Chapter 5

Fourier methods

5.1 Introduction

Of the many books on Fourier methods those by Chatfield [11], Proakis and Manolakis [51] and Bloomfield [4] are particularly good.

5.2 Sinewaves and Samples

Sines and cosines can be understood in terms of the vertical and horizontal displacement of a fixed point on a rotating wheel; the wheel has unit length and rotates *anti-clockwise*. The angle round the wheel is measured in degrees or radians ($0 - 2\pi$; for unit radius circles the circumference is 2π, radians tell us how much of the circumference we've got). If we go round the wheel a whole number of times we end up in the same place, eg.$\cos 4\pi = \cos 2\pi = \cos 0 = 1$. Frequency, f, is the number of times round the wheel per second. Therefore, given $x = \cos(2\pi f t)$, $x = 1$ at $t = 1/f, 2/f$ etc. For $x = \cos(2\pi f t + \Phi)$ we get a head start (lead) of Φ radians. Negative frequencies may be viewed as a wheel rotating *clockwise* instead of anti-clockwise.

If we assume we have samples of the signal every T_s seconds and in total we have N such samples then T_s is known as the sampling period and $F_s = 1/T_s$ is the sampling frequency in Hertz (Hz) (samples per second). The nth sample occurs at time $t[n] = nT_s = n/F_s$. The cosine of sampled data can be written

$$x[n] = \cos(2\pi f t[n]) \tag{5.1}$$

When dealing with sampled signals it is important to note that some frequencies become indistinguishable from others; at a sampling frequency F_s the only *unique* frequencies are in the range 0 to $(F_s/2)$Hz. Any frequencies outside this range become *aliases* of one of the unique frequencies.

For example, if we sample at 8Hz then a -6Hz signal becomes indistinguishable from

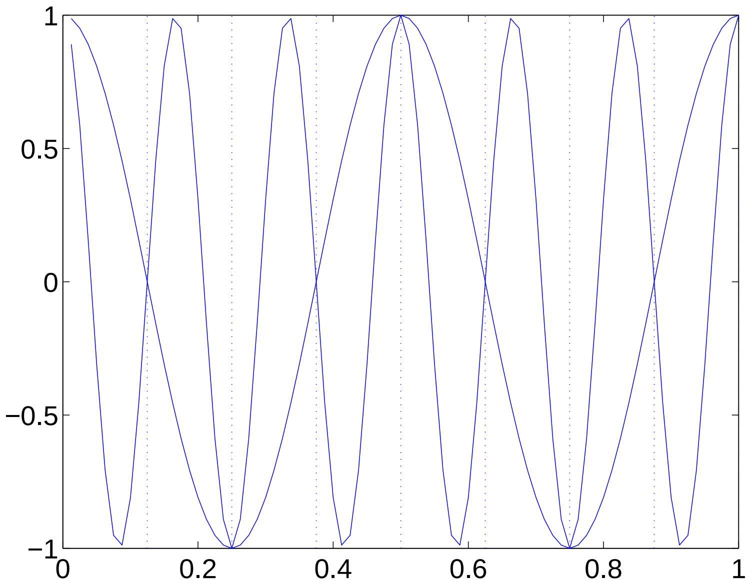

Figure 5.1: **Aliases** *The figure shows a 2Hz cosine wave and a -6Hz cosine wave as solid curves. At sampling times given by the dotted lines, which correspond to a sampling frequency of 8Hz, the* $-6Hz$ *signal is an alias of the 2Hz signal. Other aliases are given by equation 5.2.*

a 2Hz signal. This is shown in figure 5.1. More generally, if f_0 is a unique frequency then its aliases have frequencies given by

$$f = f_0 + kF_s \tag{5.2}$$

where k is any positive or negative integer, eg. for $f_0 = 2$ and $F_s = 8$ the two lowest frequency aliases, given by $k = -1$ and $k = 1$, are -6Hz and 10Hz.

Because of *aliasing* we must be careful when we interpret the results of spectral analysis. This is discussed more at the end of the lecture.

5.3 Sinusoidal models

If our time series has a periodic component in it we might think about modelling it with the equation

$$x[n] = R_0 + Rcos(2\pi f t[n] + \Phi) + e[n] \tag{5.3}$$

where R_0 is the offset (eg. mean value of x[n]), R is the amplitude of the sine wave, f is the frequency and Φ is the phase. What our model does'nt explain will be soaked up in the error term $e[n]$. Because of the trig identity

$$\cos(A + B) = \cos A \cos B - \sin A \sin B \tag{5.4}$$

the model can be written in an alternative form

$$x[n] = R_0 + a\cos(2\pi f t[n]) + b\sin(2\pi f t[n]) + e[n] \tag{5.5}$$

where $a = R\cos(\Phi)$ and $b = -R\sin(\Phi)$. This is the form we consider for subsequent analysis.

This type of model is similar to a class of models in statistics called *Generalised Linear Models* (GLIMS). They perform *nonlinear regression* by, first, taking *fixed* nonlinear functions of the inputs, these functions being called *basis functions*, and second, form an output by taking a linear combination of the basis function outputs. In sinusoidal models the basis functions are sines and cosines. In statistics a much broader class of functions is considered. However, sinewaves have some nice properties as we shall see.

