The Discrete Fourier Transform

1 Introduction

The discrete Fourier transform (DFT) is a fundamental transform in digital signal processing, with applications in frequency analysis, fast convolution, image processing, etc. Moreover, fast algorithms exist that make it possible to compute the DFT very efficiently. The algorithms for the efficient computation of the DFT are collectively called fast Fourier transforms (FFTs). The historic paper [9] by Cooley and Tukey made well known an FFT of complexity $N \log N$, where N is the length of the data vector. A sequence of early papers [1, 5, 7, 8, 9] still serve as a good reference for the DFT and FFT. In addition to texts on Digital Signal Processing, a number of books devote special attention to the DFT and FFT [2, 3, 4, 10, 11, 12, 13, 14, 15].

The importance of Fourier analysis in general is put forth very well by Leon Cohen in [6]

> . . . Bunsen and Kirchhoff, observed (around 1865) that light spectra can be used for recognition, detection, and classification of substances because they are unique to each substance.
>
> This idea, along with its extension to other waveforms and the invention of the tools needed to carry out spectral decomposition, certainly ranks as one of the most important discoveries in the history of mankind.

Some of the most basic uses of the DFT is to compute the frequency responses of filters, to implement convolution, and spectral estimation.

2 Definition

The k^{th} DFT coefficient of a length N signal $x(n)$ is defined as

$$X^d(k) = \sum_{n=0}^{N-1} x(n)\, W_N^{-kn}, \qquad k = 0, \dots, N-1 \tag{1}$$

where

$$W_N = e^{j2\pi/N} = \cos\left(\frac{2\pi}{N}\right) + j\,\sin\left(\frac{2\pi}{N}\right)$$

is the principal N-th root of unity. Because W_N^{nk} as a function of k has a period of N, the DFT coefficients $X^d(k)$ are periodic with period N when k is taken outside the range $k = 0, \dots, N-1$. The original sequence $x(n)$ can be retrieved by the inverse discrete Fourier transform (IDFT)

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X^d(k)\, W_N^{kn}, \qquad n = 0, \dots, N-1.$$

The inverse DFT can be verified by using a simple observation regarding the principal N-th root of unity W_N. Namely,

$$\sum_{n=0}^{N-1} W_N^{nk} = N \cdot \delta(k), \qquad k = 0, \dots, N-1,$$

where $\delta(k)$ is the Kronecker delta function. For example, with $N = 5$ and $k = 0$, the sum gives

$$1 + 1 + 1 + 1 + 1 = 5.$$

For $k = 1$, the sum gives

$$1 + W_5 + W_5^2 + W_5^3 + W_5^4 = 0$$

The sums can also be visualized by looking at the illustration of the DFT matrix in Figure 1. Because W_N^{nk} as a function of k is periodic with period N, we can write

$$\sum_{n=0}^{N-1} W_N^{nk} = N \cdot \delta(\langle k \rangle_N)$$

where $\langle k \rangle_N$ denotes the remainder when k is divided by N, i.e., $\langle k \rangle_N$ is k modulo N.

To verify the inversion formula, we can substitute the DFT into the expression for the IDFT:

$$\begin{aligned}
x(n) &= \frac{1}{N} \sum_{k=0}^{N-1} \left(\sum_{l=0}^{N-1} x(l)\, W_N^{-kl} \right) W_N^{kn}, && (2)\\
&= \frac{1}{N} \sum_{l=0}^{N-1} x(l) \sum_{k=0}^{N-1} W_N^{k(n-l)}, && (3)\\
&= \frac{1}{N} \sum_{l=0}^{N-1} x(l)\, N\, \delta(\langle n-l \rangle_N), && (4)\\
&= x(n). && (5)
\end{aligned}$$

3 The DFT Matrix

The DFT of a length N signal $x(n)$ can be represented as matrix-vector product. For example, a length 5 DFT can be represented as

$$\begin{bmatrix} X^d(0) \\ X^d(1) \\ X^d(2) \\ X^d(3) \\ X^d(4) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & W^{-1} & W^{-2} & W^{-3} & W^{-4} \\ 1 & W^{-2} & W^{-4} & W^{-6} & W^{-8} \\ 1 & W^{-3} & W^{-6} & W^{-9} & W^{-12} \\ 1 & W^{-4} & W^{-8} & W^{-12} & W^{-16} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \end{bmatrix}$$

where $W = W_5$, or as

$$\mathbf{X}^d = \mathbf{F}_N \cdot \mathbf{x},$$

where $\mathbf{F}_N$ is the $N \times N$ DFT matrix whose elements are given by

$$(\mathbf{F}_N)_{l,m} = W_N^{-lm} \qquad 0 \le l, m \le N-1.$$

As the IDFT and DFT formulas are very similar, the IDFT represented as a matrix is closely related to $\mathbf{F}_N$,

$$\mathbf{F}_N^{-1} = \frac{1}{N} \mathbf{F}_N^*$$

where $\mathbf{F}_N^*$ represent the complex conjugate of $\mathbf{F}_N$.

