
AC 2012-3085: THE FAST FOURIER TRANSFORMS FOR TECHNOLOGISTS, ENGINEERS, AND OTHER NON-PH.D.S

Dr. William H. Blanton, East Tennessee State University

William Blanton has taught in the Engineering Technology Department at East Tennessee State University for 12 years. He is coordinator of the biomedical engineering technology concentration. His teaching interests are medical imaging and digital signal processing.

Dr. Zhibin Tan, East Tennessee State University

Department of Engineering Technology, Surveying, and Digital Media

Mr. Davin A. Strom, East Tennessee State University

The Fast Fourier Transforms for Technologists, Engineers, and Other Non-Ph.D.s

ABSTRACT.

The number of professional areas in which Fourier transforms appear is impressive. Fourier transforms play an important role in optics, medical imaging, acoustics, waveform analysis, and even business analysis. Quite often the Fourier Transform is first encountered in a formal mathematics class after the study of the Fourier series. As needed, the Fourier transform is revisited in senior level and graduate level courses. Unfortunately, many engineering technology students, science majors, and even some undergraduate engineering students are not introduced the Fourier transform.

The relevancy of the Fourier transform to many professional areas strongly qualifies the subject for introduction at an early stage in one's academic career. A good introduction to the Fourier transform is the fast Fourier transform (FFT). The FFT is an efficient algorithm based on the discrete Fourier transform (DFT) for converting time domain sequences into frequency domain sequences. *The results obtained by the FFT are exactly the same results that are obtained by the DFT, not an approximation of the DFT.* Whereas the evaluation of an N -point DFT requires N^2 complex multiplications and additions, the FFT method (assuming N is power of 2) requires $N \log_2 N$ calculations. Thus, the FFT calculations become significantly less than the DFT calculations as N gets larger. The inclusion of the FFT in the MATLAB[®] function set provides a good entry point for the study of the Fourier transform by technologists, engineers, and other non-Ph.D.s.

Introduction

In general, we live in an analog world where the physical phenomena can take on a range of continuous values. For example, temperature may vary from 0° Celsius on a cold day to 25° Celsius on a warm day. Between these two cold and warm, temperature can take on all values for all times. That is, the amplitude of the temperature is continuous over a continuous range of times.

Many of the measurements of physical phenomena are made using digital instruments that have an embedded microprocessor or microcontroller. The computer exists in a digital world that is based on a discrete system where the amplitude and time are quantized. That is, a discrete system does not allow all values for all times. Rather it allows a limited number of values at discrete instances in time.

The interfaces between the analog and discrete world is the analog-to-digital (A/D) converter for inputs and the digital-to-analog (D/A) converter for outputs. The A/D converter converts the phenomena that we observe into information that the computer can understand, and the D/A converter converts the information that the computer develops into analog data that we can use to observe, control or manipulate a specific phenomena.

We can and often use complex mathematical methods to describe observable phenomena and to control and manipulate the phenomena. Linear transforms such as the LaPlace and Fourier transforms can often be used as a tool to alter complex problems into more approachable problems. Specifically, the Fourier transform is pervasive in the theory and practice of many branches of science and engineering [1]. The FFT is often included as a function of test equipment. Thus, the discrete Fourier transform (DFT) and its offshoot, the fast Fourier

transform (FFT), have become common and powerful procedures for analyzing, manipulating, and synthesizing discrete signals, and a solid understanding of the DFT, and specifically the FFT, is mandatory for anyone working in digital signal processing [2].

Unfortunately the DFT is terribly inefficient and time consuming. For an 8-point DFT, there are N^2 , or 64 multiplications using complex numbers. As the number of points increase, the complex multiplications increase dramatically and become excessive. Fortunately, Cooley and Tukey in 1965 developed a very efficient algorithm, the Fast Fourier Transform (FFT), to implement the DFT [3]. The number of the complex multiplications for an N-point FFT is

$$\frac{N}{2} \log_2 N, \quad (1)$$

or 12. A two-million point FFT on a desk top computer may take 10 seconds while the two-million point DFT takes more than three weeks [2]. Arguably, the FFT algorithm as a process sits with the transistor and the microprocessor as a noteworthy scientific breakthrough in history. The objective is to make the technologists and scientists and other non-PhDs comfortable with the use of the FFT.

The Discrete Fourier Transform

The Discrete Fourier Transform (DFT) is a mathematical procedure [2] used to determine the

$$X(m) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nm/N} \quad (2)$$

frequency (harmonic) content of a discrete signal sequence. Note that j is $\sqrt{-1}$. The origins of the DFT are the Continuous Fourier Transform (CFT) (Eq. 3). Like most transform methods, the

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt \quad (3)$$

Fourier transform (whether DFT, CFT, or FFT) was developed to simplify the approach to solving cumbersome problems. That is, a transform converts one way of thinking about a problem to an alternative, sometimes simpler way of thinking about a problem (Figure 1). Most

