Time Series Analysis

Richard C. Aster and Brian Borchers[1]

July 30, 2021

Preface

This online textbook arose out of a general time series and data processing course for physical scientists and engineers that was taught, variablly, either jointly and individually by us at New Mexico Tech beginning in the early 1990s. The aim of this work is to provide a concise and fundamental background in time series (and on occasion, spatial data) along with some associated analysis concepts and linear methods. The level at which the subjects are addressed assumes a basic facility with complex numbers, calculus, and trigonometry, and is thus particularly suitable for mid-level to advanced undergraduates through early graduate students. The (certainly highly incomplete) set of topics from the enormous general field of series analysis that we cover here includes linear processes, Fourier methods, characterizaton and analysis of random processes, Kalman filtering, and autoregressive moving average (ARMA) modeling. Some of these topics necessarily overlap with those covered in our textbook written with Cliff Thurber, *Parameter Estimation and Inverse Problems* [2], and we refer readers to that reference for a complementary treatment of estimation and inverse topics presented at a similar level.

We sincerely hope that students studying and colleagues working with these topics across many disciplines will find this textbook to be useful in a variety of educational and reference settings.

Rick Aster

Department of Geosciences, Warner College of Natural Resources, Colorado State University, Fort Collins, CO USA

Brian Borchers

Department of Mathematics, New Mexico Institute of Mining and Techonology, Socorro, NM USA

August 2020

Contents

Chapter 1

Linear Time Invariant Systems

Introduction to Linear Systems, Part 1: The Time Domain

Our primary goal is to understand methods of analyzing temporal and spatial series, especially as applied to *linear systems*, both in continuous and sampled (discrete) time, and to demonstrate applications to important problems in geophysics and other physical sciences. Most of the examples worked here were done using MATLAB, and we will refer to this software from time to time.

We will be mostly concerned with an important class of physical situations that can be adequately characterized by *linear systems*. A linear system is a functional transformation, ϕ, which converts an *input* signal, $x(t)$ to an *output* signal, $y(t)$

$$y(t) = \phi[x(t)] \tag{1.1}$$

and which follows the principles of superposition

$$\phi[x(t) + y(t)] = \phi[x(t)] + \phi[y(t)] \tag{1.2}$$

and amplitude scaling

$$\phi[\alpha x(t)] = \alpha\phi[x(t)] \tag{1.3}$$

where α is a scalar. Note that for positive integer values of α (1.3) is equivalent to (1.2). (1.3) also implies that the output of the system is zero when there is no input

$$\phi[0] = 0 \ . \tag{1.4}$$

Many of the phenomena which we wish to study in geophysics and other areas of science are linear. Sometimes we study very weak perturbations to a physical system (e.g., small gravity variations, seismic disturbances far away from the source; effects due to small fluctuations in the magnetic field) and the

linear approximation is valid because the system is not tweaked very far from equilibrium. Common situations where linearity does not hold up are generally instances of large amplitude (e.g., high strain elastic waves near an underground nuclear explosion or earthquake; ocean waves breaking at a shoreline). In these cases the physics of the problem depends strongly on the amplitude of the perturbation, so that superposition (1.2) and scaling (1.3) do not hold (and are not even acceptable approximations).

Many interesting systems are also *time-invariant*, i.e., the functionality of ϕ is not time dependent. In some situations, of course we intentionally look for gradual time variations in a system response, but these usually take place on time scales greater than the duration of our signals of interest. For example, earthquake prediction researchers hope that this is not the case for some aspect of evolving earth response in an incipient main shock region.

A linear system is said to be *causal* if the output at time t_0 depends only on values of the input for $t \leq t_0$. Note that all physical processes are causal (as acausal systems propagate information backwards in time!). It is very easy mathematically, to construct non-causal mathematical systems, and these formulations may be useful in processing stored information. Also keep in mind that physical spatial phenomena (e.g. spatial filters) need not obey "causality" constraints.

A linear system is said to be *stable* if every non-infinite input produces a non-infinite output. While obvious for systems in the physical world (which will become non-linear in some manner rather than produce an infinite output) stability is important consideration in mathematical models of active systems (i.e., systems that have feedback between output and input).

The simple rules defining linear systems provide far-ranging and very useful constraints on the mathematical characterization of the system. Most importantly, linear systems are especially tractable, and very useful analysis tools, embodied in *Fourier Theory* describes their behavior complementary domains of time and frequency.

It may at first appear remarkable that the input to output transformation of *any* linear, time-invariant system can be characterized by a general integral relation (a *convolution*). To derive this result, we must first define the *Dirac delta* or *impulse function*. The delta function is discontinuous; it is nonzero only exactly where its argument is zero, where it is infinite. One way of conceptualizing the delta function (and to make it mathematically rigorous) is to define it as a limiting set of functions. One definition (e.g., Bracewell) is:

$$\delta(t) = \lim_{\tau \to 0} \ \tau^{-1} \Pi(t/\tau) \tag{1.5}$$

where $\tau^{-1}\Pi(t/\tau)$ is the unit-area rectangle or *boxcar* function of height τ^{-1} and width τ. The limit of 1.5 as τ approaches zero is an infinitesimally narrow pulse of infinite amplitude centered on $t = 0$, and having unit area. It can be shown that one need not start with the rectangle function to obtain the same functional limit, we could just as easily have considered a limit of any set of unit-area functions (e.g., an appropriately scaled set of Gaussians). Although

the delta function may seem outrageously artificial, it actually has a plethora of analytical uses in the theory of physical and theoretical system behavior.

The usefulness of $\delta(t)$ in our present context arises from its *sifting property*, whereby it can retrieve a functional value at a particular argument from within an integral

$$\int_a^b f(t)\delta(t-t_0)dt = f(t_0) \tag{1.6}$$

$$= f(t_0) \quad a \leq t_0 \leq b \tag{1.7}$$

$$= 0 \quad \text{elsewhere} \tag{1.8}$$

for any $f(t)$ continuous at finite $t = t_0$.

The delta function is one of several related discontinuous functions which will be of use to us. Another is the *step function*

$$\mathrm{H}(t-t_0) \equiv \int_{-\infty}^{t} \delta(\tau - t_0)d\tau \tag{1.9}$$

which is 0 for $t < t_0$, 1 for $t > t_0$, and takes a discontinuous step at $t = t_0$. The step function is a useful mathematical construction for "turning on" a system at $t = t_0$.

We can define the *boxcar function*, $\Pi(t)$, and *sign function*, $\mathrm{sgn}(t)$, in terms of $\mathrm{H}(t)$

$$\Pi(t) = \mathrm{H}(t+1/2) - \mathrm{H}(t-1/2)\ . \tag{1.10}$$

$$\mathrm{sgn}(t) = \frac{|t|}{t} = 2\mathrm{H}(t) - 1\ . \tag{1.11}$$

$\mathrm{sgn}(t)$ is also sometimes referred to as the *signum* function.

The *impulse response* of a system is the output produced by an impulse function input

$$h(t) \equiv \phi[\delta(t)]\ . \tag{1.12}$$

We will now show the important result that the response of a linear, time-invariant system to an arbitrary input is characterizable as a convolution. First, note that any input signal, $f(t)$, can be written as a summation of impulse functions because of the sifting property (1.8) of the delta function

$$f(t) = \int_{-\infty}^{\infty} f(\tau)\delta(t-\tau)\ d\tau\ . \tag{1.13}$$

Thus, for a general linear system characterized by an operator, ϕ, the response, $g(t)$, to an arbitrary input, $f(t)$, is just that operator acting on (1.13)

$$g(t) = \phi[f(t)] = \phi\left[\int_{-\infty}^{\infty} f(\tau)\delta(t-\tau)d\tau\right] \tag{1.14}$$

or, from the definition of the integral,

$$g(t) = \phi\left[\lim_{\Delta\tau\to 0}\sum_{n=-\infty}^{\infty} f(\tau_n)\delta(t-\tau_n)\Delta\tau\right]\ . \tag{1.15}$$

If ϕ characterizes a linear process, we can move it inside of the summation using the scaling relation (1.3), where the $f(\tau_n)$ are now weights

$$g(t) = \lim_{\Delta\tau\to 0} \sum_{n=-\infty}^{\infty} f(\tau_n)\phi[\delta(t-\tau_n)]\Delta\tau \ . \tag{1.16}$$

Because $\phi[\delta(t-\tau_n)]$ is just the time-lagged impulse response, $h(t-\tau_n)$ (1.12), (1.16) defines the integral

$$g(t) = \int_{-\infty}^{\infty} f(\tau)h(t-\tau)d\tau \tag{1.17}$$

which is the *convolution* of $f(t)$ and $h(t)$, often written in shorthand as

$$g(t) = f(t) * h(t) \ . \tag{1.18}$$

Thus, convolution of a general input signal with an appropriate impulse response exactly describes the corresponding output signal for *any* linear system. An important observation regarding (1.17) in the context of a measuring device is that convolution describes the smearing action of a linear measurement tool of limited resolving power. A measurement apparatus which records signals from the outside world exactly would have a delta function impulse response (so that its output, given by the convolution of an impulse and the real world signal would exactly match the desired observable). To see this, note that (1.13) is itself a convolution; convolution with a delta function simply returns the input signal, shifted in time (delayed or advanced) by the delta function's origin time

$$f(t) * \delta(t-t_0) \ = \int_{-\infty}^{\infty} f(\tau)\delta(t-t_0-\tau)\ d\tau = f(t-t_0) \ . \tag{1.19}$$

As all functions can be thought of as continuous integral superpositions of delta functions (1.13) it is clear that a necessary and sufficient condition for system stability is that the impulse response be bounded for all t.

Convolution with a step function

$$\int_{-\infty}^{\infty} f(\tau)\mathrm{H}(t-\tau)\ d\tau = \int_{-\infty}^{\infty} f(\tau) \int_{-\infty}^{t} \delta(\xi-\tau)d\xi\ d\tau \tag{1.20}$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{t} f(\tau)\delta(\xi-\tau)d\xi\ d\tau \tag{1.21}$$

$$= \int_{-\infty}^{t} \int_{-\infty}^{\infty} f(\tau)\delta(\xi-\tau)\ d\tau d\xi = \int_{-\infty}^{t} f(\tau)\ d\tau \tag{1.22}$$

is the definite integral of f from $t=-\infty$ up to time t. Thus, while convolution with a delta function returns the system impulse response, convolution with a step function performs the definite integration operation.

$\delta(t)$ can usefully be regarded as the time derivative of $\mathrm{H}(t)$. The significance of convolution with the time derivative of $\delta(t)$ is left as an exercise.

Another useful function for the analysis of linear systems is the *sampling function* (Bracewell's *shah* function)

$$r\text{Ш}(rt) = \sum_{n=-\infty}^{\infty} r\delta(rt - n) \ . \tag{1.23}$$

Multiplication by $\text{Ш}(rt)$ produces a continuous time representation of a *sampled* time series, with nonzero weighted impulses at $t = (..., \ -2/r, \ -1/r, \ 0, \ 1/r, \ 2/r, \ ...)$, where the weights are the values of the original function at those points. r is referred to as the *sampling rate* (the additional factor of r in (1.23) is required to maintain unit-area delta functions). Sampled time series (not necessarily in one dimension, but frequently in 2 or more dimensions, and usually uniformly sampled in time or space) make up the vast majority of geophysical and other types of scientific data.

Time domain interpretation of convolution. A way to develop further insight into convolution is to graphically examine the operation of the convolution integral

$$c(t) = f_1(t) * f_2(t) = \int_{-\infty}^{\infty} f_1(\tau) f_2(t - \tau) \ d\tau \ . \tag{1.24}$$

The procedure is as follows:

1. Plot both $f_1(\tau)$ and $f_2(t-\tau)$ on the τ-axis. Note that this operation flips the function $f_2(\tau)$ about the τ-axis and shifts it by an amount t (which is the independent variable of the output function $c(t)$).

2. Visualize that as t advances, $f_2(t - \tau)$ slides along the τ-axis.

3. For each t, the convolution integral (1.24) gives the area of the product $f_1(\tau) \cdot f_2(t - \tau)$.

As an example, consider the convolution of $\Pi(t)$ (1.10) and a truncated exponential, $e^{-t}\text{H}(t)$.

$$c(t) = \int_{-\infty}^{\infty} \Pi(\tau)\text{H}(t - \tau) e^{-(t-\tau)} \ d\tau \ . \tag{1.25}$$

Because of the discontinuities in $\Pi(t)$, the solution is found by examining three cases:

- Case (a) $t \ \leq \ -1/2$

  The nonzero portions of the functions do not overlap, and $c(t) = 0$.

- Case (b) $-1/2 \ \leq \ t \ \leq 1/2$

  The sliding exponential partially overlaps the boxcar function. The appropriate integral is

$$c(t) = \int_{-1/2}^{t} 1 \cdot e^{-(t-\tau)} \ d\tau = 1 - e^{-(t+1/2)} \ . \tag{1.26}$$

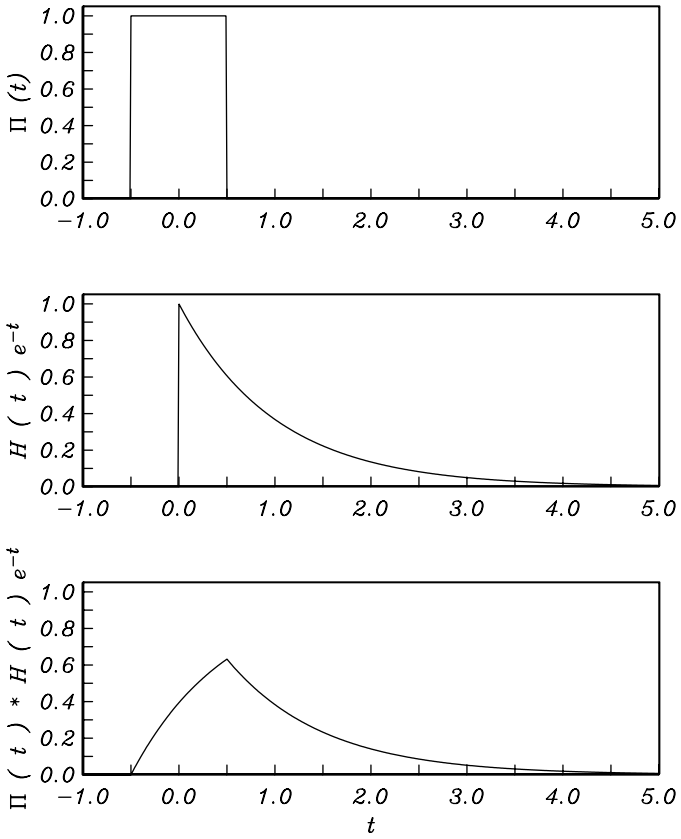

Figure 1.1: Convolution Example

- Case (c) $t \geq 1/2$

  The sliding exponential completely overlaps the boxcar function. The integral is

$$c(t) = \int_{-1/2}^{1/2} 1 \cdot e^{-(t-\tau)}\, d\tau = e^{-(t-1/2)} - e^{-(t+1/2)} \ . \tag{1.27}$$

The result of this convolution is plotted in Figure 1.1. Note that we could have equivalently written the convolution as

$$c(t) = \int_{-\infty}^{\infty} \Pi(t-\tau)\mathrm{H}(\tau)e^{-\tau}\, d\tau \ . \tag{1.28}$$

This produces the same answer with somewhat different integrals. A more efficient and elegant way of evaluating convolutions will become apparent after

we learn how to examine functions in the *frequency domain*, rather than the *time domain*.

Autocorrelation and cross-correlation. Several other integral operations, commonly used in time and spatial series analysis are closely related to convolution.

Autocorrelation is similar to *autoconvolution*

$$f(t) * f(t) = \int_{-\infty}^{\infty} f(\tau) f(t-\tau)\ d\tau \tag{1.29}$$

except that one of the functional components in the τ-domain is *not* reversed. The autocorrelation of a real function, $f(t)$, is

$$A(t) = \int_{-\infty}^{\infty} f(\xi) f(\xi - t)\ d\xi = \int_{\infty}^{-\infty} f(\xi - t) f(\xi)\ (-d\xi) \tag{1.30}$$

which is, if we let $\xi - t = -\tau$,

$$= \int_{-\infty}^{\infty} f(-\tau) f(t-\tau)\ d\tau = f(-t) * f(t) = f(t) * f(-t)\ . \tag{1.31}$$

If $f(t)$ is symmetric in time (an *even function*; $f(t) = f(-t)$), then the autoconvolution and autocorrelation are equal. Also, because the autocorrelation integral (1.31) is unchanged when we interchange $\pm t$, we see that autocorrelation always produces an even function.

It is often convenient to divide (1.31) by the signal energy to obtain a normalized autocorrelation form

$$a(t) = \frac{A(t)}{\int_{-\infty}^{\infty} f^2(\tau)\ d\tau}\ . \tag{1.32}$$

(1.32) is bounded on the interval $[-1,\ 1]$. Note that for (1.32) and (1.31) to converge, the *signal energy*

$$E = A(0) = \int_{-\infty}^{\infty} f^2(\tau)\ d\tau \tag{1.33}$$

must be finite. It is thus necessary for $f^2(t)$ to have finite area (zero mean alone is not sufficient).

The *cross-correlation* of two functions, $f_1(t)$ and $f_2(t)$ (often referred to simply as the *correlation*) is

$$C(t) = \int_{-\infty}^{\infty} f_1(\tau) f_2(\tau - t) d\tau = \int_{-\infty}^{\infty} f_1(\tau + t) f_2(\tau) d\tau \equiv f_1(t) \star f_2(t) \tag{1.34}$$

If (1.34) is divided by the *cross-signal energy* we have a normalized version of the cross-correlation corresponding to (1.31)

$$c(t) = \frac{C(t)}{\sqrt{\int_{-\infty}^{\infty} f_1^2(\tau) d\tau \cdot \int_{-\infty}^{\infty} f_2^2(\tau) d\tau}} \tag{1.35}$$

produces a value of one at zero time lag when the two functions are identical. Autocorrelation and correlation have important applications in power spectra, coherency, signal detection and timing, and array processing.

Correlations and Cross-Correlations in MATLAB. MATLAB has built in convolution *conv*, and cross-correlation (*xcorr*) time domain functions. The numerical part of MATLAB, of course, only operates on finite time series (or *sampled*) representations of functions stored as vectors or arrays of numbers which hopefully adequately represent a continuous function in nature (we will examine the issues associated with sampled functions in detail later in the course.). The *conv* function thus calculates a sample-by-sample moving dot-product rather than an integral. Note that, because these operations in MATLAB are simply vector products, you will have to scale the results by the sampling interval to get results that agree with continuous integral values. You are encouraged to experiment with these and other MATLAB functions. Note that if you have two MATLAB time series, a_1 and a_2, which are of length n_1 and n_2 samples, respectively, then the convolution output from *conv*, $a_1 * a_2$ will be of length $(n_1 + n_2 + 1)$.

Here is the MATLAB code that performs the above convolution example (Figure 1.1):

```
%MATLAB demonstration of example convolution in notes, part 1
%clear any old variables
clear

%total length of f1, f2 time series in seconds
N=10;

%time step size in seconds
dt=0.02;

%length of vectors to create
M=N/dt;

%zero time reference point
ztime=M/4;

%here is the boxcar function
%initialize f1
f1=zeros(M,1);
%insert ones into the correct elements
f1((ztime-0.5/dt):ztime+(0.5/dt))=ones(1+1/dt,1);

%here is the decaying exponential function (starting at zero time);
%initialize f2
f2=zeros(M,1);
```

```
%insert a decaying exponential into the correct elements
f2(ztime:M)=exp(-dt*(0:M-ztime));

%create the time axis vector for plotting f1 and f2
taxis = ((1:M)-ztime)*dt;

%plot f1
figure(1)
plot(taxis,f1)
grid
title('f_1(t)')
xlabel('time')

%plot f2
figure(2)
plot(taxis,f2)
grid
title('f_2(t)')
xlabel('time')

%do the convolution (normalized by dt to make the sum scale like the integral)
c=conv(f1,f2)*dt;

%create the time axis vector for the convolution (which has length(c)=2*M-1)
taxisc=((1:length(c))-2*ztime)*dt;

%plot the convolution
figure(3)
plot(taxisc,c)
grid
title('c(t)=f_1(t)*f_2(t)')
xlabel('time')
```

Chapter 2

Linear Time Invariant Systems in the Frequency Domain

Introduction to Linear Systems: The Frequency Domain

In Chapter 1, we examined signals in linear systems using time as the independent variable. We now address the fundamentals of Fourier theory, where the independent variable is the frequency of a continuum or discrete set of sinusoidal (or, equivalently, complex exponential) basis functions. The basic insight that leads to Fourier Theory is that linear systems, being subject to superposition and scaling, can be analyzed in terms of their *frequency response*, that is, in terms of their response to pure sinusoidal or exponential inputs.

Consider the response, $g(t)$ of a linear system with impulse response $\phi(t)$ to a unit-amplitude, complex input of frequency f, $e^{\imath 2\pi f t}$. The time domain response of any such system is given by the convolution of the input function and the impulse response

$$g(t) = \int_{-\infty}^{\infty} \phi(\tau) e^{\imath 2\pi f(t-\tau)}\, d\tau \ . \tag{2.1}$$

Because a time shift in the argument of an exponential is mathematically equivalent to multiplication by another exponential

$$g(t) = e^{\imath 2\pi f t} \int_{-\infty}^{\infty} \phi(\tau) e^{-\imath 2\pi f \tau}\, d\tau \ \equiv \ e^{\imath 2\pi f t} \cdot \Phi(f) \ . \tag{2.2}$$

(2.2) shows that the response of any linear system to a complex sinusoidal input is unchanged in functional form (a complex sinusoidal signal of the *same*

frequency) and is only modified in amplitude and phase (by the complex factor $\Phi(f)$). The frequency, f, in (2.2) is arbitrary. Thus, if an arbitrary input $\psi(t)$ is decomposed into a sum of sinusoidal components, then, because of superposition, the relationship between $\psi(t)$ and $g(t) = \psi(t) * \phi(t)$ can be completely characterized by $\Phi(f)$, the *transfer function* of the system. $\Phi(f)$ is the *Fourier transform* (or *spectrum*) of the impulse response of the system, $\phi(t)$.

There are several conventions that are variously used in defining the Fourier transform. The definitions that we will use are those most commonly encountered in geophysics

$$\Phi(f) = F[\phi(t)] \equiv \int_{-\infty}^{\infty} \phi(t) e^{-\imath 2\pi f t} dt \tag{2.3}$$

$$\phi(t) = F^{-1}[\Phi(f)] \equiv \int_{-\infty}^{\infty} \Phi(f) e^{\imath 2\pi f t} df \tag{2.4}$$

where F denotes the Fourier transform operation, and F^{-1} denotes the *inverse Fourier transform* operation. Be aware that in some other areas of physics and in exploration geophysics the sign convention on the complex exponentials of (2.3) and (2.4) is reversed, so that the forward transform has a plus sign in the exponent and the inverse transform has a minus sign in the exponent. This will of course not affect any fundamentals of the analysis, only the convention by which phase is measured. Other formulations use $\omega = 2\pi f$ rather than f to characterize the frequency. This introduces factors of 2π into the transform pair.

Differential equations and Fourier theory. A particularly tractable and not uncommon situation in the physical sciences occurs when a system relating two time functions, $x(t)$ and $y(t)$, is characterizable by a linear differential equation with constant coefficients. For functions of a single variable, t, the general form of such a differential equation is

$$a_n \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_1 \frac{dy}{dt} + a_0 y = b_m \frac{d^m x}{dt^m} + b_{m-1} \frac{d^{m-1} x}{dt^{m-1}} + \cdots + b_1 \frac{dx}{dt} + b_0 x \,. \tag{2.5}$$

As none of the coefficients (the a_i and b_i) depend on t, (2.5) describes a time-invariant system. Because all of the terms are linear (there are no powers or other nonlinear functions of x, y, or their derivatives), it is also a linear system, obeying superposition and scaling (note that differentiation itself is a linear operation). To obtain an expression for the transfer function corresponding to (2.5), substitute an exponential unit amplitude exponential of arbitrary frequency for the input, $x(t)$, and output, $y(t)$, so that

$$x(t) = e^{\imath 2\pi f t} \tag{2.6}$$

and, as must be the case for any linear, time-invariant system (2.2),

$$y(t) = \Phi(f) e^{\imath 2\pi f t} \,. \tag{2.7}$$

Substituting (2.6) and (2.7) into (2.5), dividing both sides by $e^{\imath 2\pi ft}$, and solving for $\Phi(f)$ produces the system transfer function, which is a ratio of two complex polynomials in f.

$$\Phi(f) = \frac{\sum_{j=0}^{m} b_j (2\pi\imath f)^j}{\sum_{k=0}^{n} a_k (2\pi\imath f)^k} \tag{2.8}$$

The values of f where the numerator is zero are referred to as *zeros* of $\Phi(f)$, as the response is zero at this frequency, regardless of the amplitude of the input signal. Conversely, frequencies for which the denominator is zero are called *poles*, as the response becomes very large at these frequencies. Note that we don't have to worry too much about any mysteries regarding $e^{\imath 2\pi ft}$ being a complex number, as

$$e^{\imath 2\pi ft} = \cos(2\pi ft) + \imath \sin(2\pi ft) \tag{2.9}$$

and we could almost have just as easily chosen to propagate the real or the imaginary part of the input signal alone through the system to reach an equivalent conclusion; in this case an input (cosine, sine) signal simply produces a scaled output (cosine, sin) with a phase shift. Note that frequencies for which we have zero or infinite response may be imaginary or complex, in which case the corresponding input function, (2.6) may be an increasing or decreasing exponential, or an increasing or decreasing exponentially damped sinusoid, respectively.

Example: Response of a seismometer. As an important example of such a linear system from geophysical instrumentation, consider (Figure 2.1) a damped vertical harmonic oscillator with a rigid case that is fixed to the Earth. A mass, M, is supported by a spring, in parallel with a damping or *dashpot* component that produces Newtonian damping (i.e., a retarding force that is proportional to velocity). Intuitively, it you may see that the motion of the mass relative to the Earth will provide some sort of representation of the true vertical ground motion. For example, if the mass were completely decoupled, so that it remained stationary in its inertial reference frame while the Earth moved, then the motion of the mass relative to its case (which is, recall, rigidly attached to the Earth) would be exactly the negative of the ground motion).

The differential equation of motion for the mass in such a *seismometer* can be obtained using Netwon's second law by equating the (upward) forces of the spring and damper acting on the mass with the (upward) acceleration times the mass.

$$F_{up} = Ma_{up} \tag{2.10}$$

or

$$-D\frac{d\xi(t)}{dt} + K[\xi_0 - \xi(t)] \; = M\frac{d^2}{dt^2}[\xi(t) + u(t)] \tag{2.11}$$

which gives rise to a homogeneous differential equation:

$$M\frac{d^2}{dt^2}[\xi(t) + u(t)] + D\frac{d\xi(t)}{dt} + K[\xi(t) - \xi_0] = 0 \; . \tag{2.12}$$

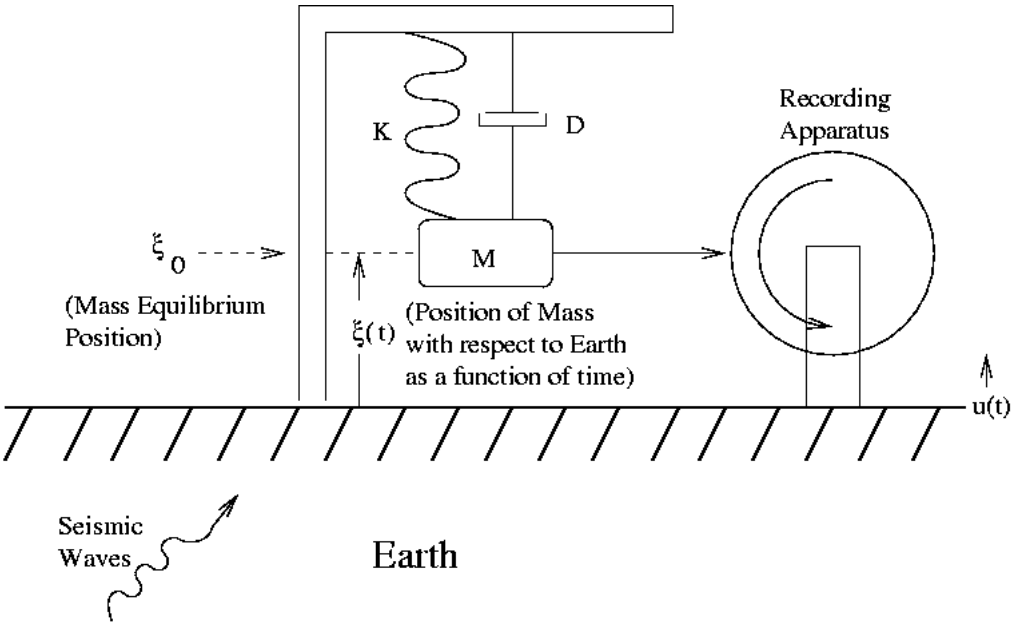

Figure 2.1: A Mechanical Seismometer

Here, u is the motion of the Earth (up positive), ξ is the position of the mass, which has an equilibrium position in the Earth's gravity field of ξ_0 (both measured up positive relative to the surface of the Earth), M is the mass of the inertial component, D is the dashpot constant (units of force per velocity), and K is the spring constant (units of force per distance).

We can simplify (2.12) somewhat by writing the equation of motion for the mass in an upward positive coordinate system (z) where $z = 0$ is the equilibrium position in the Earth's gravitational field, so that $z(t) = \xi(t) - \xi_0$. This gives

$$\ddot{z} + 2\zeta\dot{z} + \omega_s^2 z = -\ddot{u} \tag{2.13}$$

where the *damping coefficient* is

$$2\zeta \equiv D/M \tag{2.14}$$

and

$$\omega_s \equiv (K/M)^{1/2} \tag{2.15}$$

is the angular undamped or *natural* frequency of the system. (2.13) is a linear homogeneous equation where the input, u, is the displacement of the Earth, and the output, z, is the deviation of the mass from its equilibrium position, relative to the seismometer frame.

Using (2.8), we now write the transfer function of the seismometer system (seismometer displacement response to a displacement of the Earth)

$$\Phi(\omega) = \frac{z(\omega)}{u(\omega)} = \frac{-(\imath\omega)^2}{(\imath\omega)^2 + 2\zeta(\imath\omega) + \omega_s^2} = \frac{-\omega^2}{\omega^2 - 2\imath\zeta\omega - \omega_s^2} \tag{2.16}$$

or, in terms of amplitude and phase

$$|\Phi(\omega)| = \frac{\omega^2}{[(\omega^2 - \omega_s^2)^2 + 4\zeta^2\omega^2]^{1/2}} \tag{2.17}$$

$$\theta = \arg[\Phi(\omega)] = \pi - \tan^{-1}\frac{-2\zeta\omega}{\omega^2 - \omega_s^2} \ . \tag{2.18}$$

At high frequencies ($\omega \gg \omega_s$), $|\Phi(\omega)| \approx 1$, and $\theta \approx \pi$, so the seismometer displacement from equilibrium is the negative of the Earth displacement, $z \approx -u$. In this case, the Earth moves so rapidly that the mass cannot follow the motion at all, and the position of the mass relative to the frame is thus just $-u$.

At low frequencies ($\omega \ll \omega_s$), $|\phi(\omega)| \approx \omega^2/\omega_s^2$, so that response amplitude falls off quadratically with decreased frequency. From the time domain representation (2.13), we see that this response is proportional to the negative of the Earth's acceleration, $z \propto -\ddot{u}$.

The mechanical seismometer, in displacement, thus acts like a displacement sensor at high frequencies and as an accelerometer at low frequencies. Around $\omega = \omega_s$, the system undergoes a transition between these two end-member behaviors. One can already see why very low frequency natural frequencies are desirable for seismometers; if ω_s is very small, the true displacement of the Earth is recoverable directly from the instrument response.

The frequency response for displacement input and displacement output [(2.17) and (2.18)] is plotted in Figure 2.2 for various damping factors, where the complex response is plotted in terms of its amplitude and phase.

In examining Figure 2.2, first consider the amplitude response when the damping, ζ, is small relative to ω_s In this case the system exhibits a large amplitude response for input frequencies near ω_s. This occurs because the system is excited near its natural resonant frequency and there is little energy loss via the dashpot. When ζ becomes larger than ω_s, the resonance peak in the amplitude response disappears, and the system no longer oscillates freely.

Next consider the phase response. At the undamped resonance period, the phase is $-90°$, implying that the output is phase-shifted by that amount (by -$\pi/2$ radians) relative to the input. A cosine Earth motion of frequency ω_s would be phase shifted into a sine mass displacement. Regardless of damping, the phase shift approaches zero at low frequencies and approaches π (a factor of -1) at high frequencies.

Purely mechanical seismometers such as that described above were among the first such instruments used to record accurate ground motion from earthquakes or other sources (they were first widely deployed starting in the 1890's). In most modern seismometers mass motion is sensed as a voltage which is proportional to the velocity of the mass using an inductive coil and magnetic field, a

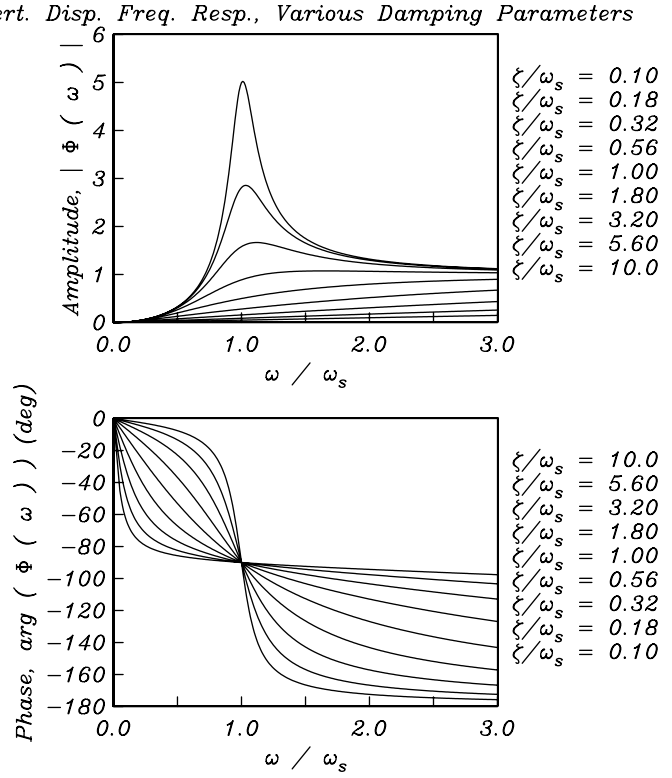

Figure 2.2: Frequency Response of the Mechanical Seismometer

method pioneered by Prince Boris Galitzin of Russia around 1906. If the mass motion is small, the induction circuit is linear and, as a bonus, the induced current in the inductive coil produces an electromagnetic force that counteracts the motion of the mass and thus provides predictable and stable damping. In the electromagnetic seismometer the output is a voltage that is proportional to the velocity, $\dot{z}$, of the mass relative to its frame (or case), and is thus the time derivative of the displacement response. The system response of a differentiator, which is characterized by the differential equation

$$y(t) = \dot{x}(t) \ , \tag{2.19}$$

can be trivially seen (2.8) to be just $i\omega$, so that the transfer function of an inductive seismometer system as voltage out versus Earth displacement is

$$\Phi_{induction}(\omega) = \frac{\dot{z}(\omega)}{u(\omega)} = \frac{-\imath\omega^3}{\omega^2 - 2\imath\zeta\omega - \omega_s^2} \ . \tag{2.20}$$

Note that if we consider the Earth velocity, $\dot{u}$ instead of the Earth displacement, u as the input signal the response of the inductive seismometer is

$$\Phi_{induction}(\omega) = \frac{\dot{z}(\omega)}{\dot{u}(\omega)} = \frac{-\omega^2}{\omega^2 - 2\imath\zeta\omega - \omega_s^2} \tag{2.21}$$

which is identical to (2.16), and the same response discussion as above applies, except that the output is in volts for a ground velocity input rather than output displacement for ground displacement. For this reason, such seismometers are sometimes referred to as *velocimeters*.

The inverse Fourier transform of a response function $\Phi(\omega)$ will give the time domain impulse response of the system. The following conditions are sufficient for existence of a Fourier transform:

1. $\phi(t)$ has only a finite number of maxima and minima in any finite time interval. This eliminates very wiggly functions (e.g., $sin(1/x)$).

2. $\phi(t)$ has only a finite number of finite discontinuities in any finite time interval. Pathological functions such as 1 where the argument is rational and 0 where the argument is irrational won't work.

3. $\phi(t)$ is has finite "energy", so that

$$\int_{-\infty}^{\infty} |\phi(t)|^2 \ dt \tag{2.22}$$

   is bounded.

There are useful functions that do not satisfy (2.22), yet still have Fourier transforms (such transforms will have delta or other discontinuous functional components). Clearly, for example, (2.22) is not satisfied for the displacement transfer function (2.16) in the seismometer system. It is a little easier to obtain

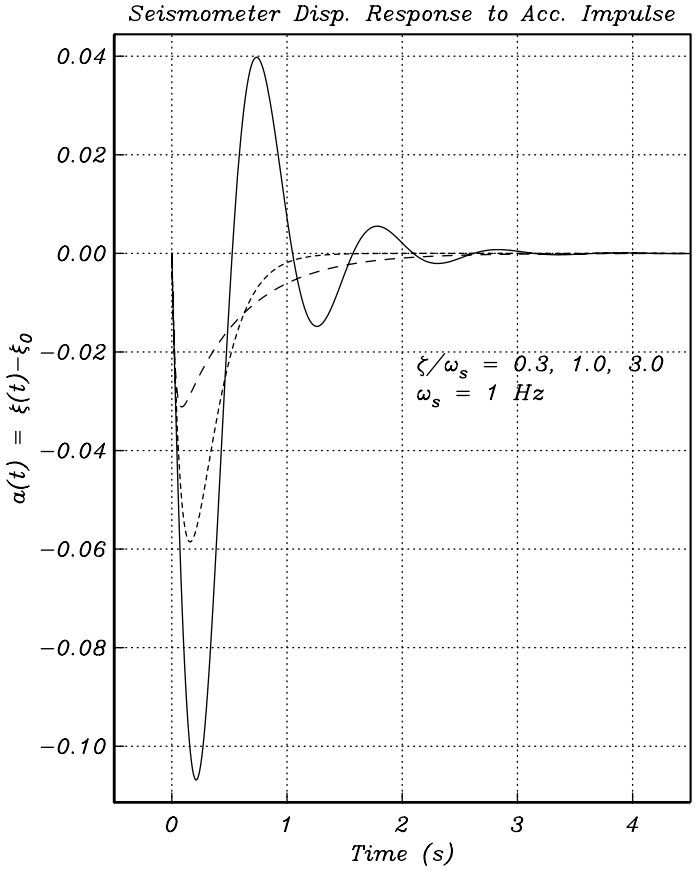

Figure 2.3: Response of the Mechanical Seismometer to an Acceleration Impulse

the displacement response to an impulsive Earth acceleration ($\ddot{u} = \delta(t)$) by the inverse Fourier transform method by solving

$$\ddot{a} + 2\zeta\dot{a} + \omega_s^2 a = -\delta(t) \tag{2.23}$$

which is shown in Figure 2.3 (we'll do the detailed calculation, and solve for the displacement response to Earth displacement later).

Energy in the Time and Frequency Domains; Parseval's theorem. The inverse Fourier transform says that time domain signals can be expressed as an infinite summation of complex exponentials. We might therefore expect a simple relationship between signal energy expressed in the time and frequency domains. Consider the total energy in a real (or complex) time domain signal, $\phi(t)$

$$E = \int_{-\infty}^{\infty} \phi(t)\phi^*(t)\ dt \tag{2.24}$$

where the asterisk denotes complex conjugation (which has no effect if $\phi(t)$ is real). Invoking (2.4), this can be written as

$$E = \int_{-\infty}^{\infty} \phi(t) \left(\int_{-\infty}^{\infty} \Phi^*(f) e^{-\imath 2\pi f t} df \right) \, dt \; . \tag{2.25}$$

Interchanging the order of integration, we get

$$E = \int_{-\infty}^{\infty} \Phi^*(f) \left(\int_{-\infty}^{\infty} \phi(t) e^{-\imath 2\pi f t} dt \right) \, df \tag{2.26}$$

which gives

$$E = \int_{-\infty}^{\infty} \Phi^*(f)\Phi(f) \; df = \int_{-\infty}^{\infty} \phi(t)\phi^*(t) \; dt \; . \tag{2.27}$$

Equation (2.27) is variously referred to as *Parseval's*, *Rayleigh's* or *Plancherel's* theorem. It says that one can evaluate the energy in a signal as either an integral of its amplitude squared time domain representation over all time, or as an integral across all of its amplitude squared frequency components over all frequencies. In a more general sense, Parseval's theorem says that the Fourier transform is *length preserving*, i.e., the "size" of the function (in the size-sense of the integral of the amplitude squared) is the same in the time and frequency domains.

Properties of the Fourier transform. We next consider the Fourier transforms of some canonical functions and discuss general symmetries and other properties. An important function in time series analysis which we saw in Chapter 1 is the boxcar function, $\Pi(t)$. The Fourier transform of the boxcar function is (Figure 2.4)

$$F[\Pi(t)] = \int_{-\infty}^{\infty} \Pi(t) e^{-\imath 2\pi f t} \; dt \tag{2.28}$$

$$= \int_{-1/2}^{1/2} e^{-\imath 2\pi f t} dt = \int_{-1/2}^{1/2} \cos(2\pi f t) \; dt \tag{2.29}$$

$$= \frac{\sin(\pi f)}{\pi f} \; \equiv \; \mathrm{sinc}(f) \; . \tag{2.30}$$

The corresponding inverse transform is thus

$$F^{-1}[\mathrm{sinc}(f)] = \int_{-\infty}^{\infty} \mathrm{sinc}(f) e^{\imath 2\pi f t} df = \Pi(t) \; . \tag{2.31}$$

Taking the complex conjugate and interchanging f and t, gives us the Fourier transform of $\mathrm{sinc}(t)$

$$\Pi(f) = \int_{-\infty}^{\infty} \mathrm{sinc}(t) e^{-\imath 2\pi f t} dt \; . \tag{2.32}$$

Note that (2.30) and (2.31) show, perhaps surprisingly, that we can get discontinuous functions by the integration smooth functions.

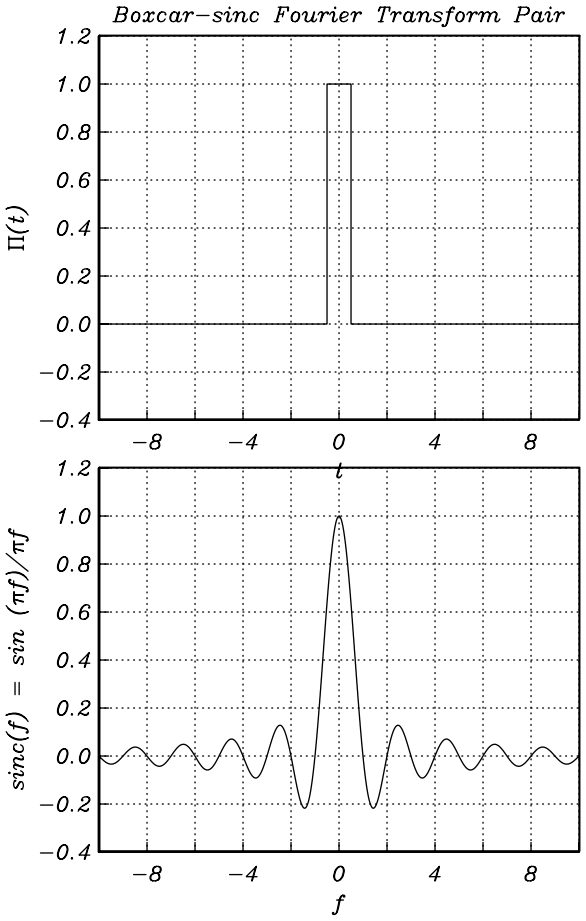

Figure 2.4: The Boxcar-Sinc Fourier Transform Pair

The Fourier transform of a delta function is easily seen to be

$$F[\delta(t)] = \int_{-\infty}^{\infty} \delta(t) e^{-\imath 2\pi f t} dt = 1 \ . \tag{2.33}$$

So a delta function can be thought of as consisting of an equal weighting of $e^{-\imath 2\pi f t}$ functions across *all* frequencies, with no relative phase shifts. Going the other direction, from the frequency to the time domain, gives

$$F^{-1}(1) = \int_{-\infty}^{\infty} e^{\imath 2\pi f t} df = \delta(t) \ . \tag{2.34}$$

One way to grasp (2.34) is to imagine the oscillating terms of the integrand all averaging out to zero, except exactly at $t = 0$, where they all have value one and will reinforce each other, i.e.,

$$F^{-1}(1) = \lim_{\epsilon \to 0} \int_{-\infty}^{-\epsilon} e^{\imath 2\pi f t} df + \int_{\epsilon}^{\infty} e^{\imath 2\pi f t} \ df = 2 \lim_{\epsilon \to 0} \int_{\epsilon}^{\infty} \cos(2\pi f t) \ df \ . \tag{2.35}$$

A very useful property of the Fourier transform is the *shifting property*; a simple time shift of a function only changes the phase (not the amplitude) of its Fourier transform. Consider the Fourier transform of a general function

$$F[\phi(t - t_0)] = \int_{-\infty}^{\infty} \phi(t - t_0) e^{-\imath 2\pi f t} dt \ . \tag{2.36}$$

Substituting $\tau = t - t_0$ gives

$$= \int_{-\infty}^{\infty} \phi(\tau) e^{-\imath 2\pi f (\tau + t_0)} \ d\tau = e^{-\imath 2\pi f t_0} \int_{-\infty}^{\infty} \phi(\tau) e^{-\imath 2\pi f \tau} \ d\tau \tag{2.37}$$

$$= e^{-\imath 2\pi f t_0} \Phi(f) \tag{2.38}$$

so that time shifts in the time domain correspond to linear (with respect to frequency) phase shifts in the frequency domain.

Another important relationship is *time-frequency scaling* or *similarity*, consider

$$F[\phi(\alpha t)] = \int_{-\infty}^{\infty} \phi(\alpha t) e^{-\imath 2\pi f t} dt \ . \tag{2.39}$$

For $\alpha > 0$, this gives

$$= \frac{1}{\alpha} \int_{-\infty}^{\infty} \phi(\tau) e^{-\imath 2\pi f \tau / \alpha} d\tau = \frac{1}{\alpha} \Phi \left(\frac{f}{\alpha} \right) \ , \tag{2.40}$$

using the substitution $\tau = \alpha t$. For $\alpha < 0$, the limits on the definite integral become reversed with the change of variable, so we get

$$F[\phi(\alpha t)] = -\frac{1}{\alpha} \Phi \left(\frac{f}{\alpha} \right) \tag{2.41}$$

so that, in general

$$F[\phi(\alpha t)] = \frac{1}{|\alpha|}\Phi\left(\frac{f}{\alpha}\right) \ . \tag{2.42}$$

Thus, when we "squeeze" a function in the time domain, its Fourier transform "spreads out" in the frequency domain (and vice-versa). An extreme end member showing this behavior is the delta function, which is an infinitely squeezed function in the time domain with an infinitely spread out transform (the 1 function; (2.33)) in the frequency domain.

As you have probably already suspected, there is a *duality* between the time and frequency domains, the precise relationship is

$$F[\phi(t)] = \Phi(f) \tag{2.43}$$

$$F[\Phi(t)] = \phi(-f) \ . \tag{2.44}$$

Any function can be decomposed into even and odd parts with respect to the origin

$$\phi(t) = \phi_e(t) + \phi_o(t) \tag{2.45}$$

$$= \frac{1}{2}[\phi(t) + \phi(-t)] + \frac{1}{2}[\phi(t) - \phi(-t)] \tag{2.46}$$

where $\phi_e(t) = \phi_e(-t)$ and $\phi_o(t) = -\phi_o(-t)$. This decomposition can be used to show that the Fourier transform exhibits various symmetry relations.

Consider the transform of a general real and even function, ϕ_e.

$$F[\phi_e(t)] = \int_{-\infty}^{\infty} \phi_e(t) e^{-\imath 2\pi f t} dt \tag{2.47}$$

$$= \int_{-\infty}^{\infty} \phi_e(t)\cos(2\pi f t)\ dt - \imath \int_{-\infty}^{\infty} \phi_e(t)\sin(2\pi f t)\ dt \tag{2.48}$$

$$= 2\int_{0}^{\infty} \phi_e(t)\cos(2\pi f t) dt \tag{2.49}$$

which is even and is purely real. Similarly, for an odd, real function, ϕ_o, the Fourier transform

$$F[\phi_o(t)] = \int_{-\infty}^{\infty} \phi_o(t) e^{-\imath 2\pi f t} dt \tag{2.50}$$

$$= \int_{-\infty}^{\infty} \phi_o(t)\cos(2\pi f t) dt - \imath \int_{-\infty}^{\infty} \phi_o(t)\sin(2\pi f t)\ dt \tag{2.51}$$

$$= -2\imath \int_{0}^{\infty} \phi_o(t)\sin(2\pi f t)\ dt \tag{2.52}$$

is odd and purely imaginary. Thus, the Fourier transform of an arbitrary real function containing both odd and even components may be evaluated as a superposition of (2.49) and (2.52), frequently referred to as the *cosine transform* and *sine transform*, respectively. Using superposition, one can derive a list of basic symmetry relationships between the time and frequency domains:

$\phi(t)$	$\Phi(f)$
even	even
odd	odd
real, even	real, even
real, odd	imaginary, odd
imaginary, even	imaginary, even
imaginary, odd	real, odd
complex, even	complex, even
complex, odd	complex, odd
real, asymmetrical	complex, Hermitian
imaginary, asymmetrical	complex, anti-Hermitian
Hermitian	real
anti-Hermitian	imaginary

where a *Hermitian* function has an even real part and an odd imaginary part ($\Phi(f) = \Phi^*(-f)$) and an *anti-Hermitian* function has an odd real part and an even imaginary part ($\Phi(f) = -\Phi^*(-f)$).

One of the most important conceptual and practical relationships between the time and frequency domains is embodied in the *convolution theorem*. Consider the Fourier transform of the convolution of two functions

$$F[\phi_1(t) * \phi_2(t)] = \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} \phi_1(\tau)\phi_2(t-\tau)\ d\tau \right) e^{-\imath 2\pi f t} dt\ . \tag{2.53}$$

Reversing the order of integration gives

$$F[\phi_1(t) * \phi_2(t)] = \int_{-\infty}^{\infty} \phi_1(\tau) \left(\int_{-\infty}^{\infty} \phi_2(t-\tau) e^{-\imath 2\pi f t} dt \right)\ d\tau\ . \tag{2.54}$$

However, by the time shift property (2.38), this is just

$$\int_{-\infty}^{\infty} \phi_1(\tau)\Phi_2(f) e^{-\imath 2\pi f \tau}\ d\tau = \Phi_1(f)\Phi_2(f) \tag{2.55}$$

so that convolution in the time domain corresponds to multiplication in the frequency domain! Similarly, we can show that multiplication in the time domain corresponds to convolution in the frequency domain

$$F[\phi_1(t)\phi_2(t)] = \Phi_1(f) * \Phi_2(f)\ . \tag{2.56}$$

This can be understood intuitively based on what we know about the response of linear systems, as the response of a linear system at each frequency is just the complex amplitude of that frequency component in the input, times the complex value of the response function of the system at that frequency.

Recall that time differentiation has a remarkably simple form in the frequency domain

$$\frac{d}{dt}\phi(t) = \frac{d}{dt} \int_{-\infty}^{\infty} \Phi(f) e^{\imath 2\pi f t}\ df \tag{2.57}$$

$$= \int_{-\infty}^{\infty} \frac{\partial}{\partial t}[\Phi(f)e^{\imath 2\pi f t}]\, df = \int_{-\infty}^{\infty} 2\pi\imath f\Phi(f)e^{\imath 2\pi f t} df = F^{-1}[2\pi\imath f\Phi(f)] \quad (2.58)$$

taking the Fourier transform of both sides gives:

$$= F[\frac{d}{dt}\phi(t)] = 2\pi\imath f\Phi(f)\ . \quad (2.59)$$

(2.59) clearly shows that differentiation amplifies high frequency signal components relative to those at low frequency, and thus belongs to a class of operators generally referred to as *high-pass filters*.

The situation for integration is somewhat more complex

$$F\left(\int_{-\infty}^{t} \phi(\tau)d\tau\right) = \frac{\Phi(f)}{2\pi\imath f} + \frac{\delta(f)}{2}\int_{-\infty}^{\infty} \phi(t)dt \quad (2.60)$$

where the delta function term accommodates the contribution of a possible non-zero mean value in $\phi(t)$. A definite integrator is thus a *low-pass filter*, as it reinforces low frequencies relative to high frequencies.

(2.59) and (2.60) are helpful in computing some otherwise nonstraightforward Fourier transforms, especially for discontinuous functions. Consider the step function. Using (2.60) gives

$$F[\mathrm{H}(t)] = F\left(\int_{-\infty}^{t} \delta(\tau)d\tau\right) \quad (2.61)$$

$$= \frac{1}{2\pi\imath f} + \frac{\delta(f)}{2}\ . \quad (2.62)$$

The Fourier transform of the sign function is thus

$$F[2\mathrm{H}(t) - 1] = \frac{1}{\pi\imath f}\ . \quad (2.63)$$

Example: Equilibrium elastic response of a loaded, buoyantly supported crust. The differentiation and integration properties of the Fourier transform provide a useful method for obtaining solutions to ordinary linear integrodifferential equations. An example of geophysical interest is the downward deflection of a rigid plate (such as the Earth's crust) buoyantly supported by an underlying liquid (to first order, the mantle) to a distributed load (such as an ice cap, volcano, or reservoir) (Figure 2.5.

The model for the small-deformation equilibrium of a deformed plate is a linear differential equation [4, 14]

$$D\bigtriangledown^4 w(r) = p(r) \quad (2.64)$$

where $w(r)$ is the upward deflection of the plate and $p(r)$ is the upward force per unit area. The forcing term, $p(r)$, arises from a topographic load, $h_l(r)$ and

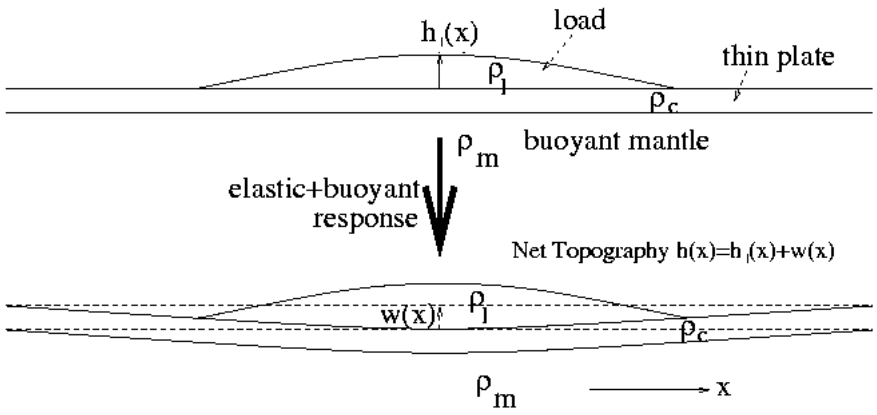

Figure 2.5: A Buoyant, Rigid Plate with a Spatial Load

from a buoyancy term due to the displaced mantle. D is the *flexural rigidity*, which depends on the thickness and elastic moduli of the plate

$$D \equiv \frac{E\tau^3}{12(1-\nu^2)} \tag{2.65}$$

where τ is the plate thickness, E is Young's Modulus, and ν is Poisson's Ratio.

In one spatial dimension, x, (2.64) becomes

$$D\frac{\partial^4 w(x)}{\partial x^4} = p(x) \ . \tag{2.66}$$

The total forcing function for a load of homogeneous density, ρ_l is the sum of the load and the opposite-directed buoyant compensation of the mantle

$$p(x) = -\rho_l g h_l(x) + B(x) \tag{2.67}$$

where ρ_l is the density of the added material, g is the acceleration of gravity, and $B(x)$ is the buoyancy term due to mantle material of density ρ_m,

$$B(x) = -\rho_m g w(x) \ . \tag{2.68}$$

We can thus write the forcing term in terms of the input load $h_l(x)$ as

$$p(x) = -g(\rho_l h_l(x) + \rho_m w(x)) \ . \tag{2.69}$$

Now we can solve for the resulting crustal deformation by separating $w(x)$ and $h_l(x)$ and taking a spatial Fourier transform

$$[(2\pi\imath k)^4 D + g\rho_m]W(k) = -\rho_l g H_l(k) \tag{2.70}$$

where k is the *spatial frequency* (units of 1/length), the spatial counterpart of f. Here, to keep our Fourier conventions unchanged from previous discussion, note that k is just $1/\lambda$, or the reciprocal wavelength (this is not to be confused with the common use of k as the wavenumber, which is $2\pi/\lambda$). Our k (a reciprocal wavelength) and the wavenumber are thus analogues to f and ω.

Note that $H_l(k)$ is the spatial Fourier transform of the input

$$H_l(k) = \int_{-\infty}^{\infty} h_l(x) e^{-\imath 2\pi k x} \, dx \tag{2.71}$$

(*not* the step function). The (spatial) frequency domain solution is thus

$$W(k) = -H_l(k) \frac{\frac{\rho_l}{\rho_m}}{1 + \frac{16\pi^4 k^4 D}{g\rho_m}} \ . \tag{2.72}$$

Note that (2.72) depends strongly on the reciprocal wavelength, k. For k large, the response of the system becomes becomes negligible. Conversely, for k small, the response becomes increasingly significant, reaching a maximum value of

$$W_{max} = W(0) = -H_l(0)\rho_l/\rho_m \tag{2.73}$$

as $k \to 0$. Thus, for long-wavelength (small k) spatial components of the landscape, we say that we have a large degree of buoyant *compensation*, as the topographic load is primarily supported by mantle buoyancy. At short spatial wavelengths, on the other hand (large k), the landscape is almost totally supported by the flexural rigidity of the crust. The degree of compensation for a spatial component of wavelength $\lambda = 1/k$, is the deflection of the system relative to W_{max}

$$C = \frac{W(k)}{W_{max}} \ . \tag{2.74}$$

We can evaluate the impulse response in the x domain by taking the inverse Fourier transform of $W(k)/H_l(k)$ (preferably with the assistance of a table of integral transforms), to obtain

$$q(x) = F^{-1}[W(k)/H_l(k)] \tag{2.75}$$

or

$$q(x) = \frac{-2g\rho_l}{D} \int_0^{\infty} \frac{\cos(2\pi k x) \, dk}{\alpha^4 + (2\pi k)^4} \tag{2.76}$$

where

$$\alpha = \left(\frac{g\rho_m}{D}\right)^{1/4} \tag{2.77}$$

so that [10]

$$q(x) = \frac{-\sqrt{2}g\rho_l}{4\alpha^3 D} \ e^{\frac{-\alpha|x|}{\sqrt{2}}} \ \left(\sin\frac{\alpha|x|}{\sqrt{2}} + \cos\frac{\alpha|x|}{\sqrt{2}}\right) \ . \tag{2.78}$$

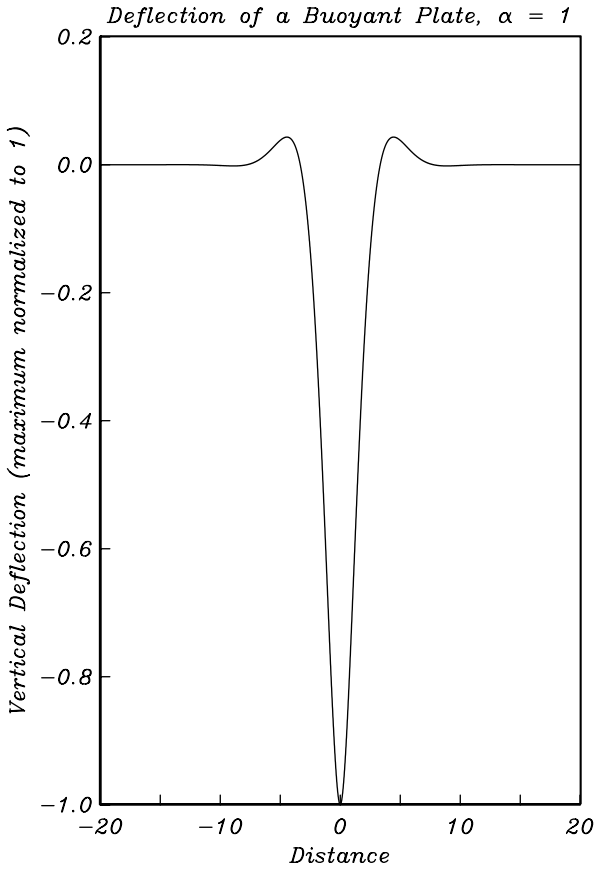

Figure 2.6: Response of a Buoyant, Rigid Plate to an Spatial Impulse Load

This function is plotted in Figure 2.6 and consists a central depression and an outboard peripheral upwarp. Note that (2.78) is the impulse response of this system, as $W(k)$ is the response for $H_l(k) = 1$ (2.72), or $h_l(x) = \delta(x)$, so that *any* 1-d deformation of a rigid plate to a load (assumed to be infinitely extending in the out-of plane direction) can thus be calculated by convolving $q(x)$ and the specific linear load distribution.

Note also that the net topography for the system in equilibrium is given by the sum of the input load topography and the system response

$$h(x) = h_l(x) + w(x) \ . \tag{2.79}$$

Example: Time domain seismometer response. We can use Fourier tools to obtain a result for the displacement response of the vertical seismometer in the time domain by noting, as above, that the time domain response to an impulsive

acceleration characterized by $\ddot{u} = \delta(t)$ is characterized by

$$\ddot{a} + 2\zeta\dot{a} + \omega_s^2 = -\delta(t) \ . \tag{2.80}$$

Taking the Fourier transform of both sides and solving for $a(\omega)$, the displacement response to an acceleration impulse input, gives the frequency domain expression

$$a(\omega) = \frac{1}{\omega^2 - 2\imath\zeta\omega - \omega_s^2} \tag{2.81}$$

Note that this is just the response of the seismometer system to the displacement impulse (2.16), divided by $-\omega^2$. This is appropriate, as the input function has been twice differentiated in the time domain and the response of a differentiator is $\imath\omega$ (2.59).

The time domain displacement response to an acceleration impulse input is therefore

$$\phi(t) = F^{-1}\left(a(\omega)\right) = F^{-1}\left(\frac{1}{\omega^2 - 2\imath\zeta\omega - \omega_s^2}\right) \tag{2.82}$$

$$= \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{e^{\imath\omega t}\ d\omega}{\omega^2 - 2\imath\zeta\omega - \omega_s^2} = \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{e^{\imath\omega t}\ d\omega}{(\omega - \omega_1 - \imath\zeta)(\omega + \omega_1 - \imath\zeta)} \tag{2.83}$$

where

$$\omega_1 = \sqrt{\omega_s^2 - \zeta^2} \tag{2.84}$$

Solving this integral is relatively straightforward using the residue theorem from complex analysis and separation into three cases. For $\omega_s > \zeta$, the system exhibits a distinct resonance near $\omega = \omega_s$ (as we have already seen from examining the frequency response; Figure 2.2) and is referred to as *underdamped*. In this case, the poles of the integrand in (2.83) lie at $(\omega_1, \imath\zeta)$ and $(-\omega_1, \imath\zeta)$. The time domain solution is found from the residues of the two complex poles of the integrand to be

$$a(t) = \frac{-\mathrm{H}(t)}{\omega_1} e^{-\zeta t}\sin(\omega_1 t) \ . \tag{2.85}$$

When $\omega_s < \zeta$, the system does not resonate, the complex poles of the integrand lie on the positive imaginary axis, and the system is referred to as being *overdamped*. ω_1^2 is negative in this case, and the result is an impulse response that is a sum of real exponentials

$$a(t) = \frac{-\mathrm{H}(t)}{2(\zeta^2 - \omega_s^2)^{1/2}}\left(e^{-(\zeta - (\zeta^2 - \omega_s^2)^{1/2})t} - e^{-(\zeta + (\zeta^2 + \omega_s^2)^{1/2})t}\right) \tag{2.86}$$

The case $\omega_s = \zeta$ is a transition between the underdamped and overdamped case,s referred to as *critically damped*. Because there is a double pole, a special case of the residue theorem must be applied to obtain the impulse response, which is

$$a(t) = -\mathrm{H}(t) t e^{-\zeta t} \ . \tag{2.87}$$

These time domain responses are shown in Figure 2.3.