It is very useful to illustrate the entries of the matrix $\mathbf{F}_N$ as in Figure 1, where each complex value is shown as a vector. In Figure 1, it can be seen that in the k^{th} row of the matrix the elements consist of a vector rotating clockwise with a constant increment of $2\pi k/N$. In the first row $k = 0$ and the vector rotates in increments of 0. In the second row $k = 1$ and the vector rotates in increments of $2\pi/N$.

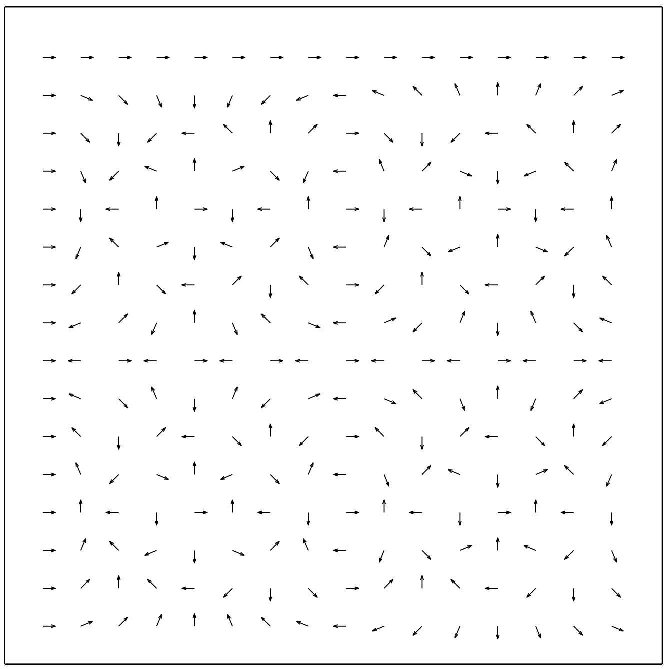

Figure 1: The 16-point DFT matrix.

4 An Example

The DFT is especially useful for representing efficiently signals that are comprised of a few frequency components. For example, the length 2048 signal shown in Figure 2 is an electrocardiogram (ECG) recording from a dog.[1] The DFT of this real signal, shown in Figure 2, is greatest at specific frequencies corresponding to the fundamental frequency and its harmonics. Clearly, the signal $x(n)$ can be represented well even when many of the small DFT $X^d(k)$ coefficients are set to zero. By discarding, or coarsely quantizing, the DFT coefficients that are small in absolute value, one obtains a more efficient representation of $x(n)$. Figure 3 illustrates the DFT coefficients when the 409 coefficients that are largest in absolute value are kept, and the remaining 1639 DFT coefficients

[1] The dog ECG data is available on the Signal Processing Information Base (SPIB) at URL `http://spib.rice.edu/`.

are set to zero. Figure 3 also shows the signal reconstructed from this "truncated" DFT. It can be seen that the reconstructed signal is a fairly accurate depiction of the original signal $x(n)$. For signals that are made up primarily of a few strong frequency components, the DFT is even more suitable for compression purposes.

