# 數位信號處理實習

# Digital Signal Processing Lab.


蔡偉和

whtsai@ntut.edu.tw

02-27712171 ext. 2257

0933-052581

綜科館 311-4


**Lecture Material:**

- 北科 i 學園 PLUS

  Matlab® 可於「北科軟體雲」上操作

**Grading:**

- Final Exam: 100%

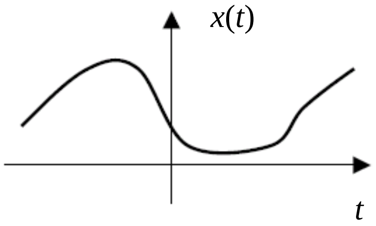
**Syllabus:**

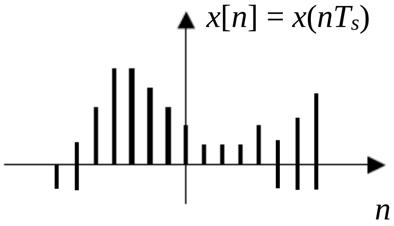
- Lab 1: Sketching Signals
- Lab 2: Convolution
- Lab 3: DTFT, DFT, and FFT
- Lab 4: Digital Systems
- Lab 5: Audio Signal Processing
- Lab 6: Filter Design
- Digital Signal Processors

# Lab 1: Sketching Signals

## [Theoretical Background]

- **Discrete-time Signals (Sequences):** A discrete-time signal $x[n]$ is obtained by sampling a continuous-time signal $x(t)$. For example:

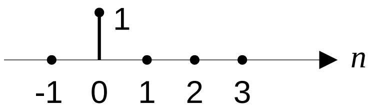
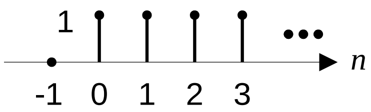
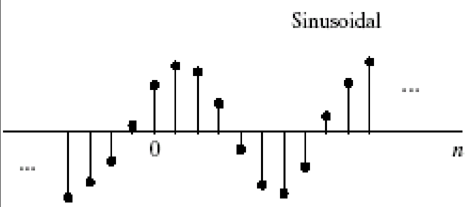
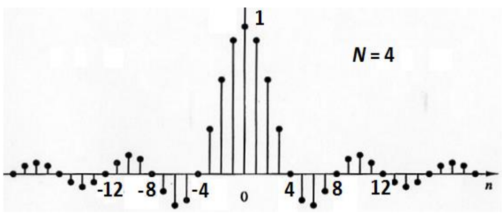
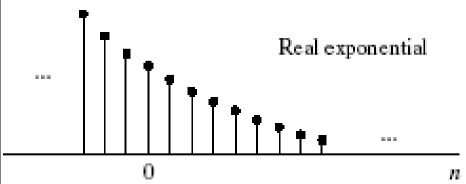



$T_s$ is the sampling period.

Remark: Digital signals usually refer to the quantized discrete-time signal. In this course, we are mostly dealing with discrete-time signals with continuous values.

- **Some basic sequences**

| | |
|---|---|
| Unit impulse sequence, $\delta[n] = \begin{cases} 1, n = 0 \\ 0, n \neq 0 \end{cases}$ |   |
| Unit step sequence, $u[n] = \begin{cases} 1, n \geq 0 \\ 0, n < 0 \end{cases}$<br>Note: $u[n] = \sum_{m=0}^{\infty} \delta[n-m]$,<br>$u[n] = \sum_{m=-\infty}^{n} \delta[m]$, and<br>$\delta[n] = u[n] - u[n-1]$ |   |
| Sinusoidal Sequences, $x[n] = A\cos(w_0 n + \phi)$,<br>A: amplitude; $w_0$: frequency; $\phi$: phase<br>Note: sinusoidal sequences are not always periodic. |   |
| Sinc sequences, $x[n] = \dfrac{\sin(\pi n / N)}{\pi n / N}$ |   |
| Exponential sequences,<br>$x[n] = A\alpha^n$<br>$x[n] = A\alpha^n u[n]$ |   |

[Matlab Example 1-1] Generate a sine signal as shown in Fig. 1-1.

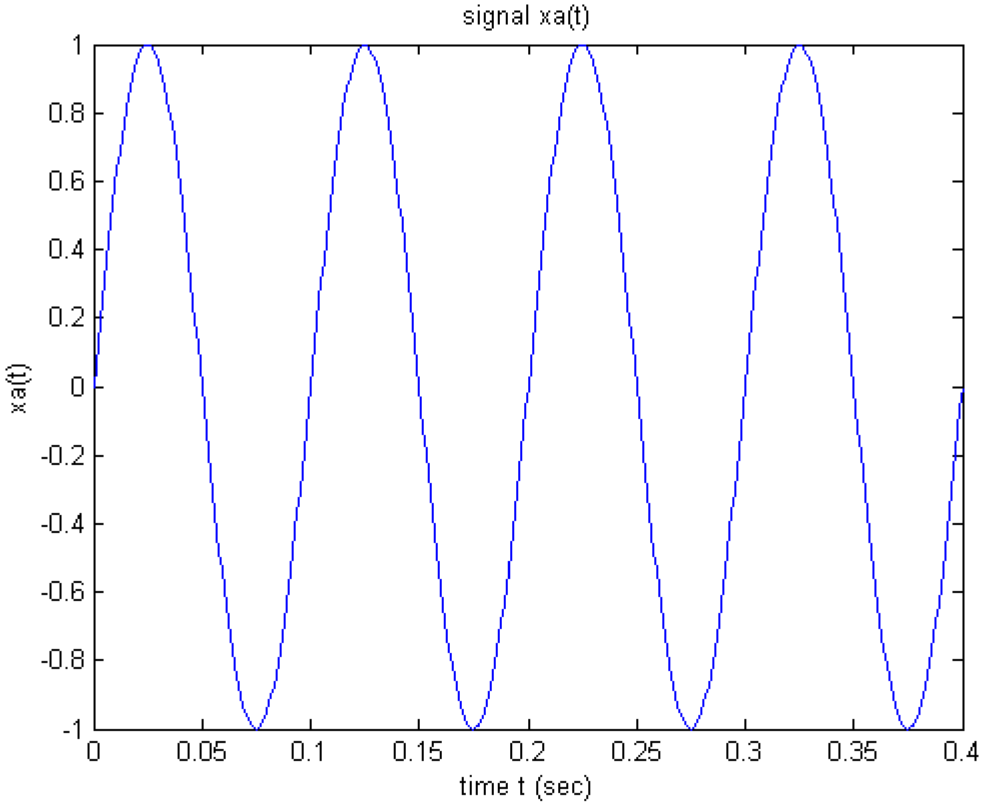



Fig. 1-1

Code Ex_1_1.m

```
% Generating a single-tone sine signal xa
clear;
f0=10;          % 10 Hz sine wave
dt=0.001;        % resolution
Length=0.4       % Total length =0.4 sec
t=0:dt:Length;
xa=sin(2*pi*f0*t);
plot(t,xa);
xlabel('time t (sec)'); ylabel('xa(t)');
title('signal xa(t)');
```

*******************************************************************

[Practice 1-1] Sketch signal $\mathrm{sinc}(t)=\dfrac{\sin \pi t}{\pi t}$ and $\mathrm{sinc}^2(t)$ for $-3 \le t \le 3$ 。

*******************************************************************

[Matlab Example1-2] Sample the analog signal in Fig. 1-1 to form the discrete-time signal in Fig. 1-2 using a sampling period of 0.01 second.

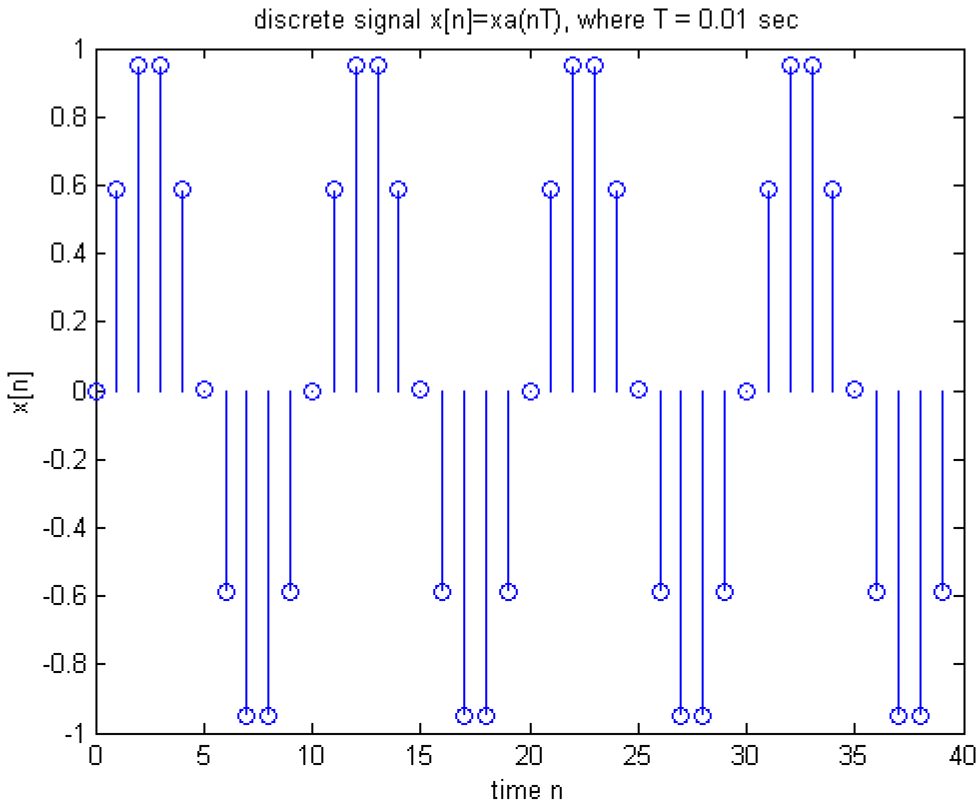



Fig. 1-2

Code Ex_1_2.m

```
% Generating a discrete-time signal x
clear;
f0=10;          % 10 Hz sine wave
Length=0.4      % Total length =0.4 sec
T=0.01;         % sampling period = 0.01 sec
N=Length/T;
n=0:1:N-1;
x=sin(2*pi*f0*n*T);
stem(n,x);
xlabel('time n'); ylabel('x[n]');
title('discrete signal x[n]=xa(nT), where T = 0.01 sec');
```

*******************************************************************

[Practice 1-2] Sketch signal $x[n]=\dfrac{\sin w_c n}{\pi n}$, where $w_c=0.2\pi,\ -30\le n\le 30$ 。

[Practice 1-3] Sketch a discrete-time signal that consists of 10Hz and 30Hz sine components based on a sampling period of 0.01 second.

*******************************************************************

# Lab 2: Convolution

## [Theoretical Background]

The convolution of two sequences $x[n]$ and $h[n]$ is defined as

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] \triangleq x[n] * h[n]$$

Note: $h[n] * \delta[n] = h[n]$.

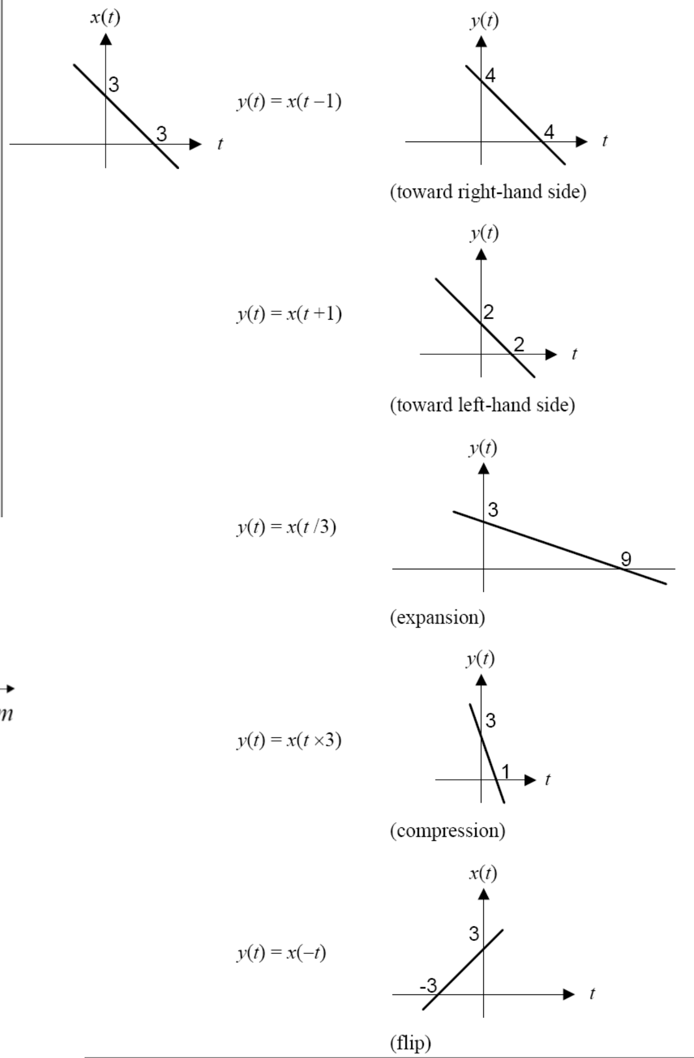
- **Procedure of convolution**

  1. Time-reverse: $h[m] \rightarrow h[-m]$
  2. Choose an $n$ value
  3. Shift $h[-m]$ by $n$: $h[n-m]$
  4. Multiplication: $x[n] \cdot h[n-m]$
  5. Summation over $m$: $y[n] = \sum_{m=-\infty}^{\infty} x[m]h[n-m]$

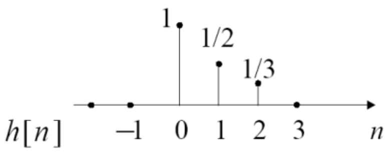
  Choose another $n$ value, go to Step 3.

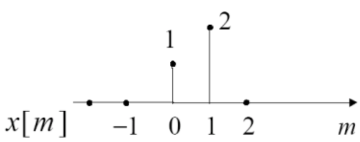
- Shifting and scaling

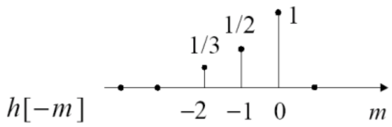
  Let $x(t) = 3 - t$, observe the following signals:



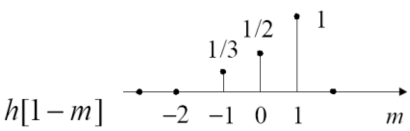

*Example:*



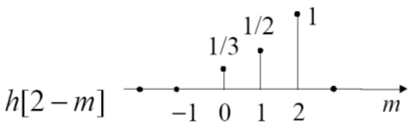



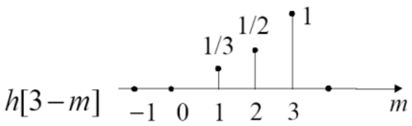

$y[0] = \sum_m x[m]h[-m] = 1$,




$y[1] = \sum_m x[m]h[1-m] = 5/2$,




$y[2] = \sum_m x[m]h[2-m] = 4/3$,




$y[3] = \sum_m x[m]h[3-m] = 2/3$.

Remark: if $x[n]$ has $M$ non-zero samples, and $h[n]$ has $N$ non-zero samples, then $y[n]$ has $(M + N - 1)$ samples.

Alternative solution 1:

| $x[n]$ \ $h[n]$ | 1 | 1/2 | 1/3 |
|---|---|---|---|
| 1 | 1 | 1/2 | 1/3 |
| 2 | 2 | 1 | 2/3 |

Alternative solution 2:

$$x[n] * h[n] = (\delta[n] + \frac{1}{2}\delta[n-1] + \frac{1}{3}\delta[n-2]) * (\delta[n] + 2\delta[n-1])$$

$$= \delta[n] + 2\delta[n-1] + \frac{1}{2}\delta[n-1] + \delta[n-1-1] + \frac{1}{3}\delta[n-2] + \frac{2}{3}\delta[n-2-1]$$

$$= \delta[n] + \frac{5}{2}\delta[n-1] + \frac{4}{3}\delta[n-2] + \frac{2}{3}\delta[n-3]$$

[Matlab Example2-1] Sketch the convolution of $x_1[n]$ and $x_2[n]$ as shown below.

$$y[n] = \sum_{k=-\infty}^{\infty} x_1[k]x_2[n-k] = \sum_{k=-\infty}^{\infty} x_2[k]x_1[n-k]$$

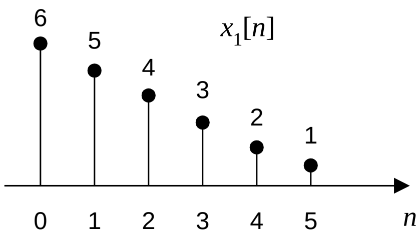



Fig. 2-1-1(a)

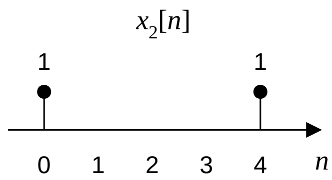



Fig. 2-1-1(b)

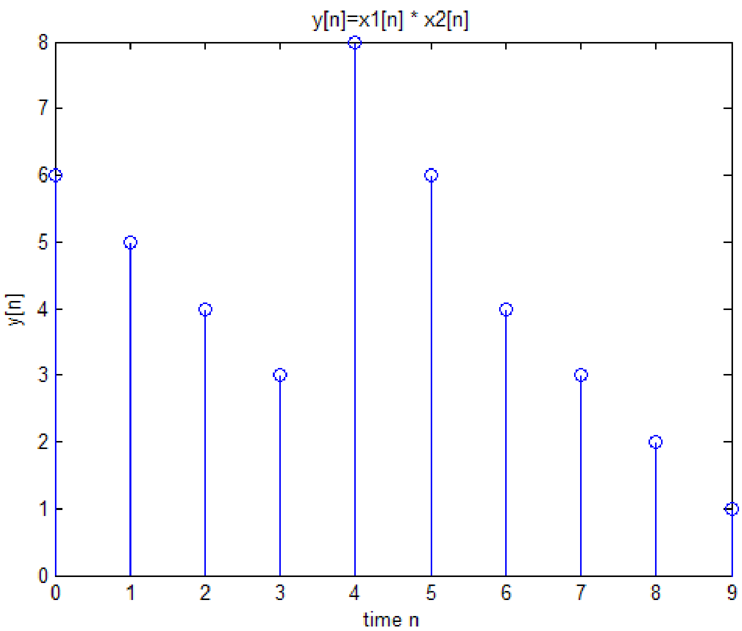



Fig. 2-1-2

Code Ex_2_1.m

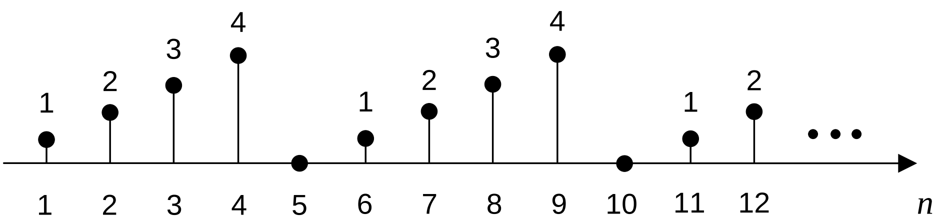
```
% y[n]=x1[n] * x2[n]
clear;
x1=[6 5 4 3 2 1];
x2=[1 0 0 0 1];
y=conv(x1,x2);
n=1:length(y);
stem(n,y);
xlabel('time n'); ylabel('y[n]');
title('y[n]=x1[n] * x2[n]');
```

*********************************************************************

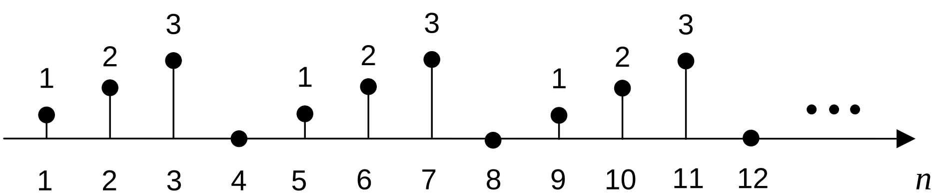
[Practice 2-1] Implement the convolution without using function conv().

[Practice 2-2] There are two signals $x_1[n] = n \% 5$, $x_2[n] = n \% 4$, $1 \le n \le 10000$. Implement their convolution using the matrix multiplication approach.

$x_1[n]$




$x_2[n]$




*********************************************************************

# Lab 3: DTFT, DFT, and FFT

## [Theoretical Background]

- **Discrete-Time Fourier Transform (DTFT)**

The Discrete-time Fourier transform decomposes a sequence into sinusoidal components of different frequencies, in which the frequency is continuous.

$$\text{Analysis:}\quad F\{x[n]\} = X(e^{jw}) = \sum_{n=-\infty}^{\infty} x[n]e^{-jwn}.$$

$$\text{Synthesis:}\quad F^{-1}\{X(e^{jw})\} = x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{jw})\, e^{jwn}dw.$$

Remark: $w$ is often chosen as between $-\pi$ and $\pi$, but any interval of length $2\pi$ can be used. Why length $2\pi$? The reason is that $X(e^{jw})$ is periodic with period $2\pi$.

$$X(e^{j(w+2\pi)}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j(w+2\pi)n} = \sum_{n=-\infty}^{\infty} x[n]e^{-jwn} = X(e^{jw}).$$

$X(e^{jw})$ is a complex value, which can be represented by $X(e^{jw}) = |X(e^{jw})|\, e^{j\angle X(e^{jw})}$, where

$|X(e^{jw})|$ is called *magnitude*, and $\angle X(e^{jw})$ is called *phase*.

