2.161 Signal Processing: Continuous and Discrete
Fall 2008

Solution of Problem Set 5: The Discrete Fourier Transform

Assigned: October 16, 2008   **Due:** October 23, 2008

Problem 1:

Given a signal of duration $T = 0.128$ ms, sampled at a rate of $F_s = 8$ kHz, the number of samples is $L = Tf_s = 1024$. If only 256 samples are taken, the frequency spacing in the computed DFT is $\Delta f = f_s/N = 31.25$ Hz.

(a) The number of multiplications required for the direct computation of the DFT is $N^2 = 256^2 = 65,536$.

(b) The number of multiplications required for the computation of the DFT using the FFT is $N/2 \log_2(N) = 128 \times 8 = 1024$.

Note that $F_m = F^*(j\frac{2\pi m}{N\Delta T}) = F^*(j2\pi f_s \frac{m}{N})$. Hence, the m$^{\text{th}}$ component of the DFT corresponds to the actual frequency of $\frac{m}{N} f_s = m\Delta f = 31.25m$ Hz.

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(j\Omega)e^{j\Omega t} d\Omega$$

$$f_n = f(n\Delta t) = \frac{1}{N} \sum_{m=0}^{N-1} F_m e^{j\frac{mn}{N}2\pi}$$

Comparing above formulas, we can realize that, *if there is a ONE-TO-ONE correspondence* between discrete values of F_m and continuous values of $F(j\Omega)$, then at $\Omega = 2\pi f_s \frac{m}{N}$ the $F(j\Omega)$ is a delta function with an amplitude equal to $\frac{2\pi}{N} F_m$. This can be verified for example for either a DC signal or a sine signal without *spectral leakage*.

Note that to find the scale factor, we cannot use the relation between F and F^* ($F^*(j\Omega) = \frac{1}{\Delta T} \sum_{n=-\infty}^{n=+\infty} F(j(\Omega - n2\pi f_s)) = \frac{1}{\Delta T} \sum_{n=-\infty}^{n=+\infty} F(j(\Omega - n\Omega_s))$). That's because in relating F_m to F^*, we assumed that the rest of the sample points are zero. On the other hand, DFT assumes that the $f(t)$ is a periodic extension of the sampled data with period $N\Delta T$.

Problem 2:

A $f_c = 10$ kHz sinusoidal signal is sampled at $f_s = 80$ kHz, and a total of $N = 64$ samples are taken to compute the DFT of the signal. The period of the signal is $T = 1/10^4 = 0.1$ ms, and the sampling period is $\Delta T = 1/f_s = 1/(80 \cdot 10^3) = 12.5$ μsec. Then 64 samples will contain 8 whole periods of the signal ($64 * 12.5$ μsec $*10$kHz$= 8$ cycle). Therefore, there is no spectral leakage. The frequency spacing of the DFT would be $\Delta f = 1/(N\Delta T) = 1/(64 \times 12.5 \cdot 10^{-6}) = 1250$ Hz, so we would expect to see one peak in the DFT $\{F_m\}$ at $m_1 = f_s/f_c = 8$ and the other peak at $m_2 = N - m_1 = 56$. The following MATLAB verifies this.

```
n=[0:63];
DT=1/(80*1e3);
f=sin(2*pi*DT*10000*n);
stem(n,abs(fft(f)))
```
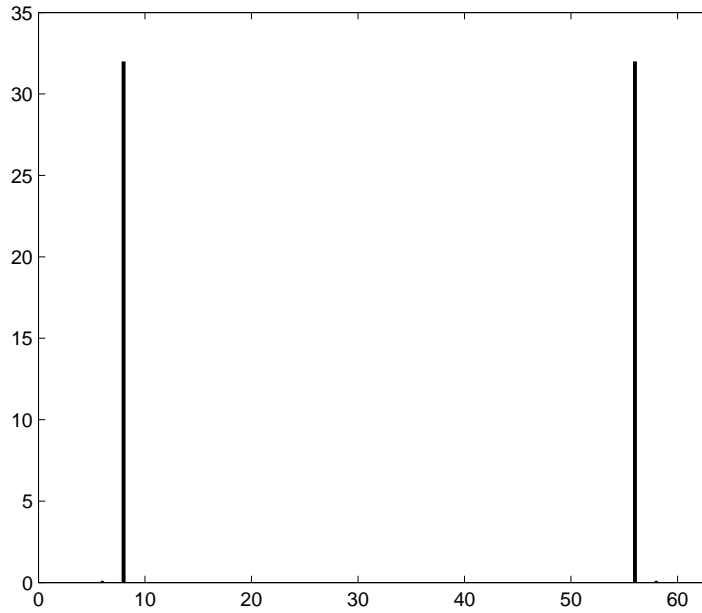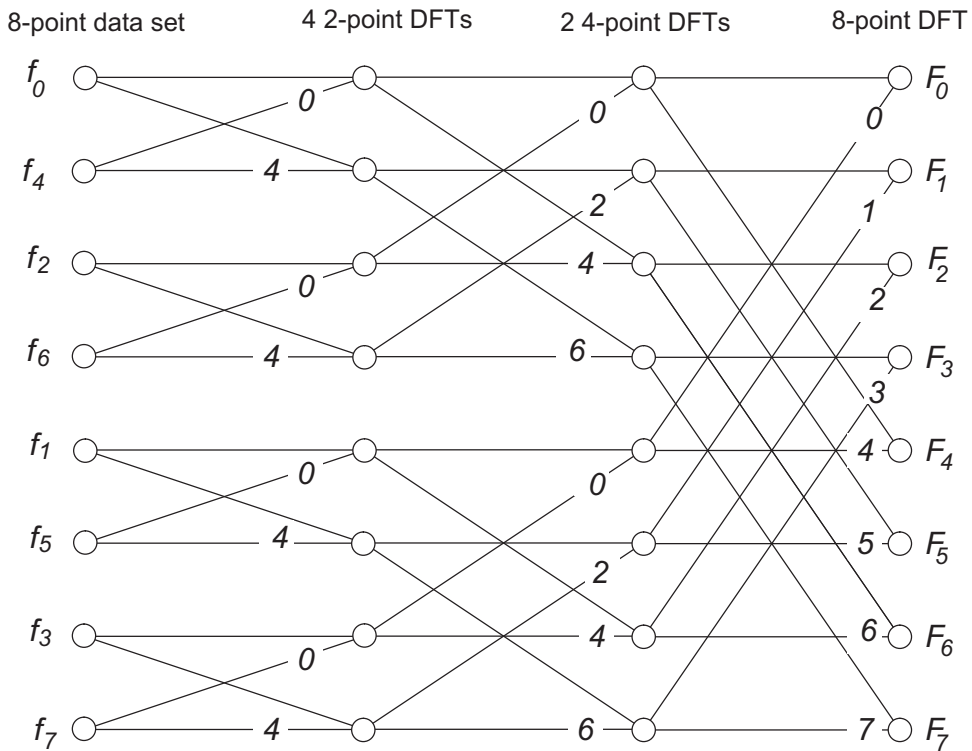
Problem 3:

We note that

$$W_8^0 = 1, \quad W_8^1 = \sqrt{(2)}/2 - j\sqrt{(2)}/2, \quad W_8^2 = 0 - j1, \quad W_8^3 = -\sqrt{(2)}/2 - j\sqrt{(2)}/2,$$
$$W_8^4 = -1, \quad W_8^5 = -\sqrt{(2)}/2 + j\sqrt{(2)}/2, \quad W_8^6 = 0 + j1, \quad W_8^7 = \sqrt{(2)}/2 + j\sqrt{(2)}/2.$$

and from Fig. 5 in the FFT handout:

we can construct the following table:

Data	2-point DFTs	4-point DFTs	8-point DFT
$x_0 = 5$	$x_0 + x_4 = 10$	$10 + W_8^0 \cdot (-6) = 4$	$4 + W_8^0 \cdot (-4) = 0 = X_0$
$x_4 = 5$	$\underline{x_0 - x_4 = 0}$	$0 + W_8^2 \cdot 0 = 0$	$0 + W_8^1 \cdot (0) = 0 = X_1$
$x_2 = -3$	$x_2 + x_6 = -6$	$10 + W_8^4 \cdot (-6) = 16$	$16 + W_8^2 \cdot (0) = 16 = X_2$
$x_6 = -3$	$\underline{x_2 - x_6 = 0}$	$\underline{0 + W_8^6 \cdot (0) = 0}$	$0 + W_8^3 \cdot (0) = 0 = X_3$
$x_1 = -1$	$x_1 + x_5 = -2$	$-2 + \overline{W_8^0 \cdot (-2)} = -4$	$4 + W_8^4 \cdot (-4) = 8 = X_4$
$x_5 = -1$	$\underline{x_1 - x_5 = 0}$	$0 + W_8^2 \cdot 0 = 0$	$0 + W_8^5 \cdot (0) = 0 = X_5$
$x_3 = -1$	$x_3 + x_7 = -2$	$-2 + W_8^4 \cdot (-2) = 0$	$16 + W_8^6 \cdot (0) = 16 = X_6$
$x_7 = -1$	$x_3 - x_7 = 0$	$0 + W_8^6 \cdot (0) = 0$	$0 + W_8^7 \cdot (0) = 0 = X_7$

The same answers are obtained using the Matlab command: `fft([5 -1 -3 -1 5 -1 -3 -1])`.

Assuming the samples are of $x(n) = 4\cos(\pi n/2) + \cos(\pi n)$ we would expect to see 4 peaks (2 for each cosine term) with amplitude ratios of 4:1. However, we see only 3 peaks and they have a ration of 2:1. Indices 2 and 6 correspond to the low frequency cosine, while index 4 correspond to the high frequency cosine. Since the sampling frequency is 1 and the frequencies of the cosines are $1/4$ and $1/2$, aliasing is present. Notice that the values for the $4\cos(\pi n/2)$ components are correct, but the value for the aliased $\cos(\pi n)$ is incorrect. No leakage is present because both components have an integral number of periods in the data record.

Problem 4:   (Proakis & Manolakis – Prob. 7.1)
Since $x(n)$ is real, the real part of the DFT is even, imaginary part odd ($x(t)$ real $\Rightarrow$ $X(-j\Omega) = \bar{X}(j\Omega)$). Thus, the remaining points are $\{0.125 + j0.0518, 0, 0.125 + j0.3018\}$.

Problem 5:   (Proakis & Manolakis – Prob. 7.3)
$\hat{X}(k)$ may be viewed as the product of $X(k)$ with:

$$\hat{F}(k) = \begin{cases} 1, & 0 \le k \le k_c, \quad N - k_c \le k \le N - 1. \\ 0, & k_c < k < N - k. \end{cases}$$

$F(k)$ represents an ideal low-pass filter removing frequency components from $(k_c + 1)\frac{2}{N}$ to π (for a zero-centered set of DFT values). Hence $\hat{x}(n)$ is a low-pass version of $x(n)$.

Problem 6:

Given the noisy signal created using the following Matlab commands:

```
>> t=[0:.01:10.23]; % 1024 Points
>> f=exp(-t).*sin(10*t); % Clean signal
>> noise=rand(1,1024)-0.5;
>> signal=f+noise; % Additive random noise
```

