Discrete Time Fourier Transformation (DTFT)

Prof. Seungchul Lee

Industrial AI Lab.

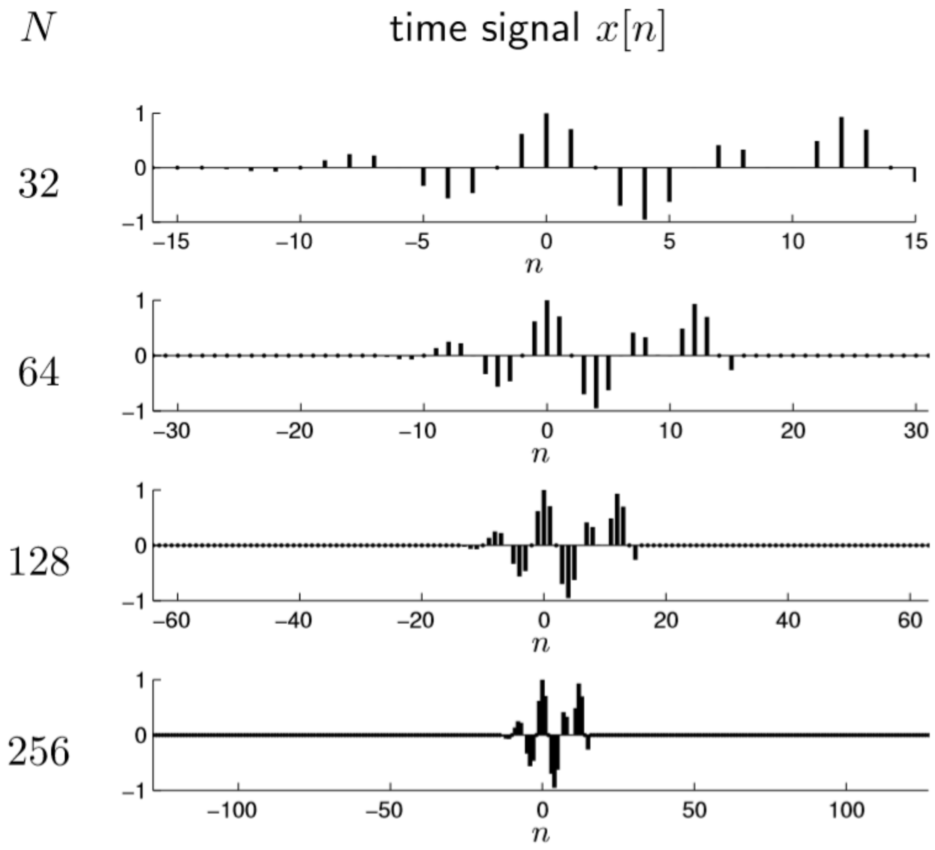
From DFT to DTFT

- DTFT is the Fourier transform of choice for analyzing infinite-length discrete signals and systems

- Useful for conceptual, but not MATLAB friendly (infinite-long vectors)

- We will derive DTFT as the limit of the DFT as the signal length $N \to \infty$

$$\omega = \frac{2\pi}{N}k$$



The Centered DFT

- Both $x[n]$ and $X[k]$ can be interpreted as periodic with period N, we will shift the intervals of interest in time and frequency to be centered around $n, k = 0$

$$-\frac{N}{2} \leq n, k \leq \frac{N}{2} - 1$$

- The modified forward and inverse DFT formulas are

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

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

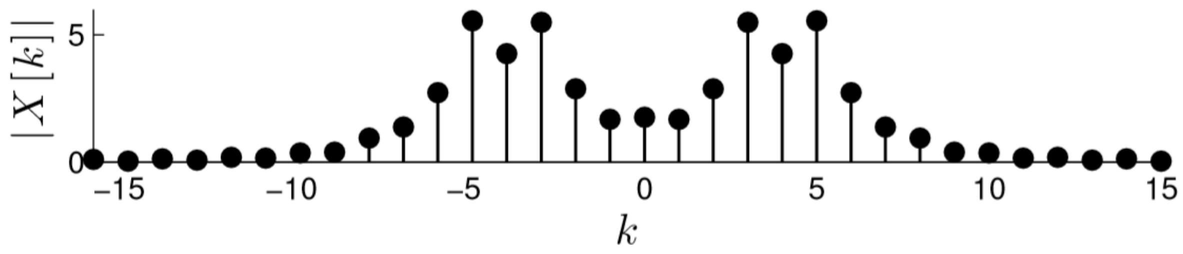
Take It To The Limit

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

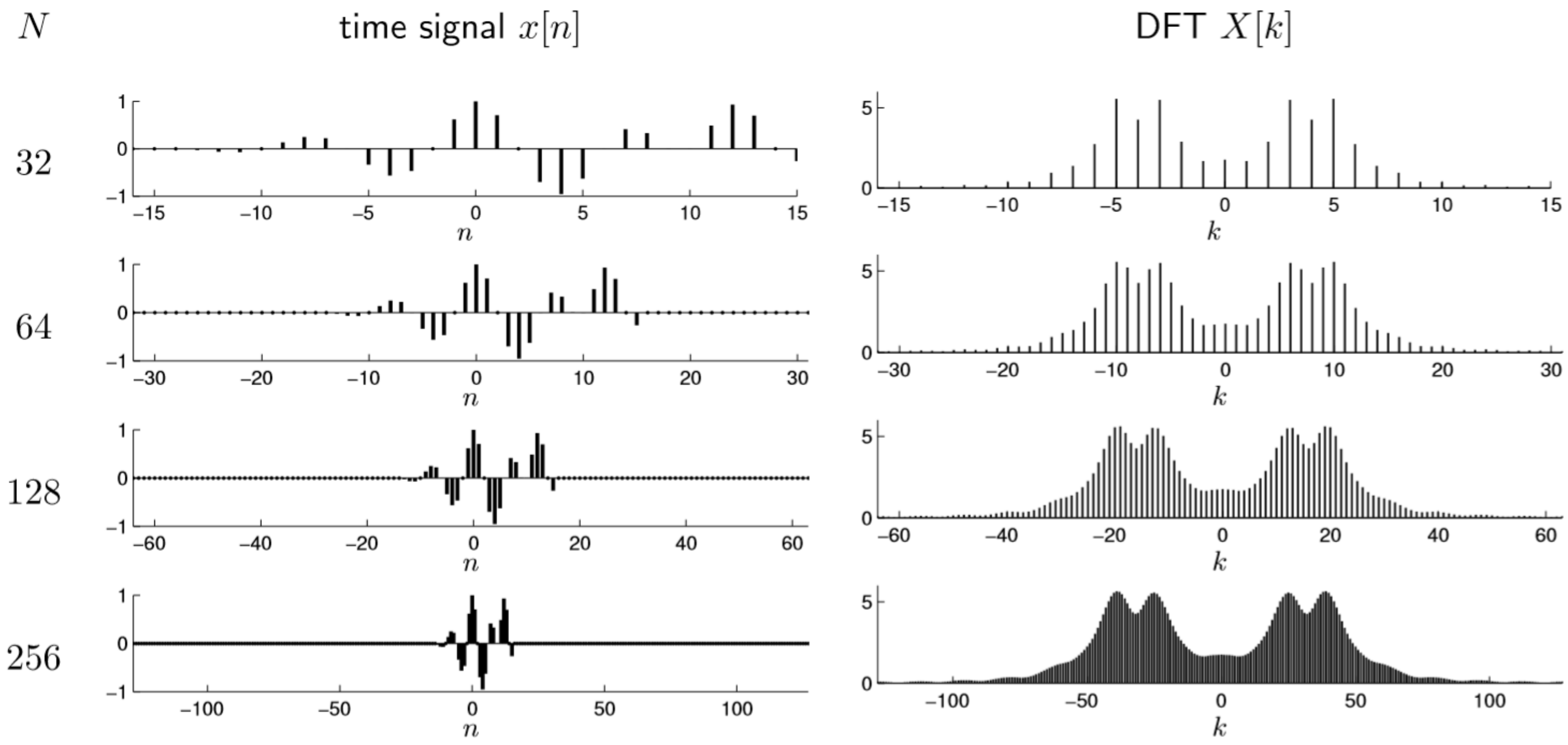
- Let the signal length N increase towards ∞ and study what happens to $X_u[k]$

