MASSACHUSETTS INSTITUTE OF TECHNOLOGY
DEPARTMENT OF MECHANICAL ENGINEERING

2.161 *Signal Processing - Continuous and Discrete*
Fall Term 2008

Lecture 11[1]

Reading:

- Class Handout: *Sampling and the Discrete Fourier Transform*
- Class Handout: *The Fast Fourier Transform*
- Proakis and Manolakis (4th Ed.) Ch. 7
- Oppenheim, Schafer & Buck (2nd Ed.) Chs. 8 & 9

1 The Discrete Fourier Transform – continued

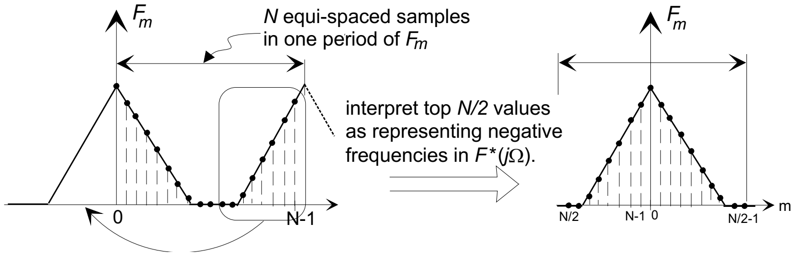
In Lecture 10 the DFT pair associated with a sample set $\{f_n\}$ of length N was defined as

$$\begin{aligned} F_m &= \sum_{n=0}^{N-1} f_n \, \mathrm{e}^{-\mathrm{j}\,2\pi mn/N} \\ f_n &= \frac{1}{N}\sum_{m=0}^{N-1} F_m \, \mathrm{e}^{\mathrm{j}\,2\pi mn/N} \end{aligned}$$

The value F_m was interpreted as $F^*(\mathrm{j}\,\Omega)$ evaluated at $\Omega = 2\pi/N\Delta T$.

1.1 Organization of the DFT

The N components in a DFT represent one period of a periodic spectrum. The first $N/2$ lines in the spectrum represent physical frequencies $0 \ldots (\pi/\Delta T)$ radians/second. The components in the upper half of the sequence, $F_{N/2+1} \ldots F_{N-1}$, may be considered to be the negative frequency components $F_{-N/2+1} \ldots F_{-1}$ in the spectrum. It is common to translate the upper half of the record to the the left side of a plot to enhance the physical meaning.


F_m
N equi-spaced samples in one period of F_m
0
N-1
interpret top *N/2* values as representing negative frequencies in $F^*(j\Omega)$.
F_m
N/2
N-1 0
N/2-1
m

[1]copyright © D.Rowell 2008

1.2 Spectral Resolution of the DFT

The DFT pair provide a transform relationship between a pair of (complex) data sets $\{f_n\}$ and $\{F_m\}$, each of length N. If the sampling interval associated with $\{f_n\}$ is ΔT units, the record duration is

$$T = N\Delta T.$$

The frequency resolution, or line spacing, $\Delta\Omega$, in the DFT is

$$\Delta\Omega = \frac{2\pi}{N\Delta T} = \frac{2\pi}{T} \text{ rad/s}, \qquad \text{or} \qquad \Delta F = \frac{1}{T} \text{ Hz.} \tag{1}$$

and the frequency range spanned by the N lines in the DFT is

$$N\Delta\Omega = \frac{2\pi}{\Delta T} \text{ rad/s}, \qquad \text{or} \qquad N\Delta F = \frac{1}{\Delta T} \text{ Hz.} \tag{2}$$

The sequence $\{F_m\}$ represents both the positive and negative frequencies in a two-sided spectrum. The highest (positive) frequency component in the spectrum is half of this range (the Nyquist frequency), that is

$$\Omega_{\max} = \frac{\pi}{\Delta T} \text{ rad/s}, \qquad \text{or} \qquad F_{\max} = \frac{1}{2\Delta T} \text{ Hz.} \tag{3}$$

We conclude therefore, that the resolution within the DFT depends on the duration T of the data record, and the maximum frequency depends on the sampling interval ΔT.