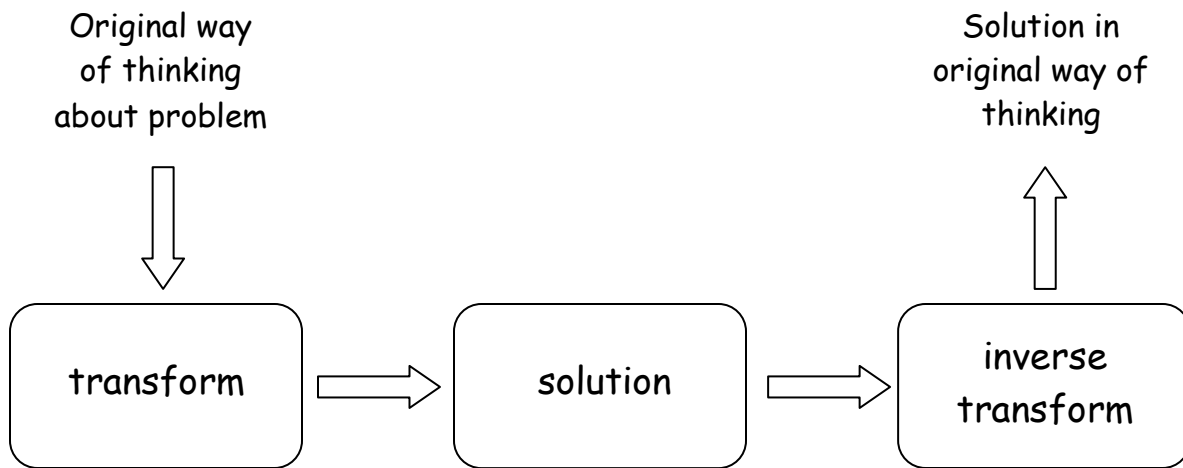


Figure 1. The Transform Method

often, the transform transfers the problem from the time domain where convolution is a complex integral into the frequency domain where convolution becomes a simple multiplication.

Once the problem has been solved in the frequency domain, the solution can be transported back into the original time domain by taking the inverse transform. Often engineering technology students are most familiar with the LaPlace transform method, but not as many are as familiar with the FFT.

One reason that many beginning engineering and science students avoid the DFT is the tangled, ominous and generally unfriendly look of the summation operator. *Euler's identity* (Eq. 4) can

$$e^{-j\theta} = \cos(\theta) - j \sin(\theta) \quad (4)$$

be used to untangle some of the mysteries of the DFT. Euler's identity can be used to separate the DFT into a real and complex part (Eq.5).

$$X(m) = \sum_{n=0}^{N-1} x(n) \left[\cos\left(\frac{2\pi nm}{N}\right) - j \sin\left(\frac{2\pi nm}{N}\right) \right] \quad (5)$$

where

- $X(m)$ = the m^{th} DFT output component, i.e., $X(0), X(1), X(2), X(3)$, etc.,
- m = the index of the DFT output in the frequency domain, $m = 0, 1, 2, 3, \dots, N-1$,
- $x(n)$ = the sequence of input samples, $x(0), x(1), x(2), x(3)$, etc.,
- n = the time domain index of the input samples, $n = 0, 1, 2, 3, \dots, N-1$,
- $j = \sqrt{-1}$, and
- N = the number of samples of the input sequence and the number of frequency points in the DFT output.

For example (see Appendix A), consider an 8-point DFT of the time function sampled at 8 kHz.

$$x(t) = \frac{1}{2} + \sin(2\pi \cdot 1000t) + 0.5 \sin\left(2\pi \cdot 3000t + \frac{3\pi}{4}\right) \quad (6)$$

From equation 6, the frequency content of the function occurs at 0 Hz (dc level), 1000 Hz, and 3000 Hz, and the signal is shown in the Figure 2. For this figure, it is difficult to determine the frequencies of the waveform. However, the DFT yields the values shown in Table 1.

	Frequency Bin	MATLAB DFT Results
X(0)	0 Hz	4
X(1)	1 kHz	-4j
X(2)	2 kHz	0
X(3)	3 kHz	1.4142 + 1.4142j
X(4)	4 kHz (folding frequency)	0
X(5)	3 kHz	1.4142 - 1.4142j
X(6)	2 kHz	0
X(7)	1 kHz	4j

Table 1. Calculated DFT Coefficients

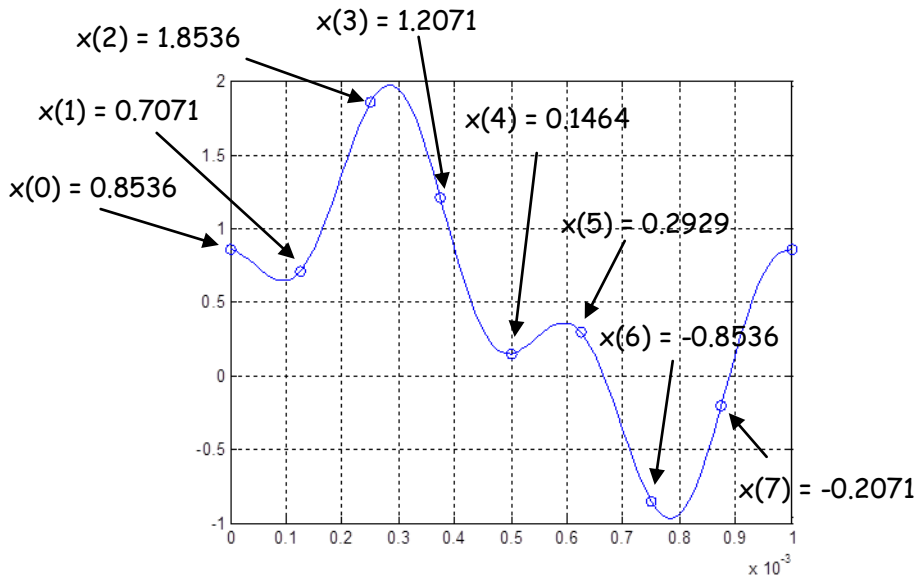


Figure 2. Example Waveform.

