Power Spectrum Estimation

Only available in hardcover textbook (Ch.14)

Dr D. Laurenson

3rd November 2020

1 Introduction

A random signal can be characterised by its autocorrelation function, and equivalently, its spectral density. Previously, these have been defined for random processes. Now we will apply the theory to practical signals. The difference between the theory and the practical case arises from the unavailability of the probability density function, and in particular for autocorrelation, the joint probability density function between any two points in time. As this is not generally known for the majority of signals that we encounter in practical situations, although we may have models for the pdf, we resort to estimating spectral densities from realisations of the random process.

Note that this material is based on Chapter 14 of the course textbook, however this is only available in the hardcover edition. (The New International Edition has removed this chapter in order to be able to sell the textbook at a much reduced price). These notes aim to give sufficient coverage of the topic without access to the textbook.

Classical techniques

Spectrum estimation falls into two classes - classical, which do not depend upon the data being analysed, and modern techniques. The classical techniques are based on performing a Fourier analysis, and suffer from the problems associated with discrete Fourier transforms. However, they are used in many situations for analysis, so worthy of study.

There are two classical approaches: direct (via the Fourier transform of the data) and indirect (via the autocorrelation function).

Modern techniques

Modern techniques can be divided into two classes: parametric, where a model of the signal is fitted to the data; and non-parametric, where properties of the data are used to optimise the spectrum. Parametric techniques are limited by the quality of the model used; a poor model will result in misleading analysis. One example of a parametric technique is the analysis of speech in mobile phones. The model specifies the effect that the vocal tract has on speech, with the parameters describing the speech itself. With non-speech input, or speech that cannot be modelled well, the audio performance is poor. Non-parametric techniques do not require such a model. However they may require the signal being analysed to conform to certain conditions dictated by the analysis technique.

We will start by concentrating on the classical techniques for spectral estimation, moving on to non-parametric estimation at the end of this module. In this course we will not cover parametric spectral estimation.

1 Approaches

All approaches assume that the signal being analysed is wide-sense stationary and ergodic, at least for the first two moments. Where this is not the case, the signal must be analysed in sections sufficiently short that they can be considered to be stationary. An example of this would be analysis of speech which can only be considered to be stationary over relatively short time-frames, of the order of 20 ms. Analysing speech blocks larger than this results in averaging out spectral features that may be important, and thus the quality of the spectral estimate would be degraded.

Periodogram

This approach is a direct approach, analysing the frequency content of a realisation of the process. Starting with the discrete-time Fourier transform of a realisation, an expression for the spectral density can be obtained.

Assuming ergodicity, and an energy signal:

$$S_{xx}(f) = \left| \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi f n} \right|^2 \tag{1.11}$$

We will estimate this for a sampled frequency domain, with N samples:

$$S_{xx}\left(\frac{k}{N}\right) = \left| \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi kn/N} \right|^2$$

Thus, the periodogram can be expressed as the magnitude square of the discrete Fourier transform (DFT) of an infinite length block of the random input signal.

The definition for an ergodic power signal can similarly be defined:

$$S_{xx}\left(\frac{k}{N}\right) = \lim_{M\to\infty} \frac{1}{2M+1} \left| \sum_{n=-M}^{M} x(n) e^{-j2\pi kn/N} \right|^2$$

In these forms, provided that the signal is ergodic (which implies strict-sense stationarity), *and* we have an infinitely long time series, or the signal is periodic and we know the period in advance, then we can use these definitions. Of course, in practice, these conditions are not met, and compromises need to be made to the estimate. We will begin by examining the length of the record: We have access to only a finite number of samples of $x(n)$, thus we must window $\{x(n)\}$ to obtain $\{\tilde{x}(n)\}$.

$$\tilde{x}(n) = \begin{cases} x(n) & ;\ 0 \le n < N \\ 0 & \text{otherwise} \end{cases}$$

Then:

$$S_{\tilde{x}\tilde{x}}\left(\frac{k}{N}\right) = \left| \sum_{n=0}^{N-1} \tilde{x}(n) e^{-j2\pi kn/N} \right|^2 \tag{1.17}$$

The question that must be asked is how close is $S_{\tilde{x}\tilde{x}}\left(\frac{k}{N}\right)$ to $S_{xx}\left(\frac{k}{N}\right)$? We continue by defining a periodogram, P_{xx} as follows.

Random signals are generally power signals, thus:

$$P_{xx}\left(\frac{k}{N}\right) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} \right|^2 \tag{1.40}$$

Thus, a periodogram is the magnitude square of the discrete Fourier transform (DFT) of one block from a realisation of a random signal, normalised by the number of samples in that block.

This latter form of periodogram is the one that is used for practical situations as we are limited to dealing with finite length records of data. The notation $P_{xx}(\cdot)$ is used to distinguish between a periodogram estimate of a spectral density, and the spectral density itself. In forming this estimate, we have initially assumed that the window is a rectangular window, i.e.

$$w(n) = \begin{cases} 1 & 0 \le n < N \\ 0 & \text{otherwise} \end{cases}$$

This will introduce similar effects to those we observed in module 2.

Zero padding

We can reduce the spacing of samples of $P_{xx}(\cdot)$ to better approximate the continuous spectrum $P_{xx}(f)$:

$$P_{xx}\left(\frac{k}{L}\right) = \frac{1}{L}\left|\sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/L}\right|^2 \tag{1.41}$$

where $N < L$. This is equivalent to an L-point DFT of the length L sequence:

$$\{x(0), x(1), \ldots, x(N-1), 0, 0, \ldots, 0\}$$

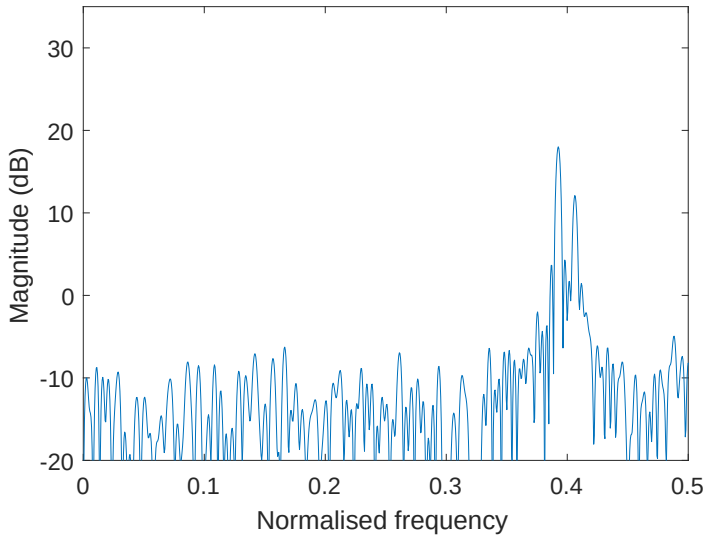
The appending of zeros to the record is called zero-padding. When this longer record is used as an input to an FFT algorithm (an efficient method of computing a DFT), then the frequency domain samples are more closely spaced than for the non-zero padded case. The overall spectral shape remains unchanged as the window that has been applied to the data has not been altered. (Remember, as the window is applied using multiplication in the time domain, this corresponds to convolving the frequency response of the signal with the frequency response of the window. Adding additional zeros does not change the window shape in time or frequency, so the underlying spectral shape remains the same - it is only the spacing of samples in frequency that is different).

