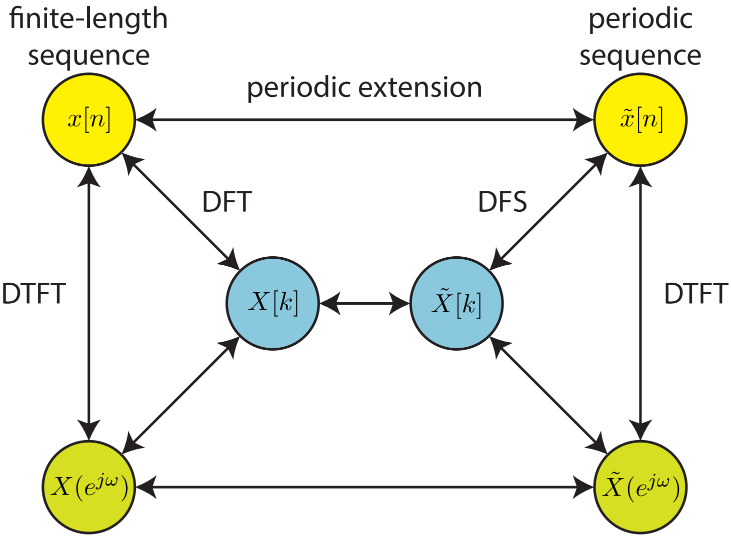
Digital Signal Processing
The Discrete Fourier Transform

D. Richard Brown III

Big Picture


finite-length
sequence
periodic
sequence
periodic extension
$x[n]$
$\tilde{x}[n]$
DFT
DFS
DTFT
$X[k]$
$\tilde{X}[k]$
DTFT
$X(e^{j\omega})$
$\tilde{X}(e^{j\omega})$

The Discrete Fourier Transform (DFT)

Definition: Given a length-$N < \infty$ sequence $\{x[n]\}$, the N-point DFT $\{X[k]\}$ is

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N} = \sum_{n=0}^{N-1} x[n] W_N^{kn} \qquad \text{for } k = 0, \dots, N-1$$

where the "twiddle factors" $W_N := e^{-j2\pi/N}$. Recall the DTFT of a finite-length sequence $\{x[n]\}$ with length N:

$$X(e^{j\omega}) = \sum_{n=0}^{N-1} x[n] e^{-j\omega n}$$

Remarks:

- The DTFT takes a discrete-time sequence $x[n]$ and produces a continuous-frequency output $X(e^{j\omega})$.
- It should be clear that the DFT is a length-N sampled version of the DTFT, i.e., $X[k] = X(e^{j\omega})|_{\omega = 2\pi k/N}$ for $k = 0, \dots, N-1$.
- The DFT takes a finite-length discrete-time sequence and produces a finite-length discrete-frequency output $X[k]$.

Relation to the Discrete Fourier Series (DFS)

Recall, for a periodic signal $\tilde{x}[n]$ with period N, the DFS coefficients can be computed as

$$\tilde{X}[k] = \sum_{n=0}^{N-1} \tilde{x}[n] e^{-j2\pi kn/N}$$

for all $k \in \mathbb{Z}$. For the DFT, we have

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$$

for $k = 0, \dots, N-1$. Since $\tilde{x}[n] = x[n]$ for $n = 0, \dots, N-1$, we can write

$$X[k] = \begin{cases} \tilde{X}[k] & k = 0, \dots, N-1 \\ 0 & \text{otherwise.} \end{cases}$$

Conversely, we can say $\tilde{X}[k] = X[k] * \sum_{r=-\infty}^{\infty} \delta[n - rN]$.

The DFT and DFS are related in the same way as the finite length sequence $x[n]$ and the periodic sequence $\tilde{x}[n]$.