Note that the table shows values for frequencies 0 Hz, 1 kHz, and 3 kHz. The folding frequency is 4 kHz, where the frequencies get folded back. So the valued calculated for 5 kHz will be the conjugate of the 3 kHz signal [2].

$$\begin{aligned}
X(3) &= 1.414 + j1.414 \Rightarrow X(5) = X(3)^* = 1.414 - j1.414, \\
X(2) &= 0 \Rightarrow X(6) = X(6)^* = 0, \text{ and} \\
X(1) &= -j4 \Rightarrow X(7) = X(1)^* = j4
\end{aligned} \tag{7}$$

where * represents the complex conjugate.

Regardless, the calculations shown in Appendix A are involved and tedious, even more so as N gets larger. Each of the DFT coefficients involved in the 8-point DFT has 8 complex multiplications and additions, and there are 8 frequency bins giving a total of 64 complex calculations while the FFT takes 24 complex calculations.

The Fast Fourier Transform

Recall that the DFT is defined as the discrete frequency domain sequence, $X(m)$, where

$$X(m) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nm/N} \tag{8}$$

Let the time-domain sequence $x(n)$ be separated into two halves:

$$x(0), x(1), \dots, x\left(\frac{N}{2} - 1\right), \text{ and} \tag{9}$$

$$x\left(\frac{N}{2}\right), x\left(\frac{N}{2} + 1\right), \dots, x(N-1). \tag{10}$$

Now there are two time domain sequences, $x(0)$ through $x(3)$ and $x(4)$ through $x(7)$.

$$X(m) = \sum_{n=0}^{\frac{N}{2}-1} x(n) e^{-j2\pi nm/N} + \sum_{n=\frac{N}{2}}^{N-1} x(n) e^{-j2\pi nm/N} \tag{11}$$

In addition, the discrete frequency domain sequence, $X(m)$, can be rewritten as:

$$X(m) = \sum_{n=0}^{\frac{N}{2}-1} x(n)W^{nm} + \sum_{n=\frac{N}{2}}^{N-1} x(n)W^{nm} \quad (12)$$

where W^{nm} is called the *twiddle factor* and is defined by:

$$W^{nm} = e^{-j2\pi\frac{nm}{N}} \quad (13)$$

Note that $\frac{N}{2} + n - n = \frac{N}{2} \Rightarrow n = \frac{N}{2} + n$. Thus, without loss in generality, the index of the second summation can be changed to:

$$X(m) = \sum_{n=0}^{\frac{N}{2}-1} x(n)W^{nm} + \sum_{n=0}^{\frac{N}{2}-1} x\left(\frac{N}{2} + n\right)W^{\left(\frac{N}{2} + n\right)m} \quad (14)$$

which is:

$$X(m) = \sum_{n=0}^{\frac{N}{2}-1} x(n)W^{nm} + W^{\frac{Nm}{2}} \sum_{n=0}^{\frac{N}{2}-1} x\left(\frac{N}{2} + n\right)W^{nm} \quad (15)$$

since $W^{\frac{Nm}{2}}$ is a constant that alternates between +1 and -1, such that:

$$W^{mN/2} = e^{-jm\pi} = (e^{-j\pi})^m = (\cos \pi - j \sin \pi)^m = (-1)^m \quad (16)$$

Substituting $m = 2m$ for even m , and $m = 2m+1$ for odd m

$$X(2m) = \sum_{n=0}^{(N/2)-1} [x(n) + x(n + N/2)]W^{2nm} \quad (17)$$

$$X(2m+1) = \sum_{n=0}^{(N/2)-1} [x(n) - x(n + N/2)]W^n W^{2nm} \quad (18)$$

The FFT algorithm takes advantage of the periodicity and symmetry of the *twiddle* constants (Figure 3) to reduce the computational requirements of the FFT [3]. From the periodicity of W ,

$$W^{m+N} = W^m \quad (19)$$

Because the twiddle constant W is a function of the length N , it can be represented as W_N . Then W_N^2 can be written as $W_{N/2} = -1$.

Substituting,

$$a(n) = x(n) + x\left(n + \frac{N}{2}\right) \text{ and } b(n) = x(n) - x\left(n + \frac{N}{2}\right) \quad (20)$$

into Eq. 17 and Eq. 18 respectively, yields

$$X(2m) = \sum_{n=0}^{(N/2)-1} a(n) W_{N/2}^{nm} \quad (21)$$

$$X(2m+1) = \sum_{n=0}^{(N/2)-1} b(n) W_N^n W_{N/2}^{nm} \quad (22)$$

The N -point DFT has been decomposed (decimated) into two $N/2$ -point DFTs, an even upper half and an odd lower half. This process can be repeated to decimate the two $N/2$ -point DFTs into four $N/4$ -point DFTs (Figure 4).