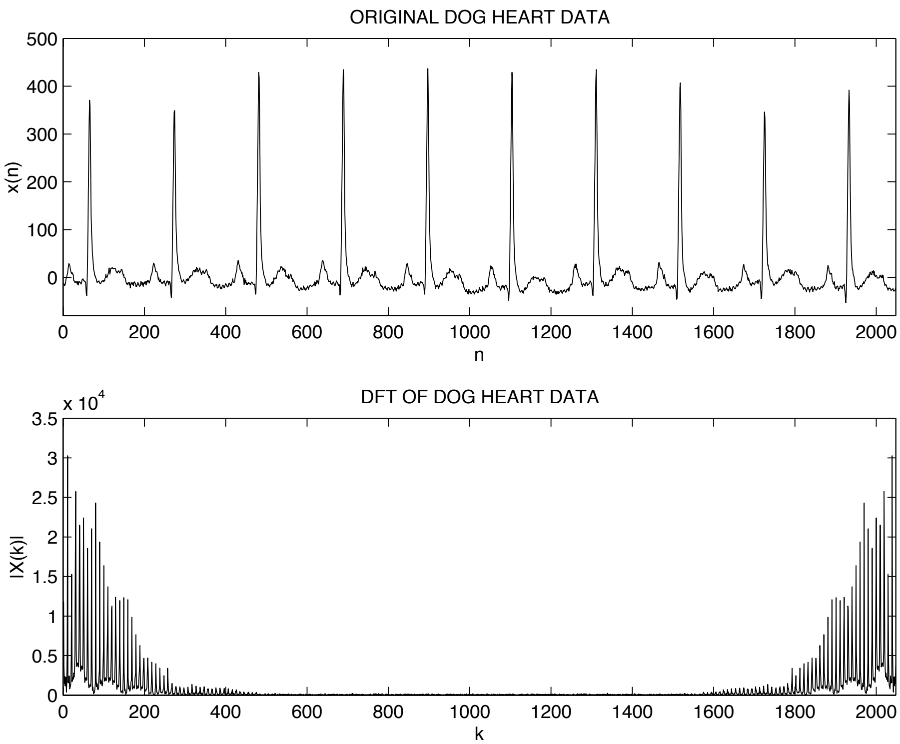

Figure 2: 2048 samples recorded of a dog heart and its DFT coefficients. The magnitude of the DFT coefficients are shown.

5 DFT Frequency Analysis

To formalize the type of frequency analysis accomplished by the DFT, it is useful to view each DFT value $X^d(k)$ as the output of a length N FIR filter $h_k(n)$. The output of the filter is given by the convolution sum

$$y_k(l) = \sum_{n=0}^{l} x(n)\, h_k(l-n).$$

When the output $y_k(l)$ is evaluated at time $l = N-1$, one has

$$y_k(N-1) = \sum_{n=0}^{N-1} x(n)\, h_k(N-1-n).$$

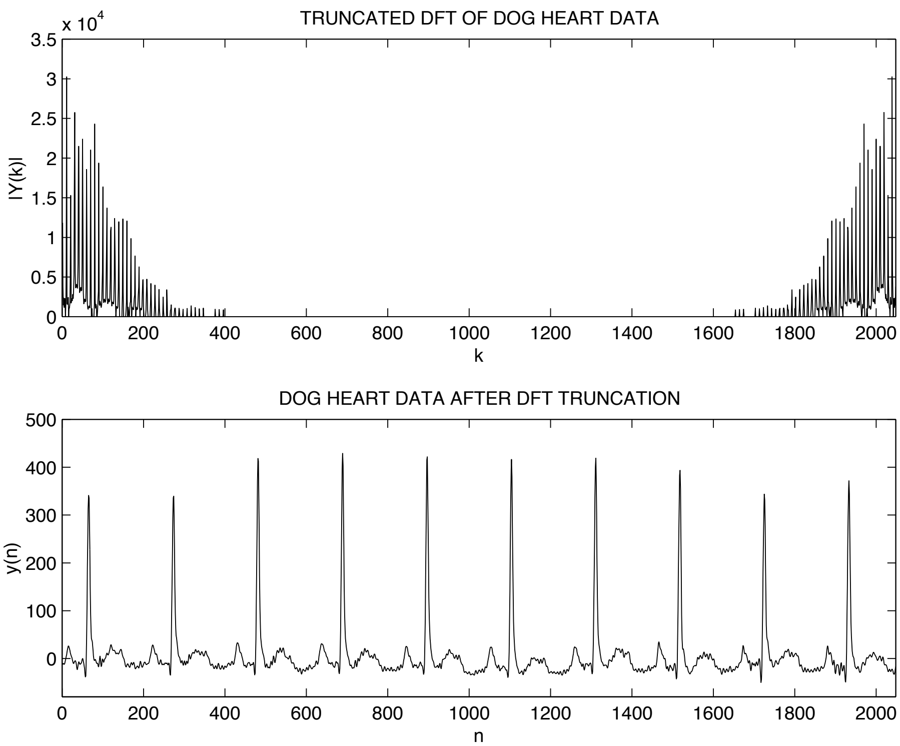

Figure 3: The truncated DFT coefficients and the time signal reconstructed from the truncated DFT.