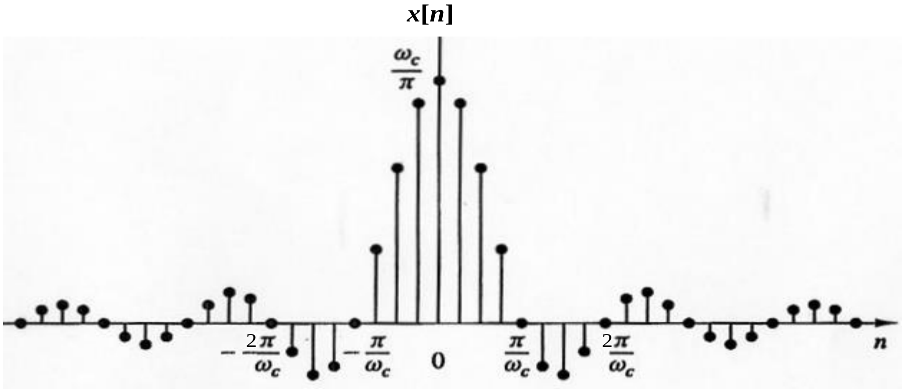
$X(e^{jw})$ does not necessary exist (converge), unless $|X(e^{jw})| < \infty$.

Example:

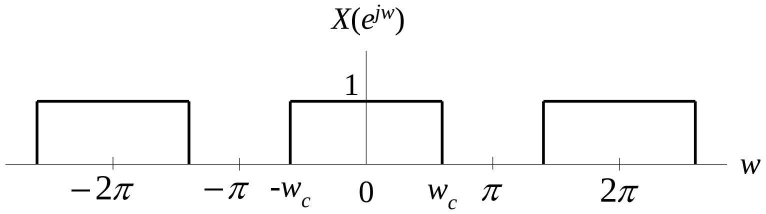
Let $x[n] = a^n u[n]$. $X(e^{jw}) = \sum_{n=0}^{\infty} a^n e^{-jwn} = \sum_{n=0}^{\infty}\left(ae^{-jw}\right)^n = \frac{1}{1-ae^{-jw}}$, if $|ae^{-jw}| < 1$, or $|a| < 1$

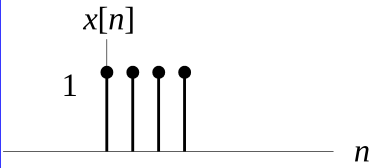
Note that $X(e^{jw})$ does not exist, if $|a| > 1$.

| Sequence | Fourier Transform |
|---|---|
| 1. $\delta[n]$ | $1$ |
| 2. $\delta[n-n_0]$ | $e^{-j\omega n_0}$ |
| 3. $1 \quad (-\infty < n < \infty)$ | $2\pi\delta(\omega)$ |
| 4. $a^n u[n] \quad (\|a\| < 1)$ | $\frac{1}{1-ae^{-j\omega}}$ |
| 5. $u[n]$ | $\frac{1}{1-e^{-j\omega}} + \pi\delta(\omega)$ |
| 6. $(n+1)a^n u[n] \quad (\|a\| < 1)$ | $\frac{1}{(1-ae^{-j\omega})^2}$ |
| 7. $\frac{\sin \omega_c n}{\pi n}$ | $X(e^{j\omega}) = \begin{cases} 1, & \|\omega\| < \omega_c, \\ 0, & \omega_c < \|\omega\| \le \pi \end{cases}$ |
| 8. $x[n] = \begin{cases} 1, & 0 \le n \le M \\ 0, & \text{otherwise} \end{cases}$ | $\frac{\sin[\omega(M+1)/2]}{\sin(\omega/2)} e^{-j\omega M/2}$ |
| 9. $e^{j\omega_0 n}$ | $2\pi\delta(\omega-\omega_0)$ |

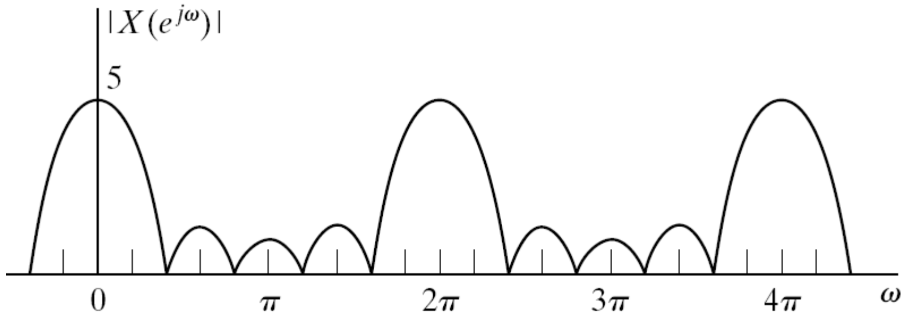


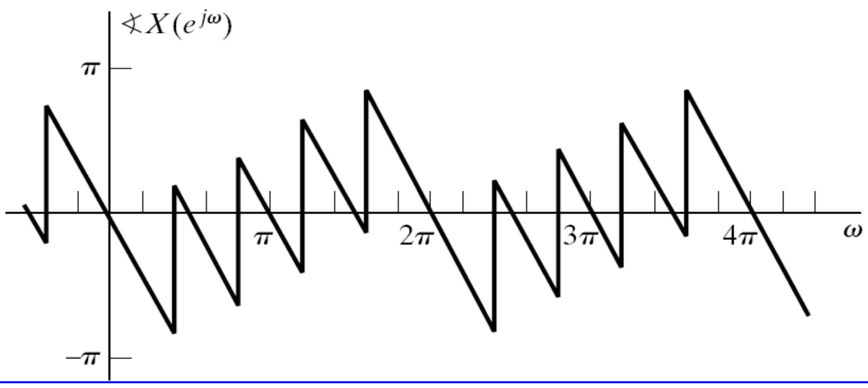

$$x[n]=\frac{\sin w_c n}{\pi n},\ -\infty\le n\le\infty \leftrightarrow X(e^{jw})=\begin{cases}1, & |w|\le w_c\\ 0, & w_c<|w|<\pi\end{cases}$$




$$x[n]=\begin{cases}1, & 0\le n\le N-1\\ 0, & \text{otherwise}\end{cases} \leftrightarrow X(e^{jw})=\frac{\sin(wN/2)}{\sin(w/2)}e^{-jw(N-1)/2}$$



- **Properties of DTFT:**

**-- Linearity**

If $x[n] \leftrightarrow X(e^{j\omega})$ and $y[n] \leftrightarrow Y(e^{j\omega})$

then $ax[n]+by[n] \leftrightarrow aX(e^{j\omega})+bY(e^{j\omega})$

**-- Time Shift**

If $x[n] \leftrightarrow X(e^{j\omega})$

then $x[n-n_d] \leftrightarrow X(e^{j\omega})e^{-j\omega n_d}$

**-- Frequency Modulation**

If $x[n] \leftrightarrow X(e^{j\omega})$

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

**--Time Reversal**

If $x[n] \leftrightarrow X(e^{j\omega})$

then $x[-n] \leftrightarrow X(e^{-j\omega})$

**-- Complex Conjugation**

If $x[n] \leftrightarrow X(e^{j\omega})$

then $x^*[n] \leftrightarrow X^*(e^{-j\omega})$

**-- Differentiation in frequency**

If $x[n] \leftrightarrow X(e^{jw})$, then $nx[n] \leftrightarrow j\dfrac{dX(e^{jw})}{dw}$,

because $$\frac{dX(e^{jw})}{dw} = \frac{d}{dw}\left(\sum_{n=-\infty}^{\infty} x[n]e^{-jwn}\right) = -j\sum_{n=-\infty}^{\infty} nx[n]e^{-jwn} = -jF\{nx[n]\}$$

**-- Convolution**

If $x[n] \leftrightarrow X(e^{jw})$, and $y[n] \leftrightarrow Y(e^{jw})$, then $x[n]*y[n] \leftrightarrow X(e^{jw})Y(e^{jw})$,

because

$$F\{x[n]*y[n]\} = \sum_{n=-\infty}^{\infty}\left(\sum_{k=-\infty}^{\infty} x[k]y[n-k]\right)e^{-jwn} = \sum_{m=-\infty}^{\infty}\sum_{k=-\infty}^{\infty} x[k]y[m]e^{-jw(m+k)}$$

$$= \sum_{m=-\infty}^{\infty} X(e^{jw})y[m]e^{-jwm} = X(e^{jw})Y(e^{jw})$$

**-- Multiplication**

If $x[n] \leftrightarrow X(e^{jw})$, and $y[n] \leftrightarrow Y(e^{jw})$,

then $x[n]y[n] \leftrightarrow \dfrac{1}{2\pi}X(e^{jw})*Y(e^{jw}) = \dfrac{1}{2\pi}\displaystyle\int_{-\pi}^{\pi} X(e^{j\theta})Y(e^{j(w-\theta)})d\theta$,

because

$$\frac{1}{2\pi}X(e^{jw})*Y(e^{jw}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\theta})Y(e^{j(w-\theta)})d\theta = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\theta})\left(\sum_{n=-\infty}^{\infty} y[n]e^{-j(w-\theta)n}\right)d\theta$$

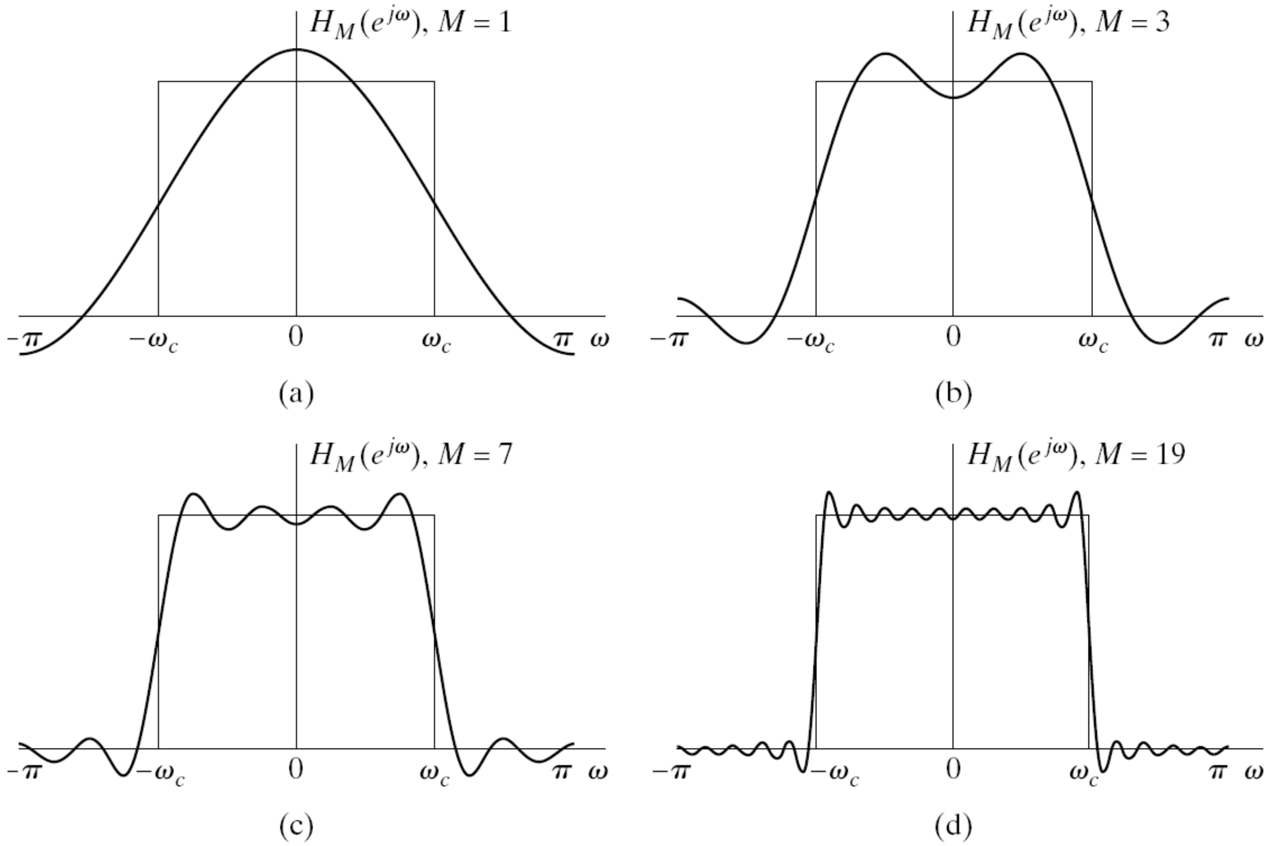
$$= \sum_{n=-\infty}^{\infty} y[n]\left(\frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\theta})e^{j\theta n}d\theta\right)e^{-jwn} = \sum_{n=-\infty}^{\infty} x[n]y[n]e^{-jwn} = F\{x[n]y[n]\}$$

- **Gibbs phenomenon**

As we shown if $X(e^{jw}) = \begin{cases} 1, & |w| \le w_c \\ 0, & w_c < |w| < \pi \end{cases}$, then $x[n] = \dfrac{1}{2\pi}\int_{-w_c}^{w_c} e^{jwn} dw = \dfrac{\sin w_c n}{\pi n}$, $-\infty \le n \le \infty$.

However, DTFT of $x[n]$, i.e., $\displaystyle\sum_{n=-\infty}^{\infty} \frac{\sin w_c n}{\pi n} e^{-jwn}$ , is not absolutely summable.

When considering the sum of a finite number of terms: $\displaystyle\sum_{n=-M}^{M} \frac{\sin w_c n}{\pi n} e^{-jwn}$, we obtain the following results.




As $M$ increases, the oscillatory behavior at $w = w_c$ is more rapid, but the size of the ripples does not decrease. When $M \to \infty$, the maximum amplitude of the oscillations does not approach zero. This is called the Gibbs phenomenon.

- **Discrete Fourier Transform (DFT)**

  $N$-pt DFT can be viewed as a discrete version of DTFT by dividing $2\pi$ into $N$ frequency bins.

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

  Synthesis: $DFT^{-1}(X[k]) = x[n] = \dfrac{1}{N}\displaystyle\sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}kn}$, $0 \le n \le N-1$.

- **DFT has a property of symmetry:**
  - If $x[n]$ is real, then $X[k] = X^*[N–k]$, because

$$X[k]=\sum_{n=0}^{N-1}x[n]e^{-j\frac{2\pi}{N}kn}=\sum_{n=0}^{N-1}x^*[n]e^{-j\frac{2\pi}{N}kn}=\left(\sum_{n=0}^{N-1}x[n]e^{j\frac{2\pi}{N}kn}\right)^*=\left(\sum_{n=0}^{N-1}x[n]e^{-j\frac{2\pi}{N}(N-k)n}\right)^*.$$

$$=X^*[N-k]$$

  - If $x[n]$ is real, and $x[n] = x[N–n]$, then $X[k]$ is also real, and $X[k] = X[N–k]$, because

$$X^*[k]=\left(\sum_{n=0}^{N-1}x[n]e^{-j\frac{2\pi}{N}kn}\right)^*=\sum_{n=0}^{N-1}x^*[n]e^{j\frac{2\pi}{N}kn}=\sum_{n=0}^{N-1}x[n]e^{j\frac{2\pi}{N}kn}=\sum_{n=N}^{1}x[N-n]e^{j\frac{2\pi}{N}kn}$$

$$=\sum_{n=N}^{1}x[N-n]e^{-j\frac{2\pi}{N}k(N-n)}=\sum_{m=0}^{N-1}x[m]e^{-j\frac{2\pi}{N}km}=X[k]$$

Example 1:

**x** = {1, 2, 3, 4, 5, 6, 7, 8}

*DFT*(**x**) = {36, -4+9.66$j$, -4+4$j$, -4+1.66$j$, -4, -4–1.66$j$, -4–4$j$, -4–9.66$j$}

$X[3] = X^*[8-3]$

Example 2:

**x** = {1, 2, 3, 4, 5, 6, 7, 8, 9}

*DFT*(**x**) = {45, -4.5+12.36$j$, -4.5+5.36$j$, -4.5+2.6$j$, -4.5+0.79$j$, -4.5–0.79$j$,
-4.5–2.6$j$, -4.5–5.36$j$, -4.5–12.36$j$}

$X[4] = X^*[9-4]$

Example 3:

**x** = {33, 1, 2, 3, 4, 4, 3, 2, 1}

*DFT*(**x**) = {53, 24.71, 32.72, 32, 32.57, 32.57, 32, 32.72, 24.71}

$DFT^{-1}$(**x**) = {5.89, 2.75, 3.64, 3.56, 3.62, 3.62, 3.56, 3.64, 2.75}

We can see that **x**, *DFT*(**x**), and $DFT^{-1}$ (**x**) are all real and symmetric.

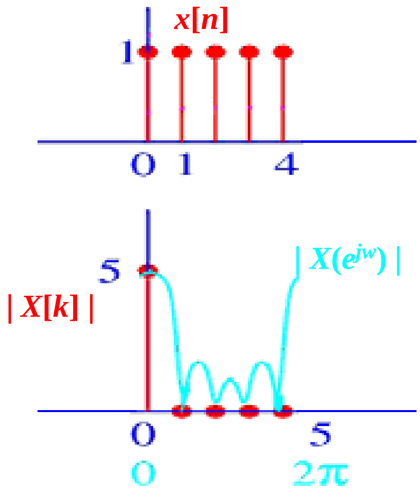
- **DFT of a rectangular window**

If $x[n]=\begin{cases}1, & 0\le n\le N-1\\ 0, & \text{otherwise}\end{cases}$, then

$$X(e^{jw})=\sum_{n=-\infty}^{\infty}x[n]e^{-jwn}=\sum_{n=0}^{N-1}e^{-jwn}=\frac{1-e^{-jwN}}{1-e^{-jw}}=\frac{\sin(wN/2)}{\sin(w/2)}e^{-jw(N-1)/2}$$

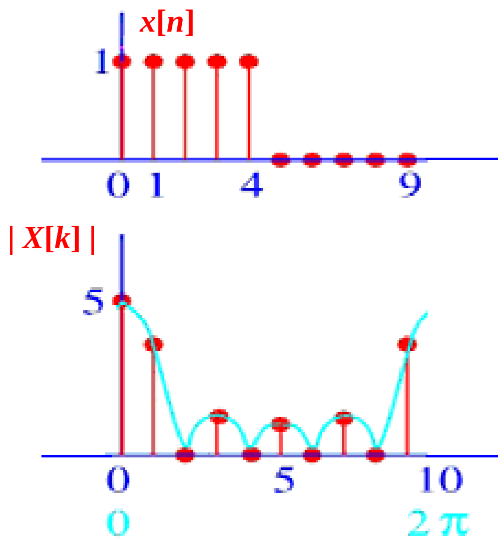
$$X[k]=\sum_{n=0}^{N-1}x[n]e^{-j\frac{2\pi}{N}kn}=\sum_{n=0}^{N-1}e^{-j\frac{2\pi}{N}kn}=\begin{cases}N, & k=0\\ \dfrac{1-e^{-j\frac{2\pi}{N}kN}}{1-e^{-j\frac{2\pi}{N}k}}=0, & k\neq 0\end{cases}$$

Example: $N$ =5




Zero padding: If we perform $M$-point DFT by appending $M$–$N$ zeros with $x[n]$, then

$$X[k]=\sum_{n=0}^{M-1} x[n]e^{-j\frac{2\pi}{M}kn}=\sum_{n=0}^{N-1} e^{-j\frac{2\pi}{M}kn}=\begin{cases} N, & k=0 \\ \dfrac{1-e^{-j\frac{2\pi}{M}kN}}{1-e^{-j\frac{2\pi}{M}k}}=\dfrac{\sin(\pi kN/M)}{\sin(\pi k/M)}e^{-j\pi k(N-1)/M}, & k\neq 0 \end{cases}$$




$\Rightarrow$ Zero padding of analyzed sequence results in “approximating” its DTFT better.

- Circular convolution

$$x_1[n] \, Ⓝ \, x_2[n]=\sum_{m=0}^{N-1} x_1[m]x_2[((n-m))_N] \xleftrightarrow{\text{DFT}} X_1[k]\cdot X_2[k]$$

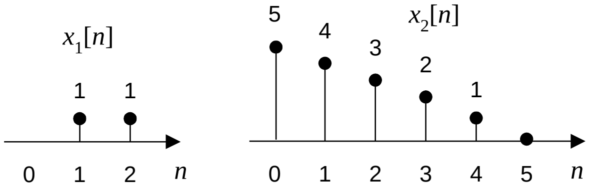
$$x_1[n]\cdot x_2[n] \xleftrightarrow{\text{DFT}} \frac{1}{N}X_1[k] \, Ⓝ \, X_2[k]=\frac{1}{N}\sum_{m=0}^{N-1} X_1[m]X_2[((k-m))_N]$$

where $x[((n))_N]=\sum_{r=-\infty}^{\infty} x[n+rN]$.