- Key fact: No matter how large N grows, the frequencies of the DFT sinusoids remain in the interval

$$-\pi \le \omega_k = \frac{2\pi}{N}k \le \pi$$



Take It To The Limit



Discrete Time Fourier Transform (Forward)

- As $N \to \infty$, the forward DFT converges to a function of the continuous frequency variable ω that we will call the forward discrete time Fourier transform (DTFT)

$$\sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} x[n]\, e^{-j\frac{2\pi}{N}kn} \quad \to \quad \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n} = X(\omega), \qquad -\pi \le \omega \le \pi$$

- Recall: inner product for infinite-length signals

$$\langle x, y\rangle = \sum_{n=-\infty}^{\infty} x[n] y[n]^*$$

- Analysis interpretation: the value of the DFFT $X(\omega)$ at frequency ω measures the similarity of the infinite-length signal $x[n]$ to the infinite-length sinusoid $e^{j\omega n}$

Discrete Time Fourier Transform (Inverse)

- Inverse unnormalized DFT

$$x[n] = \frac{1}{N} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} X[k]\, e^{j\frac{2\pi}{N}kn} \quad \rightarrow \quad x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega) e^{j\omega n}\, d\omega$$

- Synthesis interpretation: Build up the signal x as an infinite linear combination of sinusoids $e^{j\omega n}$ weighted by the DTFT $X(\omega)$

Discrete Time Fourier Transform: Summary

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}, \qquad -\pi \leq \omega < \pi$$

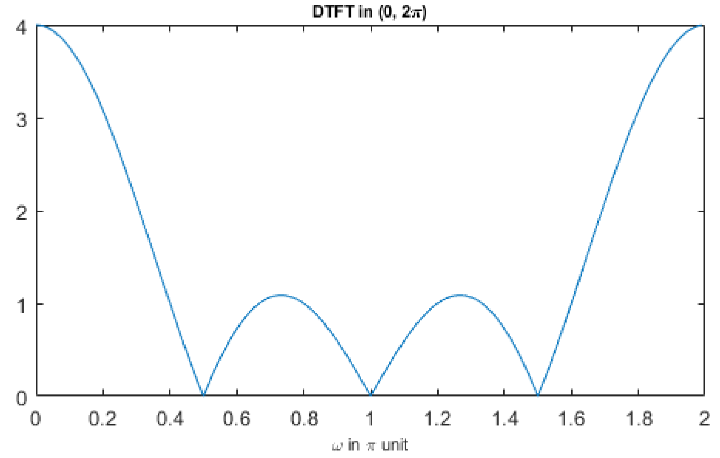
$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega)e^{j\omega n}\, d\omega, \qquad -\infty < n < \infty$$

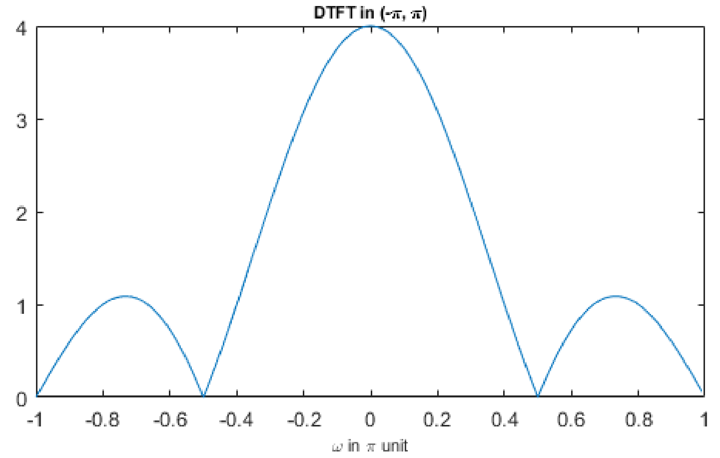
DTFT in MATLAB

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}, \qquad -\pi \leq \omega < \pi$$

```
% dtft from definition
n = 0:3;
x = [1 1 1 1];

N = 200;

w = [0:N-1]*2*pi/N;
Xdtft = 1*exp(-1j*w*0) + 1*exp(-1j*w*1) + 1*exp(-1j*w*2) + 1*exp(-1j*w*3);

plot(w/pi, abs(Xdtft))
xlabel('\omega in \pi unit', 'fontsize', 8),
title('DTFT in (0, 2\pi)', 'fontsize', 8)
```

```
k = [0:N/2-1 -N/2:-1];
w = k*2*pi/N;
ws = fftshift(w);
Xdtfts = fftshift(Xdtft);
```





