Lecture 15 - Discrete Fourier Transform (DFT) ($\neq$ DTFT)

Recall

$$\begin{array}{ccc} x_c(t) & \xleftrightarrow{\text{CTFT}} & X_c(\Omega) \\ \big\downarrow \text{sampling} & & \big\downarrow \; X_d(\omega) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left(\frac{\omega + 2k\pi}{T}\right) \\ x[n] = x_c(nT) & \xleftrightarrow{\text{DTFT}} & \underbrace{X_d(\omega)}_{2\pi\text{-periodic}} = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n} \end{array}$$

Practical problems of the DTFT:

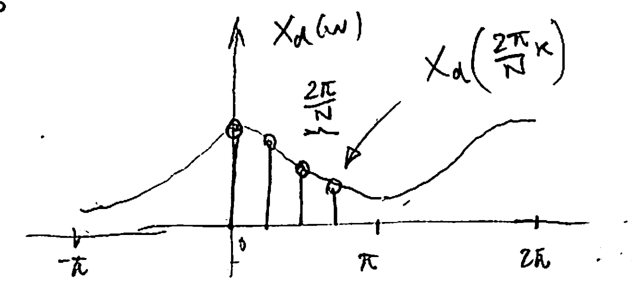
1. Infinite sum, depends on infinitely many samples
2. $X_d(\omega)$ is a continuous function of ω

Idea of DFT is to sample frequencies from DTFT

Discrete Fourier Transform (DFT)

$$\{x[n]\}_{n=0}^{N-1} \xleftrightarrow{DFT_N} \{X[k]\}_{k=0}^{N-1} \qquad k=0,1,\dots,N-1$$

DFT can be thought of as sampling DTFT.



~~Suppose~~ $\{x[n]\}_{n=-\infty}^{\infty}$ is an infinite sequence.

Suppose $x[n]=0$ for $n<0$ and $n \geq N$. In this case

$$\text{DTFT}: \quad X_d(\omega) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n} = \sum_{n=0}^{N-1} x[n] e^{-j\omega n} \quad \text{and} \quad X[k] = X_d(\omega)\Big|_{\omega=\frac{2\pi}{N}\cdot k}$$

$$\boxed{\text{DFT}: \quad X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j\left(\frac{2\pi}{N}k\right)n}}$$

$$= \sum_{n=0}^{N-1} x[n]\, W_N^{kn}$$

where $W_N = \underbrace{e^{-j\frac{2\pi}{N}}}_{\text{constant}}$

Ex: $N = 2$. $\{x[0], x[1]\} \xrightarrow{DFT_2} \{\underbrace{X[0], X[1]}_{\text{DFT coefficients}}\}$

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi}{N}kn}$$

$N = 2$:

$k=0$: $X[0] = x[0]\, W_2^{0\cdot 0} + x[1]\, W_2^{0\cdot 1}$

$k=1$: $X[1] = x[0]\, W_2^{1\cdot 0} + x[1]\, W_2^{1\cdot 1}$

or
$$\begin{bmatrix} X[0] \\ X[1] \end{bmatrix} = \begin{bmatrix} W_2^{0\cdot 0} & W_2^{0\cdot 1} \\ W_2^{1\cdot 0} & W_2^{1\cdot 1} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \end{bmatrix}$$

$W_2^0 = 1$

$W_2^1 = e^{-j\frac{2\pi}{2}\cdot 1} = -1$

$$\Rightarrow \begin{bmatrix} X[0] \\ X[1] \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}}_{\mathbb{W}_2} \begin{bmatrix} x[0] \\ x[1] \end{bmatrix} \Longrightarrow \begin{bmatrix} x[0] \\ x[1] \end{bmatrix} = \mathbb{W}_2^{-1} \begin{bmatrix} X[0] \\ X[1] \end{bmatrix}$$

$$\mathbb{W}_2^{-1} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

In general, $\mathbb{W}_N^{-1} = \frac{1}{N}\mathbb{W}_N^{*}$ (for $N=2$, $\mathbb{W}_2$ is real-valued, but not in general)

