DIGITAL SIGNAL PROCESSING LABORATORY
Assignment 6

1. Correlation

In some applications, such as radar, a system receives a known information signal that is masked by random noise. For example, in a radar system, a pulse is transmitted by the radar, then reflected by a target and received by the radar. The returning radar pulse is usually masked by random noise. The goal of this type of system is to detect the signal. Correlation provides one such tool for this task.

To approximate a vector **A** by another vector **B** or to determine how much of vector **B** is contained in vector **A**, let

$$\mathbf{A} \approx k\mathbf{B}$$

or

$$\mathbf{A} = k\mathbf{B} + \mathbf{V}_e$$

where k is a scalar and $\mathbf{V}_e$ is an error vector. If we want the vector $k\mathbf{B}$ to be the best approximation of the vector **A** in the sense that the magnitude or the magnitude square of the error vector, $\mathbf{V}_e$, is minimized then

$$k = \frac{\mathbf{A} \bullet \mathbf{B}}{\mathbf{B} \bullet \mathbf{B}}$$

where • represents the dot product or inner product operation. Thus, the component of vector **A** along vector **B** is $k\mathbf{B}$. Therefore the magnitude of k is a measure of the similarity of the two vectors. If **A** and **B** are orthogonal, then $\mathbf{A} \bullet \mathbf{B} = 0$, and $k = 0$. Therefore if **A** and **B** are orthogonal, none of vector **B** is contained in vector **A**.

Signals may be compared in a similar manner. To approximate a sequence $x[n]$ by another sequence $g[n]$ or to determine how much of the sequence $g[n]$ is contained in the sequence $x[n]$ over the interval $0 \le n \le N\text{-}1$, let

$$x[n] = kg[n] + \mathrm{e}[n]$$

where k is a scalar and $e[n]$ is an error sequence. If we want the sequence $kg[n]$ to be the best approximation of the sequence $x[n]$ over the interval $0 \le n \le N\text{-}1$ in the sense that the mean of the square of the error sequence is minimized then

$$k = \frac{x[n] \bullet g[n]}{g[n] \bullet g[n]} = \frac{\sum_{n=0}^{N-1} x[n]g^*[n]}{\sum_{n=0}^{N-1} g[n]g^*[n]}$$

where • represents the dot product or inner product operation and $g^*[n]$ represents the complex conjugate of the sequence $g[n]$. Thus, the sequence $x[n]$ contains an amount $k\,g[n]$ of the sequence $g[n]$ over the interval $0 \le n \le N\text{-}1$. Therefore the magnitude of k is a measure of the similarity of the two sequences. If $x[n]$ and $g[n]$ are orthogonal, then $x[n] \bullet g[n] = 0$, and $\mathrm{k} = 0$. Therefore if $x[n]$ and $g[n]$ are orthogonal, none of sequence $g[n]$ is contained in sequence $x[n]$ over the interval $0 \le \mathrm{n} \le N\text{-}1$. If we compare several different $x[n]$ sequences to the sequence $g[n]$, it becomes apparent that the magnitude of the numerator, $|x[n] \bullet g[n]|$, can be taken as an indication of the similarity of the two sequences.

To apply this theory to our problem, consider a radar that transmits the pulse $t[n]$ shown in Figure 1. The pulse reflects off a target and returns to the radar 20 samples later as shown in Figure 2. The two sequences are identical except that one is delayed with respect to the other. To determine the similarity between these two sequences, we calculate $t[n] \bullet s[n]$ which equals zero because the product, $t[n]\,s[n]$, is zero. Therefore $t[n]$ and $s[n]$ have no measure of similarity. This is not a useful conclusion for our application. The sequences are similar; they are just shifted in time. To test for similarity between the two sequences over time, one sequence is shifted an amount τ with respect to the other sequence. The measure of similarity then becomes a function of the shift variable, τ, and can be written as

$$\phi_{xg}(\tau) = x[n-\tau] \bullet g[n] = \sum_{n=0}^{N-1} x[n-\tau]g^{*}[n] \qquad (1)$$

where τ is an integer. This measure of similarity is known as the crosscorrelation function.

Note that the summation in Equation (1) has the form of a convolution sum and can be written as

$$\phi_{xg}(\tau) = x[-\tau] * g^{*}[\tau].$$

The value of τ for which $|\phi(\tau)|$ is maximum determines the time at which the sequences are most similar.

Once again consider the problem where a radar transmits the pulse $t[n]$ shown in Figure 1 and the return signal, $s[n]$, shown in Figure 2 returns to the radar 20 samples later masked by noise. Thus, the radar receives the signal, $s[n] + n[n]$, where $n[n]$ represents a zero mean white noise signal. Applying the correlation function to this problems yields

$$\phi_{t(s+n)}(\tau) = t[-\tau] * (s[\tau] + n[\tau]) = (t[-\tau] * s[\tau]) + (t[-\tau] * n[\tau]) = \phi_{ts}(\tau) + \phi_{tn}(\tau).$$

