A Course in the Mathematics of Signal Processing

Charles L. Byrne

January 25, 2004

TO EILEEN

Contents

Chapter 1

Introduction

The situations of interest to us here can be summarized as follows: the data has been obtained through some form of sensing; physical models, often simplified, describe how the data we have obtained relates to the information we seek; there usually isn't enough data and what we have is corrupted by noise and other distortions. Although applications differ from one another in their details they often make use of a common core of mathematical ideas; for example, the Fourier transform and its variants play an important role in many areas of signal and image processing, as do the language and theory of matrix analysis, iterative optimization and approximation techniques and the basics of probability and statistics. This common core provides the subject matter for this text. Applications of the core material to tomographic medical imaging, optical imaging and acoustic signal processing are included.

The term *signal processing* is used here in a somewhat restrictive sense to describe the extraction of information from measured data. This text is designed to provide the necessary mathematical background to understand and employ signal processing techniques in an applied environment. The emphasis is on a small number of fundamental problems and essential tools, as well as on applications. Certain topics that are commonly included in textbooks are touched on only briefly or in exercises or not mentioned at all. Other topics not usually considered to be part of signal processing, but which are becoming increasingly important, such as iterative optimization methods, are included. The book, then, is a rather personal view of the subject and reflects the author's interests.

The term *signal* is not meant to imply a restriction to functions of a single variable; indeed most of what we discuss in this text applies equally to functions of one and several variables and therefore to image processing. However, there are special problems that arise in image processing, such as edge detection, and special techniques to deal with such problems; we

shall not consider such techniques in this text. Topics discussed include the following: Fourier series and transforms in one and several variables; applications to acoustic and EM propagation models, transmission and emission tomography and image reconstruction; sampling and the limited data problem; matrix methods, singular value decomposition and data compression; optimization techniques in signal and image reconstruction from projections; autocorrelations and power spectra; high resolution methods; detection and optimal filtering; eigenvector-based methods for array processing and statistical filtering.

Chapter 2

Complex Numbers

It is standard practice in signal processing to employ complex numbers whenever possible. One of the main reasons for doing this is that it enables us to represent the important sine and cosine functions in terms of complex exponential functions and to replace trigonometric identities with the somewhat simpler rules for the manipulation of exponents.

The complex numbers are the points in the x, y-plane: the complex number $z = (a, b)$ is identified with the point in the plane having $a = Re(z)$, the *real part* of z, for its x-coordinate and $b = Im(z)$, the *imaginary part* of z, for its y-coordinate. We call (a, b) the *rectangular form* of the complex number z. The *conjugate* of the complex number z is $\overline{z} = (a, -b)$. We can also represent z in its polar form: let the *magnitude* of z be $|z| = \sqrt{a^2 + b^2}$ and the *phase angle* of z, denoted $\theta(z)$, be the angle in $[0, 2\pi)$ with $\cos\theta(z) = a/|z|$. Then the *polar form* for z is

$$z = (|z|\cos\theta(z), |z|\sin\theta(z)).$$

Any complex number $z = (a, b)$ for which the imaginary part $Im(z) = b$ is zero is identified with (treated as the same as) its real part $Re(z) = a$; that is, we identify a and $z = (a, 0)$. These real complex numbers lie along the x-axis in the plane, the so-called *real line*. If this were the whole story complex numbers would be unimportant; but they are not. It is the arithmetic associated with complex numbers that makes them important.

We add two complex numbers using their rectangular representations:

$$(a, b) + (c, d) = (a + c, b + d).$$

This is the same formula used to add two-dimensional vectors. We multiply complex numbers more easily when they are in their polar representations: the product of z and w has $|z||w|$ for its magnitude and $\theta(z)+\theta(w)$ modulo 2π for its phase angle. Notice that the complex number $z = (0, 1)$ has

$\theta(z) = \pi/2$ and $|z| = 1$, so $z^2 = (-1, 0)$, which we identify with the real number -1. This tells us that within the realm of complex numbers the real number -1 has a square root, $i = (0, 1)$; note that $-i = (0, -1)$ is also a square root of -1.

To multiply $z = (a, b) = a + ib$ by $w = (c, d) = c + id$ in rectangular form we simply multiply the binomials

$$(a + ib)(c + id) = ac + ibc + iad + i^2bd$$

and recall that $i^2 = -1$ to get

$$zw = (ac - bd, bc + ad).$$

If (a, b) is real, that is, if $b = 0$, then $(a, b)(c, d) = (a, 0)(c, d) = (ac, ad)$, which we also write as $a(c, d)$. Therefore, we can rewrite the polar form for z as

$$z = |z|(\cos\theta(z), \sin\theta(z)) = |z|(\cos\theta(z) + i\sin\theta(z)).$$

We will have yet another way to write the polar form of z when we consider the complex exponential function.

Exercise 1: Derive the formula for dividing one complex number in rectangular form by another (non-zero) one.

Exercise 2: Show that for any two complex numbers z and w we have

$$|zw| \geq \frac{1}{2}(z\overline{w} + \overline{z}w). \tag{2.1}$$

Hint: Write $|zw|$ as $|z\overline{w}|$.

Exercise 3: Show that, for any constant a with $|a| \neq 1$, the function

$$G(z) = \frac{z - \overline{a}}{1 - az}$$

has $|G(z)| = 1$ whenever $|z| = 1$.

Chapter 3

Complex Exponentials

The most important function in signal processing is the complex-valued function of the real variable x defined by

$$h(x) = \cos(x) + i\sin(x). \tag{3.1}$$

For reasons that will become clear shortly, this function is called the *complex exponential function*. Notice that the magnitude of the complex number $h(x)$ is always equal to one, since $\cos^2(x) + \sin^2(x) = 1$ for all real x. Since the functions $\cos(x)$ and $\sin(x)$ are 2π-periodic, that is, $\cos(x+2\pi) = \cos(x)$ and $\sin(x+2\pi) = \sin(x)$ for all x, the complex exponential function $h(x)$ is also 2π-periodic.

In calculus we encounter functions of the form $g(x) = a^x$, where $a > 0$ is an arbitrary constant. These functions are the *exponential* functions, the most well known of which is the function $g(x) = e^x$. Exponential functions are those with the property $g(u+v) = g(u)g(v)$ for every u and v. We show now that the function $h(x)$ in equation (3.1) has this property, so must be an exponential function; that is, $h(x) = c^x$ for some constant c. Since $h(x)$ has complex values, the constant c cannot be a real number, however.

Calculating $h(u)h(v)$ we find

$$h(u)h(v) = (\cos(u)\cos(v) - \sin(u)\sin(v)) + i(\cos(u)\sin(v) + \sin(u)\cos(v))$$

$$= \cos(u+v) + i\sin(u+v) = h(u+v).$$

So $h(x)$ is an exponential function; $h(x) = c^x$ for some complex constant c. Inserting $x = 1$ we find that c is

$$c = \cos(1) + i\sin(1).$$

Let's try to find another way to express c.

Recall from calculus that for exponential functions $g(x) = a^x$ with $a > 0$ the derivative $g'(x)$ is

$$g'(x) = a^x \ln(a) = g(x) \ln(a).$$

Since

$$h'(x) = -\sin(x) + i\cos(x) = i(\cos(x) + i\sin(x)) = ih(x)$$

we conjecture that $\ln(c) = i$; but what does this mean?

For $a > 0$ we know that $b = \ln(a)$ means that $a = e^b$. Therefore, we say that $\ln(c) = i$ means $c = e^i$; but what does it mean to take e to a complex power? To define e^i we turn to the Taylor series representation for the exponential function $g(x) = e^x$, defined for real x:

$$e^x = 1 + x + x^2/2! + x^3/3! + ...$$

Inserting i in place of x and using the fact that $i^2 = -1$, we find that

$$e^i = (1 - 1/2! + 1/4! - ...) + i(1 - 1/3! + 1/5! - ...);$$

note that the two series are the Taylor series for $\cos(1)$ and $\sin(1)$, respectively, so $e^i = \cos(1) + i\sin(1)$. Then the complex exponential function in equation (3.1) is

$$h(x) = (e^i)^x = e^{ix}.$$

Inserting $x = \pi$ we get

$$h(\pi) = e^{i\pi} = \cos(\pi) + i\sin(\pi) = -1$$

or

$$e^{i\pi} + 1 = 0,$$

which is the remarkable relation discovered by Euler that combines the five most important constants in mathematics, e, π, i, 1 and 0, in a single equation.

Note that $e^{2\pi i} = e^{0i} = e^0 = 1$, so

$$e^{(2\pi+x)i} = e^{2\pi i} e^{ix} = e^{ix}$$

for all x.

We know from calculus what e^x means for real x and now we also know what e^{ix} means. Using these we can define e^z for any complex number $z = a + ib$ by $e^z = e^{a+ib} = e^a e^{ib}$.

We know from calculus how to define $\ln(x)$ for $x > 0$ and we have just defined $\ln(c) = i$ to mean $c = e^i$. But we could also say that $\ln(c) = i(1 + 2\pi k)$ for any integer k; that is, the periodicity of the complex exponential function forces the function $\ln(x)$ to be multivalued.

For any nonzero complex number $z = |z|e^{i\theta(z)}$ we have

$$\ln(z) = \ln(|z|) + \ln(e^{i\theta(z)}) = \ln(|z|) + i(\theta(z) + 2\pi k),$$

for any integer k. If $z = a > 0$ then $\theta(z) = 0$ and $\ln(z) = \ln(a) + i(k\pi)$ for any even integer k; in calculus class we just take the value associated with $k = 0$. If $z = a < 0$ then $\theta(z) = \pi$ and $\ln(z) = \ln(-a) + i(k\pi)$ for any odd integer k. So we can define the logarithm of a negative number; it just turns out not to be a real number. If $z = ib$ with $b > 0$, then $\theta(z) = \frac{\pi}{2}$ and $\ln(z) = \ln(b) + i(\frac{\pi}{2} + 2\pi k)$, for any integer k; if $z = ib$ with $b < 0$ then $\theta(z) = \frac{3\pi}{2}$ and $\ln(z) = \ln(-b) + i(\frac{3\pi}{2} + 2\pi k)$ for any integer k.

Adding $e^{-ix} = \cos(x) - i\sin(x)$ to e^{ix} given by equation (3.1) we get

$$\cos(x) = \frac{1}{2}(e^{ix} + e^{-ix});$$

subtracting, we obtain

$$\sin(x) = \frac{1}{2i}(e^{ix} - e^{-ix}).$$

These formulas allow us to extend the definition of cos and sin to complex arguments z:

$$\cos(z) = \frac{1}{2}(e^{iz} + e^{-iz})$$

and

$$\sin(z) = \frac{1}{2i}(e^{iz} - e^{-iz}).$$

In signal processing the complex exponential function is often used to describe functions of time that exhibit periodic behavior:

$$h(\omega t + \theta) = e^{i(\omega t+\theta)} = \cos(\omega t + \theta) + i\sin(\omega t + \theta),$$

where the *frequency* ω and *phase angle* θ are real constants, and t denotes time. We can alter the magnitude by multiplying $h(\omega t + \theta)$ by a positive constant $|A|$, called the *amplitude*, to get $|A|h(\omega t + \theta)$. More generally, we can combine the amplitude and the phase, writing

$$|A|h(\omega t + \theta) = |A|e^{i\theta}e^{i\omega t} = Ae^{i\omega t},$$

where A is the complex amplitude $A = |A|e^{i\theta}$. Many of the functions encountered in signal processing can be modeled as linear combinations of such complex exponential functions or *sinusoids*, as they are often called.

Exercise 1: Show that if $\sin\frac{x}{2} \neq 0$ then

$$E_M(x) = \sum_{m=1}^{M} e^{imx} = e^{ix(\frac{M+1}{2})}\frac{\sin(Mx/2)}{\sin(x/2)}. \tag{3.2}$$

Hint: Note that $E_M(x)$ is the geometric progression

$$E_M(x) = e^{ix} + (e^{ix})^2 + (e^{ix})^3 + ... + (e^{ix})^M = e^{ix}(1 - e^{iMx})/(1 - e^{ix}).$$

Now use the fact that, for any t, we have

$$1 - e^{it} = e^{it/2}(e^{-it/2} - e^{it/2}) = e^{it/2}(-2i)\sin(t/2).$$

Exercise 2: The *Dirichlet kernel* of size M is defined as

$$D_M(x) = \sum_{m=-M}^{M} e^{imx}.$$

Use equation (3.2) to obtain the closed-form expression

$$D_M(x) = \frac{\sin((M + \frac{1}{2})x)}{\sin(\frac{x}{2})};$$

note that $D_M(x)$ is real-valued.
Hint: Reduce the problem to that of Exercise 1 by factoring appropriately.

Exercise 3: Use the result in equation (3.2) to obtain the closed-form expressions

$$\sum_{m=N}^{M} \cos mx = \cos(\frac{M+N}{2}x)\frac{\sin(\frac{M-N+1}{2}x)}{\sin\frac{x}{2}}$$

and

$$\sum_{m=N}^{M} \sin mx = \sin(\frac{M+N}{2}x)\frac{\sin(\frac{M-N+1}{2}x)}{\sin\frac{x}{2}}.$$

Hint: Recall that $\cos mx$ and $\sin mx$ are the real and imaginary parts of e^{imx}.

Exercise 4: Graph the function $E_M(x)$ for various values of M.

We note in passing that the function $E_M(x)$ equals M for $x = 0$ and equals zero for the first time at $x = 2\pi/M$. This means that the *main lobe* of $E_M(x)$, the inverted parabola-like portion of the graph centered at $x = 0$, crosses the x-axis at $x = 2\pi/M$ and $x = -2\pi/M$, so its height is M and its width is $4\pi/M$. As M grows larger the main lobe of $E_M(x)$ gets higher and thinner.

Chapter 4

Hidden Periodicities

We begin with what we call the *Ferris Wheel Problem.* A Ferris Wheel is a carnival ride, or perhaps a tourist attraction, like the London Eye, consisting of a large rotating wheel supported so that its axis of rotation is parallel to the ground. Around the rim of the wheel are seats for the riders. Once the seats are filled the wheel rotates for some number of minutes, from time $t = 0$ to $t = T$ and then it slows to let the riders off. Suppose that the radius of the wheel is R feet, the center of the wheel is $R + H$ feet off the ground and from time $t = 0$ to $t = T$ the wheel completes one revolution in P seconds, so that its frequency of rotation is $\omega = \frac{2\pi}{P}$ radians per second.

Exercise 1: Determine the formulas giving the horizontal and vertical coordinates of the position of a particular rider at an arbitrary time t in the time interval $[0, T]$.

Now let us make it a bit more complicated. Suppose that, instead of seats around the rim of the wheel, there is a smaller Ferris Wheel (or several identical smaller wheels distributed around the rim, for stability). To avoid confusion, let's let R_1 and ω_1 be the radius and frequency of rotation of the original wheel and let R_2 and ω_2 be the radius and frequency of rotation of the second wheel.

Exercise 2: Now find the formulas giving the horizontal and vertical coordinates of the position of a particular rider at an arbitrary time t in the time interval $[0, T]$.

Continuing down this road, imagine a third wheel on the rim of the second, a fourth on the rim of the third, and so on; in fact, let there be J nested Ferris wheels, the j-th wheel having radius R_j and frequency of rotation ω_j. Figure 4.1 illustrates the case of $J = 3$.

Exercise 3: Repeat the previous exercise, but for the case of J nested wheels.

What we have been doing here is solving what is called a *direct problem*. The simplest way to explain a direct problem is to contrast it with one that is not direct, a so-called *inverse problem* [82], [142]. An inverse problem involving the Ferris Wheels is the following. Suppose our data consists of the positions of a particular rider at several distinct times, $t_1, ..., t_M$. From this data alone determine J, the number of nested wheels, the radii R_j of the wheels, and their frequencies of rotation ω_j.

Direct problems usually look ahead in time to what would happen in a certain situation. The formulas involved are usually straightforward applications of the relevant concepts and there is no data involved. In contrast, inverse problems ask us to determine what did happen, given some measurements of the outcome. The measurements may be unreliable or noisy and there may not be enough measurements to determine a single unique answer. In the inverse Ferris Wheel problem we would assume that J, the number of wheels, is smaller than M, the number of measurements. Given M measurements, it is usually possible to fit those measurements exactly to a model involving more than M wheels; the hard part is to let the data tell us what J is. A second issue is the choosing of the times t_m at which the measurements are taken. If we were to take all the measurements in rapid succession, over a very small interval of time, the problem would become much more difficult and the answer much more sensitive to slight errors in the data. Just how we should select the times t_m will depend on our prior knowledge of what the frequencies of rotation might be. If some of the wheels are turning very rapidly we must sample quickly to determine that. Otherwise we get the *strobe light* type of aliasing.

The measured data giving the positions of the rider at various times is said to contain information about the *hidden periodicities* involved. There are periodicites, not always hidden, in many different data sets. For example, data giving the temperature every hour in downtown Lowell for the last one hundred million years would show several interest periodicities, or almost periodicities. Clearly there is the periodicity corresponding to the seasons of the year. There is also the periodicity associated with the passage from day to night, although this is a somewhat more complicated function of time, involving, as it does, the varying lengths of day and night in different seasons. There will be other components corresponding to the temperature changes from one day to the next, having no simple periodic aspect. On top of all this there will be components with much longer periods (so much smaller frequencies), corresponding to the climate changes from one century to the next. There will be components with even longer periods, the climate changes studied in connection with global warming, having periods of thousands of years. An interesting study is to try to

relate, or to *correlate*, the periodic components in one data set with those in another. For example, is earth weather related to the periodicities in the sun spot activity?

Many of the signals we encounter in practice contain complex exponential components having different amplitudes and frequencies. The standard model for such signals is

$$s(t) = \sum_{n=1}^{N} |A_n| e^{i(\omega_n t + \theta_n)}. \tag{4.1}$$

One of the main problems in signal processing is to determine the values of the parameters N, $|A_n|$, ω_n and θ_n from measurements of the function $s(t)$; that is, to determine the complex exponential components that constitute the signal $s(t)$. For example, in automated human voice recognition a particular individual speaker is identified by the combination of the $|A_n|$ and ω_n present in the speech of that person when pronouncing a certain sound. Our ears perform this identification task when we recognize the voice of a particular singer or actor. In digital speech processing the assumption is that the signal corresponding to the voicing of a particular sound has the form given in equation (4.1), at least for a short time interval (until the next sound is voiced). A second point of view is that equation (4.1) is a model to be used to perform certain operations on a signal, such as noise reduction or compression.

In some applications we do not have exact measurements of $s(t)$ but noisy estimates of what those exact values are. Our job is then to clean up the data to extract the parameter values. In restoration of old recordings the parameters are estimated from noisy measurements of the old recording and these parameters modified and inserted to recreate digitally the original sound. The noisy measurement data can then be modeled using equation (4.1) and (at least some of) the noise removed by subtracting certain complex exponential components attributed to the noise. At the same time the quality of the signal can be enhanced by modifying the amplitudes of the components that remain. The resulting set of numbers can then be converted back into audible sound.

In radar, sonar, radio astronomy and related remote sensing applications the variable ω may not be frequency but a direction in space relative to a fixed coordinate system. In such cases the variable t denotes the location in space at which the function $s(t)$ is measured. The various parts of the objects of interest send (or reflect) individual signals and the measuring devices record the superposition of all these signals. Whether the objects of interest are planes in radar, the stars in the heavens in optical or radio astronomy, submarines and ships at sea in sonar, regions of a patient's body in medical tomography or portions of the earth's surface in synthetic aperture radar imaging, the received signals must be analyzed, that is, broken down into their constituent parts, so that the individual sources of received

energy can be separately known. A nonzero value of $|A_n|$ then indicates the presence of a source (or reflector) of electromagnetic or acoustic energy at angle ω_n. We measure $s(t)$ at many different locations t and from that data we try to decompose the signal into its components. How well we are able to identify separate sources of energy is the *resolving capablity* of the process. Our ability to resolve will depend on several things, including the hardware we use, where we are able to measure $s(t)$ and at how many values of t we are able to employ, and also the mathematical methods we use to perform the analysis of the signal.

Common to each of these applications is the need to isolate the individual complex exponential components in the measured signal. This is the *signal analysis* problem, which we consider next.

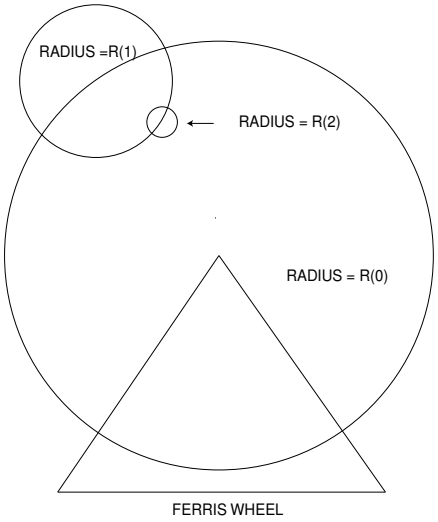


Figure 4.1: The Ferris Wheel for $J = 3$.

Chapter 5

Signal Analysis: A First Approach

We shall assume now that the signal we wish to analyze is $s(t)$ given by equation (4.1), which we rewrite as

$$s(t) = \sum_{n=1}^{N} A_n e^{i\omega_n t}, \tag{5.1}$$

with $A_n = |A_n|e^{i\theta_n}$ the complex amplitudes. Although we shall often speak of t as a time variable, that is not essential. We assume that we have determined the value of the function $s(t)$ at M points in time, called the *sampling times*. Although it is not necessary, we shall assume the sampling times are equispaced, that is, they are $t = m\Delta$, $m = 1, ..., M$, where $\Delta > 0$ is the difference between successive sampling times. So our data are the values $s(m\Delta)$, $m = 1, ..., M$. Our goal is to determine N, the number of complex exponential components in the signal $s(t)$, their complex amplitudes A_n and the frequencies ω_n. We assume that N is smaller than M.

The aliasing problem: Given our data, it is impossible for us to distinguish a frequency ω from $\omega + \frac{2\pi n}{\Delta}$, for any integer n. This can result in *aliasing*, if the sample spacing Δ is not sufficiently small.

For every m we have

$$e^{i\omega_n m\Delta} = e^{i(\omega_n + 2\pi/\Delta)m\Delta},$$

which tells us that, using the data we have, we cannot distinguish between the frequencies ω_n and $\omega_n + 2\pi/\Delta$. We shall therefore make the assumption that Δ has been selected small enough so that $|\omega_n| \leq \pi/\Delta$ for all n. If we have not selected Δ small enough, we have *undersampled* and some of the

frequencies ω_n will be mistaken for lower frequencies; this is the *aliasing problem*. We describe now an approach that determines N, the ω_n and the A_n well enough if the data is relatively noise-free, none of the ω_n are too close to one another and the M is large enough.

Our assumption: Our first approach to solving the signal analysis problem is based on a simplifying restriction on the possible locations of the frequencies ω_n. We assume that the ω_n are some of the members of the set $\{\alpha_k = -\frac{\pi}{\Delta} + k\frac{2\pi}{\Delta M},\ k = 1, 2, ..., M\}$; these are the M frequencies equispaced across the interval $(-\frac{\pi}{\Delta}, \frac{\pi}{\Delta}]$. We then rewrite $s(t)$ as

$$s(t) = \sum_{k=1}^{M} B_k e^{i\alpha_k t}; \tag{5.2}$$

values of k for which the B_k are not zero will be the ones for which α_k is one of the original ω_n and $B_k = A_n$. Our data is then

$$s(m\Delta) = \sum_{k=1}^{M} B_k e^{-im\pi} e^{i2\pi km/M},$$

for $m = 1, ..., M$.

The complex vector dot product : For any positive integer J and any two J dimensional complex column vectors $\mathbf{u}$ and $\mathbf{v}$ we define the *complex vector dot product* to be

$$\mathbf{u} \cdot \mathbf{v} = \sum_{j=1}^{J} u_j \overline{v_j}.$$

Note that $\mathbf{u} \cdot \mathbf{v} = \mathbf{v}^\dagger \mathbf{u}$, where $\mathbf{v}^\dagger$, the *conjugate transpose* of the vector $\mathbf{v}$, is the row vector whose entries are the conjugates of the entries of the vector $\mathbf{v}$. Therefore, we can and do view the complex vector dot product as a special case of matrix multiplication.

As we shall see in a later chapter on the Cauchy inequality, the dot product is a way of checking how well two vectors resemble one another. This idea is used extensively in signal processing, when we form the dot product between the data vector and each of many potential component vectors, to see how much the data resembles each of them. This is called *matching* and is the basic idea in *matched filtering*, as we shall see later. We now apply this idea of matching in our first attempt at solving the signal analysis problem.

For each $j = 1, 2, ..., M$ we ask what data we would have collected had the signal $s(t)$ consisted solely of a single complex exponential $e^{i\alpha_j t}$ with frequency α_j; the answer is $e^{i\alpha_j m\Delta}$, for $m = 1, 2, ..., M$. We now let these numbers be the entries of a vector we call $\mathbf{e}_j$; then we match $\mathbf{e}_j$ with the data vector $\mathbf{d}$ having the entries $s(m\Delta)$.

Therefore, for each $j = 1, 2, ..., M$, we let the entries of the column vector $\mathbf{e}_j$ be

$$e_{jm} = e^{i\alpha_j m\Delta} = e^{-im\pi} e^{i2\pi jm/M}.$$

Let $\mathbf{e}_j^\dagger$ denote the conjugate transpose of $\mathbf{e}_j$, that is, the row vector whose entries are $\overline{e_{jm}}$, so that the matrix multiplication $\mathbf{e}_j^\dagger \mathbf{d}$ is the complex dot product of $\mathbf{e}_j$ and $\mathbf{d}$. Then

$$\mathbf{e}_j^\dagger \mathbf{d} = \sum\nolimits_{m=1}^{M} s(m\Delta) e^{-i\alpha_j m\Delta} = \sum\nolimits_{k=1}^{M} B_k (\sum\nolimits_{m=1}^{M} e^{2\pi i(k-j)m/M}).$$

The inner sum is $E_M(x)$ for $x = 2\pi(k-j)/M$, so we can use the closed form of this sum that we derived in an exercise earlier to conclude that the inner sum equals M if $k = j$ and is zero if $k \neq j$. Therefore, for each fixed j, as we run through the index of summation k, all the terms being added are zero, except when the index k reaches the fixed value j. Therefore

$$\mathbf{e}_j^\dagger \mathbf{d} = MB_j$$

for each j. To isolate the original frequencies ω_n we select those j for which $\mathbf{e}_j^\dagger \mathbf{d}$ is not zero; then the A_n is the associated value B_j.

So we know how to isolate the individual complex exponential components of $s(t)$, so long as each of the ω_n is, at least approximately, one of the α_k, which imposes the constraint that no two of the ω_n are closer to each other than $2\pi/\Delta M$; this limits our ability to resolve components whose frequencies are closer than that limit. If we know in advance that we are seeking frequencies ω_n closer than this limit we have at least two choices: increase M or increase Δ. The latter choice is a bit dangerous in that we risk aliasing if any of the ω_n have magnitudes close to π/Δ already. A third choice is to alter the method whereby we isolated the individual components. There are many ways to do this, as we shall see.

Chapter 6

Convolution and the Vector DFT

Convolution is an important concept in signal processing and occurs in several distinct contexts. In this chapter we shall discuss *non-periodic convolution* and *periodic convolution* of vectors. Later we shall consider the convolution of infinite sequences and of functions of a continuous variable. The reader may recall an earlier encounter with convolution in a course on differential equations. The simplest example of convolution is the non-periodic convolution of finite vectors.

Non-periodic convolution:

Recall the algebra problem of multiplying one polynomial by another. Suppose

$$A(x) = a_0 + a_1 x + ... + a_M x^M$$

and

$$B(x) = b_0 + b_1 x + ... + b_N x^N.$$

Let $C(x) = A(x)B(x)$. With

$$C(x) = c_0 + c_1 x + ... + c_{M+N} x^{M+N},$$

each of the coefficients c_j, $j = 0, ..., M+N$, can be expressed in terms of the a_m and b_n (an easy exercise!). The vector $c = (c_0, ..., c_{M+N})$ is called the *non-periodic convolution* of the vectors $a = (a_0, ..., a_M)$ and $b = (b_0, ..., b_N)$. Non-periodic convolution can be viewed as a particular case of periodic convolution, as we see next.

The DFT and the vector DFT:

As we just discussed, non-periodic convolution is another way of looking at the multiplication of two polynomials. This relationship between convolution on the one hand and multiplication on the other is a fundamental aspect of convolution, whenever it occurs. Whenever we have a convolution we should ask what related mathematical objects are being multiplied. We ask this question now with regard to periodic convolution; the answer turns out to be the *vector discrete Fourier transform*.

Given the N by 1 vector $\mathbf{f}$ with complex entries $f_0, f_1, ..., f_{N-1}$ define the *discrete Fourier transform* (DFT) of $\mathbf{f}$ to be the function $DFT_{\mathbf{f}}(\omega)$, defined for ω in $[0, 2\pi)$, by

$$DFT_{\mathbf{f}}(\omega) = \sum_{n=0}^{N-1} f_n e^{in\omega}.$$

The terminology can be confusing, since the expression 'discrete Fourier transform' is often used to describe several slightly different mathematical objects.

For example, in the exercise that follows we are interested solely in the values $F_k = DFT_{\mathbf{f}}(2\pi k/N)$, for $k = 0, 1, ..., N-1$. In this case the DFT of the vector $\mathbf{f}$ often means simply the vector $\mathbf{F}$ whose entries are the complex numbers F_k, for $k = 0, ..., N-1$; for the moment let us call this the *vector DFT* of $\mathbf{f}$ and write $\mathbf{F} = vDFT_{\mathbf{f}}$. The point of Exercise 1 is to show how to use the vector DFT to perform the *periodic convolution* operation.

In some instances the numbers f_n are obtained by evaluating a function $f(x)$ at some finite number of points x_n; that is, $f_n = f(x_n)$, for $n = 0, ..., N-1$. As we shall see later, if the x_n are equispaced, the DFT provides an approximation of the Fourier transform of the function $f(x)$. Since the Fourier transform is another function of a continuous variable, and not a vector, it is appropriate, then, to view the DFT also as such a function. Since the practice is to use the term DFT to mean slightly different things in different contexts, we adopt that practice here. The reader will have to infer the precise meaning of DFT from the context.

Periodic convolution:

Given the N by 1 vectors $\mathbf{f}$ and $\mathbf{d}$ with complex entries f_n and d_n, respectively, we define a third N by 1 vector $\mathbf{f} * \mathbf{d}$, the *periodic convolution* of $\mathbf{f}$ and $\mathbf{d}$, to have the entries

$$(\mathbf{f} * \mathbf{d})_n = f_0 d_n + f_1 d_{n-1} + ... + f_n d_0 + f_{n+1} d_{N-1} + ... + f_{N-1} d_{n+1}.$$

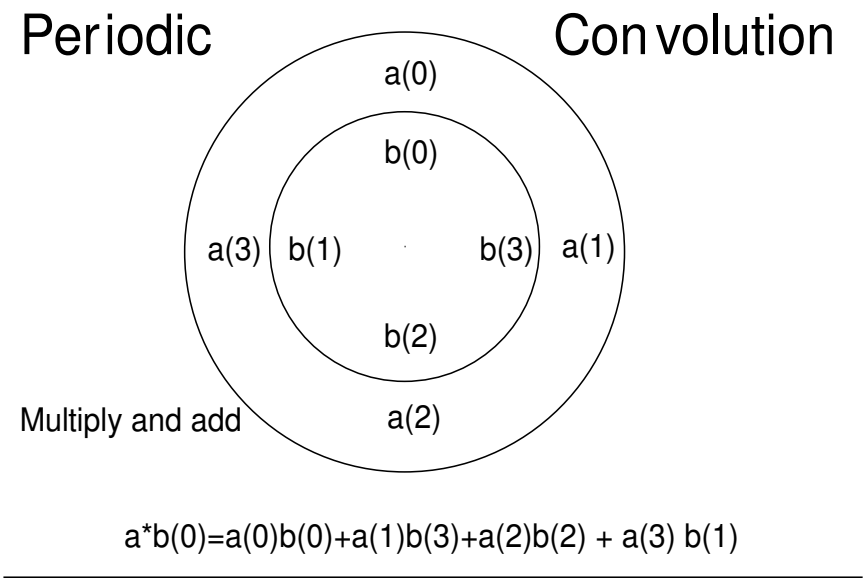
Periodic convolution is illustrated in Figure 6.1. The first exercise relates the periodic convolution to the vector DFT.

Exercise 1: Let $\mathbf{F} = vDFT_{\mathbf{f}}$ and $\mathbf{D} = vDFT_{\mathbf{d}}$. Define a third vector $\mathbf{E}$ having for its k-th entry $E_k = F_k D_k$, for $k = 0, ..., N-1$. Show that $\mathbf{E}$ is the vDFT of the vector $\mathbf{f} * \mathbf{d}$.

The vector $vDFT_{\mathbf{f}}$ can be obtained from the vector $\mathbf{f}$ by means of matrix multiplcation by a certain matrix G, called the *DFT matrix*. The matrix G has an inverse that is easily computed and can be used to go from $\mathbf{F} = vDFT_{\mathbf{f}}$ back to the original $\mathbf{f}$. The details are in Exercise 2.

Exercise 2: Let G be the N by N matrix whose entries are $G_{jk} = e^{i(j-1)(k-1)2\pi/N}$. The matrix G is sometimes called the *DFT matrix*. Show that the inverse of G is $G^{-1} = \frac{1}{N}G^{\dagger}$, where $G^{\dagger}$ is the conjugate transpose of the matrix G. Then $\mathbf{f} * \mathbf{d} = G^{-1}\mathbf{E} = \frac{1}{N}G^{\dagger}\mathbf{E}$.

As we mentioned above, nonperiodic convolution is really a special case of periodic convolution. Extend the $M+1$ by 1 vector a to an $M+N+1$ by 1 vector by appending N zero entries; similarly, extend the vector b to an $M+N+1$ by 1 vector by appending zeros. The vector c is now the periodic convolution of these extended vectors. Therefore, since we have an efficient algorithm for performing periodic convolution, namely the Fast Fourier Transform algorithm (FFT), we have a fast way to do the periodic (and thereby nonperiodic) convolution and polynomial multiplication.



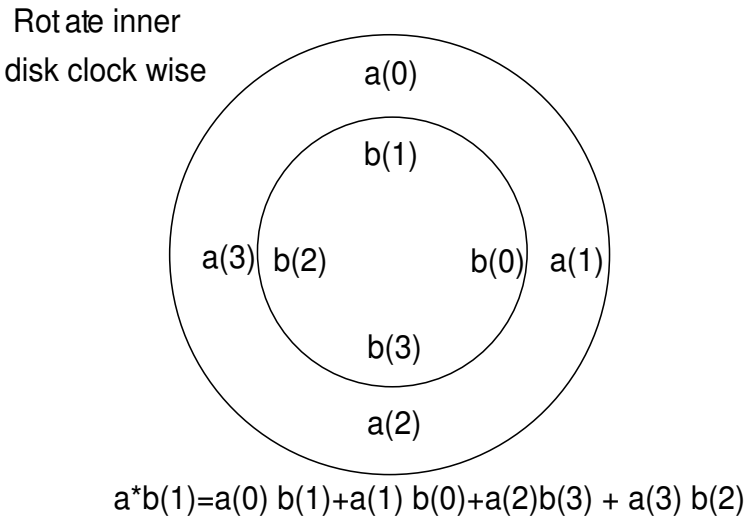


Figure 6.1: Periodic convolution of vectors $a = (a(0), a(1), a(2), a(3))$ and $b = (b(0), b(1), b(2), b(3))$.

Chapter 7

Signal Analysis: A Second Approach

As before, we assume that we have data vector $\mathbf{d}$ with entries $s(m\Delta)$, $m = 1, ..., M$ from the signal $s(t)$ given by equation (18.6). Unlike in our first approach, we do not now make any assumptions about the location of the frequencies ω_n, except that $|\omega_n| < \pi/\Delta$.

For each ω in the interval $(-\pi/\Delta, \pi/\Delta)$ let e_ω be the column vector with entries $e^{i\omega m\Delta}$, $m = 1, ..., M$. The output of the matched filter $\mathbf{e}_\omega^\dagger \mathbf{d}$, as a function of the continuous variable ω in the interval $(-\pi/\Delta, \pi/\Delta)$ is

$$DFT_{\mathbf{d}}(\omega) = \sum_{m=1}^{M} s(m\Delta)e^{-i\omega m\Delta}$$

$$= \sum_{n=1}^{N} A_n(\sum_{m=1}^{M} e^{i(\omega_n-\omega)m\Delta}).$$

We know from our earlier calculations that

$$\sum_{m=1}^{M} e^{i(\omega_n-\omega)m\Delta} = e^{i\frac{M+1}{2}(\omega_n-\omega)} \sin(\frac{M}{2}(\omega_n - \omega))/(\sin\frac{1}{2}(\omega_n - \omega)),$$

which equals M if $\omega = \omega_n$. If the ω_n are well separated then this sum is significantly smaller if ω is not near ω_n. So if the ω_n are well separated and M is significantly larger than N the function $DFT_{\mathbf{d}}(\omega)$ will be near MA_n when $\omega = \omega_n$, for each n, and will be near zero otherwise. Of course we cannot calculate $DFT_{\mathbf{d}}(\omega)$ for each ω; for the purposes of plotting we select sufficiently many values of ω and calculate $|DFT_{\mathbf{d}}(\omega)|$ at these points. Later we shall study a fast algorithm, known as the *fast Fourier transform* (FFT), which does this calculation for us in an efficient manner.

Exercise 1: Let $N = 2$ and $\omega_1 = -\alpha$, $\omega_2 = \alpha$ for some $\alpha > 0$ in $(-\pi, \pi)$. Let $A_1 = A_2 = 1$. Select a value of M that is greater than two and

calculate the values $f(m)$ for $m = 1, ..., M$. Plot the graph of the function $DFT_{\mathbf{d}}(\omega)$ on $(-\pi, \pi)$. Repeat the exercise for various values of M and values of α closer to zero. Notice how $DFT_{\mathbf{d}}(0)$ behaves as α goes to zero. For each fixed value of M there will be a critical value of α such that, for any smaller values of α, $DFT_{\mathbf{d}}(0)$ will be larger than $DFT_{\mathbf{d}}(\alpha)$. This is *loss of resolution*.

As the exercise has shown, for each fixed value of M there will be a limit to our ability to resolve closely spaced frequencies using $DFT_{\mathbf{d}}(\omega)$. If we are unable to increase the M we can try other methods of isolating the frequencies. We shall discuss these other methods later.

Chapter 8

Cauchy's Inequality

So far our methods for analyzing the measured signal have been based on the idea of matching the data against various potential complex exponential components to see which ones match best. The matching is done using the complex dot product, $\mathbf{e}_\omega^\dagger \mathbf{d}$. In the ideal case this dot product is large, for those values of ω that correspond to an actual component of the signal; otherwise it is small. Why this should be the case is the Cauchy-Schwarz inequality (or sometimes, depending on the context, just Cauchy's inequality, just Schwarz's inequality, or, in the Russian literature, Bunyakovsky's inequality).

The complex vector dot product: Let $\mathbf{u} = (a, b)$ and $\mathbf{v} = (c, d)$ be two vectors in two-dimensional space. Let $\mathbf{u}$ make the angle $\alpha > 0$ with the positive x-axis and $\mathbf{v}$ the angle $\beta > 0$. Let $||\mathbf{u}|| = \sqrt{a^2 + b^2}$ denote the length of the vector $\mathbf{u}$. Then $a = ||\mathbf{u}|| \cos\alpha$, $b = ||\mathbf{u}|| \sin\alpha$, $c = ||\mathbf{v}|| \cos\beta$ and $d = ||\mathbf{v}|| \sin\beta$. So $\mathbf{u} \cdot \mathbf{v} = ac + bd = ||\mathbf{u}||||\mathbf{v}||(\cos\alpha\cos\beta + \sin\alpha\sin\beta = ||\mathbf{u}||\,||\mathbf{v}||\cos(\alpha - \beta)$. Therefore, we have

$$\mathbf{u} \cdot \mathbf{v} = ||\mathbf{u}||\,||\mathbf{v}||\cos\theta, \tag{8.1}$$

where $\theta = \alpha - \beta$ is the angle between $\mathbf{u}$ and $\mathbf{v}$. Cauchy's inequality is

$$|\mathbf{u} \cdot \mathbf{v}| \leq ||\mathbf{u}||\,||\mathbf{v}||,$$

with equality if and only if $\mathbf{u}$ and $\mathbf{v}$ are parallel.

Cauchy's inequality extends to vectors of any size with complex entries. For example, the complex M-dimensional vectors $\mathbf{e}_\omega$ and $\mathbf{e}_\theta$ defined earlier both have length equal to $\sqrt{M}$ and

$$|\mathbf{e}_\omega^\dagger \mathbf{e}_\theta| \leq M,$$

with equality if and only if ω and θ differ by an integer multiple of π.

