Chapter 5

CORRELATION

As the word implies, correlation provides a measure of the similarity of two signals. Correlation is a powerful tool in signal detection, acoustic-field analysis, and array processing. Although we often compute correlation through the frequency domain for computational efficiency, correlation is fundamentally a time-domain concept. If implemented in the time domain, we take one time series, multiply it by the other and find the average value of the product. We repeat this process for all possible time shifts of one channel of data with respect to the other. If the two channels share a common component, the correlation will peak for one particular shift. The value of the peak indicates how good the match is and the shift required gives the delay of one signal with respect to the other.

Contrast this to coherence—another measure of the relationship of two channels of data. Coherence is fundamentally a frequency-domain measure. Coherence examines the similarity on a frequency-by-frequency basis and it requires averaging.

The serious signal analyst will want to master both of these techniques; however, in this chapter, the focus is on correlation.

But first, a brief outline of essentials from previous chapters. At this point, you should be comfortable with:

- construction of a spectrogram and the trade-off between time and frequency resolution;
- estimation of a transfer function (or impedance or propagation function) using auto- and cross-spectra;

- computation and significance of the coherence function;
- interpretation of the cross-spectrum and the relationship between stability of the phase difference between two channels and the coherence; and
- signal-to-noise ratio.

Furthermore, you should be able to write computer code to implement any of these operations using real data:

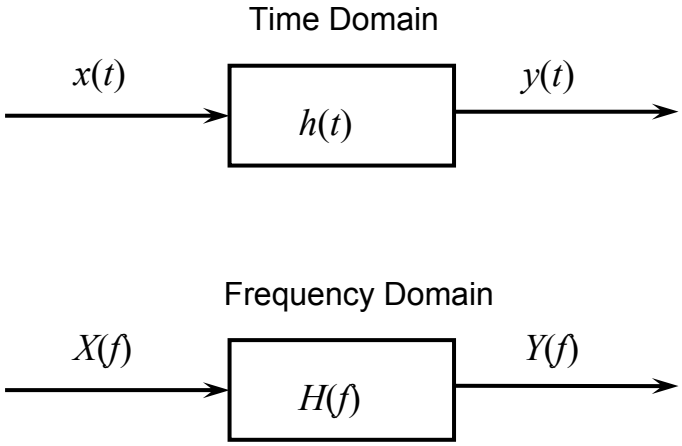
- given a time series, x, generate a spectrogram;
- given two time series, x and y, generate the cross-spectral densities, G_{XY} and S_{XY} (with averaging and windowing including the window correction factor), and the coherence, γ^2; and
- given an input time series, x, and an output time series, y, generate the transfer function from the averaged cross- and auto-spectra.

Introduction

We examined the spectral density (the auto- and cross-spectra[1]) functions in some detail in the previous chapter. A logical question is, "Are there time-domain equivalents for the spectral-density functions?" In fact, there are and these time-domain functions can be useful in some circumstances.

Recall that we can describe a system, its input, and its output, in either the time domain or the frequency domain:

[1] As a reminder, I will, at times, use the shorthand expressions auto-spectrum and cross-spectrum to mean auto-spectral density and cross-spectral density. (On the other hand, the expression, power spectrum, does not mean power spectral density. We owe this confusion to ambiguous usage in the literature.)



In the frequency domain, the relationship between the output and the input is a simple multiplication by the system's complex transfer function, H,

$$Y = X \cdot H \tag{V-1}$$

whereas in the time domain, the output is the convolution of the input and the system's impulse response, h,

$$y = x \otimes h \tag{V-2}$$

We also know that the cross-spectrum and the input auto-spectrum are related by the transfer function:

$$S_{XY} = S_{XX} \cdot H \tag{V-3}$$

so it's reasonable to ask if there is a corresponding time-domain relationship for the inverse Fourier transforms of the cross- and auto-spectra. If we call the inverse transforms, R_{xy} and R_{xx}, is there physical significance to the time-domain equivalent of the equation above,

$$R_{xy} = R_{xx} \otimes h \ ? \tag{V-4}$$

Correlation

If x and X are Fourier transforms of each other, then

$$X = \mathcal{F}[x] \quad \text{or} \quad x = \mathcal{F}^{-1}[X] \tag{V-5}$$

where we'll use $\mathcal{F}$ to denote the Fourier transform and $\mathcal{F}^{-1}$ to denote the inverse Fourier transform (whether the transform is an integral transform or a discrete transform). The inverse transform integral is

$$x(t) = \int_{-\infty}^{\infty} X\, e^{j2\pi f t}\, df \tag{V-6}$$

If we replace X by its complex conjugate and t by $-t$, then the integrand is the complex conjugate of the original integrand. Therefore,

$$x^*(-t) = \mathcal{F}^{-1}\left[X^*\right] \tag{V-7}$$

Because we are dealing with functions that are real in the time domain, the complex conjugate operation on $x(-t)$ has no effect.

Since the product of two linear spectra is the transform of the convolution of their two time functions, we can write

$$\mathcal{F}^{-1}\left[X^* X\right] = x(-t) \otimes x(t) \quad , \tag{V-8}$$

which relates the auto-spectrum in the frequency domain to a convolution ($\otimes$)[2] in the time domain.

Mechanics of convolution

The convolution of two functions is performed by reversing one of the functions in time, then shifting it with respect to the other, multiplying, then

[2] The asterisk (*) is often used to represent convolution; however, we reserve the asterisk for the complex conjugate operation.

integrating. Each time shift generates another value of the convolution. In integral form, convolution is,

$$a(t) \otimes b(t) \quad = \quad \int a(u)\, b(t-u)\, du \quad , \tag{V-9}$$

where *b* is the function that is reversed in time and shifted. Order is not important so the following form is also true,

$$a(t) \otimes b(t) \quad = \quad \int a(t-u)\, b(u)\, du \quad . \tag{V-10}$$

Here *a* is the reversed-in-time and shifted function. In these equations, *u* is the variable of integration and has the units of time. Inside the integral, *t* indicates the time shift of the reversed-in-time function with respect to the other function.

Now apply these tools to evaluate the inverse transform of X^*Y (as appears in the cross-spectral density, S_{XY}):

$$\mathcal{F}^{-1}\left[X^* Y\right] \quad = \quad x(-t) \otimes y(t) \tag{V-11}$$

The right side is an ordinary convolution but with the function, *x*, reversed in time. The time reversal is done by making the entire argument of *x* negative. As an integral, this convolution is then,

$$x(-t) \otimes y(t) \quad = \quad \int x(u-t)\, y(u)\, du \quad . \tag{V-12}$$

Make the substitution, $\tau = u - t$, so $u = t + \tau$ and $du = d\tau$,

$$x(-t) \otimes y(t) \quad = \quad \int x(\tau)\, y(t+\tau)\, d\tau \quad . \tag{V-13}$$

Since τ is the variable of integration, the right side would become a function of *t*. The convention, in correlation, is to use a different symbol for the shift time so

we'll follow that convention by swapping the roles of τ and t on the right side of the equation above. Therefore,

$$x(-t) \otimes y(t) \quad = \quad \int x(t)\, y(\tau + t)\, dt \quad . \tag{V-14}$$

The inverse transform of the cross-spectrum is then,

$$\frac{1}{T} x(-t) \otimes y(t) \quad = \quad \frac{1}{T} \int x(t)\, y(t + \tau)\, dt \quad = \quad R_{xy}(\tau) \quad . \tag{V-15}$$

The complex conjugate (in the frequency-domain expression for S_{XY}) reverses the x function in time but then the convolution reverses that function again restoring its original order. That leads to the middle expression above, which shows the two functions multiplied together with one shifted by τ. The time-domain result, R_{xy}, is the **cross-correlation** of x and y, which is the inverse transform of the cross-spectral density:

$$\mathcal{F}^{-1}\left[S_{XY}(f) \right] \quad = \quad R_{xy}(\tau) \quad . \tag{V-16}$$

The cross-spectral density is a function of frequency, f, and the cross-correlation is a function of **shift time**, τ.

Notice that, since the independent variable, τ, is shift time, τ *can be either positive or negative*. A peak in the cross-correlation at positive τ would indicate that y is delayed with respect to x; a peak at negative τ would indicate that x is delayed with respect to y.

Similarly, the auto-correlation, R_{xx}, is the inverse transform of the auto-spectral density, S_{XX},

$$\mathcal{F}^{-1}\left[S_{XX}(f) \right] \quad = \quad R_{xx}(\tau) \quad . \tag{V-17}$$

Behavior of the auto-correlation

It is often easier and faster to compute correlations by way of the frequency domain—find the linear spectra of two channels, form the spectral density, then take the inverse transform—because the *fft* function is so efficient; however, we'll start our examination of correlation in the time domain because that view is more intuitive. As an integral[3], the auto-correlation is,

$$R_{xx}(\tau) = \frac{1}{T}\int_0^T x(t)\, x(\tau + t)\, dt \quad . \tag{V-18}$$

What is the value at zero time shift ($\tau = 0$)?

$$R_{xx}(0) = \frac{1}{T}\int_0^T x(t)\, x(t)\, dt = \frac{1}{T}\int_0^T x^2\, dt \quad . \tag{V-19}$$

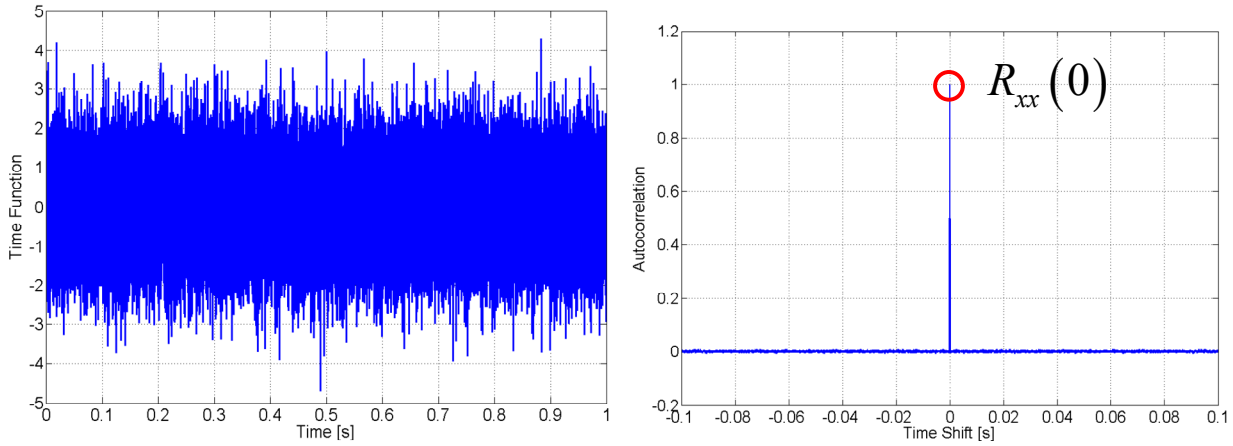
The value at zero time shift is the mean-square value of the time-domain function. This gives us a way to check calculations. If you write a routine to compute the auto-correlation, check the results at $\tau = 0$ against the mean square of x.

This property also gives us some understanding of the nature of auto-correlation. In general, we'd expect a peak in the auto-correlation when the time shift (or "lag") is zero—we're simply integrating the square of the time function and the result is the mean-square value. In fact, the auto-correlation at zero shift is always the highest value of the auto-correrlation. If the functions are shifted a bit, then they won't match as well and we wouldn't expect the product to be always positive as it must be with zero shift. The value of the auto-correlation should drop away from zero shift.

[3] You may see correlation defined without the leading factor of 1/*T*. The definition without the 1/T is inconsistent with the analyses in these notes. If you are using a packaged routine for correlation (a practice I will not recommend), you must be extremely careful to understand the scaling and options for that routine. Although perhaps less efficient, writing your own function(s) allows you to be in complete control of the operation(s). Just be sure to check your work before relying on it. If you insist on using a packaged function, check it also. One important check is described later in the paragraph.

Random noise

If, for example, the signal is purely random noise, then we'd still see a peak for zero shift (and that peak would equal the mean-square value of the noise) but we'd see a very sharp drop to much smaller values for even a one-sample ($\tau = \Delta t$) shift away from zero. This is shown below: a random time series is shown on the left and its auto-correlation is shown on the right. The random time series was generated using the *randn* MatLab function, which returns a sequence with a mean of zero and a variance of one so the mean-square value is one and the auto-correlation at zero shift is also one.

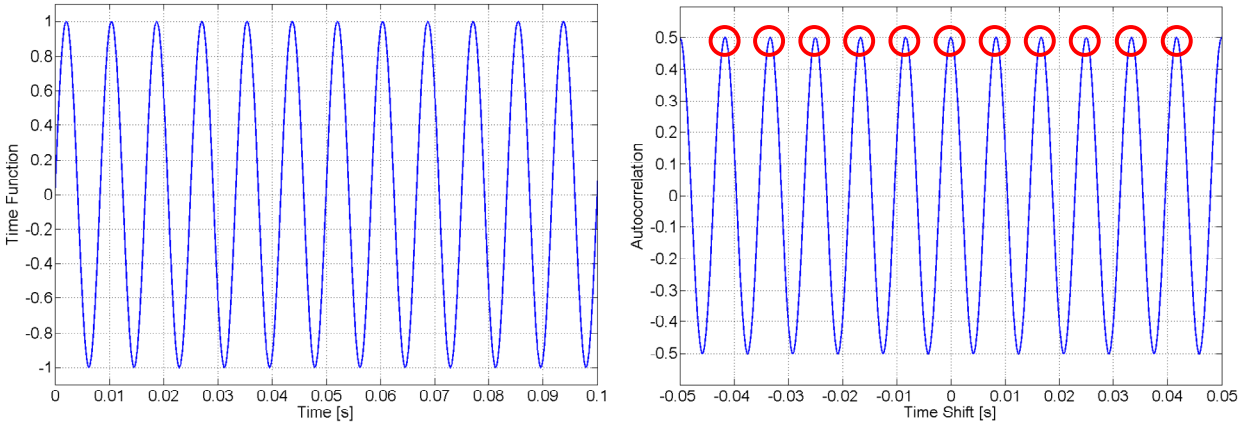


Notice that the auto-correlation looks like a delta function (an "impulse"—a single large value with all other values equal to zero). It isn't quite a delta function since the values away from zero shift are not zero, just very small, but it's close. Working in the other direction, the transform of a delta function (or impulse) results in a constant value in the frequency domain, which is consistent with a uniform distribution of power over all frequencies (i.e., "white" noise).

Periodic signals (a sine wave, for example)

If the function in time is periodic, then we'd expect another peak each time the shift, τ, is equal to an integer multiple of the period of repetition. If the function in time is sinusoidal, then the auto-correlation will also be sinusoidal. In the figure below, the sinusoidal time series is on the left and the corresponding

auto-correlation is on the right. With an amplitude of one, a sine wave would have a mean-square value of one-half,[4] which agrees with the value of the peaks in the auto-correlation (red circles) on the right.



Exercise 5.1: Basic Auto-Correlation

Use the MatLab *randn* function to create a random time series. Find and plot the auto-correlation of this time series by finding the spectral density, S_{XX}, and then taking the inverse transform. Notice where the peak is in delay-time. What is the value of that peak (and what are the units if the time series is in working units—WU)? What is the mean-square value of the original time series?

