Correlation and Convolution

(Revised Version)

This lecture describes the computation of correlation and convolution and some simple applications.
After this lecture you should be able to:
(1) compute the circular and linear auto- or cross-correlation of two sequences, (2) compute circular and linear correlation/convolution using the DFT (FFT), (3) compute the energy spectrum of a finite-duration signal from its auto-correlation function, (4) use auto-correlation to estimate the period of a signal and cross-correlation to estimate the time difference between two signals, and (5) use auto-correlation to estimate the power spectrum of a random signal and cross-correlation to estimate whether two signals are correlated.
This material is covered in Chapter 4 of the textbook.

Correlation

The term *correlation* is informally used to denote similarity between two things. For example, "there is a correlation between a weak economy and unemployment." In signal processing the term has a more precise meaning. If we have two N-point sequences $x_1(n)$ and $x_2(n)$, their correlation is given by:

$$r_{12} = \frac{1}{N} \sum_{n=0}^{N-1} x_1(n) x_2(n)$$

Exercise: What is the correlation between the sequences $\{-1, 1, -1, 1\}$ and $\{-1, 1, -1, 1\}$? The correlation between $\{1, -1, 1, -1\}$ and $\{-1, 1, 1, -1\}$?

Cross-correlation Function

In many cases we are interested in how the correlation changes when one sequence is delayed with respect to the other by j samples. This *cross-correlation function* is defined by:

$$r_{12}(j) = \frac{1}{N} \sum_{n=0}^{N-1} x_1(n) x_2(n-j)$$

where the delay j is called the *lag*.

Exercise: Compute the cross-correlation between the sequences $\{0, 1, 1\}$ and $\{0, 1, 1\}$ for lags of 0, 1 and 2. First assume the sequences are periodic with period 3, then assume they are of finite duration.

Autocorrelation Function

The correlation between a sequence and itself delayed by j samples is known as the *autocorrelation function* (ACF):

$$r_{xx}(j) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) x(n-j)$$

Exercise: Find the ACF for all lags for the finite sequence {1,1}.

Properties of the Autocorrelation Function

The ACF has some useful properties.

Maximum Value

The maximum of the ACF is at a lag of zero:

$$r_{xx}(0) \geq r_{xx}(j)$$

Power

The average power of a signal is the value of the ACF at a lag of zero:

$$P_x = \frac{1}{N} \sum_{n=0}^{N-1} x^2(n) = r_{xx}(0)$$

Even Symmetry

The ACF has even symmetry:

$$r_{xx}(j) = r_{xx}(-j)$$

Exercise: Some books define the ACF as

$$r_{xx}(j) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)x(n+j)$$

is this correct?

Power Spectrum

The power spectrum of a signal is the Discrete Fourier Transform of its autocorrelation function (the energy spectrum if x is of finite duration).

$$|X(n)|^2 = \text{DFT}\{r_{xx}(n)\}$$

Exercise: $|X(n)|^2$ is, by definition, real. What does this imply about $r_{xx}(n)$? Is this consistent with the previous property?

Correlation and Convolution

Correlation and convolution are similar operations. We saw earlier that the [circular] convolution of two sequences was defined as:

$$x_1(n) \circledast x_2(n) = \sum_{n=0}^{N-1} x_1((k)_N)x_2((n-k)_N)$$

which is the same as the [circular] correlation except that the sequence x_2 has been time-reversed in the case of convolution.

Computing Correlations using the DFT

We know that [circular] convolution of two sequences is equivalent to multiplication of their DFTs. We also know that time-reversing a sequence has the effect of complex-conjugating its DFT. Therefore correlating two functions is equivalent to multiplying the transform of one sequence with the complex conjugate of the transform of the other:

$$r_{12}(n) \rightleftharpoons X_1(k)X_2^*(k)$$

Computing the correlation or convolution directly requires O(N^2) multiply or add operations. Computing the correlation or convolution using FFTs and the above relationship requires O($N \log_2 N$) operations for each FFT and N multiplies. For large N the FFT method (called fast convolution or fast correlation for obvious reasons) is much faster.

Zero Padding to Compute Linear Correlation

The DFT can only be used to compute circular convolutions or circular correlations because the "time" sequences are implicitly periodic. If we want to compute a linear convolution or correlation we need to add zero-valued samples to each sequence ("zero pad") and extract only those values which don't exhibit the "wrap-around" effects of circular convolution/correlation.

Since the duration of the linear convolution of sequences of lengths N_1 and N_2 is $N_1 + N_2 - 1$, we must pad both sequences with zeroes to at least this length before computing the DFTs.

Exercise: We wish to use fast convolution to filter a sequence of 500 samples with a filter whose impulse response is 12 samples long. What is the shortest (power-of-2-length) FFT that we could use to do this?