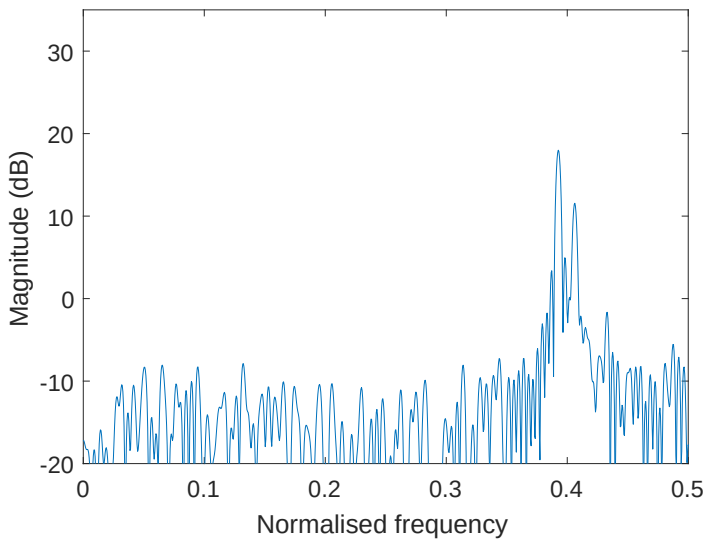
Quality of estimate

The periodogram is not a good estimate of spectral density because:

- it has a very high variance; and
- it suffers from the effects of windowing.

The high variance means that one estimate of the spectrum taken from one realisation will differ significantly from another estimate of the same spectrum taken from a different realisation, or the same realisation at a different point in time. This is a result of the underlying process being random, and us (a) assuming ergodicity and (b) considering only one section of a time-series. Consider, for example, two periodogram estimates of the same random process:





Clearly, these two example periodograms are different, although similarities can be observed. However, for any given sample point, k/L, the two estimates can differ significantly. This is the practical implication of a high variance in the estimate. In order to improve the variance of the estimate, averaging can be used. This leads to the first improvement on the periodogram approach.

1 Periodogram Approaches

1.1 Bartlett Method

In this method, assuming stationarity and ergodicity, a long time sample of the random process is divided into K consecutive blocks, each of length M. Periodograms are taken of each block, and then the resulting periodograms are averaged together. The result is the Bartlett power spectrum estimate. If the blocks of data are independent, then this estimate has a variance that is a factor of K times smaller than a single periodogram. Where the blocks are dependent, the reduction in variance is smaller than this.

Bartlett method

$$P_{xx}^{\mathrm{B}}(f) = \frac{1}{K}\sum_{i=0}^{K-1}\left\{\frac{1}{M}\left|\sum_{n=0}^{M-1} x(n+iM)e^{-j2\pi f n}\right|^2\right\}$$

The number of blocks, K, that can be obtained from a record of length N is given by

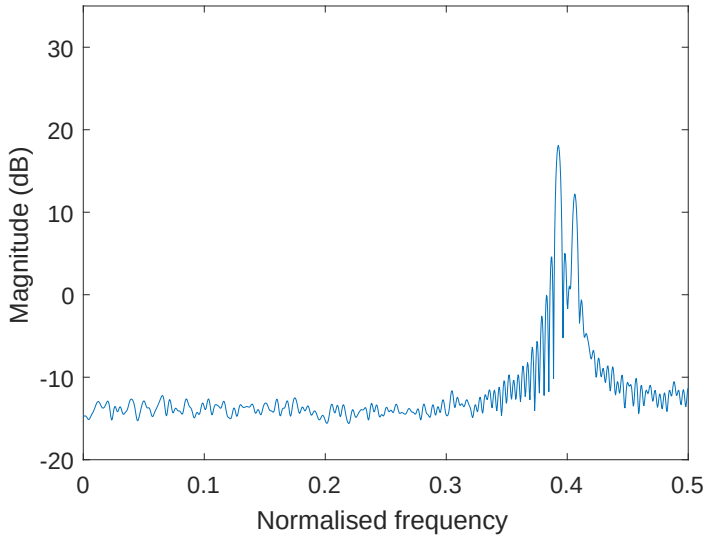
$$K = \left\lfloor \frac{N}{M} \right\rfloor$$

where $\lfloor\cdot\rfloor$ represents the floor operation, resulting in the highest integer that is not greater than the argument. (In other words, it rounds down).

The periodogram is defined in terms of f, but clearly in practice this will be sampled when using a DFT, or FFT to carry out the processing task. The figures shown below are produced with the following parameters:

Parameter	Value
Block length, M	256
FFT size	2048
Record length, N	10000

Thus, $K = 39$ in this instance.



Here it is evident that the variance of the estimate has been reduced by averaging the 39 periodograms that have been computed. This has helped to highlight the spectral components that are present in the signal. Note, though, that significant sidelobes are also clearly visible in the resulting transform close to the peaks representing the spectral components of the signal.

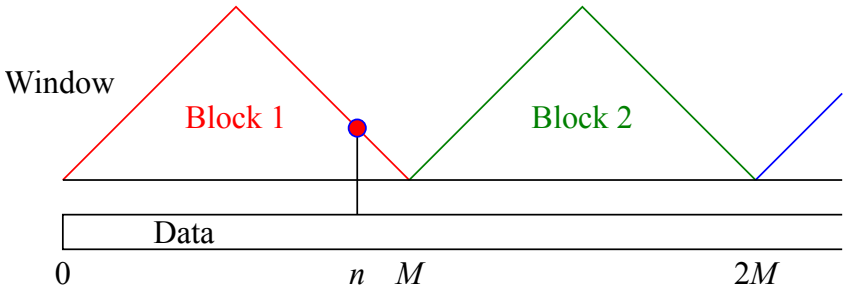
Lectures on the discrete Fourier transform covered the effect of windowing. The high sidelobes are a result of the implicit rectangular window that is applied in this case. We can improve the spectral representation by applying a shaped window. The next technique combines this with overlapping sections.

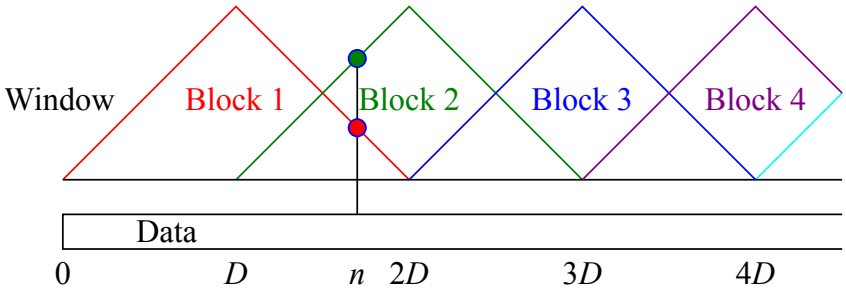
1.2 Welch method

This method applies a window function to each block of data prior to calculating the DFT. This improves the spectral characteristics of the signal by scaling the amplitude of the input samples prior to performing the transform. The window reduces the amplitude of samples near to the ends of the block of data on which the DFT is operating more than it does samples closer to the middle of the block. A second modification to the Bartlett method is rather than force

the blocks to be taken consecutively from the input record, to allow the blocks of data to be drawn from overlapping sections. The motivation for this is that when blocks are overlapped, samples that had been attenuated strongly by the window in one block are likely to be subject to much less attenuation in the next. Thus all of the input samples contribute approximately equally to the final result. A 50% overlap is common.

The effect of overlapping blocks is shown below for a triangular window. Without any overlap, the sample at n is heavily attenuated. With a 50% overlap, it is more heavily weighted in the subesquent block.





$$P_{xx}^{\mathrm{W}}(f) = \frac{1}{L}\sum_{i=0}^{L-1}\left\{\frac{1}{MU}\left|\sum_{n=0}^{M-1} x(n+iD)w(n)e^{-j2\pi f n}\right|^2\right\}$$

U denotes the power of the window:

$$U = \frac{1}{M}\sum_{n=0}^{M-1}(w(n))^2$$

The number of periodograms can be calculated from:

$$L = \left\lfloor \frac{N-M}{D} \right\rfloor + 1$$

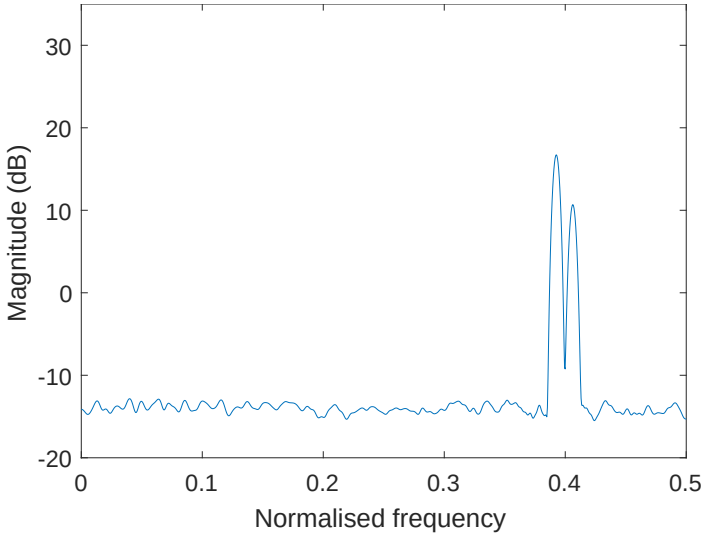
where N is the length of the sample, and $M - D$ samples overlap with the previous block.