From equation (8.1) we know that the dot product $\mathbf{u} \cdot \mathbf{v}$ is zero if and only if the angle between these two vectors is a right angle; we say then that $\mathbf{u}$ and $\mathbf{v}$ are mutually *orthogonal*. Orthogonality was at the core of our first approach to signal analysis: the vectors $\mathbf{e}_j$ and $\mathbf{e}_k$ are orthogonal if $k \neq j$. The notion of orthogonality is fundamental in signal processing and we shall return to it repeatedly in what follows. The idea of using the dot product to measure how similar two vectors are is called *matched filtering*; it is a popular method in signal detection and estimation of parameters.

Proof of Cauchy's inequality: To prove Cauchy's inequality for the complex vector dot product we write $\mathbf{u} \cdot \mathbf{v} = |\mathbf{u} \cdot \mathbf{v}|e^{i\theta}$. Let t be a real variable and consider

$$\begin{aligned} 0 \leq ||e^{-i\theta}\mathbf{u} - t\mathbf{v}||^2 &= (e^{-i\theta}\mathbf{u} - t\mathbf{v}) \cdot (e^{-i\theta}\mathbf{u} - t\mathbf{v}) \\ &= ||\mathbf{u}||^2 - t[(e^{-i\theta}\mathbf{u}) \cdot \mathbf{v} + \mathbf{v} \cdot (e^{-i\theta}\mathbf{u})] + t^2||\mathbf{v}||^2 \\ &= ||\mathbf{u}||^2 - t[(e^{-i\theta}\mathbf{u}) \cdot \mathbf{v} + \overline{(e^{-i\theta}\mathbf{u}) \cdot \mathbf{v}}] + t^2||\mathbf{v}||^2 \\ &= ||\mathbf{u}||^2 - 2Re(te^{-i\theta}(\mathbf{u} \cdot \mathbf{v})) + t^2||\mathbf{v}||^2 \\ = ||\mathbf{u}||^2 - 2Re(t|\mathbf{u} \cdot \mathbf{v}|) + t^2||\mathbf{v}||^2 &= ||\mathbf{u}||^2 - 2t|\mathbf{u} \cdot \mathbf{v}| + t^2||\mathbf{v}||^2. \end{aligned}$$

This is a nonnegative quadratic polynomial in the variable t, so cannot have two distinct real roots. Therefore, the discriminant $4|\mathbf{u} \cdot \mathbf{v}|^2 - 4||\mathbf{v}||^2||\mathbf{u}||^2$ must be non-positive; that is, $|\mathbf{u} \cdot \mathbf{v}|^2 \leq ||\mathbf{u}||^2||\mathbf{v}||^2$. This is Cauchy's inequality.

Exercise 1: Use Cauchy's inequality to show that

$$||\mathbf{u} + \mathbf{v}|| \leq ||\mathbf{u}|| + ||\mathbf{v}||;$$

this is called the *triangle inequality*.

A careful examination of the proof just presented shows that we did not explicitly use the definition of the complex vector dot product, but only certain of its properties. This suggested to mathematicians the possibility of abstracting these properties and using them to define a more general concept, an *inner product*, between objects more general than complex vectors, such as infinite sequences, random variables and matrices. Such an inner product can then be used to define the *norm* of these objects and thereby a distance between such objects. Once we have an inner product defined we also have available the notions of orthogonality and best approximation. We shall treat all of these topics in a later chapter.

Chapter 9

Orthogonal Vectors

Consider the problem of writing the two-dimensional real vector $(3, -2)$ as a linear combination of the vectors $(1, 1)$ and $(1, -1)$; that is, we want to find constants a and b so that $(3, -2) = a(1, 1) + b(1, -1)$. One way to do this, of course, is to compare the components: $3 = a + b$ and $-2 = a - b$; we can then solve this simple system for the a and b. In higher dimensions this way of doing it becomes harder, however. A second way is to make use of the dot product and orthogonality.

The dot product of two vectors (x, y) and (w, z) in R^2 is $(x, y) \cdot (w, z) = xw+yz$. If the dot product is zero then the vectors are said to be *orthogonal*; the two vectors $(1, 1)$ and $(1, -1)$ are orthogonal. We take the dot product of both sides of $(3, -2) = a(1, 1) + b(1, -1)$ with $(1, 1)$ to get

$$1 = (3, -2) \cdot (1, 1) = a(1, 1) \cdot (1, 1) + b(1, -1) \cdot (1, 1) = a(1, 1) \cdot (1, 1) + 0 = 2a,$$

so we see that $a = \frac{1}{2}$. Similarly, taking the dot product of both sides with $(1, -1)$ gives

$$5 = (3, -2) \cdot (1, -1) = a(1, 1) \cdot (1, -1) + b(1, -1) \cdot (1, -1) = 2b,$$

so $b = \frac{5}{2}$. Therefore $(3, -2) = \frac{1}{2}(1, 1) + \frac{5}{2}(1, -1)$. The beauty of this approach is that it does not get much harder as we go to higher dimensions.

Since the cosine of the angle θ between vectors $\mathbf{u}$ and $\mathbf{v}$ is

$$\cos\theta = \mathbf{u} \cdot \mathbf{v} / ||\mathbf{u}|| \, ||\mathbf{v}||,$$

where $||\mathbf{u}||^2 = \mathbf{u} \cdot \mathbf{u}$, the projection of vector $\mathbf{v}$ onto the line through the origin parallel to $\mathbf{u}$ is

$$\text{Proj}_{\mathbf{u}}(\mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\mathbf{u} \cdot \mathbf{u}} \mathbf{u}.$$

Therefore the vector $\mathbf{v}$ can be written as

$$\mathbf{v} = \text{Proj}_{\mathbf{u}}(\mathbf{v}) + (\mathbf{v} - \text{Proj}_{\mathbf{u}}(\mathbf{v})),$$

where the first term on the right is parallel to **u** and the second one is orthogonal to **u**.

How do we find vectors that are mutually orthogonal? Suppose we begin with $(1,1)$. Take a second vector, say $(1,2)$, that is not parallel to $(1,1)$ and write it as we did **v** earlier; that is, as a sum of two vectors, one parallel to $(1,1)$ and the second orthogonal to $(1,1)$. The projection of $(1,2)$ onto the line parallel to $(1,1)$ passing through the origin is

$$\frac{(1,1)\cdot(1,2)}{(1,1)\cdot(1,1)}(1,1) = \frac{3}{2}(1,1) = (\frac{3}{2},\frac{3}{2})$$

so

$$(1,2) = (\frac{3}{2},\frac{3}{2}) + ((1,2) - (\frac{3}{2},\frac{3}{2})) = (\frac{3}{2},\frac{3}{2}) + (-\frac{1}{2},\frac{1}{2}).$$

The vectors $(-\frac{1}{2},\frac{1}{2}) = -\frac{1}{2}(1,-1)$ and, therefore, $(1,-1)$ are then orthogonal to $(1,1)$. This approach is the basis for the *Gram-Schmidt* method for constructing a set of mutually orthogonal vectors.

Exercise 1: Use the Gram-Schmidt approach to find a third vector in R^3 orthogonal to both $(1,1,1)$ and $(1,0,-1)$.

Orthogonality is a convenient tool that can be exploited whenever we have an inner product defined.

Chapter 10

Discrete Linear Filters

Let $\mathbf{g} = (g_1, ..., g_M)^T$ be an M-dimensional complex column vector. The discrete linear filter obtained from $\mathbf{g}$ operates on any other M-dimensional column vector $\mathbf{h} = (h_1, ..., h_M)^T$ through the complex dot product: when the input of the filter is $\mathbf{h}$ the output of the filter is

$$\mathbf{g}^\dagger \mathbf{h} = \mathbf{h} \cdot \mathbf{g} = \sum\nolimits_{m=1}^{M} h_m \overline{g}_m.$$

Earlier we analyzed the signal $s(t)$ by applying the discrete linear filters $\mathbf{g} = \mathbf{e}_\omega$ to the data vector $\mathbf{d}$ to obtain the function $\mathbf{e}_\omega^\dagger \mathbf{d}$ of the variable ω. Such discrete linear filters are usually called *matched filters* because we use the dot product to determine the degree of similarity between the two vectors.

The term *discrete linear filter* also applies to the somewhat more general *convolution filter* whereby vectors $\mathbf{g}$ and $\mathbf{h}$ are used to produce a third vector $\mathbf{f} = \mathbf{g} * \mathbf{h}$, the periodic convolution of $\mathbf{g}$ and $\mathbf{h}$, whose entries f_n are

$$f_n = \sum\nolimits_{m=1}^{M} g_m h_{n-m}, \tag{10.1}$$

where, for notational convenience, we define $h_{n-m} = h_{n-m+M}$ whenever the index $n - m$ is less than one. Figure 10.1 illustrates the action of this convolution filter.

To better understand the action of this filtering operation we associate with each of the vectors $\mathbf{f}$, $\mathbf{g}$ and $\mathbf{h}$ a function of ω: let

$$DFT_{\mathbf{g}}(\omega) = \sum\nolimits_{m=1}^{M} g_m e^{im\omega}$$

for ω in the interval $[-\pi, \pi]$; similarly define the functions $DFT_{\mathbf{f}}(\omega)$ and $DFT_{\mathbf{h}}(\omega)$. Notice that these functions are the discrete Fourier transforms (DFT) discussed earlier. We have the option here of considering the vector

discrete Fourier transforms instead. However, since we shall also discuss the theoretical case in which we have doubly infinite sequences $\{f_n\}_{n=-\infty}^{\infty}$, it is more convenient to view the DFT as a function of the continuous variable ω throughout the discussion. As we saw in an earlier exercise, when $\mathbf{f} = \mathbf{g} * \mathbf{h}$ we also have

$$DFT_{\mathbf{f}}(\omega) = DFT_{\mathbf{g}}(\omega) DFT_{\mathbf{h}}(\omega)$$

for the values $\omega = \frac{2\pi}{M} n$, $n = 1, 2, ..., M$.

Time-invariant linear systems: Although in practice all digital filtering is performed using finite length vectors, it is convenient, in theoretical discussions, to permit the use of infinite sequences. Suppose now that $g = \{g_n\}_{n=-\infty}^{+\infty}$ and $h = \{h_n\}_{n=-\infty}^{+\infty}$ are infinite sequences of complex numbers. As above, we use g to obtain a convolution filter that, having h as the input, will have as output the convolution of sequences g and h. This is the infinite sequence $f = g * h$ with entries

$$f_n = \sum\nolimits_{m=-\infty}^{+\infty} g_m h_{n-m}.$$

This situation is commonly described by saying that the sequence $\{g_n\}$ represents a *time-invariant linear system* in which the input sequence is convolved with $\{g_n\}$ to produce the output sequence.

When dealing with infinite sequences we must be concerned with the convergence of any infinite series we encounter. In Walnut's book [145] and elsewhere an infinite sequence $\{h_n\}$ is called a *signal* if it is *absolutely summable*; that is,

$$\sum_{n=-\infty}^{\infty} |h_n| < +\infty.$$

The sequences $\{g_n\}$ used to define convolution filters are also required to be absolutely summable, so that the output $f = g * h$ is also absolutely summable and $\{f_n\}$ is therefore a signal. However, the requirement that all signals be absolutely summable is a bit restrictive. For that reason most authors, including Walnut, consider wider classes of sequences, such as *absolutely square summable* $h = \{h_n\}$ for which we have

$$\sum_{n=-\infty}^{\infty} |h_n|^2 < +\infty,$$

bounded sequences and sequences obtained from finitely nonzero ones by periodic extension. Concepts such as stability can be defined in different ways, depending on the type of signals being considered. Our discussion here will be more formal and less rigorous. The reader should remember

that integrals and infinite sums make sense only after appropriate assumptions are made.

We associate with doubly infinite sequences a function of ω: for each ω in the interval $[-\pi, \pi]$ let

$$G(\omega) = \sum_{n=-\infty}^{+\infty} g_n e^{in\omega}. \tag{10.2}$$

Define $F(\omega)$ and $H(\omega)$ similarly. Because the sequences are infinite we have a multiplication theorem that is somewhat stronger than with the vector DFT.

Exercise 1: Show that $F(\omega) = G(\omega)H(\omega)$ for all ω in $[-\pi, \pi]$.

We see from the exercise that the convolution filter obtained from the sequence $\{g_n\}$ can be understood in terms of how it affects the individual complex exponential components that make up the input. The filter converts each $H(\omega)$ into $F(\omega) = G(\omega)H(\omega)$. If $G(\omega) = 0$ for certain values of ω then whenever $h(t)$ has a complex exponential component corresponding to that value of ω it will be removed upon filtering.

Convolution filters have the important property that they amplify or depress sinusoidal inputs without distorting the frequency. Let ω be an arbitrary but fixed frequency in the interval $[-\pi, \pi]$ and let the input to the filter be the doubly infinite sequence h with entries $h_n = e^{-in\omega}$; that is, a pure sinusoid with frequency $-\omega$. Then the output sequence is f with entries

$$f_n = e^{-in\omega} \sum_{m=-\infty}^{\infty} g_m e^{im\omega}.$$

So the output is again a pure sinusoid, with the same frequency as the input, but with amplitude $G(\omega)$ instead of one.

The function $G(\omega)$ in equation (10.2) is a *Fourier series*. Here we began with an essentially arbitrary sequence g of complex numbers and formed the function G. In a number of applications we begin with a function $G(\omega)$ that is either defined on an interval of length 2π or is defined for all ω and is 2π-periodic. We then seek the complex numbers g_n so that the Fourier series obtained using these g_n gives us back the original function G as in equation (10.2). This is called the *Fourier series expansion* of the function $G(\omega)$.

Given the function $H(\omega)$ on $[-\pi, \pi]$ the numbers h_n can be determined: we have

$$h_n = \int_{-\pi}^{\pi} H(\omega) e^{-in\omega} \frac{d\omega}{2\pi}. \tag{10.3}$$

This follows from the orthogonality of the functions $e^{in\omega}$ over the interval $[-\pi, \pi]$, as we shall discuss in the next chapter. We can interpret equation

(10.3) as expressing the sequence $h = \{h_n\}$ as a continuously infinite superposition of pure sinusoids, each with their own frequency $-\omega$ and amplitude $H(\omega)/2\pi$. We know that the output from the individual sinusoidal input $\{e^{-in\omega}\}$ is $G(\omega)\{e^{-in\omega}\}$. By the linearity of the filter, the output from the input sequence h with entries given by equation (10.3) is therefore the sequence f with entries

$$f_n = \int_{-\pi}^{\pi} G(\omega)H(\omega)e^{-in\omega}\frac{d\omega}{2\pi}.$$

Since we also have

$$f_n = \int_{-\pi}^{\pi} F(\omega)e^{-in\omega}\frac{d\omega}{2\pi},$$

we are led once again to $F(\omega) = G(\omega)H(\omega)$.

Suppose that the input to the filter is an impulsive sequence; that is, let the input be the sequence $h = \delta^0$ with entries $h_n = 0$ for $n \neq 0$ and $h_0 = 1$. Then the output is the sequence f with entries $f_n = g_n$. The sequence $g = \{g_n\}$ used to build the discrete linear filter is therefore called the *impulse response* sequence of the filter and the function $G(\omega)$ is the *filter function*.

Exercise 2: The *three-point moving average* filter is defined as follows: given the input sequence $\{h_n, n = -\infty, ..., \infty\}$ the output sequence is $\{f_n, n = -\infty, ..., \infty\}$, with

$$f_n = (h_{n-1} + h_n + h_{n+1})/3.$$

Let $g_m = 1/3$, if $m = 0, 1, -1$ and $g_m = 0$, otherwise. Then we have

$$f_n = \sum_{m=-\infty}^{\infty} g_m h_{n-m},$$

so that f is the convolution of h and g. Let $F(\omega)$ be defined for ω in the interval $[-\pi, \pi]$ by equation (10.2); similarly define G and H. To recover h from f we might proceed as follows: calculate F, then divide F by G to get H, then compute h from H; does this always work?
If we let h be the sequence $\{..., 1, 1, 1, ...\}$ then $f = h$; if we take h to be the sequence $\{..., 3, 0, 0, 3, 0, 0, ...\}$ then we again get $f = \{..., 1, 1, 1, ...\}$. Therefore, we cannot expect to recover h from f in general. We know that $G(\omega) = \frac{1}{3}(1 + 2\cos(\omega))$; what does this have to do with the problem of recovering h from f?

Hint: Compute H. Where are the zeros of G?

If we take the input sequence to our convolution filter the sequence h with entries

$$h_n = \overline{g}_{-n}$$

then the output sequence is f with entries

$$f_n = \sum\nolimits_{m=-\infty}^{+\infty} g_m \overline{g}_{m-n}$$

and $F(\omega) = |G(\omega)|^2$. The sequence f is called the *autocorrelation sequence* for g and $|G(\omega)|^2$ is the *power spectrum* of g. The Cauchy inequality is valid for infinite sequences also: with the length of f defined by

$$||f|| = (\sum\nolimits_{n=-\infty}^{+\infty} |f_n|^2)^{1/2}$$

and the inner product of f and g given by

$$\langle f, g \rangle = \sum\nolimits_{n=-\infty}^{+\infty} f_n \overline{g}_n$$

we have

$$|\langle f, g \rangle| \leq ||f||\,||g||,$$

with equality if and only if g is a constant multiple of f.

Exercise 3: Let f be the autocorrelation sequence for g. Show that $f_{-n} = \overline{f}_n$ and $f_0 \geq |f_n|$ for all n.

The z-transform: It is common to consider the case in which the input to a time-invariant linear system $g = \{g_n\}$ is a discrete random process $\{X_n\}$; that is, each X_n is a random variable [119], [124]. The output sequence $\{Y_n\}$ given by

$$Y_n = \sum_{m=-\infty}^{+\infty} g_m X_{n-m}$$

is then a second discrete random process whose statistics are related to those of the input, as well as to properties of the sequence g. By analogy with what we did earlier, we would like to be able to form the functions

$$X(\omega) = \sum_{n=-\infty}^{+\infty} X_n e^{in\omega}$$

and

$$Y(\omega) = \sum_{n=-\infty}^{+\infty} Y_n e^{in\omega}$$

and use them to study the action of the system on random input. For the series for $X(\omega)$ to converge we would at least want

$$\sum_{n=-\infty}^{+\infty} |X_n|^2 < +\infty.$$

This poses a problem, because the random processes $\{X_n\}$ we usually consider do not go to zero as $|n| \to +\infty$. For this reason we need a somewhat more general tool, the z-transform.

Given a doubly infinite sequence sequence $g = \{g_n\}_{n=-\infty}^{+\infty}$ we associate with g its *z-transform*, the function of the complex variable z given by

$$G(z) = \sum\nolimits_{n=-\infty}^{+\infty} g_n z^{-n}.$$

Doubly infinite series of this form are called *Laurent series* and occur in the representation of functions analytic in an annulus. Note that if we take $z = e^{-i\omega}$ then $G(z)$ becomes $G(\omega)$ as defined by equation (10.2). The z-transform is a somewhat more flexible tool in that we are not restricted to those sequence g for which the z-transform is defined for $z = e^{-i\omega}$.

The linear system determined by g is said to be *stable* [117] if the output sequence is bounded in absolute value whenever the input sequence is.

Exercise 4: Show that the linear system determined by g is stable if and only if $\sum_{n=-\infty}^{+\infty} |g_n| < +\infty$.

Hint: If $\sum_{n=-\infty}^{+\infty} |g_n| = +\infty$, consider as input the bounded sequence $f_n = \overline{g_{-n}}/|g_n|$ and show that $h_0 = +\infty$.

Exercise 5: Consider the linear system determined by the sequence $g_0 = 2$, $g_n = (\frac{1}{2})^{|n|}$, for $n \neq 0$. Show that this system is stable. Calculate the z-transform of $\{g_n\}$ and determine its region of convergence.

The time-invariant linear system determined by g is said to be a *causal system* if the sequence $\{g_n\}$ is itself causal; that is, $g_n = 0$ for $n < 0$.

Exercise 6: Show that the function $G(z) = (z - z_0)^{-1}$ is the z-transform of a causal sequence g, where z_0 is a fixed complex number. What is the region of convergence? Show that the resulting linear system is stable if and only if $|z_0| < 1$.

Continuous time-invariant linear systems: An *operator* T associates with function f another function Tf. For example, Tf could be the derivative of f, if f is differentiable, or Tf could be F, the Fourier transform of f. The operator T is called *linear* if $T(f + h) = Tf + Th$ and

$T(\alpha f) = \alpha T f$ for any functions f and h and scalar α. For any real number τ let $f_\tau(t) = f(t+\tau)$. We say that T is *time-invariant* if $h = Tf$ implies that $h_\tau = Tf_\tau$. Suppose we fix a function g and define $Tf = f * g$; such an operator is called a *convolution operator*. Convolution operators are linear and time-invariant. As we shall see, time-invariant linear systems are convolution operators.

Exercise 7: Let $f(t) = e^{-i\omega t}$ for some fixed real number ω. Let $h = Tf$, where T is linear and time-invariant. Show that there is a constant c so that $h(t) = cf(t)$. Since the constant c may depend on ω we rewrite c as $G(\omega)$.

Exercise 8: Let T be as in the previous exercise. For

$$f(t) = \int_{-\infty}^{+\infty} F(\omega) e^{-i\omega t} d\omega / 2\pi$$

and $h = Tf$ show that $H(\omega) = F(\omega)G(\omega)$ for each ω. Conclude that T is a convolution operator whose function $g(t)$ is the inverse FT of $G(\omega)$.

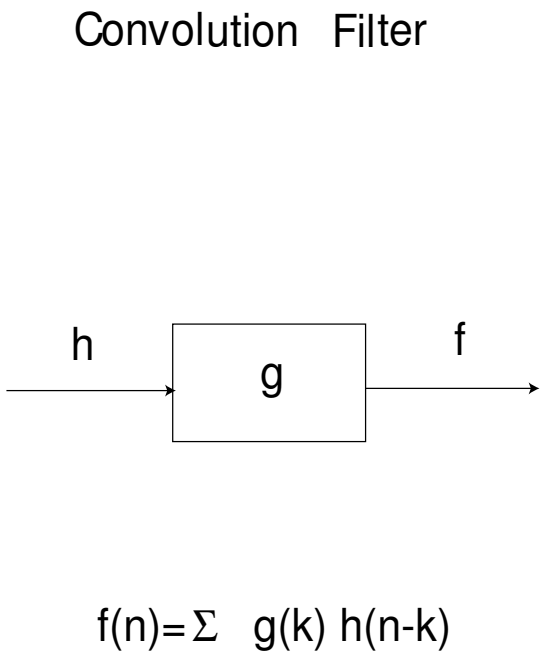


Figure 10.1: Convolution filter g operating on input h to produce out put f.

Chapter 11

Inner Products

The proof of Cauchy's inequality rests not on the actual definition of the complex vector dot product, but rather on four of its most basic properties. We use these properties to extend the concept of complex vector dot product to that of *inner product*. Later in this chapter we shall give several examples of inner products, applied to a variety of mathematical objects, including infinite sequences, functions, random variables and matrices. For now, let us denote our mathematical objects by $\mathbf{u}$ and $\mathbf{v}$ and the inner product between them as $\langle \mathbf{u}, \mathbf{v} \rangle$. The objects will then be said to be members of an *inner product space*. We are interested in inner products because they provide a notion of orthogonality, which is fundamental to best approximation and optimal estimation.

Defining an inner product: The four basic properties that will serve to define an inner product are as follows:

1: $\langle \mathbf{u}, \mathbf{u} \rangle \geq 0$, with equality if and only if $\mathbf{u} = \mathbf{0}$;

2. $\langle \mathbf{v}, \mathbf{u} \rangle = \overline{\langle \mathbf{u}, \mathbf{v} \rangle}$;

3. $\langle \mathbf{u}, \mathbf{v} + \mathbf{w} \rangle = \langle \mathbf{u}, \mathbf{v} \rangle + \langle \mathbf{u}, \mathbf{w} \rangle$;

4. $\langle c\mathbf{u}, \mathbf{v} \rangle = c\langle \mathbf{u}, \mathbf{v} \rangle$ for any complex number c.

The inner product is the basic ingredient in Hilbert space theory. Using the inner product, we define the *norm* of $\mathbf{u}$ to be

$$||\mathbf{u}|| = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle}$$

and the distance between $\mathbf{u}$ and $\mathbf{v}$ to be $||\mathbf{u} - \mathbf{v}||$.

The Cauchy-Schwarz inequality: Because these four properties were all we needed to prove the Cauchy inequality for the complex vector dot product, we obtain the same inequality whenever we have an inner product. This more general inequality is the Cauchy-Schwarz inequality:

$$|\langle \mathbf{u}, \mathbf{v} \rangle| \leq \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle} \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle}$$

or

$$|\langle \mathbf{u}, \mathbf{v} \rangle| \leq ||\mathbf{u}|| \, ||\mathbf{v}||,$$

with equality if and only if there is a scalar c such that $\mathbf{v} = c\mathbf{u}$. We say that the vectors $\mathbf{u}$ and $\mathbf{v}$ are *orthogonal* if $\langle \mathbf{u}, \mathbf{v} \rangle = 0$. We turn now to some examples.

Inner products of infinite sequences: Let $\mathbf{u} = \{u_n\}$ and $\mathbf{v} = \{v_n\}$ be infinite sequences of complex numbers. The inner product is then

$$\langle \mathbf{u}, \mathbf{v} \rangle = \sum u_n \overline{v_n},$$

and

$$||\mathbf{u}|| = \sqrt{\sum |u_n|^2}.$$

The sums are assumed to be finite; the index of summation n is singly or doubly infinite, depending on the context. The Cauchy-Schwarz inequality says that

$$|\sum u_n \overline{v_n}| \leq \sqrt{\sum |u_n|^2} \sqrt{\sum |v_n|^2}.$$

Inner product of functions: Now suppose that $\mathbf{u} = f(x)$ and $\mathbf{v} = g(x)$. Then

$$\langle \mathbf{u}, \mathbf{v} \rangle = \int f(x) \overline{g(x)} dx$$

and

$$||\mathbf{u}|| = \sqrt{\int |f(x)|^2 dx}.$$

The integrals are assumed to be finite; the limits of integration depend on the support of the functions involved. The Cauchy-Schwarz inequality now says that

$$|\int f(x) \overline{g(x)} dx| \leq \sqrt{\int |f(x)|^2 dx} \sqrt{\int |g(x)|^2 dx}.$$

Inner product of random variables: Now suppose that $\mathbf{u} = X$ and $\mathbf{v} = Y$ are random variables. Then

$$\langle \mathbf{u}, \mathbf{v} \rangle = E(X\overline{Y})$$

and

$$||\mathbf{u}|| = \sqrt{E(|X|^2)},$$

which is the standard deviation of X if the mean of X is zero. The expected values are assumed to be finite. The Cauchy-Schwarz inequality now says that

$$|E(X\overline{Y})| \leq \sqrt{E(|X|^2)}\sqrt{E(|Y|^2)}.$$

If $E(X) = 0$ and $E(Y) = 0$ the random variables X and Y are orthogonal if and only if they are *uncorrelated*.

Inner product of complex matrices: Now suppose that $\mathbf{u} = A$ and $\mathbf{v} = B$ are complex matrices. Then

$$\langle \mathbf{u}, \mathbf{v} \rangle = \text{trace}(B^\dagger A)$$

and

$$||\mathbf{u}|| = \sqrt{\text{trace}(A^\dagger A)},$$

where the trace of a square matrix is the sum of the entries on the main diagonal. As we shall see later, this inner product is simply the complex vector dot product of the vectorized versions of the matrices involved. The Cauchy-Schwarz inequality now says that

$$|\text{trace}(B^\dagger A)| \leq \sqrt{\text{trace}(A^\dagger A)}\sqrt{\text{trace}(B^\dagger B)}.$$

Weighted inner products of complex vectors: Let $\mathbf{u}$ and $\mathbf{v}$ be complex vectors and let Q be a Hermitian positive-definite matrix; that is, $Q^\dagger = Q$ and $\mathbf{u}^\dagger Q\mathbf{u} > 0$ for all nonzero vectors $\mathbf{u}$ .The inner product is then

$$\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{v}^\dagger Q\mathbf{u}$$

and

$$||\mathbf{u}|| = \sqrt{\mathbf{u}^\dagger Q\mathbf{u}}.$$

We know from the eigenvector decomposition of Q that $Q = C^\dagger C$ for some matrix C. Therefore the inner product is simply the complex vector dot product of the vectors $C\mathbf{u}$ and $C\mathbf{v}$. The Cauchy-Schwarz inequality says that

$$|\mathbf{v}^\dagger Q\mathbf{u}| \leq \sqrt{\mathbf{u}^\dagger Q\mathbf{u}}\sqrt{\mathbf{v}^\dagger Q\mathbf{v}}.$$

The weighted inner product of functions: Now suppose that $\mathbf{u} = f(x)$ and $\mathbf{v} = g(x)$ and $w(x) > 0$. Then define

$$\langle \mathbf{u}, \mathbf{v} \rangle = \int f(x)\overline{g(x)}w(x)dx$$

and

$$||\mathbf{u}|| = \sqrt{\int |f(x)|^2 w(x)dx}.$$

The integrals are assumed to be finite; the limits of integration depend on the support of the functions involved. This inner product is simply the inner product of the functions $f(x)\sqrt{w(x)}$ and $g(x)\sqrt{w(x)}$. The Cauchy-Schwarz inequality now says that

$$|\int f(x)\overline{g(x)}w(x)dx| \leq \sqrt{\int |f(x)|^2 w(x)dx}\sqrt{\int |g(x)|^2 w(x)dx}.$$

Once we have an inner product defined we can speak about orthogonality and best approximation. Important in that regard is the orthogonality principle, the topic of the next chapter.

Chapter 12

The Orthogonality Principle

Imagine that you are standing and looking down at the floor. The point B on the floor that is closest to N, the tip of your nose, is the unique point on the floor such that the vector from B to any other point A on the floor is perpendicular to the vector from N to B; that is, $\langle BN, BA\rangle = 0$. This is a simple illustration of the *orthogonality principle*. Whenever we have an inner product defined we can speak of orthogonality and apply the orthogonality principle to find best approximations.

The orthogonality principle: Let $\mathbf{u}$ and $\mathbf{v}^1, ..., \mathbf{v}^N$ be members of an inner product space. For all choices of scalars $a_1, ..., a_N$ we can compute the distance from $\mathbf{u}$ to the member $a_1\mathbf{v}^1 + ...a_N\mathbf{v}^N$. Then we minimize this distance over all choices of the scalars; let $b_1, ..., b_N$ be this best choice. The *orthogonality principle* tells us that the member $\mathbf{u} - (b_1\mathbf{v}^1 + ...b_N\mathbf{v}^N)$ is orthogonal to the member $(a_1\mathbf{v}^1 + ... + a_N\mathbf{v}^N) - (b_1\mathbf{v}^1 + ...b_N\mathbf{v}^N)$, that is,

$$\langle \mathbf{u} - (b_1\mathbf{v}^1 + ...b_N\mathbf{v}^N), (a_1\mathbf{v}^1 + ... + a_N\mathbf{v}^N) - (b_1\mathbf{v}^1 + ...b_N\mathbf{v}^N) = 0,$$

for every choice of scalars a_n. We can then use the orthogonality principle to find the best choice $b_1., , , .b_N$.

For each fixed index value j in the set $\{1, ..., N\}$ let $a_n = b_n$ if j is not equal to n and $a_j = b_j + 1$. Then we have

$$0 = \langle \mathbf{u} - (b_1\mathbf{v}^1 + ...b_N\mathbf{v}^N), \mathbf{v}^j\rangle,$$

or

$$\langle \mathbf{u}, \mathbf{v}^j\rangle = \sum\nolimits_{n=1}^{N} b_n\langle \mathbf{v}^n, \mathbf{v}^j\rangle,$$

for each j. The $\mathbf{v}^n$ are known, so we can calculate the inner products $\langle \mathbf{v}^n, \mathbf{v}^j \rangle$ and solve this system of equations for the best b_n.

We shall encounter a number of particular cases of the orthogonality principle in subsequent chapters. The example of the *least squares* solution of a system of linear equations provides a good example of the use of this principle.

The least squares solution: Let $V\mathbf{a} = \mathbf{u}$ be a system of M linear equations in N unknowns. For $n = 1, ..., N$ let $\mathbf{v}^n$ be the n-th column of the matrix V . For any choice of the vector $\mathbf{a}$ with entries a_n, $n = 1, ..., N$ the vector $V\mathbf{a}$ is

$$V\mathbf{a} = \sum\nolimits_{n=1}^{N} a_n \mathbf{v}^n.$$

Solving $V\mathbf{a} = \mathbf{u}$ amounts to representing the vector $\mathbf{u}$ as a linear combination of the columns of V .

If there is no solution of $V\mathbf{a} = \mathbf{u}$ then we can look for the best choice of coefficients so as to minimize the distance $||\mathbf{u} - (a_1\mathbf{v}^1 + ... + a_N\mathbf{v}^N)||$. The matrix with entries $\langle \mathbf{v}^n, \mathbf{v}^j \rangle$ is $V^\dagger V$ and the vector with entries $\langle \mathbf{u}, \mathbf{v}^j \rangle$ is $V^\dagger \mathbf{u}$. According to the orthogonality principle we must solve the system of equations $V^\dagger \mathbf{u} = V^\dagger V\mathbf{a}$, which leads to the least squares solution.

Exercise 1: Find polynomial functions $f(x)$, $g(x)$ and $h(x)$ that are orthogonal on the interval $[0, 1]$ and have the property that every polynomial of degree two or less can be written as a linear combination of these three functions.

Exercise 2: Show that the functions e^{inx}, n an integer, are orthogonal on the interval $[-\pi, \pi]$. Let $f(x)$ have the Fourier expansion

$$f(x) = \sum\nolimits_{n=-\infty}^{\infty} a_n e^{inx},\ |x| \leq \pi.$$

Use orthogonality to find the coefficients a_n.

We have seen that orthogonality can be used to determine the coefficients in the Fourier series representation of a function. There are other useful representations in which orthogonality also plays a role; wavelets is one such. Let $f(x)$ be defined on the closed interval $[0, X]$. Suppose that we change the function $f(x)$ to a new function $g(x)$ by altering the values for x within a small interval, keeping the remaining values the same: then all of the Fourier coefficients change. Looked at another way, a localized disturbance in the function $f(x)$ affects all of its Fourier coefficients. It would be helpful to be able to represent $f(x)$ as a sum of orthogonal functions in such a way that localized changes in $f(x)$ affect only a small number of the components in the sum. One way to do this is with wavelets, as we shall see shortly.

Chapter 13

Fourier Transforms and Fourier Series

In a previous chapter we studied the problem of isolating the individual complex exponential components of the signal function $s(t)$, given the data vector $\mathbf{d}$ with entries $s(m\Delta)$, $m = 1, ..., M$, where $s(t)$ is

$$s(t) = \sum\nolimits_{n=1}^{N} A_n e^{i\omega_n t};$$

we assume that $|\omega_n| < \pi/\Delta$. The second approach we considered involved calculating the function

$$DFT_{\mathbf{d}}(\omega) = \sum\nolimits_{m=1}^{M} s(m\Delta)e^{-i\omega m\Delta}$$

for $|\omega| < \pi/\Delta$. This sum is an example of a (finite) Fourier series. As we just saw, we can extend the concept of Fourier series to include infinite sums. In fact, we can generalize to summing over a continuous variable, using integrals in place of summation; this is what is done in the definition of the Fourier transform.

The Fourier transform:

In our discussion of linear filtering we saw that if f is a finite vector $\mathbf{f} = (f_1, ..., f_M)^T$ or an infinite sequence $f = \{f_m\}_{m=-\infty}^{+\infty}$ then it is convenient to consider the function $F(\omega)$ defined for $|\omega| \leq \pi$ by the finite or infinite Fourier series expression

$$F(\omega) = \sum f_m e^{im\omega}.$$

If $f(x)$ is a function of the real variable x, we can associate with f the function $F(\omega)$, the *Fourier transform* (FT) of $f(x)$, defined for all real ω

by

$$F(\omega) = \int f(x) e^{ix\omega} dx. \tag{13.1}$$

Once we have $F(\omega)$ we can recover $f(x)$ as the *inverse Fourier transform* (IFT) of $F(\omega)$:

$$f(x) = \int F(\omega) e^{-ix\omega} d\omega / 2\pi. \tag{13.2}$$

We say then that the functions f and F form a Fourier transform pair. It may happen that one or both of the integrals above will fail to be defined in the usual way and will be interpreted as the principal value of the integral [78].

Note that the definitions of the FT and IFT just given may differ slightly from the ones found elsewhere. The differences are minor and involve only the placement of the quantity 2π and the minus sign in the exponent. One sometimes sees the FT of the function f denoted $\hat{f}$; here we shall reserve the symbol $\hat{f}$ for estimates of the function f.

As an example of a Fourier transform pair let $F(\omega)$ be the function $\chi_\Omega(\omega)$ that equals one for $|\omega| \leq \Omega$ and is zero otherwise. Then the inverse Fourier transform of $\chi_\Omega(\omega)$ is

$$f(x) = \int_{-\Omega}^{\Omega} e^{-i\omega x} d\omega / 2\pi = \frac{\sin(\Omega x)}{\pi x}.$$

The function $\frac{\sin(x)}{x}$ is called the *sinc* function, $\mathrm{sinc}\,(x)$.

Fourier series:

If there is a positive Ω such that the Fourier transform $F(\omega)$ of the function $f(x)$ is zero for $|\omega| > \Omega$ then the function $f(x)$ is said to be Ω-*bandlimited* and $F(\omega)$ has *bandwidth* Ω; in this case the function $F(\omega)$ can be written, on the interval $[-\Omega, \Omega]$, as an infinite discrete sum of complex exponentials. For $|\omega| \leq \Omega$ we have

$$F(\omega) = \sum\nolimits_{n=-\infty}^{+\infty} f_n e^{in\omega \frac{\pi}{\Omega}}. \tag{13.3}$$

We determine the coefficients f_n in much the same way as in earlier discussions.

We know that the integral

$$\int_{-\Omega}^{\Omega} e^{i(n-m)\omega \frac{\pi}{\Omega}} d\omega$$

equals zero if $m \neq n$ and equals 2Ω for $m = n$. Therefore,

$$f_m = \frac{1}{2\Omega} \int_{-\Omega}^{\Omega} F(\omega) e^{-im\omega \frac{\pi}{\Omega}} d\omega \tag{13.4}$$

for each integer m. If we wish, we can also write the coefficient f_m in terms of the inverse Fourier transform $f(x)$ of the function $F(\omega)$: the right side of equation (13.4) also equals $\frac{\pi}{\Omega} f(m\frac{\pi}{\Omega})$, from which we conclude that $f_m = \frac{\pi}{\Omega} f(m\frac{\pi}{\Omega})$.

The Shannon Sampling Theorem: Now that we have found the coefficients of the Fourier series for $F(\omega)$ we can write

$$F(\omega) = \frac{\pi}{\Omega} \sum_{n=-\infty}^{\infty} f(n\frac{\pi}{\Omega}) e^{in\omega \frac{\pi}{\Omega}} \tag{13.5}$$

for $|\omega| \leq \Omega$. We apply the formula in equation (13.2) to get

$$f(x) = \sum_{n=-\infty}^{\infty} f(n\frac{\pi}{\Omega}) \frac{\sin(\Omega x - n\pi)}{\Omega x - n\pi}. \tag{13.6}$$

This is the famous *Shannon sampling theorem*, which tells us that if $F(\omega)$ is zero outside $[-\Omega, \Omega]$, then $f(x)$ is completely determined by the infinite sequence of values $\{f(n\frac{\pi}{\Omega})\}_{n=-\infty}^{+\infty}$. If $F(\omega)$ is continuous and $F(-\Omega) = F(\Omega)$ then $F(\omega)$ has a continuous periodic extension to all of the real line. Then the Fourier series in equation (13.3) converges to $F(\omega)$ for every ω at which the function $F(\omega)$ has a left and right derivative. In general, if $F(-\Omega) \neq F(\Omega)$, or if $F(\omega)$ is discontinuous for some ω in $(-\Omega, \Omega)$, the series will still converge, but to the average of the one-sided limits $F(\omega + 0)$ and $F(\omega - 0)$, again, provided that $F(\omega)$ has one-sided derivatives at that point. If

$$\int_{-\Omega}^{\Omega} |F(\omega)|^2 d\omega < \infty$$

then

$$\sum\nolimits_{n=-\infty}^{+\infty} |f(n\frac{\pi}{\Omega}|^2 < \infty$$

and the series in equation (13.6) converges to $f(x)$ in the L^2 sense. If, in addition, we have

$$\sum\nolimits_{n=-\infty}^{+\infty} |f(n\frac{\pi}{\Omega}| < \infty,$$

then the series converges uniformly to $f(x)$ for x on the real line. There are many books that can be consulted for details concerning convergence of Fourier series, such as [13] and [78].