How do we evaluate the displacement impulse response of the system to Earth displacement? One way is to reexpress the integrand in the inverse transform of (2.16) to strip off a delta function

$$\frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{-\omega^2 e^{\imath\omega t}\,d\omega}{\omega^2-2\imath\zeta\omega-\omega_s^2}=-\frac{1}{2\pi}\int_{-\infty}^{\infty}\left(1\;+\;\frac{2\imath\zeta\omega+\zeta^2+\omega_1^2}{(\omega-\omega_1-\imath\zeta)(\omega+\omega_1-\imath\zeta)}\right)e^{\imath\omega t}\,d\omega \tag{2.88}$$

$$=-\delta(t)-\frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{(2\imath\zeta\omega+\zeta^2+\omega_1^2)e^{\imath\omega t}\,d\omega}{(\omega-\omega_1-\imath\zeta)(\omega+\omega_1-\imath\zeta)} \tag{2.89}$$

and then evaluate the remaining integral using the residue theorem. Another way to solve (2.89) is to note that $a(t)$ is the time domain solution for the system response to an Earth acceleration of $a_0(t)=\delta(t)$. Because the seismometer system and the differentiation operation are linear, we can evaluate the seismometer displacement response from a displacement impulse by twice differentiating $a(t)$ with respect to time. For the underdamped system, for example, this gives

$$d(t)=\frac{d^2a(t)}{dt^2}=\frac{d^2}{dt^2}\left(\frac{-\mathrm{H}(t)}{\omega_1}e^{-\zeta t}\sin(\omega_1 t)\right) \tag{2.90}$$

$$\begin{aligned}=-\frac{1}{\omega_1}\big(&\mathrm{H}''(t)e^{-\zeta t}\sin(\omega_1 t)-\mathrm{H}'(t)\zeta e^{-\zeta t}\sin(\omega_1 t)\\ &+\mathrm{H}'(t)e^{-\zeta t}\omega_1\cos(\omega_1 t)-\mathrm{H}'(t)\zeta e^{-\zeta t}\sin(\omega_1 t)\mathrm{H}(t)\zeta^2 e^{-\zeta t}\sin(\omega_1 t)-\\ &+\mathrm{H}(t)\zeta e^{-\zeta t}\omega_1\cos(\omega_1 t)+\mathrm{H}'(t)e^{-\zeta t}\omega_1\cos(\omega_1 t)\\ &-\mathrm{H}(t)\zeta e^{-\zeta t}\omega_1\cos(\omega_1 t)-\mathrm{H}(t)e^{-\zeta t}\omega_1^2\sin(\omega_1 t)\big)\;.\end{aligned} \tag{2.91}$$

Using $\mathrm{H}'(t)=\delta(t)$ and $\mathrm{H}''(t)=\delta'(t)$, and noting that $\delta'(t)\sin(\omega_1 t)e^{-\zeta t}=-\delta(t)\omega_1$, and $\delta(t)e^{-\zeta t}\omega_1\cos(\omega_1 t)=\delta(t)\omega_1$ gives

$$d(t)=-\frac{1}{\omega_1}\left(\delta(t)\omega_1-2\mathrm{H}(t)\omega_1\zeta e^{-\zeta t}\cos(\omega_1 t)+\mathrm{H}(t)\zeta^2 e^{-\zeta t)}\sin(\omega_1 t)-\mathrm{H}(t)\omega_1^2 e^{-\zeta t}\sin(\omega_1 t)\right)\;. \tag{2.92}$$

In the limit as $\omega_s\to 0$, and for an undamped ($\zeta=0$) seismometer, we obtain

$$d(t)=-\delta(t)\;. \tag{2.93}$$

Note that as the resonant frequency, ω_1, becomes small (the resonant period becomes large), (2.93) and Figure 2.7 approach the ideal instrument response of a delta function (with a trivial minus sign). Because seismologists frequently want to know the true ground displacement (its long-period asymptotic spectral level is proportional to the seismic moment, among other reasons), seismometers with very long natural periods are desirable and constitute the instrumental backbone of much of modern seismology. In practice, most seismometers have an output that is proportional to velocity, but if they have suitably low noise at long periods the native output can be stably integrated to produce a displacement seismogram.

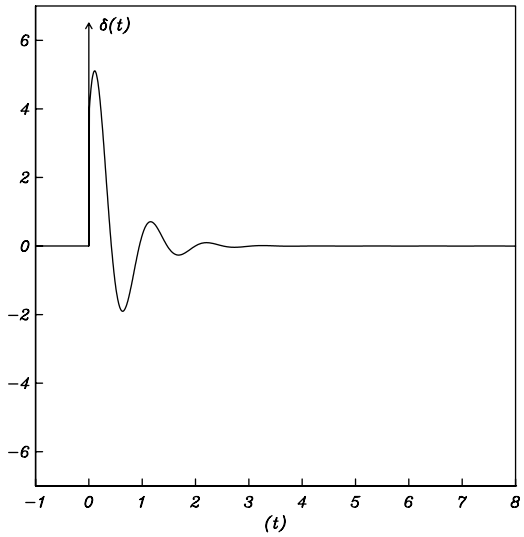

Figure 2.7: Displacement Output/Displacement Input Response of an Underdamped Seismometer ($\zeta = 0.3\omega_s$; $\omega_s = 2\pi$ Hz) to a Displacement Impulse

Moment-spectral relationships. As an additional perspective on the rich mathematics of Fourier theory can be obtained by noting that all of the moments of the time domain function, $\phi(t)$, can be expressed in terms of the behavior of $\Phi(f)$ at the origin. Consider the n^{th} moment

$$\phi_n \equiv \int_{-\infty}^{\infty} t^n \phi(t) dt \ . \tag{2.94}$$

The n^{th} derivative of $\Phi(f)$ with respect to f is

$$\frac{\partial^n \Phi(f)}{\partial f^n} = \int_{-\infty}^{\infty} (-2\pi\imath t)^n \phi(t) e^{-\imath 2\pi f t} dt \tag{2.95}$$

so that

$$\frac{1}{(-2\pi\imath)^n} \left(\frac{\partial^n \Phi(f)}{\partial f^n} \right) = \int_{-\infty}^{\infty} t^n \phi(t) e^{-\imath 2\pi f t} dt \ . \tag{2.96}$$

Evaluating both sides gives

$$\frac{1}{(-2\pi\imath)^n} \left(\frac{\partial^n \Phi(0)}{\partial f^n} \right) = \int_{-\infty}^{\infty} t^n \phi(t) \ dt = \phi_n \ . \tag{2.97}$$

Thus, we can now see that the 0^{th} moment of $\phi(t)$, the total area under $\phi(t)$, is just $\Phi(0)$. Similarly, the 1^{st} moment of $\phi(t)$ is just

$$\int_{-\infty}^{\infty} t\phi(t) dt = -\frac{1}{2\pi\imath} \left(\Phi'(f) \right)_{f=0} \tag{2.98}$$

where

$$\Phi'(f) \equiv \frac{\partial \Phi(f)}{\partial f} \tag{2.99}$$

so that the slope of $\Phi(f)$ at the origin is proportional to the expectation value of t

$$< t >_{\phi(t)} = \frac{\int_{-\infty}^{\infty} t\phi(t)\ dt}{\int_{-\infty}^{\infty} \phi(t)\ dt}\ . \tag{2.100}$$

Time functions which which are symmetrical must therefore have Fourier transforms with zero slope at $f = 0$ (we can also see this from the aforementioned symmetry relations).

The 2^{nd} moment is

$$\int_{-\infty}^{\infty} t^2\phi(t)dt = -\frac{1}{4\pi^2}\left(\Phi''(f)\right)_{f=0} \tag{2.101}$$

so that the curvature of $\Phi(f)$ at the origin is proportional to the second moment of $\phi(t)$. For functions which have an infinite second moment, the Fourier transform has a cusp at the origin, for example,

$$F\left(\frac{1}{\alpha^2 + t^2}\right) = \frac{e^{-\alpha|f|}}{2\alpha}\ . \tag{2.102}$$

Next, consider the variance of $\phi(t)$

$$\sigma^2[\phi(t)] =< (t- < t >)^2 >_{\phi(t)} = \frac{\int_{-\infty}^{\infty}(t^2 - 2t < t > + < t >^2)\phi(t)\ dt}{\int_{-\infty}^{\infty} \phi(t)\ dt} \tag{2.103}$$

$$= \frac{1}{\Phi(0)}\left(\frac{\Phi''(0)}{(-2\pi\imath)^2} - 2\frac{\Phi'(0)}{-2\pi\imath}\cdot\frac{\Phi'(0)}{-2\pi\imath\Phi(0)} + \frac{[\Phi'(0)]^2}{(-2\pi\imath)^2}\cdot\frac{\Phi(0)}{\Phi(0)^2}\right) \tag{2.104}$$

$$= \frac{1}{4\pi^2\Phi(0)}\left(-\Phi''(0) + \frac{[\Phi'(0)]^2}{\Phi(0)}\right)\ . \tag{2.105}$$

What is the variance, then, of $\phi_1(t) * \phi_2(t)$? Using the convolution theorem (2.55) makes this straightforward, as

$$\sigma^2[\phi_1(t) * \phi_2(t)] = \frac{1}{4\pi^2\Phi_1(0)\Phi_2(0)}\left(-(\Phi_1\Phi_2)''(0) + \frac{[(\Phi_1\Phi_2)'(0)]^2}{\Phi_1(0)\Phi_2(0)}\right) \tag{2.106}$$

$$= \frac{1}{4\pi^2}\left[-\frac{\Phi_1''(0)}{\Phi_1(0)} - \frac{\Phi_2''(0)}{\Phi_2(0)} + \left(\frac{\Phi_1'(0)}{\Phi_1(0)}\right)^2 + \left(\frac{\Phi_2'(0)}{\Phi_2(0)}\right)^2\right] = \sigma^2[\phi_1(t)] + \sigma^2[\phi_2(t)] \tag{2.107}$$

which gives the important result that the variance of a convolution result is just the sum of the variances of the two constituent functions. This is a quantitative measure of the amount of "spreading" that occurs in the convolution operation. Unless one or both of the constituent functions in the convolution has zero

variance, the convolution result will always have greater variance than either of the two input functions.

Causal systems and the Hilbert transform. An important relationship exists between the real and imaginary parts of the Fourier transform of a real causal function, $\phi_c(t)$, that is, a real function that is zero for all $t < 0$. To see this, we first decompose $\phi_c(t)$ into its even and odd parts

$$\phi_c(t) = \phi_e(t) + \phi_o(t) = 1/2(\phi_c(t) + \phi_c(-t)) + 1/2(\phi_c(t) - \phi_c(-t)). \quad (2.108)$$

For the causal function, we can express $\phi_o(t)$ in terms of $\phi_e(t)$, as:

$$\phi_o(t) = \phi_e(t) \quad (t > 0) \quad (2.109)$$

and

$$\phi_o(t) = -\phi_e(t) \quad (t < 0) \quad (2.110)$$

Thus

$$\phi_c(t) = [1 + \mathrm{sgn}(t)]\phi_e(t) \ . \quad (2.111)$$

By superposition, using the frequency domain convolution theorem (2.56),

$$\Phi_c(f) = \Phi_e(f) + F[\mathrm{sgn}(t)] * \Phi_e(f) \ , \quad (2.112)$$

and using the Fourier transform of the sign function (2.63), we obtain the Fourier transform of $\phi_c(t)$ explicitly in terms of the Fourier transform of $\phi_e(t)$

$$\Phi_c(f) = \Phi_e(f) + \frac{-\imath}{\pi f} * \Phi_e(f) \ . \quad (2.113)$$

Note that because $\phi_e(t)$ is real and even, so is $\Phi_e(f)$. Thus, the real and imaginary parts of $\Phi_c(f)$ are related to each other by the real convolution operator $(-1/\pi f)$. This relationship can be summarized by

$$\Im[\Phi_c(f)] = \frac{1}{\pi}\int_{-\infty}^{\infty} \frac{\Re[\Phi_c(\xi)]}{\xi - f}\, d\xi = \Re[\Phi_c(f)] * \frac{-1}{\pi f} \ \equiv \ \mathrm{H}[\Re[\Phi_c(f)]] \ . \quad (2.114)$$

and conversely,

$$\Re[\Phi_c(f)] = -\frac{1}{\pi}\int_{-\infty}^{\infty} \frac{\Im[\Phi_c(\xi)]}{\xi - f}\, d\xi = \Im[\Phi_c(f)] * \frac{1}{\pi f} \ \equiv \ \mathrm{H}^{-1}[\Im[\Phi_c(f)]] \ . \quad (2.115)$$

One can confirm (2.115) by showing that

$$-\frac{1}{\pi f} * \frac{1}{\pi f} = \delta(f) \ . \quad (2.116)$$

(2.114) is the *Hilbert transform* and (2.115) is the *inverse Hilbert transform* operator, acting on $\Re[\Phi_c(f)]$ and $\Im[\Phi_c(f)]$, respectively. This relationship puts constraints on the frequency response of all physical (causal) transfer functions.

If we take the Hilbert transform of a *time* function, we get the associated *quadrature* function.

$$\mathrm{H}[\phi(t)] = \hat{\phi}(t) \ . \tag{2.117}$$

The Fourier transform of the quadrature function has the same amplitude information as the original function, but its phase is multiplied by $\imath \ \mathrm{sgn}(f)$, so that it is phase shifted by $-\pi/2$ for negative frequencies and by $\pi/2$ for positive frequencies.

An *analytic signal* is one in which the real and imaginary parts are related by the Hilbert transform (so that its Fourier transform is zero for all negative frequencies)

$$A(t) = \phi(t) - \imath\hat{\phi}(t) \ . \tag{2.118}$$

Among its other uses, the analytic time series formulation is useful in evaluating the amplitude envelope, $|A(t)|$, of a function.

A example of a causal physical system is the attenuation which occurs when a wave propagates through a lossy medium. In seismology, such media (which of course include all real materials) are referred to as *anelastic*. The loss mechanisms need not concern us in detail here, but they include work done at grain boundaries and other irreversible changes in the material. The observational result of attenuation is that the energy arriving at the receiver is less than that which one would expect from considering the effects of geometrical spreading and other ray path effects alone.

For the idealized case of a one-dimensional plane wave propagating through a lossless medium (e.g., an electromagnetic wave propagating through a perfect vacuum, or a seismic wave propagating through a perfectly elastic medium) the signal, β, at position x and time t is simply the signal at the source delayed by the propagation time x/v

$$a(x,t) = a(t - x/v) \tag{2.119}$$

where v is the phase velocity. If the time function at the source is $a(t)$, then we can express the signal at an arbitrary time and place as

$$a(x,t) = a_0(t) * \delta(t - t_0) \tag{2.120}$$

where $t_0 = x/v$ and $a_0(t)$ is the signal at $x = 0$. We are assuming here that all frequency components propagate at a single velocity, v. Such a medium is referred to as *nondispersive*. The transfer function of a lossless, nondispersive system is therefore that of a time delay. Consider an exponential input at some frequency, f, the output of the delay system is

$$a(x,f) = \int_{-\infty}^{\infty} \delta(t - t_0) e^{-\imath 2\pi f t} \ dt = e^{-\imath 2\pi f t_0} = e^{-\imath 2\pi f x/v} \ . \tag{2.121}$$

The quality factor, Q, of an oscillating system is given by

$$\frac{1}{Q(f)} = \frac{\delta E}{2\pi E} \tag{2.122}$$

where E is the peak energy of the system and δE is the energy lost in each cycle, assuming $Q \gg 1$. For a propagating sinusoidal disturbance, then, the loss relationship as a function of x is

$$\delta E = \frac{dE}{dx}\lambda \tag{2.123}$$

as the field goes through one oscillation in a wavelength, $\lambda = v/f$. Combining (2.123) and (2.122), we have

$$\frac{2\pi E}{Q} = \frac{dE}{dx}\lambda \tag{2.124}$$

which has a solution for propagating energy of

$$E(x,\ f) = E_0(t)e^{-2\pi fx/Qv} \tag{2.125}$$

or for propagating amplitude of

$$b(x,\ f) = b_0(t)e^{-\pi fx/Qv}\ . \tag{2.126}$$

The combined transfer function for the system is thus, by the convolution theorem (2.55)

$$c(x,\ f) = F\left(\frac{1}{a_0(t)}a(x,t)\ *\ \frac{1}{b_0(t)}b(x,t)\right) \tag{2.127}$$

$$\frac{1}{a_0}a(x,\ f)\ \cdot\ \frac{1}{b_0}b(x,\ f) = e^{-\imath 2\pi fx/v}\ \cdot\ e^{-\pi fx/Qv}\ . \tag{2.128}$$

Taking the inverse Fourier transform of $c(x,\ f)$ to obtain the impulse response of the system, we have (taking the absolute value of f so that negative and positive frequencies are treated equally)

$$c(x,t) = \int_{-\infty}^{\infty} e^{2\pi(-|f|t_0/2Q+\imath f(t-t_0))}\ df \tag{2.129}$$

$$= \int_0^{\infty} e^{2\pi(-ft_0/2Q+\imath f(t-t_0))}\ df + \int_{-\infty}^{0} e^{2\pi(ft_0/2Q+\imath f(t-t_0))}\ df \tag{2.130}$$

$$= -\frac{1}{2\pi}\left[\frac{1}{(\imath t-(\imath+1/2Q)t_0} - \frac{1}{(\imath t-(\imath-1/2Q)t_0}\right] \tag{2.131}$$

$$= \frac{1}{\pi}\left(\frac{(t_0/2Q)}{(t-t_0)^2+(t_0/2Q)^2}\right) \tag{2.132}$$

which is plotted in Figure 2.8

(2.132) is a symmetrical pulse with a maximum at $t = t_0$. Note, however, that $c(x,\ t)$ is not zero for $t < t_0$. This solution is therefore acausal and cannot correspond to the behavior of the real world. Reexamining our assumptions, we find that we must reassess both the nondispersiveness of the medium and the constancy of Q across all frequencies. A moment's reflection reveals that we

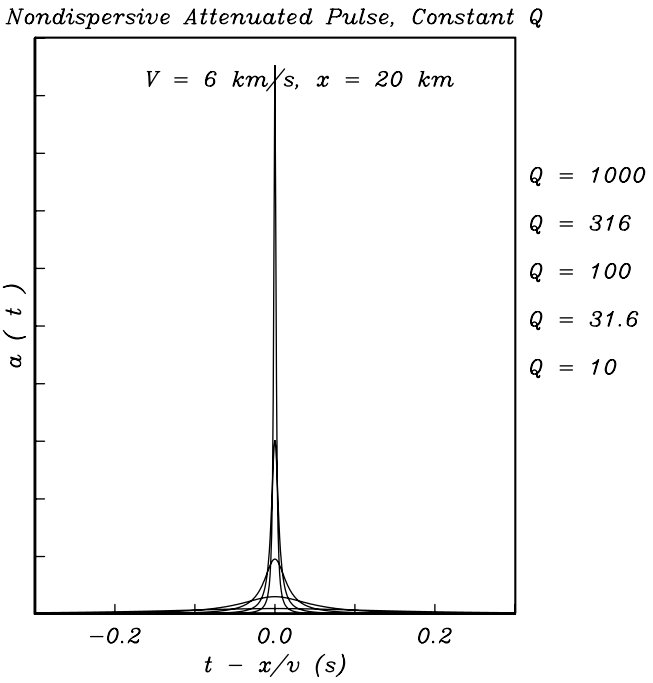

Figure 2.8: Attenuated Pulses, Constant Q

cannot get an asymmetrical, causal pulse by simply allowing Q to vary as an even function of frequency, as the Q operator will affect positive and negative frequencies equally and hence will not alter the symmetry of the pulse. Thus, we are led to the conclusion that all lossy media must be dispersive!

The general transfer function for a wave propagating towards positive x is thus a generalization of (2.128)

$$c(x,\ f) = e^{-\pi|f|x/Q(f)v(f)}\ \cdot\ e^{-\imath 2\pi f x/v(f)} \tag{2.133}$$

where v and Q are now functions of f. We can write this as

$$c(x,\ f) = e^{-2\pi\imath K x} \tag{2.134}$$

if we define the complex wavenumber, K as

$$K = \frac{-\imath|f|}{2Q(f)v(f)} + \frac{f}{v(f)}\ \equiv\ \frac{f}{v(f)} + \imath\alpha(f) \tag{2.135}$$

where $\alpha(f)$ is the attenuation factor. The impulse response is thus the inverse Fourier transform of this

$$c(x,\ t) = \int_{-\infty}^{\infty} e^{\imath 2\pi(-Kx+ft)}\ df \tag{2.136}$$

It can be shown (e.g., Aki and Richards, v. I, 1980) that constraining $c(x,\ t)$ to be causal, i.e., equal to zero for $t < t_1 = x/v_\infty$ places the following constraint on the dispersive velocity function

$$\frac{f}{v(f)} = \frac{f}{v_\infty} + \mathrm{H}[\alpha(f)] \tag{2.137}$$

where v_∞ is the phase velocity at infinite frequency and H is the Hilbert transform. Finding solutions to (2.137) is non-trivial, and there is no solution for constant Q. If we take Q to be constant over the seismic frequency range, however, we can arrive at the useful solution proposed by Azimi [3], where the phase velocity is approximately given by

$$\frac{1}{v(f)} = \frac{1}{v_\infty} + \frac{2\alpha_0}{\pi}\ln\left(\frac{1}{2\pi f\alpha_1}\right) \tag{2.138}$$

where α_0 and α_1 are constants. Using

$$\alpha_0 \approx (2v_\infty Q)^{-1}. \tag{2.139}$$

and

$$\alpha_1 = 0.01\ \mathrm{s} \tag{2.140}$$

Figure 2.9 showns the results of numerically integrating (2.138) for various values of Q to obtain attenuation pulses which are asymmetrical and exhibit a much

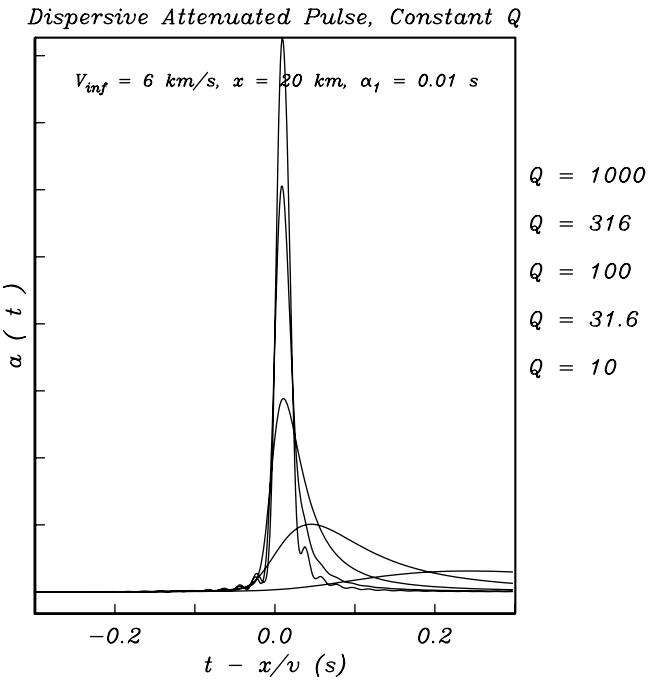

Figure 2.9: Attenuated Pulses, Quasi-Causal Q

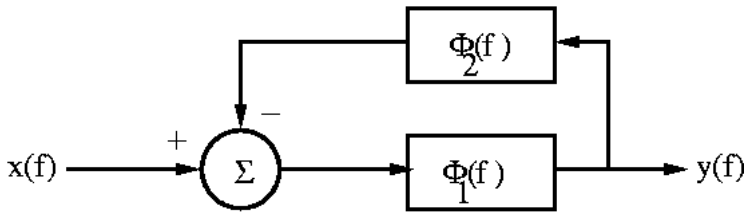

Figure 2.10: A linear system with feedback

better approximation to causal behavior than the nondispersive pulses of Figure 2.8.

The effect of feedback on the transfer function. An important engineering concept is the effect of *feedback* on the transfer function of a system. Figure 2.10 shows the situation where a filtered portion of an output signal, modified by the feedback transfer function Φ_2 is subtracted from the input signal (*negative feedback*). The effect of feedback can alter the system response significantly and, in the case of engineering applications, can do so in several highly desirable ways. The net transfer function for the system of Figure 2.10 is

$$Y(\omega) = (X(\omega) - \Phi_2(\omega)Y(\omega))\Phi_1(\omega) \tag{2.141}$$

which gives

$$\Phi_{fb}(\omega) = \frac{Y(\omega)}{X(\omega)} = \frac{\Phi_1(\omega)}{1 + \Phi_1(\omega)\Phi_2(\omega)} \ . \tag{2.142}$$

For example, consider Φ_1 to be the displacement transfer function for a seismometer (2.16) with damping ζ and natural frequency ω_s, and the feedback component transfer function being a constant $\Phi_2 = k$. In this case the transfer function of the fed back system is

$$\Phi_{fb}(\omega) = \frac{\frac{-\omega^2}{\omega^2 - 2\imath\zeta\omega - \omega_s^2}}{1 - \frac{k\omega^2}{\omega^2 - 2\imath\zeta\omega - \omega_s^2}} = \frac{-\omega^2}{(1-k)\omega^2 - 2\imath\zeta\omega - \omega_s^2} \tag{2.143}$$

which has poles at

$$\omega_{fb} = \frac{\imath\zeta \pm \sqrt{(1-k)\omega_s^2 - \zeta^2}}{1-k} \tag{2.144}$$

instead of the original poles at

$$\omega = \imath\zeta \pm \sqrt{\omega_s^2 - \zeta^2} \equiv \imath\zeta \pm \omega_1 \ . \tag{2.145}$$

A plot of the poles of the function in $z = \imath\omega_{fb}$ complex plane (Figure 2.11); see the ancillary Poles and Zeros notes), shows the system behavior as k is increased from zero for an initially $\omega_s = 2\pi$ rad/s underdamped seismometer with $\zeta = 0.1\omega_s$. The damping increases as k increases (the ratio (real(z)/imag(z) increases), and the system response approaches critical damping. As the amount

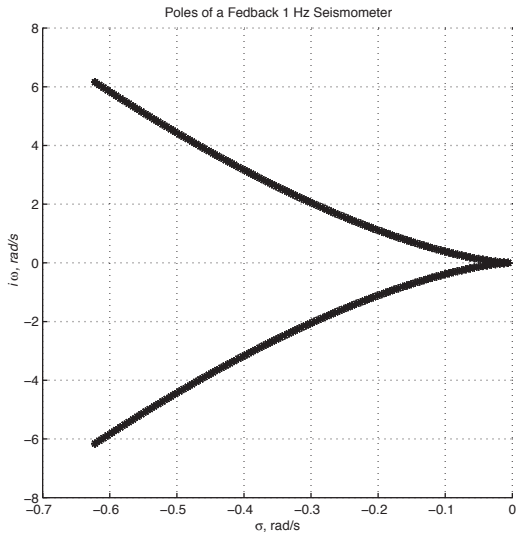

Figure 2.11: Poles for a seismometer with simple feedback ($0.01 \leq k \leq 0.99$).

of feedback is increased, the system response approaches that of a very-long period seismometer.

Feedback is the essence of modern broadband seismometer design; the feedback makes it possible to build portable stable, low noise instruments with periods $T = 2\pi/\omega_{fb}$ as long as several hundred seconds. An added technical advantage associate with feedback is, if there is enough gain in the system so that $|\Phi_1(\omega)\Phi_2(\omega)|/gg1$, (2.142) becomes

$$\Phi_{fb}(\omega) = \frac{y(\omega)}{x(\omega)} \approx \frac{1}{\Phi_2(\omega)} \ , \tag{2.146}$$

and the system response can be made effectively completely dependent on properties of the feedback elements, $\Phi_2(\omega)$ (which are typically electronic), rather than on less predictable and/or stable mechanical seismometer components.

Chapter 3

Sampled Time Series And The Discrete Fourier Transform

Sampled Time Series

Numerical scientific data are commonly organized into series or matrices, i.e., sets of spatially or temporally ordered numbers that approximate a continuous time function (e.g., seismic signals, magnetic observatory data, temperature variations). In spatial applications, the data commonly consists of a two- or three-dimensional array of samples (e.g., gravity, magnetic, or structural surveys). These data may be irregularly sampled in space and/or time. Here, we will consider Fourier theory appropriate to the case where the data are sampled at regular intervals (or where irregularly sampled data has been interpolated or otherwise transferred to a regular array of numbers).

Beginning with a continuou function, multiplication by the (uniformly spaced delta function sequence) shah function, $\mathrm{III}(t)$, can be conceptualized as performing a *regular sampling* operation for a time series. By "regular", we mean that this operation selects out instantaneous functional values at equally-spaced intervals, $1/r$ (where r is the *sampling rate* or *sampling frequency*), and ignores continuous function information between the samples. In instrumentation practice, this type of operation is in practice performed by an *analog-to-digital converter (A to D)* or *digitizer*, and the sampled values are stored as series or arrays of numbers.

To examine what sampling does to the spectral characteristics of an arbitrary function, we evaluate the Fourier transform of $\mathrm{III}(t)$ and apply the frequency-domain counterpart of the convolution theorem. We will find $\mathcal{F}[\mathrm{III}(t)]$ by evaluating the Fourier transform of a function with a limit that converges to $\mathrm{III}(t)$.

One such function is

$$\mathrm{III}(t) = \sum_{n=-\infty}^{\infty} \delta(t-n) = \lim_{\tau\to 0} \frac{1}{\tau} e^{-\pi\tau^2 t^2} \sum_{n=-\infty}^{\infty} e^{-\pi(t-n)^2/\tau^2} \ . \tag{3.1}$$

Note that (3.1) consists of a broad Gaussian envelope

$$e^{-\pi\tau^2 t^2} \tag{3.2}$$

multiplied by a periodic component

$$\frac{1}{\tau} \sum_{n=-\infty}^{\infty} e^{-\pi(t-n)^2/\tau^2} \tag{3.3}$$

You may already know that a smooth periodic function has a *Fourier series*, which is a line spectrum consisting of equally spaced delta functions (some of which may have zero amplitude so as to leave holes in the spectrum). The Fourier series for (3.3) is

$$\frac{1}{\tau} \sum_{n=-\infty}^{\infty} e^{-\pi(t-n)^2/\tau^2} = \sum_{n=-\infty}^{\infty} e^{-\pi\tau^2 n^2} e^{\imath 2\pi n t} \tag{3.4}$$

so that

$$\mathrm{III}(t) = \lim_{\tau\to 0} e^{-\pi\tau^2 t^2} \sum_{n=-\infty}^{\infty} e^{-\pi\tau^2 n^2} e^{\imath 2\pi n t} \ . \tag{3.5}$$

Thus,

$$\mathcal{F}[\mathrm{III}(t)] = \lim_{\tau\to 0} \sum_{n=-\infty}^{\infty} e^{-\pi\tau^2 n^2} \mathcal{F}[e^{-\pi\tau^2 t^2} e^{\imath 2\pi n t}] \tag{3.6}$$

and applying the frequency-domain counterpart of the time shift theorem gives

$$\mathcal{F}[\mathrm{III}(t)] = \lim_{\tau\to 0} \sum_{n=-\infty}^{\infty} e^{-\pi\tau^2 n^2} \mathcal{F}[e^{-\pi\tau^2 t^2}]|_{f=f-n} \ . \tag{3.7}$$

The Fourier transform of a Gaussian function is

$$\mathcal{F}[e^{-\alpha\pi t^2}] = \int_{-\infty}^{\infty} e^{-\alpha\pi t^2 - 2\pi\imath f t} \, dt \tag{3.8}$$

$$= e^{-\pi f^2/\alpha} \int_{-\infty}^{\infty} e^{-\pi(\alpha t^2 + 2\imath f t - f^2/\alpha)} \, dt = e^{-\pi f^2/\alpha} \int_{-\infty}^{\infty} e^{-\pi(\alpha^{1/2} t + \imath f/\alpha^{1/2})^2} \, dt \ . \tag{3.9}$$

Substituting $\xi = \alpha^{1/2} t + \imath f/\alpha^{1/2}$ gives

$$= \frac{1}{\alpha^{1/2}} e^{-\pi f^2/\alpha} \int_{-\infty}^{\infty} e^{-\pi\xi^2} d\xi = \frac{1}{\alpha^{1/2}} e^{-\pi f^2/\alpha} \ . \tag{3.10}$$

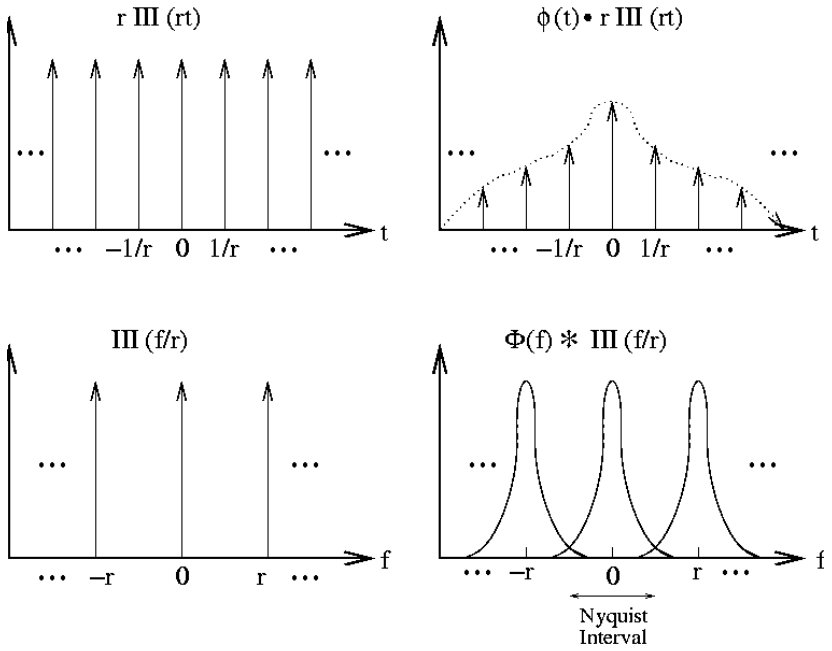

Figure 3.1: The Shah function and its Fourier Transform; Fourier Transform of a Sampled Function (slightly aliased)

So the Fourier transform of a Gaussian is just another Gaussian! Thus, we have

$$\mathcal{F}[\mathrm{III}(t)] = \lim_{\tau\to 0} \frac{1}{\tau} \sum_{n=-\infty}^{\infty} e^{-\pi\tau^2 n^2} e^{-\pi(f-n)^2/\tau^2} \ . \tag{3.11}$$

Now we take the limit as $\tau \to 0$ and see that (3.11) converges to the same limit as (3.1); the shah is, like the Gaussian, its own Fourier transform

$$\mathcal{F}[\mathrm{III}(t)] = \mathrm{III}(f) \ . \tag{3.12}$$

Sampling and Aliasing. Consider a sampled time function

$$\psi(t) = \phi(t) \cdot r\mathrm{III}(rt) \tag{3.13}$$

which is a regularly spaced (intervals of r^{-1}) sequence of delta functions in time with areas given by the values of $\phi(t)$ at those times. Using the convolution and scaling theorems, we can see that (3.12) gives (Figure 3.1)

$$\Psi(f) = \Phi(f) * \mathrm{III}(\frac{f}{r}) \ . \tag{3.14}$$

Sampling thus simply replicates the Fourier transform of $\phi(t)$, $\Phi(f)$, along the frequency axis at $\pm nr$. These copies are referred to as *aliases*. If $\Phi(f)$ is

band-limited to having its energy in the frequency interval $(-f_{max}, f_{max})$ and if $f_{max} \leq r/2$, then these aliases will not overlap. This is a crucial observation; it implies that $\phi(t)$ is fully recoverable from the sampled series via an inverse Fourier transform across one of the aliases

$$\phi(t) = F^{-1}[\Psi(f)\Pi(f/r)] \tag{3.15}$$

or (using the convolution theorem) as the convolution

$$\phi(t) = \psi(t) * r \text{ sinc } (rt) \ . \tag{3.16}$$

This remarkable result, that a continuous band-limited function can be fully recovered from time series sampled at a rate of $r > 2f_{max}$ (so that the aliases don't overlap!), leads to the definition of the *Nyquist frequency*

$$f_N = 2f_{max} \tag{3.17}$$

the minimum frequency at which we must sample for information to be recovered without corruption from a sampled time series. Thus, if we wish to sample a signal that has appreciable power up to 100 Hz, we must sample using a rate of at least $f_N = 200$ Hz. One way of intuitively appreciating the Nyquist frequency concept is that it takes slightly more than two samples per period to accurately characterize a sinusoid.

As can be seen from (3.14) that, if the sampling rate r is less than $2f_{max} = f_N$, (as in Figure 3.1), then the sampled times series aliases will overlap and corrupt each other, a condition called *undersampling*. Applying (3.16) to try and recover $\phi(t)$ in this case will produce a distorted recovered function. This undersampling distortion is called *aliasing*, and such a time series is referred to as being *aliased*. If we aren't interested in the higher frequency content in a signal, we can eliminate aliasing problems by removing the higher-frequencies from the data (using low-pass filtering) prior to sampling so that the signal contains a negligible amount of energy at frequencies near and above $f_N/2$. This type of presampling, low-pass filter is called an *antialias filter*. In data acquisition systems, antialiasing is sometimes practically accomplished by drastically oversampling the data at the analog input and then filtering and decimating the signal digitally to produce an unaliased signal at a lower, desired sampling rate. This eliminates the need for variable analog antialiasing electronics for the lower sampling rates.

It is important to understand in detail what happens if we undersample data. First, note that we never satisfy (3.17) exactly, because all real data sets are time or space limited and thus can never be truly band-limited to $\pm r/2$ (fortunately, we can get close in this regard in practical cases). One way to see this is to note that "perfect" low-pass filtering is unobtainable, as the impulse response of a perfect low-pass filter (one with a frequency response of $\Pi(f/f_{max})$) is the acausal sinc function, which has non-zero values from $t = -\infty$ to $t = \infty$. Consider the distorted spectrum, $\Phi_a(f)$, resulting from the influence of the two nearest frequency-domain aliases, which are centered at $f = \pm r$ (Figure 3.1)

$$\Phi_a(f) = \Phi(f) + \Phi(f - r) + \Phi(f + r) \ . \tag{3.18}$$

If $\phi(t)$ is real, then $\Phi(f)$ is Hermitian, so that

$$\Phi_a(f) = \Phi(f) + \Phi^*(r-f) + \Phi^*(r+f) \ . \tag{3.19}$$

The contribution to the aliased signal from the second two terms is just what one would get by adding complex-conjugated versions of the spectrum which have been "folded" in the frequency domain at $f = \pm r/2$. Note that the actual character of corruption of the original signal depends on the specific characteristics of $\Phi(f)$. The greatest distortion will occur if there is sufficient high-frequency energy above $f = r/2$ so that even the lower frequency components of $\Phi_a(f)$ (3.19) will be significantly different than those of $\Phi(f)$. A time domain sign of danger in a sampled data set would be the occurrence of lots of terms with alternating signs, as this is an indication that there is significant energy at or above $f = r/2$.

As an example of aliasing which could occur in practice, consider an undersampled voltage that is contaminated by an $f_0 = 60$ Hz AC sinusoidal noise

$$n(t) = A\cos(2\pi \cdot 60t) \ . \tag{3.20}$$

To prevent aliasing of $n(t)$, we would have to sample at a rate greater than $r \geq f_N = 2f_0 = 120$ Hz. If we instead sampled at a lesser rate, the delta function spectrum of the noise component

$$n_a(t) = n(t) \cdot r\text{III}(rt) \tag{3.21}$$

would have, in the central alias bracketing $f = 0$, its frequencies mapped to $f = \pm(r - 60)$ Hz. As an extreme case, if we sampled at half of the Nyquist frequency, (60 Hz), the 60 Hz energy in $n(t)$ would be mapped to zero frequency – producing a zero frequency component in the retrieved function. We can see why this is by looking back in the time domain and noting that this corresponds to sampling a sinusoid once per period, so that all such samples will have identical value. The specific value would depend on the phase relationship between the sampling function and $n(t)$; if the samples are centered on zero time and $n(t)$ is a cosine, then we would recover a maximum zero-frequency signal of amplitude A. Aliasing thus puts true signal into different frequency ranges. This behavior occurs because, for signal frequencies higher than the Nyquist frequency, sampling and recovery is a nonlinear process.

Fourier Theory in Discrete Time. In analyzing sampled time series, it is more practical to work in discrete (rather than continuous) time or space. As previously mentioned, essentially all practical data analysis schemes are implemented on computers, which do not process functions per se, but instead operate on discrete ordered sets of numbers. A 1-dimensional ordered set of numbers is called a *sequence* , which we will typically represent in subscript notation

$$x_n(n \in \text{integers}) \ . \tag{3.22}$$

The discrete time equivalent of the delta and step functions are the *Kronecker delta*

$$\delta_{n-m} \equiv \delta_{n,m} = \begin{cases} 1 & n = m \\ 0 & n \neq m \end{cases} \tag{3.23}$$

and its associated discrete step function

$$\mathrm{H_{n-m}} = \begin{cases} 1 & n \geq m \\ 0 & n < m \end{cases} \tag{3.24}$$

In the discrete time domain, summation supplants integration, so that the delta/step relationship integral relationship in continuous time becomes

$$H_{n-l} = \sum_{k=-\infty}^{n} \delta_{k-l} \ . \tag{3.25}$$

Analogously, convolution in the discrete world (e.g., in MATLAB) is a summation operation

$$x_n * y_n = \sum_{k=-\infty}^{\infty} x_k y_{n-k} \tag{3.26}$$

where the y index is reversed in the summation index, k, which fills in for its continuous counterpart, τ.

To investigate how Fourier concepts apply to sequences, consider the response of a linear discrete-time system (with an infinite-length impulse response sequence x_n) to a unit-amplitude, complex sinusoidal signal, s_n:

$$g_n = \sum_{k=-\infty}^{\infty} x_k s_{n-k} = \sum_{k=-\infty}^{\infty} x_k e^{2\pi\imath f(n-k)} \tag{3.27}$$

$$= e^{\imath 2\pi f n} \sum_{k=-\infty}^{\infty} x_k e^{-\imath 2\pi f k} \equiv X(f) e^{\imath 2\pi f n} \tag{3.28}$$

where $X(f)$ is the Fourier transform of x_n (keep in mind that x_n is a sequence, not a continuous function). We can unify the Fourier transform definitions for continuous and discrete functions using the sifting property of the delta function

$$X(f) \equiv \mathcal{F}[x_n] = \mathcal{F}[r\mathrm{III}(rt)x(t)] = r\mathcal{F}[\sum_{n=-\infty}^{\infty} \delta(rt-n)x(t)] \tag{3.29}$$

$$= r\int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(rt-n)x(t)e^{-\imath 2\pi f t}dt = r\sum_{n=-\infty}^{\infty} x(n/r)e^{-\imath 2\pi f n/r} \ . \tag{3.30}$$

The spectrum of (3.30) is continuous and periodic in the frequency domain, (with a spectral period of r). This periodicity reflects the spectral aliasing effects of sampling discussed earlier. It is usually most convenient to take $r = 1$, in which case the spectrum is normalized with respect to the Nyquist frequency and we need only concern ourselves with a unit Nyquist interval $-1/2 \leq f \leq 1/2$ to capture all of the information in x_n (provided that we sample rapidly enough so that the spectral aliases are non-overlapping)

$$X(f) = \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(t-n)x(t)e^{-\imath 2\pi f t}dt = \sum_{n=-\infty}^{\infty} x(n)e^{-\imath 2\pi f n} \ . \tag{3.31}$$

The original sequence can be recovered using the inverse Fourier transform, where we restrict the range of integration to the Nyquist interval

$$x_n = \int_{-1/2}^{1/2} X(f) e^{\imath 2\pi f n}\, df \ . \tag{3.32}$$

The Discrete Fourier Transform. (3.31) and (3.32) form a transform pair, but not a very useful or symmetric one, as the time sequence is infinite and the spectrum is continuous. You might imagine (and you would be right), that $X(f)$, being band limited to the Nyquist interval, could be completely specified by some sequence in the frequency domain. In this case, we would have a transform pair where both the time and frequency domain representations are discrete, and that could be used in practical situations to analyze data.

To construct such a transform pair, consider a periodic sequence, x_n, where the period is N samples. For the moment, assume that the sequence is sampled at a sampling rate of $r = 1$– we will discuss other sampling rates later. Because of its periodicity, every component of x_n of the form $e^{2\pi ikn/N}$ must also be N-sample periodic. These periodic components must therefore have frequencies $f = k/N$, where k is some integer. Because our sequence is sampled at rate $r = 1$, frequencies outside of the range $0 \le r \le 1$ would be aliased. Thus it's unnecessary to include frequencies k/N for k outside of the range from 0 to $N - 1$.

The sequence can thus be completely characterized across one of its periods via the expansion

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{2\pi ikn/N} = \mathrm{IDFT}(X_k) \ . \tag{3.33}$$

The normalization factor $1/N$ is not strictly required, but is included at this point to conform with standard conventions. Equation (3.33) defines our inverse discrete Fourier transform. The corresponding forward transform is

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi ikn/N} = \mathrm{DFT}(x_n) \ . \tag{3.34}$$

To verify this transform pair, we can begin with (3.33) and apply the forward transform to both sides of the equation

$$\sum_{n=0}^{N-1} x_n e^{-\imath 2\pi nm/N} = \frac{1}{N} \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} X_k e^{\imath 2\pi kn/N} e^{-\imath 2\pi nm/N} \ . \tag{3.35}$$

Interchanging the order of summation gives

$$\sum_{n=0}^{N-1} x_n e^{-\imath 2\pi nm/N} = \frac{1}{N} \sum_{k=0}^{N-1} X_k \sum_{n=0}^{N-1} e^{\imath 2\pi n(k-m)/N} \ . \tag{3.36}$$

$$\sum_{n=0}^{N-1} x_n e^{-i2\pi nm/N} = \frac{1}{N} \sum_{k=0}^{N-1} X_k \sum_{n=0}^{N-1} \left(e^{\imath 2\pi (k-m)/N} \right)^n \ . \tag{3.37}$$

Now, consider the innermost sum

$$\sum_{n=0}^{N-1} \left(e^{\imath 2\pi (k-m)/N} \right)^n \ . \tag{3.38}$$

Recall that the sum of a finite geometric series is given by

$$1 + r + r^2 + \ldots + r^{N-1} = \frac{1 - r^N}{1 - r} \quad r \neq 1 \ . \tag{3.39}$$

When $r = 1$, the sum is simply N. When $k - m$ is a multiple of N, then

$$e^{\imath 2\pi (k-m)/N} = 1 \tag{3.40}$$

and

$$\sum_{n=0}^{N-1} \left(e^{\imath 2\pi (k-m)/N} \right)^n = \sum_{n=0}^{N-1} 1^n = N \ . \tag{3.41}$$

When (the integer) $k - m$ is not a multiple of N, $e^{\imath 2\pi (k-m)/N}$ is not equal to one, and

$$\sum_{n=0}^{N-1} \left(e^{\imath 2\pi (k-m)/N} \right)^n = \frac{1 - \left(e^{\imath 2\pi (k-m)/N} \right)^N}{1 - e^{\imath 2\pi (k-m)/N}} \ . \tag{3.42}$$

But

$$\left(e^{\imath 2\pi (k-m)/N} \right)^N = e^{\imath 2\pi (k-m)} = 1 \tag{3.43}$$

so,

$$\sum_{n=0}^{N-1} \left(e^{\imath 2\pi (k-m)/N} \right)^n = \frac{1 - 1}{1 - e^{\imath 2\pi (k-m)/N}} = 0 \ . \tag{3.44}$$

Thus

$$\sum_{n=0}^{N-1} \left(e^{\imath 2\pi (k-m)/N} \right)^n = \begin{cases} N & (k-m) \text{ is a multiple of } N \\ 0 & \text{otherwise} \end{cases} \tag{3.45}$$

We are only interested in integer values of k and m between 0 and $N - 1$. Thus $k - m$ will only be a multiple of N when $k - m = 0$, and

$$\sum_{n=0}^{N-1} \left(e^{\imath 2\pi (k-m)/N} \right)^n = N \delta_{k,m} \ . \tag{3.46}$$

Returning to our original sum, and using the above result,

$$\sum_{n=0}^{N-1} x_n e^{-i2\pi nm/N} = \frac{1}{N} \sum_{k=0}^{N-1} X_k N \delta_{k,m} \tag{3.47}$$

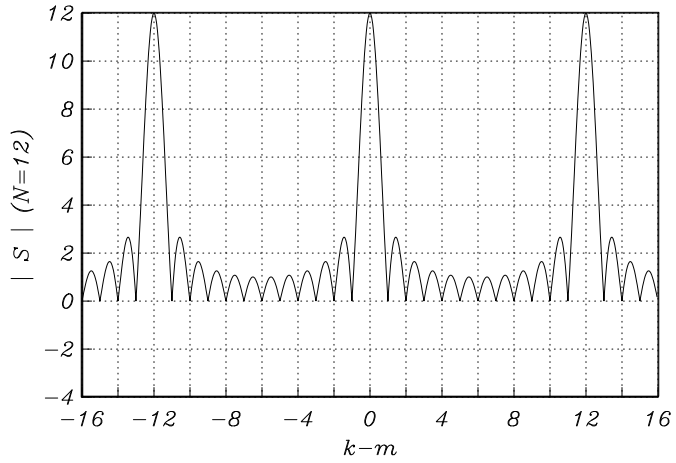

Figure 3.2: Amplitude of (3.38) as a function of $k - m$ for $N = 12$.

which reduces to

$$\sum_{n=0}^{N-1} x_n e^{-i2\pi nm/N} = X_m \ . \tag{3.48}$$

This derivation shows that DFT(IDFT(X_k)) = X_k. Similarly, it can now be easily confirmed that IDFT(DFT(x_n)) = x_n. An example DFT/IDFT pair is shown in Figure 3.3.

It is easy to see that the Discrete Fourier Transform of an N-periodic sequence also produces an N-periodic sequence

$$X_{m+N} = \sum_{n=0}^{N-1} x_n e^{-i2\pi n(m+N)/N} \tag{3.49}$$

or

$$X_{m+N} = e^{-i2\pi N/N} \sum_{n=0}^{N-1} x_n e^{-i2\pi nm/N} \ . \tag{3.50}$$

Since $e^{-i2\pi} = 1$,

$$X_{m+N} = \sum_{n=0}^{N-1} x_n e^{-i2\pi nm/N} = X_m. \tag{3.51}$$

The $k = 0$ term of the DFT is just N times the average value of x_n, while the N-periodicity of the DFT implies that

$$X_{N-k} = \sum_{n=0}^{N-1} x_n e^{-\imath 2\pi n(N-k)/N} = \sum_{n=0}^{N-1} x_n e^{\imath 2\pi nk/N} = X_{-k} \ . \tag{3.52}$$

Thus, the N periodicity here implies that the upper portion of the (N even) DFT sequence, $N/2 \le k \le N-1$, contains negative frequency spectral information, corresponding to $-N/2 \le k \le -1$ (Figure 3.4), while the lower portion contains positive frequency spectral information. If we wish to display an N-point DFT spectral sequence centered on the zero-frequency component (as we are used to picturing continuous Fourier transforms) we must therefore plot the DFT for $-N/2 \le k \le (N/2)-1$ (or $-(N-1)/2 \le k \le (N-1)/2$ for N odd) rather than $0 \le k \le N-1$, taking into account the above mapping. In MATLAB, there is an *fftshift* command that performs this rearrangement of the DFT coefficients.

Formulas for the DFT and its inverse can be written more compactly in terms of

$$w_N = e^{i2\pi/N}. \tag{3.53}$$

The DFT can be written as

$$X_m = \sum_{n=0}^{N-1} x_n w_N^{-mn}. \tag{3.54}$$

The inverse DFT becomes

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k w_N^{kn}. \tag{3.55}$$

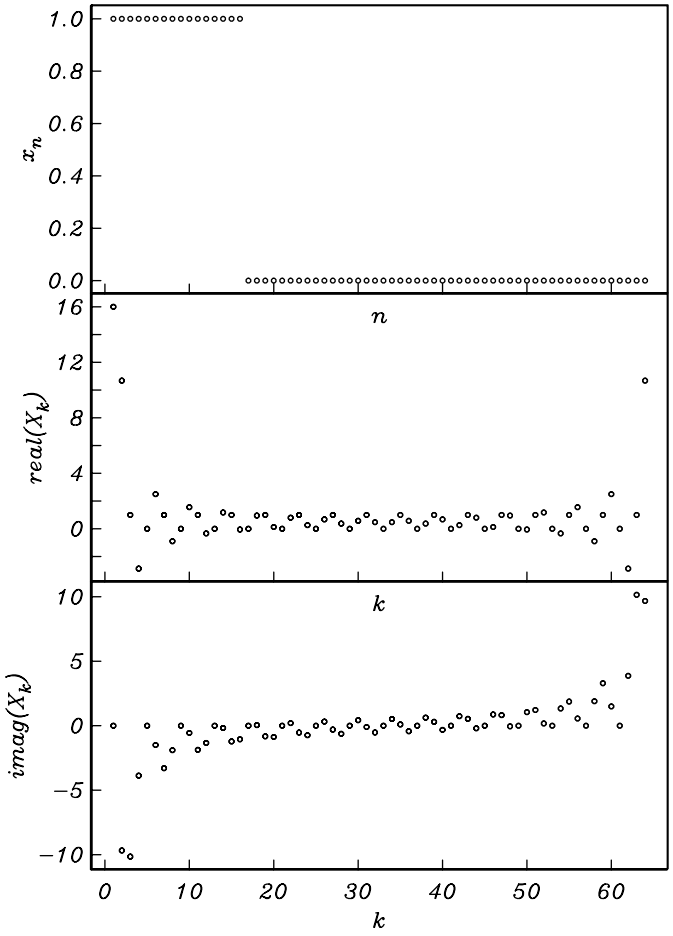

Figure 3.3: An example ($N = 64$) DFT.

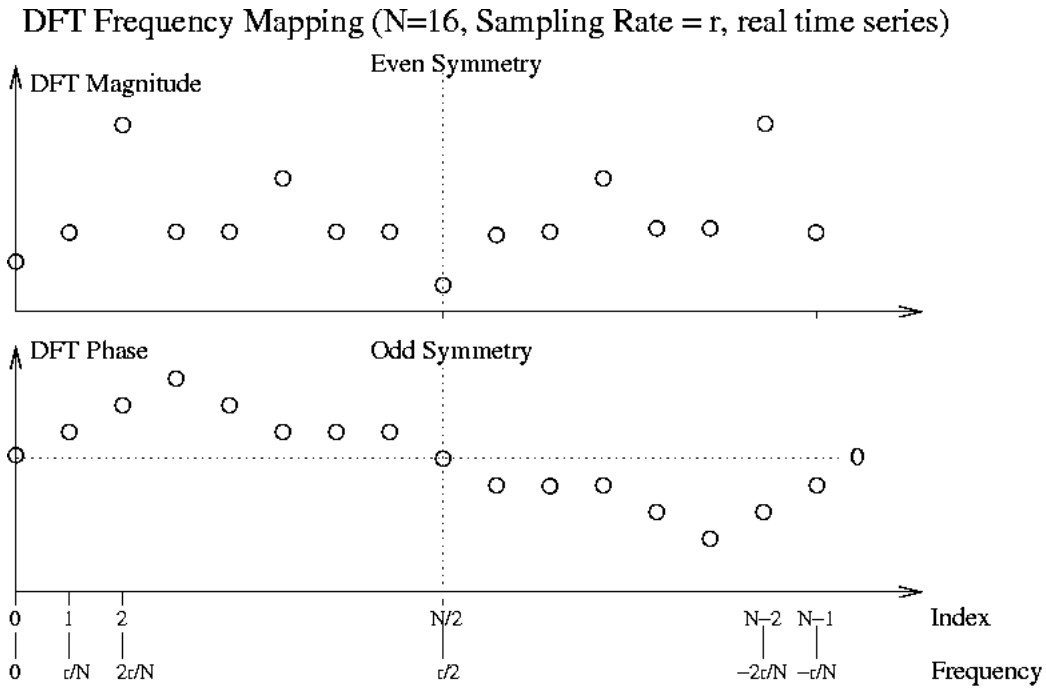

Figure 3.4: DFT frequency-index mapping.

The DFT and its inverse can also be written in matrix form as

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{N-1} \end{bmatrix} = \frac{1}{N} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w_N & w_N^2 & \cdots & w_N^{N-1} \\ 1 & w_N^2 & w_N^4 & \cdots & w_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w_N^{N-1} & w_N^{2(N-1)} & \cdots & w_N^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ \vdots \\ X_{N-1} \end{bmatrix} \tag{3.56}$$

and

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ \vdots \\ X_{N-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w_N^{-1} & w_N^{-2} & \cdots & w_N^{-(N-1)} \\ 1 & w_N^{-2} & w_N^{-4} & \cdots & w_N^{-2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w_N^{-(N-1)} & w_N^{-2(N-1)} & \cdots & w_N^{-(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{N-1} \end{bmatrix} \tag{3.57}$$

In the language of linear algebra, this shows that the DFT is a change of basis formula. It's also easy to show that the DFT basis is an orthogonal basis.

A summary of the discrete and continuous Fourier transform pairs defined here is given in Table 1. Here, (C , D) denote continuous or discrete and (P , A) denote periodic or aperiodic. Continuous periodic functions are assumed periodic on the unit interval and discrete periodic functions have period N.

$\phi(t)$	$\Phi(f)$	Transform	Forward Transform	Inverse Transform
C, A	C, A	Fourier Transform	$\Phi(f) = \int_{-\infty}^{\infty} \phi(t) e^{-\imath 2\pi f t} dt$	$\phi(t) = \int_{-\infty}^{\infty} \Phi(f) e^{\imath 2\pi f t} df$
C, P	D, A	Fourier Series	$\Phi_k = \int_{-1/2}^{1/2} \phi(t) e^{-\imath 2\pi k t} dt$	$\phi(t) = \sum_{k=-\infty}^{\infty} \Phi_k e^{\imath 2\pi k t}$
D, A	C, P	F.T. of a Sampled function	$\Phi(f) = \sum_{n=-\infty}^{\infty} \phi_n e^{-\imath 2\pi f n}$	$\phi_n = \int_{-1/2}^{1/2} \Phi(f) e^{\imath 2\pi f n} df$
D, P	D, P	DFT	$\Phi_k = \sum_{n=0}^{N-1} \phi_n e^{-\imath 2\pi k n/N}$	$\phi_n = \frac{1}{N} \sum_{k=0}^{N-1} \Phi_k e^{\imath 2\pi k n/N}$

Table 3.1: Four Discrete and Continuous Fourier Transform Pairs.

There are many results for the DFT that are analogous to results for the continuous Fourier transform. For example, the time shift theorem for the DFT is

$$\text{DFT}[x_{n-n_0}] = \sum_{n=0}^{N-1} x_{n-n_0} e^{-\imath 2\pi k n/N} \tag{3.58}$$

$$\text{DFT}[x_{n-n_0}] = \sum_{l=-n_0}^{N-n_0-1} x_l e^{-\imath 2\pi k (l+n_0)/N} \tag{3.59}$$

$$\mathrm{DFT}[x_{n-n_0}] = e^{-\imath 2\pi k n_0/N} \sum_{l=-n_0}^{N-n_0-1} x_l e^{-\imath 2\pi k l/N} \tag{3.60}$$

because of the periodicity of x_l, we can shift the summation limits to obtain

$$\mathrm{DFT}[x_{n-n_0}] = e^{-\imath 2\pi k n_0/N} \sum_{l=0}^{N-1} x_l e^{-\imath 2\pi k l/N} \tag{3.61}$$

or

$$\mathrm{DFT}[x_{n-n_0}] = e^{-\imath 2\pi k n_0/N} X_k \ . \tag{3.62}$$

Parseval's theorem for the DFT is

$$\sum_{n=0}^{N-1} |x_n|^2 = \sum_{n=0}^{N-1} x_n x_n^* \tag{3.63}$$

$$\sum_{n=0}^{N-1} |x_n|^2 = \frac{1}{N^2} \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} X_k e^{\imath 2\pi k n/N} \sum_{l=0}^{N-1} X_l^* e^{-\imath 2\pi l n/N} \tag{3.64}$$

$$\sum_{n=0}^{N-1} |x_n|^2 = \frac{1}{N^2} \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} X_k X_l^* e^{\imath 2\pi n(k-l)/N} \ . \tag{3.65}$$

Evaluating the sum over n first, using (3.46) gives

$$\sum_{n=0}^{N-1} |x_n|^2 = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} X_k X_l^* N \delta_{k,l} \tag{3.66}$$

which gives

$$\sum_{n=0}^{N-1} |x_n|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X_k|^2 \ . \tag{3.67}$$

Many properties of the continuous Fourier transform also apply to the DFT, but we must be careful, as the DFT applies to a periodic sequence, and *not* to a finite series surrounded by an infinite number of zeros, as we might at first be tempted to conceptualize from our experience with continuous time series.

A very important application of the DFT is in implementing the discrete counterpart of the convolution theorem. Suppose we are given x_n and y_n, what series has the DFT $Z_k = X_k Y_k$?

$$z_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k Y_k e^{\imath 2\pi k n/N}. \tag{3.68}$$

$$z_n = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} \sum_{m=0}^{N-1} x_l e^{-\imath 2\pi l k/N} y_m e^{-\imath 2\pi m k/N} e^{\imath 2\pi k n/N}. \tag{3.69}$$

$$z_n = \frac{1}{N} \sum_{l=0}^{N-1} x_l \sum_{m=0}^{N-1} y_m \sum_{k=0}^{N-1} e^{\imath 2\pi k(n-m-l)/N}. \tag{3.70}$$

The innermost sum is zero whenever $n-m-l$ is not a multiple of N by (3.38), so we get

$$z_n = \mathrm{IDFT}[\mathrm{X_k Y_k}] = \sum_{\mathrm{l}=0}^{\mathrm{N}-1} \mathrm{x_l}\ \mathrm{y_{n-l}} \tag{3.71}$$

where it is understood that z_n is N-periodic, as are x_n and y_n.

Because the functions that we manipulate with the DFT and IDFT are periodic, and because the length of a convolution will be greater than or equal to the maximum length of its two constituent series (and equal only in the case where one is a Kronecker delta function), it is possible to get perhaps unexpected effects when applying (3.71).

Suppose we convolve two aperiodic series x_n and y_m in the time domain to obtain the *serial product* (this is what the MATLAB *conv* function does). Further suppose that x and y have N and M contiguous non-zero terms, respectively. The convolution is then

$$z_n = x_n * y_n = \sum_{l=-\infty}^{\infty} x_l\ y_{n-l} \tag{3.72}$$

and will then have $N+M-1$ significant terms, bracketed by zeros.

What happens if we use the discrete convolution theorem to convolve the two functions? Here it is important to again keep in mind that the convolution theorem for discrete series corresponds to a convolution of periodic sequences. We must therefore take the period (i.e., the DFT size, L) to be longer than $N+M-1$, otherwise there will not be room to squeeze the $N+M-1$-length convolution result into an L-periodic result. We must therefore be careful to pad sequences with a suitable numbers of zeros to accurately mirror (3.72) using DFT techniques.

If $L < N+M-1$, we get generally undesirable *wraparound* effects and the result will be different from the serial product, especially in its tails. Because of this wraparound, (3.71) strictly applies to what is referred to as *cyclic*, or *circular* convolution (Figure 3.5). One way to avoid wraparound is to pad functions with zeros (e.g., Figure 3.6).