Given the parameters that have been used, and assuming a 50% overlap, then $D = 128$, and the number of periodograms that can be averaged together is $L = 77$. This is almost double the number of periodograms used in the Bartlett Periodogram approach.

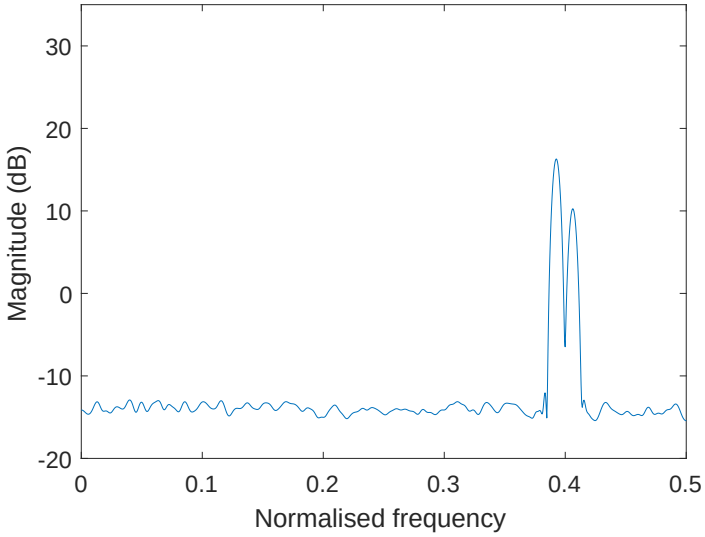
As the window function scales samples in the time domain, the energy of the scaled block of data that is the input to the DFT is lower than that of the original input data block. Thus, without any additional scaling term, an estimate of the signal energy in one Welch periodogram would be lower than the actual energy of the input. This is the reason for the scaling term, U. It represents the power of the window function, so normalising by this corrects this problem.

Different windows can be applied, with differing widths of main lobes and heights of sidelobes. The figures below compare the performance of three different windows for the same sample of the random process.

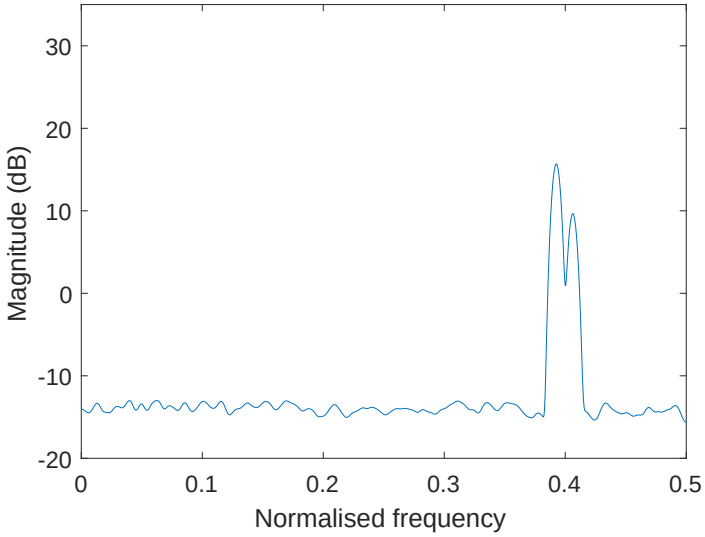
Welch method - Hamming window



Welch method - Hanning window



Welch method - Blackman window



The Hamming and Hanning windows, both being raised cosine windows, have similar main lobe widths. Recalling the lectures on the DFT, the shape of the sidelobe roll-off is different between the two windows, however this is not evident in this example because the noise floor is relatively high. Application of the Blackman window is therefore of limited use as the lower sidelobe levels do not improve the spectral estimate of frequencies that are not within the

spectral peaks. In fact, in this case, the use of the Blackman window actually degrades performance as the wider main lobes causes the main lobes for the two spectral components to overlap.

This illustrates the importance of selecting a window that meets the specified criteria for the maximum sidelobe height, but does not exceed this by a significant amount. For this problem, it is the noise level that controls the design criteria as the noise floor is approximately 30 dB lower than the maximum peak. Thus a window that has a maximum sidelobe height below -30 dB would be suitable, and of these windows, the one with the best resolution should be selected.

1 Blackman and Tukey

An alternative method of producing a spectral estimation from an average of periodograms is to use the Wiener-Khintchine relationship to compute the spectrum from an estimate of the autocorrelation function. Thus, instead of estimating multiple periodograms and averaging, a single autocorrelation estimate is computed, which may be done using an efficient convolution algorithm as discussed in module 3, and then this is the input into a discrete Fourier transform. This method was developed by R.B. Blackman and J.W. Tukey in 1958, and is either called the Correlogram, indicating its relationship to the Periodogram, or known as the Blackman-Tukey method.

First compute

$$r_{xx}(m) = \frac{1}{N}\sum_{n=m}^{N-1} x(n)x(n-m)$$

for values of $m < M$, noting that $r_{xx}(-m) = r_{xx}(m)$ for a real sequence. The reason that only M values are computed is that as M approaches N the variance, and hence unreliability of the autocorrelation estimate increases. The other practical reason is that the signal will need to be transformed by a DFT (or rather its more efficient implementation, the FFT). Thus, the number of samples required should be significantly fewer than the number of samples in the entire data record. By implication, we assume that $r_{xx}(m) = 0$ for all values of $|m| \geq M$.

A power spectral estimate is then given by:

$$P_{xx}(f) = \sum_{m=-(M-1)}^{M-1} r_{xx}(m)e^{-j2\pi fm}$$

The assumption that $r_{xx}(m) = 0$ for all values of $|m| \geq M$ is equivalent to applying a rectangular window to the autocorrelation. As we have previously discussed, this results in a distortion of the frequency response caused by the window function. As previously, a window should be applied to control the effects of this distortion:

$$P_{xx}^{\mathrm{BT}}(f) = \sum_{m=-(M-1)}^{M-1} r_{xx}(m)w(m)e^{-j2\pi fm}$$

Note that as for the periodogram case, the result should be normalised by the power of the window.

A window is required in this case, not only to improve the quality of the resulting spectral density, but also because otherwise the resulting estimate may not form a valid spectral density. We know that $S_{xx}(f) \geq 0$, thus any estimate needs to also meet this requirement

As the window is convolved with the power spectrum of the sample of the process, should a window with negative values in its transform be used, there is a possibility that the estimate will have negative values. This is clearly a poor estimate as power spectra are by definition non-negative. The windows applied so far, including the implicit rectangular window, all have transforms containing negative values, therefore they could all result in estimates where, for one or more frequency terms, $P_{XX}(f)$ may be less than zero.