DFT Zero Padding

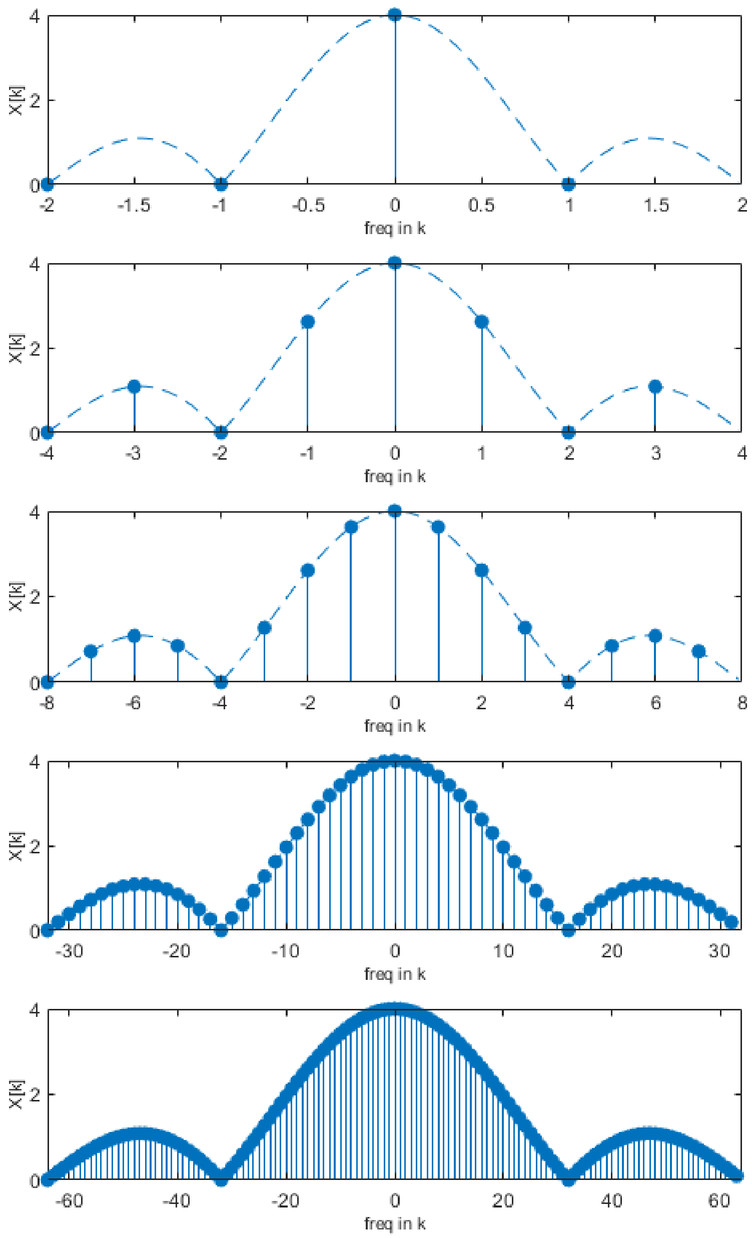
```
x = [1,1,1,1];
N = length(x);
```

```
x = [1,1,1,1,zeros(1,4)];
N = length(x);
```

```
x = [1,1,1,1,zeros(1,12)];
N = length(x);
```

```
x = [1,1,1,1,zeros(1,2^6-4)];
N = length(x);
```

```
x = [1,1,1,1,zeros(1,2^7-4)];
N = length(x);
```



DTFT in MATLAB

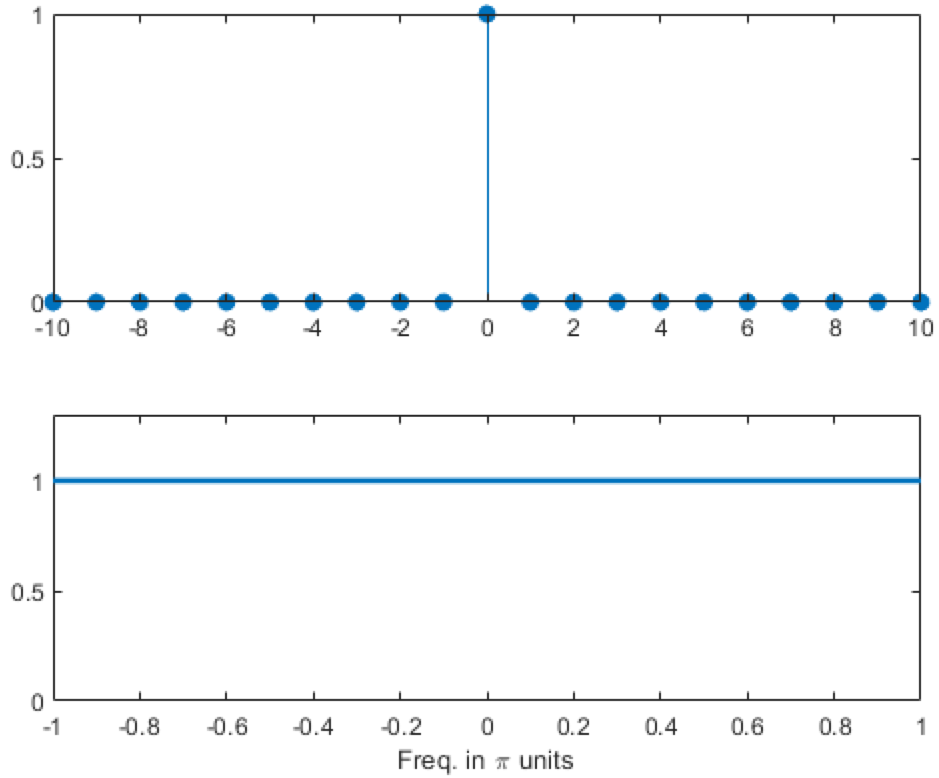
```
function X = dtft(x,n,w)
% X = dtft(x, n, w)
%
% X = DTFT values computed at w frequencies
% x = finite duration sequence over n (row vector)
% n = sample position vector
% w = frequency location vector (row vector)

X = exp(-1j*(w'*n))*x';

end
```

DTFT of the Impulse

- Fact: the impulse signal contains all the frequency components with 1

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

```
[x, n] = impseq(0, -10, 10);
w = linspace(-1,1,2^10)*pi;

X = dtft(x,n,w);
```



DTFT of $e^{j\omega_0 n}$

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n}, \qquad -\pi \le \omega < \pi$$

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega) e^{j\omega n}\, d\omega, \qquad -\infty < n < \infty$$

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} 2\pi\delta(\omega - \omega_0) e^{j\omega n}\, d\omega = e^{j\omega_0 n}$$

$$e^{j\omega_0 n} \quad \longleftrightarrow \quad 2\pi\delta(\omega - \omega_0)$$