1.3 Properties of the Discrete Fourier Transform

Because the DFT is derived directly as a sampled continuous Fourier transform, it inherits most of the properties of the Fourier transform. We repeat some of the important properties here. In addition other properties are based on the assumed periodicity of $\{f_n\}$ and $\{F_m\}$:

1. **Linearity:** If $\{f_n\}$ and $\{g_n\}$ are both length N, and

$$\{f_n\} \overset{\text{DFT}}{\Longleftrightarrow} \{F_m\} \qquad \text{and} \qquad \{g_n\} \overset{\text{DFT}}{\Longleftrightarrow} \{G_m\}$$

then

$$a\{f_n\} + b\{g_n\} \overset{\text{DFT}}{\Longleftrightarrow} a\{F_m\} + b\{G_m\}$$

2. **Symmetry Properties of the DFT:** If $\{f_n\}$ is a real-valued sequence then

$$\{F_m\} = \{\overline{F}_{-m}\}$$

from which it follows that $\Re\{\{F_m\}\}$ is an even function of m and $\Im\{\{F_m\}\}$ is an odd function of m. Similarly the magnitude of $\{F_m\}$ is an even function, and the phase is an odd function. In addition

$$\mathcal{E}\{\{f_n\}\} \overset{\text{DFT}}{\Longleftrightarrow} \Re\{\{F_m\}\} \qquad \text{and} \qquad \mathcal{O}\{\{f_n\}\} \overset{\text{DFT}}{\Longleftrightarrow} \Im\{\{F_m\}\}$$

where $\mathcal{E}$ and $\mathcal{O}$ denote the even and odd parts respectively.

3. Shifting Properties: If

$$\{f_n\} \overset{\text{DFT}}{\Longleftrightarrow} \{F_m\}$$

then

$$\{f_{n-n_0}\} \overset{\text{DFT}}{\Longleftrightarrow} \left\{ \mathrm{e}^{-jmn_0} F_m \right\}$$

and

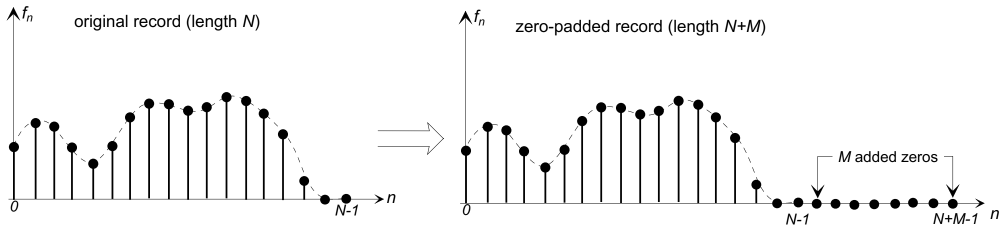
$$\left\{ \mathrm{e}^{jm_0 n} f_n \right\} \overset{\text{DFT}}{\Longleftrightarrow} \{F_{m-m_0}\}$$

where n_0 and m_0 are constants.

4. Periodicity As noted above, both $\{f_n\}$ and $\{F_m\}$ are periodic with period N.

■ Example 1

What is the effect of zero-padding (adding extra zero amplitude samples) to a data record $\{f_n\}$ before taking the DFT?


f_n
original record (length N)
0
N-1
n
f_n
zero-padded record (length $N+M$)
M added zeros
0
N-1
$N+M$-1
n

- Assume that the original N samples encapsulate $f(t)$, so that $f(t) \equiv 0$ for $T > N\Delta T$. Then let F_m^N denote the DFT of the original length N record

$$F_m^N = \sum_{n=0}^{N-1} f_n \, \mathrm{e}^{-\mathrm{j}\, 2\pi mn/N}$$

  We note that when related back to the continuous time domain, the Nyquist frequency is $\Omega_N = \pi/\Delta T$, and the line spacing is $\Delta\omega = 2\pi/N\Delta T$.

- When we ad M zeros to the end of the record

$$F_m^{N+M} = \sum_{n=0}^{N+M-1} f_n \, \mathrm{e}^{-\mathrm{j}\, 2\pi mn/(N+M)} = \sum_{n=0}^{N-1} f_n \, \mathrm{e}^{-\mathrm{j}\, 2\pi mn/(N+M)}$$

  since the M extra data point do not contribute to the DFT sums. However we now have $N + M$ spectral samples with the same Nyquist frequency $\Omega_N = \pi/\Delta T$, but with a new line spacing $\Delta\Omega = 2\pi/(N + M)\Delta T$.