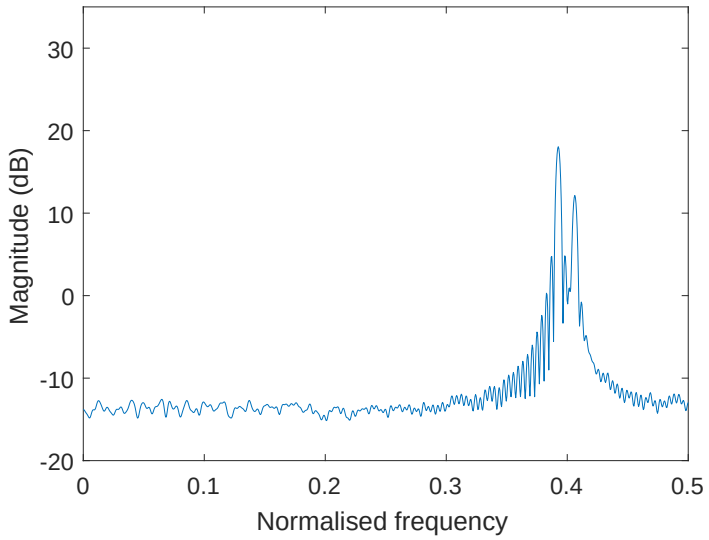
One method of producing a window that has no negative components in its frequency transform is to take an existing window, such as the Rectangular window, and square its frequency transform. This is equivalent to convolving the time domain response with itself. The result, in this case, is a triangular window, which has no negative components in its frequency transform. As there are no negative components, when it is convolved with the frequency transform of the realisation, the result will also be non-negative. Suitable transforms constructed using this technique include:

Window	**Formula**
Triangular	$w(n) = 1 - \frac{\lvert n-M\rvert}{M}$
Bartlett	$w(n) = 1 - \frac{\lvert n-M\rvert}{M+1}$
Parzen	$w(n) = \begin{cases} 1 - 6\left(\frac{\lvert n-M\rvert}{M+\frac{1}{2}}\right)^2 + 6\left(\frac{\lvert n-M\rvert}{M+\frac{1}{2}}\right)^3 \\ \qquad ;\lvert n-M\rvert \leq M/2 \\ 2\left(1 - \frac{\lvert n-M\rvert}{M+\frac{1}{2}}\right)^3 \;;M/2 < \lvert n-M\rvert \leq M \end{cases}$
Bohman	$w(n) = \left(1 - \left\lvert\frac{n}{M} - 1\right\rvert\right)\cos\left(\pi\left\lvert\frac{n}{M} - 1\right\rvert\right) + \frac{1}{\pi}\sin\left(\pi\left\lvert\frac{n}{M} - 1\right\rvert\right)$

where $0 \leq n < 2M + 1$.

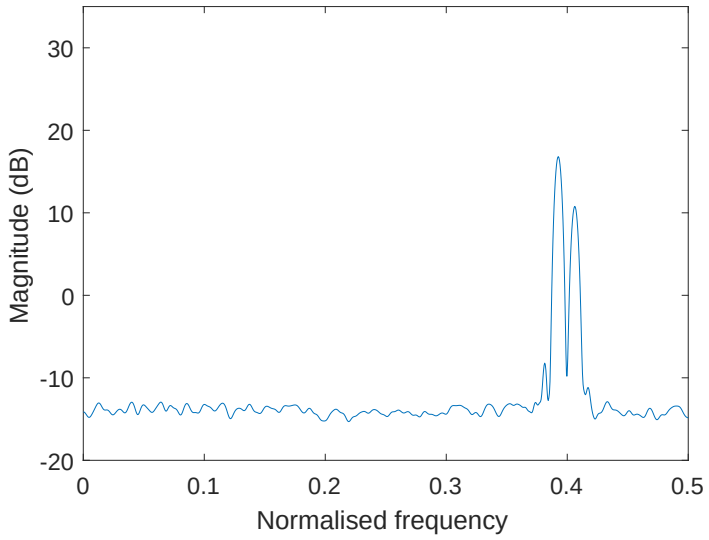
The triangular and the Bartlett windows, because of their similarity, have a very similar effect in the frequency domain. As they are strongly related to the implicit rectangular window, they have similar properties of relatively high sidelobes, but a narrow main lobe. The Parzen and Bohman windows reduce the height of the sidelobes at the expense of a wider main lobe.

Blackman and Tukey triangular



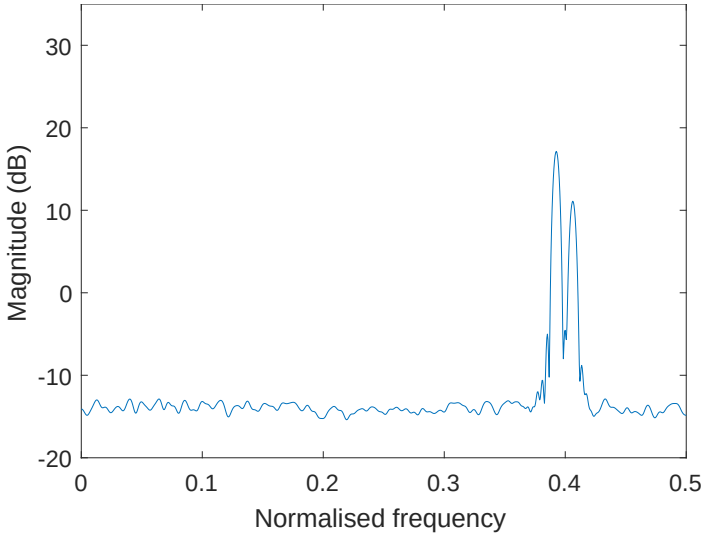
It is clear that the sidelobe height is significant, with a transform that is very similar to the Bartlett periodogram estimate from earlier in this module. This should not be surprising as the formation of the Bartlett periodogram involves taking the magnitude squared of the DFT output. This is equivalent to a convolution operation in the time domain. Thus the periodogram estimate involves the convolution of two signals with rectangular windows, which is comparable to a triangular window applied to an autocorrelation.

Blackman and Tukey Parzen



The Parzen window is the result of convolving a triangular window with itself. The convolution operation, being related to a magnitude square operation in frequency, results in sidelobe heights being reduced by a factor of 2 (in dBs), but also a doubling of the main lobe width. Thus, as a rectangular window has a maximum sidelobe height of -13 dB, a triangular window has a maximum sidelobe height of -26 dB, and the Parzen window, a maximum height of -52 dB. Other windows could be generated by convolving the Parzen window with a triangular window, or itself, to reduce the sidelobe height even further. These windows are known as B-spline windows, and are formed as k-fold convolutions of the rectangular window, where the triangular window is found from $k = 2$, and the Parzen from $k = 4$.

Blackman and Tukey Bohman



The Bohman window is derived from a cosine window, which is one positive half cycle of a cosine wave. Like the Triangular and Parzen windows, the Bohman window is formed by convolving the cosine window with itself. Thus, again the Bohman window is guaranteed to have a non-negative transform, and is suitable for use in the Blackman and Tukey method.

The Blackman-Tukey method easily extends to computation of cross-spectral densities. All that is required is that the autocorrelation be substituted by a cross-correlation, i.e.:

$$P_{yx}^{\mathrm{BT}}(f) = \sum_{m=-(M-1)}^{M-1} r_{yx}(m) w(m) e^{-j2\pi f m}$$

The rest of the process is identical. Thus, it is a very useful approach to use when determining the transfer function of an unknown linear time-invariant system as described in module 4 where:

$$H(\omega) = \frac{S_{yx}(\omega)}{S_{xx}(\omega)}$$

It is worth noting that the periodogram approaches can also be used to estimate cross-spectral densities as:

$$S_{yx}\left(\frac{k}{N}\right) = \left(\sum_{n=-\infty}^{\infty} y(n) e^{-j2\pi kn/N}\right)\left(\sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi kn/N}\right)^{*}$$

Power Spectrum Estimation

Only available in hardcover textbook (Ch.14)

Dr D. Laurenson

3rd November 2020

1 Selection of Windows and transform lengths

When selecting a window, the question should be asked: "What is the required peak sidelobe level"? This is problem dependent, but the following should be considered when attempting to answer the question:

- What is the dynamic range of input frequencies anticipated?
- What level of accuracy is required in the final result?
- What is the peak signal power compared to the noise floor?