If the filter coefficients $h_k(n)$ are defined as

$$h_k(n) = \begin{cases} W_N^{k(n-N+1)} & 0 \le n \le N-1 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

then one has

$$y_k(N-1) = \sum_{n=0}^{N-1} x(n)\, W_N^{-kn}, \tag{7}$$

$$= X^d(k). \tag{8}$$

Note that $h_k(n) = W_N^{k(n-N+1)} = W_N^k \cdot W_N^{kn}$ represents a reversal of the values W_N^{-kn} for $n = 0, \ldots, N-1$, which in turn, is the k-th row of the DFT matrix. Therefore, the DFT of a length N signal $x(n)$ can be interpreted as the output of a bank of N FIR filters of length N sampled at time $l = N-1$.

Moreover, the impulse responses $h_k(n)$ are directly related to each other through 'DFT'-modulation:

$$h_k(n) = W_N^{k(n-N+1)} \cdot p(n)$$

where the filter $h_0(n) = p(n)$ is given by

$$p(n) = \begin{cases} 1 & 0 \le n \le N-1 \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

This filter is called a *rectangular window* as it is not tapered at its ends. It follows that the Z-transforms of the filters are also simply related:

$$H_k(z) = \sum_{n=0}^{N-1} h_k(n)\, z^{-n} \tag{10}$$

$$= \sum_{n=0}^{N-1} W_N^{k(n-N+1)}\, p(n)\, z^{-n} \tag{11}$$

$$= W_N^k \sum_{n=0}^{N-1} W_N^{kn}\, p(n)\, z^{-n} \tag{12}$$

$$= W_N^k \sum_{n=0}^{N-1} \Big(W_N^{-k} z\, p(n)\Big)^{-n} \tag{13}$$

$$= W_N^k\, P(W_N^{-k} z) \tag{14}$$

where $P(z) = \sum_{n=0}^{N-1} p(n)\, z^{-n}$. That is, if each filter $h_k(n)$ in an N-channel filter bank is taken to be the time-flip of the k-th row of the DFT matrix then their Z-transforms are given by $H_k(z) = W_N^k P(W_N^{-k} z)$. $H_0(z) = P(z)$, $H_1(z) = W_N P(W_N^{-1} z)$, etc. It is instructive to view the frequency responses of the N filters $h_k(n)$, as the frequency responses of the filters $H_k(z)$ indicate the effect of the DFT on a signal. The magnitude of the frequency response of $H_k(z)$ and the zero plot in the