DTFT of The Unit Pulse

$$p[n] = \begin{cases} 1 & -M \le n \le M \\ 0 & \text{otherwise} \end{cases}$$

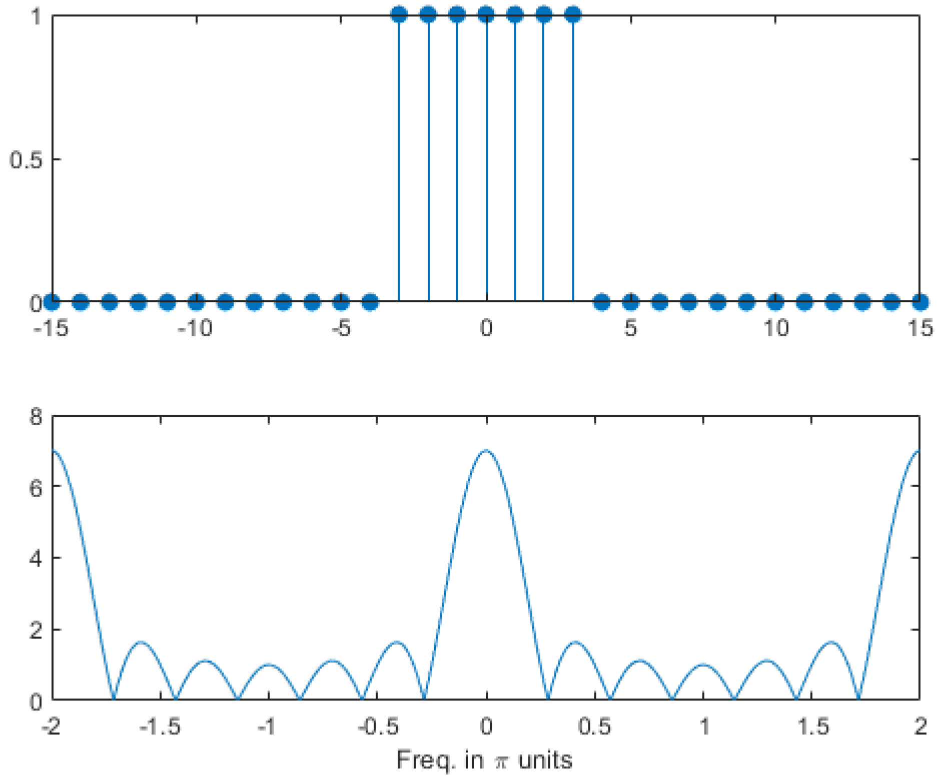
$$P(\omega) = \sum_{n=-\infty}^{\infty} p[n]\, e^{-j\omega n} = \sum_{n=-M}^{M} e^{-j\omega n} = \frac{e^{j\omega M} - e^{-j\omega(M+1)}}{1 - e^{-j\omega}}$$

$$= \frac{e^{-j\omega/2}\left(e^{j\omega\frac{2M+1}{2}} - e^{-j\omega\frac{2M+1}{2}}\right)}{e^{-j\omega/2}\left(e^{j\omega/2} - e^{-j\omega/2}\right)} = \frac{2j\ \sin\left(\omega\frac{2M+1}{2}\right)}{2j\ \sin\left(\frac{\omega}{2}\right)}$$

DTFT of The Unit Pulse

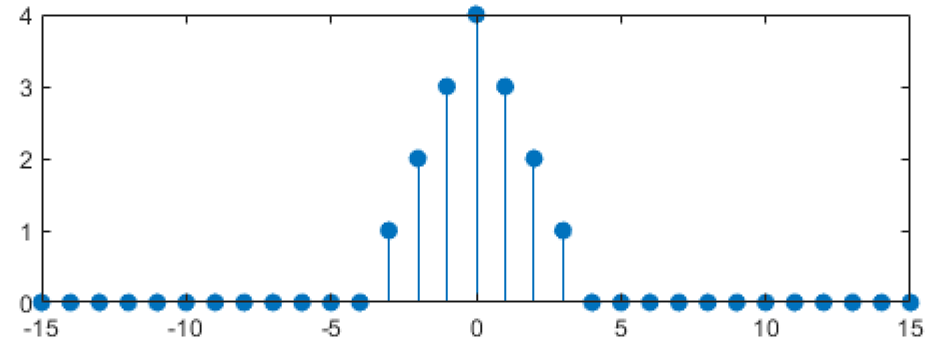
$$p[n] = \begin{cases} 1 & -M \le n \le M \\ 0 & \text{otherwise} \end{cases}$$

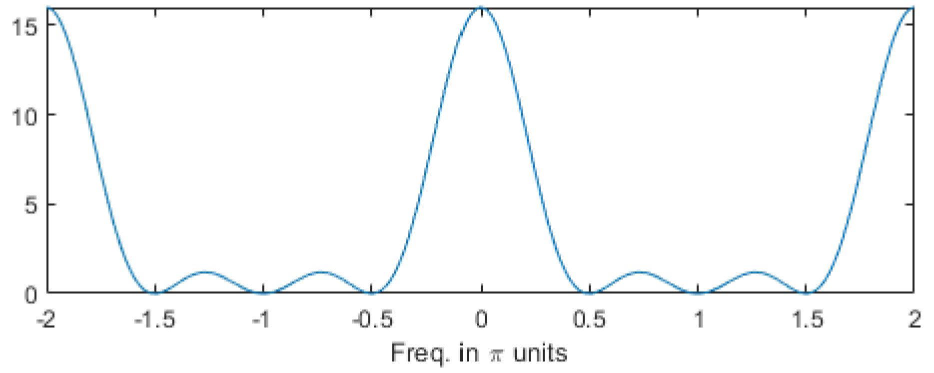
```
x = [1 1 1 1 1 1 1];
n = -3:3;
w = linspace(-2,2,2^10)*pi;

X = dtft(x,n,w);
```



DTFT of Triangle

```
x = [1 2 3 4 3 2 1];
n = -3:3;
w = linspace(-2,2,2^10)*pi;

X = dtft(x,n,w);
```