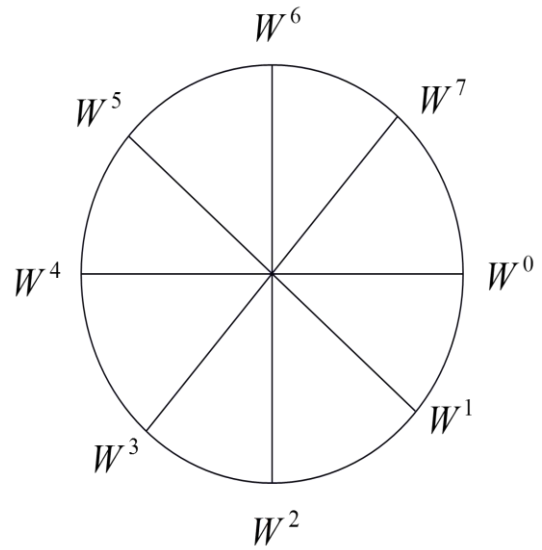


Figure 3. Twiddle Factors.

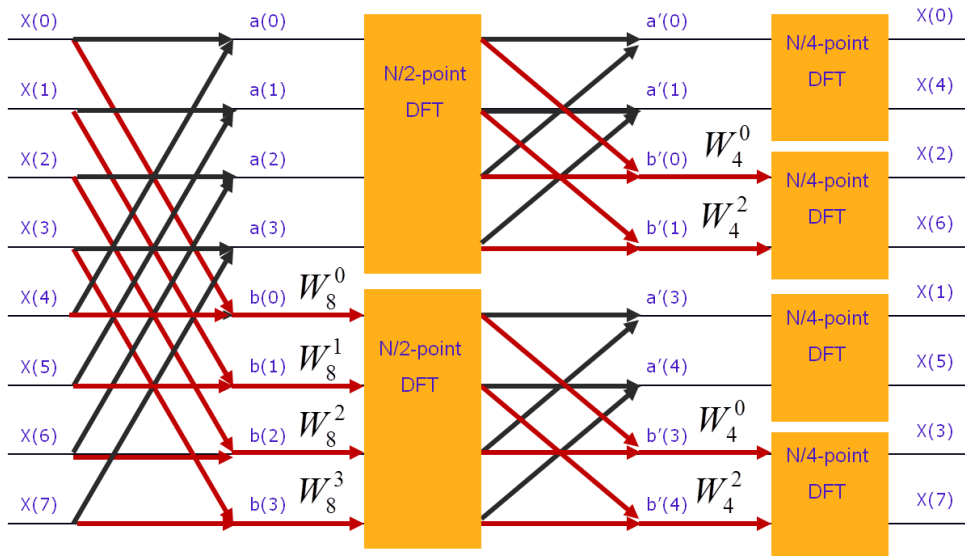
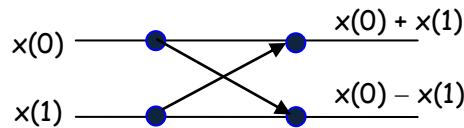


Figure 4. FFT Butterfly.

Each of the decompositions forms a butterfly, with the last decomposition being a two-point decomposition. Note that the output is scrambled with the even outputs on top and the odd



outputs on the bottom. The output index is the bit-reversed representation of the input index.

The binary index for 1_{10} is 001_2 which is 100_2 or 4_{10} when bit reversed.

Consider our example, Eq. 6.

$$x(t) = \frac{1}{2} + \sin(2\pi \cdot 1000t) + 0.5\sin\left(2\pi \cdot 3000t + \frac{3\pi}{4}\right) \quad (6)$$

Stage 1:

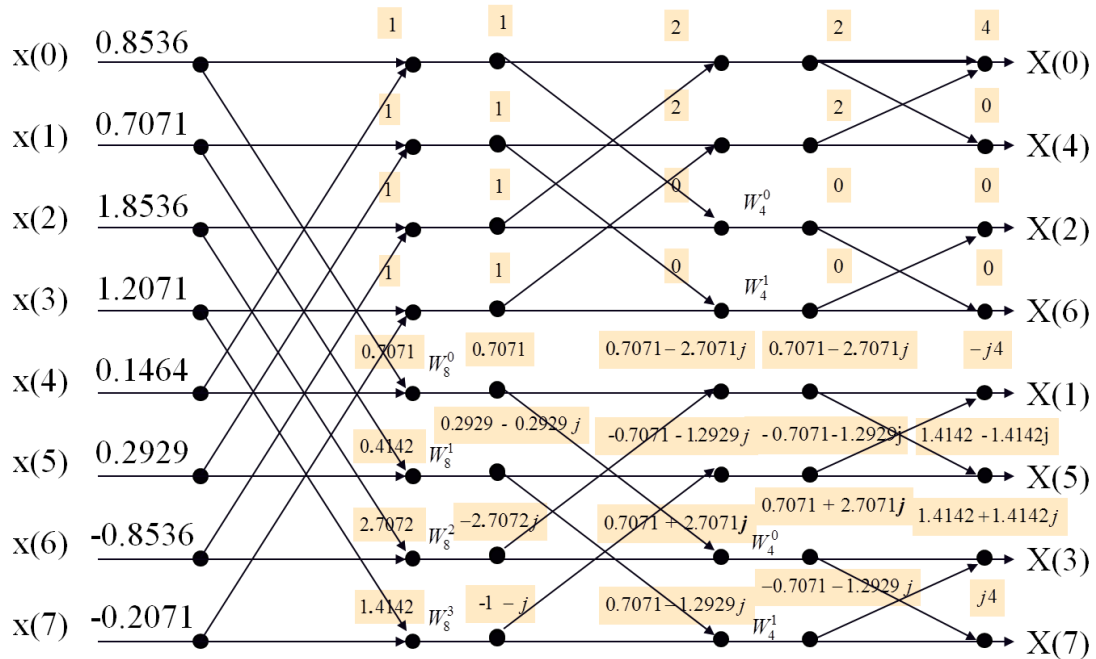
$$\begin{aligned}
 a(0) &= x(0) + x(4) = 0.8536 + 0.1464 = 1 \\
 a(1) &= x(1) + x(5) = 0.7071 + 0.2929 = 1 \\
 a(2) &= x(2) + x(6) = 1.8536 + (-0.8536) = 1 \\
 a(3) &= x(3) + x(7) = 1.2071 + (-0.2071) = 1 \\
 b(0) &= [x(0) - x(4)]W_8^0 = [0.8536 - 0.1464] \times 1 = 0.7072 \\
 b(1) &= [x(1) - x(5)]W_8^1 = [0.7071 - 0.2929] \times [0.707 - j0.707] = 0.2929 - j0.2929 \\
 b(2) &= [x(2) - x(6)]W_8^2 = [1.8536 - (-0.8536)] \times [-j] = -j2.7072 \\
 b(3) &= [x(3) - x(7)]W_8^3 = [1.2071 - (-0.2071)] \times [-0.707 - j0.707] = -1 - j
 \end{aligned} \tag{24}$$