Very often, these questions have the same answer. When selecting the window, these answers define what the peak sidelobe level should be.

The resolution of the window is the minimum frequency spacing between two components that allows them to be identified as being separate. If this separation also corresponds to the peak sidelobe level (which is likely), then the peak sidelobe level limits the smallest signal that can be guaranteed to be observed. The analogue system is generally designed to meet the system requirements, so this should correspond to the smallest signal that should be observable at the given resolution. Therefore, the peak sidelobe level should be at, or below, the system noise floor.

The window length should be chosen once the peak sidelobe level has been fixed. The length specifies the resolution of the system – that is the minimum spacing between two frequency components that still allows them to be distinguished.

- High resolution (small frequency separation) implies long window lengths.
- Long window lengths require long sequences of data.
- Care must be taken that the input remains stationary over the length of data sequence being used.
- In this course, resolution is specified as the -3 dB bandwidth. Thus for a rectangular window, the resolution is $\frac{0.89}{M}$ Hz, or $\frac{1.78\pi}{M}$ rad/s.

Other definitions of resolution are possible, such as the equivalent noise bandwidth, which specifies the bandwidth of an ideal "brickwall" filter with the same output power as the practical one. The particular definition used depends upon the application, however the -3 dB bandwidth is more common.

Up to this point we have explored two classical approaches to producing a spectral estimate - the periodogram approach and the Blackman and Tukey (or correlogram) approach. Both produced good estimates of the spectral density, so the question then needs to be asked, which is better? We will explore this question in terms of the computational expense of obtaining it. The hardcover version of the textbook also discusses the quality of the estimate, however this is out of scope for this course.

1 Computational complexity

A technique must be efficient in terms of computation, particularly if it is to be used in a system involving real-time analysis. Techniques with a low computational complexity can be executed more quickly, and run more often, than those with a high complexity. Of the common tasks for a signal processing algorithm, multiplication is generally used as the metric to gauge complexity. Division is used only sparingly as it is computationally expensive, so is not a good measure of how complex an algorithm is. In order to be fair to the different algorithms, efficiency should be sought at all stages of the computation, therefore the fast Fourier transform (FFT) is used to perform all DFT calculations. Assuming that a complex multiply is a single operation, an M-point FFT can be performed in approximately $\frac{M}{2}(\log_2(M) - 1)$ complex multiplications, which is deemed to be its complexity.

Note that in the textbook, not all steps are considered. For example, the division by K, L or M involved in averaging of periodograms is neglected, as are any additions or subtractions. In practice, division by a constant can more efficiently be computed as multiplication by its reciprocal. As this stage is only executed once at the end of the process, its impact on overall complexity is negligible, so is not included in the measure. When working with complexity in practice, small adjustments are often neglected. Thus the complexity of an FFT may equally be expressed as $\frac{M}{2}\log_2 M$. This is the approximation used in this analysis. Despite these approximations, the results are valid for comparing computational complexity of the three techniques, as the same approximations are used for all three.

To compare the complexity of the three approaches, the comparison must be fair. To achieve this, the approaches are compared in terms of achieving a particular resolution Δf. For techniques, such as the Welch periodogram, where a window is used, the length of the transform needs to be larger to achieve the same resolution as a technique such as the Bartlett, however the quality of the estimate is better. A comparison of the techniques is presented below, assuming that a triangular window is used for both the Welch and Blackman & Tukey approaches. This ensures as fair a comparison as possible:

Estimate	Complexity
Bartlett	$\frac{N}{2}\log_2 M = \frac{N}{2}\log_2 \frac{0.9}{\Delta f}$
Welch	$2N + N\log_2 M = N\log_2 \frac{5.12}{\Delta f}$
Blackman & Tukey	$\frac{N}{M}(M\log_2(2M)) = N\log_2 \frac{1.28}{\Delta f}$

The Welch method requires more computation than the other two, but all are of a similar complexity.

The complexities are computed as follows:

Bartlett The FFT length, $M = 0.9/\Delta f$. From a record of N samples, N/M periodograms can be found. Each periodogram requires $\frac{M}{2}log_2 M$ complex multiplies, resulting in a final complexity of $\frac{N}{2}\log_2 \frac{0.9}{\Delta f}$.

Welch In this case, the FFT length is $M = 1.28/\Delta f$, and due to the 50% overlap, $2N/M$ periodograms are used. In addition, the multiplication with the window is an additional step that should be considered, adding $2N$ multiplies to the process. Combining these results in the complexity being given by $N\log_2 \frac{5.12}{\Delta f}$.

Blackman & Tukey This process is detailed below showing how the computation is split into blocks, and computed using the FFT. In computing the complexity the resolution is selected when considering use of a triangular window, thus the length of the autocorrelation required is $1.28/\Delta f$.

The Blackman and Tukey approach is an attractive one as it has the benefits of applying a window without the computational complexity overhead of the Welch approach, being only a modest increase in complexity over the Bartlett approach.

Interestingly, the Blackman and Tukey complexity analysis assumes that the correlation, $r_{xx}(m)$, is computed using FFTs. As shown in module 3, this is because a convolution (which is similar in computation to a correlation) in the time domain is multiplication in the frequency domain. As the FFTs are very efficient, a significant reduction in computations can be achieved for even modest correlation lengths by taking this approach when compared to direct calculation of the correlation.

1.1 Calculation of Blackman and Tukey

Calculation of Blackman and Tukey

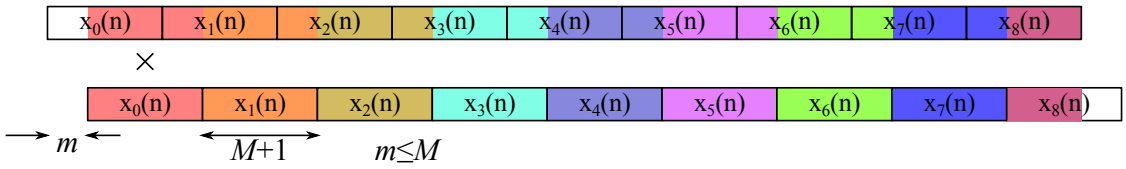
To calculate a Blackman and Tukey estimate, the estimation of $r_{xx}(m)$ is required. For real valued signals, $r_{xx}(m) = r_{xx}(-m)$, so only non-negative displacements of m are required. Assuming a maximum displacement of M, then only $M + 1$ outputs are required.

Using the equation to compute $r_{xy}(m)$ from previously, we can write:

$$r_{xx}(m) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)X^*(k)e^{j2\pi km/N}$$