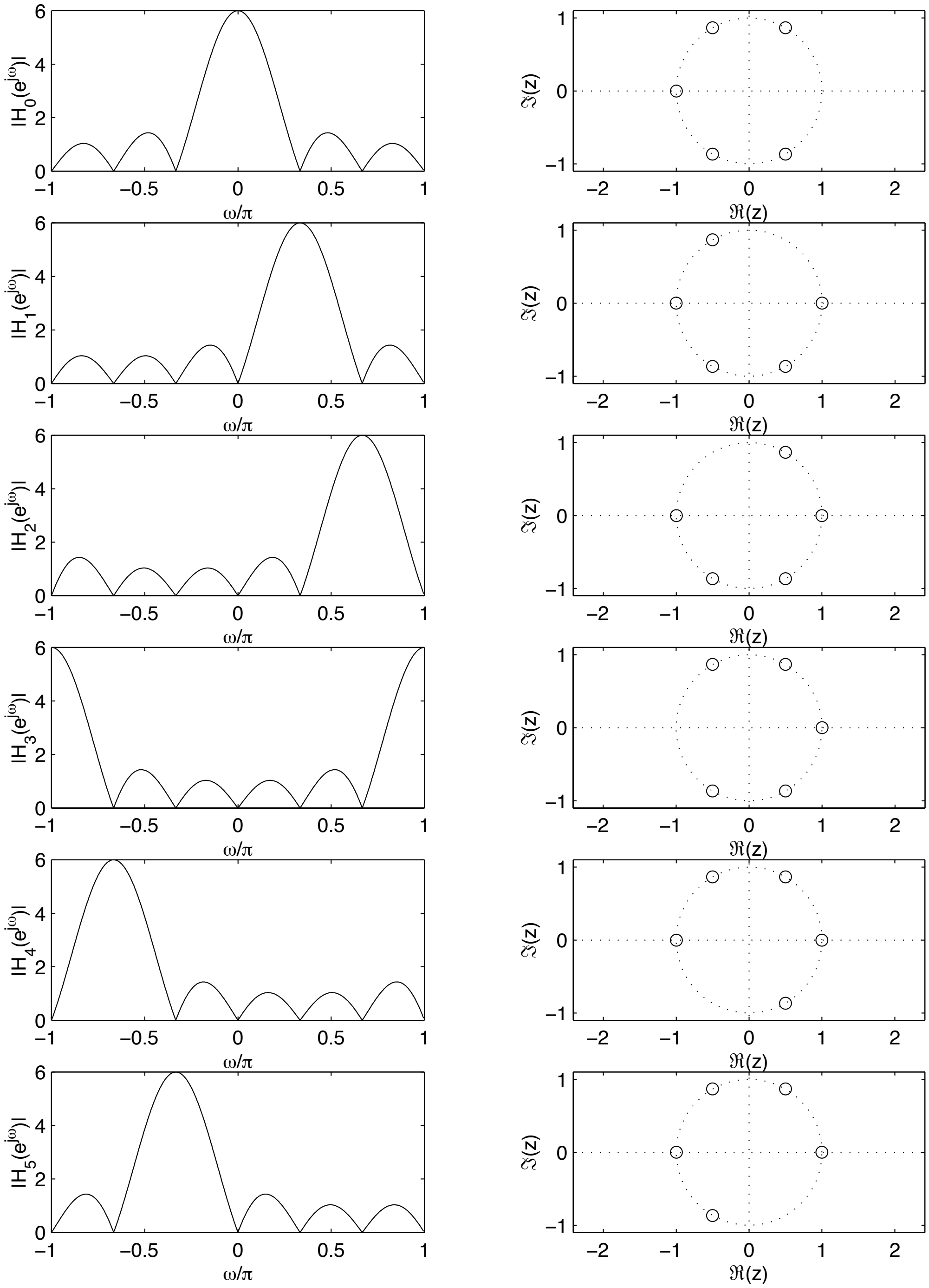

Figure 4: The magnitude of the frequency response of the filters $h_k(n)$ for $k = 0, \ldots, 5$, corresponding to a 6-point DFT. Shown on the right are the zeros of $H_k(z)$.

z-plane are given in Figure 4. Note that the zeros of $H_k(z)$ in the z-plane are simply rotated by $2\pi/N$, and that the frequency responses are shifted by the same amount. The figure makes clear the way in which the DFT performs a frequency decomposition of a signal.

The frequency response of the filter h_k is given by $H_k(e^{j\omega})$, the Discrete-time Fourier Transform (DTFT) of the impulse response:

$$H_k(e^{j\omega}) = \sum_{n=0}^{N-1} h_k(n)\, e^{-j\omega n}. \tag{15}$$

The frequency response of of the rectangular window $p(n)$ is given by

$$\begin{aligned} P(e^{j\omega}) &= \sum_{n=0}^{N-1} 1 \cdot e^{-j\omega n} && (16)\\ &= \frac{1 - e^{-jN\omega}}{1 - e^{-j\omega}} && (17)\\ &= \frac{e^{-j\omega N/2}\left(e^{j\omega N/2} - e^{-j\omega N/2}\right)}{e^{-j\omega/2}\left(e^{j\omega/2} - e^{-j\omega/2}\right)} && (18)\\ &= e^{-j\omega(N-1)/2} \cdot \frac{\sin \frac{N}{2}\omega}{\sin \frac{1}{2}\omega}. && (19) \end{aligned}$$

The function $\sin\left(\frac{N}{2}\omega\right)/\sin\left(\frac{1}{2}\omega\right)$ is called the *digital sinc* function, for its resemblance to the usual sinc function.

6 Symmetric and Anti-symmetric Signals

Because the DFT operates on finite-length data vectors, it is useful to define two types of symmetries as follows. When $x(n)$ is periodically extended outside the range $n = 0, \ldots, N-1$, the following definitions for symmetric and anti-symmetric sequences are consistent with the their usual definitions for sequences which are not finite in length.

Symmetry: Let $x(n)$ be a real-valued length N data sequence, for $n = 0, \ldots, N-1$, then $x(n)$ is *symmetric* if

$$x(N-n) = x(n), \qquad k = 1, \ldots, N-1.$$

Note that an *even*-length N symmetric sequence $x(n)$ is fully described by its first $N/2+1$ values. For example, a length 6 symmetric sequence is fully determined by its first 4 values. On the other hand, an *odd*-length N symmetric sequence $x(n)$ is fully described by its first $(N+1)/2$ values. For example, a length 7 symmetric sequence is fully determined by its first 4 values. For both even- and odd-length sequences, the number of values that determine a length N symmetric sequence is $\lfloor N/2+1 \rfloor$ where $\lfloor k \rfloor$ denotes the greatest integer smaller than or equal to k.

