"It's our own story exactly! He bold as a hawk, she soft as the dawn."

EE 311 **January 28, 2019**
Notes – Sixth Day

Chapter 3

• Fast Fourier transform (FFT)
To show how the FFT can be done we take N to be a power of 2 and break the DFT up into two sequences of even and odd terms.

$$F(k)=\sum_{\substack{n=0\\ n\ even}}^{N-1} f(n)e^{-j\frac{2\pi kn}{N}}+\sum_{\substack{n=0\\ n\ odd}}^{N-1} f(n)e^{-j\frac{2\pi kn}{N}}$$

This equation can be simplified by doing a variable change of n to 2m.

$$F(k)=\sum_{m=0}^{\frac{N}{2}-1} f(2m)e^{-j\frac{2\pi k}{N}2m}+\sum_{m=0}^{\frac{N}{2}-1} f(2m+1)e^{-j\frac{2\pi k}{N}(2m+1)}$$

From this equation we can factor $e^{-j\frac{2\pi k}{N}}$ from the right most term to get

$$F(k)=\sum_{m=0}^{\frac{N}{2}-1} f(2m)e^{-j\frac{2\pi k}{N}2m}+e^{-j\frac{2\pi k}{N}}\sum_{m=0}^{\frac{N}{2}-1} f(2m+1)e^{-j\frac{2\pi k}{N}2m} \qquad \textbf{2.18}$$

To simplify the notation let $G(k)=\sum_{m=0}^{\frac{N}{2}-1} f(2m)e^{-j\frac{2\pi k}{N}2m}$ and $H(k)=\sum_{m=0}^{\frac{N}{2}-1} f(2m+1)e^{-j\frac{2\pi k}{N}2m}$ so that equation 2.18 can be written as

$F(k)=G(k)+e^{-j\frac{2\pi k}{N}}H(k)$ where G(k) and H(k) are both N/2 term DFTs.

An N-term DFT is periodic with a period of N. Equation 2.18 is composed of two N/2-term DFT's and the period of each of these must be N/2. This implies that G(k) = G(k+N/2) and H(k) = H(k+N/2). To take advantage of this we write

$$F(k+N/2)=\sum_{m=0}^{\frac{N}{2}-1} f(2m)e^{-j\frac{2\pi(k+N/2)}{N}2m}+e^{-j\frac{2\pi(k+N/2)}{N}}\sum_{m=0}^{\frac{N}{2}-1} f(2m+1)e^{-j\frac{2\pi(k+N/2)}{N}2m}$$

or, $F(k+N/2)=\sum_{m=0}^{\frac{N}{2}-1} f(2m)e^{-j\frac{2\pi k}{N}2m}\cdot e^{-j2m\pi}+e^{-j\frac{2\pi k}{N}}\cdot e^{-j\pi}\sum_{m=0}^{\frac{N}{2}-1} f(2m+1)e^{-j\frac{2\pi k}{N}2m}\cdot e^{-j2m\pi}$

But $e^{-j2m\pi}=\cos(2m\pi)-j\sin(2m\pi)=1$ since m is an integer and $e^{-j\pi}=-1$. This leads to

$$F(k+N/2)=\sum_{m=0}^{\frac{N}{2}-1} f(2m)e^{-j\frac{2\pi k}{N}2m}-e^{-j\frac{2\pi k}{N}}\sum_{m=0}^{\frac{N}{2}-1} f(2m+1)e^{-j\frac{2\pi k}{N}2m} \qquad \textbf{2.19}$$

or $F(k+N/2)=G(k)-e^{-j\frac{2\pi k}{N}}H(k)$

Thus we see that if we calculate G(k) and H(k) to find F(k) in equation 3.16, we can make a single sign change and find F(k+N/2) using equation 2.19. To find all of the values of F(k) then, it is necessary to calculate all of the values of G(k) and H(k). This amounts to some considerable savings in computation time since G(k) and H(k) are N/2-term DFTs. For example, if N = 16, a straight forward calculation of F(k) using the equation for the DFT would have $N^2 = 256$ complex operations. Calculation of F(k) from G(k) and H(k) would have only $2(N/2)^2 + N/2 = 136$ complex operations. (The N/2 term which is added in this equation is the result of the multiplication of $e^{-j\frac{2\pi k}{N}}$ time H(k)).