Note that in this case, the lengths of $x_1[n]$ and $x_2[n]$are both $N$ points.

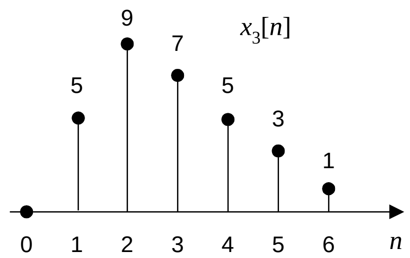
Example: Given $x_1[n]$ and $x_2[n]$ as the figures, sketch

(i) $x_3[n] = x_1[n] * x_2[n]$; (ii) $x_4[n] = x_1[n] ⑥ x_2[n]$; (iii) $x_5[n] = x_1[n] ⑦ x_2[n]$.



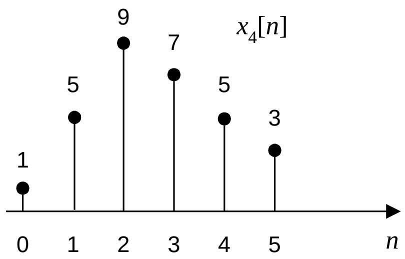

(i)

| | $n$ | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| $n$ | $x_2[n]\rightarrow$ / $x_1[n]\downarrow$ | 5 | 4 | 3 | 2 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 5 | 4 | 3 | 2 | 1 |
| 2 | 1 | 5 | 4 | 3 | 2 | 1 |



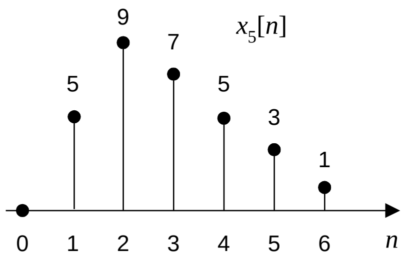

(ii)

| | $n$ | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $n$ | $x_2[((n))_6]\rightarrow$ / $x_1[n]\downarrow$ | 2 | 1 | 0 | 5 | 4 | 3 | 2 | 1 | 0 | 5 | 4 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 2 | 1 | 0 | 5 | 4 | 3 | 2 | 1 | 0 | 5 | 4 |
| 2 | 1 | 2 | 1 | 0 | 5 | 4 | 3 | 2 | 1 | 0 | 5 | 4 |




(iii)

| | $n$ | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $n$ | $x_2[((n))_7]\rightarrow$ / $x_1[n]\downarrow$ | 1 | 0 | 0 | 5 | 4 | 3 | 2 | 1 | 0 | 0 | 5 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 5 | 4 | 3 | 2 | 1 | 0 | 0 | 5 |
| 2 | 1 | 1 | 0 | 0 | 5 | 4 | 3 | 2 | 1 | 0 | 0 | 5 |



[Matlab Example 3-1] Sketch the DTFT of $x[n]$ as shown in Fig. 3-1-1.

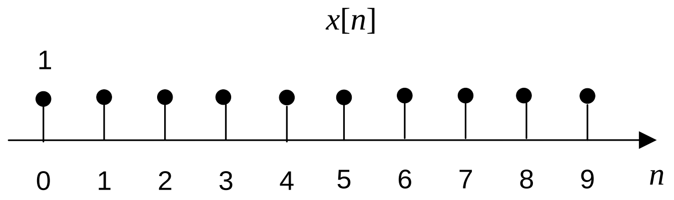



Fig. 3-1-1

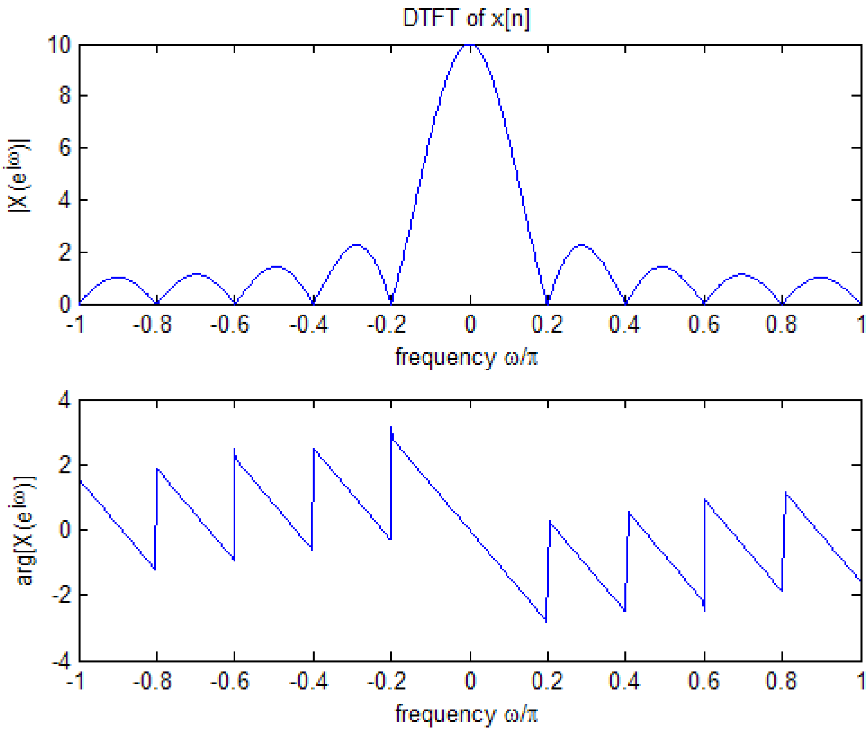



Fig. 3-1-2

Code Ex_3_1.m

```
% Computing the DTFT of signal x
clear;
x=[1 1 1 1 1 1 1 1 1 1];
n=0:length(x)-1;
K=500;
k=-K:K;
w=pi*k/K;
X=x*exp(-j*n'*w);
magX=abs(X);
angX=angle(X);
title('DTFT of x[n]');
subplot(2,1,1); plot(w/pi,magX);
xlabel('frequency \omega/\pi');    ylabel('|X(e^j^\omega)|');
subplot(2,1,2); plot(w/pi,angX);
xlabel('frequency \omega/\pi');    ylabel('arg(X(e^j^\omega))');
```

**********************************************************************

[Practice 3-1] Implement the DTFT without using functions exp(), abs(), and angle(), but instead, using the following expressions.

$X(e^{jw}) = \sum_{n=-\infty}^{\infty} x[n]e^{-jwn} = \sum_{n=-\infty}^{\infty} \{x[n]\cos(wn) - jx[n]\sin(wn)\} = X_R(e^{jw}) + jX_I(e^{jw})$, i.e., real and imaginary parts, and its magnitude and phase are $\sqrt{X_R^2(e^{jw}) + X_I^2(e^{jw})}$ and $\tan^{-1}\left[\frac{X_I(e^{jw})}{X_R(e^{jw})}\right]$, respectively.

**********************************************************************

[Matlab Example 3-2] Observe the Gibbs phenomenon.

As we known $x[n] = \frac{\sin w_c n}{\pi n} \quad \leftrightarrow \quad X(e^{jw}) = \begin{cases} 1, & |w| \le w_c \\ 0, & w_c < |w| < \pi \end{cases}$, $-\infty \le n \le \infty$. However, for finite value of *n*, e.g., $-M \le n \le M$, there exists inevitable oscillation in the magnitude of the DTFT of $x[n]$, which is called the Gibbs phenomenon.

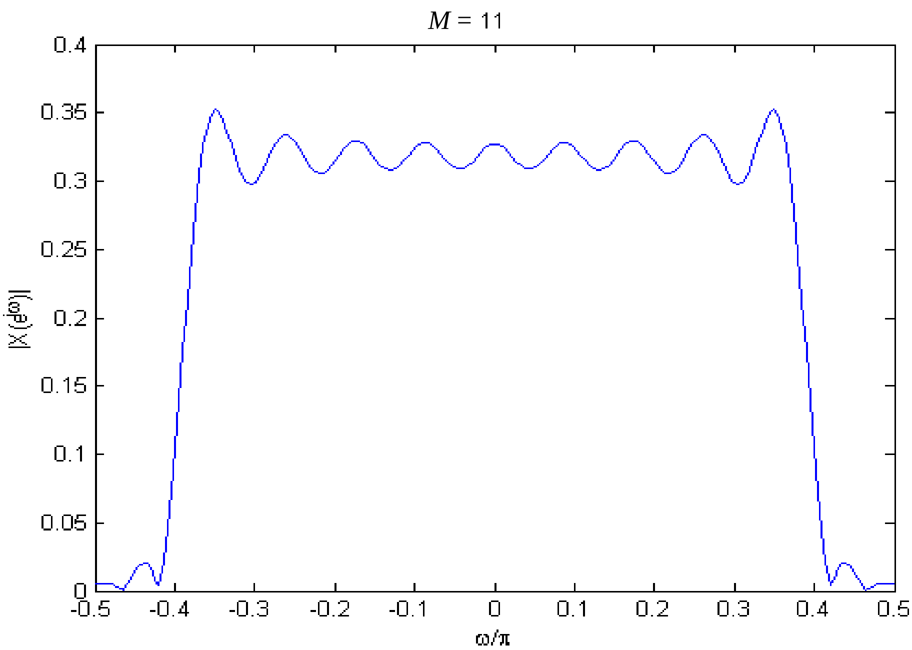


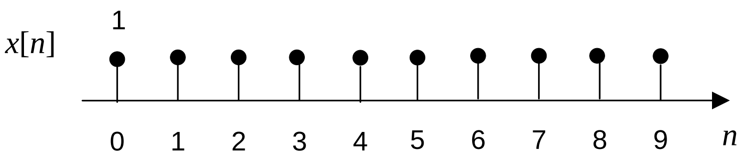

Fig. 3-2

Code Ex_3_2.m

```
f = linspace(-0.5,0.5,200);    % [-0.5pi, 0.5pi]
wc = 0.25*pi;                  % cutoff frequency

for M=1:5:2000
    n = (-M:M);
    for j=1:length(f)
       X(j) = sum(wc/pi*sinc(wc*n).*exp(-i*2*pi*n*abs(f(j))));
    end
```

```
    plot(f,abs(X));
    xlabel('\omega/\pi');
    ylabel('|X(e^j^\omega)|');
    title(M);
    pause;
end
```

[Matlab Example 3-3] Sketch the $N$-pt DFT of $x[n]$ as shown below, where the value of $N$ takes 10 and 100, respectively.



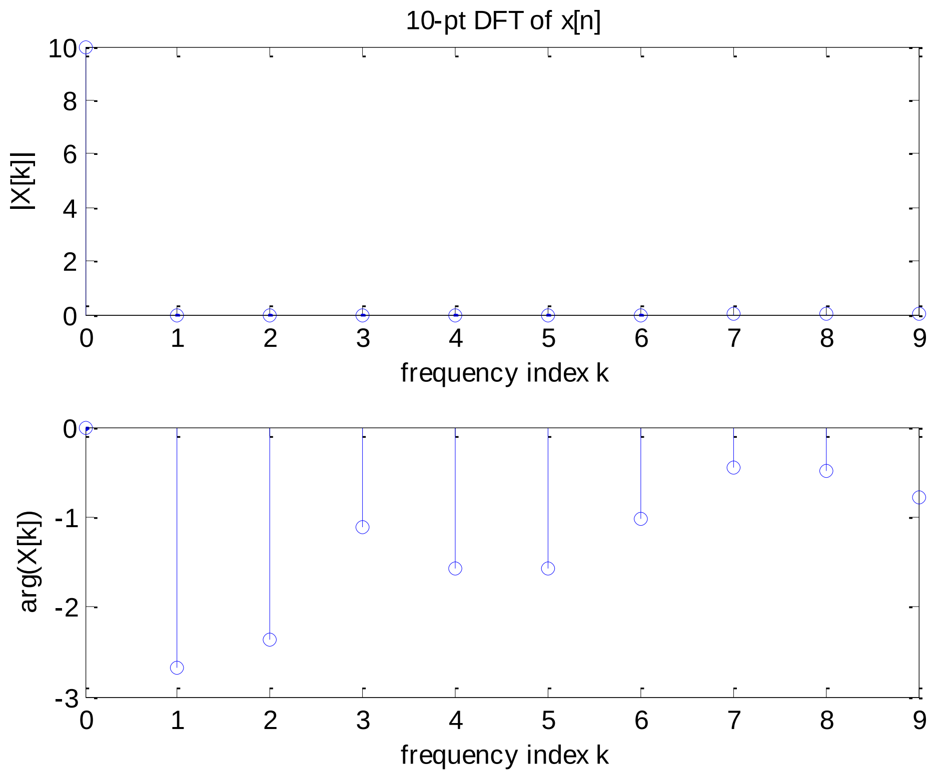




Fig. 3-3

Code Ex_3_3.m

```
% Computing the DFT of signal x
clear;
x=[1 1 1 1 1 1 1 1 1 1];
n=0:length(x)-1;
N=10;
k=0:N-1;
X=x*exp(-j*2*pi/N*n'*k);
magX=abs(X);
angX=angle(X);
```

```
subplot(2,1,1); stem(k,magX);  xlabel('frequency index k');
ylabel('|X[k]|');
title('10-pt DFT of x[n]');
subplot(2,1,2); stem(k,angX);  xlabel('frequency index k');
ylabel('arg(X[k])');
```

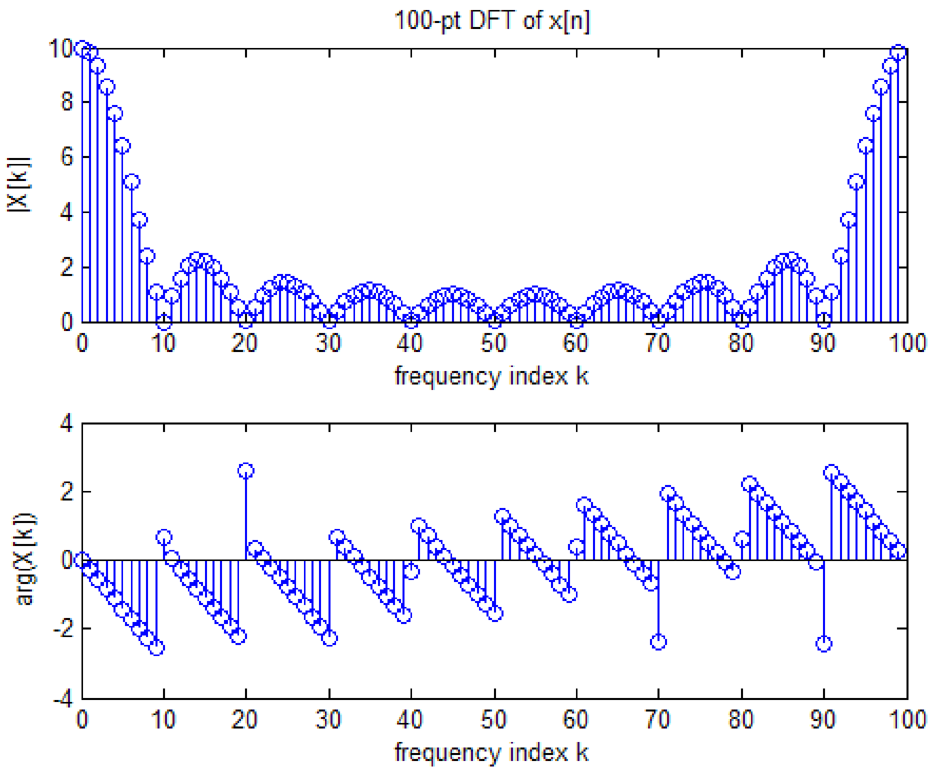



Fig. 3-4

**********************************************************************

[Practice 3-2] Take one period of the sine signal in Fig. 1-2, and then sketch its 10-pt DFT and 100-pt DFT, respectively.

[Practice 3-3] Use function fft() to compute the DFT in Practice 3-2 and sketch the DFT.

[Practice 3-4] Use function fft() to compute the DFT of the signal in Practice 1-3, i.e., a discrete-time signal that consists of 10Hz and 30Hz sine components based on a sampling period of 0.01 second.

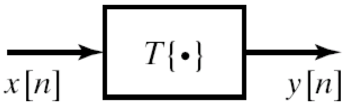
[Practice 3-5] Sketch a discrete-time signal that consists of 10Hz and 30Hz sine components based on a sampling period of 0.02 second. Then, use function fft() to compute its DFT, and compare the differences between this result and that in Practice 3-4.

**********************************************************************

# Lab 4: Digital Systems

## [Theoretical Background]

- A digital system is defined mathematically as a transformation or operator $T\{\cdot\}$ that maps an input sequence with values $x[n]$ into an output sequence with values $y[n] = T\{x[n]\}$.




- **System Classes**

1. **Memoryless**: if the output $y[n]$ at every value of $n$ depends only on the input $x[n]$ at the same value of $n$. For example, $y[n] = \{x[n]\}^2$.
2. **Linear**: if the system satisfies the principle of superposition, i.e., If $y_1[n] = T\{x_1[n]\}$, and $y_2[n] = T\{x_2[n]\}$, then $T\{ax_1[n] + bx_2[n]\} = ay_1[n] + by_2[n]$.
   Note that a linear system also satisfy $y[n] = 0$ for $x[n] = 0$, $\forall n$, because:
   if $y_1[n] = T\{0\} \neq 0$, and let $y_2[n] = T\{x_2[n]\}$, then $y_1[n] + y_2[n] \neq y_2[n] = T\{x_2[n]\} = T\{0+x_2[n]\}$
3. **Time-invariant**: a time shift or delay of the input sequence causes a corresponding shift in the output sequence, i.e., If $x_1[n] = x[n - n_d]$, then $y_1[n] = y[n - n_d]$.
4. **Causal**: $y[n]$ does not depend on the future value of $x[n]$, e.g., $x[n+1]$.
5. **Stable**: if $|\,x[n]\,| < \infty$, then $|\,y[n]\,| < \infty$ for all $n$.

- **Linear Time-invariant (LTI) Systems**

  An LTI system can be completely characterized by its impulse response $h[n]$.

  Let $x[n] = \delta[n]$ be an input to a system, then the output is $y[n] = h[n]$.

  More generally, $y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] = x[n] * h[n]$ (**convolution**)

- **Z Transform**

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n},$$

$$x[n] = \frac{1}{2\pi j}\oint_c X(z)z^{n-1}dz,$$

where $z$ is a complex, and where $C$ is a counterclockwise closed path encircling the origin and entirely in the region of convergence (ROC).

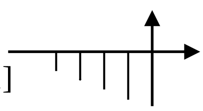
If we let $z = e^{jw}$, then z transform is equivalent to DTFT, but note that $z$ is not limited to $e^{jw}$

Example:

1. $x[n] = a^n\, u[n]$

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n} = \sum_{n=0}^{\infty} a^n z^{-n} = \sum_{n=0}^{\infty} (az^{-1})^n = \frac{1}{1-az^{-1}},\ \text{if } |\,az^{-1}\,| < 1\ \ (|z| > |a|)$$

2. $x[n] = -a^n\, u[-n-1]$



$$X(z) = -\sum_{n=-\infty}^{-1} a^n z^{-n} = -\sum_{n=1}^{\infty} a^{-n} z^n = 1 - \sum_{n=0}^{\infty} (a^{-1}z)^n = 1 - \frac{1}{1-a^{-1}z} = \frac{1}{1-az^{-1}}, \text{ if } |a^{-1}z| < 1 \quad (|z| < |a|)$$

Note that in the above example, if $a > 1$, $X(e^{jw})$ does not exist, because $\sum_{n=0}^{\infty} a^n e^{-jwn}$ is not absolutely summable. However, $X(z)$ exist, if $|z| > |a|$.