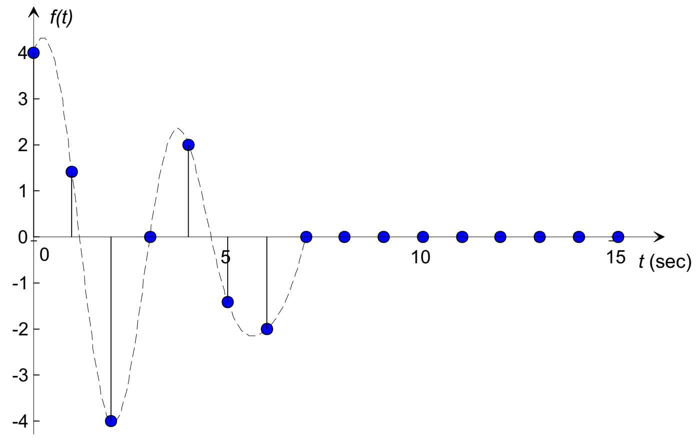
The result is that padding the data record with zeros has increased the spectral resolution of the DFT.

Note that if the number of added zeros is an integer multiple of the original record length ($M = kN$, for $k = 1, 2, 3, \ldots$), the original DFT samples F_m^N will be preserved and $k-1$ new samples will be interpolated between the original samples.

$$F_m^{kN} = \sum_{n=0}^{kN-1} f_n \, \mathrm{e}^{-\mathrm{j}\,2\pi mn/(kN)} = \sum_{n=0}^{N-1} f_n \, \mathrm{e}^{-\mathrm{j}\,2\pi mn/(kN)}$$

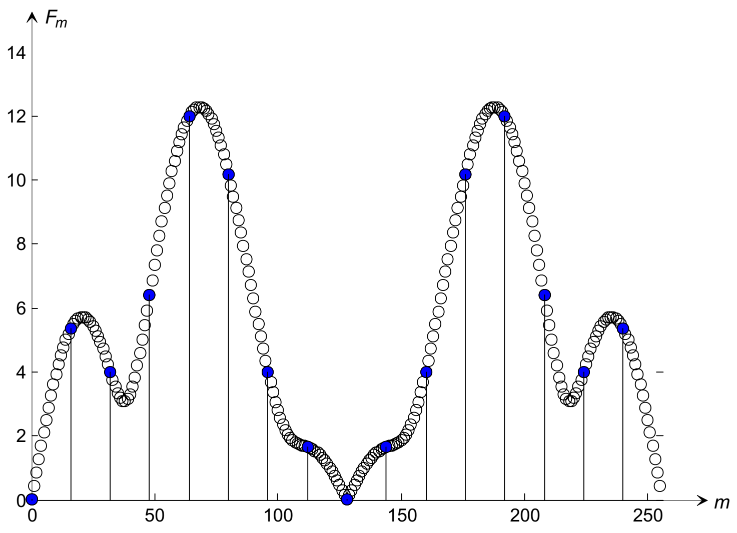
so that $F_m^N = F_{km}^{kN}$.

Example: The waveform below was sampled with $\Delta T = 1$ sec to give a length 16 record.



The data record length was then extended to a length of 256 by zero-padding with 240 zeros, and the FFT was then computed again. The following plot, demonstrating the interpolation in the DFT was generated from the MATLAB fragment:

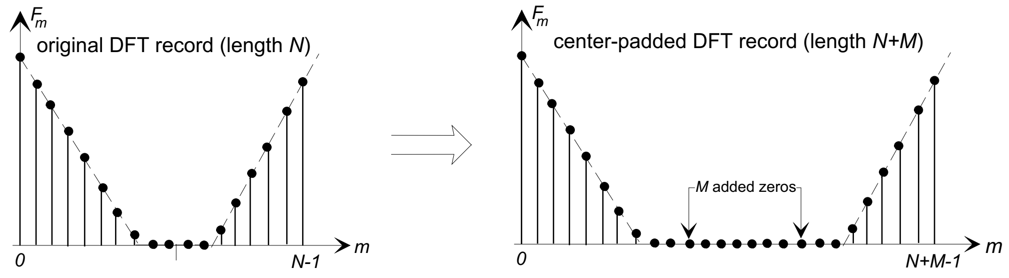
```
k    =  0:7;
f16  = [cos(2.*pi*k/8) + 3*cos(4*pi*k/8)+ sin(6*pi*k/8)  zeros(1,8)];
f256 = [f16  zeros(1,240)];
plot(0:255,abs(fft(f256)),'o');
hold;
stem(0:16:255, abs(fft(f16)),'filled');
```



The original 16 DFT points are shown as filled circles (produced by the `stem()` function). The interpolation of 15 new values between these points can be easily seen.

