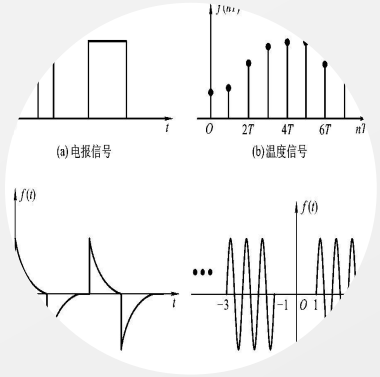
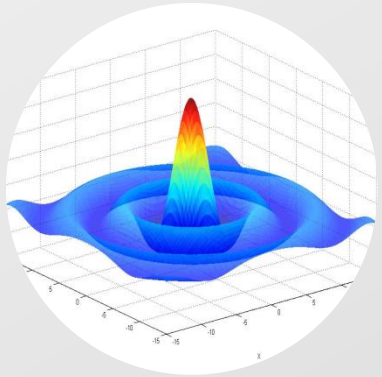
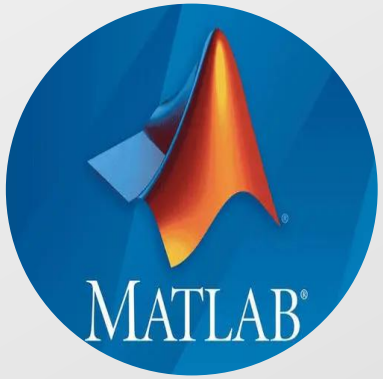
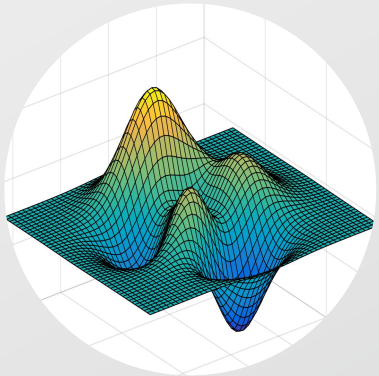
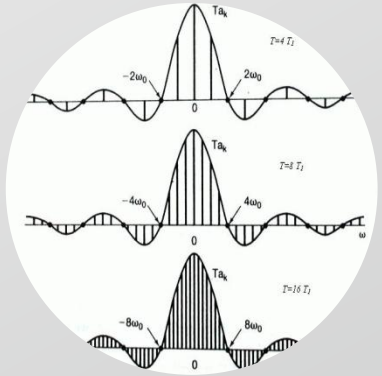










Lab3 Fourier Series Representation of Periodic Signals
周期信号的傅里叶级数表示

主讲老师：王小静
办公地点：一教131

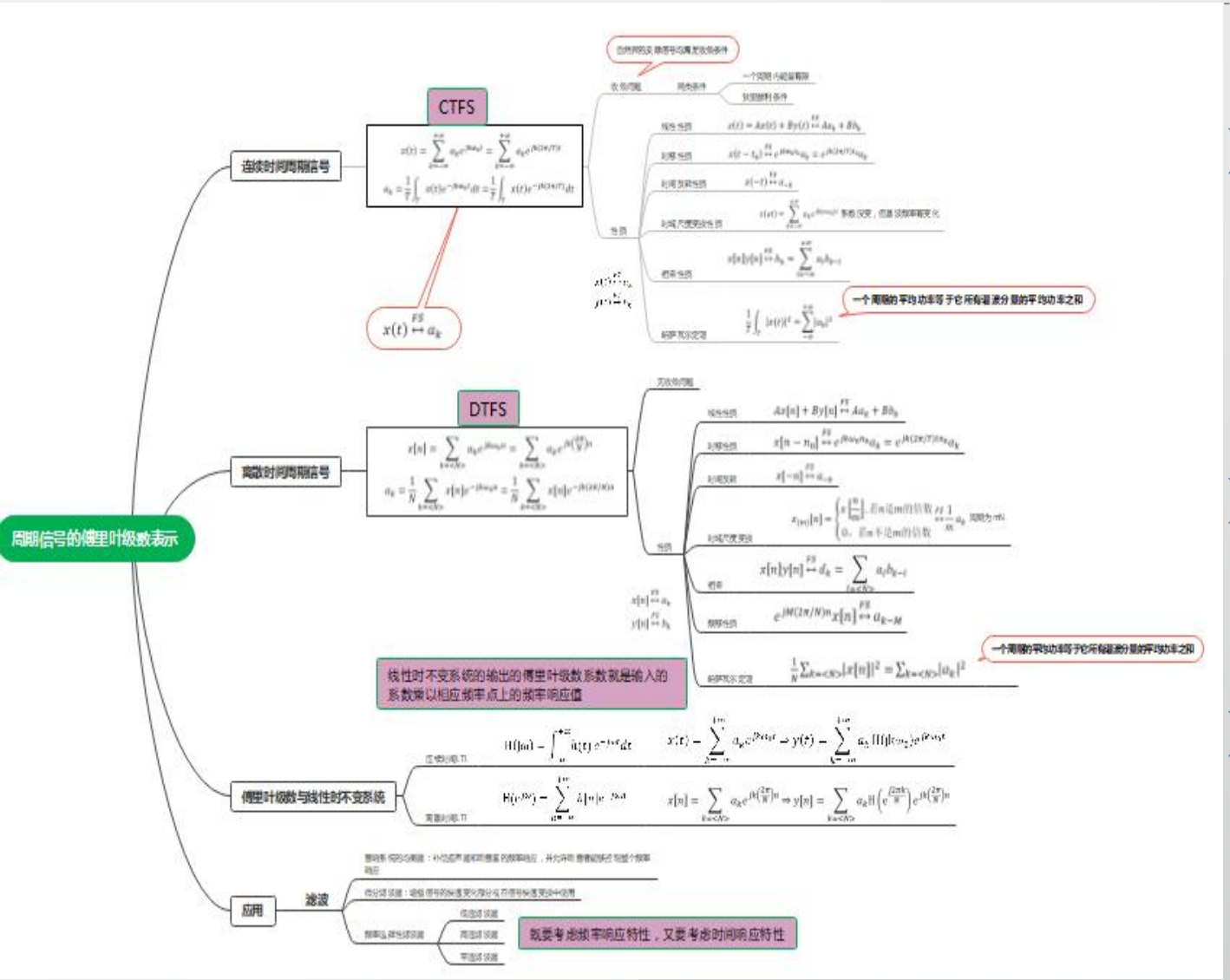
Review

Lab2 LTI System

- **DT LTI**
  - **w=conv (u,v)**
    - w=u*v，卷积计算
    - nw=nu(1)+nv(1):nu(end)+nv(end);
    - Lab2.1
    - 适用范围 ③
  - $\sum_{k=0}^{K} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k]$ 差分方程+初始松弛条件
  - **filter(B,A,x)**
    - 对于上差分方程定义的LTI系统，计算在输入为x时的输出y
    - 输出 y和输入x具有相同的时间索引范围，从而导致 输出信号y可能会被截断！
    - Lab2.2
    - 适用范围 ③
- **CT LTI**
  - **conv(u,v) . step**
    - w=u*v，卷积计算
    - Lab2.8
  - $\sum_{k=0}^{N} a_k \frac{d^k y(t)}{dt^k} = \sum_{k=0}^{N} b_k \frac{d^k x(t)}{dt^k}$ 微分方程+初始松弛条件
  - **lsim (B,A,x,t)**
    - 对于上微分方程定义的LTI系统，计算在输入为x时的输出y
    - x为行向量，lsim输出为列向量
    - Lab2.3，3.3

Autocorrelation Function

Objectives



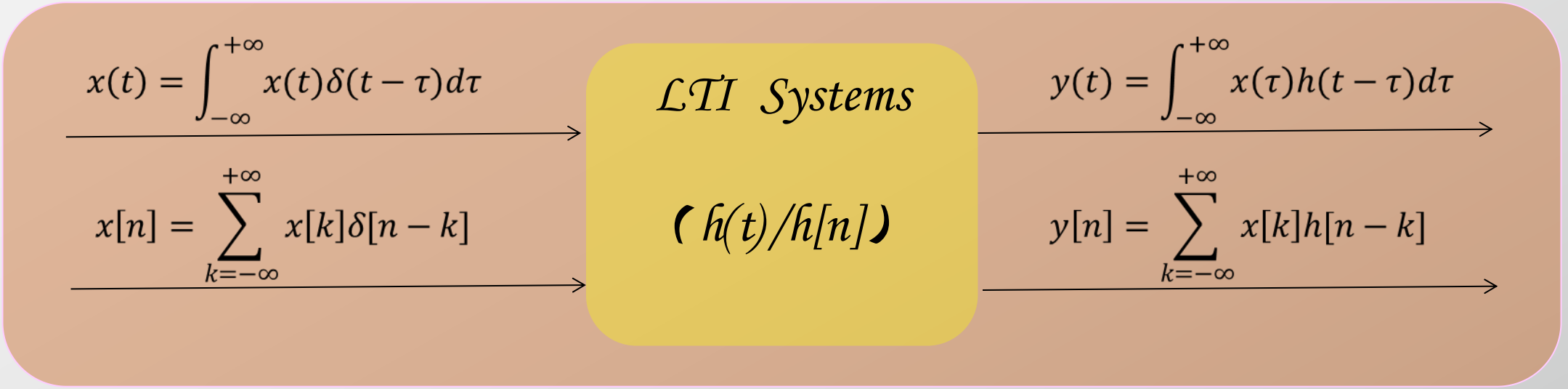
- Fourier Series Representation of Periodic Signals
- Properties of Fourier Series
- Frequency Response of a LTI System

Contents

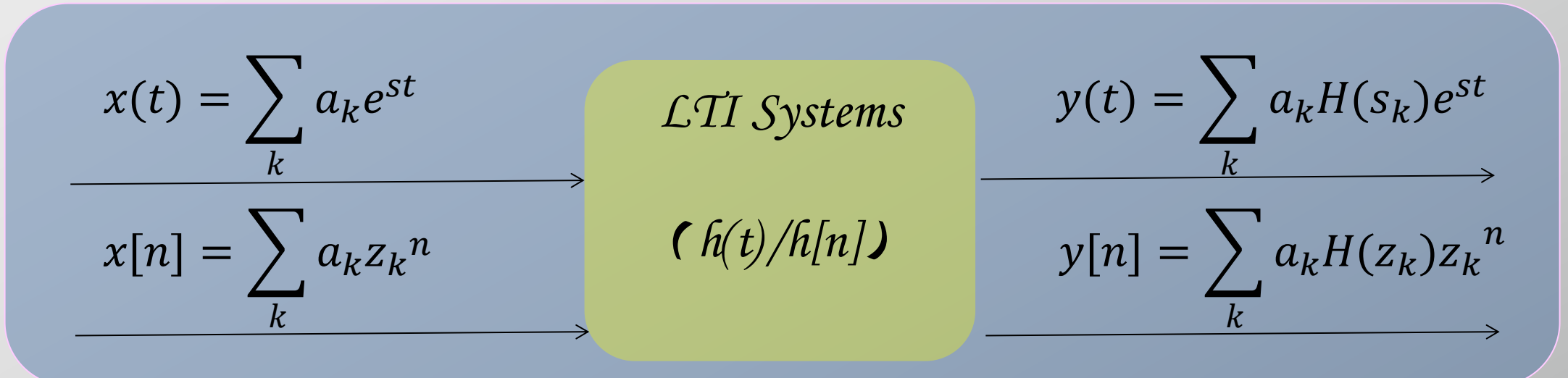
Catalogue

Overview

chapter 2



chapter 3~5



CT LTI

$$x(t) = e^{st} \rightarrow \boxed{h(t)} \rightarrow y(t) = \int_{-\infty}^{+\infty} h(\tau) e^{s(t-\tau)} d\tau$$

$$= \left[\int_{-\infty}^{+\infty} h(\tau) e^{-s\tau} d\tau \right] e^{st}$$

$$= H(s) e^{st}$$

eigenvalue

eigenfunction

DT LTI

$$x[n] = z^n \rightarrow \boxed{h[n]} \rightarrow y[n] = \sum_{k=-\infty}^{+\infty} h[k] z^{n-k}$$

$$= z^n \sum_{k=-\infty}^{+\infty} h[k] z^{-k}$$

$$= H(z) z^n$$

eigenvalue

eigenfunction

Complex exponentials are **eigenfunctions** of LTI systems, i.e., when the input equence is a complex exponential, the output is the same complex exponential only scaled in amplitude by a complex constant. This constant can be computed from the impulse response h[n].When the input to a discrete-time LTI system is $x[n] = z^n$, the output is $y[n] = H(z)z^n$, where

$$H(z) = \sum_{-\infty}^{+\infty} h[k]z^{-k}$$

The Response of LTI systems to Complex Exponentials

$x(t) = e^{st}$ / $x[n] = z^n$ → *LTI Systems* *h(t)/h[n]* → $y(t)=H(s)e^{st}$ / $y[n]=H(z)z^n$

The response of LTI Systems to Compex Exponentials *is the* Compex Exponentials *multiply a complex constant.*