Stage 2:

$$\begin{aligned}
 a'(0) &= a(0) + a(2) = 1 + 1 = 2 \\
 a'(1) &= a(1) + a(3) = 1 + 1 = 2 \\
 b'(0) &= [a(0) - a(2)]W_4^0 = 1 - 1 = 0 \\
 b'(1) &= [a(1) - a(3)]W_4^2 = [1 - 1] \times [-j] = 0 \\
 a'(2) &= b(0) + b(2) = 0.7072 + (-j2.7072) = 0.7072 - j2.7072 \\
 a'(3) &= b(1) + b(3) = 0.2929 - j0.2929 - 1 - j = -0.7072 - 1.2927j \\
 b'(2) &= [b(0) - b(2)]W_4^0 = 0.7072 - (-j2.7072) = 0.7072 + j2.7072 \\
 b'(3) &= [b(1) - b(3)]W_4^2 = [0.2929 - j0.2929 - (-1 - j)] \times [-j] = 0.7070 - 1.2927j
 \end{aligned} \tag{25}$$

Stage 3

$$\begin{aligned}
 X(0) &= a'(0) + a'(1) = 2 + 2 = 4 \\
 X(4) &= a'(0) - a'(1) = 2 - 2 = 0 \\
 X(2) &= b'(0) + b'(1) = 0 + 0 = 0 \\
 X(6) &= b'(0) - b'(1) = 0 - 0 = 0 \\
 X(1) &= a'(2) + a'(3) = (.7071 - 2.7071*j) + (-0.7071 - 1.2929*j) = -4j \\
 X(5) &= a'(2) - a'(3) = (.7071 - 2.7071*j) - (-0.7071 - 1.2929*j) = 1.4142 - 1.4142j \\
 X(6) &= b'(2) + b'(3) = 0.7072 + j2.7072 + 0.7070 - 1.2927j = 1.4142 + 1.4142j \\
 X(7) &= b'(2) - b'(3) = 0.7072 + j2.7072 - (0.7070 - 1.2927j) = 4j
 \end{aligned} \tag{26}$$



A MATLAB *decimation-in-frequency* protocol can be constructed to find the components of the FFT:

%DECIMATION IN FREQUENCY

```

W0=1;
W1=.707-.707i;
W2=-i;
W3=-.707-.707i;
x0=0.8536;
x1=0.7071;
x2=1.8536;
x3=1.2071;
x4=0.1464;
x5=0.2929;
x6=-0.8536;
x7=-0.2071;

```

%STAGE 1

$xx0=x0+x4;$
 $xx1=x1+x5;$
 $xx2=x2+x6;$
 $xx3=x3+x7;$
 $xx4=[x0-x4]*W0;$
 $xx5=[x1-x5]*W1;$
 $xx6=[x2-x6]*W2;$
 $xx7=[x3-x7]*W3;$

Stage 1 = {1 1 1 1 0.7072 0.2928 - 0.2928j - 2.7072j -1-j}

%STAGE 2

$X0=xx0+xx2;$
 $X1=xx1+xx3;$
 $X2=[xx0-xx2]*W0;$
 $X3=[xx1-xx3]*W2;$
 $X4=xx4+xx6;$
 $X5=xx5+xx7;$
 $X6=[xx4-xx6]*W0;$
 $X7=[xx5-xx7]*W2;$

Stage 2 = {2 2 0 0 0.7072 - 2.7072j -0.7070 - 1.2927j 0.7072 + 2.7072j 0.7070 - 1.2927j}

%STAGE 3

$XX0=X0+X1 ;$
 $XX4=X0-X1 ;$
 $XX2=X2+X3 ;$
 $XX6=X2-X3 ;$
 $XX1=X4+X5 ;$
 $XX5=X4-X5 ;$

Stage 3 = {4 0 0 0 -4j 1.4142+1.4142j 1.4142-1.4142j 4j}

**Reordered \Rightarrow $X0 = 4, X1 = -4j, X2 = 0, X3 = 1.414+1.424j, X4 = 0, X5 = 1.414-1.424j$
 $X6 = 0, X7 = 4j.$**

We can use MATLAB FFT function to arrive at an even easier solution to our problem, Table 1.

```

N=8;
Fs=8e3;
Ts=1/Fs;
t=0:Ts:1e-3;
x=0.5+sin(2*pi*1000*t)+0.5*sin(2*pi*3000*t+3*pi/4);
X=fft(x,N)

```

The results for the MATLAB FFT function are shown in Table 2.