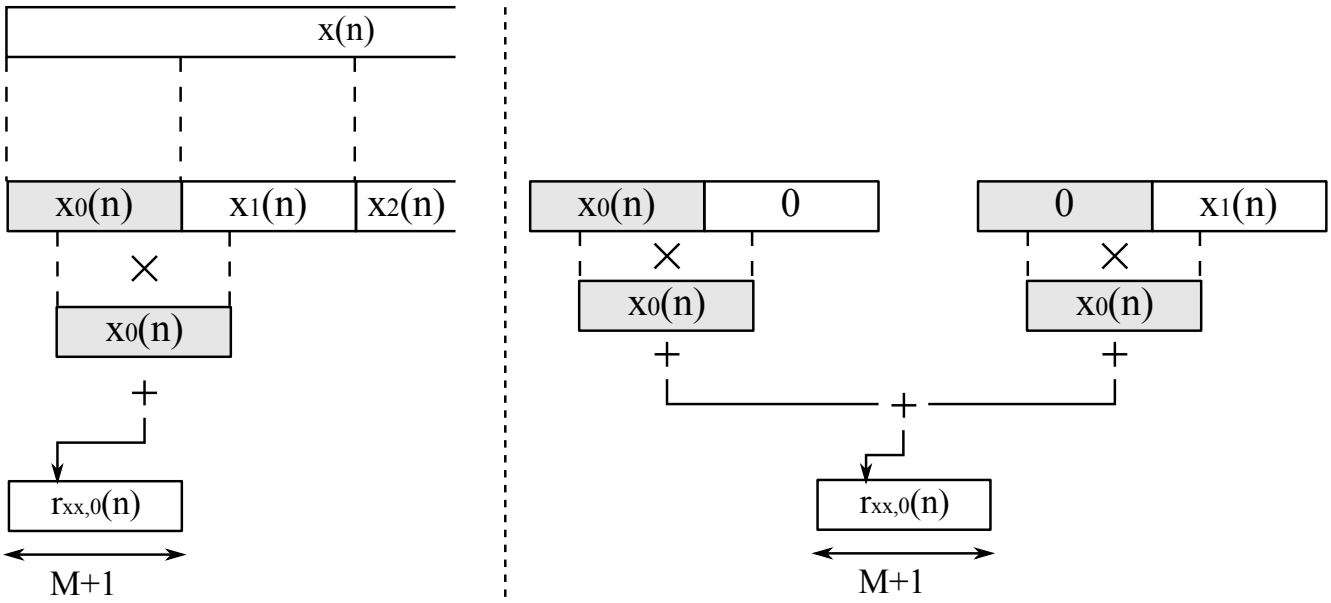
Whilst this is mathematically possible, computing an FFT of a very long sequence carries a significant computational burden. Recalling the need for zero padding from module 3, the FFT would need to be at least of length $N+M$ to allow for the maximum value of m. As only $M+1$ outputs are needed, for a large value of N the complexity of this approach may be higher that of direct computation, which has a complexity of NM.

Instead of computing the correlation by applying this equation to the complete data input, $x(n)$, it can be split into blocks, $x_i(n)$, of length $M+1$, and the correlation between blocks be computed and combined to achieve the same result. In the time domain, the process would be represented as below.



From this, it is clear that every block, $x_i(n)$, except the final one, is correlated with itself and its adjacent block, $x_{i+1}(n)$. By splitting this computation up and computing the correlation terms corresponding to $x_i(n)$ correlated with itself, and then computing the terms corresponding to $x_i(n)$ with $x_{i+1}(n)$, and then summing the result, an efficient method of computing the whole sequence can be developed.

As correlation can be efficiently computed using an FFT, the process will operate on FFTs of each block. The displacement m satisfies $|m| \leq M$, which suggests that the length of the FFT must be at least $2M+1$ for a block size of $M+1$ values. Using a block size of $M+1$, the conceptual process is given in the following figure:



The correlation for a single block, $x_i(n)$, is computed as the sum of an autocorrelation with itself, as well as the next block, $x_{i+1}(n)$. As we need to compute the correlation not only between blocks $x_i(n)$ and $x_i(n)$ but also between blocks $x_i(n)$ and $x_{i+1}(n)$, it is very useful to be able to shift a block of data in the time domain by computation in the frequency domain. If the shift is equivalent to half of an FFT length, then the computation becomes simple, as it is multiplication by $(-1)^k$. Thus the FFT size chosen is $2M+2$. When processed by the FFT, the data is zero padded at the end of the block. When the block is multiplied by $(-1)^k$ in the frequency domain, this shifts the data to the second half of the time domain data block, effectively zero-padding for the first half instead of the second half. In the diagram above, this is the structure of the block $x_1(n)$ in the right hand portion of the figure.

The aim of the approach is to avoid high complexity computation wherever possible. The FFT is the most complex operation in the process, so the minimum number of FFT computations should be made. The algorithm therefore computes the FFT of each block of data, $x_i(n)$, once only. This leads to the following approach:

1. Starting with $i=0$, select the first block of $M+1$ data points from $x(n)$ such that

$$x_i(n) = x(n+i(M+1)) \,; 0 \leq n \leq M$$

2. Compute $X_i(k)$ using an FFT of length $2M+2$

3. Find $X_i^*(k)X_i(k)$ and store. In the time domain this is equivalent to correlating the length $M+1$ sequence $x_i(n)$ with itself. Note that for displacements other than $m=0$, the resulting correlation is not the same as computing the correlation of this block of data with the original sequence. This is because the final m samples of $x_i(m)$ should be multiplied with the first m samples of $x_{i+1}(m)$, but are instead are multiplied with the zero padded values. The next three steps compute these missing values.

4. Increment i, and select the next block of data:

$$x_i(n) = x(n + i(M+1)) \; ; 0 \leq n \leq M$$

5. Compute $X_i(k)$ using an FFT of length $2M+2$.

6. In order to calculate the missing values, the block $x_i(n)$ should be zero-padded at the front so that the final m values of $x_{i-1}(n)$ can be multiplied by the first m values of $x_i(n)$. However, the FFT in step 5 assumed zero padding at the end.

   In order to rectify this, $x_i(n)$ should be shifted by $M+1$ values. A shift in time is a multiplication in the frequency domain. Thus, the FFT of $\{0, 0, \ldots, 0, x_i(0), x_i(1), \ldots, x_i(M)\}$ is equal to $e^{j2\pi k(M+1)/(2M+2)} X_i(k)$. However, $e^{j\pi k} = (-1)^k$, which is trivial to compute.

   So, in order to compute the missing results, we find $(-1)^k X^*_{i-1}(k) X_i(k)$ and add this to the result of step 3. This step is equivalent to correlating $x_{i-1}(n)$ with the length $2M+2$ sequence $\{0, 0, \ldots, 0, x_i(0), x_i(1), \ldots, x_i(M)\}$.

7. Repeat steps 3 to 6, adding all of the results together until all data has been used.

8. Inverse FFT the final accumulated result (using a length $2M+2$ IFFT).

9. Use only the first $M+1$ values of the result to determine $r_{xx}(m)$, where $0 \leq m \leq M$ and the symmetry property to determine $r_{xx}(m)$ for $-M \leq m < 0$.

10. Apply a window to $r_{xx}(m)$ and then estimate the spectral density using an FFT with a length that is at least $2M+1$ points. Although it may seem like the last three steps are redundant, without them, the result is subject to an implicit rectangular window. As time domain windows cannot be applied in the frequency domain, the time domain correlation $r_{xx}(m)$ needs to be obtained in order to apply the window.

In conclusion, the ability to control leakage in the Blackman and Tukey approach is equivalent to that of the Welch approach, yet its complexity is roughly equivalent to the Bartlett approach. This suggests that this is the best of the Classical methods to determine a spectral density.

1 Filter bank realisation

At the core of the periodogram approach is the application of the DFT. Previously we have considered windowing as applied to the input signal, thus its frequency response is convolved with the input. By re-arrangement of the terms, it can instead be viewed as being convolved with the ideal frequency response of each DFT term.