Let $f = \{f_m\}$ and $g = \{g_m\}$ be the sequences of Fourier coeffcients for the functions $F(\omega)$ and $G(\omega)$, respectively, defined on the interval $[-\pi, \pi]$; that is

$$F(\omega) = \sum_{m=-\infty}^{\infty} f_m e^{im\omega}, |\omega| \leq \pi.$$

Exercise 1: Use the orthogonality of the functions $e^{im\omega}$ on $[-\pi, \pi]$ to establish *Parseval's equation*:

$$\langle f, g \rangle = \sum_{m=-\infty}^{\infty} f_m \overline{g_m} = \int_{-\pi}^{\pi} F(\omega)\overline{G(\omega)} d\omega/2\pi,$$

from which it follows that

$$\langle f, f \rangle = \int_{-\infty}^{\infty} |F(\omega)|^2 d\omega/2\pi.$$

Similar results hold for the Fourier transform, as we shall see in the next chapter.

Exercise 2: Let $f(x)$ be defined for all real x and let $F(\omega)$ be its FT. Let

$$g(x) = \sum_{k=-\infty}^{\infty} f(x + 2\pi k),$$

assuming the sum exists. Show that g is a 2π -periodic function. Compute its Fourier series and use it to derive the *Poisson summation formula*:

$$\sum_{k=-\infty}^{\infty} f(2\pi k) = \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} F(n).$$

In certain applications our main interest is the function $f(x)$, for which we have finitely many (usually noisy) values. For example, x may be the time variable t and $f(t)$ may be a short segment of spoken speech that we wish to analyze. We model $f(t)$ as a finite, infinite discrete or continuous sum of complex exponentials, that is, as a Fourier series or Fourier transform, in order to process the data, to remove the noise, to compress the data and to identify the parameters.

In remote sensing applications (such as radar, sonar, tomography), on the other hand, we have again noisy values of $f(x)$, but it is not $f(x)$ that interests us. Instead, we are interested in $F(\omega)$, the Fourier transform of $f(x)$ or the sequence F_n of the complex Fourier coeffcients of $f(x)$, if $f(x) = 0$ outside some finite interval. We cannot measure these quantities directly, so we must content ourselves with estimating them from our measurements of $f(x)$.

In yet a third class of applications, such as linear filtering, we are concerned with constructing a digital procedure for performing certain operations on any signal we might receive as input. In such cases our goal is to construct the sequence g_n for which the associated Fourier series $G(\omega)$ will have a desired shape. For example, we may want the filter to eliminate all complex exponential components of the input signal whose frequency is not in the interval $[-\Omega, \Omega]$. Then we would want $G(\omega)$ to be one for ω within this interval and zero outside. To achieve this we would take the sequence g_n to be

$$g_n = \frac{\sin(\Omega n)}{\pi n}.$$

In these applications there is no $f(x)$ to be analyzed nor $F(\omega)$ to be estimated.

Chapter 14

More on the Fourier Transform

We begin with exercises that treat basic properties of the FT and then introduce several examples of Fourier transform pairs.

Exercise 1: Let $F(\omega)$ be the FT of the function $f(x)$. Use the definitions of the FT and IFT given in equations (13.1) and (13.2) to establish the following basic properties of the Fourier transform operation:

Symmetry: The FT of the function $F(x)$ is $2\pi f(-\omega)$. For example, the FT of the function $f(x) = \frac{\sin(\Omega x)}{\pi x}$ is $\chi_\Omega(\omega)$, so the FT of $g(x) = \chi_\Omega(x)$ is $G(\omega) = 2\pi\frac{\sin(\Omega\omega)}{\pi\omega}$.

Conjugation: The FT of $\overline{f(x)}$ is $\overline{F(-\omega)}$.

Scaling: The FT of $f(ax)$ is $\frac{1}{|a|}F(\frac{\omega}{a})$ for any nonzero constant a.

Shifting: The FT of $f(x-a)$ is $e^{-ia\omega}F(\omega)$.

Modulation: The FT of $f(x)\cos(\omega_0 x)$ is $\frac{1}{2}[F(\omega+\omega_0) + F(\omega-\omega_0)]$.

Differentiation: The FT of the n-th derivative, $f^{(n)}(x)$ is $(-i\omega)^n F(\omega)$. The IFT of $F^{(n)}(\omega)$ is $(ix)^n f(x)$.

Convolution in x: Let f, F, g, G and h, H be FT pairs, with

$$h(x) = \int f(y)g(x-y)dy,$$

so that $h(x) = (f * g)(x)$ is the convolution of $f(x)$ and $g(x)$. Then $H(\omega) = F(\omega)G(\omega)$. For example, if we take $g(x) = \overline{f(-x)}$, then

$$h(x) = \int f(x+y)\overline{f(y)}dy = \int f(y)\overline{f(y-x)}dy = r_f(x)$$

is the *autocorrelation function* associated with $f(x)$ and

$$H(\omega) = |F(\omega)|^2 = R_f(\omega) \geq 0$$

is the *power spectrum* of $f(x)$.

Convolution in ω: Let f, F, g, G and h, H be FT pairs, with $h(x) = f(x)g(x)$. Then $H(\omega) = \frac{1}{2\pi}(F * G)(\omega)$.

Exercise 2: Show that the Fourier transform of $f(x) = e^{-\alpha^2 x^2}$ is $F(\omega) = \frac{\sqrt{\pi}}{\alpha}e^{-(\frac{\omega}{2\alpha})^2}$. Hint: Calculate the derivative $F'(\omega)$ by differentiating under the integral sign in the definition of F and integrating by parts. Then solve the resulting differential equation.

Let $u(x)$ be the *Heaviside function* that is $+1$ if $x \geq 0$ and 0 otherwise. Let $\chi_X(x)$ be the *characteristic function* of the interval $[-X, X]$ that is $+1$ for x in $[-X, X]$ and 0 otherwise. Let $\text{sgn}(x)$ be the *sign function* that is $+1$ if $x > 0$, -1 if $x < 0$ and zero for $x = 0$. Denote by $\delta(\omega)$ the *Dirac delta* 'function' that is defined formally as the FT of the constant function that has the value $\frac{1}{2\pi}$ for all x.

Exercise 3: Calculate the FT of the function $f(x) = u(x)e^{-ax}$, where a is a positive constant.

Exercise 4: Calculate the FT of $f(x) = \chi_X(x)$.

Exercise 5: Show that the IFT of the function $F(\omega) = 2i/\omega$ is $f(x) = \text{sgn}(x)$. Hints: write the formula for the inverse Fourier transform of $F(\omega)$ as

$$f(x) = \frac{1}{2\pi}\int_{-\infty}^{+\infty}\frac{2i}{\omega}\cos\omega x d\omega - \frac{i}{2\pi}\int_{-\infty}^{+\infty}\frac{2i}{\omega}\sin\omega x d\omega$$

which reduces to

$$f(x) = \frac{1}{\pi}\int_{-\infty}^{+\infty}\frac{1}{\omega}\sin\omega x d\omega,$$

since the integrand of the first integral is odd. For $x > 0$ consider the Fourier transform of the function $\chi_x(t)$. For $x < 0$ perform the change of variables $u = -x$.

Exercise 6: Use the fact that $\text{sgn}(x) = 2u(x) - 1$ and the previous exercise to show that $f(x) = u(x)$ has the FT $F(\omega) = i/\omega + \pi\delta(\omega)$.

Let $f(x)$ be arbitrary and $F(\omega)$ its Fourier transform. The complex numbers $f(x)$ and $F(\omega)$ have real and imaginary parts; we wish to see how these are related.

Exercise 7: Let $F(\omega) = R(\omega) + iX(\omega)$, where R and X are real-valued functions, and similarly, let $f(x) = f_1(x) + if_2(x)$, where f_1 and f_2 are real-valued. Find relationships between the pairs R,X and f_1,f_2.

We saw earlier that the $F(\omega) = \chi_\Omega(\omega)$ has for its inverse Fourier transform the function $f(x) = \frac{\sin \Omega x}{\pi x}$; note that $f(0) = \frac{\Omega}{\pi}$ and $f(x) = 0$ for the first time when $\Omega x = \pi$ or $x = \frac{\pi}{\Omega}$. Therefore, as Ω grows larger, $f(0)$ approaches $+\infty$, while $f(x)$ goes to zero for $x \neq 0$. The limit is therefore not a function; it is a *generalized function* called *the Dirac delta function at zero*, denoted $\delta(x)$. The FT of $\delta(x)$ is the function $F(\omega) = 1$ for all ω. The Dirac delta function $\delta(x)$ enjoys the *sifting property*: for any other function $g(x)$ we have

$$g(0) = \int_{-\infty}^{\infty} g(x)\delta(x)dx.$$

The generalized function $\delta(x - x_0)$ then has the property that

$$g(x_0) = \int_{-\infty}^{\infty} g(x)\delta(x - x_0)dx.$$

It follows from the sifting and shifting properties that the FT of $\delta(x - x_0)$ is the function $e^{ix_0\omega}$.

The formula for the inverse FT nows says

$$\delta(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-ix\omega} d\omega. \tag{14.1}$$

If we try to make sense of this integral according to the rules of calculus we get stuck quickly. The problem is that the integral formula doesn't mean quite what it does ordinarily and the $\delta(x)$ is not really a function, but an operator on functions; it is sometimes called a *distribution*. The Dirac deltas are mathematical fictions, not in the bad sense of being lies or fakes, but in the sense of being made up for some purpose. They provide helpful descriptions of impulsive forces, probability densities in which a discrete point has nonzero probability, or, in array processing, objects far enough away to be viewed as occupying a discrete point in space.

We shall treat the relationship expressed by equation (14.1) as a formal statement, rather than attempt to explain the use of the integral in what is

surely an unconventional manner. Nevertheless, it is possible to motivate this relationship by proving that, for any $x \neq 0$,

$$\int_{-\infty}^{\infty} e^{-ix\omega} d\omega = 0.$$

Assume, for convenience, that $x > 0$. Notice first that we can write

$$\int_{-\infty}^{\infty} e^{-ix\omega} d\omega = \sum_{k=-\infty}^{\infty} \int_{\frac{2\pi}{x}k}^{\frac{2\pi}{x}(k+1)} e^{-ix\omega} d\omega.$$

Since

$$e^{-ix\omega} = e^{-ix(\omega + \frac{2\pi}{x})}$$

we can write

$$\begin{aligned}
\int_{\frac{2\pi}{x}k}^{\frac{2\pi}{x}(k+1)} e^{-ix\omega} d\omega &= \int_{-\frac{\pi}{x}}^{\frac{\pi}{x}} e^{-ix\omega} d\omega \\
&= \int_{0}^{\frac{\pi}{x}} [e^{-ix\omega} + e^{-ix(\omega - \frac{\pi}{x})}] d\omega \\
&= \frac{1}{x} \int_{0}^{\pi} [e^{-i\omega}(1 + e^{i\pi})] d\omega \\
&= \frac{1}{x}(1 + e^{i\pi}) \int_{0}^{\pi} e^{-i\omega} d\omega = 0.
\end{aligned}$$

Clearly, when $x = 0$ the integrand is one for all ω, which leads to the delta function supported at zero.

If we move the discussion into the ω domain and define the Dirac delta function $\delta(\omega)$ to be the FT of the function that has the value $\frac{1}{2\pi}$ for all x, then the FT of the complex exponential function $\frac{1}{2\pi} e^{-i\omega_0 x}$ is $\delta(\omega - \omega_0)$, visualized as a "spike" at ω_0, that is, a generalized function that has the value $+\infty$ at $\omega = \omega_0$ and zero elsewhere. This is a useful result, in that it provides the motivation for considering the Fourier transform of a signal $s(t)$ containing hidden periodicities. If $s(t)$ is a sum of complex exponentials with frequencies $-\omega_n$ then its Fourier transform will consist of Dirac delta functions $\delta(\omega - \omega_n)$. If we then estimate the Fourier transform of $s(t)$ from sampled data, we are looking for the peaks in the Fourier transform that approximate the infinitely high spikes of these delta functions.

Exercise 8: Let f, F be a FT pair. Let $g(x) = \int_{-\infty}^{x} f(y) dy$. Show that the FT of $g(x)$ is $G(\omega) = \pi F(0)\delta(\omega) + \frac{iF(\omega)}{\omega}$.

Hint: For $u(x)$ the Heaviside function we have

$$\int_{-\infty}^{x} f(y) dy = \int_{-\infty}^{\infty} f(y) u(x - y) dy.$$

We can use properties of the Dirac delta functions to extend the Parseval equation to Fourier transforms, where it is usually called the *Parseval-Plancherel* equation.

Exercise 9: Let $f(x), F(\omega)$ and $g(x), G(\omega)$ be Fourier transform pairs. Use equation (14.1) to establish the Parseval-Plancherel equation

$$\langle f, g\rangle = \int f(x)\overline{g(x)}dx = \frac{1}{2\pi}\int F(\omega)\overline{G(\omega)}d\omega,$$

from which it follows that

$$||f||^2 = \langle f, f\rangle = \int |f(x)|^2 dx = \frac{1}{2\pi}\int |F(\omega)|^2 d\omega.$$

Exercise 10: We define the *even part* of $f(x)$ to be the function

$$f_e(x) = \frac{f(x) + f(-x)}{2},$$

and the *odd part* of $f(x)$ to be

$$f_o(x) = \frac{f(x) - f(-x)}{2};$$

define F_e and F_o similarly for F the FT of f. Let $F(\omega) = R(\omega) + iX(\omega)$ be the decomposition of F into its real and imaginary parts. We say that f is a *causal function* if $f(x) = 0$ for all $x < 0$. Show that, if f is causal, then R and X are related; specifically, show that X is the *Hilbert transform* of R, that is,

$$X(\omega) = \frac{1}{\pi}\int_{-\infty}^{\infty} \frac{R(\alpha)}{\omega - \alpha}d\alpha.$$

Hint: If $f(x) = 0$ for $x < 0$ then $f(x)\text{sgn}(x) = f(x)$. Apply the convolution theorem, then compare real and imaginary parts.

Exercise 11: The one-sided *Laplace transform* (LT) of f is $\mathcal{F}$ given by

$$\mathcal{F}(z) = \int_0^{\infty} f(x)e^{-zx}dx.$$

Compute $\mathcal{F}(z)$ for $f(x) = u(x)$, the Heaviside function. Compare $\mathcal{F}(-i\omega)$ with the FT of u.

Chapter 15

Directional Transmission

An important example of the use of the DFT is the design of directional transmitting or receiving arrays of antennas. In this chapter we concentrate on the transmission case; we shall return to array processing and consider the passive or receiving case in a later chapter.

Parabolic mirrors behind car headlamps reflect the light from the bulb, concentrating it directly ahead. Whispering at one focal point of an elliptical room can be heard clearly at the other focal point. When I call to someone across the street I cup my hands in the form of a megaphone to concentrate the sound in that direction. In all these cases the transmitted signal has acquired *directionality*. In the case of the elliptical room, not only does the soft whispering reflect off the walls toward the opposite focal point, but the travel times are independent of where on the wall the reflections occur; otherwise, the differences in time would make the received sound unintelligible. Parabolic satellite dishes perform much the same function, concentrating incoming signals coherently. In this chapter we discuss the use of amplitude and phase modulation of transmitted signals to concentrate the signal power in certain directions. Following the lead of Richard Feynman in [72], we use radio broadcasting as a concrete example of the use of directional transmission.

Radio broadcasts are meant to be received and the amount of energy that reaches the receiver depends on the amount of energy put into the transmission as well as on the distance from the transmitter to the receiver. If the transmitter broadcasts a spherical wave front, with equal power in all directions, the energy in the signal is the same over the spherical wavefronts, so that the energy per unit area is proportional to the reciprocal of the surface area of the front. This means that, for omni-directional broadcasting, the energy per unit area, that is, the energy supplied to any receiver, falls off as the distance squared. The amplitude of the received signal is then proportional to the reciprocal of the distance.

Suppose you owned a radio station in Los Angeles. Most of the population resides along the north-south coast, with fewer to the east, in the desert, and fewer still to the west, in the Pacific Ocean. You might well want to transmit the radio signal in a way that concentrates most of the power north and south. But how can you do this? The answer is to broadcast directionally. By shaping the wavefront to have most of its surface area north and south you will enable to have the broadcast heard by more people without increasing the total energy in the transmission. To achieve this shaping you can use an array of multiple antennas.

Multiple antenna arrays: We place $2N + 1$ transmitting antennas a distance $\Delta > 0$ apart along an east-west axis. For convenience, let the locations of the antennas be $n\Delta$, $n = -N, ..., N$. To begin with, let us suppose that we have a fixed frequency ω and each of the transmitting antennas sends out the same signal $f_n(t) = \cos(\omega t)$. Let (x, y) be an arbitrary location on the ground and let $\mathbf{s}$ be the vector from the origin to the point (x, y). Let θ be the angle measured counterclockwise from the positive horizontal axis to the vector $\mathbf{s}$. Let D be the distance from (x, y) to the origin. Then, if (x, y) is sufficiently distant from the antennas, the distance from $n\Delta$ on the horizontal axis to (x, y) is approximately $D - n\Delta\cos(\theta)$. The signals arriving at (x, y) from the various antennas will have travelled for different times and so will be out of phase with one another to a degree that depends on the location of (x, y).

Since we are concerned only with wavefront shape, we omit for now the distance-dependence in the amplitude of the received signal. The signal received at (x, y) is proportional to

$$f(\mathbf{s}, t) = \sum_{n=-N}^{N} \cos(\omega(t - t_n)),$$

where

$$t_n = \frac{1}{c}(D - n\Delta\cos(\theta))$$

and c is the speed of propagation of the signal. Writing

$$\cos(\omega(t - t_n)) = \cos(\omega(t - \frac{D}{c}) + n\tau\cos(\theta))$$

for $\gamma = \frac{\omega\Delta}{c}$, we have

$$\cos(\omega(t-t_n)) = \cos(\omega(t-\frac{D}{c}))\cos(n\gamma\cos(\theta)) - \sin(\omega(t-\frac{D}{c}))\sin(n\gamma\cos(\theta)).$$

Therefore the signal received at (x, y) is

$$f(\mathbf{s}, t) = A(\theta)\cos(\omega(t - \frac{D}{c})) \tag{15.1}$$

for

$$A(\theta) = \frac{\sin((N+\frac{1}{2})\gamma\cos(\theta))}{\sin(\frac{1}{2}\gamma\cos(\theta))};$$

when the denominator equals zero the signal equals $(2N+1)\cos(\omega(t-\frac{D}{c}))$.

We see from equation (15.1) that the maximum power is in the north-south direction. What about the east-west direction? In order to have negligible signal power wasted in the east-west direction we want the numerator in equation (15.1) to be zero when $\theta = 0$. This means that $\Delta = \lambda/(2N+1)$, where $\lambda = 2\pi c/\omega$ is the wavelength. Recall that the wavelength for broadcast radio is tens to hundreds of meters.

Exercise 1: Graph the function $A(\theta)$ in polar coordinates for various choices of N and Δ.

Phase and Amplitude Modulation: In the previous section the signal broadcast from each of the antennas was the same. Now we look at what directionality can be obtained by using different amplitudes and phases at each of the antennas. Let the signal broadcast from the antenna at $n\Delta$ be

$$f_n(t) = |A_n|\cos(\omega t - \phi_n) = |A_n|\cos(\omega(t-\tau_n)),$$

for some amplitude $|A_n| > 0$ and phase $\phi_n = \omega\tau_n$. Now the signal received at $\mathbf{s}$ is proportional to

$$f(\mathbf{s},t) = \sum_{n=-N}^{N} |A_n|\cos(\omega(t-t_n-\tau_n)). \tag{15.2}$$

If we wish, we can repeat the calculations done earlier to see what the effect of the amplitude and phase changes is. Using complex notation simplifies things somewhat.

Let us consider a complex signal; suppose that the signal transmitted from the antenna at $n\Delta$ is $g_n(t) = |A_n|e^{i\omega(t-\tau_n)}$. Then the signal received at location $\mathbf{s}$ is proportional to

$$g(\mathbf{s},t) = \sum_{n=-N}^{N} |A_n|e^{i\omega(t-t_n-\tau_n)}.$$

Then we have

$$g(\mathbf{s},t) = B(\theta)e^{i\omega(t-\frac{D}{c})}$$

for

$$B(\theta) = \sum_{n=-N}^{N} A_n e^{-inx},$$

$A_n = |A_n|e^{-i\phi_n}$ and $x = \frac{\omega\Delta}{c}\sin(\theta)$. Note that the complex amplitude function $B(\theta)$ depends on our choices of N and Δ and takes the form of a finite Fourier series or DFT. We can design $B(\theta)$ to approximate the desired directionality by choosing the appropriate complex coefficients A_n and selecting the amplitudes $|A_n|$ and phases ϕ_n accordingly. We can generalize further by allowing the antennas to be spaced irregularly along the east-west axis, or even distributed irregularly over a two-dimensional area on the ground.

Exercise 2: Use the Fourier transform of the characteristic function of an interval to design a transmitting array that maximally concentrates signal power within the sectors northwest to northeast and southwest to southeast.

Chapter 16

The FT in Higher Dimensions

The Fourier transform is also defined for functions of several real variables $f(x_1, ..., x_N) = f(\mathbf{x})$. The multidimensional FT arises in image processing, scattering, transmission tomography, and many other areas.

We adopt the usual vector notation that ω and $\mathbf{x}$ are N-dimensional real vectors. We say that $F(\omega)$ is the *N-dimensional Fourier transform* of the possibly complex-valued function $f(\mathbf{x})$ if the following relation holds:

$$F(\omega) = \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} f(\mathbf{x}) e^{i\omega \cdot \mathbf{x}} d\mathbf{x},$$

where $\omega \cdot \mathbf{x}$ denotes the vector dot product and $d\mathbf{x} = dx_1 dx_2 ... dx_N$. In most cases we then have

$$f(\mathbf{x}) = \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} F(\omega) e^{-i\omega \cdot \mathbf{x}} d\omega / (2\pi)^N;$$

we describe this by saying that $f(\mathbf{x})$ is the *inverse Fourier transform* of $F(\omega)$.

Consider the FT of a function of two variables $f(x, y)$:

$$F(\alpha, \beta) = \int \int f(x, y) e^{i(x\alpha + y\beta)} dx dy.$$

We convert to polar coordinates using $(x, y) = r(\cos\theta, \sin\theta)$ and $(\alpha, \beta) = \rho(\cos\omega, \sin\omega)$. Then

$$F(\rho, \omega) = \int_0^{\infty} \int_{-\pi}^{\pi} f(r, \theta) e^{ir\rho\cos(\theta - \omega)} r dr d\theta. \tag{16.1}$$

Say that a function $f(x,y)$ of two variables is a *radial* function if $x^2+y^2 = x_1^2+y_1^2$ implies $f(x,y) = f(x_1,y_1)$, for all points (x,y) and (x_1,y_1); that is, $f(x,y) = g(\sqrt{x^2+y^2})$ for some function g of one variable.

Exercise 1: Show that if f is radial then its FT F is also radial. Find the FT of the radial function $f(x,y) = \frac{1}{\sqrt{x^2+y^2}}$.

Hints: Insert $f(r,\theta) = g(r)$ in equation (16.1) to obtain

$$F(\rho,\omega) = \int_0^\infty \int_{-\pi}^{\pi} g(r) e^{ir\rho\cos(\theta-\omega)} r\,dr\,d\theta$$

or

$$F(\rho,\omega) = \int_0^\infty r g(r)[\int_{-\pi}^{\pi} e^{ir\rho\cos(\theta-\omega)} d\theta]dr. \tag{16.2}$$

Show that the inner integral is independent of ω and then use the fact that

$$\int_{-\pi}^{\pi} e^{ir\rho\cos\theta} d\theta = 2\pi J_0(r\rho),$$

with J_0 the 0-th order Bessel function, to get

$$F(\rho,\omega) = H(\rho) = 2\pi \int_0^\infty r g(r) J_0(r\rho) dr. \tag{16.3}$$

The function $H(\rho)$ is called the *Hankel transform* of $g(r)$. Summarizing, we say that if $f(x,y)$ is a radial function obtained using g then its Fourier transform $F(\alpha,\beta)$ is also a radial function, obtained using the Hankel transform of g.

Chapter 17

The Fast Fourier Transform

A fundamental problem in signal processing is to estimate finitely many values of the function $F(\omega)$ from finitely many values of its (inverse) Fourier transform, $f(t)$. As we have seen, the DFT arises in several ways in that estimation effort. The *fast Fourier transform* (FFT), discovered in 1965 by Cooley and Tukey, is an important and efficient algorithm for calculating the vector DFT [61]. John Tukey has been quoted as saying that his main contribution to this discovery was the firm and often voiced belief that such an algorithm must exist.

To illustrate the main idea behind the FFT consider the problem of evaluating a real polynomial $P(x)$ at a point, say $x = c$: let the polynomial be

$$P(x) = a_0 + a_1 x + a_2 x^2 + ... + a_{2K} x^{2K},$$

where a_{2K} might be zero. Performing the evaluation efficiently by Horner's method,

$$P(c) = (((a_{2K} c + a_{2K-1})c + a_{2K-2})c + a_{2K-3})c + ...,$$

requires $2K$ multiplications, so the complexity is on the order of the degree of the polynomial being evaluated. But suppose we also want $P(-c)$. We can write

$$P(x) = (a_0 + a_2 x^2 + ... + a_{2K} x^{2K}) + x(a_1 + a_3 x^2 + ... + a_{2K-1} x^{2K-2})$$

or

$$P(x) = Q(x^2) + xR(x^2).$$

Therefore we have $P(c) = Q(c^2) + cR(c^2)$ and $P(-c) = Q(c^2) - cR(c^2)$. If we evaluate $P(c)$ by evaluating $Q(c^2)$ and $R(c^2)$ separately, one more

multiplication gives us $P(-c)$ as well. The FFT is based on repeated use of this idea, which turns out to be more powerful when we are using complex exponentials, because of their periodicity.

Say the data are the samples are $\{f(n\Delta), n = 1, ..., N\}$, where $\Delta > 0$ is the sampling increment or sampling spacing.

The DFT estimate of $F(\omega)$ is the function $F_{DFT}(\omega)$, defined for ω in $[-\pi/\Delta, \pi/\Delta]$, and given by

$$F_{DFT}(\omega) = \Delta \sum_{n=1}^{N} f(n\Delta) e^{in\Delta\omega}.$$

The DFT estimate $F_{DFT}(\omega)$ is data consistent; its inverse Fourier transform value at $t = n\Delta$ is $f(n\Delta)$ for $n = 1, ..., N$. The DFT is sometimes used in a slightly more general context in which the coefficients are not necessarily viewed as samples of a function $f(t)$.

Given the complex N-dimensional column vector $\mathbf{f} = (f_0, f_1, ..., f_{N-1})^T$ define the DFT of vector $\mathbf{f}$ to be the function $DFT_{\mathbf{f}}(\omega)$, defined for ω in $[0, 2\pi)$, given by

$$DFT_{\mathbf{f}}(\omega) = \sum_{n=0}^{N-1} f_n e^{in\omega}.$$

Let $\mathbf{F}$ be the complex N-dimensional vector $\mathbf{F} = (F_0, F_1, ..., F_{N-1})^T$, where $F_k = DFT_{\mathbf{f}}(2\pi k/N), k = 0, 1, ..., N-1$. So the vector $\mathbf{F}$ consists of N values of the function $DFT_{\mathbf{f}}$, taken at N equispaced points $2\pi/N$ apart in $[0, 2\pi)$.

From the formula for $DFT_{\mathbf{f}}$ we have, for $k = 0, 1, ..., N-1$,

$$F_k = F(2\pi k/N) = \sum_{n=0}^{N-1} f_n e^{2\pi ink/N}. \tag{17.1}$$

To calculate a single F_k requires N multiplications; it would seem that to calculate all N of them would require N^2 multiplications. However, using the FFT algorithm we can calculate vector $\mathbf{F}$ in approximately $N \log_2(N)$ multiplications.

Suppose that $N = 2M$ is even. We can rewrite equation(17.1) as follows:

$$F_k = \sum_{m=0}^{M-1} f_{2m} e^{2\pi i(2m)k/N} + \sum_{m=0}^{M-1} f_{2m+1} e^{2\pi i(2m+1)k/N},$$

or, equivalently,

$$F_k = \sum_{m=0}^{M-1} f_{2m} e^{2\pi imk/M} + e^{2\pi ik/N} \sum_{m=0}^{M-1} f_{2m+1} e^{2\pi imk/M}. \tag{17.2}$$

Note that if $0 \leq k \leq M-1$ then

$$F_{k+M} = \sum_{m=0}^{M-1} f_{2m} e^{2\pi imk/M} - e^{2\pi ik/N} \sum_{m=0}^{M-1} f_{2m+1} e^{2\pi imk/M}, \qquad (17.3)$$

so there is no additional computational cost in calculating the second half of the entries of $\mathbf{F}$, once we have calculated the first half. The FFT is the algorithm that results when take full advantage of the savings obtainable by splitting a DFT calculating into two similar calculations of half the size.

We assume now that $N = 2^L$. Notice that if we use equations (17.2) and (17.3) to calculate vector $\mathbf{F}$, the problem reduces to the calculation of two similar DFT evaluations, both involving half as many entries, followed by one multiplication for each of the k between 0 and $M-1$. We can split these in half as well. The FFT algorithm involves repeated splitting of the calculations of DFTs at each step into two similar DFTs, but with half the number of entries, followed by as many multiplications as there are entries in either one of these smaller DFTs. We use recursion to calculate the cost $C(N)$ of computing $\mathbf{F}$ using this FFT method. From equation (17.2) we see that $C(N) = 2C(N/2) + (N/2)$. Applying the same reasoning to get $C(N/2) = 2C(N/4) + (N/4)$, we obtain

$$C(N) = 2C(N/2) + (N/2) = 4C(N/4) + 2(N/2) = ...$$

$$= 2^L C(N/2^L) + L(N/2) = N + L(N/2).$$

Therefore the cost required to calculate $\mathbf{F}$ is approximately $N \log_2 N$.

From our earlier discussion of discrete linear filters and convolution we see that the FFT can be used to calculate the periodic convolution (or even the non-periodic convolution) of finite length vectors.

Finally, let's return to the original context of estimating the Fourier transform $F(\omega)$ of function $f(t)$ from finitely many samples of $f(t)$. If we have N equispaced samples we can use them to form the vector $\mathbf{f}$ as above and perform the FFT algorithm to get vector $\mathbf{F}$ consisting of N values of the DFT estimate of $F(\omega)$. It may happen that we wish to calculate more than N values of the DFT estimate, perhaps to produce a smooth looking graph. We can still use the FFT, but we must trick it into thinking we have more data that the N samples we really have. We do this by *zero-padding*. Instead of creating the N-dimensional vector $\mathbf{f}$, we make a longer vector by appending, say, J zeros to the data, to make a vector that has dimension $N + J$. The DFT estimate is still the same function of ω, since we have only included new zero coefficients as fake data. But the FFT thinks we have $N + J$ data values, so it returns $N + J$ values of the DFT, at $N + J$ equispaced values of ω in $[0, 2\pi)$.

Chapter 18

Discretization

Computer simulations play a large role in the design and testing of reconstruction algorithms. In such simulations functions of a continuous variable are replaced by finite vectors obtained by sampling. In this chapter we look at the effects this discretization step can have on the calculation of Fourier transform values.

Throughout this chapter we let $F(\omega)$ be defined for $\omega \in [0, 2\pi]$, with

$$f(x) = \frac{1}{2\pi} \int_0^{2\pi} F(\omega) e^{-ix\omega} d\omega. \tag{18.1}$$

In subsequent chapters we shall be concerned with the problem of reconstructing the function $F(\omega)$, having bounded support, from finitely many values of $f(x)$. In applications $F(\omega)$ usually represents some physical object of limited extent; remote sensing has provided (usually noisy) values of $f(x)$ for finitely many x.

When algorithms are being developed and tested one often works with simulations. If the $F(\omega)$ to be simulated is specified analytically we may be able to compute values of $f(x)$ by performing the integrals in equation (18.1). It may be the case, however, that the integrals cannot be performed exactly or even that $F(\omega)$ is represented by a finite vector of samples. Estimating values of $f(x)$ in such cases is the topic for this chapter.

We assume that we have the values $F_n = F(2\pi n/N)$, $n = 0, 1, ..., N-1$ and wish to estimate $f(x)$ for certain values of x. We are particularly interested in the role to be played by the vector DFT of these samples, defined for $k = 0, 1, ..., N-1$ by

$$f_k = \frac{1}{N} \sum_{n=0}^{N-1} F_n e^{-2\pi ikn/N}. \tag{18.2}$$

It is tempting to take f_k as our estimate of $f(2\pi k/N)$ for each k, but this is not a good idea.

Let us assume that $F(\omega)$ is Riemann integrable. For each x we can approximate the integral in equation (18.1) by the Riemann sum

$$rs(x;N) = \frac{1}{N}\sum_{n=0}^{N-1} F_n e^{-2\pi i n x/N}. \tag{18.3}$$

The problem is that how good an approximation of $f(x)$ $rs(x;N)$ is will depend on x; as $|x|$ gets large the integrand becomes ever more oscillatory and a larger value of N will be needed to obtain a good approximation of the integral.

To see this from another viewpoint, consider the step function approximation of $F(\omega)$ given by

$$S(\omega) = \sum_{n=0}^{N-1} F_n \chi_{\pi/N}(\omega - \frac{2n+1}{N}\pi) \tag{18.4}$$

with

$$s(x) = \frac{1}{2\pi}\int_0^{2\pi} S(\omega) e^{-2\pi i x\omega} d\omega. \tag{18.5}$$

Performing the integrations we find that

$$s(x) = rs(x;N)\frac{\sin(\pi x/N)}{\pi x/N}. \tag{18.6}$$

If N is large enough for $S(\omega)$ to provide a reasonable approximation of $F(\omega)$ then $s(x)$ should be a good estimate of $f(x)$, at least for smaller values of x. Of course, since the rate of decay of $f(x)$ as $|x|$ approaches infinity depends on the smoothness of $F(\omega)$ we must not expect $s(x)$ to approximate $f(x)$ well for larger values of x.

Notice that the first positive zero of $\sin(\pi x/N)$ occurs at $x = N$, which suggests that $rs(x;N)$ provides a reasonable estimate of $f(x)$ for $|x|$ not larger than, say, $N/2$; therefore we may use f_k to estimate $f(k)$ for $0 \leq k \leq N/2$. To be safe, we may wish to use a smaller upper bound on k. Note also that $rs(-x;N) = rs(-x+N;N)$, which means that we may use f_{N-k} to approximate $f(-k)$ for $0 < k \leq N/2$.

To summarize, the N samples of $F(\omega)$ provide useful estimates of $f(k)$ for $-N/2 < k \leq N/2$. For $N = 2K$ we have $-K < k \leq K$, so that the N samples of $F(\omega)$ provide $2K = N$ useful estimates of $f(k)$.

There is yet another way to look at this problem. If $F(\omega)$ is twice continuously differentiable when periodically extended then its Fourier series expansion

$$F(\omega) = \sum_{m=-\infty}^{\infty} f(m) e^{im\omega} \tag{18.7}$$

converges uniformly for all ω. Therefore, for M large enough, we can estimate $F(\omega)$ using the truncated Fourier series

$$T(\omega; M) = \sum_{m=-M}^{M} f(m)e^{im\omega}. \tag{18.8}$$

Let $N = 2M + 1$ now.

Substituting $\omega = 2\pi n/N$ into equation (18.8) we obtain

$$T(2\pi n/N; M) = \sum_{m=-M}^{M} f(m)e^{2\pi imn/N}. \tag{18.9}$$

For $j = -M, ..., M$ multiply both sides of equation (18.9) by $e^{-2\pi inj/N}$, sum over $n = 0, ..., N - 1$ and use orthogonality to get $f(j)$ on the right side and

$$\frac{1}{N} \sum_{n=0}^{N-1} T(2\pi n/N; M)e^{-2\pi inj/N} \tag{18.10}$$

on the left. Viewing $T(2\pi n/N; M)$ as an estimate of $F(2\pi n/N)$ and replacing the former by the latter in equation (18.10), we conclude once again that $f(k)$ is well approximated by f_k for $0 \leq k \leq M$ and $f(-k)$ by f_{N-k} for $1 \leq k \leq M$.

Chapter 19

Fourier Transform Estimation

The basic problem we want to solve is the reconstruction of an object function $F(\omega)$ from finitely many values of its inverse Fourier transform

$$f(x) = \int F(\omega) \exp(-ix\omega) d\omega / 2\pi, \tag{19.1}$$

where, for notational convenience, we use single letters x and ω to denote possibly multi-dimensional variables. We assume that the formula

$$F(\omega) = \int f(x) \exp(ix\omega) dx$$

also holds.

Let the data be $f(x_m)$, $m = 1, ..., M$. Given this data, we want to estimate $F(\omega)$. Notice that any estimate of $F(\omega)$, which we denote as $\hat{F}(\omega)$, corresponds to an estimate of $f(x)$ by inserting $\hat{F}(\omega)$ into equation (19.1); that is

$$\hat{f}(x) = \int \hat{F}(\omega) \exp(-ix\omega) d\omega / 2\pi. \tag{19.2}$$

We shall say that the estimate $\hat{F}(\omega)$ is *data consistent* if

$$\hat{f}(x_m) = f(x_m),\ m = 1, ..., M.$$

A first estimate for $F(\omega)$: It seems reasonable to take as our first attempt the estimate

$$\hat{F}(\omega) = \sum_{m=1}^{M} f(x_m) \exp(ix_m\omega). \tag{19.3}$$

Is this estimate data consistent? Let's calculate $\hat{f}(x)$ and see. Inserting $\hat{F}(\omega)$ in equation (19.3) into equation (19.2) we get

$$\hat{f}(x) = \sum_{m=1}^{M} f(x_m)\delta(x - x_m),$$

where $\delta(x - a)$ denotes the Dirac delta function supported at the point a. The estimate is not data consistent, since what we measured at $x = x_m$ was not the top of a delta function, but just a number, $f(x_m)$. Does our estimate seem reasonable now? Is it reasonable that the estimate of the function $f(x)$ just happens to have delta function components located at precisely the places we chose to sample and is zero everywhere else? Perhaps we can do better.

We go beyond our first estimation attempt by incorporating some prior knowledge in our estimate, or, at least, making reasonable assumptions about the function $F(\omega)$ being estimated. The first type of assumption we make concerns the support of $F(\omega)$, that is, the region in ω-space outside of which $F(\omega)$ is identically equal to zero.

Including a support constraint: Let $\Omega > 0$ and suppose that the function $F(\omega) = 0$ for $|\omega| > \Omega$. Let $\chi_\Omega(\omega)$ be the function that is one for $|\omega| \leq \Omega$ and zero otherwise. Building on our first attempt, we try the estimate

$$\hat{F}(\omega) = \chi_\Omega(\omega) \sum_{m=1}^{M} f(x_m) \exp(ix_m\omega). \tag{19.4}$$

Is this estimate data consistent? Inserting $\hat{F}(\omega)$ in equation (19.4) into equation (19.2) we get

$$\hat{f}(x) = \sum_{m=1}^{M} f(x_m) \frac{\sin \Omega(x - x_m)}{\pi(x - x_m)}. \tag{19.5}$$

Now we ask if it is true that

$$f(x_n) = \sum_{m=1}^{M} f(x_m) \frac{\sin \Omega(x_n - x_m)}{\pi(x_n - x_m)} \tag{19.6}$$

for $n = 1, ..., M$. The answer is, generally, no, although in special cases, the answer is yes, or almost yes.

The Nyquist case: Suppose that $\Omega = \pi$, $F(\omega)$ is zero for $|\omega| > \pi$ and the data is $f(m)$, $m = 1, ..., M$. Then the estimate

$$\hat{F}(\omega) = \chi_\pi(\omega) \sum_{m=1}^{M} f(m) \exp(im\omega)$$

is data consistent; it is then what is often called the *discrete Fourier transform* (DFT) of the data, defined for ω in the interval $[-\pi, \pi]$. For this reason we write the estimate as $F_{DFT}(\omega)$. The inversion formula gives

$$\hat{f}(x) = \sum_{m=1}^{M} f(m) \frac{\sin \pi(x-m)}{\pi(x-m)}$$

and

$$\hat{f}(n) = \sum_{m=1}^{M} f(m) \frac{\sin \pi(n-m)}{\pi(n-m)}$$

holds for each $n = 1, ..., M$, since the matrix becomes the identity matrix.

Suppose, more generally, that $\Omega = \frac{\pi}{\Delta}$ for some $\Delta > 0$, $F(\omega)$ is zero for $|\omega| > \frac{\pi}{\Delta}$ and the data is $f(m\Delta)$, $m = 1, ..., M$. Then the estimate

$$\hat{F}(\omega) = \chi_{\frac{\pi}{\Delta}}(\omega) \sum_{m=1}^{M} f(m\Delta) \exp(im\Delta\omega)$$

is almost data consistent. The inversion formula gives

$$\hat{f}(x) = \sum_{m=1}^{M} f(m\Delta) \frac{\sin \frac{\pi}{\Delta}(x-m\Delta)}{\pi(x-m\Delta)}$$

and so

$$\hat{f}(n\Delta) = \frac{1}{\Delta} \sum_{m=1}^{M} f(m\Delta) \frac{\sin \pi(n-m)}{\pi(n-m)} = \frac{1}{\Delta} f(n\Delta)$$

holds for each $n = 1, ..., M$. To get data consistency we multiply our estimate by Δ; that is, we take

$$\hat{F}(\omega) = \Delta \chi_{\frac{\pi}{\Delta}}(\omega) \sum_{m=1}^{M} f(m\Delta) \exp(im\Delta\omega).$$

Now this estimate is both data consistent and supported on the interval $[-\frac{\pi}{\Delta}, \frac{\pi}{\Delta}]$. This estimate may also be called the DFT, ignoring the Δ multiplier or redefining variables to make $\Delta = 1$.