	Frequency Bin	MATLAB FFT Results
X(0)	0 Hz	4.0000
X(1)	1 kHz	-0.0000 - 4.0000j (-4j)
X(2)	2 kHz	-0.0000 - 0.0000j (0)
X(3)	3 kHz	1.4142 + 1.4142j
X(4)	4 kHz (folding frequency)	0
X(5)	3 kHz	1.4142 - 1.4142j
X(6)	2 kHz	-0.0000 + 0.0000j (0)
X(7)	1 kHz	-0.0000 + 4.0000j (4j)

Table 2. FFT Coefficients Derived from MATLAB FFT Function.

Conclusion

The FFT is a simplified algorithm for calculating the Discrete Fourier transform, and the FFT gives exactly the same results as the DFT. Recalling the basic axioms of algebra, *values equal to the same values are equal to each other*. Therefore, functions equal to the same function are equal to each other, so that the result of the FFT function is *exactly* the same result that would be obtained by the DFT.

The FFT is a more efficient and efficacious derivation of the Discrete Fourier Transform. By taking advantage of MATLAB, especially the MATLAB FFT function, the DFT and even the

FFT derivations can be trivialized, thereby making the FFT more palatable and more usable for students in lower division engineering courses.

The Fast Fourier Transform is among the more important innovations that have revolutionized modern technology. As with many technological innovations, they appear early in the scientific journals, find their way into graduate courses and research institutions, filter down into the undergraduate courses, and eventually become general knowledge. At East Tennessee State University, we introduce the FFT in our microprocessor class at the junior level, and expand depth of the subject in our digital signal processing course that follows. However, our Agilent oscilloscopes have the FFT function that could be used in a sophomore circuit analysis course to show the frequency content of a square wave when doing Fourier series.

References

1. Bracewell, Ron. **The Fourier Transform and Its Applications**. McGraw-Hill Book Company. New York. 1965.
2. Lyons, Richard G. **Understanding Digital Signal Processing**, 2nd Edition. Pearson Education. Upper Saddle River. NJ. 2004.
3. Chassaing, Rulph and Reay, Donald. **Digital Signal Processing and Applications with the TMS320C6713 and TMS320C6416 DSK**, 2nd Edition. Wiley-Interscience. Hoboken, N.J. 2008.
4. Software was developed using MATLAB (2011a, The MathWorks, Natick, MA).

APPENDIX A DFT CALCULATIONS

From equation 6, the frequency content of the function occurs at 0 Hz (dc level), 1000 Hz, and

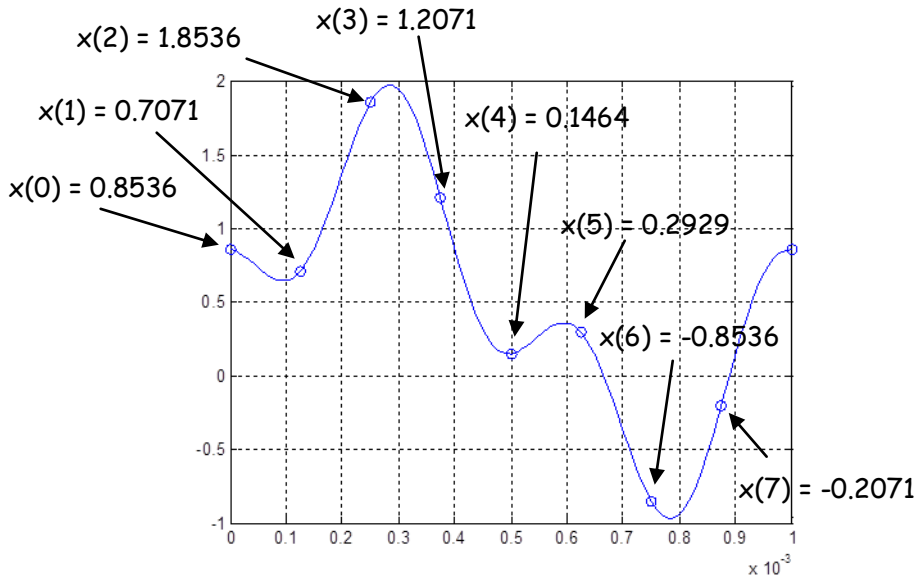


Figure 2. Example Waveform.

Applying the DFT (eq. 5) to the signal,

$$X(m) = \sum_{n=0}^{N-1} x(n) \left[\cos\left(\frac{2\pi nm}{8}\right) - j \sin\left(\frac{2\pi nm}{8}\right) \right] \quad (5)$$

The frequency samples, $X(m)$, or frequency bins, are multiples of 1000 Hz (8 kHz/8) in which $X(0)$ represents 0 Hz, $X(1)$ represents 1 kHz, $X(2)$ represents 2 kHz, $X(3)$ represents 3 kHz, and $X(4)$ represents 4kHz. An interesting phenomenon occurs at 4 kHz. The frequency content folds and DFT is the conjugate of the corresponding unfolded frequency [2]. So, $X(5)$ is the conjugate of the 3 kHz frequency bin, $X(6)$ is the conjugate of the 2 kHz frequency bin, and $X(7)$ is the conjugate of the 1 kHz frequency bin.