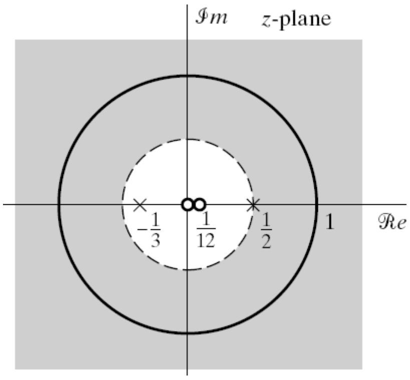
For any given $X(z)$, the set of values of $z$ for which the z-transform converges is called the region of convergence (ROC)

Example: $x[n] = \left(\frac{1}{2}\right)^n u[n] + \left(-\frac{1}{3}\right)^n u[n]$

$$X(z) = \sum_{n=-\infty}^{\infty} \left\{ \left(\frac{1}{2}\right)^n u[n] + \left(-\frac{1}{3}\right)^n u[n] \right\} z^{-n} = \sum_{n=0}^{\infty} \left(\frac{1}{2}\right)^n z^{-n} + \sum_{n=0}^{\infty} \left(-\frac{1}{3}\right)^n z^{-n}$$

$$= \frac{1}{1-\frac{1}{2}z^{-1}} + \frac{1}{1+\frac{1}{3}z^{-1}} = \frac{2z(z-\frac{1}{12})}{(z-\frac{1}{2})(z+\frac{1}{3})}, \text{ if } |\frac{1}{2}z^{-1}| < 1, \text{and} |\frac{1}{3}z^{-1}| < 1$$

$$\Rightarrow |z| > \frac{1}{2}, \text{and} |z| > \frac{1}{3} \quad \Rightarrow |z| > \frac{1}{2}$$



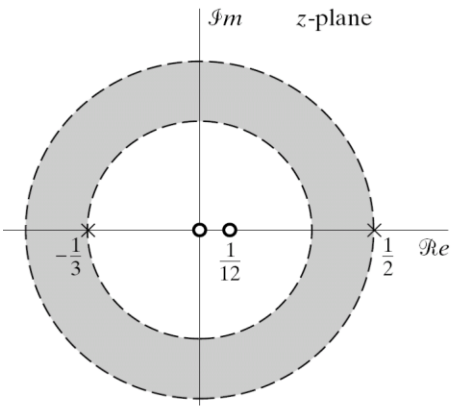

Zeros: $X(z) = 0 \Rightarrow z = 0, 1/12$
Poles: $X(z) = \infty \Rightarrow z = 1/2, -1/3$
Note: the number of pole equals the number of zeros.

Example: $x[n] = \left(-\frac{1}{3}\right)^n u[n] - \left(\frac{1}{2}\right)^n u[-n-1]$

$$X(z) = \frac{1}{1+\frac{1}{3}z^{-1}} \quad (\text{if } |z| > \frac{1}{3})$$

$$+ \frac{1}{1-\frac{1}{2}z^{-1}} \quad (\text{if } |z| < \frac{1}{2})$$

$$= \frac{2z(z-\frac{1}{12})}{(z+\frac{1}{3})(z-\frac{1}{2})}, \ |z| < \frac{1}{2}, \text{and} |z| > \frac{1}{3}$$



Inverse z-transform

Example 1: If $X(z) = z^2(1-0.5z^{-1})\ (1 + z^{-1})\ (1-z^{-1})$, find $x[n]$.

Because $X(z) = z^2 - 0.5z - 1 + 0.5z^{-1}$,

$x[n] = \delta[n + 2] - 0.5\delta[n+1] - \delta[n] + 0.5\delta[n-1]$.

Example 2: If $X(z) = \dfrac{1+2z^{-1}+z^{-2}}{1-1.5z^{-1}+0.5z^{-2}}$, find $x[n]$.

Because $X(z) = 2 - \dfrac{9}{1-0.5z^{-1}} + \dfrac{8}{1-z^{-1}}$,

$x[n] = 2\delta[n] - 9(0.5)^n u[n] + 8u[n]$

ROC: $|z| > 0.5$ and $|z| > 1 \Rightarrow |z| > 1$

- **Properties of LTI Systems**

$x[n] \longrightarrow$ LTI System, $h[n]$ (impulse response) $\longrightarrow y[n] = x[n] * h[n]$

$X(e^{jw}) \longrightarrow$ LTI System, $H(e^{jw}) = \mathcal{F}\{h[n]\}$ (frequency response) $\longrightarrow Y(e^{jw}) = X(e^{jw})\, H(e^{jw})$

$X(z) \longrightarrow$ LTI System, $H(z) = \mathcal{Z}\{h[n]\}$ (system function) $\longrightarrow Y(z) = X(z)\, H(z)$

Causality: an LTI system is causal, if $h[n] = 0$, for $n < 0$.

Because $y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k]$, and if the system is causal,

then $y[n]$ does not depend on any $x[n - k]$ for $k < 0$, $h[k]$ must be 0 for $k < 0$.

the ROC of $H(z)$ must be outside the outermost pole

For example, if $h[n] = (1/2)^n u[n] + (-1/3)^n u[n]$, we know that $h[n]$ is causal.

Taking the z-transform of $h[n]$, we have

$$H(z) = \frac{1}{1-(1/2)z^{-1}} + \frac{1}{1+(1/3)z^{-1}} = \frac{2z(z-1/12)}{(z-1/2)(z+1/3)}, \text{ ROC: } |z| > 1/2$$

The poles of $H(z)$ are 1/2 and -1/3. We can see that ROC of $H(z)$ is outside the outermost pole, 1/2.

Stability: an LTI system is stable, if $h[n]$ is absolutely summable, i.e., $\sum_{k=-\infty}^{\infty} |h[k]| < \infty$

The ROC of $H(z)$ includes the unit circle. Why?

Recall that a stable system is defined as:

if input $|x[n]| < \infty$, then output $|y[n]| < \infty$ for all $n$.

Since $|y[n]| = \left|\sum_{k=-\infty}^{\infty} h[k]x[n-k]\right| \le \sum_{k=-\infty}^{\infty} |h[k]||x[n-k]|$, a stable system satisfies

$\sum_{k=-\infty}^{\infty} |h[k]| < \infty$, which is equivalent to the condition $\sum_{k=-\infty}^{\infty} |h[k]z^{-k}| < \infty$ for $|z| = 1$.

This means that the ROC of $H(z)$ includes the unit circle.

An LTI system $H(z)$ is causal and stable, if all the poles of $H(z)$ are inside the unit circle.

- Inverse system

An inverse system $H_i(z)$ of the system $H(z)$ is defined as $H(z)\ H_i(z) = 1$, that is, $H_i(z) = 1/H(z)$. Since $H_i(z) = 1/H(z)$, the ROCs of $H_i(z)$ and $H(z)$ must overlap

Example:

If the system $H(z) = \dfrac{1-0.5z^{-1}}{1-0.9z^{-1}}$ is stable, what is the impulse response of its inverse system?

Apparently, $H_i(z) = \dfrac{1-0.9z^{-1}}{1-0.5z^{-1}} = \dfrac{1}{1-0.5z^{-1}} - \dfrac{0.9z^{-1}}{1-0.5z^{-1}}$. There are two possibilities for the ROC of $H_i(z)$, one is $|z| < 0.5$, and another is $|z| > 0.5$.

Since $H(z)$ is stable, the ROC of $H(z)$ is $|z| > 0.9$. To ensure that the ROCs of $H_i(z)$ and $H(z)$ overlap, we choose $|z| > 0.5$ as the ROC of $H_i(z)$.

Therefore, the impulse response of $H_i(z)$ is

$$h_i[n] = (0.5)^n u[n] - 0.9(0.5)^{n-1}u[n-1],$$

which is both causal and stable, or alternatively,

using $H_i(z) = 1 - \dfrac{0.4z^{-1}}{1-0.5z^{-1}}$, we have $h_i[n] = \delta[n] - 0.4(0.5)^{n-1}u[n-1]$,

which is also causal and stable.

- All-pass system

A system of the form $H_{ap}(z) = \dfrac{z^{-1}-a^*}{1-az^{-1}}$ is called an allpass system.

An allpass system has constant magnitude that is independent of frequency $w$, because

$$|H_{ap}(e^{jw})| = \left|\frac{e^{-jw}-a^*}{1-ae^{-jw}}\right| = \left|e^{-jw}\frac{1-a^*e^{jw}}{1-ae^{-jw}}\right| = 1.$$

- Minimum phase system

  If an LTI system is causal and stable, and its inverse system is also causal and stable, then this system is called "minimum phase system".

  How to check if a system is minimum phase?

  ⇒ The poles and zeros of a minimum phase system are all inside the unit circle.

  Any rational system function $H(z)$ can be decomposed as the form of

  $H(z) = H_{ap}(z)\, H_{min}(z)$

  Why? Suppose $H(z)$ has one zero $z = 1/c^*$, where $|c| < 1$, so that the zero is outside the unit circle. Then, we can express $H(z)$ as

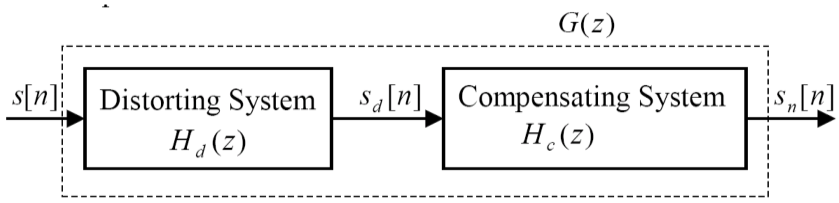
$$H(z) = H_1(z)(z^{-1} - c^*),$$

  where $H_1(z)$ is composed of the remaining poles and zeros of $H(z)$ inside the unit circle. We can also rewrite $H(z)$ as

$$H(z) = \underbrace{H_1(z)(1 - cz^{-1})}_{\text{minimum phase}} \underbrace{\frac{z^{-1} - c^*}{1 - cz^{-1}}}_{\text{allpass}}.$$

- Frequency-response compensation

  In many applications, signals are distorted by an LTI system with an undesirable frequency response. It may be of interest to process the distorted signal with a compensating system.




  Institutively, $H_c(z) = 1/\, H_d(z)$, so that the distortion can be eliminated. However, because $H_d(z)$ is sometimes not causal and stable, it may not be possible to realize $H_c(z)$. To solve this problem, we can decompose $H_d(z)$ into $H_d(z) = H_{d,min}(z)H_{ap}(z)$, and then design $H_c(z) = 1/\, H_{d,min}(z)$. This guarantees that $H_c(z)$ is causal and stable. In addition, we note that the overall system function $G(z) = H_d(z)H_c(z) = H_{ap}(z)$.

[Matlab Example 4-1] A signal as shown in Fig. 4-1 is input to a system $y[n] = x[n] - x[n-1]$. The result (response) can be computed in two ways: filtering and convolution. If filtering is performed, the result is [2 1 1 1]. If convolution is performed, the result is [2 1 1 1 -5]. In Code Ex_4_1.m, we can verify that if $x[n]$ Ⓝ $h[n] = y[n]$, then $X[k] \cdot H[k] = Y[k]$.

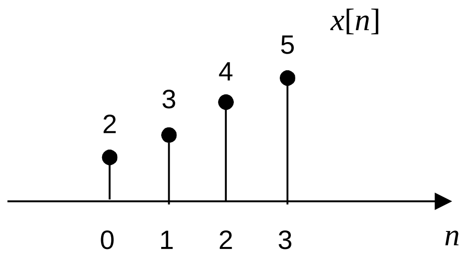



Fig. 4-1

Code Ex_4_1.m

```
% Backward difference system y[n] = x[n] - x[n-1]
a=[1];
b=[1 -1];
x=[2 3 4 5];
y=filter(b,a,x)      % y = [2 1 1 1]

h=[1 -1];
x=[2 3 4 5];
w=conv(h,x)    % w = [2 1 1 1 -5]

h1=[1 -1 0 0 0];
x1=[2 3 4 5 0];
H=fft(h1)
X=fft(x1)
Z=H.*X
W=fft(w)          % Z = W
```

*******************************************************************

[Practice 4-1] Input the signal in Fig. 4-1 to system $y[n] = 0.8y[n\text{-}1] + x[\text{n}] - x[n\text{-}1]$. Sketch the response based on filtering and convolution, respectively. Observe the differences between filtering and convolution.

*******************************************************************

[Matlab Example 4-2] By decomposing $H(z) = \dfrac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots}$ into $H(z) = \dfrac{(1 - d_1 z^{-1})(1 - d_2 z^{-1})\cdots}{(1 - c_1 z^{-1})(1 - c_2 z^{-1})\cdots}$ and plotting the pole-zero diagram, we can check if the system is stable or not. For example, if system $H(z) = \dfrac{1 - 5z^{-1} + 6z^{-2}}{1 - 4.5z^{-1} + 2z^{-2}} = \dfrac{(1 - 3z^{-1})(1 - 2z^{-1})}{(1 - 4z^{-1})(1 - 0.5z^{-1})}$ is causal, then we can see that the system is unstable, since its ROC does not contain the unit circle. Note that sometimes small numerical errors in the transfer function $H(z)$ may affect the stability significantly. For example,

Suppose $H_1(z) = \dfrac{1}{1 - 1.845z^{-1} + 0.850586z^{-2}} = \dfrac{1}{(1 - 0.943z^{-1})(1 - 0.902z^{-1})}$ is causal. Then, $H_1(z)$ is also stable, since all the poles are inside the unit circle. However, if the coefficients of $H_1(z)$ are rounded off the 2nd decimal place, i.e.,

$H_2(z) = \dfrac{1}{1 - 1.85z^{-1} + 0.85z^{-2}} = \dfrac{1}{(1 - z^{-1})(1 - 0.85z^{-1})}$, the resulting poles will be outside the unit circle, which turn the system from stable to unstable.

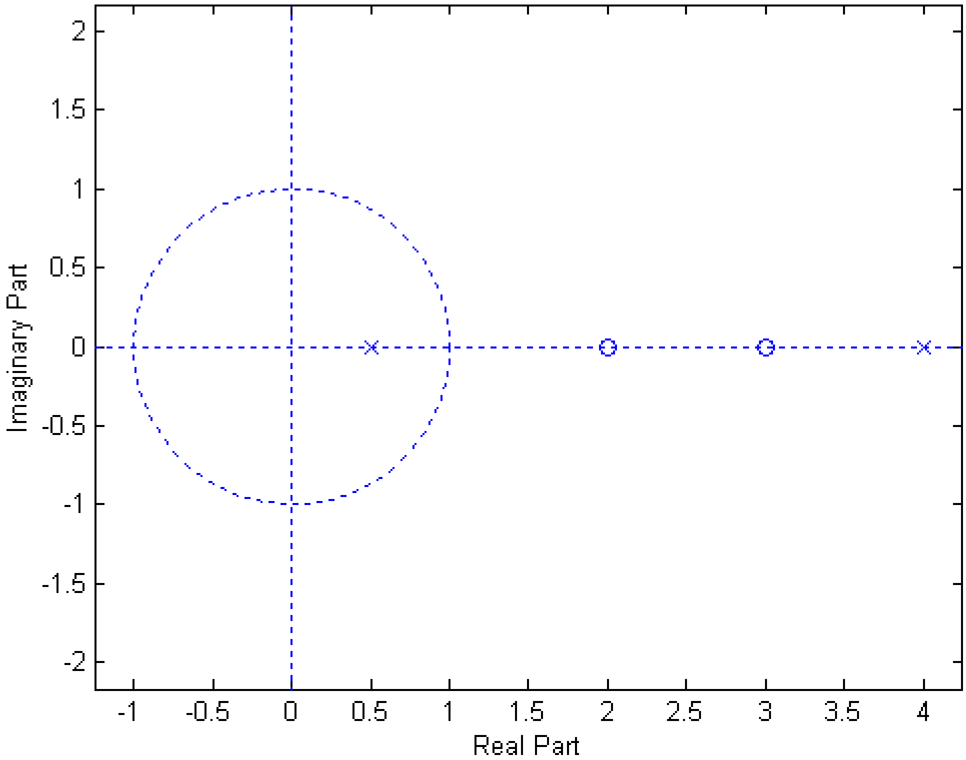



Fig. 4-2

Code Ex_4_2.m

```
% pole-zero plot of H(z)
num = input('Type in the numerator coefficients (e.g., [1 -5 6]) = ');
den = input('Type in the denominator coefficients (e.g., [1 -4.5 2])
= ');
roots(num)
roots(den)
zplane(num,den)
```

[Matlab Example 4-3] A 10Hz sine signal is input to an allpass system $H(z)=\dfrac{z^{-1}-0.5}{1-0.5z^{-1}}$. We can find that the output signal is also a 10Hz sine signal.

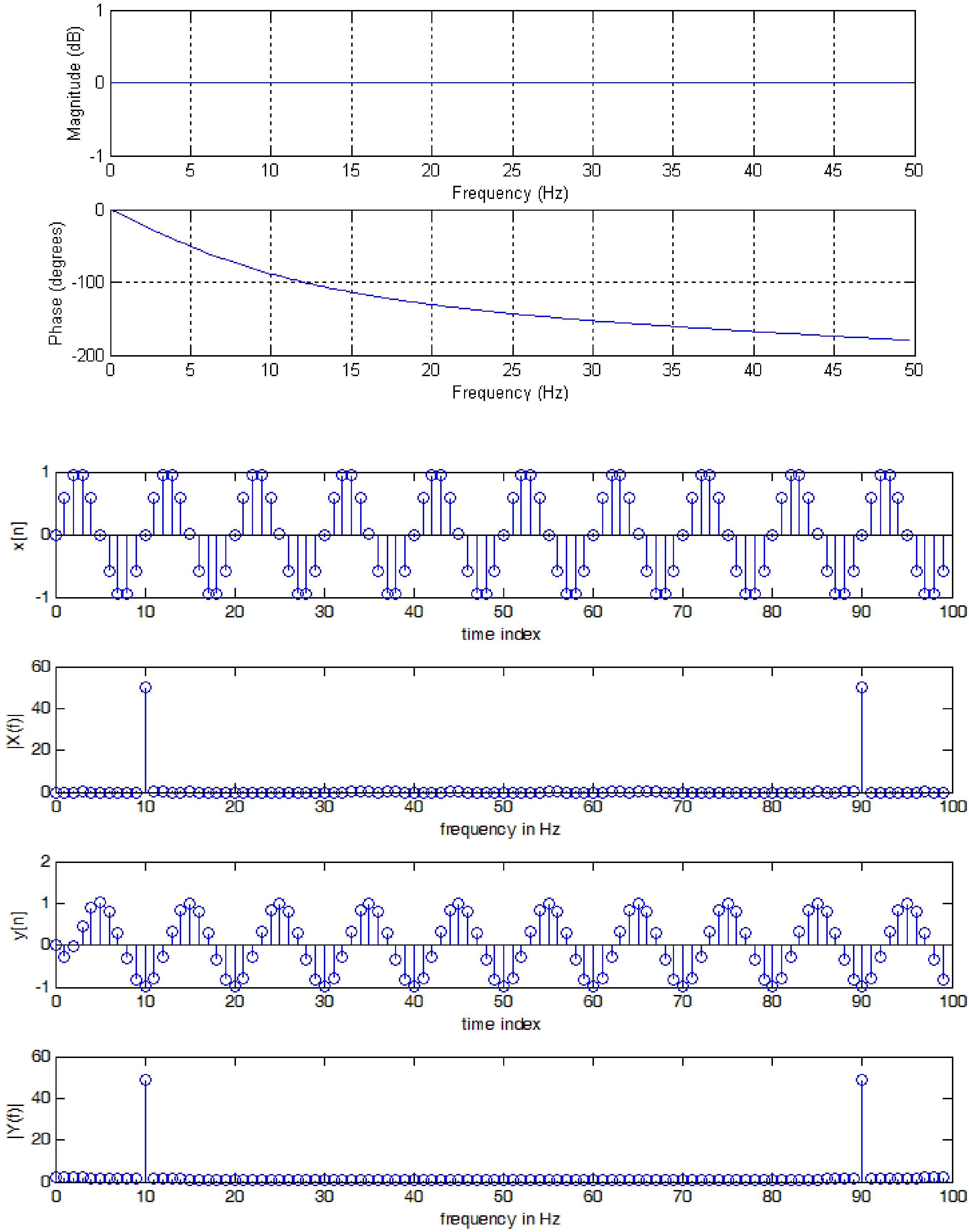



Fig. 4-3

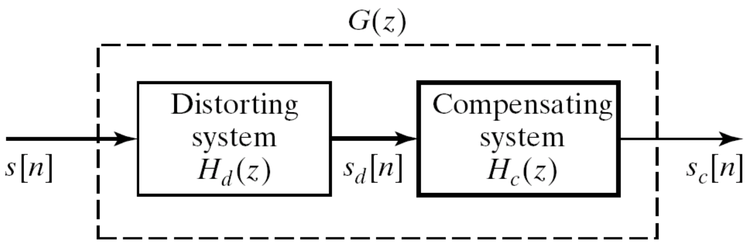
Code Ex_4_3.m

```
% a 10-Hz sine wave is input to an allpass system
num = [-0.5 1];
```

```
den = [1 -0.5];
freqz(num,den,200,100);
pause;
zplane(num,den);
pause;

f0=10;          % 10 Hz sine wave
T=0.01;         % sampling freq. = 100 Hz
N=100;
n=0:1:N-1;
x=sin(2*pi*f0*n*T);
subplot(4,1,1); stem(n,x);
xlabel('time index'); ylabel('x[n]');

f=n/T/N;
subplot(4,1,2); stem(f,abs(fft(x)));
xlabel('frequency in Hz'); ylabel('|X(f)|');


y=filter(num,den,x);
subplot(4,1,3); stem(n,y);
xlabel('time index'); ylabel('y[n]');
subplot(4,1,4); stem(f,abs(fft(y)));
xlabel('frequency in Hz'); ylabel('|Y(f)|');
```

*********************************************************************

[Practice 4-2] Design a compensating system $H_c(z)$ as shown below (must be a minimum phase system). Suppose that a sine signal $s[n]$ consisting of 10Hz and 30Hz is input to a distorting system $H_d(z) = \dfrac{1-6.9z^{-1}+13.4z^{-2}-7.2z^{-3}}{1-1.3z^{-1}+0.47z^{-2}-0.035z^{-3}}$. Plot the output $s_c[n]$ and its DFT.