5.3.1 Fitting the model

If we let $\boldsymbol{x} = [x(1), x(2), ..., x(N)]^T$, $\boldsymbol{w} = [R_0, a, b]^T$, $\boldsymbol{e} = [e_1, e_2, ..., e_N]^T$ and

$$\boldsymbol{A} = \begin{bmatrix} 1 & cos2\pi ft[1] & sin2\pi ft[1] \\ 1 & cos2\pi ft[2] & sin2\pi ft[2] \\ 1 & cos2\pi ft[3] & sin2\pi ft[3] \\ .. & .. & .. \\ 1 & cos2\pi ft[N] & sin2\pi ft[N] \end{bmatrix} \tag{5.6}$$

then the model can be written in the matrix form

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{w} + \boldsymbol{e} \tag{5.7}$$

which is in the standard form of a multivariate linear regression problem. The solution is therefore

$$\boldsymbol{w} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{x} \tag{5.8}$$

5.3.2 But sinewaves are orthogonal

Because we are dealing with sinewaves it turns out that the above solution simplifies. We restrict ourselves to a frequency f_p which is an integer multiple of the *base frequency*

$$f_p = pF_b \tag{5.9}$$

where $p = 1..N/2$ and

$$f_b = \frac{F_s}{N} \tag{5.10}$$

eg. for $F_s = 100$ and $N = 100$ (1 seconds worth of data), $f_b = 1Hz$ and we can have f_p from 1Hz up to 50Hz[1]. The *orthogonality* of sinewaves is expressed in the following equations

$$\sum_{n=1}^{N} \cos 2\pi f_k t[n] = \sum_{n=1}^{N} \sin 2\pi f_k t[n] = 0 \tag{5.11}$$

[1] To keep things simple we don't allow f_p where $p = N/2$; if we did allow it we'd get N and 0 in equations 5.14 and 5.15 for the case $k = l$. Also we must have N even.

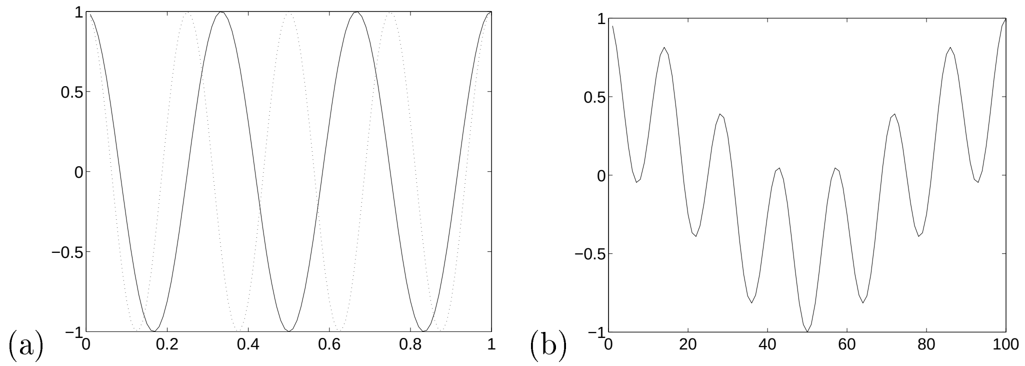

Figure 5.2: **Orthogonality of sinewaves** *Figure (a) shows* $\cos 2\pi 3 f_b t[n]$ *and* $\cos 2\pi 4 f_b t[n]$*, cosines which are 3 and 4 times the base frequency* $f_b = 1Hz$*. For any two integer multiples* k, l *we get* $\sum_{n=1}^{N} cos 2\pi f_k t[n] cos 2\pi f_l t[n] = 0$*. This can be seen from Figure (b) which shows the product* $cos 2\pi 3 f_b t[n] cos 2\pi 4 f_b t[n]$*. Because of the trig identity* $cos A cos B = 0.5 cos(A + B) + 0.5 cos(A - B)$ *this looks like a 7Hz signal superimposed on a 1Hz signal. The sum of this signal over a whole number of cycles can be seen to be zero; because each cos term sums to zero. If, however, k or l are not integers the product does not sum to zero and the orthogonality breaks down.*

$$\sum_{n=1}^{N} \cos 2\pi f_k t[n] \sin 2\pi f_l t[n] = 0 \tag{5.12}$$

$$\sum_{n=1}^{N} \cos 2\pi f_k t[n] \sin 2\pi f_l t[n] = 0 \tag{5.13}$$

$$\sum_{n=1}^{N} \cos 2\pi f_k t[n] \cos 2\pi f_l t[n] \quad = \quad \begin{array}{ll} 0 & k \neq l \\ N/2 & k = l \end{array} \tag{5.14}$$

$$\sum_{n=1}^{N} \sin 2\pi f_k t[n] \sin 2\pi f_l t[n] \quad = \quad \begin{array}{ll} 0 & k \neq l \\ N/2 & k = l \end{array} \tag{5.15}$$

These results can be proved by various trig. identities or, more simply, by converting to complex exponentials (see [5] or later in this chapter). The results depend on the fact that all frequencies that appear in the above sums are integer multiples of the base frequency; see figure 5.2.

