Digital Signal Processing
Properties of the Discrete Fourier Transform

D. Richard Brown III

Some Useful Properties of the DFT

TABLE 8.2 SUMMARY OF PROPERTIES OF THE DFT

	Finite-Length Sequence (Length N)	N-point DFT (Length N)
1.	$x[n]$	$X[k]$
2.	$x_1[n], x_2[n]$	$X_1[k], X_2[k]$
3.	$ax_1[n] + bx_2[n]$	$aX_1[k] + bX_2[k]$
4.	$X[n]$	$Nx[((-k))_N]$
5.	$x[((n-m))_N]$	$W_N^{km} X[k]$
6.	$W_N^{-\ell n} x[n]$	$X[((k-\ell))_N]$
7.	$\sum_{m=0}^{N-1} x_1[m] x_2[((n-m))_N]$	$X_1[k] X_2[k]$
8.	$x_1[n] x_2[n]$	$\frac{1}{N} \sum_{\ell=0}^{N-1} X_1[\ell] X_2[((k-\ell))_N]$
9.	$x^*[n]$	$X^*[((-k))_N]$
10.	$x^*[((-n))_N]$	$X^*[k]$

Notation: $((n))_N$ means n modulo N and $W_N = e^{-j2\pi/N}$.

Time-Shifting Property: DTFT vs. DFS vs. DFT

Recall the time-shifting property of the DTFT for the sequence $x[n] \leftrightarrow X(e^{j\omega})$:

$$x[n-m] \quad \leftrightarrow \quad e^{-j\omega m} X(e^{j\omega}) \quad \text{for } \omega \in \mathbb{R}$$

For the DFS of a periodic sequence $\tilde{x}[n] \leftrightarrow \tilde{X}[k]$, we have

$$\tilde{x}[n-m] \quad \leftrightarrow \quad e^{-j2\pi km/N} \tilde{X}[k] \quad \text{for } k \in \mathbb{Z}$$

Now denote $\tilde{x}[n] = x[n] * \tilde{p}[n]$ with $x[n]$ a length-N sequence and $\tilde{p}[n] = \sum_{r=-\infty}^{\infty} \delta[n - rN]$ a periodic impulse train. Since

$$\tilde{x}[n] = x[((n))_N] \qquad \text{and} \qquad \tilde{X}[k] = X[k] \quad k = 0, 1, \dots, N-1$$

we see a linear shift of $\tilde{x}[n]$ corresponds to a **circular shift** of $x[n]$. Hence

$$x[((n-m))_N] \quad \leftrightarrow \quad e^{-j2\pi km/N} X[k] \quad \text{for } k = 0, 1, \dots, N-1$$

for the DFT.

Convolution Property: DTFT vs. DFT

Recall the convolution property of the DTFT:

$$x_1[n] * x_2[n] \quad \leftrightarrow \quad X_1(e^{j\omega})X_2(e^{j\omega})$$

for all $\omega \in \mathbb{R}$ if the DTFTs both exist. Note this relation holds for infinite length or finite length sequences (the sequences don't need to have the same length.)

What about

$$x_1[n] * x_2[n] \quad \overset{?}{\leftrightarrow} \quad X_1[k]X_2[k]$$

for $k = 0, \dots, N-1$. One problem is that $x_1[n]$ and $x_2[n]$ must be the same length for this to make sense at all. Another problem is that the IDFT of $X_1[k]X_2[k]$ can't have the correct length of $2N-1$.