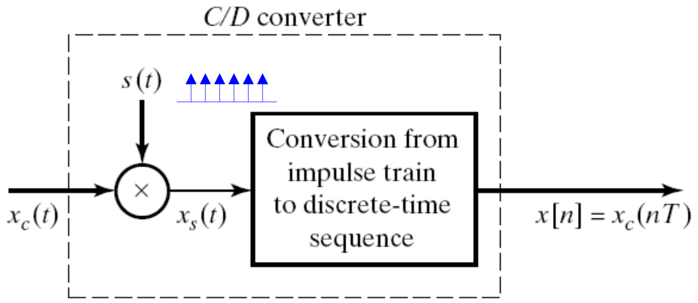


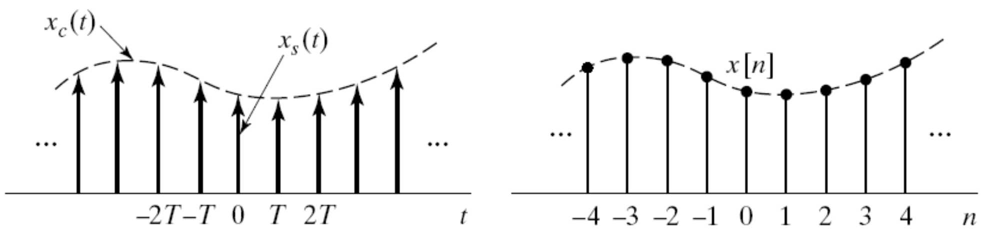

*********************************************************************

# Lab 5: Audio Signal Processing

## [Theoretical Background]

- **Analog (continuous) to discrete conversion**




In the time domain

Modulating signal $s(t)=\sum_{n=-\infty}^{\infty}\delta(t-nT)$

Sampled signal $x_s(t)=x_c(t)s(t)=x_c(t)\sum_{n=-\infty}^{\infty}\delta(t-nT)=\sum_{n=-\infty}^{\infty}x_c(nT)\delta(t-nT)$

Discrete-time signal $x[n] = x_c(nT)$, which is the non-zero terms of $x_s(t)$.

In the frequency domain

By using $x_s(t)=x_c(t)s(t)$, we can obtain

$$
\begin{aligned}
X_s(j\Omega) &= \frac{1}{2\pi}X_c(j\Omega)*S(j\Omega)\\
&= \frac{1}{2\pi}X_c(j\Omega)*\left[\frac{2\pi}{T}\sum_{k=-\infty}^{\infty}\delta(j\Omega-jk\Omega_s)\right]\\
&= \frac{1}{T}\sum_{k=-\infty}^{\infty}X_c(j\Omega-jk\Omega_s)
\end{aligned}
$$

$$s(t)=\sum_{n=-\infty}^{\infty}\delta(t-nT)=\sum_{k=-\infty}^{\infty}s_k e^{jk\Omega_s t}$$

$s_k$ is the Fourier series coefficients,

$$s_k=\frac{1}{T}\int_{-T/2}^{T/2}\delta(t)e^{-jk\Omega_s t}dt=\frac{1}{T}$$

Therefore,

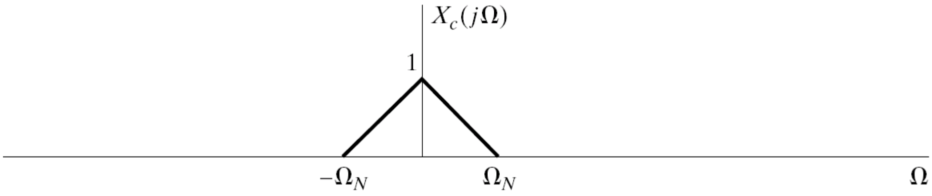
$$
\begin{aligned}
S(j\Omega) &= \mathcal{F}\left\{\sum_{k=-\infty}^{\infty}\frac{1}{T}e^{jk\Omega_s t}\right\}\\
&= \frac{2\pi}{T}\sum_{k=-\infty}^{\infty}\delta(j\Omega-jk\Omega_s)
\end{aligned}
$$

By using $x_s(t)=\sum_{n=-\infty}^{\infty}x_c(nT)\delta(t-nT)$, we get

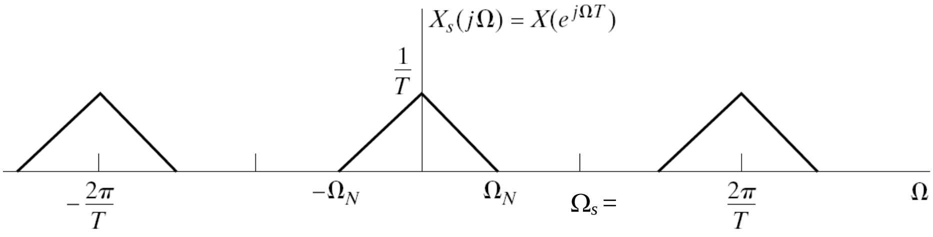
$$X_s(j\Omega)=\sum_{k=-\infty}^{\infty}x_c(nT)e^{-jn\Omega T}=\sum_{k=-\infty}^{\infty}x[n]e^{-jn\Omega T} \qquad (1)$$

Since $X(e^{jw})=\sum_{k=-\infty}^{\infty}x[n]e^{-jwn}$, $\qquad$ (2)

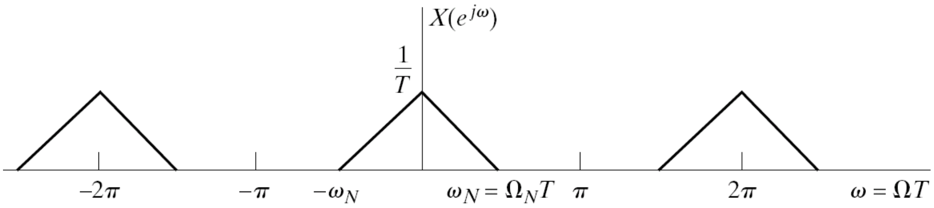
Comparing (1) and (2), we see that $X(e^{jw})=X_s(j\Omega)\big|_{\Omega=\frac{w}{T}}=\frac{1}{T}\sum_{k=-\infty}^{\infty}X_c\left(j\frac{1}{T}(w-2\pi k)\right)$



(a)



(b)



(c)

** The sampled signal spectrum is the sum of shifted copies of the original. **

Note:

$\Omega$ is the analog frequency

$w$ is the digital frequency

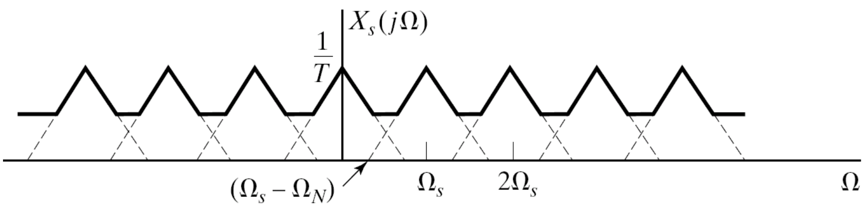
$w = \Omega T$, where $T$ is the sampling period

$T = 1/f_s$, where $f_s$ is the sampling frequency in Hz

$\Omega_s = 2\pi f_s = 2\pi / T$ is the sampling frequency in radius.

- Aliasing

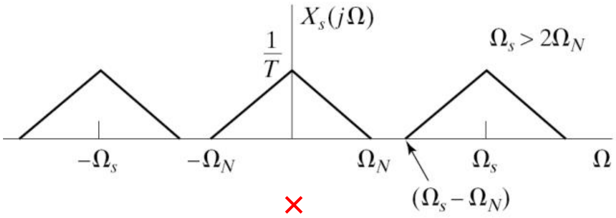
  If $\Omega_s < 2\Omega_N$, the copies of $X_c(j\Omega)$ overlap, which results in distortion.



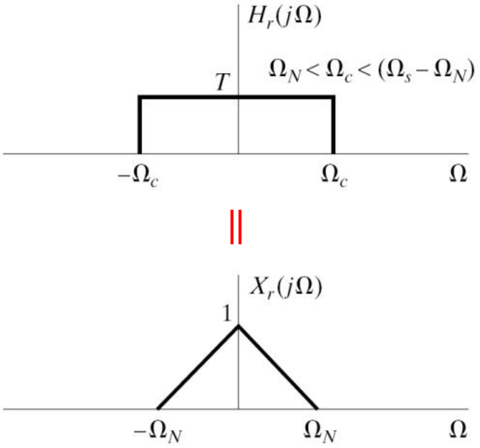

- Nyquist sampling theorem

  To avoid aliasing, the sampling frequency must be at least the double of the highest frequency of the analog signal, i.e., $\Omega_s \geq 2\Omega_N$

- Reconstruction of an analog signal from its samples



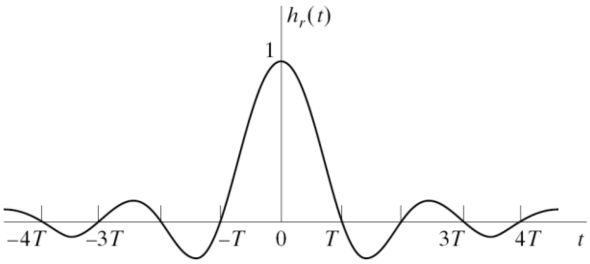

×




(e)

Usually, $\Omega_c$ is set to be $\Omega_s /2 = \pi/T$

$$h_r(t) = \frac{1}{2\pi}\int_{-\pi/T}^{\pi/T} T e^{j\Omega t} d\Omega = \frac{\sin(\pi t/T)}{\pi t/T}$$

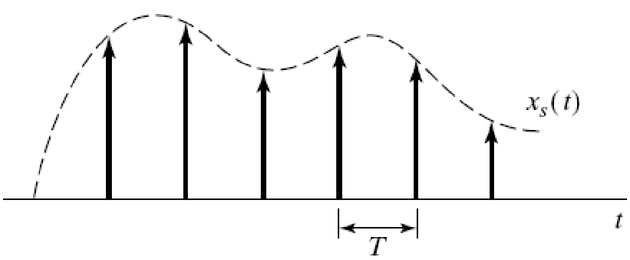


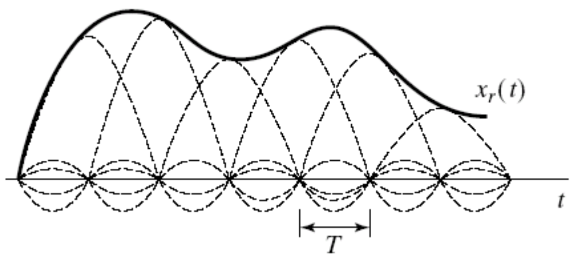

Since $X_r(j\Omega) = X_s(j\Omega)H_r(j\Omega)$,

$$x_r(t) = x_s(t) * h_r(t)$$

$$= \sum_{n=-\infty}^{\infty} x[n]\delta(t-nT) * h_r(t) = \sum_{n=-\infty}^{\infty} x[n]h_r(t-nT) = \sum_{n=-\infty}^{\infty} x[n]\frac{\sin(\pi(t-nT)/T)}{\pi(t-nT)/T}$$

The reconstruction is an interpolation.




- Sampling rate reduction by an integer factor (Decimation) $\boxed{\downarrow M}$

Sampling rate compressor: If $x_d[n] = x[Mn]$, where $x[n] = x_c(nT)$ is obtained by sampling $x_c(t)$ with period $T$, what is the relationship between $X_d(e^{jw})$ and $X(e^{jw})$?

Taking a simple example that $M = 2$,

$$X_d(e^{jw}) = \sum_{n=-\infty}^{\infty} x_d[n]e^{-jwn} = \sum_{n=-\infty}^{\infty} x[2n]e^{-jwn} = \sum_{m\in even} x[m]e^{-jwm/2}$$

$$= \sum_{m=-\infty}^{\infty} \frac{1}{2}\{x[m] + (-1)^m x[m]\}e^{-jwm/2} = \frac{1}{2}X(e^{jw/2}) + \frac{1}{2}X(e^{j(w/2-\pi)}).$$

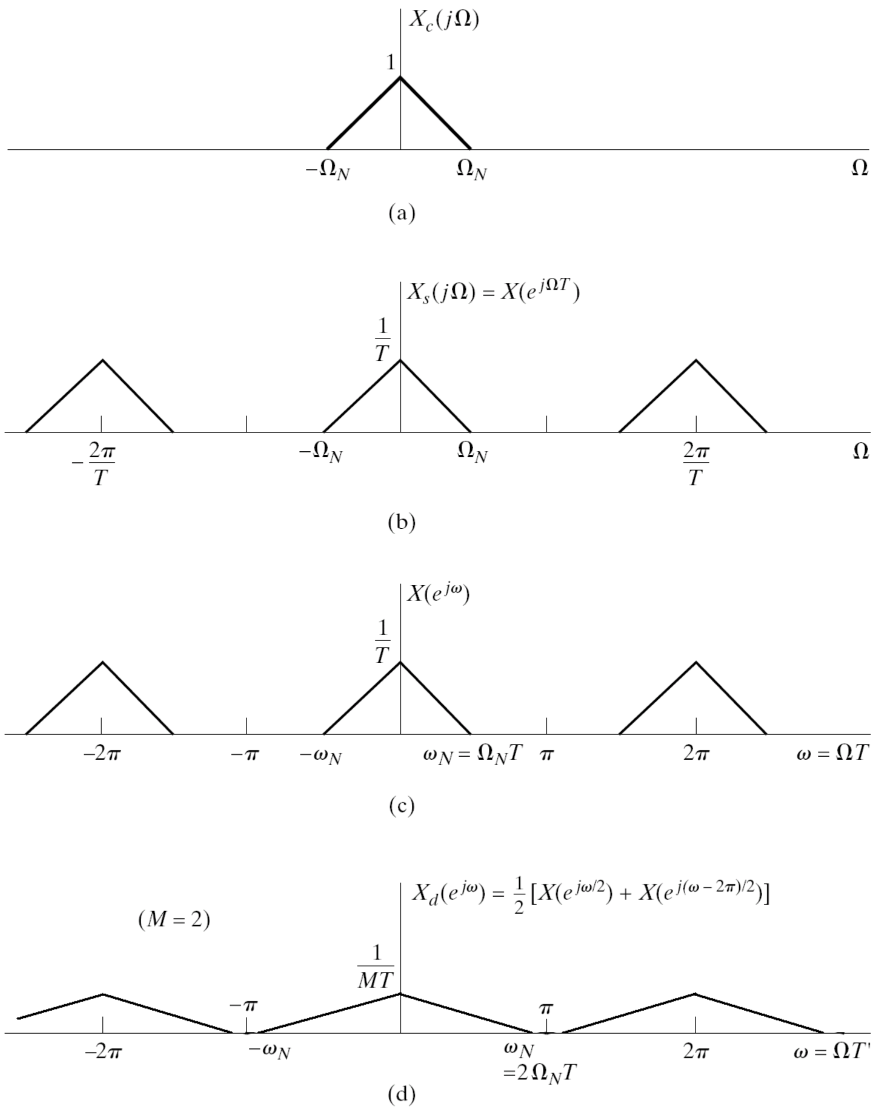
Note: $(-1)^m = e^{-j\pi m}$.

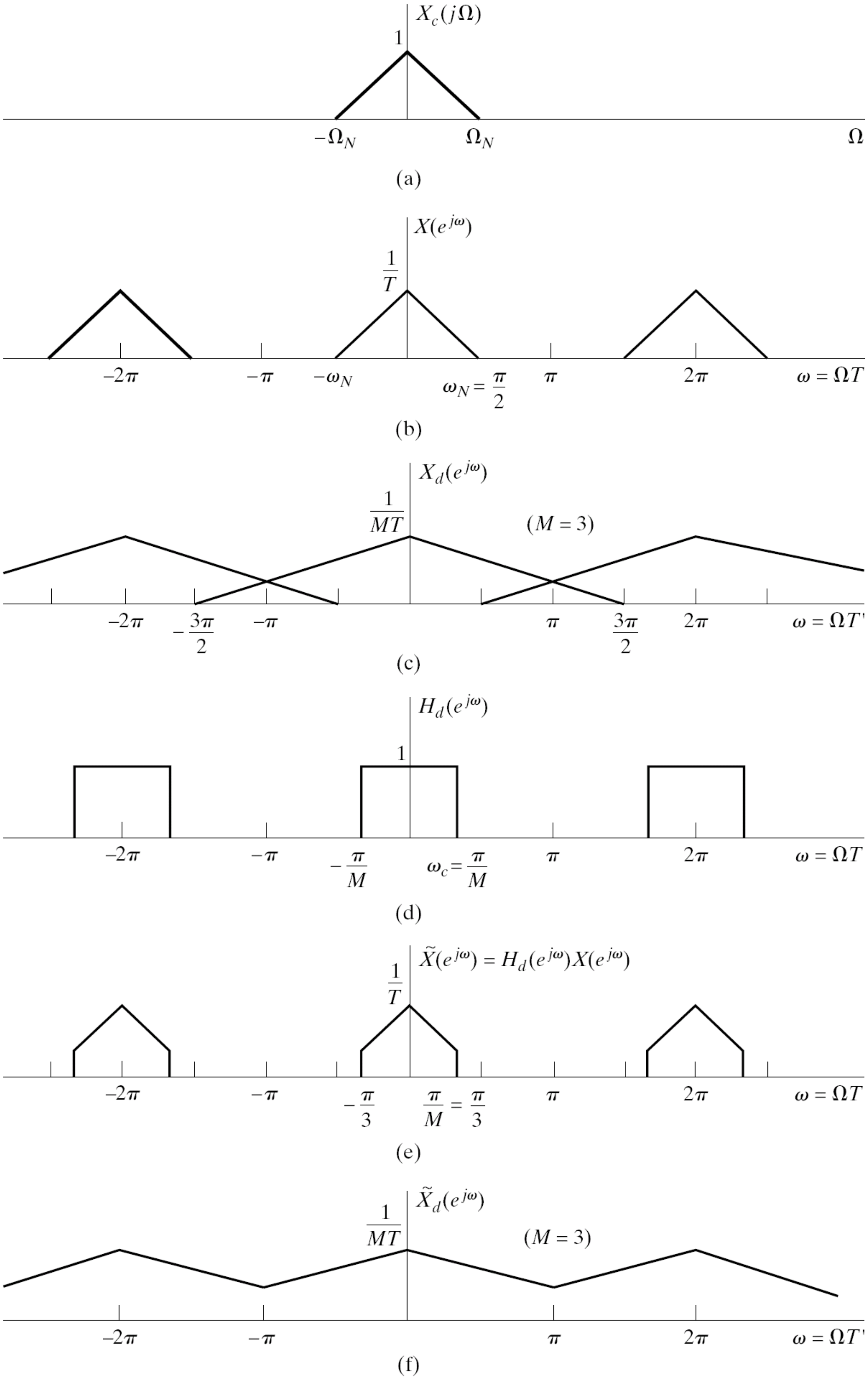
Now consider $x_d[n] = x[Mn]$,

$$X_d(e^{jw}) = \sum_{n=-\infty}^{\infty} x[Mn]e^{-jwn} = \sum_{m=0,\pm M,\pm 2M\ldots} x[m]e^{-jwm/M} = \sum_{m=-\infty}^{\infty} \frac{1}{M}\left\{\sum_{k=0}^{M-1} e^{j2\pi km/M} x[m]\right\}e^{-jwm/M}$$

$$= \frac{1}{M}\sum_{k=0}^{M-1} X(e^{j(w/M-2\pi k/M)}).$$

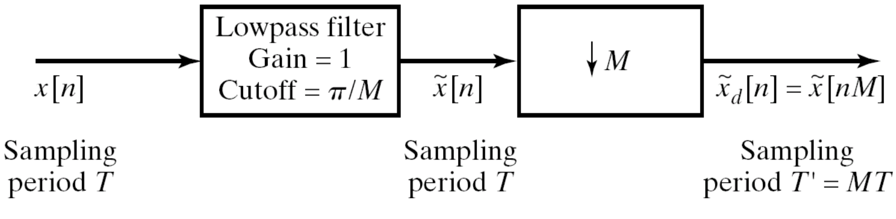
$$\boxed{\sum_{k=0}^{M-1} e^{j2\pi km/M} = \begin{cases} M & ,m = Mr \\ \dfrac{1-e^{j2\pi m}}{1-e^{j2\pi m/M}} = 0, & m \neq Mr \end{cases}}$$






(Alaising occurs due to the decimation)

Procedure of downsampling:

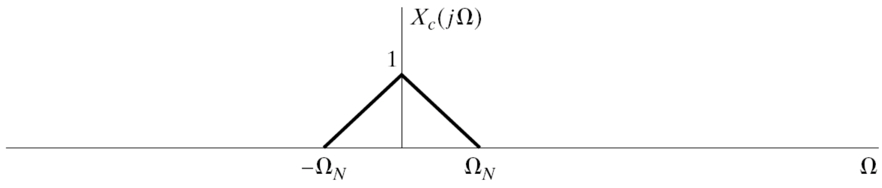


- Sampling rate increase by an integer factor (Interpolation)