Why bother to use (3.71) rather than (3.72) to evaluate convolutions? A major incentive arises because of a set of computer algorithms which first emerged in the mid 1960's (e.g., Cooley and Tukey, "An Algorithm for the Machine Computation of Complex Fourier Series" [6]. These *Fast Fourier Transform* or *FFT* algorithms evaluate the DFT, but in a much faster manner than the straightforward application of (3.48). Because large DFT's can be efficiently calculated using the FFT algorithm, it is much more efficient to evaluate a convolution by computing two DFT's, multiplying them, and then taking the inverse transform of the result, rather than by evaluating the serial product.

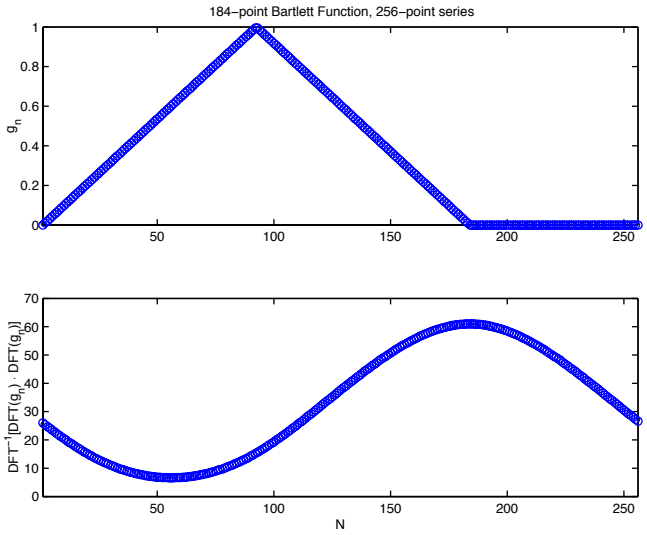

Figure 3.5: Wraparound in an $N = 256$-point Circular Convolution.

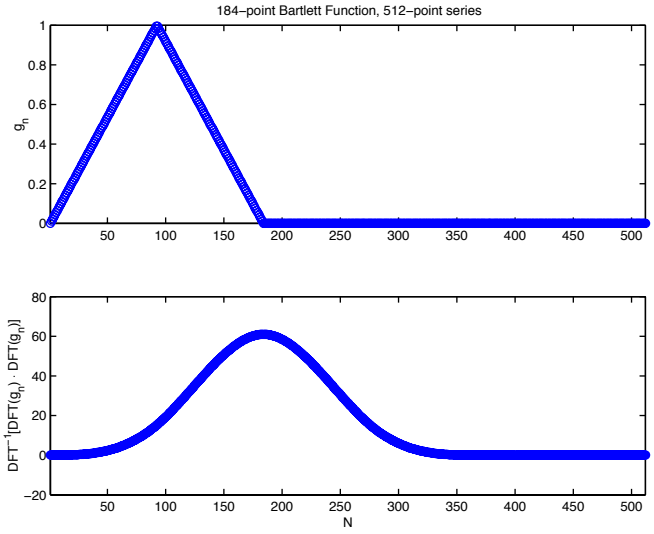

Figure 3.6: Same convolution as Figure 3.5, except with 256-point zero padding to eliminate wrap-around and thus emulate a noncircular convolution.

We will derive an FFT algorithm for the special case in which $N = 2^p$ is a power of 2. Similar ideas are used in algorithms that can work with more general values of N. Given a fixed input sequence x_n, Let

$$p(z) = x_0 + x_1 z + \ldots + x_{N-1} z^{N-1} \ . \tag{3.73}$$

Then the DFT of x is

$$X = \begin{bmatrix} p(w_N^0) \\ p(w_N^{-1}) \\ p(w_N^{-2}) \\ \vdots \\ p(w_N^{-(N-1)}) \end{bmatrix} \tag{3.74}$$

where $w_N = e^{\imath 2\pi/N}$.

We could use *Horner's rule* to evaluate the polynomial $p(z)$, which gives

$$p(z) = ((\cdots(x_{N-1}z + x_{N-2})z + x_{N-3})z + \ldots + x_1)z + x_0 \ . \tag{3.75}$$

Evaluating $p(z)$ in this way requires $N-1$ complex multiplications and $N-1$ complex additions. Computing the entire vector X takes N evaluations of $p(z)$, so computing the DFT in this manner takes $2N^2 - 2N = O(N^2)$ operations.

The FFT algorithm takes advantage of the fact that we are evaluating $p(z)$ only at powers of w_N^{-1}. We begin by breaking apart the even and odd powers in $p(z)$. Let

$$p_{\text{even}}(z) = \sum_{n=0 \ \ n \text{ even}}^{N-1} x_n z^{\frac{n}{2}} \tag{3.76}$$

and

$$p_{\text{odd}}(z) = \sum_{n=0 \ \ n \text{ odd}}^{N-1} x_n z^{\frac{n-1}{2}} \tag{3.77}$$

Then

$$p(z) = p_{\text{even}}(z^2) + z p_{\text{odd}}(z^2) \ . \tag{3.78}$$

For example, if $N = 8$, then

$$p(z) = x_0 + x_1 z + \ldots + x_7 z^7, \tag{3.79}$$

$$p_{\text{even}}(z) = x_0 + x_2 z + x_4 z^2 + x_6 z^3, \tag{3.80}$$

and

$$p_{\text{odd}}(z) = x_1 + x_3 z + x_5 z^2 + x_7 z^3. \tag{3.81}$$

Then

$$p_{\text{even}}(z^2) + z p_{\text{odd}}(z^2) = x_0 + x_1 z + \ldots + x_7 z^7 \ . \tag{3.82}$$

By using this decomposition of $p(z)$, we only need to evaluate z at even powers of w_N. Because of the periodicity of the powers of w_N, there are only

N	H_N	T_N
2	4	4
4	24	16
8	112	48
16	224	128
32	960	320
64	3972	768

Table 3.2: Operation counts for DFT by the naive algorithm (H_N) and FFT algorithm (T_N.)

$N/2$ points at which we have to evaluate the polynomial. For example, when $N = 8$, we need to evaluate $p(z)$ for

$$z = w_N^0, w_N^{-1}, \ldots, w_N^{-7}, \tag{3.83}$$

after removing multiples of $e^{i2\pi}$, the squares of these eight numbers are

$$z^2 = w_N^0, w_N^{-2}, w_N^{-4}, w_N^{-6}, w_N^0, w_N^{-2}, w_N^{-4}, w_N^{-6} \ . \tag{3.84}$$

Note that the even powers of w_N repeat twice. Thus we only need to evaluate p_{even} and p_{odd} at 4 points.

Let T_N be the number of arithmetic operations needed to evaluate the N point DFT. In other words, T_N is the number of arithmetic operations needed to evaluate an Nth degree polynomial $p(z)$ at w_N^0, w_N^{-1}, ..., $w_N^{-(N-1)}$. By using our formula, we can reduce this to 2 evaluations of polynomials of degree $N/2$ at $N/2$ points plus N multiplications and N additions. Thus

$$T_N = 2T_{N/2} + 2N \ . \tag{3.85}$$

We won't find an explicit solution to this recurrence relation. However, we can easily compute a table of values of T_N for small values of N that are powers of 2. Table 3.2 shows operations counts for the naive algorithm and our FFT algorithm. Clearly, the FFT becomes much more efficient as N gets larger. In fact, it can be shown that the growth of T_N is $O(N \log N)$, while the growth of H_N is $O(N^2)$. For long signals with thousands or millions of samples, the FFT is vastly more efficient than the naive algorithm.

Computation of the convolution of two sequences of length N takes $O(N^2)$ time by direct evaluation of the convolution formula. If we use the convolution theorem for the DFT, then we can do the job by zero padding the sequences to length $2N$, computing two FFT's of length $2N$, performing $2N$ multiplications, and then doing an inverse FFT of length $2N$. Since

$$T_{2N} = 2T_N + 4N \leq 6T_N, \tag{3.86}$$

T_{2N} is $O(N \log N)$. Thus the entire FFT convolution process takes $O(N \log N)$ operations.

If we do not have large aliasing effects, so that the sampled sequence, x_n, adequately characterizes some near-band-limited continuous function in the real world, $\phi(t)$, then the DFT of the sequence x_n is just the spectrum of $\phi(t)$, sampled at the N equally-spaced frequency points. As we go to finer and finer sampling, we expect our calculated spectrum to approach the true spectrum, $\Phi(f)$. One way to help see that this is true (in a somewhat nonrigorous way) by considering the DFT when N becomes large.

It's instructive to investigate the convergence of the DFT to the Fourier Transform. Consider a discrete function defined by the N-point sequence, x_n. Taking the N-point DFT, where we'll take N to be odd, $N = 2M + 1$, we get

$$X_k = \sum_{n=-M}^{M} x_n e^{-\imath 2\pi kn/N} \tag{3.87}$$

Heuristically in the limit as N approaches infinity (finer and finer sampling), n remains discrete, but the function becomes aperiodic (we might conceptualize that it occupies the entire number line) and k thus become continuous (Table 4.1). The Fourier transform is thus

$$X(f) \equiv \sum_{n=-M}^{M} x_n e^{-\imath 2\pi fn} \ . \tag{3.88}$$

As a special case, consider the discrete rectangle function

$$\Pi_n = \begin{cases} 1 & |n| \le M \\ 0 & |n| > M \end{cases} \tag{3.89}$$

Taking the Fourier Transform of (3.89), where $N = 2M + 1$, we get

$$\Pi(f) = \sum_{n=-M}^{M} e^{-\imath 2\pi fn} = e^{\imath 2\pi fM} \sum_{n=0}^{2M} e^{-\imath 2\pi fn} \tag{3.90}$$

$$= e^{\imath 2\pi fM} \frac{1 - e^{-\imath 2\pi (2M+1)f}}{1 - e^{-\imath 2\pi f}} = \frac{\sin(N\pi f)}{\sin(\pi f)} \equiv D(f) \tag{3.91}$$

Expressions of the form of (3.91) are a discrete, periodic analogue to the sinc function, occur frequently in discrete Fourier theory (e.g., in the kernel of the multitaper eigenfunction equation we noted in discussing power spectra. Such functions are commonly referred to as *Dirichlet kernels*. When N is large, the numerator of (3.91) oscillates much more rapidly than the denominator. Making the substitution $y = Nf$, (3.91) indeed then approaches the sinc function:

$$\lim_{N\to\infty} D(f) = \lim_{N\to\infty} \frac{\sin(\pi y)}{\sin(\pi y/N)} = \frac{\sin(\pi y)}{\pi y/N} = N \ \mathrm{sinc} \ \ y \ . \tag{3.92}$$

Chapter 4

Spectral Analysis

Energy and Power Spectra

It is frequently valuable to study the power distribution of a signal in the frequency domain. For example, we may wish to have estimates for how the power in a signal is distributed with frequency, so that we can quantitatively state how much power is in a particular band of interest relative to other frequencies. Power peaks and/or troughs across specific frequency ranges may reveal important information about a physical process. Given a power spectral density function, the power across any range of frequencies can then be estimated by integrating such a function over the band of interest.

The simplest such measure of energy (or, with scaling modifications, power) in a signal as a function of frequency is the energy spectral density, which is just the square of the spectral amplitude

$$|\Phi(f)|^2 = \Phi(f)\Phi^*(f) \ . \tag{4.1}$$

Applying the convolution theorem, and noting that phase conjugation in the frequency domain corresponds to reversal in the time domain, this can be recognized as the Fourier transform of the autocorrelation

$$\Phi(f)\Phi^*(f) = \mathcal{F}[\phi(t) * \phi^*(-t)] = \mathcal{F}[\phi(t) \text{ cor } \phi^*(t)] \ . \tag{4.2}$$

We can thus observe that function which has a sharp and narrow autocorrelation function will have a broad energy spectral density, while a function which has a broad autocorrelation function will have a narrow energy spectral density. This can perhaps be understood better by considering what is in fact required of a time-domain function for it to have narrow (in the limit, delta-like) autocorrelation function; the function must change rapidly, so that it does not resemble itself very much for a small shift from zero lag. For a function to change rapidly, it must have high frequency energy in its spectrum. Note that, because the units of a spectrum are $u \cdot \mathrm{s} = u/\mathrm{Hz}$, the units of (4.1) are u^2/Hz^2, where u denotes the physical units of $\phi(t)$ (e.g., Volts, Amperes, meters/s, etc.).

Many interesting signals, such as those arising from an incessant excitation, are, practically speaking, unbounded in time or *continuous* (as opposed to signals that are limited in time, or *transient*). If the statistical behavior of the signal (we will look at statistical aspects of time series much more later on in the class) doesn't change with time, so that the spectral and other properties of the signal are time-invariant, it is generally referred to as *stationary*. Examples where signals can often be considered to be stationary may include seismic, thermal, or electromagnetic noise, tides, winds, temperatures, and currents. Some signals of interest exhibit strong periodicities (tides, for example) because they are associated with astronomical or other periodic forcing. Because of their incessant nature, such signals have infinite total energy

$$E_T = \lim_{T->\infty} \int_{-T/2}^{T/2} |\phi(t)|^2 \, dt = \infty, \tag{4.3}$$

so that the Fourier transform of the autocorrelation (4.2) won't converge. The frequency content of such signals may, however, still be examined using *power spectral density*, or simply *PSD*.

Signal power averaged over some interval T is simply the energy (4.3) normalized by the length of the observation

$$P_T = \frac{1}{T} \int_{-T/2}^{T/2} |\phi(t)|^2 \, dt = \frac{1}{T} \int_{-\infty}^{\infty} |\phi(t) \cdot \Pi(t/T)|^2 \, dt \; . \tag{4.4}$$

As the observation interval T becomes long, this converges to the true signal power

$$P = \lim_{T\to\infty} P_T \; . \tag{4.5}$$

The PSD is defined as

$$PSD[\phi(t)] = \lim_{T\to\infty} \frac{1}{T} \Phi_T(f) \cdot \Phi_T^*(f) \tag{4.6}$$

where the time series has been *windowed* by multiplying the time series with a boxcar function of unit height and length T, so that

$$\Phi_T(f) = \mathcal{F}[\phi(t) \cdot \Pi(t/T)] \; . \tag{4.7}$$

Note that dimensional analysis shows that the units of the power spectral density in (4.6) are u^2/Hz. Further note that that PSDs will be real, symmetric functions over f for the common case where $\phi(t)$ real (and thus has a Hermitian spectrum). For this reason, as we noted for the complex Hermitian spectrum in considering the various Fourier symmetry relationships, the power spectra of real functions are typically plotted only for positive frequencies. Because we can never do calculations on an infinite-length signal, all PSDs in practice are *estimates* of P that we hope approach the "true" PSD for the continuous and (conceptually) time-infinite signal that we are studying.

The simplest (but definitely not the best!) way to estimate a PSD is to simply truncate the data with a T-length rectangular time window extending

across a time interval that we can define as being between $-T/2$ to $T/2$. This estimate, because it seems like the obvious thing to do, has a long history, and it is sometimes referred to as a *periodogram*. To understand the relationship between the periodogram estimate and the true PSD (4.6), note that for a rectangular window of width T and a real-valued time series (which, again, has a Hermitian spectrum)

$$PSD_{periodogram} = \frac{1}{T}|\Phi_T(f)|^2 = \frac{1}{T}|\mathcal{F}[\phi(t)\Pi(t/T)]|^2 \tag{4.8}$$

Using the convolution theorem, this gives

$$PSD_{periodogram} = \frac{1}{T}|\Phi(f) * \text{sinc } (Tf)|^2 \tag{4.9}$$

where, recall, the Fourier transform of $\Pi(t/T)$ is

$$\text{sinc } (Tf) = \frac{\sin \pi T f}{\pi T f} \ . \tag{4.10}$$

Thus, what we obtain in a periodogram estimate is the true PSD of the process convolved in the frequency domain with the sinc (Tf) function. Figure 4.1 shows such a periodogram estimate for a sinusoidal process. The underlying process has a delta function spectrum, with the delta function centered on the frequency of the sinusouid. However, (4.9) shows in a broader peak in the PSD estimate. The spearing effect of the convolution produced a limited *spectral resolution* view of the sinusoidal process.

The loss of resolution caused by the convolution in (4.9) is undesirable, and we typically want to minimize and characterize it. As convolution is essentially a smoothing operation (recall that variances add when we convolve two functions, thus increasing their spread), our windowed estimate in (4.9) is a blurred image of the true spectrum. In the periodogram case, this blurring takes the specific form of convolution with a sinc function because we chose an (abrupt) boxcar data truncation on the $\pm T/2$ interval, and the Fourier transform of the boxcar function is a sinc. The sinc function's slow ($(Tf)^{-1}$) fall-off and oscillatory side lobes are easily improved by modifying the estimation method, and the periodogram should thus never be used in practice except for quick and dirty estimates of the PSD.

The smearing of spectral resolution due to the convolution of the true spectrum with the Fourier transform of the windowing function is called *spectral leakage*, as the frequency domain convolution in (4.9) causes power from surrounding frequencies to "leak" into the estimate at any particular frequency. In its simplest form, spectral leakage in the periodogram will make the PSD estimate for a function that is really a sinusoid of frequency f have the appearance of sinc functions centered on the true frequencies ($\pm f$) of the continuous signal, rather than the true delta functions.

Spectral leakage can be reduced by increasing T, so that the Fourier transform of the windowing function becomes reciprocally (by a factor of $1/T$) narrower. However, for statistical reasons involving the variance of the estimate

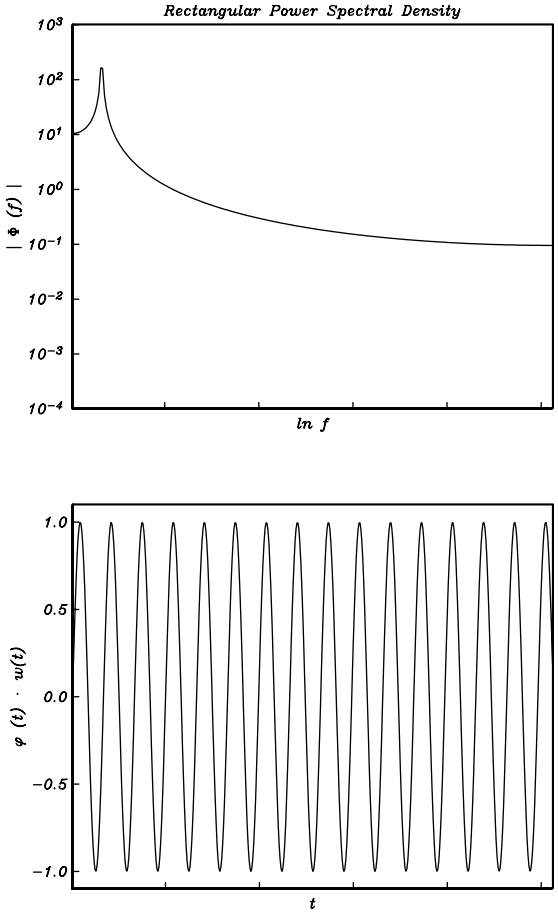

Figure 4.1: A Periodogram Estimate of a Pure Sinusoidal Process

that we will not elaborate on here, this is still a poor way to estimate the PSD. A better way to reduce spectral leakage, at the cost of eliminating the statistical contributions of data near the endpoints of the data series, is to window with a smoother time function than the boxcar that has a Fourier transform that is more delta-like by some measure. For example, consider the *Bartlett* or *Parzen* window

$$\Lambda(2t/T) = 4/T^2 \left(\Pi(2t/T) * \Pi(2t/T)\right) \tag{4.11}$$

which is a unit height triangle function spanning the interval $-T/2$ to $T/2$. $\Lambda(2t/T)$ is easily seen by the convolution theorem to have a Fourier transform given by the sinc function squared

$$\mathcal{F}[\Lambda(2t/T)] = 4/T^2 \mathcal{F}[\Pi(2t/T) * \Pi(2t/T)] = \text{sinc}^{\,2}(fT/2) \tag{4.12}$$

which falls off asymptotically as $(Tf)^{-2}$ and is positive everywhere (although it is still oscillatory; Figure 4.2).

The formulation of various data windows such as the Parzen window has historically formed a rich are of research (if not a veritable cottage industry) in signal processing, and numerous function are in common usage (may of which can be readily generated using various MATLAB functions in the signal processing toolbox). The general tradeoff in window selection arises between the width of the main lobe of the leakage function and the rate of decay away from the center frequency. A few examples of commonly used windows and their corresponding spectral leakage properties when they are applied to a true sinusoidal signal (which, again, has a "true" delta function PSD), are shown in the following figures.

An interesting issue in spectral estimation that arises from the use of windows is the "throwing out" of data resulting from tapering near the data segment endpoints. The result is that we are downweighting information and thus increasing the statistical uncertainty of the PSD estimate. For long, stationary time series, one straightforward and widely-applied method of addressing this issue is to evaluate a suite of either overlapping or nonoverlapping spectral estimates for a host of window locations, and to subsequently average them and calculate statistical bounds on the mean estimate. The most commonly used technique along these lines is called *Welch's Method* (see the *pwelch* function in the MATLAB signal processing toolbox).

An elegant, more computationally-intensive, and increasingly widely utilized method of estimating spectra (see the *pmtm* function in MATLAB's signal processing toolbox) is *multitaper spectral estimation* [13]. In multitaper spectral estimation, a family of statistically independent spectral estimates is obtained from a signal using an orthogonal set of windows on the estimation interval that are referred to as *prolate spheroidal tapers* (Figure 4.6).

In multitaper spectral estimation individual spectra obtained from the prolate spheroidal tapers are combined in a weighted sum to produce a spectral estimate with leakage that is approximately limited to some specified frequency band, $\pm W$. Specifically, for a specified time-bandwidth product, NW, the multitapers are the Fourier Transforms of solutions, U_k, to the frequency-domain

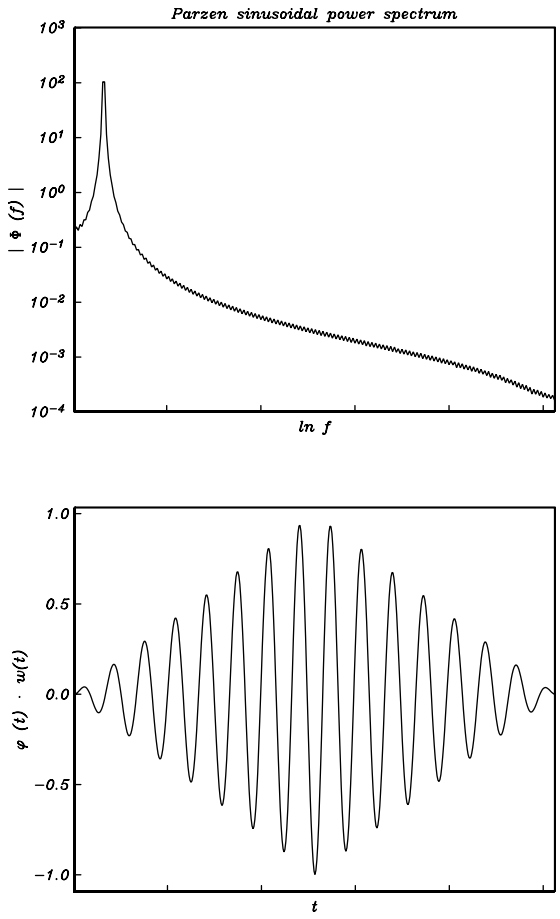

Figure 4.2: A Bartlett or Parzen window estimate of a pure sinusoidal process spectrum.

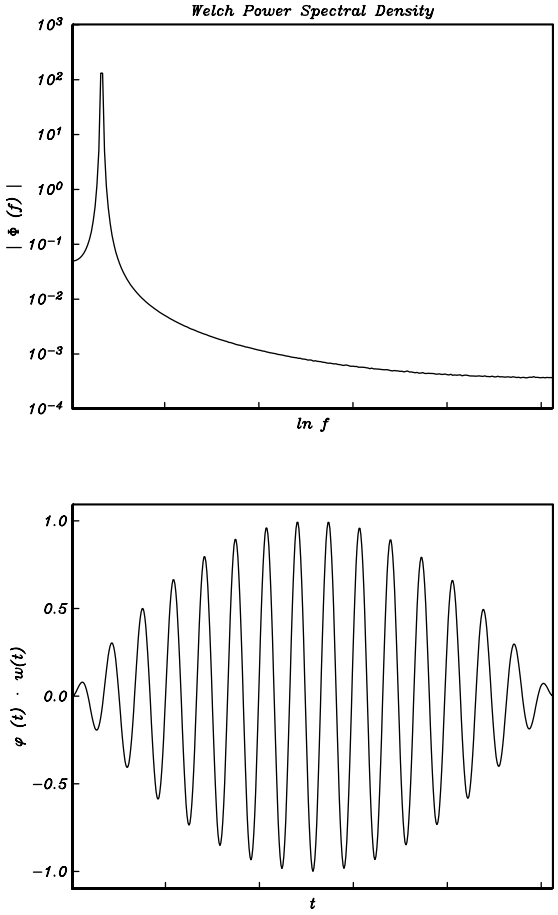

Figure 4.3: A Welch window estimate of a pure sinusoidal process spectrum.

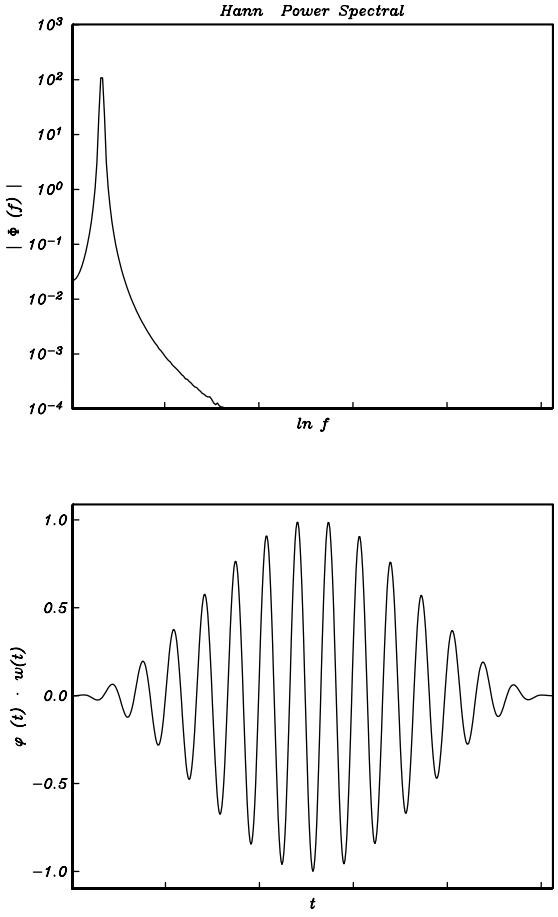

Figure 4.4: A Hann window estimate of a pure sinusoidal process spectrum.

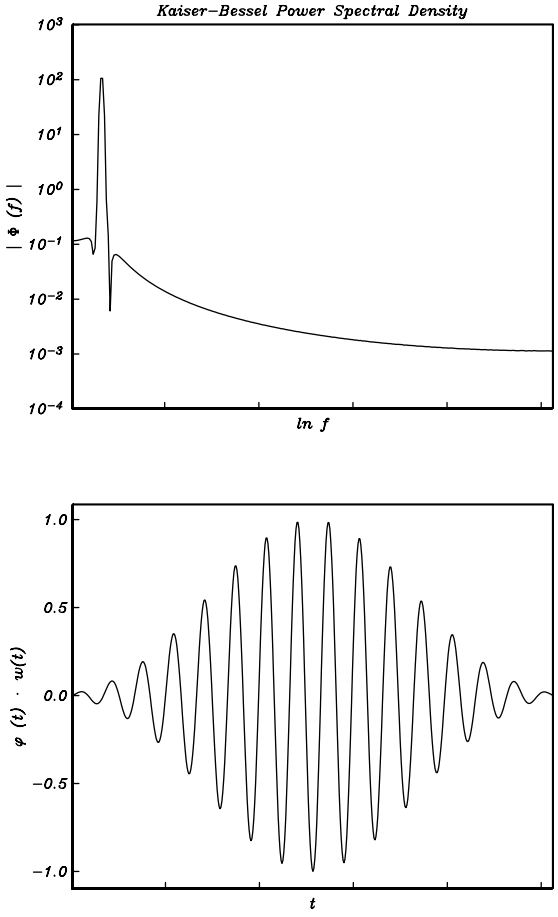

Figure 4.5: A Kaiser-Bessel window estimate of a pure sinusoidal process.

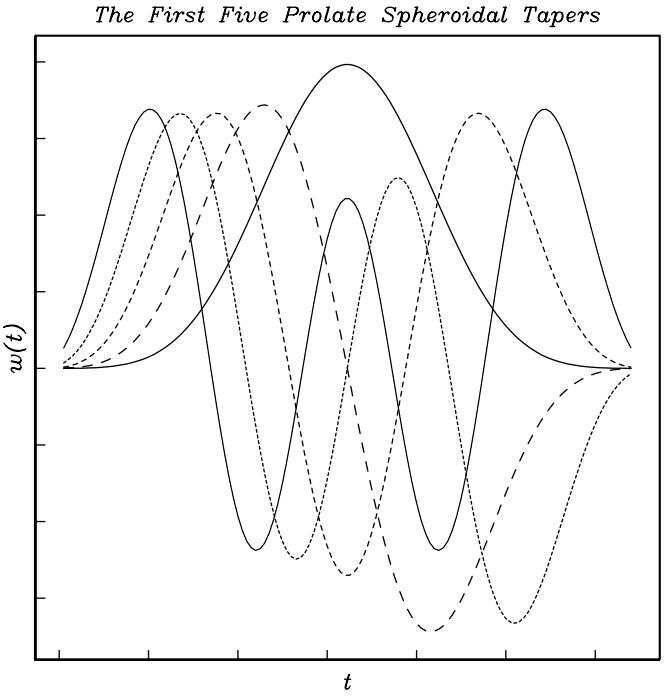

Figure 4.6: Prolate Spheroidal Taper Functions ($0 \leq k \leq 4; NW = 4$).

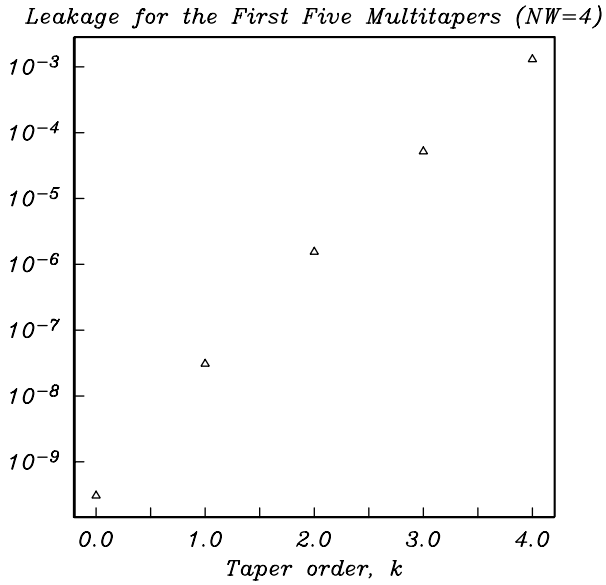

Figure 4.7: Fractional energy leakage outside of $f = (-W, W)$ for the first five multitapers ($NW = 4$).

eigenvalue-eigenfunction equation

$$\int_{-W}^{W} \frac{\sin N\pi(f-f')}{\sin \pi(f-f')} U_k(N,W;f')\, df' = \lambda_k(N,W) \cdot U_k(N,W;f)\ . \qquad (4.13)$$

where the λ_k are eigenvalues (the first $2NW$ of which are close to one), and N is the discrete length of the taper sequence (this is a discrete formulation for spectral estimation on sampled time series, which we shall discuss next shortly. The integral in (4.13) is a convolution in the frequency domain between the U_k and the Dirichlet kernel, a function that arises frequently in discrete Fourier analysis because it is the Fourier transform of the sampled counterpart of the boxcar function (more on this later). Solutions to (4.13) form an orthogonal family of functions which have the greatest fractional energy concentration in the frequency interval $(-W, W)$. The eigenvalues in (4.13) are measures of the degree to which spectral leakage is confined to $(-W, W)$. Spectral leakage becomes increasingly worse for higher-order tapers, with the energy leakage being given approximately as

$$1 - \lambda_k \approx \frac{\sqrt{2\pi}}{k!}(8c)^{k+1/2} e^{-2c} \qquad (4.14)$$

where $c = \pi NW$. Figure 4.7 shows the fractional leakage for the first five multitapers.

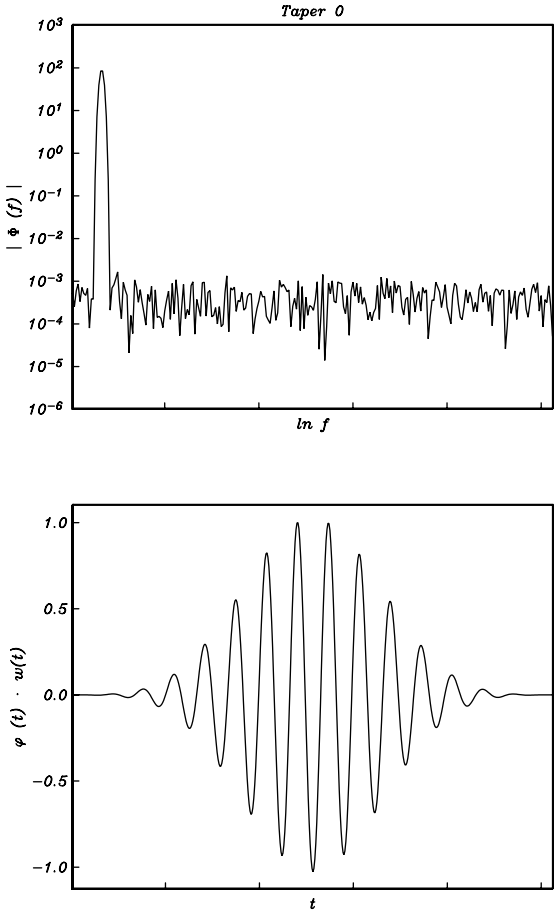

Figure 4.8: Prolate spheroidal taper spectral estimate ($k = 0$; $NW = 4$).

Because of the appreciable leakage of the higher order tapers the lowest few (typically six or so, depending on the values of N and W) are typically used in practice. Figures 4.8 through 4.12 show the five lowest order multitaper estimates for $NW = 4$ for the example sine wave signal used in the earlier figures. Figure 4.13 shows the multitaper estimate obtained by averaging them. The leakage function displayed in Figure 4.13 approximates a frequency boxcar of width $2W$.

An example geophysical application of the PSD is to quantify the background noise characteristics of seismic stations, so as to gauge, for example, how they compare to known very quiet sites, and to assess what frequency bands good or bad for signal detection. This is of considerable importance both for earthquake and Earth structure studies and for estimating detection thresholds for clandestine events (e.g., nuclear tests). Figure 4.14 shows PSD estimates for a

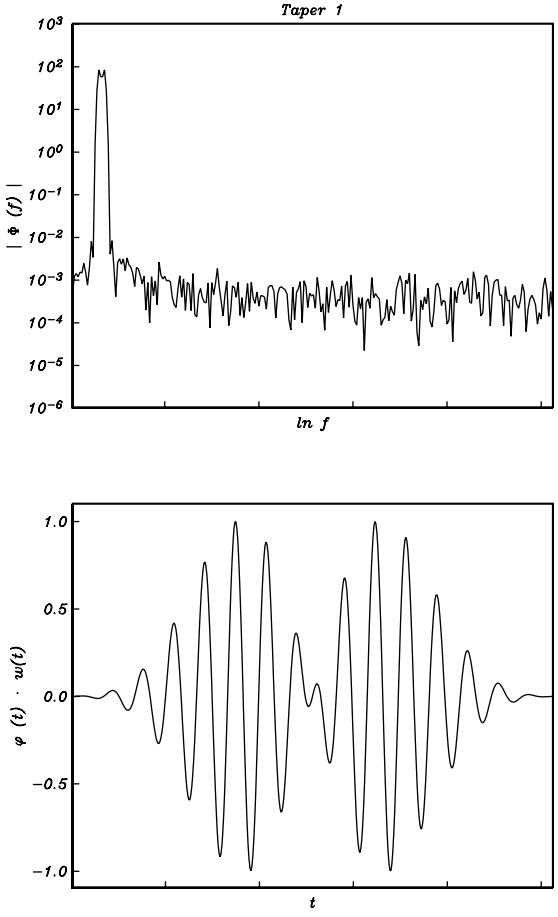

Figure 4.9: Prolate spheroidal taper spectral estimate ($k = 1$; $NW = 4$).

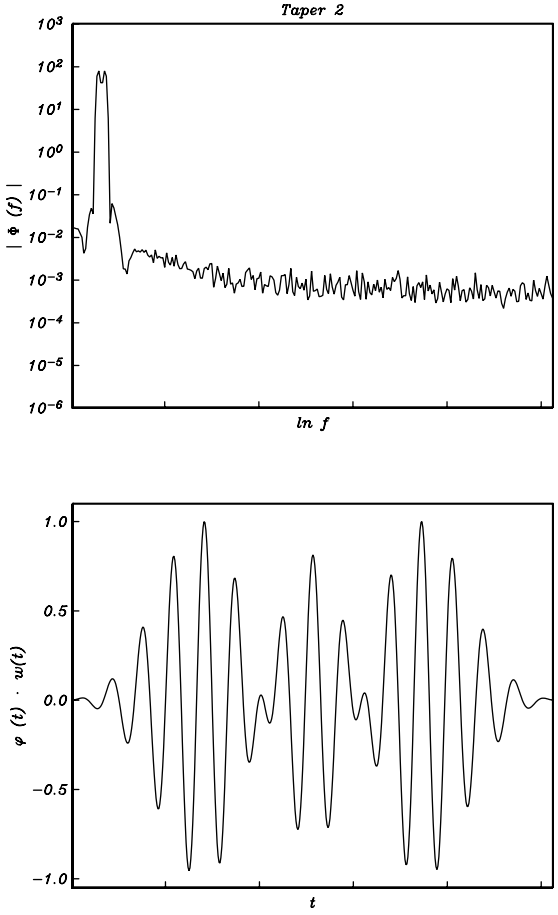

Figure 4.10: Prolate spheroidal taper spectral estimate ($k = 2$; $NW = 4$).

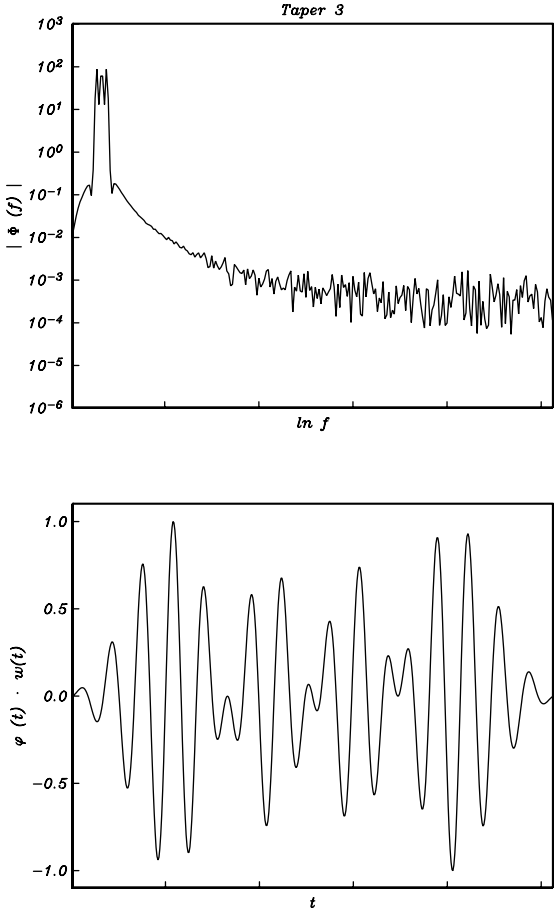

Figure 4.11: Prolate spheroidal taper spectral estimate ($k = 3$; $NW = 4$).

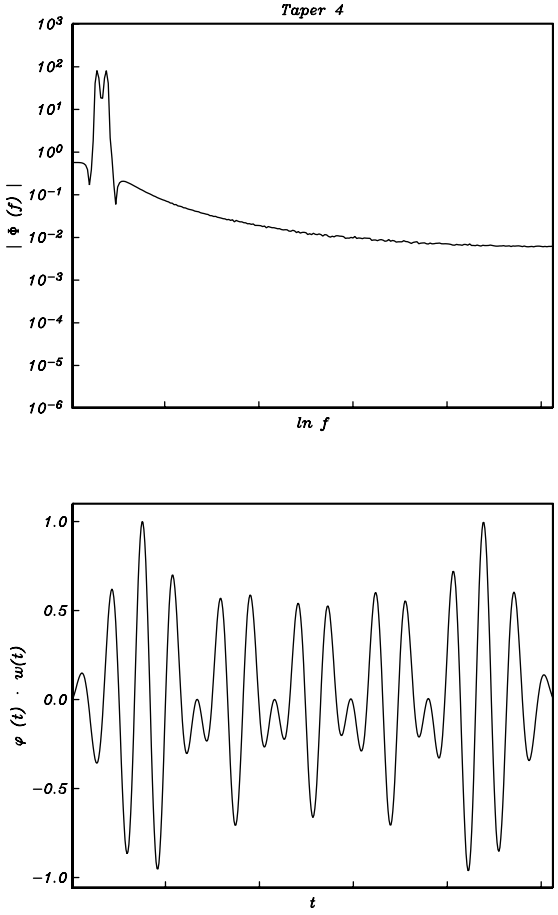

Figure 4.12: Prolate spheroidal taper spectral estimate ($k = 4$; $NW = 4$).

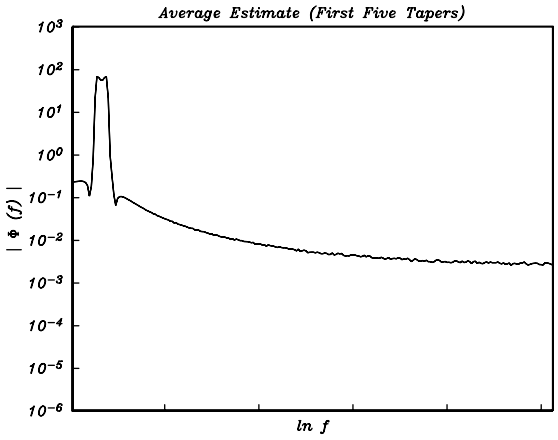

Figure 4.13: Prolate spheroidal taper average spectral estimate ($0 \leq k \leq 4$; $NW = 4$).

fairly quiet IRIS broadband seismic station in the Tien Shan mountains near Ala Archa, Kyrgyzstan, at periods ranging from 0.1 to 10^3 s (about 17 minutes). The bounding curves are empirically-based high- and low-noise extremal models for broadband stations. Noise at short periods is dominated by cultural (man-made), wind, and other rapidly varying environmental effects. The prominent noise peaks near 7 and 14 seconds are globally observed and are generated by ocean waves. The long-period power is higher on the horizontal sensors as opposed to the vertical sensors because they are sensitive to tilt caused by barometric, thermal, or other long-period noise sources. The peak near 1.6 s is unusual and may represent microseismic wave noise from the nearby Issyk Kul, one of the largest high-altitude alpine lakes in the world.

As a final indication of the great utility of the PSD, the Figure (4.15) shows processed PSDs from a broadband seismometer (Guralp CMG-3Tb) located in a 255-m deep borehole in the polar icecap near the South Pole. A great many of 1-hour data length, 50% overlap, PSDs using a hamming taper, were calculated from the month of May, 2003, and the resulting individual PSDs were used to assemble an empirical probability density function for the signal characteristics at he station The bifurcation of the high frequency noise is caused by intermittent periods where tractors are moving snow near the station. Pink misty areas concentrated around 1 and 20 s are PSDs that include teleseismic earthquake signals. At short periods this is among the quietest stations on Earth.

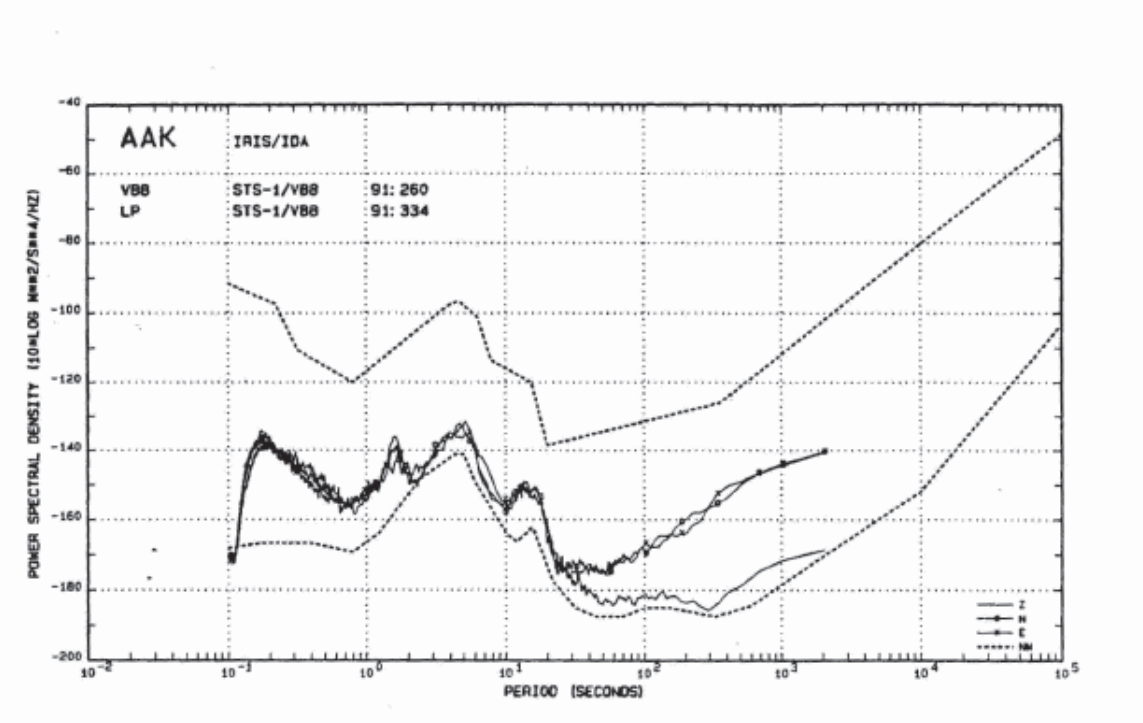

Figure 4.14: Earth Acceleration Power Spectral Density for background noise at the Ala Archa IRIS/IDA station as a function of period. Z, N, E refer to vertical, north, and east seismometer components. Curves labeled NM are the empirical noise model bounds of Peterson (1994) denoting to extremal PSD values from stations installed around the world. The reference (0 db) level is $(1\ \mathrm{m/s^2})^2/\mathrm{Hz}$. PSD estimates were obtained using Welch's method.

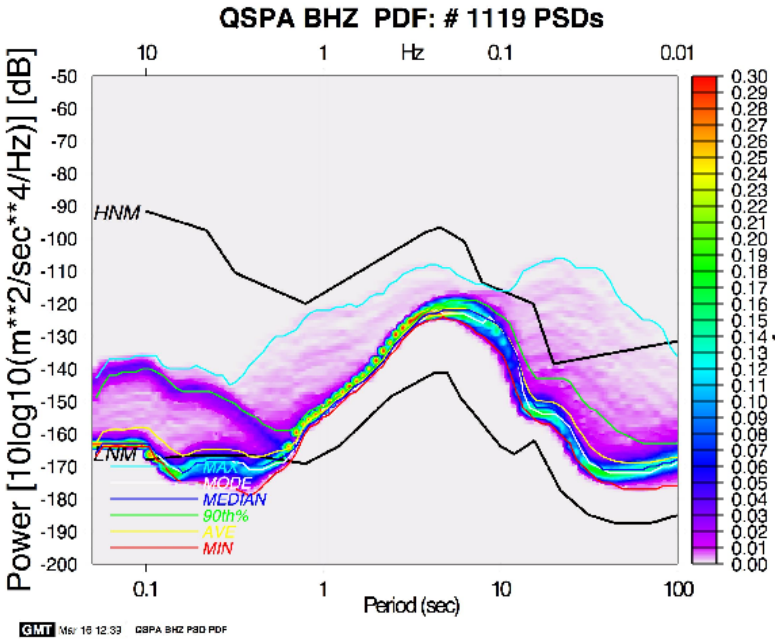

Figure 4.15: Quiet South Pole (QSPA) Global Seismic Network Station, power spectral density probability density plot

Chapter 5

Digital Filtering

Digital Filtering

We next turn to the (very broad) topic of how to manipulate a sampled signal to alter the amplitude and/or phase of different frequency components of the signal. There is an incredible amount of literature on this subject, and we will only be able to scratch the surface here. There are many reasons for wanting to filter a signal including:

1. Noise rejection/Signal enhancement
2. Remove an instrument response from the signal
3. Differentiate or integrate
4. Change the sampling rate
5. Artistic effects in audio and video.

Some particular types of filters that we will look at include **low pass**, **high pass**, and **band pass** filters which cut off portions of the frequency spectrum while allowing other frequencies to pass through the filter. In implementing these filters we will want to achieve the desired frequency response while otherwise distorting the signal as little as possible.

We will concentrate our analysis on filters which are themselves linear time invariant systems. This will enable us to apply all of the techniques that we have previously developed for LTI systems to analyze the performance of the filter. However, it is also possible to construct more complicated nonlinear filters which are not LTI systems.

In analyzing filters we will again encounter the concept of **stability**. A linear filter is stable if the corresponding LTI is stable. If we try to use an unstable filter, we're likely to find that small amounts of noise in the input build up in the output to intolerable levels. Unstable filters are practically impossible to use.

In some cases a filter is designed to process the signal in **real time** with little or no delay, while in other cases we must receive the entire signal before we can begin processing it. There is generally a preference for linear filters which are causal, since these filters can be implemented in real time. If an acausal filter needs to look ahead only a few samples, then we can implement the filter in real time by simply inserting a buffer before the filter and allowing the filter to delay its output by a few samples.

In practice, much of the human effort that goes into designing and implementing digital filters is about tradeoffs between the desired frequency and phase response of the filter and difficulty and cost of the implementation of the filter.

Filtering by Direct Manipulation of the FFT

One very simple approach to filtering a sampled signal is to compute its FFT and then manipulate the individual components of the FFT to achieve a desired frequency or phase response. This approach gives us complete freedom to control the frequency and phase response of the filter.

There are two significant costs associated with implementing a filter in this fashion. The first problem is that computing the FFT of a signal can (depending on the sampling rate and time duration of the signal) be very computationally intensive. For very long signals, computing the FFT of the entire signal may not even be practical. The second issue is that since we must have the entire signal in hand before we can begin filtering, we cannot use this method for real time filtering.

For example, consider a 20 second long recording of some guitar music. (The audio clip and MATLAB codes for this example will be made available on the class web site.) The audio is digitized according to the consumer audio CD standard at a sample rate of 44.1 Khz, with 16 bits per sample. For this example, we've combined the left and right channels into one mono channel. Thus the 20 seconds of audio requires $20 \times 44100 \times 16$ bits, or 1.76 megabytes of storage. We'll store the samples in MATLAB as 8 byte double precision numbers, which expands the storage requirements by a factor of four to about 8 megabytes. This is fairly large, but still well within the memory size limits of our computers.

At the sampling rate of 44.1 Khz, the Nyquist frequency is 22.05 Khz. Is a practical matter, most of us can't hear (and the speakers in our classroom can't reproduce) much above about 15 Khz. Before sampling, this signal was passed through an analog anti-aliasing filter that eliminated all frequencies above 22.05 Khz. Thus the sampling rate has been chosen to effectively reproduce all of the frequencies in the original music while avoiding aliasing problems.

We read the signal into MATLAB and compute its FFT. Since there are 882,000 real values in the original signal, the FFT also has 882,000 complex components. Since there are 882,000 frequency components over a frequency range of 0 to 44,100 Hz, each component of the FFT represents a frequency range of 0.05 Hz. Figure 5.1 shows a plot of the absolute values of the FFT

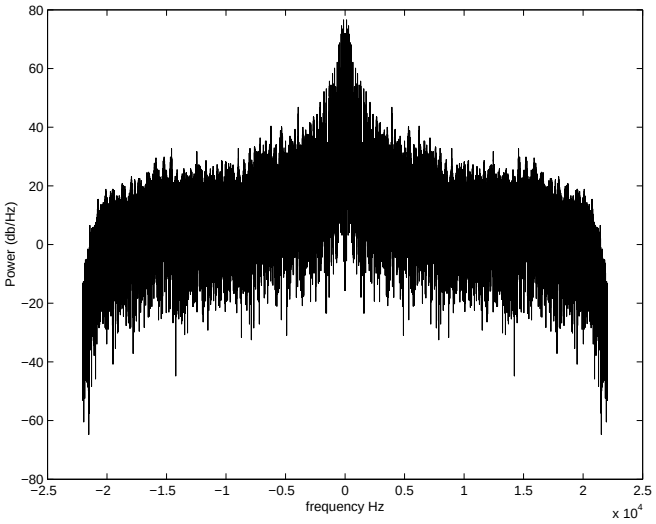

Figure 5.1: Spectrum of the original signal.

versus frequency. The vertical axis represents power. We have used a dB scale. The horizontal access represents frequency in Hz. For convenience, we have used the MATLAB command **fftshift** to rearrange the entries in the FFT so that 0 Hz is at the center of the spectrum.

Now, suppose that we want to low pass filter the signal, eliminating all frequency components above 2 Khz. We do this by simply setting to 0 those elements of the FFT that correspond to frequencies above 2 Khz. Figure 5.2 shows a plot of the revised spectrum. At all frequencies below 2 Khz, we've left the spectrum alone, while at all frequencies above 2 Khz, we've zeroed out the entries in the FFT.

We can play the filtered signal, and hear that it sounds much like the original recording, but somewhat "dull." The guitar notes are at frequencies between about 400 Hz and 1 Khz. However, as a guitar string plays a note, the string also vibrates at multiples of the base frequency. These **harmonics** are what give the guitar its particular tone. By filtering out the harmonics, we've effectively dulled the tone of the guitar.

We can also try filtering out the fundamental frequencies and just listen to the higher harmonics. Figure 5.3 shows the spectrum after filtering out everything below 1 Khz. When you listen to the playback of the filtered signal, you'll still be able to hear the original music, because the harmonics still carry the tune.

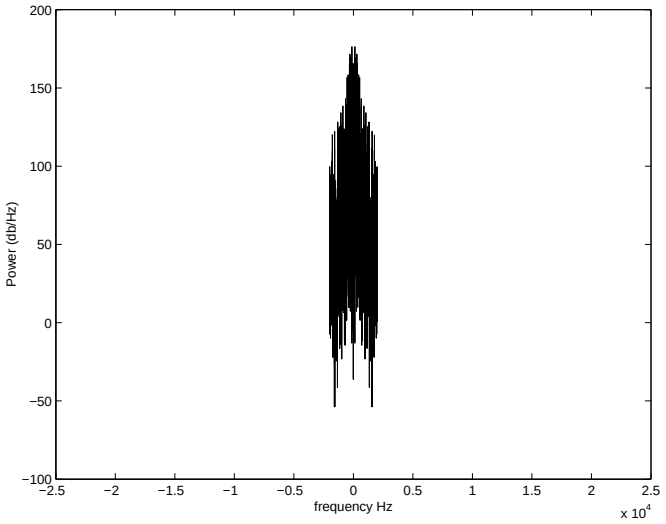

Figure 5.2: Spectrum after low pass filtering.

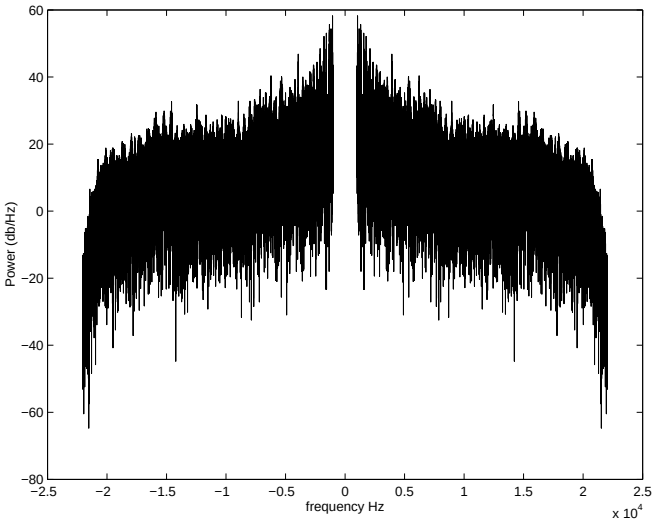

Figure 5.3: Spectrum after high pass filtering.

Phase Shifts

So far, we've only adjusted the amplitude of various frequency components in the FFT. A filter which doesn't change the phases of any of the components of the FFT is called a **zero–phase filter**. Hidden within the phase of the complex numbers in the FFT is the information about when the various notes appear in the signal. Adjusting the phases of the FFT components can do some interesting things to the signal.

Continuing our example, we'll set the phase of each complex element of the discrete spectrum to zero by taking the absolute value of each component. When played back, the resulting gong-like signal bears little resemblance to the original signal (although its amplitude spectrum is identical). The original frequencies are all present, but the order of the notes has disappeared completely; they are now all cosines that are aligned at zero time. In general, filtering that affects phase can cause distortions to the signal that will make it virtually unrecognizable (phase affects how all the Fourier components align in the time domain, after all).

However, there is one important type of phase adjustment that does not distort the relative time alignment of the Fourier components, and is fact useful in many contexts. We will consider adjusting the phase of each frequency component of the FFT by an amount proportional to its frequency.

That is, if the original signal contains a frequency component of the form

$$\phi(t) = Ae^{2\pi ift} \tag{5.1}$$

then we will adjust this to

$$\hat{\phi}(t) = Ae^{2\pi ift+icf} \tag{5.2}$$

where c is some constant of proportionality. This results in the signal is shifted by cf radians, or $cf/(2\pi)$ cycles. Since the time length of each cycle is $1/f$, $\hat{\phi}(t)$ is $\phi(t)$ shifted in time by $(cf/(2\pi))(1/f) = c/(2\pi)$. Notice that this time shift is independent of f. Thus if we apply a phase shift of cf at each frequency, then we'll get a consistent and circular time shift of $c/(2\pi)$. This is just an implementation of the time shift theorem for discrete periodic spectra.

A filter which shifts each phase in the FFT by an amount proportional to its frequency is called a **linear phase filter**. The time shift introduced by a linear phase filter can sometimes be a nuisance. However, there is a clever technique for correcting this effect; we can apply a linear phase filter to our signal, then time reverse the filtered signal and apply the same filter a second time, and finally time reverse the twice filtered signal. This has the effect of shifting the signal forward and backward in time by the same amount. It also effectively squares the frequency response of the filter. This technique is implemented in the MATLAB command **filtfilt**.

Returning to our original example, suppose that we multiply each component of the FFT by e^{i15f}. This effectively adds $15f$ to the phase angle of each component of the FFT. For example, at f=22000 Hz, the phase is shifted by

$\phi = 330000$ radians, which is 52,521 cycles, or 2.39 seconds. Similarly, at 100 Hz, the phase is shifted by $\phi = 1500$ radians, or 238.7 cycles, which is also 2.39 seconds. We then invert the FFT to recover the filtered signal.

Note that the direction of this phase shift is backward in time. That is, at time $t = 0$, we hear what was originally in the signal at $t = 2.39$ seconds. What do you expect to hear during the last 2.39 seconds of the playback? Remember that the FFT assumes that the entire signal is periodic.

Finally, we'll consider another common and often trivial or inconsequential type of phase shift. Suppose that the phase of each component of the FFT is adjusted by π. This is equivalent to multiplying each component of the FFT by $e^{i\pi}$, which is just -1. Because the FFT is a linear transformation of the original signal, we can easily compute the effect of this phase shift on the original signal. The inverse FFT of minus one times the FFT of the original signal is minus one times the inverse FFT of the FFT of the original signal, or just minus the original signal.

For many purposes, $\phi(t)$ and $-\phi(t)$ are indistinguishable signals. In our audio example, this phase phase shift makes no discernible difference, because your hearing system effectively analyzes the amplitudes of different frequency components and not their absolute phases.

Finite Impulse Response Filtering

By *finite impulse response* or *FIR* filters, we refer to linear filtering operators which have finite duration impulse responses. Such filters can be easily implemented by simply convolving the input signal with the impulse response. Since the impulse response is typically very short (perhaps just a few samples), this convolution can often be efficiently implemented directly without using the convolution theorem and the FFT.

Finite impulse response filters are invariably stable because they have no recursive components (i.e., no internal feedback in their algorithms). Once the input goes to 0, the output will thus return to zero within a finite period of time determined by the length of the impulse response. It's also trivial to make such a filter causal by simply specifying that the impulse response be zero for negative times.

In the following discussion, M will be the length of the filter sequence, N will be the length of the input sequence, n will be used as a time index, and k will be used as a frequency index. A common and easy to understand example is the symmetric, M-point (M odd) *running mean*filter defined as

$$w_n = \frac{1}{M}\Pi_M = \begin{cases} 1/M & \text{for } |n| \le (M-1)/2 \\ 0 & \text{for } |n| > (M-1)/2 \end{cases} . \tag{5.3}$$

The M filter impulse response values $w_0, w_1, ..., w_{M-1}$, are often referred to in this context as *weights* . Convolution of an arbitrary sequence, y_n, with this particular w_n results in a sequence with frequency characteristics (according to

the convolution theorem)

$$Z_k = Y_k \cdot \mathrm{DFT}[w_n] = Y_k \cdot \frac{1}{M} \sum_{n=-(M-1)/2}^{(M-1)/2} e^{-\imath 2\pi kn/N} \tag{5.4}$$

Recall from our previous lecture notes on sampled time series that

$$\sum_{n=-M}^{M} e^{-\imath 2\pi fn} = \frac{\sin(N\pi f)}{\sin(\pi f)} \tag{5.5}$$

where $N = 2M + 1$. Thus

$$Z_k = Y_k \cdot \frac{1}{M} \frac{\sin(M\pi k/N)}{\sin(\pi k/N)} \ . \tag{5.6}$$

The net result is a low-pass filter with a Dirichlet kernel frequency response function. The DFT of w_n is real. Note that at some frequencies, W_k is positive while at other frequencies it is negative. As a result, the phase of this filter "flips" from 0 to 180 degrees and back whenever W_k changes sign.

Note that although this low pass filter has a zero phase contribution, it is also acausal and thus can only be implemented on a pre-recorded signal. This is easily gotten around with the implementation of a pre-event memory in the recording system.

As we have already seen, linear phase response is a frequently desirable property of a filter because it will not distort the relative timing of the Fourier components. All M-point, real-valued FIR filters with symmetric weights have this property, as we can see by expressing the frequency response as

$$W_k = \sum_{n=0}^{M-1} w_n e^{-\imath 2\pi kn/N} = e^{-\imath \pi k(M-1)/N} \sum_{n=-(M-1)/2}^{(M-1)/2} w_n e^{-\imath 2\pi kn/N} \tag{5.7}$$

$$= e^{-\imath \pi k(M-1)/N} \left(2 \sum_{n=1}^{(M-1)/2} w_n \cos(2\pi kn/N) + w_0 \right) = P(k) \cdot A(k) \ . \tag{5.8}$$

The phase factor $P(k)$ is complex with magnitude one, so it only adjusts the phase. Furthermore, the phase adjustment is a linear function of k. Meanwhile, the amplitude factor $A(k)$ is real, so it only changes the relative amplitude at different frequencies.