We can get further computational savings if we can repeat this process on G(k) and H(k). Thus, if N is a power of 2, N/2 is also a power of 2 and G(k) and H(k) can be divided into even and odd terms just as F(k) was. This division can continue until we are down to transforms of 1-term functions. Thus a 16 term function would be divided into two 8 point functions. These would be divided into four 4 point functions which would lead to eight 2 point functions. Each division would produce fewer complex operations. For an N-point sequence where $N = 2^p$, we can repeat this reduction process p times. Since $p = \log_2(N)$, the total number of complex multiplications is reduced to $(N/2)\log_2(N)$. ($N\log_2(N)$ additions are also required.) For example, if N = 1024, $\log_2(1024) = 10$ and 5,120 complex multiplications are required as opposed to 1,048,576 by brute force.

To visualize this consider the signal flow graph of Figure 3.9. In this figure, the central darkened circle is a summing junction and the signal flow graph shows how F(k) and F(k+N/2) are calculated from G(k) and H(k). Note that the ± sign on the e term must be chosen positive when calculating F(k) and negative when calculating F(k+N/2).

If an 8 point DFT is broken down into two 4 point DFTs, the butterfly signal flow graphs would be calculated as shown in Figure 3.10. In this figure the even terms of f(n) form the first 4 terms of G(k) and the odd terms of f(n) form the first four terms of H(k). If G(k) and H(k) are further broken down into even and odd terms the signal flow graph for F(k) is shown in Figure 2.11.

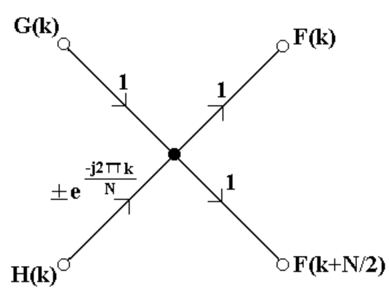

Figure 2.9

Signal flow graph for the calculation of F(k) and F(k+N/2) from G(k) and H(k) according to equations 1 and 2. This signal flow graph is commonly referred to a as a "Butterfly".

Note that in Figure 2.11 the ordering of the original signal, f(n) is skewed. In the first reduction the function was divided into even and odd parts so that the ordering is 0, 2, 4, 6, 1, 3, 5, 7. These 4-point DFTs are further divided into even and odd parts the new ordering becomes 0, 4, 2, 6, 1, 5, 3, 7.

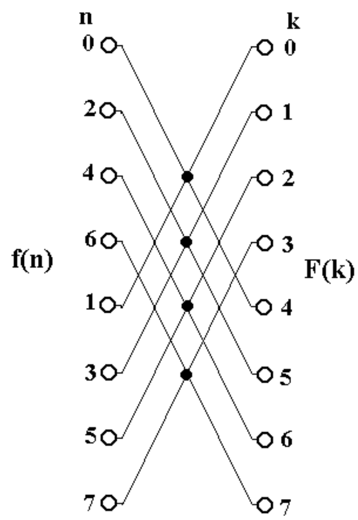

Figure 2.10
The butterfly signal flow graphs for the calculation of an 8-point FFT from two 4-point DFTs.

As it turns out this ordering of the input sequence can be easily determined by writing the sequence in binary notation and reversing the bit orders. Thus we see that

0 = 000 →	000 = 0
1 = 001 →	100 = 4
2 = 010 →	010 = 2
3 = 011 →	110 = 6
4 = 100 →	001 = 1
5 = 101 →	101 = 5
6 = 110 →	011 = 3
7 = 111 →	111 = 7

This process of reordering the input sequence is called “bit twiddling”.

```
%WavFileFFT.m
[x fs] = audioread('rossini.wav');
T = 1/fs;
k = 1:length(x);
figure(1);clf;
subplot(2,1,1)
plot(k*T,x)                %Plot x in time
axis([0 T*length(x) -1.5 1.5])
xlabel('time in seconds');
ylabel('voltage');
title('rossini.wav in Time');
%
X = fft(x);
X = X/max(abs(X));
subplot(2,1,2)
plot(k*fs/length(x), abs(X))  %Plot X in frequency
axis([0 fs/2 0 .5]);
xlabel('frequency in Hz');
ylabel('gain');
title('rossini.wav in frequency');
wavplay(x);
```

• LaPlace Transform

The LaPlace transform can be viewed as a generalization of the Fourier transform.
The Fourier transform is:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt$$

In this equation, jω is a particular set of complex numbers that forms the vertical axis in the complex plain. We can generalize this by allowing s = σ+jω to replace jω since s can be any complex number because it has both a real part and an imaginary part.