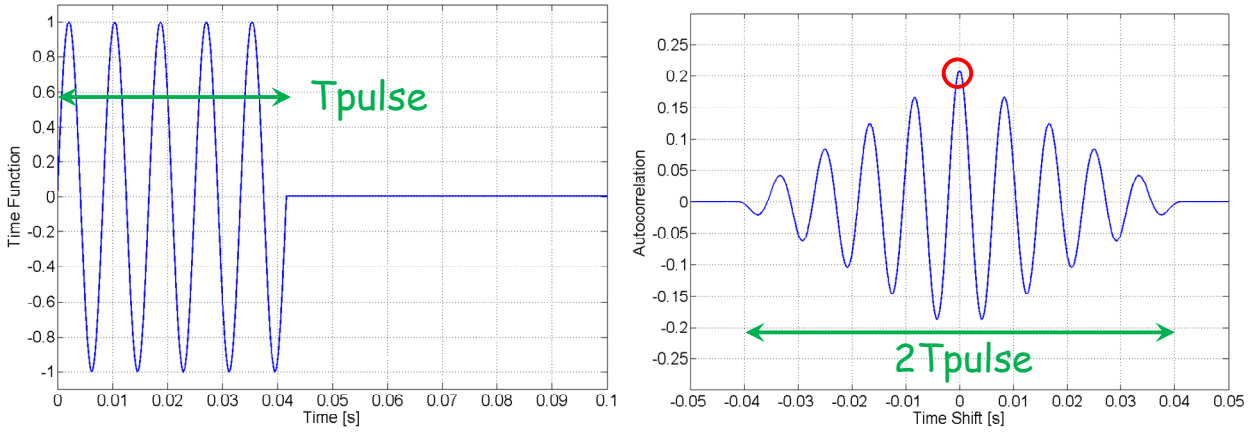
Modify your code to put the negative time-delay points before the zero- and positive-time-delay points (so that, if you plot the correlation, the delay time would run from the maximum negative value through zero to the maximum positive value). Make sure that the time axis in your plot is correct—the peak must be at delay-time = 0.

Generate a sine wave time series; find and plot the auto-correlation.

[4] The mean-square integral is not difficult to evaluate for a sine wave and the result is standard: if the amplitude of a sine (or cosine) is one, then, if there are an integer number of full cycles in the time series, the mean square is one half and the root-mean-square is one over the square root of two (about 0.707). This calculation is specific to the sine or cosine function. Don't apply these factors to anything other than sinusoids with integer numbers of cycles!

Sine pulse

Now consider a more interesting signal—a sinusoidal pulse. In this case, the time series is that of a sine wave over a limited time and zero for the remainder of time. This circumstance is shown below.



In the time domain, the sinusoidal portion is less than half the extent of the time series so the mean-square value will be much less than what you'd calculate based on a sine wave that is on for all time. In fact, the mean-square value is slightly greater than 0.2 as compared to 0.5 for a unit-amplitude sine wave on for all time. The peak in the auto-correlation (red circle) gives 0.2, which equals the mean-square value.

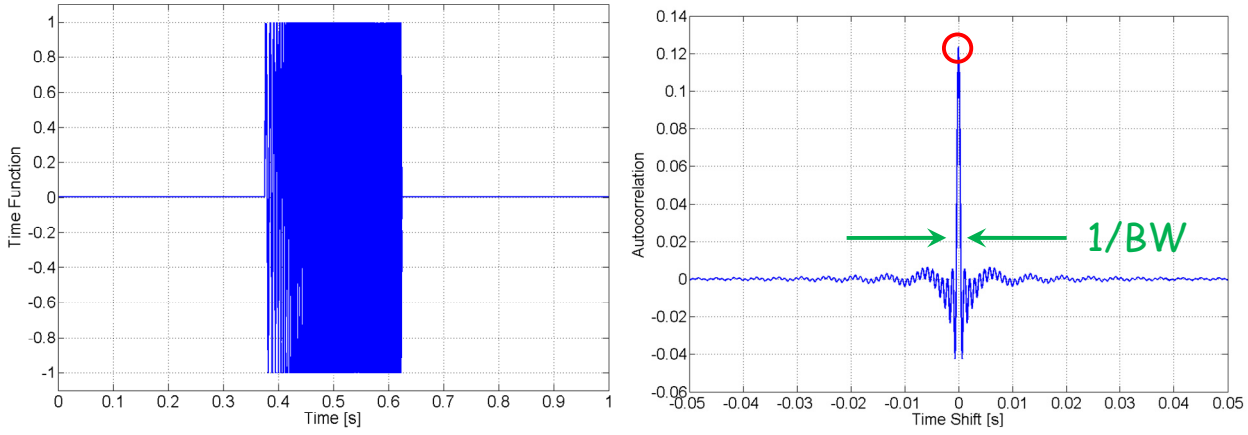
The auto-correlation has a triangular envelope: there are multiple peaks but the zero-shift peak is the largest; the other peaks slowly decrease in amplitude away from the zero-shift point. At zero shift, all features align so the average value of the sum of the products is a maximum. With increasing shift (in either direction), the mis-alignment initially increases leading to a drop in the correlation. When the shift is a full cycle, four of the five sine cycles align so there's another correlation peak but lower in amplitude. When the shift reaches two full cycles, only three of the five peaks align and the peak is even smaller. Notice that, when the shift is half a cycle, there is a negative peak. If you were to reverse the sign of all of the values of one of the functions, you'd see that four and an half cycles align when the shift is one-half a cycle.

Compare the sine-pulse case to the other two cases. For random noise, the bandwidth of the signal (the extent of the distribution of power in the frequency domain) is large—the full range of frequency—and the auto-correlation peak is as narrow as it can possibly be. For the sine wave (on for the entire record), the bandwidth of the signal is small—all of the power is at one frequency—while the auto-correlation peaks all have the same value. That is, the correlation is spread out over all time shifts. For the sine pulse, the bandwidth is intermediate (about equal to the reciprocal of *Tpulse*) and the auto-correlation shows some localization in time shift but the taper from the peak value is slow.

This relationship is fundamental: the larger the bandwidth of the time-domain signal, the more localized the auto-correlation. This feature is important in the design of waveforms for correlation-based processes.

Wideband pulses

The bandwidth of a sine-like pulse can be increased without increasing its duration by sweeping the frequency. In the next example, the sinusoidal pulse sweeps from 100 to 1100 Hz in 0.25 seconds. A constant-frequency sine pulse 0.25 seconds long would have a bandwidth of about 4 Hz; the swept-sine pulse has a bandwidth of about 1000 Hz—much larger. This should result in a narrower auto-correlation and it does as shown here:



Although the width of the correlation peak is ambiguous because of the rapid oscillations, the usable width is approximately the inverse of the bandwidth of the pulse.

Cross-Correlation

As we saw earlier, there is a cross-correlation function, R_{xy}, defined as

$$R_{xy}(\tau) = \frac{1}{T}\int_0^T x(t)\, y(\tau + t)\, dt \qquad \text{(V-20)}$$

which is related to the cross-spectral density, S_{XY}, by a Fourier transform:

$$\mathcal{F}^{-1}\left[S_{XY}(f)\right] = R_{xy}(\tau) \qquad \text{(V-21)}$$

The cross-correlation at zero time shift also has a simple relationship to the underlying signals,

$$R_{xy}(0) = \frac{1}{T}\int x(t)\, y(t)\, dt \quad , \qquad \text{(V-22)}$$

and we can use this relationship to test the accuracy of an algorithm for finding the cross-correlation.

Positive and negative time shifts

If we generate the cross-correlation from the inverse transform of the cross-spectral density, then MatLab automatically returns the correlation as if the shift time starts at $\tau = 0$ then increases in steps of Δt. As for positive and negative frequencies in a linear spectrum, we can interpret this range of shift times[5] as

[5] The range suggested here is a direct analog of the range of positive and negative frequencies for the linear spectrum in which there are N/2-1 points in the negative range. In contrast, though, to the linear spectrum of a real time series, there is no redundancy in the correlation output; no values are repeated even as complex conjugates. Consequently, you may elect to split the positive and negative shift times so that there are N/2 negative points and

consisting of positive delays from $\tau = 0$ to $\tau = T/2$ followed by negative delays from $-T/2 + \Delta t$ to $-\Delta t$.

Any such re-ordering of points should be done with regard to the context of the measurement. For some correlations, you may know that any relevant delays will be positive. Then there is no point in interpreting the upper half as corresponding to negative delays. This would be the case if one channel represented a short transmitted pulse and the other channel represented the output of a receiver some distance away. If you select the transmitted waveform as x and the received waveform as y, then the waveform, y, would only show delayed versions of the transmitted pulse and correlation peaks for each of the paths from source to receiver would all show positive delays[6]. In this case, it would make sense to display (plot) the correlation values on a shift-time axis that extended from $\tau = 0$ to $\tau = T - \Delta t$ and take a time segment (for both channels, appending zeros to the source pulse if necessary) long enough to extend at least a pulse length beyond the last arrival.

If, on the other hand, the two channels were the outputs of two microphones (or hydrophones or accelerometers) and the signal arrival could be coming from any direction, then both positive and negative delays would be possible and it would make sense to re-order the correlation points to move those points corresponding to negative delays below those corresponding to positive delays and display the results with a time axis that runs from $-T/2 + \Delta t$ to $+T/2$.

When planning correlation analyses, you must also consider the implications of the implied periodicity of our sampled signals. I will discuss these implications shortly.

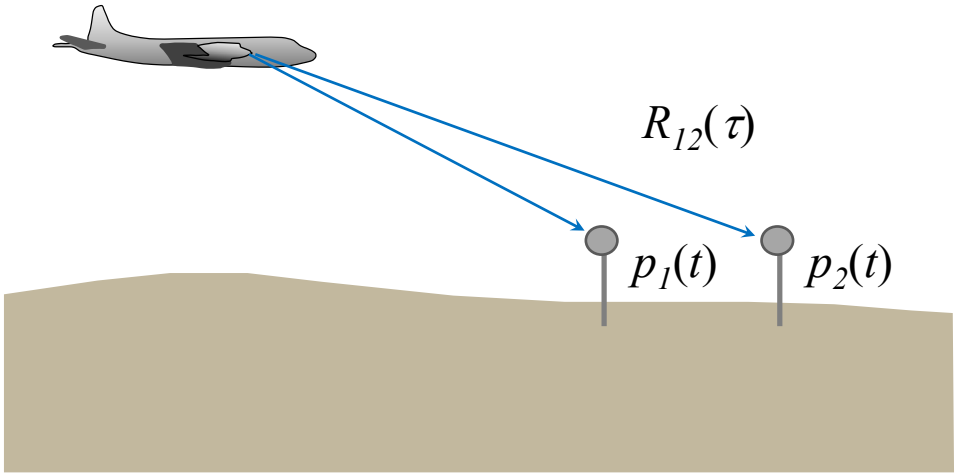
N/2 zero-plus-positive points. Whatever your preference, make sure that you keep track of where the $\tau = 0$ point wound up! This is most reliably done by re-arranging a shift-time vector in exactly the same way as you re-arranged the correlation vector.

[6] If, that is, you derived the cross-correlation from S_{XY}. If you used S_{YX}, the delay would be negative. I suggest that you be consistent to avoid interpretation of delays in the wrong sign sense. If x represents some input and y represents some output, use S_{XY} so that "normal" delays—y lagging x—will produce positive delays.

Applications for cross-correlation

Time delay between two points

Even in the presence of noise or interference, if there is a common component in the signals recorded at two locations, the cross-spectrum can be used to find the time delay of that common component between the two receivers. The common component, itself, can be noise and the wider the bandwidth of the common component the better: wider bandwidth in the source signal results in a narrower correlation peak, which translates to better accuracy in estimating the time delay.

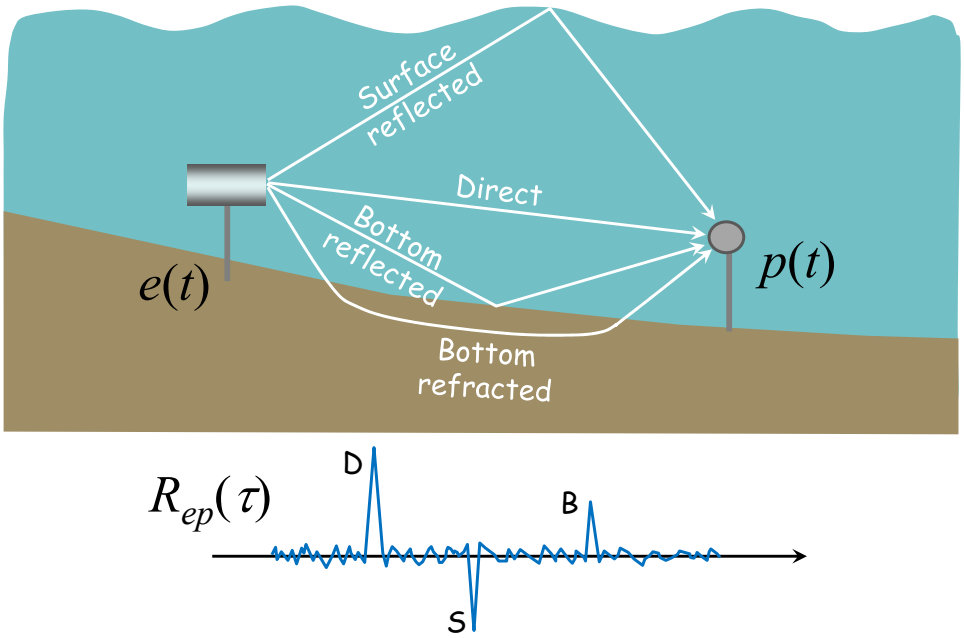


Once the time delay is known, you might be able to estimate the direction of arrival if, for example, you knew that the source was always in a certain plane passing through both receivers. By adding more receivers and calculating delays from other receiver pairs, the direction of arrival can be estimated without assuming a particular geometry.

Detection or identification of multiple propagation paths

Suppose $x(t)$ is a transmitted acoustic signal and $y(t)$ is the received signal at some distance. There will be a delay in time between the transmission and the reception. The cross-correlation function would show a peak at the shift, τ, that

corresponds to the propagation delay. If there were several paths by which the transmitted signal could reach the receiver ("multipath" propagation), there would be several peaks in the cross-correlation.



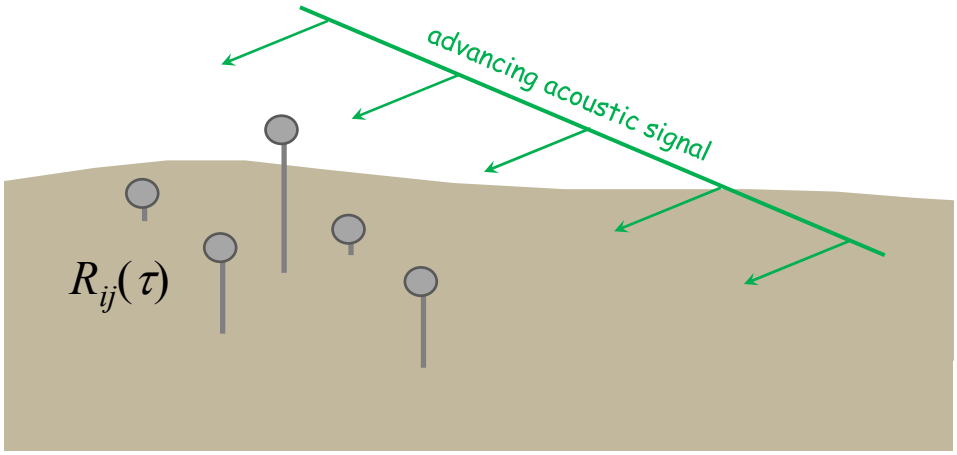
If the transmitted signal is sufficiently strong, the actual transmitted pulse might be visible at times corresponding to the arrivals of the different paths. More commonly, the pulses would be obscured by noise. The correlation process improves the detection signal-to-noise ratio and improves estimation of arrival times or arrival levels. The improvement in arrival-time estimation is directly related to the degree of time compression available[7] from the source pulse. Consequently, a transmitted pulse with a wide bandwidth (and correspondingly narrow auto-correlation) is an effective strategy.