The MATLAB command **conv** can be used to convolve a filter sequence w with the input sequence x. One problem with this is that the convolution will lengthen the sequence by $M - 1$ samples. This is because the response of the filter continues after the end of the input signal. If these samples are unwanted or zero, you can simply truncate the filtered signal to produce an output with the same number of samples as the input

```
>> y=conv(x,w);
>> y=y(1:N);
```

An alternative is to use the MATLAB command **filter**. This command is designed for more complicated IIR filters (discussed below) which are specified by two vectors. However, it can be used with an FIR filter by specifying the filter weights as the first argument, and "[1]" as the second argument. e.g.

```
>> y=filter(w,[1],x);
```

Now suppose we have some desired continuous (*analog*) filter characteristic, $\Omega(f)$, and we wish to construct an FIR realization, specified by N weights, w_n. As our realization is discrete, we let $\Omega(f)$ be periodic in f, and apply the inverse Fourier transform on the Nyquist interval to obtain

$$w_n = \int_{-1/2}^{1/2} \Omega(f) e^{\imath 2\pi f n} \, df \tag{5.9}$$

where f is normalized to the Sampling rate, r. However, there is a complication in applying this recipe, as the resulting sequence may have an infinite number of nonzero w_n. Consider, for example, the perfect low pass filter, with a desired cutoff frequency of f_s/α, defined by

$$\Omega(f) = \Pi(\alpha f/2) \ . \tag{5.10}$$

The inverse Fourier transform gives

$$w_n = 2\int_0^{1/\alpha} \cos(2\pi f n) \, df = \frac{2}{\alpha} \text{sinc} \ (2n/\alpha) \tag{5.11}$$

which has an infinite number of nonzero w_n.

The sinc function decays as n^{-1}. What happens if we simply truncate the series to M terms, bounded by $\pm(M-1)/2$? In this case we are convolving the ideal frequency response with the DFT of the boxcar function, which is the by now familiar Dirichlet kernel

$$\sum_{n=-(M-1)/2}^{(M-1)/2} e^{-\imath 2\pi k n/N} = \frac{\sin(M\pi k/N)}{\sin(\pi k/N)} \equiv D(M,N,k) \ . \tag{5.12}$$

The frequency response of our truncated realization is thus the convolution of the desired response with the Fourier transform of the discrete boxcar function weighting. This particular realization is thus not especially desirable because the Dirichlet kernel is a very oscillatory function which doesn't fall off particularly rapidly with frequency. The result is the introduction of large side lobes to the frequency response of this filter realization. We can reduce this problem by applying less abrupt truncation and/or by taking N to be as large as possible. This brings us back once again to the issue of *windowing*, which arose previously

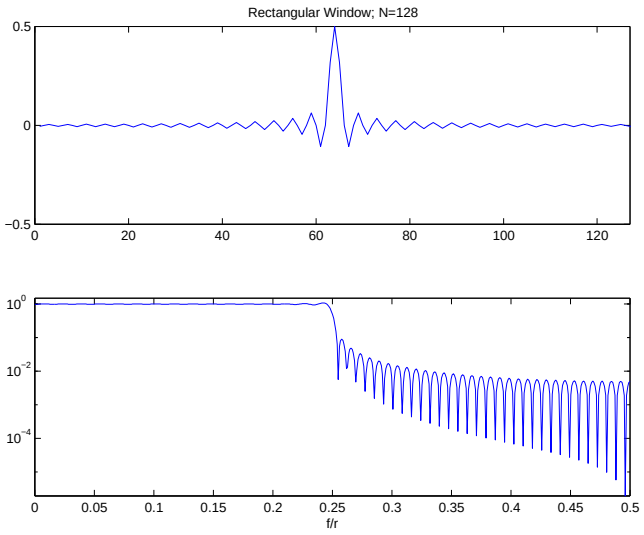

Figure 5.4: FIR weights and response for a 128-point rectangular window FIR realization of a low pass filter with a desired cutoff frequency of $f = r/4$.

in these notes in different contexts associated with estimating power spectral densities and in sampling.

Although usually not an optimal way to design filters, windowing the infinite sequence defined by (5.9) provides a simple way of obtaining useful closed forms for FIR filter weights. Some examples of windowed realizations of ideal low-pass filters for $\alpha = 4$ (filter corner at 1/2 of the Nyquist frequency) are shown in Figures 5.4, 5.5, 5.6.

Because of the unique correspondence between an N-length sequence and its N DFT coefficients, an N-length FIR filter can be uniquely specified by N DFT coefficients. Another design method for obtaining FIR filter weights, called *Frequency Sampling*, is thus to specify frequency characteristics at up to N desired frequencies and then take the IDFT, rather than the inverse continuous FFT, as we did in (5.9). This gets around the problem of truncating an infinite number of weights, because the IDFT produces exactly N weights. For example, the perfect low pass filter realization, where the passband is defined from $k = -(M-1)/2$ to $k = (M-1)/2$ becomes

$$w_n = \frac{1}{N} \sum_{k=-(M-1)/2}^{(M-1)/2} e^{\imath 2\pi nk/N} = \frac{1}{N} \frac{\sin(\pi nM/N)}{\sin(\pi n/N)} = \frac{1}{N} D(M, N, k) \ . \qquad (5.13)$$

Convolution of an input series of length N with (5.13) is identical to simply taking the DFT of the input series, setting the frequency components for $|k| > M$ equal to zero, and then inverse transforming the modified k-series back to

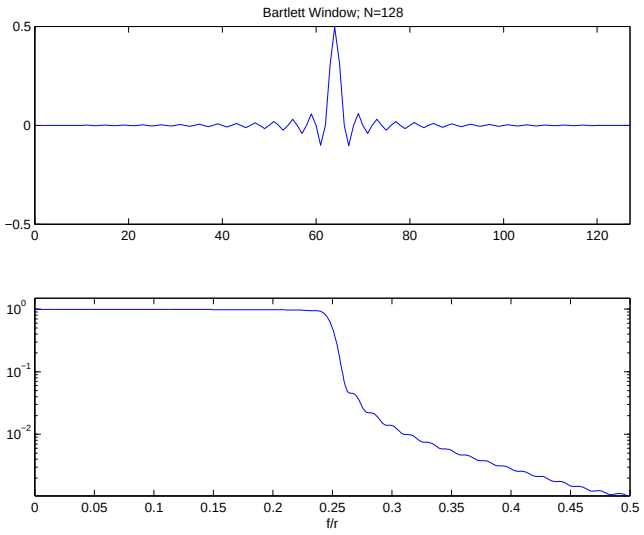

Figure 5.5: FIR weights and response for a 128-point Bartlett window FIR realization of a low pass filter with a desired cutoff frequency of $f = r/4$. Note the reduction in ripple near the transition band relative to the simple truncation series (Figure 5.4).

the n domain via the IDFT.

In practice, we determine the desired filter length M, then pick N so that a filter of length M covers all of the frequencies for which we want a nonzero response. Once the filter sequence is computed, we can apply the filter to a sequence of arbitrary length by convolving the filter sequence with the input sequence.

The problem with this type of filtering is that we have only defined the frequency response at N points. what happens to the frequency response at frequencies that are not constrained?

The frequency response of the sequence w_n is given by (5.13). Taking a unit sampling interval (so that f is normalized to the Nyquist frequency) gives (when the Hermitian terms are collapsed into a cosine function)

$$W(f) = \frac{2}{N} \sum_{n=1}^{N/2-1} \left(\frac{\sin(\pi n M/N)}{\sin(\pi n/N)} \cos(2\pi n f) \right) + \frac{M}{N} + \frac{1}{N} \cos(2\pi n f) \qquad (5.14)$$

where the last two terms are for $n = 0$ and $n = N/2$, respectively. (5.14) is plotted as a function of normalized frequency in Figures 5.7 and 5.8 for $N = 128$, $M = 31$ and for $N = 512$, $M = 127$.

We see that the frequency response oscillates wildly between the frequency sample points, even though it dutifully follows the ideal low pass specification

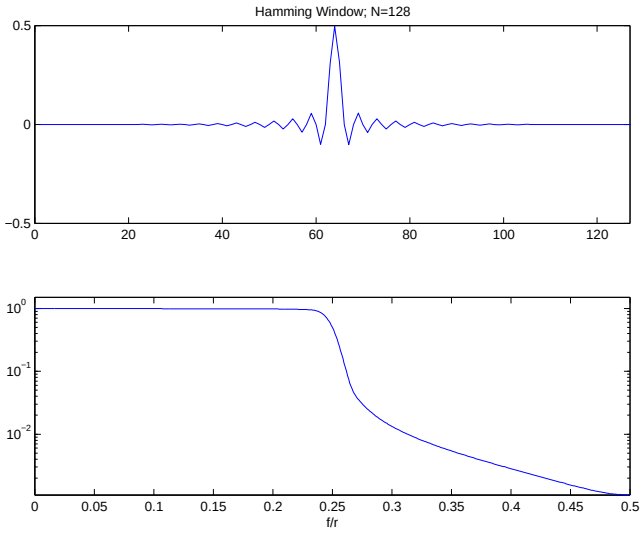

Figure 5.6: FIR weights and response for a 128-point Hamming window FIR realization of a low pass filter with a desired cutoff frequency of $r/4$. Note the reduction in ripple near the transition band relative to the simple truncation series (Figure 5.4) and the Bartlett window (Figure 5.5). The tradeoff for smoother response and better rejection outside of the desired passband is to have a more gradual transition.

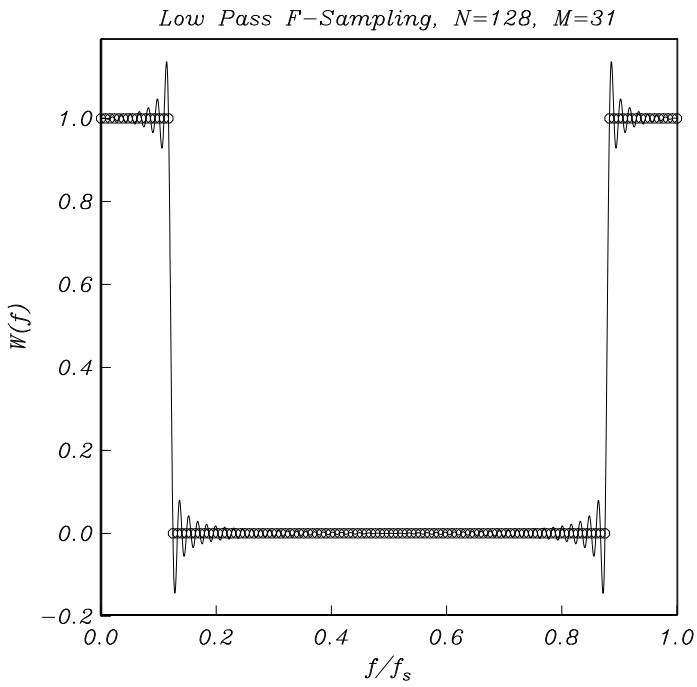

Figure 5.7: Frequency sampling frequency response in attempting to realize an ideal low pass filter; N=128

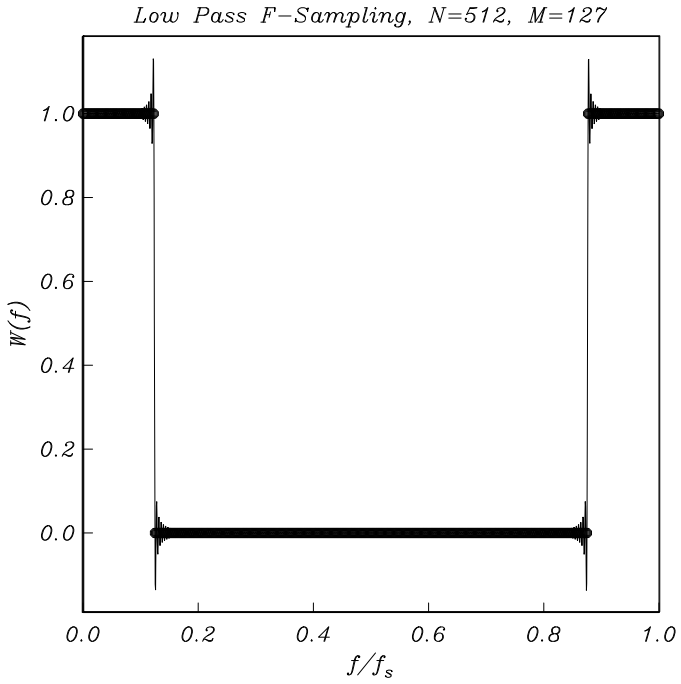

Figure 5.8: Another frequency sampling frequency response in attempting to realize an ideal low pass filter; N=512. Note that increasing the number of frequency specifications does not reduce the amplitude of the undesirable response ripple.

exactly at the proscribed frequencies. The largest overshoots occur near the transition band. This type of behavior at the intermediate frequencies is called the *Gibbs phenomenon* and Figures 5.7 and 5.8 show that it has the unfortunate property that the percent overshoot does not decrease as N increases, although the width of the ripples does decrease as we squeeze them by stubbornly specifying more and more frequencies in our frequency sampling procedure.

If frequency sampling is really equivalent to direct manipulation of the FFT, then why didn't we notice any problems when we directly manipulated the FFT of the 20 second audio clip? In that case, the FFT had 882,000 frequencies, so the equivalent FIR filter would consist of a sequence of over 80,000 weights. Thus the ripple was confined to extremely narrow frequency bands near the cutoff at 2 Khz.

It turns out that one can in fact design much better behaved (smaller ripple) filters by using more sophisticated design methods. Although we won't get into this in these notes, one very popular approach is the use of the Remez exchange algorithm to design FIR filters with specified maximum and minimum amplitudes in each of several frequency bands. Typically, a quite small filter (say 15 points) can adequately match the desired frequency response with very little ripple. The MATLAB command **firpm** implements this approach to designing a FIR filter.

More compact filter representations are possible if we allow *recursive*elements in our filters, where a component of the output is fed back into the input. In addition, there are systems of interest that have impulse responses with non-zero values at $t = \infty$ (e.g., integrators) which cannot be expressed at all in a finite length FIR series. To fully appreciate this and to get a more general outlook on discrete realizations of continuous idealizations, we need to introduce some new types of transforms that are related to the Fourier transform.

The Laplace Transform

The *One-Sided Laplace transform* is a generalized Fourier transform which explicitly allows for complex frequency, $s = \sigma + \imath\omega$, where σ and ω are real

$$\Phi(s) \equiv L[\phi(t)] = \int_0^\infty \phi(t) e^{-st}\, dt\ . \tag{5.15}$$

The convergence of the integral is very much an issue. Assuming that s is a positive real number or is complex with a positive real part, the function e^{-st} will go to 0 as t goes to infinity. For the integral to converge, $\phi(t)$ must not grow too quickly as t goes to infinity. If $|\phi(t)| \leq Ke^{bt}$, for some real constants K and b, and $\mathrm{Re}(s) > b$, then the integral will converge.

Note that an alternative *Two-Sided Laplace Transform* is used by some authors. In the two–sided Laplace transform, the integral is evaluated from minus infinity to plus infinity instead of from 0 to plus infinity. The two sided Laplace transform of $H(t)\phi(t)$ is precisely the one sided transform of $\phi(t)$.

If we make the substitution $s = 2\pi\imath f = \imath\omega$, we get

$$L[\phi(t)] = \int_0^\infty \phi(t)e^{-st}\,dt = \int_{-\infty}^\infty H(t)\phi(t)e^{-2\pi\imath ft}\,dt = F[H(t)\phi(t)]\ . \quad (5.16)$$

The the Laplace transform of $\phi(t)$ is equivalent to the Fourier transform of $H(t)\phi(t)$. An alternative way to look at this is to say that as long as our signals are zero before time $t = 0$, the Fourier transform and Laplace transform are equivalent. This equivalence will be used frequently. In practice, we will often assume that signals begin after time $t = 0$, so that multiplying by $H(t)$ isn't necessary. Because of this relationship between the Laplace transform and the Fourier transform, many properties of the Laplace transform can be proved by using the already known properties of the Fourier transform.

For example, consider the action of a linear time invariant system on a signal $x(t)$, which we'll assume is zero for all t before $t = 0$. Let $\phi(t)$ be the impulse response of the system, and let $\Phi(f)$ be the Fourier transform of the impulse response. Assume further that the system is causal so that the output, $y(t)$, is zero before time $t = 0$. We know from our work with the Fourier transform that the Fourier transform of the output is $Y(f) = X(f)\Phi(f)$. Using our substitution $s = 2\pi\imath f$, we get that $Y(s) = X(s)\Phi(s)$. Here we've abused notation slightly by using $Y(s)$ for the Laplace transform of $y(t)$ and $Y(f)$ for the Fourier transform of $y(t)$. As long as all of the functions involved are zero before time 0, this works beautifully.

Recall that the Fourier transform of the derivative of $f(t)$ is given by $F[f'(t)] = 2\pi\imath f F[f(t)]$. Using the equivalence of the Fourier and Laplace transforms for functions which are zero before time $t = 0$, we would get that $L[f'(t)] = sL[f(t)]$. This is almost, but not quite correct. The problem occurs because $f(0)$ might be nonzero. Using the definition of the Laplace transform and integration by parts, it's easy to show that $L[f'(t)] = sL[f(t)] - f(0)$. In general,

$$L[f^{(n)}(t)] = s^n L[f(t)] - s^{n-1}f(0) - \ldots - sf^{(n-2)}(0) - f^{n-1}(0)\ . \quad (5.17)$$

Next, we consider a linear time invariant system that is governed by a nth order linear differential equation with constant coefficients.

$$a_n\frac{d^n y}{dt^n} + \ldots + a_1\frac{dy}{dt} + a_0 y = b_m\frac{d^m x}{dt^m} + \ldots + b_1\frac{dx}{dt} + b_0 x\ . \quad (5.18)$$

Many (but by no means all) LTI's can be written in this form. If we assume that $y(0)$, $y'(0)$, ..., $y^{(n-1)}(0) = 0$, then by the rule for the Laplace transform of a derivative,

$$(a_n s^n + \ldots a_1 s + a_0)\,Y(s) = (b_m s^m + \ldots b_1 s + b_0)\,X(s). \quad (5.19)$$

This can be rewritten as

$$\frac{Y(s)}{X(s)} = \Phi(s) = \frac{\sum_{j=0}^m b_j s^j}{\sum_{k=0}^n a_k s^k}\ . \quad (5.20)$$

As in the Fourier transfer function definition, the m roots of the numerator of (5.20) are called zeros, because $\Phi(s)$ is zero there, and the n roots of the denominator are called poles, because $\Phi(s)$ is infinite there. If the coefficients, a_i and b_i in (5.18) are real, then the poles and zeros are either real or form complex conjugate pairs. Note that at a pole frequency, s_p, an output can occur even for zero input. As we have seen before, a stable system has all of its poles on the left hand side of the complex plane (i.e., $\mathrm{Re}(s_p) < 0$), so that the pole frequencies have negative real parts

Another qualitative point is that closely-spaced poles and zeros cancel and can be ignored unless we are very close to them. Indeed for large frequencies all poles and zeros will start to cancel in this manner, so that $\Phi(s)$ asymptotically approaches

$$G(s) = \frac{b_m}{a_n}(s)^{m-n} \tag{5.21}$$

which changes by some multiple of about 6 dB for every doubling in frequency (6 dB per octave).

The Inverse Laplace Transform

For $t \geq 0$, the inverse Laplace transform is given by

$$\phi(t) = \frac{1}{2\pi\imath}\int_{\gamma-\imath\infty}^{\gamma+\imath\infty} \Phi(s)e^{st}\, ds \tag{5.22}$$

where γ is selected to be large enough so that the integral converges. if $|\phi(t)| \leq Ke^{bt}$ for some constants K and b, then any value of γ larger than b will suffice.

In practice, this integral is usually evaluated by the technique of *contour integration*, using a contour c which includes the line $\mathrm{Re}(s) = \gamma$ and a semicircular arc to the left. See figure 5.9. If the integral over the semicircular arc is 0 (because $\Phi(s)$ goes to zero fast enough as the radius increases), then we can replace the integral over the line with an integral around the entire contour

$$\phi(t) = \frac{1}{2\pi\imath}\int_{c} \Phi(s)e^{st}\, ds\ . \tag{5.23}$$

Why bother with the contour integral? An important theorem of complex analysis states that if $f(z)$ has a finite number of poles, then the counter clockwise integral around a closed contour, which contains the poles of $f(z)$ can be evaluated by

$$\int_c f(z)dz = 2\pi\imath \sum_{\alpha=\text{poles of f(z)}} \text{residue}(\alpha) \tag{5.24}$$

where the residue at a pole $z = \alpha$ of order m is

$$\text{residue}(\alpha) = \frac{1}{(m-1)!}\lim_{z\to\alpha}\frac{d^{m-1}}{dz^{m-1}}(z-\alpha)^m f(z)\ . \tag{5.25}$$

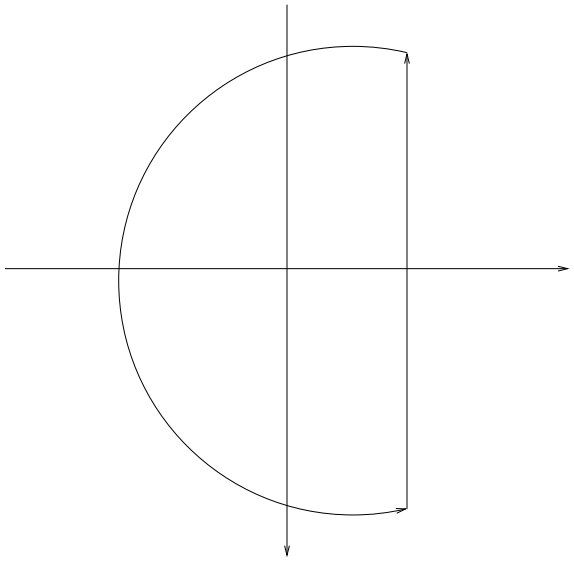

Figure 5.9: The contour used in computing the inverse Laplace transform.

Notice that the value of the contour integral depends only on the residues and the locations of the poles. Any contour which surrounds the same collection of poles will result in the same value for the integral! We can apply this formula to (5.23) to evaluate the inverse Laplace transform of $\Phi(s)$.

For example, suppose that our linear time invariant system is governed by the differential equation

$$2y'(t) + y(t) = x(t) \ . \tag{5.26}$$

We find that

$$\Phi(s) = \frac{1}{1+2s} \ . \tag{5.27}$$

In this case the system has a single pole of order 1 at s=-1/2. We will now find the impulse response by computing the inverse Laplace transform using (5.23) and contour integration. We can use the integration path from $-\imath\infty$ to $+\imath\infty$.

$$\phi(t) = \frac{1}{2\pi\imath} \int_{-\imath\infty}^{\imath\infty} \frac{1}{1+2s} e^{st} \ ds \ . \tag{5.28}$$

Our integrand goes to 0 very rapidly as our semicircular arc expands, so that in the limit, the integral over the semicircular arc is in fact 0. To show this, we use the substitution $s = Re^{\imath\theta}$, and take the limit as R goes to infinity

$$\int_{\pi/2}^{3\pi/2} \frac{1}{1+2Re^{\imath\theta}} e^{Re^{\imath\theta}t} Re^{i\theta} \ d\theta \ . \tag{5.29}$$

In the limit as R goes to infinity, this integrand goes to 0 and the integral goes to 0, so it is safe in this case to replace (5.22) with (5.23). A very common mistake is to make the switch to the contour without checking that the integral over the semicircular arc is 0. In such cases, contour integration will give the wrong answer, so beware!

The residue at s=-1/2 is

$$\text{residue}(-1/2) = \lim_{z \to -1/2} (s + 1/2) \frac{1}{1 + 2s} e^{st} = \frac{1}{2} e^{(-1/2)t} \tag{5.30}$$

The factors of $2\pi\imath$ in (5.23) and (5.24) cancel out, so

$$\phi(t) = \frac{1}{2} e^{(-1/2)t} \quad t \geq 0 \ . \tag{5.31}$$

Although any inverse Laplace transform can in theory be computed by this method, in practice it's usually easier to refer to a table of Laplace transforms or to use a symbolic computation package such as Maple to do the work. Table 5.1 gives some useful Laplace transforms.

A very common problem in practice is to find the inverse Laplace transform of a rational function $\Phi(s) = p(s)/q(s)$, where $p(s)$ and $q(s)$ are polynomial functions of s. We can perform a partial fraction decomposition of $\Phi(s)$ in terms of its poles a_1, a_2, ..., a_m.

$$\Phi(s) = \frac{p(s)}{q(s)} = \sum_{j=1}^{m} \sum_{k=1}^{n_j} \frac{c_{j,k}}{(s - a_j)^k}. \tag{5.32}$$

Here n_j is the multiplicity of the pole a_j. Note that since we're working with complex poles, there are no irreducible quadratic factors. This partial fraction decomposition can be done by hand, or it can be done with the help of a symbolic computation package such as Maple, Mathematica, or with MATLAB's symbolic computation toolbox.

From the table of inverse Laplace transforms, we can see that

$$L^{-1}\left[\frac{1}{(s-a)^n}\right] = \frac{t^{(n-1)}}{(n-1)!} e^{at}. \tag{5.33}$$

Thus

$$\phi(t) = L^{-1}\left[\Phi(s)\right] = \sum_{j=1}^{m} \sum_{k=1}^{n_j} \frac{c_{j,k}}{(k-1)!} t^{k-1} e^{a_j t}. \tag{5.34}$$

This expression of $\phi(t)$ in terms of the poles of $\Phi(s)$ is very useful, because it provides us with a stability criterion. If $\text{Re}(a_j) < 0$, then $e^{a_j t}$ will go to 0 as t goes to infinity. However, if $\text{Re}(a_j) \geq 0$, then $e^{a_j t}$ will not decay as t goes to infinity. Thus our filter will be stable if and only if $\text{Re}(a_j) < 0$ for $j = 1$, ..., m. That is, the filter will be stable if all of the poles are in the left half plane.

Table 5.1: Table of Laplace Transforms

$f(t)$	$F(s)$	where valid
1	$\frac{1}{s}$	$s > 0$
e^{at}	$\frac{1}{s-a}$	$s > a$
$\frac{t^{(n-1)}}{(n-1)!}e^{at}$	$\frac{1}{(s-a)^n}$	$s > a$
t^n	$\frac{n!}{s^{n+1}}$	$s > 0$
$\sin(at)$	$\frac{a}{s^2+a^2}$	$s > 0$
$\cos(at)$	$\frac{s}{s^2+a^2}$	$s > 0$
$e^{at}\sin(bt)$	$\frac{b}{(s-a)^2+b^2}$	$s > a$
$e^{at}\cos(bt)$	$\frac{s-a}{(s-a)^2+b^2}$	$s > a$
$H(t-c)$	$\frac{e^{-cs}}{s}$	$s > 0$
$\delta(t-c)$	e^{-cs}	
$f^{(n)}(t)$	$s^nF(s) - s^{n-1}f(0) - \ldots - f^{(n-1)}(0)$	
$e^{ct}f(t)$	$F(s-c)$	
$H(t-c)f(t-c)$	$e^{-cs}F(s)$	

The Chandler Wobble

As an example of a geophysical system transfer function with one complex pole in the s plane and a complex forcing and response, we next consider the *Chandler wobble* or *free nutation* response of the earth's spin axis, which changes due to some combination of mass shifts in the Earth due to oceanic or atmospheric circulation, glaciation, vegetation variations, snow or surface water accumulation, large earthquakes, mantle motions, core-mantle interactions, etc. Lately, it has been claimed that the most important processes, at least from 1985-1996, were atmospheric and oceanic processes, with the dominant mechanism being ocean-bottom pressure variations [7]. In a Cartesian grid is laid out at the north pole with an origin at the mean pole position (the axis of greatest moment of inertia), the spin axis at a given time can be specified as being at at (y_1, y_2) (Figure 5.10.)

If the forcing function, in this case, the migration of the Earth's principal axis of maximum rotational inertia due to mass movements, in the same coordinate system, is (x_1, x_2), the governing differential equations of motion are those of a body rotating slightly off from its maximum moment of inertia principal axis

$$\frac{\dot{y}_1}{\omega_c} + y_2 = x_2 \tag{5.35}$$

$$\frac{-\dot{y}_2}{\omega_c} + y_1 = x_1 \tag{5.36}$$

where, for a rigid body,

$$\omega_c = \left(\frac{C-A}{C}\right)\Omega \tag{5.37}$$

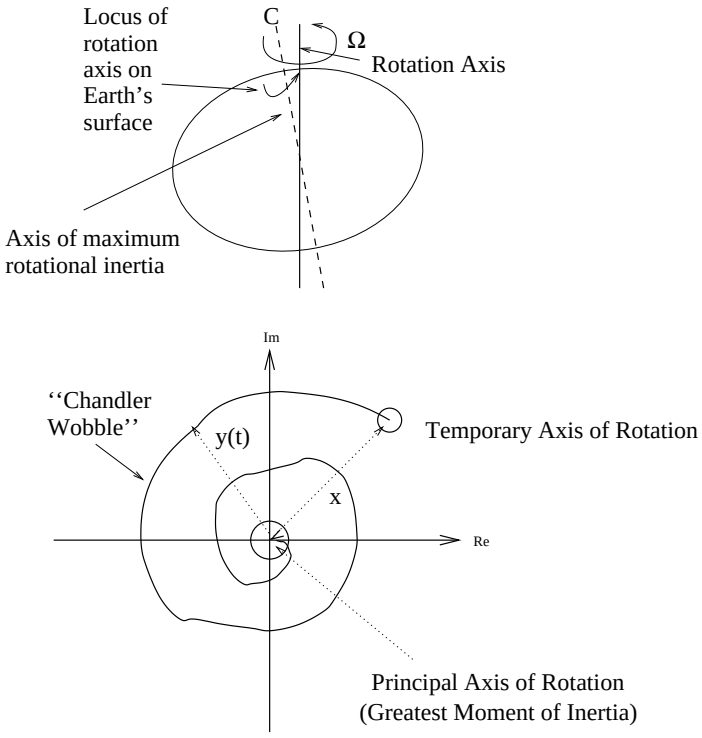

Figure 5.10: Geometry of the Chandler Wobble

.

where C and A are the polar and equatorial rotational moments of inertia and Ω is the spin rate. In the Earth, the components of the Chandler wobble have amplitudes of tens of meters, and are thus readily detectable using astronomical or other techniques. The ideal rigid body frequency (5.37) for a solid Earth is about 305 days, $(C - A)/C \approx 1/305.51$) but the observed decay constant is significantly longer (about 430 days) due to the Earth not being a perfectly elastic body.

We can jointly consider the two equations (5.35 and 5.36) by defining the complex quantities

$$x = x_1 + \imath x_2 \tag{5.38}$$

$$y = y_1 + \imath y_2 \tag{5.39}$$

to obtain

$$i\frac{\dot{y}}{\omega_c} + y = x \ . \tag{5.40}$$

Taking the Laplace transform of both sides gives

$$Y(s)\left(\frac{is}{\omega_c} + 1\right) = X(s) \tag{5.41}$$

so that the transfer function is

$$\frac{Y(s)}{X(s)} = \frac{\omega_c}{is + \omega_c} \tag{5.42}$$

which has a single pole at $s = \imath\omega_c$ (as $y(t)$ and $x(t)$ are complex valued, there is no complex conjugate pole at $s = -\imath\omega_c$ in this case). Physically, this means that the locus of the rotational axis on the earth's surface will indefinitely precess from west to east, once the system is excited. This asymmetry arises from the gyroscopic nature of the system. Dissipation in the earth (the principal cause or causes for the damping of the Chandler wobble are, again, controversial) can be accommodated by making ω_c complex

$$\omega_c = \frac{2\pi}{T_c}\left(1 + \frac{i}{2Q_c}\right) = \frac{\pi}{T_c}\left(2 + \frac{i}{Q_c}\right) \tag{5.43}$$

where Q_c is the quality factor (see Chapter 2) of the system and T_c is the natural frequency. The pole of the system response (5.42) then becomes

$$p = \frac{\pi}{T_c}\left(2\imath - \frac{1}{Q_c}\right) \tag{5.44}$$

which has a negative real part and hence describes a decaying sinusoidal motion. The impulse response is thus

$$\phi(t) = L^{-1}[Y(S)/X(s)] = \frac{1}{2\pi\imath}\int_c \frac{-i\omega_c}{s - \imath\omega_c} e^{st}\, ds \tag{5.45}$$

and may be found via contour integration and the residue theorem to be the complex sinusoid

$$= -i\omega_c e^{\imath\omega_c t} \tag{5.46}$$

where the phase of (5.46) signifies the phase relationship between the complex forcing and response functions, $x(t)$ and $y(t)$. In the problem of the Chandler wobble, the interesting physics are tied up in the measurement of T_c (which is around 430 days) and of the forcing function, $x(t)$. The wobble is continuously excited by mass movements in the solid Earth, oceans, and the atmosphere which change its moments of inertia and averages about 0.14 seconds of arc (6.8×10^{-7} rad), which corresponds to a root mean square (rms) polar discrepancy of about 4.5 m).

It is worth noting that in some interesting situations, such as the excitation of the normal modes of the earth, we can examine the response and estimate the pole positions without worrying about the exact spectrum of the excitation function. This is because the excitation function is broad-band relative to our observational bandwidth and thus, on average, excites many frequencies.

The Z Transform

Just as the discrete Fourier transform as an alternative to the Fourier transform to analyze discretized signals, the *Z transform* is the discrete analog of the continuous Laplace transform.

Consider a complex variable z and define the *z transform* of a sequence x_n as

$$X(z) = Z[x_n] = \sum_{n=-\infty}^{\infty} x_n z^{-n}. \tag{5.47}$$

A warning: a few authors use z^n rather than z^{-n} in their z transform definitions. Again, this is mere convention, akin to choosing $e^{-\imath 2\pi ft}$ or $e^{\imath 2\pi ft}$ in the Fourier transform definitions, but can lead to misinterpretations. Also, some authors will define a one-sided version of the z transform in which the sum runs from $n = 0$ to infinity. As a rule, always check to see what conventions a given author is using!

Multiplication by z^{-l} in the z domain is equivalent to a time delay (rightward shift) of l samples and multiplication by z^l is equivalent to a time advance (leftward shift) of l samples; the exponent of z in each term is a place holder to designate where a particular value fits into the time series. The time shift theorem for the z transform is thus

$$Z[x_{n-i}] = \sum_{n=-\infty}^{\infty} x_{n-i} z^{-n} = z^{-i} \sum_{n=-\infty}^{\infty} x_{n-i} z^{-(n-i)} = z^{-i} \sum_{m=-\infty}^{\infty} x_m z^{-m} \tag{5.48}$$

or

$$Z[x_{n-i}] = z^{-i} X(z)\ . \tag{5.49}$$

Finding closed-form expressions for the z transforms of common time sequences relies on the specific properties of each series, but as an example, consider an exponential series

$$x_n = \begin{cases} c^n & n \geq 0 \\ 0 & n < 0 \end{cases} . \tag{5.50}$$

In this case, we can use the standard procedure for collapsing geometric series to obtain

$$Z[x_n] = \sum_{n=0}^{\infty} c^n z^{-n} = \frac{1}{1 - cz^{-1}} = \frac{z}{z - c} \tag{5.51}$$

when $|z| > |c|$.

The case when $c = 1$ gives the z transform of the discrete unit step function

$$Z[H_n] = \frac{z}{z - 1} . \tag{5.52}$$

The convolution theorem relationship for the z transform is particularly easy to see. For a particular m, the terms in the product

$$W(z) = X(z)Y(z) = \sum_{m=-\infty}^{\infty} w_m z^{-m} \tag{5.53}$$

can be seen from polynomial multiplication of $X(z)$ and $Y(z)$ to be

$$x_n z^{-n} y_{m-n} z^{-(m-n)} . \tag{5.54}$$

It follows that

$$W_m = \sum_{n=-\infty}^{\infty} x_n y_{m-n} = \sum_{n=-\infty}^{\infty} y_n x_{m-n} \tag{5.55}$$

which is just the discrete (linear) convolution of x_n and y_n.

To evaluate the inverse z transform, we again make use of the technique of contour integration. By the residue theorem, the counterclockwise contour integration around a pole of degree $-k + 1$ is

$$\frac{1}{2\pi\imath} \int_c \frac{dz}{z^{-k+1}} = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases} . \tag{5.56}$$

The inverse z transform is thus

$$x_n = \frac{1}{2\pi\imath} \int_c X(z) z^{n-1} \, dz \tag{5.57}$$

where c is a counterclockwise closed contour selected so that the integral will converge.

To discuss issues of convergence, we express the z transform as a function of complex z in a polar coordinate system $z = Re^{\imath 2\pi f}$, where R is a real number. The z transform is then

$$X(Re^{\imath 2\pi f}) = \sum_{n=-\infty}^{\infty} x_n \cdot (Re^{\imath 2\pi f})^{-n} = \sum_{n=-\infty}^{\infty} x_n R^{-n} e^{-\imath 2\pi f n} \tag{5.58}$$

for $R = 1$, z lies on the unit circle in the complex plane, and the z transform is equivalent to the Fourier series of the sequence x_n. The infinite series defined by the z transform (5.47) converges when

$$\sum_{n=-\infty}^{\infty} |x_n z^{-n}| = \sum_{n=-\infty}^{\infty} |x_n R^{-n}| < \infty \ . \tag{5.59}$$

As the z transform contains terms for both positive and negative n, the general situation is that the sequence converges in some annular region, where R is not so large that the negative n part of the sequence diverges, but not so small that the positive n part of the sequence diverges, i.e.,

$$R_{h-} < |z| < R_{h+} \tag{5.60}$$

where R_{h-} and R_{h+} designate the inner and outer radii of the annulus, respectively.

Given the inverse z transform (5.57) we can now examine what happens in the z domain when we multiply two time series together

$$w_n = x_n \cdot y_n \ . \tag{5.61}$$

Taking the z transform of both sides yields

$$W(z) = \sum_{n=-\infty}^{\infty} x_n y_n z^{-n} = \frac{1}{2\pi\imath} \sum_{n=-\infty}^{\infty} x_n \int_c Y(v) \left(\frac{v}{z}\right)^n \frac{dv}{v} \tag{5.62}$$

$$= \frac{1}{2\pi\imath} \int_c Y(v) \left\{ \sum_{n=-\infty}^{\infty} x_n \left(\frac{v}{z}\right)^n \right\} \frac{dv}{v} = \frac{1}{2\pi\imath} \int_c Y(v) X(z/v) \frac{dv}{v} \tag{5.63}$$

setting $z = Re^{\imath\phi}$ and letting the contour, c be the circular path $v = \rho e^{\imath\theta}$ gives

$$W(Re^{\imath\phi}) = \frac{1}{2\pi} \int_0^{2\pi} Y(\rho e^{\imath\theta}) X(\frac{R}{\rho} e^{\imath(\phi-\theta)}) \ d\theta \tag{5.64}$$

This is a generalized case of the convolution relationship for the Fourier transform. To see this, evaluate (5.64) on the unit circle, where $R = \rho = 1$, $\phi = 2\pi f$, and $\theta = 2\pi f'$ to obtain

$$W(f) = \int_0^1 Y(e^{2\pi\imath f'}) X(e^{2\pi\imath(f-f')}) \ df' \tag{5.65}$$

which is a circular convolution! In fact, the DFT of a sequence,

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\imath 2\pi nk/N} \tag{5.66}$$

is just the z transform evaluated at N equiangular points around the unit circle, i.e.,

$$X_k = X(z = e^{\imath 2\pi k/N}) k = 0, 1, 2, ..., N-1 \ . \tag{5.67}$$

How can we relate the discrete-time z transform to the continuous-time Laplace transform? How we do this is fundamental to designing discrete systems which mimic continuous ones.

Consider the Laplace transform of a sampled version of a continuous function, $x(t)$, which is assumed to be 0 before $t = 0$:

$$\int_0^\infty x(t)\text{III}(t)e^{-st}\ dt = \int_0^\infty \sum_{n=0}^{\infty} x(n)\delta(t-n)e^{-st}\ dt \tag{5.68}$$

$$= \sum_{n=0}^{\infty} x(n) \int_0^\infty \delta(t-n)e^{-st}\ dt = \sum_{n=0}^{\infty} x(n)e^{-sn} = \sum_{n=0}^{\infty} x_n z^{-n} \ . \tag{5.69}$$

Notice that if we let $z = e^s$, then $z^{-n} = e^{-sn}$. Thus the mapping between z and s is simply $z = e^s$!

The general relationship between the z transform and the Fourier domain is shown in Figure 5.11. The imaginary axis in the s-plane corresponds to the unit circle in the z-plane. Similarly, the right half s-plane maps outside of the z-plane unit circle and the left half of the s-plane maps inside of the z-plane unit circle. Note that this mapping is multivalued, with a periodicity of 2π in the s-plane imaginary dimension, i.e., all of the points $s = R(2\pi\imath f + 2\pi\imath m)$ map to the same point on the unit circle in the z plane, $z = e^{\imath R 2\pi f}$.

Recall the stability criterion, that a filter is stable if and only if the poles of $\Phi(s)$ lie in the left half plane. In terms of z, the condition is that a filter is stable if and only if the poles of $\Phi(z)$ lie inside the unit circle.

IIR filtering

We will now consider recursive filters which effectively take weighted averages of the input and previous output samples. The filter equation is of the form

$$\sum_{k=0}^{K} a_k y_{n-k} = \sum_{m=0}^{M} b_m x_{n-m} \tag{5.70}$$

where y is the output sequence and x is the input sequence. This can be rewritten to show how y_n can be computed

$$y_n = \frac{\sum_{m=0}^{M} b_m x_{n-m} - \sum_{k=1}^{K} a_k y_{n-k}}{a_0} \ . \tag{5.71}$$

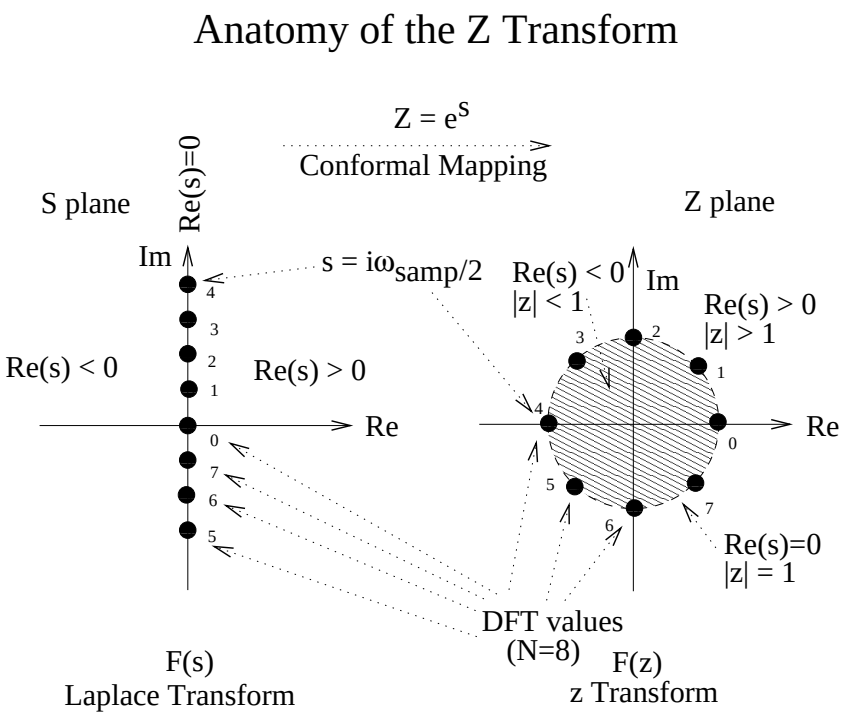

Figure 5.11: The Z Transform, and its relationship to the Fourier domain.

As equation (5.71) shows, recursive filters of this type are always causal. In this form, the filter is trivial to program. Given the filter coefficients b, a, and the input sequence x, the MATLAB command **filter** can be used to compute y.

We can compute the z transform of the impulse response by multiplying equation (5.70) by z^{-n}, and summing up terms from $n = -\infty$ to ∞.

$$\sum_{n=-\infty}^{\infty} \sum_{k=0}^{K} a_k y_{n-k} z^{-n} = \sum_{n=-\infty}^{\infty} \sum_{m=0}^{M} b_m x_{n-m} z^{-n} \tag{5.72}$$

$$Y(z) \sum_{k=0}^{K} a_k z^{-k} = X(z) \sum_{m=0}^{M} b_m z^{-m}. \tag{5.73}$$

Thus

$$\Phi(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{m=0}^{M} b_m z^{-m}}{\sum_{k=0}^{K} a_k z^{-k}} \ . \tag{5.74}$$

Note the similarities between (5.70), (5.74) and (5.18), (5.20), this is because delays map into powers of z^{-1} in the z transform, just as differentiation maps into powers of s in the Laplace transform.

When is our recursive filter stable? For our filter to be stable, we must have that the impulse response sequence ϕ_n goes to 0 as n goes to infinity. Recall that if all of the poles of $\Phi(s)$ are in the left half plane (or have negative real parts), then an LTI is stable. If all of the poles of $\Phi(z)$ are contained within the unit circle, then by our transformation $z = e^s$, all of the poles of $\Phi(s)$ will be in the left half plane, and our filter will be stable. Thus the stability condition for a recursive filter of the form (5.70) is that the poles of $\Phi(z)$ must lie within the unit circle.

The Impulse Invariance Method

Consider the simple continuous-time system defined by

$$\tau \frac{dy}{dt} + y = x \tag{5.75}$$

where τ is real. Solving for the transfer function using the Laplace transform yields

$$\frac{Y(s)}{X(s)} = \frac{1}{1 + \tau s} \tag{5.76}$$

which has a pole at $s = -1/\tau$ and is stable for $\tau > 0$. The frequency response is found by letting $s = 2\pi\imath f$

$$\frac{Y(f)}{X(f)} = \frac{1}{1 + \imath 2\pi\tau f} \tag{5.77}$$

which is 1 at zero frequency, and becomes smaller as f increases. This system is thus a low pass filter. The impulse response is

$$L^{-1}[Y(s)/X(s)] = \frac{1}{2\pi\imath}\int_c \frac{e^{st}\ ds}{1+\tau s} = H(t)\tau^{-1}e^{st}|_{s=\tau^{-1}} = \frac{H(t)}{\tau}e^{-t/\tau}\ . \quad (5.78)$$

and the step response is thus

$$H(t) * H(t)\tau^{-1}e^{-t/\tau} = H(t)(1-e^{-t/\tau})\ . \quad (5.79)$$

This response is nonzero for all non-negative $t < \infty$, and thus cannot be modeled at large t with any FIR filter, unless we are willing to use an arbitrarily large number of filter terms. However, a very simple recursive filter can come much closer to mimicking the desired response.

In the *impulse invariance method*, we pick a recursive filter so that the impulse response of the digital filter matches the desired impulse response of the continuous filter.

Consider the step response of the discrete system defined by

$$y_n - \alpha y_{n-1} = x_n(1-\alpha) \quad (5.80)$$

For a step sequence input, we get

$$\begin{array}{rcl} y_0 &=& 1-\alpha \\ y_1 &=& \alpha(1-\alpha)+(1-\alpha) \\ y_2 &=& \alpha^2(1-\alpha)+\alpha(1-\alpha)+(1-\alpha) \end{array} \quad (5.81)$$

and so forth. In general,

$$y_n = (1-\alpha)\sum_{k=0}^{n}\alpha^k = 1-\alpha^{n+1} = 1-e^{(n+1)\ln(\alpha)} \quad (5.82)$$

which has the form of a sampled version of the desired continuous response (5.79). (5.80) is thus an IIR filter realization of (5.79).

To express this IIR filter in the z domain, recall the z^{-1} is the z transform of a one sample delay. We can thus map the x_n and y_n in (5.80) to the z domain by multiplying each term by z^{-n} and summing over all n

$$\sum_{n=-\infty}^{\infty} y_n z^{-n} - \alpha\sum_{n=-\infty}^{\infty} y_{n-1}z^{-n} = (1-\alpha)\sum_{n=-\infty}^{\infty} x_n z^{-n} \quad (5.83)$$

which can be factored as

$$Y(z)(1-\alpha z^{-1}) = (1-\alpha)X(z) \quad (5.84)$$

to obtain the z transfer function

$$\frac{Y(z)}{X(z)} = \frac{1-\alpha}{1-\alpha z^{-1}}\ . \quad (5.85)$$

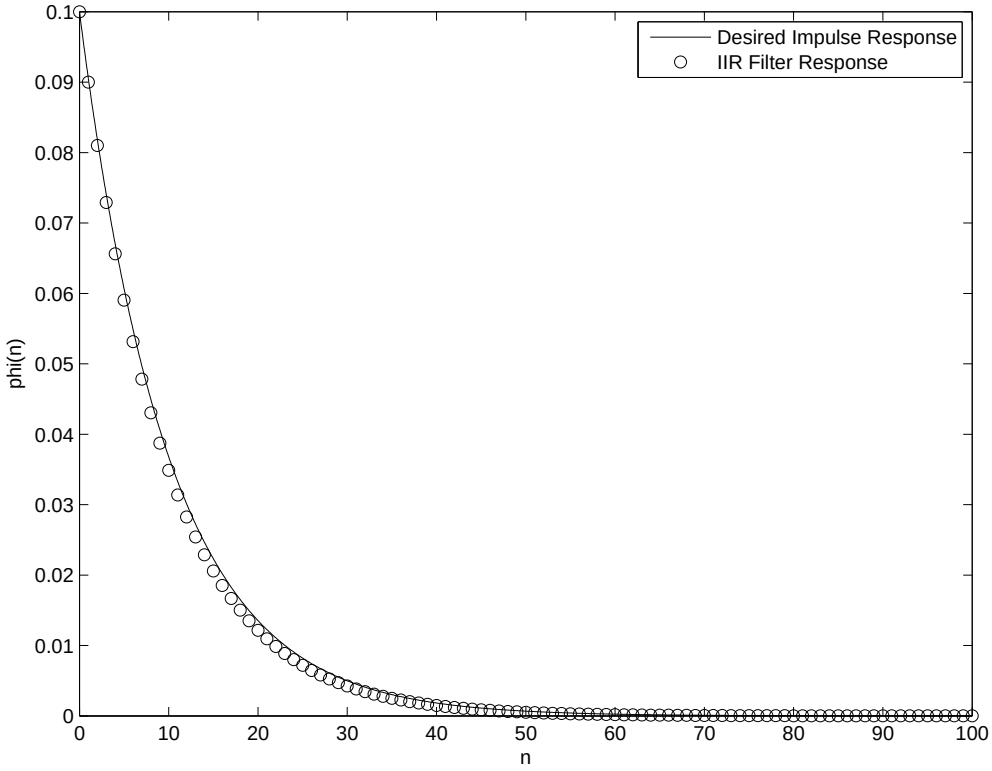

Figure 5.12: Impulse invariance discrete realization compared to a target continuous response in the time domain.

To evaluate the frequency response of (5.85), we evaluate the transfer function on the unit circle, or at $z = e^{2\pi\imath f/f_s}$, where f_s is the sampling frequency, and the Nyquist interval is the range of frequencies is thus $-f_s/2 \leq f \leq f_s/2$

$$\Phi(z = e^{2\pi\imath f/f_s}) = \frac{1-\alpha}{1-\alpha e^{-\imath 2\pi f/f_s}} \ . \tag{5.86}$$

The corresponding frequency response of the continuous system is given by (5.77). Both the continuous and discrete response functions are plotted in Figure 5.13, using $\tau = 10$ and the corresponding value for α, $\alpha = 1 - 1/\tau$, so that values for the discrete and continuous time functions agree at $n = 0$ and $t = 0$, respectively.

The major discrepancy in the frequency domain is that a discrete system has a periodic frequency response, and so, for this filter, must return to a value of 1 at $f = f_s$, while the continuous system continues to approach zero response with increasing frequency at a rate of about 6 dB per octave.

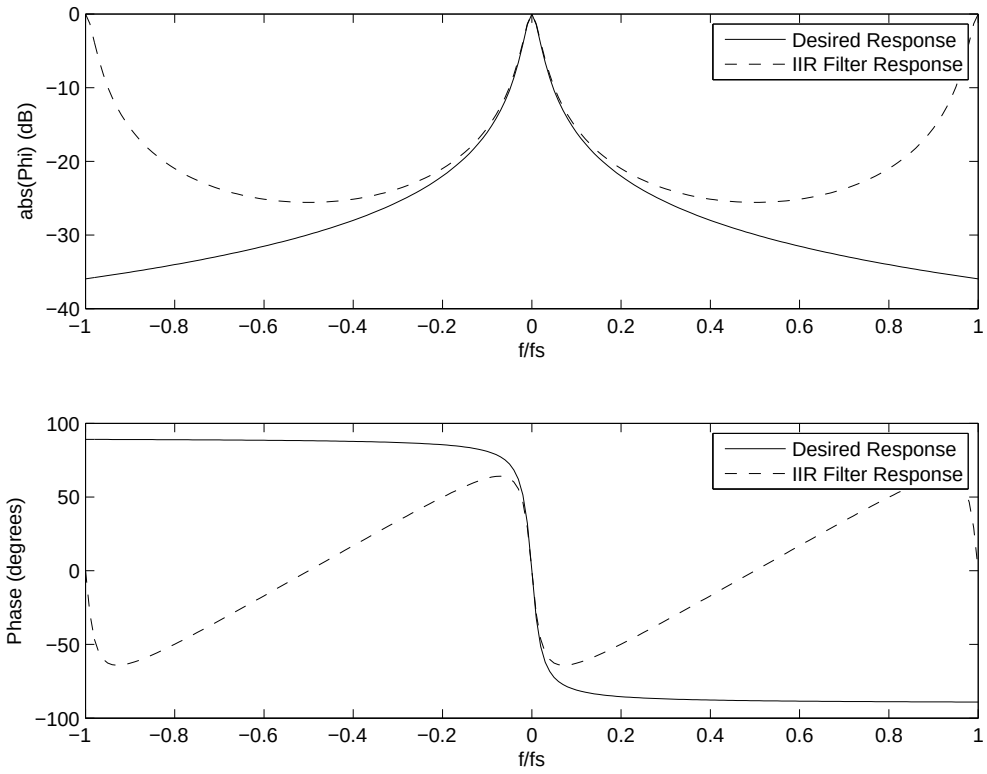

Figure 5.13: Impulse invariance discrete realization compared to a target continuous response in the frequency domain.

The Bilinear Transformation

Consider writing the discrete y sequence at a sampling interval of Δ as

$$y(n\Delta) = \int_{\Delta(n-1)}^{n\Delta} \dot{y}(u)du + y(\Delta[n-1]) \ . \tag{5.87}$$

Approximating the integral in (5.87) by the trapezoidal rule then gives

$$y(n\Delta) \approx \frac{\Delta}{2}\left[\dot{y}(\Delta[n-1]) + \dot{y}(n\Delta)\right] + y(\Delta[n-1]) \ . \tag{5.88}$$

which has the discrete time counterpart

$$y_n = \frac{\Delta}{2}\left[\dot{y}_{n-1} + \dot{y}_n\right] + y_{n-1} \tag{5.89}$$

The original (rewritten) differential equation (5.75) is

$$\dot{y}_n = \frac{1}{\tau}(x_n - y_n) \tag{5.90}$$

or equivalently

$$\dot{y}_{n-1} = \frac{1}{\tau}(x_{n-1} - y_{n-1}) \ . \tag{5.91}$$

Thus, we can eliminate the time derivatives by evaluating $\dot{y} + \dot{y}_{n-1}$ from the sum of (5.90) and (5.91) and substituting the result into (5.89). This yields

$$y_n = \frac{\Delta}{2\tau}\left[x_{n-1} - y_{n-1} + x_n - y_n\right] + y_{n-1} \tag{5.92}$$

or

$$y_n(1 + \frac{\Delta}{2\tau}) - y_{n-1}(1 - \frac{\Delta}{2\tau}) = \frac{\Delta}{2\tau}(x_n + x_{n-1}) \ . \tag{5.93}$$

(5.91) has the z transform

$$\Phi(z) = \frac{Y(z)}{X(z)} = \frac{\frac{\Delta}{2\tau}(1 + z^{-1})}{(1 + \frac{\Delta}{2\tau}) - (1 - \frac{\Delta}{2\tau})z^{-1}} \tag{5.94}$$

$$= \frac{(1 + z^{-1})}{(\frac{2\tau}{\Delta} + 1) - (\frac{2\tau}{\Delta} - 1)z^{-1}} \tag{5.95}$$

$$= \frac{1}{1 + \left(\frac{2\tau}{\Delta}\right)\left(\frac{1-z^{-1}}{1+z^{-1}}\right)} \ . \tag{5.96}$$

Evaluating the frequency response of (5.96) by taking $z = e^{2\pi\imath f/f_s}$, we get

$$\Phi(z = e^{i2\pi f/f_s}) = \frac{1}{1 + \left(\frac{2\tau}{\Delta}\right)\left(\frac{1-e^{-\imath 2\pi f/f_s}}{1+e^{-\imath 2\pi f/f_s}}\right)} \tag{5.97}$$

$$= \frac{1}{1 + \left(\frac{2\tau}{\Delta}\right) \imath \tan \pi f / f_s} \ . \tag{5.98}$$

(5.98) is thus the response of the continuous system (5.76) with the substitution

$$s = \frac{2i}{\Delta} \tan \pi f / f_s \ . \tag{5.99}$$

(5.98) is plotted along with the continuous response in Figure 5.15).

Recalling that the continuous frequency response (5.77) is just (5.76), evaluated at $s = \imath 2\pi f$, we can see that the frequency mapping between (5.98) and 5.77) is just

$$2\pi f_c = \frac{2}{\Delta} \tan \pi f_d / f_s \tag{5.100}$$

where f_d is the digital frequency and f_c is the continuous frequency. The continuous system frequency response tends to zero as $f_c \to \infty$. The bilinear z transform frequency response, on the other hand tends to zero where

$$\frac{\pi f_d}{f_s} = \frac{\pi}{2}(2m + 1) \tag{5.101}$$

or

$$f_d = \frac{f_s}{2}(2m + 1) \tag{5.102}$$

where m is an integer, which is just at odd multiples of the Nyquist frequency, $f_N = f_s/2$. The bilinear z-transform substitution (5.99) thus maps the semi-infinite frequency interval of the continuous system $(-\infty, \ \infty)$ into the Nyquist interval $[-f_N, \ f_N]$. To obtain the digital transfer function, $\Phi_d(z)$, from a given analog filter transfer function, $\Phi_a(s)$, we simply substitute

$$s = \frac{2}{\Delta} \frac{1 - z^{-1}}{1 + z^{-1}} \ . \tag{5.103}$$

An alternative explanation of the bilinear transform approach is that if $z = e^s$, and

$$\hat{s} = 2 \frac{1 - \frac{1}{z}}{1 + \frac{1}{z}} \tag{5.104}$$

then

$$\hat{s} = 2 \frac{1 - e^{-s}}{1 + e^{-s}} = 2 \frac{1 - e^{-2\pi \imath f}}{1 + e^{-2\pi \imath f}} = 2\imath \tan(\pi f) \ . \tag{5.105}$$

For small frequencies f, $\tan \pi f \approx \pi f$. Thus

$$\hat{s} \approx 2\pi \imath f \ . \tag{5.106}$$

That is, $\hat{s}$ is an approximation to s. By using $\hat{s}$ in place of s in the transfer function, we obtain a transfer function that can be expressed as a rational function of $1/z$.

Of course, we can never match the analog response with a digital system because of aliasing, but we can match some desirable characteristic of the analog

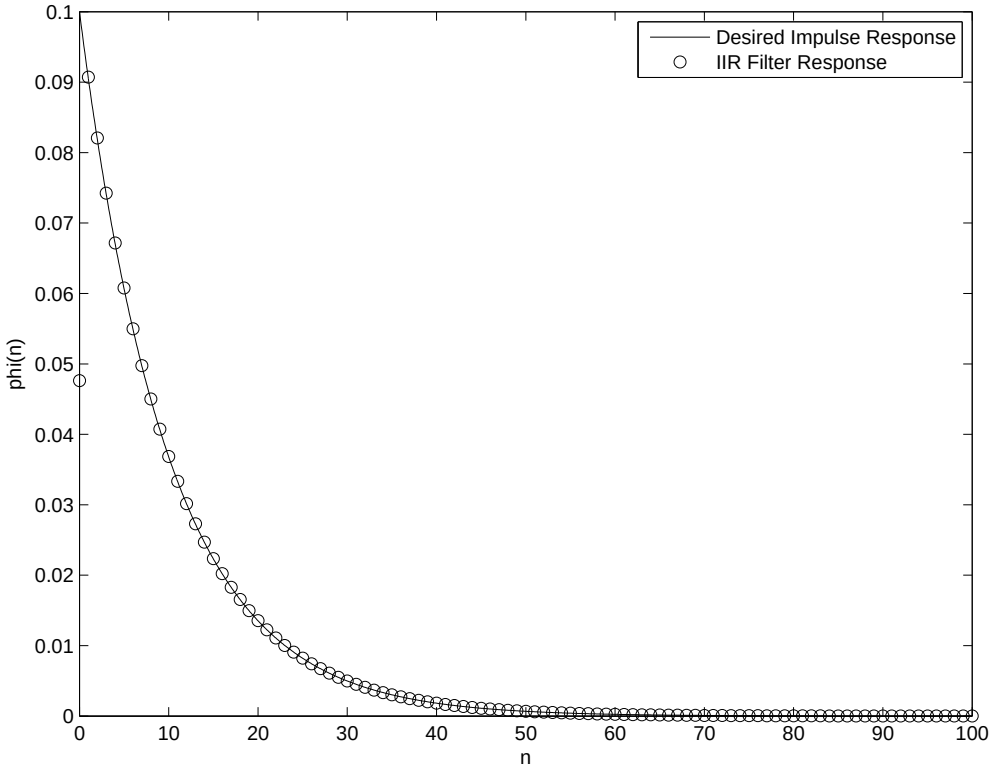

Figure 5.14: Bilinear z transform discrete realization response compared to a target continuous response in the time domain.

system (e.g., ripple height, corner frequency, etc.) within the Nyquist interval. In general, we can do this far more compactly with an IIR filter, but as always, there is a price, in this case IIR filters will have more complicated (non-linear) phase characteristics than FIR filters. We can see this directly by noting that the z transform of an FIR filter is just a polynomial in z^{-1}, while the z transform of a recursive filter is a ratio of two polynomials (a rational function) in z^{-1}.

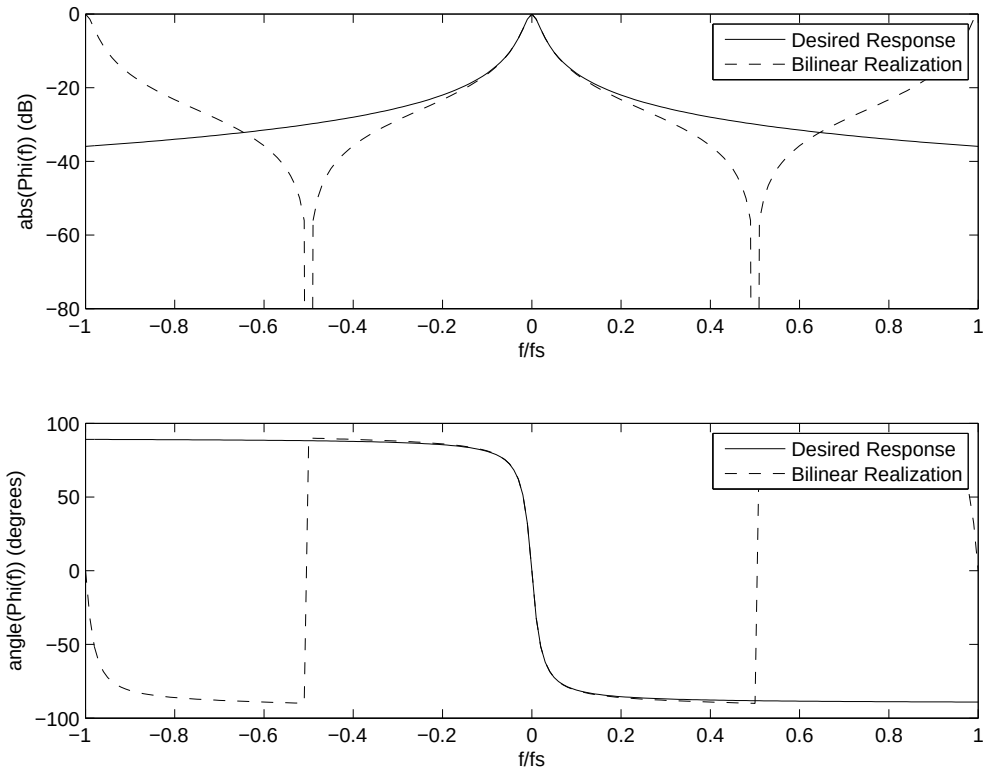

Figure 5.15: Bilinear z transform discrete realization response compared to a target continuous response in the frequency domain.