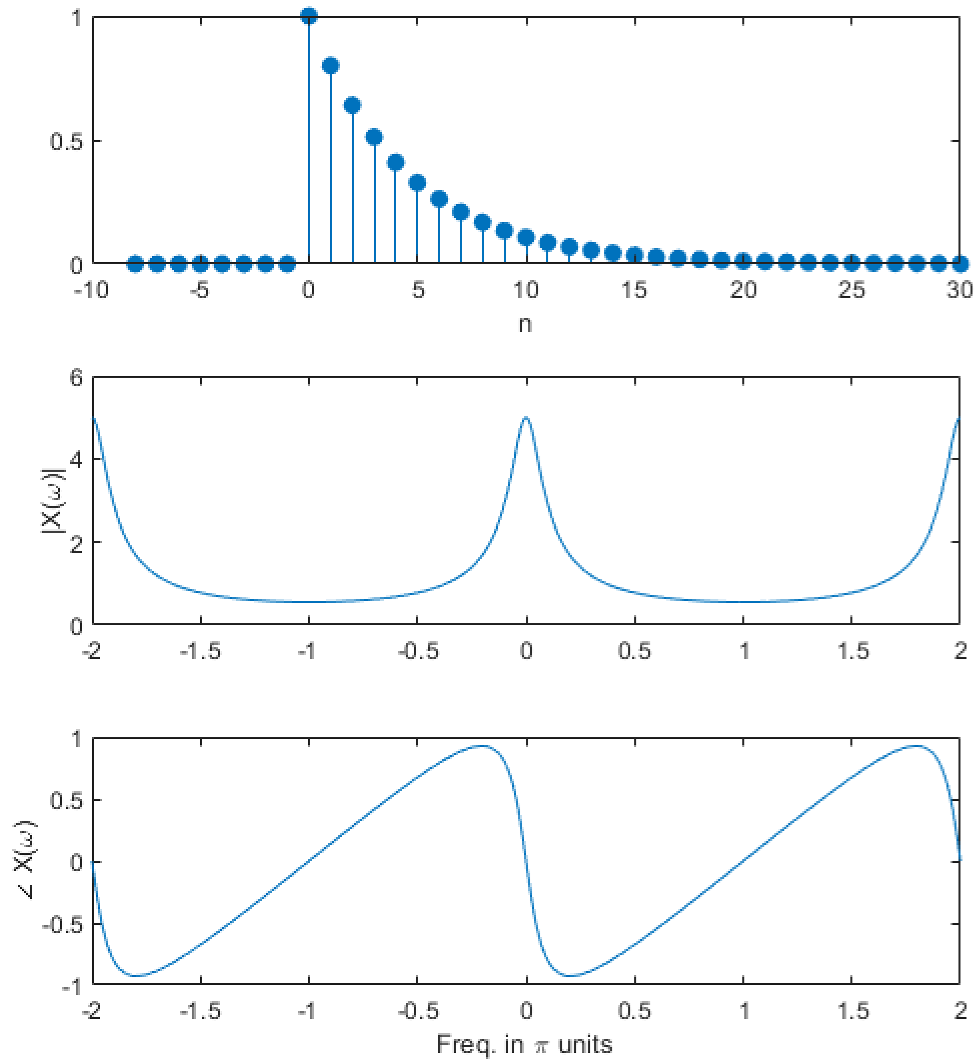




DTFT of a One-sided Exponential

$$h[n] = \alpha^n u[n] \qquad \longleftrightarrow \qquad H(\omega) = \frac{1}{1 - \alpha e^{-j\omega}}$$

```
N = 30;

x = zeros(1,N);
for i = 1:N
    x(i) = 0.8^(i-1);
end

n = 0:N-1;
w = linspace(-2,2,2^10)*pi;

X = dtft(x,n,w);
```

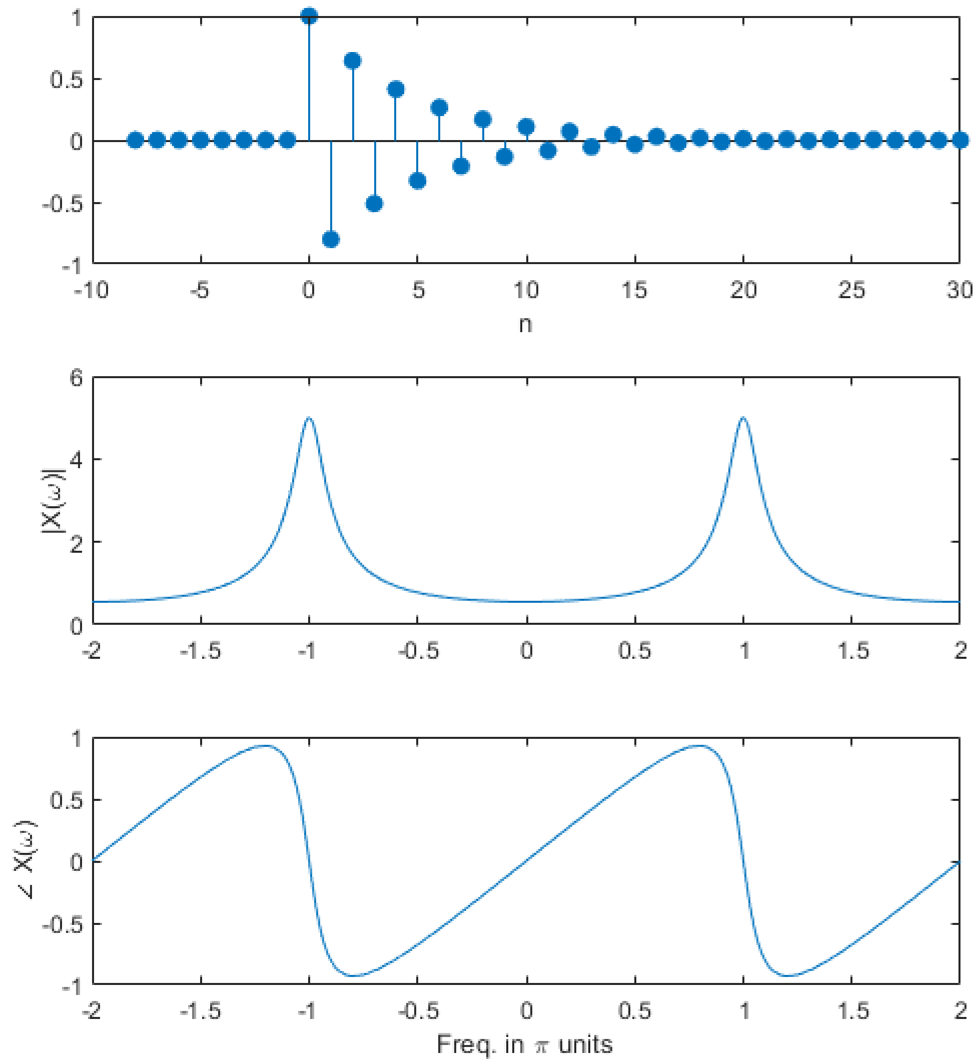


Property: Modulation

$$e^{j\omega_0 n}x[n] \qquad \longleftrightarrow \qquad X(\omega - \omega_0)$$

$$e^{j\omega_0 n}x[n] = (-1)^n x[n] \qquad \text{when} \quad \omega_0 = \frac{2\pi}{N}\frac{N}{2} = \pi$$

```
N = 30;
nd = -8:N;
xd = zeros(size(nd));

x = zeros(1,N);
for i = 1:N
    x(i) = (-0.8)^(i-1);
end
n = 0:N-1;
%w = linspace(-1,1,2^10)*pi;
w = linspace(-2,2,2^10)*pi;

X = dtft(x,n,w);
```