For CT LTI System: Compex Exponentials e^{st} *is its eigenfunction.* The constant H(s) for a specific value of s is then the eigenvalue

$$H(s) = \int_{-\infty}^{+\infty} h(\tau)\, e^{-s\tau} d\tau$$

For DT LTI System: Compex Exponentials z^n *is its eigenfunction, The constant H(Z) for a specific value of z is then the eigenv*

$$H(z) = \sum_{-\infty}^{+\infty} h[k] z^{-k}$$

$x(t) = e^{st}$ → *LTI Syetems* (*h(t)/h[n]*) → $y(t) = H(s)e^{st}$

$x[n] = z^n$ → $y[n] = H(z)z^n$

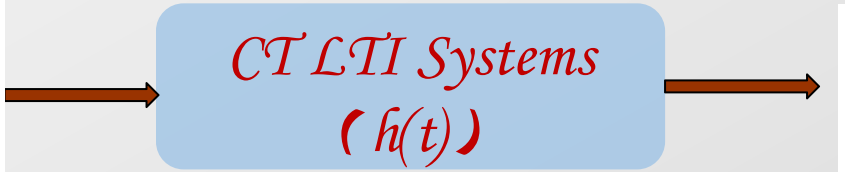
Superpositon

$$x(t) = \sum_k a_k e^{s_k t} \rightarrow \text{CT LTI Syetems } (h(t)) \rightarrow y(t) = \sum_k a_k H(s_k) e^{s_k t}$$

$$x[n] = \sum_k a_k z_k^{\,n} \rightarrow \text{DT LTI Syetems } (h[n]) \rightarrow y[n] = \sum_k a_k H(z_k) z_k^{\,n}$$

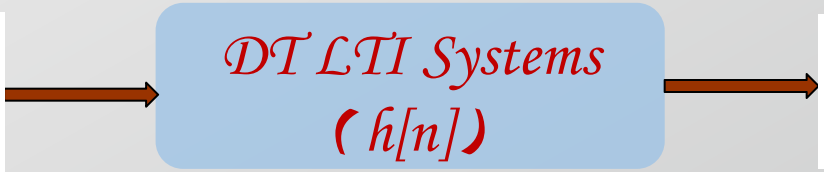
we consider only when s=jw z=e^jw

$$x(t) = \sum_k a_k\, e^{jw_k t}$$



$$y(t) = \sum_k a_k\, H(jw_k) e^{jw_k t}$$

$$x[n] = \sum_k a_k\, e^{jw_k n}$$



$$y[n] = \sum_k a_k\, H(e^{jw_k}) e^{jw_k n}$$

$$x(t) = e^{st} \longrightarrow \boxed{h(t)} \longrightarrow y(t) = \int_{-\infty}^{+\infty} h(\tau) e^{s(t-\tau)} d\tau$$

$$= \left[\int_{-\infty}^{+\infty} h(\tau) e^{-s\tau} d\tau \right] e^{st}$$

$$= H(s) e^{st}$$

$$H(s) = \int_{-\infty}^{+\infty} h(\tau)\, e^{-s\tau} d\tau$$

when s is arbitrary complex number, H(s) is System Function.
when s=jw，H(s) is H(jw), that we named Frequency Response.

$$x[n] = z^n \longrightarrow \boxed{h[n]} \longrightarrow y[n] = \sum_{k=-\infty}^{+\infty} h[k] z^{n-k}$$

$$= z^n \sum_{k=-\infty}^{+\infty} h[k] z^{-k}$$

$$= H(z) z^n$$

$$H(z) = \sum_{-\infty}^{+\infty} h[k] z^{-k}$$

when z is arbitrary complex number, H(z) is System Function.
when z=ejw，H(z) is H(ejw), that we named Frequency Response.

单选题 1分

设置

$$x(t) = \sum_k a_k e^{jw_k t} \longrightarrow \boxed{\text{CT LTI Systems } (h(t))} \longrightarrow y(t) = \sum_k a_k H(jw_k) e^{jw_k t}$$

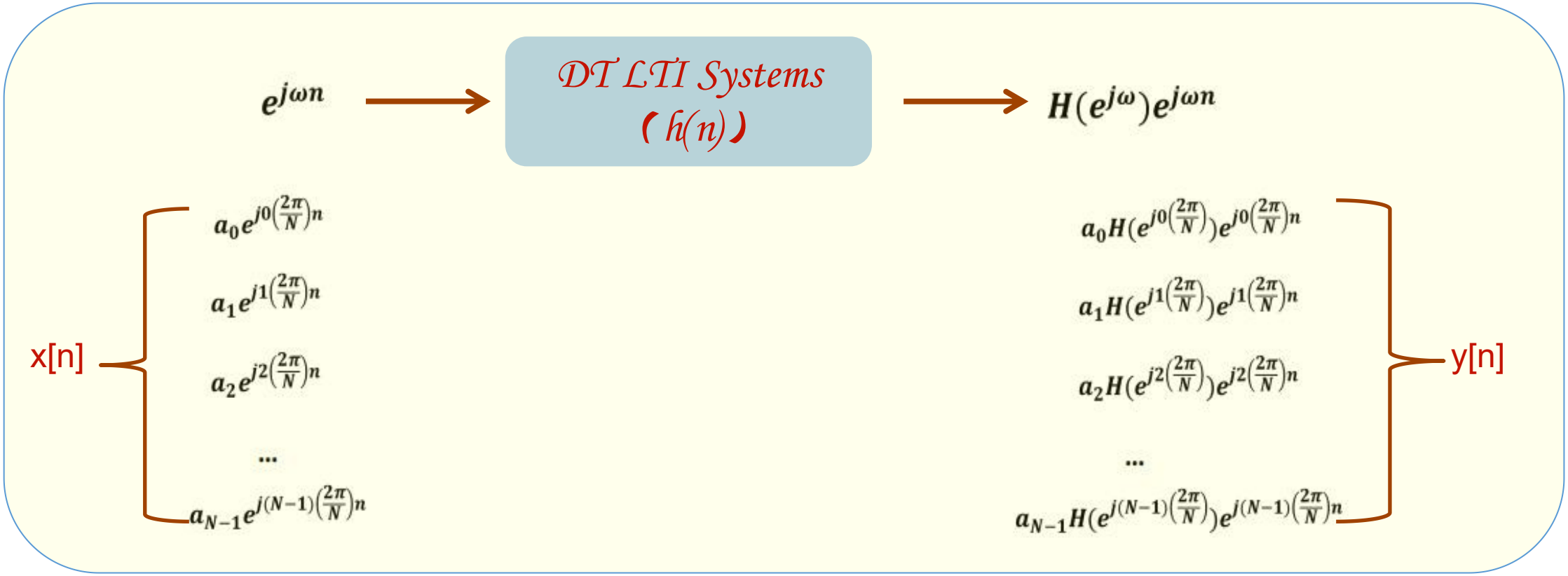
$$x[n] = \sum_k a_k e^{jw_k n} \longrightarrow \boxed{\text{DT LTI Systems } (h[n])} \longrightarrow y[n] = \sum_k a_k H(e^{jw_k}) e^{jw_k n}$$

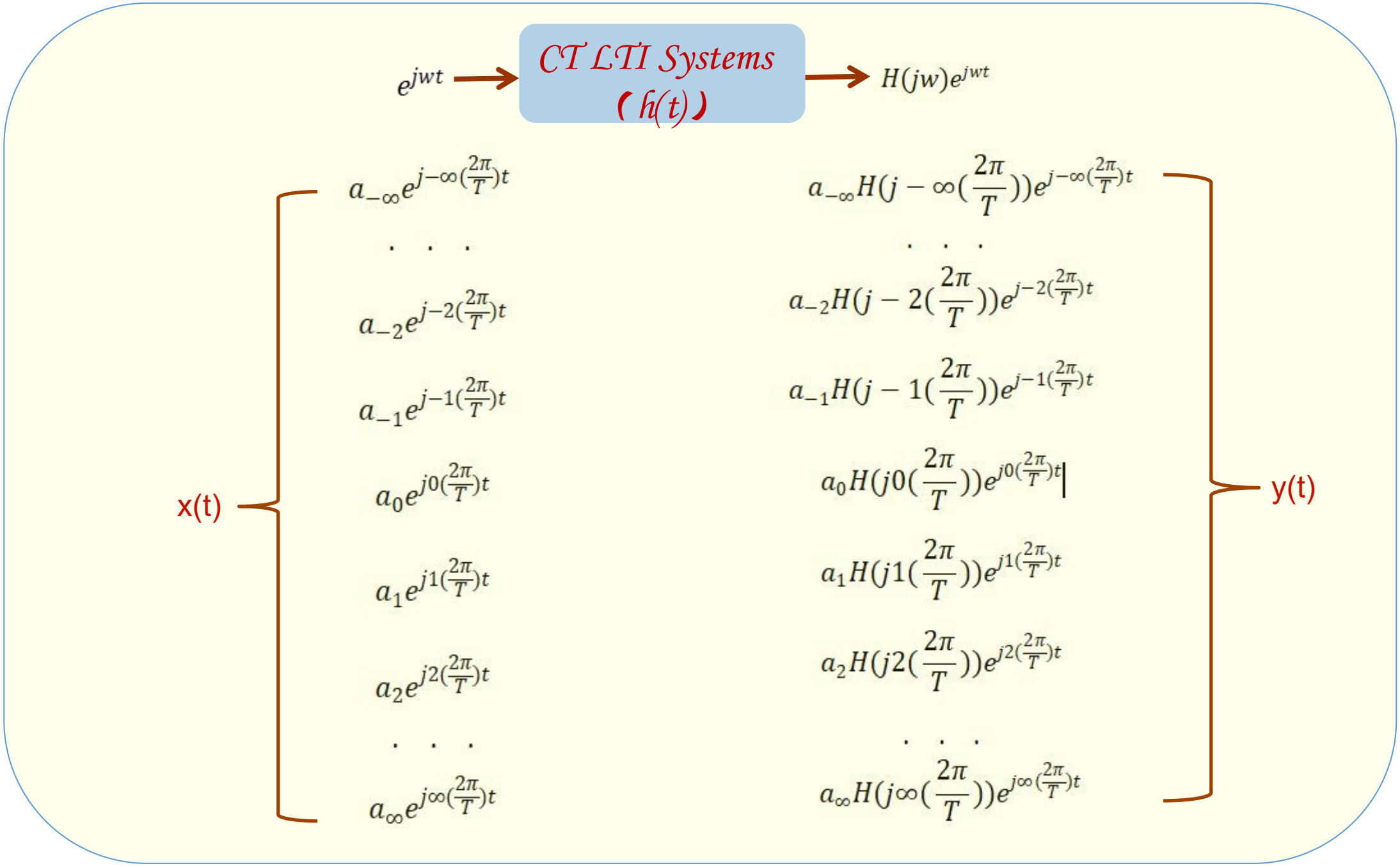
Let x(t) or x[n] correspond to a linear combination of some complex exponentials, and we also know the frequency response H(jw) or $H(e^{jw})$ of the LTI system, then we can easily calculate the response of the LTI system to x(t) or x[n] .

A Yes

B No

提交


$e^{j\omega n}$
DT LTI Systems
(h(n))
$H(e^{j\omega})e^{j\omega n}$
x[n]
$a_0 e^{j0\left(\frac{2\pi}{N}\right)n}$
$a_1 e^{j1\left(\frac{2\pi}{N}\right)n}$
$a_2 e^{j2\left(\frac{2\pi}{N}\right)n}$
...
$a_{N-1} e^{j(N-1)\left(\frac{2\pi}{N}\right)n}$
$a_0 H(e^{j0\left(\frac{2\pi}{N}\right)})e^{j0\left(\frac{2\pi}{N}\right)n}$
$a_1 H(e^{j1\left(\frac{2\pi}{N}\right)})e^{j1\left(\frac{2\pi}{N}\right)n}$
$a_2 H(e^{j2\left(\frac{2\pi}{N}\right)})e^{j2\left(\frac{2\pi}{N}\right)n}$
...
$a_{N-1} H(e^{j(N-1)\left(\frac{2\pi}{N}\right)})e^{j(N-1)\left(\frac{2\pi}{N}\right)n}$
y[n]


e^{jwt}
CT LTI Systems
(h(t))
$H(jw)e^{jwt}$
x(t)
$a_{-\infty}e^{j-\infty(\frac{2\pi}{T})t}$
. . .
$a_{-2}e^{j-2(\frac{2\pi}{T})t}$
$a_{-1}e^{j-1(\frac{2\pi}{T})t}$
$a_0e^{j0(\frac{2\pi}{T})t}$
$a_1e^{j1(\frac{2\pi}{T})t}$
$a_2e^{j2(\frac{2\pi}{T})t}$
. . .
$a_{\infty}e^{j\infty(\frac{2\pi}{T})t}$
$a_{-\infty}H(j-\infty(\frac{2\pi}{T}))e^{j-\infty(\frac{2\pi}{T})t}$
. . .
$a_{-2}H(j-2(\frac{2\pi}{T}))e^{j-2(\frac{2\pi}{T})t}$
$a_{-1}H(j-1(\frac{2\pi}{T}))e^{j-1(\frac{2\pi}{T})t}$
$a_0H(j0(\frac{2\pi}{T}))e^{j0(\frac{2\pi}{T})t}$
$a_1H(j1(\frac{2\pi}{T}))e^{j1(\frac{2\pi}{T})t}$
$a_2H(j2(\frac{2\pi}{T}))e^{j2(\frac{2\pi}{T})t}$
. . .
$a_{\infty}H(j\infty(\frac{2\pi}{T}))e^{j\infty(\frac{2\pi}{T})t}$
y(t)