Some Common Filter Types

Some filter types are commonly encountered in a wide variety of applications. A high pass filter is designed to pass through all frequencies above a cut-off frequency f_c. The ideal high pass filter would have $|\Phi(f)| = 1$ for $f \geq f_c$ and $|\Phi(f)| = 0$ for $f < f_c$. Similarly, a low pass filter would have $|\Phi(f)| = 1$ for $f < f_c$ and $|\Phi(f)| = 0$ for $f \geq f_c$. An ideal band pass filter has $|\Phi(f)| = 1$ for $f_1 \leq f \leq f_2$ and $|\Phi(f)| = 0$ for frequencies outside of the pass band. An ideal band stop filter has $|\Phi(f)| = 0$ for $f_1 \leq f \leq f_2$ and $|\Phi(f)| = 1$ for frequencies outside of the pass bad.

In practice it's simply impossible to achieve these ideal frequency responses. However, it is possible to design filters that are optimal with respect to some objective criterion. The most common goal is to minimize the "ripple" in the pass band. This is simply the difference (in dB) between the largest value of $|\Phi(f)|$ and the smallest value of $|\Phi(f)|$ in the pass band. In the stop band, the typical goal is to minimize the maximum value of $\|\Phi(f)|$ in the stop band.

The Butterworth filter is designed to be optimally flat in its passband for amplitude response (i.e., to be "ripple"-less). Another very commonly applied filter is the Chebyshev filter. The type 1 Chebyshev filter has no more than R dB of ripple in the pass band, while the type 2 Chebyshev filter has $|\Phi(f)|$ at least R dB down within the pass band. The sharpness of the corner is controlled by the order of the filter (the number of poles in its transfer function). In general, higher order filters can realize very sharp transition bands in their amplitude response, but at a cost of complex (i.e., non-linear-phase) frequency response. See Figures 5.16 through 5 for examples of the frequency response that can be obtained with these filter designs.

Unfortunately, these highly optimized filters tend to have very odd phase responses. There is a useful trick that can be used to produce a zero phase filter with amplitude response that is the square of the amplitude response of the original filter. Given a signal $x(t)$, let $u(t) = \phi[x(t)]$. Let $v(t) = u(-t)$, effectively time reversing the filtered signal. Let $w(t) = \phi[v(t)]$. Finally, let $y(t) = w(-t)$, effectively time reversing the signal again. The MATLAB command **filtfilt** implements this procedure. Note that the resulting filter will be acausal. See Figures 5.22 and 5.23.

The amplitude of the frequency response of this filter is $|\Phi(f)|^2$. To see this, let $X(f)$ be the Fourier transform of $x(t)$. Then

$$U(f) = \Phi(f)X(f). \tag{5.107}$$

By the time reversal theorem, the Fourier transform of $u(-t)$ is $U(-f)$. Thus

$$V(f) = U(-f) = \Phi(-f)X(-f). \tag{5.108}$$

Then

$$W(f) = \Phi(f)V(f) = \Phi(f)\Phi(-f)X(-f). \tag{5.109}$$

Finally,

$$Y(f) = W(-f) = \Phi(-f)\Phi(f)X(f). \tag{5.110}$$

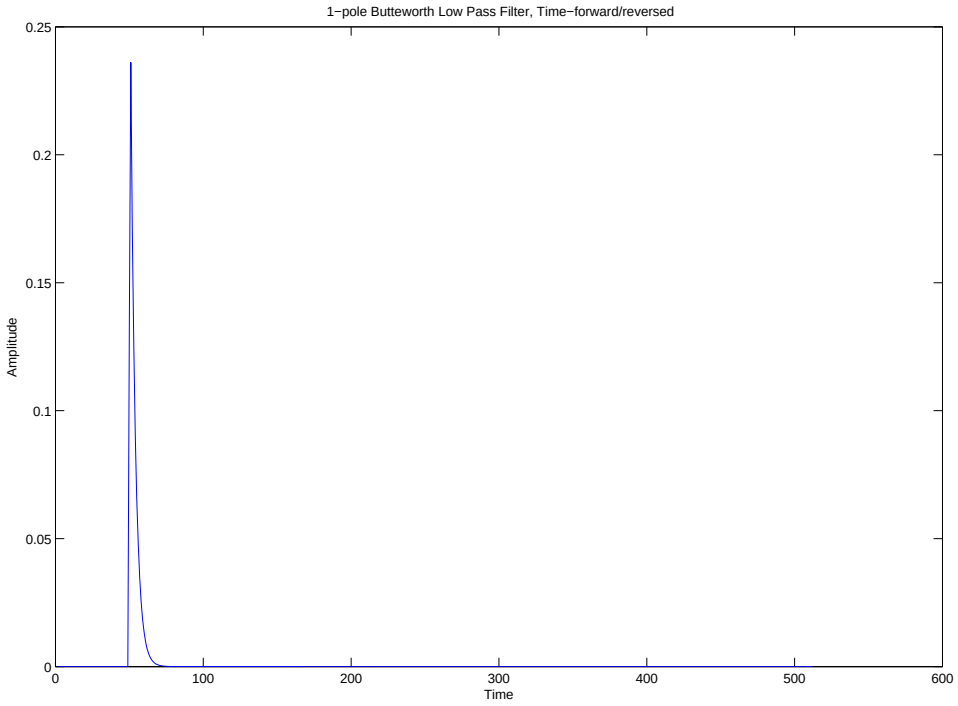

Figure 5.16: 1-pole Butterworth low-pass realization impulse response.

Assuming that the impulse response $\phi(t)$ is real, $\Phi(f)$ is Hermitian. Thus

$$Y(f) = \Phi(f)^* \Phi(f) X(f) = |\Phi(f)|^2 X(f). \tag{5.111}$$

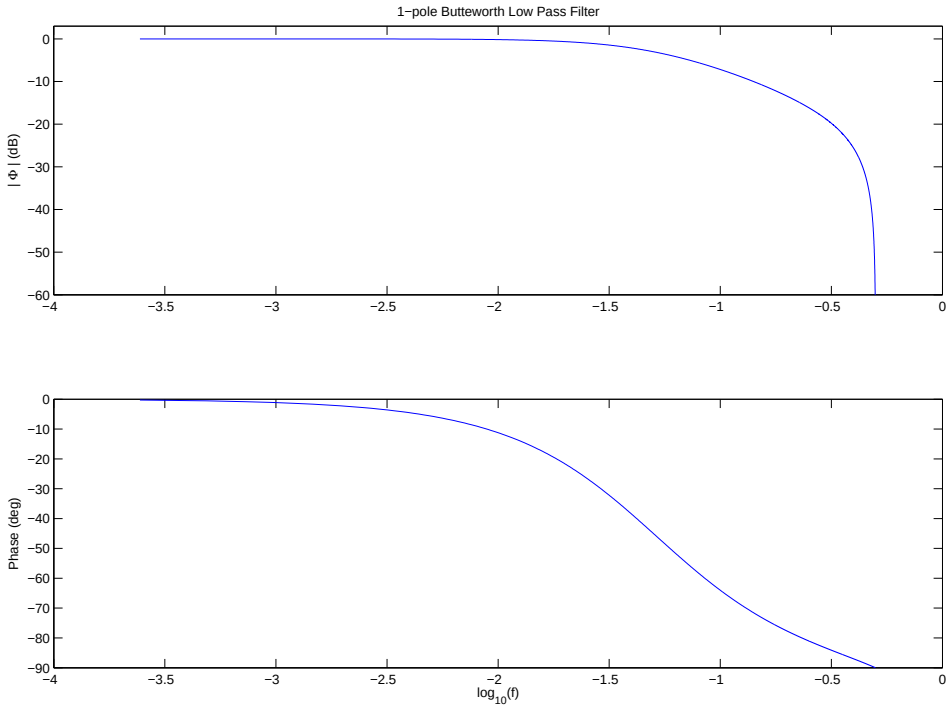

Figure 5.17: 1-pole Butterworth low-pass realization transfer function.

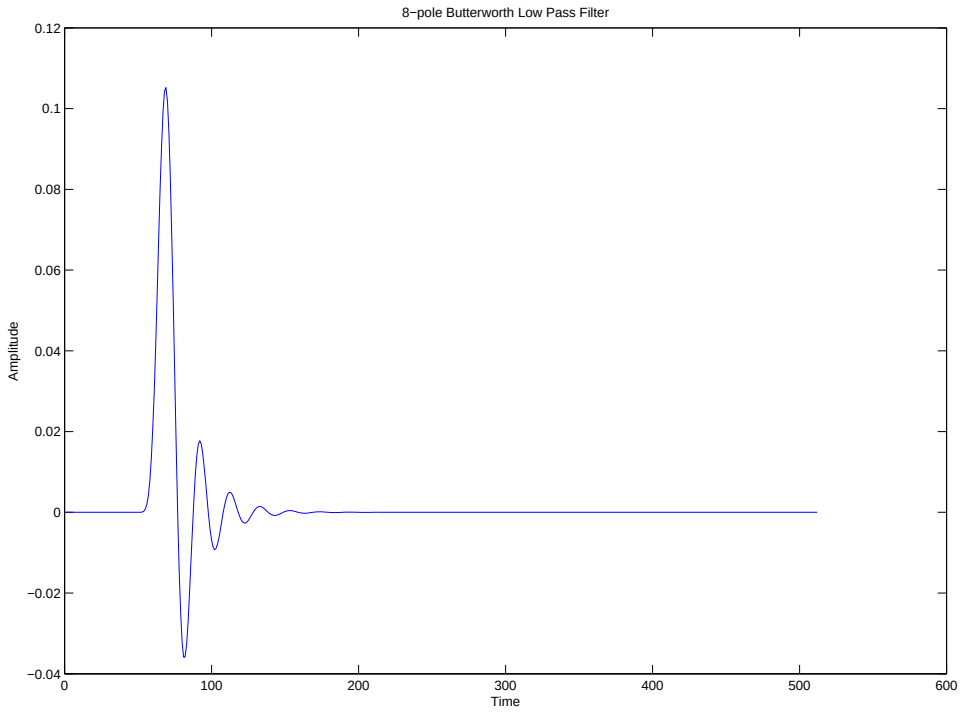

Figure 5.18: 8-pole Butterworth low-pass realization impulse response.

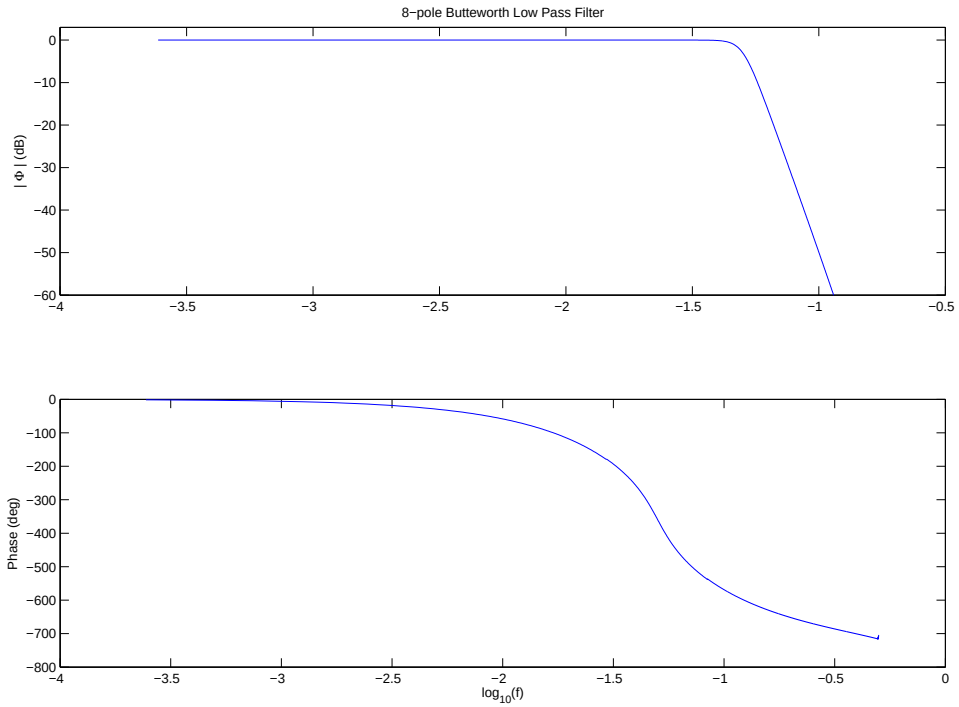

Figure 5.19: 8-pole Butterworth low-pass realization transfer function.

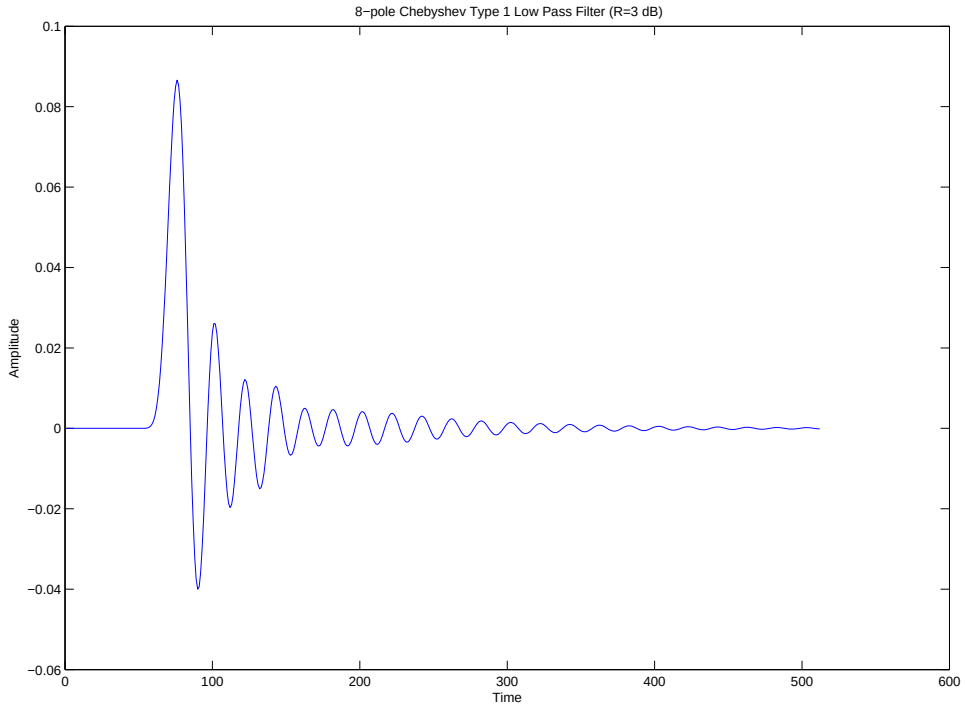

Figure 5.20: 8-pole Chebyshev (type 1) low-pass realization impulse response.

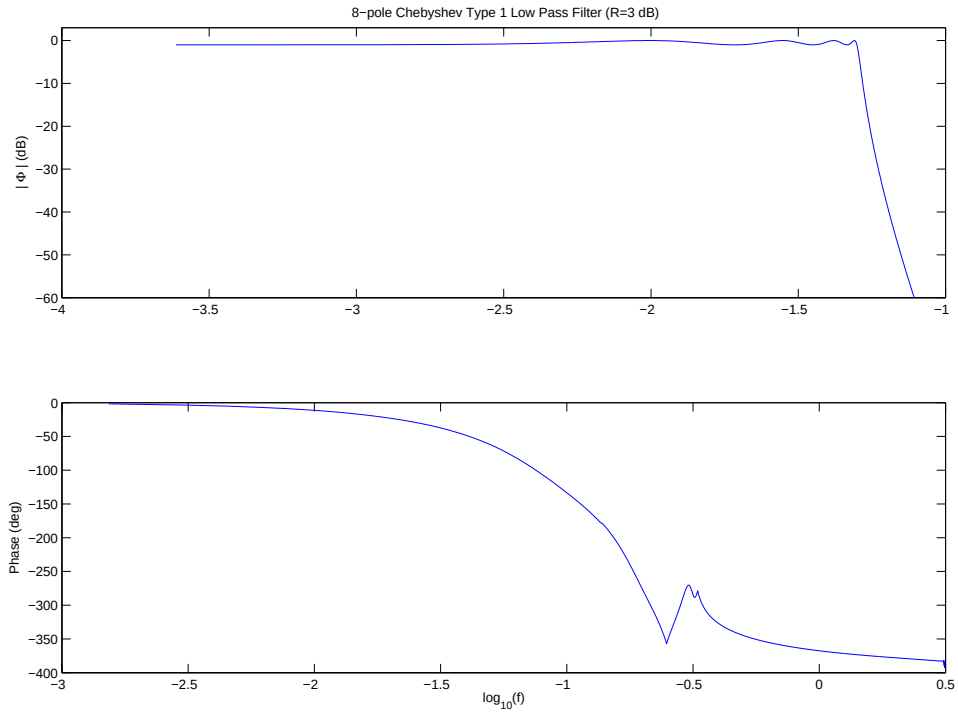

Figure 5.21: 8-pole Chebyshev (type 1) low-pass realization transfer function.

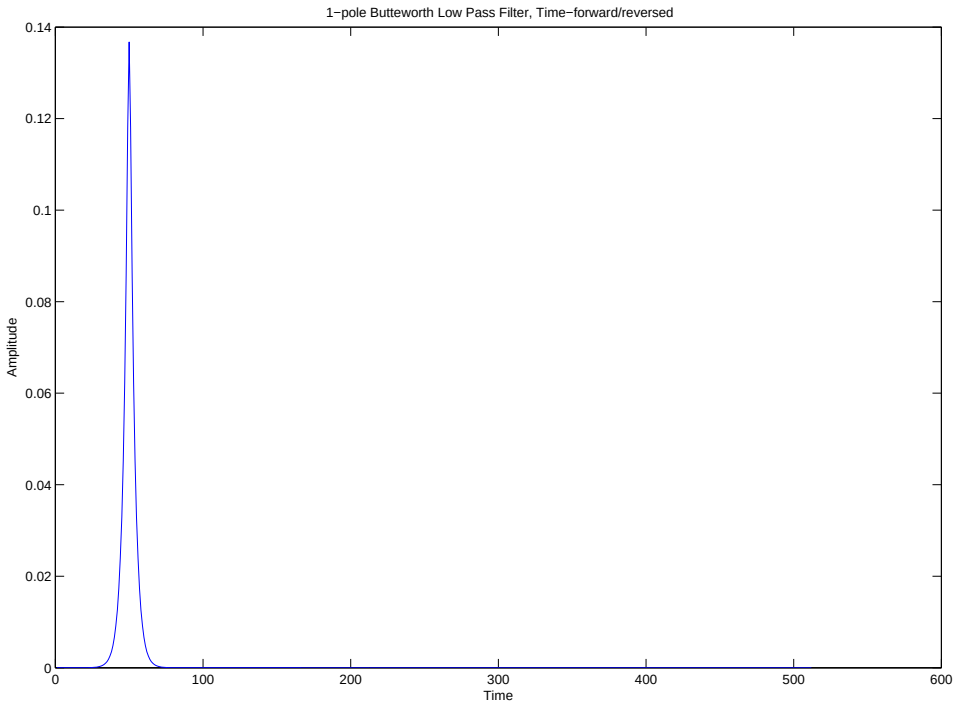

Figure 5.22: 1-pole Butterworth low-pass realization impulse response with forward-reverse-time filtering (2 poles, effectively).

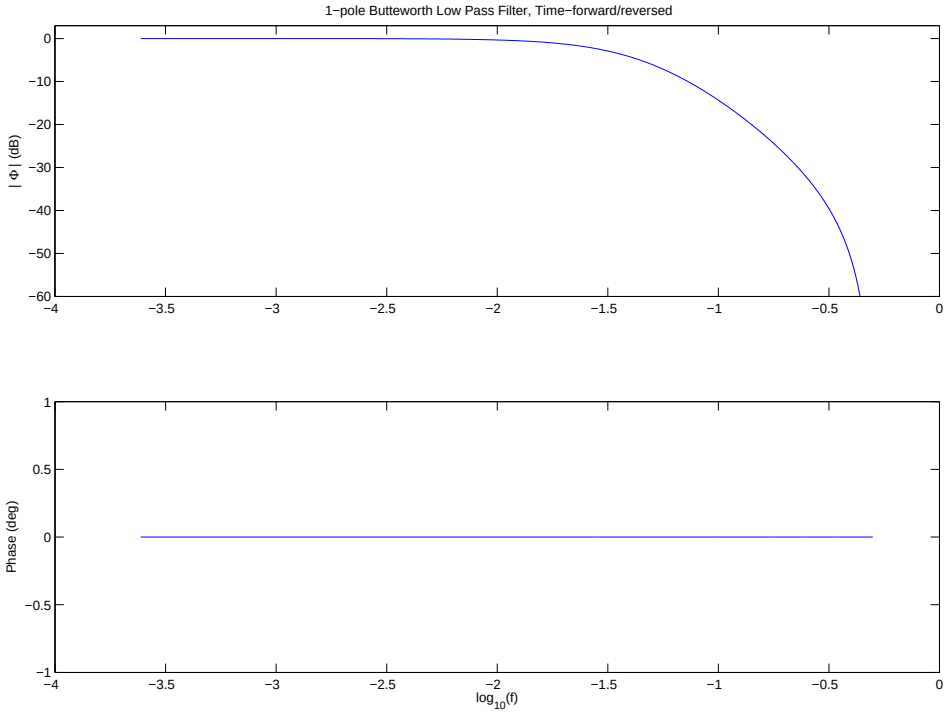

Figure 5.23: 1-pole Butterworth low-pass realization transfer function with forward-reverse-time filtering (2 poles, effectively).

Chapter 6

Deconvolution

We have seen how to perform convolution of discrete and continuous signals in both the time domain and with the help of the Fourier transform. In these lectures, we'll consider the problem of reversing convolution or deconvolving an input signal, given an output signal and the impulse response of a linear time invariant system.

We begin with the equation

$$d(t) = g(t) * m(t) \tag{6.1}$$

where $d(t)$ and $g(t)$ are known. Our goal is to solve for the unknown $m(t)$.

Although there's no obvious way to use the convolution integral to solve this equation, the equation becomes much easier to solve in the frequency domain. By the convolution theorem,

$$D(f) = G(f)M(f). \tag{6.2}$$

Thus

$$M(f) = \frac{D(f)}{G(f)}. \tag{6.3}$$

Once we have $M(f)$, we can invert the Fourier transform to obtain $m(t)$. Similarly, if we have discrete time signals and

$$d_n = (g_n * m_n)\Delta t \tag{6.4}$$

then

$$D_k = G_k M_k \Delta t \tag{6.5}$$

for $k = 0, 1, \ldots, N-1$. Solving for M_k, we get

$$M_k = \frac{D_k}{G_k \Delta t}. \tag{6.6}$$

Once we have the vector M we can invert the discrete Fourier transform to obtain m_n. This simple approach to solving the deconvolution problem is called **spectral division**.

Unfortunately, this method seldom works in practice. The first problem is that denominator in (6.3) might be zero, at least for some frequencies. In that case, $M(f)$ is undefined, and we can't invert the Fourier transform to obtain $m(t)$. Another way of looking at this is to consider what output the system will produce for sine waves at different frequencies. If the system produces zero output for a sine wave at a particular frequency f_0, then it's clear that we can't solve the deconvolution problem for any input signal that contains a sine wave at frequency f_0 because there's no evidence of this sine wave in the output!

What about noise? First, suppose that noise $n(t)$ is mixed with the true signal before the convolution. In that case we have

$$d(t) = g(t) * (m(t) + n(t)) \tag{6.7}$$

or

$$D(f) = G(f)(M(f) + N(f)). \tag{6.8}$$

If we perform spectral division, we obtain

$$M(f) + N(f) = \frac{D(f)}{G(f)}. \tag{6.9}$$

In this situation, the deconvolution hasn't made the noise any worse than it was before the deconvolution. Later in the course we'll discuss approaches to removing noise with a known frequency spectrum from such a signal.

Things get trickier if the noise is added after the convolution with $g(t)$. In that case, we have

$$d(t) = g(t) * m(t) + n(t) \tag{6.10}$$

or

$$D(f) = G(f)M(f) + N(f). \tag{6.11}$$

If we try to perform spectral division, we end up with

$$M(f) + \frac{N(f)}{G(f)} = \frac{D(f)}{G(f)}. \tag{6.12}$$

The $N(f)/G(f)$ term will introduce noise into the recovered signal. At frequencies where $G(f)$ is small but nonzero, the deconvolution process can greatly increase the magnitude of the noise.

Various techniques have been developed to deal with this noise. The basic idea is to avoid division by zero by somehow modifying the denominator in (6.6). This **regularizes** the deconvolution problem. In performing the regularization, we want to do as little as possible to frequencies where the noise is insignificant, while damping out the noise at frequencies where it is larger than the signal. Because the DFT of a real input signal is always Hermitian (i.e. $M_k = M^*_{N-k}$) it is important that we perform the regularization in a way that produces a Hermitian M sequence and a real signal m_n.

For example, we might try

$$M_k = \frac{D_k}{(G_k + \lambda)\Delta t} \tag{6.13}$$

where λ is a small positive real number. When G_k is much larger than λ, then this will have little effect on M_k. However, when G_k is very small compared to λ, this will effectively zero out the response at frequency k. One problem with this scheme is that if $G_k = -\lambda$, we can still get division by zero. It would obviously be better to work with the absolute value of G_k.

A scheme called **water level regularization** is widely used in geophysics. Since problems only occur at frequencies where $|G(f)|$ is small, we pick a critical level w and adjust $G(f)$ only when $|G(f)| \leq w$. At frequencies where $|G(f)| > w$ we simply perform spectral division. This has the advantage of not altering the spectral division method at good frequencies. At frequencies where $|G(f)|$ is small, we need to replace $G(f)$ with something that isn't too small. We could simply use w, but it is slightly better to use a complex number that at least has the same phase as $G(f)$. So we, use

$$\hat{G}(f) = w\frac{G(f)}{|G(f)|}. \tag{6.14}$$

If $G(f)$ is exactly zero this still causes problems! In that case, we'll use $\hat{G}(f) = w$. In discrete time, the water level deconvolution scheme can be written as

$$\hat{M}_k = \frac{D_k}{\hat{G}_k \Delta t} \tag{6.15}$$

where

$$\hat{G}_k = \begin{cases} G_k & |G_k| > w \\ \frac{wG_k}{|G_k|} & 0 < |G_k| \leq w \\ w & G_k = 0. \end{cases} \tag{6.16}$$

Note that $\hat{M}_k$ will be a Hermitian sequence. When we invert the transform to obtain m_n, we'll get back a real signal.

In order for the water level regularization to work we need to make sure that $w\Delta t$ is somewhat larger than $|N_k|$. If w is too large, then we simply get back d_n scaled down by a factor of w. If w is too small, than the result will be overly noisy, often at higher frequencies where $|G(f)|$ is smaller.

In the following example, A small amount of noise in the data makes spectral division unstable, but water level regularization produces very good results.

The input signal is $m(t) = te^{-t}$ and the impulse response is $g(t) = e^{-5t}\sin(10t)$. See Figures 6.1 and 6.2. This signal was sampled at intervals of $\Delta t = 0.01$ seconds. The convolved signal $d(t)$ is 10 seconds long, so there are 1001 samples.

Random noise was added to the signal with a normal distribution with mean 0 and standard deviation 0.0001. Figure 6.4 shows the noisy data. Figure 6.5 shows the unfortunate result of simple spectral division- the high frequency noise is greatly increased in amplitude.

The white noise that we added to the signal has approximately equal energy at all frequencies. Recall Parseval's theorem for the DFT,

$$\sum_{j=0}^{N-1} |x_j|^2 = \frac{1}{N}\sum_{k=0}^{N-1} |X_k|^2. \tag{6.17}$$

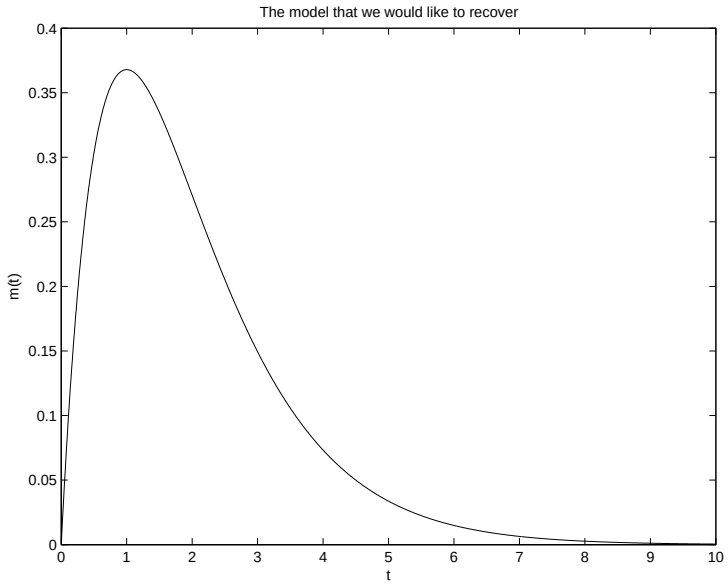

Figure 6.1: The input signal.

Since $N = 1001$, we expect the $|N_k|$ values of our noise to be about $\sqrt{1001}$ or roughly 30 times larger than the $|n_n$ values. Thus a typical value of $|N_k|$ should be about 0.003. In order to make the values of $w\Delta t$ larger than 0.003, we'd like to have $w \geq 1$.

Figures 6.6 through 6.8 shows the results obtained with $w = 0.1$, $w = 1$, and $w = 10$. Although the solution is under regularized at $w = 0.1$, the solution is quite good by the time we get to $w = 10$.

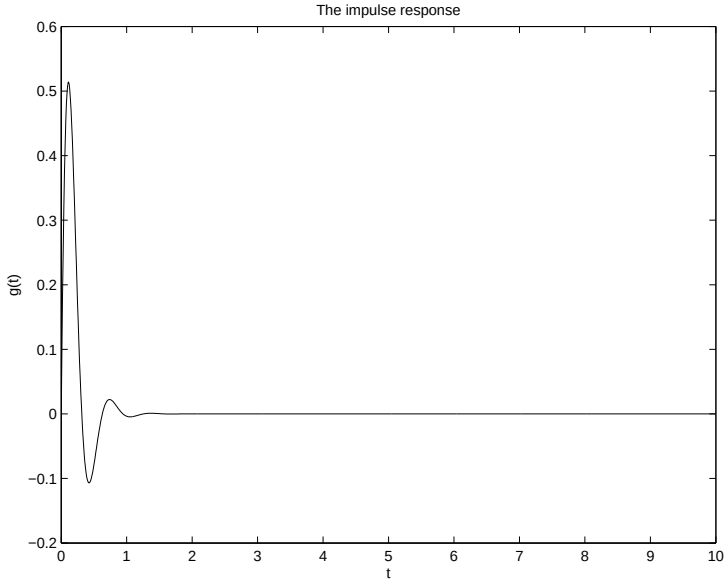

Figure 6.2: The impulse response.

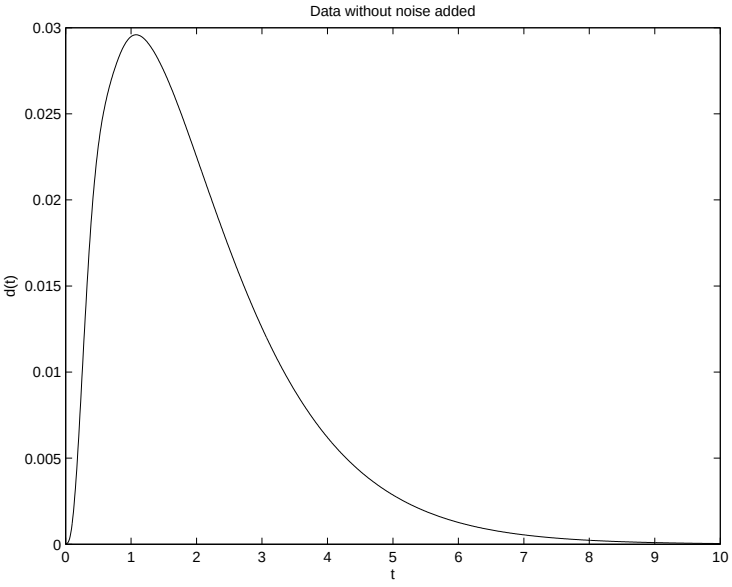

Figure 6.3: Clean data.

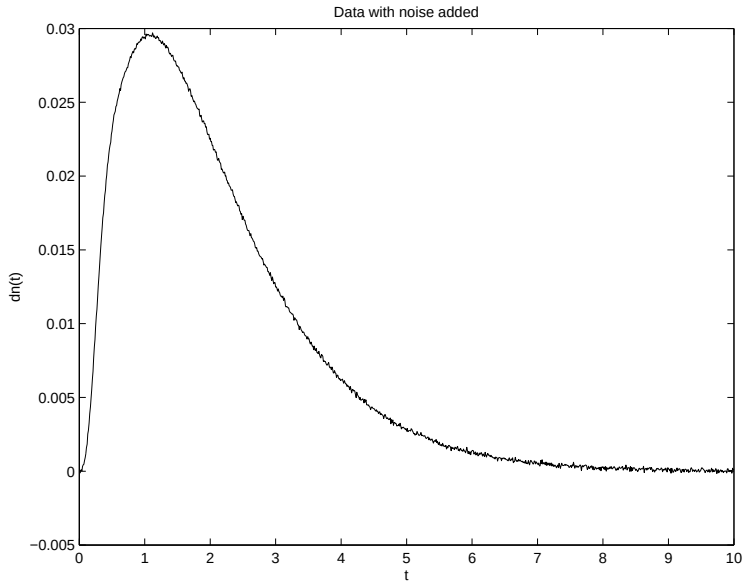

Figure 6.4: The data with a small amount of noise added.

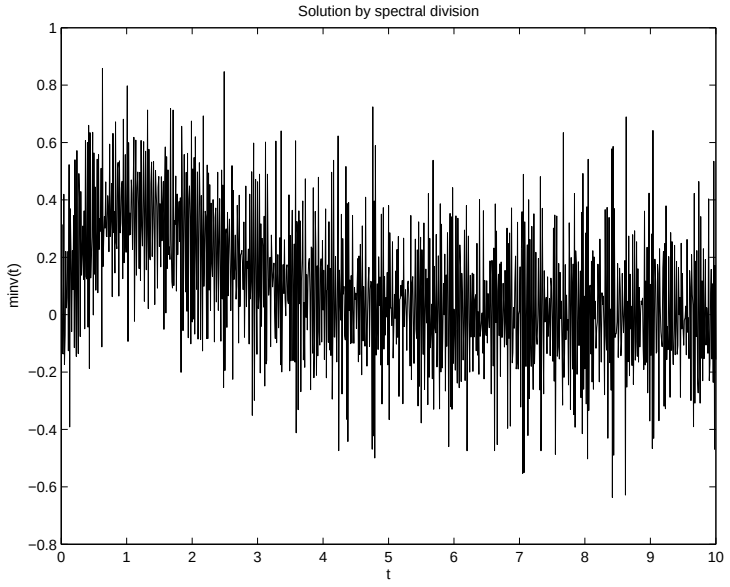

Figure 6.5: Deconvolution by spectral division, no regularization.

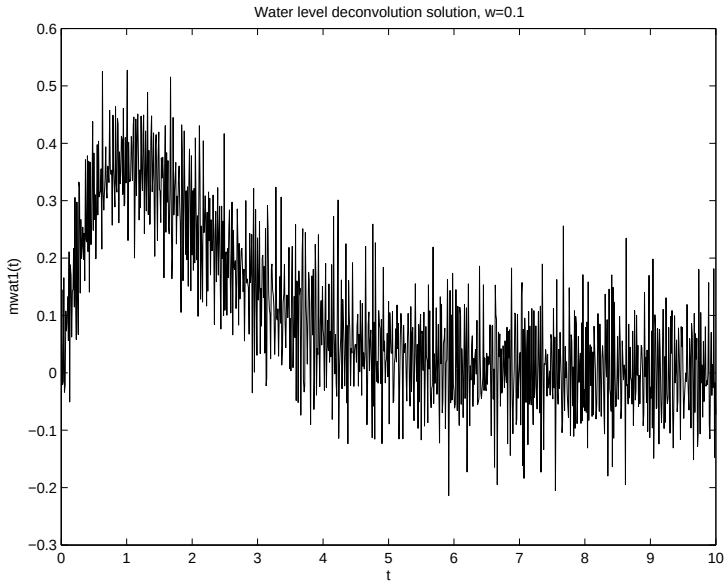

Figure 6.6: Water level solution, $w = 0.1$.

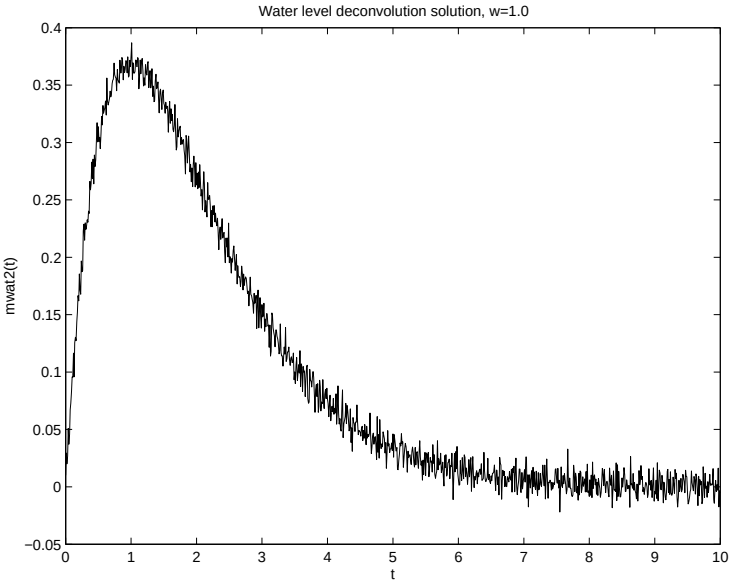

Figure 6.7: Water level solution, $w = 1.0$.

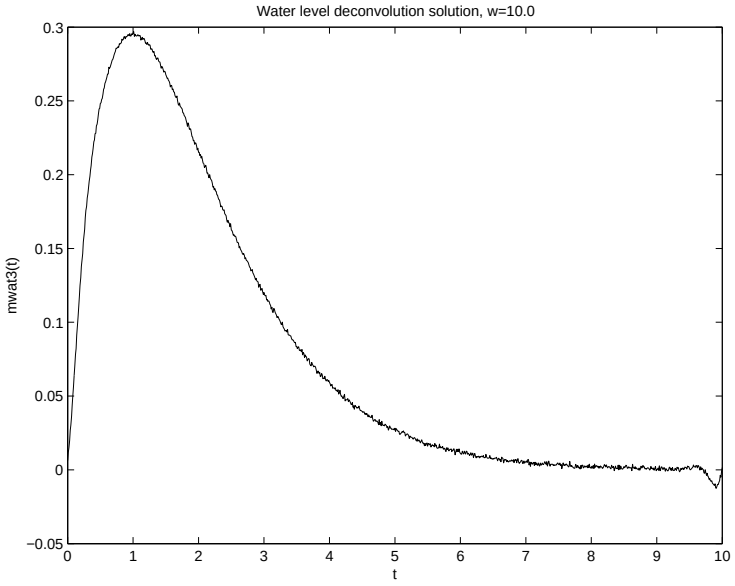

Figure 6.8: Water level solution, $w = 10.0$.

In Tikhonov regularization, we use

$$\hat{M}_k = \frac{G_k^* D_k}{(G_k^* G_k + \lambda)\Delta t} \tag{6.18}$$

where λ is a small positive parameter. This is similar to (6.13), in that when $|G_k|$ is much larger in magnitude than λ, we get essentially (6.6). However, when $|G_k|$ is much smaller than λ, M_k is reduced in magnitude. It's not hard to show that if M_k is obtained by Tikhonov regularization then M_k will be Hermitian. Furthermore, the denominator in this formula can never be 0.

The size of the factor

$$\frac{G_k^* N_k}{(G_k^* G_k + \lambda)\Delta t} \tag{6.19}$$

determines whether a noise frequency k will be effectively eliminated from the deconvolved signal. To get rid of the noise, we want

$$|G_k^* N_k| < \lambda \Delta t. \tag{6.20}$$

This gives us a very simple criteria for picking λ. We'll discuss more sophisticated methods for picking λ in the inverse problems course.

Returning to our earlier example, we know that $|N_k|$ is typically about 0.003, while $|G_k|$ is typically around 3. Thus we need $\lambda \Delta t > 0.01$ or $\lambda > 1$ to cover the noise. Figures 6.9 through 6.11 show the effect of different values of the regularization parameter λ.

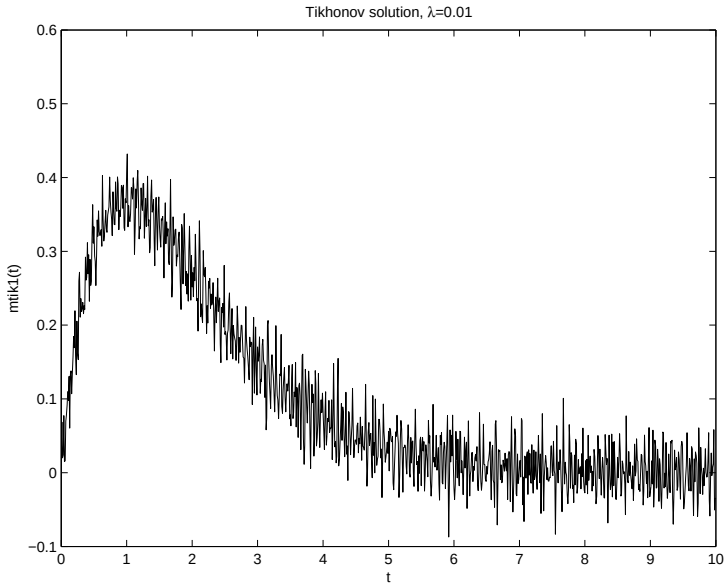

Figure 6.9: Deconvolution with Tikhonov regularization, $\lambda = 0.01$.

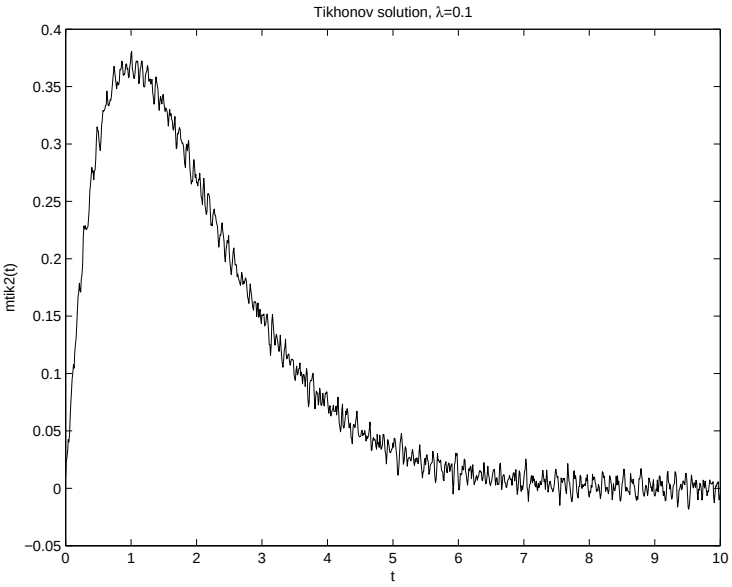

Figure 6.10: Deconvolution with Tikhonov regularization, $\lambda = 0.1$.

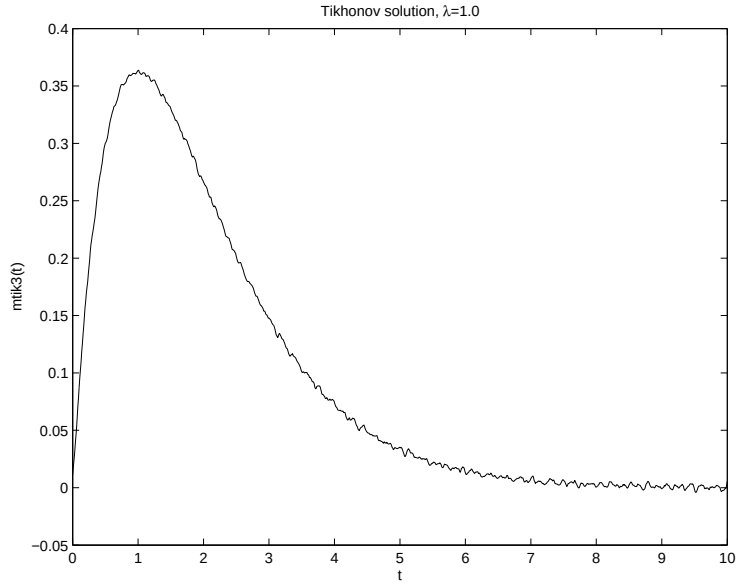

Figure 6.11: Deconvolution with Tikhonov regularization, $\lambda = 1.0$.

It can be shown that Tikhonov regularization minimizes

$$\min \|G \cdot M - D\|_2^2 + \lambda \|M\|_2^2. \tag{6.21}$$

By Parseval's theorem, this is equivalent to minimizing

$$\min \|g * m - d\|_2^2 + \lambda \|m\|_2^2. \tag{6.22}$$

The objective function is a weighted sum of a term that measures how well the model m fits the data d and a term that measures the energy of the model m. Tikhonov regularization is effectively picking the smallest energy signal that fits the data reasonably well, with the relative balance of these two factors controlled by the regularization parameter λ.

An alternative formulation of Tikhonov regularization sets a limit δ on the data misfit and then minimizes $\|m\|_2$.

$$\begin{array}{rcl} \min & \|m\|_2 & \\ & \|g * m - d\|_2 & \leq \quad \delta. \end{array} \tag{6.23}$$

There are situations in which other kinds of regularization are appropriate. We'll consider an example in which a controlled source (e.g. a vibroseis truck) is used to send a seismic wave down into the earth. The wave bounces back from reflecting layers at various depths and a seismograph of the reflected signal is recorded. We'd like to recover the depths of these reflecting layers.

Here, $g(t)$ is the known source signal, $d(t)$ is the recorded seismograph, and $m(t)$ is the unknown. The reflector should appear in $m(t)$ as scaled delta functions, with a reflect at time "depth" $t_0/2$ appearing as a scaled $\delta(t - t_0)$.

In this case, we want m to be a simple sequence of spikes. Rather than using Tikhonov regularization to minimize $\|m\|_2$, we want to minimize the number of nonzero entries in m. Let $\|m\|_0$ be the number of nonzero entries in m. Than we can formulate our regularization problem as

$$\begin{array}{rcl} \min & \|m\|_0 & \\ & \|g * m - d\|_2 & \leq \quad \delta. \end{array} \tag{6.24}$$

Unfortunately, these kinds of optimization problems are extremely difficult to solve.

A surprisingly effective alternative is to instead minimize

$$\|m\|_1 = \sum_{j=1}^{n} |m_j|. \tag{6.25}$$

The regularization problem is then

$$\begin{array}{rcl} \min & \|m\|_1 & \\ & \|g * m - d\|_2 & \leq \quad \delta. \end{array} \tag{6.26}$$

It turns out that these problems can be effectively solved by convex optimization techniques.

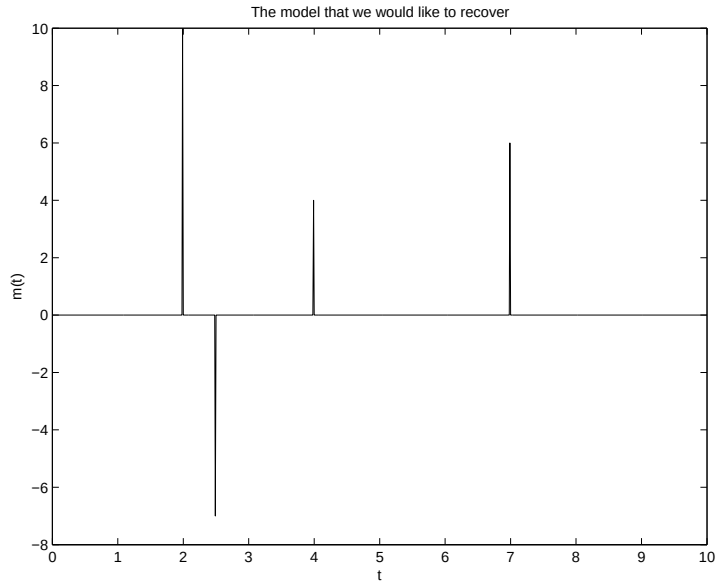

Figure 6.12: The target model $m(t)$.

For an example, we'll use the same impulse response from our previous example.

$$g(t) = e^{-5t}\sin(10t). \tag{6.27}$$

This time our target model $m(t)$ will be

$$m(t) = 10\delta(t-2) - 7\delta(t-2.5) + 4\delta(t-4) + 6\delta(t-7). \tag{6.28}$$

Again we'll add random noise to the convolved signal and then attempt to recover $m(t)$.

Figure 6.12 shows the target model. Figure 6.13 shows the data with noise added. It's quite hard to pick out the impulses in this plot. Figures 6.14 shows the best result that could be obtained with Tikhonov regularization. The impulses are artificially broadened, and the noise is not completely removed from the signal. Figure 6.15 shows using (6.26) produces an amazingly good recovery of $m(t)$. Notice that the spikes are correctly placed in time. The amplitude of the spikes is reduced and the spikes are slightly broader than they should be, but the results are vastly better than the results obtained with Tikhonov regularization.

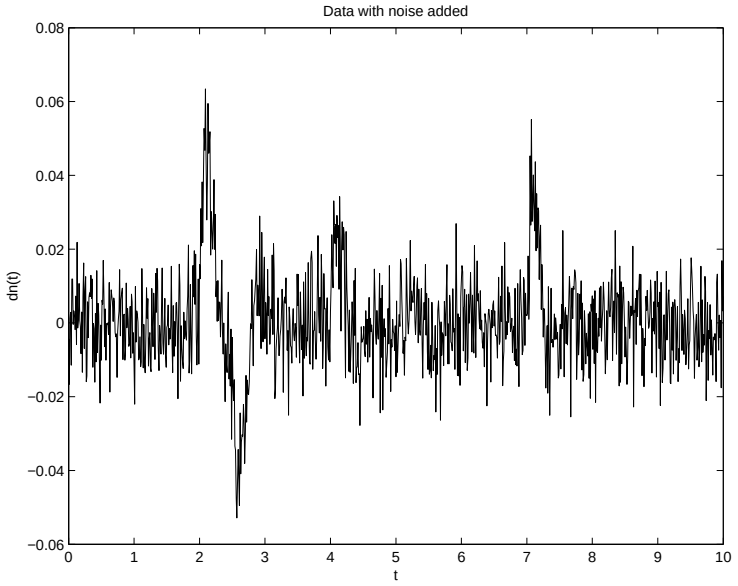

Figure 6.13: Data with noise added.

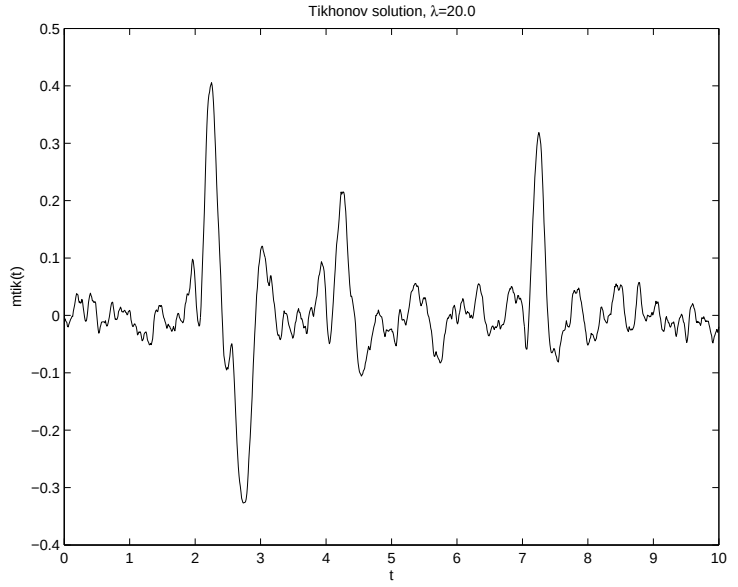

Figure 6.14: Tikhonov solution.

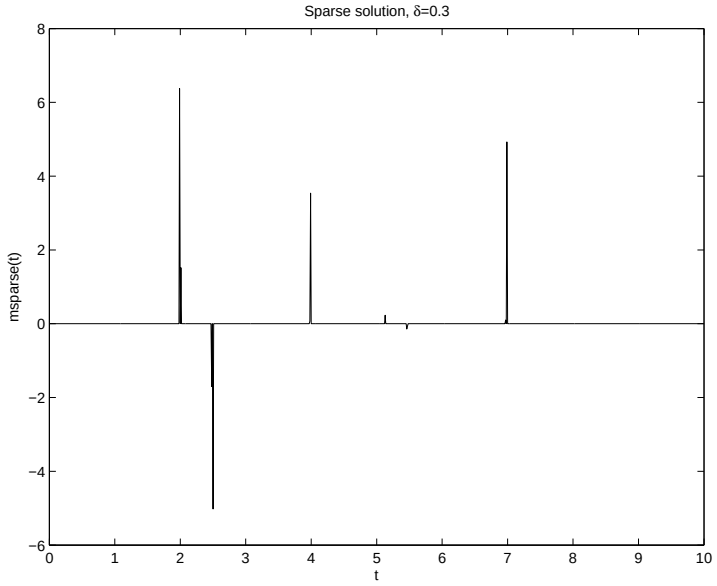

Figure 6.15: 1-norm regularized solution.

Chapter 7

Introduction to Multidimensional and Multichannel Processing

We have now covered most of the basic tools in analyzing one-dimensional time or spatial series. Many data sets in geophysics and other fields, however, are inherently multi-dimensional, either because the independent variable is multidimensional (e.g., a 2-dimensional survey or a 3-dimensional structure) or because the data itself is a vector quantity (e.g., three-component seismic or electromagnetic data).

Two or higher dimensional data sets require a multidimensional analysis technique. Some examples include photographic records, remote sensing data, or other 2-d images, seismic records from a 2-dimensional array, and gravity and magnetic surveys. Other signals may be considered multidimensional, with the two axes being physically different, such as a linear array of seismometers, where one dimension is temporal and the other is spatial or a two-dimensional array with a third time dimension. In general, much of one's intuition developed from analyzing 1-dimensional systems may be applied, although there are some very important concepts of 1-dimensional systems which do not apply in more dimensions.

Let $x(n_1, n_2)$ be a two-dimensional sequence defined for integer n_1 and n_2. Such a 2-d sequence is usually obtained from sampling a continuous 2-dimensional function. Some examples of 2-d sequences would be the unit impulse:

$$\delta(n_1, n_2) = \begin{cases} 1 \text{ for } n_1 = n_2 = 0 \\ 0 \text{ otherwise} \end{cases} \tag{7.1}$$

the step function

$$H(n_1, n_2) = \begin{cases} 1 \text{ for } n_1, n_2 \geq 0 \\ 0 \text{ otherwise} \end{cases} \tag{7.2}$$

the exponential

$$x(n_1, n_2) = \begin{cases} \alpha_1^{n_1}\alpha_2^{n_2} \text{ for } n_1, n_2 \geq 0 \\ 0 \text{ otherwise} \end{cases} \tag{7.3}$$

and the sinusoid

$$x(n_1, n_2) = e^{\imath 2\pi(f_1 n_1 + f_2 n_2)} \ . \tag{7.4}$$

If a system is linear and time invariant, then convolution is a valid concept in dimensions higher than 1, thus if $x(n_1, n_2)$ is an input to a two dimensional system which has an impulse response of $\phi(n_1, n_2)$, then the output is

$$y(n_1, n_2) = x(n_1, n_2) * \phi(n_1, n_2) = \sum_{m_1=-\infty}^{\infty} \sum_{m_2=-\infty}^{\infty} \phi(m_1, m_2) x(n_1 - m_1, n_2 - m_2) \tag{7.5}$$

$$= \sum_{m_1=-\infty}^{\infty} \sum_{m_2=-\infty}^{\infty} \phi(n_1 - m_1, n_2 - m_2) x(m_1, m_2) \tag{7.6}$$

(7.6) is usually difficult to apply, however, consider a simple case given in [11], where

$$\phi(n_1, n_2) = \alpha^{n_1 n_2} \tag{7.7}$$

and

$$x(n_1, n_2) = \begin{cases} 1 \text{ for } 0 \leq n_1, n_2 \leq 2 \\ 0 \text{ otherwise} \end{cases} \tag{7.8}$$

the response, $\phi(n_1, n_2) * x(n_1, n_2)$ is thus

$$y(n_1, n_2) = \sum_{m_1=0}^{2} \sum_{m_2=0}^{2} \alpha^{(n_1 - m_1)(n_2 - m_2)} \tag{7.9}$$

which, in general must be evaluated term by term for each (n_1, n_2) where each term requires $3^2 = 9$ operations. If $\phi(n_1, n_2)$ is *separable* , i.e., it can be written as

$$\phi(n_1, n_2) = g(n_1) \cdot f(n_2) \tag{7.10}$$

then the response can be calculated in terms of consecutive 1-dimensional convolutions, as (7.6) now becomes

$$y(n_1, n_2) = \sum_{m_1=-\infty}^{\infty} \sum_{m_2=-\infty}^{\infty} g(m_1) f(m_2) x(n_1 - m_1, n_2 - m_2) \tag{7.11}$$

$$= \sum_{m_1=-\infty}^{\infty} g(m_1) \left(\sum_{m_2=-\infty}^{\infty} f(m_2) x(n_1 - m_1, n_2 - m_2) \right) \tag{7.12}$$

where the term inside of the parentheses is a sequence of 1-d convolutions where m_1 is allowed to range from $-\infty$ to ∞. If the input sequence is also separable, so that $x(n_1, n_2) = a(n_1) \cdot b(n_2)$, then

$$y(n_1, n_2) = \sum_{m_1=-\infty}^{\infty} \sum_{m_2=-\infty}^{\infty} g(m_1) f(m_2) a(n_1 - m_1) b(n_2 - m_2) \tag{7.13}$$

$$= \left(\sum_{m_1=-\infty}^{\infty} g(m_1)a(n_1 - m_1) \right) \left(\sum_{m_2=-\infty}^{\infty} f(m_2)b(n_2 - m_2) \right) \tag{7.14}$$

which is also separable, i.e.,

$$y(n_1, n_2) = \alpha(n_1) \cdot \beta(n_2) \tag{7.15}$$

where $\alpha(n_1)$ and $\beta(n_2)$ are 1-dimensional convolutions (7.14).

As in 1-d systems, sinusoidal inputs play the fundamental functional role in the Fourier analysis of 2-d systems. This is because 2-dimensional sinusoidal functions

$$x(n_1, n_2) = e^{\imath 2\pi f_1 n_1} e^{\imath 2\pi f_2 n_2} \tag{7.16}$$

are eigenfunctions of the 2-d convolution operation. Consider the output of a system with impulse response $\phi(n_1, n_2)$ to a complex exponential input

$$y(n_1, n_2) = \sum_{m_1=-\infty}^{\infty} \sum_{m_2=-\infty}^{\infty} \phi(m_1, m_2) e^{\imath 2\pi f_1 (n_1 - m_1)} e^{\imath 2\pi f_2 (n_2 - m_2)} \tag{7.17}$$

$$= e^{\imath 2\pi f_1 n_1} e^{\imath 2\pi f_2 n_2} \sum_{m_1=-\infty}^{\infty} \sum_{m_2=-\infty}^{\infty} \phi(m_1, m_2) e^{-\imath 2\pi f_1 m_1} e^{-\imath 2\pi f_2 m_2} = x(n_1, n_2)\Phi(f_1, f_2) \tag{7.18}$$

where $\Phi(f_1, f_2)$ is the frequency response of the system in two dimensions and hence defines a 2-d Fourier transform of a 2-d sampled function. The corresponding inverse transformation (see table below) is just

$$\phi(n_1, n_2) = \int_{-1/2}^{1/2} \int_{-1/2}^{1/2} \Phi(f_1, f_2) e^{\imath 2\pi f_1 n_1} e^{\imath 2\pi f_2 n_2} \, df_1 \, df_2 \; . \tag{7.19}$$

Note that $\Phi(f_1, f_2)$ is periodic in frequency with unit period for both f_1 and f_2, as we'd expect for a sampled function

$$\Phi(f_1, f_2) = \phi(f_1 + l, f_2 + m) \; (l, m \text{ integers}) \tag{7.20}$$

which is two-dimensional aliasing. If $\phi(n_1, n_2)$ is real, then

$$\Phi(f_1, f_2) = \sum_{m_1=-\infty}^{\infty} \sum_{m_2=-\infty}^{\infty} \phi(m_1, m_2) e^{-\imath 2\pi f_1 m_1} e^{-\imath 2\pi f_2 m_2} = \Phi^*(-f_1, -f_2) \tag{7.21}$$

so that $\Phi(f_1, f_2)$ is Hermitian in a 2-d sense.

We now have the tools for performing windowed filter design in 2-dimensions in a manner entirely analogous to that which we previously examined for 1-d FIR filters. Consider the perfect low pass filter with a response given by

$$\Phi(f_1, f_2) = \begin{cases} 1 \text{ for } -\alpha \le f_1 \le \alpha, -\beta \le f_2 \le \beta \\ 0 \text{ otherwise} \end{cases} \tag{7.22}$$

taking the inverse Fourier transform gives the n domain series

$$\phi(n_1, n_2) = \int_{-\alpha}^{\alpha} \int_{-\beta}^{\beta} e^{\imath 2\pi f_2 n_2} e^{\imath 2\pi f_1 n_1} df_2 df_1 \tag{7.23}$$

if the frequency response is separable, so is the n domain response, so

$$\phi(n_1, n_2) = \left(\int_{-\alpha}^{\alpha} e^{\imath 2\pi f_1 n_1} df_1 \right) \left(\int_{-\beta}^{\beta} e^{\imath 2\pi f_2 n_2} df_2 \right) \tag{7.24}$$

$$= \left(\frac{e^{\imath 2\pi f_1 n_1}}{\imath 2\pi n_1} \right) \Bigg|_{f_1=-\alpha}^{\alpha} \left(\frac{e^{\imath 2\pi f_2 n_2}}{\imath 2\pi n_2} \right) \Bigg|_{f_2=-\beta}^{\beta} = \left(\frac{\sin(2\pi\alpha n_1)}{\pi n_1} \right) \left(\frac{\sin(2\pi\beta n_2)}{\pi n_2} \right). \tag{7.25}$$

This frequency response and a plot of its corresponding filter weights is shown on the following page.

