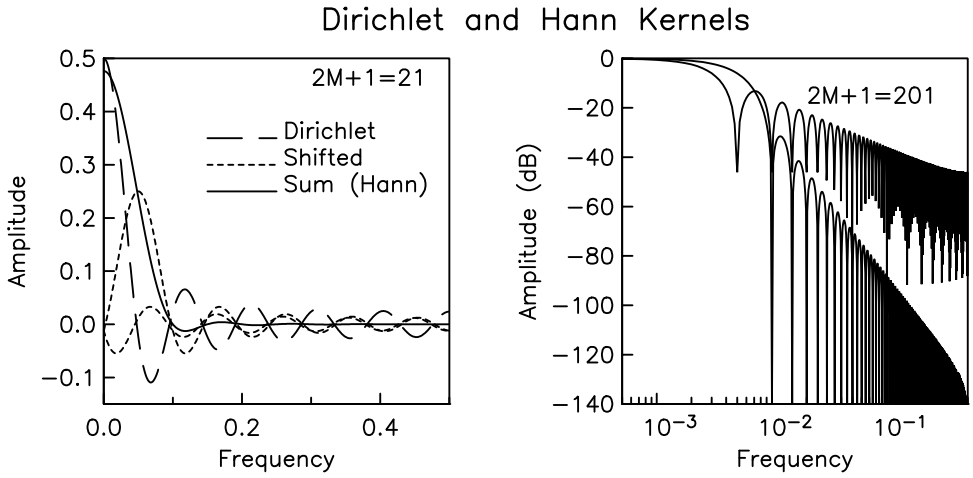



Figure 5.3: The three Dirichlet kernels used in forming the response of the von Hann taper (left, dashed), and that response itself (solid). The right plot shows $A(f)$ for the Dirichlet and von Hann kernels on a log-log scale, for a larger value of $M$.

important that the peak of $A(f)$ be narrow or the values away from the peak be small? Our answer to this will depend on what we want the filter to do, which is why there are no universal rules for filter designs.

The simplest taper is the rectangular taper, with $a_n = 1$ for $|n| \le M$; this corresponds to simply truncating the sequence of ideal weights. We can see that this is not a particularly good solution if we look at the corresponding $A(f)$:

$$A(f) = \sum_{n=-M}^{M} e^{-2\pi i f n} = (2M+1) D_{2M+1}(f)$$

where $D_N(f)$ is the Dirichlet kernel discussed in Chapter 3. As described there, successive maxima of $|D_N(f)|$ do decrease, but only as $f^{-1}$: if what we want is small values away from the peak, this $A(f)$ is not a good approximation to a delta-function. Convolving this $A(f)$ with the ideal response causes much of the passband response to "leak" into the stopband, and vice-versa. Figure 5.2 shows this effect clearly, and also shows that simply increasing the number of weights $M$ does not diminish its importance.

If you are familiar with Fourier series theory, another way to view this result is to realize that in taking the rectangular taper we are simply finding partial sums of the Fourier series expression for $W_d(f)$; such partial sums always suffer from Gibbs' phenomenon, with poor convergence near

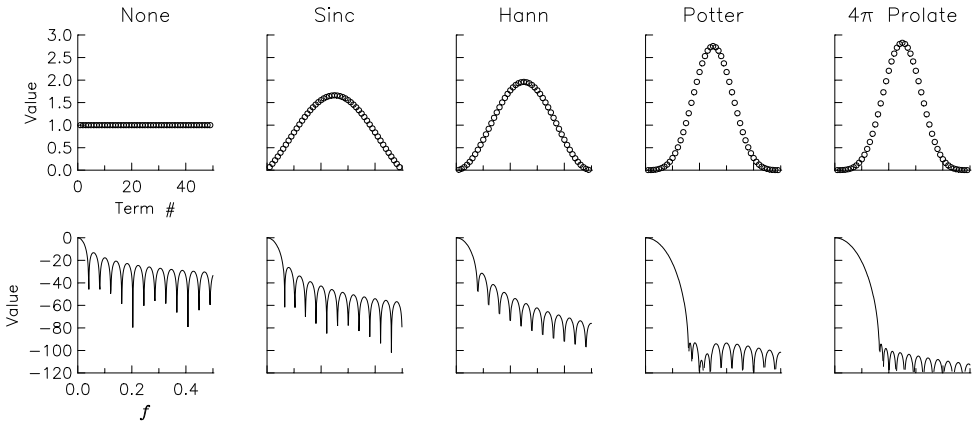



Figure 5.4: A collection of data tapers (top plots) and their Fourier transforms $A(f)$, for $2M+1 = 52$ (the transforms have been interpolated for plotting, and normalized to 1 at zero frequency). The sinc taper is given by $(\pi n/M)^{-1}\sin(\pi n/M)$, and the Potter taper by $\sum_{k=0}^{3} a_k \cos(\pi k n/(2M+1))$ with $a_0 = 0.3557702$, $a_1 = 0.4873966$, $a_2 = 0.1442299$, and $a_3 = 0.0126033$.

the discontinuities in the frequency response.[2]

There is a taper that is almost as simple, but that has much smaller values away from the central peak: this is the von Hann taper, also called the hanning, $\cos^2$, or $1+\cos$ taper:

$$a_n = \tfrac{1}{2}[1+\cos(\pi n/M)] = \cos^2(\pi n/2M) \qquad \text{for } n = -M, \ldots M$$

which has the transform:

$$\begin{aligned} A(f) &= \sum_{n=-M}^{M} a_n e^{-2\pi i f n} = \tfrac{1}{2}\sum_{n=-M}^{M} e^{-2\pi i f n} + \tfrac{1}{4}\sum_{n=-M}^{M} [e^{\pi i n/M} + e^{-\pi i n/M}] e^{-2\pi i f n} \\ &= \tfrac{1}{2}\sum_{n=-M}^{M} e^{-2\pi i f n} + \tfrac{1}{4}\sum_{n=-M}^{M} e^{-2\pi i (f-(M/2))n} + \tfrac{1}{4}\sum_{n=-M}^{M} e^{-2\pi i (f+(M/2))n} \\ &= (2M+1)[\tfrac{1}{2}D_{2M+1}(f) + \tfrac{1}{4}D_{2M+1}(f-(M/2)) + \tfrac{1}{4}D_{2M+1}(f+(M/2))] \end{aligned}$$

The three Dirichlet kernels and their sum are shown in Figure 5.3: it is clear that, as $f$ increases, the combination approaches zero much more rapidly than does the original Dirichlet kernel, a point made even more dramatically by a log-log plot.

[2] For a good introduction, both historical and mathematical, to Gibbs' phenomenon, see Hewitt and Hewitt (1979).

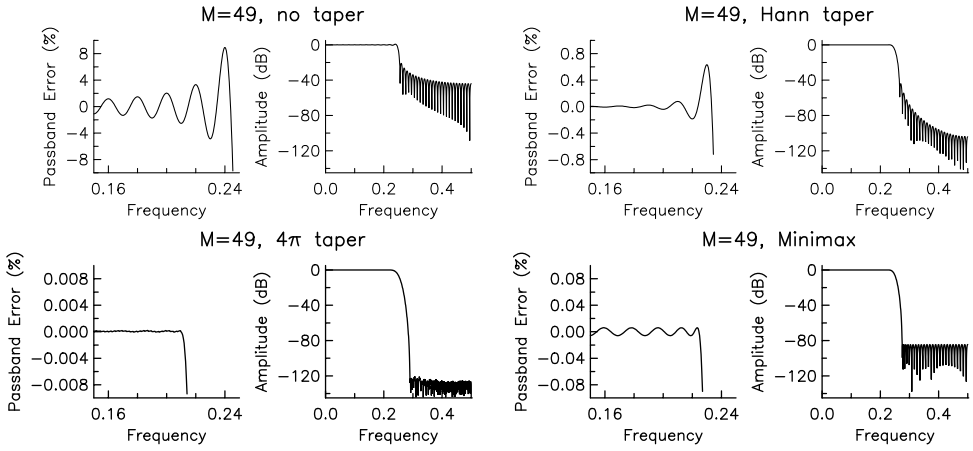



Figure 5.5: Frequency responses of lowpass filters for three data tapers, and also for the minimax design procedure described in Section 5.2.2 For each, the error (departure from unity) is shown over part of the passband (note the different scales), and the complete response is shown in log form.

There are many taper sequences available;[3] the frequency responses of these can vary considerably, but inevitably have a tradeoff between width of the central peak and smallness of the fluctuations (these are called the **sidelobes**) away from it. Figure 5.4 gives a small selection of tapers, including two with especially small sidelobes: the Potter taper (an empirically-designed one) and the $4\pi$ prolate spheroidal taper. The last, as we will see in Chapter **??**, is the function for which $A(f)$ is most concentrated within a frequency band around zero, and is one of several that are important when estimating a power spectrum.

The effect of using different tapers for filters is shown in Figure 5.5. As expected, the use of a Hann taper gives much lower sidelobes than does a rectangular taper; the $4\pi$ prolate spheroidal taper gives even lower ones, with sidelobe amplitudes due in part to roundoff error in computing the DFT. But filter design using tapers is somewhat inflexible: given a particular taper the tradeoff between sidelobe level and the width of the transition between passband and stopband is fixed. A somewhat more flexible ap-

[3] Harris (1976) is a kind of bestiary of every taper then thought of, with plots of the time and frequency responses. Most of these tapers deserve to be forgotten, since much better ones, notably the prolate spheroidal tapers, have been developed since. It is worth noting that the 'Kaiser-Bessel' tapers described in this paper closely approximate the prolates.

proach has been developed by Kaiser and Reed (1977) and Kaiser and Reed (1978), using an adjustable taper and some empirical rules for setting it (and the filter length) to meet a given sidelobe level and transition width; this method offers a convenient design that is adequate for many applications. However, even more flexibility is possible using different techniques, one of which we now describe.

### 5.2.2 Design by Optimal Fitting of the Frequency Response

A more powerful technique in designing filters is to look at the difference between the ideal response $W_d(f)$ and the actual response $W(f)$, and choose the filter weights to minimize this difference in some way. What seems like the most obvious approach does not however work very well, this being to minimize the mean-square misfit. Minimizing this is just like the least-squares criterion, though in this case expressed as an integral over frequency: the quantity to minimize is

$$\epsilon^2 = \int_{-1/2}^{1/2} |W_d(f) - W(f)|^2 df$$

The difference between the ideal and actual responses, in terms of the weights, is

$$W_d(f) - W(f) = \sum_{n=-\infty}^{\infty} (w_n^d - w_n) e^{-2\pi i f n}$$

where the actual weights $w_n$ are extended to infinity by adding zeroes at both ends. But then we may apply Parseval's theorem for infinite sequences to get:

$$\epsilon^2 = \sum_{n=-\infty}^{\infty} |w_n^d - w_n|^2 = \sum_{n=-\infty}^{-M-1} |w_n^d|^2 + \sum_{n=M+1}^{\infty} |w_n^d|^2 + \sum_{n=-M}^{M} |w_n^d - w_n|^2$$

and since only the last term is adjustable, we minimize $\epsilon^2$ by setting $w_n = w_n^d$. But this is no different from using a rectangular taper on the ideal sequence – and we have already seen that this produces filters with large sidelobes. Least squares, for all its merits, is not a panacea.

What is more useful is to minimize the maximum value that the misfit $|W_d(f) - W(f)|$ attains over a range of frequencies. This is computationally more complicated than anything we have discussed so far, and we will not

describe how it is done; suffice it to say that it is possible to find weights that will do this. Specifically, given $K$ non-overlapping frequency bands $f_{L_k} < f \le f_{H_k}$, for $k = 1, 2 \ldots K$, we can find the $M$ weights which minimize

$$b_k \max_{[f_L, f_H]} |W_d(f) - W(f)|$$

over all the bands; $b_k$ is a weight applied to each band, which allows the fit to the ideal response to be tighter or looser. The actual amount of misfit is largely a function of the filter length $M$; what this misfit actually is, can only be determined after the optimal weights are found. Usually some trial and error is needed to decide on the appropriate tradeoff between length and misfit. This approach applies the $L_\infty$ norm, whereas the mean-square-error criterion applies the $L_2$ norm; and in general gives very good designs, with more flexibility than the tapering method. The resulting filters are termed **equiripple** filters, because the frequency response shows a uniform level of departure from the ideal.

A filter designed using this method is shown in the fourth example in Figure 5.5: by sacrificing low sidelobes at high frequency, the equiripple design can have a narrower transition band than any of the tapered designs. The procedure for designing equiripple filters is called the Parks-McClellan algorithm; it is included in and in MATLAB as routine `firpm`. The underlying algorithm is called the Remez exchange method.

There are many other variants in this method of filter design; we may, for example, require, as well as a good fit to $W_d$, that the derivative $dW/df$ be of one sign in the passband: this will completely eliminates ripple there.[4] In general, most such refinements, and how to design filters where the word length is short and roundoff a problem, are rarely important in geophysical data analysis.

### 5.2.3 A Filtering Example

We close with an example that illustrates how frequency-selective filters can be used, and also illustrates why there are no fixed rules for designing them – what filter you choose depends on the problem you want to solve. The data to be filtered are measurements of strain at Piñon Flat Observatory, made at one-second intervals with a long-base laser strainmeter. For

[4] Steiglitz *et al.* (1992) present a fairly general program for including different kinds of constraints using linear programming: relatively slow, but very flexible. See `http://www.cs.princeton.edu/~ken/meteor.html` for source code and examples.

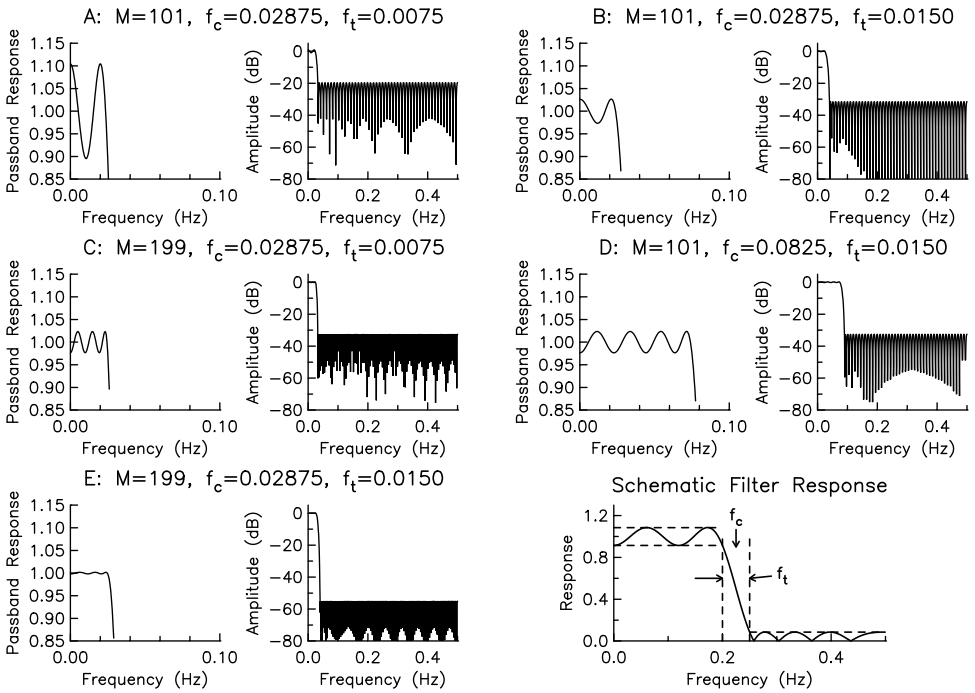



Figure 5.6: Equiripple filter designs. The lower right shows a schematic response, with the frequency parameters indicated. The different designs (A through E), have different frequency settings or lengths $M$. For each, the passband response is shown on a linear scale, the overall response in decibels.

the time period shown, these strain data show the signal from the 1994 Northridge earthquake, 203 km away. This signal, though dominated by the seismic waves from the earthquake, also contains the static change in strain caused by the elastic rebound of the rocks, which is what drives the faulting. The aim of the filtering will be to remove the "high-frequency" seismic signal so as to show the static change more clearly.

For maximum flexibility we use the equiripple design; Figure 5.6 shows some possible filter responses. The lower right corner of this figure shows parameters which we can choose: the transition frequency $f_c$ and the width of the transition band $f_t$. Once we choose these parameters, we next pick a length for the filter; the program will then find the best design possible, with the smallest amount of ripple in both the passband and stopband. In this case we choose these amounts to be the same; while they can be in any proportion to each other, making one smaller makes the other larger.