Exercise 1: Use the orthogonality principle to show that the DFT minimizes the distance

$$\int_{-\pi}^{\pi} |F(\omega) - \sum\nolimits_{m=1}^{M} a_m e^{im\omega}|^2 d\omega.$$

When the data is $f(m\Delta)$, so is equispaced, we assume that $F(\omega) = 0$ for $|\omega| > \frac{\pi}{\Delta}$; that is, we assume that our sample spacing Δ is small enough to

avoid aliasing. What happens when we *oversample*; that is, when $F(\omega) = 0$ for $|\omega| > \Omega$, where $\Omega < \frac{\pi}{\Delta}$?

The general case: Even for integer spaced data $f(m)$, $m = 1, ..., M$, the estimate

$$\hat{F}(\omega) = \chi_\Omega(\omega) \sum_{m=1}^{M} f(m) \exp(im\omega)$$

will not be data consistent if $\Omega < \pi$. For more generally spaced data $f(x_m)$, $m = 1, ..., M$ the estimate

$$\hat{F}(\omega) = \chi_\Omega(\omega) \sum_{m=1}^{M} f(x_m) \exp(ix_m\omega)$$

will not be data consistent. The approach we take is to retain the algebraic form of these estimators, but to allow the coefficients to be determined by data consistency.

Take as the estimate of $F(\omega)$ the function

$$F_\Omega(\omega) = \chi_\Omega(\omega) \sum_{m=1}^{M} a_m \exp(ix_m\omega), \tag{19.7}$$

with the coefficients a_m chosen to give data consistency. This means we must select the a_m to satisfy the equations

$$f(x_n) = \sum_{m=1}^{M} a_m \frac{\sin \Omega(x_n - x_m)}{\pi(x_n - x_m)}$$

for $n = 1, ..., M$. The resulting estimate $F_\Omega(\omega)$ is both data consistent and supported on the interval $[-\Omega, \Omega]$. We shall refer to this estimator as the *non-iterative bandlimited extrapolation method*. Figure 19.1 below shows the advantage of the non-iterative bandlimited extrapolation method, in the top frame, over the DFT below. The true object to be reconstructed is the solid figure. The sampling spacing is $\Delta = 1$, but $\Omega = \pi/30$, so the 129 data points are thirty times oversampled.

A paradox: It follows from what we just did that for any finite data and any $\alpha < \beta$ there is a function $\hat{F}(\omega)$ supported on the interval $[\alpha, \beta]$ and consistent with the data. Does the data contain no information about the actual support of $F(\omega)$? This would seem to say that the data we have measured contains essentially no information, since we can generate thousands of additional data points, select any α and β and still find a data consistent estimate of $F(\omega)$. How can this be true when, at the same time,

we have plenty of simulation cases in which we are able to generate fairly accurate estimates of the correct answer using these techniques?

The answer is that while the data we have does not eliminate any possible support for the function $F(\omega)$ it is capable of indicating preferences. When we use equation (19.7) we do get an estimate that is data consistent, but if the support $[-\Omega, \Omega]$ is a poor choice we usually have an indication of that in the norm of the estimate. The norm of $F_\Omega(\omega)$ is

$$||F_\Omega|| = \sqrt{\int_{-\Omega}^{\Omega} |F_\Omega(\omega)|^2 d\omega}$$

and can be quite large if the data and the Ω are poorly matched. Usually, the true $F(\omega)$ is a physically meaningful function that does not have unusually large norm, so any estimate $F_\Omega(x)$ with a large norm is probably incorrect and a better Ω should be sought.

Properties of the estimate $F_\Omega(\omega)$: In addition to being data consistent and having for its support the interval $[-\Omega, \Omega]$ the estimate $F_\Omega(\omega)$ given by equation (19.7) has two additional properties that are worth mentioning. The choice $G(\omega) = F_\Omega(\omega)$ minimizes the integral

$$\int_{-\Omega}^{\Omega} |G(\omega)|^2 dx$$

over all estimates $G(\omega)$ that are data consistent. It also minimizes the approximation error

$$\int_{-\Omega}^{\Omega} |F(\omega) - \sum_{m=1}^{M} a_m \exp(ix_m\omega)|^2 d\omega \tag{19.8}$$

over all choices of coefficients a_m. So in this sense it is the best approximation of the truth that we can find that has its particular algebraic form, provided, of course, that $F(\omega)$ is supported on $[-\Omega, \Omega]$.

Exercise 2: Suppose that $0 < \Omega$ and $F(\omega) = 0$ for $|\omega| > \Omega$. Let $f(x)$ be the inverse Fourier transform of $F(\omega)$ and suppose that the data is $f(x_m)$, $m = 1, ..., M$. Use the orthogonality principle to find the coefficients a_m that minimize the error given by equation (19.8). Show that the resulting estimate of $F(\omega)$ is consistent with the data.

The choice of Ω is left up to us. Suppose that our choice is too big. Then the estimate in equation (19.7) gives the best estimate of its algebraic form over the interval $[-\Omega, \Omega]$, but since $F(\omega)$ is zero on a portion of this interval, the estimate spends some effort estimating the value zero. If we

can get a more accurate estimate of the true support of $F(\omega)$ then we can modify the Ω and get a better estimate of $F(\omega)$.

Once we have calculated the estimate $F_\Omega(\omega)$ we obtain a procedure for extrapolating the data by computing its inverse Fourier transform:

$$f_\Omega(x) = \sum_{m=1}^{M} a_m \frac{\sin \Omega(x - x_m)}{\pi(x - x_m)}$$

estimates the values $f(x)$ we did not measure. This procedure is called *bandlimited extrapolation*. The Gerchberg-Papoulis (GP) method for bandlimited extrapolation is an iterative algorithm for calculating $F_\Omega(\omega)$ and, equivalently, $f_\Omega(x)$. For large data sets it provides an alternative to solving a large system of linear equations. Discrete versions of the GP algorithm work with finite vectors and exploit the FFT to perform the estimation quickly. We consider the GP method in a later chapter.

The PDFT: The estimate $F_\Omega(\omega)$ is the product of two terms: the first is $\chi_\Omega(\omega)$, which incorporates prior knowledge about the function $F(\omega)$, and the second is the sum, whose coefficients are calculated to insure data consistency. We obtain a more flexible class of estimators by replacing the first term, $\chi_\Omega(\omega)$, with $P(\omega) \geq 0$, a prior estimate of the magnitude of $F(\omega)$. The resulting estimate, called the PDFT, is the subject of the next chapter.

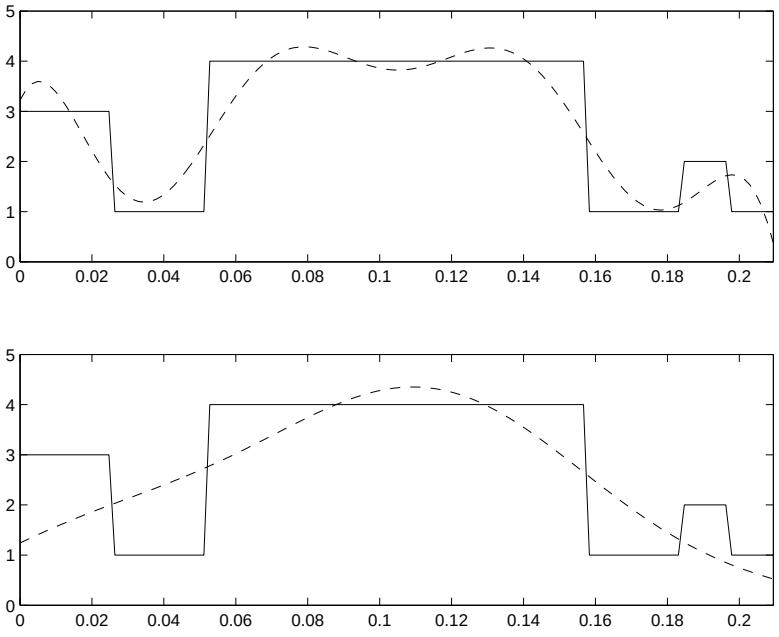


Figure 19.1: The non-iterative bandlimited extrapolation method (top) and the DFT (below) for $M = 129$, $\Delta = 1$ and $\Omega = \pi/30$.

Chapter 20

The PDFT

Most of the time the data we have is noisy, the data we have isn't really the data we want, the locations where we measured the data were the ones available, not the ones we wanted to use, the physical model we are using to interpret the data is not quite right, but is the best we can do, and we don't have enough data. All these difficulties are important and we shall deal with each one of them in one way or another. Beginning with the discussion of bandlimited extrapolation and continuing through this chapter, we focus on the last problem, the limited data problem.

In many estimation and reconstruction problems we have a limited amount of data that is not sufficient, by itself, to provide a useful result; additional information is needed. In the bandlimited extrapolation problem just discussed we were able to use the information about the support of the Fourier transform function $F(\omega)$ to improve our estimate. We may, at times, have some prior estimate not only of the support, but of its overall shape; such prior profile information can be useful in estimating $F(\omega)$. The PDFT [35], [36] is a generalization of the non-iterative bandlimited extrapolation method in equation (19.7), designed to permit the use of such prior profile estimates.

Suppose now that the data is $f(x_m)$, $m = 1, ..., M$. Suppose also that we have some prior estimate of the magnitude of $F(\omega)$ for each real ω, in the form of a function $P(\omega) \geq 0$. In the previous chapter $P(\omega)$ appeared as $\chi_\pi(\omega)$ and $\chi_\Omega(\omega)$. We take as our estimate of F the function of the form

$$F_{PDFT}(\omega) = P(\omega) \sum_{m=1}^{M} c_m \exp(ix_m\omega), \tag{20.1}$$

where the c_m are chosen to give data consistency.

Exercise 1: Show that the c_m must satisfy the equations

$$f(x_n) = \sum_{m=1}^{M} c_m p(x_n - x_m),\ n = 1, ..., M, \tag{20.2}$$

where $p(x)$ is the inverse Fourier transform of $P(\omega)$. Note that for $P(\omega) = \chi_\Omega(\omega)$ we have $p(x) = \frac{\sin(\Omega x)}{\pi x}$.

Both of the estimates $F_{DFT}(\omega)$ and $F_\Omega(\omega)$ provide a best approximation of its form and support for $F(\omega)$. The same is true of the PDFT.

Exercise 2: Show that the estimate $F_{PDFT}(\omega)$ minimizes the distance

$$\int |F(\omega) - P(\omega) \sum_{m=1}^{M} a_m \exp(ix_m\omega)|^2 P(\omega)^{-1} d\omega$$

over all choices of the coefficients a_m.

Both of the estimates $F_{DFT}(\omega)$ and $F_\Omega(\omega)$ minimize an energy, subject to data consistency. Something similar happens with the PDFT; the PDFT minimizes the weighted energy

$$\int_{-\pi}^{\pi} |F_{PDFT}(\omega)|^2 P(\omega)^{-1} d\omega, \tag{20.3}$$

subject to data consistency, with the understanding that $P(\omega)^{-1} = 0$ if $P(\omega) = 0$. That the PDFT is a minimum weighted energy solution will be important later when we turn to the discrete PDFT.

For relatively small M the PDFT is easily calculated. The difficult part is constructing the matrix P having the entries $P_{m,n} = p(x_m - x_n)$, which requires the calculation of the inverse Fourier transform of $P(\omega)$ at the irregularly spaced points $x_m - x_n$. In addition, the matrix P is often ill-conditioned, meaning that some of its (necessarily positive) eigenvalues are near zero. Noise in the data $f(x_m)$ can lead to unreasonably large values of c_m and to a PDFT estimate that is useless. To combat this problem we can multiply the terms $P_{n,n}$ on the main diagonal of P by (say) 1.001. This prevents the eigenvalues from becoming too small.

For large data sets it is more difficult to work with the PDFT as formulated. The matrix P is very large, its entries difficult to compute, storage becomes a problem and solving the resulting system of equations is expensive. To avoid all these problems and to have a formulation of the PDFT that is conceptually easier to use we turn to a discrete formulation, which we call the DPDFT.

In a recent article [123] Poggio and Smale discuss the use of positive-definite kernels for interpolation, in the context of artificial intelligence and supervised learning.

Chapter 21

Bandlimited Extrapolation

The continuous formulation of *the bandlimited extrapolation problem* is the following: let $f(x)$ and $F(\omega)$ be a Fourier transform pair. We assume that $F(\omega) = 0$, for $|\omega| > \Omega$, where Ω is a positive quantity. The function $f(x)$ is then said to be Ω-*bandlimited*. If we know $f(x)$ for x in some bounded interval of the real line, then this data determines $F(\omega)$ uniquely, by analyticity; the extension of $f(x)$ to complex z, given by the *Fourier-Laplace* transform

$$f(z) = \int_{-\infty}^{\infty} F(\omega) e^{-izw} d\omega/2\pi, \tag{21.1}$$

can be differentiated under the integral sign, since the limits of integration are finite. In fact, the function $f(z)$ is a complex-valued function that is analytic through the complex plane. Therefore, the known values of $f(x)$ determine $f(z)$ for all other values of z; we can, in theory, extrapolate f outside the data window. The iterative and non-iterative methods we describe below are usually called *super-resolution techniques* in the signal processing literature. Similar methods applied in sonar and radar array processing are called *super-directive* methods [62].

Exercise 1: Show that there can be no Fourier transform pair f, F for which positive constants a and b exist such that $f(x) = 0$ for $|x| > a$ and $F(\omega) = 0$ for $|\omega| > b$. Thus it is not possible for both f and F to be band-limited.
Hint: Use the analyticity of the function $f(z)$.

In practice, we have only finitely many values of $f(x)$ and these are typically noisy. We shall not address the noise problem here, except to say

that it is usually handled by including regularization in the solving of each of the systems of linear equations we encounter in what follows.

The finitely many values of f, say $f(x_1), ..., f(x_N)$, may be obtained at irregularly spaced sample points $\{x_n\}$ but often correspond to uniformly spaced sampling points $\{x_n = a + n\Delta\}$. We rstrict ourselves to uniformly spaced data when discussing the iterative Gerchberg-Papoulis (GP) method. The non-iterative estimate $F_\Omega(\omega)$ permits non-uniformly spaced data.

The case of uniformly spaced data:

We assume that the function $F(\omega)$ is supported on the interval $[-\Omega, \Omega]$, for some $\Omega < \pi$. The sequence of Fourier coefficients of F is denoted f. Our data are the Fourier coefficients $f(n)$, for $n \in \{M, M+1, ..., N\}$, forming the vector $\mathbf{d}$. The function $\chi_\Omega(\omega)$ is one for $|\omega| \leq \Omega$ and zero otherwise. For notational convenience we denote by $(\Omega G)(\omega)$ the product of the functions $\chi_\Omega(\omega)$ and $G(\omega)$.

The Gerchberg-Papoulis algorithm [80], [118] is an iterative procedure that works as follows. Begin with the DFT estimate, $F^0(\omega) = F_{DFT}(\omega)$ defined for ω in the interval $[-\pi, \pi]$, which is data consistent but not supported on $[-\Omega, \Omega]$. Multiply $F^0(\omega)$ by the function $\chi_\Omega(\omega)$; the result is now supported on $[-\Omega, \Omega]$, but is no longer data consistent. Take the doubly infinite sequence of its Fourier coeffcients and replace those for $n = M, ..., N$ with the known data. Now use this new sequence as the Fourier coefficients of a function $F^1(\omega)$, which is then data consistent, but not supported on $[-\Omega, \Omega]$. Multiply $F^1(\omega)$ by $\chi_\Omega(\omega)$, take its Fourier coefficients, etc. In the limit we obtain a function supported on $[-\Omega, \Omega]$ that is consistent with the data. Let us consider the GP algorithm in more detail.

For any sequence of Fourier coefficients $g = \{g(n)\}$ let Dg denote the sequence whose terms are $g(n)$ for $n \in \{M, M+1, ..., N\}$ and zero otherwise. Let $\mathcal{F}g = G$ be the operator taking a sequence of Fourier coefficients g into the function

$$G(\omega) = \sum\nolimits_{n=-\infty}^{+\infty} g(n) \exp(in\omega),$$

for $\omega \in (-\pi, \pi)$.

Let $\mathcal{H} = L^2(-\pi, \pi)$, $C_1 = L^2(-\Omega, \Omega)$ and C_2 the set of all members $G(\omega)$ of $\mathcal{H}$ whose Fourier coefficients satisfy $g(n) = f(n)$ for $n = M, M+1, ..., N$. The *metric projection* of a function $G(\omega) \in \mathcal{H}$ onto C_1 is $(\chi_\Omega G)(\omega)$; this is the function in C_1 closest to $G(\omega)$. The metric projection onto C_2 is implemented by passing from $G(\omega)$ to the sequence of its Fourier coefficients $\mathcal{F}^{-1}G = g$, then replacing those coefficients for $n = M, M+1, ..., N$ with $f(n)$ and calculating the resulting Fourier series; that is, the metric projection of G onto C_2 is $\mathcal{F}(Df + (I - D)\mathcal{F}^{-1}G)$.

We begin the Gerchberg-Papoulis (GP) iteration with the function $F^0(\omega) = 0$ for all $\omega \in (-\pi, \pi)$. For $k = 0, 1, ...$ having calculated F^k with f^k its sequence of Fourier coefficients, we define F^{k+1} by

$$F^{k+1} = \Omega\mathcal{F}(Df + (I - D)\mathcal{F}^{-1}F^k).$$

It would appear that, in order to implement this algorithm, we must calculate the entries of the sequence $\{(I - D)\mathcal{F}^{-1}F^k\}$ for all integers n not in the set $\{M, M + 1, ..., N\}$; this is not the case, fortunately. Note that

$$F^{k+1} - F^k = \Omega\mathcal{F}D(f - f^k) = \Omega\mathcal{F}a^k,$$

where the entries of the sequence $D(f - f^k) = a^k$ are zero, except for $n = M, ..., N$. Since $F^0 = 0$ it follows that each F^k has the form $F^k = \Omega\mathcal{F}b^k$, for some sequence b^k with $b^k(n) = 0$ for n not in the set $\{M, M+1, ..., N\}$. From this we conclude that the limit F^∞ has the form

$$F^\infty(\omega) = \Omega \sum\nolimits_{n=M}^{N} c_n \exp(in\omega)$$

for appropriate c_n. The coefficients c_n can then be determined by equating the Fourier coefficients of both sides of this equation. To do this we must solve the finite system of linear equations

$$f(m) = \sum_{m=M}^{N} c_n \frac{\sin \Omega(m - n)}{\pi(m - n)}, \tag{21.2}$$

where $m = M, ..., N$. This, of course, can also be done iteratively, if we desire. This leads us to the non-iterative bandlimited extrapolation estimate $F_\Omega(\omega)$ given by equation (19.7).

A different approach is frequently used, resulting in a slightly different extrapolation. This second approach formulates the problem entirely in terms of finite vectors and interprets the Fourier transform as a linear transformation between finite vectors, as is done with the Fast Fourier Transform (FFT) algorithm.

From the discussion above we see that for an arbitrary data vector d and an arbitrary choice of the band $[-\Omega, \Omega]$ in $[-\pi, \pi]$ there is a function $F_\Omega(\omega)$ supported on $[-\Omega, \Omega]$ that is consistent with the data in the vector d. The function F_Ω has the form

$$F_\Omega(\omega) = \chi_\Omega(\omega) \sum\nolimits_{n=M}^{N} c_n \exp(in\omega), \tag{21.3}$$

where $\chi_\Omega(\omega) = 1$ if $|\omega| \leq \Omega$ and zero otherwise. The coefficients c_n solve the equations (21.2). To perform data extrapolation one now evaluates the Fourier transform of F_Ω at the desired points. Note that this method applies equally to uniformly and nonuniformly spaced data and is easily extended to higher dimensions. This noniterative implementation of the GP extrapolation is not new; it was presented in [34], and has been rediscovered several times since then (see p. 209 of [136]).

Chapter 22

More on Bandlimited Extrapolation

Let our data be $f(x_m)$, $m = 1, ..., M$, where the x_m are arbitrary values of the variable x. If $F(\omega)$ is zero outside $[-\Omega, \Omega]$, then minimizing the energy over $[-\Omega, \Omega]$ subject to data consistency produces an estimate of the form

$$F_\Omega(\omega) = \chi_\Omega(\omega) \sum\nolimits_{m=1}^{M} b_m \exp(ix_m\omega),$$

with the b_m satisfying the equations

$$f(x_n) = \sum\nolimits_{m=1}^{M} b_m \frac{\sin(\Omega(x_m - x_n))}{\pi(x_m - x_n)},$$

for $n = 1, ..., M$. The matrix S_Ω with entries $\frac{\sin(\Omega(x_m - x_n))}{\pi(x_m - x_n)}$ we call a *sinc* matrix.

Although it seems reasonable that incorporating the additional information about the support of $F(\omega)$ should improve the estimation, it would be more convincing if we had a more mathematical argument to make. For that we turn to an analysis of the eigenvectors of the sinc matrix.

Exercise 1: The purpose of this exercise is to show that, for an Hermitian nonnegative-definite M by M matrix Q, a norm-one eigenvector $\mathbf{u}^1$ of Q associated with its largest eigenvalue, λ_1, maximizes the quadratic form $\mathbf{a}^\dagger Q\mathbf{a}$ over all vectors $\mathbf{a}$ with norm one. Let $Q = ULU^\dagger$ be the eigenvector decomposition of Q, where the columns of U are mutually orthogonal eigenvectors $\mathbf{u}^n$ with norms equal to one, so that $U^\dagger U = I$, and $L = diag\{\lambda_1, ..., \lambda_M\}$ is the diagonal matrix with the eigenvalues of Q as its entries along the main

diagonal. Assume that $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_M$. Then maximize

$$\mathbf{a}^\dagger Q\mathbf{a} = \sum_{n=1}^{M} \lambda_n \, |\mathbf{a}^\dagger \mathbf{u}^n|^2,$$

subject to the constraint

$$\mathbf{a}^\dagger \mathbf{a} = \mathbf{a}^\dagger U^\dagger U\mathbf{a} = \sum_{n=1}^{M} |\mathbf{a}^\dagger \mathbf{u}^n|^2 = 1.$$

Hint: Show $\mathbf{a}^\dagger Q\mathbf{a}$ is a convex combination of the eigenvalues of Q.

Exercise 2: Show that for the sinc matrix $Q = S_\Omega$ the quadratic form $\mathbf{a}^\dagger Q\mathbf{a}$ in the previous exercise becomes

$$\mathbf{a}^\dagger S_\Omega \mathbf{a} = \frac{1}{2\pi} \int_{-\Omega}^{\Omega} |\sum\nolimits_{n=1}^{M} a_n e^{in\omega}|^2 d\omega.$$

Show that the norm of the vector $\mathbf{a}$ is the integral

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} |\sum\nolimits_{n=1}^{M} a_n e^{in\omega}|^2 d\omega.$$

Exercise 3: For $M = 30$ compute the eigenvalues of the matrix S_Ω for various choices of Ω, such as $\Omega = \frac{\pi}{k}$, for $k = 2, 3, ..., 10$. For each k arrange the set of eigenvalues in decreasing order and note the proportion of them that are not near zero. The set of eigenvalues of a matrix is sometimes called its *eigenspectrum* and the nonnegative function $\chi_\Omega(\omega)$ is a power spectrum; here is one time in which different notions of a *spectrum* are related.

Suppose that the vector $\mathbf{u}^1 = (u_1^1, ..., u_M^1)^T$ is an eigenvector of S_Ω corresponding to the largest eigenvalue, λ_1. Associate with $\mathbf{u}^1$ the function

$$U^1(\omega) = \sum\nolimits_{n=1}^{M} u_n^1 e^{in\omega}.$$

Then

$$\lambda_1 = \int_{-\Omega}^{\Omega} |U^1(\omega)|^2 d\omega / \int_{-\pi}^{\pi} |U^1(\omega)|^2 d\omega$$

and $U^1(\omega)$ is the function of its form that is most concentrated within the interval $[-\Omega, \Omega]$.

Similarly, if $\mathbf{u}^M$ is an eigenvector of S_Ω associated with the smallest eigenvalue λ_M, then the corrsponding function $U^M(\omega)$ is the function of its form least concentrated in the interval $[-\Omega, \Omega]$.

Exercise 4: Plot for $|\omega| \leq \pi$ the functions $|U^m(\omega)|$ corresponding to each of the eigenvectors of the sinc matrix S_Ω. Pay particular attention to the places where each of these functions is zero.

The eigenvectors of S_Ω corresponding to different eigenvalues are orthogonal, that is $(\mathbf{u}^m)^\dagger \mathbf{u}^n = 0$ if m is not n. We can write this in terms of integrals:

$$\int_{-\pi}^{\pi} U^n(\omega)\overline{U^m(\omega)}d\omega = 0$$

if m is not n. The mutual orthogonality of these functions is related to the locations of their roots, which were studied in the previous exercise.

Any Hermitian matrix Q is invertible if and only if none of its eigenvalues is zero. With λ_m and $\mathbf{u}^m$, $m = 1, ..., M$ the eigenvalues and eigenvectors of Q the inverse of Q can then be written as

$$Q^{-1} = (1/\lambda_1)\mathbf{u}^1(\mathbf{u}^1)^\dagger + ... + (1/\lambda_M)\mathbf{u}^M(\mathbf{u}^M)^\dagger.$$

Exercise 5: Show that the non-iterative bandlimited extrapolation estimate (19.7) $F_\Omega(\omega)$ can be written as

$$F_\Omega(\omega) = \chi_\Omega(\omega) \sum\nolimits_{m=1}^{M} \frac{1}{\lambda_m}(\mathbf{u}^m)^\dagger \mathbf{d}\, U^m(\omega),$$

where $\mathbf{d}$ is the data vector.

Exercise 6: Show that the DFT estimate of $F(\omega)$, restricted to the interval $[-\Omega, \Omega]$, is

$$F_{DFT}(\omega) = \chi_\Omega(\omega) \sum\nolimits_{m=1}^{M} (\mathbf{u}^m)^\dagger \mathbf{d}\, U^m(\omega).$$

From these two exercises we can learn why it is that the estimate $F_\Omega(\omega)$ resolves better than the DFT. The former makes more use of the functions $U^m(\omega)$ for higher values of m, since these are the ones for which λ_m is closer to zero. Since those functions are the ones having most of their roots within the interval $[-\Omega, \Omega]$, they have the most flexibility within that region and are better able to describe those features in $F(\omega)$ that are not resolved by the DFT.

Chapter 23

A Little Matrix Theory

The 2 by 2 matrix $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ has an inverse

$$A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

whenever the *determinant* of A, $\det(A) = ad - bc \neq 0$. More generally, associated with every complex square matrix is the complex number called its determinant, which is obtained from the entries of the matrix using formulas that can be found in any text on linear algebra. The significance of the determinant is that the matrix is invertible if and only if its determinant is not zero. This is of more theoretical than practical importance, since no computer can tell when a number is precisely zero.

Given N by N complex matrix A, we say that a complex number λ is an *eigenvalue* of A if there is a nonzero vector $\mathbf{u}$ with $A\mathbf{u} = \lambda\mathbf{u}$. The column vector $\mathbf{u}$ is then called an *eigenvector* of A associated with eigenvalue λ; clearly, if $\mathbf{u}$ is an eigenvector of A, then so is $c\mathbf{u}$, for any constant $c \neq 0$. If λ is an eigenvalue of A then the matrix $A - \lambda I$ fails to have an inverse, since $(A - \lambda I)\mathbf{u} = \mathbf{0}$ but $\mathbf{u} \neq \mathbf{0}$. If we treat λ as a variable and compute the determinant of $A - \lambda I$ we obtain a polynomial of degree N in λ. Its roots $\lambda_1, ..., \lambda_N$ are then the eigenvalues of A. If $||\mathbf{u}||^2 = \mathbf{u}^\dagger\mathbf{u} = 1$ then $\mathbf{u}^\dagger A\mathbf{u} = \lambda\mathbf{u}^\dagger\mathbf{u} = \lambda$.

Suppose that $A\mathbf{x} = \mathbf{b}$ is a consistent linear system of M equations in N unknowns, where $M < N$. Then there are infinitely many solutions. A standard procedure in such cases is to find that solution $\mathbf{x}$ having the smallest norm

$$||\mathbf{x}|| = \sqrt{\sum_{n=1}^{N} |x_n|^2}.$$

As we shall see shortly, the *minimum norm* solution of $A\mathbf{x} = \mathbf{b}$ is a vector of the form $\mathbf{x} = A^\dagger\mathbf{z}$, where $A^\dagger$ denotes the conjugate transpose of the matrix

A. Then $A\mathbf{x} = \mathbf{b}$ becomes $AA^\dagger\mathbf{z} = \mathbf{b}$. Typically $(AA^\dagger)^{-1}$ will exist and we get $\mathbf{z} = (AA^\dagger)^{-1}\mathbf{b}$, from which it follows that the minimum norm solution is $\mathbf{x} = A^\dagger(AA^\dagger)^{-1}\mathbf{b}$. When M and N are not too large forming the matrix $AA^\dagger$ and solving for $\mathbf{z}$ is not prohibitively expensive and time-consuming. However, in image processing the vector $\mathbf{x}$ is often a vectorization of a two-dimensional (or even three-dimensional) image and M and N can be on the order of tens of thousands or more. The ART algorithm gives us a fast method for finding the minimum norm solution without computing $AA^\dagger$.

We begin by proving that the minimum norm solution of $A\mathbf{x} = \mathbf{b}$ has the form $\mathbf{x} = A^\dagger\mathbf{z}$ for some M-dimensional complex vector $\mathbf{z}$.

Let the *null space* of the matrix A be all N-dimensional complex vectors $\mathbf{w}$ with $A\mathbf{w} = \mathbf{0}$. If $A\mathbf{x} = \mathbf{b}$ then $A(\mathbf{x} + \mathbf{w}) = \mathbf{b}$ for all $\mathbf{w}$ in the null space of A. If $\mathbf{x} = A^\dagger\mathbf{z}$ and $\mathbf{w}$ is in the null space of A then

$$\begin{aligned} ||\mathbf{x} + \mathbf{w}||^2 = ||A^\dagger\mathbf{z} + \mathbf{w}||^2 = (A^\dagger\mathbf{z} + \mathbf{w})^\dagger(A^\dagger\mathbf{z} + \mathbf{w}) \\ = (A^\dagger\mathbf{z})^\dagger(A^\dagger\mathbf{z}) + (A^\dagger\mathbf{z})^\dagger\mathbf{w} + \mathbf{w}^\dagger(A^\dagger\mathbf{z}) + \mathbf{w}^\dagger\mathbf{w} \\ = ||A^\dagger\mathbf{z}||^2 + (A^\dagger\mathbf{z})^\dagger\mathbf{w} + \mathbf{w}^\dagger(A^\dagger\mathbf{z}) + ||\mathbf{w}||^2 \\ = ||A^\dagger\mathbf{z}||^2 + ||\mathbf{w}||^2, \end{aligned}$$

since

$$\mathbf{w}^\dagger(A^\dagger\mathbf{z}) = (A\mathbf{w})^\dagger\mathbf{z} = \mathbf{0}^\dagger\mathbf{z} = 0$$

and

$$(A^\dagger\mathbf{z})^\dagger\mathbf{w} = \mathbf{z}^\dagger A\mathbf{w} = \mathbf{z}^\dagger\mathbf{0} = 0.$$

Therefore $||\mathbf{x} + \mathbf{w}|| = ||A^\dagger\mathbf{z} + \mathbf{w}|| > ||A^\dagger\mathbf{z}|| = ||\mathbf{x}||$ unless $\mathbf{w} = \mathbf{0}$. This completes the proof.

Exercise 1: Show that if $\mathbf{z} = (z_1, ..., z_N)^T$ is a column vector with complex entries and $H = H^\dagger$ is an N by N Hermitian matrix with complex entries then the quadratic form $\mathbf{z}^\dagger H\mathbf{z}$ is a real number. Show that the quadratic form $\mathbf{z}^\dagger H\mathbf{z}$ can be calculated using only real numbers. Let $\mathbf{z} = \mathbf{x} + i\mathbf{y}$, with $\mathbf{x}$ and $\mathbf{y}$ real vectors and let $H = A + iB$, where A and B are real matrices. Then show that $A^T = A$, $B^T = -B$, $\mathbf{x}^T B\mathbf{x} = 0$ and finally,

$$\mathbf{z}^\dagger H\mathbf{z} = [\,\mathbf{x}^T \quad \mathbf{y}^T\,] \begin{bmatrix} A & -B \\ B & A \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}.$$

Use the fact that $\mathbf{z}^\dagger H\mathbf{z}$ is real for every vector $\mathbf{z}$ to conclude that the eigenvalues of H are real.

It can be shown that it is possible to find a set of N mutually orthogonal eigenvectors of the Hermitian matrix H; call them $\{\mathbf{u}^1, ..., \mathbf{u}^N\}$. The matrix H can then be written as

$$H = \sum\nolimits_{n=1}^{N} \lambda_n \mathbf{u}^n (\mathbf{u}^n)^\dagger,$$

a linear superposition of the *dyad* matrices $\mathbf{u}^n(\mathbf{u}^n)^\dagger$. We can also write $H = ULU^\dagger$, where U is the matrix whose n-th column is the column vector $\mathbf{u}^n$ and L is the diagonal matrix with the eigenvalues down the main diagonal and zero elsewhere.

The matrix H is invertible if and only if none of the λ are zero and its inverse is

$$H^{-1} = \sum\nolimits_{n=1}^{N} \lambda_n^{-1}\mathbf{u}^n(\mathbf{u}^n)^\dagger.$$

We also have $H^{-1} = UL^{-1}U^\dagger$.

A Hermitian matrix Q is said to be nonnegative- (positive-)definite if all the eigenvalues of Q are nonnegative (positive). The matrix Q is a nonnegative-definite matrix if and only if there is another matrix C such that $Q = C^\dagger C$. Since the eigenvalues of Q are nonnegative, the diagonal matrix L has a square root, $\sqrt{L}$. Using the fact that $U^\dagger U = I$ we have

$$Q = ULU^\dagger = U\sqrt{L}U^\dagger U\sqrt{L}U^\dagger;$$

we then take $C = U\sqrt{L}U^\dagger$, so $C^\dagger = C$. Then $\mathbf{z}^\dagger Q\mathbf{z} = \mathbf{z}^\dagger C^\dagger C\mathbf{z} = ||C\mathbf{z}||^2$, so that Q is positive-definite if and only if C is invertible.

Exercise 2: Let A be an M by N matrix with complex entries. View A as a linear function with domain C^N, the space of all N-dimensional complex column vectors, and range contained within C^M, via the expression $A(\mathbf{x}) = A\mathbf{x}$. Suppose that $M > N$. The range of A, denoted $R(A)$, cannot be all of C^M. Show that every vector $\mathbf{z}$ in C^M can be written uniquely in the form $\mathbf{z} = A\mathbf{x} + \mathbf{w}$, where $A^\dagger\mathbf{w} = \mathbf{0}$. Show that $\|\mathbf{z}\|^2 = \|A\mathbf{x}\|^2 + \|\mathbf{w}\|^2$, where $\|\mathbf{z}\|^2$ denotes the square of the norm of $\mathbf{z}$. Hint: If $\mathbf{z} = A\mathbf{x} + \mathbf{w}$ then consider $A^\dagger\mathbf{z}$. Assume $A^\dagger A$ is invertible.

Exercise 3: When the complex M by N matrix A is stored in the computer it is usually *vectorized*; that is, the matrix

$$A = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1N} \\ A_{21} & A_{22} & \dots & A_{2N} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ A_{M1} & A_{M2} & \dots & A_{MN} \end{bmatrix}$$

becomes

$$\mathbf{vec}(A) = (A_{11}, A_{21}, ..., A_{M1}, A_{12}, A_{22}, ..., A_{M2}, ..., A_{MN})^T.$$

a: Show that the complex dot product $\mathbf{vec}(A)\cdot\mathbf{vec}(B) = \mathbf{vec}(B)^\dagger\mathbf{vec}(A)$ can be obtained by

$$\mathbf{vec}(A)\cdot\mathbf{vec}(B) = \text{trace}\,(AB^\dagger) = tr(AB^\dagger),$$

where, for a square matrix C, trace (C) means the sum of the entries along the main diagonal of C. We can therefore use the trace to define an inner product between matrices: $< A, B >= \text{trace}\,(AB^\dagger)$.

b: Show that $\text{trace}\,(AA^\dagger) \geq 0$ for all A, so that we can use the trace to define a norm on matrices: $||A||^2 = \text{trace}\,(AA^\dagger)$.

Exercise 4: Let $B = ULD^\dagger$ be an M by N matrix in diagonalized form; that is, L is an M by N diagonal matrix with entries $\lambda_1, ..., \lambda_K$ on its main diagonal, where $K = \min(M, N)$, and U and V are square matrices. Let the nth column of U be denoted $\mathbf{u^n}$ and similarly for the columns of V. Such a diagonal decomposition occurs in the *singular value decomposition* (SVD). Show that we can write

$$B = \lambda_1\mathbf{u^1}(\mathbf{v^1})^\dagger + ... + \lambda_K\mathbf{u^K}(\mathbf{v^K})^\dagger.$$

If B is an N by N Hermitian matrix then we can take $U = V$ and $K = M = N$, with the columns of U the eigenvectors of B, normalized to have Euclidean norm equal to one, and the λ_n to be the eigenvalues of B. In this case we may also assume that U is a *unitary* matrix, that is, $UU^\dagger = U^\dagger U = I$, where I denotes the identity matrix.

Regularization of linear systems of equations:

A consistent linear system of equations $A\mathbf{x} = \mathbf{b}$ is *ill-conditioned* if small changes in the entries of vector $\mathbf{b}$ can result in large changes in the solution. Such situations are common in signal processing and are usually dealt with by regularization. We consider regularization in this subsection.

We assume, throughout this subsection, that A is a real M by N matrix with full rank; then either AA^T or A^TA is invertible, whichever one has the smaller size.

Exercise 5: Show that the vector $\mathbf{x} = (x_1, ..., x_N)^T$ minimizes the mean squared error

$$\|A\mathbf{x} - \mathbf{b}\|^2 = \sum_{m=1}^{N}(Ax_m - b_m)^2,$$

if and only if $\mathbf{x}$ satisfies the system of linear equations $A^T(A\mathbf{x} - \mathbf{b}) = \mathbf{0}$, where $Ax_m = (A\mathbf{x})_m = \sum_{n=1}^{N} A_{mn}x_n$.

Hint: Calculate the partial derivatives of $\|A\mathbf{x} - \mathbf{b}\|^2$ with respect to each x_n.

Exercise 6: Let ϵ be in $(0, 1)$ and let I be the identity matrix whose dimensions are understood from the context. Show that

$$((1-\epsilon)AA^T + \epsilon I)^{-1}A = A((1-\epsilon)A^TA + \epsilon I)^{-1},$$

and, taking transposes,

$$A^T((1-\epsilon)AA^T + \epsilon I)^{-1} = ((1-\epsilon)A^TA + \epsilon I)^{-1}A^T.$$

Hint: use the identity

$$A((1-\epsilon)A^TA + \epsilon I) = ((1-\epsilon)AA^T + \epsilon I)A.$$

Exercise 7: Show that any vector $\mathbf{p}$ in R^N can be written as $\mathbf{p} = A^T\mathbf{q} + \mathbf{r}$, where $A\mathbf{r} = 0$.

We want to solve $A\mathbf{x} = \mathbf{b}$, at least in some approximate sense. Of course, there may be no solution, a unique solution or even multiple solutions. It often happens in applications that, even when there is an exact solution of $A\mathbf{x} = \mathbf{b}$, noise in the vector $\mathbf{b}$ makes such as exact solution undesirable; in such cases a *regularized solution* is usually used instead. Let $\epsilon > 0$ and define

$$F_\epsilon(x) = (1-\epsilon)\|A\mathbf{x} - \mathbf{b}\|^2 + \epsilon\|\mathbf{x} - \mathbf{p}\|^2.$$

Exercise 8: Show that F_ϵ always has a unique minimizer $\hat{\mathbf{x}}_\epsilon$ given by

$$\hat{\mathbf{x}}_\epsilon = ((1-\epsilon)A^TA + \epsilon I)^{-1}((1-\epsilon)A^T\mathbf{b} + \epsilon\mathbf{p});$$

this is a regularized solution of $A\mathbf{x} = \mathbf{b}$. Here $\mathbf{p}$ is a prior estimate of the desired solution. Note that the inverse above always exists.

What happens to $\hat{\mathbf{x}}_\epsilon$ as ϵ goes to zero? This will depend on which case we are in:

Case 1: $N \leq M$, A^TA invertible; or

Case 2: $N > M$, AA^T invertible.