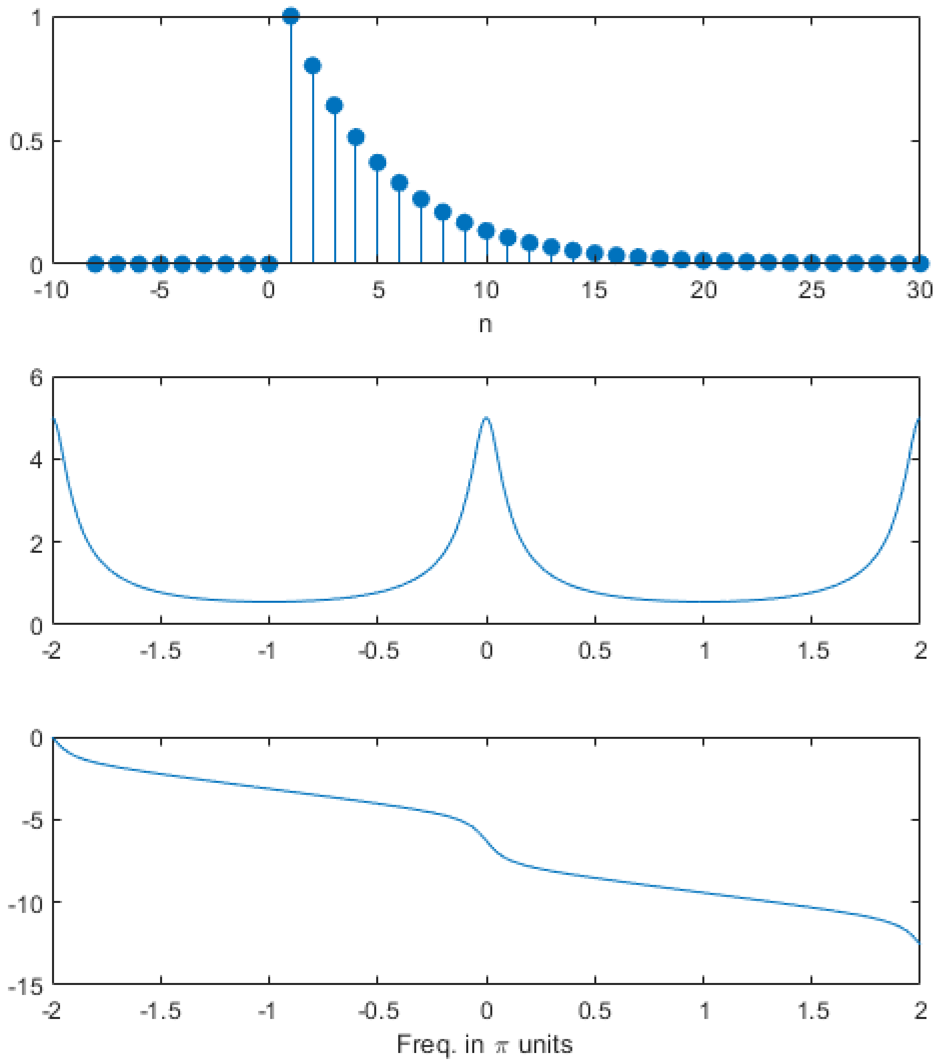


Property: Time Shift

$$x[n-m] \quad \longleftrightarrow \quad e^{-j\omega m}X(\omega)$$

- Same amplitude
- Phase changed (linearly $-\angle\omega m$)

```
N = 30;
nd = -8:N;
xd = zeros(size(nd));

x = zeros(1,N);
for i = 1:N
    x(i) = 0.8^(i-1);
end

m = 1;
n = 0+m:N-1+m;
[y,ny] = sigadd(xd,nd,x,n);

w = linspace(-2,2,2^10)*pi;

X = dtft(x,n,w);
```



DTFT and Convolution

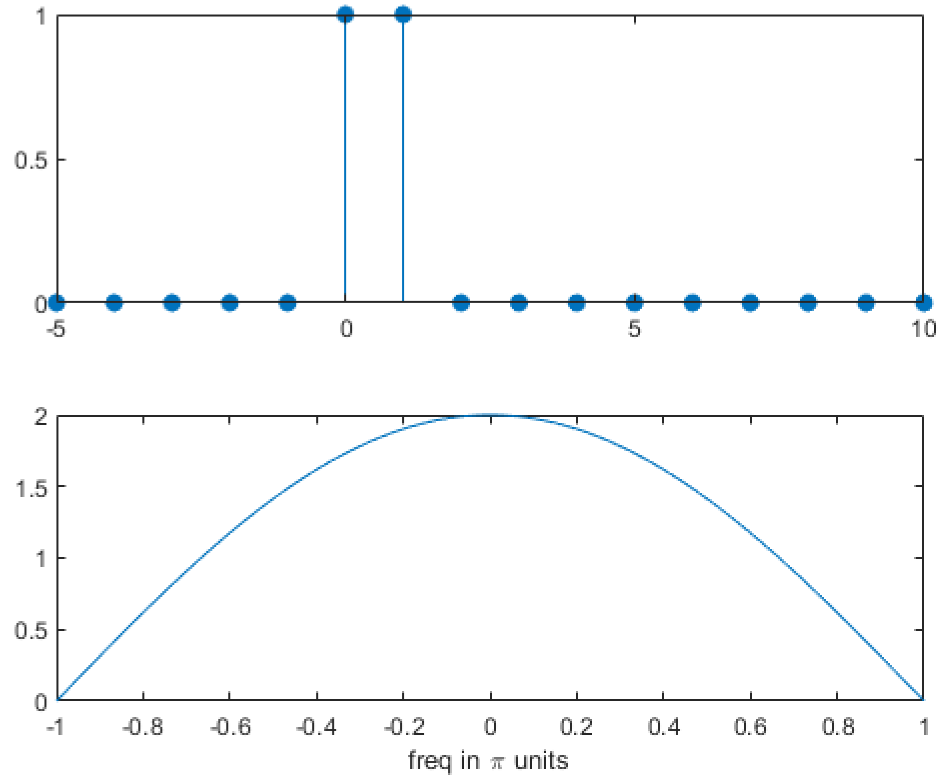
- Convolution in time domain = multiplication in frequency domain

$$\begin{aligned} x[n] &\quad\longleftrightarrow\quad X(\omega) \\ h[n] &\quad\longleftrightarrow\quad H(\omega) \end{aligned}$$