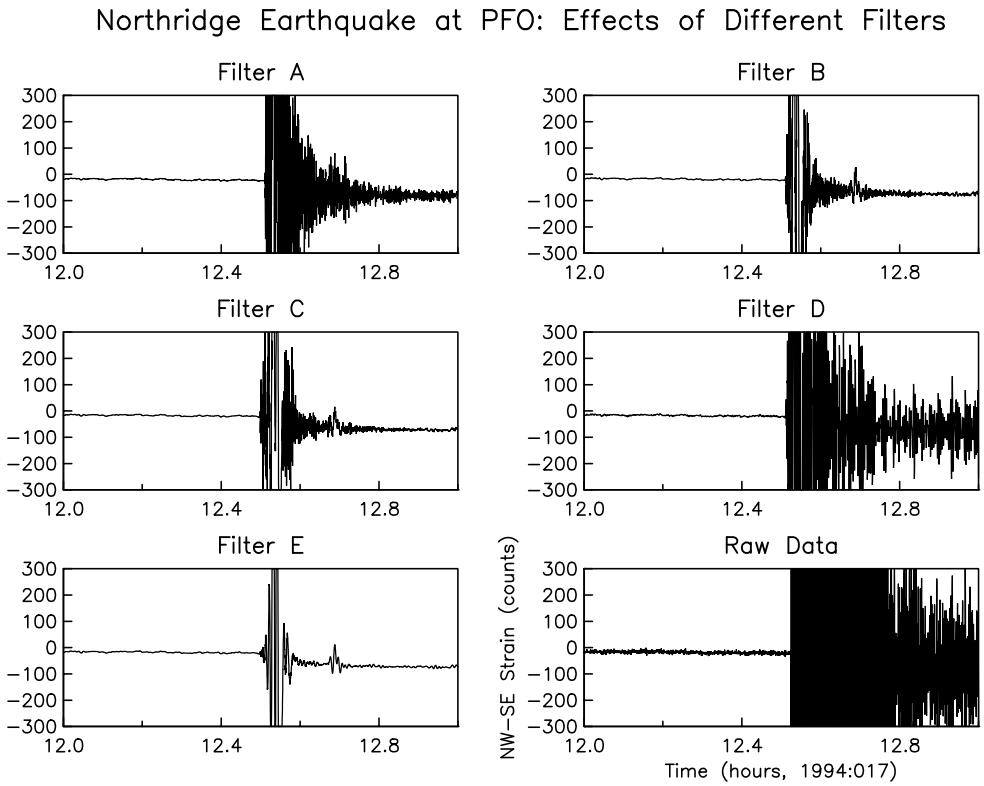



Figure 5.7: Strain variations from the Northridge earthquake recorded on the NW-SE laser strainmeter at Piñon Flat Observatory (PFO). The lower right plot is the original unfiltered data; this actually has a much larger range than is shown. Plots A through E show the results of lowpassing the data with the filters whose response is shown in Figure 5.6.

Our first design (A) has a narrow passband, and a narrow transition band – and the consequence is that the ripple is large, nearly 10%. This means that in the stopband any signal will be only 0.1 of what it was originally – we would say, 20 dB down, which is not a large reduction. To do better, we can try two approaches: (B) make the transition band wider, and (C) use more filter weights (199 instead of 101). Either one gives much less ripple; for these two examples we are trading computation time (filter length) against the width of the passband response. In (D) we try making the passband much larger by increasing $f_c$; in (E) we combine the larger $f_t$ of (B) with the more numerous weights of (C) to get a response even closer to the ideal.

Figure 5.7 shows what these filters do to the data; without filtering the static offset is well hidden in the seismic coda. Filter A reduces this enough that the offset can be seen, but still leaves a lot of energy: looking at this, you would (or should) want to say that the "true" signal would be a smoothed version of this: a sure sign that the data need more filtering. Filters B and C provide more filtering: since their stopband levels are the same, the results look about the same as well; between the two, B is probably preferable because it is shorter. A much wider passband (D) turns out to be a poor idea, at least for showing the static offset. Filter E is perhaps "best" for this application.

CHAPTER 6

# DIGITAL FILTERS II: RECURSIVE FILTERS AND SIMULATING LINEAR SYSTEMS

## 6.1 Introduction

We now turn to another kind of digital filter: one that will allow us to use a computer to imitate what some physical system does. We might need this when, for example, we want to model a seismogram. The first step would be to have a way of computing the the ground motion input to the seismometer. The next step would be to simulate what the output of the seismometer (a physical system) is for this ground motion. The FIR filters of the previous chapter are not well-suited for this, but other designs are, and it is these we will now describe.

But, to discuss these filters and how to design them we need to first spend some time introducing additional mathematics for the linear systems we discussed in Chapter 2. We will then return to the problem of designing digital filters, introducing what is known as a **recursive filter**; finally, we will show how to make such a digital filter accurately simulate an analog system.

## 6.2 Lumped-Parameter Systems

We saw in Chapter 2 that a linear time-invariant system could be characterized in three different ways:

- By its frequency response $\tilde{g}(f)$; this expresses the ratio (a complex number) between output and input when the input is a pure sinusoid, $e^{2\pi i f t}$.

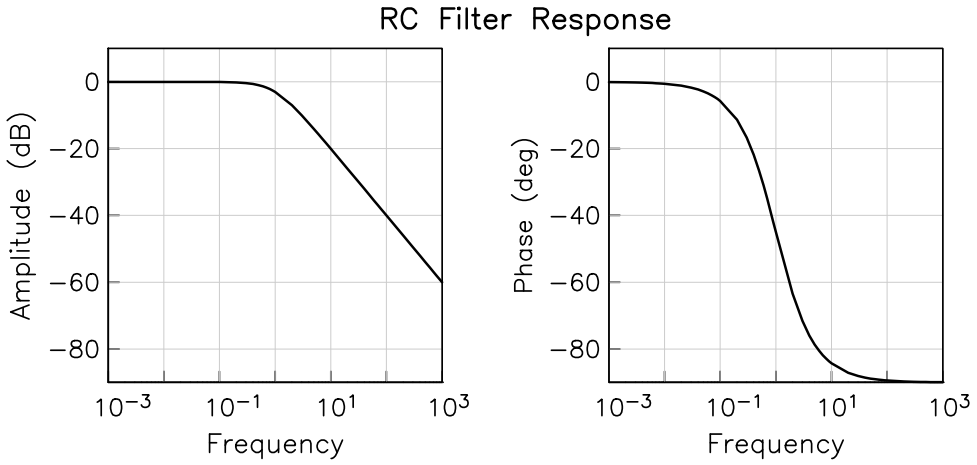



Figure 6.1: Frequency response of an RC filter with $\tau = 1$, shown as log amplitude, and phase, plotted against log frequency.

.

- By its impulse response $g(t)$: the time-domain behavior when the input is a delta-function. This is also the inverse Fourier transform of $\tilde{g}(f)$: $gfo(f) = \mathcal{F}[g(t)]$.

- By the step response $h(t)$, which is the integral of $g(t)$.

We now look at these characterizations for a special class of linear time-invariant systems – though one that includes many actual cases. This class is systems that are described by constant-coefficient linear differential equations; a common term, especially in electrical engineering, for these is **lumped-parameter systems**. For such systems the relationship between the input $x$ and output $y$ is

$$\begin{aligned} a_n \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \ldots + a_1 \frac{dy}{dt} + a_0 y = \\ b_m \frac{d^m x}{dt^m} + b_{m-1} \frac{d^{m-1} x}{dt^{m-1}} + \ldots + b_1 \frac{dx}{dt} + b_0 x \end{aligned} \tag{6.1}$$

The $a$'s and $b$'s are the parameters that describe the system.

There are standard procedures for solving such an equation to get explicit forms for $y(t)$, especially for certain classes of inputs $x(t)$; we shall instead look at the frequency response of such systems. But first we provide a few examples.

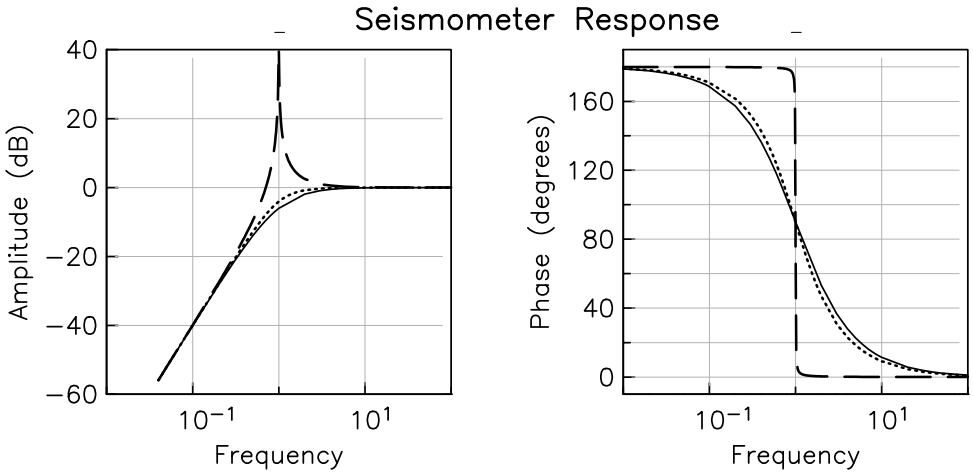
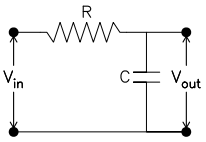



Figure 6.2: Frequency response of a seismometer to displacement. The natural frequency is $\omega_0 = 1$. The response is shown for $\lambda = 0$ (dashed), $\lambda = 0.8$ (dotted), and $\lambda = 1$ (solid).




- The first example is the simplest analog electronic filter: the RC lowpass filter, which consists of a resistor and capacitor in series, with input and output voltages as shown. The input voltage $x(t) = V_{in}$ is just the output voltage $y(t) = V_{out}$ plus the voltage drop across the resistor, which is given by $RI(t)$, where $I(t)$ is the current flowing through the capacitor. This current is given by $I(t) = C\dot{y}(t)$, making the differential equation

$$y(t) + RC\frac{dy}{dt}(t) = x(t) \tag{6.2}$$

To get the frequency response, we assume, as usual, a sinusoidal input $x(t) = e^{2\pi i f t}$; by definition of $\tilde{g}$, $y(t) = \tilde{g}(f)e^{2\pi i f t}$. Substituting these expressions into (6.2), we get

$$\tilde{g}(f) = \frac{1}{1 + 2\pi i f \tau} \tag{6.3}$$

where $\tau = RC$ is the time constant of the filter. The two plots in Figure 6.1 show the amplitude and phase of $\tilde{g}(f)$ for $\tau$ equal to 1, plotted against the logarithm of the frequency (and with the amplitude plotted in dB); these are called **Bode plots**. Clearly this is a lowpass filter: high frequencies are attenuated.

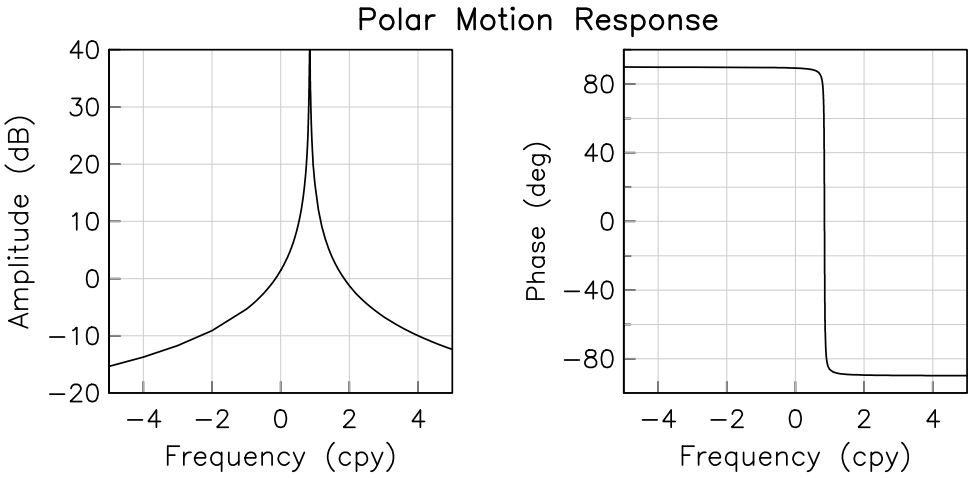



Figure 6.3: Response of the Earth's polar motion to excitation; note that since we represent the input and output (both 2-vectors) as complex numbers. we need to show both positive and negative frequency. The response is given for a $Q$ of 100; the Chandler resonance is at 0.849 cycles/year. The rapid change of phase at this frequency makes it difficult to compare polar motion with excitations.

- A seismometer (Figure 6.2). The simplest form (a mass on a spring, or a pendulum) obeys the well-known equation for a simple harmonic oscillator:

$$\ddot{y} + 2\lambda\omega_0\dot{y} + \omega_0^2 y = \ddot{x}$$

  where $y$ is the displacement of the mass with respect to the frame, and $x$ the displacement of the frame with respect to inertial space. The system parameters are the natural frequency $\omega_0$, and the damping $\lambda$. Again, we substitute $e^{2\pi i f t}$ for $x(t)$ and $\tilde{g}(f)e^{2\pi i f t}$ for $y(t)$, and find

$$\tilde{g}(f) = \frac{-4\pi^2 f^2}{-4\pi^2 f^2 + 4i\pi\lambda\omega_0 f + \omega_0^2}$$

  whose phase and amplitude response are shown in Bode plots (Figure 6.2) for several values of $\lambda$: $\lambda = 0$ is undamped, $\lambda = 0.8$ gives the flattest response, and $\lambda = 1$ is critically damped.

- The Earth's polar motion. The position of the rotation pole of the solid earth changes because of variations in the angular momentum and mass distribution of the fluid parts. The pole position can be described by two coordinates, $p_1$ and $p_2$, giving the (dimensionless)

angular displacement of (say) the North rotation pole in directions toward the Greenwich meridian and at 90° to it. If these changes are small, it can be shown that the relationship between the pole position and the motions of the fluid parts is

$$\frac{1}{\omega_c}\frac{dp_1(t)}{dt} + p_2(t) = \psi_2(t) \qquad \frac{1}{\omega_c}\frac{dp_2(t)}{dt} + p_1(t) = \psi_1(t) \tag{6.4}$$

where $\psi_1$ and $\psi_2$ are integrals over the mass distribution and velocity of the fluid parts of the earth (usually termed the "excitation").[1] The frequency $\omega_c$ is determined by the properties of the solid earth; for a rigid ellipsoidal body $\omega_c = [(C-A)/A]\Omega$, where $C$ and $A$ are the polar and equatorial moments of inertia, and $\Omega$ is the frequency of the Earth's spin (once per day). The values of $C$ and $A$ for the Earth give a value for $\omega_c$ that corresponds to a period of 305 days; various corrections, too complicated to discuss here, make the actual period equal to 430 days. We can write this pair of equations as a single equation by forming the complex variables $\mathbf{p} = p_1 + ip_2$ and $\psi = \psi_1 + i\psi_2$; then we can combine the two equations in (6.4) to get:

$$\frac{i}{\omega_c}\dot{\mathbf{p}}(\mathbf{t}) + \mathbf{p}(\mathbf{t}) = \psi(\mathbf{t}) \tag{6.5}$$

If we make the input $eft$, and the output $\tilde{g}eft$, we find

$$\tilde{g}(f) = \frac{1}{1-(2\pi f/\omega_c)}$$

which has the response shown in Figure 6.3, in this case plotted against both positive and negative frequency, since the response is different for these: a common attribute of gyroscopic systems. (Remember that, with this trial function, negative frequency corresponds to clockwise [deasil] rotation, and positive to counterclockwise [widdershins].)

We will use these examples in the discussion below, both to illustrate concepts about systems, and also as examples for filter design.

[1] For a derivation of this equation, see Munk and McDonald (1960); the version given here has been revised to match the newer and more accurate result of Gross (1992).

## 6.3 The Laplace Transform: System Poles and Zeros

Yet another way of looking at the response of a system comes if we look at another integral transform: the **Laplace transform**. The Laplace transform of a function $x(t)$ is defined as

$$\tilde{x}(s) = \int_{-\infty}^{\infty} x(t) e^{-st}\, dt$$

for $s$ a complex number: $\tilde{x}(s)$ is defined on the complex plane. The Fourier transform is $\tilde{x}(f)$ is just $\tilde{x}(s)$ evaluated along the imaginary axis of the complex plane; that is, $\tilde{x}(f) = \tilde{x}(s)$ for $s = 2\pi i f$. The Laplace transform can thus be thought of as a generalization of the Fourier transform; but, viewed as a transform, it is nowhere near as useful. And, mostly because $\tilde{x}(s)$ is a complex-valued function over the complex plane, the Laplace transform is much more difficult to visualize than the Fourier transform. The inverse Laplace transform is

$$x(t) = \frac{1}{2pi} \int_{c-i\infty}^{c+i\infty} \tilde{x}(s) e^{st}\, ds$$

where the integral is evaluated over a line parallel to the imaginary axis, the value of the real part depending on the nature of $\tilde{x}(s)$. The Laplace transform and its inverse are thus much less symmetrical than the Fourier transform pair.

The utility of the Laplace transform for us comes through applying it to the lumped-parameter systems described by equation (6.1). The value of the Laplace transform is that it enables us to get significant understanding of such a system without solving the differential equation at all.[2]

Taking the Fourier transform of both sides of equation (6.1) gives us the same result as we would get (and did get for the examples of the previous section) by substituting $e^{2\pi i f t}$ as the input. What we mean by "taking the Fourier transform" of the system is to find the equation that describes the connection between the Fourier transforms of the input and output: that is, between $\tilde{x}(f)$ and $\tilde{y}(f)$. We get this result by applying the theorem for the Fourier transform of the derivative of a function to equation (6.1); this gives us a polynomial in $f$:

$$[a_n(2\pi i f)^n + a_n - 1(2\pi i f)^n - 1 + \ldots + a_1(2\pi i f) + a_0]\tilde{y} =$$
$$[b_m(2\pi i f)^m + b_m - 1(2\pi i f)^m - 1 + \ldots + b_1(2\pi i f) + \ + b_0]\tilde{x}$$

[2] Deakin (1992) describes how this approach grew to dominance.

where $\tilde{y}$ and $\tilde{x}$ are the Fourier transforms of $y$ and $x$. Therefore, the frequency response of the system is

$$\tilde{g}(f) = \frac{\tilde{y}(f)}{\tilde{x}(f)} = \frac{\sum_{j=0}^{m} b_j (2\pi i f)^j}{\sum_{k=0}^{n} a_k (2\pi i f)^k} \tag{6.6}$$

We can follow a similar route if we take the Laplace transform of both sides of (6.1). While this might seem like an unnecessary generalization (who needs the system response at a complex frequency?) it actually leads to some very useful insights into how the system will behave. We need the derivative theorem for the Laplace transform, with we simply state without proof: if $\tilde{x}(s)$ is the Laplace transform of $x(t)$, the Laplace transform of $\dot{x}$ is $s\tilde{x}(s)$.