■ Example 2

What is the effect of adding M zeros to the center of a DFT record $\{F_m\}$ before taking the inverse DFT?



F_m
original DFT record (length N)
0
N-1
m
F_m
center-padded DFT record (length N+M)
M added zeros
0
N+M-1
m

We assume that no aliasing has taken place in the original sampling operation.

Let's answer this by considering what the DFT would have been if we had sampled $f(t)$ at a higher rate, for example let's consider twice the original rate.

- Since no aliasing was present in the original record, there will ne no aliasing at the higher rate.

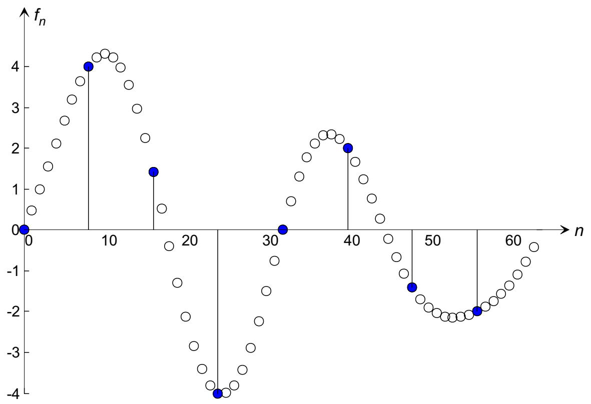
- The total duration $T = N\Delta T = 2N(\Delta T/2)$, therefore the line spacing $\Delta\Omega$ remains constant.
- The Nyquist frequency $\Omega_N = \pi/(\Delta T/2)$ has been doubled.
- The spectrum has been scaled.

This argument says that doubling the sampling rate (halving the interval ΔT) has simply inserted N samples into the center of the DFT record $\{F_m\}$, and scaled the data points. In general, inserting kN zeros into the center of a DFT record and then taking the inverse DFT of the new record of length $(k+1)N$, is to interpolate $k-1$ samples between each of the original N data points (and to scale the data set).

The following MATLAB fragment was used to demonstrate the interpolation.

```
 i=0:7;
 f8=cos(2*pi*(i-1)/8) + 3*cos(4*pi*(i-1)/8)+ sin(6*pi*(i-1)/8) ;
% Take the DFT
 Fin = fft(f8);
% Pad the center of the DFT with zeros.
 Fout = zeros(56,1);
 Fout(1:4) = Fin(1:4);
 Fout(61: 64) = Fin(5:8);
% Take the inverse DFT, scale, and preserve the real part
 fout = real(ifft(Fout))*8;
 plot(0:63,fout,'o');
 hold on;
 stem(0:8:63,f8,'filled');
```

A data record of length 8 is created, and the DFT is computed. The center of this record is padded with zeros to form a length 64 DFT. The inverse DFT is then computed and plotted. The results are shown below. The original length 8 data sequence is shown with filled circles, and the interpolated data points are shown as open circles.



2 The Fast Fourier Transform (FFT)

Although the DFT was known for many decades, its utility was severely limited because of the computational burden. The calculation of the DFT of an input sequence of an N length sequence $\{f_n\}$

$$F_m = \sum_{n=0}^{N-1} f_n \,\mathrm{e}^{-\mathrm{j}\,\frac{2\pi mn}{N}}, \qquad m = 0, \ldots, N-1 \tag{4}$$

requires N complex multiplications to compute each on the N values, F_m, for a total of N^2 multiplications. Early digital computers had neither fixed-point nor floating-point hardware multipliers, and multiplication was performed by binary shift-and-add software algorithms. Multiplication was therefore a computationally "expensive" and time consuming operation, rendering machine computation of the DFT impractical for common usage.