An FIR filter can be written as:

$$y(i) = \sum_{n=-\infty}^{\infty} x(n)h(i-n)$$

Starting with the definition of the power spectral density estimate computed by a periodogram, the power spectrum estimate can be recast as an FIR filter. Given a rectangular window:

$$P_{XX}(f) = \frac{1}{N}\left|\sum_{n=0}^{N-1} x(n)e^{-j2\pi fn}\right|^2$$

for any particular frequency, f_k, the summation is a form of FIR filter where

$$h_k(n) = e^{j2\pi f_k n}$$

The output can then be sampled every N^{th} sample, and squared.[1]

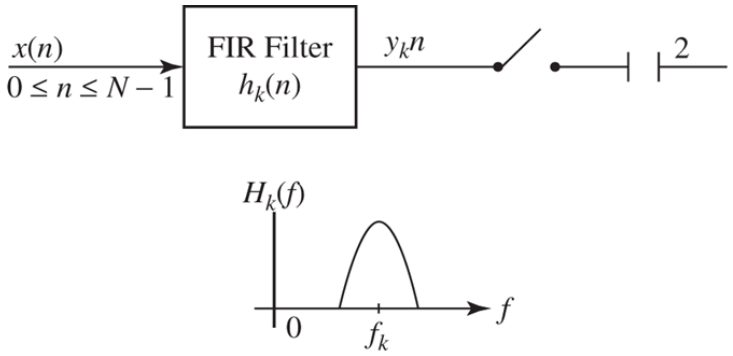


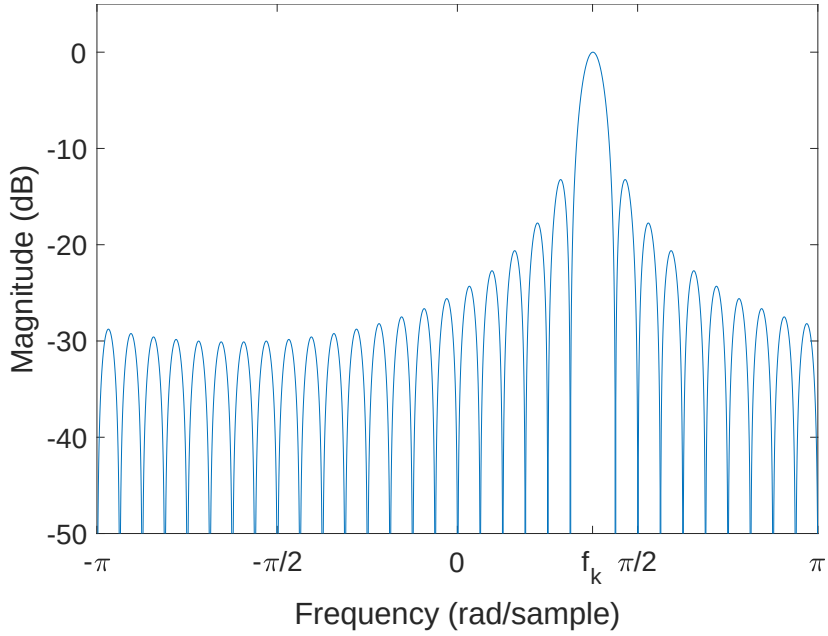
Figure 14.4.1 Measurement of signal power at a frequency in the vicinity of f_k.

The frequency transform of the filter is given by:

$$\begin{aligned} H_k(f) &= \sum_{n=0}^{N-1} h_k(n)e^{-j2\pi fn} \\ &= \sum_{n=0}^{N-1} (e^{-j2\pi(f-f_k)})^n \\ &= \frac{e^{-j\pi(f-f_k)N}}{e^{-j\pi(f-f_k)}} \frac{e^{j\pi(f-f_k)N} - e^{-j\pi(f-f_k)N}}{e^{j\pi(f-f_k)} - e^{-j\pi(f-f_k)}} \\ &= e^{-j\pi(f-f_k)(N-1)} \frac{\sin(\pi(f-f_k)N)}{\sin(\pi(f-f_k))} \end{aligned}$$

The maximum response is obtained when $f = f_k$, however it is obvious that the output of the filter is not zero for other values of f. This is another way to view leakage of the DFT. It implies that leakage is the non-zero response of a filter that is distant from the frequency causing the leakage. The frequency response of the output for $f = f_k$ is shown below:

[1] Selected figures taken from "Digital Signal Processing, New International Edition/4th", Proakis & Manolakis, ©Pearson Education Limited, 2014. ISBN: 978-1-29202-573-5



The full periodogram can be viewed as a bank of FIR filters:

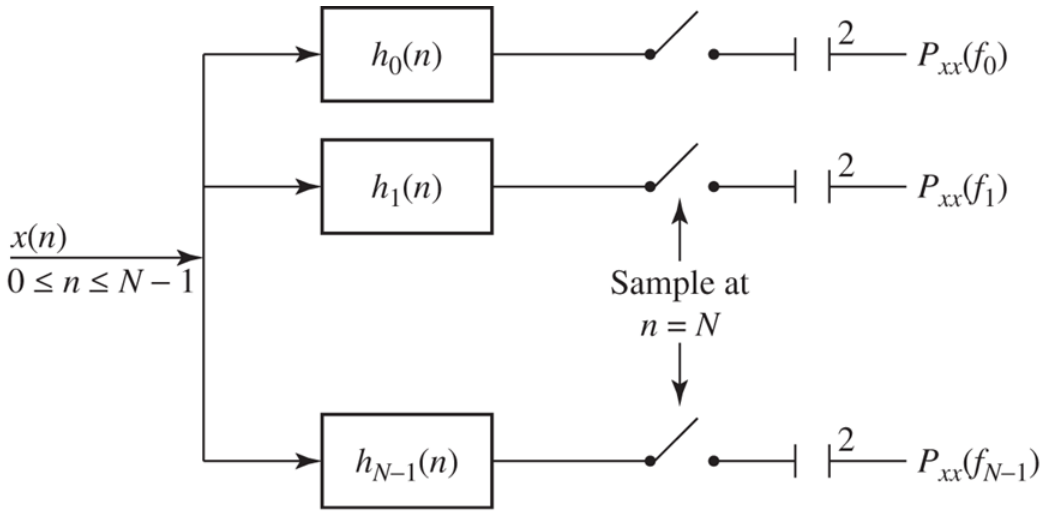


Figure 14.4.2 Filter bank implementation of periodogogram estimator.

Applying a windowed periodogram changes the shape of the frequency response of the filter, keeping its centre frequency at f_k. The same filter shape is retained for all frequencies. However, this may be inappropriate for certain problems where a mixture of strong and weak signals are present. In such cases, the dominant components can have a significant influence, obscuring the presence of the weaker frequency component. We will now examine a method of estimating a spectrum that adapts to the data being analysed.

2 Minimum variance spectral estimates

The ideal filter to estimate the spectral content at a frequency f_s has a transfer function that:

- is 1 at the frequency f_s; and
- is 0 at all other frequency components in the signal

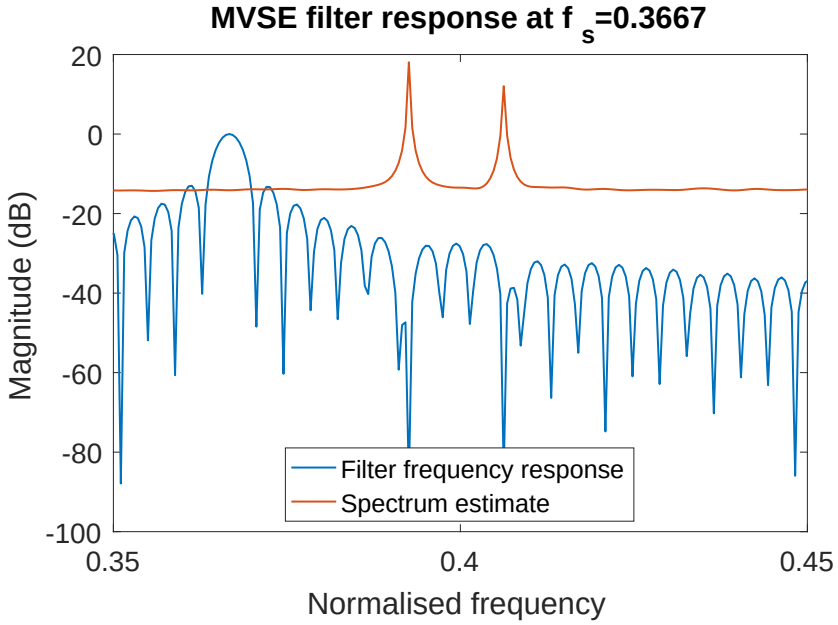
Thus, for a signal with two frequency components present, the ideal spectral estimate would have filters that placed null(s) on the component(s) not being analysed. Thus, only two of the filters in a filterbank would have non-zero outputs.