Applying this rule to the differential equation shows that the ratio of the Laplace transforms of the input and output, which is called the **transfer function**, is

$$\frac{\tilde{y}(s)}{\tilde{x}(s)} = \frac{\sum_{j=0}^{m} b_j s^j}{\sum_{k=0}^{n} a_k s^k} \tag{6.7}$$

Comparing this with equation (6.6), we see that the frequency response is a special case of the transfer function: the frequency response $\tilde{g}(f)$ is just the transfer function $\tilde{g}(s)$ evaluated on the imaginary axis.

The additional insight to be gotten from looking at the transfer function comes if, instead of expressing the polynomials in (6.7) in terms of their coefficients, we instead write them as products of their roots:

$$\frac{\tilde{y}(s)}{\tilde{x}(s)} = C \frac{\prod_{j=1}^{m} (s - r_j)}{\prod_{k=1}^{n} (s - p_k)}$$

The roots of these polynomials in $s$ come in two forms, the **zeros** $r_j$ of the numerator, and the **poles** $p_k$ of the denominator. At the zeros the transfer function is zero; and at the poles it is infinite. The scaling value, $C$, is needed to make the description complete. Looking at the locations of the poles and zeros, and particularly how close they are to the imaginary axis, will show where the frequency response will be large and where it will be small. Finding the locations for either set of roots has to be done numerically if $m$ or $n$ exceeds five, but our three examples can all be done analytically:

- The RC filter has a transfer function

$$\tilde{g}(s) = (1 + \tau s)^{-1} \tag{6.8}$$

which has a single pole on the negative real axis at $s = \tau^{-1}$. This is shown in the left-hand plot of Figure 6.4; as is conventional, we use a cross to show the location of a pole.

- The seismometer transfer function is

$$\tilde{g}(s) = \frac{s^2}{s^2 + \lambda\omega_0 s + \omega_0^2}$$

which has a pair of zeros at the origin of the $s$-plane. The poles lie at $\omega_0[-1 \pm \sqrt{\lambda^2 - 1}]$. For $\lambda = 0$, the poles are on the imaginary axis, at $\pm i\omega_0$. Since this axis corresponds to the frequency axis for the Fourier transform, these on-axis poles make $\tilde{g}(\pm\omega_0$ infinite. As $\lambda$ increases from zero, the two poles leave the imaginary $s$-axis and follow a circular path about the origin: because the coefficients of the polynomial are real, its roots (these poles) must be complex conjugates. The case of $\lambda = 0.8$, which gives the flattest response, would be termed a two-pole Butterworth filter. At $\lambda = 1$ the poles meet; for larger values they lie on the negative real axis, one approaching and the other receding from the origin.

- The transfer function for polar motion is

$$\tilde{g}(s) = \frac{\mathbf{P(s)}}{\Psi(\mathbf{s})} = \frac{\frac{1}{1+is}}{\omega_c} = \frac{\omega_c}{\omega_c + is}$$

which has a single pole, on the imaginary axis, at $s = i\omega_c$. Again, this gives an infinite response for $f = \omega_c$: in the Earth this resonance gives rise to the Chandler wobble, hence the subscript $c$ on $\omega$. In reality dissipation keeps the response from being infinite; we can add this effect to the equations by moving the pole of the transfer function away from the imaginary axis. We make this pole complex, rather than imaginary, by defining $\omega_c$ as

$$\omega_c = \frac{2\pi}{T_c}\left(1 + \frac{i}{2\pi Q_c}\right)$$

where $T_c$ is now the period of the resonance, and $Q_c$ a measure of the amount of dissipation.[3]

[3] It is certain that dissipation occurs in the Chandler wobble, but what causes it remains unclear. Smith and Dahlen (1981) provide an exhaustive discussion of the theoretical values for these two parameters for a given Earth model.

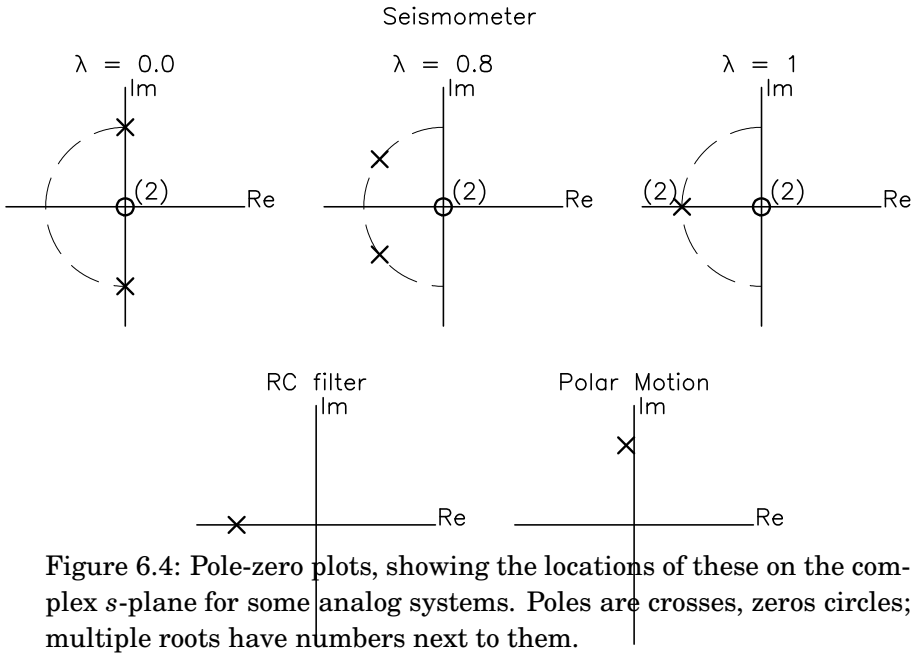



Figure 6.4: Pole-zero plots, showing the locations of these on the complex $s$-plane for some analog systems. Poles are crosses, zeros circles; multiple roots have numbers next to them.

For all our examples, including the last one with finite $Q_c$, the poles are left of the imaginary axis. This is good, because for any lumped-parameter system, stability requires that all the poles of the transfer function have negative real parts, lying in the left half-plane. And for an unstable system, any nonzero input leads to an infinite output. Whether or not a system is stable is not obvious from looking at the differential equation; but finding the poles shows this immediately. Likewise, looking at the poles we can easily see what at what frequencies the response is large: it will be those that are near poles that are close to the imaginary axis. Conversely, zeros close to the imaginary axis will produce dips in the amplitude response.

## 6.4 The *z*-transform for Digital Filters

In Chapter 3 we examined the Fourier transform for sequences. We have just seen that the Laplace transform can be regarded as the generalization of the Fourier transform; the equivalent generalization for sequences is called the **z-transform**. An infinite sequence $x_n$, has a $z$-transform $\zeta[x_n]$

that is a function of the complex variable $z$:

$$\zeta[x(n)] = \tilde{x}(z) = \sum_{n=-\infty}^{\infty} x_n z^{-n}$$

It is important to note that this is *not* a definition that is universally followed. In particular, the geophysical exploration industry (which does a lot of signal processing) defines the $z$-transform as a sum over $x_n z^n$, Our usage is that found in electrical engineering; if we call our $z$-variable $z_{\mathrm{EE}}$, and the exploration one $z_{\mathrm{oil}}$, $z_{\mathrm{EE}} = z_{\mathrm{oil}}^{-1}$. Since $z$ is a complex variable, this usage is equivalent to an inversion of the complex plane on the unit circle: the outside of the circle becomes the inside, and vice-versa, with the origin and infinity mapping into each other. Either convention is consistent, but in looking at results (and software) you need to know which one is being used.

We have seen that the Fourier transform can be viewed as the Laplace transform evaluated on the imaginary axis of the $s$-plane. In just the same way, the Fourier transform of a sequence is a special case of the $z$-transform. Remember that the Fourier transform of a sequence is

$$\tilde{x}(f) = \sum_{n=-\infty}^{\infty} x_n e^{-2\pi i f n} \tag{6.9}$$

This is equivalent to the $z$-transform for $z = e^{2\pi i f}$; in words,

> The Fourier transform of a sequence is the $z$-transform evaluated on the unit circle in the complex $z$-plane.

There are other ways of looking at the $z$-transform. We can, for example, regard $z^{-1}$ as a unit delay. This may sound peculiar, but is easily derived. Suppose $y$ is the same series as $x$, but delayed by $m$ terms: $y_n = x_{n-m}$; then the $z$-transform of $y$ is just

$$\tilde{y}(z) = \sum_{n=-\infty}^{\infty} x_{n-m} z^{-n} = \sum_{l=-\infty}^{\infty} x_l z^{-(l+m)} = \tilde{x}(z) z^{-m} = \tilde{x}(z)(z^{-1})^m \tag{6.10}$$

so that the effect of delaying $x$ is to multiply the $z$-transform by $z^{-1}$ a total of $m$ times: $z^{-1}$ "represents" a delay. (That is, $z_{\mathrm{EE}}^{-1}$ does; in the usage of the exploration industry, $z_{\mathrm{oil}}$ does.)

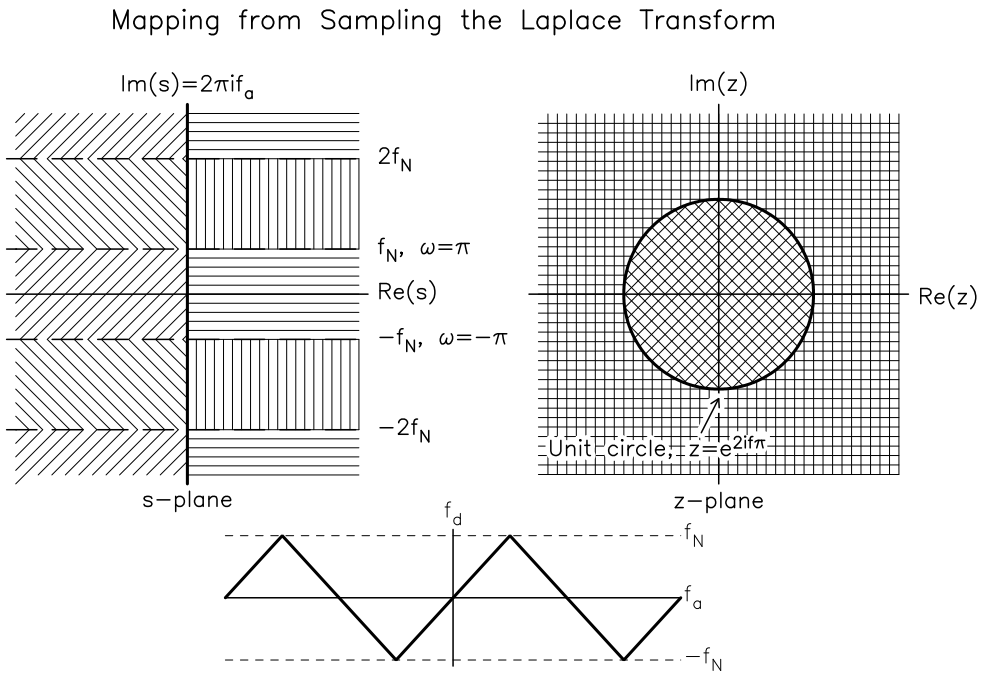



Figure 6.5: Mapping between the $s$-plane and the $z$-plane if we get our sequence in discrete time by sampling in continuous time. The right-hand side of the $s$-plane maps outside the unit circle in the $z$-plane, and the left-hand side to the inside of the unit circle, but in both cases the mapping is not one-to-one. The bottom plot shows how the continuous-time frequency $f_a$ maps into discrete-time frequency $f_d$.

Another way to look at the $z$-transform comes from taking the Laplace transform of the function which is equivalent to the sequence $x_n$, namely $\sum_{n=-\infty}^{\infty} x(n)\delta(t-n)$, which we used in Chapter 4 to discuss sampling of a function given in continuous time. The Laplace transform of this infinite sum of delta-functions is

$$\int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x_n \delta(t-n) e^{-st}\, dt = \sum_{n=-\infty}^{\infty} x_n \int_{-\infty}^{\infty} e^{-st}\delta(t-n)\, dt = \sum_{n=-\infty}^{\infty} x_n e^{-sn}$$

which becomes equivalent to the $z$-transform if we set

$$z = e^s \tag{6.11}$$

which for any $s$ gives a value $z$. This derivation of the $z$-transform thus introduces a **mapping** from the $s$-plane to the $z$-plane; as we will see below,

such mappings are important in devising digital (discrete-time) models of continuous-time systems. The particular mapping given by equation (6.11) has an important feature, namely that it is nonunique: different points in the $s$-plane all map to the same point in the $z$-plane.

If we consider the strip for which $|\Im(s)| \le \pi$, we can see that the region of the strip with negative real part maps inside the unit circle, the region with positive real part maps to outside the unit circle, and the segment of the imaginary axis that lies in the strip maps onto the unit circle.[4] But exactly the same mapping occurs for each strip defined by $m - \pi \le \Im(s) \le m + \pi$, for any integer $m$. This is yet another example of aliasing: when we form a sequence by sampling a function, one frequency in the sequence can come from many different frequencies in the function; with the mapping (6.11) we have generalized this to complex frequencies.

It is possible to use complex-variable theory to find the inverse of the $z$-transform, and also to derive extensions to the convolution theorems in order to show, for example, how to convolve two $z$-transforms to get the $z$-transform of the product of two sequences. Since these results are not needed to design filters for system simulation, we do not discuss them further; see Oppenheim and Schafer (1989) for the details.

## 6.5 Recursive Digital Filters

In a FIR filter, the output is just a weighted combination of input values. This kind of filter is inadequate for simulating an analog system, for which purpose we introduce a more general form: the **recursive filter**, where the output value depends on previous values of the output, as well as on the input values. The equation for such a digital filter is

$$\sum_{k=0}^{N-1} a_k y_{n-k} = \sum_{l=0}^{L-1} b_l x_{n-l} \tag{6.12}$$

where $x_n$ is the input sequence and $y_n$ is the output. Because the sums are one-sided, we can compute the "current" value of $y$, $y_n$, without needing any values of $y$ except for the "past" ones that we have already computed. Using only past values of $x$ is not necessary except in real-time processing; but is conventional and not restrictive in practice.

[4] In geophysical-exploration usage, the mappings to the inside and outside of the circle would be reversed.

Filters of this type, like the nonrecursive filters of the previous chapter, go by several different names: electrical engineers call them **Infinite Impulse Response** (IIR) filters, and statisticians call them **auto-regressive moving-average** (ARMA) systems. If $L = 1$, so that only the current value of the input is used, the statisticians' term is that the system is **autoregressive** (AR).

Just as we used the Laplace transform to find the transfer function of a differential equation, we may use the $z$-transform to find a transfer function for a discrete-time filter. If we take the $z$-transforms of both sides of equation (6.12), equation, and use the result (6.10) that the $z$-transform of a delayed sequence is $z^{-m}$ times the transform of the original sequence, the two sides of (6.12) become polynomials in $z$. The transfer function becomes the ratio of these polynomials:

$$\frac{\tilde{y}(z)}{\tilde{x}(z)} = \frac{\sum_{l=0}^{L-1} b_l z^l}{\sum_{k=0}^{N-1} a_k z^k} \tag{6.13}$$

This is actually a general equation for any type of digital filter, recursive or nonrecursive; note that for zero phase shift FIR filters the range of summation will include negative as well as positive values of $l$.

The parallel between (6.12) and (6.13) on the one hand, and the differential equation (6.1) and its transfer function (6.7) is quite intentional: this parallelism shows how to reach our goal of simulating a system with a digital filter. The simulation problem now becomes how to find the coefficients of the filter (6.12), with $z$-transform (6.13), that will best imitate the behavior described by the differential equation (6.1) with transfer function (6.7).

We start with our simplest example, the lowpass RC filter, and begin with the step response. If $x$ is constant and equal to $x_0$, the solution to the differential equation (6.2) is $y = x_0$. If at time $t = 0$, $x$ then becomes zero, the solution for $y$ for $t > 0$ is $y = x_0 e^{-t/\tau}$. An FIR filter cannot reproduce the infinitely long response to a step that this system shows; even approximating it would require a large number of weights. But a very simple recursive filter can give a similar response. This is

$$y_n - a y_{n-1} = x_n(1-a)$$

For $x$ constant this gives a constant $y$ equal to $x$. If $x_n$ is 0 for $n \le 0$, and 1 for $n \ge 1$, it is easy to compute $y$:

$$y_0 = 1 \quad y_1 = a \quad y_2 = a^2 \qquad y_m = a^m = e^{m \ln a}$$

giving the same exponential falloff as in the analog filter. The transfer function of this digital filter is

$$\tilde{y}(z)(1-az^{-1}) = \tilde{x}(z)(1-a) \quad \text{or} \quad \frac{\tilde{y}(z)}{\tilde{x}(z)} = \frac{1-a}{1-az^{-1}}$$

which has the frequency response (using $z = e^{2\pi if}$)

$$\frac{1-a}{1-ae^{-2\pi if}} \tag{6.14}$$

where $f$ is now the digital frequency, running over the range from $-\frac{1}{2}$ to $\frac{1}{2}$ (the range between the Nyquist frequencies). Clearly the analog response (6.3) and the digital response (6.14), though similar, are not the same; the problem we will now address is to devise a recursive filter that will be closer.