For situations in which there are one or more distinct signal paths, correlation analysis in the time domain can be useful for isolating or identifying those paths.

[7] That is, the ratio of the width (in shift time) of the auto-correlation peak to the duration of the source pulse.

Signal detection and direction finding

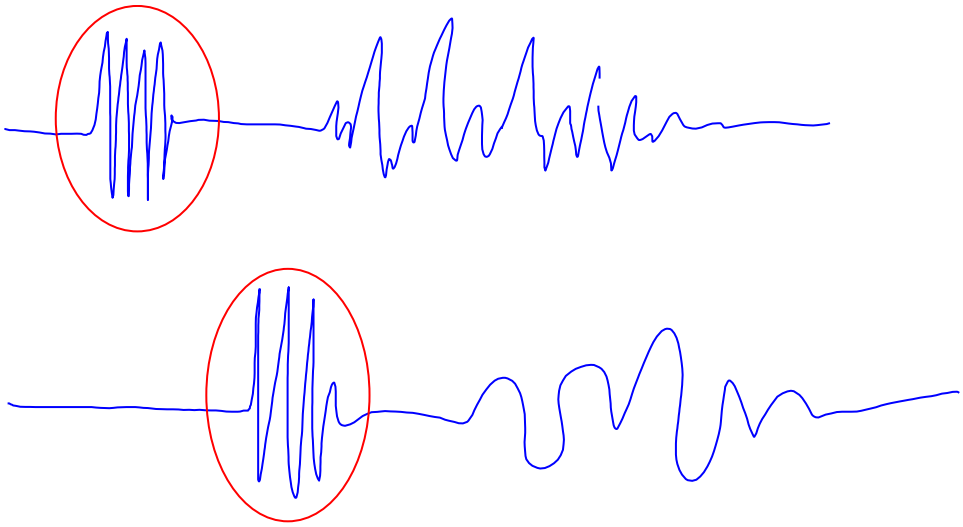
With more than two sensors, we can use the multiple time delays to estimate the direction of a signal arrival and we can use the normalized cross-correlation (described later) values to determine the likelihood that there is a genuine signal arriving.



Because of the importance of these processes—signal detection and angle-of-arrival estimation—a subsequent chapter will consider them in detail.

Synchronizing two or more channels

If we need to synchronize two or more channels, we can embed a special pulse in each (or just in the transmitted waveform in a transmit/receive measurement). If we use a pulse with a sharply peaked autocorrelation, we can use a replica of that pulse in cross-correlation with a channel to establish a time reference point.



Normalized Cross-Correlation

The cross-correlation gives a measure of the similarity of two signals and an estimate of the time delay between similar features. With a small modification, the cross-correlation can be normalized so that a value of one corresponds to perfect similarity. By perfect similarity, we mean that the curve shapes in the two channels are identical but one could be shifted in time or scaled by a constant factor. In general, then, x and y are perfectly similar if,

$$y(t) \quad = \quad A\,x(t - t_0) \quad . \tag{V-23}$$

In this case, the cross-correlation peak would occur at a time shift of t_0,

$$R_{xy}(t_0) \quad = \quad \frac{1}{T}\int_0^T x(t)\,A\,x(t)\,dt \quad = \quad A\overline{x^2} \quad , \tag{V-24}$$

and be equal to the scale factor, A, times the mean-square value of x. (This is not strictly true for the integral as written above; however, all of our signals are sampled so the implied periodic repetition prevents any errors from non-overlapping portions of the two signals.)

If we divide the ordinary cross-correlation by the *rms* value of x and by the *rms* value of y, we form the normalized cross-correlation, C_{xy},

$$C_{xy}(\tau) = \frac{R_{xy}(\tau)}{\sqrt{\overline{x^2}}\sqrt{\overline{y^2}}} = \frac{R_{xy}(\tau)}{x_{rms}\, y_{rms}} , \qquad \text{(V-25)}$$

which has a maximum magnitude of one,

$$-1 < C_{xy}(\tau) < +1 , \qquad \text{(V-26)}$$

and that value is achieved only if x and y are perfectly similar[8]. C_{xy} is sometimes called the **correlation coefficient**. Alternatively, the rms values of x and y can be expressed as their autocorrelations at zero time shift:

$$C_{xy}(\tau) = \frac{R_{xy}(\tau)}{\sqrt{R_{xx}(0)R_{yy}(0)}} . \qquad \text{(V-27)}$$

Cross-correlations and time delays

One of the most useful features of the cross-correlation is that the locations (in shift time) of the peaks correspond to the relative timing between the two channels. If the correlation peak appears at a positive delay time, does that mean that the signal reached the point at which $x(t)$ was measured first or the point at which $y(t)$ was measured? Sometimes the positioning of the sensors with respect to the signal propagation path is known and the significance of the sign of the delay is clear; however, we can determine, from the definitions, what the time delay means even if we don't know the signal direction.

[8] If the shapes are a perfect match then the normalized cross-correlation is +1; if the shapes are a perfect match except that one is inverted from the other, then the normalized cross-correlation is -1.

Consider a plane wave traveling along the line connecting the two sensor locations. For our exponential time convention ($e^{+j\omega t}$), the complex-exponential form for a plane wave traveling in the z direction is,

$$A e^{j(\omega t - k z)} \quad , \tag{V-28}$$

where k is the wave number, ω/c. If the sensor producing $x(t)$ is located at z_1 and the sensor producing $y(t)$ is located further along the z axis at z_2, then,

$$X = A e^{j(\omega t - k z_1)} \quad ; \quad Y = A e^{j(\omega t - k z_2)} \quad , \tag{V-29}$$

or

$$X = A e^{j\omega(t - z_1/c)} \quad ; \quad Y = A e^{j\omega(t - z_2/c)} \quad . \tag{V-30}$$

From our definition of the cross-spectrum[9],

$$S_{XY} = \frac{1}{T} X^* Y = \frac{A^* A}{T} e^{j\omega(t_1 - t_2)} \quad , \tag{V-31}$$

where $t_1 = z_1/c$ and $t_2 = z_2/c$, the arrival times at the locations of the x and y measurements, respectively.

The cross-correlation is the inverse transform of the cross-spectrum. For the convention used in these notes,

$$R_{xy}(\tau) = \int S_{XY} e^{j\omega\tau} \, df = \int \frac{A^* A}{T} e^{j\omega(\tau + t_1 - t_2)} \, df \, , \tag{V-32}$$

which has a peak at $\tau = t_2 - t_1$. Therefore, a positive τ indicates that t_1 is less than t_2: the signal arrived at z_1 *before* it arrived at z_2.

[9] Some authors define the cross-spectrum as $(XY^*)/T$ instead of $(X^*Y)/T$. That definition reverses the phase in the frequency domain and reverses the time-delay axis in the time domain. If you are using a packaged software routine to find the cross-spectrum, make sure you understand exactly how it is implemented.

Another way of remembering the direction associated with τ is to remember that taking the complex conjugate of a linear spectrum is equivalent to reversing the time-domain function in time. For S_{XY}, we take the conjugate of X so the proper direction of time is preserved in Y (and, of course, y). So a positive τ is later in time in the y function.

Exercise 5.2: Cross-Correlation and Time Delay

Create a time series, x, 200 points long. Assume a sample rate of 1000 samples per second. Set point 10 equal to one but make all of the other points zero. Create another time series, y, also 200 points long. Set point 15 equal to one but make all of the other points zero. Find and plot the cross-correlation of this time series by finding the cross-spectral density, S_{XY}, and then taking the inverse transform. Notice where the peak is. What is the time delay that corresponds to the peak?

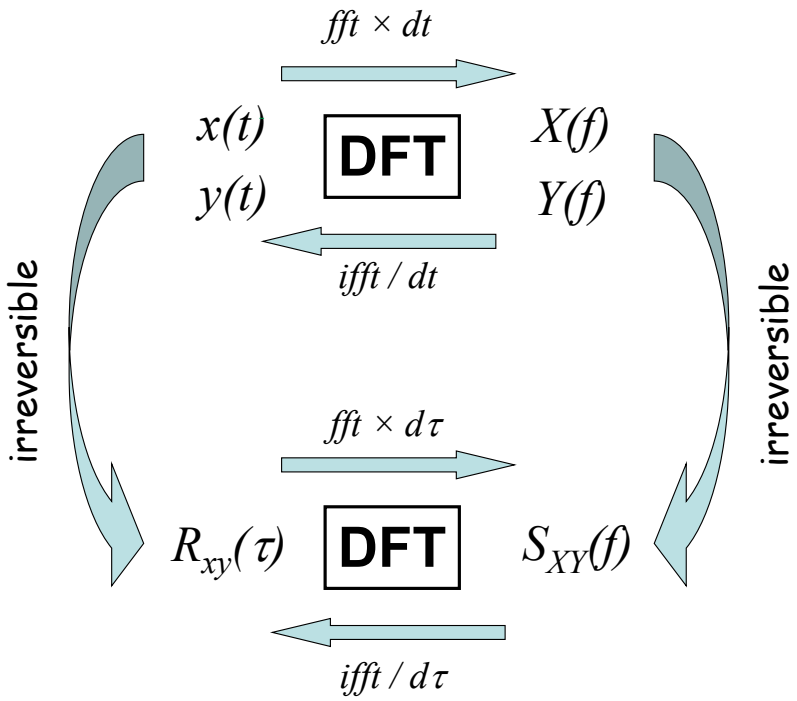
Modify y so that the only non-zero value is at point 5 instead of at point 15. Plot the new cross-correlation. Notice where the peak is. What is the time delay that corresponds to the peak?

As you did in the previous exercise, modify your code to put the negative time-delay points before the zero- and positive-time-delay points. Make sure that the time axis in your plot is correct – the peak for the first case must be at positive or negative 5 time increments; the peak for the second case must be at negative or positive 5 increments. When y is delayed with respect to x (the first case), is the cross-correlation time delay positive or negative?

Summary of Frequency- and Time-Domain Relationships

The following diagram summarizes the major relationships that we've established so far. We can move back and forth between ordinary time series (x, y) and linear spectra (X, Y) with the discrete transform (*fft, ifft*). These operations are reversible. We can form the auto- and cross-correlations (R_{xx}, R_{yy}, R_{xy}) from x and y; however, these operations are not reversible. We can also form the auto- and cross-correlations (S_{XX}, S_{YY}, S_{XY}) from X and Y. The linear spectra, X and Y, cannot be reconstructed from their auto-correlations; however, as long as no averaging has been done, the linear spectra can be reconstructed if you have both the auto- *and*

cross-spectra. Finally, the correlations and the spectral densities are reversible transforms of each other.



We also have several relationships that are extremely useful for checking the validity of our calculations and computer code:

- the "integral" (the sum times Δf) of either $G_{XX}(f)$ or $S_{XX}(f)$ equals the mean-square value of $x(t)$;
- the "integral" of $S_{XY}(f)$ equals the average value of the point-by-point product of $x(t)$and $y(t)$;
- the value of R_{xx} at zero delay time is equal to the mean-square value of $x(t)$; and,
- the value of R_{xy} at zero delay time is equal to the average value of the point-by-point product of $x(t)$and $y(t)$.

Although it is not obvious, the most efficient way to find auto- or cross-correlations is usually the "long" way: finding the linear spectra, then the spectral

densities, and then inverse transforming to the correlations. This is also the approach that is easiest to program; however, the implied periodicity of the sampled functions has consequences as we will see shortly.

Finding the correlation in practice

The correlation between two functions can be found directly in the time domain or by means of the cross- or auto-spectral density in the frequency domain.

Time domain

Implementation in the time domain is straightforward to understand since the basic definition of correlation is in the time domain. You multiply the two time series together (point by point), average the result, shift one series by one Δt, multiply, average, shift again, and so on. While straightforward, this process is inefficient computationally unless one time series is much shorter than the other. Normally, the frequency-domain process yields faster code[10].

Frequency domain

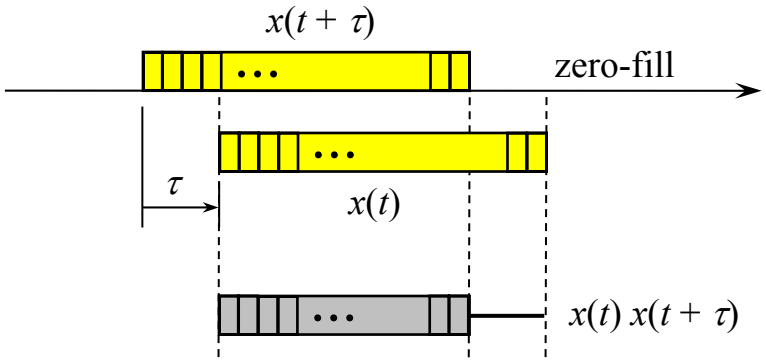
The auto- (cross-) correlation is the inverse transform of the auto- (cross-) spectral density and the spectral densities are simple to calculate from the original time series. There is no need to find the single-sided spectral density; S_{XX} (or S_{XY}) is already in the proper order for MatLab's *ifft*. Consequently, the frequency-domain approach is simple to code. If filtering is necessary, the filtering can be done in the frequency domain before the inverse transform step. If calculating the correlation using the frequency-domain approach, you must, however, respect the implied periodicity of the functions. This subject is discussed next.

[10] The code is also simpler so errors are less likely; however, no matter how simple, make sure you check the code!

Sampled versions of the correlation function

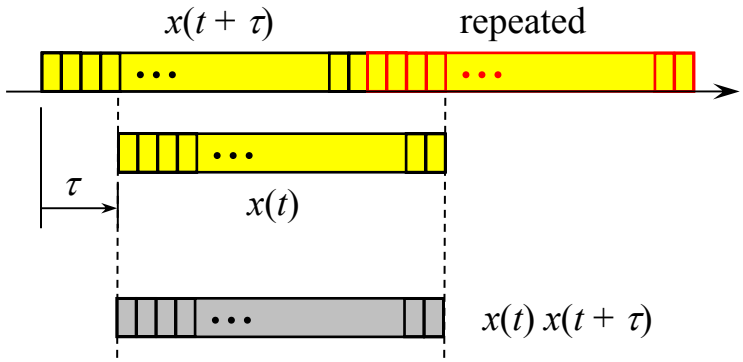
If you are calculating correlations from spectral densities, you must not forget the implied periodicity in both the frequency and time domains for sampled signals. From the integral definition of the auto-correlation, there is no restriction that the time shift always be positive; however, MatLab's inverse fft (*ifft*) produces a time-domain series corresponding to the time shifts, $\tau = 0, \Delta t, 2\Delta t, \ldots (N-1)*\Delta t$, which are all positive. Because of the implied periodicity, the result for a shift of $\tau = -\Delta t$ is identical to the result for the positive shift of $\tau = (N-1)*\Delta t$, the result for a shift of $\tau = -2\Delta t$ is identical to the result for the positive shift of $\tau = (N-2)*\Delta t$, and so on. This is particularly important to remember for the cross-correlation: the cross-correlation between two channels of time domain data may show a peak for a small *negative* relative shift but, in the output of the *ifft* operation, the peak will appear as a large positive shift. It is often easier to visualize the significance of the cross- (or auto-) correlation if you move the upper half of the time-domain result down below the lower half so that $\tau = 0$ is at the middle with negative shifts below and positive shifts above.

