CS 370 Winter 2009: Assignment 3

Instructor: G. Labahn Office: DC3629 e-mail: *glabahn@uwaterloo.ca*
Lectures: MWF 9:30 MC2038 Office Hours: Tues 2:30-3:30 PM
Office Hours: Thurs 9:30-10:30 AM

Web Site: www.cs.uwaterloo.ca/~glabahn/cs370/
Due Wednesday Mar 18, 5:00PM, in assignment boxes, 3rd floor MC

CORRECTED VERSION : March 10, 2009

Analytic Questions

1. (**5 marks**) Calculate the Discrete Fourier Transform of the following periodic time sequences by hand, both using the direct DFT formula.

(i)

$$f[n] = (2, -2, 1, -1) \quad (n = 0, \ldots, 3;\ N = 4)$$

(ii)

$$f[n] = (1, 2, 4, 8) \quad (n = 0, \ldots, 3;\ N = 4)$$

2. (15 marks) Let $\{F_0, \ldots, F_{N-1}\}$ be the DFT of a sequence $\{f_0, \ldots, f_{N-1}\}$ with

$$F_k = \frac{1}{N} \sum_{n=0}^{N-1} f_n W^{-nk}$$

where $W = e^{\frac{2\pi}{N} i}$.

(a) Give a simple formula for F_k when $f_n = e^{\frac{6\pi n}{N} i} = W^{3n}$ for $n = 0, ..., N-1$.

(b) Give a simple formula for F_k when $f_n = \cos(\frac{4n\pi}{N})$ for $n = 0, ..., N-1$.

(c) Give a simple formula for F_k when $N = 4m$ and $f_n = 0$ for $m \le n < 3m$ and 1 otherwise.

3. (15 marks) Consider the sequence of eight numbers $f = [-1,\ -2,\ -2,\ -1,\ 1,\ 2,\ 2,\ 1]$. As in the previous question, please use the unnormalized definition of the DFT for this question.

(a) What are the two arrays (g and h, each of length 4) that are used in computing the DFT of f by the FFT method?

(b) Compute G and H, the two DFTs for $\{g_i\}$ and $\{h_i\}$, respectively, using the definition of DFT of 4 values. Simplify if possible.

(c) Let F be the DFT of the array f. Using G and H from part b), write out F.

4. (10 marks) Let f_n, $n = 0, \ldots, N-1$ be given real data values and F_k, $k = 0, \ldots, N-1$ be the DFT of f_n. Show that

$$\sum_{k=0}^{N-1} F_k \, \overline{F_k} = \frac{1}{N} \sum_{n=0}^{N-1} f_n^2 \, .$$

Programming Questions

The objective of the programming component of this assignment is to use the FFT to carry out various signal processing operations. The FFT is a basic building block for DSP functions.

Note: There are many plots in this assignment. Do not hand in one plot per page. Use the Matlab *subplot* command to organize multiple plots per page.

5 (15 marks) We look at the problem of identifying the location of a known signal buried in noise. Consider the following scenario, which comes from CSI Waterloo

- The CSI team finds a dead body, with the murder weapon (a gun) nearby.
- A tape recorder was on during the murder. However, the microphone was pointed outside, and there is a lot of traffic noise. The gun had a silencer, so it is very difficult to hear the gunshot on the tape.
- By making a test firing of the gun, we can obtain a clean (noise free) sample of the sound of the gun.

The objective here is to determine the time of the shooting by finding the sound of the gunshot in the noisy recording.

In this assignment, we will take a reference signal, and then try to find out the most likely time (i.e. the location) where this signal occurs in a noisy recording.

On the course web site, you will find:

Data Files signal1_w09, signal2_w09, signal3_w09, signal4_w09, signal5_w09

Five files containing signal samples, one sample per record, 1024 samples. These files can be read in using the *load* command.

Given two real input signals $x_i, y_i, i = 0, ..., N-1$, then the correlation function:

$$\phi_k = \frac{1}{N} \sum_{l=0}^{N-1} x_l y_{l+k} \tag{1}$$

is a measure of how well two signals are correlated, i.e. how much they are similar. Although the correlation function can be computed directly, it is usually more efficient to use an FFT (see the course notes).

Note that to avoid *wrap around pollution*, the computation of the correlation function using an FFT usually requires that the original signals be padded with zeros for twice the original length. This has already been done in *signal1_w09, signal2_w09, signal3_w09, signal4_w09, signal5_w09*, for a total length of 1024.

(a) Using the FFT approach, compute the correlation function of the two signals (signal1_w09, signal2_w09). Of course we want ϕ not its Fourier transform. Plot the original signals, and the correlation function. Give an interpretation of the correlation function with reference to these two plots.