This property of sinewaves leads to the result

$$\boldsymbol{A}^T \boldsymbol{A} = \boldsymbol{D} \tag{5.16}$$

where $\boldsymbol{D}$ is a diagonal matrix. The first entry is N (from the inner product of two columns of 1's of length N; the 1's are the coefficients of the constant term R_0) and all the other entries are $N/2$. A matrix $\boldsymbol{Q}$ for which

$$\boldsymbol{Q}^T \boldsymbol{Q} = \boldsymbol{D} \tag{5.17}$$

is said to be orthogonal. Therefore our $\boldsymbol{A}$ matrix is orthogonal. This greatly simplifies the fitting of the model which now reduces to

$$\boldsymbol{w} = \boldsymbol{D}^{-1} \boldsymbol{A}^T \boldsymbol{x} \tag{5.18}$$

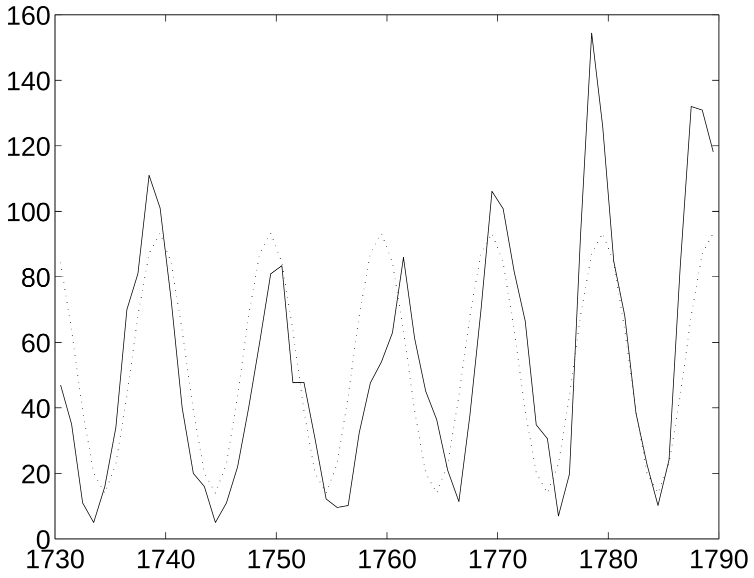

Figure 5.3: *Sunspot index (solid line) and prediction of it from a simple sinusoidal model (dotted line).*

which is simply a *projection* of the signal onto the basis matrix, with some pre-factor ($\boldsymbol{D}^{-1}$; remember the inverse of a diagonal matrix is simply the inverse of each of the diagonal terms, so this is easy to compute). Given that $\boldsymbol{w} = [a, b, R_0]^T$ we can see that, for example, a is computed by simply projecting the data onto the second column of the matrix $\boldsymbol{A}$, eg.

$$a = \frac{2}{N} \sum_{n=1}^{N} \cos(2\pi f t) x_t \tag{5.19}$$

Similarly,

$$b = \frac{2}{N} \sum_{n=1}^{N} \sin(2\pi f t) x_t \tag{5.20}$$

$$R_0 = \frac{1}{N} \sum_{n=1}^{N} x_t \tag{5.21}$$

We applied the simple sinusoidal model to a 'sunspot data set' as follows. We chose 60 samples between the years 1731 and 1790 (because there was a fairly steady mean level in this period). The sampling rate $F_s = 1$Year. This gives a base frequency of $f_b = 1/60$. We chose our frequency $f = pf_b$ with p=6; giving a complete cycle once every ten years. This gave rise to the following estimates; $R_0 = 53.64$, $a = 39.69$ and $b = -2.36$. The data and predictions are shown in Figure 5.3.

5.4 Fourier Series

We might consider that our signal consists of lots of periodic components in which case the *multiple sinusoidal model* would be more appropriate

$$x(t) = R_0 + \sum_{k=1}^{p} R_k \cos(2\pi f_k t + \Phi_k) + e_t \tag{5.22}$$

where there are p sinusoids with different frequencies and phases. In a discrete Fourier series there are $p = N/2$ such sinusoids having frequencies

$$f_k = \frac{kF_s}{N} \tag{5.23}$$

where $k = 1..N/2$ and F_s is the sampling frequency. Thus the frequencies range from F_s/N up to $F_s/2$. The Fourier series expansion of the signal $x(t)$ is

$$x(t) = R_0 + \sum_{k=1}^{N/2} R_k \cos(2\pi f_k t + \Phi_k) \tag{5.24}$$

Notice that there is no noise term. Because of the trig identity

$$\cos(A + B) = \cos A \cos B - \sin A \sin B \tag{5.25}$$

this can be written in the form

$$x(t) = a_0 + \sum_{k=1}^{N/2} a_k \cos(2\pi f_k t) + b_k \sin(2\pi f_k t) \tag{5.26}$$

where $a_k = R_k \cos(\Phi_k)$ and $b_k = -R_k \sin(\Phi_k)$. Alternatively, we have $R_k^2 = a_k^2 + b_k^2$ and $\Phi = \tan^{-1}(b_k/a_k)$. The signal at frequency f_k is known as the *kth harmonic*. Equivalently, we can write the nth sample as

$$x[n] = a_0 + \sum_{k=1}^{N/2} a_k \cos(2\pi f_k t[n]) + b_k \sin(2\pi f_k t[n]) \tag{5.27}$$

where $t[n] = nT_s$.

The important things to note about the sinusoids in a Fourier series are (i) the frequencies are *equally* spread out, (ii) there are $N/2$ of them where N is the number of samples, (iii) Given F_s and N the frequencies are *fixed*. Also, note that in the Fourier series 'model' there is *no noise*. The Fourier series aims to represent the data *perfectly* (which it can do due to the excessive number of basis functions)[2].

The Fourier coefficients can be computed by a generalisation of the process used to compute the coefficients in the simple sinusoidal model.

$$a_k = \frac{2}{N} \sum_{n=1}^{N} \cos(2\pi f_k t[n]) x[n] \tag{5.28}$$

[2] Statisticians would frown on fitting a model with N coefficients to N data points as the estimates will be very noisy; the Fourier series is a low bias (actually zero), high variance model. This underlines the fact that the Fourier methods are transforms rather than statistical models.