Anti-symmetry: A real-valued length N data sequence is *anti*-symmetric if

$$x(0) = 0 \qquad \text{and} \qquad x(N-k) = -x(k), \qquad k = 1, \ldots, N-1.$$

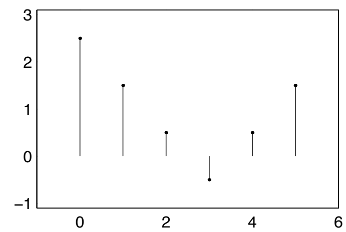

Figure 5: Illustration of even-length symmetric sequence.

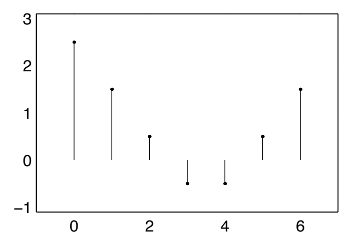

Figure 6: Illustration of odd-length symmetric sequence.

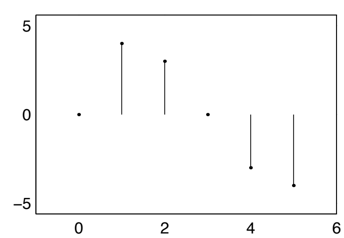

Figure 7: Illustration of even-length anti-symmetric sequence.

Note that an *even*-length N anti-symmetric sequence $x(n)$ is fully described by $N/2 - 1$ values. For example, a length 6 anti-symmetric sequence is fully determined by 2 values. On the other hand, an *odd*-length N anti-symmetric sequence $x(n)$ is fully described by $(N-1)/2$ values. For example, a length 7 anti-symmetric sequence is fully determined by 3 values: For both even- and

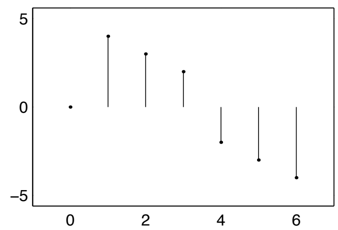

Figure 8: Illustration of odd-length anti-symmetric sequence.

odd-length sequences, the number of values that determine a length N anti-symmetric sequence is $\lceil N/2 - 1 \rceil$ where $\lceil k \rceil$ denotes the smallest integer greater than or equal to k.

7 Symmetry Properties of the DFT

To state the symmetry properties of the DFT, it is useful to introduce the notation $X_r^d(k)$ and $X_i^d(k)$ for the real and imaginary parts of $X^d(k)$. Similarly, $x_r(n)$ and $x_i(n)$ are used to denote the real and imaginary parts of $x(n)$.

If $x(n)$ is a length N data vector and ...

1. if $x(n)$ is *real-valued*, then

$$X^d(k) = X^{d*}(N-k), \qquad k = 1, \dots, N-1,$$

i.e., the real part of $X^d(k)$ is symmetric, and the imaginary part of $X^d(k)$ is anti-symmetric.

2. if $x(n)$ is *real-valued and symmetric*, then

$$X^d(k) = X_r^d(k), \qquad X_r^d(k) = X_r^d(N-k), \quad k = 1, \dots, N-1,$$

i.e., $X^d(k)$ is purely real and symmetric.

3. if $x(n)$ is *real-valued and anti-symmetric*, then

$$X^d(k) = j\, X_i^d(k), \qquad X_i^d(k) = -X_i^d(N-k), \quad k = 1, \dots, N-1,$$

i.e., $X^d(k)$ is purely imaginary and anti-symmetric.

4. if $x(n)$ is *purely imaginary*, then

$$X^d(k) = -X^{d*}(N-k), \qquad k = 1, \dots, N-1,$$

i.e., the real part of $X^d(k)$ is anti-symmetric, and the imaginary part of $X^d(k)$ is symmetric.