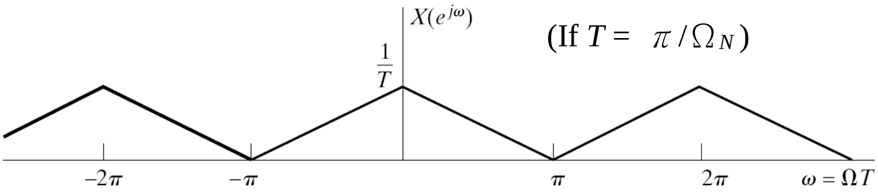
Sampling rate expander:

If $x_e[n]=\begin{cases} x[n/L], n=0,\pm L,\pm 2L,\ldots \\ 0 \ , \quad \text{otherwise} \end{cases}$ $\qquad \boxed{\uparrow L}$

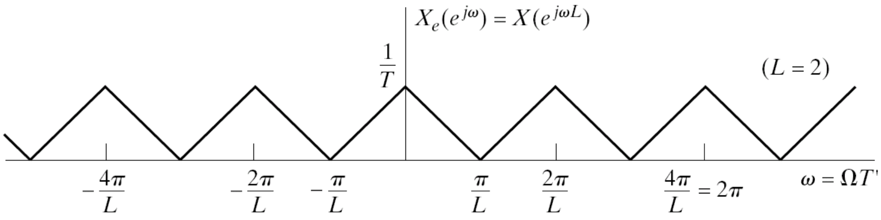
$$X_e(e^{jw})=\sum_{n=0,\pm L,\pm 2L\ldots}^{\infty} x[n/L]e^{-jwn}=\sum_{m=-\infty}^{\infty}x[m]e^{-jwmL}=X(e^{jwL}).$$



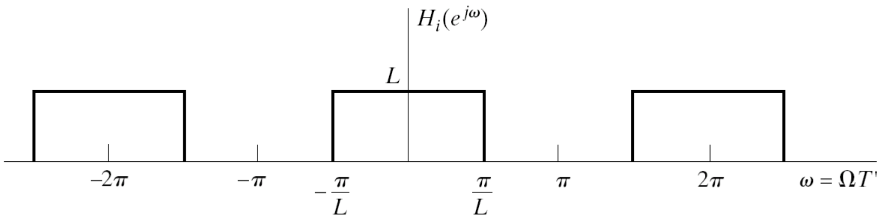

(a)



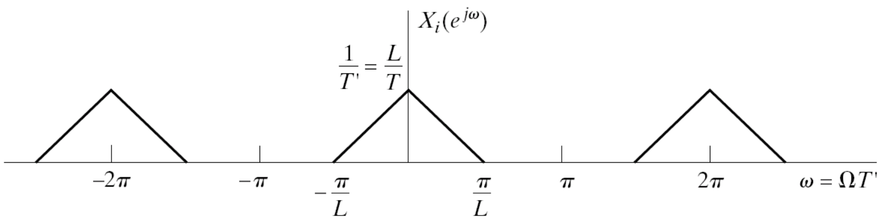

(b)




(c)

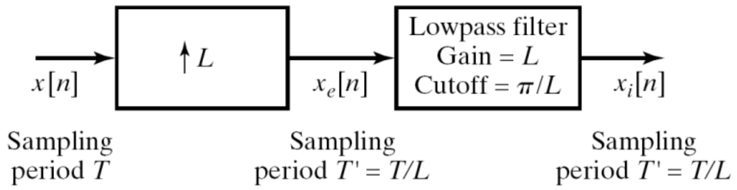



(d)




(e)

To obtain a signal $x_i[n]=x_c(nT')$, where $T'=T/L$, we need $H_e(e^{jw})=\begin{cases} L, |w|<\pi/L \\ 0, \text{otherwise} \end{cases}$



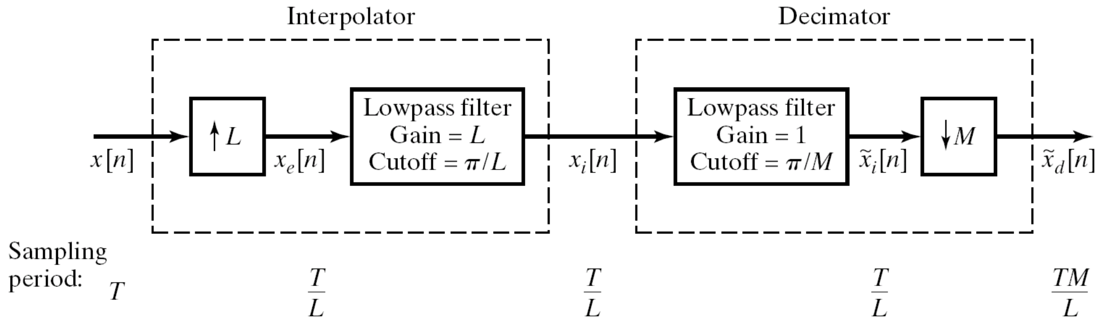

$X_i(e^{jw}) = X_e(e^{jw})H_i(e^{jw})$.

$$x_i[n] = x_e[n] * h_i[n] = x_e[n] * \frac{\sin(\pi n / L)}{\pi n / L} = \sum_{m=0,\pm L,\pm 2L\ldots} x[m/L] \cdot \frac{\sin\left(\pi(n-m)/L\right)}{\pi(n-m)/L}$$

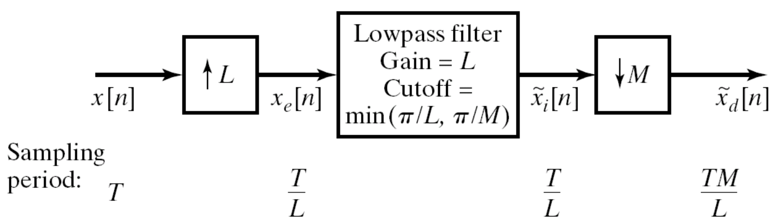
$$= \sum_{k=-\infty}^{\infty} x[k] \cdot \frac{\sin\left(\pi(n-kL)/L\right)}{\pi(n-kL)/L}$$

- Sampling rate change by a non-integer factor *k*

If *k* is rational, i.e., $k = M/L$, the change of sampling rate can be done by an interpolation followed by a decimation.




(a)




(b)

If *k* is not rational, the change of sampling rate must be done by converting the discrete-time signals into the continuous signals and then sampling the continuous signals again.

[Matlab Example 5-1] Record an audio signal using Matlab functions, in which the sampling frequency is set as 16000Hz. The recorded audio is saved as a binary file. We can then read the file and plot the waveform and spectrogram.

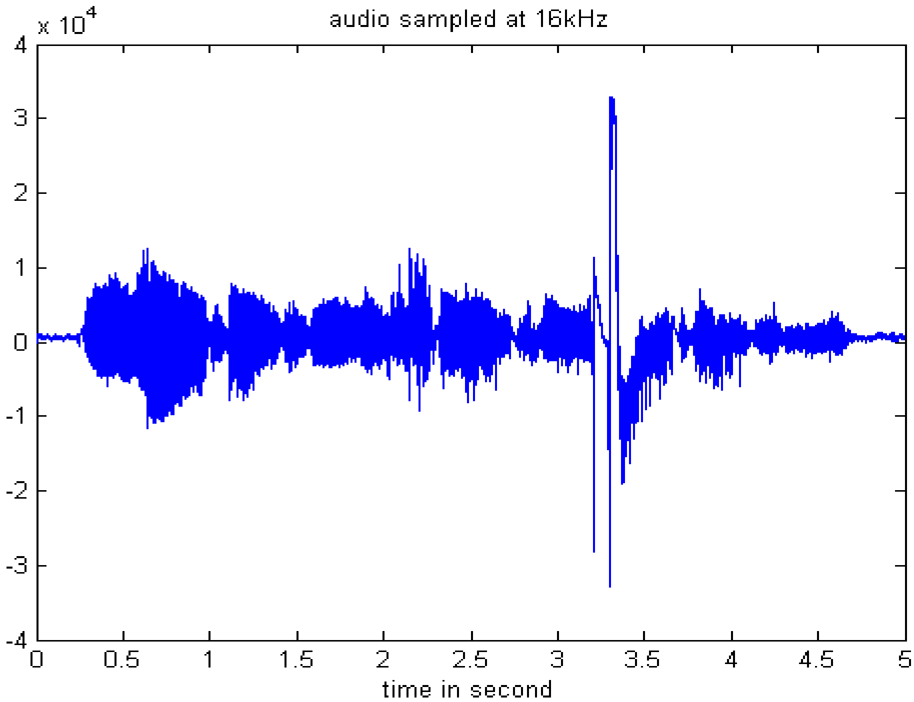



Fig. 5-1-1

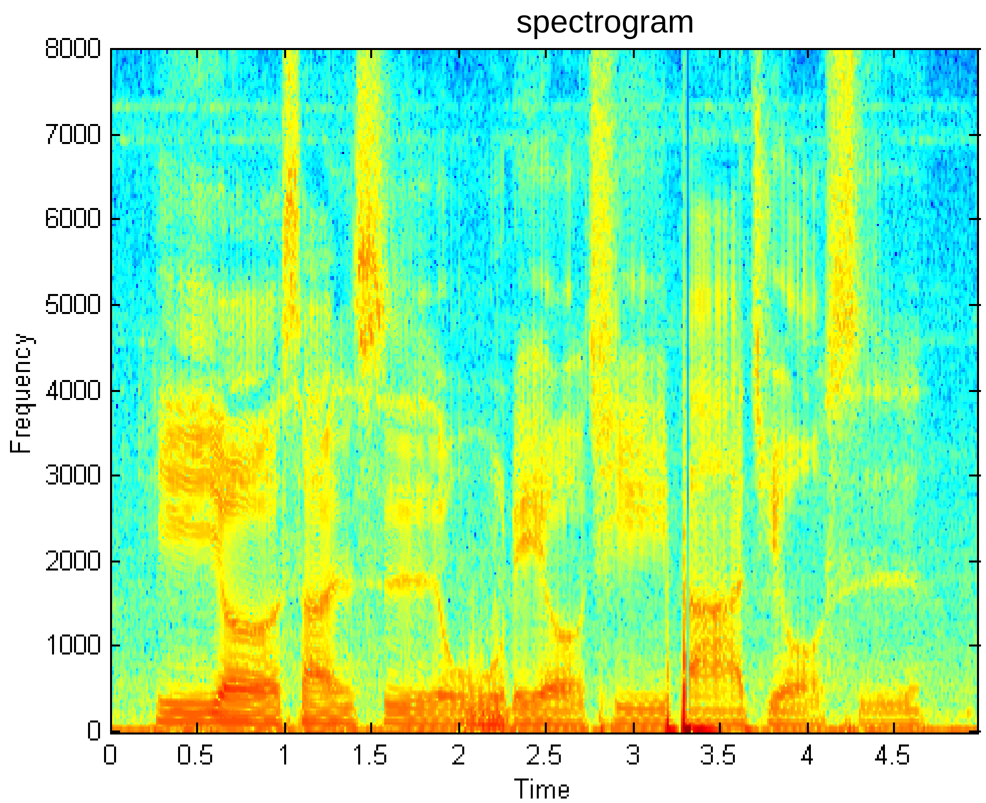



Fig. 5-1-2

Code Ex_5_1.m

```
% Record and play back 5 seconds of 16-bit audio sampled at 16000 kHz.
clear;
recObj = audiorecorder(16000, 16, 1);
```

```
disp('Start speaking.')
recordblocking(recObj, 5);
disp('End of Recording.');

% Play back the recording.
play(recObj);

% Store data in double-precision array.
x = getaudiodata(recObj,'int16');

% Plot the waveform.
plot(x);

%save the sound in a raw pcm file
fp=fopen('16kHz.pcm','wb');
fwrite(fp,x,'short');
fclose(fp);
```

Code Ex_5_2.m

```
% Read a pcm file.
clear;
fp=fopen('16kHz.pcm','rb');
x=fread(fp,'short');
fclose(fp);
% Read, plot, and play a wav file.
Fs=16000;
n=0:length(x)-1;
t=n/Fs;
plot(t,x);
xlabel('time in second')
title('audio sampled at 16kHz');
sound(x./32766,Fs,16)
specgram(x,512,Fs,320);
```

[Matlab Example5-2] Downsample and upsample via Matlab functions. Note that in Matlab, function `downsample` means the decimation (↓), and function `upsample` means the sampling rate expansion (↑). And, function `decimate` means the downsampling (Lowpass filtering + decimation), and function `interp` means the upsampling (sampling rate expansion + Lowpass filtering).

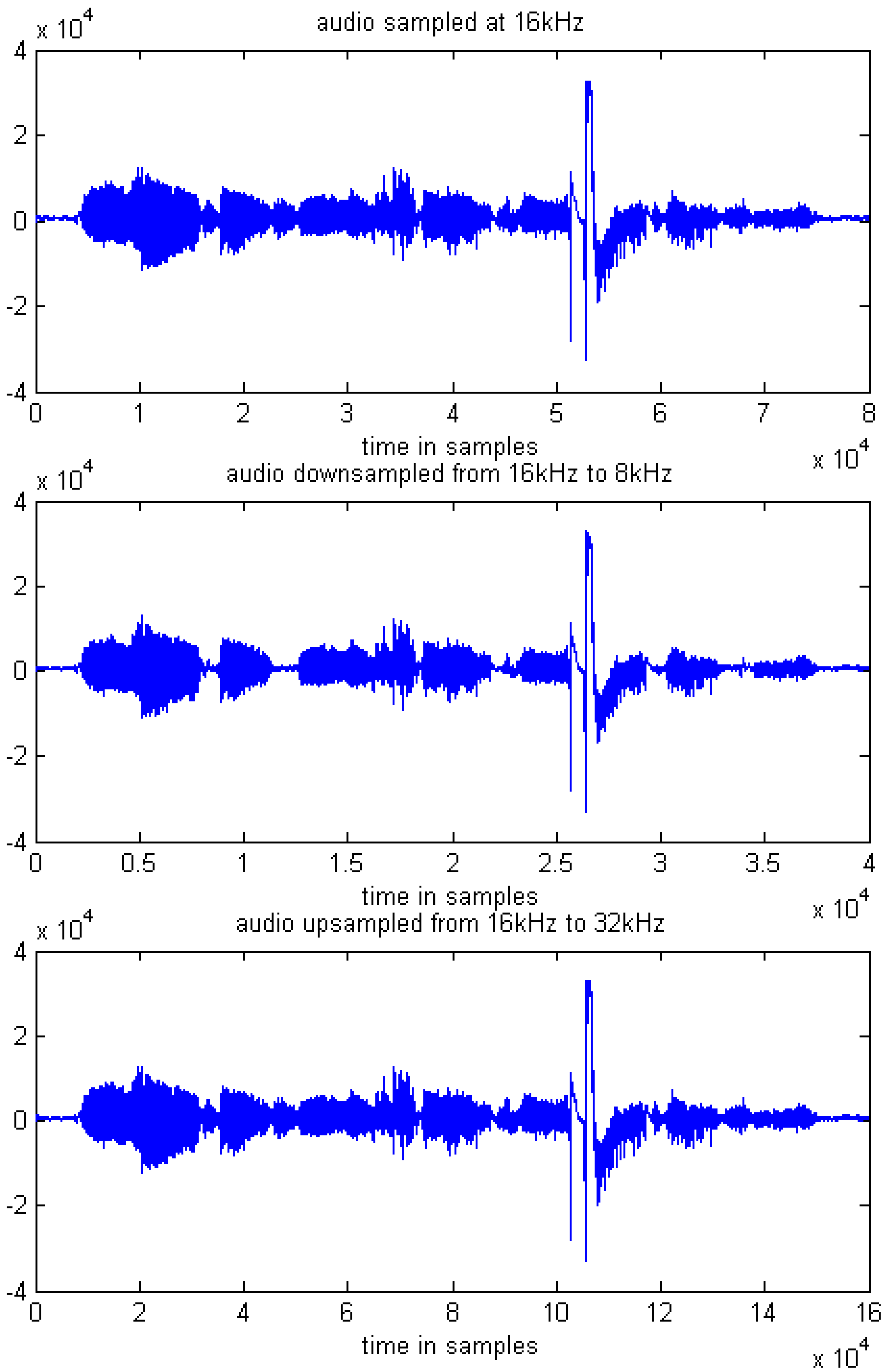

x 10^4
audio sampled at 16kHz
4
2
0
-2
-4
0
1
2
3
4
5
6
7
8
time in samples
x 10^4
x 10^4
audio downsampled from 16kHz to 8kHz
4
2
0
-2
-4
0
0.5
1
1.5
2
2.5
3
3.5
4
time in samples
x 10^4
x 10^4
audio upsampled from 16kHz to 32kHz
4
2
0
-2
-4
0
2
4
6
8
10
12
14
16
time in samples
x 10^4

Fig. 5-2-1

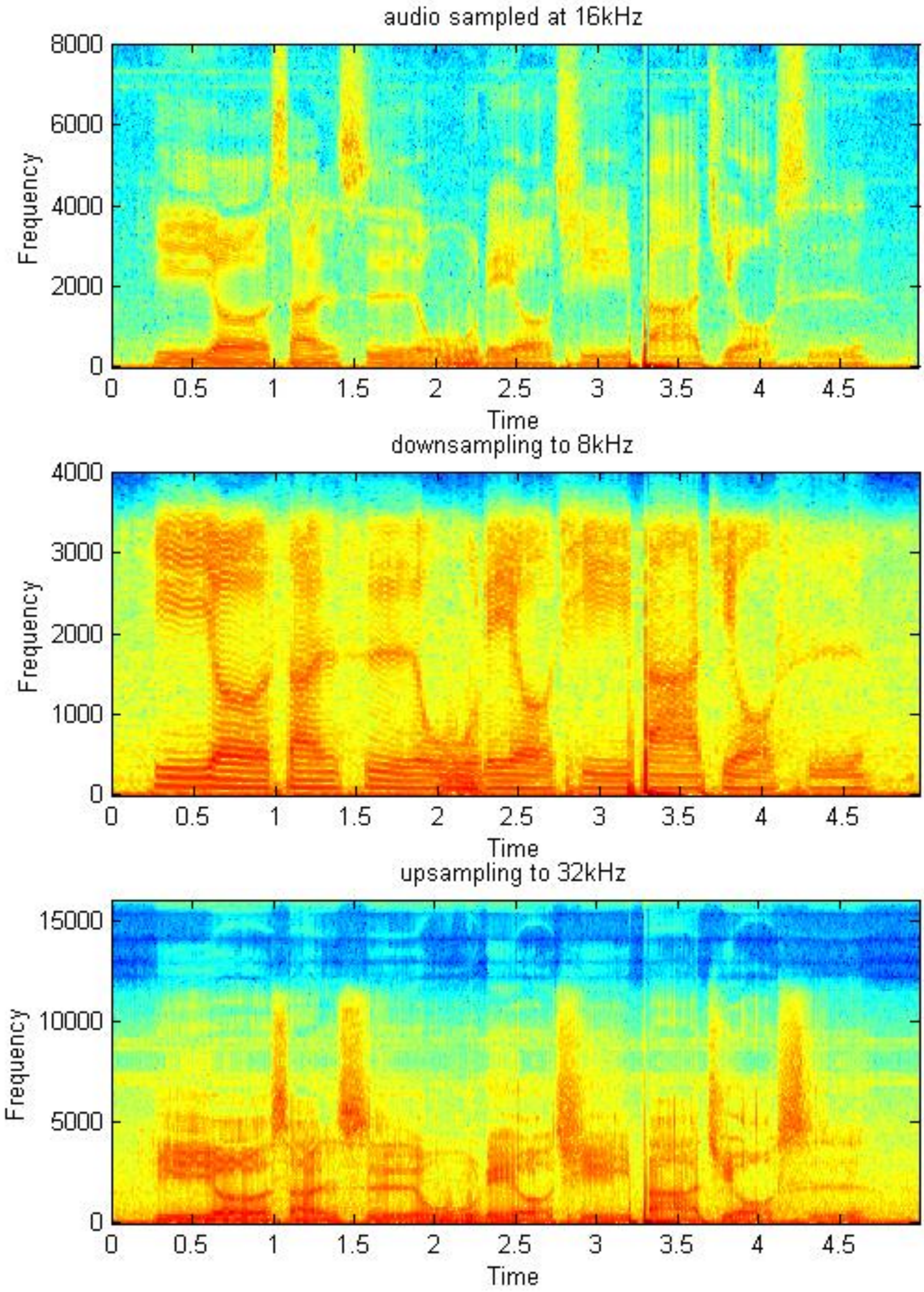

audio sampled at 16kHz
Frequency
Time
downsampling to 8kHz
Frequency
Time
upsampling to 32kHz
Frequency
Time

Fig. 5-2-2

Code Ex_5_3.m

```
% downsampling & upsampling
clear;
fp=fopen('16kHz.pcm','rb');
x=fread(fp,'short');
Fs=16000;
subplot(3,1,1); plot(x);
xlabel('time in samples')
title('audio sampled at 16kHz');
sound(x./32766,Fs,16)
pause;
prompt='press any key'

% downsampling
y=decimate(x,2);
subplot(3,1,2); plot(y);
xlabel('time in samples')
title('audio downsampled from 16kHz to 8kHz');
sound(y./32766,Fs/2,16)
pause;
prompt='press any key'

% upsampling
z=interp(x,2);
subplot(3,1,3); plot(z);
xlabel('time in samples')
title('audio upsampled from 16kHz to 32kHz');
sound(z./32766,Fs*2,16)
prompt='press any key'

subplot(3,1,1); specgram(x,512,Fs,320);
title('audio sampled at 16kHz');
subplot(3,1,2); specgram(y,512,Fs/2,320);
title('downsampling to 8kHz');
subplot(3,1,3); specgram(z,512,Fs*2,320);
title('upsampling to 32kHz');
```

*******************************************************************

[Practice 5-1] Perform ↑4 and then ↓3 of an audio signal without using the Matlab functions. Plot the spectrogram of the resulting signal.