Fast Convolution/Correlation of Long Sequences

Fast convolution is often used to filter continuous sequences of data. To do this the input sequence is segmented, convolved with an M-sample impulse response and the result is output. However at the boundary between segments there will be $M-1$ values that depend on samples from two different adjacent segments. Two methods are used to properly compute the values of these samples.

In one method ("overlap-save") the $M-1$ values that would have been incorrectly computed are discarded and the last portion of each segment is saved and pre-pended to the next segment. In another method ("overlap-add") the $M-1$ incomplete output values from the end of one segment are added to the first $M-1$ samples of the next segment to obtain the correct values.

ACF of a Random Source

The autocorrelation function is often used characterize random sequences. Like other statistics (e.g. the distribution), the ACF helps describe the properties of a random variable. The ACF of a random process is given by:

$$R_{xx}(j) = E\{x(n)x(n-j)\}$$

where $E\{\cdot\}$ is the *ensemble* average, that is, the average over all possible sequences that could be generated by the random process.

The ACF of a sequence computed as described previously is often used to *estimate* the unknown ACF of a stochastic processes (a random variable that is a function of time) in the same way that the sample[1] mean is used to estimate the expected value.

It's important not to confuse the ACF of the underlying random process (which is fixed and often unknown) and the ACF of one sequence generated by that process (which will vary from sequence to sequence and can be computed exactly).

ACF of the sum of two signals

If we have a sequence which is the sum of two signals,

$$y(n) = x_1(n) + x_2(n)$$

the ACF of the sum can be found to be

$$r_{yy} = \sum_{n=0}^{N-1} (x_1(n) + x_2(n))(x_1(n+j) + x_2(n+j))$$

$$= r_{11}(j) + r_{21}(j) + r_{12}(j) + r_{22}(j)$$

If x_1 and x_2 are *independent* (or *uncorrelated*, a weaker condition) then $E\{r_{12}\} = E\{r_{21}\} = 0$. This is often the case when one of the two signals is additive noise due to things such as coupling of unwanted signals onto sensors, noise added by electronic circuits, quantization errors, etc. In this case

$$E\{r_{yy}\} = E\{r_{11}(j)\} + E\{r_{22}(j)\}$$

Furthermore, if x_2 is a *white* noise source (the samples are independent (or uncorrelated) with each other) then ($E\{r_2 2(j)\} = 0 \text{for} j \neq 0$). Therefore $r_{yy}(j)$ can be used as an estimate for the ACF of a noisy waveform $x_1(n)$. In practical terms this means the ACF of a sequence can be used to estimate the ACF of a signal when it is corrupted by additive white (or uncorrelated) noise.

Just as when we estimate the expected value from the mean of a sample, the "quality" of our estimate (the probability that the ACF we compute from a particular sequence will be close to the underlying ACF) will improve as the number of samples in the sequence increases. We can improve the quality of our estimate by increasing the length of the sample sequence that we use to compute the ACF.

Another conclusion that we can draw from the above is that the expected value of the power of the sum of two signals (their ACFs at lag of zero) will be the sum of their powers if the two signals are uncorrelated.

Applications

Correlation is often used to estimate delay-related parameters of random or noisy signals.

Estimation of Period

From the expression for the ACF we can see that the ACF of a periodic sequence will also be periodic with the same period.

Exercise: Prove this.

We can therefore compute the ACF of a signal and then estimate the period from the ACF. The period could be computed by looking for peaks in the ACF (which we know occur at lags of 0 and multiples of the period) or by computing the spectrum (e.g. by using a DFT) and looking for peaks in the spectrum.

You might wonder why we would bother to do this. Since both the ACF and the original signal are periodic, why not estimate the period from the original signal? The reason is that, as we saw above, if the

[1] Here we mean the a sample of a random process rather than a sample of a function of time.

signal is corrupted by additive white noise we can get a better estimate of the ACF of the underlying signal by computing the sample ACF over longer and longer sequences. The process of computing the ACF allows us to “average out” the noise and obtain a better estimate of the period.

Delay Estimation

Many system (e.g. radar and sonar) work by measuring the delay between a transmitted signal and its echo. The echo is often weaker than the transmitted signal and corrupted by added noise that is uncorrelated with the original signal. If we compute the cross-correlation between the received and the transmitted signal (presumably scaled versions of each other) we should get maximum correlation for a lag equal to the true delay.

The use of the cross-correlation sum allows us to obtain the same “noise-canceling” effect as when estimating the ACF of a noisy signal. We will obtain best results from such a systems by transmitting a signal whose autocorrelation is strongly peaked. One such signal is an impulse, but there are other signals that have this property (e.g. the PN sequences described in the text on page 205).