Then,
to analyze the LTI Systems, we only need to do **two things:**

1： Let x(t) or x[n] correspond to a linear combination of some complex exponentials

$$x(t) = \sum_k a_k e^{jw_k t}$$

$$x[n] = \sum_k a_k e^{jw_k n}$$

2： figure out the System Function of the LTI systems

$$H(s) = \int_{-\infty}^{+\infty} h(\tau) e^{-s\tau} d\tau$$

$$H(z) = \sum_{-\infty}^{+\infty} h[k] z^{-k}$$

When s=jw z=e^{jw}

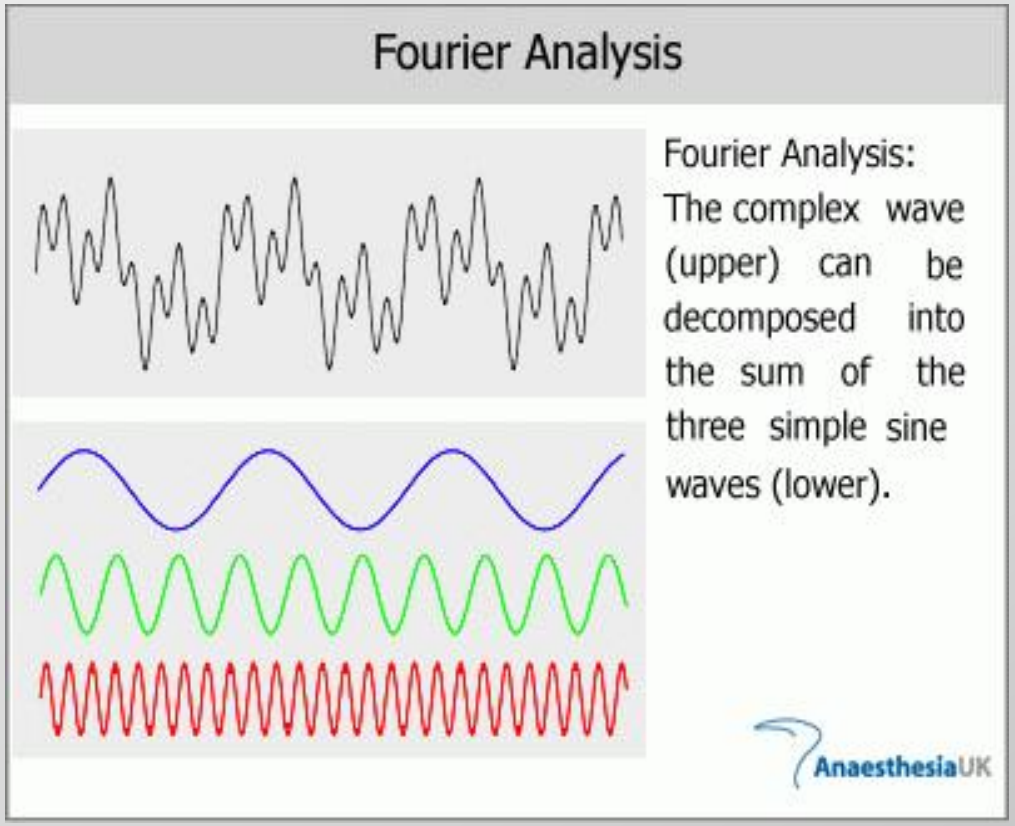
$$H(j\omega) = \int_{-\infty}^{+\infty} h(t) e^{-j\omega t} dt$$

$$H(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} h[n] e^{-j\omega n}$$

In the lab, we focus on DT Signals、DT LTI Systems more

Part I: Fourier Series Respresentation of DT Periodic Signals

Joseph Fourier



DTFS

$$x[n] = \sum_{k=<N>} a_k e^{jk\omega_0 n} = \sum_{k=<N>} a_k e^{jk(2\pi/N)n}$$

→ *Synthesis Equation*

$$a_k = \frac{1}{N} \sum_{n=<N>} x[n]\, e^{-jk\omega_0 n} = \frac{1}{N} \sum_{n=<N>} x[n]\, e^{-jk(2\pi/N)n}$$

→ *Analysis Equation*

$\{a_k\}$ are often called the Fourier series coefficients or the **spectral coefficients**.

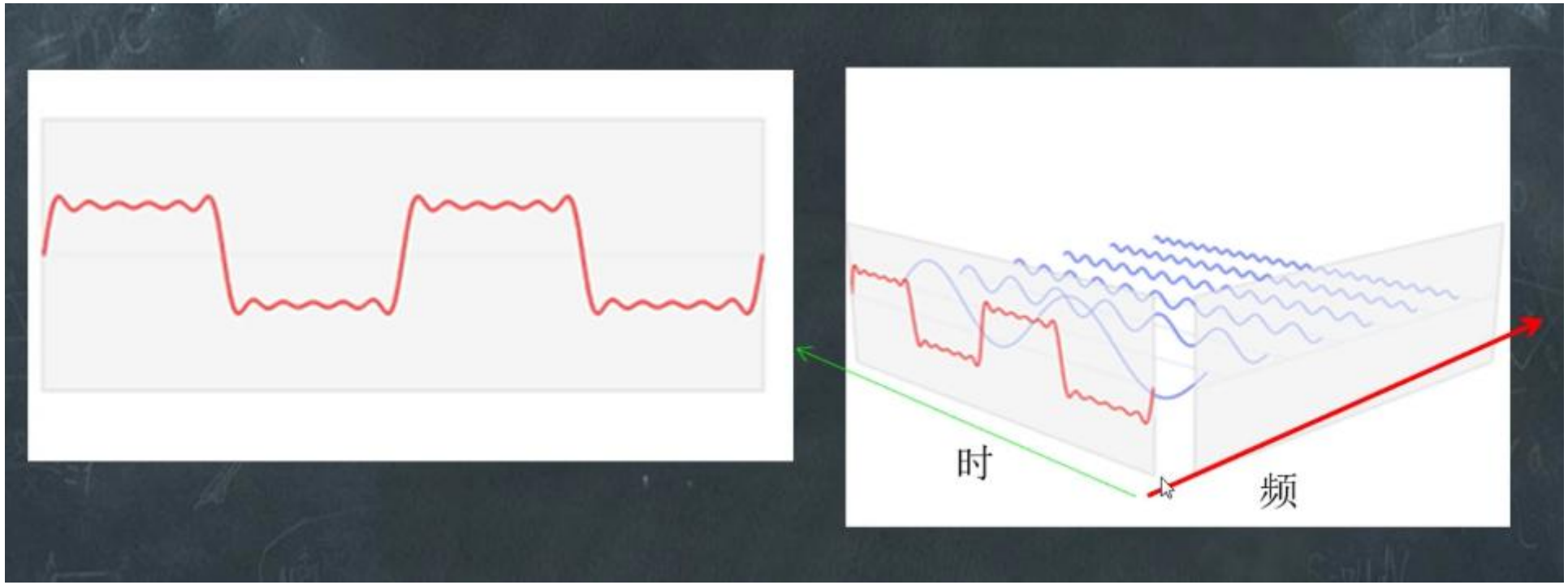
There is N values of k, that is a_k repeat periodiclly with period N.

Linearity、Time shifting、Frequency Shifting、Time Reversal、Time Scaling、Periodic Convolutiion、Multiplication、Conjugate Symmetry for Real Signals、Parseval's Relation for Periodic Signals

$$a_0 = \frac{1}{N} \sum_{n=<N>} x[n]$$

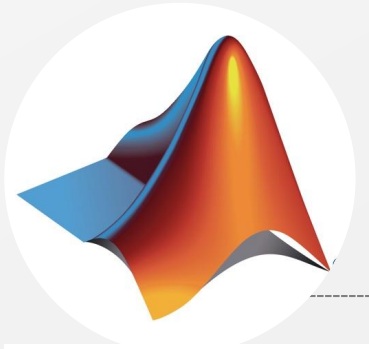
$$\frac{1}{N} \sum_{k=<N>} |x[n]|^2 = \sum_{k=<N>} |a_k|^2$$

Lab3.5



$$x[n] = \sum_{k=<N>} a_k e^{jk\omega_0 n} = \sum_{k=<N>} a_k e^{jk(2\pi/N)n}$$

$$a_k = \frac{1}{N} \sum_{n=<N>} x[n]\, e^{-jk\omega_0 n} = \frac{1}{N} \sum_{n=<N>} x[n]\, e^{-jk(2\pi/N)n}$$



Calculate DTFS

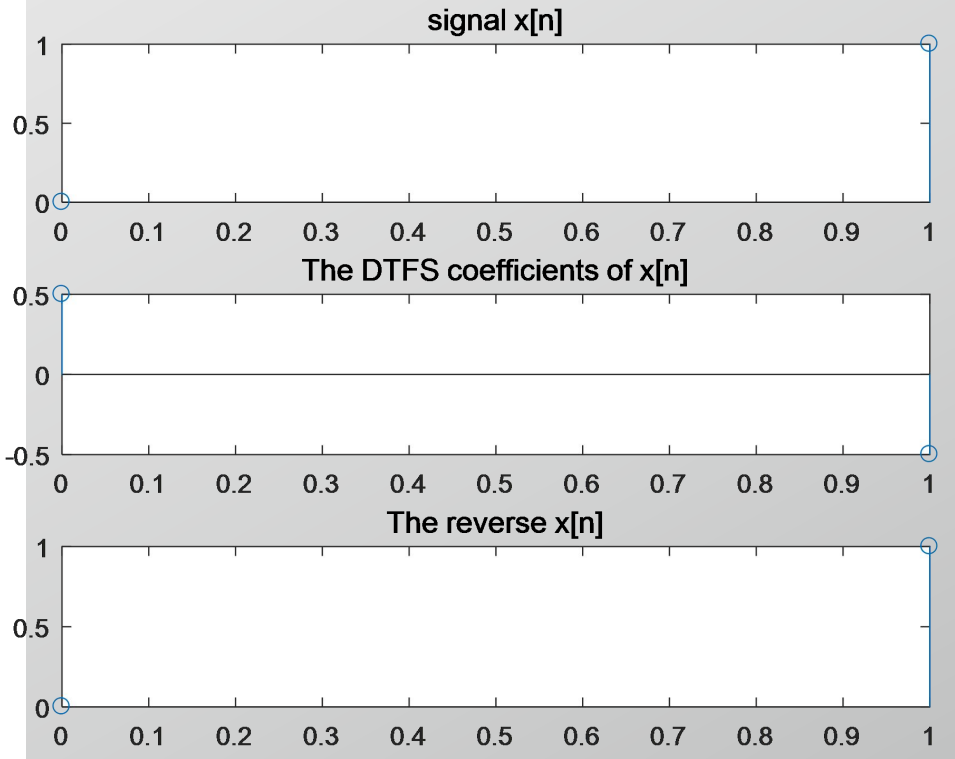
according to the Synthesis and Analysis Equations

$$x[n] = \begin{cases} 1, n = 1 \\ 0, n = 0 \end{cases}, \quad \text{preiod is 2}$$

$$a_k = \frac{1}{N} \sum_{n=<N>} x[n]\, e^{-jk(2\pi/N)n} = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-jk(2\pi/N)n}$$

$$x[n] = \sum_{k=<N>} a_k e^{jk\omega_0 n} = \sum_{k=0}^{N-1} a_k e^{jk(2\pi/N)n}$$

```
clear,close all
x=[0,1];  N=2;
%dtfs
x0=x(1);  x1=x(2);
a0=1/N*(x0+x1);  a1=1/N*(x0+x1*exp(-j*1*2*pi/N*1));
a=[a0,a1];
%Synthesize  x
x_revers(1)=a(1)+a(2);
x_revers(2)=a(1)+a(2)*exp(-j*1*2*pi/N*1);
subplot(3,1,1),stem([0:N-1],x),title('signal x[n]')
subplot(3,1,2),stem([0:N-1],a),title('The DTFS coefficients of x[n]');
subplot(3,1,3),stem([0:N-1],x_revers),title('The reverse x[n]');
```



Complexity Analysis of DTFS

$$a_k = \frac{1}{N}\sum_{n=<N>} x[n]\, e^{-jk\omega_0 n} = \frac{1}{N}\sum_{n=<N>} x[n]\, e^{-jk(2\pi/N)n}$$

Suppose we know the matrix $E(n,k) = e^{-jk(2\pi/N)n}$

? How many multiplications & additions are needed to calculate Fourier series [a0，a1，…，aN-1]

$$a_k = \frac{1}{N}\sum_{n=<N>} x[n]\, e^{-jk\omega_0 n} = \frac{1}{N}\sum_{n=<N>} x[n]\, e^{-jk(2\pi/N)n}$$

$$E(n,k) = e^{-jk(2\pi/N)n}$$

$$a_0 = \frac{1}{N}\sum_{n=<N>} x[n]e^{-j0(2\pi/N)n} = \frac{1}{N}\sum_{n=<N>} x[n]E(n,0)$$

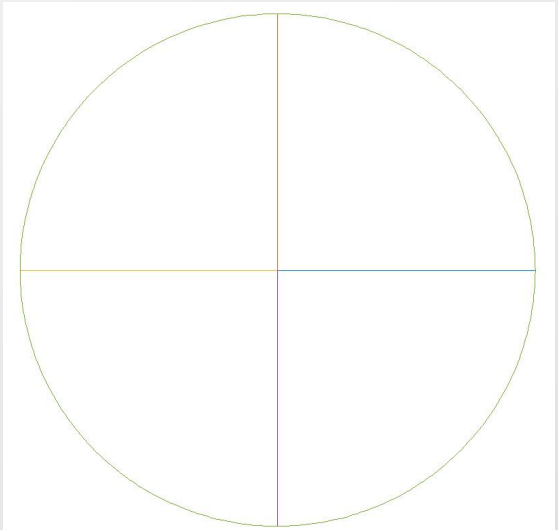
$$a_1 = \frac{1}{N}\sum_{n=<N>} x[n]e^{-j1(2\pi/N)n} = \frac{1}{N}\sum_{n=<N>} x[n]E(n,1)$$

•
•
•

Each: $N + 1$ $\times$; $N - 1$ +
Total: $N(N + 1)$ $\times$; $N(N - 1)$ +

Fast Fourier Transform (FFT)

$$E(n,k) = e^{-jk(2\pi/N)n}$$



$E(n,k)$

2 π is divided into 4 parts;

Although there are 16 elements, only four values are different from each other, and there is a symmetric relationship between them.