Ex : $N = 3$

$$W_3 = \begin{bmatrix} W_3^{0\cdot 0} & W_3^{0\cdot 1} & W_3^{0\cdot 2} \\ W_3^{1\cdot 0} & W_3^{1\cdot 1} & W_3^{1\cdot 2} \\ W_3^{2\cdot 0} & W_3^{2\cdot 1} & W_3^{2\cdot 2} \end{bmatrix} = \overbrace{\begin{bmatrix} 1 & 1 & 1 \\ 1 & W_3 & W_3^{2} \\ 1 & W_3^{2} & W_3^{4} \end{bmatrix}}^{\text{Symmetric matrix}} \Rightarrow W_3^{-1} = \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 1 & W_3^{-1} & W_3^{-2} \\ 1 & W_3^{-2} & W_3^{-4} \end{bmatrix}$$

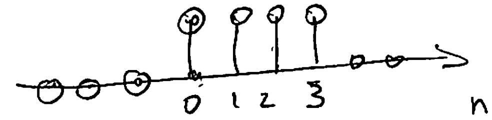
Inverse DFT_N :

$$\boxed{x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, e^{j\frac{2\pi}{N}kn}}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, W_N^{-kn}$$

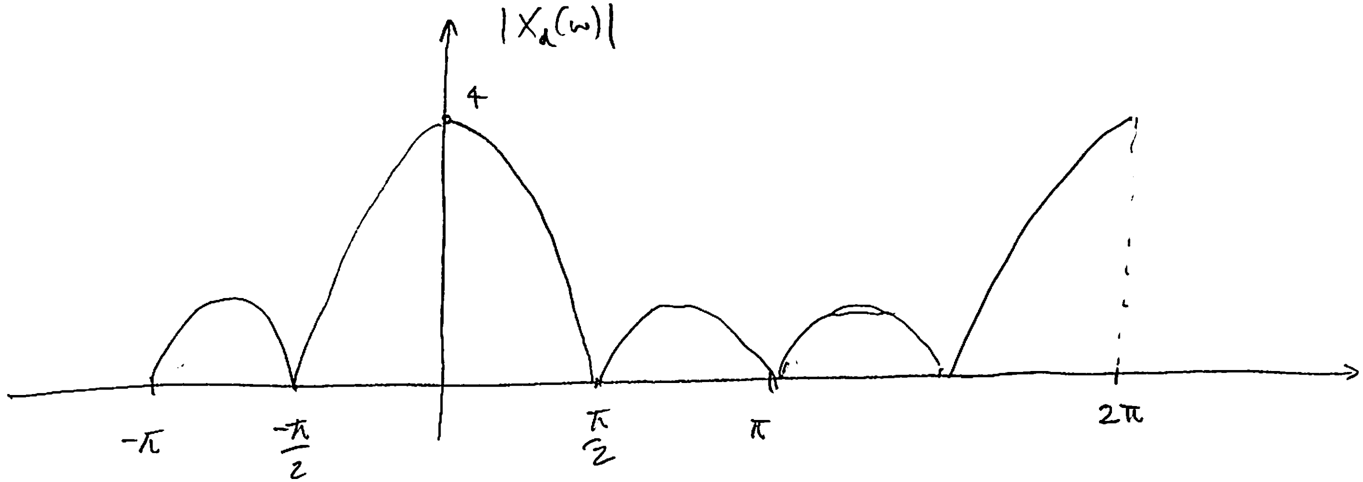
Example for MATLAB Demo

$$x[n] = \delta[n] + \delta[n-1] + \delta[n-2] + \delta[n-3]$$



DTFT. $$X_d(w) = \sum_{n=-\infty}^{\infty} x[n] e^{-jwn} = 1 + e^{-jw} + e^{-j2w} + e^{-j3w} = \frac{1 - e^{-j4w}}{1 - e^{-jw}}$$