Similarly,

$$b_k = \frac{2}{N} \sum_{n=1}^{N} \sin(2\pi f_k t[n]) x[n] \tag{5.29}$$

$$a_0 = \frac{1}{N} \sum_{n=1}^{N} x[n] \tag{5.30}$$

These equations can be derived as follows. To find, for example, a_k, multiply both sides of equation 5.27 by $\cos(2\pi f_k t[n])$ and sum over n. Due to the orthogonality property of sinusoids (which still holds as all frequencies are integer multiples of a base frequency) all terms on the right go to zero except for the one involving a_k. This just leaves $a_k(N/2)$ on the right giving rise to the above formula.

5.4.1 Example

The plots on the right of Figure 5.4 show four components in a Fourier series expansion. The components have been ordered by amplitude. The plots on the left of the Figure show the corresponding Fourier approximation.

5.5 Fourier Transforms

Fourier series are representations of a signal by combinations of sinewaves of different magnitudes, frequencies and offsets (or *phases*). The magnitudes are given by the *Fourier coefficients*. These sinewaves can also be represented in terms of *complex exponentials* (see the appendix for a quick review of complex numbers); a representation which ultimately leads to algorithms for computing the Fourier coefficients; the Discrete Fourier Transform (DFT) and the Fast Fourier Transform (FFT).

5.5.1 Discrete Fourier Transform

Fourier series can be expressed in terms of complex exponentials. This representation leads to an efficient method for computing the coefficients. We can write the cosine terms as complex exponentials

$$a_k \cos(2\pi f_k t[n]) = a_k \frac{\exp(i2\pi f_k t[n]) + \exp(-i2\pi f_k t[n])}{2} \tag{5.31}$$

where $i^2 = -1$. Picture this as the addition of two vectors; one above the real axis and one below. Together they make a vector on the real axis which is then halved.

We can also write the sine terms as

$$b_k \sin(2\pi f_k t[n]) = b_k \frac{\exp(i2\pi f_k t[n]) - \exp(-i2\pi f_k t[n])}{2i} \tag{5.32}$$