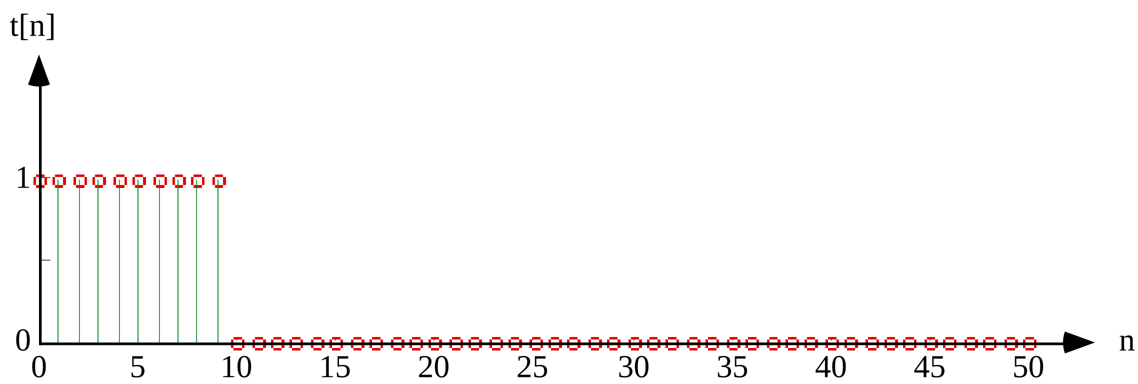

Figure 1. Transmitted Signal.

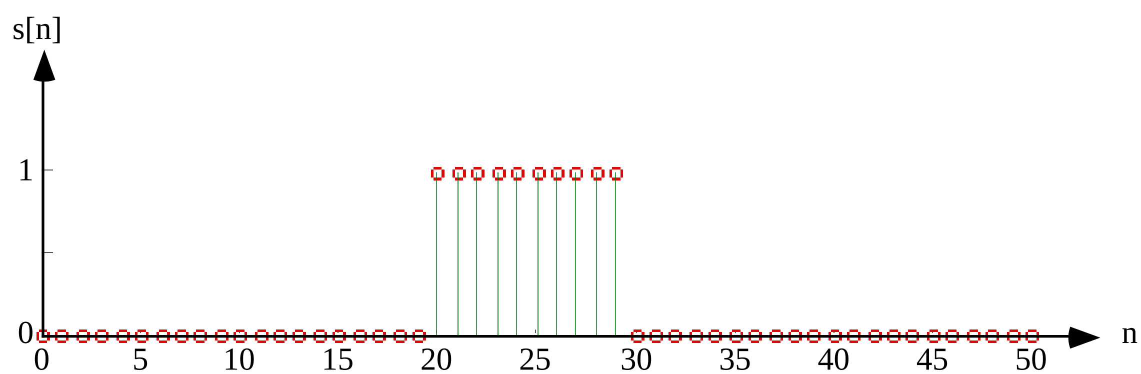

Figure 2. Received Signal

Examining this equation, it can be seen that $\phi(\tau)$ for a noisy signal is equal to the correlation of the transmitted signal and the received signal without noise, $t[-\tau] * s[t]$, plus the correlation of the transmitted signal and the noise, $t[-\tau] * n[\tau]$. It can be shown that the mean value of $\phi_{tn}(\tau)$ where $n[n]$ is a zero mean white noise source is zero. Thus the mean value of $\phi_{t(s+n)}(\tau)$ is $\phi_{ts}(\tau)$. Therefore correlation can be used to detect signals obscured by noise.

a) Consider a radar system which transmits the following signal

$$t[n] = \begin{cases} 1 & 0 \le n \le 9 \\ 0 & \text{otherwise} \end{cases}$$

and receives the signal $r[n] = s[n] + n[n]$ where

i) $s[n] = t[n-20]$ and $n[n] = 0$.
ii) $s[n] = 0.5\, t[n-20]$ and $n[n] = 0$.
iii) $s[n] = t[n-20]$ and $n[n]$ = 2*(**rand**(n)-0.5) where **rand** is set to **rand**(**'uniform'**).
iv) $s[n] = t[n-20]$ and $n[n]$ = 4*(**rand**(n)-0.5) where **rand** is set to **rand**(**'uniform'**).
v) $s[n] = t[n-20]$ and $n[n]$ = 0.5***rand**(n) where **rand** is set to **rand**(**'normal'**).
vi) $s[n] = t[n-20]$ and $n[n]$ = **rand**(n) where **rand** is set to **rand**(**'normal'**).

Assume $N = 50$. Plot $r[n]$ and $\phi_{\text{tr}}(\text{n})$ for each of the received signals. Comment on your plots.

2. 2 Real FFT's (DFT's) Simultaneously

The FFT algorithms discussed in class were derived assuming the FFT's input sequences have complex values. If the sequences of interest are real, it is possible to compute the FFT of two real N point sequences using one N point FFT computation.

a) Consider two real N point sequences, $x[n]$ and $y[n]$. Let

$$z[n] = x[n] + j\, y[n].$$

Show that

$$X[k] = (Z[k] + Z^*[[N-k]]_N) / 2$$

$$Y[k] = (Z[k] - Z^*[[N-k]]_N) / 2j.$$

b) Verify your answer by

i) generating 2 distinct real sequences, $x[n]$ and $y[n]$, of length N where $64 \le N \le 1024$,
ii) performing an FFT on $x[n]$ and $y[n]$ (plot the magnitude of $X(k)$ and $Y(k)$),
iii) performing an FFT on $x[n]$ and $y[n]$ using the method described in part a) of this exercise (plot the magnitude of $Z(k)$, $X(k)$ and $Y(k)$), and
iv) comparing your results (subtract your answers in part ii) from your answers in part iii)).