Table 1: DFT Symmetry Properties

$\mathbf{x}$ is purely real		$\mathbf{X}_r^d$ is symmetric,	$\mathbf{X}_i^d$ is anti-symmetric
$\mathbf{x}$ is purely real,	$\mathbf{x}_r$ is symmetric	$\mathbf{X}_r^d$ is symmetric,	$\mathbf{X}^\mathbf{d}$ is purely real
$\mathbf{x}$ is purely real,	$\mathbf{x}_r$ is anti-symmetric	$\mathbf{X}^\mathbf{d}$ is purely imaginary,	$\mathbf{X}_i^d$ is anti-symmetric
$\mathbf{x}$ is purely imaginary		$\mathbf{X}_r^d$ is anti-symmetric,	$\mathbf{X}_i^d$ is symmetric
$\mathbf{x}$ is purely imaginary,	$\mathbf{x}_i$ is symmetric	$\mathbf{X}^\mathbf{d}$ is purely imaginary,	$\mathbf{X}_i^d$ is symmetric
$\mathbf{x}$ is purely imaginary,	$\mathbf{x}_i$ is anti-symmetric	$\mathbf{X}_r^d$ is anti-symmetric,	$\mathbf{X}^\mathbf{d}$ is purely real

5. if $x(n)$ is *purely imaginary and $x_i(n)$ is symmetric*, then

$$X^d(k) = X_i^d(k), \qquad X_i^d(k) = X_i^d(N-k), \quad k = 1, \ldots, N-1,$$

i.e., $X^d(k)$ is purely imaginary and symmetric.

6. if $x(n)$ is *purely imaginary and $x_i(n)$ is anti-symmetric*, then

$$X^d(k) = X_r^d(k), \qquad X_r^d(k) = -X_r^d(N-k), \quad k = 1, \ldots, N-1,$$

i.e., $X^d(k)$ is purely real and anti-symmetric.

These properties are summarized in Table 1.

These properties explain why the total number of parameters needed to describe the original data sequence $x(n)$ is the same after the DFT is performed. For example, consider a real-valued length 6 sequence $x(n)$ and its DFT:

$$\mathbf{x} = \begin{bmatrix} 1 \\ 3 \\ 5 \\ 6 \\ 7 \\ 2 \end{bmatrix} \qquad \mathbf{X}^d = \begin{bmatrix} 24.0000 \\ -8.5000 \\ -1.5000 \\ 2.0000 \\ -1.5000 \\ -8.5000 \end{bmatrix} + j \begin{bmatrix} 0 \\ 0.8660 \\ -2.5981 \\ 0 \\ 2.5981 \\ -0.8660 \end{bmatrix}.$$

It is clear that there are a total of 6 distinct values in the DFT coefficients $X^d(k)$ for this example.

In general, for a length N real-valued sequence $x(n)$, the symmetric $X_r^d(k)$ is determined by $\lfloor N/2+1 \rfloor$ values, and the anti-symmetric $X_i^d(k)$ is determined by $\lceil N/2-1 \rceil$ values. Therefore, even though the DFT $X^d(k)$ of a length N real-valued sequence $x(n)$ is complex-valued, it is fully determined by exactly N values. The number of parameters is the same in both $x(n)$ and $X^d(k)$.

Recall that an even-length real-valued symmetric sequence $x(n)$ is determined by its first $N/2+1$ values. By the symmetry property above, the same is true for the DFT $X^d(k)$. An odd-length real-valued symmetric sequence $x(n)$ is determined by its first $(N+1)/2$ values. By the symmetry property above, the same is true for the DFT $X^d(k)$ The symmetry properties for real-valued symmetric sequences are especially useful because they can be used to develop useful DFT-based transforms that yield real-valued coefficients.

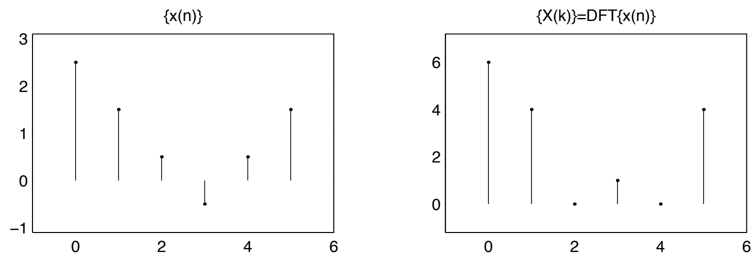

Figure 9: Illustration of DFT symmetric property.

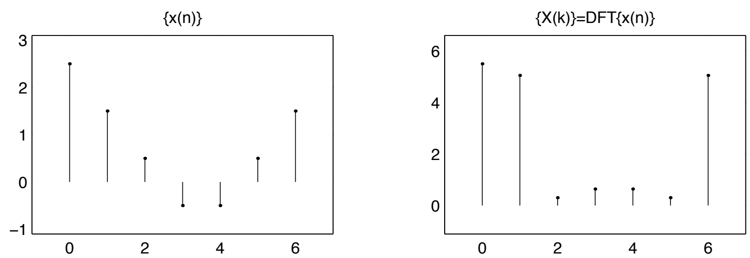

Figure 10: Illustration of DFT symmetric property.