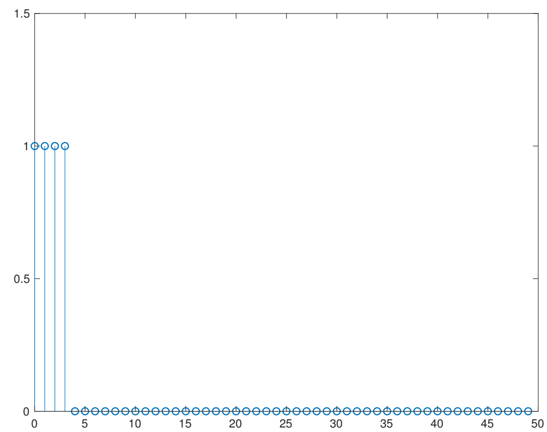
$$= e^{-j\frac{3w}{2}} \frac{\sin(2w)}{\sin(w/2)}$$



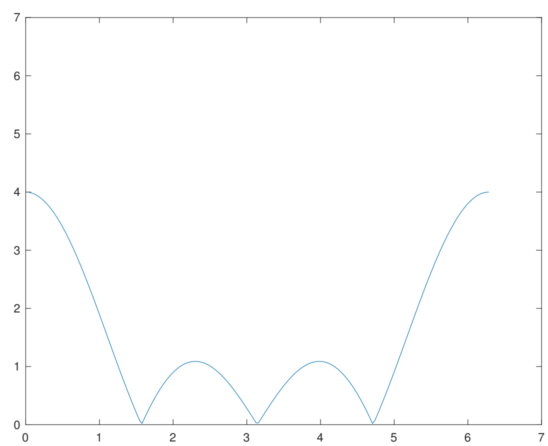
Let's implement the DFT of $x[n]$

```matlab
function X = DFT(x)
% DFT Compute the Discrete-Time Fourier Transform
% Input: x (discrete-time signal in an array)
% Output: X (DFT coefficients)

N = length(x);
W_N = exp(-j*2*pi/N);

% For each frequency index k
for k = 0:(N-1)
    % Compute DFT sum over signal samples
    X(k+1) = 0;
    for n = 0:(N-1)
        X(k+1) = X(k+1) + x(n+1) * W_N^(k*n);
    end
end
```

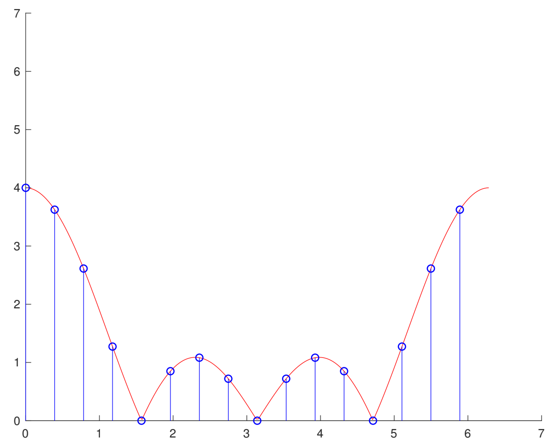
```
x0 = zeros(1,50);

x0(1:4) = [1,1,1,1];

figure(1)
stem(0:length(x0)-1,x0);
ylim([0,1.5]);
```




```
% We know what the DTFT of x[n] should look like:

w_dtft = linspace(0,2*pi,200);

X_d = exp(-j*3/2*w_dtft).*(sin(2*w_dtft))./sin(w_dtft/2);

figure(2)
plot(w_dtft,abs(X_d));
ylim([0,7]);
```



```
% To compute an N-point DFT, we pick a segment of length N of x

N = 16;

x = x0(1:N);

X = DFT(x);

% points where we are "sampling" the DTFT
w_k = (0:N-1)*2*pi/N;

figure(3);
hold off;
hold on;
plot(w_dtft,abs(X_d),'r');
stem( w_k, abs(X) ,'b');
ylim([0,7]);
xlim([0,7])
hold off;
```



Suppose $x[n]$ is an infinite sequence

Take N samples $x[0], \ldots, x[N-1]$ and take DFT.