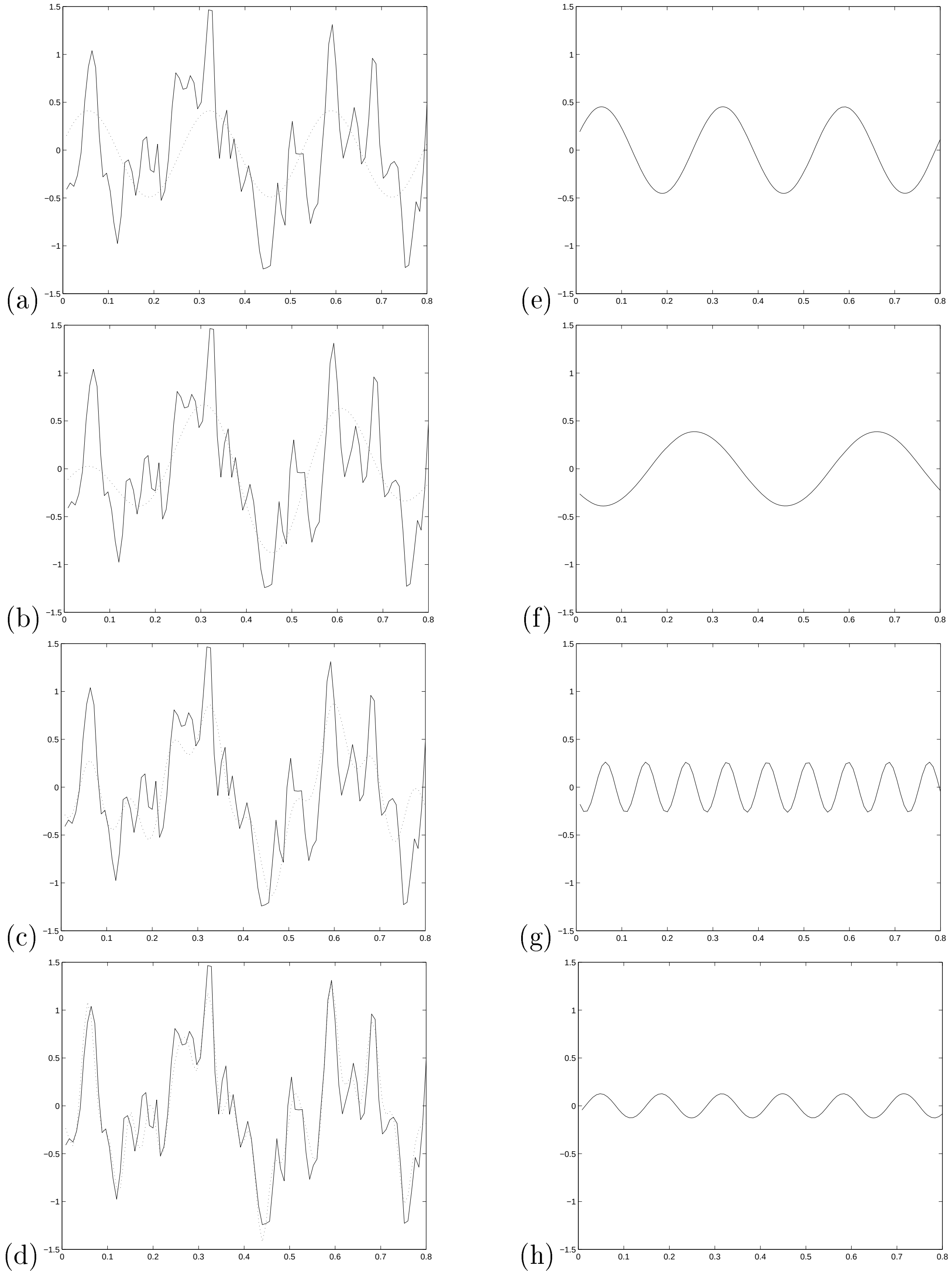

Figure 5.4: *Signal (solid line) and components of the Fourier series approximation* $\sum_{k=1}^{p} R_k cos(2\pi f_k + \Phi_k)$ *(dotted lines) with (a)* $p = 1$*, (b)* $p = 2$*, (c)* $p = 3$ *and (d)* $p = 11$ *where we have ordered the components according to amplitude. The corresponding individual terms are (e)* $R^2 = 0.205, f = 3.75$ *and* $\Phi = 0.437$*, (f)* $R^2 = 0.151$*,* $f = 2.5$ *and* $\Phi = 0.743$*, (g)* $R^2 = 0.069$*,* $f = 11.25$ *and* $\Phi = 0.751$ *and (h)* $R^2 = 0.016$*,* $f = 7.5$ *and* $\Phi = -0.350$*.*

Picture this as one vector above the real axis minus another vector below the real axis. This results in a purely imaginary (and positive) vector. The result is halved and then multiplied by the vector $\exp(3\pi/2)$ ($-i$, from multplying top and bottom by i) which provides a rotation to the real axis.

Adding them (and moving i to the numerator by multiplying b_k top and bottom by i) gives

$$\frac{1}{2}(a_k - b_k i)\exp(i2\pi f_k t[n]) + \frac{1}{2}(a_k + b_k i)\exp(-i2\pi f_k t[n]) \tag{5.33}$$

Note that a single term at frequency k has split into a complex combination (the coefficients are complex numbers) of a positive frequency term and a *negative frequency* term. Substituting the above result into equation 5.27 and noting that $f_k t[n] = kn/N$ we get

$$x[n] = a_0 + \frac{1}{2}\sum_{k=1}^{N/2}(a_k - b_k i)\exp(i2\pi kn/N) + \frac{1}{2}\sum_{k=1}^{N/2}(a_k + b_k i)\exp(-i2\pi kn/N) \tag{5.34}$$

If we now let

$$\tilde{X}(k) = \frac{N}{2}(a_k - b_k i) \tag{5.35}$$

and note that for real signals $\tilde{X}(-k) = \tilde{X}^*(k)$ (negative frequencies are reflections across the real plane, ie. conjugates) then the $(a_k + b_k i)$ terms are equivalent to $\tilde{X}(-k)$. Hence

$$x[n] = a_0 + \frac{1}{2N}\sum_{k=1}^{N/2}\tilde{X}(k)\exp(i2\pi kn/N) + \frac{1}{2N}\sum_{k=1}^{N/2}\tilde{X}(k)\exp(-i2\pi kn/N) \tag{5.36}$$

Now, because $\tilde{X}(N-k) = \tilde{X}(-k)$ (this can be shown by considering the Fourier transform of a signal $x[n]$ and using the decomposition $\exp(-i2\pi(N-k)n/N) = \exp(-i2\pi N/N)\exp(i2\pi kn/N)$ where the first term on the right is unity) we can write the second summation as

$$x[n] = a_0 + \frac{1}{2N}\sum_{k=1}^{N/2}\tilde{X}(k)\exp(i2\pi kn/N) + \frac{1}{2N}\sum_{k=N/2}^{N-1}\tilde{X}(k)\exp(-i2\pi(N-k)n/N) \tag{5.37}$$

Using the same exponential decomposition allows us to write

$$x[n] = a_0 + \frac{1}{N}\sum_{k=1}^{N-1}\tilde{X}(k)\exp(i2\pi kn/N) \tag{5.38}$$

If we now let $X(k+1) = \tilde{X}(k)$ then we can absorb the constant a_0 into the sum giving

$$x[n] = \frac{1}{N}\sum_{k=1}^{N}X(k)\exp(i2\pi(k-1)n/N) \tag{5.39}$$

which is known as the *Inverse Discrete Fourier Transform* (IDFT). The terms $X(k)$ are the *complex valued Fourier coefficients*. We have the relations

$$a_0 \;=\; Re\{X(1)\} \tag{5.40}$$

$$
\begin{aligned}
a_k &= \frac{2}{N} Re\{X(k+1)\} \\
b_k &= \frac{-2}{N} Im\{X(k+1)\}
\end{aligned}
$$

The complex valued Fourier coefficients can be computed by first noting the orthogonality relations

$$
\sum_{n=1}^{N} \exp(i2\pi(k-1)n/N) = \begin{array}{ll} N & k = 1, \pm(N+1), \pm(N+2) \\ 0 & otherwise \end{array} \tag{5.41}
$$

If we now multiply equation 5.39 by $\exp(-i2\pi ln/N)$, sum from 1 to N and re-arrange we get

$$
X(k) = \sum_{n=1}^{N} x(n) \exp(-i2\pi(k-1)n/N) \tag{5.42}
$$

which is the Discrete Fourier Transform (DFT).