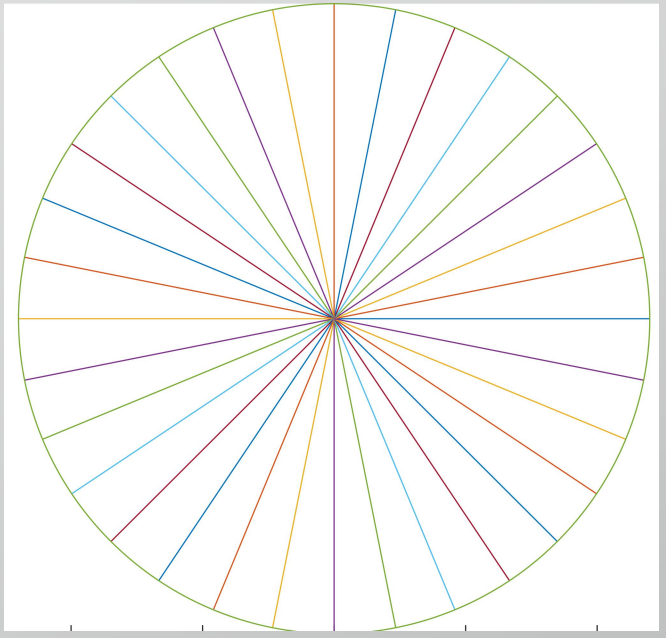
N=4

a_0=(x[0]E(0,0)+x[1]E(1,0)+x[2]E(2,0)+x[3]E(3,0))/N

a_2=(x[0]E(0,2)+x[1]E(1,2)+x[2]E(2,2)+x[3]E(3,2))/N

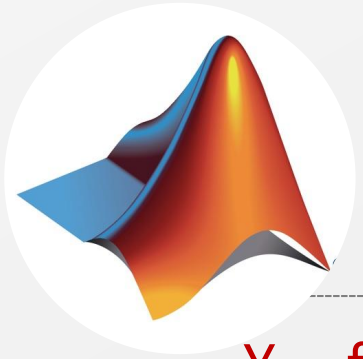
a_1=(x[0]E(0,1)+x[1]E(1,1)+x[2]E(2,1)+x[3]E(3,1))/N

a_3=(x[0]E(0,3)+x[1]E(1,3)+x[2]E(2,3)+x[3]E(3,3))/N



To calculate the DFT and FFT of a 1024*1024 image:

CPU	Clock Frequency	DFT	FFT
1941	60 Hz	152.3 y	271.4 d
1971 (4004)	108KHz	30.8 d	3.6 h
1978 (8086)	10MHz	8.0 h	2.3 min
1982 (80286)	20MHz	4.0 h	1.2min
1985 (80386)	33MHz	2.4h	42.6s
1989 (80486)	100MHz	48.0min	14.1s
1995 (Pentium)	200MHz	24.0min	7.0s
1999 (Pentium Ⅲ)	450MHz	10.7min	3.1s
2000 (Pentium 4)	1.4GHz	3.4min	1.0s
2001 (Pentium 4)	2GHz	2.4min	0.7s



Calculate DTFS by Matlab

Y = fft(X) computes the discrete Fourier transform (DFT) of X using a fast Fourier transform (FFT) algorithm.

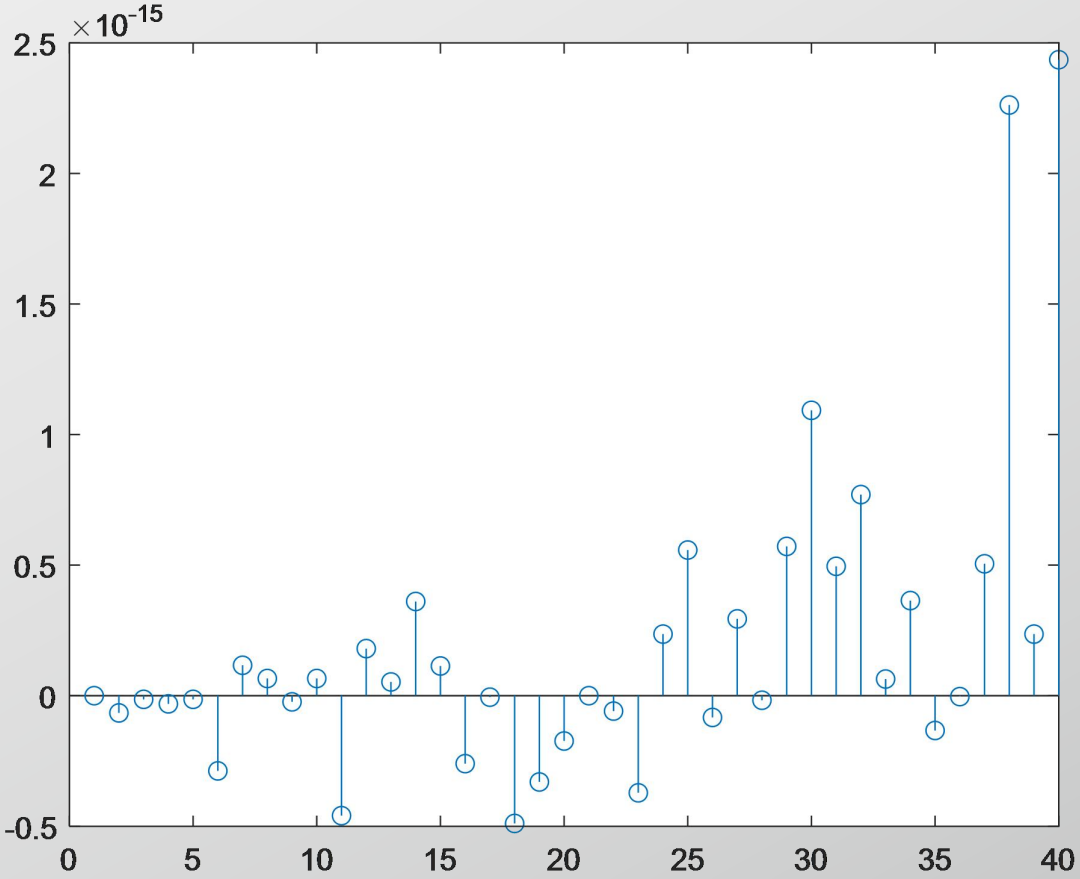
```
>> help fft
                    N
     X(k) =       sum  x(n)*exp(-j*2*pi*(k-1)*(n-1)/N), 1 <= k <= N.
                  n=1
```

- Compare with DTFS's Definition:

$$a_k = \frac{1}{N}\sum_{n=<N>} x[n]\, e^{-jk(2\pi/N)n} = \frac{1}{N}\sum_{n=0}^{N-1} x[n] e^{-jk(2\pi/N)n}$$

- Calculate the DTFS of vector x whose period is N: a = (1/N) * fft(x)

```
clc,clear,close all
x=[zeros(1,20) ones(1,20)];
stem(0:39, x); xlim([0 39]); ylim([0 1.2]);
%DTFS by equations
N=length(x);
a=zeros(1,N);
for k=0:1:N-1
    for n=0:1:N-1
        a(k+1)=a(k+1)+x(n+1)*exp(-j*k*(2*pi/N)*n);
    end
end
a=a./N;
%DTFS by fft
b=fft(x)/N;
stem(a-b)
```



X = ifft(Y), computes the inverse discrete Fourier transform of Y using a fast Fourier transform algorithm.

Reconstruct signals from DTFS coefficients

```
>> help fft

The inverse DFT (computed by IFFT) is given by
                    N
      x(n) = (1/N) sum  X(k)*exp( j*2*pi*(k-1)*(n-1)/N), 1 <= n <= N.
                    k=1
```

- Compare with our definition:

$$x[n] = \sum_{k=<N>} a_k e^{jk\omega_0 n} = \sum_{k=0}^{N-1} a_k e^{jk(2\pi/N)n}$$

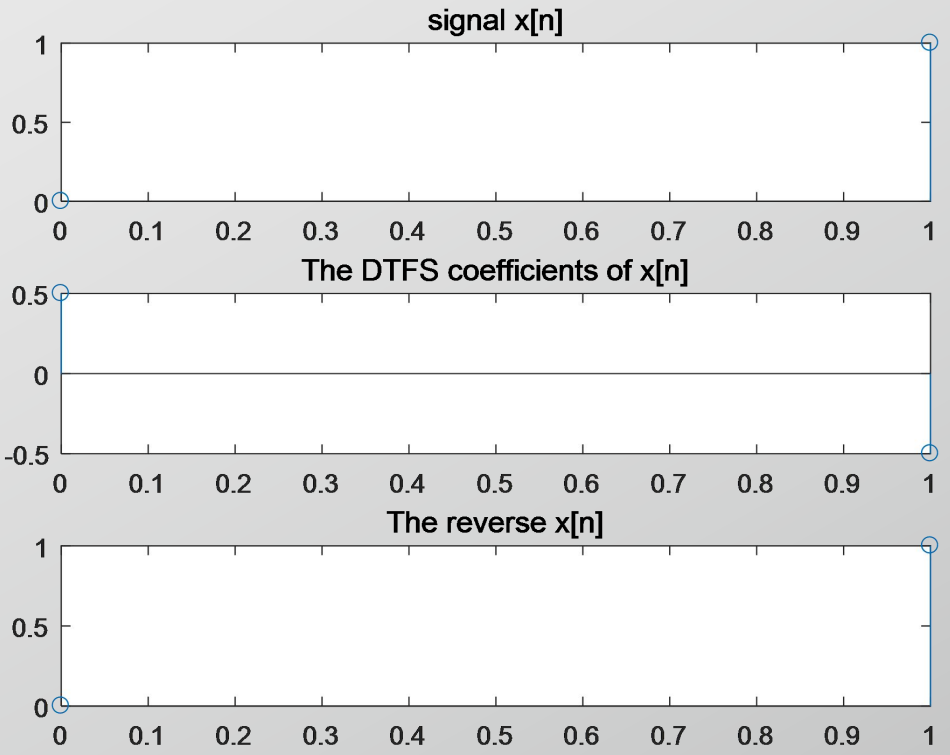
- Reconstruct signal x from DTFS coefficients a：

```
x=N*ifft(a)
```

$a=1/N*fft(x),\quad x=N*ifft(a)$

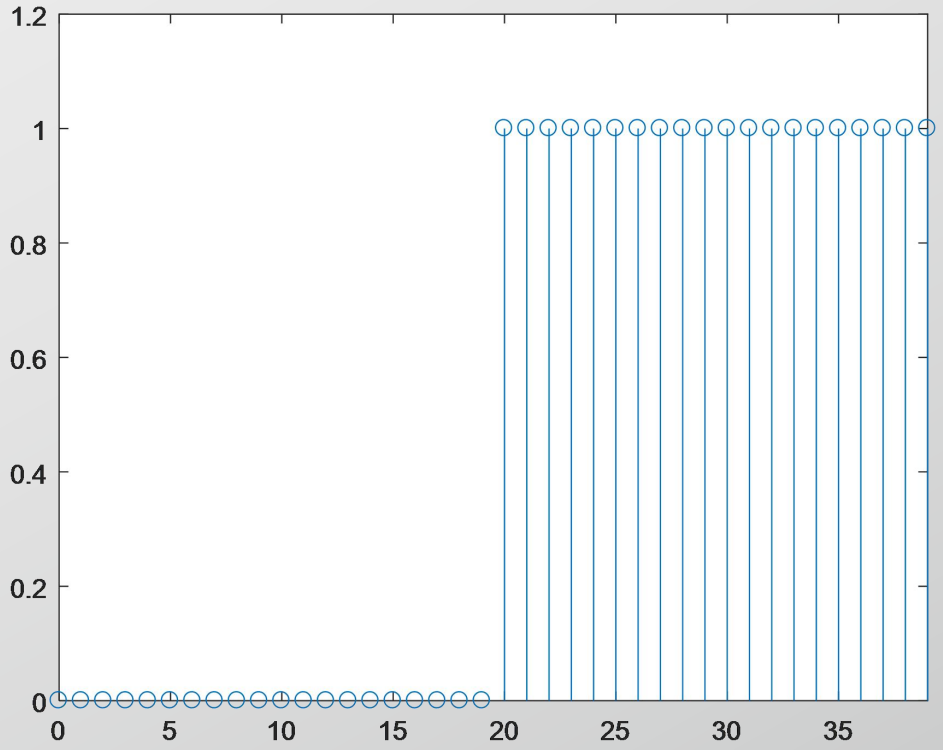
$$x[n]=\begin{cases}1, n=1\\0, n=0\end{cases}, \text{ Period is 2}$$

```
clear,close all
x=[0,1];
N=2;
%dtfs
a=1/N*fft(x);
%Synthesize  x
x_revers=N*ifft(a);
subplot(3,1,1),stem([0:N-1],x),title('signal x[n]')
subplot(3,1,2),stem([0:N-1],a),title('The DTFS coefficients of x[n]');
subplot(3,1,3),stem([0:N-1],x_revers),title('The reverse x[n]')
```