Exercise 9: Show that, in Case 1, taking limits as $\epsilon \to 0$ on both sides of the expression for $\hat{\mathbf{x}}_\epsilon$ gives $\hat{\mathbf{x}}_\epsilon \to (A^TA)^{-1}A^T\mathbf{b}$, the least squares solution of $A\mathbf{x} = \mathbf{b}$.

We consider Case 2 now. Write $\mathbf{p} = A^T\mathbf{q} + \mathbf{r}$, with $A\mathbf{r} = \mathbf{0}$. Then

$$\hat{\mathbf{x}}_\epsilon = A^T((1-\epsilon)AA^T + \epsilon I)^{-1}((1-\epsilon)\mathbf{b} + \epsilon\mathbf{q}) + ((1-\epsilon)A^TA + \epsilon I)^{-1}(\epsilon\mathbf{r}).$$

Exercise 10: (a): Show that

$$((1-\epsilon)A^TA + \epsilon I)^{-1}(\epsilon\mathbf{r}) = \mathbf{r}, \forall\epsilon.$$

Hint: let

$$\mathbf{t}_\epsilon = ((1-\epsilon)A^TA + \epsilon I)^{-1}(\epsilon\mathbf{r}).$$

Then multiplying by A gives

$$A\mathbf{t}_\epsilon = A((1-\epsilon)A^TA + \epsilon I)^{-1}(\epsilon\mathbf{r}).$$

Now show that $A\mathbf{t}_\epsilon = 0$.
(b): Now take the limit of $\hat{\mathbf{x}}_\epsilon$, as $\epsilon \to 0$, to get $\hat{\mathbf{x}}_\epsilon \to A^T(AA^T)^{-1}\mathbf{b} + \mathbf{r}$. Show that this is the solution of $A\mathbf{x} = \mathbf{b}$ closest to $\mathbf{p}$.
Hint: Draw a diagram for the case of one equation in two unknowns.

Some useful matrix identities: In the exercise that follows we consider several matrix identities that are useful in developing the Kalman filter.

Exercise 11: Establish the following identities, assuming that all the products and inverses involved are defined:

$$CDA^{-1}B(C^{-1} - DA^{-1}B)^{-1} = (C^{-1} - DA^{-1}B)^{-1} - C; \tag{23.1}$$

$$(A - BCD)^{-1} = A^{-1} + A^{-1}B(C^{-1} - DA^{-1}B)^{-1}DA^{-1}; \tag{23.2}$$

$$A^{-1}B(C^{-1} - DA^{-1}B)^{-1} = (A - BCD)^{-1}BC; \tag{23.3}$$

$$(A - BCD)^{-1} = (I + GD)A^{-1}, \tag{23.4}$$

for

$$G = A^{-1}B(C^{-1} - DA^{-1}B)^{-1}.$$

Hints: To get equation (23.1) use

$$C(C^{-1} - DA^{-1}B) = I - CDA^{-1}B.$$

For the second identity, multiply both sides of equation (23.2) on the left by $A - BCD$ and at the appropriate step use the identity (23.1). For (23.3) show that

$$BC(C^{-1} - DA^{-1}B) = B - BCDA^{-1}B = (A - BCD)A^{-1}B.$$

For (23.4), substitute what G is and use (23.2).

Chapter 24

The Singular Value Decomposition

We saw earlier that an N by N Hermitian matrix H can be written in terms of its eigenvalues and eigenvectors as $H = ULU^\dagger$ or as

$$H = \sum_{n=1}^{N} \lambda_n \mathbf{u^n} (\mathbf{u^n})^\dagger.$$

The *singular value decomposition* (SVD) is a similar result that applies to any rectangular matrix. It is an important tool in image compression and pseudo-inversion.

Let C be any N by K complex matrix, with $K \geq N$. Let $A = C^\dagger C$ and $B = CC^\dagger$; we assume, reasonably, that B, the smaller of the two matrices, is invertible, so all the eigenvalues $\lambda_1, ..., \lambda_N$ of B are positive. Then write the eigenvalue/eigenvector decomposition of B as $B = ULU^\dagger$.

Exercise 1: Show that the nonzero eigenvalues of A and B are the same.

Let V be the K by K matrix whose first N columns are those of the matrix $C^\dagger U L^{-1/2}$ and whose remaining $K - N$ columns are any mutually orthogonal norm-one vectors that are all orthogonal to each of the first N columns. Let M be the N by K matrix with diagonal entries $M_{nn} = \sqrt{\lambda_n}$ for $n = 1, ..., N$ and whose remaining entries are zero. The nonzero entries of M, $\sqrt{\lambda_n}$, are called the *singular values* of C. The *singular value decomposition* (SVD) of C is $C = UMV^\dagger$. The SVD of $C^\dagger$ is $C^\dagger = VM^T U^\dagger$.

Exercise 2: Show that $UMV^\dagger$ equals C.

Using the SVD of C we can write

$$C = \sum_{n=1}^{N} \sqrt{\lambda_n} \mathbf{u^n} (\mathbf{v^n})^\dagger,$$

where $\mathbf{v^n}$ denotes the n-th column of the matrix V.

In image processing matrices such as C are used to represent discrete two-dimensional images, with the entries of C corresponding to the grey level or color at each pixel. It is common to find that most of the N singular values of C are nearly zero, so that C can be written approximately as a sum of far fewer than N dyads; this is SVD image compression.

If $N \neq K$ then C cannot have an inverse; it does, however, have a *pseudo-inverse*, $C^* = VM^*U^\dagger$, where M^* is the matrix obtained from M by taking the inverse of each of its nonzero entries and leaving the remaining zeros the same.

Some important properties of the pseudo-inverse are the following:

a. $CC^*C = C$;

b. $C^*CC^* = C^*$;

c. $(C^*C)^\dagger = C^*C$;

d. $(CC^*)^\dagger = CC^*$.

The pseudo-inverse C^* can be used in much the same way as the inverse is, to obtain exact or approximate solutions of systems of equations $C\mathbf{x} = \mathbf{d}$ by multiplying $\mathbf{d}$ by C^*, as the examples in the next two exercises illustrate.

Exercise 3: If $N > K$ the system $C\mathbf{x} = \mathbf{d}$ probably has no exact solution. Show that $C^* = (C^\dagger C)^{-1}C^\dagger$ so that the vector $\mathbf{x} = C^*\mathbf{d}$ is the least squares approximate solution.

Exercise 4: If $N < K$ the system $C\mathbf{x} = \mathbf{d}$ probably has infinitely many solutions. Show that the pseudo-inverse is now $C^* = C^\dagger(CC^\dagger)^{-1}$, so that the vector $\mathbf{x} = C^*\mathbf{d}$ is the exact solution of $C\mathbf{x} = \mathbf{d}$ closest to the origin; that is, it is the minimum norm solution.

Chapter 25

Discrete Random Processes

The most common model used in signal processing is that of a sum of complex exponential functions plus noise. The noise is viewed as a sequence of random variables, and the signal components also may involve random parameters, such as random amplitudes and phase angles. Such models are best studied as *discrete random processes*.

A discrete random process is an infinite sequence $\{X_n\}_{n=-\infty}^{+\infty}$ in which each X_n is a complex-valued random variable. The *autocorrelation function* associated with the random process is defined for all index values m and n by $r_x(m,n) = E(X_m\overline{X_n})$, where $E(\cdot)$ is the expectation or expected value operator. For $m = n$ we get $r(n,n) = \text{variance}(X_n)$. We say that the random process is *wide-sense stationary* if $E(X_n)$ is independent of n and $r_x(m,n)$ is a function only of the difference, $m-n$, so that variance(X_n) is independent of n. The autocorrelation function can then be redefined as $r_x(k) = E(X_{n+k}\overline{X_n})$. The *power spectrum* $R_x(\omega)$ of the random process is defined using the values $r_x(k)$ as its Fourier coeffcients:

$$R_x(\omega) = \sum_{k=-\infty}^{+\infty} r_x(k)e^{ik\omega},$$

for all ω in the interval $[-\pi,\pi]$. It can be proved that the power spectrum is a nonnegative function of the form $R_x(\omega) = |G(\omega)|^2$ and the autocorrelation sequence $\{r_x(k)\}$ satisfies the equations

$$r_x(k) = \sum_{n=-\infty}^{+\infty} g_{k+n}\overline{g_n},$$

for

$$G(\omega) = \sum_{n=-\infty}^{+\infty} g(n)e^{in\omega}.$$

In practice we will have actual values $X_n = x_n$, for only finitely many of the X_n, say for $n = 1, ..., m$. These can be used to estimate the values $r_x(k)$, at least for values of k between, say, $-M/5$ and $M/5$. For example, we could estimate $r_x(k)$ by averaging all the products of the form $x_{k+m}\overline{x_m}$ that we can compute from the data. Clearly, as k gets farther away from zero we have fewer such products, so our average is a less accurate estimate.

Once we have $r_x(k)$, $|k| \leq N$ we form the $N+1$ by $N+1$ *autocorrelation matrix* R having the entries $R_{m,n} = r_x(m-n)$. This autocorrelation matrix is what is used in the design of optimal filtering.

The matrix R is *Hermitian*, that is, $R_{n,m} = \overline{R_{m,n}}$, so that $R^\dagger = R$. An M by M Hermitian matrix H is said to be *nonnegative-definite* if, for all complex column vectors $\mathbf{a} = (a_1, ..., a_M)^T$, the *quadratic form* $\mathbf{a}^\dagger H\mathbf{a}$ is a nonnegative number and *positive-definite* if such a quadratic form is always positive.

Exercise 1: Show that the autocorrelation matrix R is nonnegative definite. Hint: Let

$$A(\omega) = \sum_{n=1}^{N+1} a_n e^{in\omega}$$

and express the integral

$$\int |A(\omega)|^2 R(\omega) d\omega$$

in terms of the a_n and the $R_{m,n}$. Under what conditions can R fail to be positive-definite?

Later we shall consider the *maximum entropy* method for estimating the power spectrum from finitely many values of $r_x(k)$.

Autoregressive processes: We noted at the beginning of the chapter that the case of a discrete-time signal with additive random noise provides a good example of a discrete random process; there are others. One particularly important type is the *autoregressive* (AR) process, which is closely related to ordinary linear differential equations.

When a smooth periodic function has noise added the new function is rough. Imagine, though, a fairly weighty pendulum of a clock, moving smoothly and periodically. Now imagine that a young child is throwing small stones at the bob of the pendulum. The movement of the pendulum is no longer periodic, but it is not rough. The pendulum is moving randomly in response to the random external disturbance, but not as if a random noise component has been added to its motion. To model such random processes we need to extend the notion of an ordinary differential equation. That leads us to the AR processes.

Recall that an ordinary linear M-th order differential equation with constant coefficients has the form

$$x^{(M)}(t) + c_1 x^{(M-1)}(t) + c_2 x^{(M-2)}(t) + ... + c_{M-1} x'(t) + c_M x(t) = f(t),$$

where $x^{(m)}(t)$ denotes the m-th derivative of the function $x(t)$ and the c_m are constants. In many applications the variable t is time and the function $f(t)$ is an external effect driving the linear system, with system response given by the unknown function $x(t)$. How the system responds to a variety of external drivers is of great interest. It is sometimes convenient to replace this continuous formulation with a discrete analog, called a *difference equation*.

In switching from differential equations to difference equations we discretize the time variable and replace the driving function $f(t)$ with f_n, $x(t)$ with x_n, the first derivative at time t, $x'(t)$, with the first difference, $x_n - x_{n-1}$, the second derivative $x''(t)$ with the second difference, $(x_n - x_{n-1}) - (x_{n-1} - x_{n-2})$, and so on. The differential equation is then replaced by the difference equation

$$x_n - a_1 x_{n-1} - a_2 x_{n-2} - ... - a_M x_{n-M} = f_n \tag{25.1}$$

for some constants a_m; the negative signs are a technical convenience only.

We now assume that the driving function is a discrete random process $\{f_n\}$, so that the system response becomes a discrete random process, $\{X_n\}$. If we assume that the driver f_n is white noise, independent of the $\{X_n\}$, then the process $\{X_n\}$ is called an autoregressive (AR) process. What the system does at time n depends partly on what it has done at the M discrete times prior to time n, as well as what the external disturbance f_n is at time n. Our goal is usually to determine the constants a_m; this is *system identification*. Our data is typically some number of consecutive measurements of the X_n.

Multiplying both sides of equation (25.1) by $\overline{X_{n-k}}$, for some $k > 0$ and taking the expected value, we obtain

$$E(X_n \overline{X_{n-k}}) - ... - a_M E(X_{n-M} \overline{X_{n-k}}) = 0.$$

or

$$r_x(k) - a_1 r_x(k-1) - ... - a_M r_x(k-M) = 0.$$

Taking $k = 0$ we get

$$r_x(0) - a_1 r_x(-1) - ... - a_M r_x(-M) = E(|f_n|^2) = \text{var}\,(f_n).$$

To find the a_m we use the data to estimate $r_x(k)$ at least for $k = 0, 1, ..., M$. Then we use these estimates in the linear equations above, solving them for the a_m.

Linear systems with random input: In our discussion of discrete linear filters, also called time-invariant linear systems, we noted that it is common to consider as the input to such a system a discrete random process, $\{X_n\}$. The output is then another random process $\{Y_n\}$ given by

$$Y_n = \sum_{m=-\infty}^{+\infty} g_m X_{n-m},$$

for each n.

Exercise 2: Show that if the input process is wide-sense stationary then so is the output. Show that the power spectrum $R_y(\omega)$ of the output is

$$R_y(\omega) = |G(\omega)|^2 R_x(\omega).$$

Chapter 26

Best Linear Unbiased Estimation

In most signal and image processing applications the measured data includes unwanted components termed *noise*. Noise often appears as an additive term, which we then try to remove. If we knew precisely the noisy part added to each data value we would simply subtract it; of course, we never have such information. How then do we remove something when we don't know what it is? Statistics provides a way out.

The basic idea in statistics is to use procedures that perform well on average, when applied to a class of problems. The procedures are built using properties of that class, usually involving probabilistic notions, and are evaluated by examining how they would have performed had they been applied to every problem in the class. To use such methods to remove additive noise we need a description of the class of noises we expect to encounter, not specific values of the noise component in any one particular instance. We also need some idea about what signal components look like. In this chapter we discuss solving this noise removal problem using the *best linear unbiased estimation* (BLUE) . We begin with the simplest case and then proceed to discuss increasingly complex scenarios.

The simplest problem:

Suppose our data is $z_j = c + v_j$, for $j = 1, ..., J$, where c is an unknown constant to be estimated and the v_j are additive noise. We assume that $E(v_j) = 0, E(v_j\overline{v_k}) = 0$, for $j \neq k$ and $E(|v_j|^2) = \sigma_j^2$. So the additive noises are assumed to have mean zero and to be independent (or at least uncorrelated). In order to estimate c we adopt the following rules:

a. The estimate $\hat{c}$ is *linear* in the data $\mathbf{z} = (z_1, ..., z_J)^T$; that is, $\hat{c} = \mathbf{k}^\dagger \mathbf{z}$,

for some vector $\mathbf{k} = (k_1, ..., k_J)^T$.

b. The estimate is *unbiased*; that is $E(\hat{c}) = c$. This means $\sum_{j=1}^{J} k_j = 1$.

c. The estimate is best in the sense that it minimizes the expected error squared; that is, $E(|\hat{c} - c|^2)$ is minimized.

The resulting vector $\mathbf{k}$ is calculated to be

$$k_i = \sigma_i^{-2} / (\sum_{j=1}^{J} \sigma_j^{-2})$$

and the BLUE estimator of c is then

$$\hat{c} = \sum_{i=1}^{J} z_i \sigma_i^{-2} / (\sum_{j=1}^{J} \sigma_j^{-2}).$$

The general case of the BLUE:

Suppose now that our data vector is $\mathbf{z} = H\mathbf{x} + \mathbf{v}$. Here $\mathbf{x}$ is a random vector whose value is to be estimated, the random vector $\mathbf{v}$ is additive noise whose mean is $E(\mathbf{v}) = 0$ and whose correlation matrix is $Q = E(\mathbf{v}\mathbf{v}^\dagger)$, not necessarily diagonal, and the known matrix H is J by N, with $J > N$. Now we seek an estimate of the vector $\mathbf{x}$. The rules we use are now

a. The estimate $\hat{\mathbf{x}}$ must have the form $\hat{\mathbf{x}} = K^\dagger \mathbf{z}$, where the matrix K is to be determined.

b. The estimate is unbiased; that is, $E(\hat{\mathbf{x}}) = E(\mathbf{x})$.

c. The K is determined as the minimizer of the expected squared error; that is, once again we minimize $E(|\hat{\mathbf{x}} - \mathbf{x}|^2)$.

Exercise 1: Show that

$$E(|\hat{\mathbf{x}} - \mathbf{x}|^2) = \text{trace}\, K^\dagger Q K.$$

Hints: Write the left side as

$$E(\text{trace}\,((\hat{\mathbf{x}} - \mathbf{x})(\hat{\mathbf{x}} - \mathbf{x})^\dagger).$$

Also use the fact that the trace and expected value operations commute.

Exercise 2: Show that for the estimator to be unbiased we need $K^\dagger H = I$, the identity matrix.

The problem then is to minimize trace $K^\dagger QK$ subject to the constraint equation $K^\dagger H = I$. We solve this problem using a technique known as *prewhitening*.

Since the noise correlation matrix Q is Hermitian and nonnegative definite, we have $Q = UDU^\dagger$, where the columns of U are the (mutually orthogonal) eigenvectors of Q and D is a diagonal matrix whose diagonal entries are the (necessarily nonnegative) eigenvalues of Q; therefore, $U^\dagger U = I$. We call $C = UD^{1/2}U^\dagger$ the Hermitian square root of Q, since $C^\dagger = C$ and $C^2 = Q$. We assume that Q is invertible, so that C is also. Given the system of equations

$$\mathbf{z} = H\mathbf{x} + \mathbf{v},$$

as above, we obtain a new system

$$\mathbf{y} = G\mathbf{x} + \mathbf{w}$$

by multiplying both sides by $C^{-1} = Q^{-1/2}$; here $G = C^{-1}H$ and $\mathbf{w} = C^{-1}\mathbf{v}$. The new noise correlation matrix is

$$E(\mathbf{w}\mathbf{w}^\dagger) = C^{-1}QC^{-1} = I,$$

so the new noise is white. For this reason the step of multiplying by C^{-1} is called *prewhitening*.

With $J = CK$ and $M = C^{-1}H$ we have

$$K^\dagger QK = J^\dagger J$$

and

$$K^\dagger H = J^\dagger M.$$

Our problem then is to minimize trace $J^\dagger J$, subject to $J^\dagger M = I$.

Let $L = L^\dagger = (M^\dagger M)^{-1}$ and let $f(J)$ be the function

$$f(J) = \text{trace}[(J^\dagger - L^\dagger M^\dagger)(J - ML)].$$

The minimum value of $f(J)$ is zero, which occurs when $J = LM$. Note that this choice for J has the property $J^\dagger M = I$. So minimizing $f(J)$ is equivalent to minimizing $f(J)$ subject to the constraint $J^\dagger M = I$ and both problems have the solution $J = LM$. But minimizing $f(J)$ subject to $J^\dagger M = I$ is equivalent to minimizing trace $J^\dagger J$ subject to $J^\dagger M = I$, which is our original problem. Therefore the optimal choice for J is $J = LM$. Consequently the optimal choice for K is

$$K = Q^{-1}HL = Q^{-1}H(H^\dagger Q^{-1}H)^{-1}.$$

and the BLUE estimate of $\mathbf{x}$ is

$$\hat{\mathbf{x}} = K^\dagger \mathbf{z} = (H^\dagger Q^{-1}H)^{-1}H^\dagger Q^{-1}\mathbf{z}.$$

The simplest case can be obtained from this more general formula by taking $N = 1$, $H = (1, 1, ..., 1)^T$ and $\mathbf{x} = c$.

Note that if the noise is *white*, that is, $Q = \sigma^2 I$, then $\hat{\mathbf{x}} = (H^\dagger H)^{-1} H^\dagger \mathbf{z}$, which is the least squares solution of the equation $\mathbf{z} = H\mathbf{x}$. The effect of requiring that the estimate be unbiased is that, in this case, we simply ignore the presence of the noise and calculate the least squares solution of the noise-free equation $\mathbf{z} = H\mathbf{x}$.

The BLUE with a prior estimate

In Kalman filtering we have the situation in which we want to estimate the random vector $\mathbf{x}$ given measurements $\mathbf{z} = H\mathbf{x} + \mathbf{v}$, but also given a prior estimate $\mathbf{y}$ of $\mathbf{x}$. It is the case there that $E(\mathbf{y}) = E(\mathbf{x})$, so we write $\mathbf{y} = \mathbf{x} + \mathbf{w}$, with $\mathbf{w}$ independent of both $\mathbf{x}$ and $\mathbf{v}$ and $E(\mathbf{w}) = \mathbf{0}$. The covariance matrix for $\mathbf{w}$ we denote by $E(\mathbf{w}\mathbf{w}^\dagger) = R$. We now require that the estimate $\hat{\mathbf{x}}$ be linear in both $\mathbf{z}$ and $\mathbf{y}$; that is, the estimate has the form

$$\hat{\mathbf{x}} = C^\dagger \mathbf{z} + D^\dagger \mathbf{y},$$

for matrices C and D to be determined.

The approach is to apply the BLUE to the combined system of linear equations

$$\mathbf{z} = H\mathbf{x} + \mathbf{v},$$

$$\mathbf{y} = \mathbf{x} + \mathbf{w}.$$

In matrix language this combined system becomes $\mathbf{u} = J\mathbf{x} + \mathbf{n}$, with $\mathbf{u}^T = [\mathbf{z}^T \ \mathbf{y}^T]$, $J^T = [H^T \ I^T]$ and $\mathbf{n}^T = [\mathbf{v}^T \ \mathbf{w}^T]$. The noise covariance matrix becomes

$$P = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix}.$$

The BLUE estimate is $K^\dagger \mathbf{u}$, with $K^\dagger J = I$. Minimizing the variance, we find that the optimal $K^\dagger$ is

$$K^\dagger = (J^\dagger P^{-1} J)^{-1} J^\dagger P^{-1}.$$

The optimal estimate is then

$$\hat{\mathbf{x}} = (H^\dagger Q^{-1} H + R^{-1})^{-1} (H^\dagger Q^{-1} \mathbf{z} + R^{-1} \mathbf{y}).$$

Therefore

$$C^\dagger = (H^\dagger Q^{-1} H + R^{-1})^{-1} H^\dagger Q^{-1}$$

and

$$D^\dagger = (H^\dagger Q^{-1} H + R^{-1})^{-1} R^{-1}.$$

Using the matrix identities in equations (23.2) and (23.3) we can rewrite this estimate in the more useful form

$$\hat{\mathbf{x}} = \mathbf{y} + G(\mathbf{z} - H\mathbf{y}),$$

for

$$G = RH^\dagger(Q + HRH^\dagger)^{-1}. \tag{26.1}$$

The covariance matrix of the optimal estimator is $K^\dagger PK$, which can be written as

$$K^\dagger PK = (R^{-1} + H^\dagger Q^{-1} H)^{-1} = (I - GH)R.$$

In the context of the Kalman filter R is the covariance of the prior estimate of the current state, G is the Kalman gain matrix and $K^\dagger PK$ is the posterior covariance of the current state. The algorithm proceeds recursively from one state to the next in time.

Adaptive BLUE

We have assumed so far that we know the covariance matrix Q corresponding to the measurement noise. If we do not, then we may attempt to estimate Q from the measurements themselves; such methods are called *noise-adaptive*. To illustrate, let the *innovations* vector be $\mathbf{e} = \mathbf{z} - H\mathbf{y}$. Then the covariance matrix of $\mathbf{e}$ is $S = HRH^\dagger + Q$. Having obtained an estimate $\hat{S}$ of S from the data, we use $\hat{S} - HRH^\dagger$ in place of Q in equation (26.1).

In this chapter we have focused on the filtering problem: given the data vector $\mathbf{z}$, estimate $\mathbf{x}$, assuming that $\mathbf{z}$ consists of noisy measurements of $H\mathbf{x}$; that is, $\mathbf{z} = H\mathbf{x} + \mathbf{v}$. An important extension of this problem is that of stochastic prediction. In the next chapter we discuss the Kalman filter method for solving this more general problem.

Chapter 27

Kalman Filters

One area in which prediction plays an important role is the tracking of moving targets, such as ballistic missiles, using radar. The range to the target, its angle of elevation and its azimuthal angle are all functions of time governed by linear differential equations. The *state vector* of the system at time t might then be a vector with nine components, the three functions just mentioned, along with their first and second derivatives. In theory, if we knew the initial state perfectly and our differential equations model of the physics was perfect, that would be enough to determine the future states. In practice neither of these is true and we need to assist the differential equation by taking radar measurements of the state at various times. The problem then is to estimate the state at time t using both the measurements taken prior to time t and the estimate based on the physics.

When such tracking is performed digitally the functions of time are replaced by discrete sequences. Let the state vector at time $k\Delta t$ be denoted by $\mathbf{x}_k$, for k an integer and $\Delta t > 0$. Then, with the derivatives in the differential equation approximated by divided differences, the physical model for the evolution of the system in time becomes

$$\mathbf{x}_k = A_{k-1}\mathbf{x}_{k-1} + \mathbf{m}_{k-1}.$$

The matrix A_{k-1}, which we assume is known, is obtained from the differential equation, which may have nonconstant coefficients, as well as from the divided difference approximations to the derivatives. The random vector sequence $\mathbf{m}_{k-1}$ represents the error in the physical model due to the discretization and necessary simplification inherent in the original differential equation itself. We assume that the expected value of $\mathbf{m_k}$ is zero for each k. The covariance matrix is $E(\mathbf{m}_k\mathbf{m}_k^\dagger) = M_k$.

At time $k\Delta t$ we have the measurements

$$\mathbf{z}_k = H_k\mathbf{x}_k + \mathbf{v}_k,$$

where H_k is a known matrix describing the nature of the linear measurements of the state vector and the random vector $\mathbf{v}_k$ is the noise in these measurements. We assume that the mean value of $\mathbf{v}_k$ is zero for each k. The covariance matrix is $E(\mathbf{v}_k\mathbf{v}_k^\dagger) = Q_k$. We assume that the initial state vector $\mathbf{x}_0$ is random and independent of the noise sequences.

Given an estimate $\hat{\mathbf{x}}_{k-1}$ of the state vector $\mathbf{x}_{k-1}$, our prior estimate of $\mathbf{x}_k$ based solely on the physics is

$$\mathbf{y}_k = A_{k-1}\hat{\mathbf{x}}_{k-1}.$$

Exercise 1: Show that $E(\mathbf{y}_k - \mathbf{x}_k) = 0$, so the prior estimate of $\mathbf{x}_k$ is unbiased. We can then write $\mathbf{y}_k = \mathbf{x}_k + \mathbf{w}_k$, with $E(\mathbf{w}_k) = \mathbf{0}$.

Kalman filtering: The *Kalman filter* [99], [79], [56] is a recursive algorithm to estimate the state vector $\mathbf{x}_k$ at time $k\Delta t$ as a linear combination of the vectors $\mathbf{z}_k$ and $\mathbf{y}_k$. The estimate $\hat{\mathbf{x}}_k$ will have the form

$$\hat{\mathbf{x}}_k = C_k^\dagger\mathbf{z}_k + D_k^\dagger\mathbf{y}_k, \tag{27.1}$$

for matrices C_k and D_k to be determined. As we shall see, this estimate can also be written as

$$\hat{\mathbf{x}}_k = \mathbf{y}_k + G_k(\mathbf{z}_k - H_k\mathbf{y}_k), \tag{27.2}$$

which shows that the estimate involves a prior prediction step, the $\mathbf{y}_k$, followed by a correction step, in which $H_k\mathbf{y}_k$ is compared to the measured data vector $\mathbf{z}_k$; such estimation methods are sometimes called *predictor-corrector methods*.

In our discussion of the BLUE we saw how to incorporate a prior estimate of the vector to be estimated. The trick was to form a larger matrix equation and then to apply the BLUE to that system. The Kalman filter does just that.

The correction step in the Kalman filter uses the BLUE to solve the combined linear system

$$\mathbf{z}_k = H_k\mathbf{x}_k + \mathbf{v}_k$$

and

$$\mathbf{y}_k = \mathbf{x}_k + \mathbf{w}_k.$$

The covariance matrix of $\hat{\mathbf{x}}_{k-1} - \mathbf{x}_{k-1}$ is denoted P_{k-1} and we let $Q_k = E(\mathbf{w}_k\mathbf{w}_k^\dagger)$. The covariance matrix of $\mathbf{y}_k - \mathbf{x}_k$ is

$$\mathrm{cov}(\mathbf{y}_k - \mathbf{x}_k) = R_k = M_{k-1} + A_{k-1}P_{k-1}A_{k-1}^\dagger.$$

It follows from our earlier discussion of the BLUE that the estimate of $\mathbf{x}_k$ is

$$\hat{\mathbf{x}}_k = \mathbf{y}_k + G_k(\mathbf{z}_k - H\mathbf{y}_k),$$

with

$$G_k = R_k H_k^\dagger (Q_k + H_k R_k H_k^\dagger)^{-1}.$$

Then the covariance matrix of $\hat{\mathbf{x}}_k - \mathbf{x}_k$ is

$$P_k = (I - G_k H_k) R_k.$$

The recursive procedure is to go from P_{k-1} and M_{k-1} to R_k, then to G_k, from which $\hat{\mathbf{x}}_k$ is formed, and finally to P_k, which, along with the known matrix M_k, provides the input to the next step. The time-consuming part of this recursive algorithm is the matrix inversion in the calculation of G_k. Simpler versions of the algorithm are based on the assumption that the matrices Q_k are diagonal, or on the convergence of the matrices G_k to a limiting matrix G [56].

There are many variants of the Kalman filter, corresponding to variations in the physical model, as well as in the statistical assumptions. The differential equation may be nonlinear, so that the matrices A_k depend on $\mathbf{x}_k$. The system noise sequence $\{\mathbf{w}_k\}$ and the measurement noise sequence $\{\mathbf{v}_k\}$ may be correlated. For computational convenience the various functions that describe the state may be treated separately. The model may include known external inputs to drive the differential system, as in the tracking of spacecraft capable of firing booster rockets. Finally, the noise covariance matrices may not be known *a priori* and adaptive filtering may be needed. We discuss this last issue briefly in the next section.

Adaptive Kalman filtering: As in [56] we consider only the case in which the covariance matrix Q_k of the measurement noise $\mathbf{v}_k$ is unknown. As we saw in the discussion of adaptive BLUE, the covariance matrix of the innovations vector $\mathbf{e}_k = \mathbf{z}_k - H_k \mathbf{y}_k$ is

$$S_k = H_k R_k H_k^\dagger + Q_k.$$

Once we have an estimate for S_k, we estimate Q_k using

$$\hat{Q}_k = \hat{S}_k - H_k R_k H_k^\dagger.$$

We might assume that S_k is independent of k and estimate $S_k = S$ using past and present innovations; for example, we could use

$$\hat{S} = \frac{1}{k-1} \sum_{j=1}^{k} (\mathbf{z}_j - H_j \mathbf{y}_j)(\mathbf{z}_j - H_j \mathbf{y}_j)^\dagger.$$

Chapter 28

The Vector Wiener Filter

The vector Wiener filter (VWF) provides another method for estimating the vector $\mathbf{x}$ given noisy measurements $\mathbf{z}$, where

$$\mathbf{z} = H\mathbf{x} + \mathbf{v},$$

with $\mathbf{x}$ and $\mathbf{v}$ independent random vectors and H a known matrix. We shall assume throughout this chapter that $E(\mathbf{v}) = \mathbf{0}$ and let $Q = E(\mathbf{v}\mathbf{v}^\dagger)$.

It is common to formulate the VWF in the context of filtering a signal vector $\mathbf{s}$ from signal plus noise. The data is the vector

$$\mathbf{z} = \mathbf{s} + \mathbf{v}$$

and we want to estimate $\mathbf{s}$. Each entry of our estimate of the vector $\mathbf{s}$ will be a linear combination of the data values; that is, our estimate is $\hat{\mathbf{s}} = B^\dagger \mathbf{z}$ for some matrix B to be determined. This B will be called the *vector Wiener filter*. To extract the signal from the noise we must know something about possible signals and possible noises. We consider several stages of increasing complexity and correspondence with reality.

Suppose, initially, that all signals must have the form $\mathbf{s} = a\mathbf{u}$, where a is an unknown scalar and $\mathbf{u}$ is a known vector. Suppose that all noises must have the form $\mathbf{v} = b\mathbf{w}$, where b is an unknown scalar and $\mathbf{w}$ is a known vector. Then to estimate $\mathbf{s}$ we must find a. So long as $J \geq 2$ we should be able to solve for a and b. We form the two equations

$$\mathbf{u}^\dagger \mathbf{z} = a\mathbf{u}^\dagger \mathbf{u} + b\mathbf{u}^\dagger \mathbf{w}$$

and

$$\mathbf{w}^\dagger \mathbf{z} = a\mathbf{w}^\dagger \mathbf{u} + b\mathbf{w}^\dagger \mathbf{w}.$$

This system of two equations in two unknowns will have a unique solution unless $\mathbf{u}$ and $\mathbf{w}$ are proportional, in which case we cannot expect to distinguish signal from noise.

We move now to a somewhat more complicated model. Suppose now that all signals must have the form

$$\mathbf{s} = \sum_{n=1}^{N} a_n \mathbf{u}^n,$$

where the a_n are unknown scalars and the $\mathbf{u}^n$ are known vectors. Suppose that all noises must have the form

$$\mathbf{v} = \sum_{m=1}^{M} b_m \mathbf{w}^m,$$

where the b_m are unknown scalars and $\mathbf{w}^m$ are known vectors. Then to estimate $\mathbf{s}$ we must find the a_n. So long as $J \geq N+M$ we should be able to solve for the unique a_n and b_m. However, we usually do not know a great deal about the signal and the noise, so we find ourselves in the situation in which the N and M are large. Let U be the J by N matrix whose nth column is $\mathbf{u}^n$ and W the J by M matrix whose mth column is $\mathbf{w}^m$. Let V be the J by $N+M$ matrix whose first N columns contain U and whose last M columns contain W; so $V = [U \quad W]$. Let $\mathbf{c}$ be the $N+M$ by 1 column vector whose first N entries are the a_n and whose last M entries are the b_m. We want to solve $\mathbf{z} = V\mathbf{c}$. But this system of linear equations has too many unknowns when $N+M > J$, so we seek the minimum norm solution. In closed form this solution is

$$\hat{\mathbf{c}} = V^\dagger (VV^\dagger)^{-1} \mathbf{z}.$$

The matrix $VV^\dagger = (UU^\dagger + WW^\dagger)$ involves the *signal correlation matrix* $UU^\dagger$ and the *noise correlation matrix* $WW^\dagger$. Consider $UU^\dagger$. The matrix $UU^\dagger$ is J by J and the (i,j) entry of $UU^\dagger$ is given by

$$UU^\dagger_{ij} = \sum_{n=1}^{N} u_i^n u_j^n,$$

so the matrix $\frac{1}{N} UU^\dagger$ has for its entries the average, over all the $n = 1, ..., N$, of the product of the ith and jth entries of the vectors $\mathbf{u}^n$. Therefore, $\frac{1}{N} UU^\dagger$ is statistical information about the signal; it tells us how these products look, on average, over all members of the family $\{\mathbf{u}^n\}$, the *ensemble*, to use the statistical word.

To pass to a more formal statistical framework, we let the coefficient vectors $\mathbf{a} = (a_1, a_2, ..., a_N)^T$ and $\mathbf{b} = (b_1, b_2, ..., b_M)^T$ be independent random white noise vectors, both with mean zero and covariance matrices $E(\mathbf{a}\mathbf{a}^\dagger) = I$ and $E(\mathbf{b}\mathbf{b}^\dagger) = I$. Then

$$UU^\dagger = E(\mathbf{s}\mathbf{s}^\dagger) = R_s$$

and

$$WW^\dagger = E(\mathbf{v}\mathbf{v}^\dagger) = Q = R_v.$$

The estimate of $\mathbf{s}$ is the result of applying the vector Wiener filter to the vector $\mathbf{z}$ and is given by

$$\hat{\mathbf{s}} = UU^\dagger(UU^\dagger + WW^\dagger)^{-1}\mathbf{z}.$$

Exercise 1: Apply the vector Wiener filter to the simplest problem discussed earlier; here let $N = 1$. It will help to use the *matrix inversion identity*

$$(Q + \mathbf{u}\mathbf{u}^\dagger)^{-1} = Q^{-1} - (1 + \mathbf{u}^\dagger Q^{-1}\mathbf{u})^{-1}Q^{-1}\mathbf{u}\mathbf{u}^\dagger Q^{-1}. \tag{28.1}$$

The VWF and the BLUE: To apply the VWF to the problem considered in the discussion of the BLUE let the vector $\mathbf{s}$ be $H\mathbf{x}$. We assume, in addition, that the vector $\mathbf{x}$ is a white noise vector; that is, $E(\mathbf{x}\mathbf{x}^\dagger) = \sigma^2 I$. Then $R_s = \sigma^2 HH^\dagger$.

In the VWF approach we estimate $\mathbf{s}$ using

$$\hat{\mathbf{s}} = B^\dagger \mathbf{z},$$

where the matrix B is chosen so as to minimize the mean squared error, $E|\hat{\mathbf{s}} - \mathbf{s}|^2$. This is equivalent to minimizing

$$\text{trace}\, E((B\mathbf{z} - \mathbf{s})(B\mathbf{z} - \mathbf{s})^\dagger).$$

Expanding the matrix products and using the definitions above, we see that we must minimize

$$\text{trace}\,(B^\dagger(R_s + R_v)B - R_sB - B^\dagger R_s + R_s).$$

Differentiating with respect to the matrix B using equations (**??**) and (**??**), we find

$$(R_s + R_v)B - R_s = 0,$$

so that

$$B = (R_s + R_v)^{-1}R_s.$$

Our estimate of the signal component is then

$$\hat{\mathbf{s}} = R_s(R_s + R_v)^{-1}\mathbf{z}.$$

With $\mathbf{s} = H\mathbf{x}$, our estimate of $\mathbf{s}$ is

$$\hat{\mathbf{s}} = \sigma^2 HH^\dagger(\sigma^2 HH^\dagger + Q)^{-1}\mathbf{z}$$

and the VWF estimate of $\mathbf{x}$ is

$$\hat{\mathbf{x}} = \sigma^2 H^\dagger(\sigma^2 HH^\dagger + Q)^{-1}\mathbf{z}.$$

How does this estimate relate to the one we got from the BLUE?

The BLUE estimate of $\mathbf{x}$ is

$$\hat{\mathbf{x}} = (H^\dagger Q^{-1} H)^{-1} H^\dagger Q^{-1} \mathbf{z}.$$

From the matrix identity in equation (23.3) we know that

$$(H^\dagger Q^{-1} H + \sigma^{-2} I)^{-1} H^\dagger Q^{-1} = \sigma^2 H^\dagger (\sigma^2 H H^\dagger + Q)^{-1}.$$

Therefore the VWF estimate of $\mathbf{x}$ is

$$\hat{\mathbf{x}} = (H^\dagger Q^{-1} H + \sigma^{-2} I)^{-1} H^\dagger Q^{-1} \mathbf{z}.$$

Note that the BLUE estimate is unbiased and unaffected by changes in the signal strength or the noise strength. In contrast, the VWF is not unbiased and does depend on the signal-to-noise ratio; that is, it depends on the ratio $\sigma^2/\text{trace}\,(Q)$. The BLUE estimate is the limiting case of the VWF estimate, as the signal-to-noise ratio goes to infinity.

The BLUE estimates $\mathbf{s} = H\mathbf{x}$ by first finding the BLUE estimate of $\mathbf{x}$ and then multiplying it by H to get the estimate of the signal $\mathbf{s}$.

Exercise 2: Show that the mean squared error in the estimation of $\mathbf{s}$ is

$$E(|\hat{\mathbf{s}} - \mathbf{s}|^2) = \text{trace}\,(H(H^\dagger Q^{-1} H)^{-1} H^\dagger).$$

The VWF finds the linear estimate of $\mathbf{s} = H\mathbf{x}$ that minimizes the mean squared error $E(|\hat{\mathbf{s}} - \mathbf{s}|^2)$. Consequently, the mean squared error in the VWF is less than that in the BLUE.