Please note that we have changed the noise, such that its average value is equal to zero. Otherwise, we can subtract the DC value manually from DFT component. The DC magnitude is equal to Nn_0 where N is the number of points and n_0 is the average value of Noise.
Plots of the clean and noisy signals are:
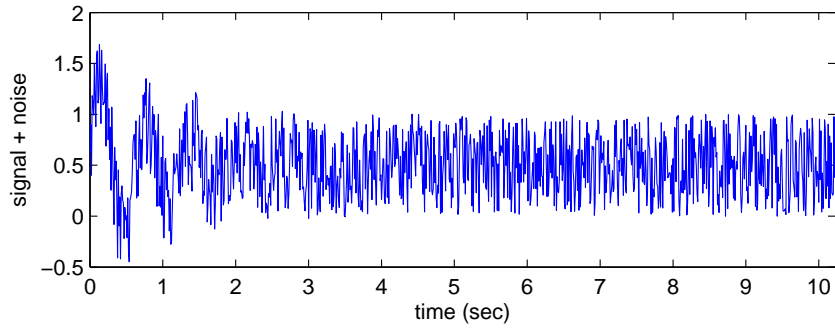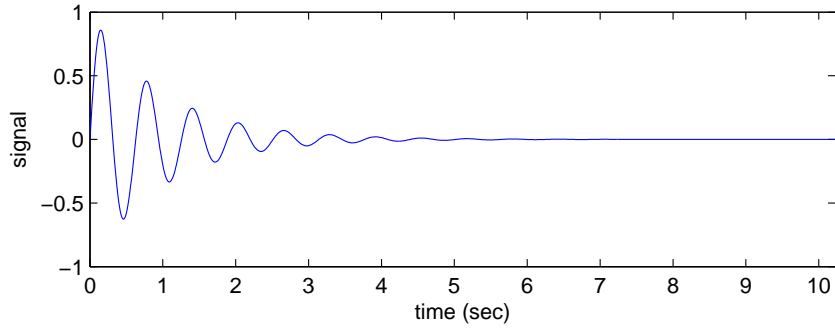
The DFT magnitude plots are
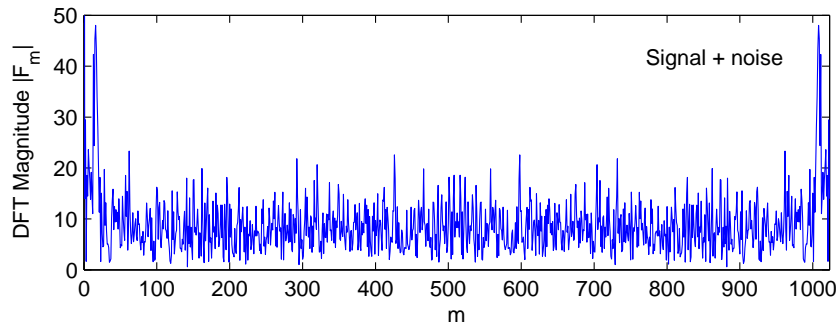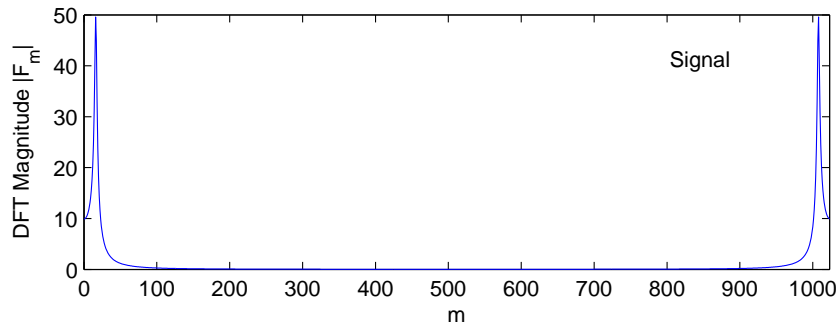
The following is the MATLAM script that was used to do crude FFT-based low-pass filtering on the data. The FFT of the data set is computed and K terms F_m, $|m| = 0, \ldots, K$ are retained before takin the IFFT and plotting the result.