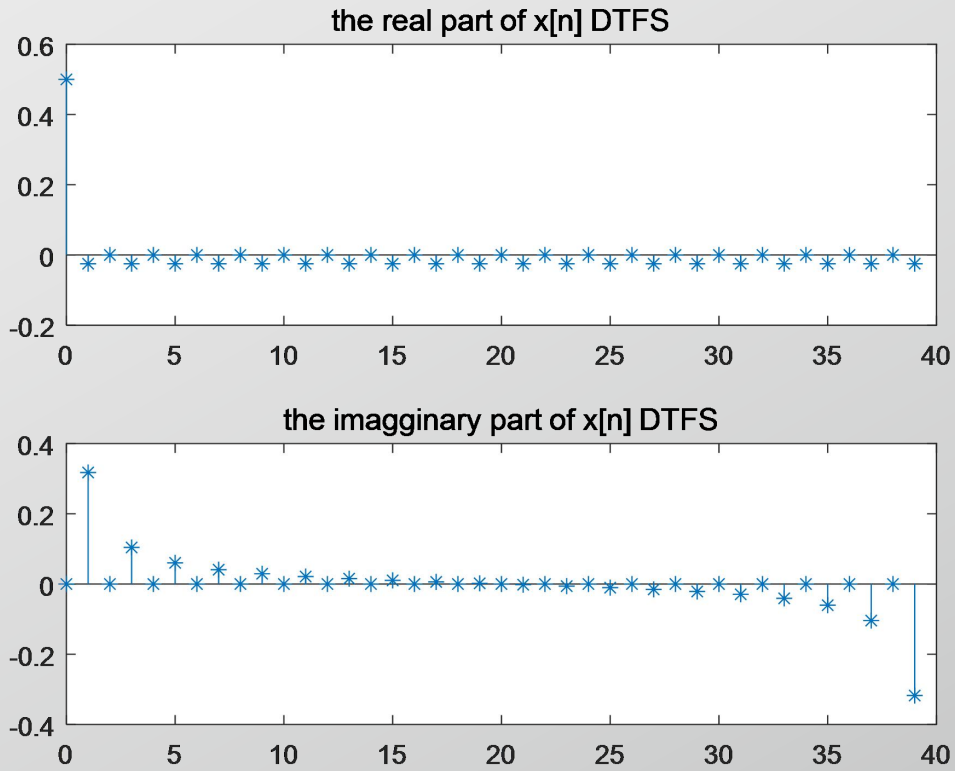
Practice: Periodic Signal x=[zeros(1,20) ones(1,20)];nx=0:1:39；N=40. Calculate its Fourier Series coefficients by Matlab function fft;

```
x=[zeros(1,20) ones(1,20)];
stem(0:39, x);
xlim([0 39]);
ylim([0 1.2]);
```



```
N=length(x);
A =1/N* fft(x);
figure
subplot(2,1,1), stem(0: N-1,real(A),'*-');
title('the real part of x[n] DTFS')
subplot(2,1,2), stem(0: N-1,imag(A),'*-');
title('the imagginary part of x[n] DTFS')
```

$$a=1/N*fft(x)$$



A =

1 至 7 列

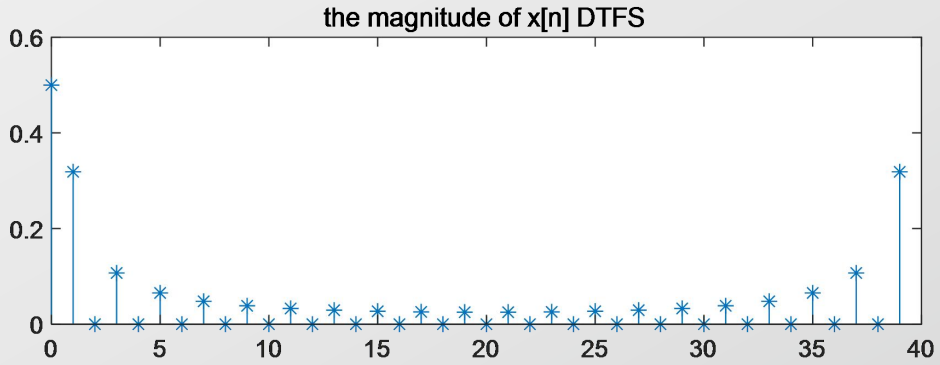
0.5000 + 0.0000i -0.0250 + 0.3177i 0.0000 + 0.0000i -0.0250 + 0.1041i 0.0000 + 0.0000i -0.0250 + 0.0604i 0.0000 + 0.0000i

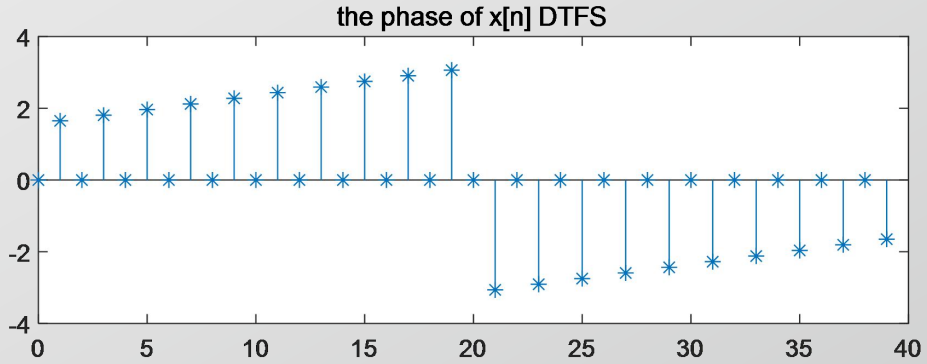
……

36 至 40 列

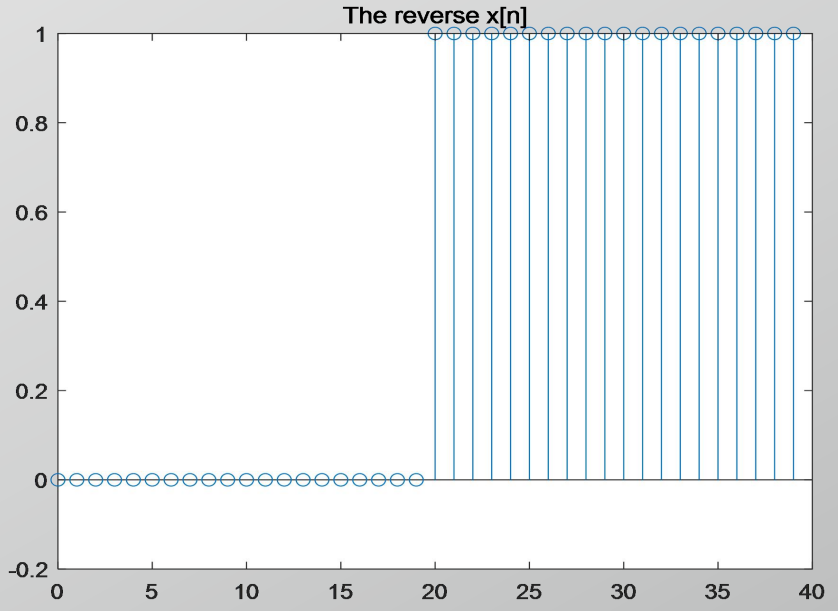
-0.0250 - 0.0604i 0.0000 + 0.0000i -0.0250 - 0.1041i 0.0000 + 0.0000i -0.0250 - 0.3177i

```
figure
subplot(2,1,1),stem(0:N-1,abs(A),'*-');
title('the magnitude of x[n] DTFS')
subplot(2,1,2),stem([0:N-1],angle(A),'*-');
title('the phase of x[n] DTFS')
```



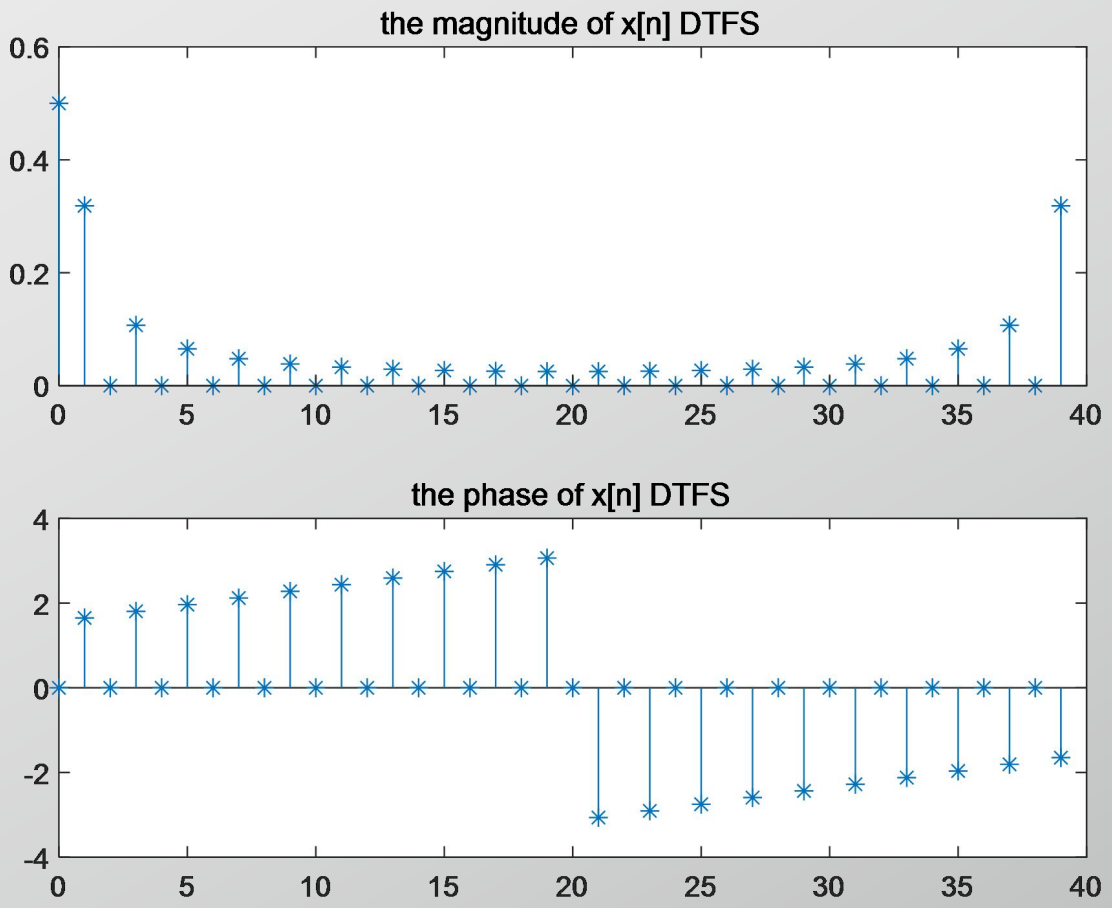



```
%Synthesize  x
x_revers=N*ifft(A)
figure
stem(0:N-1,x_revers),title('The reverse x[n]')
```