The Discrete Fourier Transform (DFT)

The DFT is a linear operation that can be conveniently represented as a matrix-vector product. For example, suppose $N = 3$. The DFT is

$$\begin{aligned} X[0] &= W_3^{00} x[0] + W_3^{01} x[1] + W_3^{02} x[2] \\ X[1] &= W_3^{10} x[0] + W_3^{11} x[1] + W_3^{12} x[2] \\ X[2] &= W_3^{20} x[0] + W_3^{21} x[1] + W_3^{22} x[2] \end{aligned}$$

which is the same thing as

$$\begin{aligned} \begin{bmatrix} X[0] \\ X[1] \\ X[2] \end{bmatrix} &= \begin{bmatrix} W_3^{00} & W_3^{01} & W_3^{02} \\ W_3^{10} & W_3^{11} & W_3^{12} \\ W_3^{20} & W_3^{21} & W_3^{22} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & 1 \\ 1 & e^{-j2\pi/3} & e^{-j4\pi/3} \\ 1 & e^{-j4\pi/3} & e^{-j8\pi/3} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \end{bmatrix} \\ \boldsymbol{X} &= \boldsymbol{G}\boldsymbol{x} \end{aligned}$$

Matrix Hermitian

The notation $\boldsymbol{G}^H$ means "matrix Hermitian" and is defined as the complex conjugate transpose of a matrix. For example:

$$\boldsymbol{G} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad \boldsymbol{G}^H = \begin{bmatrix} a^* & c^* \\ b^* & d^* \end{bmatrix}.$$

Let's look at the $\boldsymbol{G}$ from our length-3 DFT:

$$\boldsymbol{G} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & e^{-j2\pi/3} & e^{-j4\pi/3} \\ 1 & e^{-j4\pi/3} & e^{-j8\pi/3} \end{bmatrix} \qquad \boldsymbol{G}^H = \begin{bmatrix} 1 & 1 & 1 \\ 1 & e^{j2\pi/3} & e^{j4\pi/3} \\ 1 & e^{j4\pi/3} & e^{j8\pi/3} \end{bmatrix}.$$

It is not difficult to confirm that

$$\frac{1}{3}\boldsymbol{G}^H\boldsymbol{G} = \frac{1}{3}\boldsymbol{G}\boldsymbol{G}^H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The DFT is an orthogonal transform. Since $\frac{1}{N}\boldsymbol{G}^H\boldsymbol{G}\boldsymbol{x} = \boldsymbol{x}$, this also tells us how to compute the IDFT.

The Inverse Discrete Fourier Transform (IDFT)

Given our definition of the DFT matrix $\boldsymbol{G}$, the IDFT can be computed as

$$\boldsymbol{x} = \frac{1}{N}\boldsymbol{G}^H\boldsymbol{X}$$

or, equivalently,

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j2\pi kn/N} = \frac{1}{N}\sum_{k=0}^{N-1} X[k]W_N^{-kn} \qquad \text{for } n = 0,\dots,N-1$$

See Matlab functions `fft` and `ifft`. You can get an N-point DFT matrix in Matlab with the following code:

```
N = 3;
x = eye(N);
G = fft(x);
```

Note the FFT computes the same result as the DFT but has less computational complexity.

DFT Intuition

Look at our $N = 3$ example again:

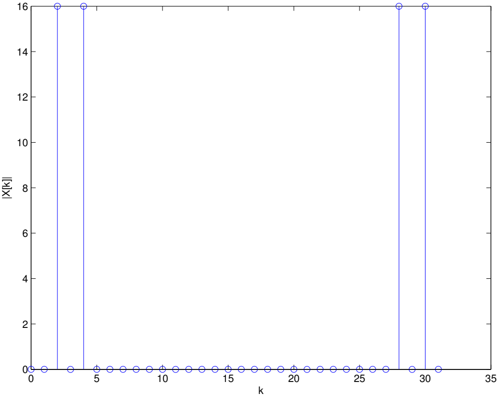
$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & e^{-j2\pi/3} & e^{-j4\pi/3} \\ 1 & e^{-j4\pi/3} & e^{-j8\pi/3} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \end{bmatrix}$$

Intuition:

- Each row of the DFT matrix is a sampled complex exponential at a specific frequency.
- The matrix-vector product is a inner product between the time domain sequence $\{x[n]\}$ and each row of the DFT matrix.
- $X[k]$ is a measure of how much $2\pi k/N$-frequency component is present in $\{x[n]\}$.
- Each row of the DFT matrix is orthogonal to the other rows.
- The DFT operation can be thought of as a change of basis.

Interpretation of the DFT Frequency Axis

Suppose you plot the DFT magnitude of an $N = 32$ point signal via `plot(0:31,abs(fft(x)))` and see the following result:



What frequencies are present in the signal? If the signal was sampled at $f_s = 44100$ Hz, what frequencies are present in the original analog signal?