Assignment 5 - Correlation and Convolution

Due February 13, 1996

Question 1

Part (a)

This part asks for the linear and circular correlation and convolution between the two sequences: a ={1, 2, 3}, and b ={2, 1}.

Linear Correlation

The correlation will be non-zero for lags from -1 to 2. The calculations are as follows:

For $j = -1$			
n	$a(n)$	$b(n-j)$	$a(n)b(n-j)$
0	1	1	1
1	2	0	0
2	3	0	0
			$\sum = 1$

For $j = 0$			
n	$a(n)$	$b(n-j)$	$a(n)b(n-j)$
0	1	2	2
1	2	1	2
2	3	0	0
			$\sum = 4$

For $j = 1$			
n	$a(n)$	$b(n-j)$	$a(n)b(n-j)$
0	1	0	0
1	2	2	4
2	3	1	3
			$\sum = 7$

For $j = 2$			
n	$a(n)$	$b(n-j)$	$a(n)b(n-j)$
0	1	0	0
1	2	0	0
2	3	2	6
			$\sum = 6$

Linear Convolution

Similarly, the linear convolution can be computed for lags from 0 to 3 as:

For $j = 0$			
n	$a(n)$	$b(j-n)$	$a(n)b(j-n)$
0	1	2	2
1	2	0	0
2	3	0	0
			$\sum = 2$

For $j = 1$			
n	$a(n)$	$b(j-n)$	$a(n)b(j-n)$
0	1	1	1
1	2	2	4
2	3	0	0
			$\sum = 5$

For $j = 2$			
n	$a(n)$	$b(j-n)$	$a(n)b(j-n)$
0	1	0	0
1	2	1	2
2	3	2	6
			$\sum = 8$

For $j = 3$			
n	$a(n)$	$b(j-n)$	$a(n)b(j-n)$
0	1	0	0
1	2	0	0
2	3	1	3
			$\sum = 3$

Circular Correlation/Convolution

The circular convolution or correlation of two periodic sequences of different periods is not well defined. The method used in the text book (page 191) assumes that one sequence is finite and the other is periodic. In this case the result is periodic with the period of the shorter sequence.

An alternate method (that gives a different result – a constant) would compute the correlation by periodically extending both of the sequences until they were the same length (which would be the lowest common multiple of the two periods).

I can't think of a practical reason why you would have to compute the circular correlation or convolution of two sequences of unequal lengths so for the purposes of this course you can ignore this problem. In this course you will not be asked to compute the *circular* convolution or *circular* correlation of sequences of *different* lengths.

Part (b)

The Matlab commands to compute the linear convolutions above are:

```
>> a=[1 2 3];
>> b=[2 1];
>> conv(a,b)

ans =

     2     5     8     3

>>
```

which agrees with the values computed by hand. Note that the first result is the result for the smallest lag (-1), not to the result for a lag of 0.

The correlation can be computed by time-reversing the second sequence:

```
>> conv(a,b(length(b):-1:1))

ans =

     1     4     7     6
```

Part (c)

For the reasons described above, we won't deal with circular convolutions or correlations of unequal length sequences.

Part (d)

To compute the linear correlation or convolution using the DFT, we need to pad each vector with zeros to the length of the convolution/correlation: `(length(a) + length(b) - 1` = 4. The linear correlation and convolution are computed as follows:

```
>> a=[1 2 3 0];
>> b=[2 1 0 0];
>> ifft(fft(a).*conj(fft(b)));
>> real(ans)

ans =

     4     7     6     1

>> ifft(fft(a).*fft(b));
>> real(ans)

ans =

     2     5     8     3

>>
```

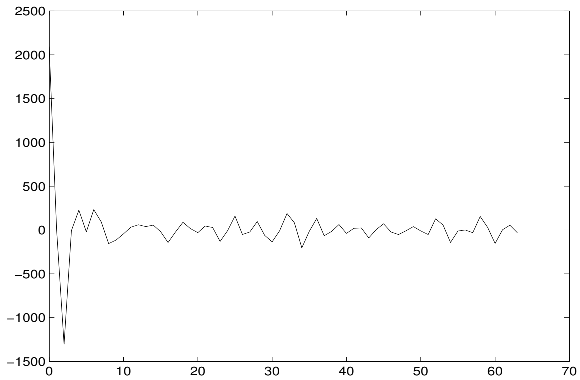
Note that in this case the results are given in order for lags from 0 to N. Since the results of the inverse DFT are periodic, the result for a lag of -1 is the same as for a lag of $N-1$.

Question 2

To compute the linear ACF of `s` using the DFT method we need to zero-pad to a length of at least 1024+1024-1. To make the DFT computations a little bit faster I'll pad `s` to a length of 2048. The following Matlab commands compute the ACF, remove the imaginary parts caused by finite precision computations and plot the first 64 values:

```
>> load asg5sigs
>> sp=[s zeros(1,1024)];
>> acf=ifft(fft(sp).*conj(fft(sp)));
>> acf=real(acf);
>> plot([0:63],acf(1:64));
>> print -deps sol5.q2a.eps
```

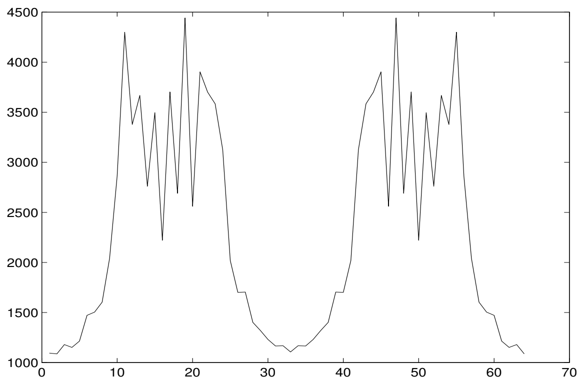
The plot of the ACF is:



The following commands compute and plot the estimated power spectrum of `s` from the first 64 points of the ACF:

```
>> psd=fft(acf(1:64));
>> plot(abs(psd));
>> print -deps sol5.q2b.eps
```

The result is:



The samples of `s` were generated by passing white noise through the filter designed in Assignment 4. Notice that: (1) the power spectrum has a shape similar to the (square of) the amplitude response of the transfer function and (2) there is some "randomness" in the ACF since this is an *estimate* of the true ACF.

Question 3

To estimate the delay we find the lag that produces the maximum value of the cross-correlation. If we use the `conv()` function we need to keep in mind that the first value corresponds to a lag of -6 since `x` has a length of 7. To find the maximum we use the `max()` and `find()` functions and the == operators:

```
>> ccf=conv(y,x(length(x):-1:1));
>> find(ccf == max(ccf))

ans =

    18

>>
```

The peak of the cross-correlation occurs at an index of 18 in the output of `conv()` function. Since index 1 corresponds to a lag of -6, index 18 corresponds to a lag of 11. We therefore estimate that the signal `y` contains the signal `x` delayed by 11 samples.

An alternate (and faster) way of computing the cross-correlation involves using zero-padding `y` and `x` to a length of `length(y) + length(x) - 1` = 27+7-1=33 and using the FFT to compute the circular cross-correlation. The following example also illustrates the use of the `sort()` function to find the index of the largest value:

```
>> xp=[x zeros(1,26)];
>> yp=[y zeros(1,6)];
>> ccf=ifft(fft(yp).*conj(fft(xp)));
>> [dummy i]=sort(ccf);
>> i(length(i))

ans =

    12

>>
```

Since Matlab uses 1-based indexing an index of 1 corresponds to a lag of 0 and the estimated delay is again 11.