The DFT is taking samples from what DTFT?

Let $y[n] = \begin{cases} x[n] & \text{for } n = 0, \ldots, N-1 \\ 0 & \text{otherwise} \end{cases}$

$\text{DFT}_N((x[0], \ldots, x[N-1]))$ is sampling the DTFT of $y[n]$.

If $x[n] \approx 0$ for $n < 0$ and $n \geq N$, then $Y_d(\omega)$

and $X_d(\omega)$ should be close

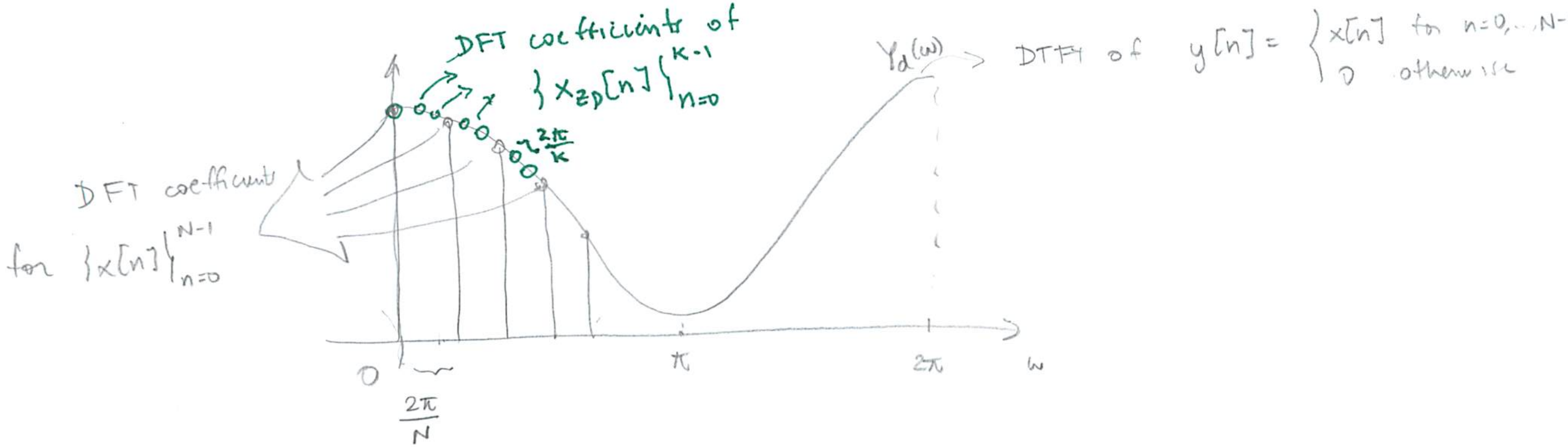
Suppose we only have N samples $x[0], \ldots, x[N-1]$,

but we want to get $K > N$ DFT coefficients.

<u>zero-padding</u> We simply pad $\{x[0], \ldots, x[N-1]\}$ with zeros.

If we want K DFT coefficients, add $K-N$ zeros.

$$x_{zp}[n] = \{x[0], x[1], \ldots, x[N-1], \overbrace{0, 0, \ldots, 0}^{K-N}\}$$



$$X[k] = Y_d\left(k \frac{2\pi}{N}\right) \qquad X_{zp}[k] = Y_d\left(\frac{k\,2\pi}{K}\right)$$

$$X[k_1] = X_{zp}[k_2] \quad \text{if} \quad \frac{k_1}{N} = \frac{k_2}{K}$$