However, before doing so, we need to discuss the stability criterion for digital filters: what makes them stable or not. If we look at the output of the discrete-time filter for the case given above (equation (6.13), we see that for $a$ less than 1, the output decays when $x$ goes to zero (from one); but if $a$ were greater than 1, the output would increase exponentially. The value of $a$ thus determines if the filter is stable or unstable, the boundary being $a = 1$, for which the filter is metastable. In the $z$-plane, the one filter pole is at $z = a$, so the condition for stability is that the pole is inside the unit circle. This can be shown to be true in general. Just as a continuous-time system, to be stable, must have all poles on the negative real half of the $s$-plane, for a discrete-time filter, all its poles must fall within the unit circle. These criteria matter because we want any mapping from the $s$-plane to the $z$-plane to preserve stability: the "stable" part of each plane should map only into the stable part of the other. The mapping given by equation (6.11), corresponding to sampling, fulfills this criterion; some others do not. For example, creating discrete-time systems by replacing derivatives by forward differences does not; if we perform this substitution on a stable differential equation, the resulting discrete-time filter may not be stable.

A better mapping than (6.11) for system simulation is called the **bilinear transformation**. To motivate it, we again construct a discrete-time system from the differential equation, but instead of using numerical differences to approximate derivatives, we use numerical approximations to integrals.

For example, we can re-express the differential equation for an RC lowpass filter in integral form as:

$$y(t) = \int_{t_0}^{t} \dot{y}(u)du + y(t_0)$$

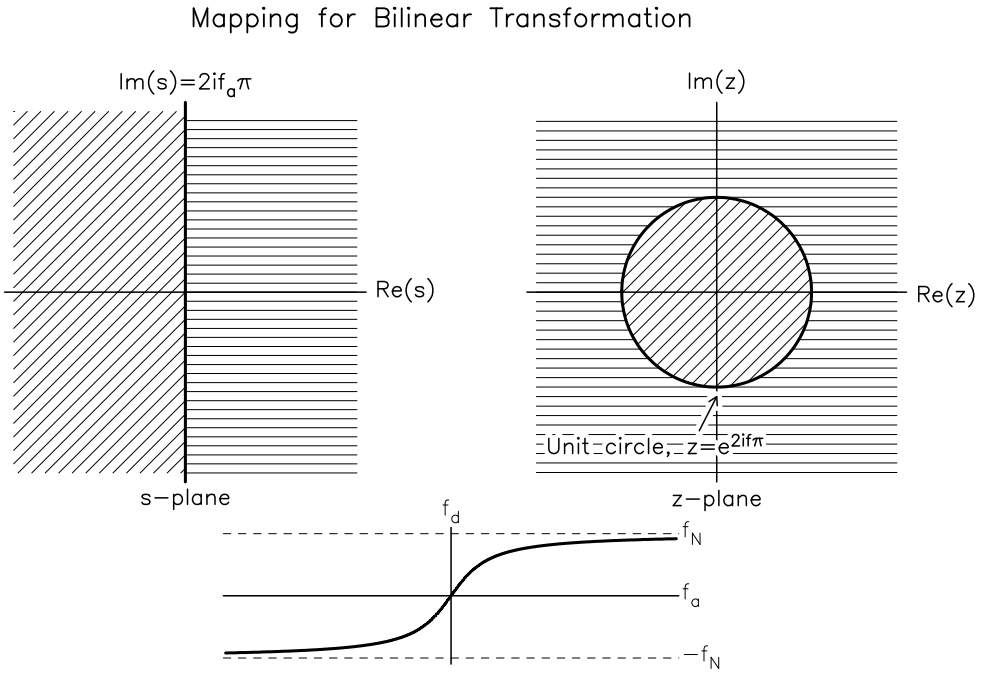



Figure 6.6: Mapping between the $s$-plane and the $z$-plane if the bilinear transformation is used. The right-hand side of the $s$-plane maps outside the unit circle in the $z$-plane, and the left-hand side to the inside of the unit circle; in both cases the mapping is one-to-one. The bottom plot shows how the continuous-time frequency $f_a$ maps into discrete-time frequency $f_d$: the mapping is one-to-one, but nonlinear, though approximately linear for $f_a \ll f_n$.

For equispaced intervals, which we set equal to 1, this becomes

$$y(n) = \int_{n-1}^{n} \dot{y}(u)du + y(n-1)$$

We now use the trapezoidal rule for this integral to write

$$y(n) = \tfrac{1}{2}[\dot{y}(n) + \dot{y}(n-1)] + y(n-1)$$

The differential equation itself gives an expression for $\dot{y}$: $\dot{y}(t) = \tau^{-1}[x(t) - y(t)]$. We may then write the above expression for $y(n)$ as

$$y(n) = \frac{1}{2\tau}[x(n) + x(n-1) - y(n) - y(n-1)] + y(n-1)$$

which we may, finally, write in the form (6.12) of a recursive filter; using subscripts rather than arguments since we are now working with a sequence, we get

$$y_n\left(1+\frac{1}{2\tau}\right)-y_{n-1}\left(1+\frac{1}{2\tau}\right)=\frac{1}{2\tau}(x_n+x_{n-1})$$

The transfer function in the $z$-plane of this discrete-time filter is found in the usual way; we take $z$-transforms to get

$$\tilde{y}(z)\left[\left(1+\frac{1}{2\tau}\right)-z^{-1}\right]=\tilde{x}(z)[1+z^{-1}]$$

whence the transfer function is

$$\frac{1}{1+2\tau}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)$$

We can equate this with the transfer function for the continuous-time case, equation (6.8), if we make the relationship between $s$ and $z$

$$s=2\left(\frac{1-z^{-1}}{1+z^{-1}}\right) \tag{6.15}$$

which is the bilinear transformation.

In the more general case where the sampling interval is $\Delta$, the same derivation may be used to give the same result, with the leading 2 replaced by $2/\Delta$. The inverse mapping from $s$ to $z$ has a similar form, namely

$$z=\frac{1+\frac{1}{2}s\Delta}{1-\frac{1}{2}s\Delta} \tag{6.16}$$

While our derivation used a particular differential equation, using multiple integrals for any constant-coefficient linear differential equation will produce the same mapping. This mapping takes the imaginary axis of the $s$-plane onto the unit circle in the $z$-plane; since it maps the entire left half of the $s$-plane into the inside of the unit circle on the $z$-plane, it maintains stability.

The bilinear transformation thus gives us a design procedure for getting discrete-time filters from continuous-time ones. The steps are:

1. Find the transfer function in the $s$-plane from the differential equation of the system; this will be a ratio of polynomials in $s$.

2. Perform the bilinear mapping (6.16) to produce a transfer function in the $z$-plane; reduce this to a ratio of polynomials in $z$.

3. Get the filter weights from the coefficients of the polynomials, as in the relation between (6.12) and (6.13).

There is often a step before step 1: we may have to modify the differential equation to minimize a distortion from the bilinear mapping. Let the frequency of the discrete-time system be $f_d$, so that (equation (6.9)) $z = e^{2\pi i f_d}$, and of the analog system be $f_a$, so that $s = 2\pi i f_a$. Then

$$\frac{1 - z^{-1}}{1 + z^{-1}} = i \tan(\pi f_d)$$

so that the bilinear transformation of the frequencies is

$$f_d = \frac{1}{\pi} \arctan(\pi f_a \Delta)$$

which maps the entire analog frequency axis, from $-\infty$ to $\infty$, onto the digital frequencies from $-\frac{1}{2}$ to $\frac{1}{2}$: that is, from the lower to the upper Nyquist frequencies. This warping of the frequency axis, though it avoids aliasing of the filter response, means that we must adjust the time constants of the differential equation so that after applying the bilinear transformation these have the correct frequency, an adjustment known as **prewarping**.

To show how to do this, we design a filter to simulate the polar-motion equation (6.4). If we apply the bilinear transformation (6.15) to the transfer function (6.5), and compute the frequency response of the resulting digital filter, we find that it is

$$\frac{\omega_c'}{\omega_c' - (2/\Delta)\tan(\pi f_d)}$$

where we have used $\omega_c'$ to denote that this parameter is not necessarily the same one as in the actual continuous-time system. For $\omega_c$ real, the frequency of the resonance in the continuous-time system is at $f_a = \omega_c/2\pi$; when the series is sampled at an interval $\Delta$, this becomes the (nondimensional) frequency $\omega_c\Delta/2\pi$, providing it is not aliased. In the discrete-time system the resonance will be at

$$f_d = \frac{1}{\pi} \arctan\left(\frac{\omega_c'\Delta}{2}\right)$$

Equating these two digital frequencies gives us the relationship

$$\omega_c' = \frac{2}{\Delta}\tan\left(\frac{\Delta}{2}\omega_c\right)$$

for the time constant (in this case a frequency) used in the discrete-time filter as a function of the one actually present in the system. If $\omega_c\Delta \ll 1$, the two are nearly equal; but as this product approaches $\pi$ more and more correction is needed. In this particular example, the usual sample interval $\Delta$ for $\mathbf{p}$ and $\psi$ is 30 days; for $\omega_c = 0.01461$rad/day, $\omega_c' = 0.01485$, only a 2% change.

Having made this correction, we can compute the actual filter weights by the procedure just described. We find that the polynomial in $z$ is

$$\frac{\omega_c'\Delta + \omega_c'\Delta z^{-1}}{(\omega_c'\Delta + 2i) + (\omega_c'\Delta - 2i)z^{-1}}$$

which gives a recursive filter

$$\mathbf{p_n}(\omega_c'\Delta + 2i) = -\mathbf{p_{n-1}}(\omega_c'\Delta - 2i) + \omega_c'\Delta(\psi_\mathbf{n} + \psi_\mathbf{n-1})$$

for producing simulated polar motion from a possible excitation.

CHAPTER 7

# DIGITAL FILTERS III: DIFFERENTIATORS, LEAST-SQUARES, AND MINIMUM-PHASE FILTERS

## 7.1 FIR Filters: Other Applications

In this chapter we look at some other uses for FIR filters than removing bands of frequencies: differentiation, Hilbert transforms, and linear regression. We also describe the most important class of non-symmetric FIR filters for frequency selection, the minimum-phase filters; these are sometimes important for use in seismology.

## 7.2 Differentiators and Digital Hilbert Transformers

We have mentioned before (Section 2.8) that differentiation can be thought of as a linear system; this makes it appropriate for digital filtering. Equation (2.5)

$$\mathcal{F}[\dot{x}(t)] = 2\pi i f \mathcal{F}[x(t)]$$

shows that differentiation is a filter with frequency response $2\pi i f$: that is, a constant phase shift of $\pi/2$ (the $i$ part of the coefficient) and an amplitude response that is proportional to frequency.

To imitate this digitally, we make the number of weights odd, with $N = 2M + 1$, and also make them antisymmetric:

$$w_{-n} = -w_n \quad \text{for} \quad n = 0, \ldots M$$

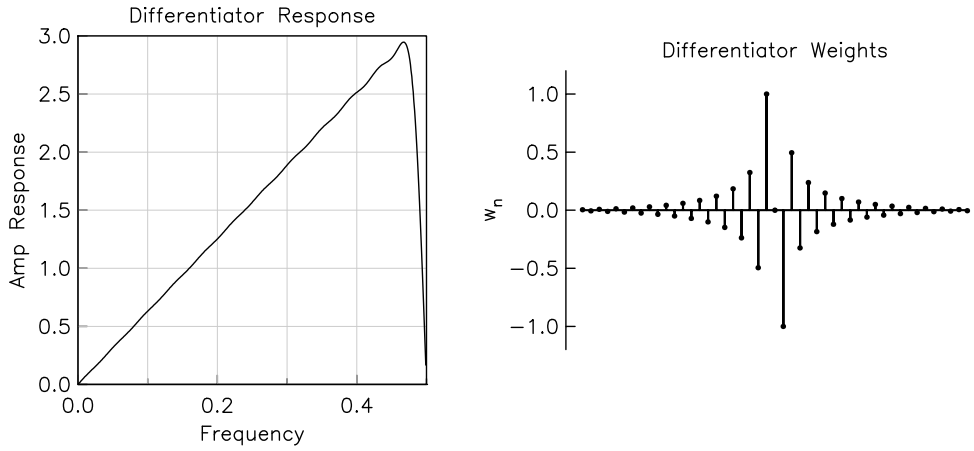



Figure 7.1: The amplitude response, and weights, for a differentiating filter, constructed by tapering the weights for an ideal response with a Kaiser-Bessel taper so that the response at the Nyquist is −40dB.

where we have centered the filter about 0 rather than term $M$; this asymmetry means that $w_0 = 0$. Substituting these weights into the general expression (equation (5.1)) for the frequency response of a FIR filter, we find

$$W(f) = -2i \sum_{n=1}^{M} w_n \sin 2\pi n f$$

which we may compare with equation (5.3) for a symmetric acausal FIR filter.

An antisymmetric filter thus will have exactly the same phase response as differentiation: purely imaginary. To get an amplitude response that approximates differentiation, we take the ideal response to be $W(F) = 2\pi i f$ – but, to avoid over-amplifying the energy close to the Nyquist, only up to some cutoff frequency $\alpha$. That is, we want the filter to have an amplitude response that is zero at zero frequency, increases linearly to the cutoff frequency, and then falls to zero.

As with the frequency-selective filters, we can get the filter weights corresponding to this ideal response by taking the inverse Fourier transform of it:

$$w_n = \int_{-\frac{1}{2}\alpha}^{\frac{1}{2}\alpha} W(f) e^{2\pi i f n} df = 4\pi i \int_0^{\frac{1}{2}\alpha} f i \sin 2\pi f n \, df = -4\pi \int_0^{\frac{1}{2}\alpha} f \sin 2\pi n f \, df$$

$$= \frac{1}{\pi n^2} \int_0^{\frac{1}{2}\alpha} 2\pi n f \sin 2\pi n f \, d(2\pi n f) = \frac{1}{\pi n^2} \int_0^{\pi n \alpha} x \sin x \, dx = \frac{\alpha \cos(\pi n)}{n} - \frac{\sin \pi n \alpha}{\pi n^2}$$

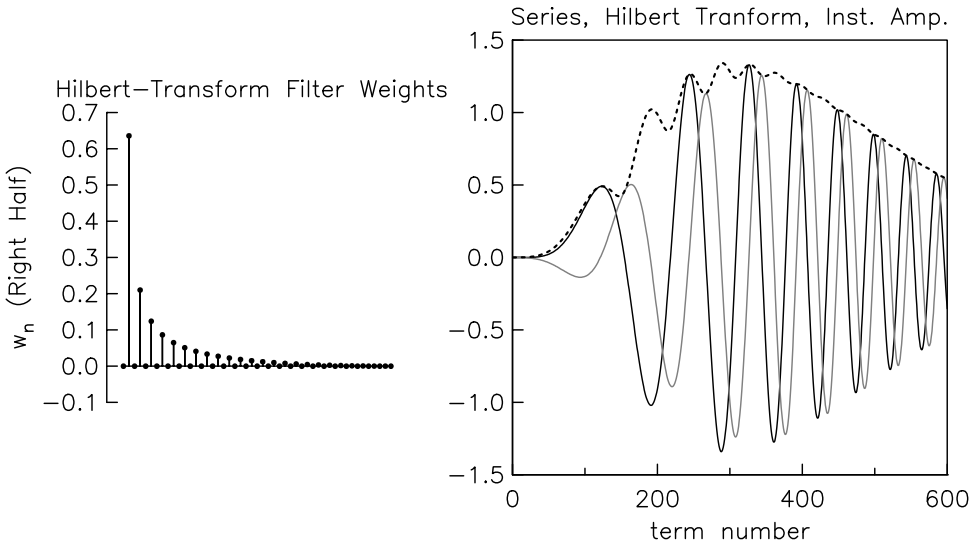



Figure 7.2: On the left, half of the weights for a (windowed) Hilbert transform filter with total length of 97. On the right, a time varying series ($x^3 e^{-x} \sin(1.5x^2)$, where $x = n/10$), shown as the black solid line, its Hilbert transform (gray solid line) and the instantaneous amplitude (dashed line).

where we have made use of the antisymmetry of $f$ around zero frequency. As with the ideal lowpass filter, this series of weights goes on forever; and, as in that case, we can approximate the ideal response better if we taper the series of weights instead of simply truncating it. Figure 7.1 shows an example. It is also possible to adapt the Parks-McClellan procedure to this design task.

Antisymmetric filters may have other amplitude responses. An important case is when there is no frequency dependence at all: an amplitude response equal to one from zero frequency to the Nyquist. This is the digital **Hilbert transform** filter, whose ideal response is

$$W(f) = \begin{cases} i & f < 0 \\ \text{for} & \\ -i & f > 0 \end{cases}$$

Note that this has the opposite phase shift from differentiation. The ideal weights are

$$w_n = \frac{2\sin^2(n\pi/2)}{n\pi}$$

Again, actual filters can be designed by using tapers (as was done for the filter in Figure 7.2) or using the Parks-McClellan algorithm. Either of these methods will produce a filter whose amplitude response is only approximately flat over the full frequency band, but whose phase response is, like that of the differentiator, exact, thanks to the antisymmetry of the filter, and its having an odd number of weights.