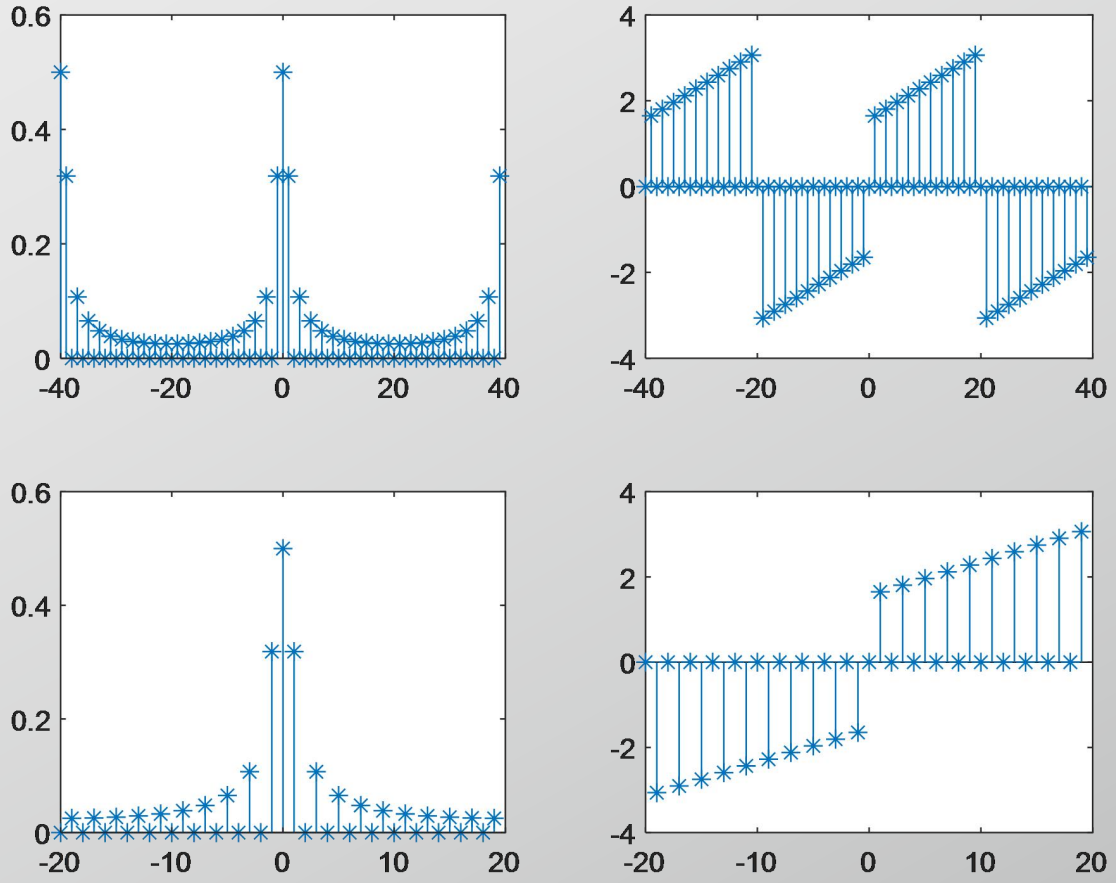
When x(t) is real, its Fourier Series coefficients will be *Conjugate Symmetric*,i.e.,$a_{-k}=a_k^*$

```
figure
subplot(1,2,1),stem(0:length(x)-1,abs(A),'*-');
title('the magnitude of x[n] DTFS')
subplot(1,2,2),stem([0:length(x)-1],angle(A),'*-');
title('the phase of x[n] DTFS')
```



When x(t) is real, its Fourier Series coefficients will be *Conjugate Symmetric*,i.e.,$a_{-k}=a_k^*$

```
figure
subplot(2,2,1),stem([-N:N-1],[abs(A),abs(A)],'*-');
subplot(2,2,2),stem([-N:N-1],[angle(A),angle(A)],'*-');
subplot(2,2,3),stem([-N/2:N/2-1],abs(fftshift(A)),'*-');
subplot(2,2,4),stem([-N/2:N/2-1],angle(fftshift(A)),'*-');
```



A useful Matlab Function: **fftshift**

fftshift : Shift zero-frequency component to center of spectrum.

Y=fftshift(X), rearranges a Fourier transform X by shifting the zero-frequency component to the center of the array.

Tests：

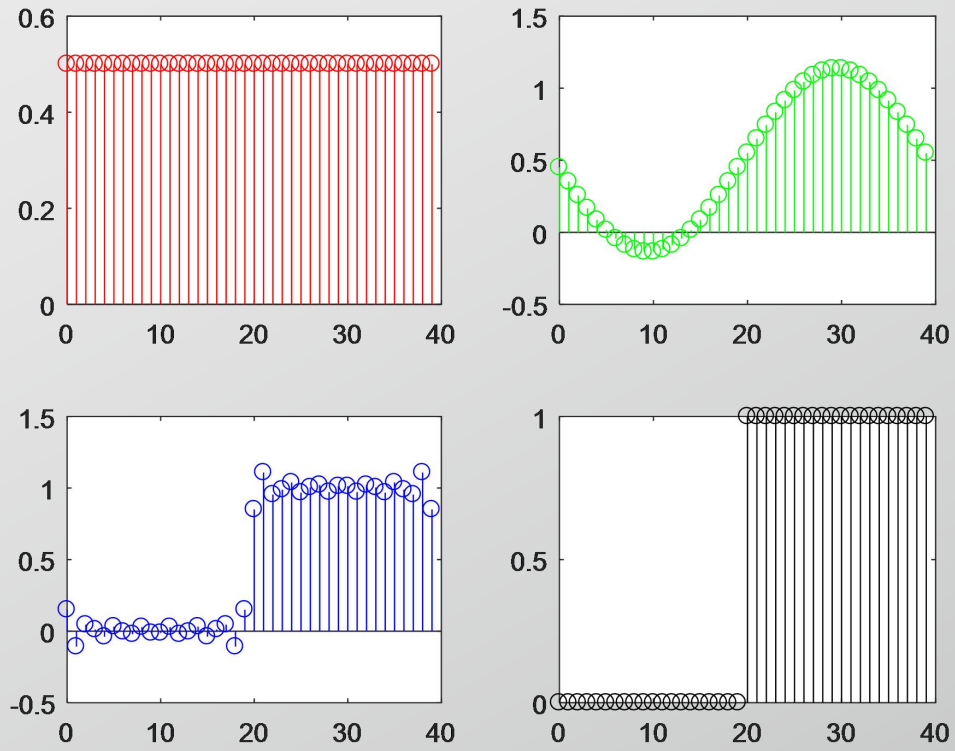
```
ak1=[0 1 2 3 4 5 6 7];
fftshift(ak1)

ak2=[0 1 2 3 4 5 6 7 8];
fftshift(ak2)
```

Practice: x[n] is a periodic signal and its Fourier Series coefficients is A. Please reconstruct signal x[n]

$$x = N * ifft(a)$$

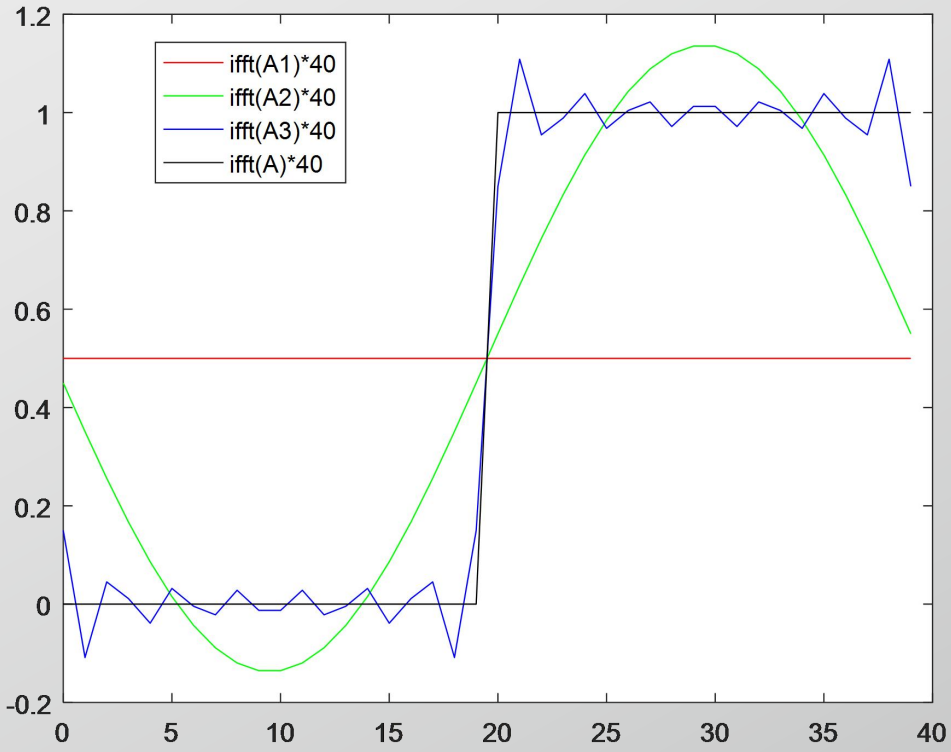
```
A1 = [A(1) zeros(1,39)];
A2 = [A(1) A(2) zeros(1,37) A(40)];
A3 = [A(1) A(2:15) zeros(1,11) A(27:40)];
figure
subplot(2,2,1), stem(0:39,ifft(A1)*40, 'r');
subplot(2,2,2), stem(0:39,ifft(A2)*40, 'g');
subplot(2,2,3), stem(0:39,ifft(A3)*40, 'b');
subplot(2,2,4), stem(0:39,ifft(A)*40, 'k');
```



A= 0.5000 + 0.0000i -0.0250 + 0.3177i 0.0000 + 0.0000i -0.0250 + 0.1041i 0.0000 + 0.0000i
-0.0250 + 0.0604i 0.0000 + 0.0000i -0.0250 + 0.0408i 0.0000 + 0.0000i -0.0250 +0.0293i
0.0000 + 0.0000i -0.0250 + 0.0214i ……

$$x = N * ifft(a)$$

```
figure
plot(0:39,ifft(A1)*40, 'r', ...
     0:39,ifft(A2)*40, 'g', ...
     0:39,ifft(A3)*40, 'b',...
     0:39,ifft(A)*40, 'k');
legend('ifft(A1)*40','ifft(A2)*40','ifft(A3)*40','ifft(A)*40')
```



No Gibbs phenomenon

Fourier Series Respresentation of CT Periodic Signals

CTFS

$$x(t) = \sum_{k=-\infty}^{+\infty} a_k e^{jkw_0 t} = \sum_{k=-\infty}^{+\infty} a_k e^{jk(2\pi/T)t}$$

→ *Synthesis Equation*

$$a_k = \frac{1}{T}\int_T x(t)e^{-jkw_0 t}dt = \frac{1}{T}\int_T x(t)e^{-jk(2\pi/T)}dt$$

→ *Analysis Equation*

$\{a_k\}$ are often called the Fourier series coefficients or the **spectral coefficients**.

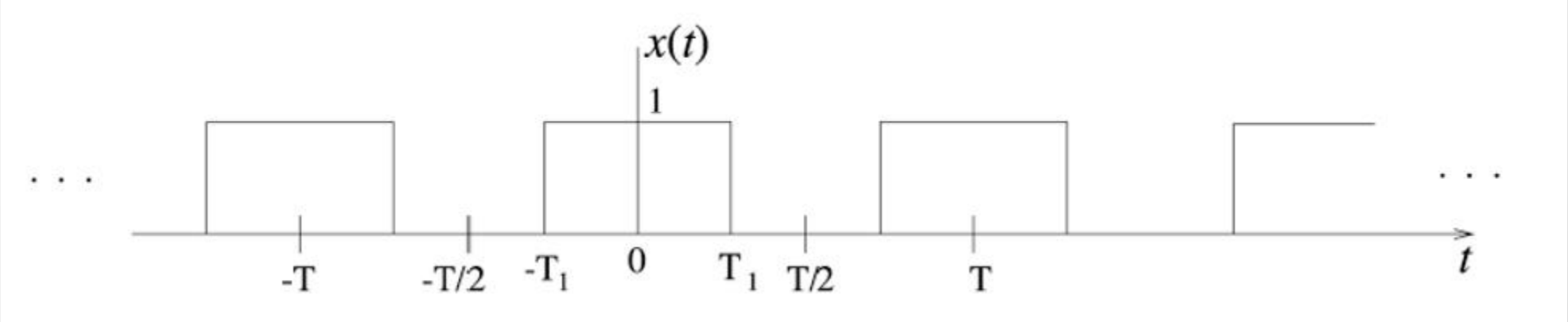
$$k \in (-\infty, +\infty)$$

$$a_0 = \frac{1}{T}\int_T x(t)dt$$

$$\frac{1}{T}\int_T |x(t)|^2 = \sum_{-\infty}^{+\infty} |a_k|^2$$

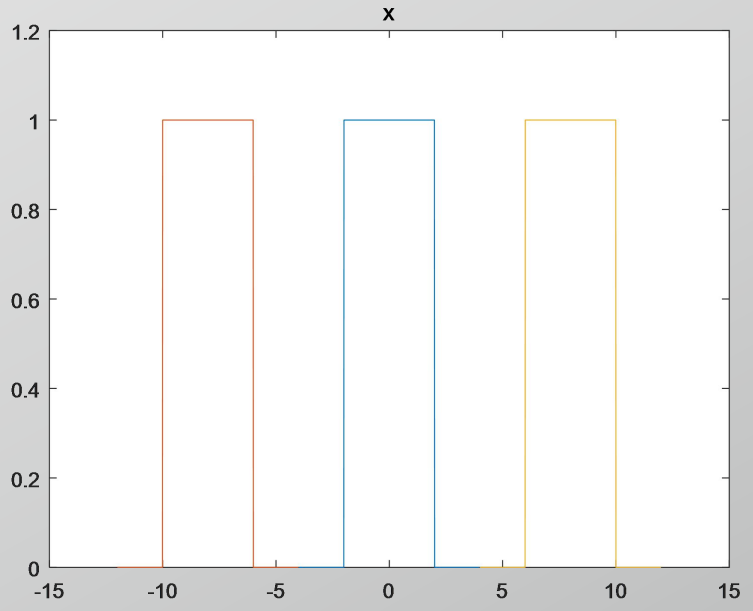
Linearity、Time shifting、Frequency Shifting、Time Reversal、Time Scaling、Periodic Convolutiion、Multiplication、Conjugate Symmetry for Real Signals、Parseval's Relation for Periodic Signals