Exercise 3: Assume that $E(\mathbf{x}\mathbf{x}^\dagger) = \sigma^2 I$. Show that the mean squared error for the VWF estimate is

$$E(|\hat{\mathbf{s}} - \mathbf{s}|^2) = \text{trace}\,(H(H^\dagger Q^{-1} H + \sigma^{-2} I)^{-1} H^\dagger).$$

The functional Wiener filter The Wiener filter is often presented in the context of random functions of, say, time. In this model signal is $s(t)$ and noise is $q(t)$, where these functions of time are viewed as random functions (stochastic processes). The data is taken to be $z(t)$, a function of t, so that the matrices $UU^\dagger$ and $WW^\dagger$ are now *infinite matrices*; the discrete index $j = 1, ..., J$ is now replaced by the continuous index variable t. Instead of the finite family $\{\mathbf{u}^n, n = 1..., N\}$, we now have an infinite family of functions $u(t)$ in $\mathcal{U}$. The entries of $UU^\dagger$ are essentially the average values of the products $u(t_1)\overline{u(t_2)}$ over all the members of $\mathcal{U}$. It is often assumed that this average of products is a function not of t_1 and t_2 separately, but only of their difference $t_1 - t_2$; this is called *stationarity*. So, $aver\{u(t_1)\overline{u(t_2)}\} = r_s(t_1 - t_2)$ comes from a function $r_s(\tau)$ of a

single variable. The Fourier transform of $r_s(\tau)$ is $R_s(\omega)$, the signal power spectrum. The matrix $UU^\dagger$ is then an infinite Toeplitz matrix, constant on each diagonal. The Wiener filtering can actually be achieved by taking Fourier transforms and multiplying and dividing by power spectra, instead of inverting infinite matrices. It is also common to discretize the time variable and to consider the Wiener filter operating on infinite sequences, as we see in the next chapter.

Chapter 29

Wiener Filter Approximation

As we saw in the previous chapter, when the data is a finite vector composed of signal plus noise the vector Wiener filter can be used to estimate the signal component, provided we know something about the possible signals and possible noises. In theoretical discussion of filtering signal from signal plus noise it is traditional to assume that both components are doubly infinite sequences of random variables. In this case the Wiener filter is a convolution filter that operates on the input signal plus noise sequence to produce the output estimate of the signal-only sequence. The derivation of the Wiener filter is in terms of the autocorrelation sequences of the two components, as well as their respective power spectra.

Suppose now that the discrete stationary random process to be filtered is the doubly infinite sequence $\{z_n = s_n + q_n\}_{n=-\infty}^{\infty}$, where $\{s_n\}$ is the signal component with autocorrelation function $r_s(k) = E(s_{n+k}\overline{s_n})$ and power spectrum $R_s(\omega)$ defined for ω in the interval $[-\pi, \pi]$, $\{q_n\}$ is the noise component with autocorrelation function $r_q(k)$ and power spectrum $R_q(\omega)$ defined for ω in $[-\pi, \pi]$. We assume that for each n the random variables s_n and q_n have mean zero and that the signal and noise are independent of one another. Then the autocorrelation function for the signal plus noise sequence $\{z_n\}$ is

$$r_z(n) = r_s(n) + r_q(n)$$

for all n and

$$R_z(\omega) = R_s(\omega) + R_q(\omega).$$

is the signal plus noise power spectrum.

Let $h = \{h_k\}_{k=-\infty}^{\infty}$ be a linear filter with *transfer function*

$$H(\omega) = \sum_{k=-\infty}^{\infty} h_k e^{ik\omega},$$

for ω in $[-\pi, \pi]$. Given the sequence $\{z_n\}$ as input to this filter, the output is the sequence

$$y_n = \sum_{k=-\infty}^{\infty} h_k z_{n-k}. \tag{29.1}$$

The goal of Wiener filtering is to select the filter h so that the output sequence y_n approximates the signal s_n sequence as well as possible. Specifically, we seek h so as to minimize the expected squared error, $E(|y_n - s_n|^2)$, which, because of stationarity, is independent of n. We have

$$E(|y_n|^2) = \sum_{k=-\infty}^{\infty} h_k (\sum_{j=-\infty}^{\infty} \overline{h_j}(r_s(j-k) + r_q(j-k)))$$

$$= \sum_{k=-\infty}^{\infty} h_k (r_z * \overline{h})_k$$

which, by the Parseval equation, equals

$$\frac{1}{2\pi} \int H(\omega) R_z(\omega) \overline{H(\omega)} d\omega = \frac{1}{2\pi} \int |H(\omega)|^2 R_z(\omega) d\omega.$$

Similarly,

$$E(s_n \overline{y_n}) = \sum_{j=-\infty}^{\infty} \overline{h_j} r_s(j)$$

which equals

$$\frac{1}{2\pi} \int R_s(\omega) \overline{H(\omega)} d\omega,$$

and

$$E(|s_n|^2) = \frac{1}{2\pi} \int R_s(\omega) d\omega.$$

Therefore,

$$E(|y_n - s_n|^2) = \frac{1}{2\pi} \int |H(\omega)|^2 R_z(\omega) d\omega - \frac{1}{2\pi} \int R_s(\omega) \overline{H(\omega)} d\omega$$

$$-\frac{1}{2\pi} \int R_s(\omega) H(\omega) d\omega + \frac{1}{2\pi} \int R_s(\omega) d\omega.$$

As we shall see shortly, minimizing $E(|y_n - s_n|^2)$ with respect to the function $H(\omega)$ leads to the equation

$$R_z(\omega) H(\omega) = R_s(\omega),$$

so that the transfer function of the optimal filter is

$$H(\omega) = R_s(\omega) / R_z(\omega).$$

The *Wiener filter* is then the sequence $\{h_k\}$ of the Fourier coefficients of this function $H(\omega)$.

To prove that this choice of $H(\omega)$ minimizes $E(|y_n - s_n|^2)$ we note that

$$|H(\omega)|^2 R_z(\omega) - R_s(\omega)\overline{H(\omega)} - R_s(\omega)H(\omega) + R_s(\omega)$$

$$= |H(\omega) - R_s(\omega)/R_z(\omega)|^2 - R_s(\omega) + R_s(\omega)^2/R_z(\omega).$$

Only the first term involves the function $H(\omega)$.

Since $H(\omega)$ is a nonnegative function of ω, therefore real-valued, its Fourier coefficients h_k will be *conjugate symmetric*, that is, $h_{-k} = \overline{h_k}$. This poses a problem when the random process z_n is a discrete time series, with z_n denoting the measurement recorded at time n. From the equation (29.1) we see that to produce the output y_n corresponding to time n we need the input for every time, past and future. To remedy this we can obtain the best causal approximation of the Wiener filter h.

A filter $g = \{g_k\}_{k=-\infty}^{\infty}$ is said to be *causal* if $g_k = 0$ for $k < 0$; this means that given the input sequence $\{z_n\}$, the output

$$w_n = \sum\nolimits_{k=-\infty}^{\infty} g_k z_{n-k} = \sum\nolimits_{k=0}^{\infty} g_k z_{n-k}$$

requires only values of z_m up to $m = n$. To obtain the causal filter g that best approximates the Wiener filter, we find the coeffcients g_k that minimize the quantity

$$\int_{-\pi}^{\pi} |H(\omega) - \sum\nolimits_{k=0}^{+\infty} g_k e^{ik\omega}|^2 R_z(\omega) d\omega. \tag{29.2}$$

The orthogonality principle tells us that the optimal coefficients must satisfy the equations

$$r_s(m) = \sum\nolimits_{k=0}^{+\infty} g_k r_z(m-k), \tag{29.3}$$

for all m. These are the *Wiener-Hopf equations* [119].

Even having a causal filter does not completely solve the problem, since we would have to record and store the infinite past. Instead, we can decide to use a filter $f = \{f_k\}_{k=-\infty}^{\infty}$ for which $f_k = 0$ unless $-K \leq k \leq L$ for some positive integers K and L. This means we must store L values and wait until time $n + K$ to obtain the output for time n. Such a linear filter is a *finite memory, finite delay* filter, also called a *finite impulse response* filter.

To obtain the filter f of this type that best approximates the Wiener filter, we find the coefficients f_k that minimize the quantity

$$\int_{-\pi}^{\pi} |H(\omega) - \sum\nolimits_{k=-K}^{L} f_k e^{ik\omega}|^2 R_z(\omega) d\omega. \tag{29.4}$$

The orthogonality principle tells us that the optimal coefficients must satisfy the equations

$$r_s(m) = \sum_{k=-K}^{L} f_k r_z(m-k), \tag{29.5}$$

for $-K \leq m \leq L$.

In [39] it was pointed out that the linear equations that arise in Wiener filter approximation also occur in image reconstruction from projections, with the image to be reconstructed playing the role of the power spectrum to be approximated. The methods of Wiener filter approximation were then used to derive linear and nonlinear image reconstruction procedures.

Chapter 30

Adaptive Wiener Filters

Once again, we consider a stationary random process $z_n = s_n + v_n$ with autocorrelation function $E(z_n\overline{z_{n-m}}) = r_z(m) = r_s(m) + r_v(m)$. The finite causal Wiener filter (FCWF) $\mathbf{f} = (f_0, f_1, ..., f_L)^T$ is convolved with $\{z_n\}$ to produce an estimate of s_n given by

$$\hat{s}_n = \sum_{k=0}^{L} f_k z_{n-k}.$$

With $\mathbf{y}_n^\dagger = (z_n, z_{n-1}, ..., z_{n-L})$ we can write $\hat{s}_n = \mathbf{y}_n^\dagger \mathbf{f}$. The FCWF $\mathbf{f}$ minimizes the expected squared error

$$J(\mathbf{f}) = E(|s_n - \hat{s}_n|^2)$$

and is obtained as the solution of the equations

$$r_s(m) = \sum\nolimits_{k=0}^{L} f_k r_z(m-k),$$

for $0 \leq m \leq L$. Therefore, to use the FCWF we need the values $r_s(m)$ and $r_z(m-k)$ for m and k in the set $\{0, 1, ..., L\}$. When these autocorrelation values are not known we can use adaptive methods to approximate the FCWF.

An adaptive least mean square approach: We assume now that we have $z_0, z_1, ..., z_N$ and $p_0, p_1, ..., p_N$, where p_n is a prior estimate of s_n, but that we do not know the correlation functions r_z and r_s.

The gradient of the function $J(\mathbf{f})$ is

$$\nabla J(\mathbf{f}) = R_{zz}\mathbf{f} - \mathbf{r}_s,$$

where R_{zz} is the square matrix with entries $r_z(m-n)$ and $\mathbf{r}_s$ is the vector with entries $r_s(m)$. An iterative gradient descent method for solving the system of equations $R_{zz}\mathbf{f} = \mathbf{r}_s$ is

$$\mathbf{f}_\tau = \mathbf{f}_{\tau-1} - \mu_\tau \nabla J(\mathbf{f}_{\tau-1}),$$

for some step-size parameters $\mu_\tau > 0$.

The adaptive *least mean square* (LMS) approach [45] replaces the gradient of $J(\mathbf{f})$ with an approximation of the gradient of the function $G(\mathbf{f}) = |s_n - \hat{s}_n|^2$, which is $-2(s_n - \hat{s}_n)\mathbf{y}_n$. Since we do not know s_n we replace that term with the estimate p_n. The iterative step of the LMS method is

$$\mathbf{f}_\tau = \mathbf{f}_{\tau-1} + \mu_\tau(p_\tau - \mathbf{y}_\tau^\dagger \mathbf{f}_{\tau-1})\mathbf{y}_\tau, \tag{30.1}$$

for $L \leq \tau \leq N$. Notice that it is the approximate gradient of the function $|s_\tau - \hat{s}_\tau|^2$ that is used at this step, in order to involve all the data $z_0, ..., z_N$ as we iterate from $\tau = L$ to $\tau = N$. We illustrate the use of this method in adaptive interference cancellation.

Adaptive interference cancellation: Adaptive interference cancellation (AIC) [146] is used to suppress a dominant noise component v_n in the discrete sequence $z_n = s_n + v_n$. It is assumed that we have available a good estimate q_n of v_n. The main idea is to switch the roles of signal and noise in the adaptive LMS method and design a filter to estimate v_n. Once we have that estimate, we subtract it from z_n to get our estimate of s_n.

In the role of z_n we use

$$q_n = v_n + \epsilon_n,$$

where ϵ_n denotes a low level error component. In the role of p_n we take z_n, which is approximately v_n, since the signal s_n is much lower than the noise v_n. Then $\mathbf{y}_n^\dagger = (q_n, q_{n-1}, ..., q_{n-L})$. The iterative step used to find the filter $\mathbf{f}$ is then

$$\mathbf{f}_\tau = \mathbf{f}_{\tau-1} + \mu_\tau(z_\tau - \mathbf{y}_\tau^\dagger \mathbf{f}_{\tau-1})\mathbf{y}_\tau,$$

for $L \leq \tau \leq N$. When the iterative process has converged to $\mathbf{f}$ we take as our estimate of s_n

$$\hat{s}_n = z_n - \sum_{k=0}^{L} f_k q_{n-k}.$$

It has been suggested that this procedure be used in computerized tomography to correct artifacts due to patient motion [69].

Recursive least squares: An alternative to the LMS method is to find the least squares solution of the system of $N - L + 1$ linear equations

$$p_n = \sum_{k=0}^{L} f_k z_{n-k},$$

for $L \leq n \leq N$. The *recursive least squares* (RLS) method is a recursive approach to solving this system.

For $L \leq \tau \leq N$ let Z_τ be the matrix whose rows are $\mathbf{y}_n^\dagger$ for $n = L, ..., \tau$, $\mathbf{p}_\tau^T = (p_L, p_{L+1}, ..., p_\tau)$ and $Q_\tau = Z_\tau^\dagger Z_\tau$. The least squares solution we seek is

$$\mathbf{f} = Q_N^{-1} Z_N^\dagger \mathbf{p}_N.$$

Exercise 1: Show that $Q_\tau = Q_{\tau-1} + \mathbf{y}_\tau \mathbf{y}_\tau^\dagger$, for $L < \tau \leq N$.

Exercise 2: Use the matrix inversion identity in equation (28.1) to write Q_τ^{-1} in terms of $Q_{\tau-1}^{-1}$.

Exercise 3: Using the previous exercise, show that the desired least squares solution $\mathbf{f}$ is $\mathbf{f} = \mathbf{f}_N$, where, for $L \leq \tau \leq N$ we let

$$\mathbf{f}_\tau = \mathbf{f}_{\tau-1} + (\frac{p_\tau - \mathbf{y}_\tau^\dagger \mathbf{f}_{\tau-1}}{1 + \mathbf{y}_\tau^\dagger Q_{\tau-1}^{-1} \mathbf{y}_\tau}) Q_{\tau-1}^{-1} \mathbf{y}_\tau.$$

Comparing this iterative step with that given by equation (30.1) we see that the former gives an explicit value for μ_τ and uses $Q_{\tau-1}^{-1}\mathbf{y}_\tau$ instead of $\mathbf{y}_\tau$ as the direction vector for the iterative step. The RMS iteration produces a more accurate estimate of the FCWF than does the LMS method, but requires more computation.

Chapter 31

Entropy Maximization

The problem of estimating the nonnegative function $R(\omega)$, for $|\omega| \leq \pi$, from the finitely many Fourier transform values

$$r(n) = \int_{-\pi}^{\pi} R(\omega) \exp(-in\omega) d\omega/2\pi,\ n = -N, ..., N$$

is an *underdetermined problem*, meaning that the data alone is insufficient to determine a unique answer. In such situations we must select one solution out of the infinitely many that are mathematically possible. The obvious questions we need to answer are: What criteria do we use in this selection? How do we find algorithms that meet our chosen criteria? In this chapter we look at some of the answers people have offered and at one particular algorithm, Burg's *maximum entropy* method (MEM) [19], [20].

These values $r(n)$ are autocorrelation function values associated with a random process having $R(\omega)$ for its power spectrum. In many applications, such as seismic remote sensing, these autocorrelation values are estimates obtained from relatively few samples of the underlying random process, so that N is not large. The DFT estimate,

$$R_{DFT}(\omega) = \sum_{n=-N}^{N} r(n) \exp(in\omega),$$

is real-valued and consistent with the data, but is not necessarily nonnegative. For small values of N the DFT may not be sufficiently resolving to be useful. This suggests that one criterion we can use to perform our selection process is to require that the method provide better resolution than the DFT for relatively small values of N, when reconstructing power spectra that consist mainly of delta functions.

A brief side trip to philosophy:

Generally speaking, we would expect to do a better job of estimating a function from data pertaining to that function if we also possess additional prior information about the function to be estimated and are able to employ estimation techniques that make use of that additional information. There is the danger, however, that we may end up with an answer that is influenced more by our prior guesses than by the actual measured data. Striking a balance between including prior knowledge and letting the data speak for itself is a noble goal; how to achieve that is the question. At this stage, we begin to suspect that the problem is as much philosophical as it is mathematical.

We are essentially looking for principles of induction that enable us to extrapolate from what we have measured to what we have not. Unwilling to turn the problem over entirely to the philosophers, a number of mathematicians and physicists have sought mathematical solutions to this inference problem, framed in terms of what the *most likely* answer is, or which answer involves the smallest amount of additional prior information. This is not, of course, a new issue; it has been argued for centuries with regard to the use of what we now call Bayesian statistics; *objective* Bayesians allow the use of prior information, but only if it is the right prior information. The interested reader should consult the books [134] and [135], containing papers by Ed Jaynes, Roy Frieden and others originally presented at workshops on this topic held in the early 1980's.

The maximum entropy method is a general approach to such problems that includes Burg's algorithm as a particular case. It is argued that by maximizing entropy we are, in some sense, being maximally noncommittal about what we do not know and thereby introducing a minimum of prior knowledge (some would say prior guesswork) into the solution. In the case of Burg's MEM a somewhat more mathematical argument is available.

Let $\{x_n\}_{n=-\infty}^{\infty}$ be a stationary random process with autocorrelation sequence $r(m)$ and power spectrum $R(\omega)$, $|\omega| \leq \pi$. The prediction problem is the following: suppose we have measured the values of the process prior to time n and we want to predict the value of the process at time n. On average, how much error do we expect to make in predicting x_n from knowledge of the infinite past? The answer, according to Szegö's theorem [90], is

$$\exp[\int_{-\pi}^{\pi} \log R(\omega) d\omega];$$

the integral

$$\int_{-\pi}^{\pi} \log R(\omega) d\omega$$

is the *Burg entropy* of the random process [124]. Processes that are very predictable have low entropy, while those that are quite unpredictable, or,

like white noise, completely unpredictable, have high entropy; to make entropies comparable we assume a fixed value of $r(0)$. Given the data $r(n)$, $|n| \leq N$, Burg's method selects that power spectrum consistent with these autocorrelation values that corresponds to the most unpredictable random process.

Other similar procedures are also based on selection through optimization. We have seen the minimum norm approach to finding a solution to an underdetermined system of linear equations, the minimum expected squared error approach in statistical filtering and later we shall see the maximum likelihood method used in detection. We must keep in mind that, however comforting it may be to know that we are on solid philosophical ground (if such exists) in choosing our selection criteria, if the method does not work well, we must use something else. As we shall see, the MEM, like every other reasonable method, works well sometimes and not so well other times. There is certainly philosophical precedent for considering the consequences of our choices, as Blaise Pascal's famous wager about the existence of God nicely illustrates. As an attentive reader of the books [134] and [135] will surely note, there is a certain theological tone to some of the arguments offered in support of entropy maximization. One group of authors (reference omitted) went so far as to declare that entropy maximization was what one did if one cared what happened to one's data.

The objective of Burg's MEM for estimating a power spectrum is to seek better resolution by combining nonnegativity and data-consistency in a single closed-form estimate. The MEM is remarkable in that it is the only closed-form (that is, noniterative) estimation method that is guaranteed to produce an estimate that is both nonnegative and consistent with the autocorrelation samples. Later we shall consider a more general method, the inverse PDFT (IPDFT), that is both data-consistent and positive in most cases.

Properties of the sequence $\{r(n)\}$:

We begin our discussion with a look at important properties of the sequence $\{r(n)\}$. Because $R(\omega) \geq 0$, the values $r(n)$ are often called *autocorrelation values*.

Since $R(\omega) \geq 0$, it follows immediately that $r(0) \geq 0$. In addition, $r(0) \geq |r(n)|$ for all n:

$$|r(n)| = |\int_{-\pi}^{\pi} R(\omega) \exp(-in\omega) d\omega/2\pi|$$

$$\leq \int_{-\pi}^{\pi} R(\omega) |\exp(-in\omega)| d\omega/2\pi = r(0).$$

In fact, if $r(0) = |r(n)| > 0$ for some $n > 0$, then R is a sum of at most $n + 1$ delta functions with nonnegative amplitudes. To see this, suppose

that $r(n) = |r(n)| \exp(i\theta) = r(0) \exp(i\theta)$. Then

$$\int_{-\pi}^{\pi} R(\omega)|1 - \exp(i(\theta + n\omega))|^2 d\omega/2\pi$$

$$= \int_{-\pi}^{\pi} R(\omega)(1 - \exp(i(\theta + n\omega))(1 - \exp(-i(\theta + n\omega))d\omega/2\pi$$

$$= \int_{-\pi}^{\pi} R(\omega)[2 - \exp(i(\theta + n\omega)) - \exp(-i(\theta + n\omega))]d\omega/2\pi$$

$$= 2r(0) - \exp(i\theta)\overline{r(n)} - \exp(-i\theta)r(n) = 2r(0) - r(0) - r(0) = 0.$$

Therefore, $R(\omega) > 0$ only at the values of ω where $|1-\exp(i(\theta+n\omega))|^2 = 0$; that is, only at $\omega = n^{-1}(2\pi k - \theta)$ for some integer k. Since $|\omega| \leq \pi$ there are only finitely many such k.

In discussing the Burg MEM estimate we shall need to refer to the concept of *minimum phase* vectors. We consider that briefly now.

Minimum phase vectors:

We say that the finite column vector with complex entries $(a_0, a_1, ..., a_N)^T$ is a *minimum phase* vector if the complex polynomial

$$A(z) = a_0 + a_1 z + ... + a_N z^N$$

has the property that $A(z) = 0$ implies that $|z| > 1$; that is, all roots of $A(z)$ are outside the unit circle. Consequently, the function $B(z)$ given by $B(z) = 1/A(z)$ is analytic in a disk centered at the origin and including the unit circle. Therefore, we can write

$$B(z) = b_0 + b_1 z + b_2 z^2 + ...$$

and taking $z = \exp(i\omega)$, we get

$$B(\exp(i\omega)) = b_0 + b_1 \exp(i\omega) + b_2 \exp(2i\omega) + ...$$

The point here is that $B(\exp(i\omega))$ is a one-sided trigonometric series, with only terms corresponding to $\exp(in\omega)$ for nonnegative n.

Burg's MEM:

The approach is to estimate $R(\omega)$ by the function $S(\omega) > 0$ that maximizes the so-called Burg entropy, $\int_{-\pi}^{\pi} \log S(\theta) d\theta$, subject to the data constraints.

The Euler-Lagrange equation from the calculus of variations allows us to conclude that $S(\omega)$ has the form

$$S(\omega) = 1/H(\omega)$$

for

$$H(\omega) = \sum_{n=-N}^{N} h_n e^{in\omega} > 0.$$

From the Fejér-Riesz theorem **??** we know that $H(\omega) = |A(e^{i\omega})|^2$ for minimum phase $A(z)$ as above. As we now show, the coefficients a_n satisfy a system of linear equations formed using the data $r(n)$.

Given the data $r(n), |n| \leq N$, we form the *autocorrelation matrix* R with entries $R_{mn} = r(m-n)$, for $-N \leq m, n \leq N$. Let δ be the column vector $\delta = (1, 0, ..., 0)^T$. Let $\mathbf{a} = (a_0, a_1, ..., a_N)^T$ be the solution of the system $R\mathbf{a} = \delta$. Then Burg's MEM estimate is the function $S(\omega) = R_{MEM}(\omega)$ given by

$$R_{MEM}(\omega) = a_0/|A(\exp(i\omega))|^2, |\omega| \leq \pi.$$

Once we show that $a_0 \geq 0$ then it will be obvious that $R_{MEM}(\omega) \geq 0$. We also must show that R_{MEM} is data-consistent; that is,

$$r(n) = \int_{-\pi}^{\pi} R_{MEM}(\omega) \exp(-in\omega) d\omega/2\pi =, \; n = -N, ..., N.$$

Let us write $R_{MEM}(\omega)$ as a Fourier series; that is

$$R_{MEM}(\omega) = \sum_{n=-\infty}^{+\infty} q(n) \exp(in\omega), \; |\omega| \leq \pi.$$

From the form of $R_{MEM}(\omega)$ we have

$$R_{MEM}(\omega)\overline{A(\exp(i\omega))} = a_0 B(\exp(i\omega)).$$

Suppose, as we shall shortly show, that $A(z)$ has all its roots outside the unit circle and so $B(\exp(i\omega))$ is a one-sided trigonometric series, with only terms corresponding to $\exp(in\omega)$ for nonnegative n. Then, multiplying on the left side of the equation above and equating coefficients corresponding to $n = 0, -1, -2, ...$, we find that, provided $q(n) = r(n)$, for $|n| \leq N$, we must have $R\mathbf{a} = \delta$. Notice that these are precisely the same equations we solve in calculating the coefficients of an AR process. For that reason the MEM is sometimes called an autoregressive method for spectral estimation.

We now show that if $R\mathbf{a} = \delta$ then $A(z)$ has all its roots outside the unit circle. Let $r\exp(i\theta)$ be a root of $A(z)$. Then write

$$A(z) = (z - r\exp(i\theta))C(z),$$

where

$$C(z) = c_0 + c_1 z + c_2 z^2 + ... + c_{N-1} z^{N-1}.$$

Then the vector $\mathbf{a} = (a_0, a_1, ..., a_N)^T$ can be written as $\mathbf{a} = -r\exp(i\theta)\mathbf{c}+\mathbf{d}$, where $\mathbf{c} = (c_0, c_1, ..., c_{N-1}, 0)^T$ and $\mathbf{d} = (0, c_0, c_1, ..., c_{N-1})^T$. So $\delta = R\mathbf{a} = -r\exp(i\theta)R\mathbf{c} + R\mathbf{d}$ and

$$0 = \mathbf{d}^\dagger\delta = -r\exp(i\theta)\mathbf{d}^\dagger R\mathbf{c} + \mathbf{d}^\dagger R\mathbf{d},$$

so that

$$r\exp(i\theta)\mathbf{d}^\dagger R\mathbf{c} = \mathbf{d}^\dagger R\mathbf{d}.$$

From the Cauchy inequality we know that

$$|\mathbf{d}^\dagger R\mathbf{c}|^2 \leq (\mathbf{d}^\dagger R\mathbf{d})(\mathbf{c}^\dagger R\mathbf{c}) = (\mathbf{d}^\dagger R\mathbf{d})^2, \tag{31.1}$$

where the last equality comes from the special form of the matrix R and the similarity between $\mathbf{c}$ and $\mathbf{d}$.

With

$$D(\omega) = c_0e^{i\omega} + c_1e^{2i\omega}... + c_{N-1}e^{iN\omega}$$

and

$$C(\omega) = c_0 + c_1e^{i\omega} + ... + c_{N-1}e^{i(N-1)\omega},$$

we can easily show that

$$\mathbf{d}^\dagger R\mathbf{d} = \mathbf{c}^\dagger R\mathbf{c} = \frac{1}{2\pi}\int_{-\pi}^{\pi} R(\omega)|D(\omega)|^2 d\omega$$

and

$$\mathbf{d}^\dagger R\mathbf{c} = \frac{1}{2\pi}\int_{-\pi}^{\pi} R(\omega)\overline{D(\omega)}C(\omega)d\omega.$$

If there is equality in the Cauchy inequality (31.1) then $r = 1$ and we would have

$$\exp(i\theta)\frac{1}{2\pi}\int_{-\pi}^{\pi} R(\omega)\overline{D(\omega)}C(\omega)d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi} R(\omega)|D(\omega)|^2 d\omega.$$

From the Cauchy inequality for integrals, we can conclude that

$$\exp(i\theta)\overline{D(\omega)}C(\omega) = |D(\omega)|^2$$

for all ω for which $R(\omega) > 0$. But

$$\exp(i\omega)C(\omega) = D(\omega).$$

Therefore we cannot have $r = 1$ unless $R(\omega) = \delta(\omega - \theta)$. In all other cases we have

$$|\mathbf{d}^\dagger R\mathbf{c}|^2 < |r|^2|\mathbf{d}^\dagger R\mathbf{c}|^2,$$

from which we conclude that $|r| > 1$.

Solving $R\mathbf{a} = \delta$ using Levinson's algorithm: Because the matrix R is Toeplitz (constant on diagonals) and positive definite, there is a fast algorithm for solving $R\mathbf{a} = \delta$ for $\mathbf{a}$. Instead of a single R we let R_M be the matrix defined for $M = 0, 1, ..., N$ by

$$R_M = \begin{bmatrix} r(0) & r(-1) & ... & r(-M) \\ r(1) & r(0) & ... & r(-M+1) \\ . & & & \\ . & & & \\ . & & & \\ r(M) & r(M-1) & ... & r(0) \end{bmatrix}$$

so that $R = R_N$. We also let δ^M be the $M+1$-dimensional column vector $\delta^M = (1, 0, ..., 0)^T$. We want to find the column vector $\mathbf{a}^M = (a_0^M, a_1^M, ..., a_M^M)^T$ that satisfies the equation $R_M\mathbf{a}^M = \delta^M$. The point of Levinson's algorithm is to calculate $\mathbf{a}^{M+1}$ quickly from $\mathbf{a}^M$.

For fixed M find constants α and β so that

$$\delta^M = R_M\left\{\alpha \begin{bmatrix} a_0^{M-1} \\ a_1^{M-1} \\ . \\ . \\ . \\ a_{M-1}^{M-1} \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ \overline{a}_{M-1}^{M-1} \\ \overline{a}_{M-2}^{M-1} \\ . \\ . \\ . \\ \overline{a}_0^{M-1} \end{bmatrix}\right\}$$

$$= \left\{\alpha \begin{bmatrix} 1 \\ 0 \\ . \\ . \\ . \\ 0 \\ \gamma^M \end{bmatrix} + \beta \begin{bmatrix} \overline{\gamma}^M \\ 0 \\ . \\ . \\ . \\ 0 \\ 1 \end{bmatrix}\right\},$$

where

$$\gamma^M = r(M)a_0^{M-1} + r(M-1)a_1^{M-1} + ... + r(1)a_{M-1}^{M-1}.$$

We then have

$$\alpha + \beta\overline{\gamma^M} = 1,\ \alpha\gamma^M + \beta = 0$$

or

$$\beta = -\alpha\gamma^M,\ \alpha - \alpha|\gamma^M|^2 = 1,$$

so

$$\alpha = 1/(1 - |\gamma^M|^2),\ \beta = -\gamma^M/(1 - |\gamma^M|^2).$$

Therefore, the algorithm begins with $M = 0$, $R_0 = [r(0)], a_0^0 = r(0)^{-1}$. At each step calculate the γ^M, solve for α and β and form the next $\mathbf{a}^M$.

The MEM resolves better than the DFT when the true power spectrum being reconstructed is a sum of delta functions plus a flat background. When the background itself is not flat performance of the MEM degrades rapidly; the MEM tends to interpret any non-flat background in terms of additional delta functions. In the next chapter we consider an extension of the MEM, called the indirect PDFT (IPDFT), that corrects this flaw.

Why Burg's MEM and the IPDFT are able to resolve closely spaced sinusoidal components better than the DFT is best answered by studying the eigenvalues and eigenvectors of the matrix R; we turn to this topic in a later chapter.

A sufficient condition for positive-definiteness:

If the function

$$R(\omega) = \sum_{n=-\infty}^{\infty} r(n)e^{in\omega}$$

is nonnegative on the interval $[-\pi, \pi]$ then the matrices R_M above are nonnegative-definite for every M. Theorems by Herglotz and by Bochner go in the reverse direction [3]. Katznelson [100] gives the following result.

Theorem 31.1 *Let $\{f(n)\}_{n=-\infty}^{\infty}$ be a sequence of nonnegative real numbers converging to zero, with $f(-n) = f(n)$ for each n. If, for each $n > 0$, we have*

$$(f(n-1) - f(n)) - (f(n) - f(n+1)) > 0,$$

then there is a nonnegative function $R(\omega)$ on the interval $[-\pi, \pi]$ with $f(n) = r(n)$ for each n.

Chapter 32

Eigenvector Methods

Prony's method showed that information about the signal can sometimes be obtained from the roots of certain polynomials formed from the data. Eigenvector methods assume the data is correlation values and involve polynomials formed from the eigenvectors of the correlation matrix. Schmidt's *multiple signal classification* (MUSIC) algorithm is one such method [129]. A related technique used in direction-of-arrival array processing is the *estimation of signal parameters by rotational invariance techniques* (ESPRIT) of Paulraj, Roy and Kailath [120].

We suppose now that the function $f(t)$ being measured is signal plus noise, with the form

$$f(t) = \sum_{j=1}^{J} A_j e^{i\theta_j} e^{i\omega_j t} + n(t) = s(t) + n(t),$$

where the phases θ_j are random variables, independent and uniformly distributed in the interval $[0, 2\pi)$ and $n(t)$ denotes the random complex stationary noise component. Assume that $E(n(t)) = 0$ for all t and that the noise is independent of the signal components. We want to estimate J, the number of sinusoidal components, their magnitudes $|A_j|$ and their frequencies ω_j.

The autocorrelation function associated with $s(t)$ is

$$r_s(\tau) = \sum_{j=1}^{J} |A_j|^2 e^{-i\omega_j \tau}$$

and the signal power spectrum is the Fourier transform of $r_s(\tau)$,

$$R_s(\omega) = \sum_{j=1}^{J} |A_j|^2 \delta(\omega - \omega_j).$$

The noise autocorrelation is denoted $r_n(\tau)$ and the noise power spectrum is denoted $R_n(\omega)$. For the remainder of this section we shall assume that the noise is *white noise*, that is, $R_n(\omega)$ is constant and $r_n(\tau) = 0$ for $\tau \neq 0$.

We collect samples of the function $f(t)$ and use them to estimate some of the values of $r_s(\tau)$. From these values of $r_s(\tau)$ we estimate $R_s(\omega)$, primarily looking for the locations ω_j at which there are delta functions.

We assume that the samples of $f(t)$ have been taken over an interval of time sufficiently long to take advantage of the independent nature of the phase angles θ_j and the noise. This means that when we estimate the $r_s(\tau)$ from products of the form $f(t+\tau)\overline{f(t)}$ the cross terms between one signal component and another, as well as between a signal component and the noise, are nearly zero, due to destructive interference coming fro the random phases.

Suppose now that we have the values $r_f(m)$ for $m = -(M-1), ..., M-1$, where $M > J$, $r_f(m) = r_s(m)$ for $m \neq 0$ and $r_f(0) = r_s(0) + \sigma^2$, for σ^2 the variance (or *power*) of the noise. We form the M by M autocorrelation matrix R with entries $R_{m,k} = r_f(m-k)$.

Exercise 1: Show that the matrix R has the following form:

$$R = \sum_{j=1}^{J} |A_j|^2 \mathbf{e}_j \mathbf{e}_j^\dagger + \sigma^2 I,$$

where $\mathbf{e}_j$ is the column vector with entries $e^{-i\omega_j m}$, for $m = -(M-1), ..., M-1$.

Let $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_M > 0$ be the eigenvalues of R and let $\mathbf{u}^m$ be a norm-one eigenvector associated with λ_m.

Exercise 2: Show that $\lambda_m = \sigma^2$ for $m = J+1, ..., M$, while $\lambda_m > \sigma^2$ for $m = 1, ..., J$. Hint: since $M > J$ the $M - J$ orthogonal eigenvectors $\mathbf{u}^m$ corresponding to λ_m for $m = J+1, ..., M$ will be orthogonal to each of the $\mathbf{e}_j$. Then consider the quadratic forms $(\mathbf{u}^m)^\dagger R \mathbf{u}^m$.

By calculating the eigenvalues of R and noting how many of them are greater than the smallest one we find J. Now we seek the ω_j.

For each ω let $\mathbf{e}_\omega$ have the entries $e^{-i\omega m}$ and form the function

$$T(\omega) = \sum_{m=J+1}^{M} |\mathbf{e}_\omega^\dagger \mathbf{u}^m|^2.$$

This function $T(\omega)$ will have zeros at precisely the values $\omega = \omega_j$, for $j = 1, ..., J$. Once we have determined J and the ω_j we estimate the magnitudes $|A_j|$ using Fourier transform estimation techniques already discussed. This is basically Schmidt's MUSIC method.

We have made several assumptions here that may not hold in practice and we must modify this eigenvector approach somewhat. First, the time over which we are able to measure the function $f(t)$ may not be long enough

to give good estimates of the $r_f(\tau)$. In that case we may work directly with the samples of $f(t)$. Second, the smallest eigenvalues will not be exactly equal to σ^2 and some will be larger than others. If the ω_j are not well separated, or if some of the $|A_j|$ are quite small, it may be hard to tell what the value of J is. Third, we often have measurements of $f(t)$ that have errors other than those due to background noise; inexpensive sensors can introduce their own random phases that can complicate the estimation process. Finally, the noise may not be white, so that the estimated $r_f(\tau)$ will not equal $r_s(\tau)$ for $\tau \neq 0$, as above. If we know the noise power spectrum or have a decent idea what it is we can perform a *prewhitening* to R, which will then return us to the case considered above, although this can be a tricky procedure.

Chapter 33

Signal Detection and Estimation

In this chapter we consider the problem of deciding whether or not a particular signal is present in the measured data; this is the *detection* problem. The underlying framework for the detection problem is optimal estimation and statistical hypothesis testing [79].

The general model of signal in additive noise:

The basic model used in detection is that of a signal in additive noise. The complex data vector is $\mathbf{x} = (x_1, x_2, ..., x_N)^T$. We assume that there are two possibilities:

Case 1: noise only

$$x_n = z_n,\ n = 1, ..., N,$$

or

Case 2: signal in noise

$$x_n = \gamma s_n + z_n,$$

where $\mathbf{z} = (z_1, z_2, ..., z_N)^T$ is a complex vector whose entries z_n are values of random variables that we call *noise*, about which we have only statistical information (that is to say, information about the average behavior), $\mathbf{s} = (s_1, s_2, ..., s_N)^T$ is a complex signal vector that we may known exactly, or at least for which we have a specific parametric model and γ is a scalar that may be viewed either as deterministic or random (but unknown, in either case). Unless otherwise stated, we shall assume that γ is deterministic.

The *detection problem* is to decide which case we are in, based on some calculation performed on the data $\mathbf{x}$. Since Case 1 can be viewed as a special case of Case 2 in which the value of γ is zero, the detection problem is closely related to the problem of estimating γ, which we discussed in the chapter dealing with the best linear unbiased estimator, the BLUE.

We shall assume throughout that the entries of $\mathbf{z}$ correspond to random variables with means equal to zero. What the variances are and whether or not these random variables are mutually correlated will be discussed below. In all cases we shall assume that this information has been determined previously and is available to us in the form of the correlation matrix $Q = E(\mathbf{z}\mathbf{z}^\dagger)$ of the vector $\mathbf{z}$; the symbol E denotes expected value, so the entries of Q are the quantities $Q_{mn} = E(z_m\overline{z}_n)$. The diagonal entries of Q are $Q_{nn} = \sigma_n^2$, the variance of z_n.

Note that we have adopted the common practice of using the same symbols, z_n, when speaking about the random variables and about the specific values of these random variables that are present in our data. The context should make it clear to which we are referring.

In case 2 we say that the *signal power* is equal to $|\gamma|^2 \frac{1}{N}\sum_{n=1}^{N}|s_n|^2 = \frac{1}{N}|\gamma|^2\mathbf{s}^\dagger\mathbf{s}$ and the *noise power* is $\frac{1}{N}\sum_{n=1}^{N}\sigma_n^2 = \frac{1}{N}tr(Q)$, where $tr(Q)$ is the trace of the matrix Q, that is, the sum of its diagonal terms; therefore the noise power is the average of the variances σ_n^2. The *input signal-to-noise ratio* (SNR_{in}) is the ratio of the signal power to that of the noise, prior to processing the data; that is,

$$\mathrm{SNR}_{\mathrm{in}} = \frac{1}{N}|\gamma|^2\mathbf{s}^\dagger\mathbf{s}/\frac{1}{N}tr(Q) = |\gamma|^2\mathbf{s}^\dagger\mathbf{s}/tr(Q).$$

Optimal linear filtering for detection:

In each case to be considered below, our detector will take the form of a linear estimate of γ; that is, we shall compute the estimate $\hat{\gamma}$ given by

$$\hat{\gamma} = \sum_{n=1}^{N}\overline{b}_n x_n = \mathbf{b}^\dagger\mathbf{x},$$

where $\mathbf{b} = (b_1, b_2, ..., b_N)^T$ is a vector to be determined. The objective is to use what we know about the situation to select the optimal $\mathbf{b}$, which will depend on $\mathbf{s}$ and Q.