If a Hilbert-transform filter, is convolved with a series it converts all the cosine components into sines and vice versa. This can be useful for a series that is approximately sinusoidal, since then we can create what is called a complex-valued **analytic series**

$$\{x^a\} = \{x\} + i\{x^h\}$$

where $\{x^h\}$ is the digital Hilbert transform of the original sequence $\{x\}$. From this, we can find the **instantaneous amplitude** (also called the **envelope function**) and **instantaneous phase**; these are the amplitude and phase of each term of $\{x^a\}$; differentiating the phase gives the **instantaneous frequency**. Figure 7.2 shows an example for a gradually-changing sinusoid.

## 7.3 Least-squares Fits as Filters

Next we show that parameters estimated through least-squares can be found by filtering the data – and that sometimes, thinking of least-squares fitting as a filter can help us understand what the fit is doing. The basis of this is that the solution to any least-squares problem gives the parameters in terms of a linear combination of the data; but any linear combination of the data is a linear time-invariant system, and hence a filtering operation.

More formally, suppose we have a data vector $x$, which we aim to fit parameters $y$ to by least squares. The equations for $y$ are the normal equations:

$$y = (A^T A)^{-1} A^T x$$

where A is the design matrix.[1] Now suppose we want to find a particular linear combination of the parameters, $z = b^T y$; $z$ is a scalar (and might just be one of the parameters). Then the solution is

$$z = b^T (A^T A)^{-1} A^T x \stackrel{\text{def}}{=} w^T x$$

[1] We have assumed that the errors are identical and independent; if they are not the equations are slightly more complicated but we get an equivalent end result.

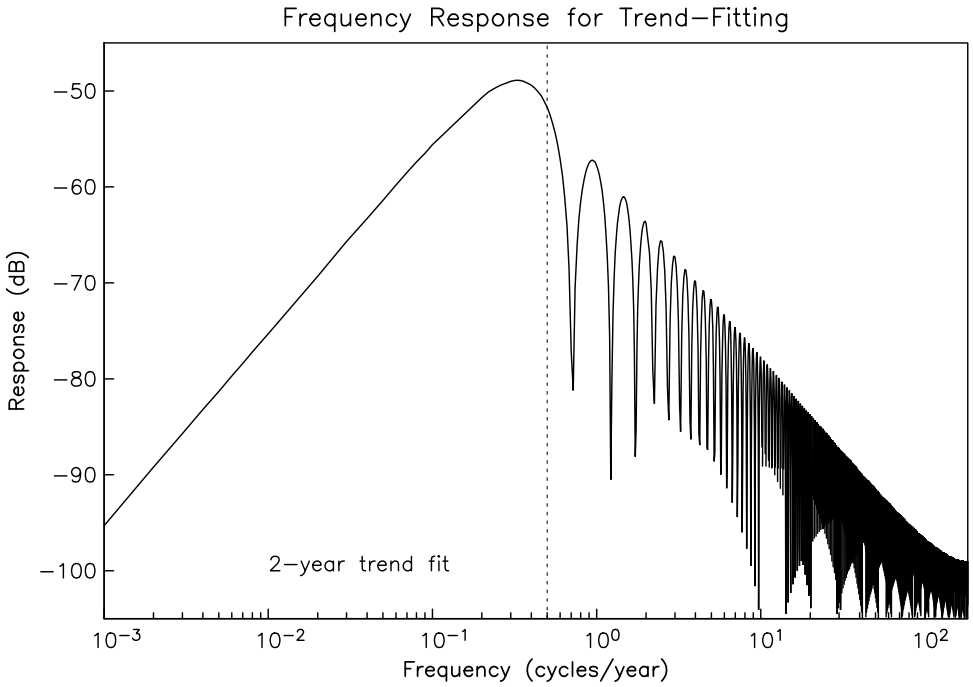



Figure 7.3: An example of the frequency response of a least-squares fit: in this case, for the slope of daily data fit over a two-year span.

where the weights $w$ are given by $b^T(A^TA)^{-1}A^T$, which can be seen to be an $N$-term vector, $N$ being the number of data. But this is just a digital convolution (albeit with only one output value) – so we may regard the weights as filter weights, and compute the frequency response by taking their Fourier transform.

A simple, and common, example is fitting a straight line to data; assume we have $2M+1$ values, centered about 0. Then if $y_1$ is the value of the fit at $n=0$ and $y_2$ is the slope, the design matrix is

$$A^T = \begin{bmatrix} 1 & 1 & \dots & 1 & 1 & 1 & \dots & 1 & 1 \\ -M & -M+1 & \dots & -1 & 0 & 1 & \dots & M-1 & M \end{bmatrix}$$

which, conveniently, means that the matrix for the normal equations is diagonal:

$$A^TA = \begin{bmatrix} 2M+1 & 0 \\ 0 & M(M+1)(2M+1) \end{bmatrix}$$

which in turn makes the weights for finding the slope

$$w_n = \frac{3n}{M(M+1)(2M+1)} \qquad \text{for} \qquad n = -M, \ldots M$$

Note that this is quite different from the weights for finding the derivative, which is a local measure of slope: the weights for the fitted slope are largest far from the center, something that is often stated by saying that such parts of the series have more influence. The frequency response, which then is

$$W(f) = \frac{6}{M(M+1)(2M+1)} \sum_{n=1}^{M} n \sin(2\pi n f)$$

so that the frequency response depends on $M$, the length over which the fit is made. Figure 7.3 shows the frequency response for $N = 730$: this might be, for example, 2 years of daily geodetic data. What is interesting is that the response actually peaks at a frequency of about 0.3 cycles/year. This is lower than the lowest frequency that is usually resolvable from data, namely one cycle over the record, which would be 0.5 cycles/year. The location of this peak in the response becomes more obvious if we think about what kind of sinusoid "looks like" a straight line: namely, one with about a quarter of a cycle over the range of data. Much lower frequencies look more and more like a constant.

## 7.4 Minimum-Phase Filters

Next, we turn to what are called *minimum-phase* filters. As mentioned in Section 5.2, sometimes we do not want to use zero-phase filters because they lack causality: if the data have sudden changes (notably from earthquakes) we may not want leakage of energy into times earlier than the time at which the change occurs. But, also, we usually want the filter to be as compact as possible: that is, we would like the filter to spread any impulses or steps over the minimum possible time span. Making the filter compact, but still causal, turns out to be equivalent to minimizing the phase shift.

The design of such filters depends on the locations of the roots of the $z$-transform polynomial; these are, for weights $w_0, w_1, \ldots w_N$,

$$W(z) = \sum_{k=0}^{N-1} w_k z^{-k}$$

where $z$ is complex. This polynomial has $N-1$ roots, $r_1, r_2, \ldots r_{N-1}$; given these roots, the $z$-transform can alternatively be written as a product of degree-one polynomials:

$$W(z) = w_0 \prod_{k=1}^{N-1} (1 - r_k z^{-k})$$

with the $w_0$ providing appropriate scaling.

For common choices of the weights $w_n$, there are a number of "symmetry" relationships that apply to the roots $r_k$. If the weights are real, the roots $r_k$ must either be real, or occur as complex conjugate pairs. If the filter is symmetric, with an odd number $N = 2M+1$ of weights, then if $r_k$ is a root, so is $r_l = 1/r_k$: the roots occur as reciprocal pairs unless they are on the unit circle. To see this, write the $z$-transform as

$$W(z) = \sum_{k=0}^{2M} w_k z^{-k} = z^{-M} \left[ w_M + \sum_{k=0}^{M-1} w_k (z^{M-k} + z^{k-M}) \right] = z^{-2M} W(z^{-1})$$

so that if $W(r_k) = 0$, then $W(1/r_k) = 0$ as well.

Combining these results, we have that for real and symmetric weights (the commonest case), the roots of $W(z)$ occur in reciprocal pairs plus single roots on the unit circle, and all pairs (or singlets) not purely real occur as complex conjugate pairs.

If the weights are real and symmetric, the response on the unit circle is

$$W(f) = e^{-2\pi f M} \left[ w_M + 2 \sum_{k=0}^{M-1} w_k \cos 2\pi k f \right] = e^{-2\pi f M} W_a(f)$$

where $W_a(f)$ is the amplitude response, which is purely real. If there are single roots of $W(z)$ on the unit circle, there must be values of $f$ for which $W_a(f) \leq 0$. Conversely, if $W_a(f) > 0$ for all $f$, there can be no roots on the unit circle: all roots occur as reciprocal pairs. Quite commonly, $W_a(f)$ is nonnegative, but equal to zero for particular values of $f$; these values of $f$ then correspond to double roots of $W(z)$, for which the root and its reciprocal are equal.

We have discussed the locations of the roots in such detail because of some less obvious results that relate to the goal of obtaining a compact filter. Suppose some sequence $w_n$, with $z$-transform $W(z)$, has a root within the unit circle; without loss of generality we can take this to be the "last"

root $r_{N-1}$. Now consider another $z$-transform polynomial

$$W'(z) = W(z)\frac{z^{-1} - r_{N-1}^*}{1 - r_{N-1}z^{-1}} \stackrel{\text{def}}{=} W(z)A(z) \tag{7.1}$$

The form of the term $A(z)$ means that for the polynomial $W'$ the root at $r_{N-1}$ has been divided out of the polynomial for $W(f)$ and replaced by its reciprocal complex conjugate, which will be outside the unit circle. For $z = e^{-2i\pi f}$,

$$|A(Z)|^2 = \left| z^{-1}\frac{1 - r_{N-1}^* z}{1 - r_n z^{-1}} \right|^2 = 1$$

which means that the amplitude responses $|W(f)|^2$ and $|W'(f)|^2$ are identical. For this reason filter expressed by the $A(z)$ polynomial is termed an **allpass filter**; note that this is a recursive filter.

We can use the factorization (7.1) to get a relationship between the two sets of weights, $w_k$ and $w'_k$, that have the responses $W(z)$ and $W'(z)$. Let $V(z)$ be a polynomial of degree $N-2$, obtained by factoring out the last root (this is called deflation of the polynomial). Then we can rearrange equation (7.1), to get

$$W(z) = V(z)(1 - r_{N-1}z^{-1}) \qquad \text{and} \qquad W'(z) = V(z)(z^{-1} - r_{N-1}^*)$$

But these polynomial relationships imply the following expressions for the coefficients of $W$ and $W'$:

$$w_k = v_n * (1, -r_n) = v_k - r_{N-1}v_{k-1}$$
$$w'_k = v_n * (-r_n^*, 1) = -r_{N-1}v_k + v_{k-1}$$

Next, we compute the difference of the sums of squares of the weights up to the penultimate term. Most of the terms in the sum cancel, giving

$$\sum_{k=0}^{N-2} |w_k|^2 - |w'_k|^2 = (1 - |r_{N-1}|^2)|v_{N-2}|^2$$

Because $|r_{N-1}| < 1$ (by assumption, we chose a root with this property), this sum has to be positive. This means that the sequence $w_k$ has more energy (measured by the sum of squared amplitudes) concentrated before the last point than the sequence $w'_k$ does.

This result can be extended to show that, among all FIR filters with the same amplitude response $|W(f)|^2$, the one whose $z$-transform has all its

roots on or inside the unit circle will have the most concentration of energy towards the early terms of the sequence; that is,

$$\sum_{k=0}^{m} |w_k|^2$$

will be maximized for all $m < N-1$. (For $m = N-1$ the sum is the same, being the filter response at $f = 0$.) This choice of roots thus makes the filter as compact as possible.

Additional results follow if we express $W(f)$ as amplitude and phase: $W(f) = W_a(f)e^{-\phi(f)}$. The phase of $W$ and $W'$ are related, by equation (7.1), though the phase of the allpass term in equation (7.1). If we write the root in polar form as $r_{N-1} = \rho e^{2\pi i\theta}$, we find that its phase is the phase of

$$e^{-2\pi if}\left(\frac{1-\rho e^{2\pi i(f-\theta)}}{1-\rho e^{-2\pi i(f-\theta)}}\right)$$

which gives a phase

$$\phi(f) = -2\pi f - 2\arctan\left[\frac{\rho\sin 2\pi(f-\theta)}{1-\rho\cos 2\pi(f-\theta)}\right] \tag{7.2}$$

A more interesting quantity is the **group delay** $D(f)$, which defines the delay energy in a signal with frequency $f$ (exactly analogous to group velocity in wave propagation), which is given by the derivative of the phase. If the phase is given by

$$\phi(f) = \arctan\left(\frac{S(f)}{C(f)}\right)$$

the group delay is given by

$$D(f) = \frac{1}{2\pi}\frac{d\phi}{df} = \frac{1}{2\pi}\frac{S'(f)C(f) - C'(f)S(f)}{S^2(f) + C^2(f)}$$

which when applied to (7.2) gives

$$D(f) = \frac{-1-\rho^2}{1+\rho^2-2\rho\cos 2\pi(f-\theta)}$$

which is negative for $\rho < 0$ (the root inside the unit circle) so that $W'(f)$ has a greater delay, and a larger (more negative) phase than $W(f)$. The sequence with all the roots inside the unit circle is thus not only the most

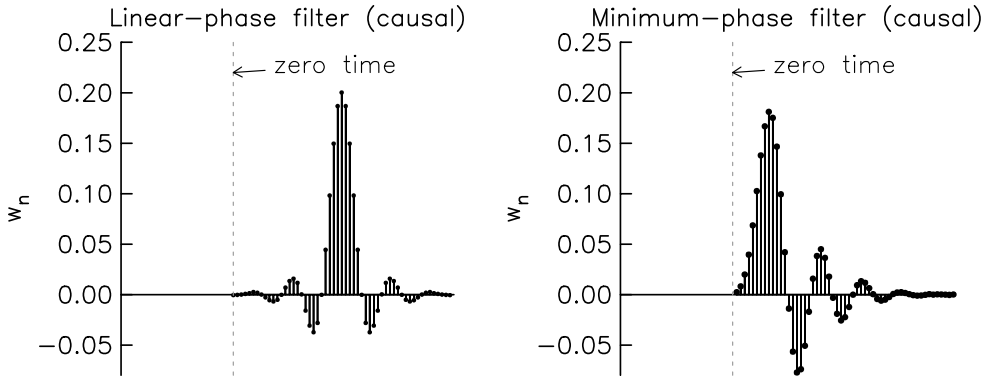



Figure 7.4: Weights for two filters with the same (lowpass) frequency response: on the left, a symmetric filter, on the right, its minimum-phase equivalent. Both are shown as being applied causally.

compact; of all the filters with a given amplitude response, it minimizes group delay, and makes the phase lag as small as possible. Such sequences are therefore usually referred to as providing **minimum-phase** FIR filters.

To design such filters requires, besides approximating the desired frequency response, that the roots of the $z$-transform all fall inside of or on the unit circle in the complex plane. There are two different ways to accomplish this.

One, suggested by equation (7.1), is to replace all roots outside the unit circle with their reciprocal complex conjugates, leaving roots on the unit circle in place. This procedure is called the **allpass decomposition**, because of the allpass term in equation (7.1). This equation, generalized to contain as many such factors as there are roots outside the unit circle, becomes the result that any FIR filter can be expressed as the convolution of a minimum-phase FIR filter and an allpass filter. Note that the allpass filter itself is not a FIR filter, but would need to be computed recursively; this is acceptable for correcting for the previous application of a non-causal FIR filter to data (Scherbaum and Bouin, 1997).

A second approach, called **spectral factorization**, depends on the result that, for symmetric filters, if $W_a(f) \geq 0$ on the unit circle, then all the roots can be paired into reciprocals, with double roots on the unit circle. The factorization consists of taking all the roots inside the unit circle and half of the roots on it, and constructing the corresponding sequence.[2] The

[2] A good description of the procedure, and some pitfalls, is given in Orchard and Willson (2003) – though there have been difficulties with the Newton's method procedure given

resulting filter will have an amplitude response $|W(f)| = \sqrt{W_a(f)}$, which therefore requires that the initial filter have a nonnegative amplitude response, $W(f) \geq 0$. Such a filter can be created using the Parks-McClellan algorithm if we constrain the stopband response to oscillate around the error level $\epsilon$, rather than (as is more usual) zero.

Figure 7.4 shows an example of filter weights for a zero-phase and minimum-phase filter with the same amplitude response.[3] It is obvious that the symmetric (linear-phase) filter, applied as a causal filter. is less compact than the minimum-phase filter applied the same way.

there.

[3] This is one of several filters combined to produce 5-minute data from 1-Hz data recorded with the strainmeters installed by the Plate Boundary Observatory. For more details, see Agnew and Hodgkinson (2007),

CHAPTER 8

# STOCHASTIC PROCESSES

## 8.1 Introducing Ordered Random Data

We now turn to sequences of random data: that is, data in which (1) the order of the values is important, and (2) the data are random. The first attribute is the same kind of specification that we have been dealing with in signal processing: we have a sequence of data $x_1, x_2, x_3, \ldots x_n$, which we will often call, as statisticians do, a **time series**. As before, we assume that the index $n$ denotes a time $t = n\Delta$ at which the observation was made: we are again sampling uniformly in time at an interval $\Delta$. Obviously we could equally well be treating a data set sampled in space or, much more rarely, some other variable. For theoretical purposes it is sometimes useful to be able to treat time as a continuous variable, and to work with functions $x(t)$; but as we will see, if we do this for a random variable things can become mathematically extremely difficult without some further quite severe simplifications. We will, as before, often want to consider infinite sequences, maybe infinite in both directions.

The time series of data $\{x_n\}$ is modeled by a **stochastic process** $\{X_n\}$ or $\{X(t)\}$. Such a process is a family of random variables indexed by the integer $n$ (when it is a discrete process) or by the real number $t$ (when it is a continuous process).[1]

What do we mean when we say "random variable"? The conventional variables we have been dealing with up to now (and which we did not, up to now, need a special name for) were a particular kind of mathematical entity that obeyed certain rules. Random variables are also mathematical entities, with different rules.