If we chose to work solely in the time domain, translation of the integral forms for correlation into sampled forms is straightforward with one exception. The time shift, τ, means that the N points in one time function will not line up completely with the N points in the shifted time function (unless $\tau = 0$). If we were only concerned about finding those delays (shifts) that produced peaks in the correlation, we could just "zero-pad" one of the functions:



so that any values that did not overlap would produce a zero result. (With careful coding, you could, of course, use actual values from the time series instead of zero filling.)

But sampling in the time and frequency domain and the discrete Fourier transform imply that the record in the time (or frequency) domain is repeated over and over. So, if we are shifting one of the time functions relative to the other, the shifted function will overlap partially the periodic extension of the other function. If we accept that overlap onto the periodic extension, we are performing a *circular correlation*:



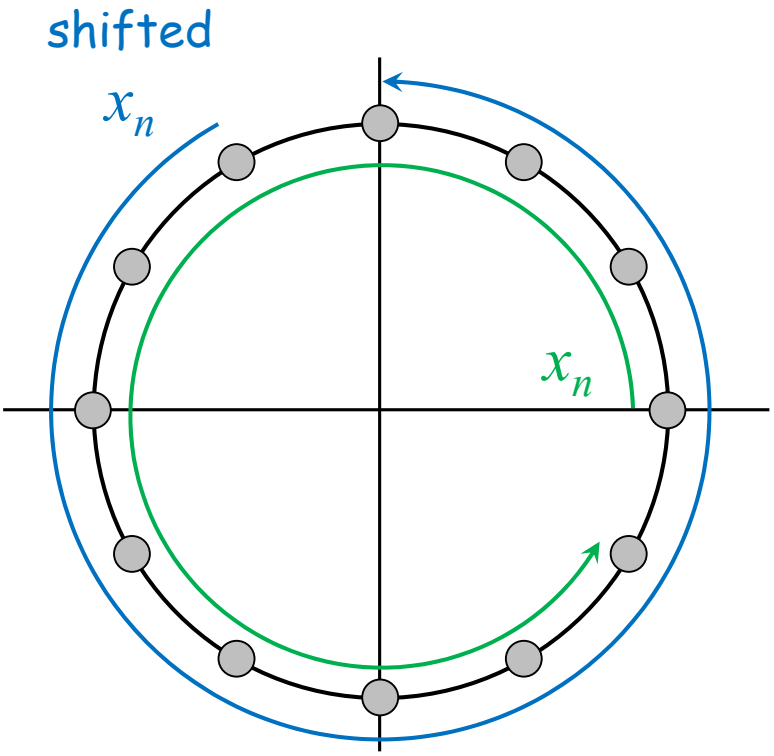
If working in the time domain (which would rarely be a good idea), the most convenient way to code these operations is to make one vector twice as long as the other by appending N zeros (for the simple correlation) or appending a copy of itself (for the circular correlation). Then simply adding the shift constant to the indexes for the double-length vector will index the proper values for the multiplication:

$$R_{xx}(n_s \Delta t) = \frac{1}{T}\sum_{n=0}^{N-1} x_n \, x_{n+n_s} \, \Delta t = \frac{1}{N}\sum_{n=0}^{N-1} x_n \, x_{n+n_s} \qquad \text{(V-33)}$$

Here, $n_s \Delta t$ is the discrete version of τ, and n_s is the shift in index value. Notice that, when $n_s = 0$, the right side is the mean-square value of x_n.

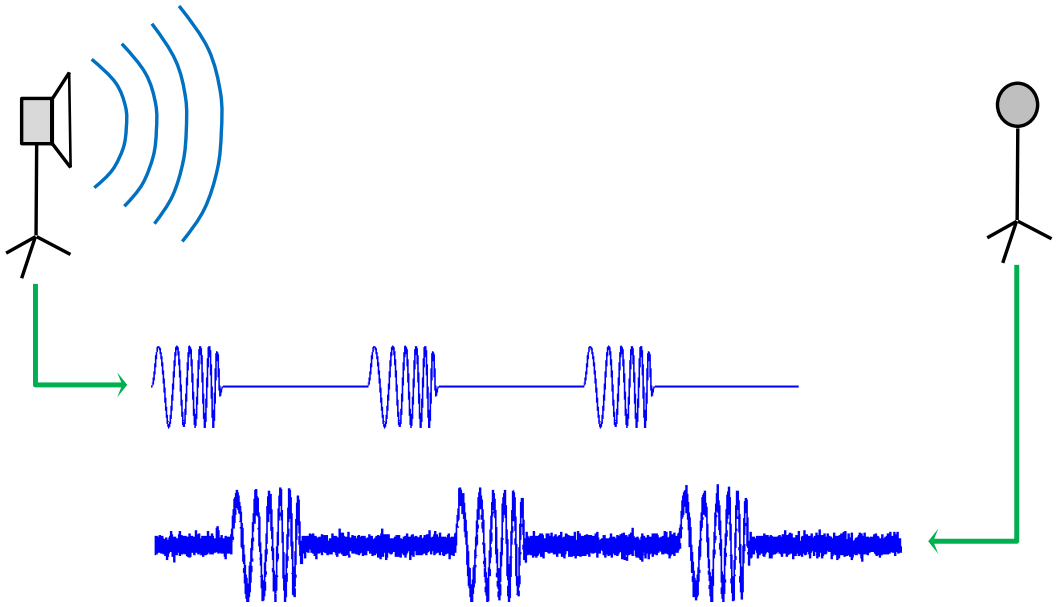
Generating correlations by first translating into the frequency domain is simpler to program and generally faster in execution. The natural result is a

circular correlation because of the implied periodicity of the functions. The unit circle provides an alternate view of the circularity of a correlation generated from the inverse transform of a spectral density. In the figure below, $N = 12$. The unshfted series is represented by the green arrow that runs from the $\tau = 0$ point to the $\tau = 11*\Delta t$ point. The shifted series (shifted by 4 points) is represented by the blue arrow. Notice that the blue arrow wraps back around the zeroeth point and also overlaps the first four points of the unshifted series.

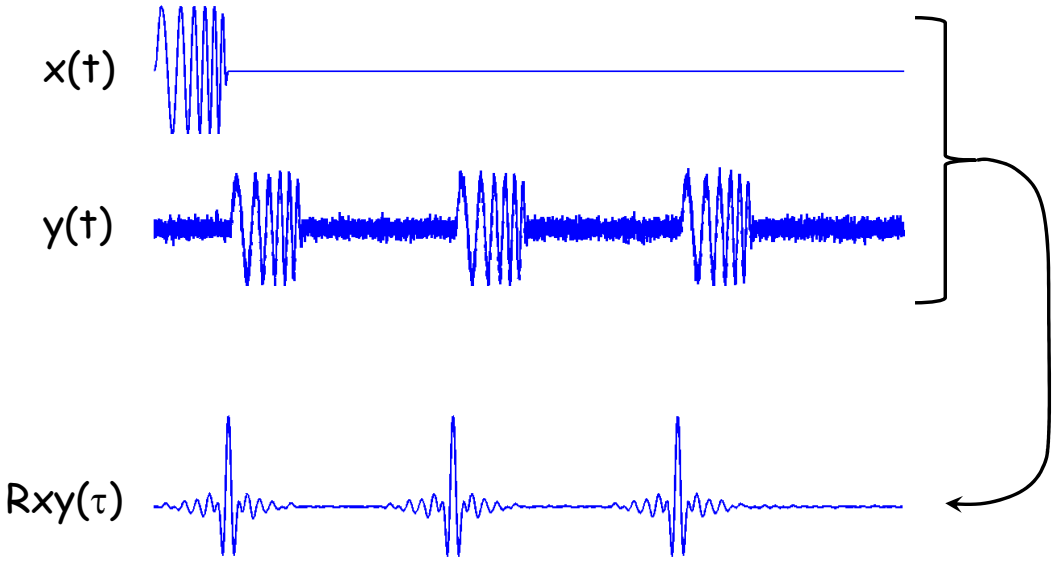


If an ordinary correlation (non-circular) is required, then add N zeros to both time series (making them $2T$ long in time), calculate the correlation from the spectral densities, and discard delay times greater than T.

The ability to visualize this circularity is valuable in designing correlation processes. Circular correlation can always be avoided by doubling the lengths of the time series but this extension in length is not always necessary. Consider an experiment in which short pulses are sent from a transmitting transducer and received some distance away.



Computing the correlation between the transmitted waveform (three pulses in the above example) and the received waveform is often not be useful as it simply returns an average of all of the individual transmitted-pulse-to-received-pulse correlations. If the intent is to study the individual pulse receptions, then we would correlate between a *single* transmitted pulse (often called a "replica") and the string of received pulses.



The resulting cross-correlation (bottom trace above) gives the relative arrival times of each of the received pulses and the amplitudes of the correlation peaks gives an estimate of the individual received-pulse levels.

Because the replica waveform has a long interval of zero values and there is a more-than-replica-length no-arrival-interval after the last received pulse, the wrap-around that results from the implied periodicity doesn't cause the replica to run into the first received pulse after passing the last received pulse. This is apparent if you picture the *x* and *y* waveforms on strips of paper then rolled into cylinders, stacked one on top of the other and shifted (by rotation) over the entire range of correlation shifts.

Circular or non-circular correlations in practice

Time domain

Direct implementation of a correlation in the time domain would result in an ordinary correlation. To reproduce a circular correlation in the time domain, a copy of one of the signals must be appended to that signal (its length would double); however, if a circular correlation were required[11], it would normally be better to work from the spectral density where circular correlation is automatic.

Frequency domain

For computation of cross-correlation from the cross-spectral density, the two time series must have the same length. If one series is shorter than the other (a "replica" of a pulse, for example), zeros must be appended to make its length equal to the length of the other time series.

[11] For example, the maximum-length sequence, described later, is a waveform with a special auto-correlation but that auto-correlation is only achieved under circular correlation.

To avoid circular correlation[12], you may need to add additional zeros to both time series so that any overlap with the periodic extensions of the time series are overlaps in which at least one of the time series has zero value at the overlapped points.

If you are in doubt about these operations, draw a sketch of the time series including the periodic repetition. Either make a series of sketches showing various shifts or sketch each time series on a separate strip of paper and slide one past the other to see what overlaps there are and what you need to do to control the behavior in the regions beyond the ends of a single copy of each series. Visualization[13] is a powerful tool for understanding these operations.

Correlation spectrogram

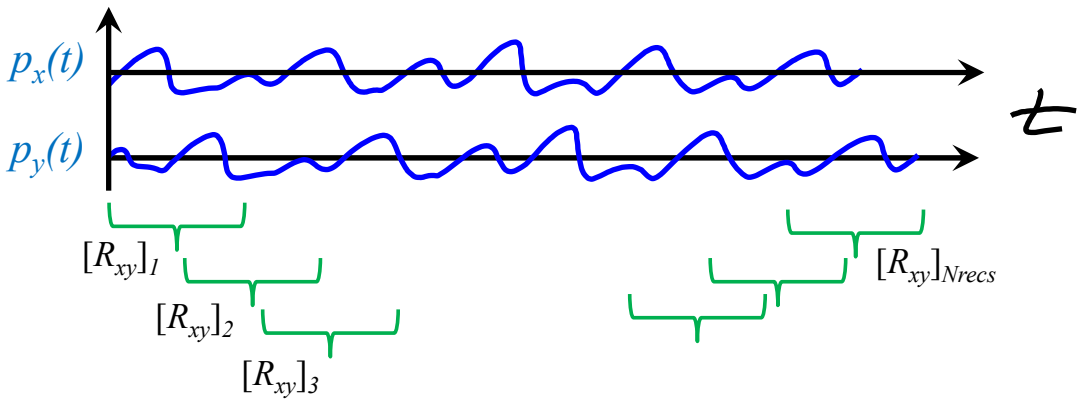
In a previous chapter, we considered the spectrogram—an image that displays signal variations in both time and frequency. The spectrogram is particularly useful as an exploratory tool; a tool to illustrate general behavior. This ordinary spectrogram is based on the spectral density (and, most often, on the logarithm of the spectral density to accommodate a large range of values). A spectrogram can also be created based on cross-correlation. In this case, we would normally use the normalized cross-correlation with no need for the data compression of the logarithm function. For normalized cross-correlation, the range of values is, at most, -1 to +1.

The horizontal axis for the cross-correlation spectrogram is the same as for the ordinary spectrogram: normal time. The vertical axis is shift time (delay time, time lag), τ, rather than frequency, with a maximum range of shift time from $-T/2$ to $T/2$.

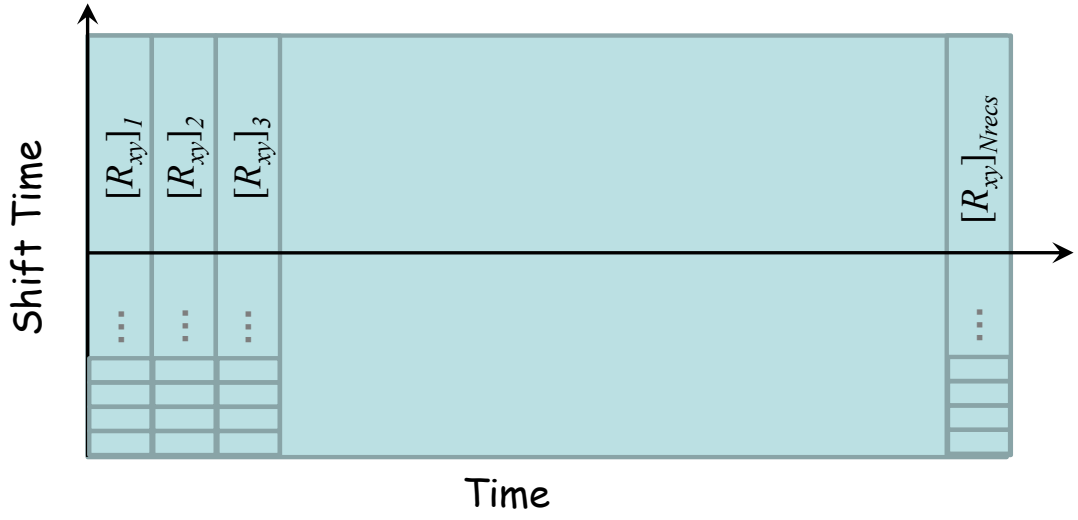
[12] To avoid the *consequences* of circular correlation, that is. All correlations calculated from the inverse transform of spectral density are circular.

[13] Many people learn more quickly from pictures, diagrams, and sketches. Develop the skills to make your own diagrams or sketches. This comes only through practice. If you can make your own visualizations, then you won't have to rely on authors and lecturers to do it for you and you can enhance your own ability to learn concepts. If you are a visual learner but have not tried to develop your own ability to make diagrams, you are hurting yourself.