The FFT is a computationally efficient algorithm to compute the DFT, by eliminating redundant *complex multiplications.* In this course we look at a single implementation - the *radix-2 FFT with decimation in time and input-bit-reversal.* Many other algorithms exist.

We start by writing the DFT as

$$F_m = \sum_{n=0}^{N-1} f_n \,\mathrm{e}^{-\mathrm{j}\,\frac{2\pi mn}{N}} = \sum_{n=0}^{N-1} f_n W_N^{mn}, \qquad m = 0, \ldots, N-1$$

where $W_N = \mathrm{e}^{-\mathrm{j}\,2\pi/N}$. We also note the following periodic and symmetry properties of W_N

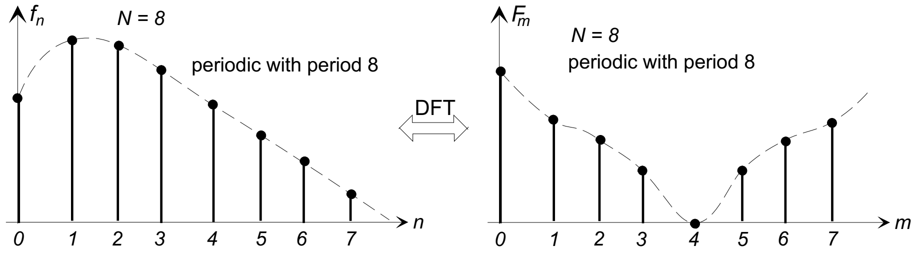
- $W_N^{k(N-n)} = W_N^{-kn} = \overline{W_N^{kn}}$ (complex conjugate symmetry),
- $W_N^{kn} = W_N^{k(n+N)} = W_N^{n(k+N)}$ (periodicity in n and k),
- $W_N^{n} = -W_N^{n-N/2}$ for $n \geq N/2$.

The FFT recognizes that these properties render many of the N^2 complex multiplications in the DFT redundant.

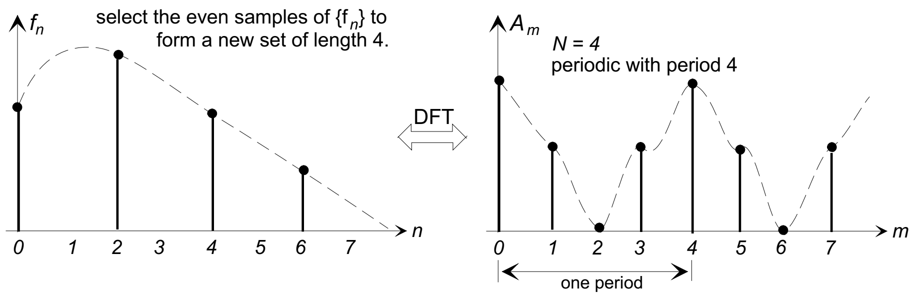
Assume that the input sequence length N is even. We then ask whether any computational efficiency might be gained from splitting the calculation of $\{F_m\}$ into two sub-computations, each of length $N/2$, involving the even samples, $\{f_{2n}\}$, and odd samples $\{f_{2n+1}\}$ for $n = 0 \ldots N/2-1$. Then

$$\begin{aligned}
F_m &= \sum_{n=0}^{P-1} f_{2n} W_N^{2mn} + \sum_{n=0}^{P-1} f_{2n+1} W_N^{m(2n+1)} \\
&= \sum_{n=0}^{P-1} f_{2n} W_N^{2mn} + W_N^{m} \sum_{n=0}^{P-1} f_{2n+1} W_N^{2mn} \\
&= \sum_{n=0}^{P-1} f_{2n} W_P^{mn} + W_N^{m} \sum_{n=0}^{P-1} f_{2n+1} W_P^{mn} \\
&= A_m + W_N^m B_m, \qquad m = 0, \ldots, N-1.
\end{aligned}$$

where $P = N/2$, $\{A_m\}$ is a DFT of length $N/2$, based on the even sample points, and similarly $\{B_m\}$ is a DFT of length $N/2$ based on the odd sample points of $\{f_n\}$. For example, if $N = 8$ the DFT relationship of $\{f_n\}$ shown as



may decomposed into even and odd sample sets with a DFT relationship for the even set indicated as