A. A conventional variable is assumed to have a definite value, although we may not know it: it might, for example, be integer-valued, real-

[1] **Notation alert**: random variables or functions will normally be denoted by uppercase letters, but not all uppercase letters denote random variables.

valued, complex, or a vector. A random variable does not have a definite value; rather, it is described by its **probability distribution function** (pdf), giving the probability density over some range. The range, and any parameters of the function used for the pdf, are however conventional variables.

B. Conventional variables obey the rules of algebra (arithmetic): we can add, subtract, multiply, or (for real-valued ones) divide them. Formally, we can perform the same operations on random variables, but what happens when we do is described by a new set of rules. For example, the sum of two random variables is another one: but the pdf of this sum is not the sum of the pdf's of the two variables that have been added.

Conventional variables are appropriate models for things we want to represent as having a definite value: for example, data values that we have measured or otherwise acquired. This makes them appropriate for signal processing: we assume the signal has some definite value. The terms of a sequence may not be equal to a known number (that is why they are variables, after all), but we treat them as though at some point they will. In contrast, random variables are appropriate models for things that are somehow uncertain. We might be using a random variable to model the outcome of some experiment, in which case we can think of it as one of an infinite collection of values from which the experiment will extract a single one. Or it might be used to model something about which we are simply uncertain.

A key point is that the choice to model something by a random variable is just that: a choice of what mathematical entity we decide is appropriate to what we are studying. There are no absolute rules for this, and it may well be that, depending on what different models are for, we might treat the same thing as a conventional variable in one, and as a random variable in the other.

One way to think about a given series of random variables is as being the result of an experiment that could be repeated as many times as we like. We say that each repetition produces one **realization**, drawn from an **ensemble** of series that would be produced if we made many repetitions; in that sense the ensemble is generated by the underlying random process. So we can think of such operations as finding the **expected value** at a particular $n$ or $t$, as the average, over an infinite number of realizations, of the variable for that $n$ or $t$. Figure 8.1 shows realizations of a several stochastic processes.

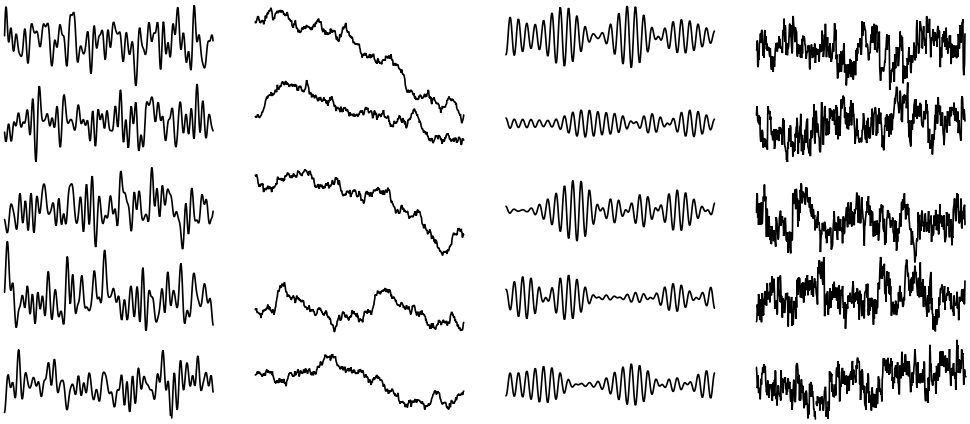


Figure 8.1: Five realizations of four different stochastic processes: from left to right, lowpassed white noise, a random walk, a randomly modulated sinusoid, and $f^{-1}$, or "flicker" noise.

Even in discrete time the general stochastic process is horribly complex. A single random variable is specified by a single pdf; but a set of them is specified completely only if we have the joint pdf of every element $X_n$ with every other element. Suppose all the elements have a Gaussian distribution. Then for a sequence of length $N$ the joint pdf is a function $\phi$ of an $N$-vector $\mathbf{X} = (X_1, X_2, \ldots X_N)^T$

$$\phi(\mathbf{X}) = \frac{1}{(2\pi)^{N/2}} \det(C) \exp[-\tfrac{1}{2}(\mathbf{X} - \bar{\mathbf{x}})^T C^{-1} (\mathbf{X} - \bar{\mathbf{x}})] \tag{8.1}$$

This expression contains $N$ parameters $\bar{\mathbf{x}} \in |\Re^N|$, the vector of mean values, and $C \in |\Re^{N \times N}|$, the covariance matrix. This is a symmetric, positive definite matrix that describes the correlations among the random variables $X_n$, and has $\frac{1}{2}N(N+1)$ values. So, to describe completely these $N$ random variables, we need a total of $\frac{1}{2}N(N+3)$ parameters, far more than we are likely to have observations: even a short time series of 50 numbers would need 1325 parameters to be fully specified, and for a reasonably long data series of 2000 terms, we would need more than 2 million parameters. If we instead consider continuous time the problem is almost completely intractable, and such a general treatment has no practical value because it would be impossible to make estimates of the necessary parameters, even if the mathematics were possible – and this turns out to be very hard for the general case (see Priestley (1981), Chapter 3).

## 8.2 Stationary Processes and Autocovariance

Since the perfectly general stochastic process is much too general to be useful, we follow the conventional approach of limiting ourselves to a much more restricted class of random processes: those called **stationary processes.** These have the property that, while the actual observables vary over time, the underlying statistical description is *time invariant*. This restriction can be weakened a bit, but we will consider only these so-called **completely stationary** processes. Assuming **stationarity** (as this restriction is called) has the gratifying effect of reducing the number of parameters needed to a manageable number. For example, the mean value of a stationary process is

$$\mathscr{E}[X_n] = \bar{x}$$

where $\mathscr{E}$ stands for "the expectation of"; we could call this the expectation operator, which converts a random variable into a conventional one. For a stationary process the mean is one number: a constant independent of $n$, or of time $t$ if the process is continuous. It is often assumed the mean is zero, since it is a trivial operation to remove or add the mean value into a time series, if necessary. Of course, many observational series do not look as if the mean is constant – there may be a **secular trend**, as there was in the first example we gave in Chapter 1. Stationarity is such a powerful and useful property that one often attempts to convert an evidently nonstationary series into a stationary process, for example, by fitting and removing a straight line trend, or forming a new stationary sequence by differencing:

$$Y_n = X_{n+1} - X_n$$

The **variance** of a single random variable is defined as

$$\sigma_X^2 = \mathscr{V}[X_n] = \mathscr{E}[(X_n - \bar{x})^2]$$

which defines the variance operator $\mathscr{V}$; again, this converts from a random to a conventional variable. With stationarity $\sigma_X^2$ must also be independent of $n$ (or $t$). But most importantly, the covariance between any two random variables in the sequence cannot depend on where we are in the series; it must look like

$$\mathscr{C}[X_m, X_n] = \mathscr{E}[(X_m - \bar{x})(X_n - \bar{x})] \stackrel{\text{def}}{=} R_X(m-n) \tag{8.2}$$

which is to say that the covariance can be expressed by a function $R_X$, whose argument is the interval between the two points. (Note that though we use a capital letter for this it is not itself random). The function $R_X$ is called the **autocovariance.** For continuous processes equation (8.2) is usually written

$$\mathscr{C}[X(t), X(t+\tau)] = R_X(\tau) \tag{8.3}$$

and $\tau$ is called the **lag**. Observe that by definition

$$R_X(0) = \sigma_X^2$$

Also notice that, because of stationarity, one can set $s = t - \tau$ in (8.3) and the result will be the same, since the answer is independent of which time was selected. This yields:

$$R_X(-\tau) = R_X(\tau)$$

which shows that the autocovariance function is an even function of the lag. A stationary process does not contain information on which way time is flowing – it is the same if time is reversed.

For a stochastic process with a Gaussian pdf as in (8.1), we see that in place of the vector $\bar{\mathbf{x}}$ of mean values we have a single number. How about the covariance matrix? You may recall that the $j$-$k$th entry of $C$ is

$$C_{jk} = \mathscr{C}[X_j, X_k] \stackrel{\text{def}}{=} R_X(j-k)$$

and so, for a stationary process, the covariance matrix has the same values on all its diagonals:

$$C = \begin{bmatrix} R_X(0) & R_X(1) & R_X(2) & R_X(3) & \dots & R_X(N) \\ R_X(1) & R_X(0) & R_X(1) & R_X(2) & \dots & R_X(N-1) \\ R_X(2) & R_X(1) & R_X(0) & R_X(1) & \dots & R_X(N-2) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ R_X(N) & R_X(N-1) & R_X(N-2) & R_X(N-3) & \dots & R_X(0) \end{bmatrix}$$

This type of matrix is called a **Toeplitz** matrix. So we only need $N+1$ parameters, instead of $\frac{1}{2}N(N+3)$ parameters, to describe the pdf of the process.

You should note that just because a stochastic process is composed of random variables, this does not mean it is completely unpredictable. Recall the correlation coefficient of two random variables:

$$\rho_{XY} = \frac{\mathscr{C}[X,Y]}{\sqrt{\mathscr{V}[X]\mathscr{V}[Y]}} \tag{8.4}$$

From (8.3) we find the correlation coefficient between any two points in a sequence in continuous time:

$$\rho(\tau) = \frac{R_X(\tau)}{\sigma_X^2}$$

with a similar result for discrete processes. The function $\rho$ is called the **autocorrelation** function; it has the advantage over the autocovariance that the variance has been removed from the definition, so that for any process $-1 \le \rho \le 1$. Unless the autocovariance function is zero for the lag $\tau$, there is some correlation between the value $X(t)$, and the later value $X(t+\tau)$; we can predict something about one given the other.

So far we have considered the average of the variables, and also averages of the products of variables, $(X(t_1) - \bar{x})(X(t_2) - \bar{x})$. Such averages are examples of **moments** of the process. The second-order moments are the ones involving products, and provide the variance and the autocorrelation function (or the autocovariance, which includes both), nothing else. We can define higher order moments, for example the third-order moments that come from the product of three $X$'s. But if the process is based on a Gaussian pdf, we have seen that the complete pdf is specified by the second-order moments alone: all the higher-order moments, if we wanted them, can be found from these second-order moments. And even if the pdf is not Gaussian, a great deal can be learned about a process from its second-order moments; the higher order moments are rarely studied–and we will not do so.

## 8.3 White Noises and their Relatives

Let us look a few concrete examples of stationary stochastic processes. The simplest one is what is called **white noise**. This is defined as a stationary process in which the random variable at any time (or sequence number) is independent of the variable at any other time.[2] It is commonly assumed that the mean value is zero. In the discrete case we have

$$R_W(n) = \sigma_0^2 \delta_{n0}$$

where $\delta_{jk}$ is the Kronecker delta symbol. In this case the random process

[2] More precisely, that the correlation is zero; for non-Gaussian variables this can be true even if the variables have some dependence. But this gets into the matters of higher moments that we just promised to ignore.

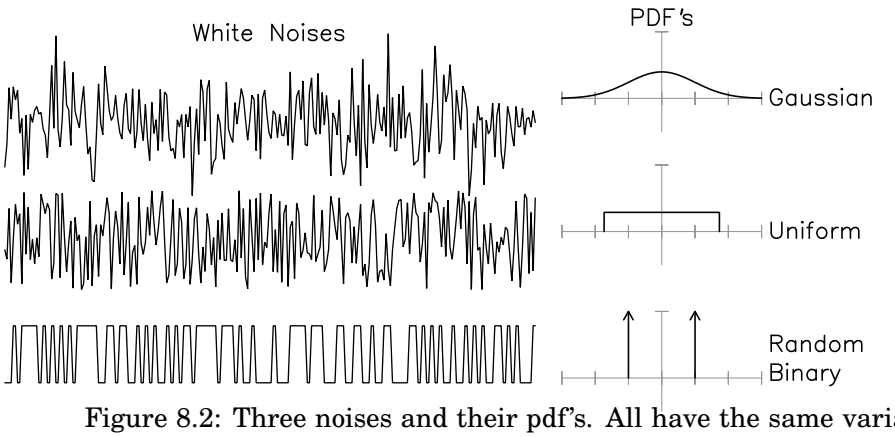



Figure 8.2: Three noises and their pdf's. All have the same variance $\sigma^2$. The top two are white noises; the random telegraph turns out to be slightly nonwhite.

really is unpredictable: we learn nothing about the value at one time from knowing the value at another. However, the values are still not completely unpredictable; we can say something about the range of values to be expected based on the known variance, $\sigma_0^2$. For continuous processes, it turns out that a white noise is singular because the variance at any point must be infinite:

$$R_W(\tau) = \sigma_0^2 \delta(\tau) \tag{8.5}$$

But these definitions do *not* completely specify the stochastic process. At any particular time $t$ (or index $n$) $X$ is a random variable with a pdf; that pdf will be the same for every $t$, but so far we have not specified it. Of course, the most common choice is the Gaussian, for which

$$\phi(X) = \frac{1}{\sigma_0\sqrt{2\pi}} e^{-\frac{1}{2}(X/\sigma_0)^2}$$

Because of the statistical independence, the joint pdf is

$$\phi(X_1, X_2, X_3, \ldots) = \phi(X_1)\phi(X_2)\phi(X_3)\ldots \tag{8.6}$$

This is **Gaussian white noise** which is shown in Figure 8.2.

We could allow any other pdf for $X_n$; suppose we choose a uniform distribution:

$$\phi(X) = b^{-1}\Pi(X/b)$$

so that $\mathcal{V}[X] = b^2/12$. This is a different kind of white noise, as is easily seen by comparing the top two panels in Figure 8.2. The joint pdf is given

by (8.6) again, with the appropriate choice of $\phi$. We have actually encountered this kind of white noise already, since it is what quantization noise (Section 4.7) looks like. If the series varies by much more than $\pm 1$ between samples, the value recorded is the true value of the measurement plus an unpredictable (that is, random) amount that is distributed uniformly over the interval $(-\frac{1}{2}, \frac{1}{2})$. Since the consecutive values of the rounding error are uncorrelated, the recorded series appears to be the true signal with a white noise added. The noise is zero mean and its variance is 1/12. If the last digit doesn't change very often in the series, then the round-off noise added is no longer uncorrelated; but then the signal is most likely not being recorded with enough accuracy.

Another white series with a limited range is the **random telegraph signal**: this one switches discontinuously between two values, say 0 and 1, at random times:

$$\phi(X) = \tfrac{1}{2}[\delta(X) + \delta(X-1)]$$

Here the mean value is not zero but $\frac{1}{2}$, and the variance is a $\frac{1}{4}$. The random telegraph signal is used as a calibration signal for seismometers or other instruments since it is easy to generate electronically. A zero mean version of it has sometimes been suggested as a model for the Earth's dipole moment over time scales of $10^5$ to $10^6$ years, but this is actually very implausible because the moment is far from constant between reversals. The bottom panel of Figure 8.2 shows a sample.

These three examples of white noise are clearly different to the eye. Somewhat remarkably, if they are normalized to the same variance and converted into a sound track, they sound identical – the ear doesn't have a very good density-function discriminator.

These three sequences were not observational; they were made with a random number generator. We can obtain other kinds of stochastic sequences by filtering white noise in various ways; the series in Figure 8.1 were generated that way. Note that as consequence of the Central Limit Theorem. filtered white noise tends to have a Gaussian distribution. Many physical processes can be thought of as resulting from filtering applied to some stochastic process, so Gaussian distributions are often assumed in a signal, and are quite often. though not always, observed too. A common exception, an example of which we will see later, is marine magnetic anomalies; these are usually heavy in the tails compared with a Gaussian distribution.

Suppose we apply a FIR filter, of any type, to white noise; that is, we

convolve the white noise sequence $W_j$ with a finite number of weights $w_k$:

$$Y_n = \sum_{k=1}^{K} w_k W_{n-k}$$

Assume for simplicity that the mean of the white noise is zero. Then we can calculate the autocovariance of the new sequence using the definition (8.2):

$$\begin{aligned} R_Y(l) &= \mathscr{C}[Y_n + l, Y_n] = \mathscr{E}[Y_{n+l} Y_n] \\ &= \mathscr{E}\left[ \sum_{j=1}^{K} w_j W_{n+l-j} \sum_{k=1}^{K} w_k W_{n-k} \right] \\ &= \sum_{j=1}^{K} \sum_{k=1}^{K} w_j w_k \mathscr{E}[W_{n+l-j} W_{n-k}] \\ &= \sum_{j=1}^{K} \sum_{k=1}^{K} w_j w_k \sigma_0^2 \delta_{l-j+k,0} \end{aligned}$$

The delta symbol vanishes except when $l - j + k = 0$ or $j = l + k$; so

$$R_Y(l) = \sigma_0^2 \sum_{k=1}^{K} w_k w_{k+l} = \sigma_0^2 w_k * w_{-k}$$

And so we see that the previously uncorrelated (zero covariance) white noise $W_k$ has become correlated. Notice that $R_Y$ is zero for $|l| > K$.