Unless we have a physical reason for wishing to treat the n_1 and n_2 directions unequally, we would generally want to have a response which is circularly symmetric in the time and frequency domains. Such a filter is specified by

$$\Phi(f_1, f_2) = \begin{cases} 1 & f_1^2 + f_2^2 \leq f_{\max}^2 \\ 0 & \text{otherwise} \end{cases} \tag{7.26}$$

and the corresponding filter weights are obtainable as

$$\phi(n_1, n_2) = \int_{-f_{\max}}^{f_{\max}} \int_{-(f_{\max}^2 - f_1^2)^{1/2}}^{(f_{\max}^2 - f_1^2)^{1/2}} e^{\imath 2\pi f_1 n_1} e^{\imath 2\pi f_2 n_2} \; df_2 \; df_1 \; . \tag{7.27}$$

An easy way to evaluate this integral is to note that both the n and frequency response of the system are circularly symmetric, thus, we can obtain the general solution by finding $\phi(n_1, 0)$ and then substituting $(n_1^2 + n_2^2)^{1/2}$ for n_1.

$$\phi(n_1, 0) = \int_{-f_{\max}}^{f_{\max}} \int_{-(f_{\max}^2 - f_1^2)^{1/2}}^{(f_{\max}^2 - f_1^2)^{1/2}} e^{\imath 2\pi f_1 n_1} \; df_2 \; df_1 \tag{7.28}$$

$$= \int_{-f_{\max}}^{f_{\max}} e^{\imath 2\pi f_1 n_1} \cdot 2(f_{\max}^2 - f_1^2)^{1/2} \; df_1 \tag{7.29}$$

using the polar substitution $f_1 = f_{\max} \sin\theta$ gives

$$= \int_{-\pi/2}^{\pi/2} 2(f_{\max}^2 - f_{\max}^2 \sin^2\theta)^{1/2} e^{\imath 2\pi f_{\max} n_1 \sin\theta} \cdot f_{\max} \cos\theta \; d\theta \tag{7.30}$$

$$= 2f_{\max}^2 \int_{-\pi/2}^{\pi/2} \cos^2\theta e^{\imath 2\pi f_{\max} n_1 \sin\theta} d\theta \tag{7.31}$$

$$= \frac{2\pi f_{\max} J_1(2\pi f_{\max} n_1)}{n_1} \tag{7.32}$$

where J_1 is the first-order Bessel function. Thus

$$\phi(n_1, n_2) = \frac{2\pi f_{\max} J_1(2\pi f_{\max}(n_1^2 + n_2^2)^{1/2})}{(n_1^2 + n_2^2)^{1/2}} \ . \tag{7.33}$$

Before proceeding further with the topic of 2-d filtering, we must define a 2-d DFT. The utility of the multidimensional DFT arises for the same reasons as for 1-d series; it enables us to deal with limited time series (with the added implication that our sampled signals are now periodic), and it is implementable with highly efficient FFT routines.

A periodic signal in two dimensions satisfies

$$x(n_1, n_2) = x(n_1 + m_1 N_1, n_2 + m_2 N_2) \tag{7.34}$$

where (N_1, N_2) are the periods of the 2-d signal (in samples) along the two grid axes and

$$(m_1, m_2) \text{ integers} \tag{7.35}$$

As in one dimension, such 2-d signals can be decomposed into a linear combination of a finite number of exponential basis functions which have periods which are submultiples of (N_1, N_2). Thus,

$$x(n_1, n_2) = \frac{1}{N_1 N_2} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} X(k_1, k_2) e^{\imath 2\pi n_1 k_1/N_1} e^{\imath 2\pi n_2 k_2/N_2} \tag{7.36}$$

where $X(k_1, k_2)$ is the 2-d DFT of $x(n_1, n_2)$. The corresponding DFT is therefore

$$X(k_1, k_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(n_1, n_2) e^{-\imath 2\pi n_1 k_1/N_1} e^{-\imath 2\pi n_2 k_2/N_2}. \tag{7.37}$$

Noe that we could also define a 2-d z transform

$$x(z_1, z_2) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} x(n_1, n_2) z_1^{-n_1} z_2^{-n_2} \tag{7.38}$$

and a corresponding inverse z transform (with contours c_1 and c_2)

$$x(n_1, n_2) = \frac{1}{(\imath 2\pi)^2} \int_{c_1} \int_{c_2} X(z_1, z_2) z_1^{n_1 - 1} z_2^{n_2 - 1} \ dz_1 \ dz_2 \ . \tag{7.39}$$

A general 2-d digital filter is thus characterizable by a difference equation

$$y(n_1, n_2) = \sum_{i=-p}^{p} \sum_{j=-q}^{q} \alpha_{ij} x(n_1 - i, n_2 - j) - \sum_{i=-r}^{r} \sum_{j=-s}^{s} \beta_{ij} y(n_1 - i, n_2 - j) \tag{7.40}$$

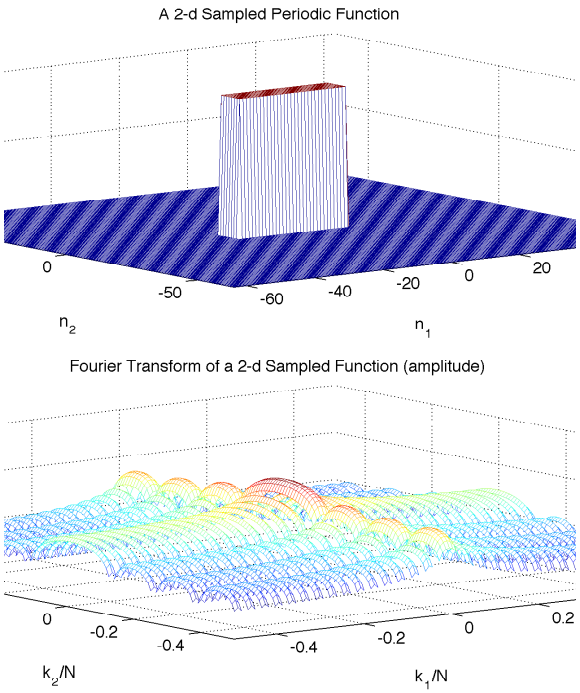

Figure 7.1: A 2-dimensional sampled function and its DFT

where i and j are not both zero in the second summation, and we have made the constant coefficients symmetric about $y(0,0)$. This has a z transform given by

$$Y(z_1, z_2) = \frac{\sum_{i=-p}^{p} \sum_{j=-q}^{q} \beta_{ij} z_1^{-i} z_2^{-j}}{\sum_{i=-r}^{r} \sum_{j=-s}^{s} \alpha_{ij} z_1^{-i} z_2^{-j}} \tag{7.41}$$

where $\alpha_{00} = 1$ has poles and zeros in a 4-dimensional space defined by the real and imaginary parts of z_1 and z_2. Evaluating stability for such filters is difficult, primarily because one cannot, in general, factor the 2-dimensional numerator and denominator to obtain a simple view of the zero and pole frequencies. As a result of this property (or non-property) of higher-dimensional polynomials, the cascade of two stable IIR filters may not even be stable! (this issue is still a current research topic). Because of these difficulties, we will primarily concern ourselves with FIR higher-dimensional filters here (this is the case where there are no poles and thus no potential stability problems).

The two dimensional function shown in Figure (7.1) could be applied as an FIR filter to effect low-pass filtering by the use of the convolution theorem (direct manipulation of the DFT) or via convolution with a corresponding kernel in two dimensions. However, this filter is anisotropic in the (k_1, k_2) plane, in the sense the wavenumber components along the diagonals will experience different filtering than along the k_1 or k_2 directions, and the filtering in k_1 and k_2 directions has different cutoff wavenumbers. Consider, instead, a circularly symmetric, low-pass filter case defined by the ideal response (Figure 7.2), which has the transfer function

$$\Phi(f_1, f_2) = \begin{cases} 1 & f_1^2 + f_2^2 \leq 1/4 \\ 0 & \text{otherwise} \end{cases} \tag{7.42}$$

from (7.33), we know that the corresponding filter weights are given by

$$w(n_1, n_2) = \begin{cases} \pi f_{\max} J_1((\pi/2)(n_1^2 + n_2^2)^{1/2}) \\ 2(n_1^2 + n_2^2)^{1/2} \end{cases} \tag{7.43}$$

Taking a N by N -point rectangular window (a simple truncation of the 2-d series) produces a filter with a frequency response

$$W(f_1, f_2) = \sum_{n_1=-(N-1)/2}^{(N-1)/2} \sum_{n_2=-(N-1)/2}^{(N-1)/2} w(n_1, n_2) e^{-\imath 2\pi n_1 f_1} e^{-\imath 2\pi n_2 f_2} \tag{7.44}$$

where (f_1, f_2) is normalized to the Nyquist interval, so that both frequencies span $(-1/2, 1/2)$.

As in the 1-d case, we can improve the ripple features of the filtering by applying a windowing function with better spectral leakage characteristics than the 2-dimensional rectangular window implied by simply convolving with a truncated $w(n_1, n_2)$. As we usually wish our window to be circularly symmetric in

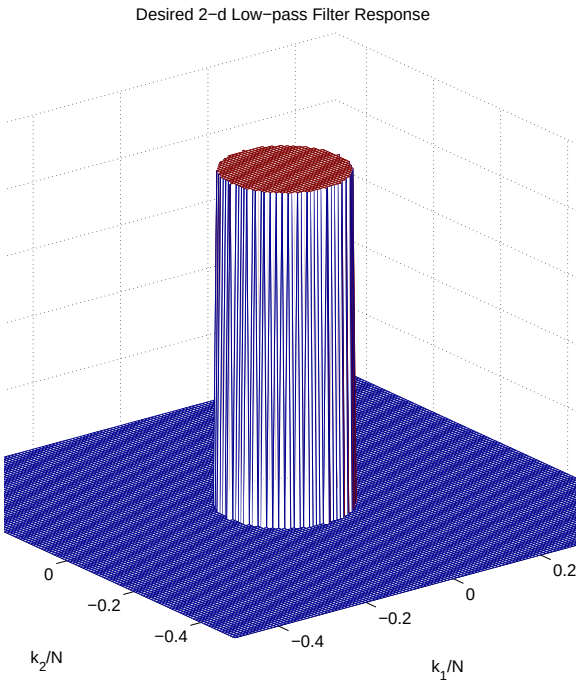

Figure 7.2: A 2-dimensional ideal lowpass filter response.

the (f_1, f_2) and (n_1, n_2) planes, we can take a window function, $\hat{w}$, from 1-dimensional analysis and substitute the radius in (n_1, n_2)-space for n to obtain a circularly symmetric 2-dimensional window

$$w(n_1, n_2) = \hat{w}(n_1^2 + n_2^2)^{1/2} \ . \tag{7.45}$$

As in 1-d processing, the Kaiser-Bessel window is a good candidate for a windowing function due to its low spectral leakage. An N by N, 2-d Kaiser-Bessel window is

$$w(n_1, n_2) = \frac{I_0\left[2\pi\sqrt{1 - (n_1^2 + n_2^2)/N^2)}\right]}{I_0(2\pi)} \tag{7.46}$$

for $n_1^2 + n_2^2 \leq N^2$ and

$$w(n_1, n_2) = 0 \tag{7.47}$$

for $n_1^2 + n_2^2 > N^2$, where $I_0(x)$ is the modified Bessel function of the first kind and 0^{th} order. The response of the Kaiser-Bessel windowed low pass filter is superior in smoothness and in attenuation (reduction of spectral leakage) to the rectangular window, as shown in the plots on the following pages.

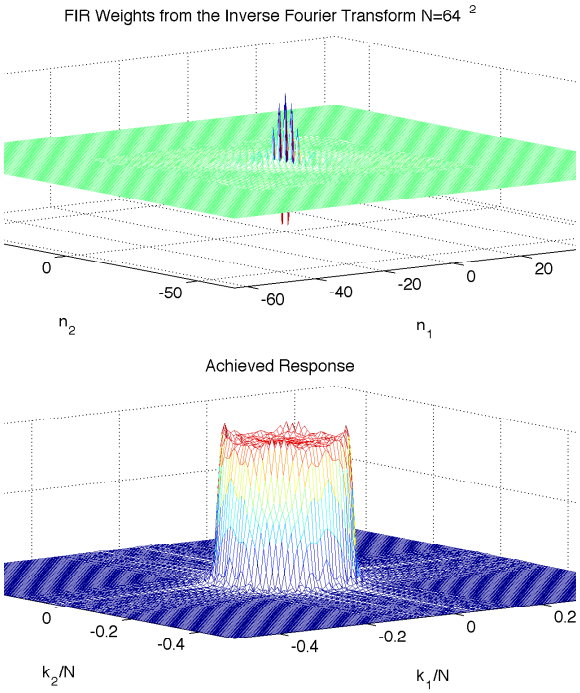

Figure 7.3: A 64 by 64 truncated FIR realization of the ideal low pass filter response.

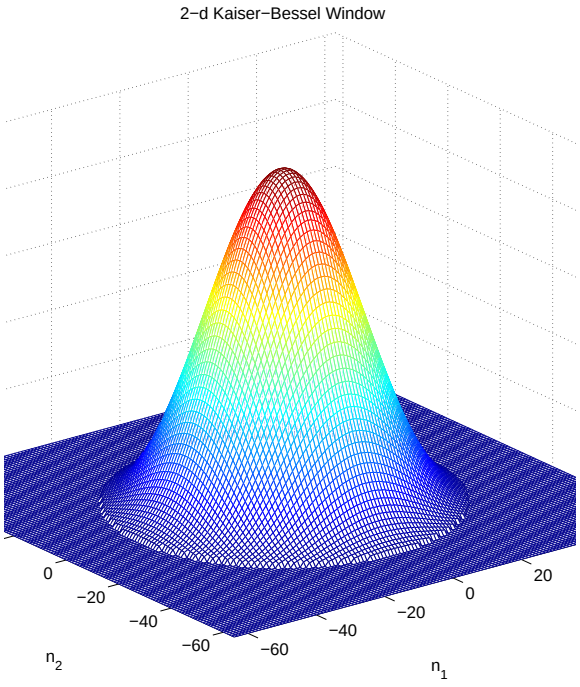

Figure 7.4: A Kaiser Bessel window in 2 dimensions.

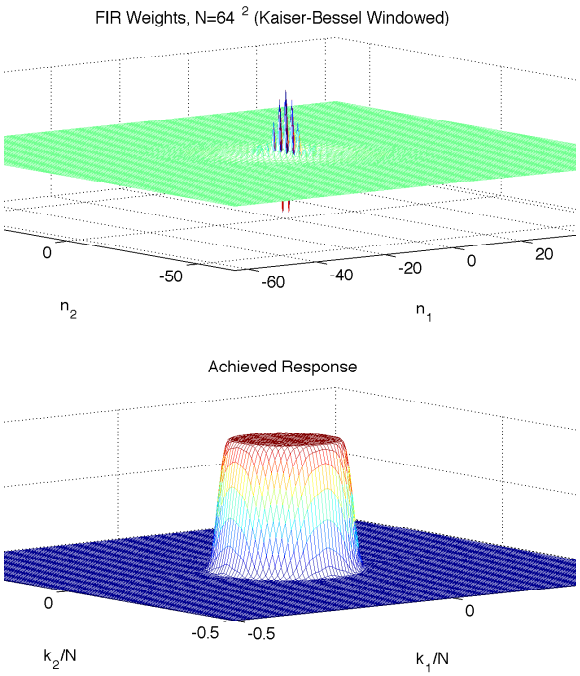

Figure 7.5: A 64 by 64 Kaiser Bessel-windowed FIR realization of the ideal low pass filter response.

Frequency-Wavenumber Filtering

We next consider some aspects of filtering in a two-dimensional system where the two-dimensions do not have the same units. Consider a linear array of seismometers or antennae deployed in the $\hat{x}$ direction with a constant spacing. Signals from such an array can be displayed in a 2-dimensional *record section*, where we have t as the ordinate and channel number, or x, as the abscissa (or vice-versa). The response of such a system to a traveling, sinusoidal plane wave of frequency f_0

$$\phi(t,x) = e^{\imath 2\pi f_0(t-x/v_0)} \tag{7.48}$$

where v_0 is the *apparent phase velocity* of the wave across the array, is of particular interest, as such signals impinge upon the array at specific angles given by

$$\theta = \sin^{-1}(c/v_0) \tag{7.49}$$

where c is the true wave velocity in the medium and θ is the angle between the planar wavefront and the $\hat{x}$ direction. Thus, when $\theta = 0$, the apparent phase velocity $v_0 = \infty$, as the wavefront strikes all of the sensors simultaneously. Conversely, when $\theta = 90$, $v_0 = c$, as the plane wave is propagating directly along the array axis (in the $\hat{x}$ direction).

If we arrange the data in (t,x)-space to form a 2-dimensional array (practically speaking, we may have to resample the traces to form an evenly-spaced array in the sampled case), we can take a 2-d Fourier transform of (7.48) as

$$\Phi(f,k) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \phi(t,x)e^{-\imath 2\pi ft}e^{\imath 2\pi xf/v}\,dt\,dx = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \phi(t,x)e^{-\imath 2\pi ft}e^{\imath 2\pi kx}\,dt\,dx \tag{7.50}$$

where the *wavenumber* (or *spatial frequency*) is, here, defined as the reciprocal length

$$k = 1/\lambda = f/v\ . \tag{7.51}$$

The $f-k$ transform of the plane wave evaluated using (7.48)) is thus

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{\imath 2\pi f_0 t}e^{-\imath 2\pi xk_0}e^{-\imath 2\pi ft}e^{\imath 2\pi xk}\,dt\,dx = \delta(f-f_0,k-k_0) \tag{7.52}$$

so that every traveling sinusoidal wave of a given frequency and wavenumber in (x,t)-space maps to a delta function in (f,k)-space!

Note that we have chosen a mixed exponential sign convention for the $f-k$ transform, where the frequency portion has a minus sign in the exponent, consistent with our previous convention for 1- and 2-dimensional transforms, but the wavenumber transform exponent has a plus sign. We do this so that waves propagating towards increasing x for increasing t (like 7.48) will map into the first quadrant of the $f-k$ plane. Of course there are three other conventions of exponent signs which could be chosen here.

In $f-k$ space, arbitrary signals of a given apparent phase velocity, v_0 are specified by (7.51), so that such signals lie along lines which intersect the $f-k$

origin and have slopes of v_0 in an f vs. k presentation. Now suppose that we wish to selectively resolve waves within a range of apparent velocities. This procedure is called *beam forming*, as it was first developed in radar and radio transmission applications. In seismological applications, because of Snell's law, the horizontal phase velocity of a signal remains constant throughout a given ray path in a horizontally homogeneous medium. Thus, beam forming using seismic array data selectively examines waves which turn within a particular depth range (as our array is generally deployed horizontally). For a simple 1-d array of sensors we can preferentially extract signals with a specific phase velocity above some cutoff value, v_0, by using a filter with an $f-k$ response given by

$$Y(f,k) = \begin{cases} 1 & -|f|/v_0 \le k \le |f|/v_0 \\ 0 & \text{otherwise} \end{cases} \tag{7.53}$$

It's instructive to examine the impulse response of (e.g., [9]), given by the inverse $f-k$ transform

$$y(t,x) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} Y(f,k) e^{\imath 2\pi f t} e^{-\imath 2\pi k x}\, dk\, df\ . \tag{7.54}$$

Of course, in practical situations, x and t are both discrete variables, so that, for unit time sampling interval, $\Delta t = 1$ and unit spatial sampling interval, $\Delta x = 1$

$$y(n\Delta t,(m{+}1/2)\Delta x) = y(n,m{+}1/2) = \int_{-1/2}^{1/2}\int_{-1/2}^{1/2} Y(f,k) e^{\imath 2\pi f n} e^{-\imath 2\pi k(m+1/2)}\, dk\, df \tag{7.55}$$

where we have assumed that there are an even number of receivers in the array, so that the half-integer spatial index, $m+1/2$ gives a symmetric deployment relative to the x origin. Evaluating the integral over k for $Y(f,k)$ gives

$$\int_{-1/2}^{1/2} e^{\imath 2\pi f n} \left(\frac{e^{-\imath 2\pi k(m+1/2)}}{-\imath 2\pi(m+1/2)} \right)\Bigg|_{k=-|f|/v_0}^{|f|/v_0} df \tag{7.56}$$

$$= \frac{1}{\pi(m+1/2)} \int_{-1/2}^{1/2} e^{\imath 2\pi f n} \sin(2\pi(m+1/2)|f|/v_0)\ df \tag{7.57}$$

$$= \frac{2}{\pi(m+1/2)} \int_{0}^{1/2} \cos 2\pi f n \sin(2\pi(m+1/2)f/v_0)\ df \tag{7.58}$$

using unity apparent velocity as the cutoff value for the sake of illustration gives

$$v_0 = \Delta x/\Delta t = 1 \tag{7.59}$$

so that

$$y(n,m+1/2) = \frac{2}{\pi(m+1/2)} \int_{0}^{1/2} \sin(2\pi f(m+1/2)) \cos(2\pi f n)\ df\ . \tag{7.60}$$

Because, for $m^2 \neq n^2$,

$$\int \sin(mx)\cos(nx)dx = \frac{-\cos(m-n)x}{2(m-n)} - \frac{\cos(m+n)x}{2(m+n)} + C \tag{7.61}$$

we have

$$y(n, m+1/2) = \frac{2}{\pi(m+1/2)} \left(\frac{-\cos(2\pi f(n+m+1/2))}{4\pi(n+m+1/2)} - \frac{\cos(2\pi f(-n+m+1/2))}{4\pi(-n+m+1/2)} \right) \Bigg|_0^{1/2} \tag{7.62}$$

$$= \frac{2}{\pi(m+1/2)} \times \tag{7.63}$$

$$\left(\frac{-\cos(\pi(n+m+1/2))}{4\pi(n+m+1/2)} - \frac{\cos(\pi(-n+m+1/2))}{4\pi(-n+m+1/2)} + \frac{1}{4\pi(n+m+1/2)} + \frac{1}{4\pi(-n+m+1/2)} \right) . \tag{7.64}$$

As m and n are integers, the cosine terms are zero, so that

$$y(n, m+1/2) = \frac{1}{2\pi^2(m+1/2)} \left(\frac{1}{(n+m+1/2)} + \frac{1}{(-n+m+1/2)} \right) \tag{7.65}$$

or

$$y(n, m+1/2) = \frac{1}{\pi^2 \left[(m+1/2)^2 - n^2 \right]} . \tag{7.66}$$

As is usual in FIR filter design problems, the weights are nonzero for large indices (n and m) and we are forced into a truncation procedure to produce a finite set of filter weights. As in our previous examples, the Kaiser Bessel window provides a good choice for truncating the 2-d weights. Rectangular and Kaiser-Bessel windowed realizations of the velocity filter (7.66) for 64 channels of 64 sample data are shown on the following page.

As the 3-d perspective plots make it difficult to see the $x-t$ domain impulse response, we also show a plot of the impulse response traces for 16 traces of 64 samples. Each time series in the impulse response consists of a simple convolving kernel. The response of the filter, $r(n, m+1/2)$ to an arbitrary input, $\phi(n, m+1/2)$, is thus given by the 2-d convolution of (7.66) with the input traces

$$r(n, m+1/2) = \sum_{i=1}^{N} \sum_{j=-M/2}^{M/2-1} \phi(i, j+1/2) y(n-i, m+1/2-j) \tag{7.67}$$

$r(n, 1/2)$ is thus obtainable by convolving each time series in the input with the corresponding time series in the impulse response (7.66), followed by a summation (stack) of the resultant M convolutions all m.

A particularly simple $f-k$ filter has weights given by

$$y(n, m) = \delta(n = 0) \tag{7.68}$$

The $m = 0$ output of such a filter is just a zero-lag stack of the input traces. The $f-k$ impulse response of such a system is just the Discrete Fourier transform of $y(n, m)$

$$Y(\nu, \mu) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \delta(n = 0) e^{-\imath 2\pi \nu n/N} e^{\imath 2\pi \mu m/M} \tag{7.69}$$

where our frequency-wavenumber indices are the integers (ν, μ).

$$= \sum_{m=0}^{M-1} e^{\imath 2\pi \mu m/M} = \frac{1 - e^{\imath 2\pi \mu}}{1 - e^{\imath 2\pi \mu/M}} \tag{7.70}$$

which has the amplitude response given by the Dirichlet kernel

$$|Y(\nu, \mu)| = \frac{\sin(\pi \mu)}{\sin(\pi \mu/M)} \tag{7.71}$$

which is independent of the Nyquist-normalized frequency, ν. The $t-x$ and $f-k$ plots are shown on the following page. The zero-lag stack, then, acts like a low pass filter in k and a high pass filter in v, so that waves with large k (short wavelengths) and low v (less vertical ray paths) will be attenuated, while those with small k (long wavelengths) will be relatively unaffected.

Consider now what happens if we stack the time series with some time lag, Δ, imposed between the channels, so that the impulse response is now

$$y(n, m) = \delta(n + \Delta m). \tag{7.72}$$

Such a system is called a *phased array* and has many applications in geophysics, optics, and electromagnetics (e.g., RADAR). The $f-k$ response then becomes

$$Y(\nu, \mu) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \delta(n + \Delta m) e^{-\imath 2\pi \nu n/N} e^{\imath 2\pi \mu m/n} \tag{7.73}$$

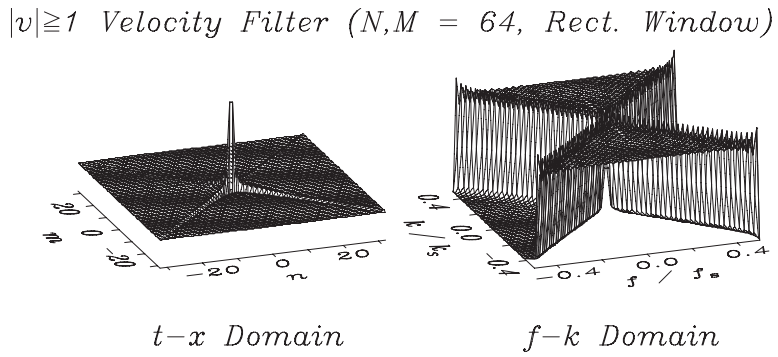

Figure 7.6: A rectangular-widowed velocity filter.

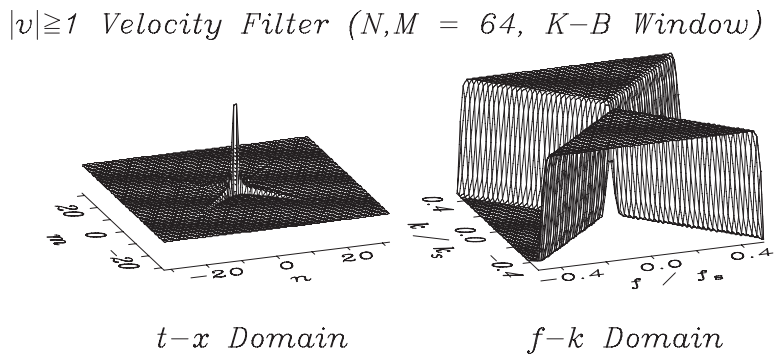

Figure 7.7: A Kaiser-Bessel-windowed velocity filter.

$$= \sum_{m=0}^{M-1} e^{\imath 2\pi\nu\Delta m/N} e^{\imath 2\pi\mu m/M} \tag{7.74}$$

for the symmetric case $N = M$, we have

$$= \sum_{m=0}^{M-1} e^{\imath 2\pi m(\nu\Delta+\mu)/N} \tag{7.75}$$

which gives the amplitude response

$$|Y(\nu,\mu)| = \frac{\sin(\pi(\nu\Delta+\mu))}{\sin(\pi(\nu\Delta+\mu)/M)} \tag{7.76}$$

which is shown on the following page for $\Delta = 1$, along with the response of the unlagged series. Rotating the impulse response in the $t-x$ domain has thus simply rotated the Fourier Transform by the same angle (in this case, 45°). We now know how to modify the velocity filter to enclose some other hourglass-shaped swath of the $f-k$ plane – we simply must impose a linear lag between the initial time traces to rotate the response function to the desired angle.

An important application of phased arrays is to receive or transmit narrow frequency band energy preferentially from a small range of azimuths. Consider a linear hydrophone array trailed from a ship with an array element spacing of $\Delta x = 30$ m and a length of 3600 m ($M = 121$ elements in all). If such an array is receiving energy from a narrow-band source (so that we are only interested in a small range of frequencies), we can calculate the width of the main lobe of the Dirichlet kernel response if we know the sound speed (about 1500 m/s in water).

For a $f_1 = 50$ Hz source, the wavelength is thus about 30 m. The $f-k$ response of the streamer for stacked traces is

$$|Y(\nu,\mu)| = \frac{\sin(\pi\mu)}{\sin(\pi\mu/M)} \tag{7.77}$$

where we can convert a general discrete $f-k$ transform to a function of Nyquist-normalized wavenumber, k, and Nyquist-normalized frequency, f, using the transformations

$$\mu = Mk/k_s \tag{7.78}$$

$$\nu = Nf/f_s \tag{7.79}$$

where k_s is the spatial sampling frequency

$$k_s = 1/(\Delta x) = 1/30 \text{ m}^{-1} \tag{7.80}$$

and f_s is the time sampling frequency to obtain

$$|Y(f,k)| = \frac{\sin(M\pi k/k_s)}{\sin(\pi k/k_s)} \ . \tag{7.81}$$

The first zero of this function occurs at $k = k_1$, defined by

$$\sin(M\pi k_1/k_s) = 0 \ (k_1 \neq 0) \tag{7.82}$$

or where

$$k_1 = k_s/M \approx 2.75 \times 10^{-4} \text{ m}^{-1} \tag{7.83}$$

which occurs at a plane wave emergence angle of

$$\theta = \sin^{-1}(c/v_1) = \sin^{-1}(ck_1/f_1) = \sin^{-1}(1500 \cdot 2.75 \times 10^{-4}/50) \approx 0.47^\circ \tag{7.84}$$

(corresponding to a phase lag of 2π between the first and last hydrophones) so that the total width of the main lobe is $\pm\theta$, or about 1°. The second major maximum occurs when the contributions of the plane wave are again in phase at all of the receivers, where $k = k_s$ and

$$\theta = \sin^{-1}(ck_s/f_1) = \sin^{-1}(1500/(30 \cdot 50)) = 90^\circ. \tag{7.85}$$

If frequency is doubled to $f_2 = 100$ Hz, then the wavelength is halved, and the main lobe becomes narrower, with the first zero now occurring at

$$\theta = \sin^{-1}(ck_1/f_2) = \sin^{-1}(1500 \cdot 2.75 \times 10^{-4}/100) \approx 0.24^\circ \ . \tag{7.86}$$

The second major maximum now occurs at only

$$\theta = \sin^{-1}(ck_s/f_2) = \sin^{-1}(1500 \cdot 2/(60 \cdot 100)) = 30^\circ. \tag{7.87}$$

so that the main beam has become narrower, but we now have a second maximum to contend with at 30° from normal incidence.

Frequency-Wavenumber Filtering with 2-dimensional arrays

Next, we consider data from a 2-dimensional array of instruments. Again, we can decompose incident energy into a superposition of traveling waves, but we now have an additional spatial dimension to contend with because our signals now have two spatial dimensions.

A particular wave field sampled by a two-dimensional array can be decomposed into plane waves

$$\phi(t,x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \Phi(f,k_x,k_y)e^{\imath 2\pi ft}e^{-\imath 2\pi k_x x}e^{-\imath 2\pi k_y y}\ df\ dk_x\ dk_y \tag{7.88}$$

where k_x and k_y are the wavenumbers in the x and y directions and $\Phi(k_x,k_y,f)$ is a 3-dimensional frequency-wavenumber spectrum. A particular plane wave propagates at an azimuth, ϕ, specified by

$$\phi = \tan^{-1}(k_y/k_x) \tag{7.89}$$

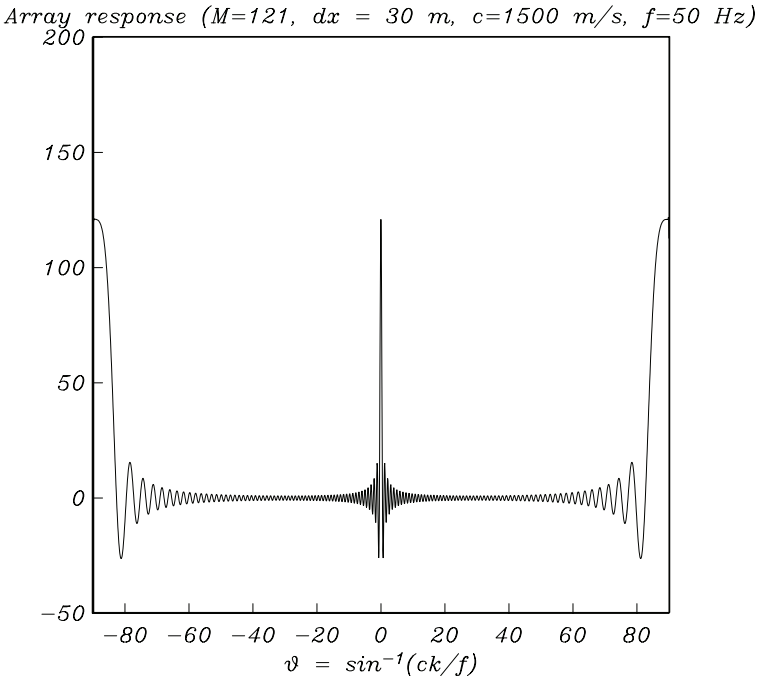

Figure 7.8: A linear array response as a function of incident angle.

k_x and k_y are thus not independent, but are related by the Pythagorean theorem

$$k_x^2 + k_y^2 = f^2/v^2. \tag{7.90}$$

The $f-k$ spectrum of a 2-dimensional time signal is thus

$$\Phi(f, k_x, k_y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \phi(t,x,y) e^{-\imath 2\pi f t} e^{\imath 2\pi k_x x} e^{\imath 2\pi k_y y} dt dx dy \tag{7.91}$$

and its discrete counterpart is

$$\Phi(\nu, \mu_x, \mu_y) = \sum_{n=0}^{N-1}\sum_{l=0}^{L-1}\sum_{m=0}^{M-1} \phi(l,n,m) e^{-\imath 2\pi n\nu/N} e^{\imath 2\pi l\mu_x/L} e^{\imath 2\pi m\mu_y/M} \ . \tag{7.92}$$

As in the case of an ideal 1-dimensional array, we can (theoretically at least) calculate a frequency-wavenumber spectrum from real data using (7.92) to determine the nature of the incident energy in terms of a plane wave decomposition. Unfortunately, this is not usually the case in seismology, particularly at high frequencies, as spatial heterogeneity induces scattering which fragments the wavefront near the array, reducing the signal coherence from sensor to sensor. One can improve the situation somewhat by introducing station corrections (e.g., [1]), so that the wavefront is best reconstructed (this procedure is analogous to the adaptive optical techniques used in modern large telescopes).

Upward and Downward Continuation of Remotely Sensed Data

Consider a point mass, m, located at the origin, which produces a gravitational field

$$\vec{g}(\hat{r}) = \frac{-mG\hat{r}}{r^2} = \frac{-mG(x\hat{x} + y\hat{y} + z\hat{z})}{(x^2+y^2+z^2)^{3/2}} \tag{7.93}$$

where G is Newton's gravitational constant. At a general position (x, y, z), the vertical (z component) of the gravity field will thus be

$$g_z = \hat{z}\cdot\vec{g} = \frac{-mGz}{(x^2+y^2+z^2)^{3/2}} \tag{7.94}$$

The integral of g_z over the xy plane is

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g_z \ dx \ dy = -mGz \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x^2+y^2+z^2)^{-3/2} \ dx \ dy \tag{7.95}$$

$$= -2\pi mGz \int_0^{\infty} \frac{r \ dr}{(z^2+r^2)^{3/2}} \tag{7.96}$$

$$= -2\pi mGz \left(\frac{-1}{(z^2+r^2)^{1/2}}\right)\Bigg|_0^{\infty} = -2\pi mG \tag{7.97}$$

which, interestingly, does not depend on z. If we take the output of our system to be the vertical field at $z = 0$, then we clearly have a delta function at the origin with a magnitude given by (7.94), as the field has no vertical component except exactly at the origin. Next consider a surface at a height h above the xy plane. The vertical field there is just

$$g_z(h) = -\frac{h}{2\pi(x^2 + y^2 + h^2)^{3/2}} = -\frac{h}{2\pi(r^2 + h^2)^{3/2}} \tag{7.98}$$

where we have normalized the response by (7.94). As field quantities obey superposition and linearity, vertical field measurements of a general field obtained at an arbitrary height $z = h$ are thus specified by the 2-dimensional convolution of (7.98) with the field at $z = 0$.

We can examine the frequency response of this filter by taking the Fourier transform of (7.98)

$$g(k_x, k_y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{h}{2\pi(x^2 + y^2 + h^2)^{3/2}} e^{-\imath 2\pi k_x x} e^{-\imath 2\pi k_y y}\, dx\, dy \tag{7.99}$$

which can be solved to obtain

$$g(k_x, k_y) = e^{-2\pi h(k_x^2 + k_y^2)^{1/2}} \tag{7.100}$$

which is the frequency response of the upward continuation filter. Note that (7.100) is thus a low pass filter – as we move away from the ($z = 0$) plane, we loose the high frequencies in our survey. Conversely, if we wish to extrapolate downwards to the earth's surface, we need to implement the (unstable) inverse filter, $g^{-1}(k_x, k_y)$. This 2-dimensional deconvolution can be achieved in a stable way by a regularized (e.g., 2-d water level) deconvolution in the frequency domain.

Multi-dimensional filtering in MATLAB

Basic filtering operations can be done with the functions *filter2* and *conv2*. There are also 2-dimensional DFT operations (*fft2* and *ifft2*), as well as a routine (*fftn* and *ifftn*) for arbitrary dimensionality. 2-dimensional FIR filter design programs are also available using the windowing and frequency sampling methods (*fwind1/fwind2*, *fsamp2*, respectively) in the image processing toolbox. This toolbox also has two-dimensional functions (*fspecial*) and many, many other useful functions for operating on 2-dimensional arrays.

Chapter 8

Notes on Random Processes

A Brief Review of Probability

In this section of the course, we will work with random variables which are denoted by capital letters, and which we will characterize by their **probability density functions** (pdf) and **cumulative density functions** (CDF.) We will use the notation $f_X(x)$ for the pdf and $F_X(a)$ for the CDF of X. Here, the subscript X tells us which random variable's pdf or CDF we're working with. The relation between the pdf and CDF is

$$P(X \leq a) = F_X(a) = \int_{-\infty}^{a} f_X(x)dx. \tag{8.1}$$

Since probabilities are always between 0 and 1, the limit as a goes to negative infinity of $F(a)$ is 0, and the limit as a goes to positive infinity of $F(a)$ is 1. Also, $\int_{-\infty}^{\infty} f(x)dx = 1$. By the fundamental theorem of calculus, $F'(a) = f(a)$.

The most important distribution that we'll work with is the **normal distribution**.

$$P(X \leq a) = \int_{-\infty}^{a} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2} dx. \tag{8.2}$$

Unfortunately, there's no simple formula for this integral. Instead, tables or numerical approximation routines are used to evaluate it. The normal distribution has a characteristic bell shaped pdf. The center of the bell is at $x = \mu$, and the parameter σ^2 controls the width of the bell. The particular case in which $\mu = 0$, and $\sigma^2 = 1$ is referred to as the **standard normal random variable**. The letter Z is typically used for the standard normal random variable. Figure 8.1 shows the pdf of the standard normal.

The **expected value** of a random variable X is

$$\mu_X = E[X] = \int_{-\infty}^{\infty} x f_X(x)dx. \tag{8.3}$$

Note that this integral does not always converge!

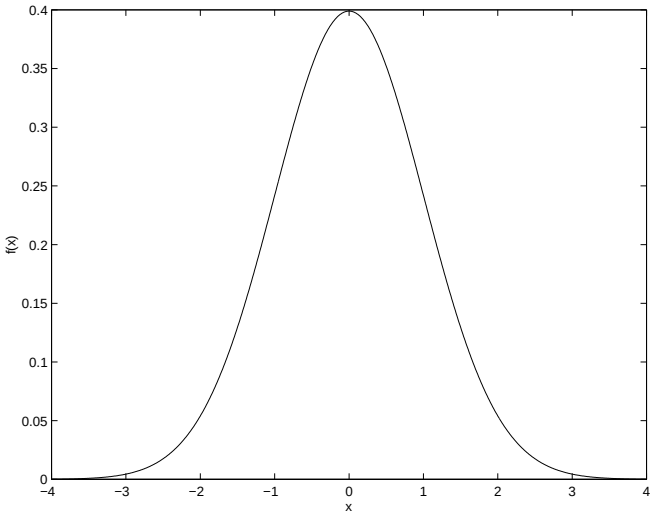

Figure 8.1: The standard normal pdf.

For a normal random variable, it turns out (after a bit of work to evaluate the integral) that $E[X] = \mu$.

We'll often work with random variables that are functions of other random variables. If X is a random variable with pdf $f_X(x)$ and $g()$ is a function, then $g(X)$ is also a random variable, and

$$E[g(X)] = \int_{-\infty}^{\infty} g(x) f_X(x) dx. \tag{8.4}$$

Because integration is a linear operator,

$$E[X + Y] = E[X] + E[Y] \tag{8.5}$$

and

$$E[sX] = sE[X]. \tag{8.6}$$

The **variance** of a random variable X is

$$Var(X) = E[(X - E[X])^2] \tag{8.7}$$

$$Var(X) = E[X^2 - 2XE[X] + E[X]^2] \tag{8.8}$$

Using the linearity of $E[]$ and the fact that the expected value of a constant is the constant, we get that

$$Var(X) = E[X^2] - 2E[X]E[X] + E[X]^2 \tag{8.9}$$

$$Var(X) = E[X^2] - E[X]^2. \tag{8.10}$$

For a normal random variable, it's relatively easy to show that $Var(X) = \sigma^2$.

If we have two random variables X and Y, they *may* have a **joint probability density** $f(x, y)$ with

$$P(X \leq a \text{ and } Y \leq b) = \int_{-\infty}^{a} \int_{-\infty}^{b} f(x, y)\, dy\, dx \tag{8.11}$$

Two random variables X and Y are **independent** if they have a joint density and

$$f(x, y) = f_X(x) f_Y(y). \tag{8.12}$$

If X and Y have a joint density, then the **covariance** of X and Y is

$$Cov(X, Y) = E[(X - E[X])(Y - E[Y])] = E[XY] - E[X]E[Y]. \tag{8.13}$$

It turns out that if X and Y are independent, then $E[XY] = E[X]E[Y]$, and $Cov(X, Y) = 0$. However, there are examples where X and Y are dependent, but $Cov(X, Y) = 0$. If $Cov(X, Y) = 0$, then we say that X and Y are **uncorrelated**.

The **correlation** of X and Y is

$$\rho_{XY} = \frac{Cov(X, Y)}{\sqrt{Var(X)Var(Y)}}. \tag{8.14}$$

The correlation is a sort of scaled version of the covariance that we will make frequent use of.

Some important properties of Var, Cov and correlation include:

$$Var(X) \geq 0 \tag{8.15}$$

$$Var(sX) = s^2 Var(X) \tag{8.16}$$

$$Var(X + Y) = Var(X) + Var(Y) + 2Cov(X, Y) \tag{8.17}$$

$$Cov(X, Y) = Cov(Y, X) \tag{8.18}$$

$$-1 \leq \rho_{XY} \leq 1 \tag{8.19}$$

The following example demonstrates the use of some of these properties.

Example 8.1

Suppose that Z is a standard normal random variable. Let

$$X = \mu + \sigma Z. \tag{8.20}$$

Then

$$E[X] = E[\mu] + \sigma E[Z] \tag{8.21}$$

so

$$E[X] = \mu. \tag{8.22}$$

Also,

$$Var(X) = Var(\mu) + \sigma^2 Var(Z) \tag{8.23}$$

$$Var(X) = \sigma^2. \tag{8.24}$$

Thus if we have a program to generate random numbers with the standard normal distribution, we can use it to generate random numbers with any desired normal distribution. The MATLAB command **randn** generates N(0,1) random numbers.

Suppose that X_1, X_2, ..., X_n are independent realizations of a random variable X. How can we estimate $E[X]$ and $Var(X)$?

Let

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n} \tag{8.25}$$

and

$$s^2 = \frac{\sum_{i=1}^{n} (X_i - \bar{X})^2}{n-1} \tag{8.26}$$

These estimates for $E[X]$ and $Var(X)$ are unbiased in the sense that

$$E[\bar{X}] = E[X] \tag{8.27}$$

and

$$E[s^2] = Var(X). \tag{8.28}$$

We can also estimate covariances with

$$\widehat{Cov}(X,Y) = \frac{\sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y})}{n} \tag{8.29}$$

Random Vectors

In digital signal processing we've been dealing with signals, represented in discrete time by vectors. Thus it's important to be able to work with random variables that are vectors. We'll consider random vectors of the form

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} \tag{8.30}$$

where the individual random variables X_i are assumed to have a joint probability density function.

The expected value of a random vector is

$$\mu = E[X] = \begin{bmatrix} E[X_1] \\ E[X_2] \\ \vdots \\ E[X_n] \end{bmatrix}. \tag{8.31}$$

The covariance matrix of X is

$$C = Cov(X) = E[XX^T] - E[X]E[X]^T. \tag{8.32}$$

Since

$$XX^T = \begin{bmatrix} X_1X_1 & X_1X_2 & X_1X_3 & \cdots & X_1X_n \\ X_2X_1 & X_2X_2 & X_2X_3 & \cdots & X_2X_n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ X_nX_1 & X_nX_2 & X_nX_3 & \cdots & X_nX_n \end{bmatrix}, \tag{8.33}$$

$$C_{i,j} = E[X_iX_j] - E[X_i]E[X_j] = Cov(X_i, X_j). \tag{8.34}$$

We will also work with the correlation matrix

$$P_{i,j} = \frac{Cov(X_i, X_j)}{\sqrt{Cov(X_i, X_i)}\sqrt{Cov(X_j, X_j)}}. \tag{8.35}$$

Just as with scalar random variables, the expected value and covariance of a random vector have many useful properties. In deriving these properties we have to be somewhat careful, since matrix multiplication is not commutative. Thus

$$E[AX] = AE[X], \tag{8.36}$$

but

$$E[XA] = E[X]A, \tag{8.37}$$

An analogous result to $Var(sX) = s^2Var(X)$ is that

$$Cov(AX) = E[(AX)(AX)^T] - E[AX]E[AX]^T. \tag{8.38}$$

$$Cov(AX) = E[AXX^TA^T] - AE[X]E[X]^TA^T. \tag{8.39}$$

$$Cov(AX) = AE[XX^T]A^T - AE[X]E[X]^TA^T. \tag{8.40}$$

$$Cov(AX) = A(E[XX^T] - E[X]E[X]^T)A^T. \tag{8.41}$$

$$Cov(AX) = ACov(X)A^T. \tag{8.42}$$

Recall that a symmetric matrix A is positive semidefinite (PSD) if $x^TAx \geq 0$, for all x. Also A is positive definite (PD) if $x^TAx \geq 0$, for all nonzero x.

Corresponding to the property that $Var(X) \geq 0$, we find that the covariance matrix C of a random variable is always positive semidefinite. To show this, let

$$W = \alpha_1X_1 + \ldots \alpha_nX_n = \alpha^TX. \tag{8.43}$$

Then

$$Var(W) = E[(W - E[W])(W - E[W])^T]. \tag{8.44}$$

$$Var(W) = E[(W - E[W])(W - E[W])^T]. \tag{8.45}$$

Since

$$W - E[W] = \alpha^Tx - \alpha^T\mu, \tag{8.46}$$

$$Var(W) = E[\alpha^T (x-\mu)(x-\mu)^T \alpha]. \tag{8.47}$$
$$Var(W) = \alpha^T E[(x-\mu)(x-\mu)^T]\alpha. \tag{8.48}$$
$$Var(W) = \alpha^T C \alpha. \tag{8.49}$$

But $Var(W) \geq 0$. Thus $\alpha^T C \alpha \geq 0$, for every vector α, and C is positive semidefinite.

We can estimate $E[X]$ and $Cov(X)$ from a sample of random vectors drawn from the distribution. Suppose that the columns of an n by m matrix X are m random vectors drawn from the distribution. Then we can estimate

$$E[X_j] \approx \mu = \frac{\sum_{j=1}^{n} X_{1,j}}{m} \tag{8.50}$$

or

$$E[X] \approx \mu = \frac{Xe}{m}, \tag{8.51}$$

where e is the vector of all ones. We can also estimate that

$$Cov(X) \approx C = \frac{XX^T}{m} - \mu\mu^T. \tag{8.52}$$

The Multivariate Normal (MVN) Distribution

The **multivariate normal distribution** (MVN) is an important joint probability distribution. If the random variables X_1, ..., X_n have an MVN, then the probability density is

$$f(x_1, x_2, \ldots, x_n) = \frac{1}{(2\pi)^{n/2}} \frac{1}{\sqrt{|C|}} e^{-(x-\mu)^T C^{-1} (X-\mu)/2}. \tag{8.53}$$

Here μ is a vector of the mean values of X_1, ..., X_n, and C is a matrix of covariances with

$$C_{i,j} = Cov(X_i, X_j). \tag{8.54}$$

The multivariate normal distribution is one of a very few multivariate distributions with useful properties. Notice that the vector μ and the matrix C completely characterize the distribution.

We can generate vectors of random numbers according to an MVN distribution by using the following process, which is very similar to the process for generating random normal scalars.

1. Find the Cholesky factorization $C = LL^T$.
2. Let Z be a vector of n independent N(0,1) random numbers.
3. Let $X = \mu + LZ$.

To see that X has the appropriate mean and covariance matrix, we'll compute them.

$$E[X] = E[\mu + LZ] = \mu + E[LZ] = \mu + LE[Z] = \mu. \tag{8.55}$$
$$Cov[X] = E[(X-\mu)(X-\mu)^T] = E[(LZ)(LZ)^T]. \tag{8.56}$$
$$Cov[X] = LE[ZZ^T]L^T = LIL^T = LL^T = C. \tag{8.57}$$

Covariance Stationary processes

A **discrete time stochastic process** is a sequence of random variables Z_1, Z_2, ... In practice we will typically analyze a single realization z_1, z_2, ..., z_n of the stochastic process and attempt to estimate the statistical properties of the stochastic process from the realization. We will also consider the problem of predicting z_{n+1} from the previous elements of the sequence.

We will begin by focusing on the very important class of **stationary** stochastic processes. A stochastic process is **strictly stationary** if its statistical properties are unaffected by shifting the stochastic process in time. In particular, this means that if we take a subsequence Z_{k+1}, ..., Z_{k+m}, then the joint distribution of the m random variables will be the same no matter what k is.

In practice, we're often only interested in the means and covariances of the elements of a time series. A time series is **covariance stationary**, or **second order stationary** if its mean and its autocovariances (or autocorrelations) at all lags are finite and constant. For a covariance stationary process, the **autocovariance at lag** m is $\gamma_m = Cov(Z_k, Z_{k+m})$. Since covariance is symmetric, $\gamma_{-m} = \gamma_m$. The correlation of Z_k and Z_{k+m} is the **autocorrelation at lag** m. We will use the notation ρ_m for the autocorrelation. It is easy to show that

$$\rho_k = \frac{\gamma_k}{\gamma_0}. \tag{8.58}$$

The autocovariance and autocorrelation matrices

The covariance matrix for the random variables Z_1, ..., Z_n is called an **autocovariance matrix**.

$$\Gamma_n = \begin{bmatrix} \gamma_0 & \gamma_1 & \gamma_2 & \cdots & \gamma_{n-1} \\ \gamma_1 & \gamma_0 & \gamma_1 & \cdots & \gamma_{n-2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \gamma_{n-1} & \gamma_{n-2} & \cdots & \gamma_1 & \gamma_0 \end{bmatrix} \tag{8.59}$$

Similarly, we can form an **autocorrelation matrix**

$$P_n = \begin{bmatrix} 1 & \rho_1 & \rho_2 & \cdots & \rho_{n-1} \\ \rho_1 & 1 & \rho_1 & \cdots & \rho_{n-2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \rho_{n-1} & \rho_{n-2} & \cdots & \rho_1 & 1 \end{bmatrix}. \tag{8.60}$$

Note that

$$\Gamma_n = \sigma_Z^2 P_n. \tag{8.61}$$

Since the autocovariance matrix is a covariance matrix, it is positive semidefinite. It's easy to show that the autocorrelation matrix is also positive semidefinite.

An important example of a stationary process that we will work with occurs when the joint distribution of Z_k, ..., Z_{k+n} is multivariate normal. In this situation, the autocovariance matrix Γ_n is precisely the covariance matrix C for the multivariate normal distribution.

Estimating the mean, autocovariance, and autocorrelation

Given a realization z_0, z_2, ..., z_{N-1}, of a stochastic process, how can we estimate the mean, variance, autocovariance and autocorrelation?

We will estimate the mean by

$$\bar{z} = \frac{\sum_{j=0}^{N-1} z_j}{N}. \tag{8.62}$$

We will estimate the autocovariance at lag k with

$$c_k = \frac{1}{N} \sum_{j=0}^{N-1} (z_j - \bar{z})(z_{j+k} - \bar{z}). \tag{8.63}$$

Here we have used the convention that z_k is a periodic sequence to get z_{j+k} in cases where $j + k > N - 1$.

Note that c_0 is an estimate of the variance, but it is not the same unbiased estimate that we used in the last lecture. The problem here is that the z_i are correlated, so that the formula from the last lecture no longer provides an unbiased estimator. The formula given here is also biased, but is considered to work better in practice.

We will estimate the autocorrelation at lag k with

$$r_k = \frac{c_k}{c_0}. \tag{8.64}$$

The following example demonstrates the computation of autocorrelation and autocovariance estimates.

Example 8.2

Consider the time series of yields from a batch chemical process given in Table 1. The data is plotted in Figure 8.2. These data are taken from p 31 of Box, Jenkins, and Reinsel. Read the table by rows. Figure 8.3 shows the estimated autocorrelation for this data set. The fact that r_1 is about -0.4 tells us that whenever there is a sample in the data that is well above the mean, it is likely to be followed by a sample that is well below the mean, and vice versa. Notice that the autocorrelation tends to alternate between positive and negative values and decays rapidly towards a noise level. After about $k = 6$, the autocorrelation seems to have died out.

Just as with the sample mean, the autocorrelation estimate r_k is a random quantity with its own standard deviation. It can be shown that

$$Var(r_k) \approx \frac{1}{n} \sum_{v=-\infty}^{\infty} (\rho_v^2 + \rho_{v+k}\rho_{v-k} - 4\rho_k\rho_v\rho_{v-k} + 2\rho_v^2\rho_k^2). \tag{8.65}$$

47	64	23	71	38	64	55	41	59	48
71	35	57	40	58	44	80	55	37	74
51	57	50	60	45	57	50	45	25	59
50	71	56	74	50	58	45	54	36	54
48	55	45	57	50	62	44	64	43	52
38	59	55	41	53	49	34	35	54	45
68	38	50	60	39	59	40	57	54	23

Table 8.1: An example time series.

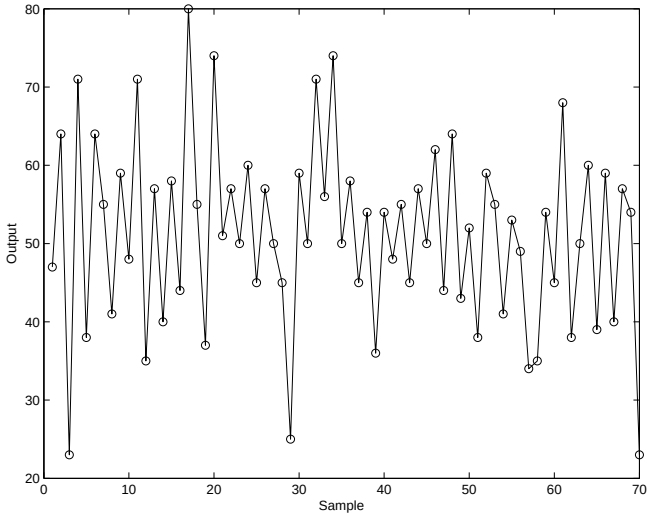

Figure 8.2: An example time series.

The autocorrelation function typically decays rapidly, so that we can identify a lag q beyond which r_k is effectively 0. Under these circumstances, the formula simplifies to

$$Var(r_k) \approx \frac{1}{n}(1 + 2\sum_{v=1}^{q} \rho_v^2), \quad k > q. \tag{8.66}$$

In practice we don't know ρ_v, but we can use the estimates r_v in the above formula. This provides a statistical test to determine whether or not an autocorrelation r_k is statistically different from 0. An approximate 95% confidence interval for r_k is $r_k \pm 1.96 * \sqrt{Var(r_k)}$. If this confidence interval includes 0, then we can't rule out the possibility that r_k really is 0 and that there is no correlation at lag k.

Example 8.3

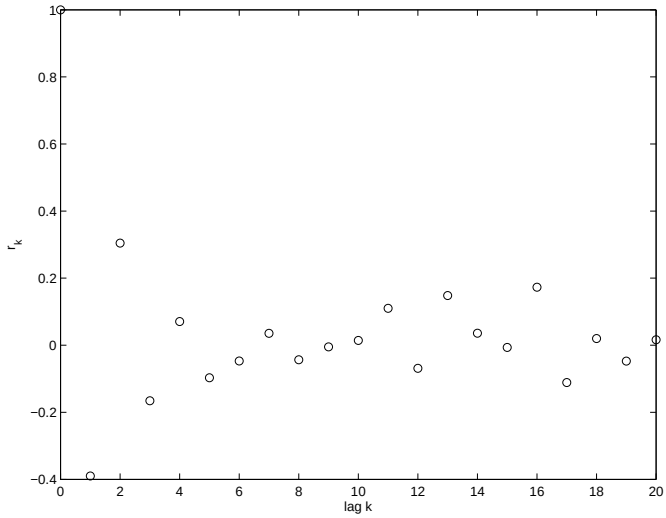

Figure 8.3: Estimated autocorrelation for the example data.

Returning to our earlier data set, consider the variance of our estimate of r_6. Using $q = 5$, we estimate that $Var(r_6) = .0225$ and that the standard deviation is about 0.14. Since $r_6 = -0.0471$ is considerably smaller than the standard deviation, we will decide to treat r_k as essentially 0 for $k \geq 6$.

The spectrum and autocorrelation

In continuous time, the spectrum of a signal $\phi(t)$ is given by

$$PSD(f) = |\Phi(f)|^2 = \Phi(f)\Phi(f)^*. \tag{8.67}$$

Since

$$\Phi(f) = \int_{t=-\infty}^{\infty} \phi(t) e^{-2\pi i f t} dt, \tag{8.68}$$

$$\Phi(f)^* = \int_{t=-\infty}^{\infty} \phi(t)^* e^{+2\pi i f t} dt. \tag{8.69}$$

Let $\tau = -t$. Then $d\tau = -dt$, and

$$\Phi(f)^* = \int_{\tau=-\infty}^{\infty} \phi(-\tau)^* e^{-2\pi i f \tau} d\tau. \tag{8.70}$$

$$\Phi(f)^* = F\left[\phi(-t)^*\right]. \tag{8.71}$$

Thus

$$PSD(f) = F\left[\phi(t)\right] F\left[\phi(-t)^*\right], \tag{8.72}$$

or by the convolution theorem,

$$PSD(f) = F\left[\phi(t) * \phi(-t)^*\right] = F\left[\text{autocorr } \phi(t)\right]. \tag{8.73}$$

We can derive a similar connection in discrete time between the periodogram and the autocovariance. Given a N-periodic sequence z_n, the autocovariance is

$$c_n = \frac{1}{N}\sum_{j=0}^{N-1}(z_j - \bar{z})(z_{j+n} - \bar{z}). \tag{8.74}$$

$$c_n = \frac{1}{N}\left(\sum_{j=0}^{N-1} z_j z_{j+n} - 2\sum_{j=0}^{N-1} z_j \bar{z} + \sum_{j=0}^{N-1} \bar{z}^2\right). \tag{8.75}$$

Since

$$\bar{z} = \frac{1}{N}\sum_{j=0}^{N-1} z_j, \tag{8.76}$$

$$c_n = \frac{1}{N}\left(\sum_{j=0}^{N-1} z_j z_{j+n} - N\bar{z}^2\right). \tag{8.77}$$

Now, we'll compute the DFT of c_n.

$$C_m = \sum_{n=0}^{N-1} c_n e^{-2\pi i n m/N}. \tag{8.78}$$

$$C_m = \sum_{n=0}^{N-1}\left(\sum_{j=0}^{N-1}\frac{z_j z_{j+n}}{N} - \bar{z}^2\right) e^{-2\pi i n m/N}. \tag{8.79}$$

By our "technical result",

$$\sum_{n=0}^{N-1} -\bar{z}^2 e^{-2\pi i n m/N} = -N\bar{z}^2\delta_m. \tag{8.80}$$

When $m = 0$, $e^{-2\pi i m n/N} = 1$, so we get

$$C_0 = \left(\sum_{n=0}^{N-1}\sum_{j=0}^{N-1}\frac{z_j z_{j+n}}{N}\right) - N\bar{z}^2. \tag{8.81}$$

Since

$$\sum_{n=0}^{N-1}\sum_{j=0}^{N-1}\frac{z_j z_{j+n}}{N} = N\bar{z}^2, \tag{8.82}$$

$$C_0 = 0. \tag{8.83}$$

Note that by definition,

$$C_0 = \sum_{n=0}^{N-1} c_n e^{-2\pi i 0n/N} = \sum_{n=0}^{N-1} c_n, \tag{8.84}$$

So $C_0 = 0$ implies that the average of the autocovariances must be 0.

When $m \neq 0$, things are more interesting. In this case,

$$C_m = \sum_{n=0}^{N-1} \sum_{j=0}^{N-1} \frac{z_j z_{j+n}}{N} e^{-2\pi inm/N}. \tag{8.85}$$

$$C_m = \frac{1}{N} \sum_{n=0}^{N-1} \sum_{j=0}^{N-1} z_j z_{j+n} e^{-2\pi inm/N}. \tag{8.86}$$

$$C_m = \frac{1}{N} \sum_{j=0}^{N-1} z_j \sum_{n=0}^{N-1} z_{j+n} e^{-2\pi inm/N}. \tag{8.87}$$

$$C_m = \frac{1}{N} \sum_{j=0}^{N-1} z_j e^{+2\pi ijm/N} \sum_{n=0}^{N-1} z_{j+n} e^{-2\pi i(j+n)m/N}. \tag{8.88}$$

Using the fact that z is real we get,

$$C_m = \frac{1}{N} \sum_{j=0}^{N-1} z_j^* e^{+2\pi ijm/N} \sum_{n=0}^{N-1} z_{j+n} e^{-2\pi i(j+n)m/N}. \tag{8.89}$$

$$C_m = \frac{1}{N} Z_m^* \sum_{n=0}^{N-1} z_{j+n} e^{-2\pi i(j+n)m/N}. \tag{8.90}$$

Using the fact that z is N-periodic, we get

$$C_m = \frac{1}{N} Z_m^* Z_m. \tag{8.91}$$

Note that because c_n is symmetric, C_m is real. Also note that the right hand side of this equation is always nonnegative. This means that $C_m \geq 0$. It turns out that $C_m \geq 0$ is equivalent to the autocovariance matrix being positive semidefinite.

Thus knowing the spectrum of z is really equivalent to knowing the autocovariance, c, or its DFT, C. In practice, the sample spectrum from a short time series is extremely noisy, so it's extremely difficult to make sense of the spectrum. On the other hand, it is much easier to make sense of the autocorrelation function of a short time series. For this reason, the autocorrelation is more often used in analyzing shorter time series.

Example 8.4

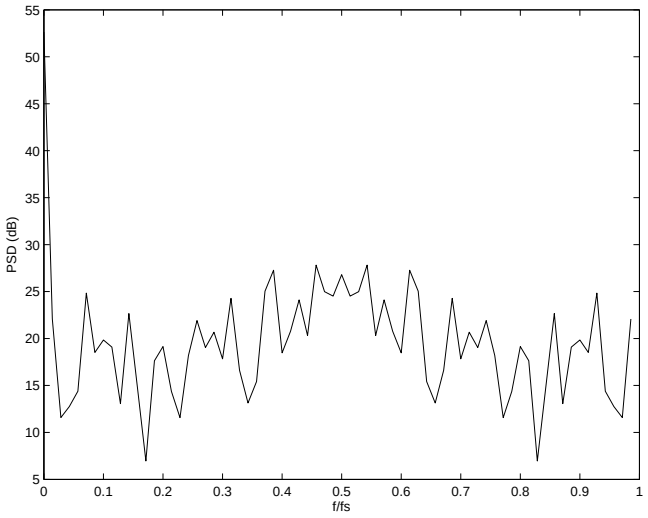

Figure 8.4: Periodogram of the sample time series.

Figure 8.4 shows the periodogram for our example data. It's very difficult to detect any real features in this spectrum. The problem is that with a short time series you get little frequency resolution, and lots of noise. Longer time series make it possible to obtain both better frequency resolution (by using a longer window) and reduced noise (by averaging over many windows.) However, if you're stuck with only a short time series, the first few autocorrelations may be more informative than the periodogram.