The ordinary spectrogram uses a sequence of spectral densities from a single time series; the cross-correlation spectrogram uses a sequence of cross-correlations from a pair of time series,



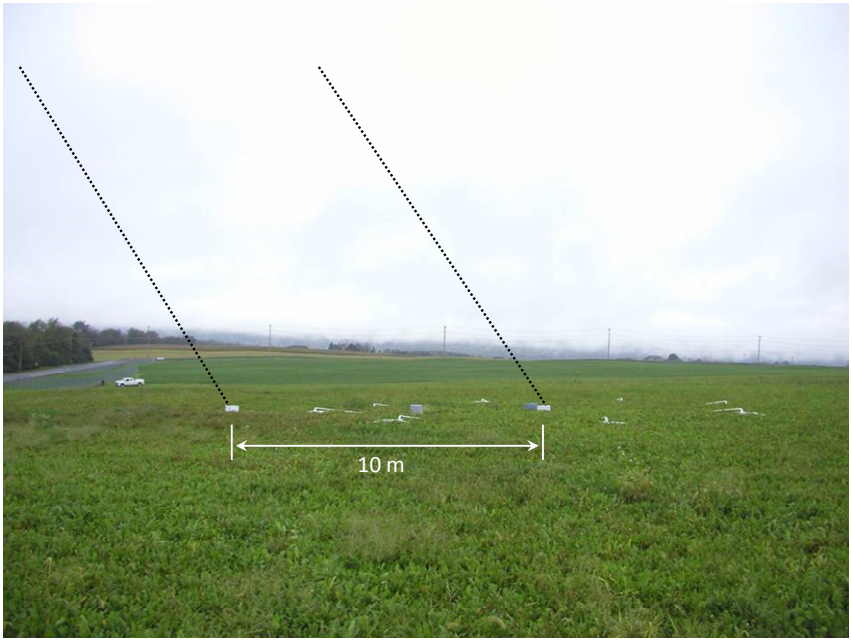
and each cross-correlation is used as a single column of pixels in the spectrogram image,



Examples of correlation spectrograms

A few examples will illustrate the correlation spectrogram and also reinforce some of the properties of the cross-correlation. Two microphones were placed

some distance from the end of the departure runway at University Park airport in State College, PA. The microphones were spaced 10 meters apart in the flight direction, sampled at 1000 samples per second, and recorded continuously for several days.



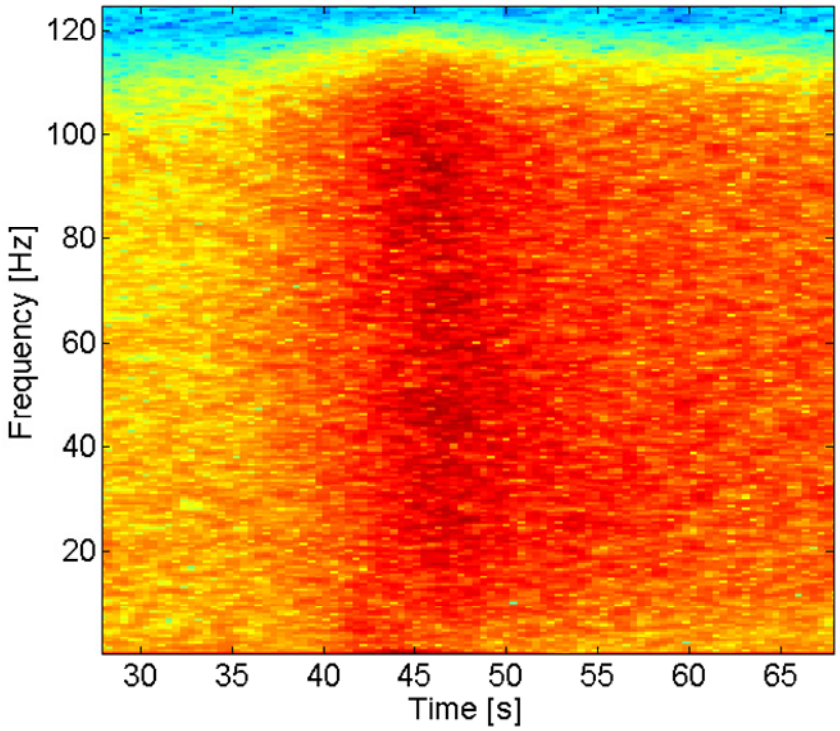
As aircraft passed directly overhead after takeoff, the paths to each microphone would be equal in length and show no delay between the two channels. Before and after being directly overhead, though, there is a delay, negative[14] before the overhead position and positive past the overhead position.

Regional jet (CRJ-200)

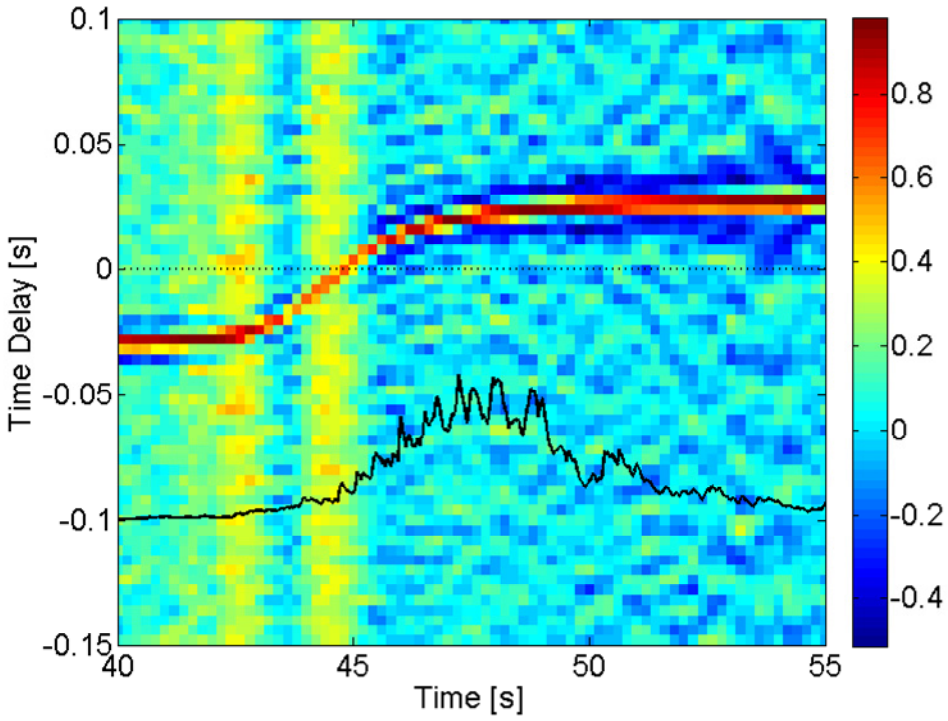
The Canadair regional jet (CRJ) is a twin-engine, short-range passenger jet. The ordinary spectrogram from 0 to 120 Hz for an overhead passage shows no tonal or line structure. On the color scale selected, blue represents the lowest levels and dark red represents the highest levels. (Remember, the pixel values are based on the logarithm of the spectral density.) The highest sound level occurs between

[14] Whether the delay is negative or positive depends on the order of the two channels in the cross-spectral density. If the cross-spectral density is S_{XY}, then a positive delay (i.e., shift time) would mean that the signal arrived at x first and then at y.

45 and 50 seconds on this time scale. There is a smooth increase in level up to this time and then a smooth decrease in level after this time. Not surprising.



In contrast, the cross-correlation spectrogram reveals a distinct feature that shows the progression of the jet with respect to the microphone pair clearly. (Ignore the black line for the moment.)



To emphasize the range of values, I have included a color bar (right side of plot) showing the correspondence of color to value. The range displayed is roughly -0.5 to 1.0 in normalized cross-correlation. The main feature is in the upper half of the spectrogram. The curved line with values approaching 1.0 shows the time delay between the two microphones. The jet is directly overhead at 45 seconds (zero delay). The delay ranges from about -0.3 seconds[15] well before overhead passage to +0.3 seconds well after the overhead point.

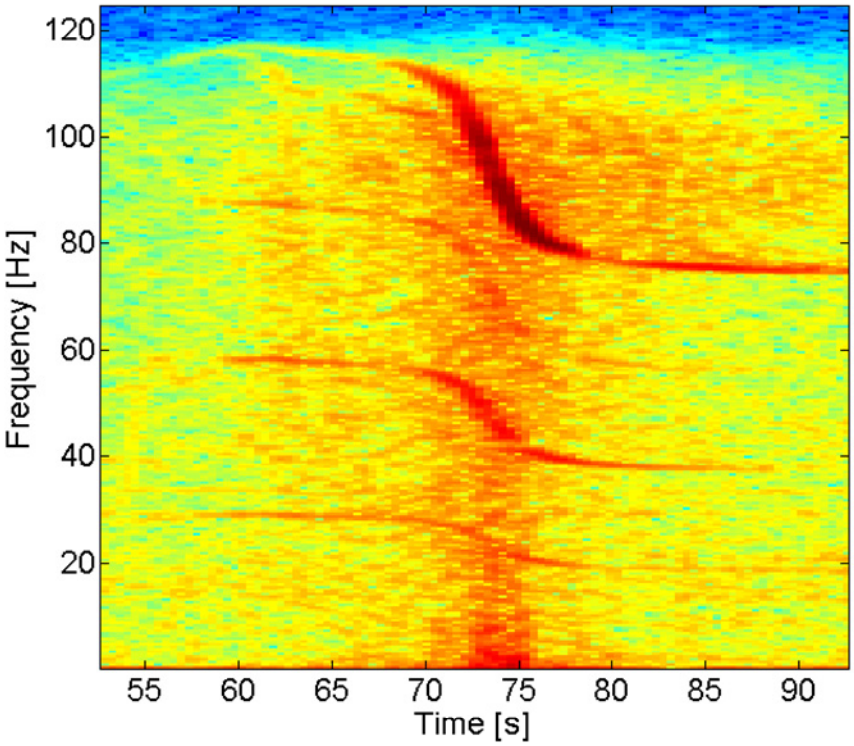
In this frequency range, the acoustic emission from the jet engines is broadband noise, which results in a well-defined correlation peak.

[The black curve is an overlay. The black curve represents the logarithm of the total power for each record as time advances as if it were plotted as power as a function of time. *Notice that the received power peaks about 3 seconds after the jet is directly overhead*. This is most likely a consequence of the directionality of sound radiation. For a jet engine, the radiation of sound peaks in the direction roughly 135 degrees aft of the flight direction.]

[15] Does this make sense given the 10 meter spacing?

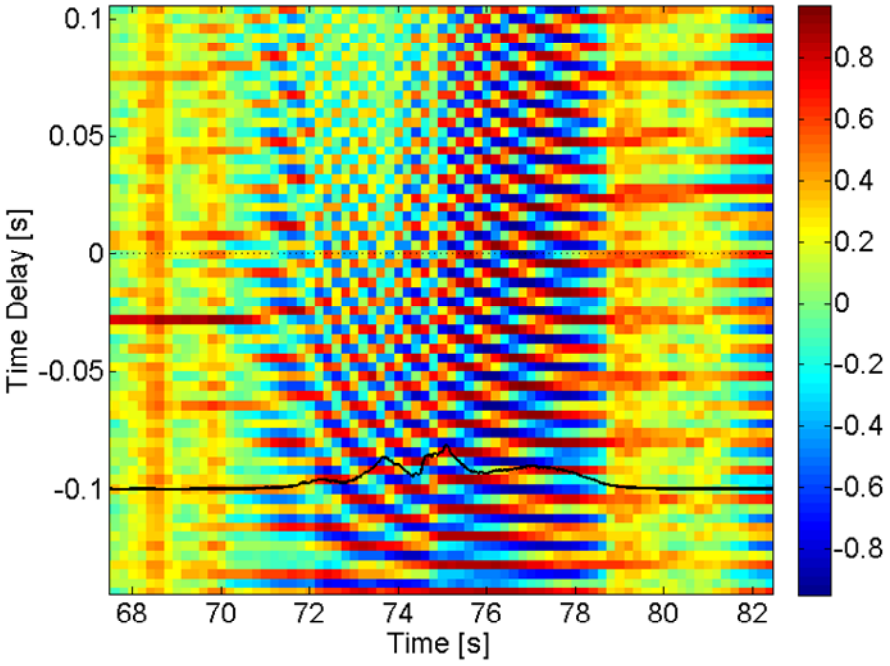
Turboprop commuter (Saab 340)

The Saab 340 is a twin-engine turboprop airplane with four-bladed propellers. Over the range of frequency examined here, the primary acoustic radiation is from the propellers and is tonal in nature. The tonal radiation appears as curved lines in the ordinary spectrogram:



The strongest line (the uppermost line) shows the blade-passage rate, which is four times the propeller shaft rate (since there are four propeller blades). The Doppler shift is evident as the airplane passes overhead.

The cross-correlation spectrogram is, on first view, confusing:



There are so many lines that it is difficult to see their trajectories. Recall that the auto-correlation of a sine wave is also a sine wave. That is what is causing the complicated pattern seen here. The strong tonals (sine waves with varying frequency) are generating correlation peaks not only at the time delay corresponding to the path difference but also at time delays that correspond to the path delays plus or minus an integer number of periods of the underlying tonal.

While the cross-correlation spectrogram is useful for the jet overflight, the complexity of the cross-correlation spectrogram for a signal dominated by tonals for the turboprop plane is not nearly as illuminating.

If a signal is a mix of broadband components and tonal components, correlation may still be useful if the signal can be filtered to remove the tonal components. Also, if you see a cross-correlation spectrogram that looks like that of the turboprop plane, you should suspect that there is a strong tonal (or tonals) in the data.

Correlation: a summary of issues

Correlation is often simpler and faster to implement as the inverse transform of the spectral density; however, the circular nature of that implementation must be respected in order to avoid spurious results related to unintended "collision" of waveforms.

As for the ordinary spectrogram, choice of record length has a significant impact on the quality of the image. Long records blur the temporal resolution; short records limit the range of observable delay times.

If the delay between common components approaches the record length, then the correlation may drop simply because the entire common component is not contained in both records. In that case, it may be worth considering using the first delay estimate to shift the time series and then extract the records again from the pre-shifted series. This process will give a better estimate of the true correlation magnitude (and should also improve the delay-time estimate if both the pre-shift and the correlation shift are considered).

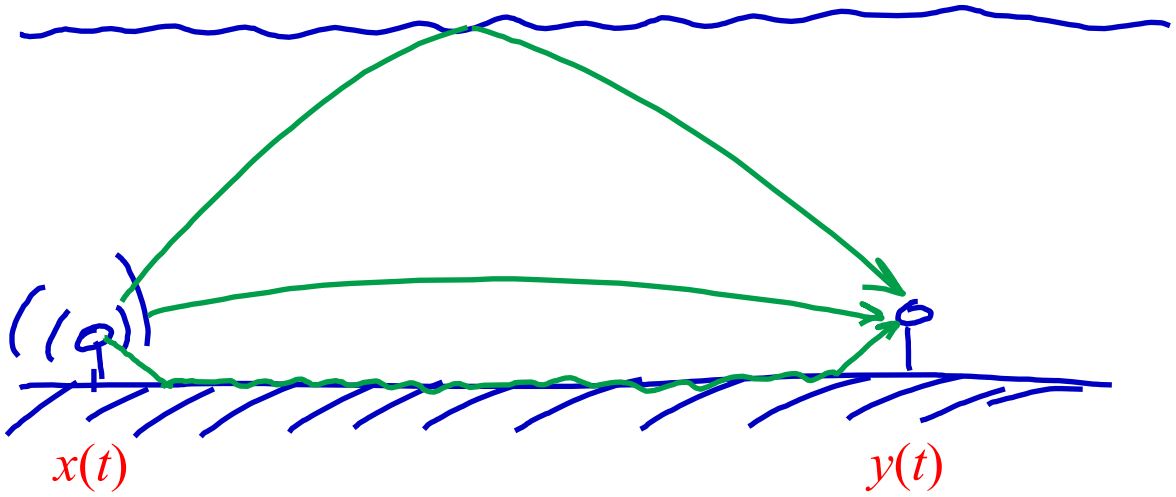
If there are strong tonals in the two time series, then the cross-correlation can be difficult to interpret. If those tonals are in a different frequency range from the desired common component, they should be removed by filtering before correlation. In fact, if the desired common component has a limited bandwidth (like a swept-sine pulse, for example), the time series should be filtered to reduce the signal power outside that band.