For any given vector $\mathbf{b}$, the quantity

$$\hat{\gamma} = \mathbf{b}^\dagger\mathbf{x} = \gamma\mathbf{b}^\dagger\mathbf{s} + \mathbf{b}^\dagger\mathbf{z}$$

is a random variable whose mean value is equal to $\gamma\mathbf{b}^\dagger\mathbf{s}$ and whose variance is

$$var(\hat{\gamma}) = E(|\mathbf{b}^\dagger\mathbf{z}|^2) = E(\mathbf{b}^\dagger\mathbf{z}\mathbf{z}^\dagger\mathbf{b}) = \mathbf{b}^\dagger E(\mathbf{z}\mathbf{z}^\dagger)\mathbf{b} = \mathbf{b}^\dagger Q\mathbf{b}.$$

Therefore, the *output signal-to-noise ratio* (SNR_{out}) is defined to be

$$\text{SNR}_{\text{out}} = |\gamma \mathbf{b}^\dagger \mathbf{s}|^2 / \mathbf{b}^\dagger Q \mathbf{b}.$$

The advantage we obtain from processing the data is called the *gain* associated with $\mathbf{b}$ and is defined to be the ratio of the SNR_{out} to SNR_{in}; that is

$$\text{gain}(\mathbf{b}) = \frac{|\gamma \mathbf{b}^\dagger \mathbf{s}|^2/(\mathbf{b}^\dagger Q \mathbf{b})}{|\gamma|^2(\mathbf{s}^\dagger \mathbf{s})/tr(Q)} = \frac{|\mathbf{b}^\dagger \mathbf{s}|^2\, tr(Q)}{(\mathbf{b}^\dagger Q \mathbf{b})(\mathbf{s}^\dagger \mathbf{s})}.$$

The best $\mathbf{b}$ to use will be the one for which gain($\mathbf{b}$) is the largest. So, ignoring the terms in the gain formula that do not involve $\mathbf{b}$, we see that the problem becomes *maximize* $\frac{|\mathbf{b}^\dagger \mathbf{s}|^2}{\mathbf{b}^\dagger Q \mathbf{b}}$, *for fixed signal vector* $\mathbf{s}$ *and fixed noise correlation matrix* Q.

The Cauchy inequality plays a major role in optimal filtering and detection:

Cauchy's inequality: for any vectors $\mathbf{a}$ and $\mathbf{b}$ we have

$$|\mathbf{a}^\dagger \mathbf{b}|^2 \leq (\mathbf{a}^\dagger \mathbf{a})(\mathbf{b}^\dagger \mathbf{b}),$$

with equality if and only if $\mathbf{a}$ is proportional to $\mathbf{b}$, that is, there is a scalar β such that $\mathbf{b} = \beta \mathbf{a}$.

Exercise 1: Use Cauchy's inequality to show that, for any fixed vector $\mathbf{a}$, the choice $\mathbf{b} = \beta \mathbf{a}$ maximizes the quantity $|\mathbf{b}^\dagger \mathbf{a}|^2/\mathbf{b}^\dagger \mathbf{b}$, for any constant β.

Exercise 2: Use the definition of the correlation matrix Q to show that Q is Hermitian and that, for any vector $\mathbf{y}$, $\mathbf{y}^\dagger Q \mathbf{y} \geq 0$. Therefore Q is a nonnegative definite matrix and, using its eigenvector decomposition, can be written as $Q = CC^\dagger$, for some invertible square matrix C.

Exercise 3: Consider now the problem of maximizing $|\mathbf{b}^\dagger \mathbf{s}|^2/\mathbf{b}^\dagger Q \mathbf{b}$. Using the two previous exercises, show that the solution is $\mathbf{b} = \beta Q^{-1}\mathbf{s}$, for some arbitrary constant β.

We can now use the results of these exercises to continue our discussion. We choose the constant $\beta = 1/(\mathbf{s}^\dagger Q^{-1}\mathbf{s})$ so that the optimal $\mathbf{b}$ has $\mathbf{b}^\dagger \mathbf{s} = 1$; that is, the **optimal filter b** is

$$\mathbf{b} = (1/(\mathbf{s}^\dagger Q^{-1}\mathbf{s}))Q^{-1}\mathbf{s}$$

and the *optimal estimate* of γ is

$$\hat{\gamma} = \mathbf{b}^\dagger \mathbf{x} = (1/(\mathbf{s}^\dagger Q^{-1}\mathbf{s}))(\mathbf{s}^\dagger Q^{-1}\mathbf{x}).$$

The random variable $\hat{\gamma}$ has mean equal to $\gamma \mathbf{b}^\dagger \mathbf{s} = \gamma$ and variance equal to $1/(\mathbf{s}^\dagger Q^{-1}\mathbf{s})$. Therefore, the output signal power is $|\gamma|^2$, the output noise power is $1/(\mathbf{s}^\dagger Q^{-1}\mathbf{s})$ and so the *output signal-to-noise ratio* (SNR_{out}) is

$$\text{SNR}_{\text{out}} = |\gamma|^2(\mathbf{s}^\dagger Q^{-1}\mathbf{s}).$$

The gain associated with the optimal vector $\mathbf{b}$ is then

$$\text{maximum gain} = \frac{(\mathbf{s}^\dagger Q^{-1}\mathbf{s})\, tr(Q)}{(\mathbf{s}^\dagger \mathbf{s})}.$$

The calculation of the vector $C^{-1}\mathbf{x}$ is sometimes called *prewhitening* since $C^{-1}\mathbf{x} = \gamma C^{-1}\mathbf{s} + C^{-1}\mathbf{z}$ and the new noise vector, $C^{-1}\mathbf{z}$, has the identity matrix for its correlation matrix. The new signal vector is $C^{-1}\mathbf{s}$. The filtering operation that gives $\hat{\gamma} = \mathbf{b}^\dagger \mathbf{x}$ can be written as

$$\hat{\gamma} = (1/(\mathbf{s}^\dagger Q^{-1}\mathbf{s}))(C^{-1}\mathbf{s})^\dagger C^{-1}\mathbf{x};$$

the term $(C^{-1}\mathbf{s})^\dagger C^{-1}\mathbf{x}$ is described by saying that we *prewhiten, then do a matched filter*. Now we consider some special cases of noise.

The case of white noise:

We say that the noise is *white noise* if the correlation matrix is $Q = \sigma^2 I$, where I denotes the identity matrix that is one on the main diagonal and zero elsewhere and $\sigma > 0$ is the common standard deviation of the z_n. This means that the z_n are mutually uncorrelated (independent, in the Gaussian case) and share a common variance.

In this case the optimal vector $\mathbf{b}$ is $\mathbf{b} = \frac{1}{(\mathbf{s}^\dagger \mathbf{s})}\mathbf{s}$ and the gain is N. Notice that $\hat{\gamma}$ now involves only a matched filter. We consider now some special cases of the signal vectors $\mathbf{s}$.

Constant signal: Suppose that the vector $\mathbf{s}$ is constant, that is, $\mathbf{s} = \mathbf{1} = (1, 1, ..., 1)^T$. Then we have

$$\hat{\gamma} = \frac{1}{N}\sum_{n=1}^{N} x_n.$$

This is the same result we found in our discussion of the BLUE, when we estimated the mean value and the noise was white.

Sinusoidal signal - known frequency: Suppose

$$\mathbf{s} = \mathbf{e}(\omega_0) = (\exp(-i\omega_0), \exp(-2i\omega_0), ..., \exp(-Ni\omega_0))^T,$$

where ω_0 denotes a known frequency in $[-\pi, \pi)$. Then $\mathbf{b} = \frac{1}{N}\mathbf{e}(\omega_0)$ and

$$\hat{\gamma} = \frac{1}{N}\sum_{n=1}^{N} x_n \exp(in\omega_0);$$

so we see yet another occurrence of the DFT.

Sinusoidal signal - unknown frequency: If we do not know the value of the signal frequency ω_0 a reasonable thing to do is to calculate the $\hat{\gamma}$ for each (actually, finitely many) of the possible frequencies within $[-\pi, \pi)$ and base the detection decision on the largest value; that is, we calculate the DFT as a function of the variable ω. If there is only a single ω_0 for which there is a sinusoidal signal present in the data, the values of $\hat{\gamma}$ obtained at frequencies other than ω_0 provide estimates of the noise power σ^2, against which the value of $\hat{\gamma}$ for ω_0 can be compared.

The case of correlated noise:

We say that the noise is *correlated* if the correlation matrix is Q is not a multiple of the identity matrix. This means either that the z_n are mutually correlated (dependent, in the Gaussian case) or that they are uncorrelated, but have different variances.

In this case, as we saw above, the optimal vector $\mathbf{b}$ is

$$\mathbf{b} = \frac{1}{(\mathbf{s}^\dagger Q^{-1}\mathbf{s})} Q^{-1}\mathbf{s}$$

and the gain is

$$\text{maximum gain} = \frac{(\mathbf{s}^\dagger Q^{-1}\mathbf{s})\, tr(Q)}{(\mathbf{s}^\dagger \mathbf{s})}.$$

How large or small the gain is depends on how the signal vector $\mathbf{s}$ relates to the matrix Q.

For sinusoidal signals, the quantity $\mathbf{s}^\dagger\mathbf{s}$ is the same, for all values of the parameter ω; this is not always the case, however. In passive detection of sources in acoustic array processing, for example, the signal vectors arise from models of the acoustic medium involved. For far-field sources in an (acoustically) isotropic deep ocean, planewave models for $\mathbf{s}$ will have the property that $\mathbf{s}^\dagger\mathbf{s}$ does not change with source location. However, for near-field or shallow-water environments, this is usually no longer the case.

It follows from an earlier exercise that the quantity $\frac{\mathbf{s}^\dagger Q^{-1}\mathbf{s}}{\mathbf{s}^\dagger\mathbf{s}}$ achieves its maximum value when $\mathbf{s}$ is an eigenvector of Q associated with its smallest eigenvalue, λ_N; in this case, we are saying that the signal vector does not look very much like a typical noise vector. The maximum gain is then

$\lambda_N^{-1} tr(Q)$. Since $tr(Q)$ equals the sum of its eigenvalues, multiplying by $tr(Q)$ serves to normalize the gain, so that we cannot get larger gain simply by having all the eigenvalues of Q small.

On the other hand, if $\mathbf{s}$ should be an eigenvector of Q associated with its largest eigenvalue, say λ_1, then the maximum gain is $\lambda_1^{-1} tr(Q)$. If the noise is signal-like, that is, has one dominant eigenvalue, then $tr(Q)$ is approximately λ_1 and the maximum gain is around one, so we have lost the maximum gain of N we were able to get in the white noise case. This makes sense, in that it says that we cannot significantly improve our ability to discriminate between signal and noise by taking more samples, if the signal and noise are very similar.

Constant signal with unequal-variance uncorrelated noise: Suppose that the vector $\mathbf{s}$ is constant, that is, $\mathbf{s} = \mathbf{1} = (1, 1, ..., 1)^T$. Suppose also that the noise correlation matrix is $Q = \text{diag}\{\sigma_1, ..., \sigma_N\}$.

In this case the optimal vector $\mathbf{b}$ has entries

$$b_m = \frac{1}{(\sum_{n=1}^N \sigma_n^{-1})} \sigma_m^{-1},$$

for $m = 1, ..., N$, and we have

$$\hat{\gamma} = \frac{1}{(\sum_{n=1}^N \sigma_n^{-1})} \sum_{m=1}^N \sigma_m^{-1} x_m.$$

This is the BLUE estimate of γ in this case.

Sinusoidal signal - known frequency, in correlated noise: Suppose

$$\mathbf{s} = \mathbf{e}(\omega_0) = (\exp(-i\omega_0), \exp(-2i\omega_0), ..., \exp(-Ni\omega_0))^T,$$

where ω_0 denotes a known frequency in $[-\pi, \pi)$. In this case the optimal vector $\mathbf{b}$ is

$$\mathbf{b} = \frac{1}{\mathbf{e}(\omega_0)^\dagger Q^{-1} \mathbf{e}(\omega_0)} Q^{-1} \mathbf{e}(\omega_0)$$

and the gain is

$$\text{maximum gain} = \frac{1}{N} [\mathbf{e}(\omega_0)^\dagger Q^{-1} \mathbf{e}(\omega_0)] tr(Q).$$

How large or small the gain is depends on the quantity $q(\omega_0)$, where

$$q(\omega) = \mathbf{e}(\omega)^\dagger Q^{-1} \mathbf{e}(\omega).$$

The function $1/q(\omega)$ can be viewed as a sort of noise power spectrum, describing how the noise power appears when decomposed over the various

frequencies in $[-\pi, \pi)$. The maximum gain will be large if this *noise power spectrum* is relatively small near $\omega = \omega_0$; however, when the noise is similar to the signal, that is, when the noise power specrum is relatively large near $\omega = \omega_0$, the maximum gain can be small. In this case the noise power spectrum plays a role analogous to that played by the eigenvalues of Q earlier.

To see more clearly why it is that the function $1/q(\omega)$ can be viewed as a sort of noise power spectrum, consider what we get when we apply the optimal filter associated with ω to data containing only noise. The average output should tell us how much power there is in the component of the noise that resembles $\mathbf{e}(\omega)$; this is essentially what is meant by a noise power spectrum. The result is $\mathbf{b}^\dagger \mathbf{z} = (1/q(\omega))\mathbf{e}(\omega)^\dagger Q^{-1}\mathbf{z}$. The expected value of $|\mathbf{b}^\dagger \mathbf{z}|^2$ is then $1/q(\omega)$.

Sinusoidal signal - unknown frequency: Again, if we do not know the value of the signal frequency ω_0 a reasonable thing to do is to calculate the $\hat{\gamma}$ for each (actually, finitely many) of the possible frequencies within $[-\pi, \pi)$ and base the detection decision on the largest value. For each ω the corresponding value of $\hat{\gamma}$ is

$$\hat{\gamma}(\omega) = [1/(\mathbf{e}(\omega)^\dagger Q^{-1}\mathbf{e}(\omega))] \sum_{n=1}^{N} a_n \exp(in\omega),$$

where $\mathbf{a} = (a_1, a_2, ..., a_N)^T$ satisfies the linear system $Q\mathbf{a} = \mathbf{x}$ or $\mathbf{a} = Q^{-1}\mathbf{x}$. It is interesting to note the similarity between this estimation procedure and the PDFT discussed in earlier notes; to see the connection view $[1/(\mathbf{e}(\omega)^\dagger Q^{-1}\mathbf{e}(\omega))]$ in the role of $P(\omega)$ and Q its corresponding matrix of Fourier transform values. The analogy breaks down when we notice that Q need not be Toeplitz, as in the PDFT case; however, the similarity is intriguing.

Chapter 34

Random Signal Detection

We consider now the detection and estimation problem for the case in which the signal components have random aspects as well.

Random amplitude sinusoid in noise:

A somewhat more general model for sinusoids in additive noise is the following. The complex data vector is $\mathbf{x} = (x_1, x_2, ..., x_N)^T$. We assume that there are two possibilities:

Case 1: noise only

$$x_n = z_n,\ n = 1, ..., N,$$

or

Case 2: signal in noise

$$x_n = \gamma s_n + z_n,$$

where $\gamma = |\gamma| \exp(i\theta)$ is an unknown value of a complex random variable whose magnitude $|\gamma|$ and phase θ are mutually independent and independent of the noise. In this case the mean value of γ can be zero, if θ is distributed uniformly over $[-\pi, \pi)$. The presence of a nonzero signal component is detected through the increase in the variance, not through a nonzero mean value, as above. The calculations are basically the same as the earlier ones and we shall not consider this case further.

Multiple independent sinusoids in noise:

We mention briefly the case in which there may be more than one sinusoid present. For this case a random model is typically used, in which the

magnitudes and phases of the different sinusoids are taken to be mutually independent. Statistical hypothesis testing theory tells us that we should detect in two steps now:

1: perform a maximum likelihood estimation of the number and location (in frequency space) of the sinusoidal components; then

2: use the optimal linear filtering to estimate their respective coefficients, the γ's.

The first step is computationally intractible and various suboptimal, but computationally efficient, alternatives are commonly used. These alternative methods can involve the eigenvector- or singular value decomposition of certain matrices formed from the data vector $\mathbf{x}$, and so are nonlinear procedures. How well we can detect two or more separate signals will, of course, depend on how distinct their $\mathbf{s}$ vectors are, how distinct each is from the noise, how accurate our knowledge of the noise correlation matrix Q is, how accurate our model of the $\mathbf{s}$ is and on the value of N; this is the *resolution problem*. Our ability to resolve will also depend on the accuracy of the measurements, therefore on the hardware used to collect the measurements.

Data-adaptive high resolution methods:

In all of the discussion so far, we have assumed that the noise correlation matrix Q was available to use in forming the optimal filter $\mathbf{b}$. The Q may depend on data previously obtained or may simply be the result of a model chosen to describe the physical situation. In some applications, such as sonar array processing, the Q may vary from minute to minute; it would be helpful if we could obtain as good an estimate as possible of the current value of Q, but this would require measurements, at the present moment, of the noise without the embedded signal, which is impossible. One approach, due to Capon [46], is a *data-adaptive high resolution detection*; it has been used in the case in which there are potentially more than one signal present, to achieve higher resolution than that obtainable by the methods we have discussed so far.

Data-adaptive high resolution methods- sinusoidal signals

The idea behind these methods is to use the data vector $\mathbf{x}$ to estimate the noise correlation matrix. Since the vector $\mathbf{x}$ may also contain signals, it would seem that we would be lumping signals in with noise and designing a filter $\mathbf{b}$ to suppress everything. The constraint $\mathbf{b}^\dagger\mathbf{e}(\omega) = 1$ saves us, however.

Suppose that there are two signals present: then the vector $\mathbf{x}$ has components

$$x_n = \gamma_1 \exp(-in\omega_1) + \gamma_2 \exp(-in\omega_2) + z_n,$$

for $n = 1, ..., N$. When we are trying to detect $\mathbf{e}(\omega_1)$ it is fine if the $\mathbf{e}(\omega_2)$ component is viewed as noise, and vice versa. High resolution depends on what the output of our filter is when we look at a frequency ω that is between ω_1 and ω_2; now it is advantageous that the signal components are lumped in with the noise.

To obtain a substitute for Q we partition the N by 1 data vector $\mathbf{x}$ into K smaller M by 1 vectors, denoted $\mathbf{y}^k$, for $k = 1, ..., K$ and $N = MK$. Specifically, we let

$$y_m^k = x_{(k-1)M+m},\ m = 1, ..., M,$$

for $k = 1, 2, ..., K$. We then define the M by M matrix R as follows:

$$R_{jm} = \frac{1}{K} \sum_{k=1}^{K} y_j^k \, \overline{y}_m^k,$$

for j, $m = 1, 2, ..., M$. The matrix R is then Hermitian and nonnegative definite. The signal components involving $\mathbf{e}(\omega_1)$ and $\mathbf{e}(\omega_2)$ are transformed into shorter components of the form

$$\tilde{\mathbf{e}}(\omega) = (\exp(-i\omega), ..., \exp(-iM\omega))^T.$$

To obtain our data-adaptive estimate of the γ of the potential signal component $\tilde{\mathbf{e}}(\omega)$ we apply the optimal filtering, as before, but to each of the vectors $\mathbf{y}^k$ separately, using R instead of Q and using $\tilde{\mathbf{e}}(\omega)$ instead of $\mathbf{e}(\omega)$. We then average the squared magnitudes of the resulting estimates over $k = 1, ..., K$, to obtain our estimate of the $|\gamma|^2$ associated with ω.

Capon's data-adaptive estimator:

$$|\hat{\gamma}(\omega)|^2 = 1/(\tilde{\mathbf{e}}(\omega)^\dagger R^{-1}(\tilde{\mathbf{e}}(\omega)).$$

Exercise 1: (or better, Research Project 1.) What is going on here? Why is this method 'high resolution'? What does R look like? What are its eigenvalues and eigenvectors? Can we apply it to signals other than sinusoids? Is it important that the signal coefficients (the γ's) be random? What can go wrong? How can it be fixed?

Chapter 35

The Wave Equation

In this chapter and the next we demonstrate how the problem of Fourier transform estimation from sampled data arises in the processing of measurements obtained by sampling electromagnetic or acoustic field fluctuations, as in radar or sonar.

In many areas of remote sensing what we measure are the fluctuations in time of an electromagnetic or acoustic field. Such fields are described mathematically as solutions of certain partial differential equations, such as the *wave equation*. A function $u(x, y, z, t)$ is said to satisfy the *three-dimensional wave equation* if

$$u_{tt} = c^2(u_{xx} + u_{yy} + u_{zz}) = c^2\nabla^2 u,$$

where u_{tt} denotes the second partial derivative of u with respect to the time variable t twice and $c > 0$ is the (constant) speed of propagation. More complicated versions of the wave equation permit the speed of propagation c to vary with the spatial variables x, y, z, but we shall not consider that here.

We use the method of *separation of variables* at this point, to get some idea about the nature of solutions of the wave equation. Assume, for the moment, that the solution $u(t, x, y, z)$ has the simple form

$$u(t, x, y, z) = f(t)g(x, y, z).$$

Inserting this separated form into the wave equation we get

$$f''(t)g(x, y, z) = c^2 f(t)\nabla^2 g(x, y, z)$$

or

$$f''(t)/f(t) = c^2\nabla^2 g(x, y, z)/g(x, y, z).$$

The function on the left is independent of the spatial variables, while the one on the right is independent of the time variable; consequently, they

must both equal the same constant, which we denote $-\omega^2$. From this we have two separate equations,

$$f''(t) + \omega^2 f(t) = 0, \tag{35.1}$$

and

$$\nabla^2 g(x,y,z) + \frac{\omega^2}{c^2} g(x,y,z) = 0. \tag{35.2}$$

The equation (35.2) is the *Helmholtz equation*.

Equation (35.1) has for its solutions the functions $f(t) = \cos(\omega t)$ and $\sin(\omega t)$, or, in complex form, the complex exponential functions $f(t) = e^{i\omega t}$ and $f(t) = e^{-i\omega t}$. Functions $u(t,x,y,z) = f(t)g(x,y,z)$ with such time dependence are called *time-harmonic* solutions.

In three-dimensional spherical coordinates with $r = \sqrt{x^2+y^2+z^2}$ a radial function $u(r,t)$ satisfies the wave equation if

$$u_{tt} = c^2(u_{rr} + \frac{2}{r}u_r).$$

Exercise 1: Show that the radial function $u(r,t) = \frac{1}{r}h(r-ct)$ satisfies the wave equation for any twice differentiable function h.

Radial solutions to the wave equation have the property that at any fixed time the value of u is the same for all the points on a sphere centered at the origin; the curves of constant value of u are these spheres, for each fixed time.

Suppose at time $t = 0$ the function $h(r,0)$ is zero except for r near zero; that is, initially, there is a localized disturbance centered at the origin. As time passes that disturbance spreads out spherically. When the radius of a sphere is very large, the surface of the sphere appears planar, to an observer on that surface, who is said then to be in the *far field*. This motivates the study of solutions of the wave equation that are constant on planes; the so-called *planewave solutions*.

Exercise 2: Let $\mathbf{s} = (x,y,z)$ and $u(\mathbf{s},t) = u(x,y,z,t) = e^{i\omega t}e^{i\mathbf{k}\cdot\mathbf{s}}$. Show that u satisfies the wave equation $u_{tt} = c^2\nabla^2 u$ for any real vector $\mathbf{k}$, so long as $||\mathbf{k}||^2 = \omega^2/c^2$. This solution is a planewave associated with frequency ω and *wavevector* $\mathbf{k}$; at any fixed time the function $u(\mathbf{s},t)$ is constant on any plane in three dimensional space having $\mathbf{k}$ as a normal vector.

Chapter 36

Array Processing

In radar and sonar the field $u(\mathbf{s}, t)$ being sampled is usually viewed as a discrete or continuous superposition of planewave solutions with various amplitudes, frequencies and wavevectors. We sample the field at various spatial locations $\mathbf{s}_m$, $m = 1, ..., M$, for t in some finite interval of time. We simplify the situation a bit now by assuming that all the planewave solutions are associated with the same frequency, ω. If not, we perform an FFT on the functions of time received at each sensor location $\mathbf{s}_m$ and keep only the value associated with the desired frequency ω.

In the continuous superposition model the field is

$$u(\mathbf{s}, t) = e^{i\omega t} \int f(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{s}} d\mathbf{k}.$$

Our measurements at the sensor locations $\mathbf{s}_m$ give us the values

$$F(\mathbf{s}_m) = \int f(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{s}_m} d\mathbf{k},$$

for $m = 1, ..., M$. The data are then Fourier transform values of the complex function $f(\mathbf{k})$; $f(\mathbf{k})$ is defined for all three-dimensional real vectors $\mathbf{k}$, but is zero, in theory, at least, for those $\mathbf{k}$ whose squared length $||\mathbf{k}||^2$ is not equal to ω^2/c^2. Our goal is then to estimate $f(\mathbf{k})$ from finitely many values of its Fourier transform. Since each $\mathbf{k}$ is a normal vector for its planewave field component, determining the value of $f(\mathbf{k})$ will tell us the strength of the planewave component coming from the direction $\mathbf{k}$.

The collection of sensors at the spatial locations s_m, $m = 1, ..., M$, is called *an array* and the size of the array, in units of the wavelength $\lambda = 2\pi c/\omega$, is called the *aperture* of the array. Generally the larger the aperture the better, but what is a large aperture for one value of ω will be a smaller aperture for a lower frequency. The book by Haykin [84] is a useful reference, as is the review paper by Wright, Pridham and Kay [148].

In some applications the sensor locations are essentially arbitrary, while in others their locations are carefully chosen. Sometimes, the sensors are collinear, as in sonar towed arrays. Let's look more closely at the collinear case.

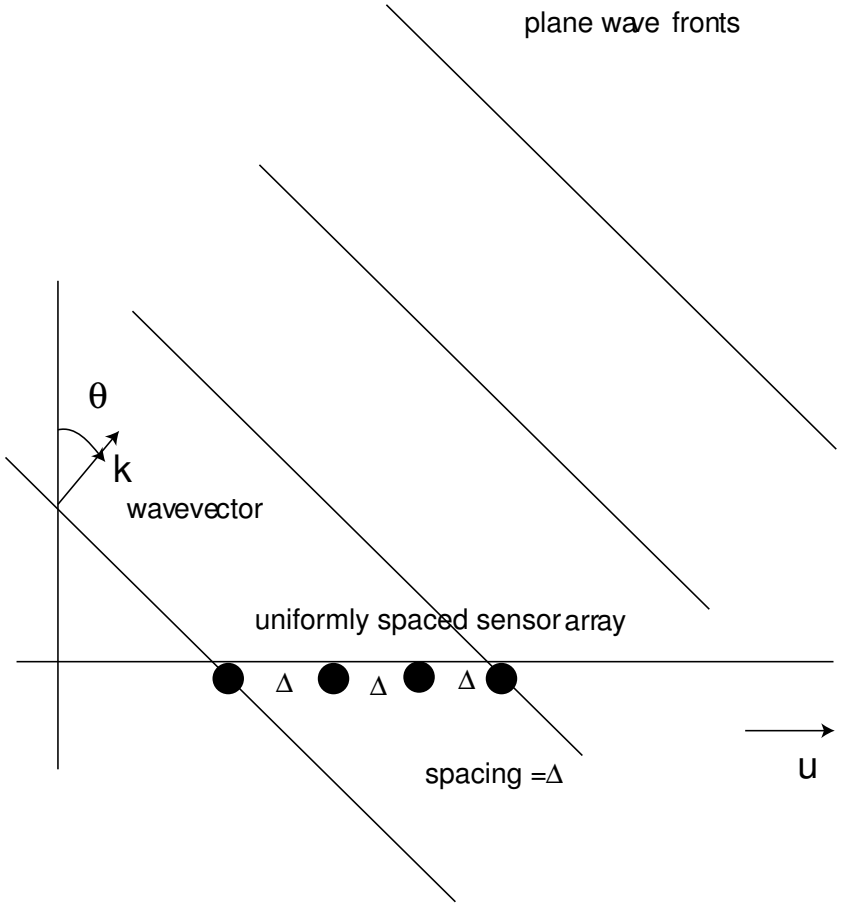


Figure 36.1: A uniform line array sensing a planewave field.

We assume now that the sensors are equispaced along the x-axis, at locations $(m\Delta, 0, 0)$, $m = 1, ..., M$, where $\Delta > 0$ is the sensor spacing; such an arrangement is called a *uniform line array*; this setup is illustrated in Figure 36.1. Our data is then

$$F_m = F(\mathbf{s}_m) = F((m\Delta, 0, 0)) = \int f(\mathbf{k}) e^{im\Delta \mathbf{k} \cdot (1,0,0)} d\mathbf{k}.$$

Since $\mathbf{k} \cdot (1,0,0) = \frac{\omega}{c}\cos\theta$, for θ the angle between the vector $\mathbf{k}$ and the x-axis, we see that there is some ambiguity now; we cannot distinguish the cone of vectors that have the same θ. It is common then to assume that the wavevectors $\mathbf{k}$ have no z-component and that θ is the angle between two vectors in the x, y-plane, the so-called *angle of arrival*. The *wavenumber* variable $k = \frac{\omega}{c}\cos\theta$ lies in the interval $[-\frac{\omega}{c}, \frac{\omega}{c}]$ and we imagine that $f(\mathbf{k})$ is now $f(k)$, defined for $|k| \leq \frac{\omega}{c}$. The Fourier transform of $f(k)$ is $F(s)$, a function of a single real variable s. Our data is then viewed as the values $F(m\Delta)$, for $m = 1, ..., M$. Since the function $f(k)$ is zero for $|k| > \frac{\omega}{c}$ the Nyquist spacing in s is $\frac{\pi c}{\omega}$, which is $\frac{\lambda}{2}$, where $\lambda = \frac{2\pi c}{\omega}$ is the wavelength.

To avoid aliasing, which now means mistaking one direction of arrival for another, we need to select $\Delta \leq \frac{\lambda}{2}$. When we have oversampled, so that $\Delta < \frac{\lambda}{2}$, the interval $[-\frac{\omega}{c}, \frac{\omega}{c}]$, the so-called *visible region*, is strictly smaller than the interval $[-\frac{\pi}{\Delta}, \frac{\pi}{\Delta}]$. If the model of propagation is accurate all the signal component planewaves will correspond to wavenumbers k in the visible region and the background noise will also appear as a superposition of such propagating planewaves. In practice, there can be components in the noise that appear to come from wavenumbers k outside of the visible region; this means these components of the noise are not due to distant sources propagating as planewaves, but, perhaps, to sources that are in the *near field*, or localized around individual sensors, or coming from the electronics within the sensors.

Using the formula $\lambda\omega = 2\pi c$ we can calculate the Nyquist spacing for any particular case of planewave array processing. For electromagnetic waves the propagation speed is the speed of light, which we shall take here to be $c = 3 \times 10^8$ meters per second. The wavelength λ for gamma rays is around one Angstrom, which is 10^{-10} meters; for x-rays it is about one millimicron, or 10^{-9} meters; the visible spectrum has wavelengths that are a little less than one micron, that is, 10^{-6} meters. Shortwave radio has wavelength around one millimeter; broadcast radio has a λ running from about 10 meters to 1000 meters, while the so-called long radio waves can have wavelengths several thousand meters long. At the one extreme it is impractical (if not physically impossible) to place individual sensors at the Nyquist spacing of fractions of microns, while at the other end, managing to place the sensors far enough apart is the challenge.

The wavelengths used in primitive early radar at the start of World War II were several meters long. Since resolution is proportional to aperture, which, in turn, is the length of the array, in units of wavelength, antennae for such radar needed to be quite large. As Körner notes in [102], the general feeling at the time was that the side with the shortest wavelength would win the war. The cavity magnetron, invented during the war by British scientists, made possible 10 cm wavelength radar, which could then easily be mounted on planes.

In ocean acoustics it is usually assumed that the speed of propagation of sound is around 1500 meters per second, although deviations from this *ambient sound speed* are significant, and since they are caused by such things as temperature differences in the ocean, can be used to estimate these differences. At around the frequency $\omega = 50$ Hz we find sound generated by man-made machinery, such as motors in vessels, with higher frequency harmonics sometimes present also; at other frequencies the main sources of acoustic energy may be wind-driven waves or whales. The wavelength for 50 Hz is $\lambda = 30$ meters; sonar will typically operate both above and below this wavelength. It is sometimes the case that the array of sensors is fixed in place, so what may be Nyquist spacing for 50 Hz will be oversampling for 20 Hz.

It is often the case that we are primarily interested in the values $|f(\mathbf{k})|$, not the complex values $f(\mathbf{k})$. Since the Fourier transform of the function $|f(\mathbf{k})|^2$ is the autocorrelation function obtained by convolving the function F with $\overline{F}$, we can mimic the approach used earlier for power spectrum estimation to find $|f(\mathbf{k})|$. We can now employ the nonlinear methods such as Burg's MEM and Capon's maximum likelihood method.

In array processing, as in other forms of signal and image processing, we want to remove the noise and enhance the information-bearing component, the signal. To do this we need some idea of the statistical behavior of the noise, we need a physically accurate description of what the signals probably look like and we need a way to use this information. Much of our discussion up to now has been about the many ways in which such prior information can be incorporated in linear and nonlinear procedures. We have not said much about the important issue of the sensitivity of these methods to mismatch; that is, What happens when our physical model is wrong or the statistics of the noise is not what we thought it was? We did note earlier how Burg's MEM resolves closely spaced sinusoids when the background is white noise, but when the noise is correlated, MEM can degrade rapidly.

Even when the physical model and noise statistics are reasonably accurate, slight errors in the hardware can cause rapid degradation of the processor. Sometimes acoustic signal processing is performed with sensors that are designed to be expendable and are therefore less expensive and more prone to errors than more permanent equipment. Knowing what a sensor has received is important, but so is knowing when it received it. Slight phase errors caused by the hardware can go unnoticed when the data is processed in one manner, but can ruin the performance of another method.

The information we seek is often stored redundantly in the data and hardware errors may harm only some of these storage locations, making robust processing still possible. As we saw in our discussion of eigenvector methods, information about the frequencies of the complex exponential

components of the signal are stored in the roots of the polynomials obtained from some of the eigenvectors. In [42] it was demonstrated that, in the presence of correlated noise background, phase errors distort the roots of some of these polynomials more than others; robust estimation of the frequencies is still possible if the stable roots are interrogated.

We have focused here exclusively on planewave propagation, which results when the source is far enough way from the sensors and the speed of propagation is constant. In many important applications these conditions are violated, different versions of the wave equation are needed, which have different solutions. For example, sonar signal processing in environments such as shallow channels, in which some of the sound reaches the sensors only after interacting with the ocean floor or the surface, requires more complicated parameterized models for solutions of the appropriate wave equation. Lack of information about the depth and nature of the bottom can also cause errors in the signal processing. In some cases it is possible to use acoustic energy from known sources to determine the needed information.

Array signal processing can be done in *passive* or *active* mode. In passive mode the energy is either reflected off of or originates at the object of interest: the moon reflects sunlight, while ships generate their own noise. In the active mode the object of interest does not generate or reflect enough energy by itself, so the energy is generated by the party doing the sensing: active sonor is sometimes used to locate quiet vessels, while radar is used to locate planes in the sky or to map the surface of the earth. In the February 2003 issue of Harper's is an article on scientific apocalypse, dealing with the search for near-earth asteroids. These objects are initially detected by passive optical observation, as small dots of reflected sunlight; once detected, they are then imaged by active radar to determine their size, shape, rotation and such.

Chapter 37

Transmission Tomography

In this chapter we show how the two dimensional Fourier transform arises in transmission tomographic image processing. See the texts [115] and [116] for more detailed discussion of these matters.

As an x-ray beam passes through the body it encounters various types of matter, soft tissue, bone, ligaments, air, each weakening the beam to a greater or lesser extent. If the strength of the beam upon entry is S_{in} and S_{out} is its lesser strength after passing through the body, then

$$S_{out} = S_{in} e^{-\int_L f},$$

where $f = f(x, y) \geq 0$ is the *attenuation function* describing the two-dimensional distribution of matter within the slice of the body being scanned and $\int_L f$ is the integral of the function f over the line L along which the x-ray beam has passed. From knowledge of S_{in} and S_{out} we can determine $\int_L f$. As we shall see, if we know $\int_L f$ for every line in the x, y-plane we can reconstruct the attenuation function f. In actual *computer-assisted tomography* (CAT) scans we know line integrals only approximately and only for finitely many lines. Figure 37.1 illustrates the situation. In practice the function f is replaced by a grid of pixels, as shown in Figure 37.2.

Let θ be a fixed angle in the interval $[0, \pi)$ and consider the rotation of the x, y coordinate axes to produce the t, s axis system, where

$$t = x \cos\theta + y \sin\theta,$$

and

$$s = -x \sin\theta + y \cos\theta.$$

We can then write the attenuation function f as a function of the variables t and s. For each fixed value of t we compute the integral $\int f(x, y) ds$, obtaining the integral of $f(x, y) = f(t\cos\theta - s\sin\theta, t\sin\theta + s\cos\theta)$ along

the single line L corresponding to the fixed values of θ and t. We repeat this process for every value of t and then change the angle θ and repeat again. In this way we obtain the integrals of f over every line L in the plane. We denote by $r_f(\theta, t)$ the integral

$$r_f(\theta, t) = \int_L f(x, y) ds.$$

The function $r_f(\theta, t)$ is called the *Radon transform* of f.

For fixed θ the function $r_f(\theta, t)$ is a function of the single real variable t; let $R_f(\theta, \omega)$ be its Fourier transform. Then

$$R_f(\theta, \omega) = \int (\int f(x, y) ds) e^{i\omega t} dt,$$

which we can write as

$$R_f(\theta, \omega) = \int \int f(x, y) e^{i\omega(x \cos\theta + y \sin\theta)} dx dy = F(\omega \cos\theta, \omega \sin\theta),$$

where $F(\omega \cos\theta, \omega \sin\theta)$ is the two-dimensional Fourier transform of the function $f(x, y)$, evaluated at the point $(\omega \cos\theta, \omega \sin\theta)$; this relationship is called the *central slice theorem*. For fixed θ as we change the value of ω we obtain the values of the function F along the points of the line making the angle θ with the horizontal axis. As θ varies in $[0, \pi)$ we get all the values of the function F. Once we have F we can obtain f using the formula for the two-dimensional inverse Fourier transform. We conclude that we are able to determine f from its line integrals.

The inversion formula tells us that the function $f(x, y)$ can be obtained as

$$f(x, y) = \frac{1}{4\pi^2} \int \int F(u, v) e^{-i(xu+yv)} du dv.$$

Expressing the double integral in polar coordinates (ω, θ), with $\omega \geq 0$, $u = \omega \cos\theta$ and $v = \omega \sin\theta$, we get

$$f(x, y) = \frac{1}{4\pi^2} \int_0^{2\pi} \int_0^{\infty} F(u, v) e^{-i(xu+yv)} \omega d\omega d\theta,$$

or

$$f(x, y) = \frac{1}{4\pi^2} \int_0^{\pi} \int_{-\infty}^{\infty} F(u, v) e^{-i(xu+yv)} |\omega| d\omega d\theta.$$

Now write

$$F(u, v) = F(\omega \cos\theta, \omega \sin\theta) = R_f(\theta, \omega),$$

where $R_f(\theta, \omega)$ is the FT with respect to t of $r_f(\theta, t)$ so that

$$\int_{-\infty}^{\infty} F(u, v) e^{-i(xu+yv)} |\omega| d\omega = \int_{-\infty}^{\infty} R_f(\theta, \omega) |\omega| e^{-i\omega t} d\omega.$$

The function $h_f(\theta, t)$ defined for $t = x\cos\theta + y\sin\theta$ by

$$h_f(\theta, x\cos\theta + y\sin\theta) = \int_{-\infty}^{\infty} R_f(\theta,\omega)|\omega|e^{-i\omega t}d\omega$$

is the result of a linear filtering of $r_f(\theta, t)$ using a *ramp filter* with transfer function $G(\omega) = |\omega|$. Then

$$f(x, y) = \int_0^{\pi} h_f(\theta, x\cos\theta + y\sin\theta)d\theta$$

gives $f(x, y)$ as the result of a *backprojection operator*; for every fixed value of (θ, t) add $h_f(\theta, t)$ to the current value at the point (x, y) for all (x, y) lying on the straight line determined by θ and t by $t = x\cos\theta+y\sin\theta$. The final value at a fixed point (x, y) is then the sum of all the values $h_f(\theta, t)$ for those (θ, t) for which (x, y) is on the line $t = x\cos\theta + y\sin\theta$. It is therefore said that $f(x, y)$ can be obtained by *filtered backprojection* (FBP) of the line integral data.

Knowing that $f(x, y)$ is related to the complete set of line integrals by filtered backprojection suggests that when only finitely many line integrals are available a similar ramp filtering and backprojection can be used to estimate $f(x, y)$; in the clinic this is the most widely used method for the reconstruction of tomographic images.