You are invited to verify that the same result is obtained for the continuous time version: if

$$Y = w * W$$

then

$$R_Y = \sigma^2 w(t) * w(-t)$$

It is possible to produce an even wider range of stochastic processes if we use recursive filters. For example, suppose our recursive filter is

$$y_n = y_{n-1} + x_n$$

which is about as simple as you could ask for. If we apply this to a white-noise sequence $X_n$, the resulting $Y$ is a process known as a **random walk**

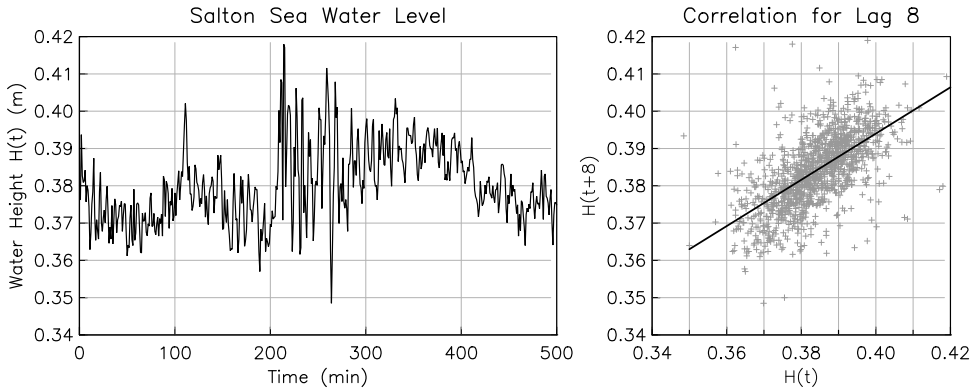



Figure 8.3: On the left, a time series of water height on the shore of the Salton Sea, showing a seiche. The right plot shows the correlation between data at two times eight minutes apart.

or a **Brownian process**.[3] This process is actually nonstationary, with a variance that grows with time.

A great deal of the statistical literature on time series is devoted to processes defined by filters; in Chapter 7 we introduced the terms autoregressive (AR) and autoregressive/moving-average (ARMA) as names for different kinds of recursive filters; and from these come the names AR and ARMA processes. The utility of this approach for geophysical data, outside some special cases, seems to be limited, and we shall not speak of it again.

## 8.4 Examples from the Real World

Instead, we look at some examples of actual observations which a stationary process might be an appropriate model for. Our first example is the water height at the edge of a lake[4] over several hours. The water rises and falls in response to the wind, and because of standing waves in the basin, there are well-defined oscillations, known as **seiches**. Figure 8.3 shows nearly 10 hours of data. Note that the mean is not zero; on the other hand, the zero level is purely arbitrary. and there is no reason not to make the

[3] The "Brownian" name comes from the applicability of this process to explaining the motion of small particles that are moved about by the kinetic motions of (much smaller) molecules, something known as Brownian motion because it was first observed. in pollen, by the microscopist Robert Brown.

[4] A very large, shallow one: the Salton Sea

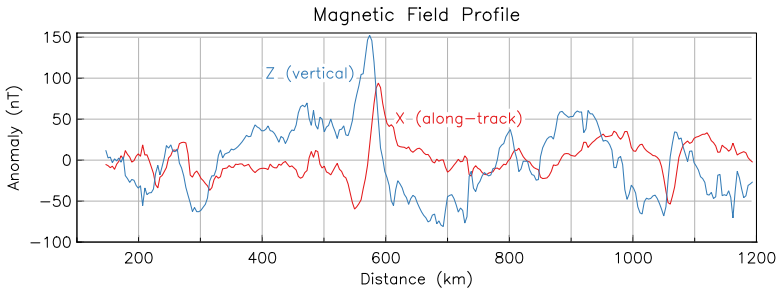



Figure 8.4: Magnetic anomaly profile over the eastern Pacific Ocean. A standard geomagnetic model has been removed, so that the mean values are nearly zero. The sample interval (in space) is 0.35 km.

mean zero. Next notice the oscillations, which are not completely regular, but nonetheless suggest periodicity. This definitely looks like a stochastic process, but one might be skeptical that it is stationary, given the amplitude increase at about 210 minutes. However, we will stick with that model because it is so much more mathematically tractable than the alternative.

Do these data look like a good approximation to a white noise? If they did, values at one time would not be related, to values at other times. That looks improbable to the eye. If we draw a scatter plot (Figure 8.3) where the observed height at one time is plotted against the height 8 minutes later, we see what looks like a very clear correlation; and indeed, the correlation coefficient estimated using (8.4) is 0.626. We used 1293 data values in finding this; the probability that $\rho$ would be this big in an uncorrelated Gaussian sample[5] is less than $10^{-180}$. Perhaps surprisingly, the distribution of the data is almost perfectly Gaussian.

Our second example of a geophysical data sequence is a series of values in space: the magnetic field measured on an aircraft flying 7 km above the south-eastern Pacific Ocean. Figure 8.4 shows the vertical component $Z$, and the horizontal components $X$, which is along the flight path. As with the lake-level data, we see an irregular line with a certain amount of order. A stochastic process seems like a good model, but there seems to be little evidence of a regular oscillation, or even an irregular one. A feature to notice here is that the two components appear to be related, varying together in some fashion: there is a lag and perhaps a suggestion of differentiation of $Z$ to obtain $X$. We will discuss later how to look at pairs of time series for evidence of this kind of relationship.

[5] Found using the $t$ test; $t_{N-2} = 28.8$.

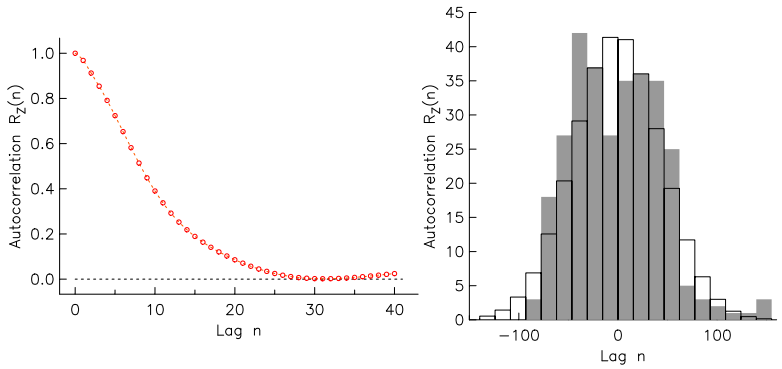



Figure 8.5: Left: estimated autocorrelation function for magnetic component $Z$. Right: pdf of $Z$ data (gray) and of a Gaussian distribution with the same mean and variance (black).

Concentrating for the moment on the $Z$ component, let us look at an estimate of the autocorrelation function, shown in Figure 8.5. We do not describe yet how that estimate was made; that will be the subject of a later lecture. For now notice how the $R_Z$ dies away monotonically from one. This means that neighboring values are likely to be very similar indeed, but as one separates two samples in space, their correlation fades away so that by a lag of 30 samples (about 100 km), they are uncorrelated. It is easy to believe this series could be generated by applying a suitable filter to white noise.

The histogram of the data shows something mentioned earlier: the magnetic anomaly data is somewhat non-Gaussian, being asymmetric, with a large positive tail and a compressed lower tail. A test for Gaussian behavior shows that Gaussian variables would depart this much from a true Gaussian about 18% of the time: not a resounding rejection of the Gaussian model, but a reason to be suspicious of it. The reason for this commonly observed behavior is not understood.

The final example (Figure 8.6) is a bathymetry profile across one side of the East Pacific Rise: another example of a spatial series, though this would be a true time series if we used the age of seafloor rather than distance. Here the data are very obviously not stationary, because the seafloor becomes deeper with age: as a realization of a stationary process, this is a miserable failure. But if we remove a average trend, (the line shown), we get a much more satisfactory-looking approximation to a stationary series, as we see from the gray line in Figure 8.6. Now in fact we should

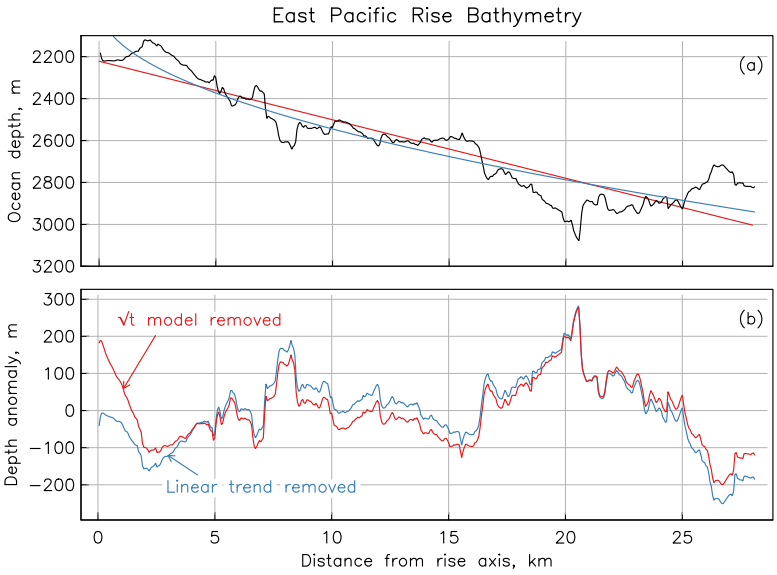



Figure 8.6: The upper plot shows bathymetry along a profile perpendicular to the East Pacific rise, along with a linear trend and a function where depth increases as $\sqrt{t}$. The bottom plot shows the "depth anomaly" found by removing these nonstationary features.

have removed, not a straight line but a parabola, because of the famous $\sqrt{t}$ approximation to depth changes caused by the cooling of the plate as it ages. The least-squares best-fitting parabolic curve is also shown; it fits the observations slightly better than the straight line does, and produces a residual that is the black curve in Figure 8.6.

Here we have done something very common (we saw another example in the sea-level data in Chapter 1), which is to model the observations as a steadily evolving part, plus a random, stationary part. Unusually in this example, we have a good model for the evolving piece of the model; normally we would just take a straight-line trend to represent that. The pdf of the depth anomaly is again not quite Gaussian – the distribution is heavy tailed, but not enough for a test to reject it as not being Gaussian.

APPENDIX A

# COMPUTING THE DISCRETE FOURIER TRANSFORM

## A.1 Introduction

As we noted in Chapter 3, the nominal form of the Discrete Fourier Transform (equation 3.8), which is

$$\tilde{x}_k = \sum_{n=0}^{N-1} x_n e^{2\pi i n k/N} \quad \text{for} \quad k = 0, \ldots N-1$$

does not well express how to actually perform the computation. First of all, as noted in that chapter, we would first compute the exponents using $e^{2\pi i n k/N} = e^{-2\pi i (nk)_{\mathrm{mod} N}/N}$; then the formula above would imply doing $N$ "operations" (additions and multiplications) to get a single $\tilde{x}_k$ – and we would then do this $N$ times to get all the DFT values, for a total of $N^2$ operations. The Fast Fourier Transform gives us the same result with many fewer operations.

## A.2 The Basis of the Fast Fourier Transform

The first discovery of an FFT algorithm was by Gauss in 1805 Heideman *et al.* (1985); like all the others before Cooley and Tukey's, it was not noticed. In fact, these algorithms are the same, at least if Gauss' procedure is updated to use complex exponentials. We now outline how it works.

The key to the whole procedure lies in assuming that $N$, the series length, be a composite integer, so that we can write $N = N_1 N_2$, where $N_1$ and $N_2$ are both integers. We will show how to compute the DFT in fewer operations for this case; and then we will assume that $N_1$ is also composite, apply the same procedure, and continue until we run out of factors. The simplest algorithm is for all the composite factors of $N$ to be the same size,

and as small as possible – which amounts to requiring that $N$ be a power of two.

We divide the $N$-length sequence of data $x_n$ into $N_2$ sequences of length $N_1$, and the $N$-length sequence of DFT values $\tilde{x}_k$ into $N_1$ sequences of length $N_2$. Then we re-index the subscripts for the two sequences as

$$\begin{aligned} n = N_2 n_1 + n_2 \quad &\text{and} \quad k = k_1 + N_1 k_2 \\ \text{where} n_1, k_1 = 0, \ldots N_1 - 1 \quad &\text{and} \quad n_2, k_2 = 0, \ldots N_2 - 1 \end{aligned}$$

We apply this indexing to the $e^{2\pi i n k/N}$ in the DFT expression:

$$e^{-2\pi i (k_1 n_1 N_2 + k_1 n_2)/N} e^{-2\pi i N_1 k_2 n_2/N} e^{-2\pi i k_2 n_1}$$

and note that the last exponential is always one. Then the full DFT expression becomes

$$\tilde{x}_{k_1 + N_1 k_2} = \sum_{n_2=0}^{N_2-1} \left[ \sum_{n_1=0}^{N_1-1} x_n e^{-2\pi i (k_1 n_1 N_2 + k_1 n_2)/N} \right] e^{-2\pi i N_1 k_2 n_2/N}$$

which we can write more compactly as

$$\tilde{x}_{k_1 + N_1 k_2} = \sum_{n_2=0}^{N_2-1} S_{k_1 n_2} e^{-2\pi i N_1 k_2 n_2/N}$$

where $S_{k_1 n_2}$ is the sum in brackets: we can view this as an $N_1$ by $N_2$ matrix, each of whose terms takes $N_1$ operations to compute.[1] Each of the columns of this matrix is just the result of performing a DFT on a sequence of length $N-1$ to get $N-1$ coefficients; So the total operations count to get all the elements of $S$ is $N_1(N_1 N_2) = N_1^2 N_2$. Then to get the final result requires the second (outer) sum, multiplying the $S$ matrix into the $N_2$-length vector represented by the final exponential to get a second vector of length $N_1$ – which we have to do for $N_2$ values of $k_2$. The operations count for each matrix multiplication is $N_1 N_2$; doing it $N_2$ times makes the total operations count for this step $N_1 N_2^2$. The total number of operations is the sum: $N_1^2 N_2 + N_2^2 N_1$ and remembering that $N = N_1 N_2$, this is

$$\frac{N^2}{N_2} + N \cdot N_2$$

[1] Actually $2N_1$ operations if we count both additions and multiplications as operations; but here and in the rest of the derivation we ignore expressions that do not depend on $N$.

which for $N$ large and $N_2$ small would be less than $N^2$; for $N_2 = 2$ it would be $\frac{1}{2}N^2 + 2N$.

But now suppose that $N_1$ is composite, with $N_1 = N_2 N_3$, which means that $N = N_3 N_2^2$. then we can decompose each of the $N_1$-length sequences in the same way, and use the same decomposition to compute each of the DFT's in $S$ more efficiently. The total number of operations then becomes, replacing the $N_1^2$ term in the above,

$$N_2(N_3^2 N_2 + N_2^2 N_3) + N_2^2 N_1 = \frac{N^2}{N_2^2} + 2N_2 N$$

And finally, if $N = N_2^q$ we can do this decomposition $q$ times, so that the total operations count is

$$\frac{N^2}{N_2^q} + qN_2 N = N(1 + qN_2)$$

Since $q = \log_{N_2} N$, the operations count scales as $N \log_{N_2}$, or for $N_2 = 2$, $N \log_2 N$: a much smaller number than $N^2$, even for $N$ as small as 1024, 100 times smaller.

## A.3 Computing DFT's of Real Series

If we are computing the DFT of a real-valued series, we can always reduce the computation time, and storage requirements, by a factor of two. The standard definition of the DFT, which is followed by almost all FFT programs, assumes that the sequence $x_n$ is complex. Transforming $N$ real numbers then means transforming $N$ complex numbers, all with imaginary parts equal to zero – and then half the output will be redundant, because the $\tilde{x}_n$'s will be Hermitian. While we could replace the DFT with a purely real transform, (such as the Hartley transform), this is not necessary. Instead, we represent our $N$ real numbers as $N/2$ complex numbers, take an $N/2$-length complex DFT, and rearrange the output to give the $N/2$ complex $\tilde{x}_k$'s that we want.

To do this, we combine adjacent pairs of real $x_n$'s to form our complex-valued sequence; we may not even have to do this explicitly, because in many computer languages (certainly FORTRAN), passing a real-valued sequence to a subroutine that expects complex numbers will automatically

cause this reassignment to occur. The DFT is then

$$C_k = \sum_{n=0}^{N/2-1} (x_{2n} + i x_{2n+1}) e^{-2\pi k n/M} \quad \text{for} \quad k = 0, \ldots M-1$$

where $M = N/2$. To see what to do with the $C_k$'s, construct two complex-valued sequences $A$ and $B$, defined by the sums

$$A_k = \sum_{n=0}^{M-1} x_{2n} e^{-2\pi i n k/M} \qquad B_k = \sum_{n=0}^{M-1} x_{2n+1} e^{-2\pi i n k/M}$$

which means that $C_k = A_k + iB_k$, If we could find the $A$ and $B$ sequences, we can compute the $X_k$ that we actually want, since

$$\begin{aligned} X_k &= \sum_{n=0}^{2M-1} x_n e^{-2\pi i n k/2M} = \sum_{n=0}^{M-1} x_{2n} e^{-2\pi i 2nk/2M} + \sum_{n=0}^{M-1} x_{2n} e^{-2\pi i (2n+1)k/2M} \\ &= A_k + e^{-\pi i k/N} B_k \quad \text{for} \quad k = 0, \ldots N/2-1 \end{aligned}$$

To get the $A$'s and $B$'s from the $C$'s that we have computed with out DFT, consider

$$C_{N-k} + C_k = A_{n-k} + A_k + i(B_{N-k} + B_k)$$

Sine $A$ and $B$ are both derived from real sequences, both are Hermitian, with $A_{N-k} = A_k^*$ and $B_{N-k} = B_k^*$. Therefore,

$$C_{N-k} + C_k = 2\mathscr{R}[A_k] + 2i\mathscr{R}[B_k]$$

which means that

$$\mathscr{R}[A_k] = \tfrac{1}{2}\mathscr{R}[C_{N-k} + C_k] \qquad \mathscr{R}[B_k] = \tfrac{1}{2}\mathscr{I}[C_{N-k} + C_k]$$

and similarly,

$$\mathscr{I}[A_k] = \tfrac{1}{2}\mathscr{I}[C_k - C_{N-k}] \qquad \mathscr{I}[B_k] = -\tfrac{1}{2}\mathscr{R}[C_k - C_{N-k}]$$

So, by computing first $C$ then $A$ and $B$, and finally $X$, we have achieved our aim. We can get slightly greater efficiency using DFT algorithms specifically designed to handle real sequences (or, for the inverse transform, Hermitian ones), though the improvements are not large enough to be important unless you plan to do a lot of transforms; Sorensen *et al.* (1987) give both algorithms and comparisons.