The input samples, $x(n)$, for the waveform (Figure 2) can be obtained using the following

MATLAB script:

```
%Decimation in Frequency
t=0:1e-3/100:1e-3;
y=0.5+sin(2*pi*1000*t)+0.5*sin(2*pi*3000*t+(3*pi/4));
plot(t,y)
hold on
t=0:1e-3/8:1e-3;
y=0.5+sin(2*pi*1000*t)+0.5*sin(2*pi*3000*t*(3*pi/4))
plot(t,y,'o')
```

$y = \{ 0.85355 \ 0.70711 \ 1.8536 \ 1.2071 \ 0.14645 \ 0.29289 \ -0.85355 \ -0.20711 \ 0.85355 \}$

As the MATLAB script indicates, the input samples are determined by taking 8 samples of the waveform over one period of the waveform. Using these values, the 8-point DFT can be calculated as shown below. Each requires 8 complex multiplications for a total of 64 (N^2) complex multiplications.

$$\begin{aligned}
 X(0) &= \sum_{n=0}^7 x(n) \left[\cos\left(\frac{2\pi nm}{8}\right) - j \sin\left(\frac{2\pi nm}{8}\right) \right] \\
 &= 0.8536[\cos(0) - j \sin(0)] + 0.7071[\cos(0) - j \sin(0)] \\
 &\quad + 1.8536[\cos(0) - j \sin(0)] + 1.2071[\cos(0) - j \sin(0)] \\
 &\quad + 0.1464[\cos(0) - j \sin(0)] + 0.2929[\cos(0) - j \sin(0)] \\
 &\quad - 0.8536[\cos(0) - j \sin(0)] - 0.2071[\cos(0) - j \sin(0)] \\
 X(0) &= 0.8536 + 0.7071 + 1.8536 + 1.2071 + 0.1464 + 0.2929 - 0.8536 - 0.2071 = 4.0
 \end{aligned}$$

$$\begin{aligned}
X(1) &= \sum_{n=0}^7 x(n) \left[\cos\left(\frac{2\pi nm}{8}\right) - j \sin\left(\frac{2\pi nm}{8}\right) \right] \\
&= 0.8536[\cos(0) - j \sin(0)] + 0.7071 \left[\cos\left(\frac{2\pi}{8}\right) - j \sin\left(\frac{2\pi}{8}\right) \right] + 1.8536 \left[\cos\left(\frac{2\pi \times 2}{8}\right) - j \sin\left(\frac{2\pi \times 2}{8}\right) \right] \\
&\quad + 1.2071 \left[\cos\left(\frac{2\pi \times 3}{8}\right) - j \sin\left(\frac{2\pi \times 3}{8}\right) \right] + 0.1464 \left[\cos\left(\frac{2\pi \times 4}{8}\right) - j \sin\left(\frac{2\pi \times 4}{8}\right) \right] \\
&\quad + 0.2929 \left[\cos\left(\frac{2\pi \times 5}{8}\right) - j \sin\left(\frac{2\pi \times 5}{8}\right) \right] - 0.8536 \left[\cos\left(\frac{2\pi \times 6}{8}\right) - j \sin\left(\frac{2\pi \times 6}{8}\right) \right] \\
&\quad - 0.2071 \left[\cos\left(\frac{2\pi \times 7}{8}\right) - j \sin\left(\frac{2\pi \times 7}{8}\right) \right]
\end{aligned}$$

$$\begin{aligned}
X(1) &= 0.8536 + 0.7071(0.707 - j0.707) + 1.8536(0 - j) + 1.2071(-0.707 - j0.707) + 0.1464(-1 - 0) \\
&\quad + 0.2929(-0.707 + j0.707) - 0.8536(0 + j) - 0.2071(0.707 + j0.707) \\
&= 0.8536 + 0.50 - j0.5 - j1.8536 - 0.8536 - j0.8536 - .1464 - 0.2071 + j0.2071 - j0.8536 - .1464 - j0.1464 \\
X(1) &= 0 - j4 = -j4
\end{aligned}$$

$$\begin{aligned}
X(2) &= \sum_{n=0}^7 x(n) \left[\cos\left(\frac{2\pi nm}{8}\right) - j \sin\left(\frac{2\pi nm}{8}\right) \right] \\
&= 0.8536[\cos(0) - j \sin(0)] + 0.7071 \left[\cos\left(\frac{2\pi \times 1 \times 2}{8}\right) - j \sin\left(\frac{2\pi \times 1 \times 2}{8}\right) \right] \\
&\quad + 1.8536 \left[\cos\left(\frac{2\pi \times 2 \times 2}{8}\right) - j \sin\left(\frac{2\pi \times 2 \times 2}{8}\right) \right] + 1.2071 \left[\cos\left(\frac{2\pi \times 3 \times 2}{8}\right) - j \sin\left(\frac{2\pi \times 3 \times 2}{8}\right) \right] \\
&\quad + 0.1464 \left[\cos\left(\frac{2\pi \times 4 \times 2}{8}\right) - j \sin\left(\frac{2\pi \times 4 \times 2}{8}\right) \right] + 0.2929 \left[\cos\left(\frac{2\pi \times 5 \times 2}{8}\right) - j \sin\left(\frac{2\pi \times 5 \times 2}{8}\right) \right] \\
&\quad - 0.8536 \left[\cos\left(\frac{2\pi \times 6 \times 2}{8}\right) - j \sin\left(\frac{2\pi \times 6 \times 2}{8}\right) \right] - 0.2071 \left[\cos\left(\frac{2\pi \times 7 \times 2}{8}\right) - j \sin\left(\frac{2\pi \times 7 \times 2}{8}\right) \right]
\end{aligned}$$