(b) The correlation function can be used to detect a specific signal form in noisy data. If we regard the signal in *signa11_w09* as the reference signal, then we want to detect where this waveform occurs in the noisy data sets *signal3_w09, signal4_w09, signal5_w09.*

That is, the files *signal3_w09, signal4_w09, signal5_w09* contain a signal of the same shape as in *signal1_w09*, except that it is shifted by an unknown amount (a different amount in each file). These files have had white noise added to the signal (progressively larger amounts). Use the correlation function technique to estimate how much the noisy signal is shifted from the reference signal *signal1_w09* for each of *signal3_w09, signal4_w09, signal5_w09.*

Submit plots of the original noisy signal and the correlation function in each case. Submit hard copy of your Matlab code.

6. (10 marks) On the course website, you will find the data file **inputsound.mat**. This sound file can be loaded into your workspace using the following command

```
load inputsound
```

Now, you have the signal array y and the sampling rate Fs in your workspace. You can play this sound by using

```
soundsc(y,Fs)
```

Note that *soundsc* automatically scales the data so that it lies in the range $[-1, +1]$. Otherwise, the data is clipped so that it lies in this range before being played. (Note that *sound* clips any values outside this range).

You will hear a bird chirping and a train whistle. Plot the original signal, and the magnitude of the DFT of this signal (use the *stem* command). Submit these plots.

Read the appendix to this assignment on *filtering.* Design a low pass filter to isolate the train sound, and a high pass filter to isolate the bird chirp. Describe your filter, and submit plots of the final filtered signals, as well as your matlab code.

Filtering

Recorded sounds are often processed by carrying out a *filtering* operation in the frequency domain. Suppose we are given an input signal $x_i, i = 0, ..., N-1$. Let $X = FFT(x)$. The Fourier representation has frequencies in the range $\{0, ..., +N/2\}$, However, note that $X_{N-k} = X_{-k}, k < N/2$ really represents frequencies of size k not $N-k$, since we have used a complex representation of the Fourier series, and we have defined $X_{k \pm N} = X_k$.

If we use the conventional range of $X_k, k \in [0, N-1]$, this means that we have do the following to construct a *low-pass* filter. Let $p < N/2$ be the maximum value of the frequency which will be allowed to pass our *lowpass* filter. In other words, we will kill any frequencies in the signal having frequency $> p$. This is easily accomplished using the lowpass filter

$$\begin{aligned} Q_k &= 0 \;;\; k = p+1, ..., N-p-1 \\ &= 1 \;;\; \text{otherwise} \end{aligned} \tag{2}$$

Note that due to the symmetry of the DFT of a real signal, the filter should be symmetric about $N/2$.

Another way to think about equation (2) is to imagine plotting X in the range $[-N/2 + 1, .., +N/2]$. Then, we want to zero all the X_k such that $k > p$ or $k < -p$. This defines a filter

$Q_k, k \in [-N/2+1, ..., N/2]$. Now, define the filter in the range $[0, ..., N-1]$ by a periodic extension $Q_{N-k} = Q_{-k}, k = 1, .., N/2 - 1$.

The filtered signal in the frequency domain $\hat{X}$ is then

$$\hat{X}_k = X_k Q_k \;\; ; \;\; k = 0, ..., N-1 \tag{3}$$

and the filtered signal in the time domain is $\hat{x} = Real(IFFT(\hat{X}))$.

A high pass filter is constructed in a similar way, except that we want to kill any frequencies $< p$.

Hearing sound with Matlab

The Unix x-terminals cannot play sound files. The Macs should be able to play sound files if Matlab is installed correctly. If you have trouble hearing the sound, you might try this. Write your sound files using Matlab using "au" format or "wav" format. Here are examples

```
load handel; % creates variables y and Fs

    % writes the handel sound track to the file myfile.au
auwrite(y, Fs, 'myfile.au');

    % writes the handel sound track to the file myfile.wav

wavwrite(y, Fs, 'myfile.wav');
```

You should then be able to play back the sound files on any Mac (Quicktime and RealPlayer) or PC (RealPlayer and Windows Media Player). Note: *wavwrite, auwrite* clip the data so that values lie between $[-1, +1]$. You may have to scale the data so that all the values are in the range $[-1, +1]$ before using *wavwrite, auwrite*.

A note on Matlab conventions

The definition of the FFT in Matlab is different from that used in class. In class, the FFT, IFT pair was defined as

$$F_k = \frac{1}{N} \sum_{n=0}^{N-1} f_n \; W^{-kn}$$
$$f_n = \sum_{k=0}^{N-1} F_k \; W^{kn}$$

where

$$W = \exp(\sqrt{-1} \; 2\pi/N)$$

whereas Matlab defines the above pair as

$$F_k = \sum_{n=1}^{N} f_n \; W^{-(k-1)(n-1)}$$
$$f_n = \frac{1}{N} \sum_{n=1}^{N} F_n \; W^{(k-1)(n-1)}$$

The two definitions differ in the place where the $1/N$ is multiplied, as well as the fact that Matlab arrays start at one. In all analytic work, use the definition as in class unless otherwise specified.