Lab3.11、3.12

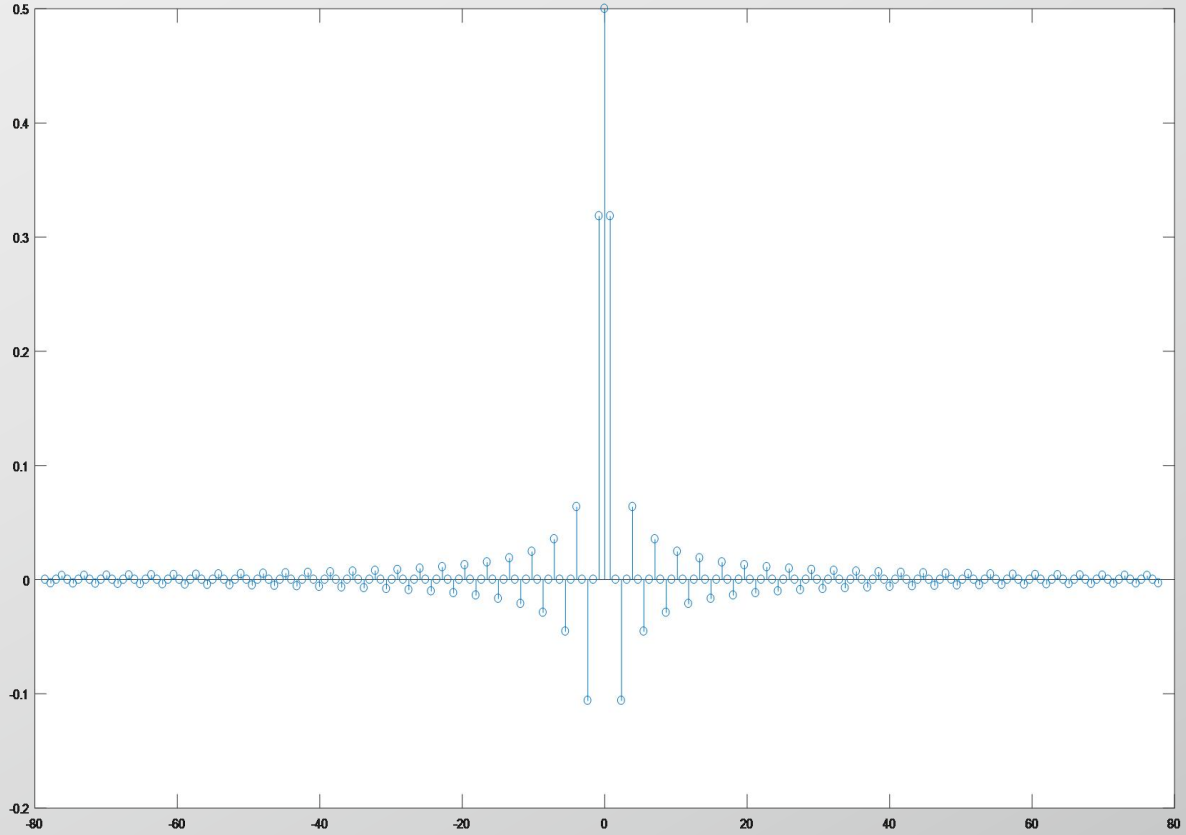


$$a_k = \frac{1}{T}\int_{-T/2}^{T/2} x(t)e^{-jkw_0t}dt = \frac{\sin(kw_0T_1)}{k\pi}$$

```
clear,close all
step=0.01;
t=-4:step:4-step;
x=[zeros(1,length(t)/4),ones(1,length(t)/2),zeros(1,length(t)/4)];
figure
plot(t,x),title('x'),ylim([0 1.2]),hold on
plot((t-8),x),hold on,plot((t+8),x)
```



```
N=100;
k=-N:1:N-1;
T=8;
T1=2;
w0=2*pi/T;
T1=2;
a0=0.5;
a=(sin(k*w0*T1)/pi)./k
a(N+1)=0.5;
stem(k,a)
```



Properties of DTFS

线性性质 $Ax[n] + By[n] \overset{FS}{\leftrightarrow} Aa_k + Bb_k$

时移性质 $x[n - n_0] \overset{FS}{\leftrightarrow} e^{jk\omega_0 n_0} a_k = e^{jk(2\pi/T)tn_0} a_k$

时间反转 $x[-n] \overset{FS}{\leftrightarrow} a_{-k}$

时域尺度变换 $x_{(m)}[n] = \begin{cases} x\left[\frac{n}{m}\right], \text{若} n \text{是} m \text{的倍数} \\ 0, \text{若} n \text{不是} m \text{的倍数} \end{cases} \overset{FS}{\leftrightarrow} \frac{1}{m} a_k$ 周期为mN

共轭和共轭对称性质 $x^*[n] \overset{FS}{\leftrightarrow} a^*_{-k}$

实信号的共轭对称 $a_k = a^*_k$ 当x[n]为实信号

周期卷积 $\sum_{r=<N>} x[r]h[n-r] \overset{FS}{\leftrightarrow} Na_k b_k$

相乘 $x[n]y[n] \overset{FS}{\leftrightarrow} d_k = \sum_{l=<N>} a_l b_{k-l}$

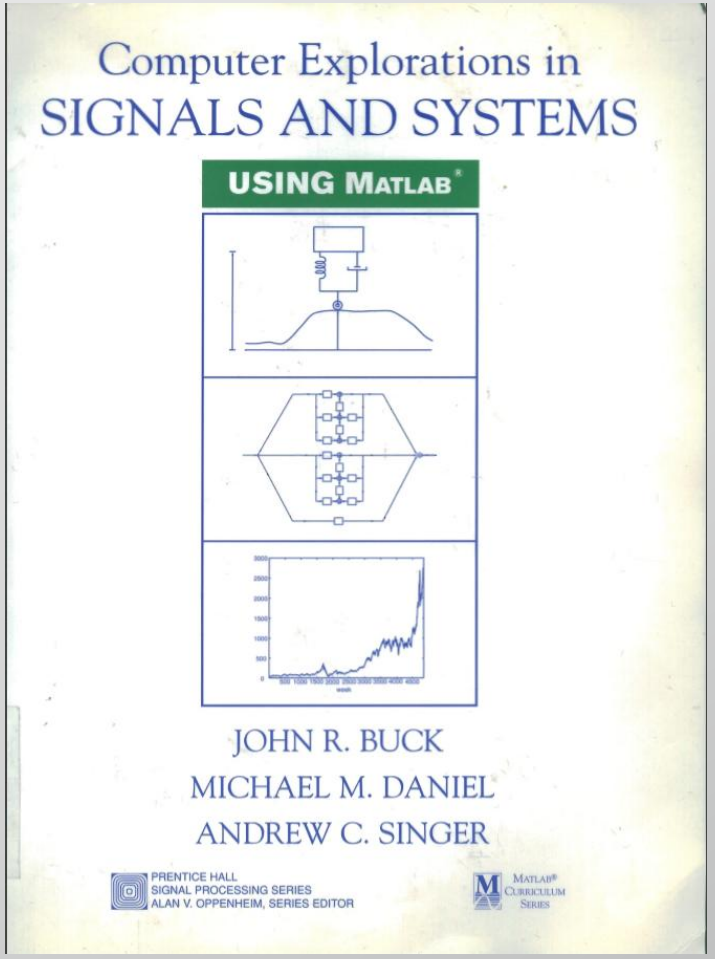
频移性质 $e^{jM(2\pi/N)n} x[n] \overset{FS}{\leftrightarrow} a_{k-M}$

一个周期的平均功率等于它所有谐波分量的平均功率之和

帕萨瓦尔定理 $\frac{1}{N} \sum_{k=<N>} |x[n]|^2 = \sum_{k=<N>} |a_k|^2$

Lab3 Assignments a

- Read tutorial 3.1、3.2
- Assignments：3.5 & 3.10



Tips :

1

Several parts of this exercise require you to generate vectors which should be purely real, but have very small imaginary parts due to numerical roundoff errors. Use **real** to remove these residual imaginary parts from these vectors.

■ 3.5 Synthesizing Signals with the Discrete-Time Fourier Series

The discrete-time Fourier series (DTFS) is a frequency-domain representation for periodic discrete-time sequences. The synthesis and analysis equations for the DTFS are given by Eqs. (3.1) and (3.2). This exercise contains three sets of problems to give you practice working with these equations. The Basic Problems allow you to synthesize a very simple periodic discrete-time signal from its DTFS coefficients. In the Intermediate Problems, you will both analyze a set of periodic discrete-time signals to obtain their DTFS coefficients, and construct one of these signals by adding in a few coefficients at a time. For the Advanced Problem, you will write a function to find the DTFS coefficients of an arbitrary periodic discrete-time signal from one period of samples.

Basic Problems

In these problems, you will synthesize a periodic discrete-time signal with period $N = 5$ and the following DTFS coefficients

$$a_0 = 1,\ a_2 = a_{-2}^* = e^{j\pi/4},\ a_4 = a_{-4}^* = 2e^{j\pi/3}.$$

(a). Based on the DTFS coefficients, do you expect $x[n]$ to be complex-valued, purely real, or purely imaginary? Why?

(b). Using the DTFS coefficients given above, determine the values of a_0 through a_4 and specify a vector **a** containing these values.

2 *Lab 3.5*

- The DTFS coefficients of a periodic discrete-time signal with period N = 5:

$$a_0 = 1,\ a_2 = a_{-2}^* = e^{j\pi/4},\ a_4 = a_{-4}^* = 2e^{j\pi/3}.$$

a_1 ? ?

a_3 ? ?

$$a_k = a_{k+N}$$

3

$$x[n] = \sum_{k=<N>} a_k e^{jk\omega_0 n} = \sum_{k=<N>} a_k e^{jk\left(\frac{2\pi}{N}\right)n}$$

$$a_k = \frac{1}{N} \sum_{k=<N>} x[n] e^{-jk\omega_0 n} = \frac{1}{N} \sum_{k=<N>} x[n] e^{-jk(2\pi/N)n}$$

→ *Synthesis Equation x[n]=N*ifft(a)*

→ *Analysis Equation a=(1/N)*fft(x)*

4

Lab 3.10 (b) (c)

Suggested in the lab book: measure the number of operations by using the internal flops '**flops**'

This is an obsolete function

- ✓ **Measure the function performance difference by comparing the time cost by functions rather than numbers of floating point perations.**
- ✓ **Try to keep the PC conditions unchanged while measuring the functions timecosts**

To measure the elapsed time by a function, you can choose one of the followings:

etime

tic, toc

timeit

➢ *etime*

```
t0 = clock;
X = dtfs(x,0)
dtfstime = etime(clock,t0)
t1 = clock;
X2 = fft(x)
ffttime = etime(clock,t1)
```

```
>> help etime
 etime  Elapsed time.
    etime(T1,T0) returns the time in seconds that has
elapsed between  vectors T1 and T0.  The two vectors
must be six elements long, in  the format returned by
CLOCK:

        T = [Year Month Day Hour Minute Second]
```

```
>> help clock
 clock  Current date and time as date vector.
    C = clock returns a six element date vector containing
the current time  and date in decimal form:

       [year month day hour minute seconds]
```

➢ *tic ,toc*

```
tic
X = dtfs(x,0);
toc
tic
X2 = fft(x);
toc
```

```
>> help tic
 tic Start a stopwatch timer.
    tic and TOC functions work together to measure
elapsed time.
    tic, by itself, saves the current time that TOC uses later
to  measure the time elapsed between the two.
```

➢ *timeit*

```
f1 = @()dtfs(x,0);
t1_timeit = timeit(f1)
f2 = @()fft(x)
t2_timeit = timeit(f2)
```

```
>> help timeit
 timeit Measures time required to run function.
    T = timeit(F) measures the typical time (in seconds)
required to run  the function specified by the function
handle F that takes no input  argument.
```

To measure performance, it is recommended to use the timeit or tic and toc functions

If your PC is so good that the time cost by fft is always 0,
try to increase the value of N. ☺
Or, put your codes in a for-loop, … …