Generating Correlated Gaussian Noise

The connection between the autocovariance and spectrum also provides us with another way to generate random Gaussian noise with specified autocovariance. In this approach, introduced by Shinozuka and Jan [12], we start with a desired autocovariance c_n, compute the DFT C_m, and then use (8.91) to get a real, nonnegative square root, Z_m. We could simply invert this to obtain a real sequence z_n. However, this sequence wouldn't be random. Shinozuka's clever idea was to compute Z_m, and then apply random phases to each of the Z_m components, while keeping the sequence Z_m Hermitian. To do this, we multiply Z_k by $e^{\theta_k i}$, and multiply Z_{-k} by $e^{-\theta_k i}$, where θ_k is uniformly distributed between 0 and 2π. We can then invert the discrete Fourier transform to get a random sequence z_n with exactly the required autocovariances.

Note that in order to make this work, we must insure that the average of covariances c is 0 and $C_0 = 0$. If we start with a given sequence of covariances it may be necessary to additional positive or negative covariances at an extremely long lag to make the mean of c equal 0. Fortunately, we can simply extend c so that it is much longer than the desired random output sequence z, in much the same way that we used 0 padding in computing convolution by the FFT.

An important advantage of this spectral method for generating correlated Gaussian noise is that it does not require computing and storing the Cholesky factorization of the autocovariance matrix. This makes the generation of long (millions of points) sequences or 2-D or 3-D random fields computationally tractable.

Chapter 9

Kalman Filtering

Introduction

Data Assimilation is the problem of merging model predictions with measurements of a system to produce an optimal estimate of the current state of the system and/or predictions of the future state of the system. For example, weather forecasters run massive computational models that predict winds, temperature, and other physical conditions. As time progresses, it is important to incorporate available weather observations into the mathematical model. Since these weather observations are noisy, the problem of incorporating the observations into the model is inherently statistical in nature.

Data Assimilation is a important topic in many areas of science, including atmospheric physics, oceanography, and hydrology. In the next few lectures, we'll introduce Kalman filtering, which is one of the simplest approaches to data assimilation. The Kalman filter was introduced in a 1960 paper by R. E. Kalman [8].

The Model Of The System

Consider a discrete time dynamical system governed by the equation

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}. \tag{9.1}$$

Here, x_k, u_{k-1}, and w_{k-1} are vectors and the subscripts refer to the time steps rather than indexing elements of the vectors. The state of the system at time k is given by the vector x_k. Deterministic inputs to the system at time $k-1$ are given by u_{k-1}. Random noise affecting the system at time $k-1$ is given by w_{k-1}. We'll assume that w_{k-1} has a multivariate normal distribution with mean 0 and covariance matrix Q.

We'll obtain a vector of measurements z_k at time k, where z_k is given by

$$z_k = Hx_k + v_k. \tag{9.2}$$

Here v_k represents random noise in the observation z_k. We'll assume that v_k is normally distributed with mean 0 and covariance matrix R.

The matrices A, B, H, Q, and R are all assumed to be known, although the Kalman filter can be extended to simultaneously estimate these matrices along with x_k. For now, our goal is to estimate x_k and predict x_{k+1}, x_{k+2}, ..., as accurately as possible given z_1, z_2, ..., z_k.

The estimate that we will obtain will come in the form of a multivariate normal distribution with a specified mean $\hat{x}_k$ and covariance matrix $\hat{P}_k$. We will want to measure the "tightness" of this multivariate normal distribution. A convenient measure of the tightness of an MVN distribution with covariance matrix C is

$$\text{trace}(C) = C_{1,1} + C_{2,2} + \ldots + C_{n,n}. \tag{9.3}$$

$$\text{trace}(C) = Var(X_1) + Var(X_2) + \ldots + Var(X_n). \tag{9.4}$$

Example 9.5

For example, x_k might be a six element vector containing the position (3 coordinates) and velocity (3 coordinates) of an aircraft at time k. The vector u_{k-1} might represent control inputs (thrust, elevator, rudder, etc.) to the aircraft at time $k-1$, and w_{k-1} might represent the effects of turbulence on the aircraft. We may be using a very simple radar to observe the aircraft, so that we get measurements of the position, z, but not the velocity of the aircraft at each moment in time. These measurements of the aircraft's position might also be noisy.

In many cases, the system that we're interested in is described by a system of differential equations in continuous time:

$$x'(t) = Ax(t) + Bu(t). \tag{9.5}$$

We can discretize this system of equations using time steps of length Δt, to get

$$x(t + \Delta t) = x(t) + \Delta t x'(t). \tag{9.6}$$

$$x(t + \Delta t) = x(t) + \Delta t(Ax(t) + Bu(t)). \tag{9.7}$$

Letting $x_k = x(t + \Delta t)$ and $x_{k-1} = x(t)$, we get

$$x_k = (I + \Delta t A)x_{k-1} + \Delta t B u_{k-1}. \tag{9.8}$$

In many practical applications of Kalman filtering the mathematical model of the system consists of an even more complicated system of partial differential equations. Such systems are commonly discretized using finite difference or finite element methods. Rather than diving into the details of the numerical analysis used in discretizing PDE's, we will simply assume that our problem has been cast in the form of (9.1).

The Kalman Filter

We have two sources of information that can help us in estimating the state of the system at time k. First, we can use the equations that describe the dynamics of the system. Substituting $w_{k-1} = 0$ into (9.1), we might reasonably estimate

$$\hat{x}_k = Ax_{k-1} + Bu_{k-1} \tag{9.9}$$

A second useful source of information is our observation z_k. We might pick $\hat{x}_k$ so as to minimize $\|z_k - Hx_k\|$. There's an obvious trade-off between these two methods of estimating x_k. The Kalman filter produces a weighted combination of these two estimates that is optimal in the sense that it minimizes the uncertainty of the resulting estimate.

We'll begin the estimation process with an initial guess for the state of the system at time 0. Since we want to keep track of the uncertainty in our estimates, we'll have to specify the uncertainty in our initial guess. We describe this by using a multivariate normal distribution

$$x_0 \sim N(\hat{x}_0, \hat{P}_0). \tag{9.10}$$

In the *prediction* step, we are given an estimate $\hat{x}_{k-1}$ of the state of the system at time $k-1$, with associated covariance matrix $\hat{P}_{k-1}$. We substitute the mean value of $w_{k-1} = 0$ into (9.1) to obtain the estimate

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}. \tag{9.11}$$

The minus superscript is used to distinguish this estimate from the final estimate that we get after including the observation z_k. The covariance of our new estimate is

$$\hat{P}_k^- = Cov(\hat{x}_k^-). \tag{9.12}$$

$$\hat{P}_k^- = Cov(A\hat{x}_{k-1} + Bu_{k-1} + w_{k-1}). \tag{9.13}$$

The Bu_{k-1} term is not random, so its covariance is zero. The covariance of w_{k-1} is Q. The covariance of $A\hat{x}_{k-1}$ is $A\,Cov(\hat{x}_{k-1})A^T$. Thus

$$\hat{P}_k^- = ACov(\hat{x}_{k-1})A^T + Q. \tag{9.14}$$

$$\hat{P}_k^- = A\hat{P}_{k-1}A^T + Q. \tag{9.15}$$

We could simply repeat this process for x_1, x_2, ... If no observations of the system are available, that would be an appropriate way to estimate the system state.

In the *update* step, we modify the prediction estimate to include the observation.

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{9.16}$$

$$\hat{x}_k = (I - K_kH)\hat{x}_k^- + K_kz_k. \tag{9.17}$$

Here the factor K_k is called the Kalman gain. It adjusts the relative influence of z_k and $\hat{x}_k^-$. In many applications of Kalman filtering the factor K is simply set

once at the time a system is designed and it is not dynamically adjusted during the operation of the filter. In other applications the Kalman gain is dynamically adjusted to take into account the latest information on the covariance of x_{k-1}.

No matter how K_k is determined, we can produce an updated covariance matrix by applying the rule for the covariance of a matrix times an MVN vector. The new covariance is

$$\hat{P}_k = (I - K_k H)\hat{P}_k^{-1}(I - K_k H)^T + K_k R K_k^T. \tag{9.18}$$

We will next show that

$$K_k = \hat{P}_k^- H^T (H\hat{P}_k^- H^T + R)^{-1} \tag{9.19}$$

is optimal in the sense that it minimizes the trace of $\hat{P}_k$.

The covariance of our updated estimate is

$$\hat{P}_k = (I - K_k H)\hat{P}_k^-(I - K_k H)^T + K_k Cov(z_k) K_k^T. \tag{9.20}$$

Since $Cov(z_k) = R$,

$$\hat{P}_k = (I - K_k H)\hat{P}_k^-(I - K_k H)^T + K_k R K_k^T. \tag{9.21}$$

This simplifies to

$$\hat{P}_k = \hat{P}_k^- - K_k H\hat{P}_k^- - \hat{P}_k^- H^T K_k^T + K_k(H\hat{P}_k^- H^T)K_k^T + K_k R K_k^T. \tag{9.22}$$

$$\hat{P}_k = \hat{P}_k^- - K_k H\hat{P}_k^- - \hat{P}_k^- H^T K_k^T + K_k(H\hat{P}_k^- H^T + R)K_k^T. \tag{9.23}$$

We want to minimize the trace of $\hat{P}_k$. Using vector calculus, it can be shown that

$$\frac{\partial \text{trace}(\hat{P}_k)}{\partial K_k} = -2(H\hat{P}_k^-)^T + 2K_k(H\hat{P}_k^- H^T + R). \tag{9.24}$$

Setting the derivative equal to 0,

$$-2(H\hat{P}_k^-)^T + 2K_k(H\hat{P}_k^- H^T + R) = 0. \tag{9.25}$$

$$K_k = (H\hat{P}_k^-)^T(H\hat{P}_k^- H^T + R)^{-1}. \tag{9.26}$$

$$K_k = \hat{P}_k^- H^T(H\hat{P}_k^- H^T + R)^{-1}. \tag{9.27}$$

Using this optimal Kalman gain, $\hat{P}_k$ simplifies further.

$$\hat{P}_k = \hat{P}_k^- - K_k H\hat{P}_k^- - \hat{P}_k^- H^T K_k^T + K_k(H\hat{P}_k^- H^T + R)K_k^T. \tag{9.28}$$

$$\begin{aligned}\hat{P}_k = \hat{P}_k^- - \hat{P}_K^- H^T(H\hat{P}_k^- H^T + R)^{-1} H\hat{P}_k^- - \\ \hat{P}_k^- H^T(\hat{P}_K^- H^T(H\hat{P}_k^- H^T + R)^{-1}))^T + \\ \hat{P}_K^- H^T(H\hat{P}_k^- H^T + R)^{-1}(H\hat{P}_k^- H^T + R)(\hat{P}_k^- H^T(H\hat{P}_k^- H^T + R)^{-1})^T\end{aligned} \tag{9.29}$$

$$\hat{P}_k = \hat{P}_k^- - \hat{P}_k^- H^T (H\hat{P}_k^- H^T + R)^{-1} H\hat{P}_k^- \tag{9.30}$$

$$\hat{P}_k = (I - K_k H)\hat{P}_k^-. \tag{9.31}$$

Again, remember that this formula is only correct when we use the optimal Kalman gain given by (9.19).

The algorithm can be summarized as follows. For $k = 1, 2, \ldots,$

1. Let $\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$.
2. Let $\hat{P}_k^- = A\hat{P}_{k-1}A^T + Q$.
3. Let $K_k = \hat{P}_k^- H^T (H\hat{P}_k^- H^T + R)^{-1}$.
4. Let $\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$.
5. Let $\hat{P}_k = (I - K_k H)\hat{P}_k^-$.

In practice, we may not have an observation at every time step. In that case, we can use predictions at each time step and compute updates steps whenever observations become available.

Example 9.6

In this example, we'll consider a system governed by the second order differential equation

$$y''(t) + 0.01y'(t) + y(t) = \sin(2t) \tag{9.32}$$

with the initial conditions $y(0) = 0.1$, $y'(0) = 0.5$.

We must first use a standard trick to convert this second order ordinary differential equation into a system of two first order differential equations. Let

$$x_1(t) = y(t) \tag{9.33}$$

and

$$x_2(t) = y'(t). \tag{9.34}$$

The relation between $x_1(t)$ and $x_2(t)$ is

$$x_1'(t) = x_2(t). \tag{9.35}$$

Also, (9.32) becomes

$$x_2'(t) = -x_1(t) - 0.01x_2(t) + \sin(2t). \tag{9.36}$$

This system of two first order equations can be written as

$$x'(t) = Ax(t) + Bu(t) \tag{9.37}$$

where

$$A = \begin{bmatrix} 0 & 1 \\ -1 & -0.01 \end{bmatrix}, \tag{9.38}$$

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{9.39}$$

and

$$u(t) = \begin{bmatrix} 0 \\ \sin(2t) \end{bmatrix}. \tag{9.40}$$

This system of differential equations will be discretized using (9.8) with time steps of $\Delta t = 0.01$. At each time step, the state vector will be randomly perturbed with $N(0, Q)$, noise, where

$$Q = \begin{bmatrix} 0.0005 & 0.0 \\ 0.0 & 0.0005 \end{bmatrix}. \tag{9.41}$$

We will observe $x_1(t)$ once per second (every 100 time steps.) Thus

$$H = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{9.42}$$

Our observations will have a variance of 0.0005.

For the initial conditions we will begin with the estimate

$$\hat{x}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{9.43}$$

and covariance

$$\hat{P}_0 = \begin{bmatrix} 0.5 & 0.0 \\ 0.0 & 0.5 \end{bmatrix}. \tag{9.44}$$

Figure 9.1 shows the true state of the system. Figure 9.2 shows the estimate of the system state using only prediction steps. The dotted lines in this plot are one standard-deviation error bars. The initial uncertainty in $x(t)$ is due to uncertainty in the initial conditions. Later, this uncertainty increases due to the effect of noise on the state of the system.

Figure 9.3 shows the Kalman filter estimates including observations of $x_1(t)$ once per second. Although the initial uncertainty is quite high, the Kalman filter quickly "learns" the actual state of the system and then tracks it quite closely. Each circle on the $x_1(t)$ plot represents an observation of the system. Notice that when an observation is obtained the Kalman estimate "jumps" to incorporate the new observation. Also note that although we only observe $x_1(t)$, the Kalman filter also manages to track $x_2(t)$. This happens because the system of differential equations connects $x_1(t)$ and $x_2(t)$. Figure 9.4 shows the true state of the system and the Kalman filter estimate on the same plot.

Figure 9.5 shows the differences between the system state and the simple prediction. Figure 9.6 shows the difference between the system state and the Kalman prediction. Notice that the Kalman filter produced much tighter estimates of both $x_1(t)$ and $x_2(t)$ using only a few observations of $x_1(t)$.

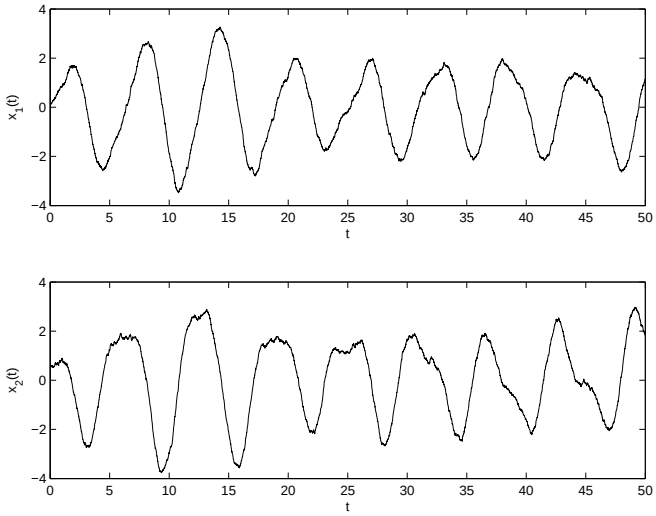

Figure 9.1: Plot of the system state $x(t)$.

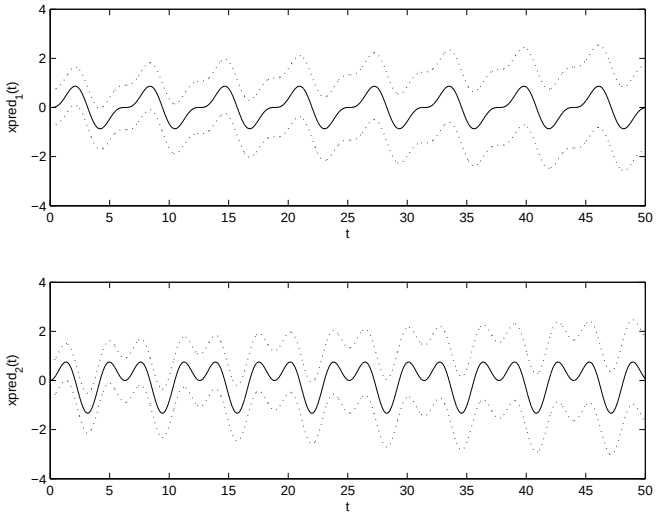

Figure 9.2: Estimate of $x(t)$ using prediction steps only.

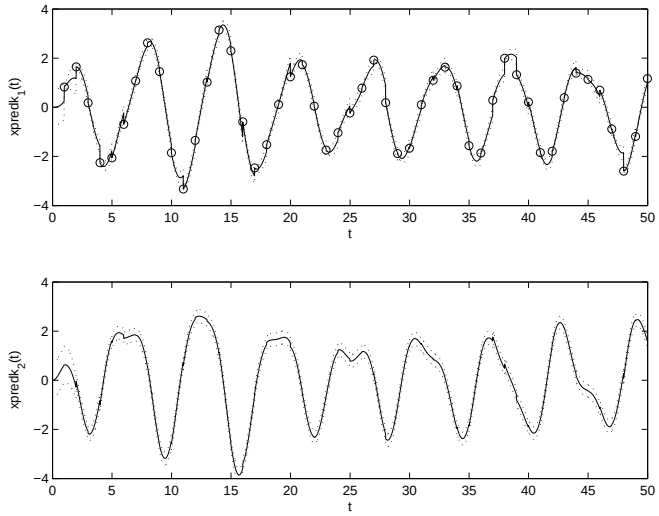

Figure 9.3: Kalman estimates of the $x_1(t)$ and $x_2(t)$.

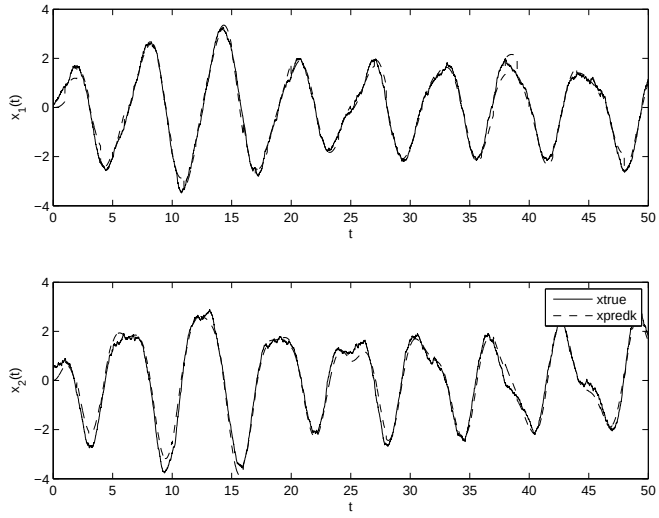

Figure 9.4: Kalman estimate versus the true values of $x_1(t)$ and $x_2(t)$.

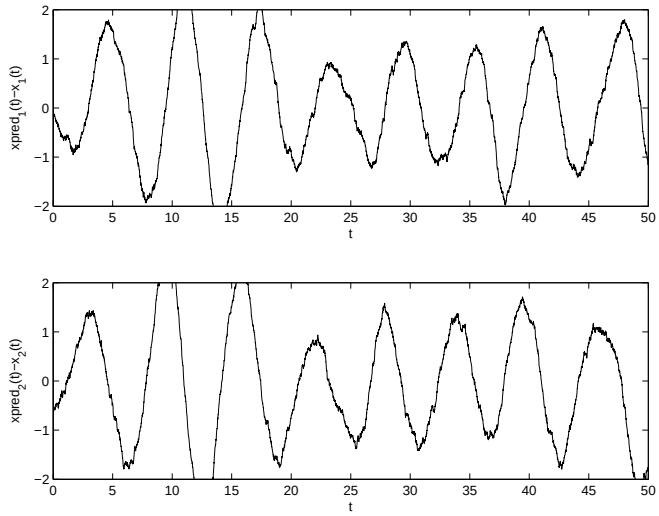

Figure 9.5: Difference between the system state and prediction estimate.

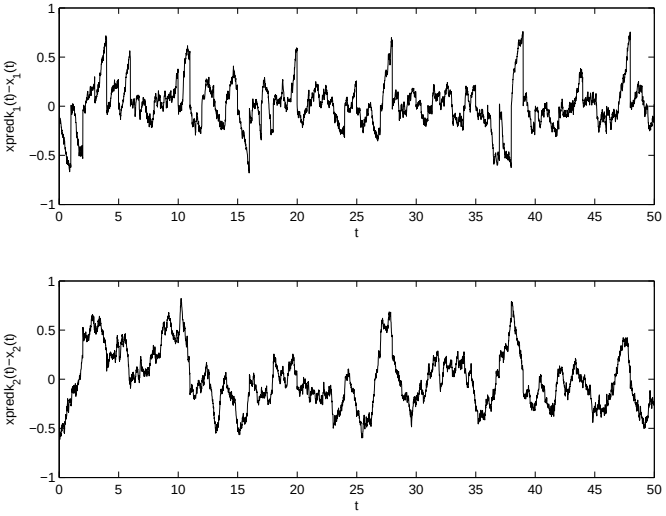

Figure 9.6: Difference between the system state and Kalman estimate.

The Extended Kalman Filter

The Extended Kalman Filter (EKF) extends the Kalman filtering concept to problems with nonlinear dynamics. Our new equation for the time evolution of the system state will be of the form

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}) \tag{9.45}$$

where w_{k-1} is a random perturbation of the system. This time, we'll assume that w_{k-1} has a multivariate normal distribution with mean 0 and covariance matrix Q_{k-1}. That is, the covariance is allowed to be time dependent.

Our new measurement model will be

$$z_k = h(x_k, v_k) \tag{9.46}$$

where v_k is a multivariate normal $N(0, R_k)$ noise vector.

The prediction step is a straight forward generalization of what we have previously done in the Kalman filter.

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, 0). \tag{9.47}$$

We'll also introduce a new notation for the predicted observation

$$\hat{z}_k^- = h(\hat{x}_k^-, 0). \tag{9.48}$$

In general, for a nonlinear function f, $\hat{x}_k^-$ will not have a multivariate normal distribution. However, we can reasonably hope that $f(x, u, w)$ will be approximately linear for relatively small changes in x and w, so that $\hat{x}_k^-$ will be at least approximately normally distributed.

We linearize $f(x, u, w)$ around $(\hat{x}_{k-1}, u_{k-1}, 0)$ as

$$f(x, u, w) = f(\hat{x}_{k-1}, u_{k-1}, 0) + A_{k-1}(x - \hat{x}_{k-1}) + W_{k-1}(w - 0) \tag{9.49}$$

where A and W are matrices of partial derivatives of f with respect to x and w. Note that since u_{k-1} is assumed to be known exactly, we don't need to linearize in the u variable. The entries in A_{k-1} and W_{k-1} are given by

$$A_{i,j,k-1} = \frac{\partial f_i(\hat{x}_{k-1}, u_{k-1}, 0)}{\partial x_j}. \tag{9.50}$$

$$W_{i,j,k-1} = \frac{\partial f_i(\hat{x}_{k-1}, u_{k-1}, 0)}{\partial w_j}. \tag{9.51}$$

Using this linearization, we end up with an approximate covariance matrix for $\hat{x}_k^-$,

$$\hat{P}_k^- = A_{k-1}\hat{P}_{k-1}A_{k-1}^T + W_{k-1}Q_{k-1}W_{k-1}^T. \tag{9.52}$$

Similarly, we can linearize $h()$. Let

$$H_{i,j,k} = \frac{\partial h_i(\hat{x}_k^-, 0)}{\partial x_j}. \tag{9.53}$$

$$V_{i,j,k} = \frac{\partial h_i(\hat{x}_k^-, 0)}{\partial v_j}. \tag{9.54}$$

Now, let

$$\hat{e}_{x_k}^- = x_k - \hat{x}_k^- \tag{9.55}$$

and

$$\hat{e}_{z_k}^- = z_k - \hat{z}_k^-. \tag{9.56}$$

We don't actually know x_k, but we do expect $x_k - \hat{x}_k^-$ to be relatively small. Thus we can use our linearization of $f()$ to derive an approximation for $\hat{e}_{x_k}^-$.

$$\hat{e}_{x_k}^- = f(x_{k-1}, u_{k-1}, w_{k-1}) - f(\hat{x}_{k-1}, u_{k-1}, 0). \tag{9.57}$$

By the linearization,

$$\hat{e}_{x_k}^- \approx A_{k-1}(x_{k-1} - \hat{x}_{k-1}) + \epsilon_k \tag{9.58}$$

where ϵ_k accounts for the effect of the random w_{k-1}. The distribution of ϵ_k is $N(0, W_{k-1}Q_{k-1}W_{k-1}^T)$. Similarly,

$$\hat{e}_{z_k}^- = h(x_k, v_k) - h(\hat{x}_k^-, 0). \tag{9.59}$$

By the linearization this is approximately

$$\hat{e}_{z_k}^- \approx H\hat{e}_{x_k}^- + \eta_k \tag{9.60}$$

where η_k has an $N(0, V_k R_k V_k^T)$ distribution.

Ideally, we could update $\hat{x}_k^-$ to get x_k by

$$x_k = \hat{x}_k^- + \hat{e}_{x_k}^-. \tag{9.61}$$

Of course, we don't know $\hat{e}_{x_k}^-$, but we can estimate it. Let

$$\hat{e}_{x_k} = K_k(z_k - \hat{z}_k^-) \tag{9.62}$$

where K_k is a Kalman gain factor to be determined. Then let

$$\hat{x}_k = \hat{x}_k^- + \hat{e}_{x_k}. \tag{9.63}$$

By a derivation similar to our earlier derivation of the optimal Kalman gain for the linear Kalman filter, it can be shown that the optimal Kalman gain for the EKF is

$$K_k = \hat{P}_k^- H_k^T (H_k \hat{P}_k^- H_k^T + V_k R_k V_k^T)^{-1}. \tag{9.64}$$

Using this optimal Kalman gain, the covariance matrix for the updated estimate $\hat{x}_k$ is

$$\hat{P}_k = (I - K_k H_k)\hat{P}_k^-. \tag{9.65}$$

The EKF algorithm can be summarized as follows. For $k = 1, 2, \ldots$,

1. Let $\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, 0)$.
2. Let $\hat{P}_k^- = A_{k-1}\hat{P}_{k-1}A_{k-1}^T + W_{k-1}Q_{k-1}W_{k-1}^T$.
3. Let $K_k = \hat{P}_k^- H_k^T (H_k \hat{P}_k^- H_k^T + V_k R_k V_k^T)^{-1}$.
4. Let $\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-, 0)))$.
5. Let $\hat{P}_k = (I - K_k H_k)\hat{P}_k^-$.

The Ensemble Kalman Filter

A fundamental problem with the EKF is that we must compute the partial derivatives of $f()$ so that they're available for computing the covariance matrix in the prediction step. An alternative approach involves using Monte Carlo simulation. In the Ensemble Kalman Filter (EnKF), we generate a collection of state variables according to the MVN distribution and time $k = 0$, and then follow the evolution of this ensemble through time.

In the following, we'll assume that our system state evolves according to

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}) \tag{9.66}$$

and that our observation model is

$$z_k = Hx_k + v_k. \tag{9.67}$$

Instead of using partial derivatives of $f()$ to estimate $\hat{x}_k^-$ and $\hat{P}_k^-$, we'll use Monte Carlo simulation. Suppose that we're given a collection (or ensemble) of random state vectors at time $k-1$, $\hat{x}_{k-1}^i$, $i = 1, 2, \ldots, m$. For each random state vector, we can generate a random $N(0, Q_{k-1})$ vector w_{k-1}^i, and then update the vector with

$$\hat{x}_k^{i,-} = f(\hat{x}_{k-1}^i, u_{k-1}, w_{k-1}^i), \;\; i = 1, 2, \ldots, m. \tag{9.68}$$

Now, we can estimate $\hat{x}_k^-$ and the covariance matrix $\hat{P}_k^-$ from the vectors $\hat{x}_k^{i,-}$.

Next, we need to derive an update algorithm. We could simply use the observation z_k in the update formula from the Kalman filter for each state vector in the ensemble. However, this tends to drive all of the members of the ensemble towards the same state. Rather, at time k, we obtain a new observation z_k, which is assumed to include MVN $N(0, R)$ noise. By adding $N(0, R)$ noise to z_k, we obtain an ensemble of simulated observations, z_k^i, $i = 1, 2, \ldots, m$.

We update our ensemble of solutions with

$$\hat{x}_k^i = \hat{x}_k^{i,-} + \hat{P}_k^- H^T (H\hat{P}_k^- H^T + R)^{-1}(z_k^i - H\hat{x}_k^{i,-}). \tag{9.69}$$

This is simply the Kalman filter update, but using the estimated covariance matrix $\hat{P}_k^-$, and the Monte Carlo simulated observations z_k^i.

Finally, we can use the ensemble of states $\hat{x}_k^i$, $i = 1, 2, \ldots, m$ to estimate the mean state, $\hat{x}_k$, and the covariance $\hat{P}_k$.

The algorithm can be summarized as follows. Begin by constructing an ensemble of m state vectors with mean $\hat{x}_0$ and covariance $\hat{P}_0$. For $k = 1, 2, \ldots$,

1. Let $\hat{x}_k^{i,-} = f(\hat{x}_{k-1}^i, u_{k-1}, w_{k-1}^i)$, $i = 1, 2, \ldots, m$.
2. Estimate $\hat{P}_k^-$ and $\hat{x}_k^-$ from the $\hat{x}_k^{i,-}$.
3. Generate an ensemble of observations $z_k^i = z_k + v_k^i$, for $i = 1, 2, \ldots, m$.
4. Let $\hat{x}_k^i = \hat{x}_k^{i,-} + CH^T(HCH^T + R)^{-1}(z_k^i - H\hat{x}_k^{i,-})$ for $i = 1, 2, \ldots, m$.
5. Estimate $\hat{x}_k$ and $\hat{P}_k$ from the $\hat{x}_k^i$.

Example 9.7

Continuing our previous example, we repeated the computations using the ensemble Kalman figure with $m = 1,000$ state vectors in the ensemble. Figure 9.7 shows the resulting solution, which is virtually identical to the solution obtained using the Kalman filter shown in Figure 9.4.

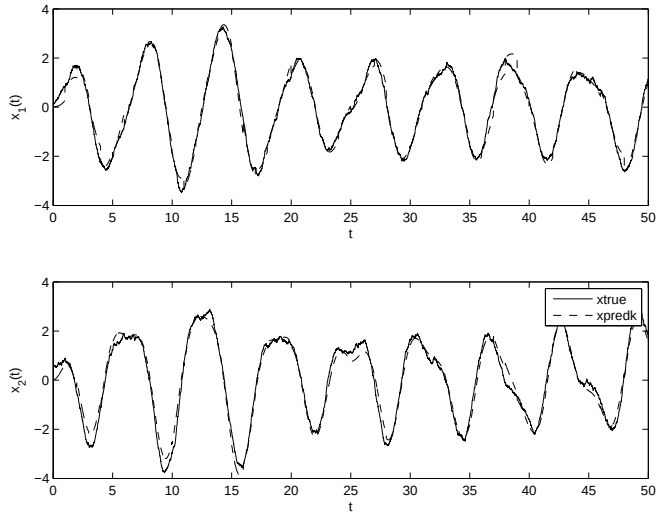

Figure 9.7: Ensemble Kalman estimate versus the true values of $x_1(t)$ and $x_2(t)$.

Chapter 10

ARMA Modeling

Some notation

In the following, we will make use of forward and backward shifts in time. The B operator is defined by

$$Bz_n = z_{n-1} \tag{10.1}$$

while the F operator is

$$Fz_n = z_{n+1}. \tag{10.2}$$

Note that B and F are not numbers but rather operators that act on a time series z_n. We can extend this notation to include powers of B and F

$$B^k z_n = z_{n-k} \tag{10.3}$$

and

$$F^k z_n = z_{n+k}. \tag{10.4}$$

Also, we can build polynomials from the B and F operators. For example,

$$(1 - B + 0.5B^2)z_n = z_n - z_{n-1} + 0.5z_{n-2}. \tag{10.5}$$

Gaussian White Noise

We will frequently make use of a **Gaussian white noise**. The white noise process has A_n normally distributed with mean 0, variance σ_A^2, and autocovariance $\gamma_k = 0$, $k = 1, 2, \ldots$ and autocorrelation $\rho_k = 0$, $k = 0, 1, \ldots$. White noise can easily be generated in MATLAB using the **randn** command.

Using our formula for the spectrum of a stationary process from its autocovariance, it's easy to show that the white noise process should have $I(f) = 2\sigma_A^2$, $0 \leq f \leq 1/2$. In the limit as n goes to infinity, the spectrum is constant for all frequencies. However, for any actual realization of the white noise process, the sample spectrum will contain considerable noise.

The ARMA process

An **autoregressive moving average (ARMA)** process is obtained by applying a recursive filter to Gaussian white noise. In terms of the elements of the z_n and a_n sequences,

$$z_n = \phi_1 z_{n-1} + \phi_2 z_{n-2} + \ldots + \phi_p z_{n-p} + a_n - \theta_1 a_{n-1} - \ldots - \theta_q a_{n-q}. \quad (10.6)$$

The terms $\phi_1 z_{n-1}$ through $\phi_p z_{n-p}$ are the autoregressive portion of the filter. The terms a_n through $\theta_q a_{n-q}$ are a moving average of the white noise input process. Notice that this has the form of the recursive IIR filter that we previously considered, except that the first coefficients have been normalized to 1.

In terms of our operator notation,

$$\phi(B) z_n = \theta(B) a_n \quad (10.7)$$

where

$$\phi(B) = 1 - \phi_1 B - \ldots - \phi_p B^p \quad (10.8)$$

and

$$\theta(B) = 1 - \theta_1 B - \ldots - \theta_q B^q. \quad (10.9)$$

Note the unusual notational convention of minus signs in front of each coefficient.

Let

$$\psi(B) = \frac{\theta(B)}{\phi(B)}. \quad (10.10)$$

$$\psi(B) = \frac{1 - \theta_1 B - \theta_2 B^2 - \ldots - \theta_q B^q}{1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p}. \quad (10.11)$$

Then we can write

$$z_n = \psi(B) a_n. \quad (10.12)$$

Earlier, when we found the z-transform transfer function for a filter, we wrote the transfer functions in powers of z^{-1}. The z-transform transfer function for our filter would be

$$\Phi(z) = \frac{1 - \theta_1 z^{-1} - \theta_2 z^{-2} - \ldots - \theta_q z^{-q}}{1 - \phi_1 z^{-1} - \phi_2 z^{-2} - \ldots - \phi_p z^{-p}} \quad (10.13)$$

It's apparent that in our new notation, $\Psi(B) = \theta(B)/\phi(B)$ is equivalent to $\Phi(z)$ in the z-transform notation, with $B = 1/z$.

The power spectrum can be obtained by substituting $B = e^{-2\pi i f}$ in the transfer function

$$I(f) = 2\sigma_a^2 \left| \frac{\theta(e^{-2\pi i f})}{\phi(e^{-2\pi i f})} \right|^2. \quad (10.14)$$

Note that the minus sign in the exponent of $B = e^{-2\pi i f}$ is again because of the difference in notation between the Z transform and the B notation- $B = 1/z$.

We can expand $\psi(B)$ as

$$\psi(B) = 1 + \psi_1 B + \psi_2 B^2 + \dots \tag{10.15}$$

where the coefficients ψ_k can be obtained by Taylor series expansion. This allows us to write z_n in terms of the inputs at time n and previous times.

$$z_n = a_n + \psi_1 a_{n-1} + \psi_2 a_{n-2} + \dots . \tag{10.16}$$

Note that the constant coefficient is always 1 and that this time (only) we've used positive signs in front of the coefficients. From time to time it will be helpful to use the notation $\psi_0 = 1$, so that we don't have to treat the first term in this power series as a special case.

Example 10.8

Consider the filter in which

$$\phi(B) = 1 - 0.5B \tag{10.17}$$

and

$$\theta(B) = 1. \tag{10.18}$$

In terms of the filter equations,

$$z_n - 0.5 z_{n-1} = a_n \tag{10.19}$$

or

$$z_n = a_n + 0.5 z_{n-1}. \tag{10.20}$$

We can recursively apply the equation to write this as

$$z_n = a_n + 0.5(a_{n-1} + 0.5 z_{n-2}). \tag{10.21}$$

$$z_n = a_n + 0.5(a_{n-1} + 0.5(a_{n-2} + 0.5 z_{n-3}). \tag{10.22}$$

We end up with

$$z_n = a_n + 0.5 a_{n-1} + 0.25 a_{n-2} + 0.125 a_{n-3} + \dots . \tag{10.23}$$

Using the operator notation, it is much simpler to get the same result by doing a Taylor series expansion.

$$\Psi(B) = \frac{1}{1 - 0.5B} = 1 + 0.5B + 0.25B^2 + 0.125B^3 + \dots . \tag{10.24}$$

An alternative is to let

$$\pi(B) = \frac{1}{\psi(B)} = \frac{\phi(B)}{\theta(B)}. \tag{10.25}$$

In this case,

$$a_n = \pi(B) z_n. \tag{10.26}$$

Clearly,

$$\pi(B) = \frac{1}{\psi(B)}. \tag{10.27}$$

We can expand $\pi(B)$ in a Taylor's series as

$$\pi(B) = 1 - \pi_1 B - \pi_2 B^2 - \dots \tag{10.28}$$

Example 10.9

Consider the ARMA process

$$Z_n - 0.5 Z_{n-1} = A_n - 0.3 A_{n-1} + 0.2 A_{n-2}. \tag{10.29}$$

Here $\phi(B) = 1 - 0.5B$ and $\theta(B) = 1 - 0.3B + 0.2B^2$. Using Maple to compute the Taylor's series, we obtain

$$\psi(B) = \frac{\theta(B)}{\phi(B)} = 1 + 0.2B + 0.3B^2 + 0.15B^3 + \dots \tag{10.30}$$

Thus $\psi_1 = 0.2$, $\psi_2 = 0.3$, and $\psi_3 = 0.15$. Similarly,

$$\pi(B) = \frac{\phi(B)}{\theta(B)} = 1 - 0.2B - 0.26B^2 - 0.038B^3 - \dots. \tag{10.31}$$

Thus $\pi_1 = 0.2$, $\pi_2 = 0.26$, and $\pi_3 = 0.038$.

Stationarity and Invertibility

Unfortunately, it is easy to write down an ARMA process which is not covariance stationary. For example, let

$$\psi(B) = \frac{1}{1 - B} = 1 + B + B^2 + \dots \tag{10.32}$$

Then

$$Z_n = \sum_{k=-\infty}^{n} A_k \tag{10.33}$$

and

$$Var(Z_n) = \sum_{k=-\infty}^{n} Var(A_k) = \infty. \tag{10.34}$$

It can be shown that if

$$\sum_{j=1}^{\infty} |\psi_j| < \infty \tag{10.35}$$

then the ARMA process is stationary. This happens if the series $\psi(B)$ converges for every B with $|B| \leq 1$. Since $\psi(B)$ is a rational function, it can also be shown that the series converges for every B with $|B| \leq 1$ if the complex zeros of $\phi(B)$ lie outside the unit circle.

Recall that when we worked with the z transform of a digital filter, the stability condition was that the poles of the transfer function must lie within the unit circle. Why is the stability condition for an ARMA process that the zeros of $\phi(B)$ must lie outside the unit circle? The problem here is one of notation. In the digital filtering case, the transfer function was a rational function of $1/z$. Here in the ARMA case, the transfer function is a rational function of B. Thus B and $1/z$ are effectively playing the same role. When a pole lies inside the unit circle in the z plane, the corresponding pole lies outside of the unit circle in the B plane, where $B = 1/z$.

If we have a stationary ARMA process, then since $Z_n = \psi(B)A_n$, and the expected values of A_n are all 0, the expected value of Z_n is also 0.

A related issue is that of **invertibility**. Recall that we can write z_n in terms of a_n and previous values of z_{n-k}. That is,

$$\pi(B)z_n = a_n \tag{10.36}$$

or

$$z_n = a_n + \pi_1 z_{n-1} + \pi_2 z_{n-2} + ... \tag{10.37}$$

This **inverted form** of the process provides a very useful way of generating a random sequence according to our ARMA process. However, this infinite sum must be truncated in practice. If the π_j coefficients do not decay to zero, then it isn't possible to approximate this infinite sum by truncating it.

We say that the process is **invertible** if

$$\sum_{j=1}^{\infty} |\pi_j| < \infty \tag{10.38}$$

Since $\pi(B)$ is a rational function, the series is invertible if the complex zeros of $\theta(B)$ lie outside of the unit circle.

Example 10.10

Recall the ARMA process of example 5. In this case, since $\phi(B) = 1 - 0.5B$, the only zero of $\phi(B)$ is at $B = 2$, which is outside of the unit circle, so the process is stationary. The zeros of $\theta(B)$ are at $B = 0.75 \pm 2.1i$, so the process is also invertible.

Finding the autocovariance and autocorrelation of an ARMA process

The $\psi()$ form of the ARMA model can be used to find $Var(Z_n)$. Since

$$Z_n = \sum_{k=0}^{\infty} \psi_k A_{n-k} \tag{10.39}$$

and the A_i are independent with mean 0 and variance σ_A^2, we can compute

$$Var(Z_n) = \sum_{k=0}^{\infty} \psi_k^2 Var(A_{n-k}) = \sigma_A^2 \sum_{k=0}^{\infty} \psi_k^2. \tag{10.40}$$

If we know the value of this infinite sum, then we're all set. If we don't know the infinite sum, but the ψ_k coefficients decay quickly to 0, then we can truncate the infinite series and get a good approximation to $Var(Z_n)$.

Similarly, we can use the $\psi()$ form of the ARMA process to find covariances between Z_n and A_{n-k} for $k = 0, \ldots$.

$$Cov(Z_n, A_{n-k}) = Cov\left(\sum_{k=0}^{\infty} \psi_k A_{n-k}, A_{n-k}\right). \tag{10.41}$$

Since the A_i are independent of each other,

$$Cov(Z_n, A_{n-k}) = Cov(\psi_k A_{n-k}, A_{n-k}) = \psi_k \sigma_A^2. \tag{10.42}$$

In order to find the autocovariance of an ARMA process, we start with the model in the recursive filter form.

$$Z_n = \phi_1 Z_{n-1} + \phi_2 Z_{n-2} + \ldots + \phi_p Z_{n-p} + A_n - \theta_1 A_{n-1} - \ldots - \theta_q A_{n-q}. \tag{10.43}$$

Next, we multiply both sides by Z_{n-k} and take expected values. Since $E[Z_n] = E[z_{n-k}] = E[A] = 0$, $Cov(Z_n, Z_{n-k}) = E[Z_n Z_{n-k}]$. Thus

$$\begin{aligned} Cov(Z_n, Z_{n-k}) &= \phi_1 Cov(Z_{n-1}, Z_{n-k}) + \ldots + \phi_p Cov(Z_{n-p}, Z_{n-k}) \\ &\quad + Cov(Z_{n-k}, A_n) - \theta_1 Cov(Z_{n-k}, A_{n-1}) - \ldots - \theta_q Cov(Z_{n-k}, A_{n-q}) \end{aligned}$$

So,

$$\gamma_k = \phi_1 \gamma_{k-1} + \ldots + \phi_p \gamma_{k-p} + \gamma_{ZA}(k) - \theta_1 \gamma_{ZA}(k-1) - \ldots - \theta_q \gamma_{ZA}(k-q) \tag{10.44}$$

where

$$\gamma_{ZA}(k-j) = Cov(Z_{n-k}, A_{n-j}) \tag{10.45}$$

Since Z_{n-k} is independent of the white noise at times after $n-k$, these covariances are 0. Also, since $Z_{n-k} = \sum_{j=0}^{\infty} \psi_j A_{n-k-j}$, the remaining covariances are given by $Cov(Z_{n-k}, A_{n-k-j}) = \psi_j \sigma_A^2$. Thus

$$\gamma_{ZA}(j) = \begin{cases} 0 & j > 0 \\ \psi_{-j} \sigma_A^2 & j \leq 0 \end{cases} \tag{10.46}$$

So, we can express the autocovariance at lag k as

$$\gamma_k = \phi_1\gamma_{k-1} + \ldots + \phi_p\gamma_{k-p} + \sigma_A^2(-\theta_k\psi_0 - \theta_{k+1}\psi_1 - \ldots - \theta_q\psi_{q-k}) \quad (10.47)$$

When $k \geq q+1$, this simplifies to

$$\gamma_k = \phi_1\gamma_{k-1} + \ldots + \phi_p\gamma_{k-p}. \quad (10.48)$$

Another important case is $k = 0$. The variance γ_0 is given by

$$\gamma_0 = \phi_1\gamma_1 + \ldots + \phi_p\gamma_p + \sigma_A^2(1 - \theta_1\psi_1 - \ldots - \theta_q\psi_q) \quad (10.49)$$

These recurrence relations can be solved to obtain the autocovariance and autocorrelation.

As an example, consider the second order autoregressive process

$$Z_n = \phi_1 Z_{n-1} + \phi_2 Z_{n-2} + A_n \quad (10.50)$$

It can be show that this process is stationary if $\phi_1 + \phi_2 < 1$, $\phi_2 - \phi_1 < 1$, and $-1 < \phi_2 < 1$. Because $\theta(B) = 1$ this process is always invertible.

To compute the autocovariance, we multiply the above formula by Z_{n-k} and take expected values.

$$Cov(Z_n, Z_{n-k}) = \phi_1 Cov(Z_{n-1}, Z_{n-k}) + \phi_2 Cov(Z_{n-2}, Z_{n-k}) + Cov(A_n, Z_{n-k}) \quad (10.51)$$

When $k = 0$, we get

$$\gamma_0 = \phi_1\gamma_1 + \phi_2\gamma_2 + Cov(A_n, Z_n) \quad (10.52)$$

But $Z_n = \phi_1 Z_{n-1} + \phi_2 Z_{n-2} + A_n$, and A_n is independent of Z_{n-1} and Z_{n-2}, so

$$\gamma_0 = \phi_1\gamma_1 + \phi_2\gamma_2 + Cov(A_n, A_n) \quad (10.53)$$

or

$$\gamma_0 = \phi_1\gamma_1 + \phi_2\gamma_2 + \sigma_A^2 \quad (10.54)$$

When $k > 0$, $Cov(A_n, Z_{n-k}) = 0$, and we get

$$\gamma_k = \phi_1\gamma_{k-1} + \phi_2\gamma_{k-2} \quad (10.55)$$

In terms of the autocorrelation function, we have

$$\rho_0 = 1 \quad (10.56)$$

and

$$\rho_1 = \phi_1\rho_0 + \phi_2\rho_1. \quad (10.57)$$

Solving this equation for ρ_1, we get

$$\rho_1 = \frac{\phi_1}{1 - \phi_2} \quad (10.58)$$

For $k > 2$, we get

$$\rho_k = \phi_1\rho_{k-1} + \phi_2\rho_{k-2} \quad k > 2 \quad (10.59)$$

Example 10.11

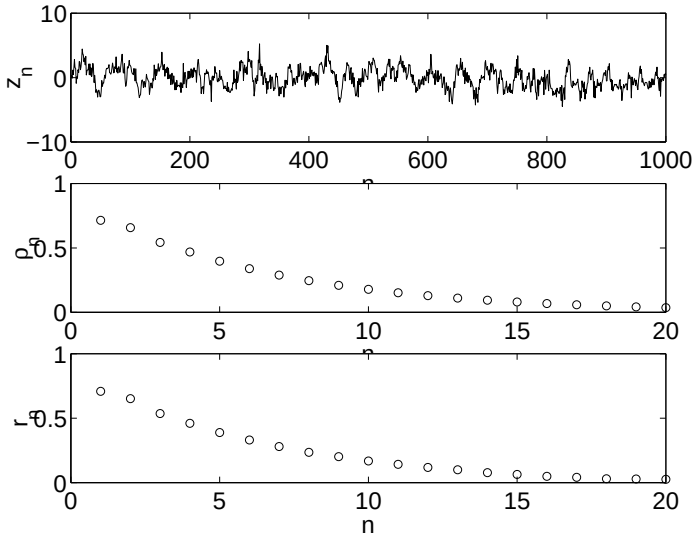

Figure 10.1: An AR(2) process with $\phi_1 = 0.5$, $\phi_2 = 0.3$.

Consider an AR(2) process with $\phi_1 = 0.5$ and $\phi_2 = 0.3$. We generated a random sequence according to this process. Figure 10.1a shows the first 1000 points of this random process. Figure 10.1b shows the theoretical autocorrelation. Figure 10.1 shows the autocorrelation as estimated from the 20,000 point sequence.

Next, consider the ARMA(1,1) process

$$Z_n - \phi_1 Z_{n-1} = A_n - \theta_1 A_{n-1} \tag{10.60}$$

Here $\phi(B) = 1 - \phi_1 B$ and $\theta(B) = 1 - \theta_1 B$. We need to make sure that the roots of $\phi(B)$ and $\theta(B)$ are outside of the unit circle. This process is stationary if $-1 < \phi_1 < 1$ and invertible when $-1 < \theta_1 < 1$. We can also compute $\psi(B) = 1 + (\phi_1 - \theta_1)B + \ldots$. The recurrence relations for the autocovariance give

$$\gamma_0 = \phi_1 \gamma_1 + \sigma_A^2 (1 - \theta_1 \psi_1) \tag{10.61}$$

$$\gamma_1 = \phi_1 \gamma_0 - \theta_1 \sigma_A^2 \tag{10.62}$$

$$\gamma_k = \phi_1 \gamma_{k-1} \quad k \geq 2 \tag{10.63}$$

These equations can be solved for the autocovariance. We can then convert the solution to an autocorrelation function. The result is

$$\rho_1 = \frac{(1 - \phi_1 \theta_1)(\phi_1 - \theta_1)}{1 + \theta_1^2 - 2\phi_1 \theta_1} \tag{10.64}$$

Name	p	q	ρ_1	ρ_2	ρ_k, $k \geq 3$
AR(1)	1	0	$\rho_1 = \phi_1$	$\rho_2 = \phi_1^2$	$\rho_k = \phi_1^k$
AR(2)	2	0	$\rho_1 = \frac{\phi_1}{1-\phi_2}$	$\rho_2 = \phi_1\rho_1 + \phi_2$	$\rho_k = \phi_1\rho_{k-1} + \phi_2\rho_{k-2}$
ARMA(1,1)	1	1	$\rho_1 = \frac{(1-\phi_1\theta_1)(\phi_1-\theta_1)}{1+\theta_1^2-2\phi_1\theta_1}$	$\rho_2 = \phi_1\rho_1$	$\rho_k = \phi_1\rho_{k-1}$
MA(1)	0	1	$\rho_1 = \frac{-\theta_1}{1+\theta_1^2}$	$\rho_2 = 0$	$\rho_k = 0$
MA(2)	0	2	$\rho_1 = \frac{-\theta_1(1-\theta_2)}{1+\theta_1^2+\theta_2^2}$	$\rho_2 = \frac{-\theta_2}{1+\theta_1^2+\theta_2^2}$	$\rho_k = 0$

Table 10.1: Autocorrelations for ARMA(p,q) processes. Based on formulas from Chapter 3 of [5].

Name	p	q	Stationarity	Invertibility
AR(1)	1	0	$\lvert\phi_1\rvert < 1$	none
AR(2)	2	0	$\lvert\phi_2\rvert < 1$, $\phi_1 + \phi_2 < 1$, $\phi_2 - \phi_1 < 1$	none
ARMA(1,1)	1	1	$\lvert\phi_1\rvert < 1$	$\lvert\theta_1\rvert < 1$
MA(1)	0	1	none	$\lvert\theta_1\rvert < 1$
MA(2)	0	2	none	$\lvert\theta_2\rvert < 1$, $\theta_1 + \theta_2 < 1$, $\theta_2 - \theta_1 < 1$

Table 10.2: Stationarity and Invertibility Conditions ARMA(p,q) processes. Based on formulas from Chapter 3 of Box, Jenkins, and Reinsel.

$$\rho_k = \phi_1\rho_{k-1} \quad k >= 2 \tag{10.65}$$

These computations can all be performed for arbitrary ARMA(p,q) processes. However, in practice, the most important processes have p and q quite small, and general solutions for these particular ARMA(p,q) processes have been developed. Box, Jenkins, and Reinsel contains specific solutions for the autocorrelations of a variety of ARMA(p,q) processes with small values of p and q [5]. Table 10.1 summarizes these formulas. Table 10.2 gives the stationarity and invertibility conditions for these ARMA models.

The Partial Autocorrelation Function

Suppose that our ARMA process is purely autoregressive of order k. That is,

$$Z_n = A_n + \phi_{k1}Z_{n-1} + \phi_{k2}Z_{n-2} + \ldots + \phi_{kk}Z_{n-k} \tag{10.66}$$

In this case, the equations for the autocorrelations ρ_j, $j = 1, 2, \ldots k$ are particularly simple. They take the form

$$\rho_j = \phi_{k1}\rho_{j-1} + \phi_{k2}\rho_{j-2} + \ldots + \phi_{kk}\rho_{j-k}, \quad j = 1, 2, \ldots k \tag{10.67}$$

Here we have used the notation ϕ_{kj} for the ϕ_j coefficient in an autoregressive model of order k.

These **Yule-Walker equations** can be written in matrix form as

$$\begin{bmatrix} 1 & \rho_1 & \rho_2 & \cdots & \rho_{k-1} \\ \rho_1 & 1 & \rho_1 & \cdots & \rho_{k-2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \rho_{k-1} & \rho_{k-2} & \cdots & \rho_1 & 1 \end{bmatrix} \begin{bmatrix} \phi_{k1} \\ \phi_{k2} \\ \cdots \\ \phi_{kk} \end{bmatrix} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \cdots \\ \rho_k \end{bmatrix} \tag{10.68}$$

or

$$P_k \phi_{\mathbf{k}} = \rho_{\mathbf{k}}. \tag{10.69}$$

In general, there will be nonzero autocorrelations at lags greater than k, and this system of equations doesn't help us determining those autocorrelations.

Example 10.12

Recall the AR(2) process

$$Z_n = A_n + \phi_1 Z_{n-1} + \phi_2 Z_{n-2} \tag{10.70}$$

The Yule-Walker equations are

$$\begin{bmatrix} 1 & \rho_1 \\ \rho_1 & 1 \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = \begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix} \tag{10.71}$$

Solving these equations, we obtain

$$\rho_1 = \frac{\phi_1}{1 - \phi_2} \tag{10.72}$$

and

$$\rho_2 = \frac{\phi_1^2}{1 - \phi_2} + \phi_2 \tag{10.73}$$

which matches our earlier calculation.

The Yule-Walker equations can be used in two important ways. If we know the coefficients ϕ_{k1} through ϕ_{kk}, then we can use the equations to compute the autocorrelations ρ_1 through ρ_k. Conversely, if we know (or can estimate) the autocorrelations, we can solve the equations to obtain estimates of the coefficients ϕ_{k1}, ..., ϕ_{kk}.

The **partial autocorrelation function (PACF)** associated with a sequence z_n consists of the sequence $\hat{\phi}_{11}$, $\hat{\phi}_{22}$, $\hat{\phi}_{33}$, ... of partial autocorrelations estimated from the sequence z_n. This sequence can be obtained by estimating the autocorrelations, inserting the autocorrelations into the Yule-Walker equations (10.68), and then solving the Yule-Walker equations for $k = 1$, $k = 2$, ...

In practice, a recursive formula due to Durbin is more efficient. The Durbin formula is

$$\hat{\phi}_{p+1,j} = \hat{\phi}_{p,j} - \hat{\phi}_{p+1,p+1} \hat{\phi}_{p,p-j+1} \tag{10.74}$$

$$\hat{\phi}_{p+1,p+1} = \frac{r_{p+1} - \sum_{j=1}^{p} \hat{\phi}_{p,j} r_{p+1-j}}{1 - \sum_{j=1}^{p} \hat{\phi}_{p,j} r_j} \tag{10.75}$$

where

$$\hat{\phi}_{1,1} = r_1 \tag{10.76}$$

The PACF is very useful in identifying an autoregressive process. If our original process is autoregressive of order p, then for $k > p$, we should have $\hat{\phi}_{kk} = 0$. This provides a very useful test for whether or not a process is autoregressive. Of course, we need to know when the $\hat{\phi}_{kk}$ are effectively zero. It can be shown that the variance of $\hat{\phi}_{kk}$ is approximately $1/n$ when we have n points from an AR(p) process and $k \geq p + 1$.

The PACF also turns out to be important in forecasting. It can be shown that the best (least squares) predictor of z_n using the $k-1$ previous values z_{n-1}, z_{n-2}, ..., z_{n-k+1} is

$$z_n = \phi_{k-1,1} z_{n-1} + \phi_{k-1,2} z_{n-2} + \ldots + \phi_{k-1,k-1} z_{n-k+1} \tag{10.77}$$

ARMA Modeling in Practice

Now that we understand the theoretical behavior of ARMA processes, we will consider how to take an actual observed time series, fit an ARMA model to the data, and forecast future values of the time series.

The stages in our process for ARMA modeling a time series beginning with observed values z_1, z_2, ..., z_n are:

1. Remove any nonzero mean from the time series.
2. Estimate the autocorrelation and PACF of the time series. Use these to determine the autoregressive order p and the moving average order q.
3. Estimate the coefficients ϕ_1, ..., ϕ_p, θ_1, ..., θ_q.
4. Estimate a_1, a_2, ..., a_n.
5. Use the fitted model to forecast z_{n+1}, z_{n+2}, ...

The first step in our process is removing any nonzero mean from the time series z. This is a very straight forward computation- just compute the mean of the time series, and subtract it from each element of the time series.

The next step in the process is determining the autoregressive order p and the moving average order q. Table 1 (taken from BJR) summarizes the behavior of ARMA(p,q) processes for $p = 0, 1, 2$ and $q = 0, 1, 2$. If the observed autocorrelation and PACF match up with one of the lines in this table, then it's reasonable to fit a model of that type.

Example 10.13

Order	ρ_k	ϕ_{kk}
(1,0)	exp decay	only ϕ_{11} nonzero
(0,1)	only ρ_1 nonzero	exp decay
(2,0)	exp or damped sine wave	only ϕ_{11}, ϕ_{22} nonzero
(0,2)	only $\rho_{1,2}$ nonzero	exp or damped sine wave
(1,1)	exp decay	exp decay

Table 10.3: Rules for selecting p and q.

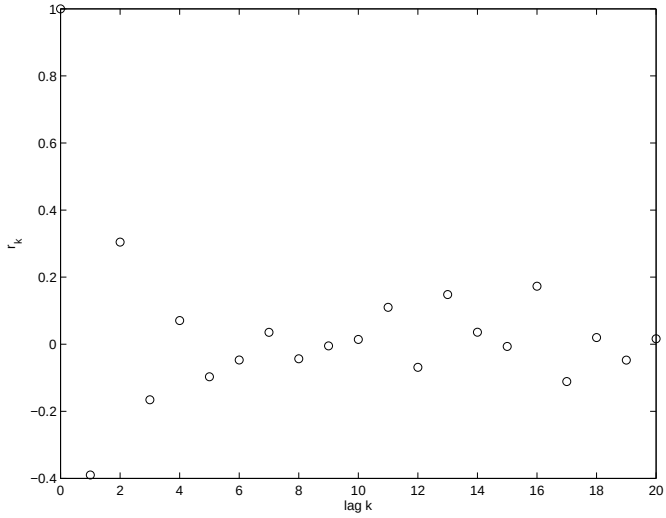

Figure 10.2: Autocorrelations for the batch data.

Recall the time series of yields from a batch chemical process that we previously analyzed. Figure 10.2 shows the autocorrelations for this data. The autocorrelations seem to follow an exponentially damped sine wave, but they quickly hit a noise level beyond a lag of four or five.

We can also estimate the PACF for this data. We get $\hat{\phi}_{1,1} = -0.3889$, $\hat{\phi}_{2,2} = 0.1797$, $\hat{\phi}_{3,3} = 0.0023$, $\hat{\phi}_{4,4} = -0.0443$, ... In this case, since $n = 70$, we'd expect the standard deviation of the estimates to be about $1/\sqrt{70} = 0.12$ once we get out past a lag of p. Thus it appears that only the first two coefficients $\hat{\phi}_{1,1}$ and $\hat{\phi}_{2,2}$ are definitely nonzero.

The autocorrelation and PACF suggests an AR(2) model for this data set.

ARMA(p,q)	ρ_1	ρ_2
(1,0)	$\rho_1 = \phi_1$	
(0,1)	$\rho_1 = -\theta_1/(1+\theta_1^2)$	
(2,0)	$\rho_1 = \phi_1/(1-\phi_2)$	$\rho_2 = (\phi_1^2)/(1-\phi_2)+\phi_2$
(0,2)	$\rho_1 = -\theta_1(1-\theta_2)/(1+\theta_1^2+\theta_2^2)$	$\rho_2 = -\theta_2/(+\theta_1^2+\theta_2^2)$
(1,1)	$\rho_1 = (1-\theta_1\phi_1)(\phi_1-\theta_1)/(1+\theta_1^2-2\phi_1\theta_1)$	$\rho_2 = \rho_1\phi_1$

Table 10.4: Equations to be solved for ϕ and θ.

Once we've determined p and q, the next step is to estimate the actual parameters $\phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q$. One very simple approach can be used if we have formulas for the autocorrelations in terms of the parameters. For example, for an AR(2) process, we know that

$$\rho_1 = \frac{\phi_1}{1-\phi_2} \tag{10.78}$$

and

$$\rho_2 = \frac{\phi_1^2}{1-\phi_2} + \phi_2. \tag{10.79}$$

These equations can be solved for ϕ_1 and ϕ_2 to get

$$\phi_1 = \frac{\rho_1(\rho_2-1)}{\rho_1^2-1} \tag{10.80}$$

and

$$\phi_2 = \frac{\rho_1^2-\rho_2}{\rho_1^2-1}. \tag{10.81}$$

If we use our estimates r_1 and r_2, we can obtain estimates of ϕ_1 and ϕ_2.

A similar approach can be used to estimate the parameters of other low order ARMA models. Table 2 summarizes the equations to be solved for the (1,0), (0,1), (2,0), (0,2) and (1,1) cases.

A more sophisticated approach is to use maximum likelihood estimation to obtain the parameters. Unfortunately, there aren't any functions for this in the MATLAB toolboxes available at NMT. However, this can be done with more sophisticated statistical packages such as Minitab and R.

Example 10.14

Continuing with the batch process data, using (10.80) and (10.81), we estimate that

$$\phi_1 = -0.3198 \tag{10.82}$$

and

$$\phi_2 = 0.1797. \tag{10.83}$$

Suppose that we are now at time n, we've found p and q, and fitted the parameters $\phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q$. Now we want to predict the observations at times $n+1, n+2, \ldots, n+l$. We will use the notation z_{n-k} for the known observations up to time n. As before, we will use Z_{n+k} for unknown (random) future values of the time series. We will use $\hat{z}_n(l)$ for the predicted observation at time $n+l$ based on observations through time n.