There is a second way to recover $f(x, y)$ using backprojection and filtering, this time in the reverse order; that is, we backproject the Radon transform and then ramp filter the resulting function of two variables. We begin again with the relation

$$f(x, y) = \frac{1}{4\pi^2}\int_0^{2\pi}\int_0^{\infty} F(u, v)e^{-i(xu+yv)}\omega d\omega d\theta,$$

which we write as

$$f(x, y) = \frac{1}{4\pi^2}\int_0^{2\pi}\int_0^{\infty} \frac{F(u, v)}{\sqrt{u^2+v^2}}\sqrt{u^2+v^2}e^{-i(xu+yv)}\omega d\omega d\theta$$

$$= \frac{1}{4\pi^2}\int_0^{2\pi}\int_0^{\infty} G(u, v)\sqrt{u^2+v^2}e^{-i(xu+yv)}\omega d\omega d\theta, \tag{37.1}$$

using

$$G(u, v) = \frac{F(u, v)}{\sqrt{u^2+v^2}}$$

for $(u, v) \neq (0, 0)$. Equation (37.1) expresses $f(x, y)$ as the result of ramp filtering $g(x, y)$, the inverse Fourier transform of $G(u, v)$. We show now

that $g(x,y)$ is the backprojection of the function $r_f(\omega,t)$; that is, we show that

$$g(x,y) = \int_0^\pi r_f(\theta, x\cos\theta + y\sin\theta)d\theta.$$

From the central slice theorem we know that $g(x,y)$ can be written as

$$g(x,y) = \int_0^\pi h_g(\theta, x\cos\theta + y\sin\theta)d\theta,$$

where

$$h_g(\theta, x\cos\theta + y\sin\theta) = \int_{-\infty}^{\infty} R_g(\theta,\omega)|\omega|e^{-i\omega(x\cos\theta + y\sin\theta)}d\omega.$$

Since

$$R_g(\theta,\omega) = G(\omega\cos\theta, \omega\sin\theta)$$

we have

$$g(x,y) = \int_0^\pi \int_{-\infty}^{\infty} G(\omega\cos\theta,\omega\sin\theta)|\omega|e^{-i\omega(x\cos\theta+y\sin\theta)}d\omega d\theta$$

$$= \int_0^\pi \int_{-\infty}^{\infty} F(\omega\cos\theta,\omega\sin\theta)e^{-i\omega(x\cos\theta+y\sin\theta)}d\omega d\theta$$

$$= \int_0^\pi \int_{-\infty}^{\infty} R_f(\theta,\omega)e^{-i\omega(x\cos\theta+y\sin\theta)}d\omega d\theta$$

$$= \int_0^\pi r_f(\theta, x\cos\theta + y\sin\theta)d\theta.$$

This is what we wanted.

We have found that the recovery of $f(x,y)$ from its line integrals can be accomplished using filtering and backprojection in two different ways: one way is to filter the function $r_f(\theta,t)$, viewed as a function of t, with a ramp filter, then backproject; the other way is to backproject $r_f(\theta,t)$ first and then filter the resulting function of two variables with a ramp filter in two dimensions. Both of these filtered backprojection methods have their analogs in the processing of actual finite data.

As we noted above, in actual CAT scans only finitely many θ are used and for each θ only finitely many t are employed. Therefore at each step along the way we are dealing only with approximations of what the theory would provide. In addition to that, the data we have are not exactly line integrals of f but more precisely integrals of f along narrow strips.

Although the one and two dimensional Fourier transforms do play roles in CAT scan imaging there are better reconstruction methods based on iterative algorithms such as ART and the EMML.

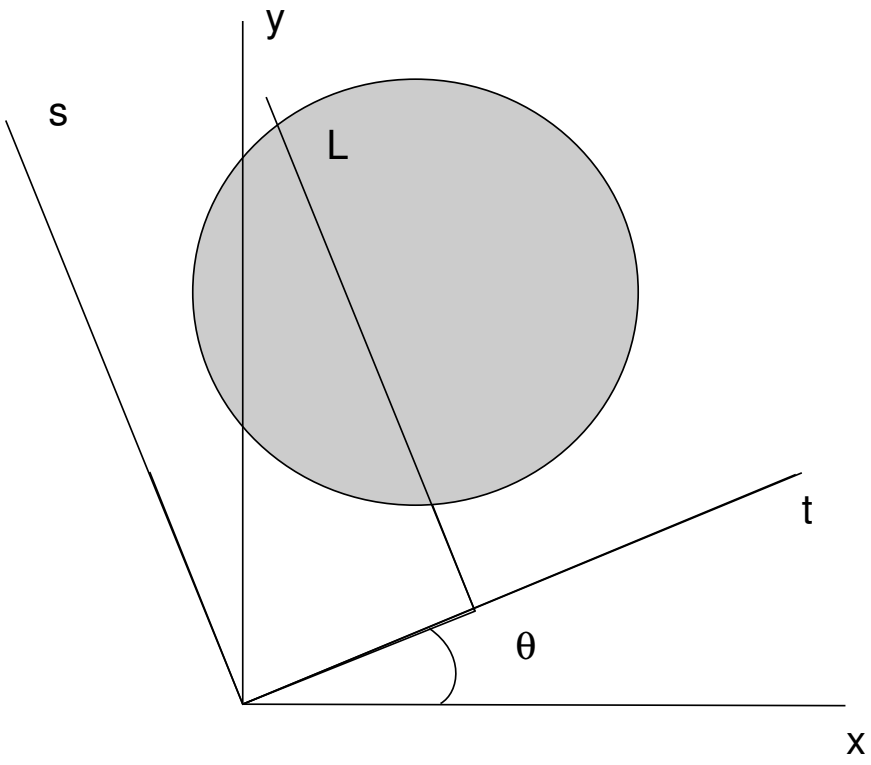


Figure 37.1: The Radon transform of f at (t, θ) is the line integral of f along line L.

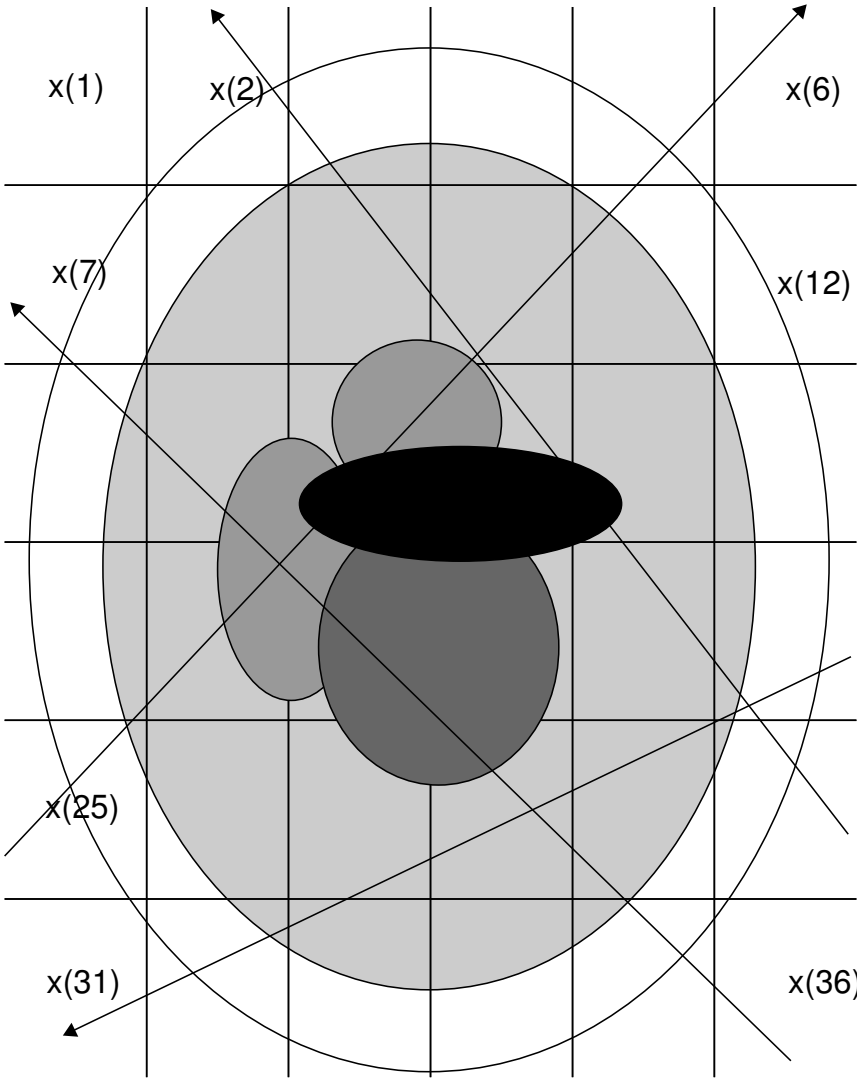


Figure 37.2: The Radon transform for a discretized object.

Chapter 38

Resolution Limits

We began in the introductory chapter by saying that our data has been obtained through some form of sensing; physical models, often simplified, describe how the data we have obtained relates to the information we seek; there usually isn't enough data and what we have is corrupted by noise and other distortions. All of the models and algorithms we have considered have as their aim the overcoming of this inherent problem of limited data. But just how limited is the data and in what sense limited? After all, if Burg's maximum entropy method (MEM) resolves peaks that are left unresolved by the DFT, the problem would seem to lie not with the data, which must still retain the desired information, but with the method used. When Burg's MEM produces incorrect reconstructions in the presence of a background that is not flat, but the IPDFT is able to use an estimate of the background to provide a better answer, is it the data or the method that is limiting? On the other hand, when we say MEM has produced an incorrect answer what do we mean? We know that MEM gives a positive estimate of the power spectrum that is exactly consistent with the autocorrelation data; it is only incorrect because we know the true spectrum, having created it in our simulations. Such questions concern everyone using inversion methods, and yet have no completely satisfying answers. Bertero's paper [8] is a good place to start one's education in these matters. In this chapter we consider some of these issues, in so far as they concern the methods we have discussed in this text.

The DFT:

The exercise following our discussion of the second approach to signal analysis uses the DFT to illustrate the notion of *resolution limit*. The signal there was the sum of two sinusoids, at frequencies $\omega_1 = -\alpha$ and $\omega_2 = \alpha$. As the α approached zero resolution in the DFT was eventually lost; for

larger data lengths the α could be smaller before this happened. We know from successful application of high-resolution methods that this does not mean that the information about the two sinusoids has been lost. What does it mean?

The DFT shows up almost everywhere in signal processing. As a finite Fourier series it can be viewed as a best approximation of the infinite Fourier series; as a matched filter it is the optimal linear method for detecting a single sinusoid in white noise. However, it is not the optimal linear method for detecting two sinusoids in white noise. If we know that the signal is the sum of two sinusoids (with equal amplitudes, for now) in additive white noise, the optimal linear filter is a matched filter of the form $\mathbf{e}_{\alpha\beta}^{\dagger}\mathbf{d}$, where $\mathbf{d}$ is the data vector and $\mathbf{e}_{\alpha\beta}$ is the data we would have received had the signal consisted solely of $e^{i\alpha t} + e^{i\beta t}$. The output of the matched filter is a function of the two variables α and β. We plot the magnitude of this function of two variables and select the pair for which the magnitude is greatest. If we apply this procedure to the signal in the exercise we would find that we could still determine that there are sinusoids at α and $\beta = -\alpha$. The DFT manages to resolve sinusoids when they are far enough apart to be treated as two separate signals, each with a single sinusoid. Otherwise, the DFT is simply not the proper estimate of frequency location for multiple sinusoids. A proper notion of resolution limit should be based on something other than the behavior of the DFT in the presence of two sinusoids.

Bandlimited extrapolation reconsidered:

Suppose we want to estimate the function $F(\omega)$, known to be zero for $|\omega| > \Omega$, where $0 < \Omega < \pi$. Our data will be samples of the inverse Fourier transform, $f(x)$. Suppose, in addition, that we are able to select our finitely many samples only for x within the bounded interval $[0, X]$, but are otherwise unrestricted; that is, we can take as many samples at whichever x values we wish. What should we do?

Shannon's *sampling theorem* tells us that we can reconstruct $F(\omega)$ exactly if we know the values $f(n\frac{\pi}{\Omega})$ for all the integers n. Then we have

$$F(\omega) = \frac{\pi}{\Omega}\sum\nolimits_{n=-\infty}^{\infty} f(n\frac{\pi}{\Omega})e^{in\frac{\pi}{\Omega}\omega}.$$

The sampling rate of $\Delta = \frac{\pi}{\Omega}$ is the *Nyquist rate* and the doubly infinite sequence of samples at this rate is all we need. But, of course, we cannot actually measure infinitely many values of $f(x)$. Furthermore, we are restricted to the interval $[0, X]$. If

$$(N-1)\frac{\pi}{\Omega} \leq X < N\frac{\pi}{\Omega}$$

then there are N Nyquist samples available within the interval $[0, X]$. Some have concluded that the sampling theorem tells us that we can do no better than to take the N samples $f(n\frac{\pi}{\Omega})$, $n = 0, 1, ..., N-1$, that we have N *degrees of freedom* in selecting data from within the interval $[0, X]$ and our freedom is thus exhausted when we have taken these N samples. The questions are: Can we do better? and Is there a quantifiable limit to our freedom to extract information under these restrictions? If someone offered to give you the value of $f(x)$ at one new point x within the interval $[0, X]$, would you take it?

No one would argue that the N Nyquist samples determine completely the values of $f(x)$ for the remaining x within the interval $[0, X]$. The problem is more how to use this new data value. The DFT

$$F_{DFT}(\omega) = \frac{\pi}{\Omega}\chi_\Omega(\omega)\sum\nolimits_{n=0}^{N-1} f(n\frac{\pi}{\Omega})e^{in\frac{\pi}{\Omega}\omega}$$

is zero outside the interval $[-\Omega, \Omega]$, is consistent with the data and therefore could be the right answer. If we are given the additional value $f(a)$ the estimate

$$\frac{\pi}{\Omega}\chi_\Omega(\omega)[f(a)e^{ia\omega} + \sum\nolimits_{n=0}^{N-1} f(n\frac{\pi}{\Omega})e^{in\frac{\pi}{\Omega}\omega}]$$

is not consistent with the data.

Using the non-iterative bandlimited extrapolation estimate given in equation (19.7) we can get an estimate with is consistent with this no longer uniformly spaced data as well as with the band limitation. So it is possible to make good use of the additional sample offered to us; we should accept it. Is there no end to this, however? Should we simply take as many samples as we desire, equispaced or not? Is there some limit to our freedom to squeeze information out of the behavior of the function $f(x)$ within the interval $[0, X]$? The answer is Yes, there are limits, but the limits depend in sometimes subtle ways on themethod being used and the amount and nature of the noise involved, which must include round-off error and quantization. Let's consider this more closely, with respect to the non-iterative bandlimited extrapolation method.

As we saw earlier, the non-iterative Gerchberg-Papoulis bandlimited extrapolation method leads to the estimate

$$F_\Omega(\omega) = \chi_\Omega(\omega)\sum\nolimits_{m=1}^{M} \frac{1}{\lambda_m}(\mathbf{u}^m)^\dagger \mathbf{d}\, U^m(\omega),$$

where $\mathbf{d}$ is the data vector. In contrast, the DFT estimate is

$$F_{DFT}(\omega) = \sum\nolimits_{m=1}^{M} (\mathbf{u}^m)^\dagger \mathbf{d}\, U^m(\omega).$$

The estimate $F_\Omega(\omega)$ can provide better resolution within the interval $[-\Omega, \Omega]$ because of the multiplier $1/\lambda_m$, causing the estimate to rely more heavily on

those functions $U_m(\omega)$ having more roots, therefore more structure, within that interval. But therein lies the danger, as well.

When the data is noise-free the dot product $(\mathbf{u}^m)^\dagger\mathbf{d}$ is relatively small for those eigenvectors $\mathbf{u}_m$ corresponding to the small eigenvalues; therefore the product $(1/\lambda_m)(\mathbf{u}^m)^\dagger\mathbf{d}$ is not large. However, when the data vector $\mathbf{d}$ contains noise, the dot product of the noise component with each of the eigenvectors is about the same size. Therefore, the product $(1/\lambda_m)(\mathbf{u}^m)^\dagger\mathbf{d}$ is now quite large and the estimate is dominated by the noise. This sensitivity to the noise is the limiting factor in the bandlimited extrapolation. Any reasonable definitions of *degrees of freedom* and *resolution limit* must include the signal-to-noise ratio, as well as the fall-off rate of the eigenvalues of the matrix. In our bandlimited extrapolation problem the matrix is the sinc matrix. The proportion of nearly zero eigenvalues will be approximately $1 - \frac{\Omega}{\pi}$; the smaller the ratio $\frac{\Omega}{\pi}$ the fewer essentially nonzero eigenvalues there will be. For other extrapolation methods, such as the PDFT, the fall-off rate may be somewhat different. For analogous methods in higher dimensions the fall-off rate may be quite different [8].

High-resolution methods:

The bandlimited extrapolation methods we have studied are linear in the data, while the high-resolution methods are not. The high-resolution methods we have considered, such as MEM, Capon's method, the IPDFT and the eigenvector techniques, exploit the fact that the frequencies of sinusoidal components can be associated with the roots of certain polynomials obtained from eigenvectors of the autocorrelation matrix. When the roots are disturbed by phase errors or are displaced by the presence of a non-flat background, the methods that use these roots perform badly. As we mentioned earlier, there is some redundancy in the storage of information in these roots and stable processing is still possible in many cases. Not all the eigenvectors store this information and a successful method must interrogate the ones that do. Additive white noise causes MEM to fail by increasing all the eigenvalues, but does not hurt explicit eigenvector methods. Correlated noise that cannot be effectively prewhitened hurts all these methods, by making it more difficult to separate the information-bearing eigenvectors from the others. Correlation between sinusoidal components, as may occur in multipath arrivals in shallow water, causes additional difficulty, as does short data length, which corrupts the estimates of the autocorrelation values.

Bibliography

[1] Agmon, S. (1954) The relaxation method for linear inequalities, *Canadian Journal of Mathematics*, **6**, pp. 382–392.

[2] Anderson, A. and Kak, A. (1984) Simultaneous algebraic reconstruction technique (SART): a superior implementation of the ART algorithm, *Ultrasonic Imaging*, **6**, pp. 81–94.

[3] Ash, R., and Gardner, M. (1975) *Topics in Stochastic Processes*, Academic Press.

[4] Baggeroer, A., Kuperman, W., and Schmidt, H. (1988) Matched field processing: source localization in correlated noise as optimum parameter estimation, *Journal of the Acoustical Society of America*, **83**, pp. 571–587.

[5] Bauschke, H. (2001) Projection algorithms: results and open problems, in *Inherently Parallel Algorithms in Feasibility and Optimization and their Applications*, Butnariu, D., Censor, Y. and Reich, S., editors, Elsevier Publ., pp. 11–22.

[6] Bauschke, H., and Borwein, J. (1996) On projection algorithms for solving convex feasibility problems, *SIAM Review*, **38 (3)**, pp. 367–426.

[7] Bauschke, H., Borwein, J., and Lewis, A. (1997) The method of cyclic projections for closed convex sets in Hilbert space, *Contemporary Mathematics: Recent Developments in Optimization Theory and Nonlinear Analysis*, **204**, American Mathematical Society, pp. 1–38.

[8] Bertero, M. (1992) Sampling theory, resolution limits and inversion methods, in [10], pp. 71–94.

[9] Bertero, M., and Boccacci, P. (1998) *Introduction to Inverse Problems in Imaging*, Institute of Physics Publishing, Bristol, UK.

[10] Bertero, M., and Pike, E.R. (eds.) (1992) *Inverse Problems in Scattering and Imaging*, Malvern Physics Series, Adam Hilger, IOP Publishing, London.

[11] Bertsekas, D.P. (1997) A new class of incremental gradient methods for least squares problems, *SIAM J. Optim.*, **7**, pp. 913-926.

[12] Blackman, R., and Tukey, J. (1959) *The Measurement of Power Spectra*, Dover.

[13] Boggess, A., and Narcowich, F. (2001) *A First Course in Wavelets, with Fourier Analysis*, Prentice-Hall, NJ.

[14] Born, M., and Wolf, E. (1999) *Principles of Optics: 7-th edition*, Cambridge University Press.

[15] Borwein, J., and Lewis, A. (2000) *Convex Analysis and Nonlinear Optimization*, Canadian Mathematical Society Books in Mathematics, Springer, New York.

[16] Bregman, L.M. (1967) The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming, *USSR Computational Mathematics and Mathematical Physics*, **7**: 200–217.

[17] Browne, J. and A. DePierro, A. (1996) A row-action alternative to the EM algorithm for maximizing likelihoods in emission tomography, *IEEE Trans. Med. Imag.*, **15**, 687-699.

[18] Bucker, H. (1976) Use of calculated sound fields and matched field detection to locate sound sources in shallow water, *Journal of the Acoustical Society of America*, **59**, pp. 368–373.

[19] Burg, J. (1967) Maximum entropy spectral analysis, *paper presented at the 37th Annual SEG meeting, Oklahoma City, OK.*

[20] Burg, J. (1972) The relationship between maximum entropy spectra and maximum likelihood spectra, *Geophysics*, **37**, pp. 375–376.

[21] Burg, J. (1975) *Maximum Entropy Spectral Analysis*, Ph.D. dissertation, Stanford University.

[22] Byrne, C. (1992) Effects of modal phase errors on eigenvector and nonlinear methods for source localization in matched field processing, *Journal of the Acoustical Society of America*, **92(4)**, pp. 2159–2164.

[23] Byrne, C. (1993) Iterative image reconstruction algorithms based on cross-entropy minimization, *IEEE Transactions on Image Processing*, **IP-2**, pp. 96–103.

[24] Byrne, C. (1995) Erratum and addendum to "Iterative image reconstruction algorithms based on cross-entropy minimization", *IEEE Transactions on Image Processing*, **IP-4**, pp. 225–226.

[25] Byrne, C. (1996) Iterative reconstruction algorithms based on cross-entropy minimization, in: *Image Models (and their Speech Model Cousins)*, (S.E. Levinson and L. Shepp, Editors), the IMA Volumes in Mathematics and its Applications, Volume 80, Springer-Verlag, New York, pp. 1–11.

[26] Byrne, C. (1996) Block-iterative methods for image reconstruction from projections, *IEEE Transactions on Image Processing*, **IP-5**, pp. 792–794.

[27] Byrne, C. (1997) Convergent block-iterative algorithms for image reconstruction from inconsistent data, *IEEE Transactions on Image Processing*, **IP-6**, pp. 1296–1304.

[28] Byrne, C. (1998) Accelerating the EMML algorithm and related iterative algorithms by rescaled block-iterative (RBI) methods, *IEEE Transactions on Image Processing*, **IP-7**, pp. 100-109.

[29] Byrne, C. (1999) Iterative projection onto convex sets using multiple Bregman distances, *Inverse Problems*, **15**, pp. 1295-1313.

[30] Byrne, C. (2000) Block-iterative interior point optimization methods for image reconstruction from limited data, *Inverse Problems*, **16**, pp. 1405–1419.

[31] Byrne, C. (2001) Bregman-Legendre multidistance projection algorithms for convex feasibility and optimization, in *Inherently Parallel Algorithms in Feasibility and Optimization and their Applications*, Butnariu, D., Censor, Y. and Reich, S., editors, Elsevier Publ., pp. 87–100.

[32] Byrne, C. (2002) Iterative oblique projection onto convex sets and the split feasibility problem, *Inverse Problems*, **18**, pp. 441-453.

[33] Byrne, C. and Censor, Y. (2001) Proximity function minimization using multiple Bregman projections, with applications to split feasibility and Kullback-Leibler distance minimization, *Annals of Operations Research*, **105**, pp. 77–98.

[34] Byrne, C. and Fitzgerald, R. (1979) A unifying model for spectrum estimation, *Proceedings of the RADC Workshop on Spectrum Estimation- October 1979*, Griffiss AFB, Rome, NY.

[35] Byrne, C. and Fitzgerald, R. (1982) Reconstruction from partial information, with applications to tomography, *SIAM J. Applied Math.*, **42(4)**, pp. 933–940.

[36] Byrne, C., Fitzgerald, R., Fiddy, M., Hall, T. and Darling, A. (1983) Image restoration and resolution enhancement, *J. Opt. Soc. Amer.*, **73**, pp. 1481–1487.

[37] Byrne, C. and Fitzgerald, R. (1984) Spectral estimators that extend the maximum entropy and maximum likelihood methods, *SIAM J. Applied Math.*, **44(2)**, pp. 425–442.

[38] Byrne, C. and Fiddy, M. (1987) Estimation of continuous object distributions from Fourier magnitude measurements, *JOSA A*, **4**, pp. 412–417.

[39] Byrne, C., and Fiddy, M. (1988) Images as power spectra; reconstruction as Wiener filter approximation, *Inverse Problems*, **4**, pp. 399–409.

[40] Byrne, C., Haughton, D., and Jiang, T. (1993) High-resolution inversion of the discrete Poisson and binomial transformations, *Inverse Problems*, **9**, pp. 39–56.

[41] Byrne, C., Levine, B.M., and Dainty, J.C. (1984) Stable estimation of the probability density function of intensity from photon frequency counts, *JOSA Communications*, **1(11)**, pp. 1132–1135.

[42] Byrne, C., and Steele, A. (1985) Stable nonlinear methods for sensor array processing, *IEEE Transactions on Oceanic Engineering*, **OE-10(3)**, pp. 255–259.

[43] Byrne, C., Brent, R., Feuillade, C., and DelBalzo, D (1990) A stable data-adaptive method for matched-field array processing in acoustic waveguides, *Journal of the Acoustical Society of America*, **87(6)**, pp. 2493–2502.

[44] Byrne, C., Frichter, G., and Feuillade, C. (1990) Sector-focused stability methods for robust source localization in matched-field processing, *Journal of the Acoustical Society of America*, **88(6)**, pp. 2843–2851.

[45] Candy, J. (1988) *Signal Processing: The Modern Approach*, McGraw-Hill.

[46] Capon, J. (1969) High-resolution frequency-wavenumber spectrum analysis, *Proc. of the IEEE*, **57**, pp. 1408–1418.

[47] Cederquist, J., Fienup, J., Wackerman, C., Robinson, S., and Kryskowski, D. (1989) Wave-front phase estimation from Fourier intensity measurements, *Journal of the Optical Society of America A*, **6(7)**, pp. 1020–1026.

[48] Censor, Y. (1981) Row-action methods for huge and sparse systems and their applications, *SIAM Review*, **23**: 444–464.

[49] Censor, Y. and Elfving, T. (1994) A multiprojection algorithm using Bregman projections in a product space, *Numerical Algorithms*, **8**: 221–239.

[50] Censor, Y., Eggermont, P.P.B., and Gordon, D. (1983) Strong underrelaxation in Kaczmarz's method for inconsistent systems, *Numerische Mathematik*, **41**, pp. 83-92.

[51] Censor, Y. and Segman, J. (1987) On block-iterative maximization, *J. of Information and Optimization Sciences*, **8**, pp. 275-291.

[52] Censor, Y. and Zenios, S.A. (1997) *Parallel Optimization: Theory, Algorithms and Applications*, Oxford University Press, New York.

[53] Childers, D. (ed.)(1978) *Modern Spectral Analysis*, IEEE Press, New York.

[54] Christensen, O. (2003) *An Introduction to Frames and Riesz Bases*, Birkhäuser, Boston.

[55] Chui, C. (1992) *An Introduction to Wavelets*, Academic Press, Boston.

[56] Chui, C., and Chen, G. (1991) *Kalman Filtering*, second edition, Springer-Verlag, Berlin.

[57] Cimmino, G. (1938) Calcolo approssimato per soluzioni die sistemi di equazioni lineari, *La Ricerca Scientifica XVI, Series II, Anno IX*, **1**, pp. 326–333.

[58] Combettes, P. (1993) The foundations of set theoretic estimation, *Proceedings of the IEEE*, **81 (2)**, pp. 182–208.

[59] Combettes, P. (1996) The convex feasibility problem in image recovery, *Advances in Imaging and Electron Physics*, **95**, pp. 155–270.

[60] Combettes, P., and Trussell, J. (1990) Method of successive projections for finding a common point of sets in a metric space, *Journal of Optimization Theory and Applications*, **67 (3)**, pp. 487–507.

[61] Cooley, J., and Tukey, J. (1965) An algorithm for the machine calculation of complex Fourier series, *Math. Comp.*, **19**, pp. 297–301.

[62] Cox, H. (1973) Resolving power and sensitivity to mismatch of optimum array processors, *Journal of the Acoustical Society of America*, **54**, pp. 771–785.

[63] Csiszár, I., and Tusnády, G. (1984) Information geometry and alternating minimization procedures, *Statistics and Decisions*, Supp. 1, pp. 205–237.

[64] Dainty, C., and Fiddy, M. (1984) The essential role of prior knowleldge in phase retrieval, *Optica Acta*, **31**, pp. 325–330.

[65] Darroch, J., and Ratcliff, D. (1972) Generalized iterative scaling for log-linear models, *Annals of Mathematical Statistics*, **43**, pp. 1470–1480.

[66] De Bruijn, N. (1967) Uncertainty principles in Fourier analysis, in *Inequalties*, O. Shisha, (ed.), Academic Press, pp. 57–71.

[67] Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977) Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society, Series B*, **37**: 1–38.

[68] De Pierro, A., and Iusem, A. (1990) On the asymptotic behaviour of some alternate smoothing series expansion iterative methods, *Linear Algebra and its Applications*, **130**, pp. 3–24.

[69] Dhanantwari, A., Stergiopoulos, S., and Iakovidis, I. (2001) Correcting organ motion artifacts in x-ray CT medical imaging systems by adaptive processing. I. Theory, *Med. Phys.*, **28(8)**, pp. 1562–1576.

[70] Everitt, B., and Hand, D. (1981) *Finite Mixture Distributions*, Chapman and Hall, London.

[71] Feuillade, C., DelBalzo, D., and Rowe, M. (1989) Environmental mismatch in shallow-water matched-field processing: geoacoustic parameter variability, *Journal of the Acoustical Society of America*, **85**, pp. 2354–2364.

[72] Feynman, R., Leighton, R., and Sands, M. (1963) *The Feynman Lectures on Physics, Vol. 1*, Addison-Wesley.

[73] Fiddy, M. (1983) The phase retrieval problem, in *Inverse Optics*, SPIE Proceedings 413 (A.J. Devaney, ed.), pp. 176–181.

[74] Fienup, J. (1979) Space object imaging through the turbulent atmosphere, *Optical Engineering*, **18**, pp. 529–534.

[75] Fienup, J. (1987) Reconstruction of a complex-valued object from the modulus of its Fourier transform using a support constraint, *Journal of the Optical Society of America A*, **4(1)**, pp. 118–123.

[76] Frieden, B. R. (1982) *Probability, Statistical Optics and Data Testing*, Springer.

[77] Gabor, D. (1946) Theory of communication, *Journal of the IEE (London)*, **93**, pp. 429–457.

[78] Gasquet, C., and Witomski, F. (1998) *Fourier Analysis and Applications*, Springer.

[79] Gelb, A. (1974) (ed.) *Applied Optimal Estimation*, written by the technical staff of The Analytic Sciences Corporation, MIT Press.

[80] Gerchberg, R. W. (1974) Super-restoration through error energy reduction, *Optica Acta*, **21**, pp. 709–720.

[81] Gordon, R., Bender, R., and Herman, G.T. (1970) Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and x-ray photography, *J. Theoret. Biol.*, **29**, pp. 471-481.

[82] Groetsch, C. (1999) *Inverse Problems: Activities for Undergraduates*, The Mathematical Association of America.

[83] Gubin, L.G., Polyak, B.T. and Raik, E.V. (1967) The method of projections for finding the common point of convex sets, *USSR Computational Mathematics and Mathematical Physics*, **7**: 1–24.

[84] Haykin, S. (1985) *Array Signal Processing*, Prentice-Hall.

[85] Herman, G.T. (1999) *private communication.*

[86] Herman, G. T. and Meyer, L. (1993) Algebraic reconstruction techniques can be made computationally efficient, *IEEE Transactions on Medical Imaging*, **12**, pp. 600-609.

[87] Hildreth, C. (1957) A quadratic programming procedure, *Naval Research Logistics Quarterly*, **4**, pp. 79–85. Erratum, ibid., p. 361.

[88] Hinich, M. (1973) Maximum likelihood signal processing for a vertical array, *Journal of the Acoustical Society of America*, **54**, pp. 499–503.

[89] Hinich, M. (1979) Maximum likelihood estimation of the position of a radiating source in a waveguide, *Journal of the Acoustical Society of America*, **66**, pp. 480–483.

[90] Hoffman, K. (1962) *Banach Spaces of Analytic Functions*, Prentice-Hall.

[91] Hogg, R., and Craig, A. (1978) *Introduction to Mathematical Statistics*, MacMillan.

[92] Holte, S., Schmidlin, P., Linden, A., Rosenqvist, G. and Eriksson, L. (1990) Iterative image reconstruction for positron emission tomography: a study of convergence and quantitation problems, *IEEE Transactions on Nuclear Science*, **37**, pp. 629–635.

[93] Hubbard, B. (1998) *The World According to Wavelets*, A.K. Peters, Publ., Natick, MA.

[94] Hudson, H. M., and Larkin, R. S. (1994) Accelerated image reconstruction using ordered subsets of projection data, *IEEE Transactions on Medical Imaging*, **13**, pp. 601-609.

[95] Hutton, B., Kyme, A., Lau, Y., Skerrett, D., and Fulton, R. (2002) A hybrid 3-D reconstruction/registration algorithm for correction of head motion in emission tomography, *IEEE Transactions on Nuclear Science*, **49 (1)**, pp. 188–194.

[96] Johnson, R. (1960) *Advanced Euclidean Geometry*, Dover.

[97] Kaczmarz, S. (1937) Angenäherte Auflösung von Systemen linearer Gleichungen, *Bulletin de l'Academie Polonaise des Sciences et Lettres*, **A35**, 355-357.

[98] Kaiser, G. (1994) *A Friendly Guide to Wavelets*, Birkhäuser, Boston.

[99] Kalman, R. (1960) A new approach to linear filtering and prediction problems, *Trans. ASME, J. Basic Eng.*, **82**, pp. 35–45.

[100] Katznelson, Y. (1983) *An Introduction to Harmonic Analysis*, Wiley.

[101] Körner, T. (1988) *Fourier Analysis*, Cambridge University Press.

[102] Körner, T. (1996) *The Pleasures of Counting*, Cambridge University Press.

[103] Kullback, S. and Leibler, R. (1951) On information and sufficiency, *Annals of Mathematical Statistics*, **22**: 79–86.

[104] Landweber, L. (1951) An iterative formula for Fredholm integral equations of the first kind, *Amer. J. of Math.*, **73**, pp. 615-624.

[105] Lane, R. (1987) Recovery of complex images from Fourier magnitude, *Optics Communications*, **63(1)**, pp. 6–10.

[106] Lange, K. and Carson, R. (1984) EM reconstruction algorithms for emission and transmission tomography, *Journal of Computer Assisted Tomography*, **8**: 306–316.

[107] Lange, K., Bahn, M. and Little, R. (1987) A theoretical study of some maximum likelihood algorithms for emission and transmission tomography, *IEEE Trans. Med. Imag.*, **MI-6(2)**, 106-114.

[108] Leahy, R., and Byrne, C. (2000) Guest editorial: Recent development in iterative image reconstruction for PET and SPECT, *IEEE Trans. Med. Imag.*, **19**, pp. 257-260.

[109] Lent, A. (1998) *private communication*.

[110] Liao, C.-W., Fiddy, M., and Byrne, C. (1997) Imaging from the zero locations of far-field intensity data, *Journal of the Optical Society of America -A*, **14 (12)**, pp. 3155–3161.

[111] McLachlan, G.J. and Krishnan, T. (1997) *The EM Algorithm and Extensions*, John Wiley and Sons, New York.

[112] Meidunas, E. (2001) *Re-scaled Block Iterative Expectation Maximization Maximum Likelihood (RBI-EMML) Abundance Estimation and Sub-pixel Material Identification in Hyperspectral Imagery*, MS thesis, Department of Electrical Engineering, University of Massachusetts Lowell, Lowell MA.

[113] Meyer, Y. (1993) *Wavelets: Algorithms and Applications*, SIAM, Philadelphia, PA.

[114] Motzkin, T., and Schoenberg, I. (1954) The relaxation method for linear inequalities, *Canadian Journal of Mathematics*, **6**, pp. 393–404.

[115] Natterer, F. (1986) *Mathematics of Computed Tomography*, Wiley and Sons, NY.

[116] Natterer, F., and Wübbeling, F. (2001) *Mathematical Methods in Image Reconstruction*, SIAM.

[117] Oppenheim, A., and Schafer, R. (1975) *Digital Signal Processing*, Prentice-Hall.

[118] Papoulis, A. (1975) A new algorithm in spectral analysis and band-limited extrapolation, *IEEE Transactions on Circuits and Systems*, **22**, pp. 735–742.

[119] Papoulis, A. (1977) *Signal Analysis*, McGraw-Hill.

[120] Paulraj, A., Roy, R., and Kailath, T. (1986) A subspace rotation approach to signal parameter estimation, *Proceedings of the IEEE*, pp. 1044–1045.

[121] Peressini, A., Sullivan, F., and Uhl, J. (1988) *The Mathematics of Nonlinear Programming*, Springer.

[122] Pisarenko, V. (1973) The retrieval of harmonics from a covariance function, *Geoph. J. R. Astrom. Soc.*, **30**.

[123] Poggio, T., and Smale, S. (2003) The mathematics of learning: dealing with data, *Notices of the American Mathematical Society*, **50 (5)**, pp. 537–544.

[124] Priestley, M. B. (1981) *Spectral Analysis and Time Series*, Academic Press.

[125] Prony, G.R.B. (1795) Essai expérimental et analytique sur les lois de la dilatabilité de fluides élastiques et sur celles de la force expansion de la vapeur de l'alcool, à différentes températures, *Journal de l'Ecole Polytechnique* (Paris), **1(2)**, pp. 24–76.

[126] Qian, H. (1990) Inverse Poisson transformation and shot noise filtering, *Rev. Sci. Instrum.*, **61**, pp. 2088–2091.

[127] Rockafellar, R. (1970) *Convex Analysis*, Princeton University Press.

[128] Schmidlin, P. (1972) Iterative separation of sections in tomographic scintigrams, *Nucl. Med.*, **15(1)**, Schatten Verlag, Stuttgart.

[129] Schmidt, R. (1981) *A Signal Subspace Approach to Multiple Emitter Location and Spectral Estimation*, PhD thesis, Stanford University, CA.

[130] Schuster, A. (1898) On the investigation of hidden periodicities with application to a supposed 26 day period of meteorological phenomena, *Terrestrial Magnetism*, **3**, pp. 13–41.

[131] Shang, E. (1985) Source depth estimation in waveguides, *Journal of the Acoustical Society of America*, **77**, pp. 1413–1418.

[132] Shang, E. (1985) Passive harmonic source ranging in waveguides by using mode filter, *Journal of the Acoustical Society of America*, **78**, pp. 172–175.

[133] Shang, E., Wang, H., and Huang, Z. (1988) Waveguide characterization and source localization in shallow water waveguides using Prony's method, *Journal of the Acoustical Society of America*, **83**, pp. 103–106.

[134] Smith, C. Ray, and Grandy, W.T., eds. (1985) *Maximum-Entropy and Bayesian Methods in Inverse Problems*, Reidel.

[135] Smith, C. Ray, and Erickson, G., eds. (1987) *Maximum-Entropy and Bayesian Spectral Analysis and Estimation Problems*, Reidel.

[136] Stark, H. and Yang, Y. (1998) *Vector Space Projections: A Numerical Approach to Signal and Image Processing, Neural Nets and Optics*, John Wiley and Sons, New York.

[137] Strang, G., and Nguyen, T. (1997) *Wavelets and Filter Banks*, Wellesley-Cambridge Press.

[138] Tanabe, K. (1971) Projection method for solving a singular system of linear equations and its applications, *Numer. Math.*, **17**, 203-214.

[139] Therrien, C. (1992) *Discrete Random Signals and Statistical Signal Processing*, Prentice-Hall.

[140] Tindle, C., Guthrie, K., Bold, G., Johns, M., Jones, D., Dixon, K., and Birdsall, T. (1978) Measurements of the frequency dependence of normal modes, *Journal of the Acoustical Society of America*, **64**, pp. 1178–1185.

[141] Tolstoy, A. (1993) *Matched Field Processing for Underwater Acoustics*, World Scientific.

[142] Twomey, S. (1996) *Introduction to the Mathematics of Inversion in Remote Sensing and Indirect Measurement*, Dover.

[143] Van Trees, H. (1968) *Detection, Estimation and Modulation Theory*, Wiley, New York.

[144] Vardi, Y., Shepp, L.A. and Kaufman, L. (1985) A statistical model for positron emission tomography, *Journal of the American Statistical Association*, **80**: 8–20.

[145] Walnut, D. (2002) *An Introduction to Wavelets*, Birkhäuser, Boston.

[146] Widrow, B., and Stearns, S. (1985) *Adaptive Signal Processing*, Prentice-Hall.

[147] Wiener, N. (1949) *Time Series*, MIT Press.

[148] Wright, W., Pridham, R., and Kay, S. (1981) Digital signal processing for sonar, *Proc. IEEE*, **69**, pp. 1451–1506.

[149] Yang, T.C. (1987) A method of range and depth estimation by modal decomposition, *Journal of the Acoustical Society of America*, **82**, pp. 1736–1745.

[150] Youla, D. (1978) Generalized image restoration by the method of alternating projections, *IEEE Transactions on Circuits and Systems*, **CAS-25 (9)**, pp. 694–702.

[151] Youla, D.C. (1987) Mathematical theory of image restoration by the method of convex projections, in: Stark, H. (Editor) (1987) *Image Recovery: Theory and Applications*, Academic Press, Orlando, FL, USA, pp. 29–78.

[152] Young, R. (1980) *An Introduction to Nonharmonic Fourier Analysis*, Academic Press.

Index