```
t=[0:.01:10.23];
signal = exp(-t).*sin(10*t);
noise = rand(1,1024)-0.5;
s_plus_n = signal+noise;
m = [0:1023];
Fm = fft(s_plus_n);
% Do low-pass filtering by zeroing out the center of
% the DFT.  K is the band-limit
K = 60;
% Create an empty array and move the K low frequency
% components in.
Fout=zeros(size(Fm));
Fout(1:K) = Fm(1:K);
Fout(1024-K-1:1024) = Fm(1024-K-1:1024);
subplot(2,1,1), plot(m,abs(Fout))
subplot(2,1,2), plot(t,signal,'r--',t,real(ifft(Fout)),'b')
```
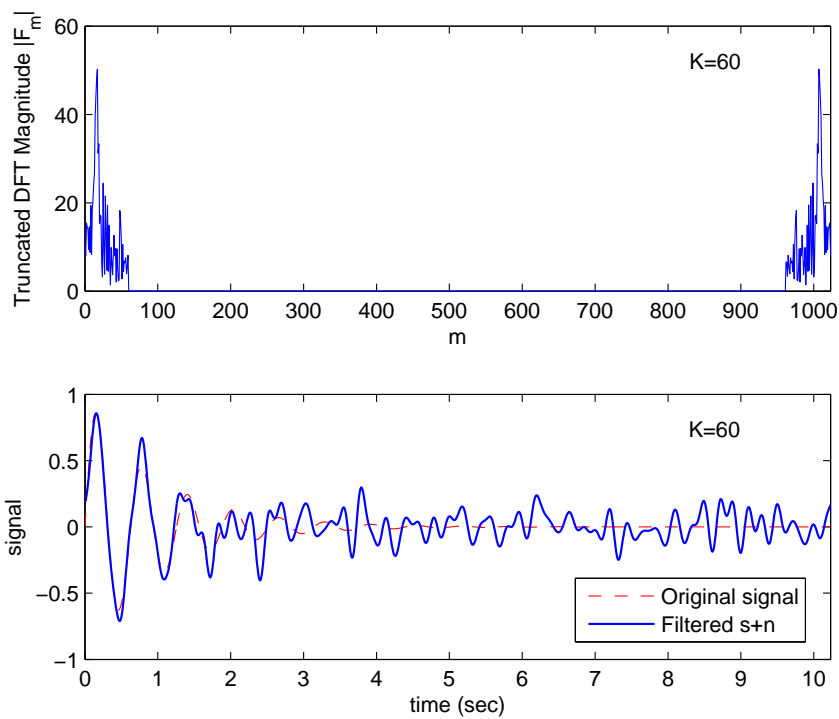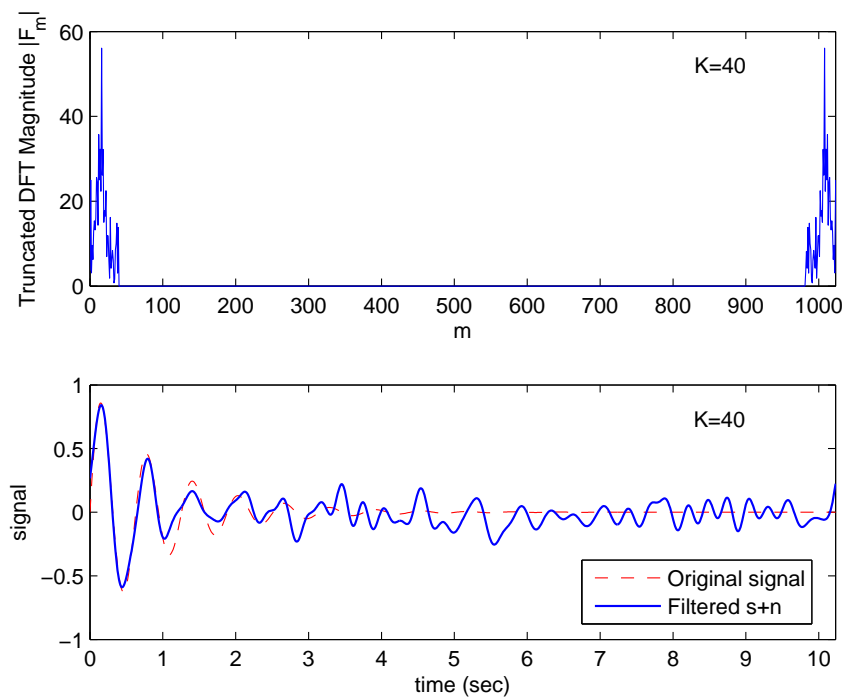
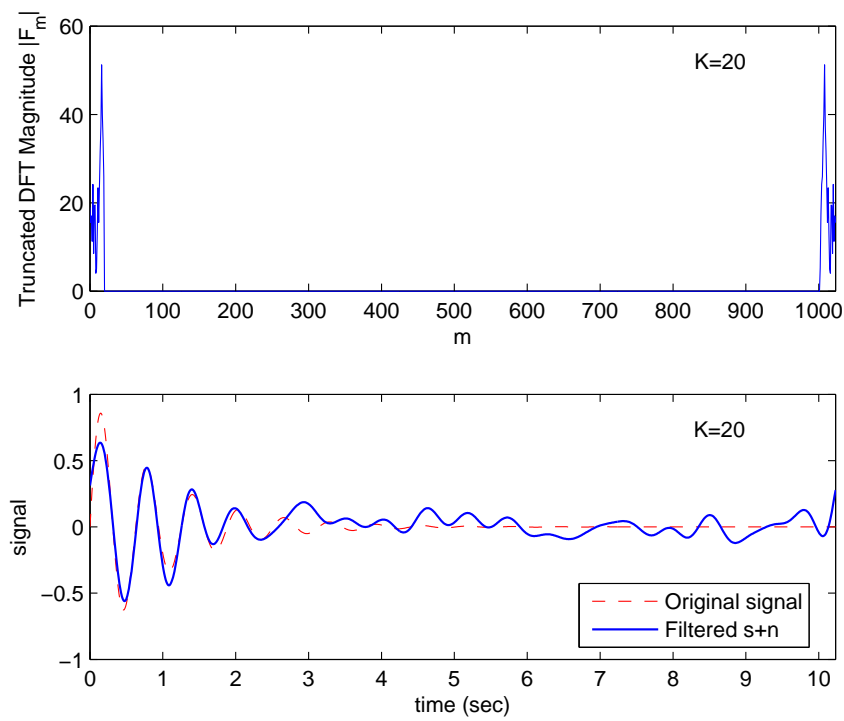Results for some typical values of K are:

(a) $K = 100$:
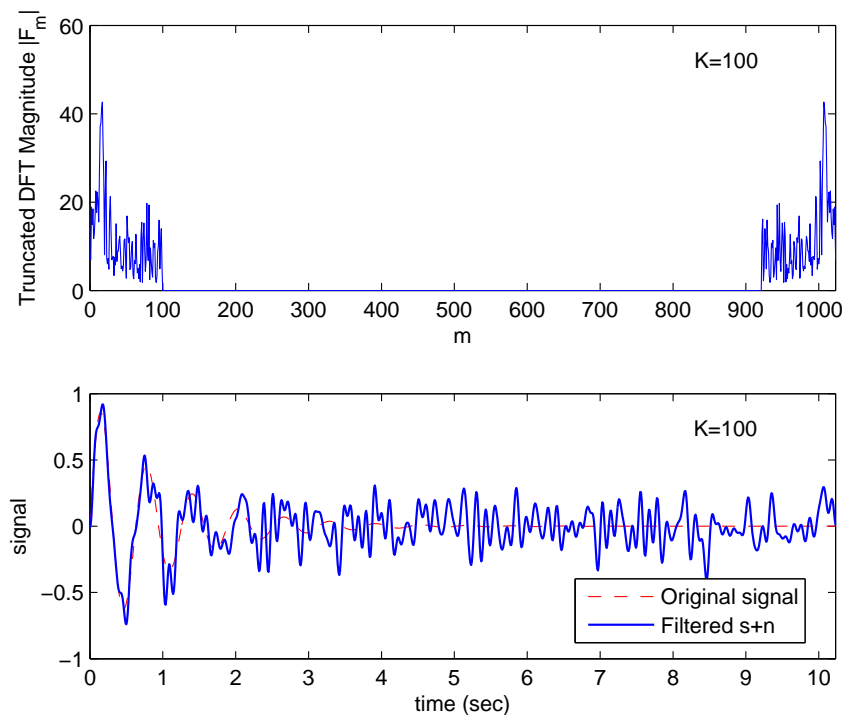
(b) $K = 60$:

(c) $K = 40$:

(d) $K = 20$:

We can argue about it, but I think that $K = 40$ looks to be about a good compromise between residual noise and fidelity of the original signal???