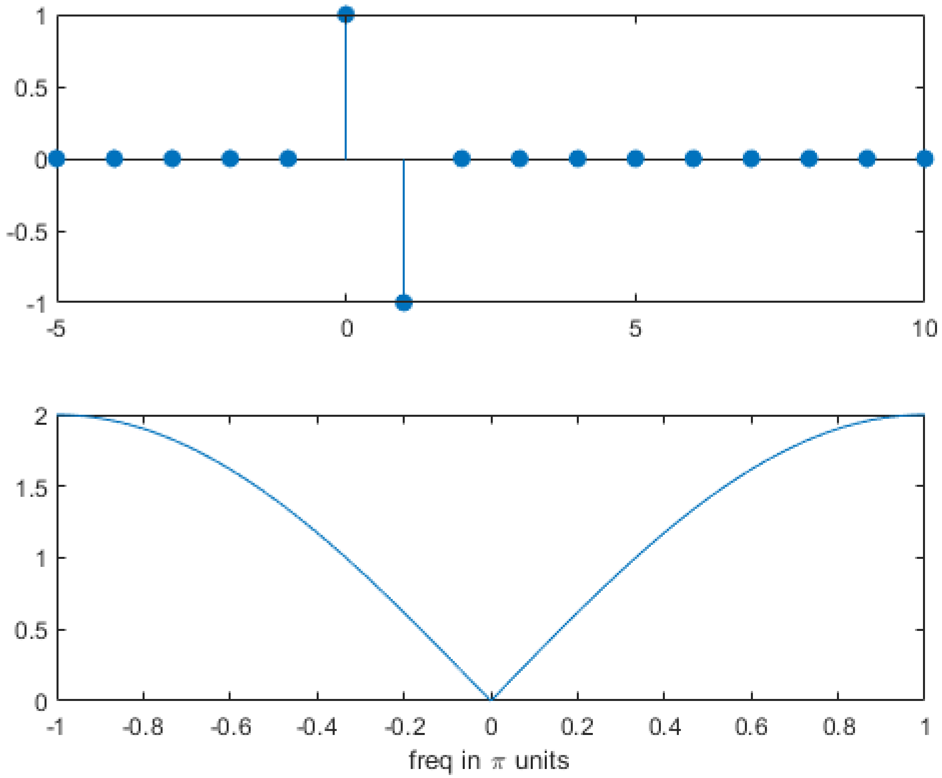
$$y[n] = x[n] * h[n] \quad\longleftrightarrow\quad H(\omega)X(\omega)$$

Filters

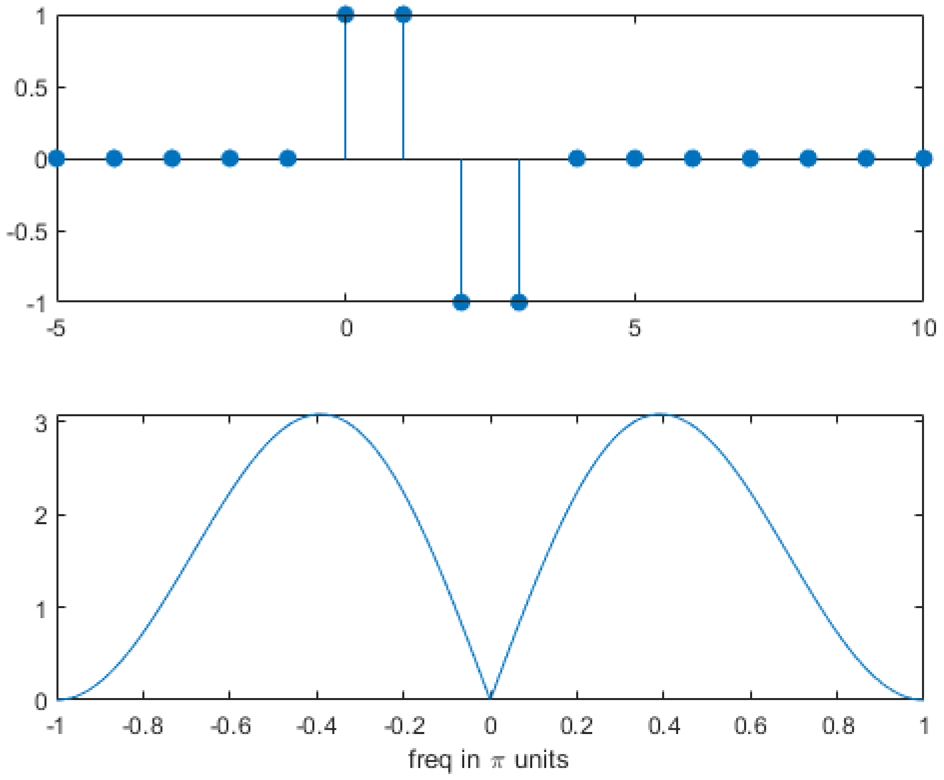
Filters: Low-Pass



Filters: High-Pass



Filters: Band-Pass



Ideal Low-pass Filters

- Impulse Response of the Ideal Low-pass Filter
- The frequency response $H(\omega)$ of the ideal low-pass filter passes low frequencies (near $\omega = 0$), but blocks high frequencies (near $\omega = \pm\pi$)

$$H(\omega) = \begin{cases} 1 & -\omega_c \leq \omega \leq \omega_c \\ 0 & \text{otherwise} \end{cases}$$

- Compute the impulse response $h[n]$ given this $H(\omega)$

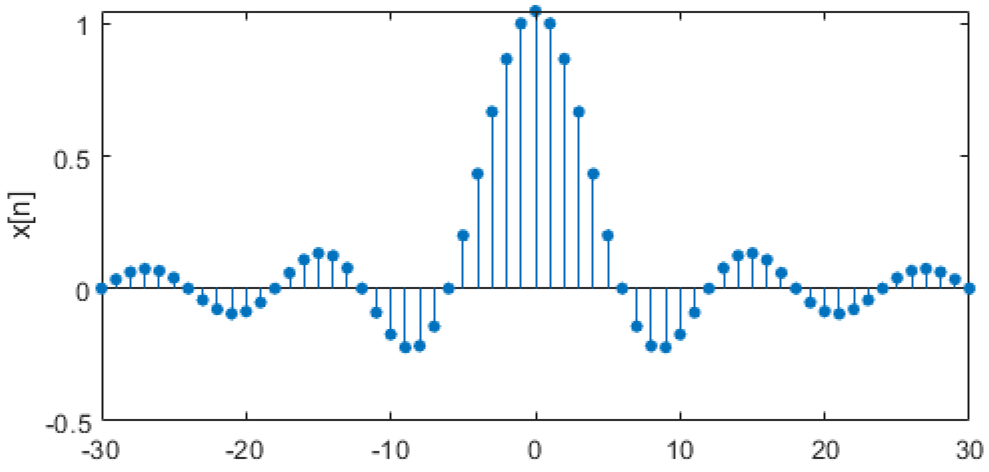
$$h[n] = \int_{-\pi}^{\pi} H(\omega)\, e^{j\omega n} \frac{d\omega}{2\pi} = \int_{-\omega_c}^{\omega_c} e^{j\omega n} \frac{d\omega}{2\pi} = \frac{e^{j\omega_c n} - e^{-j\omega_c n}}{jn} = 2\omega_c \frac{\sin(\omega_c n)}{\omega_c n}$$