The advantage of doing this is that $f(t)e^{-st}$ converges for many more functions than does $f(t)e^{-j\omega t}$. If there is no convergence, the integral does not exist.

The properties of the LaPlace transform as shown in Table 3.8 in the text. The most important properties are convolution, integration, and differentiation.

We can also write transfer functions in terms of the LaPlace transform. To find the frequency response of such a transfer function we need only replace s by jω to get the Fourier transform where we can evaluate the magnitude and phase with respect to frequency.

• *z*-transform

Just as the LaPlace transform is seen as a generalization of the Fourier transform, we can generalize the DTFT and get the *z*-transform. The DTFT is

$$F(\omega nT) = \sum_{n=-\infty}^{\infty} f(nT) e^{-j\omega nT}$$

In this equation the term $e^{-j\omega T}$ represents a particular set of complex numbers which have a magnitude of 1. In the complex plane $e^{-j\omega T}$ maps into the unit circle in the z-plane. To generalize this term we let z = α + jβ replace $e^{-j\omega T}$. This gives

$$F(z) = \sum_{n=-\infty}^{\infty} f(nT) z^{-n}$$

Which is the *z*-transform.

Since z can be any complex number we have far more functions which converge for the z-transform than we do for the DTFT.

The inverse z-transform is given by

$$\mathbf{Z}^{-1}[X(z)] = X[n] = \frac{1}{2\pi j} \int_{\Gamma} X(z) z^{n-1} dz,$$

Like the LaPlace transform, the inverse is very difficult to evaluate so we rely on tables of simpler functions and form the inverse z-transform by breaking more complex functions into the simpler functions for which the transform is known

The z-transform can be two-sided or one-sided. For the two-sided transform we have

$$F(z) = \sum_{n=-\infty}^{\infty} f(nT)z^{-n}$$

Where the one-sided transform is given by

$$F(z) = \sum_{n=0}^{\infty} f(nT)z^{-n}$$

You should be careful when looking at z-transform tables that you get the correct table for the transform you want. The one-sided z-transforms always include an initial condition which is the value at n = 0 where the two-sided transform goes from -∞ to +∞ and needs no initial condition.

When we use the z-transform to solve difference equations we use the one-sided transforms since we often have the initial conditions. Otherwise, we use the two-sided transform.

In this class we will solve few difference equations.

Some easy z-transforms:

Impulse

$$F(z) = \sum_{n=-\infty}^{\infty} \delta(nT)z^{-n} = z^0 = 1$$

Step

$$F(z) = \sum_{n=-\infty}^{\infty} u(nT)z^{-n} = 1 + z^{-1} + z^{-2} + ...$$

From this we can write:

$$F(z) = 1 + z^{-1} + z^{-2} + z^{-3} + \dots$$

$$zF(z) = z + z^0 + z^{-1} + z^{-2} + \dots$$

Subtract these two equations to get:

$$F(z)(1-z) = 1 - z - 1 = -z$$

Sovle to get

$$F(z) = \frac{z}{z-1}$$

Exponential

$$F(z) = \sum_{n=-\infty}^{\infty} a^n u(n) z^{-n} = 1 + a/z + a^2/z^2 + a^3/z^3 + \cdots$$

From this we write

$$F(z) = 1 + a/z + a^2/z^2 + a^3/z^3 + \cdots$$

$$(a/z)F(z) = a/z + a^2/z^2 + a^3/z^3 + \cdots$$

Subtract these two equations to get

$$F(z)(1-a/z)=1$$

Or

$$F(z)=\frac{z}{z-a}$$

Table 3.9 in the text gives many more two-sided z-transforms.

• Convergence

• Inverse