The difference between correlation and convolution

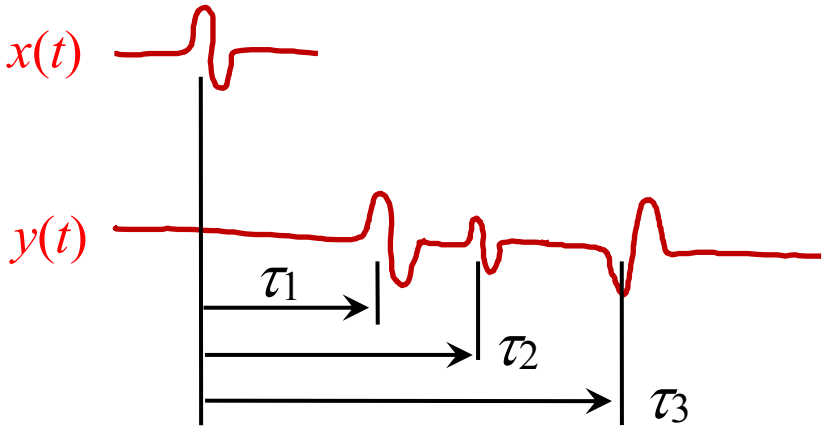
There is often some confusion between correlation and convolution. Their time-domain definitions are similar but their applications are distinctly different. In correlation, two waveforms with some relative time shift are multiplied point-by-point and then averaged for many values of shift; in convolution, two waveforms—one of which is reversed in time—with some relative shift are

multiplied point-by-point and averaged, also for many values of shift. The only difference in the mathematical expressions is the reversal in time of one waveform in the case of convolution.

The key difference in application is in the nature of the two waveforms and the intent of the process. *Correlation is a process of determining the similarity of two waveforms*. For example, if we want to locate in time multiple arrivals of a single pulse, we can cross-correlate the received signal with the transmitted pulse waveform and we should see peaks at time shifts that correspond to each arrivals travel time. If we transmitted a pulse, $x(t)$, in an environment in which there were multiple paths from source to receiver,

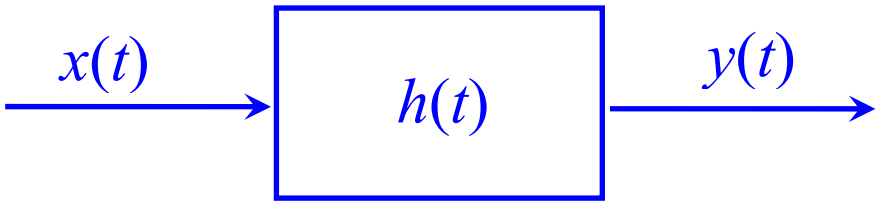


then the received signal, $y(t)$, would have several replicas of the transmitted pulse shifted in time and amplitude:

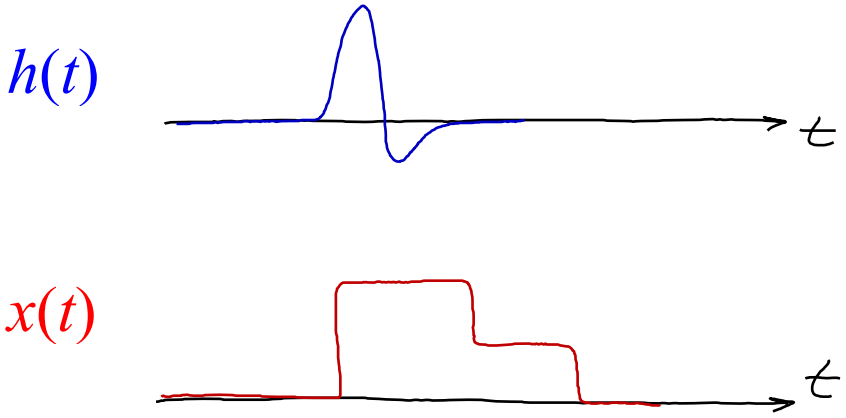


If we perform the cross-correlation between x and y, peaks would appear at the delay times corresponding to the individual path propagation times. The cross-correlation identifies regions of y that are similar in shape to x.

Convolution is a process of determining the response of some system to a simulus. If we know the response of some system to an ideal impulse, we can compute the response of that system to an arbitrary input waveform by convolving the input waveform with the impulse response.



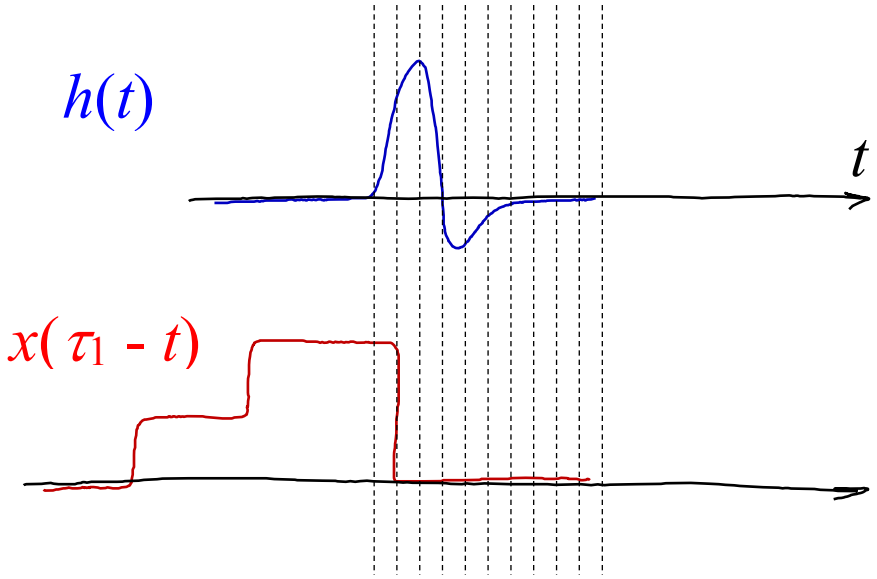
With correlation, we would study the similarity between the system output, y, and the system input, x. With convolution, we could determine the system output given the system input and the system impulse response, h.



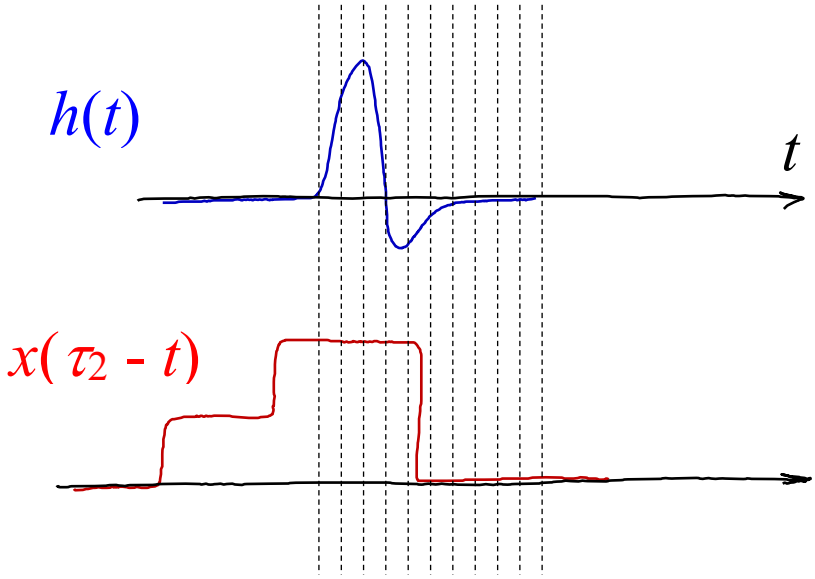
The impulse response describes a system output if the input is an impulse. We can consider that the real input signal is a sum of many impulses, one at each sample point of the input, with their amplitudes adjusted to match the input signal amplitude. Each input impulse generates a copy of the impulse response and the output is the sum of all of those impulse response copies. In order for this representation to work, the first sample of the input must "trigger" the first impulse response copy, the second sample of the input must trigger the second copy and so

on. We can arrange this by reversing the input waveform in time and sliding it past the impulse response.

Since the input waveform is reversed in time, the leading edge is the first to overlap with the impulse response:



Subsequent shifts bring the later parts of the input waveform into overlap,



For each increment of time shift, we multiply the two together and integrate. This operation is equivalent to the superposition of all of the impulse-response copies generated in the proper order. The net result is the proper response for the given input stimulus.

Transfer Function Measurements and Correlation Functions

As we've discussed earlier, one of the general types of measurements is measurement of a system's (or an environment's) complex transfer function, $H(f)$. Here, an output is related to an input by the characteristics of the system. We can express the relationship between the complex linear spectra of the input and output as

$$Y(f) = X(f) \cdot H(f) \quad . \tag{V-34}$$

If we were to choose this relationship to characterize a system, we would use an excitation (input) that had power distributed over the range of frequencies of interest. We might use a swept sine source or random noise or some repetative broadband waveform (and we might use synchronous averaging if possible).

The time domain equivalent to the transfer-function relation expresses the output time signal as the input time signal convolved with the system's *impulse response*:

$$y(t) = x(t) \otimes h(t) \quad . \tag{V-35}$$

If we chose this relationship, we would use an excitation that was sharply limited in time—an impulse or a step function, for example. And we would have to collect data for a time sufficiently long to capture the response decay ("reverberation" or "ringing" in some cases).

We also found a relationship between the complex transfer function and the auto- and cross-spectra:

$$S_{XY} \quad = \quad S_{XX} \cdot H(f) \tag{V-36}$$

This expression was particularly useful if we were forced to perform ordinary (un-synchronized) averages because we didn't have control over the excitation. Here, we would often use broadband input waveforms.

But, we can also write a corresponding time-domain form of this expression since the auto- and cross-correlations are Fourier transforms of the auto- and cross-spectra:

$$R_{xy}(\tau) \quad = \quad R_{xx}(\tau) \otimes h(\tau) \quad . \tag{V-37}$$

We'll find this expression to be useful in certain circumstances. It is especially useful if the *auto-correlation* of the excitation signal is an impulse (or nearly so). This can be useful for isolating propagation paths in a noisy environment or an environment with several significant paths of propagation.

In the ordinary time-domain convolution expression, notice that, if $x(t)$ is a delta function, $\delta(t)$, then the output, $y(t)$, is exactly the impulse response, $h(t)$:

$$y(t) \quad = \quad \delta(t) \otimes h(t) \quad \equiv \quad h(t) \quad . \tag{V-38}$$

Similarly, if the auto-correlation, R_{xx}, of the excitation function, $x(t)$, is a delta function, then the impulse response is equal to the cross-correlation, R_{xy}:

$$R_{xy}(\tau) \quad = \quad \delta(\tau) \otimes h(\tau) \quad \equiv \quad h(\tau) \quad . \tag{V-39}$$

This expression gives us yet another option for measuring the response of a system (or measuring the characteristics of a propagation path, or measuring impedance):

a) Use a signal having an impulse-like auto-correlation as the excitation;

b) Measure the output and calculate the cross-correlation, R_{xy}.

c) The resulting R_{xy} is the system impulse response; take the Fourier transform to find the system transfer function.

Practical considerations for transfer function measurements

If we want to measure the transfer function over some range of frequencies, then it's clear that we need to have an input signal that has power distributed over that range of frequencies. For classical random-data analysis, the choice is often random noise with a constant spectral density over the desired frequency range. However, this is often a poor choice for transfer-function measurement. There are a number of possible input signal types that can be used:

a. Sine wave. Since the sine wave has its power concentrated at a single frequency, we would need to use many different sine waves at different frequencies to assess the frequency dependence of a transfer function. The prinicipal disadvantage is that this process can be slow; however, there are a number of marked advantages. The amplitude of each sine wave can be adjusted depending on the system response: a large amplitude can be used at frequencies for which the response is small and a small amplitude can be used when the system response is large. In contrast, a single, random noise input signal must be adjusted so that it does not overdrive the system where the system response is large (a resonance, for example) but that means that the output may be extremely weak for frequencies at which the system response is small. In addition, synchronous averaging is straightforward with a sine-wave input. The ability to adjust the input amplitude to optimize the measurement coupled with the potential for synchronous averaging opens the door to making measurements that would be impossible or impractical using random noise. It is also much easier to assess nonlinear behavior with a sine-wave input signal – clipping and distortion are easy to detect with a sine wave but virtually impossible to detect with a random-noise signal. Some commercial signal analyzers allow setting up a stepped-sine "sweep" over some range of frequencies and, in addition, permit automatic adjustment of the input amplitude. *You should never bcome so enamoured with noise and statistical*

analysis that you forget the power of analyzing a system a single frequency at a time.

b. Chirps. Later, we will discuss pulses that sweep rapidly from one frequency to another. Such pulses can have a wide frequency range and are also suitable for synchronous averaging.

c. Impulse. The ideal impulse – a time series with only one non-zero value – has a perfectly flat spectrum. While the broadband, flat spectrum is desirable for measuring transfer functions, the impulse as an excitation is generally a poor choice. Measurement systems saturate (clip, become nonlinear) based on peak voltage or peak pressure but the ultimate signal-to-noise ratio of a transfer function measurement is related to the power in the excitation signal. The impulse has the poorest ratio of power to peak amplitude of any signal. Its utility is principally limited to theoretical calculations[16] rather than practical measurements.

d. Random noise. Typically, a random noise signal is a continuous time series in which each subsequent value has little relationship to previous values. Because the theory for white, Gaussian, stationary noise is well developed, analysis of measurements made with such noise signals is also well developed. Unfortunately, this has also led to the use of noise signals to make measurements that would be much better made with some other waveform. *Sophisticated statistical analysis will not turn a mediocre measurement into a good measurement*.

These are rather general categories, however, and it's instructive to dig deeper. One consideration is that the data acquisition system will always have an upper limit to the voltage that it will record faithfully. Many digital data acquisition systems "clip" at some voltage level (with a limit on both positive and negative voltage excursions); analog systems may clip or they may introduce unacceptable distortion from nonlinearity prior to clipping. In any case, there is a

[16] The impulse is, however, exceptionally useful for numerical testing of filters. A true impulse integrates to one so the sampled impulse would have one non-zero value equal to 1/dt. The linear spectrum of the filter's output is, then, the complex frequency response.

limit to the maximum signal excursion before distortion of the waveform becomes unacceptable.

The power distributed over the frequency spectrum is, however, related to the mean-square value of the waveform, not the peak value squared. If the peak value is far above the *rms* value, then we're forced to set the acquisition system's input range to accomodate that peak value while the low *rms* value means less power across the spectrum. Less power across the spectrum can lead to poorer measurement of the system response because we'd be closer to the unwanted noise (of the acquisition system or of extraneous noise in the measurement). So the ratio of peak value to *rms* value tells us something of the utility of a waveform for these kinds of measurements.