5.5.2 The Fourier Matrix

If we write $X(k)$ as a vector $\boldsymbol{X} = [X(1), X(2), ..., X(N)]^T$ and the input signal as a vector $\boldsymbol{x} = [x(0), x(1), ..., x(N-1)]^T$ then the above equations can be written in matrix form as follows. The Inverse Discrete Fourier Transform is

$$
\boldsymbol{x} = \boldsymbol{F}\boldsymbol{X} \tag{5.43}
$$

where $\boldsymbol{F}$ is the *Fourier Matrix* and the Discrete Fourier Transform is

$$
\boldsymbol{X} = \boldsymbol{F}^{-1}\boldsymbol{x} \tag{5.44}
$$

If we let

$$
w_N = \exp(i2\pi/N) \tag{5.45}
$$

we can write the *Fourier matrix* as

$$
\boldsymbol{F}_N = \tfrac{1}{N} \begin{bmatrix} 1 & 1 & 1 & . & 1 \\ 1 & w_N & w_N^2 & . & w_N^{(N-1)} \\ 1 & w_N^2 & w_N^4 & . & w_N^{2(N-1)} \\ . & . & . & . & . \\ 1 & w_N^{(N-1)} & w_N^{2(N-1)} & . & w_N^{(N-1)^2} \end{bmatrix} \tag{5.46}
$$

which has elements[3]

$$
(\boldsymbol{F}_N)_{kn} = w_N^{(k-1)(n-1)} \tag{5.47}
$$

[3] We have re-indexed such that we now have $x(0)$ to $x(N-1)$. Hence we have $(n-1)$ instead of n.

Now, the inverse Fourier matrix is

$$\boldsymbol{F}_N^{-1} = \begin{bmatrix} 1 & 1 & 1 & . & 1 \\ 1 & w_N^{-1} & w_N^{-2} & . & w_N^{-(N-1)} \\ 1 & w_N^{-2} & w_N^{-4} & . & w_N^{-2(N-1)} \\ . & . & . & . & . \\ 1 & w_N^{-(N-1)} & w_N^{-2(N-1)} & . & w_N^{-(N-1)^2} \end{bmatrix} \tag{5.48}$$

where the elements are

$$(\boldsymbol{F}_N^{-1})_{kn} = w_N^{-(k-1)(n-1)} \tag{5.49}$$

In the *Fast Fourier Transform* (FFT) an N-dimensional matrix multiplication can be replaced by 2 M-dimensional multiplications, where $M = N/2$. This is because the exponential elements in the Fourier matrix have the key property

$$w_N^2 = w_M \tag{5.50}$$

eg. $\exp(i2\pi/64)^2 = \exp(i2\pi/32)$. Cooley and Tukey realised you could use this property as follows. If you split the IDFT

$$x_j = \sum_{k=0}^{N-1} w_N^{jk} X_k \tag{5.51}$$

into a summation of even parts and a summation of odd parts

$$x_j = \sum_{k=0}^{M-1} w_N^{2jk} X_{2k} + \sum_{k=0}^{M-1} w_N^{(2k+1)j} X_{2k+1} \tag{5.52}$$

then we can use the identity $w_N^2 = w_M$ to give

$$x_j = \sum_{k=0}^{M-1} w_M^{jk} X_{2k} + w_N^j \sum_{k=0}^{M-1} w_M^{kj} X_{2k+1} \tag{5.53}$$

which is the summation of two IDFTs of dimension M (a similar argument applies for the DFT).

This reduces the amount of computation by, approximately, a factor of 2. We can then replace each M-dimensional multiplication by an $M/2$-dimensional one, etc. FFTs require N to be a power of 2, because at the lowest level we have lots of 2-dimensional operations. For $N = 4096$ we get an overall speed-up by a factor of 680. For larger N the speed-ups are even greater; we are replacing N^2 multiplications by $\frac{N}{2} \log_2 N$.

5.6 Time-Frequency relations

Signals can be operated on in the *time domain* or in the *frequency domain*. We now explore the relations between them.

5.6.1 Power Spectral Density

The power in a signal is given by

$$P_x = \sum_{n=1}^{N} |x[n]|^2 \tag{5.54}$$

We now derive an expression for P_x in terms of the Fourier coefficients. If we note that $|x[n]|$ can also be written in its conjugate form (the conjugate form has the same magnitude; the phase is different but this does'nt matter as we're only interested in magnitude)

$$|x[n]| = \frac{1}{N} \sum_{k=1}^{N} X^*(k) \exp(-i2\pi(k-1)n/N) \tag{5.55}$$

then we can write the power as

$$P_x = \sum_{n=1}^{N} |x[n] \frac{1}{N} \sum_{k=1}^{N} X^*(k) \exp(-i2\pi(k-1)n/N)| \tag{5.56}$$

If we now change the order of the summations we get

$$P_x = \frac{1}{N} \sum_{k=1}^{N} |X^*(k) \sum_{n=1}^{N} x(n) \exp(-i2\pi(k-1)n/N)| \tag{5.57}$$

where the sum on the right is now equivalent to $X(k)$. Hence

$$P_x = \frac{1}{N} \sum_{k=1}^{N} |X(k)|^2 \tag{5.58}$$

We therefore have an equivalence between the power in the time domain and the power in the frequency domain which is known as *Parseval's relation*. The quantity

$$P_x(k) = |X(k)|^2 \tag{5.59}$$

is known as the *Power Spectral Density* (PSD).