$$X(2) = 0.8536 - j0.7071 - 1.8536 + j1.2071 + 0.1464 - j0.2929 \\ + 0.8536 - j0.2071 = 0 + j0 = 0$$

$$X(3) = \sum_{n=0}^7 x(n) \left[\cos\left(\frac{2\pi nm}{8}\right) - j \sin\left(\frac{2\pi nm}{8}\right) \right] \\ = 0.8536[\cos(0) - j \sin(0)] + 0.7071 \left[\cos\left(\frac{2\pi \times 1 \times 3}{8}\right) - j \sin\left(\frac{2\pi \times 1 \times 3}{8}\right) \right] \\ + 1.8536 \left[\cos\left(\frac{2\pi \times 2 \times 3}{8}\right) - j \sin\left(\frac{2\pi \times 2 \times 3}{8}\right) \right] + 1.2071 \left[\cos\left(\frac{2\pi \times 3 \times 3}{8}\right) - j \sin\left(\frac{2\pi \times 3 \times 3}{8}\right) \right] \\ + 0.1464 \left[\cos\left(\frac{2\pi \times 4 \times 3}{8}\right) - j \sin\left(\frac{2\pi \times 4 \times 3}{8}\right) \right] + 0.2929 \left[\cos\left(\frac{2\pi \times 5 \times 3}{8}\right) - j \sin\left(\frac{2\pi \times 5 \times 3}{8}\right) \right] \\ - 0.8536 \left[\cos\left(\frac{2\pi \times 6 \times 3}{8}\right) - j \sin\left(\frac{2\pi \times 6 \times 3}{8}\right) \right] - 0.2071 \left[\cos\left(\frac{2\pi \times 7 \times 3}{8}\right) - j \sin\left(\frac{2\pi \times 7 \times 3}{8}\right) \right]$$

$$X(3) = 0.8536 + 0.7071(-0.707 - j0.707) + j1.8536 + 1.2071(0.707 - j0.707) \\ - 0.1464 + 0.2929(0.707 + j0.707) + j0.8536 + 0.2071(0.707 - j0.707)$$

$$X(3) = 0.8536 - 0.5 - j0.5 + j1.8536 + 0.8536 - j0.8536 - 0.1464 + 0.2071 + j0.2071 \\ + j0.8536 + 0.1464 - j0.1464 = 1.414 + 1.414j$$

$$X(4) = \sum_{n=0}^7 x(n) \left[\cos\left(\frac{2\pi nm}{8}\right) - j \sin\left(\frac{2\pi nm}{8}\right) \right] \\ = 0.8536[\cos(0) - j \sin(0)] + 0.7071 \left[\cos\left(\frac{2\pi \times 1 \times 4}{8}\right) - j \sin\left(\frac{2\pi \times 1 \times 4}{8}\right) \right] \\ + 1.8536 \left[\cos\left(\frac{2\pi \times 2 \times 4}{8}\right) - j \sin\left(\frac{2\pi \times 2 \times 4}{8}\right) \right] + 1.2071 \left[\cos\left(\frac{2\pi \times 3 \times 4}{8}\right) - j \sin\left(\frac{2\pi \times 3 \times 4}{8}\right) \right] \\ + 0.1464 \left[\cos\left(\frac{2\pi \times 4 \times 4}{8}\right) - j \sin\left(\frac{2\pi \times 4 \times 4}{8}\right) \right] + 0.2929 \left[\cos\left(\frac{2\pi \times 5 \times 4}{8}\right) - j \sin\left(\frac{2\pi \times 5 \times 4}{8}\right) \right] \\ - 0.8536 \left[\cos\left(\frac{2\pi \times 6 \times 4}{8}\right) - j \sin\left(\frac{2\pi \times 6 \times 4}{8}\right) \right] - 0.2071 \left[\cos\left(\frac{2\pi \times 7 \times 4}{8}\right) - j \sin\left(\frac{2\pi \times 7 \times 4}{8}\right) \right]$$

$$X(4) = 0.8536 - 0.7071 + 1.8536 - 1.2071 + 0.1464 - 0.2929 + 0.8536 + 0.2071 = 0 + j0 = 0 \quad (16)$$

Fortunately, $X(4)$ is the folding frequency (bin) for $N = 8$. Therefore, each of the remaining frequency bins will be folded back from 4 kHz (half the sampling frequency of 8 kHz) toward 0 Hz, and the DFT components will be the complex conjugates of corresponding frequency bins [2] as shown in Table 1.

	Frequency Bin	MATLAB DFT Results
X(0)	0 Hz	4
X(1)	1 kHz	-4j
X(2)	2 kHz	0
X(3)	3 kHz	1.4142 + 1.4142j
X(4)	4 kHz (folding frequency)	0
X(5)	3 kHz	1.4142 - 1.4142j
X(6)	2 kHz	0
X(7)	1 kHz	4j

Table 1. Calculated DFT Coefficients