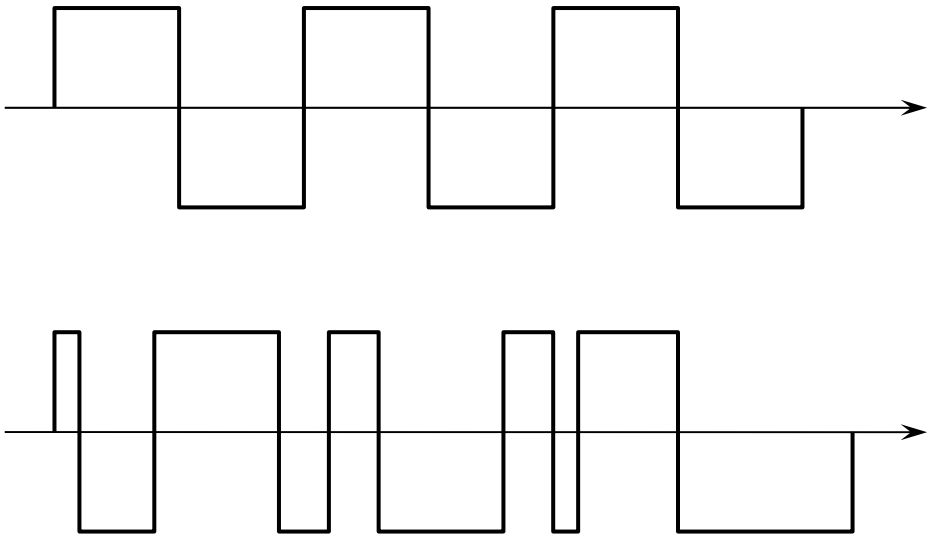
The ratio of peak to *rms* for a sine wave is $\sqrt{2}$. (Don't forget that this is only true for a sine wave!) This fact and the very narrow bandwidth of the sine wave makes it well suited to point-by-point measurement of a system's transfer function (but the process can be slow). For random noise with a gaussian distribution, the peak value can, in theory, take on any value. In practice, however, excursions of more than 4 or 5 standard deviations would be rare so we can take the peak-to-*rms* ratio to be 4 or 5.

An impulse is sometimes used to measure the system's impulse response (and then the transfer function is found from the transform of the impulse response) but the impulse has a very large peak-to-*rms* ratio. For a sampled signal, the impulse is a single non-zero value, equal to one, followed by N-1 zeros. Since all values except one are zero, the mean square value is one squared divided by N. Consequently, the ratio of peak to rms is $\sqrt{N}$. For 1024 samples, the peak-to-*rms* ratio would be 32. Not only is generating a clean impulse difficult with a real transducer; it contains very little spectral density for its amplitude.

At the other extreme is the square wave[17], which has an *rms* value equal to its peak value. The spectral distribution for a square wave is not attractive – it

[17] We'll assume that every waveform in this discussion has a zero average value ("zero-mean").

consists of a series of harmonics (odd harmonics) with amplitudes that decrease with increasing frequency – so it is not used for transfer function analysis. However, if we modify the square-wave waveform to introduce randomness, we can get a better distribution of the power spectral density. One possible modification involves shifting the positive-pulse duration and negative-pulse duration randomly from step to step:



While we could just use a random number generator to determine the width of the positive and negative excursions, there is a much better way to construct this sort of waveform.

Maximum Length Sequences

As described above, the square-like waveform has the optimal ratio of peak to rms value. Is it possible to design a waveform that altenates between positive and negative one in such a way that's its auto-correlation is exactly an impulse (so it's power distribution in the frequency domain would be perfectly uniform)?

Almost. The Maximum Length Sequence (MLS or "m-sequence") is nearly that ideal waveform. Much has been written about the MLS[18] and the theoretical development of these waveforms is arcane. In this course, I will show application of these waveforms but I won't attempt a derivation.

An MLS waveform either has the value 1 or -1 for any sample. Any MLS has an "order" N_0: the sequence of order N_0 has $N = 2^{N_0} - 1$ points. The *circular* auto-correlation of an MLS of order N_0 is almost an impulse:

$$R_{xx}(k) = \begin{cases} 1 & \text{if } k = 0 \\ -\dfrac{1}{N} & \text{if } k \neq 0 \end{cases} . \qquad \text{(V-40)}$$

At zero shift ($k = 0$), the auto-correlation is one; at any other shift, the auto-correlation is nearly zero (-1/N) for large N. The auto-correlation[19] of an MLS with large N is so close to an ideal impulse, that the MLS can be used as if it were ideal for system transfer function measurement.

> Exercise 5.3: Maximum Length Sequences
>
> The appendix on Maximum Length Sequences at the end of this chapter contains a MatLab function for generating MLS's. [Note: this function is no longer at the end of this chapter but it is included in the MatLab routines on Canvas.] Use that function to generate an MLS that is 15 points long. (What "order" is that?) Find and plot the circular auto-correlation of this sequence and show that the amplitudes are as predicted. Find and plot G_{XX} for this single record.
>
> Add a single point with a value of either +1 or -1 to this series to make it 16 points long. How does this change G_{XX} and the auto-correlation?

Further Properties of Random Noise Sequences

The maximum-length sequence has the ideal ratio of peak to *rms* (one) and it has a "perfect" auto-correlation; however, the MLS length can only be one less

[18] One of many summary articles is J. Borish and J. Angell, "An efficient algorithm for measuring the impulse response using pseudo-random noise," J. Audio Eng. Soc. **31**, 478-488, 1983.
[19] The MLS is designed to be used with *circular* correlation: Eq. III-23 is only guaranteed for circular correlation.

than an integer power of two. If we wanted to use a noise-like excitation signal but we also wanted to choose the length, what options are there? In this section, we will compare four different random-noise sequences and we'll see that we can generate a noise sequence of arbitrary length that has perfect auto-correlation. The four sequences are:

a. "Perfect" white noise: We can generate perfectly white noise by generating the linear spectrum first and then transforming to the time domain. Generate the positive-frequency values of the linear spectrum by setting all of the magnitudes to one and making all[20] of the phases random (with a uniform distribution between $+\pi$ and $-\pi$). Then generate the negative-frequency values of the linear spectrum from the complex conjugates of the positive-frequency values in the reverse order. This process is discussed in more detail in an earlier chapter of these notes. The resulting time-domain signal will have a perfectly flat spectrum and, consequently, an auto-correlation with a large peak at zero time shift and very small equal values at all other shifts.

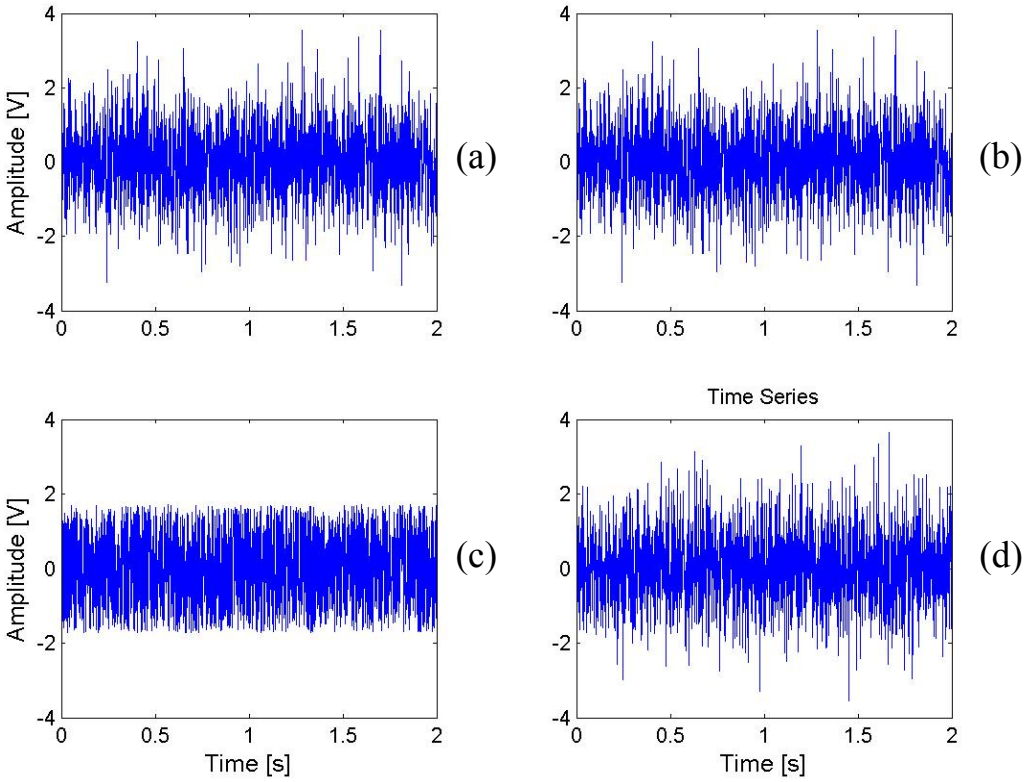
b. Perfect white noise with a defect: Having generated perfectly white noise, it is instructive to see how sensitive the sequence is to perturbation. For this example, we will simply drop one point (the last point) in the sequence.

c. Noise produced by a uniform-distribution random-number generator: This sequence is produced by the MatLab *rand* function.

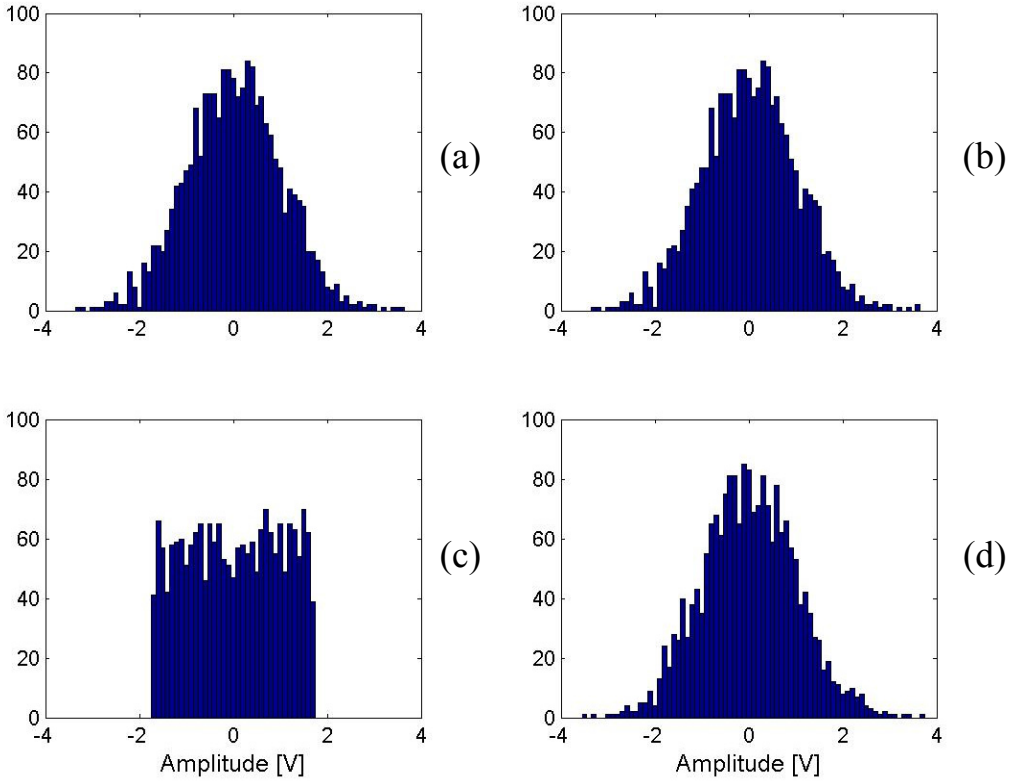
d. Noise produced by a normal-distribution random-number generator: This sequence is produced by the MatLab *randn* function.

Typical time series plots for these four types of noise are shown below. The plots are arranged so that types *a* and *b* are in the top set and *c* and *d* are in the bottom set:

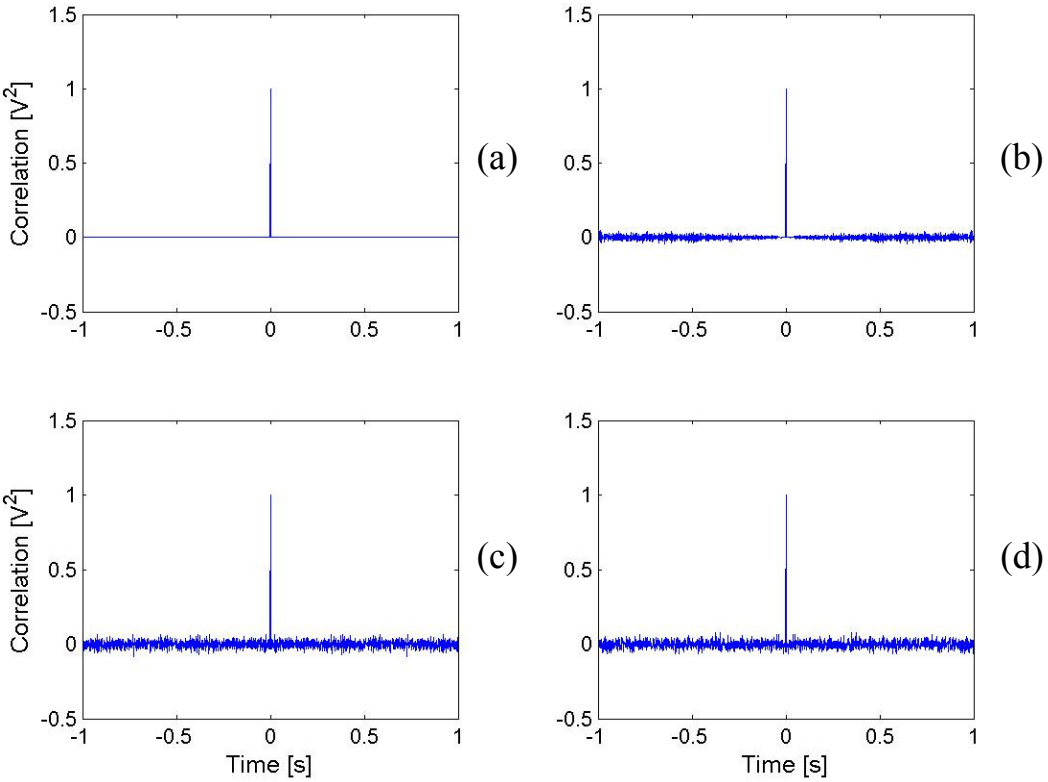
[20] Except, of course, for the values that must be real. The zero-frequency value must be real and, if the sequence has an even number of points, the $f_s/2$ point must also be real. The phase in those two cases would be zero.