5.6.2 Filtering

The filtering process

$$x[n] = \sum_{l=-\infty}^{\infty} x_1(l) x_2(n-l) \tag{5.60}$$

is also known as *convolution*

$$x[n] = x_1(n) * x_2(n) \tag{5.61}$$

We will now see how it is related to frequency domain operations. If we let $w = 2\pi(k-1)/N$, multiply both sides of the above equation by $\exp(-iwn)$ and sum over n the left hand side becomes the Fourier transform

$$X(w) = \sum_{n=-\infty}^{\infty} x[n] \exp(-iwn) \tag{5.62}$$

and the right hand side (RHS) is

$$\sum_{n=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} x_1(l)x_2(n-l)\exp(-iwn) \tag{5.63}$$

Now, we can re-write the exponential term as follows

$$\exp(-iwn) = \exp(-iw(n-l))\exp(-iwl) \tag{5.64}$$

Letting $n' = n - l$, we can write the RHS as

$$\sum_{l=-\infty}^{\infty} x_1(l)\exp(-iwl) \sum_{n'=-\infty}^{\infty} x_2(n')\exp(-iwn') = X_1(w)X_2(w) \tag{5.65}$$

Hence, the filtering operation is equivalent to

$$X(w) = X_1(w)X_2(w) \tag{5.66}$$

which means that convolution in the time domain is equivalent to multiplication in the frequency domain. This is known as the *convolution theorem.*

5.6.3 Autocovariance and Power Spectral Density

The autocovariance of a signal is given by

$$\sigma_{xx}(n) = \sum_{l=-\infty}^{\infty} x(l)x(l-n) \tag{5.67}$$

Using the same method that we used to prove the convolution theorem, but noting that the term on the right is $x(l-n)$ not $x(n-l)$ we can show that the RHS is equivalent to

$$X(w)X(-w) = |X(w)|^2 \tag{5.68}$$

which is the Power Spectral Density, $P_x(w)$. Combining this with what we get for the left hand side gives

$$P_x(w) = \sum_{n=-\infty}^{\infty} \sigma_{xx}(n)\exp(-iwn) \tag{5.69}$$

which means that the PSD is the Fourier Transform of the autocovariance. This is known as the *Wiener-Khintchine Theorem.* This is an important result. It means that the PSD can be estimated from the autocovariance and vice-versa. It also means that the PSD and the autocovariance contain the same information about the signal.

It is also worth noting that since both contain no information about the phase of a signal then the signal cannot be uniquely constructed from either. To do this we need to know the PSD *and* the *Phase spectrum* which is given by

$$\Phi(k) = \tan^{-1}(\frac{b_k}{a_k}) \tag{5.70}$$

where b_k and a_k are the real Fourier coefficients.

We also note that the Fourier transform of a symmetric function is real. This is because symmetric functions can be represented entirely by cosines, which are themselves symmetric; the sinewaves, which constitute the complex component of a Fourier series, are no longer necessary. Therefore, because the autocovariance is symmetric the PSD is real.

5.7 Spectral Estimation

5.7.1 The Periodogram

The periodogram of a signal x_t is a plot of the normalised power in the kth harmonic versus the frequency, f_k of the kth harmonic. It is calculated as

$$I(f_k) = \frac{N}{4\pi}(a_k^2 + b_k^2) \tag{5.71}$$

where a_k and b_k are the Fourier coefficients.

The periodogram is a low bias (actually unbiased) but high variance [4] estimate of the power at a given frequency. This is therefore a problem if the number of data points is small; the estimated spectrum will be very spiky.

To overcome this, a number of algorithms exist to smooth the periodogram ie. to reduce the variance. The Bartlett method, for example, takes an N-point sequence and subdivides it into K nonoverlapping segments and calculates $I(f_k)$ for each. The final periodogram is just the average over the K estimates. This results in a reduction in variance by a factor K at the cost of reduced spectral resolution (by a factor K).

The Welch method is similar but averages modified periodograms, the modification being a windowing of each segment of data. Also, the segments are allowed to overlap. For further details of this and other smoothing methods see Chapter 12 in Proakis and Manolakis [51]. This smoothing is necessary because at larger lags there are fewer data points, so the estimates of covariance are commensurately more unreliable.

5.7.2 Autocovariance methods

The PSD can be calculated from the autocovariance. However, as the sample autocovariance on short segments of data has a high variance then so will the resulting spectral estimates.

To overcome this a number of proposals have been made. The autocovariance function can first be smoothed and truncated by applying various smoothing windows,

[4] It is an *inconsistent* estimator, because the variance does'nt reduce to zero as the number of samples tends to infinity.

for example Tukey, Parzen, Hanning or Hamming windows. For further details see Chatfield p.114 [11] or Chapter 12 in Proakis and Manolakis [51].

5.7.3 Aliasing

Because of aliasing if we wish to uniquely identify frequencies up to BHz then we must sample the data at a frequency $f_s > 2B$Hz.

Alternatively, given a particular sample rate f_s, in order to uniquely identify frequencies up to $f_s/2$Hz (and not confuse them with higher frequencies) we must ensure that there is no signal at frequencies higher than $f_s/2$. This can be achieved by applying a Low-Pass Filter (LPF).

5.7.4 Filtering

There are two main classes of filters; IIR filters and FIR filters. Their names derive from how the filters respond to a single pulse of input, their so-called impulse response. The output of an Infinite Impulse Response (IIR) filter is fed-back to its input. The response to a single impulse is therefore a gradual decay which, though it may drop rapidly towards zero (no output), will never technically reach zero; hence the name IIR.

In Finite Impulse Response (FIR) filters the output is not fed-back to the input so if there is no subsequent input there will be no output. The output of an FIR filter ([51], page 620) is given by

$$y[n] = \sum_{k=0}^{p-1} b_k x[n-k] \tag{5.72}$$

where $x[n]$ is the original signal and b_k are the filter coefficients.