Lab 3.10 （g）

- **Periodic convolution:**

  Suppose x and y are two periodic signals with common period N, the periodic convolution between x and y is defined as

$$x[n] \circledast y[n] = \sum_{k=<N>} x[k]y[n-k]$$

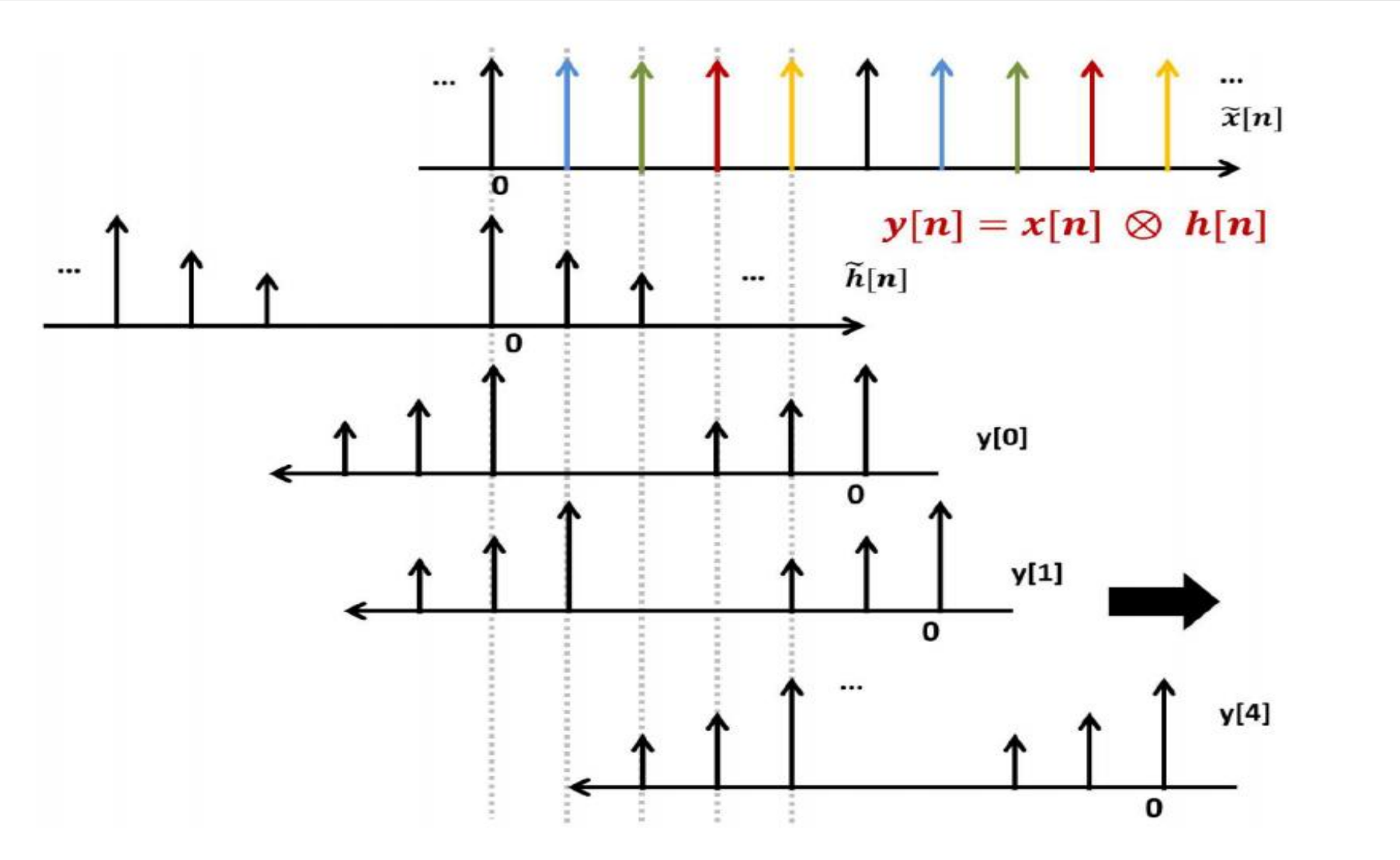
- **Suppose $x[n] \rightarrow a_k$ and $y[n] \rightarrow b_k$, then**

$$x[n] \circledast y[n] \rightarrow \mathrm{Na_k b_k} \quad \text{and} \quad x[n]y[n] \rightarrow a_k \circledast b_k$$

➢ Periodic Convolution with the FFT

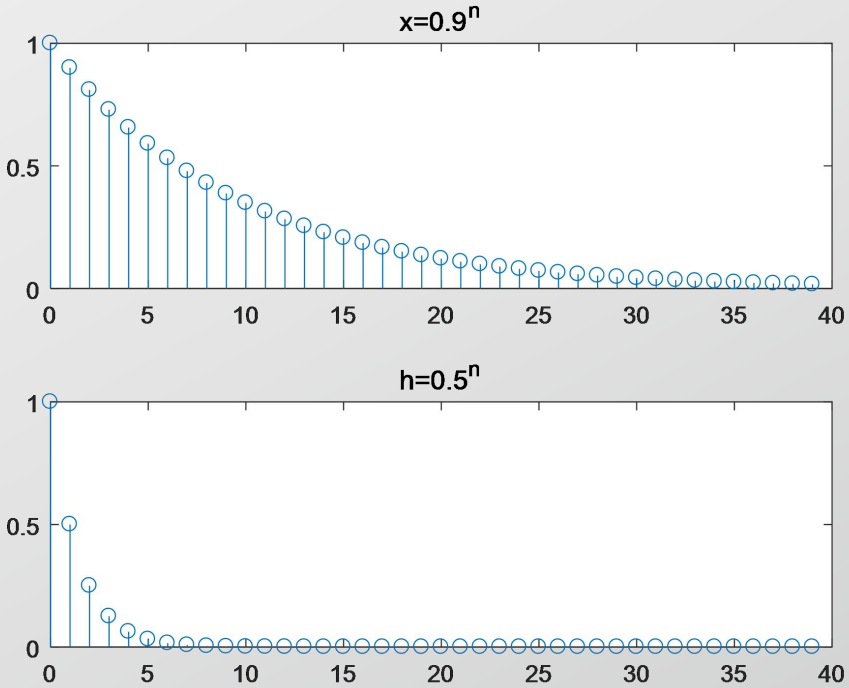
Periodic convolution in time domain is equivalent to multiplication in frequency domain

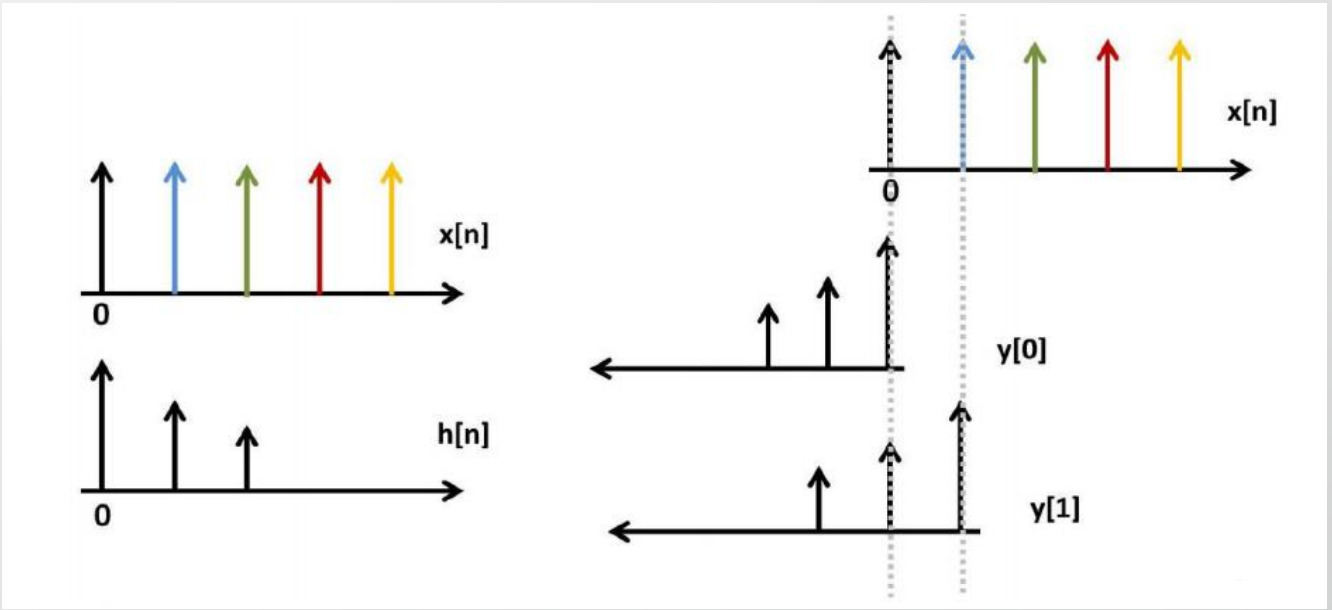
$$y[n] = \sum_{r=<N>} x[r]h[n-r] = \sum_{r=0}^{N-1} x[r]h[n-r] = x[0]h[n] + x[1]h[n-1] + \ldots . x[N-1]h[n-(N-1)]$$



Lab 3.10 (g)

```
N=40;
n=0:1:N-1;
x=0.9.^(n);
h=0.5.^(n);
figure
subplot(2,1,1),stem(n,x),title('x=0.9^n');
subplot(2,1,2),stem(n,h),title('h=0.5^n');
```



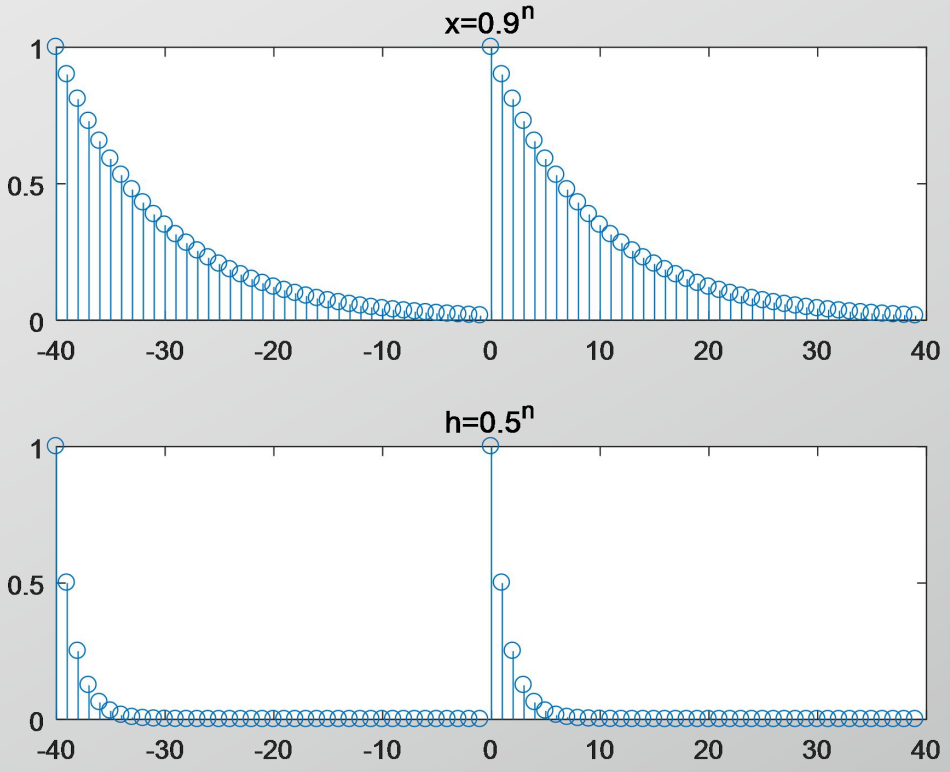


conv(x,h)

Lab 3.10 （g）

$$y[n] = \sum_{r=<N>} x[r]h[n-r] = \sum_{r=0}^{N-1} x[r]h[n-r] = x[0]h[n] + x[1]h[n-1] + \ldots . x[N-1]h[n-(N-1)]$$

```
N=40;
n=0:1:N-1;
x=0.9.^(n);
h=0.5.^(n);
figure
subplot(2,1,1),stem(-40:39,[x,x]),title('x=0.9^n');
subplot(2,1,2),stem(-40:39,[h,h]),title('h=0.5^n');
```

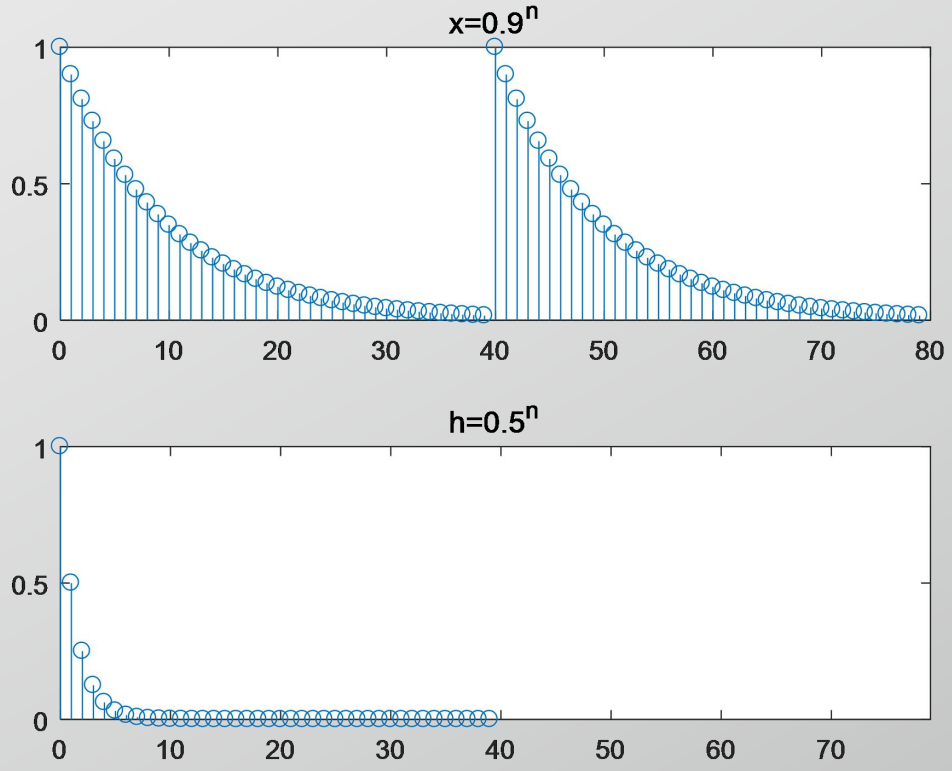


Calculate Periodic Convolutions Using Linear Convolutions

Lab 3.10 （g）

conv([x,x],h)

```
N=40;
n=0:1:N-1;
x=0.9.^(n);
h=0.5.^(n);
figure
subplot(2,1,1),stem(0:79,[x,x]),title('x=0.9^n');
subplot(2,1,2),stem(0:39,h),title('h=0.5^n'),xlim([0 79]);
```



Calculate Periodic Convolutions Using Linear Convolutions