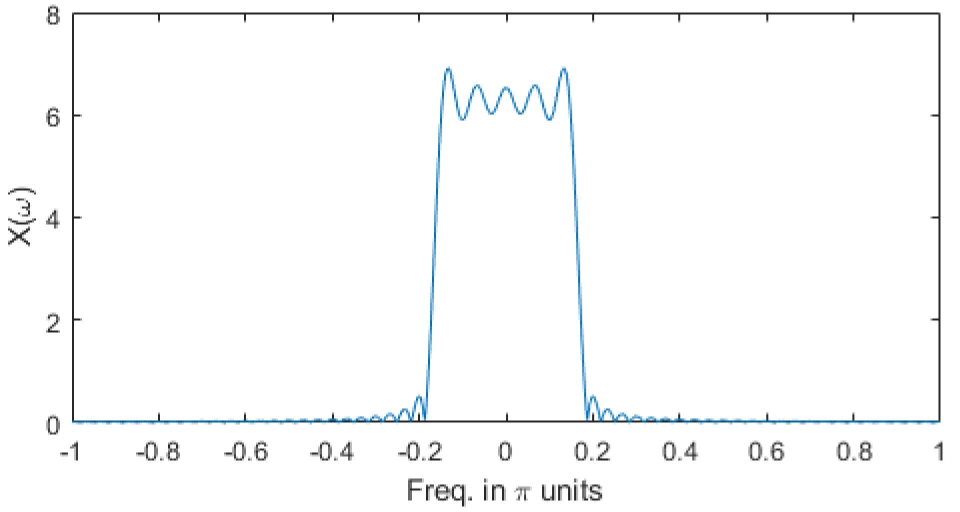
POSTECH

Ideal Low-Pass Filter in MATLAB

$$h[n] = 2\omega_c \frac{\sin(\omega_c n)}{\omega_c n} \qquad \mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$$

```
wc = pi/6;

N = 30;
n = -N:N;

h = zeros(1,length(n));
for i = 1:length(n)
    h(i) = 2*wc*sinc(1/pi*wc*(i-N-1));
end

w = linspace(-1,1,2^10)*pi;

X = dtft(h,n,w);
```

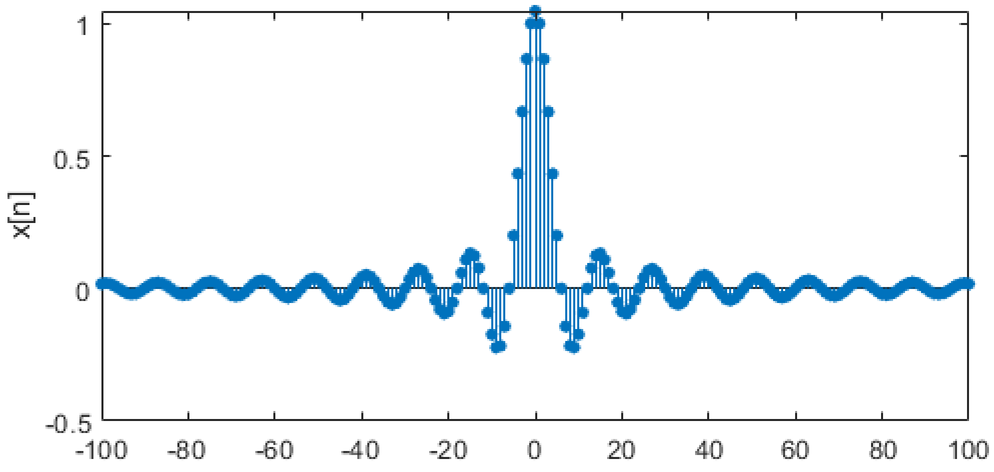


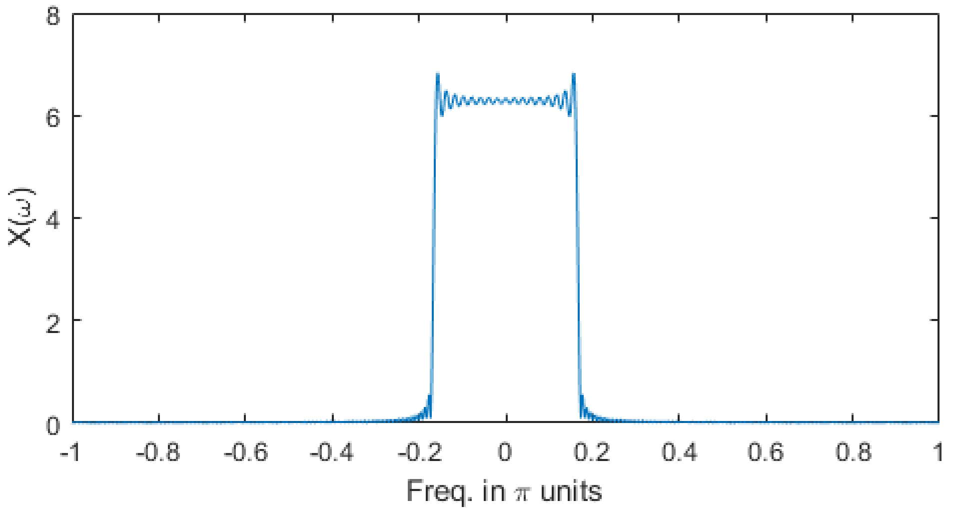


Ideal Low-Pass Filter in MATLAB

$$h[n] = 2\omega_c \frac{\sin(\omega_c n)}{\omega_c n} \qquad \mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$$

```
wc = pi/6;

N = 100;
n = -N:N;

h = zeros(1,length(n));
for i = 1:length(n)
    h(i) = 2*wc*sinc(1/pi*wc*(i-N-1));
end

w = linspace(-1,1,2^10)*pi;

X = dtft(h,n,w);
```





Ideal High-Pass Filter in MATLAB

```
wc = pi/6;

N = 30;
n = -N:N;

d = zeros(1,length(n));
h = zeros(1,length(n));
d(N+1) = d(N+1) + 1;

for i = 1:length(n)
    h(i) =  (-1)^(i-N-1)*2*wc*sinc(1/pi*wc*(i-N-1));
end

w = linspace(-1,1,2^10)*pi;

X = dtft(h,n,w);
```