## A.4 Computing the Fourier Transform for a Few Frequencies

Finally, suppose we want to compute, instead of the DFT coefficients, the transform $\tilde{x}(f)$ for arbitrary $f$:

$$\tilde{x}(f) = \sum_{n=0}^{N-1} x_n e^{-2\pi i f n} \tag{A.1}$$

Certainly you might imagine (correctly) that even for the DFT case, where $f = k/N$, that the FFT will not be the most efficient procedure if we only want a few values of $f$. But again, we do not need to do the sum as written; by using a recursive method known as the **Goertzel algorithm** we need to compute only one trigonometric function for each frequency.

We define a series of sums:

$$U_k(f) = \frac{1}{\sin 2\pi f} \sum_{N-k}^{N-1} x_n \sin 2\pi f[n-(N-k)+1] \quad \text{for} \quad k = 1, \ldots N$$

and set $U_0 = U_{-1} = 0$. Then the difference between the final sum and the penultimate one, multiplied by $e^{2\pi i f}$ turns out to be the Fourier transform we seek:

$$\begin{aligned}
&U_N(f) - e^{2\pi i f} U_{N-1}(f) \\
&= \frac{1}{\sin 2\pi f}\left[\sum_{n=0}^{N-1} x_n \sin 2\pi(n+1)f - \cos 2\pi f \sum_{n=1}^{N-1} x_n \sin 2\pi n f\right] - i\sum_{n=1}^{N-1} x_n \sin 2\pi n f \\
&= x_0 + \frac{1}{sin 2\pi f}\left[\sum_{n=1}^{N-1} x_n(\sin 2\pi n f \cos 2\pi f + \cos 2\pi n f \sin 2\pi f - \cos 2\pi f \sin 2\pi n f)\right] \\
&- i\sum_{n=1}^{N-1} x_n \sin 2\pi n f \\
&= x_0 + \sum_{n=1}^{N-1} x_n(\cos 2\pi n f - i \sin 2\pi n f) = \sum_{n=1}^{N-1} x_n e^{-2\pi i n f}
\end{aligned}$$

The point of all this is that the we can compute the $U_k$'s recursively, and need to compute only one cosine, once, to do so. As above, we start with the

answer and show that it gives the result we wish. Take

$$\begin{aligned} & x_{N-k} + 2U_{k-1}\cos[2\pi f - U_{k-2}] \\ =& x_{N-k} + \frac{1}{\sin 2\pi f}\sum_{N-k+1}^{N-1} x_n[2\cos[2\pi f \sin 2\pi(n-N+k)f] \\ & - \sin[2\pi(n-N+k-1)f]] \end{aligned}$$

A general expression for the trigonometric functions is

$$2\cos u \sin[(m-1)u] - \sin[(m-2)u] = \sin^m u$$

which, applied to the sum in A.4, gives

$$\begin{aligned} & x_{N-k} + 2U_{k-1}\cos[2\pi f - U_{k-2}] \\ =& x_{N-k} + \frac{1}{\sin 2\pi f}\sum_{N-k+1}^{N-1} x_n \sin[2\pi(n-N+k+1)] \\ =& \frac{1}{\sin 2\pi f}\sum_{N-k}^{N-1} x_n \sin[2\pi(n-(N-k)+1)] \\ =& U_k \end{aligned}$$

a recursion that allows us to find $U_1, U_2 \ldots U_N$ in $N$ real multiplications and $2N$ real additions; and once we have the $U_{N-1}$ and $U_N$, we can find $\tilde{x}(f)$.

One difficulty with this approach is that, for values of $f$ near 0 or ½, it can be inaccurate because of roundoff (Gentleman, 1969). For example, for $f = 0$, the recursion becomes

$$\begin{aligned} U_1 &= x_{N-1} \\ U_2 &= x_{N-2} + 2x_{N-1} \\ U_3 &= x_{N-3} + 2x_{N-2} + 3x_{N-1} \\ &\cdots \end{aligned}$$

Finally, we would compute $\tilde{x}(0) = U_N - U_{N-1}$ by differencing two numbers that are potentially very large: this is a sure invitation to roundoff error. There is however a simple solution to this: define $\Delta U_k = U_k - U_{k-1}$. Then the recursion in terms of $\Delta U$ is

$$\begin{aligned} \Delta U_k = U_k - U_{k-1} &= x_{N-k} + 2(\cos 2\pi f - 1)U_{k-1} - U_{k-2} + U_{k-1} \\ &= x_{N-k} + 2(\cos 2\pi f - 1)U_{k-1} + \Delta U_{k-1} \end{aligned}$$

and the expression for $\tilde{x}(f)$ is

$$X(f) = \Delta U_N - (\cos 2\pi f - 1)U_{N-1} - i(\sin 2\pi f)U_{N-1}$$

Now, for $f$ small, $\cos 2\pi f - 1$ becomes small, and the recursion and its final processing become numerically stable.

There is a similar problem with the original recursion for $f$ close to ½; for $f = ½$ the original recursion is for which the original recursion gives

$$\begin{aligned} U_1 &= x_{N-1} \\ U_2 &= x_{N-2} - 2x_{N-1} \\ U_3 &= x_{N-3} - 2x_{N-2} - 3x_{N-1} \\ &\cdots \end{aligned}$$

which again means (potentially) differencing large numbers. The cure is again to define a new quantity (for which we use the same notation):

$$\Delta U_k = U_k + U_{k-1}$$

in terms of which the recursion becomes

$$\Delta U_k = x_{N-k} + 2(\cos 2\pi f + 1)U_{k-1} - \Delta U_{k-1}$$

and the final step becomes

$$X(f) = \Delta U_N - (\cos 2\pi f + 1)U_{N-1} - (-i(\sin 2\pi f)U_{N-1})$$

The need for three recursions complicates the process somewhat, but not so much so as to render this method unattractive for special purposes.

APPENDIX B

# OTHER READING

## B.1 Introduction

There are a many books that cover different aspects of this course, but none that provide the mix we think most useful. Most books tend to either focus on the statistical viewpoint, without much discussion either of Fourier theory or of actual algorithms; or else focus on the signal-processing topics with little discussion of randomness. The list here is certainly not complete, but includes some of our own favorites, both textbooks and reference monographs.

## B.2 Linear Systems and Fourier Analysis

Pippard (1978) gives a particularly detailed and readable discussion of one class of linear system: the damped oscillator (or sets of oscillators), including a number of interesting ways of looking at the Fourier transform. Books on the Fourier transform differ substantially in their level of mathematical rigor. The two least rigorous are James (1995), which is at about the level of these notes, and Bracewell (1986), which is much more complete, and is probably the best book on Fourier theory for non-mathematicians. It emphasizes getting a "feel" for transforms, being able to visualize the connection between the time and frequency domains, and the ubiquity of Fourier transforms, as shown by a wide range of examples (most from electrical engineering). Editions after the first one are basically unchanged, except for a discussion of the Hartley transform, about which the author is, in our view, overenthusiastic.

Moving up the scale of rigor, Champeney (1987) covers the Fourier theorems rigorously, with due attention to what kinds of functions they apply to and what the integrals really mean, though results are given rather than proved. Korner (1988) is a collection of anecdotes, theorems and applica-

tions, mostly to do with Fourier analysis; many of the essays can be read in isolation from the rest of the book. It provides an entertaining look at some mathematical culture. Lighthill (1958) is a brief, elegant, readable, and rigorous exposition of Fourier analysis and its connection with generalized functions; if you want to see the proofs done right, but readably, this is the place to look to fill in many of the details we have skipped. It is not, however, complete: there is no discussion of convolution. Finally, for the mathematically inclined Dym and McKean (1972) treats the subject with modern notation, and full rigor but in a lively style. They show lots and lots of unexpected applications ranging from differential equations to the Prime Number Theorem. They do not make any use of generalized functions.

## B.3 Digital Signal Processing

Because digital signal processing is, now, used inside devices ranging from cell phones to satellites to DVD players and washing machines, it has a huge literature on it; and because this subject is a standard course for electrical engineers, many authors have tried their hands at textbooks. Perusal of the relevant classification number (TK5102.9) in the library catalog will reveal dozens of books, all with similar content, though with levels varying from introductions for computer musicians whose mathematical background may be quite modest, to detailed treatments for graduate-level engineers. Many of these books address issues of finite word length and computational efficiency that, while of considerable engineering importance, are rarely relevant to data analysis. Many books also include a section on spectrum analysis, but usually without doing justice to the statistical questions involved.

Given this abundance, we can only offer a selection. Two basic introductions are Scherbaum (2001), which is oriented towards seismology but not very complete, and Steiglitz (1996), which is oriented towards people doing computer music – and thus avoids the assumption of many of these books, which is that you are an electrical engineer. Another in this class is Hamming (1983), which is very readable, and pays lots of attention to common stumbling blocks. It is however somewhat idiosyncratic in its methods and its coverage. One particular strong point is the especially good treatment of what some common operations (e.g. numerical integration) look like if viewed as filters.

Oppenheim and Schafer (1989) is a standard introductory textbook,

with probably all the information that you are likely to ever need, presented in a way that shows the authors' command of the subject and how to teach it. It is much more detailed than the books just mentioned; and it does assume some familiarity with continuous-time theory and (in places) an electrical-engineering background. The notation will be familiar, since the notation in our class notes comes from an earlier version of this text.

Signal processing as done in exploration geophysics, has very different goals and standards from the rest of the field. Robinson and Treitel (1980), by two of the founders of the subject, begins at a very basic level, but can be quite advanced in the later chapters, which tend to focus on special topics the authors happen to be interested in.

## B.4 Time Series and Spectral Estimation

Jenkins and Watts (1968) was long the standard text in the field, but it is now showing its age in terms of estimation methods (the theory is still perfectly correct). It is probably still one of the best discussions of multivariate methods. Chapter Four, on statistical inference, is still well worth reading. Priestley (1981) provides a careful but readable treatment of all the standard material with a nice balance between proof, discussion and illustration. This book focuses on the statistical issues, and includes some discussion of time-domain estimation. Bendat and Piersol (1986) has a strong emphasis on estimation, particularly of response functions for linear systems. The orientation is towards engineering, not statistics.

Percival and Walden (1993) introduces univariate spectral analysis at a reasonably high level. They cover stochastic processes, filtering, Fourier theory, and spectral estimation, with special emphasis on direct multitaper estimation techniques. They also cover parametric spectral estimation and harmonic analysis. A promised volume on cross spectra has not yet appeared. This is probably the best single book on univariate spectral estimation, though it is not always easy going; the authors have emulated the inventor of multitaper methods, Dave Thomson, in the lavish use of subscripts and superscripts, which tends to obscure the discussion.

# Bibliography


Agnew, D. C., and K. M. Hodgkinson (2007), Designing compact causal digital filters for low-frequency strainmeter data, *Bull. Seismol. Soc. Amer.*, **97**, 91–99.

Alsop, L. E., and A. A. Nowroozi (1966), Faster Fourier analysis, *J. Geophys. Res.*, **71**, 5482–5483, doi:10.1029/JZ071i022p05482.

Ardhuin, F., B. Chapron, and F. Collard (2009), Observation of swell dissipation across oceans, *Geophys. Res. Lett.*, **36**, L06,607, doi: 10.1029/2008GL037030.

Bendat, J. S., and A. G. Piersol (1986), *Random Data: Analysis and Measurement Procedures*, John Wiley, New York.

Bracewell, R. (1986), *The Fourier Transform and its Applications*, McGraw-Hill, New York.

Champeney, D. C. (1987), *A Handbook of Fourier Theorems*, Cambridge University Press), New York.

Cooley, J. W., and J. W. Tukey (1965), An algorithm for the machine computation of complex Fourier series, *Math, Comp.*, **19**, 297–301, doi: 10.2307/2003354.

Deakin, M. A. B. (1992), The ascendancy of the Laplace Transform and how it came about, *Arch. Hist. Exact Sci*, **44**, 265–286.

Dym, H., and H. P. McKean (1972), *Fourier Series and Integrals*, Academic Press, New York.

Gentleman, W. M. (1969), An error analysis of Goertzel's (Watt's) algorithm, *Computer J.*, **12**, 160–165.

Gross, R. (1992), Correspondence between theory and observations of polar motion, *Geophys. J. Internat.*, **109**, 162–170.

Hamming, R. W. (1983), *Digital Filters*, Prentice-Hall, Englewood Cliffs.

Harris, F. J. (1976), On the use of windows for harmonic analysis with the discrete Fourier transform, *IEEE Proc.*, **66**, 51–83.

Heideman, M. T., D. H. Johnson, and C. S. Burrus (1985), Gauss and the history of the Fast Fourier Transform, *Arch. Hist. Exact. Sci.*, **34**, 265–277.

Hewitt, E., and R. Hewitt (1979), The Gibbs-Wilbraham phenomenon: an episode in Fourier analysis, *Arch. Hist. Exact Sci.*, **21**, 129–169.

James, J. F. (1995), *A Student's Guide to Fourier Transforms: With Applications in Physics and Engineering*, Cambridge University Press), New York.

Jenkins, G. M., and D. G. Watts (1968), *Spectral Analysis and its Applications*, 525 pp., Holden-Day, San Francisco.

Kaiser, J. F., and W. A. Reed (1977), Data smoothing using low-pass digital filters, *Rev. Sci. Instrum.*, **48**, 1447–1454.

Kaiser, J. F., and W. A. Reed (1978), Bandpass (bandstop) digital filter design routine„ *Rev. Sci. Instrum.*, **49**, 1103–1106.

Korner, T. W. (1988), *Fourier Analysis*, Cambridge University Press, Cambridge.

Lighthill, M. J. (1958), *Fourier Analysis and Generalized Functions*, Cambridge University Press, Cambridge.

Munk, W. H., and G. McDonald (1960), *The Rotation of the Earth: a Geophysical Discussion*, Cambridge University Press, Cambridge.

Oppenheim, A. V., and R. W. Schafer (1989), *Discrete-time Signal Processing*, Prentice Hall, Englewood Cliffs, N.J.

Orchard, H. J., and A. N. Willson (2003), On the computation of a minimum-phase spectral factor, *IEEE Trans. Circuits Systems I*, **50**, 365–375.

Parker, P. R., M. A. Zumberge, and R. L. Parker (1995), A new method of fringe-signal processing in absolute gravity meters, *Manuscrip. Geodet.*, **20**, 173–181.

Percival, D. B., and A. T. Walden (1993), *Spectral Analysis for Physical Applications: Multitaper and Conventional Univariate Techniques*, Cambridge University Press, Cambridge.

Pippard, A. B. (1978), *The Physics of Vibration*, Cambridge University Press, New York.

Priestley, M. B. (1981), *Spectral Analysis and Time Series*, 890 pp., Academic, Orlando, Fla.

Robinson, E. A., and S. Treitel (1980), *Geophysical Signal Analysis*, Prentice Hall, Englewood Cliffs, N.J.

Scherbaum, F. (2001), *Of Poles and Zeros: Fundamentals of Digital Seismology*, Kluwer Academic Publishers, Boston.

Scherbaum, F., and M.-P. Bouin (1997), FIR filter effects and nucleation phases, *Geophys. J. Internat.*, **130**, 661–668.

Singleton, R. C. (1969), An algorithm for computing the mixed radix fast Fourier transform, *IEEE Trans. Audio Electroacous.*, **AU-17**, 93–103.

Slepian, D. (1976), On bandwidth, *Proc. IEEE*, **6**, 292–300.

Smith, M. L., and F. A. Dahlen (1981), The period and Q of the Chandler wobble, *Geophys. J. R. Astron. Soc.*, **64**, 223–281.

Snodgrass, F. E., G. W. Groves, K. F. Hasselmann, G. R. Miller, W. H. Munk, and W. H. Powers (1966), Propagation of ocean swell across the Pacific, *Phil. Trans. Roy. Soc., Ser. A*, **259**, 431–497.

Sorensen, H., D. L. Jones, M. T. Heideman, and C. S. Burrus (1987), Real-valued Fast Fourier Transform algorithms, *IEEE Trans. Acoustics Speech Sign. Proc.*, **35**, 849–863.

Steiglitz, K. (1996), *A DSP Primer: with Applications to Digital Audio and Computer Music*, Addison-Wesley, Menlo Park.

Steiglitz, K., T. W. Parks, and J. F. Kaiser (1992), METEOR: A constraint-based FIR filter design program, *IEEE Trans. Signal Proc.*, **40**, 1901–1909.

# Index

Amplitude of complex number, *see* Complex numbers
Argand diagram, *see* Phasor diagram

Complex numbers
    amplitude, 11
    phase, 11

Decibels, 5

Fechner's law, 5
Fourier series, 34
Fourier transform
    symmetry relations, 22

Linear systems, 9
    definition, 9
    time-invariant, 9

Phase
    of complex number, *see* Complex numbers
    sign convention for, 12
Phasor diagram, 11

Quadrature, defined, 11

Sea level
    at SIO pier, 1
    steric effect, 1
    swell, 2
Swell, propagation, 3
Symmetry relations of Fourier transform, *see* Fourier transform
Systems, *see* Linear systems

Time invariance, *see* Linear systems, time-invariant

Vibration diagram, *see* Phasor diagram