We also note from the properties of the DFT that both $\{A_m\}$ and $\{B_m\}$ are periodic with period $N/2$, that is

$$A_{m+N/2} = A_m, \qquad \text{and} \qquad B_{m+N/2} = B_m$$

so that

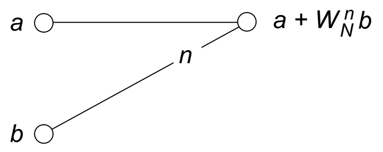
$$\begin{aligned} F_m &= A_m + W_N^m B_m && \text{for } m = 0 \ldots (N/2-1), \text{ and} \\ F_m &= A_{m-N/2} - W_N^{m-N/2} B_{m-N/2} && \text{for } m = N/2 \ldots (N-1) \end{aligned}$$

These equations show that a DFT of length N may be synthesized by combining two shorter DFTs from an even/odd decomposition of the original data set. For example, if $N = 8$, F_3 and F_7 are simply related:

$$\begin{aligned} F_3 &= A_3 + W_8^3 B_3 \\ F_7 &= A_7 + W_8^7 B_7 = A_3 - W_8^3 B_3 \end{aligned}$$

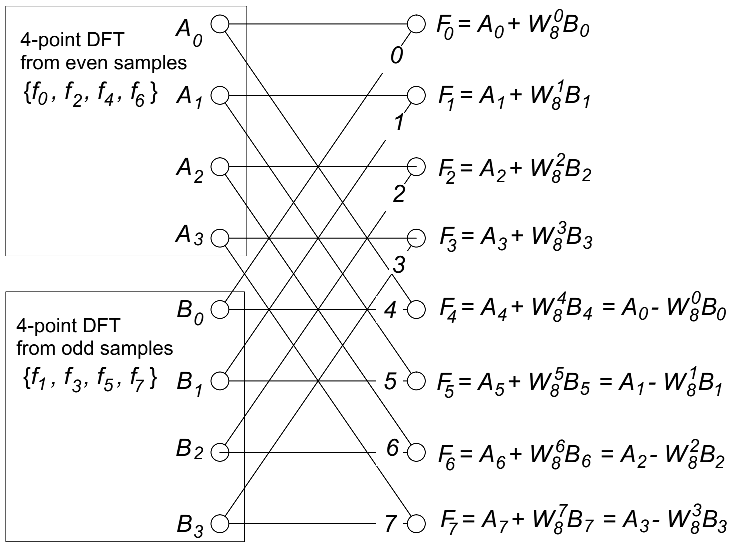
so that $N/2$ multiplications are required to combine the two sets. Each of the two shorter DFTs requires $(N/2)^2$ complex multiplications, therefore the total required is $N^2/2 + N/2 < N^2$, for $N > 2$, indicating a computational saving.

A modified discrete form of Mason's signal-flow graph is commonly used to display the algorithmic structure of the synthesis. The figure below shows the signal-flow graph - consisting of a network of nodes connected by line segments.



The algorithm works from left to right, and each right-hand node is assigned a value that is the weighted sum of the connected left-hand nodes, where the indicated weight n is the exponent of W_N. If no weight is indicated, it is assumed to be unity (or equivalent to W_N^0). Thus the output of the step shown above is $c = a + W_N^n b$.

With this notation the combining of the two length $N/2$ DFTs is illustrated below for N=8. Each right-hand node is one of the F_m, formed by the appropriate combination of A_m and B_m.


4-point DFT from even samples
$\{f_0, f_2, f_4, f_6\}$
4-point DFT from odd samples
$\{f_1, f_3, f_5, f_7\}$
A_0
A_1
A_2
A_3
B_0
B_1
B_2
B_3
0
1
2
3
4
5
6
7
$F_0 = A_0 + W_8^0 B_0$
$F_1 = A_1 + W_8^1 B_1$
$F_2 = A_2 + W_8^2 B_2$
$F_3 = A_3 + W_8^3 B_3$
$F_4 = A_4 + W_8^4 B_4 = A_0 - W_8^0 B_0$
$F_5 = A_5 + W_8^5 B_5 = A_1 - W_8^1 B_1$
$F_6 = A_6 + W_8^6 B_6 = A_2 - W_8^2 B_2$
$F_7 = A_7 + W_8^7 B_7 = A_3 - W_8^3 B_3$

This only the first step in the development of the FFT. The complete algorithm will be developed in Lecture 12.