In making a forecast $\hat{z}_n(l)$, we want to minimize the expected value of the square of the error in the forecast.

$$\min E[(Z_{n+l} - \hat{z}_n(l))^2] \tag{10.84}$$

where this expected value is conditioned on all of the observations through time n. It can be shown that picking $\hat{z}_n(l) = E[Z_{n+l}]$ minimizes the expected value of the squared error. Using the $\psi()$ function, we know that

$$Z_{n+l} = \sum_{j=0}^{\infty} \psi_j A_{n+l-j} = A_{n+l} + \psi_1 A_{n+l-1} + \ldots \tag{10.85}$$

This infinite sum contains some terms which correspond to times up to time n and other terms which lie in the future and are still random. To get the expected value of Z_{n+l}, we take the expected value of each term on the right hand side. For A_{n+l} and other future inputs from the white noise, this expected value is 0. For A_n and other past white noise inputs, the expected value of A_{n-k} is the actual value a_{n-k} that was observed. Thus our prediction is given by

$$\hat{z}(l) = \sum_{j=l}^{\infty} \psi_j a_{n+l-j}. \tag{10.86}$$

The random error associated with our forecast is

$$e_n(l) = A_{n+l} + \psi_1 A_{n+l-1} + \ldots + \psi_{l-1} A_{n+1}. \tag{10.87}$$

Clearly, the expected value of $e_n(l)$ is 0. Furthermore, we can work out the variance associated with our prediction.

$$Var(e_n(l)) = Var(A_{n+l}) + \psi_1^2 Var(A_{n+l-1}) + \ldots + \psi_{l-1}^2 Var(A_{n+1}) \tag{10.88}$$

$$Var(e_n(l)) = \sigma_A^2(1 + \psi_1^2 + \ldots + \psi_{l-1}^2). \tag{10.89}$$

There are three important practical issues that we need to resolve before we can actually start computing forecasts. The first problem is that we have a time series z_k, but not the corresponding a_k series. To compute the a sequence, notice that

$$e_n(1) = A_{n+1} \tag{10.90}$$

Thus

$$z_n - \hat{z}_{n-1}(1) = a_n. \tag{10.91}$$

We can use this to compute the values a_k for $k \leq n$. Just compute the lag 1 predictions, and subtract them from the actual values. In doing this, we may have to refer to z_k and a_k values from before the start of our observations. Set these to 0. In practice, the 0 initial conditions will have little effect on the forecasts.

The second issue is that we may not know σ_A^2. In this case, we use the sample variance of the a values that we have computed as an estimate for σ_A^2.

The third issue is that evaluating the infinite sum

$$\hat{z}(l) = \sum_{j=l}^{\infty} \psi_j a_{n+l-j}. \tag{10.92}$$

may be impractical. If the ψ_j weights decay rapidly, we can safely truncate the series, but if the ψ_j weights decay slowly this may be impractical. Fortunately, it is also possible to use the two other main forms of the model

$$Z_n = A_n + \pi_1 Z_{n-1} + \pi_2 Z_{n-2} + \ldots \tag{10.93}$$

or

$$\phi(B) Z_n = \theta(B) A_n \tag{10.94}$$

for forecasting. If the model is purely autoregressive, then the π weights are the way to go. If the model is purely moving average, then it's best to use the ψ weights. For mixed models, the form $\phi(B)Z_n = \theta(B)A_n$ is usually the easiest to work with.

In making a forecast using any of the three forms of the model, we use the same basic idea. We start by computing the previous a_k values. Next, we substitute observed or expected values for all terms in the model to get $\hat{z}_n(l)$. The expected values of all future a_{n+k} values are 0. The expected values of future z_{n+k} values are given by our predictions $\hat{z}_n(k)$, $k = 1, 2, \ldots$. We compute $\hat{z}_n(1), \hat{z}_n(2), \ldots, \hat{z}_n(l)$, and then use the variance formula to get confidence intervals for our predictions.

Example 10.15

Figure 10.3 shows next five points predicted from the batch data. The general pattern of alternating high and low values is predicted to continue, but the error bars on these predictions are quite broad. The problem is that the original data is quite noisy. The estimated value of σ_A is 10.7 which is about 20% of the typical data points of around 50.

ARMA modeling and a slightly more sophisticated variation called ARIMA modeling are very widely used in time series forecasting. The technique is also known as Box-Jenkins forecasting after its inventors. When a model can be found that fits the data well and σ_A is relatively small, it can provide very good

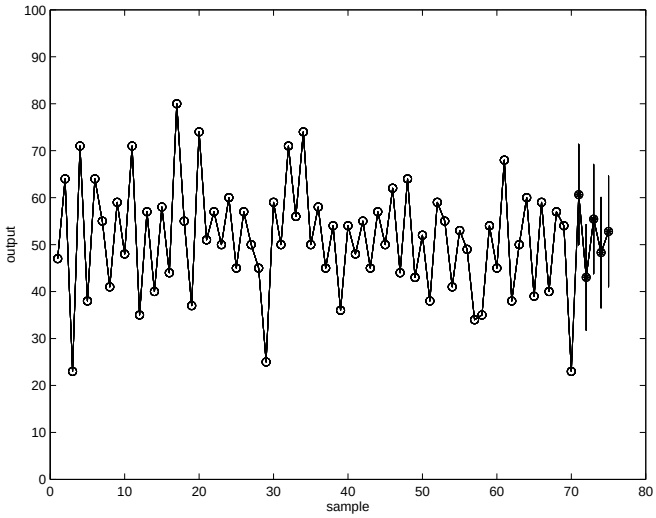

Figure 10.3: Predicted data points for the batch data.

predications. A huge advantage of this approach is that it produces error barsmany other simple forecasting schemes do not provide any indication of the uncertainty of the predictions. However, in some cases ARIMA modeling can fail, either because the underlying dynamics of the time series are too complicated to be captured by a simple ARIMA model, or because the noise level σ_A is simply too large.

Appendix A

Discrete Approximation of a Convolution

This note discusses how to approximate a continuous convolution with a discrete convolution, and how MATLAB can easily be used to compute this approximation. MATLAB works with vectors and arrays of numbers, not continuous functions, so it is essential to develop a familiarity for moving between continuous and discrete methods to apply MATLAB to simulating physical systems and solving problems.

We start by selecting a sampling interval- a period of time which is short relative to the phenomenon that we're interested in (we will make this concept much more quantitative once we discuss Fourier theory and the Nyquist theorem in Chapter 4). For example, if we're working with a function that varies over a period of several seconds, then a sampling interval of $\Delta t = 0.01$ seconds will probably provide adequately dense sampling.

For each sampling interval, we select an "average" value to assign to the associated sample. This might be the true average over the interval, or the function value at the midpoint of the interval, or the value at some other reasonable point in the interval. In the following we'll we'll use times

$$t_j = t_0 + j\Delta t \;\; j = -\infty \ldots \infty \tag{A.1}$$

as the midpoints of the intervals, and evaluate or "sample" the function at these times.

$$x_j = x(t_j) \;\; j = -\infty \ldots \infty. \tag{A.2}$$

Here we have adopted the convention of using $x(t)$ for the function and x_j for the discrete approximation of the function. This dual use of x should not cause problems as long as we remember that $x_j = x(t_j)$. See Figure A.1.

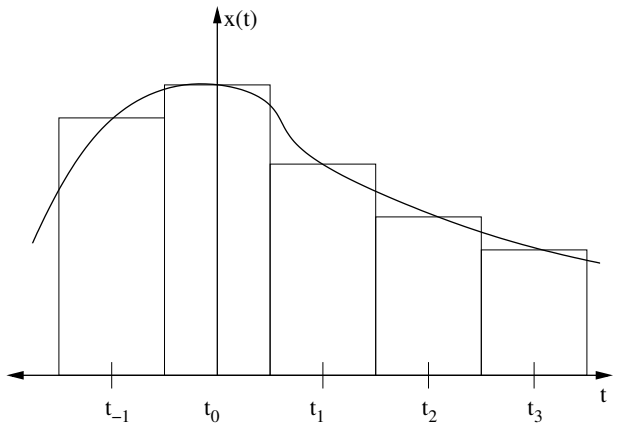

Figure A.1: Discretization of $x(t)$.

Notice that since we cannot store vectors of infinite length in a computer, we cannot approximate functions which are nonzero everywhere. Typically, it is possible to use enough sampling intervals to cover the portion of $x(t)$ that we're interested in. Also note that although it is convenient in writing our equations to use zero or negative indices on x for times before t_1, MATLAB does not allow for array indices that are less than one. Thus we may need to shift our indices.

For example, the following MATLAB code represents the function $x(t) = \sin(2\pi ft)$, between $t = -1$ and $t = 1$ seconds using 100 sampling intervals of width $\Delta t = 0.02$ seconds and stores the result in the vector x. The factor of 2π in this function converts the frequency f in cycles per second into radians per second. Note that the MATLAB sin function does not work with arguments in degrees! For our example, we'll use a frequency of $f = 5$ cycles per second. The function $x(t)$ is evaluated at the midpoint of each sampling interval, at times -0.99, -0.97, ..., 0.99. For convenience, we also store the mid points of the time intervals in the vector t.

```
deltat=0.02;
f=5;
for j=1:100,
  t(j)=-1.01+j*deltat;
  x(j)=sin(2*pi*f*t(j));
end;
```

MATLAB's colon notation and vectorized evaluation of functions can be used to simplify and greatly speed up the above code. The following code produces exactly the same results without the use of a for loop.

```
deltat=0.02;
f=5;
t=-0.99:deltat:0.99;
x=sin(2*pi*f*t);
```

Once we've generated x, we can plot $x(t)$ with

```
figure(1);
plot(t,x,'k');
xlabel('t');
ylabel('x(t)');
print -deps signal.eps
```

The signal is shown in Figure A.2. Notice that MATLAB has interpolated between the points with straight lines. Since Δt is 0.02 seconds, this piecewise linear interpolation is not perfectly smooth, and there are places where you can see this in the plot.

Integrating a function represented in sampled fashion is straightforward by using rectangular strips to approximate the area under the function during each sampling interval. We then integrate $x(t)$ by adding up the area under $x(t)$ across all sampling intervals. The following bit of MATLAB code integrates $x(t)$ from -1 to 1 using the x vector that we just generated.

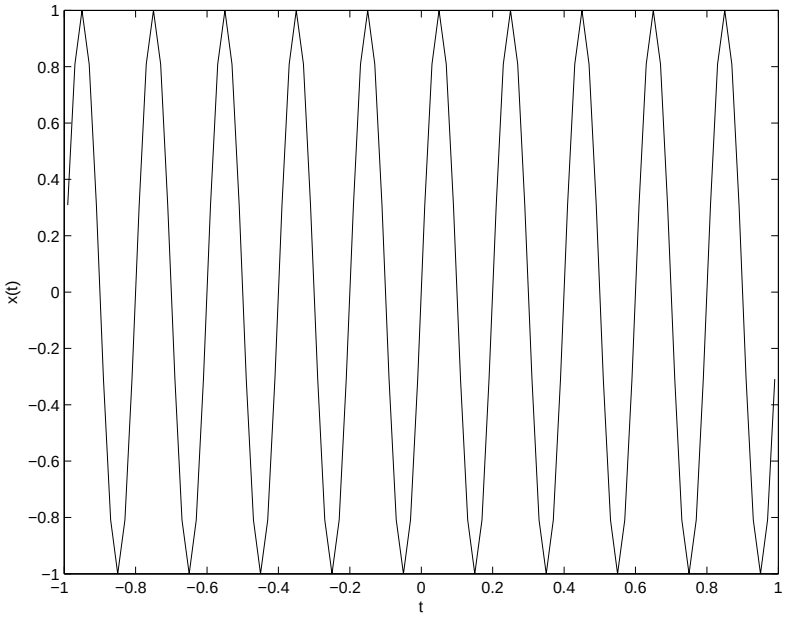

Figure A.2: $x(t)$.

```
deltat=0.02;
s=0.0;
for j=1:100,
  s=s+x(j)*deltat;
end;
```

Again, it's possible to greatly simplify this code to

```
deltat=0.02;
s=sum(x)*deltat;
```

Notice that the length of the sampling interval Δt plays an important part in this formula.

Infinite sequences can be convolved in a fashion very similar to convolution of functions in time. The convolution of the sequence x_i with the sequence y_j is defined as

$$z_k = \sum_{j=-\infty}^{\infty} x_j y_{k-j}. \tag{A.3}$$

In practice, we don't store vectors of infinite length, but rather treat all of the entries outside of our finite length vectors as if they were 0.

The MATLAB command conv implements a discrete convolution of two finite length vectors. Given a vector x of length r, and a vector y of length s,

the convolution operation produces a vector z of length $r + s - 1$. Since x has entries 1 through r, and y has entries 1 through s, the nonzero entries in the convolution should be in positions 2 through $r + s$. However, since MATLAB arrays always start with index 1 (sorry, c programmers), the entries are shifted one place to the left.

For example, suppose that $x_1 = 1$ and $x_2 = 2$ and all other entries of x are zero. Also suppose that $y_1 = 3$, and $y_2 = 4$, and all other entries of y are 0. In the convolution, using (A.3), we get that $z_2 = 3$, $z_3 = 10$, $z_4 = 8$, and all other entries in z are 0. In MATLAB, we would have $x = [1\ \ 2]$, $y = [3\ \ 4]$, and $z = [3\ \ 10\ \ 8]$.

Now, how can we compute the discrete approximation to a continuous convolution? Recall that we can find $z(t) = x(t) * y(t)$ by evaluating the integral

$$z(t) = \int_{-\infty}^{\infty} x(\tau)y(t - \tau)d\tau. \tag{A.4}$$

We will approximate this integral by setting up intervals of length $\Delta\tau$, and evaluating the functions at times τ_j that are the mid points of each interval. For convenience, we will use the same discretization for t and τ, so that $t_j = \tau_j$.

$$z(t_k) = \int_{-\infty}^{\infty} x(\tau)y(t_k - \tau)d\tau. \tag{A.5}$$

Approximating the integral by the rectangle rule, we get

$$z_k = \sum_{j=-\infty}^{\infty} x(\tau_j)y(t_k - \tau_j)\Delta\tau. \tag{A.6}$$

Because the τ_j's are evenly spaced, $t_k - \tau_j = (k-j)\Delta\tau$ and $y(t_k - \tau_j) = y(t_{k-j})$. Thus

$$z_k = \sum_{j=-\infty}^{\infty} x_j y_{k-j}\Delta\tau. \tag{A.7}$$

This is just the discrete convolution of x and y defined in (A.3) multiplied by a scaling factor of $\Delta\tau$. So, to approximate $z(t) = x(t) * y(t)$, start by representing the nonzero parts of $x(t)$ and $y(t)$ by vectors x and y. Next, compute $z = \text{conv}(x, y)$. Finally, scale z by $\Delta\tau$. It's unfortunate that MATLAB doesn't multiply by $\Delta\tau$ inside the conv function. However, the sampling interval $\Delta\tau$ is not stored with the vectors x and y so it simply isn't available to the conv function.

Continuing our example, let's convolve $x(t)$ with an impulse response $y(t) = H(t)e^{-5t}$. Note that this function is nonzero for all $t > 0$. However, by $t = 1$ second, $y(t)$ is nearly 0. So, we'll truncate y at one second. We compute the discrete convolution of x and y and then scale by Δt to get an approximation to $z(t)$.

```
%
% First, setup y, using 50 points.
%
deltat=0.02;
ty=0.01:deltat:0.99;
y=exp(-5*ty);
%
% Do the convolution.
%
z=conv(x,y)*deltat;
%
% Figure out the time points for z.
%
for j=1:length(x)+length(y)-1
  tz(j)=-1.01+j*deltat;
end;
%
% Now, plot the result.
%
plot(tz,z,'k');
```

The result is shown in Figure A.3.

We can also compute discrete approximations to the cross correlation of two functions or the autocorrelation of a function. For discrete sequences x_j and y_j, the cross correlation is defined by

$$z_k = \sum_{j=-\infty}^{\infty} x_j y_{j-k}. \tag{A.8}$$

The MATALB function xcorr(x, y) takes input vectors x and y and computes the discrete cross correlation. Again, if x and y were sampled at an interval of Δt, then the discrete cross correlation must be scaled by Δt. The xcorr function can also be used compute the autocorrelation of a sequence x.

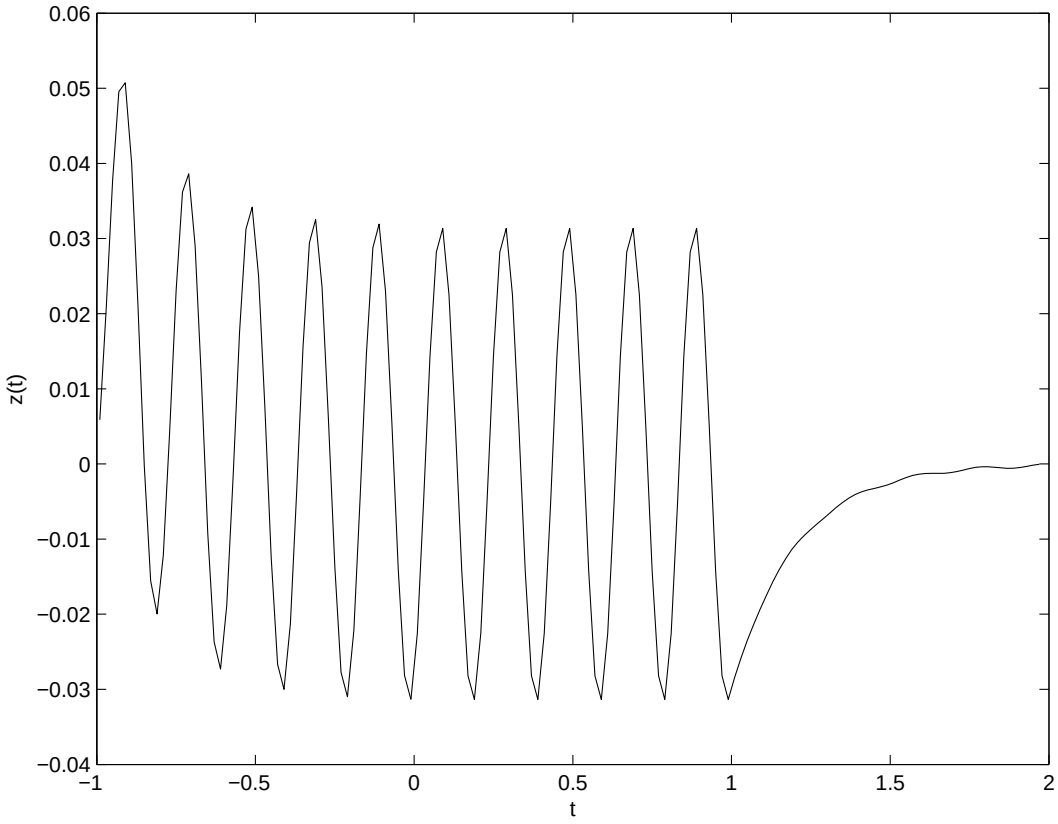

Figure A.3: Convolution of x and y.

Appendix B

Primer on Complex Numbers and Arithmetic

These notes summarize some important facts about complex numbers and their arithmetic.

Rectangular Form

If we define $i = \sqrt{-1}$, then we can construct a system of complex numbers of the form $z = a + bi$. In this system,

$$(a + bi) + (c + di) = (a + c) + (b + d)i \tag{B.1}$$

$$(a + bi) - (c + di) = (a - c) + (b - d)i \tag{B.2}$$

$$(a + bi)(c + di) = ac + bci + adi + bdi^2 = (ac - bd) + (bc + ad)i \tag{B.3}$$

$$\frac{a + bi}{c + di} = \frac{(a + bi)(c - di)}{(c + di)(c - di)} = \frac{(ac + bd) + (bc - ad)i}{c^2 + d^2} \tag{B.4}$$

The *complex conjugate* of a complex number is given by

$$(a + bi)^* = \overline{a + bi} = a - bi. \tag{B.5}$$

Euler's Formula

Using the power series

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots \tag{B.6}$$

we can derive a formula for e raised to an imaginary power.

$$e^{ix} = 1 + ix + \frac{(ix)^2}{2!} + \frac{(ix)^3}{3!} + \dots$$

Using the facts that $i^2 = -1$, $i^3 = -i$, and $i^4 = 1$, we find that

$$e^{ix} = 1 + ix - \frac{x^2}{2!} - i\frac{x^3}{3!} + \frac{x^4}{4!} + \dots$$

Rearranging the terms, we get

$$e^{ix} = (1 - \frac{x^2}{2!} + \frac{x^4}{4!} + \dots) + i(x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots)$$

Finally, using the Taylor's series for sin and cos, we get

$$e^{ix} = \cos(x) + i\sin(x) \tag{B.7}$$

For general complex numbers $a + bi$, we find that

$$e^{a+bi} = e^a(\cos(b) + i\sin(b)) \tag{B.8}$$

Polar Form

Using Euler's formula, we can take any complex number

$$z = a + bi$$

and rewrite it as

$$z = Re^{i\theta} \tag{B.9}$$

where

$$R = |z| = \sqrt{z^* z} = \sqrt{a^2 + b^2} \tag{B.10}$$

is variously called the *amplitude*, *modulus*, or *complex norm* and

$$\theta = \angle z = \tan^{-1}\frac{b}{a}. \tag{B.11}$$

is variously called the *complex angle*, *phase* or *argument* of z. Because sin and cos are 2π periodic, we can add any multiple of 2π to the phase of a complex number without changing its value.

We can also go the other way. If

$$z = Re^{i\theta}$$

then $z = a + bi$, where

$$a = R\cos(\theta) \tag{B.12}$$

and

$$b = R\sin(\theta). \tag{B.13}$$

Polar form is very useful for multiplication, division, and exponentiation, but hopeless for addition and subtraction.

$$Ae^{i\theta}Be^{i\phi} = ABe^{\theta+\phi} \tag{B.14}$$

$$\frac{Ae^{i\theta}}{Be^{i\phi}} = \frac{A}{B}e^{\theta-\phi} \tag{B.15}$$

$$(Ae^{i\theta})^x = (A^x)e^{ix\theta}. \tag{B.16}$$

It's also easy to find the complex conjugate of a number in polar form.

$$(Ae^{i\theta})^* = Ae^{-i\theta}. \tag{B.17}$$

Cosine and Sine in terms of complex exponentials

Using Euler's formula, it's easy to derive formulas for sin and cos in terms of complex exponentials.

$$\cos(\theta) = \frac{e^{i\theta} + e^{-i\theta}}{2}. \tag{B.18}$$

$$\sin(\theta) = \frac{e^{i\theta} - e^{-i\theta}}{2i}. \tag{B.19}$$

MATLAB and Complex Numbers

When you first start MATLAB, The variable i is set equal to $\sqrt{-1}$. However, if you change the value of i (for example by using it as the index in a for loop!), then it will no longer have this value. Thus it is a good idea to avoid using i as a loop index.

Nearly all of the functions that are built into MATLAB operate correctly on complex numbers. Thus you can add, subtract, multiply, and divide complex numbers. You can also compute exponentials, logs, sines, cosines, and other functions of complex numbers. MATLAB has several useful functions for manipulating complex numbers. The conj function computes the complex conjugate of a number. The abs function computes the absolute value of a complex number. The angle function computes the phase angle of a complex number.

How LTI's operate on complex exponentials, sines, and cosines

A linear time invariant (LTI) system operates in a simple fashion when a complex exponential, sine, or cosine is input to the system. Recall that if $\phi(t)$ is the impulse response of the system (that is, the response of the system when the input $\delta(t)$), then response to an input function, $x(t)$, is given by the convolution of $x(t)$ with the impulse response.

$$y(t) = \phi(t) * x(t) \tag{B.20}$$

or

$$y(t) = \int_{-\infty}^{\infty} \phi(\tau)x(t-\tau)d\tau. \tag{B.21}$$

Consider the special case where

$$x(t) = e^{i2\pi ft}. \tag{B.22}$$

In this case,

$$y(t) = \int_{-\infty}^{\infty} \phi(\tau) e^{i2\pi f(t-\tau)} d\tau. \tag{B.23}$$

$$y(t) = e^{i2\pi ft} \int_{-\infty}^{\infty} \phi(\tau) e^{-i2\pi f\tau} d\tau. \tag{B.24}$$

Notice that this integral depends only on f, and not t. We can define

$$\Phi(f) = \int_{-\infty}^{\infty} \phi(\tau) e^{-i2\pi f\tau} d\tau. \tag{B.25}$$

Where $\Phi(f)$ is the *Fourier Transform* of $\phi(t)$. Then

$$y(t) = e^{i2\pi ft}\Phi(f). \tag{B.26}$$

Since $\Phi(f)$ is just a complex number, we can write it in exponential form as $\Phi(f) = A(f)e^{i\theta(f)}$, where $A(f)$ and $\theta(f)$ are real numbers. Now, we can write $y(t)$ as

$$y(t) = e^{i2\pi ft}A(f)e^{i\theta(f)}. \tag{B.27}$$

This says that if we use a complex exponential signal as the input to our LTI system, we'll get a complex exponential signal as the output. The factor $A(f)$ amplifies or attenuates the signal, and the factor $e^{i\theta(f)}$ shifts the phase of the signal.

What if $x(t) = \cos(2\pi ft)$ where f and t are a real frequency and time? We can use (B.18) to write the cosine in terms of complex exponentials.

$$x(t) = \frac{e^{i2\pi ft} + e^{-i2\pi ft}}{2}. \tag{B.28}$$

Then using the principles of scaling and superposition and (B.26), we get that

$$y(t) = \frac{e^{i2\pi ft}\Phi(f)}{2} + \frac{e^{-i2\pi ft}\Phi(-f)}{2}. \tag{B.29}$$

It can be shown that if $\phi(t)$ is real, then

$$\Phi(-f) = \Phi(f)^* \tag{B.30}$$

To see this, simply take the complex conjugate of $\Phi(f)$.

$$\Phi(f)^* = \left(\int_{-\infty}^{\infty} \phi(\tau) e^{-i2\pi f\tau} d\tau.\right)^* \tag{B.31}$$

$$\Phi(f)^* = \int_{-\infty}^{\infty} \left(\phi(\tau) e^{-i2\pi f\tau}\right)^* d\tau. \tag{B.32}$$

$$\Phi(f)^* = \int_{-\infty}^{\infty} \phi(\tau) e^{+i2\pi f\tau} d\tau. \tag{B.33}$$

$$\Phi(f)^* = \int_{-\infty}^{\infty} \phi(\tau) e^{-i2\pi(-f)\tau} d\tau. \tag{B.34}$$

$$\Phi(f)^* = \Phi(-f). \tag{B.35}$$

Equivalently, this says that $\Phi(f)$ for a real-valued impulse response $\phi(t)$ has even symmetry for its real part, and odd symmetry for its imaginary part (relative to $f = 0$.)

Since $\Phi(f) = A(f)e^{i\theta(f)}$, and $\Phi(-f) = \Phi(f)^*$, $\Phi(-f) = A(f)e^{-i\theta(f)}$. Thus

$$y(t) = \frac{e^{i2\pi ft} A(f) e^{i\theta(f)}}{2} + \frac{e^{-i2\pi ft} A(f) e^{-i\theta(f)}}{2}. \tag{B.36}$$

$$y(t) = \frac{e^{i(2\pi ft + \theta(f))} + e^{-i(2\pi ft + \theta(f))}}{2} A(f). \tag{B.37}$$

$$y(t) = \cos(2\pi ft + \theta(f)) A(f). \tag{B.38}$$

This shows that the output of the LTI system with a real-valued $\phi(t)$ and a cosine input is also a cosine, but with its magnitude scaled by the amplitude, $A(f)$, and shifted in phase by the angle $\theta(f)$. Similarly, it's easy to show that if the input is a sine wave with frequency f, then the output will be a sine wave scaled by the amplitude $A(f)$ and shifted in phase by the angle $\theta(f)$.

Appendix C

Finding an Impulse Response via Contour Integration

In the Chapter 2 notes, we noted that the time domain displacement response to an acceleration impulse input is

$$\phi(t) = F^{-1}\left(\frac{1}{\omega^2 - 2\imath\zeta\omega - \omega_s^2}\right) \tag{C.1}$$

$$= \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{e^{\imath\omega t}\, d\omega}{\omega^2 - 2\imath\zeta\omega - \omega_s^2} = \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{e^{\imath\omega t}\, d\omega}{(\omega - \omega_1 - \imath\zeta)(\omega + \omega_1 - \imath\zeta)} \tag{C.2}$$

where

$$\omega_1 = \sqrt{\omega_s^2 - \zeta^2}\ . \tag{C.3}$$

To solve (C.2), we utilize a remarkable and useful theorem from complex analysis, the *residue theorem*. Succinctly stated, the residue theorem says that, for a complex function in the complex plane that is defined and differentiable with a region except at an isolated singularity at a finite point z_0 (e.g., a pole in a transfer function), then, for a closed path, or contour, C, encompassing the singularity

$$\oint_C f(z)dz = 2\pi\imath a \tag{C.4}$$

where a is called the *residue* of $f(z)$ at z_0, where, for a pole of order m,

$$a = \frac{1}{(m-1)!}\frac{d^{m-1}}{dz^{m-1}}[(z - z_0)^m f(z)]_{z=z_0}\ . \tag{C.5}$$

for a single pole at z_0, the residue is simply

$$a = [(z - z_0)f(z)]_{z=z_0}\ . \tag{C.6}$$

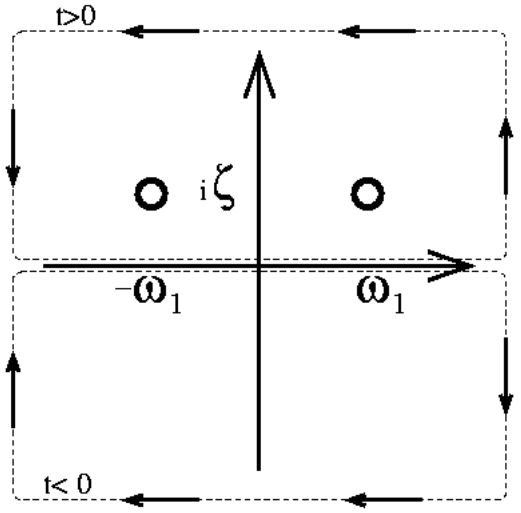

Figure C.1: Contour integration in the complex ω plane.

Evaluation of a contour integral in the complex plane thus involves evaluating the integrand at $z = z_0$ with the pole "removed" by first multiplying by the factor $(z - z_0)$ More generally, if more than one distinct pole is enclosed by the integration path

$$\oint_C f(z)dz = 2\pi\imath \sum_i a_i \tag{C.7}$$

where the a_i are the residues at the encosed poles. If there are no poles enclosed by C, the integral will be zero (this is the same as saying that the function is *conservative*, or that the integral of $f(z)$ between two complex points doesn't depend on the integration path).

We can now use the residue theorem to evaluate the inverse Fourier transform (C.2). The poles of the integrand lie at $(\pm\omega_1, \imath\zeta)$. We conceptualize the inverse Fourier transform as a contour integration by integrating in the complex ω plane along the ω axis from $-\infty$ to ∞, and then closing the countour at $|z = \infty|$ (where the value of the integrand is zero). For $t < 0$ the contour is clockwise because of the $e^{\imath\omega t}$ factor and encompasses no poles (Figure C.1). Thus

$$\phi(t) = 0 \ (t < 0) \ . \tag{C.8}$$

For $t > 0$ the contour is clockwise and encompases poles, so that the residue theorem gives

$$\phi(t) = \frac{\imath}{\omega_1}\left(\frac{e^{-\imath\omega_1 t}}{-2} + \frac{e^{\imath\omega_1 t}}{2}\right) e^{-\zeta t} \ . \tag{C.9}$$

For the underdamped case, where $\omega_2 > \zeta$, ω_1 is real, so that (setting the function to be zero for $t < 0$ with a step function) we have the impulse response

$$\phi_{underdamped} = \mathrm{H}(t)\frac{-1}{\omega_1} e^{-\zeta t}\sin(\omega_1 t) \ . \tag{C.10}$$

For the overdamped case, where $\omega_s < \zeta$, $\omega_1 = \imath\sqrt{\zeta^2 - \omega_s^2}$, and the poles lie on the negative real axis at $-\zeta \pm \sqrt{\zeta^2 - \omega_s^2}$. The impulse response function in this case can be written entirely with real exponentials as

$$\phi_{overdamped}(t) = \frac{-\mathrm{H}(t)}{2(\zeta^2 - \omega_s^2)^{1/2}}\left(e^{-(\zeta - (\zeta^2 - \omega_s^2)^{1/2})t} - e^{-(\zeta + (\zeta^2 + \omega_s^2)^{1/2})t}\right) \ . \tag{C.11}$$

For the critically damped case, $\omega_1 = 0$, and we have a repeated (order 2) pole at $\omega = \imath\zeta$. Application of (C.5) for $t > 0$ gives

$$\phi_{critical}(t) = \imath\frac{d}{d\omega}\mathrm{H}(t)e^{\imath\omega t}|_{\omega=\imath\zeta} = -\mathrm{H}(t)te^{-\zeta t} \ . \tag{C.12}$$

Appendix D

Plotting Spectra Using Decibels

Because the amplitudes describing the spectral responses of physical systems in nature as well as filters and instruments, frequently span many orders of magnitude, amplitude responses are frequently plotted as a function of frequency using either as log-linear or log-log displays. The standard way to do this is using a *decibel* (dB) scale. The 'Bel' was originally a unit of sound intensity, after Alexander Graham Bell- a decibel is one tenth of a Bel.

The decibel relationship between two power levels is defined as

$$d = 10 \log_{10} \frac{P_1}{P_2} \tag{D.1}$$

In examining system responses, P_1/P_2 in (D.1) is commonly the ratio of output power level over input power level so that 0 dB corresponds to unit gain and amplification by a factor greater than one corresponds to $d > 0$.

Note that the definition we have given is for power ratios. In practice it is often necessary to consider voltage ratios as well. Assuming that the power is dissipated by a load of constant resistance R, the power is $P = V^2/R$, so

$$d = 10 \log_{10} \frac{P_1}{P_2} = 10 \log_{10} \frac{V_1^2/R}{V_2^2/R} = 10 \log_{10} \frac{V_1^2}{V_2^2} = 10 \log_{10} \left(\frac{V_1}{V_2} \right)^2 = 20 \log_{10} \frac{V_1}{V_2}.$$

Similarly, in seismic work power is proportional to velocity squared, so the factor of 20 is used. In general, the factor of 20 is used with amplitudes and the factor of 10 is used with power. An amplitude change of a factor of two is equal to about 6 dB, because $\log_{10} 2 = 0.3010$. An amplitude factor of $\sqrt{2}$ is equal to about 3 dB, and so forth.

An obvious question is what voltage to use for a time varying signal? The most common convention is to use the root mean square (RMS) average voltage.

$$V_{\text{RMS}} = \sqrt{\frac{\int_0^T V(t)^2 dt}{T}}$$

If not specified you can reasonably assume that RMS voltage was intended.

Decibels are also conveniently used to express rates of exponential falloff in a system response. This is especially common in engineering specifications. For example, a response that is proportional to $1/f$ (e.g., a single-pole system with no zeros such as a simple RC low-pass analog filter) decays at (approximately) $6 \approx 20\log_{10}(2)$ dB per octave (per frequency doubling), or at $20 = 20\log_{10}(10)$ dB per decade (per 10-fold frequency increase.)

In a dB vs. log frequency plot, such asymptotic power law behavior is easy to predict (and sketch), because a falloff of f^{-n} is just a straight line with a slope of $20n$ dB/decade. One can thus approximately sketch the amplitude response of system as a set of simple lines with differing slopes (such plots are called *Bode plots*).

Recall from the lecture notes on linear time invariant systems in the frequency domain that the frequency response of an underdamped seismometer is given by

$$|\Phi(\omega)| = \frac{\omega^2}{\sqrt{(\omega^2 - \omega_2^2)^2 + 4\xi^2\omega^2}}. \tag{D.2}$$

Expanding this in a Taylor series around $\omega = 0$, we get that

$$|\Phi(\omega)| = \frac{\omega^2}{\omega_s^2} + O(\omega^4). \tag{D.3}$$

Since $\omega = 2\pi f$,

$$|\Phi(f)| = O(f^2) \tag{D.4}$$

as f goes to 0. We would expect to see the frequency response drop off by 40 dB/decade or 12 dB/octave as f approaches 0.

Figures D.1, D.2, and D.3 show the amplitude displacement-displacement response of an underdamped seismometer with $\zeta = \omega_s/\sqrt{2}$ in linear-linear, dB (log)-linear, and dB-log plots, respectively. Note that the Figure D.3 plot is most easily interpretable, shows the essential characteristics of $|\Phi(f)|$ most clearly, and has the expected quadratic ($O(f^2)$) response fall-off of 40 dB/decade at low frequencies.

Note that we have looked the behavior of $|\Phi(f)|$ as f goes to 0. Some other authors consider the behavior of the power spectral density, $|\Phi(f)|^2$, as f goes to 0. Following that convention, the power spectral density is $O(f^4)$ as f approaches 0. However, in either case we would still see a decrease of 40 dB/decade on the plot.

So far, we have only used dB's as a measure of relative power. In situations where we want to establish an absolute scale, we must first pick a reference level for 0 dB. For example, in acoustics, a commonly used reference level for the sound pressure level (SPL) is an amplitude of 20 μPa in air. In this system, the unit is "dB(SPL)". In electronics, a commonly used reference level is one milliwatt. In this system the unit is "dBm". A voltage based scheme that is independent of the particular impedance uses a reference level of 0.775 V. This voltage happens to produce 1 milliwatt of power in a 600 ohm resistor (600 ohms

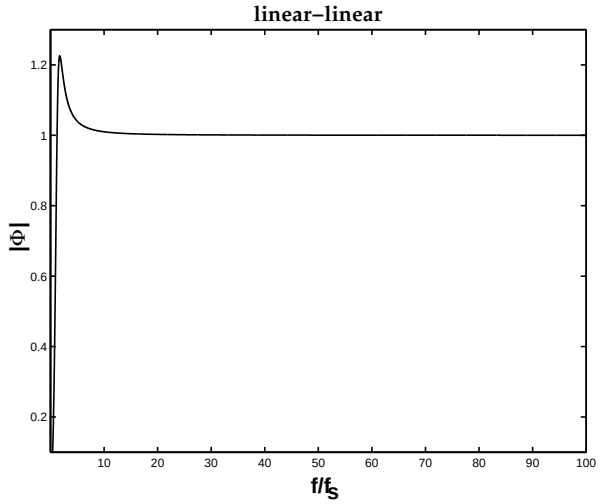

Figure D.1: Linear-linear plot of the amplitude response of a seismometer, $\zeta = \omega_s/\sqrt{2}$.

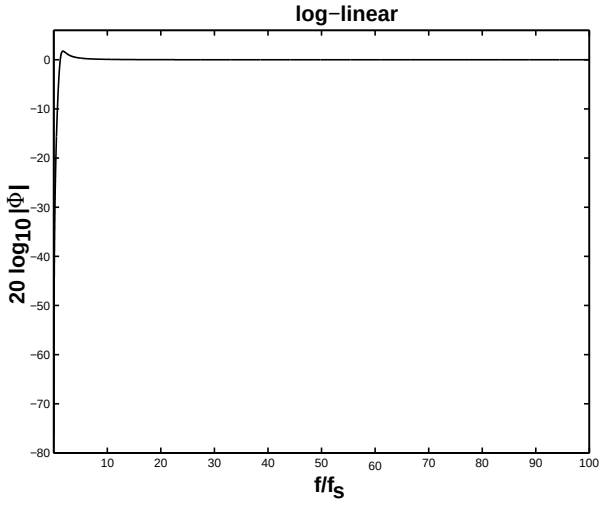

Figure D.2: Log-linear plot of the amplitude response of a seismometer, $\zeta = \omega_s/\sqrt{2}$.

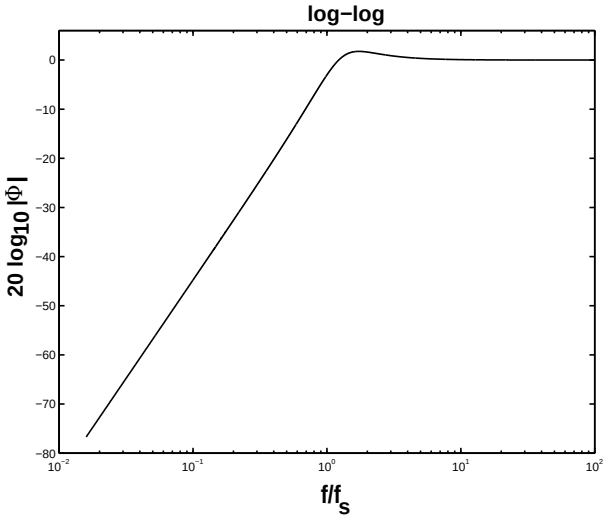

Figure D.3: Log-log plot of the amplitude response of a seismometer, $\zeta = \omega_s/\sqrt{2}$.

is the standard impedance in audio work.) In this system, the unit is "dBu" for "unloaded". You'll also see "dBV" for voltage relative to 1 volt, and "dBv" which is an older name for "dBu".

Appendix E

Plotting Spectra Using the FFT

Plotting the spectrum of a signal from its FFT is a very common activity. In this set of notes, we discuss how to produce such plots on both absolute and dB scales.

Suppose that x_n represents a voltage. Then the power of the signal at time $t = n\Delta t$, assuming a constant load R is

$$P_n = x_n^2/R. \tag{E.1}$$

For convenience, we'll take $R = 1$ ohm. The scaling factor of R can easily be reinserted into the equations, but it has no important effects.

The average power of the signal is

$$P_{avg} = \frac{\sum_{n=0}^{N-1} |x_n|^2 \Delta t}{T} \tag{E.2}$$

where the total length of the signal is $T = N\Delta t$. Thus

$$P_{avg} = \frac{\sum_{n=0}^{N-1} |x_n|^2}{N}. \tag{E.3}$$

Recall that by Parseval's theorem for the DFT,

$$\sum_{n=0}^{N-1} |x_n|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X_k|^2. \tag{E.4}$$

Thus

$$P_{avg} = \frac{\sum_{k=0}^{N-1} |X_k|^2}{N^2}. \tag{E.5}$$

Now, consider the power spectral density. We want to have

$$P_{avg} = \int_0^r PSD(f) df \tag{E.6}$$

where r is the sampling frequency. In terms of the discrete Fourier transform, we want

$$P_{avg} = \sum_{k=0}^{N-1} PSD_k \Delta f \tag{E.7}$$

where $\Delta f = r/N$. Combining (E.5) and (E.7) we see that

$$PSD_k = \frac{|X_k|^2}{Nr}. \tag{E.8}$$

The units of the PSD are x^2/Hz, whatever the units of x are. If we want to plot the spectrum with a dB scale, relative to to an amplitude of 1, then we should plot

$$PSD_k^{dB} = 10 \log_{10} \frac{|X_k|^2}{Nr}. \tag{E.9}$$

Note that a factor of 10 is used here because the amplitudes are already squared. The frequency associated with point k of the DFT is

$$f_k = \frac{kr}{N}.$$

We can also shift the range of frequencies to run from $-r/2$ to $r/2$ if desired.

Note that the PSD values will automatically adjust to changes in the length N of the sampled signal and the sampling rate r. We should be able to recover the total power of the signal by integrating the PSD from $f = 0$ to $f = r$.

A common problem with spectral estimation is that due to short term random variations in the signal (noise), the spectral estimate can be noisy. By computing the spectrum for each of many sections of the input signal and then averaging the spectra, we can average out these variations to get at the long term behavior of the signal. In computing the average, there are several options- we could average the values of $|X_k|$, the values of $|X_k|^2$, or even the dB values. This produces subtle changes in the results. In Welch's method, the method most commonly used in practice, values of $|X_k|^2$ are averaged.

In the following example, we consider the spectral analysis of a signal containing what is known as "Gaussian white noise". The signal consists of samples that are independent and normally distributed with expected value 0 and standard deviation 1. We'll assume a sampling frequency of $r = 100$ Hz. For such a signal, the average value of x_n^2 is 1, so the average power of the signal should be 1 Watt. It can be shown (later in the course) that the spectrum of such a signal is flat, with equal energy at all frequencies from 0 to r. However, because we have only a random sample of finite length, the actual spectrum that we obtain from our sample will have some sampling variability. We generate a signal of 1,000,000 samples. Both approaches to computing the average power give $P = 1.0002$ Watts.

Figure 1 shows the periodogram estimate of the spectrum. One problem with this plot is that 1,000,000 different frequencies are represented and there simply isn't enough resolution on the paper to show all of these frequencies. In

Figure 2, we've plotted 1,000 of the frequencies. Notice that the average value of the PSD is about 0.01. When this is multiplied by the frequency range of 100 Hz, we get an average power of 1 Watt, as expected. Figure 3 shows the same figure, on a dB scale. The average is at -20 dB, corresponding to an average x^2/Hz of 0.01. Again, when multiplied by 100, this gives a total power of 1 Watt.

These spectra are somewhat noisy. We can improve upon them by breaking the signal into sections of length $M = 1000$, computing spectra for each section, and averaging the spectra. Figure 4 shows the averaged spectrum in x^2/Hz units. Figure 5 shows the same spectrum in dB units. These spectra are much smoother than the first three spectra. Note that the vertical axes have changed to cover a much smaller range.

Finally, Figure 6 shows the spectrum produced by MATLAB's pwelch command. This closely matches Figure 5. A few small differences can be attributed to the fact that pwelch uses a Hamming window before computing the FFT to help reduce spectral leakage.

There are many other varieties of "colored noise" that have been identified over the years. For example, in "red noise", a white noise signal x_n is integrated to obtain a new signal y_n. Since differentiation effectively multiplies the Fourier transform of a signal by $2\pi if$, one would expect integration to divide the Fourier transform (and thus the spectrum) by the same factor of $2\pi if$. Since $|\Phi(f)|$ falls off proportionally to $1/f$ as f increases, we would expect the power spectrum of red noise to fall off as $1/f^2$. Figure 7 shows the averaged spectrum of a red noise signal plotted against a logarithmic frequency scale. Notice that the PSD falls off at a rate of 20 dB per decade. Similarly, in *pink noise* or *1/f noise*, $|\Phi(f)|$ is proportional to $1/\sqrt{f}$ and the power spectral density is proportional to $1/f$ as f increases. Here $1/f$ refers to the power spectral density, not $|\Phi(f)|$.

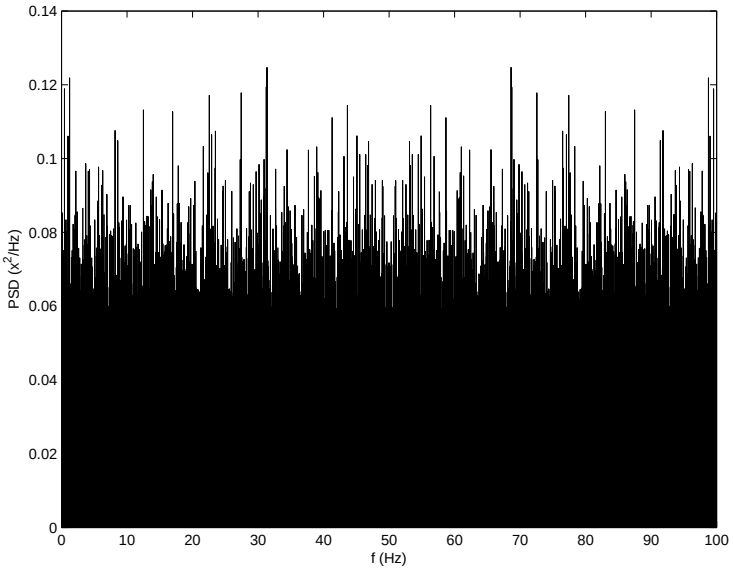

Figure E.1: Periodogram of the white noise signal, $N = 1,000,000$.

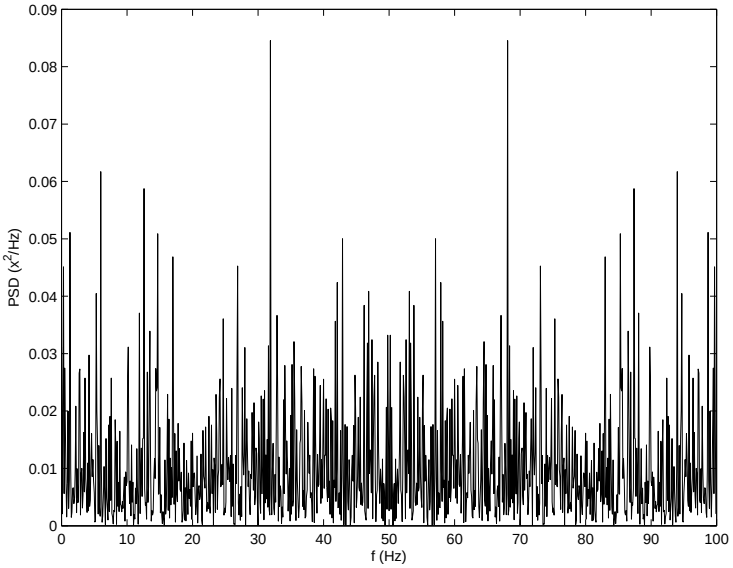

Figure E.2: Periodogram of the white noise signal, $N = 1,000,000$. This graph shows only 1,000 equally spaced frequency values.

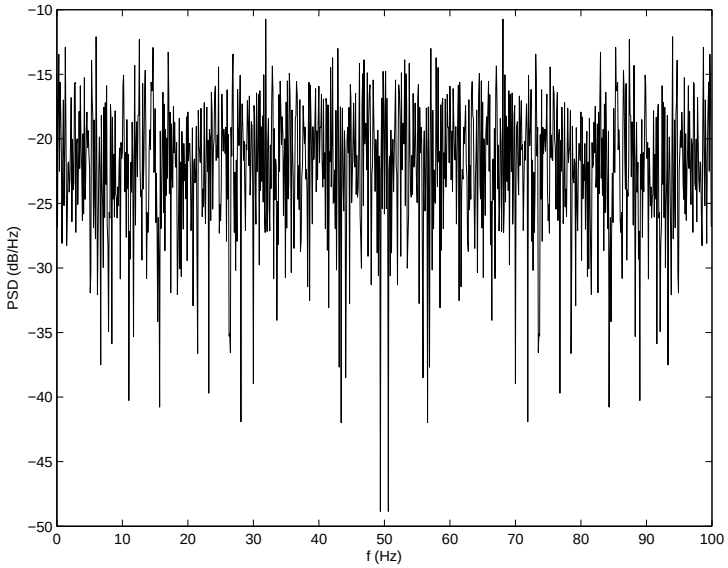

Figure E.3: Periodogram of the white noise signal, $N = 1,000,000$. This graph shows only 1,000 equally spaced frequency values. This version of the plot has dB/Hz units.

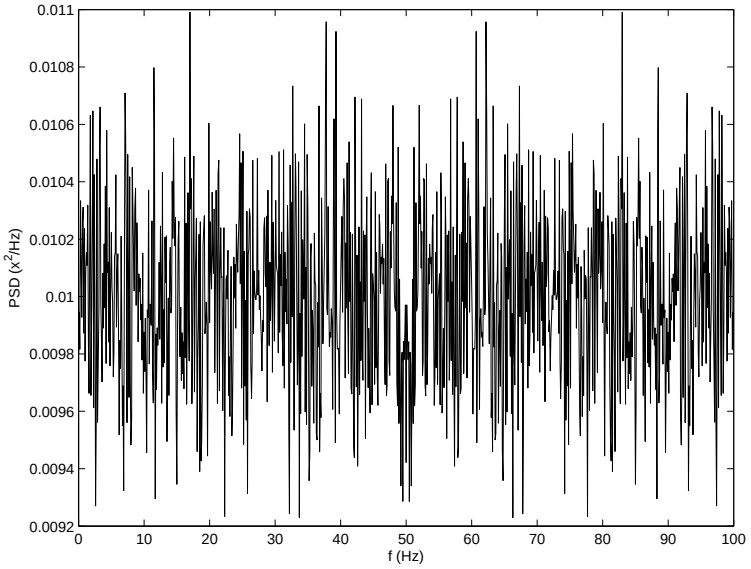

Figure E.4: Averaged periodogram of the white noise signal, blocks of $M = 1,000$.

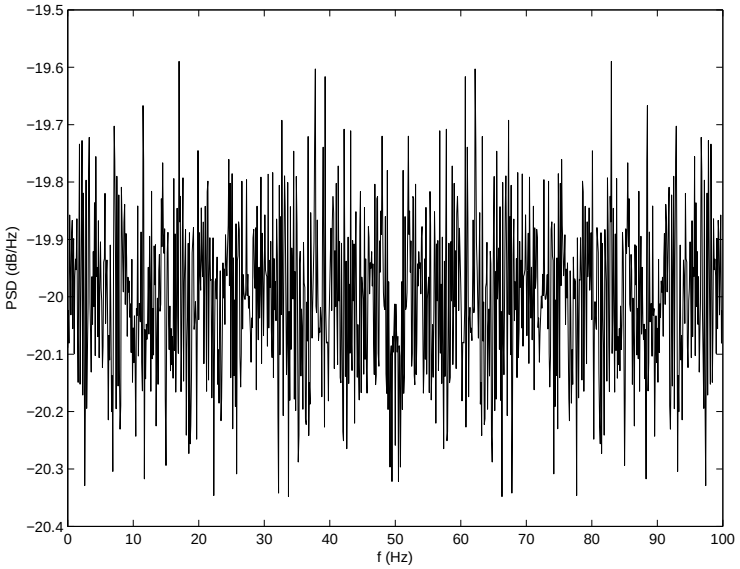

Figure E.5: Averaged periodogram of the white noise signal, blocks of $M = 1,000$, dB/Hz units.

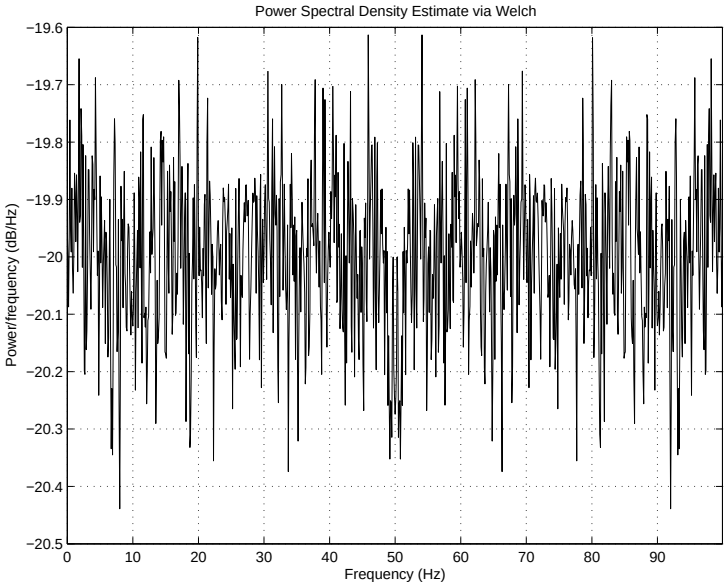

Figure E.6: PSD estimate produced by pwelch.

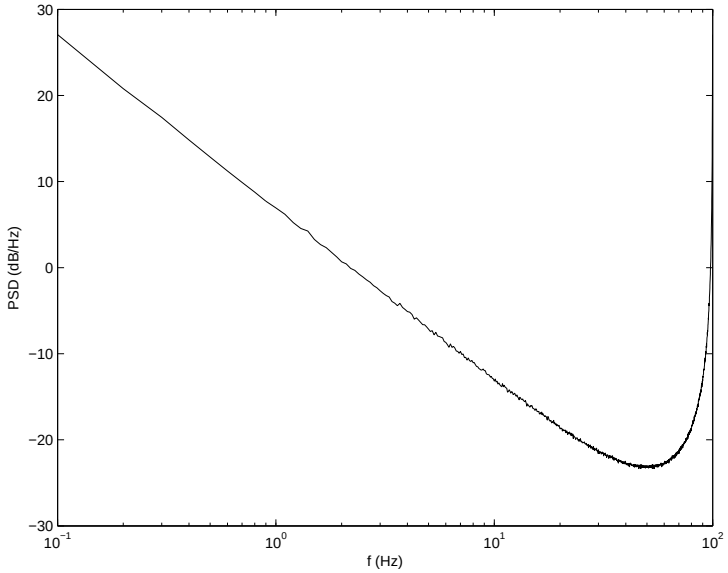

Figure E.7: Spectrum of red noise.

Appendix F

Summary of Fourier Transform Properties

This appendix summarizes some important facts about Fourier transforms and convolutions. Note that there are several different definitions of the Fourier transform in common use. If you refer to other books which use different definitions then of course the formulas will be different. This appendix has been made consistent with the following definition of the Fourier transform and its inverse.

$$\mathcal{F}(\phi(t)) = \Phi(f) = \int_{-\infty}^{\infty} f(t)e^{-2\pi ift}dt$$

$$\mathcal{F}^{-1}(\Phi(f)) = \phi(t) = \int_{-\infty}^{\infty} \Phi(f)e^{2\pi ift}df$$

The notation $\phi(t) \supset \Phi(f)$ is also used to indicate that $\Phi(f)$ is the Fourier transform of $\phi(t)$. Another common notational convention is that capital letters are used for the names of functions in the frequency domain, and corresponding lower case letters are used for functions in the time domain.

Because these two definitions are very nearly symmetric (note the sign change in the exponent of e), it is possible to use a Fourier transform pair $\phi(t) \supset \Phi(f)$ to construct a second Fourier transform pair $\Phi(t) \supset \phi(-f)$. For example, the transform

$$\mathcal{F}(\text{sgn } t) = \frac{1}{\pi if}$$

translates into

$$\mathcal{F}\left(\frac{1}{\pi it}\right) = \text{sgn} - f = -\text{sgn } f$$

General Properties

$\phi(t)$	$\Phi(f)$
$a\phi(t) + b\psi(t)$	$a\Phi(f) + b\Psi(f)$
$\phi(at)$	$\frac{1}{\|a\|}\Phi(f/a)$
$\phi'(t)$	$2\pi i f\Phi(f)$
$e^{2\pi iat}\phi(t)$	$\Phi(f-a)$
$\phi(t-a)$	$e^{-2\pi iaf}\Phi(f)$
$\phi(t)\cos(2\pi f_0 t)$	$(\Phi(f-f_0)+\Phi(f+f_0))/2$
$\phi(t) * \psi(t)$	$\Phi(f)\Psi(f)$
$\phi(t)\psi(t)$	$\Phi(f) * \Psi(f)$
$\int_{-\infty}^{t}\phi(\tau)d\tau$	$\frac{\Phi(f)}{2\pi if} + \frac{\delta(f)}{2}\int_{-\infty}^{\infty}\phi(\tau)d\tau$

Parseval's Theorem

$$\int_{-\infty}^{\infty}\phi(t)\phi^*(t)dt = \int_{-\infty}^{\infty}\Phi(f)\Phi^*(f)df.$$

Symmetry Properties

$\phi(t)$	$\Phi(f)$
even	even
odd	odd
real, even	real, even
real, odd	imaginary, odd
imaginary, even	imaginary, even
imaginary, odd	real, odd
complex, even	complex, even
complex, odd	complex, odd
real, asymmetrical	complex, Hermitian
imaginary, asymmetrical	complex, anti-Hermitian
Hermitian	real
anti-Hermitian	imaginary

Specific Fourier Transform Pairs

$\phi(t)$	$\Phi(f)$	
$\delta(t)$	1	
1	$\delta(f)$	
$e^{-a\pi t^2}$	$\frac{1}{\sqrt{a}}e^{-a\pi f^2/a}$	$a > 0$
$\mathrm{II}(t)$	$\mathrm{sinc}\ f = \frac{\sin \pi f}{\pi f}$	
$\mathrm{III}(t)$	$\mathrm{III}(f)$	
$\frac{1}{t}$	$-\pi i \mathrm{sgn}\ f$	
$\mathrm{sgn}\ t$	$\frac{1}{\pi i f}$	
$\cos(2\pi f_0 t)$	$(\delta(f+f_0)+\delta(f-f_0))/2$	
$\sin(2\pi f_0 t)$	$i(\delta(f+f_0)-\delta(f-f_0))/2$	
$H(t)$	$\frac{1}{2\pi i f} + \frac{\delta(f)}{2}$	
$H(t)e^{-at}$	$\frac{1}{a+2\pi i f}$	$a > 0$
$H(t)e^{-at}\sin(bt)$	$\frac{b}{(2\pi i f+a)^2+b^2}$	$a > 0$
$H(t)e^{-at}\cos(bt)$	$\frac{2\pi i f+a}{(2\pi i f+a)^2+b^2}$	$a > 0$
$H(t)te^{-at}$	$\frac{1}{(2\pi i f+a)^2}$	$a > 0$

Other Definitions of the Fourier Transform

Much more extensive tables of Fourier transforms are available in various reference books. These tables are often based on slightly different definitions of the Fourier transform. One typical variation is using $+2\pi ift$ in the exponent in the Fourier transform formula instead of $-2\pi ift$. Another common variation involves pulling the 2π out of the exponential and putting a factor of 2π in front of the integral in either the forward or inverse transform or putting a factor of $\sqrt{2\pi}$ in both the forward and inverse transforms. Transforms based on these alternative definitions can be converted to our system without too much difficulty.

For example one common definition of the Fourier transform is

$$\Phi_{\text{alt}}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \phi(t) e^{-i\omega t} dt$$

Here Φ_{alt} denotes the Fourier transform under the alternate definition. Under this definition,

$$\Phi_{\text{alt}}(2\pi f) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \phi(t) e^{-i2\pi ft} dt.$$

Thus

$$\Phi_{\text{alt}}(2\pi f) = \frac{1}{\sqrt{2\pi}} \Phi(f)$$

where $\Phi(f)$ is the transform under our definition. Solving for $\Phi(f)$, we get

$$\Phi(f) = \sqrt{2\pi} \Phi_{\text{alt}}(2\pi f).$$

Maple's inttrans package has fourier() and invfourier() functions that use angular frequency $\omega = 2\pi f$ (in units of radians per time) instead of circular frequency f (in units of cycles per time.) The definitions of the transform pair are:

$$\Phi_{\text{Maple}}(w) = \int_{-\infty}^{\infty} \phi(t) e^{-i\omega t} dt$$

and

$$\phi(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \Phi_{\text{Maple}}(\omega) e^{i\omega t} d\omega.$$

To use Maple's functions with our definitions, you can substitute $\omega = 2\pi f$ into the result obtained by Maple.

$$\Phi(f) = \Phi_{\text{Maple}}(2\pi f).$$

In using Maple's Fourier transform functions, note that that the Dirac $\delta()$ function has the unusual scaling property

$$\delta(\omega) = \delta(2\pi f) = \delta(f)/(2\pi).$$

This must be taken into account in converting from Maple's definition into our definition. For example, Maple's fourier() function returns $\Phi_{\text{Maple}}(\omega) = 2\pi\delta(\omega)$ for the Fourier transform of $\phi(t) = 1$. We have $\Phi(f) = \delta(f)$ where the factors of 2π cancel out.

Bibliography

[1] Keiiti Aki and Paul G Richards. *Quantitative Seismology*. University Science Books, 2002.

[2] R. C. Aster, C. H. Thurber, and B. Borchers. *Parameter Estimation and Inverse Problems*. Elsevier, 3rd edition, 2018.

[3] Sh A Azimi, A. V. Kalinin, Kalinin V. V., and B. I. Piyovaryoy. Impulse and transient characteristics of media with linear and quadratic absorption laws, izvestiya. *Physics of the Solid Earth*, pages 88–93, 1968.

[4] R. J. Banks, R. L. Parker, and S. P. Huestis. Isostatic compensation on a continental scale: local versus regional mechanisms. *Geophysical Journal International*, 51(2):431–452, 1977.

[5] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 5th edition, 2015.

[6] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

[7] Richard S Gross. The excitation of the Chandler wobble. *Geophysical Research Letters*, 27(15):2329–2332, 2000.

[8] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82, March 1960.

[9] Ernest R Kanasewich. *Time Sequence Analysis in Geophysics*. University of Alberta, 1981.

[10] Bateman manuscript project, Arthur Erdélyi, and Harry Bateman. *Tables of integral transforms: Based in part on notes left by Harry Bateman and compiled by the staff of the Bateman manuscript project*. McGraw-Hill, 1954.

[11] Lawrence R Rabiner, Bernard Gold, and CK Yuen. *Theory and Application of Digital Signal Processing*. Prentice-Hall, 2016.

[12] Masanobu Shinozuka and C-M Jan. Digital simulation of random processes and its applications. *Journal of sound and vibration*, 25(1):111–128, 1972.

[13] David J Thomson. Spectrum estimation and harmonic analysis. *Proceedings of the IEEE*, 70(9):1055–1096, 1982.

[14] Donald L Turcotte and Gerald Schubert. *Geodynamics*. Cambridge university press, second edition, 2002.