8 Other Properties of the DFT

The text book also gives the other main properties of the DFT:

1. linearity
2. periodicity
3. time-reversal
4. circular time shift
5. circular frequency shift
6. complex conjugation
7. Parseval's theorem
8. convolution

The convolution property of the DFT is somewhat different from the convolution property for the continuous-time Fourier transform, so it deserves special attention.

9 Circular Convolution

Circular convolution is also called periodic or cyclic convolution.

Suppose $x(n)$ and $g(n)$ are both finite length signals, and that they have the same length:

$$\begin{aligned} &x(n), \quad n = 0, \ldots, N-1 \\ &g(n), \quad n = 0, \ldots, N-1. \end{aligned}$$

Regular (linear) convolution can be written as

$$\begin{aligned} x(n) * g(n) = x(0)\, g(n) + x(1)\, g(n-1) + x(2)\, g(n-2) + \\ \cdots + x(N-1)\, g(n-(N-1)) \end{aligned}$$

Now, for periodic, or cyclic, convolution, the shift $g(n-m)$ is taken to be a *circular* shift.

So the circular convolution

$$v(n) = [x(0)\ x(1)\ x(2)\ x(3)] \otimes [g(0)\ g(1)\ g(2)\ g(3)]$$

can be written out as

$$\begin{aligned} v(n) = \ & x(0)\ [g(0)\ g(1)\ g(2)\ g(3)] + \\ & x(1)\ [g(3)\ g(0)\ g(1)\ g(2)] + \\ & x(2)\ [g(2)\ g(3)\ g(0)\ g(1)] + \\ & x(3)\ [g(1)\ g(2)\ g(3)\ g(0)] \end{aligned}$$

Circular convolution property of the DFT

The the DFT of the circular convolution of two finite (equal) length signals is the product of the DFT. If

$$z(n) = x(n) \otimes y(n)$$

then

$$Z^d(k) = X^d(k) \cdot Y^d(k)$$

9.1 Linear Convolution via Circular Convolution

The linear convolution of two finite signals can be computed by appending zeros to each of the signals and then employing circular convolution. In this way, the DFT can be used to implement linear convolution. (See the text). Appending zeros is commonly called *zero padding*.

References

[1] G. D. Bergland. A guided tour of the fast Fourier transform. *IEEE Spectrum*, 6:41–52, 1969. Also in [14].

[2] R. E. Blahut. *Fast Algorithms for Digital Signal Processing*. Addison-Wesley, 1985.

[3] E. O. Brigham. *The Fast Fourier Transform and its Applications.* Prentice-Hall, 1988.

[4] C. S. Burrus and T. W. Parks. *DFT/FFT and Convolution Algorithms.* John Wiley and Sons, 1985.

[5] W. T. Cochran, J. W. Cooley, D. L. Favin, H. D. Helms, R. A. Kaenel, W. W. Lang, G. C. Maling, Jr., D. E. Nelson, C. M. Rader, and P. D. Welch. What is the Fast Fourier Transform? *IEEE Trans. Audio Electroacoust.*, 15:45–55, 1967. Also in [14].

[6] L. Cohen. *Time-Frequency Analysis.* Prentice Hall, 1995.

[7] J. W. Cooley, P. A. W. Lewis, and P. D. Welch. Historical notes on the fast Fourier transform. *IEEE Trans. Audio Electroacoust.*, 15:76–79, 1967. Also in [14].

[8] J. W. Cooley, P. A. W. Lewis, and P. D. Welch. The finite Fourier transform. *IEEE Trans. Audio Electroacoust.*, 17:77–85, 1969. Also in [14].

[9] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301, 1965. Also in [14].

[10] D. F. Elliott and K. R. Rao. *Fast Transforms: algorithms, analyses, applications.* Academic Press, 1982.

[11] C. Van Loan. *Computational Frameworks for the Fast Fourier Transform.* SIAM, 1992.

[12] J. H. McClellan and C. M. Rader. *Number Theory in Digital Signal Processing.* Prentice Hall, 1979.

[13] H. J. Nussbaumer. *Fast Fourier Transform and Convolution Algorithms.* Springer-Verlag, 1982.

[14] L. R. Rabiner and C. M. Rader, editors. *Digital Signal Processing.* IEEE Press, 1972.

[15] W. W. Smith and J. M. Smith. *Handbook of real-time Fast Fourier Transforms.* IEEE Press, 1995.