*******************************************************************

*******************************************************************

[Practice 5-2] Generate a music with meoldy: So Mi Mi Fa Re Re Do Re Mi Fa So So So ; So Mi Mi Fa Re Re Do Mi So So Do, via Matlab function `sound.m` at sampling frequency of 8000Hz.

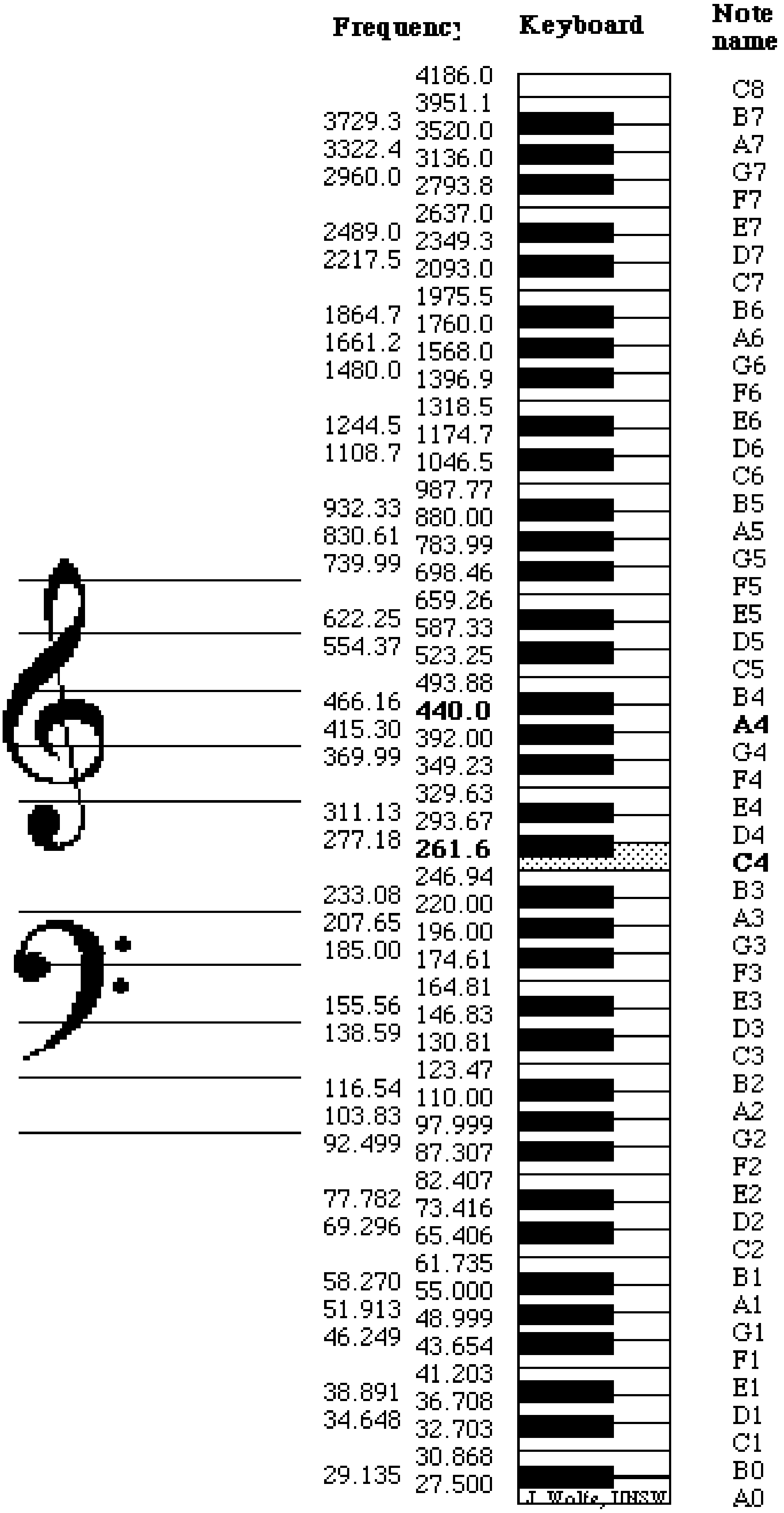



Fig. 5-3

*******************************************************************

# Lab 6: Filter Design

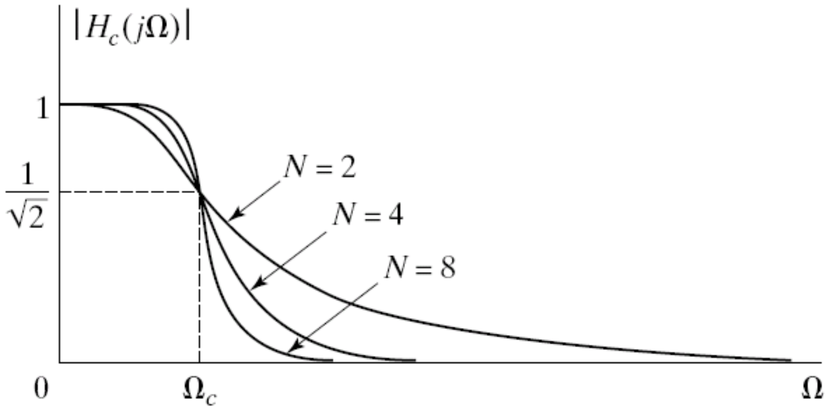
## [Theoretical Background]

- Typical analog filters

(1) Butterworth lowpass filter

$$|H_c(j\Omega)|^2 = \frac{1}{1+(j\Omega/j\Omega_c)^{2N}},$$

where $N$ is filter order, and $\Omega_c$ is the 3-dB frequency, i.e., magnitude is $1/\sqrt{2}$.




Poles of the filter:

Let $s = j\Omega$. Then, $H_c(s)H_c(-s) = \dfrac{1}{1+(s/j\Omega_c)^{2N}}$

By letting $H_c(s)H_c(-s) = \infty$, we have $1+(s/j\Omega_c)^{2N} = 0$.

Thus, $(s/j\Omega_c)^{2N} = -1 = e^{\pm j\pi} = e^{\pm j\pi + 2k\pi}$, where $k$ is an integer.

The roots of $H_c(s)H_c(-s) = \infty$ are

$$s_k = j\Omega_c e^{j(2k\pi \pm \pi)/2N} = \Omega_c e^{j\pi/2} e^{j(2k\pi\pm\pi)/2N} = \Omega_c e^{(j\pi/2N)(2k+N\pm 1)},$$

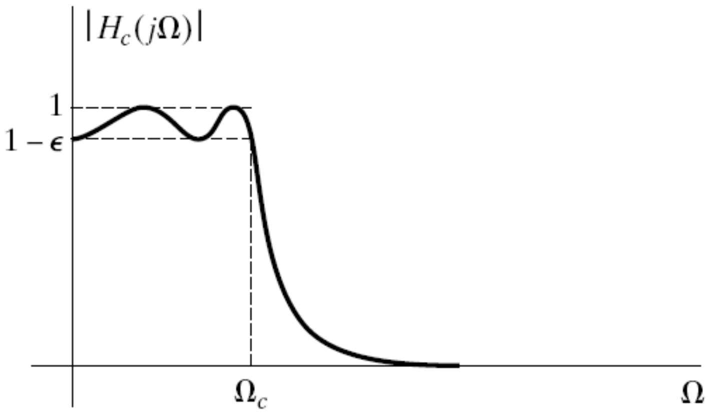
where $k$ = 0,1,…, (2$N$-1) (there are 2$N$ roots)

If $H_c(s)$ is stable, the ploes of $H_c(s)$ must lie in the left side of s-plane, i.e., the real part of each pole of $H_c(s)$ is negative.

Hence, $H_c(s) = A \prod_{\forall k, \mathrm{Re}(s_k)<0} \dfrac{1}{s-s_k}$, where $A$ is a constant making $|H_c(0)| = 1$.

(2) Chebyshev lowpass filter

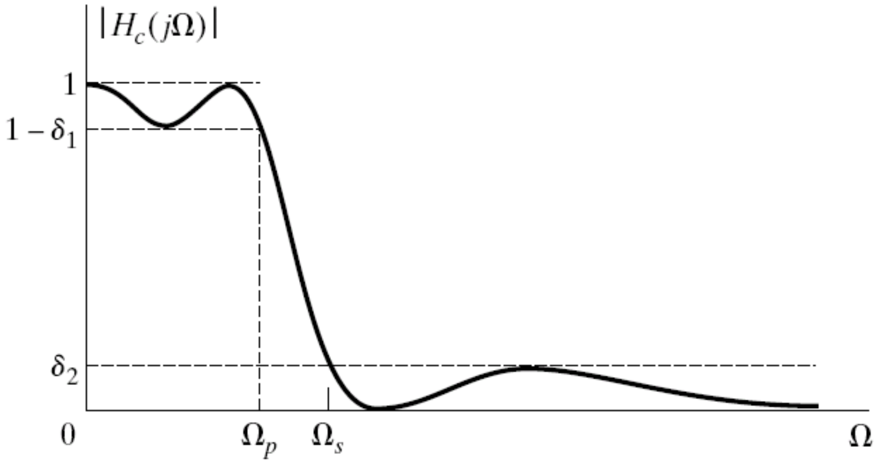
$$|H_c(j\Omega)|^2 = \frac{1}{1+\varepsilon^2 V_N^2(\Omega/\Omega_c)}, \text{ where } V_N(x) = \cos(N\cos^{-1} x)$$




Note: Chebyshev lowpass filter has a sharper transition band than Butterworth filter does, but it has ripples in the passband.

(3) Elliptic lowpass filter

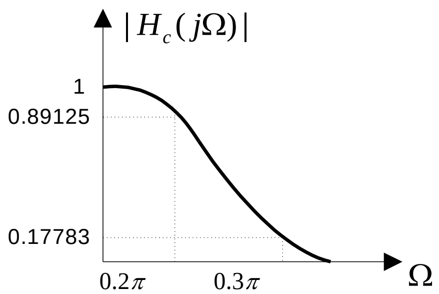
$|H_c(j\Omega)|^2 = \dfrac{1}{1+\varepsilon^2 U_N^2(\Omega/\Omega_p)}$, where $U_N()$ is a Jacobian elliptic function




Note: Elliptic lowpass filter has the smallest ($\Omega_s$ - $\Omega_p$), compared to those of the Butterworth lowpass filter and Chebyshev lowpass filter.

Example: Design an analog Butterworth lowpass filter with the following specification:

-1dB at $0.2\pi$ rad/sec, and -15dB at $0.3\pi$ rad/sec




(-1dB = 20 log 0.89125)

(-15dB = 20 log 0.17783)

The task is to find $\Omega_c$ and $N$ that fit the specification.

Since $|H_c(j\Omega)|^2 = \dfrac{1}{1+(j\Omega/j\Omega_c)^{2N}}$, we have

$$\begin{cases} 1+\left(\dfrac{0.2\pi}{\Omega_c}\right)^{2N} = \left(\dfrac{1}{0.89125}\right)^2 \\ 1+\left(\dfrac{0.3\pi}{\Omega_c}\right)^{2N} = \left(\dfrac{1}{0.17783}\right)^2 \end{cases} \Rightarrow \begin{cases} \left(\dfrac{0.2\pi}{\Omega_c}\right)^{2N} = 0.2589 & (1) \\ \left(\dfrac{0.3\pi}{\Omega_c}\right)^{2N} = 30.622 & (2) \end{cases}$$

Dividing (1) by (2), we can solve $N = 5.8858$.

Taking an integer $N = 6$, we obtain $\Omega_c = 0.7032$ rad/sec.

Thus, the roots of $H_c(s)H_c(s) = \infty$ are

$$s_k = \Omega_c e^{(j\pi/2N)(2k+N-1)} = 0.7032\, e^{j\frac{\pi}{12}(2k+5)},\ k = 0,1,\ldots 11.$$

The poles of $H_c(s)$ are those $s_k$ with negative real part:

$-0.182 \pm j(0.679)$, $-0.497 \pm j(0.497)$, $-0.679 \pm j(0.182)$

Thus, $H_c(s) =$

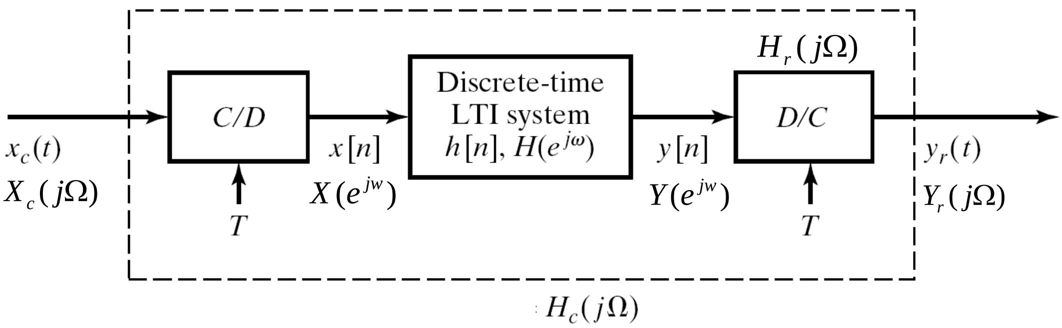
$$A\frac{1}{(s+0.182)^2+(0.679)^2}\,\frac{1}{(s+0.497)^2+(0.497)^2}\,\frac{1}{(s+0.679)^2+(0.182)^2},$$

where $A$ can be computed by letting $|H_c(0)| = 1$.

We have

$$H_c(s) = \frac{0.12093}{(s^2+0.364s+0.495)(s^2+0.995s+0.495)(s^2+1.359s+0.495)}$$

- Transformation of an analog filter to digital filter




We want to find the relationship between $H_c(j\Omega)$ and $H(e^{jw})$, so that $H(e^{jw})$ can be designed by a transformation of $H_c(j\Omega)$.

$$X(e^{jw}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left(j\frac{w}{T} - j\frac{2\pi}{T}k\right)$$

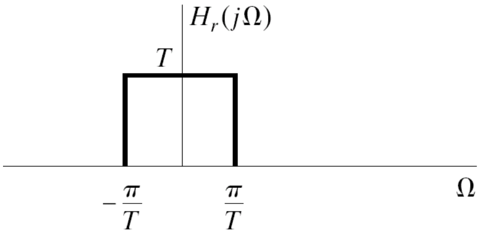
$Y(e^{jw}) = H(e^{jw})\, X(e^{jw})$

$$Y_r(j\Omega) = H_r(j\Omega)\, Y(e^{j\Omega T}) = H_r(j\Omega)\, H(e^{j\Omega T}) \cdot \frac{1}{T} \sum_{k=-\infty}^{\infty} X_c\left(j\frac{w}{T} - j\frac{2\pi}{T}k\right)$$

$$= \begin{cases} H(e^{j\Omega T}) X_c(j\Omega), & |\Omega| \le \dfrac{\pi}{T} \\ 0, & \text{otherwise} \end{cases} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\text{(a)}$$

Recall that

$$H_r(j\Omega) = \begin{cases} T, & |\Omega| \le \dfrac{\pi}{T} \\ 0, & \text{otherwise} \end{cases}$$




In the continuous-time domain, $Y_r(j\Omega) = H_c(j\Omega)\, X_c(j\Omega)$ ……………………….(b)

Comparing Eqs. (a) and (b), we have

$H_c(j\Omega) = H(e^{j\Omega T})$, $|\Omega| \le \frac{\pi}{T}$. Or equivalently, $H(e^{jw}) = H_c(j\frac{w}{T})$, $|w| \le \pi$.

According to the sampling theory, if a discrete-time signal $d[n]$ is sampled from an analog signal $d_c(t)$, with sampling period $T$, then $d[n] = d_c(nT)$, and

$D(e^{jw}) = \frac{1}{T} D_c(j\frac{w}{T})$, $|w| \le \pi$.

Since $H(e^{jw}) = H_c(j\frac{w}{T})$, $|w| \le \pi$, we have $h[n] = T\, h_c(nT)$. This is the so-called *impulse invariance*

Let $H_c(s)$ be the frequency response of a well-designed filter.

Assume $H_c(s)$ is of the form:

$$H_c(s) = A \prod_{\forall k, \mathrm{Re}(s_k)<0} \frac{1}{s - s_k} = \sum_{k=1}^{K} \frac{B_k}{s - s_k}.$$ (by a partial fraction expansion)

Taking the inverse Laplace transform, we have the impulse response of $H_c(s)$ as

$$h_c(t) = \sum_{k=1}^{K} B_k e^{s_k t} u(t).$$

By using the impulse invariance, a digital filter corresponding to $h_c(t)$ is

$$h[n] = T h_c(nT) = \sum_{k=1}^{K} B_k T e^{s_k n T} u[n] = \sum_{k=1}^{K} B_k T (e^{s_k T})^n u[n]$$

By taking the z-transform, we have

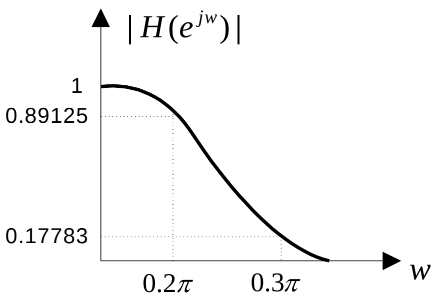
$$H(z) = \sum_{k=1}^{K} \frac{B_k T}{1 - e^{s_k T} z^{-1}}.$$

And, we note that

$$H_c(s) = \frac{1}{s-a} \xleftarrow{} \underline{\text{Transformation via the impulse invariance}} \rightarrow H(z) = \frac{T}{1 - e^{aT} z^{-1}}$$

Note: impulse invariance is only valid for the band-limited system. This is because if analog filter $H_c(j\Omega)$ is not band-limited, there must be aliasing in the resulting digital filter $H(e^{jw})$. However, in practice, most analog filters are not band-limited.

Example: Using a Butterworth lowpass filter to design a digital lowpass filter with the following specification:




$0.89125 \le |H(e^{jw})| \le 1, \quad 0 \le |w| \le 0.2\pi$

$|H(e^{jw})| \le 0.17783, \quad 0.3\pi \le |w| \le \pi$

Step 1: converting the specification to the analog domain by impulse invariance:

$0.89125 \le |H_c(j\Omega)| \le 1, \quad 0 \le |\Omega| \le 0.2\pi / T$

$|H_c(j\Omega)| \le 0.17783, \quad 0.3\pi / T \le |\Omega| \le \pi / T$

Step 2: designing a Butterworth filter by determining a proper $\Omega_c$ and $N$ that fit the above specification.

By $\begin{cases} 1+\left(\dfrac{0.2\pi}{T\Omega_c}\right)^{2N} = \left(\dfrac{1}{0.89125}\right)^2 \\ 1+\left(\dfrac{0.3\pi}{T\Omega_c}\right)^{2N} = \left(\dfrac{1}{0.17783}\right)^2 \end{cases}$, we can solve $N$ = 5.8858.

Taking $N$ = 6, we obtain $T\Omega_c$ = 0.7032.

Thus, the roots of $H_c(s)H_c(s) = \infty$ are

$$s_k = \Omega_c e^{(j\pi/2N)(2k+N-1)} = \left(\frac{0.7032}{T}\right) e^{j\frac{\pi}{12}(2k+5)}, \; k = 0,1,\ldots 11.$$

The poles of $H_c(s)$ are those $s_k$ with negative real part:

$[-0.182 \pm j(0.679)] / T, \quad [-0.497 \pm j(0.497)] / T, \quad [-0.679 \pm j(0.182)] / T$

Step 3: the frequency response of the digital filter can be represented by

$$H(z) = \frac{TA_1}{1-e^{0.182+j(0.679)}z^{-1}} + \frac{TA_2}{1-e^{0.182-j(0.679)}z^{-1}} + \frac{TA_3}{1-e^{0.479+j(0.479)}z^{-1}} + \frac{TA_4}{1-e^{0.479-j(0.479)}z^{-1}} + \frac{TA_5}{1-e^{0.679+j(0.182)}z^{-1}} + \frac{TA_6}{1-e^{0.679-j(0.182)}z^{-1}}$$

where $A_1$, $A_2$, $A_3$, $A_4$, $A_5$, and $A_6$ are subject to $|H(e^{j0})| = 1$

[Matlab Example 6-1] Design a Butterworth lowpass digital filter, in which the cut-off frequency is set to 0.4π rad/sec. Note that if the signal is sampled at frequency of 100 Hz, then the cut-off frequency is 20 Hz. The filter is then applied to the signal in Practice 1-3. We can see that most of the 30Hz components are filtered out.