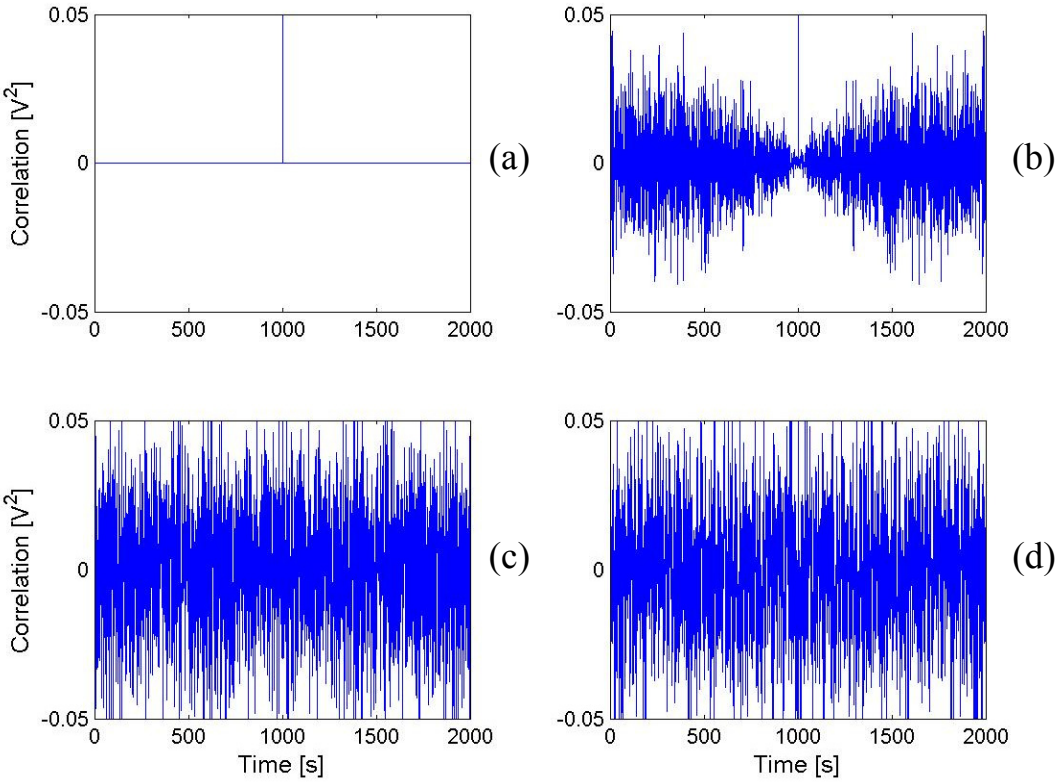
The only noise type that is clearly different is the one produced by the random-number generator with uniform distribution (*c*). All of these noise sequences are scaled to have the same *rms* value. Amplitude histograms for the four noise types are shown in the next figure. With the exception of (*c*), the histograms appear to be at least roughly normally distributed. Notice that there was no explicit use of the normal distribution in the construction of (a) or (b); the phases used to construct the linear spectrum for (a) were distributed uniformly.



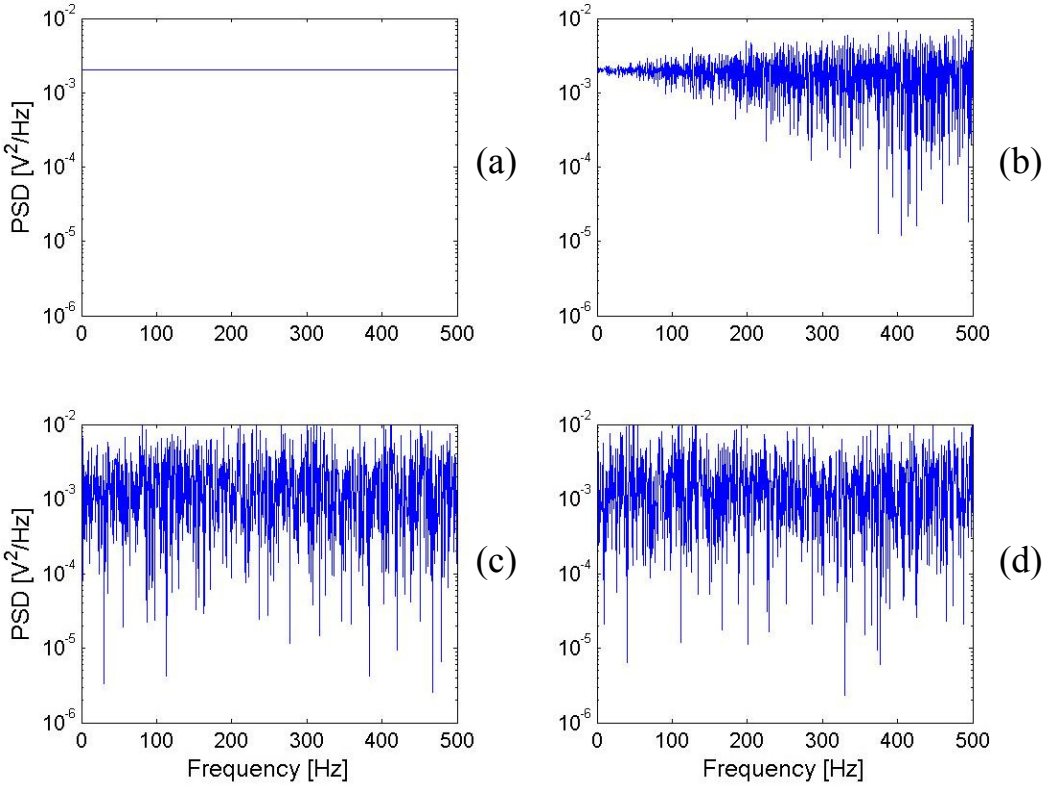
Of more interest to us are the auto-correlations and the spectral densities. The following set of plots shows the auto-correlations computed for each of the four noise types. The values at zero time shift are all equal (because the *rms* values of all four sequences were forced to be equal); however, the behavior for non-zero time shifts varies. The non-zero time-shift values appear to be very small for (*a*), the values increase with increasing time shift for (*b*), and the non-zero time-shift values have an apparently constant range of amplitudes regardless of time shift for both (*c*) and (*d*).



If we display a much smaller range of amplitude values (± 0.05 times the peak at zero time shift), we can see that the non-zero time-shift values for (*a*) are, in fact, much smaller than any of the other noise types.



The "perfect" correlation of (*a*) is reflected also in the next set of plots. The spectral density of (*a*) is constant for all frequencies. This was guaranteed by the method used to generate the noise. Notice that, for both the auto-correlation and the spectral density, removal of a single point from the "perfect" noise sequence has a substantial impact. (This is why we can't "fix" the MLS by adding a single point to make the length an integer power of two; the correlation and spectral properties are sensitive to small perturbations.)



Whereas "ordinary" random noise has a statistically flat spectrum and a reasonably sharp auto-correlation peak, it is often useful to remember that you can generate a noise sequence that has virtually perfect properties. Furthermore, by repeating the same sequence over and over, synchronous averaging can be used if necessary.

This "perfect" noise still has a relatively large ratio of peak to rms (in practice, peaks larger than five times the rms value are exceedingly rare) and the MLS also has the advantage that it can be generated with great accuracy with purely digital electronics; however, both techniques are valuable and can be considerably more powerful than simple random noise sequences.

Exercise 5.4: Frequency Response of a Digital Filter

We will discuss time-domain filters in a later chapter. For now, accept the idea that coefficients can be generated and applied in the time domain to produce a particular frequency response. For example, the following code in MatLab implements a particular frequency response by filtering a time series, xx:

```
aa = [1.0, -1.365921, 0.931704];  %  The filter coefficients

bb = [1, 0, -1]*0.034148;  %  Don't round off the coefficient values!

yy = filter(bb, aa, xx);  %  The filtering operation
```

This filter is a narrow-band filter with peak response at 0.125 times the sampling frequency.

Instead of averaging, use a single record and find the filter's frequency-domain transfer function by dividing the two complex linear spectra: H = X/Y. For each of the input time series below, plot the magnitude and phase of the transfer function:

a) an impulse: a value of (1/dt) followed by 2047 zeros (N = 2048)

b) an MLS of order 11 (N = 2047)

c) "perfect" white noise (or ordinary white noise using *randn*) of length, N = 2048.

Pick whatever sample rate you like. Notice that the result from the impulse is very clean while the results from the other two have considerable scatter. This results because the time-domain filter has a transient (or "start-up") response. For the noise-like inputs, a simple way to allow the filter's transient response to die out is to double the input time series. Take the original time series and append an identical copy to make it twice as long. Filter that time series and then throw away the first half of both the input and the output time series before computing the response. Try this with inputs (b) and (c). The resulting transfer function is much cleaner.

Exercise 5.5: Dynamic Range in Transfer-Function Measurements

If you did the previous exercise, you may have wondered why the MLS or the white noise had any advantage at all. The impulse generated a clean transfer function and did not require the "trick" of doubling the sequence. In fact, the impulse is a convenient input signal for evaluating the transfer function of digital filters when the operations are strictly computation – that is, with no noise except for round-off errors in the computations.

In typical measurements, however, the recordings will have specific amplitude limits and there will be additional noise in the output signal. Repeat the previous exercise but consider the practical limits of a data-acquisition system: suppose that your recording system would "clip" at ±1 unit. Generate the three excitation signals as you did in the previous exercise but scale the signals so that the peaks do not exceed ±1. (Use the double-length technique for the MLS and noise.) After passing a signal, *xx*, through the filter but before computing the transfer function, add a small amount of noise to the output, *yy*. Generate the noise with the *randn* function but multiply the result by 0.001 so that the added noise is very small.

Compare the frequency response (magnitude and phase) for each of the three excitation signals. The results show why the impulse is a poor choice for real measurements. The impulse has all of its energy in a single point and the amplitude of that point is limited by the range of the recording system.

Simple shift in time

Among other attributes, correlations estimate time shifts. An important basic concept is the relationship between the time- and frequency-domain expressions of a simple time delay. If a time function, z, is delayed by t_0 seconds from a time function, x, we can write z as,

$$z(t) = a \cdot x(t - t_0) \quad , \tag{V-41}$$

where a is an amplitude change. The shape of z is identical to x; z is a shifted and scaled version of x. In the frequency domain, this equation translates to

$$Z(f) = a \cdot X(f) \cdot e^{-j2\pi f t_0} \quad . \tag{V-42}$$

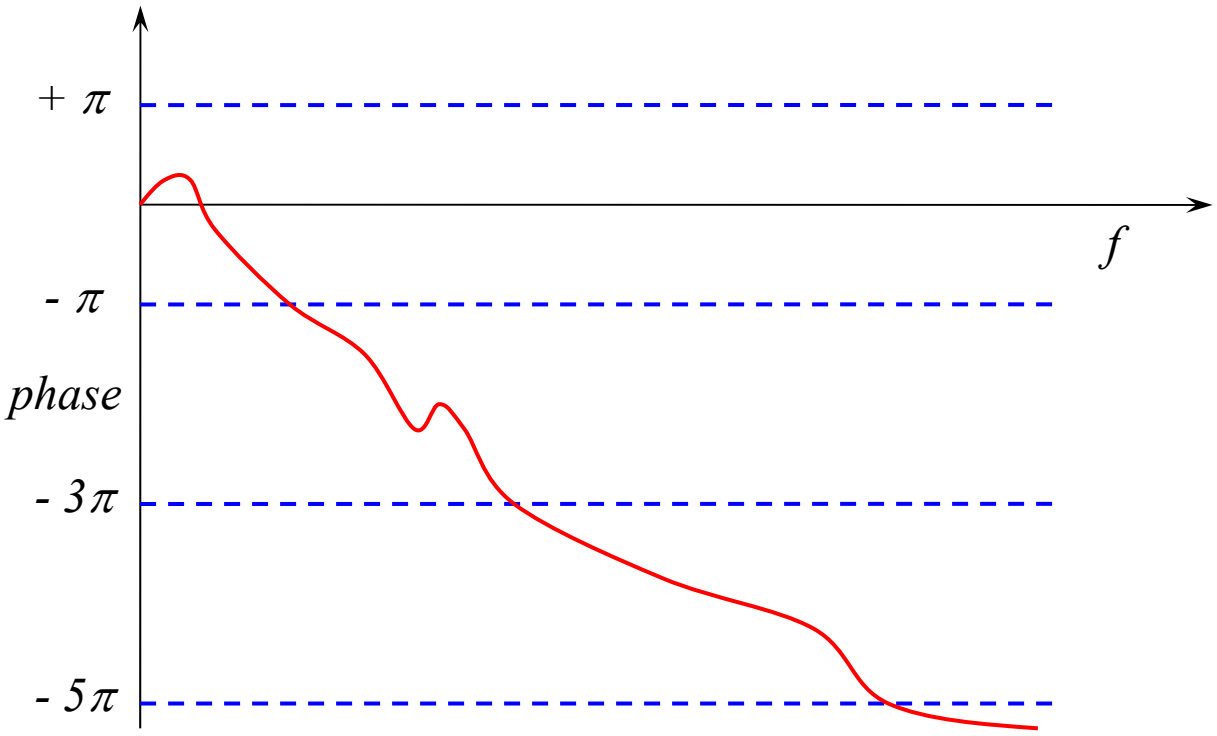
Consequently, a simple shift or delay in time is equivalent to a linear phase shift in the frequency domain. If you know the frequency dependence of the phase (either by measurement or from theory), then the delay time or shift time can be determined from

$$\tau_{delay} = -\frac{1}{2\pi}\frac{d\phi}{df} \quad . \tag{V-43}$$

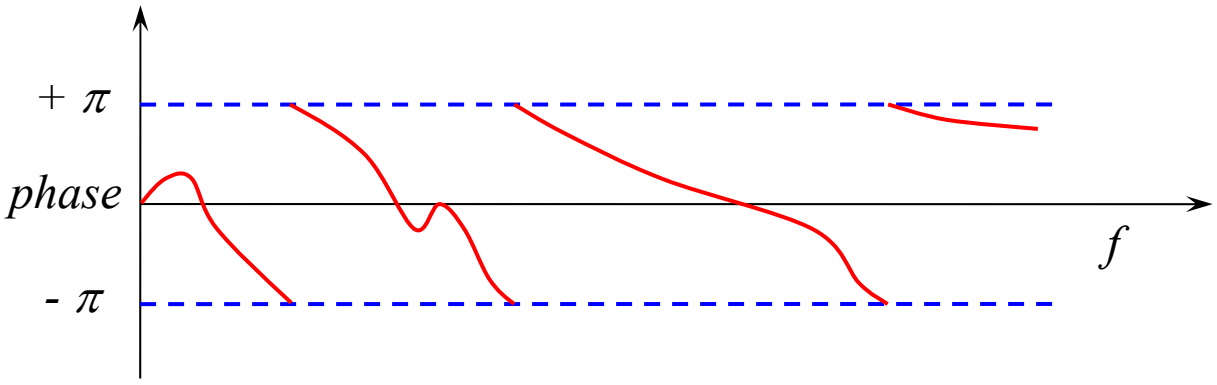
If you multiply the complex linear spectrum of a signal by the complex exponential in which the phase (the argument of the exponential) is a linear function of frequency ($2\pi f t_0$, in this example), the result in the time domain is a simple shift in time (in this example, by t_0).

Suppose, for example, you were computing the transfer function of a system with a significant delay between the input and output (it might include a propagation path), then the phase of the complex transfer function will have a linear "ramp" component. That ramp component may be steep enough that interesting phase variations in the response are obscured. This is especially true if the phase is not "unwrapped." For example, the phase of the complex number, 2 + j2, could be $\pi/4$, $\pi/4+2\pi$, $\pi/4+200\pi$, or $\pi/4-10\pi$, a few of the infinite number of possibilities. Adding or subtracting any multiple of 2π does not change the real and imaginary parts of the complex number. When using the *angle* function in MatLab to find the phase of a complex number, the value returned will always be the so-called principal value, a value between $-\pi$ and $+\pi$. A significant propagation delay can easily cause the "true" range of the phase to exceed greatly the 2π span of principal values. So you will often see what appear to be discontinuities in the phase but these discontinuities are merely the jumps of 2π required to bring the phase back into the principal-value range. The MatLab *unwrap* function is sometimes successful in removing these jumps (as long as the principal-value jumps are not obscured by noise in the phase function).

If the actual phase as a function of frequency looked like this:



the result of a plot of the angle function applied to the complex linear spectrum might look like this:



Understanding this characteristic of phase and the association of a phase linear in frequency with a simple time delay can aid in interpretation of phase plots.