The simplest FIR filter is the (normalised) rectangular window which takes a moving average of a signal. This smooths the signal and therefore acts a low-pass filter. Longer windows cut down the range of frequencies that are passed.

Other examples of FIR filters are the Bartlett, Blackman, Hamming and Hanning windows shown in Figure 5.5. The curvier shape of the windows means their frequency characteristics are more sharply defined. See Chapter 8 in [51] for more details. FIR filters are also known as Moving Average (MA) models which we will encounter in the next lecture.

The output of an IIR filter is given by

$$y[n] = \sum_{k=0}^{p_a-1} a_k y[n-k] + \sum_{k=0}^{p_b-1} b_k x[n-k] \tag{5.73}$$

where the first term includes the feedback coefficients and the second term is an FIR

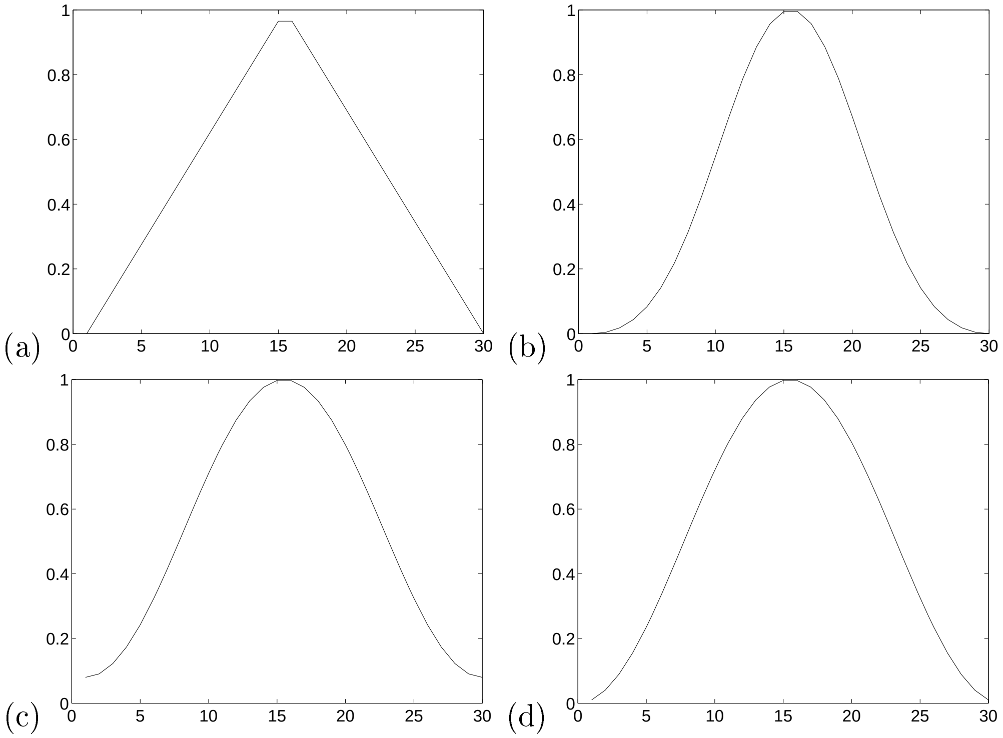

Figure 5.5: *Filter coefficients of (a) Bartlett (triangular), (b) Blackman, (c) Hamming and (d) Hanning windows for $p = 30$.*

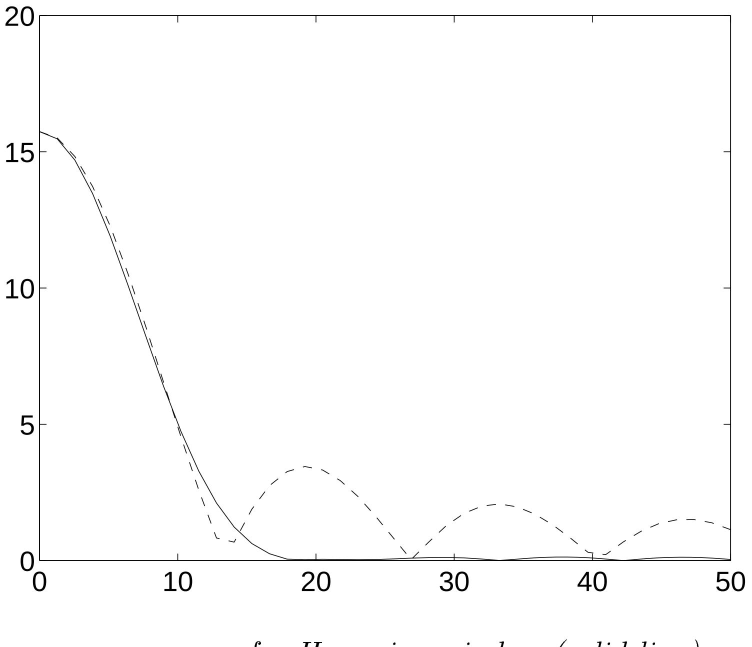

Figure 5.6: *Frequency response of a Hamming window (solid line) and a rectangular window (dotted line). The Hamming window cuts of the higher frequencies more sharply.*

model. This type of filter is also known as a Autoregressive Moving Average (ARMA) model (the first term being the Autoregressive (AR) part).

IIR filters can be designed by converting analog filters into the above IIR digital form. See [51] (section 8.3) for details. Examples of resulting IIR implementations are the Butterworth, Chebyshev, Elliptic and Bessel filters.