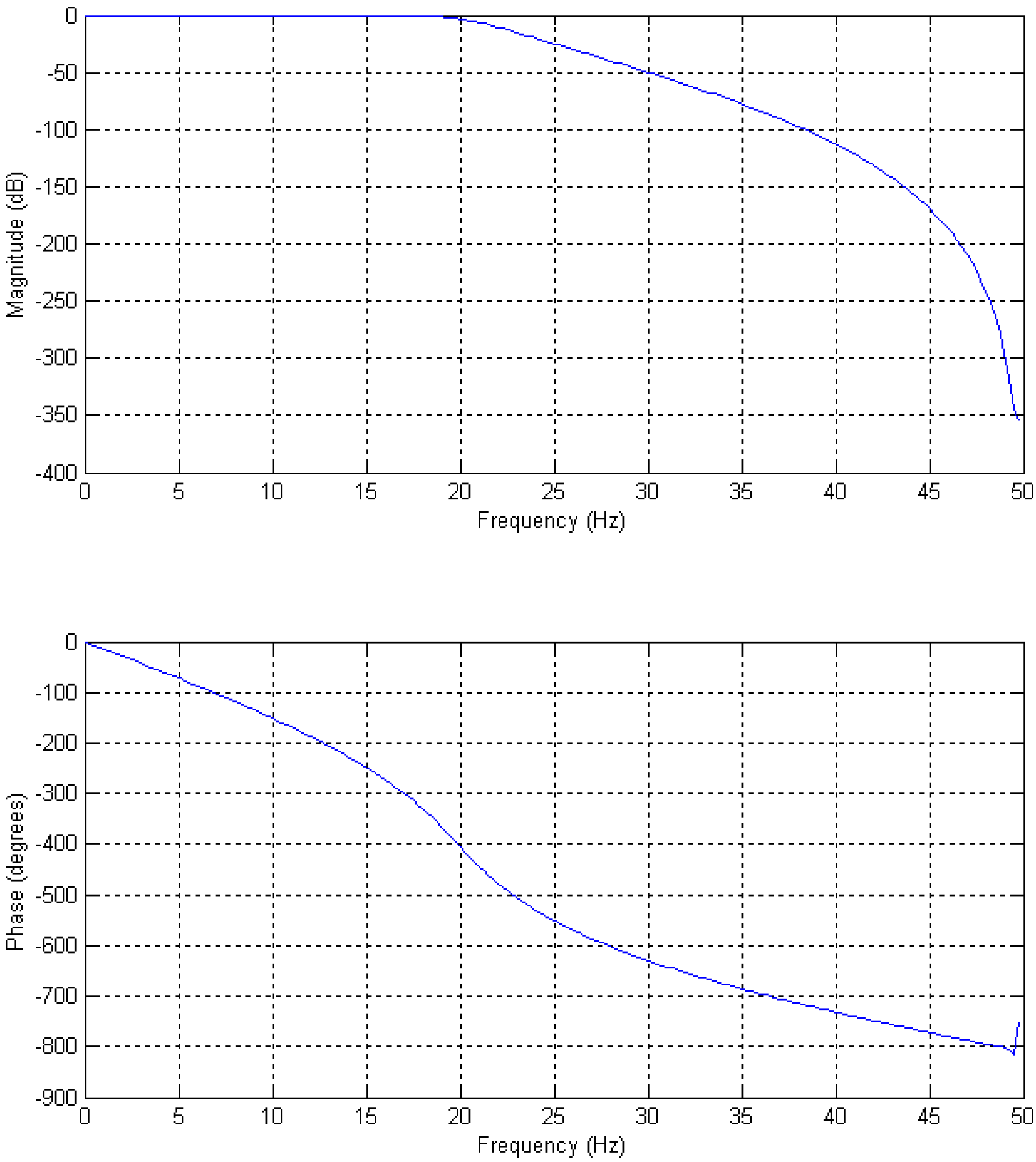

0
-50
-100
-150
-200
-250
-300
-350
-400
Magnitude (dB)
0
5
10
15
20
25
30
35
40
45
50
Frequency (Hz)
0
-100
-200
-300
-400
-500
-600
-700
-800
-900
Phase (degrees)
0
5
10
15
20
25
30
35
40
45
50
Frequency (Hz)

Fig. 6-1-1

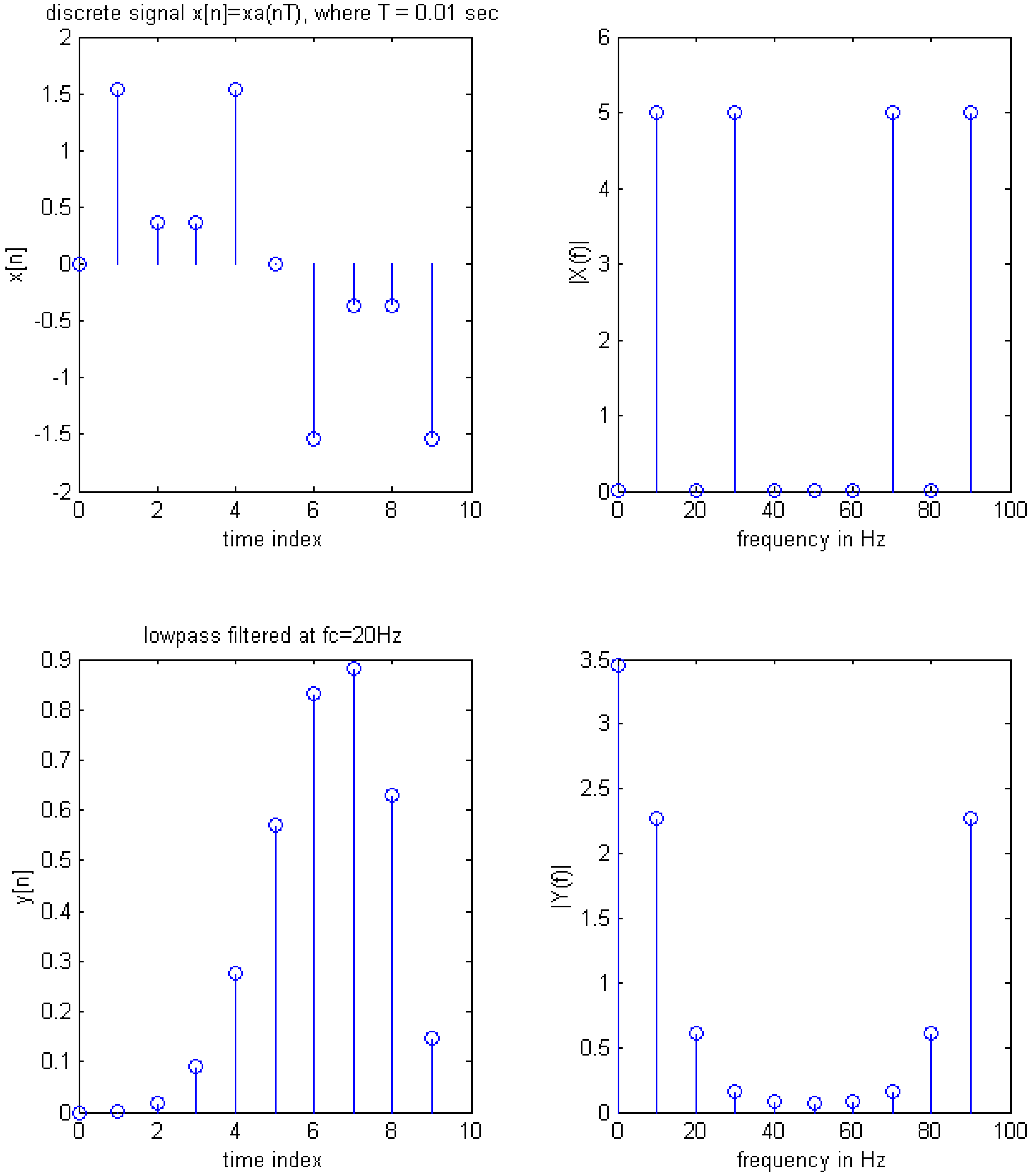

discrete signal x[n]=xa(nT), where T = 0.01 sec
x[n]
time index
|X(f)|
frequency in Hz
lowpass filtered at fc=20Hz
y[n]
time index
|Y(f)|
frequency in Hz

Fig. 6-1-2

Code Ex_6_1.m

```
% Butterworth lowpass filter
[b,a] = butter(9,0.4,'low');       % cut-off freq. = 0.4 pi = 20 Hz
freqz(b,a,200,100);
pause;

% signal x
f1=10;        % 10 Hz sine wave
f2=30;        % 30 Hz sine wave
T=0.01;       % sampling freq. = 100 Hz
N=10;
n=0:1:N-1;
x=sin(2*pi*f1*n*T)+sin(2*pi*f2*n*T);
subplot(2,2,1); stem(n,x);
xlabel('time index'); ylabel('x[n]');
title('discrete signal x[n]=xa(nT), where T = 0.01 sec');

% DFT of x
f=n/T/N;
subplot(2,2,2); stem(f,abs(fft(x)));
xlabel('frequency in Hz'); ylabel('|X(f)|');

% lowpass filtering
y=filter(b,a,x);
subplot(2,2,3); stem(n,y);
xlabel('time index'); ylabel('y[n]');
title('lowpass filtered at fc=20Hz');

% DFT of y
f=n/T/N;
subplot(2,2,4); stem(f,abs(fft(y)));
xlabel('frequency in Hz'); ylabel('|Y(f)|');
```

[Matlab Example 6-2] Design a Chebyshev lowpass digital filter, in which the cut-off frequency is set to 0.8π rad/sec. If the signal is sampled at 1000 Hz, then the cut-off frequency is 400 Hz. Now the filter is applied to a sine signal consisted of 10Hz and 300Hz. The resulting signal is then decimated by 2. We will see that aliasing occurs, because the value of sampling frequency (now is 500Hz) is not larger than the double value of the highest signal frequency (300Hz). In Fig. 6-2-2 (b), if we modify the cut-off frequency from 0.8π to 0.4π rad/sec (200 Hz), then no aliasing occurs after the decimation, since 300Hz component has been filtered out.

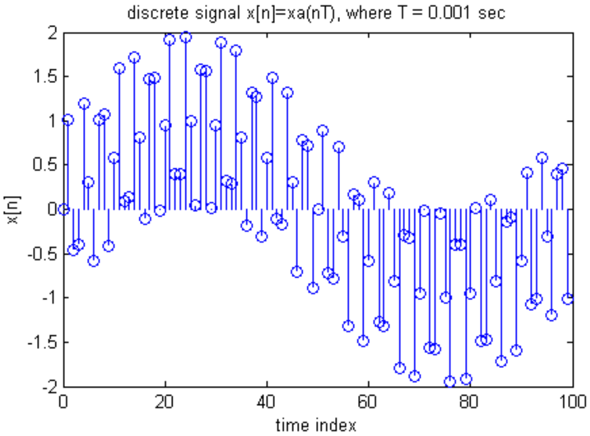


Fig. 6-2-1(a)

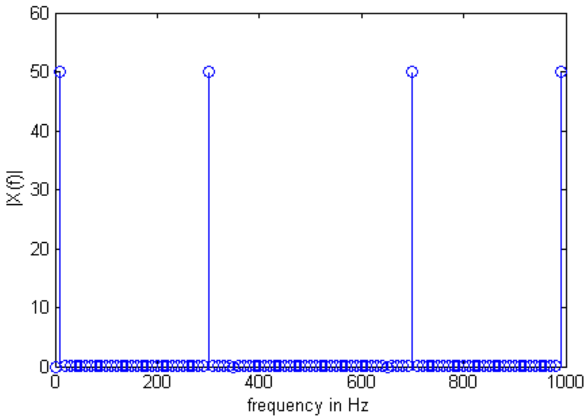


Fig. 6-2-1(b)

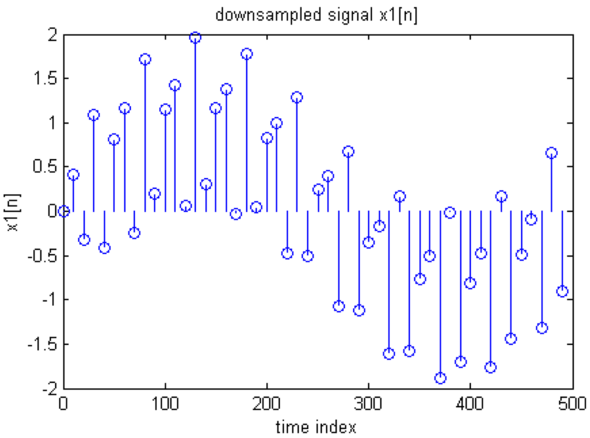


Fig. 6-2-2(a)

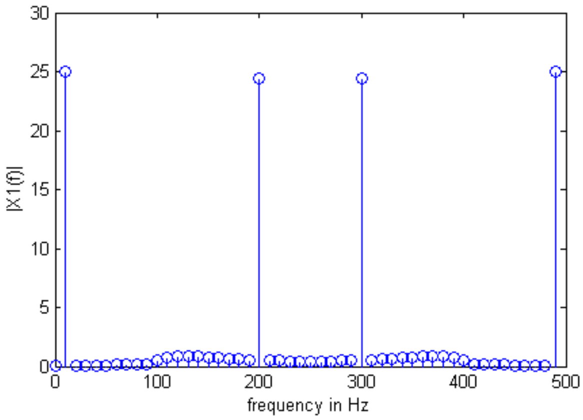


Fig. 6-2-2(b)

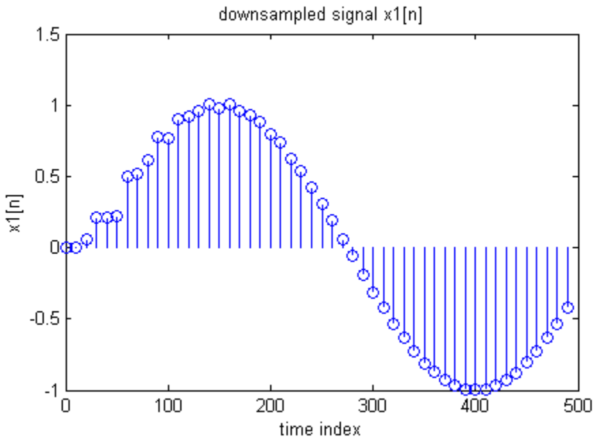


Fig. 6-2-3(a)

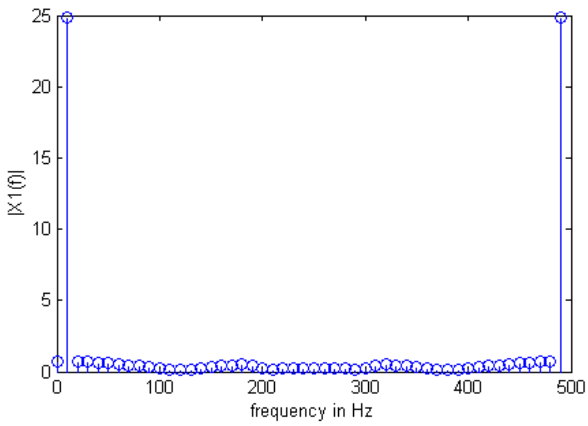
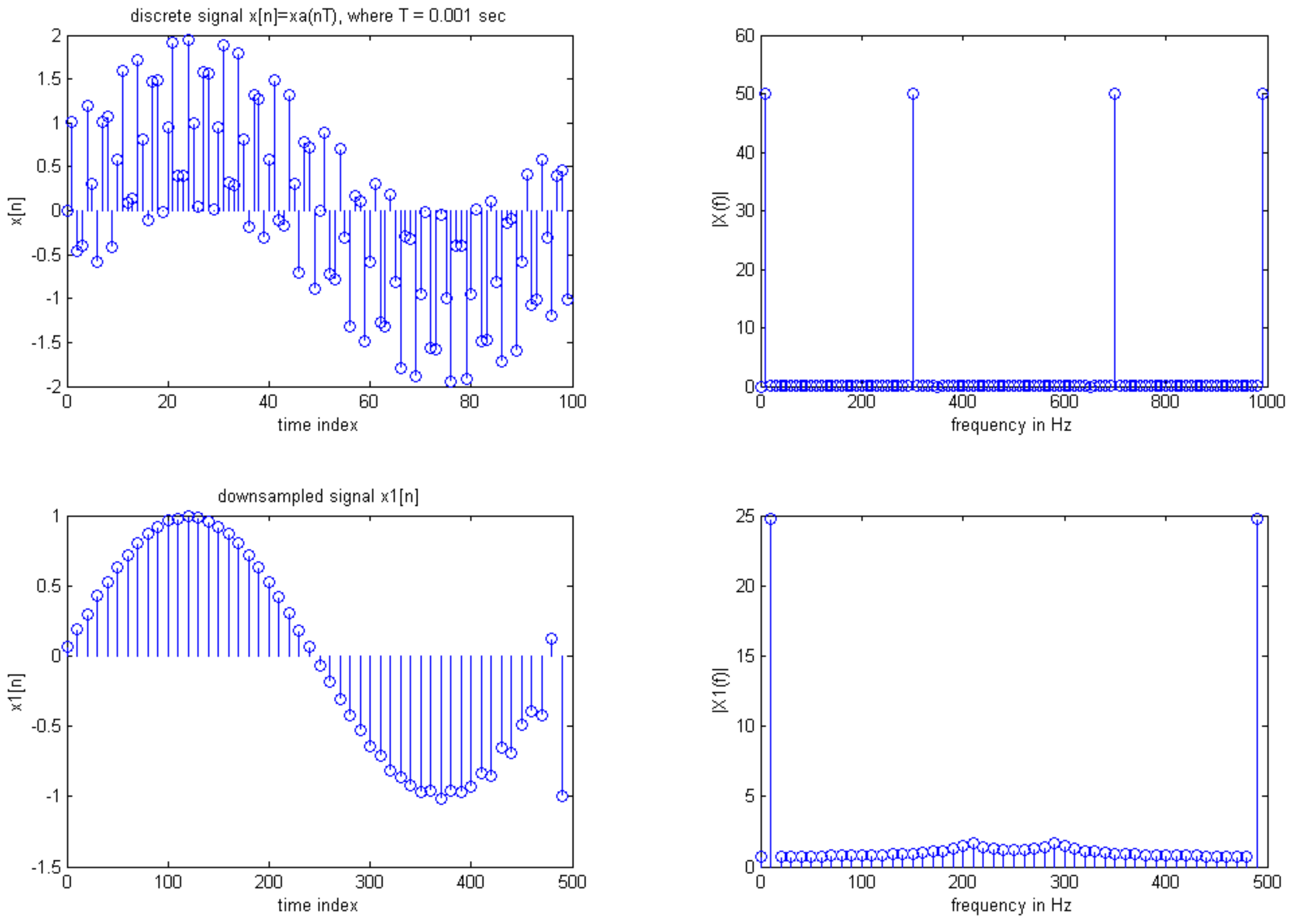


Fig. 6-2-3(b)

Code P6_2.m

```
% Chebyshev lowpass filter
[b,a] = cheby1(9,0.05,0.4);       % cut-off freq. = 0.8 pi = 400 Hz

% signal x
f1=10;         % 10 Hz sine wave
f2=300;         % 300 Hz sine wave
T=0.001;       % sampling freq. = 1000 Hz
N=100;
n=0:1:N-1;
x=sin(2*pi*f1*n*T)+sin(2*pi*f2*n*T);
subplot(2,2,1); stem(n,x);
xlabel('time index'); ylabel('x[n]');
title('discrete signal x[n]=xa(nT), where T = 0.001 sec');

% DFT of x
f=n/T/N;
subplot(2,2,2); stem(f,abs(fft(x)));
xlabel('frequency in Hz'); ylabel('|X(f)|');

% lowpass filtering & Decimation & DFT
y=filter(b,a,x);
z=downsample(y,2);
n2=0:1:N/2-1;
f=n2/(2*T)/(N/2);
subplot(2,2,3); stem(f,z);
xlabel('time index'); ylabel('x1[n]');
title('downsampled signal x1[n]');
subplot(2,2,4); stem(f,abs(fft(z)));
xlabel('frequency in Hz'); ylabel('|X1(f)|');
```

Example 6-2 can also be implemented using Matlab function decimate(), which is carried out by Code Ex_6_3.m




Code Ex_6_3.m

```
% downsampling & DFT
z=decimate(x,2);
n2=0:1:N/2-1;
f=n2/(2*T)/(N/2);
subplot(2,2,3); stem(f,z);
xlabel('time index'); ylabel('x1[n]');
title('downsampled signal x1[n]');
subplot(2,2,4); stem(f,abs(fft(z)));
xlabel('frequency in Hz'); ylabel('|X1(f)|');
```

*********************************************************************

[Practice 6-1] Design a Chebyshev lowpass digital filter using Matlab function upsample.m to perform upsampling of the signal in Example 6-2 by a factor 2. Sketch the resulting waveform and spectrogram.

*********************************************************************