Example:

$$x_1[n] = \{\underline{1}, 1\} \Leftrightarrow X_1[k] = \{\underline{2}, 0\}$$

$$x_2[n] = \{\underline{1}, -1, 1, -1\} \Leftrightarrow X_2[k] = \{\underline{0}, 0, 4, 0\}$$

$$x_1[n] * x_2[n] = \{\underline{1}, 0, 0, 0, -1\} \Leftrightarrow \text{DFT}(x_1[n] * x_2[n]) = \{\underline{0}, 0.69 - 0.95i, 1.81 - 0.59i, 1.81 + 0.59i, 0.69 + 0.95i\}$$

Periodic Convolution (DFS) and Circular Convolution (DFT)

For the DFS, we have the **periodic convolution** property

$$\tilde{x}_3[n] = \sum_{k=0}^{N-1} \tilde{x}_1[k]\tilde{x}_2[n-k] \quad \forall n \in \mathbb{Z} \quad \leftrightarrow \quad \tilde{X}_3[k] = \tilde{X}_1[k]\tilde{X}_2[k] \quad \forall k \in \mathbb{Z}$$

where $\tilde{x}_1[n] = x_1[n] * \tilde{p}[n]$ and $\tilde{x}_2[n] = x_2[n] * \tilde{p}[n]$ with $x_1[n]$ and $x_2[n]$ both length-N sequences. Since $\tilde{x}_i[n] = x_i[((n))_N]$ and $\tilde{X}_i[n] = X_i[((k))_N]$ for $n = 0, \dots, N-1$ and $k = 0, \dots, N-1$, we can use the periodic convolution property of the DFS to write

$$x_3[n] = \sum_{k=0}^{N-1} x_1[k]x_2[((n-k))_N] \quad n = 0, \dots, N-1$$

$$\leftrightarrow \quad X_3[k] = X_1[k]X_2[k] \quad \forall k = 0, \dots, N-1$$

We can define the circular convolution operator for two length-N sequences as

$$x_3[n] = x_1[n] \circledN x_2[n] = \sum_{k=0}^{N-1} x_1[k]x_2[((n-k))_N] = \sum_{k=0}^{N-1} x_1[((n-k))_N]x_2[k].$$

Note the circular convolution result $x_3[n]$ will also be length-N.

See Matlab function `cconv`.

Circular Convolution Matrix

To illustrate the idea, suppose we want to circularly convolve $\{a[0], a[1], a[2]\}$ and $\{b[0], b[1], b[2]\}$. Note $N = 3$ here. Applying the definition, we can write

$$\begin{array}{rcllll} c[0] & = & a[0]b[0] & +a[1]b[2] & +a[2]b[1] \\ c[1] & = & a[0]b[1] & +a[1]b[0] & +a[2]b[2] \\ c[2] & = & a[0]b[2] & +a[1]b[1] & +a[2]b[0] \end{array}$$

This is the same as

$$\begin{bmatrix} c[0] \\ c[1] \\ c[2] \end{bmatrix} = \underbrace{\begin{bmatrix} a[0] & a[2] & a[1] \\ a[1] & a[0] & a[2] \\ a[2] & a[1] & a[0] \end{bmatrix}}_{\text{circular conv matrix}} \begin{bmatrix} b[0] \\ b[1] \\ b[2] \end{bmatrix} = \underbrace{\begin{bmatrix} b[0] & b[2] & b[1] \\ b[1] & b[0] & b[2] \\ b[2] & b[1] & b[0] \end{bmatrix}}_{\text{circular conv matrix}} \begin{bmatrix} a[0] \\ a[1] \\ a[2] \end{bmatrix}$$

Like linear convolution, the circular convolution matrix has a Toeplitz (actually "circulant") structure. Unlike linear convolution, the circular convolution matrix is square ($N \times N$).

Circular Convolution Example

Suppose $N = 5$ and

$$x_1[n] = \delta[n-1]$$
$$x_2[n] = N - n$$

for $n = 0, \dots, 4$. Both finite-length sequences are equal to zero for all other values of n. This figure illustrates how to compute

$$x_3[n] = \sum_{m=0}^{4} x_1[m] x_2[((n-m))_N]$$

for $n = 0, \dots, 4$. The final result is $x_3[n] = x_2[((n-1))_4]$, i.e., a circular shift of $x_2[n]$ by one sample. You can confirm this result easily in MATLAB as well by computing both DFTs, computing the product $X_1[k]X_2[k]$, and computing the IDFT.