An approximation would be to define a transfer function that:

- is 1 at the frequency f_s; and
- minimises the response to all other frequency components in the signal

In other words the filter minimises its output power subject to a unity response at f_s

The minimum variance spectral estimation uses this approximation to produce individual filters for every frequency of interest. These filters aim to minimise the power from strong spectral components not at the frequency of interest, whilst maintaining a unity response to the frequency being considered. Below is an example of the frequency response for a particular frequency term, f_s in the estimate, along with the final estimate over all frequencies. It can be seen that the filter frequency response for f_s has attenuated the strong frequency components more than other frequency terms.



Filter output power

The output of the filter is given by

$$y(n) = \sum_{k=0}^{p} h(k)x(n-k) \equiv \mathbf{X}^T(n)\mathbf{h} \tag{4.7}$$

where $\mathbf{X}^T(n) = [x(n)\ x(n-1)\ \ldots\ x(n-p)]$ and $\mathbf{h} = [h(0)\ h(1)\ \ldots\ h(p)]^T$. Assuming $x(n)$ is zero mean, the variance is given by

$$\sigma_y^2 = E\left[|y(n)|^2\right] = E[\mathbf{h}^H\mathbf{X}^*(n)\mathbf{X}^T(n)\mathbf{h}]$$

For a real input, $x(n)$, this becomes:

$$\sigma_y^2 = \mathbf{h}^H E[\mathbf{X}(n)\mathbf{X}^T(n)]\mathbf{h} = \mathbf{h}^H\mathbf{\Gamma}_{XX}\mathbf{h}$$

The aim will be to find an $\mathbf{h}$ that minimises the output power, whilst maintaining $H(f_s) = 1$ where f_s is the frequency of interest.

The autocorrelation matrix, $\mathbf{\Gamma}_{XX}$, is a Toeplitz matrix with elements defined by the autocorrelation of $x(n)$. (Note, the notation used here could cause confusion. $\mathbf{\Gamma}_{XX}$ is a matrix of correlation values, while $\Gamma_{xx}(\omega)$ is a density spectrum). Where $x(n)$ is real, the matrix is defined as:

$$\mathbf{\Gamma}_{XX} = \begin{bmatrix} \gamma_{xx}(0) & \gamma_{xx}(1) & \gamma_{xx}(2) & \cdots & \gamma_{xx}(p) \\ \gamma_{xx}(1) & \gamma_{xx}(0) & \gamma_{xx}(1) & \cdots & \gamma_{xx}(p-1) \\ \gamma_{xx}(2) & \gamma_{xx}(1) & \gamma_{xx}(0) & \cdots & \gamma_{xx}(p-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \gamma_{xx}(p) & \gamma_{xx}(p-1) & \gamma_{xx}(p-2) & \cdots & \gamma_{xx}(0) \end{bmatrix}$$

Response to f_s

The response of $\mathbf{h}$ to frequency f_s is to be unity:

$$\begin{aligned} H(f_s) &= 1 \\ \Rightarrow \sum_{k=0}^{p} h(k)e^{-j2\pi k f_s} &= 1 \end{aligned}$$

Define $\mathbf{E}^T(f_s) = [1\ e^{j2\pi f_s}\ \dots\ e^{j2\pi p f_s}]$, then this is written as

$$\mathbf{E}^H\mathbf{h} = 1$$

This is the *constraint*

Using the definition of the variance, and the constraint, sets of filter coefficients, $\mathbf{h}$, are to be found, one set for each frequency output. The coefficients are to be chosen to minimise the output power for that frequency, subject to the constraint. This can be carried out using a Lagrange minimisation process. The derivation of this minimum is out of scope of this lecture course, so only the solution is presented.

Minimisation

The minimisation of the output power, subject to the constraint, results in:

$$\widehat{\mathbf{h}_{\text{opt}}} = \frac{\mathbf{\Gamma}_{XX}^{-1}\mathbf{E}(f_s)}{\mathbf{E}^H(f_s)\mathbf{\Gamma}_{XX}^{-1}\mathbf{E}(f_s)} \tag{4.10}$$

The filter has a different impulse response for each frequency, f_s, dependent upon the autocorrelation of the input signal, $\mathbf{\Gamma}_{XX}$ By including the autocorrelation of the data in the filter coefficients, the estimator adapts to the input. Should a dominant component be present at a given frequency, estimators for other frequencies will tend to attenuate the frequency corresponding to the dominant component heavily. Then, only outputs corresponding to the frequency of the dominant component, and frequencies that are close to this, will present a strong output.

We do not require the filter itself, but instead the output power of the filter, so substituting this into the previous equation:

$$\sigma^2_{\text{min}} = \frac{1}{\mathbf{E}^H(f_s)\mathbf{\Gamma}_{XX}^{-1}\mathbf{E}(f_s)} \tag{4.11}$$

As we can only estimate the autocorrelation function, $r_{xx}(n)$,

$$P_{xx}^{\text{MV}}(f) = \frac{1}{\mathbf{E}^H(f)\mathbf{R}_{XX}^{-1}\mathbf{E}(f)} \tag{4.12}$$

where $\mathbf{R}_{XX}$ is defined as

$$\mathbf{R}_{XX} = \begin{bmatrix} r_{xx}(0) & r_{xx}(1) & r_{xx}(2) & \cdots & r_{xx}(p) \\ r_{xx}(1) & r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(p-1) \\ r_{xx}(2) & r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(p-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{xx}(p) & r_{xx}(p-1) & r_{xx}(p-2) & \cdots & r_{xx}(0) \end{bmatrix}$$

This is the minimum variance spectral estimate.

Optimised calculation

Note that a detailed understanding of the optimised calculation is not examined. It is presented because it is an efficient calculation of a useful technique.

- Using these calculations directly, each frequency term requires separate computation
- Additional algebraic manipulations can be used to express as an Eigenvalue decomposition and DFT, which can be efficiently computed

As presented, this requires a separate computation for each frequency term being evaluated, as $\mathbf{E}(f_s)$ is different for each frequency. Through the use of transforms, computation of the result for all of the frequency terms can be done simultaneously. This is achieved by decomposing $\mathbf{R}_{XX}$ into its eigenvectors and eigenvalues, and using transforms. For equally spaced f, the multiplication by the set of $\mathbf{E}(f)$ values can be efficiently computed as a fast Fourier transform (FFT) of the eigenvectors. (Note that the derivation of the algorithm is out of scope of this course. The result is presented

here so that the technique can be applied efficiently in practice). Mathematically:

$$
\begin{aligned}
\mathbf{R}_{XX} &= \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1} \\
\mathbf{Q} &= [\mathbf{q}_0\ \mathbf{q}_1\ \mathbf{q}_2\ \cdots\ \mathbf{q}_p] \\
\mathbf{\Lambda} &= \begin{bmatrix} \lambda_0 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_1 & 0 & \cdots & 0 \\ 0 & 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_p \end{bmatrix} \\
u_i &= \frac{1}{\lambda_i} \\
\mathbf{u} &= \begin{bmatrix} u_0\ u_1\ u_2\ \cdots\ u_p \end{bmatrix}^T \\
\mathbf{v}_i &= |\mathrm{FFT}\,(\mathbf{q}_i)|^2 \\
\mathbf{V} &= [\mathbf{v}_0\ \mathbf{v}_1\ \mathbf{v}_2\ \cdots\ \mathbf{v}_p] \\
\mathbf{P}_{xx}^{\mathrm{MV}} &= p\frac{1}{\mathbf{V}\mathbf{u}} \\
\mathbf{P}_{xx}^{\mathrm{MV}} &= [P_{xx}^{\mathrm{MV}}(0)\ P_{xx}^{\mathrm{MV}}(1)\ P_{xx}^{\mathrm{MV}}(2)\ \cdots\ P_{xx}^{\mathrm{MV}}(p)]
\end{aligned}
$$

where, in the penultimate line, the fraction represents inversion of each term in the vector $\mathbf{Vu}$. Matlab code for this, and the other spectral estimation approaches, is available on the virtual learning environment for this course.

For the signal used earlier in this module, the resulting spectrum is: