Lecture 12

Fourier Analysis

Ye Ding (丁烨)

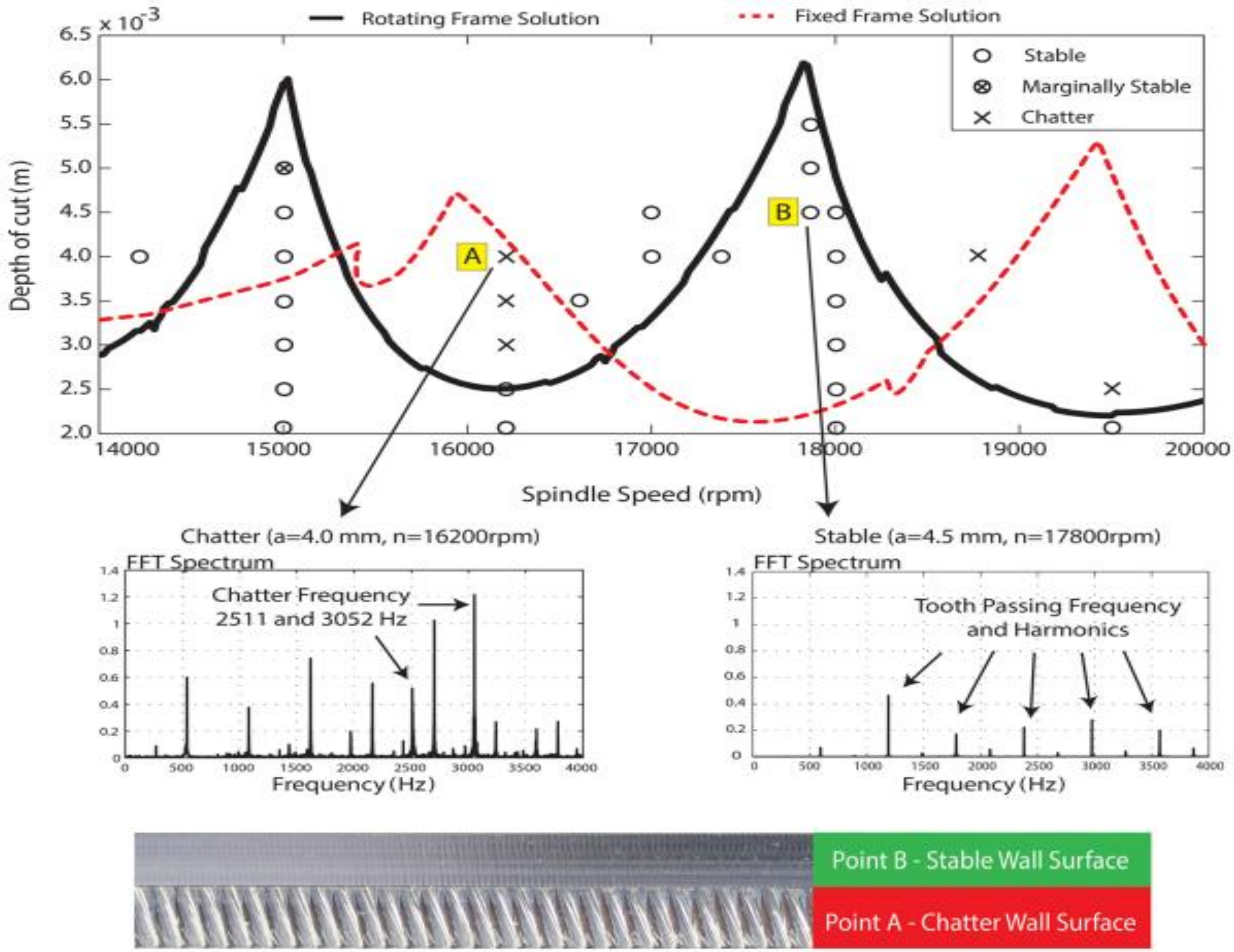
Email: y.ding@sjtu.edu.cn

School of Mechanical Engineering

Shanghai Jiao Tong University

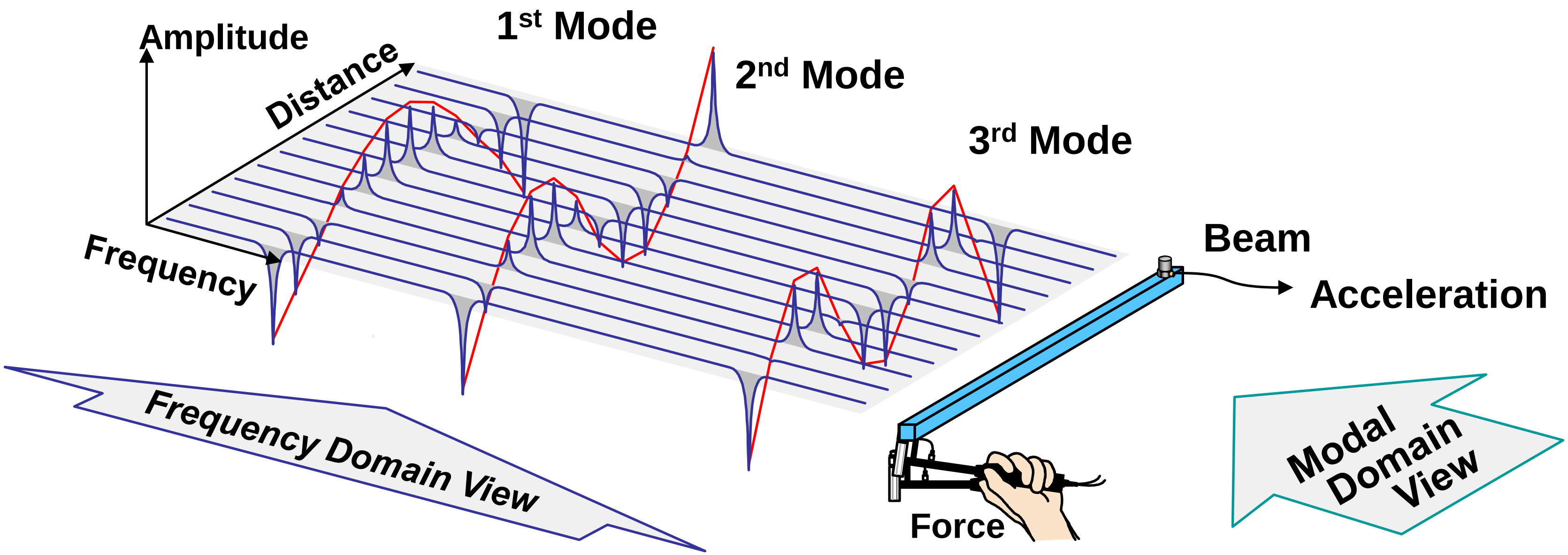
Fourier Analysis

Motivation: from Machining Processes



Fourier Analysis

- **Motivation: from Experimental Modal Analysis**



Time Domain: F(t) → Mechanical System → X(t)

Frequency Domain: $F(\omega) \times [H(\omega)] = X(\omega)$

Fourier Analysis

- **Motivation: from PDEs**

Heat Equation:

$$u_t(x,t) = u_{xx}(x,t), \qquad t > 0, 0 \le x \le \pi,$$

$$u(x,0) = f(x), \qquad 0 \le x \le \pi,$$

$$u(0,t) = A, \qquad u(\pi,t) = B.$$

Separation of Variables:

$$u(x,t) = \sum_{k=1}^{\infty} u_k(x,t)$$

$$= \sum_{k=1}^{\infty} b_k e^{-k^2 t} \sin(kx)$$

Fourier Analysis

- **References for FFT**

[1] Cleve Moler, Numerical Computing with MATLAB, Society for Industrial and Applied Mathematics, 2004. Chapter 8

[2] Timothy Sauer, Numerical Analysis, 2nd ed., Pearson Education, 2012. Chapter 10

[3] Albert Boggess, Francis J. Narcowich, A First Course in Wavelets with Fourier Analysis, Pearson Education, 2002. Chapters 1-3

[4] 李庆扬等，数值分析（第5版），清华大学出版社，2008. 第三章

Fourier Analysis

- **Fourier Analysis**
  - Fourier Series
  - Fourier Transform
  - Discrete Time Fourier Transform (DTFT)
  - Discrete Fourier Transform (DFT)
  - Fast Fourier Transform (FFT)
- **Applications**
  - DFT Interpolation
  - Least Squares Fitting

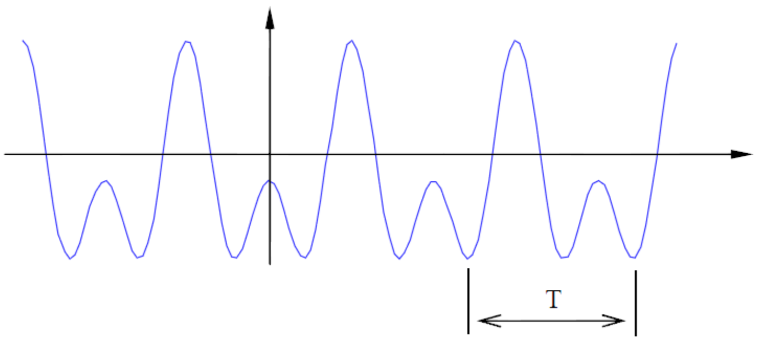
上海交通大学 SHANGHAI JIAO TONG UNIVERSITY

Fourier Series

Fourier series expansion for periodic function

For periodic function:

$$x(t) = x(t+T)$$



Fourier series expansion:

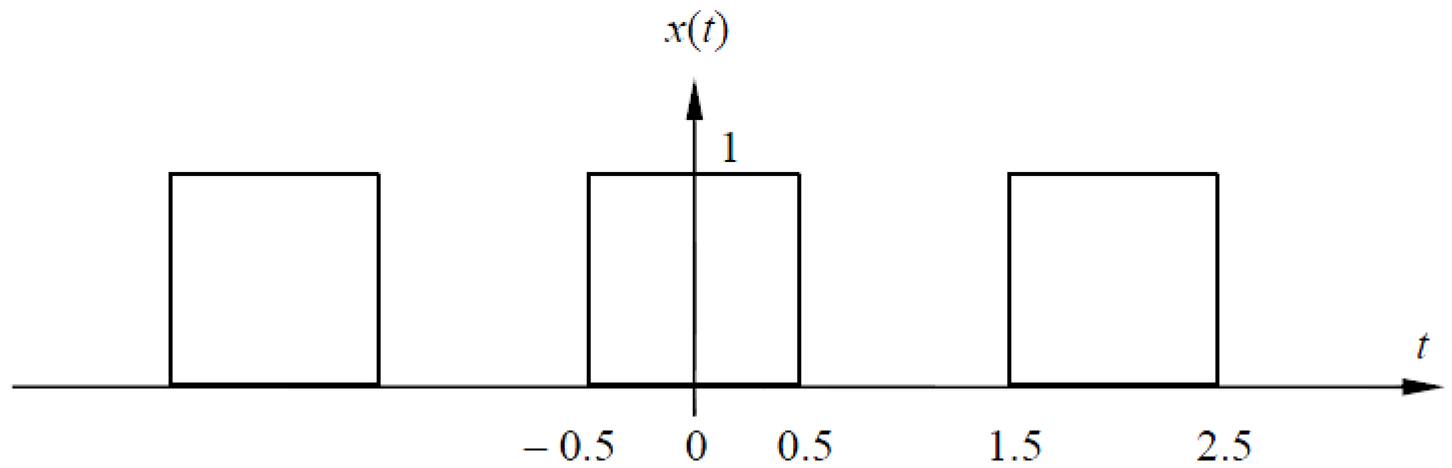
$$x(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty}\left(a_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t)\right)$$

$\omega_0 = 2\pi / T$ **is the fundamental frequency,** T **is the period**

$$\begin{cases} a_0 = \dfrac{2}{T}\displaystyle\int_{-T/2}^{T/2} x(t)dt \ , \qquad a_n = \dfrac{2}{T}\displaystyle\int_{-T/2}^{T/2} x(t)\cos(n\omega_0 t)dt \ , \\ b_n = \dfrac{2}{T}\displaystyle\int_{-T/2}^{T/2} x(t)\sin(n\omega_0 t)dt \end{cases}$$

Fourier Series

- **Case 1: The Fourier series expansion for square wave**



$$T = 2, \qquad \omega_0 = \frac{2\pi}{T} = \pi$$

$$a_0 = 1, \qquad b_n = 0$$

$$a_n = \frac{2}{T}\int_{-T/2}^{T/2} x(t)\cos(n\omega_0 t)dt = \int_{-0.5}^{0.5} \cos n\pi t dt = \frac{2}{n\pi}\sin\frac{n\pi}{2}$$

Fourier Series

- **Case 1: The Fourier series expansion for square wave**

$$x(t)=\frac{1}{2}+\frac{2}{\pi}[\cos\omega_0 t-\frac{1}{3}\cos 3\omega_0 t+\frac{1}{5}\cos 5\omega_0 t+\ldots]$$

$$=\frac{1}{2}+\frac{2}{\pi}\sum_{n=1}^{\infty}\frac{(-1)^{n-1}}{2n-1}\cos(2n-1)\omega_0 t$$

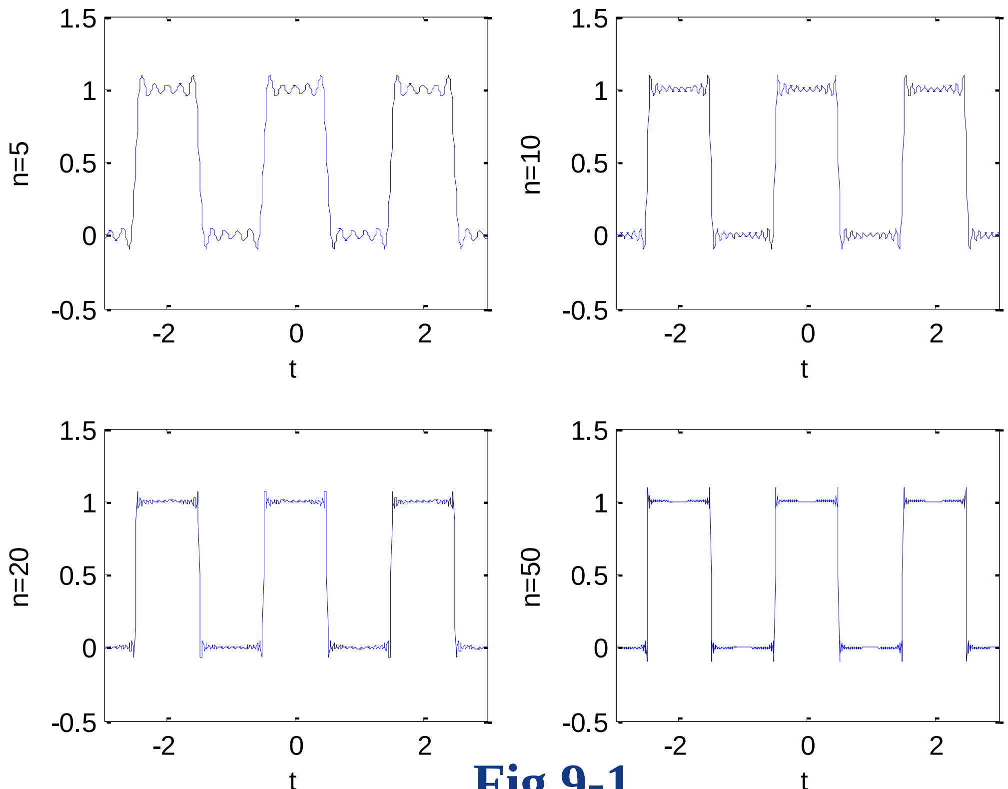


Fig.9-1

Fourier Series

The complex form of Fourier series

Using $e^{\pm i\theta} = \cos\theta \pm i\sin\theta,$ **then**

$$a_n \cos n\omega_0 t = \frac{a_n}{2}\left(e^{-in\omega_0 t} + e^{in\omega_0 t}\right), \qquad b_n \sin n\omega_0 t = \frac{ib_n}{2}\left(e^{-in\omega_0 t} - e^{in\omega_0 t}\right)$$

Fourier series can be rewritten as

$$x(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \frac{1}{2}(a_n + ib_n)e^{-in\omega_0 t} + \sum_{n=1}^{\infty} \frac{1}{2}(a_n - ib_n)e^{in\omega_0 t}$$

Define $c_0 = \dfrac{a_0}{2}, \quad c_{-n} = \dfrac{a_n + ib_n}{2}, \quad c_n = c_{-n}^{*} = \dfrac{a_n - ib_n}{2}$

then $x(t) = c_0 + \sum_{n=1}^{\infty} c_{-n} e^{-in\omega_0 t} + \sum_{n=1}^{\infty} c_n e^{in\omega_0 t}$

or $x(t) = \sum_{n=-\infty}^{\infty} c_n e^{in\omega_0 t}, \qquad c_n = \dfrac{1}{T}\int_{-T/2}^{T/2} x(t) e^{-in\omega_0 t}\, dt$

Fourier Series

- **The complex form of Fourier series: Symbolic computation**

```
syms t
T = 1; % Period of the signal
w = 2*pi/T; % radian frequency omega
for n=0:3
    sList = sprintf('n = %d : ', n);
    disp(sList)
    Cn=(1/T)*int(cos(w*t)*exp(-i*w*n*t), t, 0, 1)
end
```

$$f(t) = \ldots + 0 + \frac{1}{2}e^{-j\omega t} + 0 + \frac{1}{2}e^{j\omega t} + 0 + \ldots = \frac{e^{j\omega t} + e^{-j\omega t}}{2} = \cos\omega t$$

Fourier Series

- **Case 2: The complex form of Fourier series for square pulse**

$$T=2,\quad \omega_0=\frac{2\pi}{T}=\pi,\quad c_0=\frac{1}{2}$$

$$c_n=\frac{1}{T}\int_{-T/2}^{T/2}x(t)e^{-in\omega_0 t}dt=\frac{1}{T}\int_{-0.5}^{0.5}(\cos n\omega_0 t-i\sin n\omega_0 t)dt=\frac{2}{n\omega_0 T}\sin\frac{n\omega_0}{2}$$

$$x(t)=\sum_{n=-\infty}^{\infty}\frac{1}{n\pi}\sin\frac{n\pi}{2}e^{in\omega_0 t}=\frac{1}{2}+\frac{2}{\pi}[\cos\omega_0 t-\frac{1}{3}\cos 3\omega_0 t+\frac{1}{5}\cos 5\omega_0 t+\ldots]$$

The frequency spectrum is represented as Tc_n

For square pulse,

$$Tc_n=\int_{-T/2}^{T/2}x(t)e^{-in\omega_0 t}dt=\frac{2}{n\omega_0}\sin\frac{n\omega_0}{2}$$

Fourier Series

Case 2: The complex form of Fourier series for square pulse

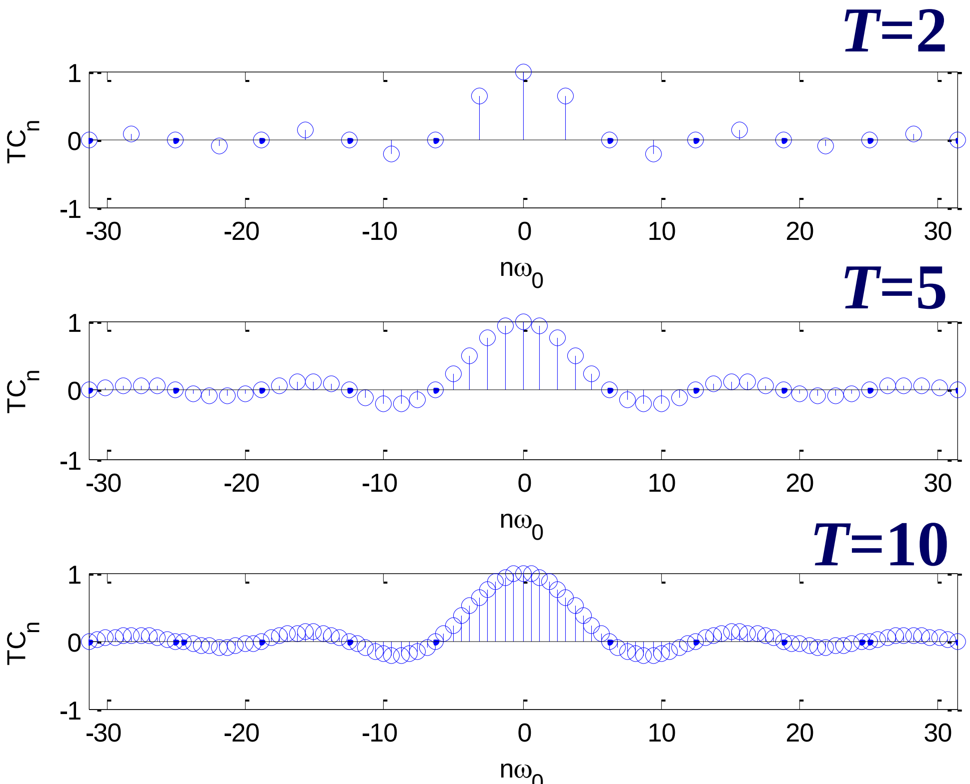


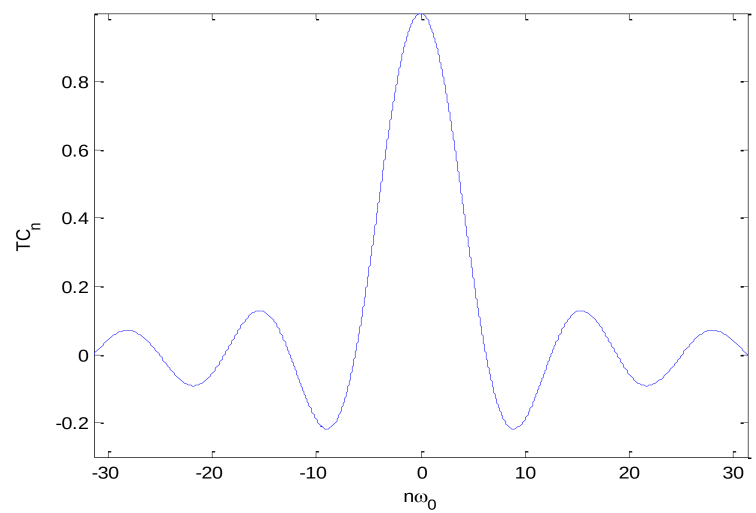
Fig.9-2

if $T \to \infty, n\omega_0 = n\frac{2\pi}{T} = n\Delta\omega = \omega$

then $\lim\limits_{T\to\infty} Tc_n = \lim\limits_{T\to\infty} \int_{-T/2}^{T/2} x(t) e^{-in\omega_0 t} dt$

$$= \int_{-\infty}^{\infty} x(t) e^{-i\omega t} dt$$

$$= X(\omega) = \frac{2}{\omega} \sin\frac{\omega}{2}$$



Fourier Transform

- **Definition of Fourier transform**

$$X(\omega)=\int_{-\infty}^{+\infty} x(t)e^{-i\omega t}dt \quad ,-\infty<\omega<+\infty$$

- **Inverse Fourier transform**

$$x(t)=\frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{i\omega t}d\omega$$

- **Parseval's theorem**

$$\int_{-\infty}^{\infty} x^2(t)dt=\frac{1}{2\pi}\int_{-\infty}^{\infty}\left|\mathrm{X}(\omega)\right|^2 d\omega=\int_{-\infty}^{\infty}\left|X(f)\right|^2 df$$

This quantity is known as the total power in a signal.

Fourier Transform

- **Fourier transform: Symbolic computation**

$$e^{-\frac{1}{2}t^2} \Leftrightarrow \sqrt{2\pi}\, e^{-\frac{1}{2}\omega^2}$$

```
syms t v w x;
ft = exp(-t^2/2);
Fw = fourier(ft)
```

Fw =

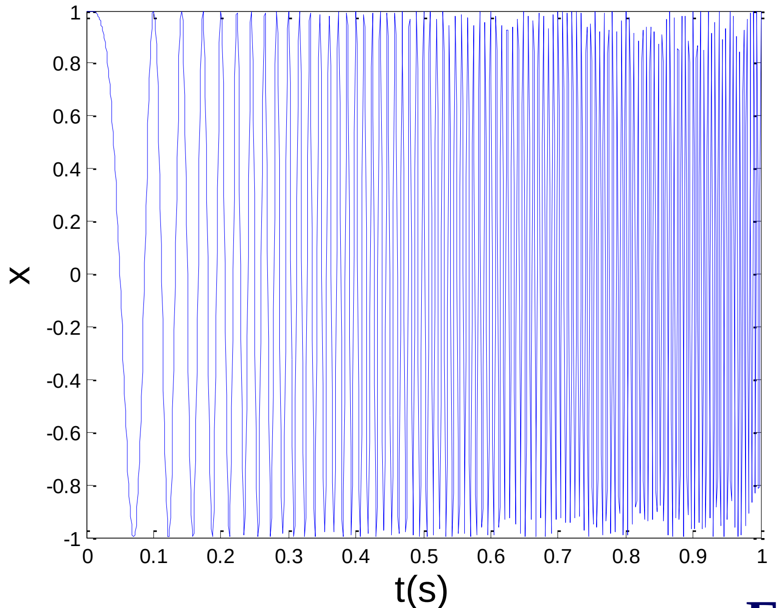
2^(1/2)*pi^(1/2)*exp(-w^2/2)

```
% Check answer by computing the Inverse using "ifourier"
ft = ifourier(Fw)
```

Fourier Transform

The Parseval theorem

For linear sweep signal, *x*=chirp(*t*,0,1,200), the time duration is 1s, and the frequency range is 0Hz-200Hz



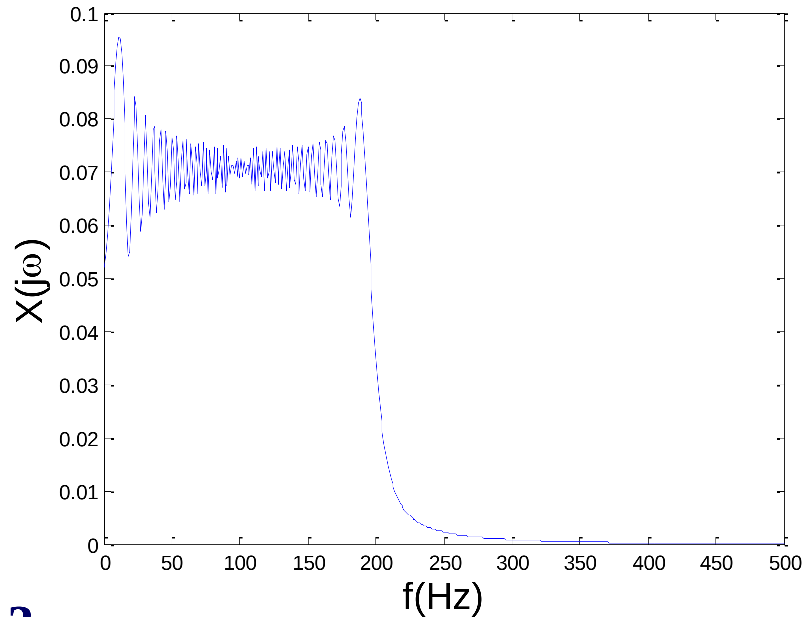


Fig.9-3

$$\int_{-\infty}^{\infty} x^2(t)dt = 0.5098$$

$$\int_{-\infty}^{\infty} |X(f)|^2 df = 0.5093$$

上海交通大学 SHANGHAI JIAO TONG UNIVERSITY

Fourier Transform

Fourier transform of classical functions

For square pulse

According to the definition of Fourier transform, the Fourier transform is

$$X(\omega) = \int_{-\infty}^{+\infty} x(t) e^{-i\omega t} dt = \frac{A \sin \omega T / 2}{\omega / 2}$$

```
syms A T t w
f = A * (heaviside(t + T/2) - heaviside(t - T/2));
Fw = fourier(f,t,w);
Fw_s = simplify(Fw)
pretty(Fw_s)
```

Fourier Transform

Fourier transform of classical functions

For square pulse

According to the definition of Fourier transform, the Fourier transform is

$$X(\omega) = \int_{-\infty}^{+\infty} x(t)e^{-i\omega t}dt = \frac{A\sin \omega T/2}{\omega/2}$$

If $A=1/T$, which means that the area of square pulse is 1, then

$$X(\omega) = \frac{\sin \omega T/2}{\omega T/2}$$

The square pulse is named as unit impulse function δ(t)

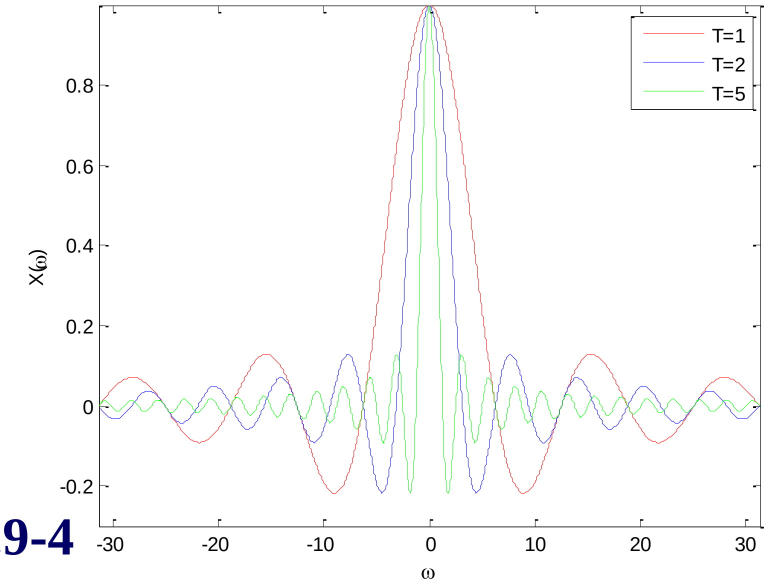


Fig.9-4

Fourier Transform

- **Unit impulse function $\delta(t)$**

$$\delta(t)=\begin{cases}\infty, & t=0\\ 0, & t\neq 0\end{cases}$$

$$\int_{-\infty}^{\infty}\delta(t)dt=1$$

$$\int_{-\infty}^{\infty}f(t)\delta(t)dt=f(0)$$

$$\int_{-\infty}^{\infty}f(t)\delta(t-t_0)dt=f(t_0)$$

The Fourier transform of impulse function δ(t) is $\mathcal{F}\left[\delta(t)\right]=1$

```
syms t
fourier(dirac(t))
```

Fourier Transform

- **Harmonic function cos$\omega_0 t$**

The Fourier transform of Harmonic function cos$\omega_0 t$ is

$$\cos(\omega_0 t) \leftrightarrow \pi\left[\delta(\omega+\omega_0)+\delta(\omega-\omega_0)\right]$$

- **For example ω_0 =10*2π Hz**

Unilateral amplitude spectrum

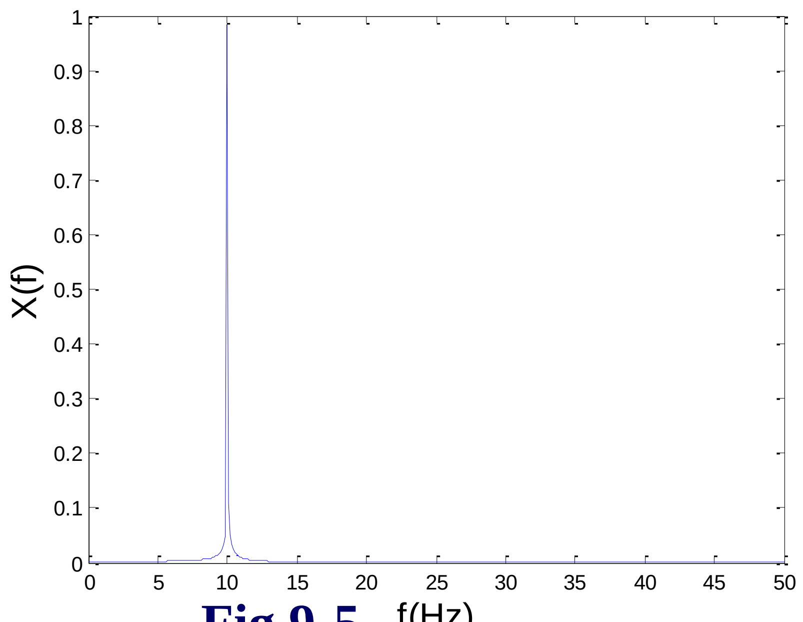


Fig.9-5

Fourier Transform

- **Several important properties of Fourier transform**

(1) $X(-\omega)=X^*(\omega)$

(2) $\mathcal{F}[x(t-t_0)]=e^{-i\omega t_0}X(\omega)$

(3) $\mathcal{F}[x(t)e^{i\omega t_0}]=X(\omega-\omega_0)$

(4) $\mathcal{F}[h(t)*x(t)]=H(\omega)X(\omega)$

(5) $\mathcal{F}[h(t)x(t)]=\dfrac{1}{2\pi}H(\omega)*X(\omega)$

(6) $\mathcal{F}(\dot{x}(t))=\mathrm{i}\omega X(\omega)$

(7) $\mathcal{F}[\int_{-\infty}^{t}x(t)dt]=\dfrac{1}{i\omega}X(\omega)$

```
syms t f(t) t0 w
fourier(f(t-t0),t,w)
```

ans =
exp(-t0*w*i)*fourier(f(t), t, w)

Fourier Transform

Sampling Theorem

Let $x(t)$ be a band-limited signal with $X(j\omega)=0$ for $|\omega|> \omega_M$. Then $x(t)$ is uniquely determined by its samples $x(nT)$, $n=0, \pm 1, \pm 2, \ldots$, if

$$\omega_s > 2\omega_M$$

where,

$$\omega_s = \frac{2\pi}{T}$$

Given these samples, we can reconstruct $x(t)$ by generating a periodic impulse train in which successive impulses have amplitudes that are successive sample values.

Discrete Time Fourier Transform

Discrete Time Fourier Transform (DTFT)

Discrete time series $x[n]$ is the sampling signal $x(n\Delta)$ of continuous time signal $x(t)$, Δ=1. Substituting $x[n]$ into the definition equation of Fourier transform,

$$X(\omega) = \int_{-\infty}^{+\infty} x(t) e^{-i\omega t} dt \quad , -\infty < \omega < +\infty$$

and enabling dt=1, t=$n\Delta$, then DTFT $X(\Omega)$ of $x[n]$ can be obtained.

$$X(\Omega) = \sum_{n=-\infty}^{\infty} x[n] e^{-i\Omega n}$$

DTFT $X(\Omega)$ is a periodic function of Ω, and the period is 2π.

$$X(\Omega) = X(\Omega + 2\pi)$$

Inverse DTFT

$$x[n] = \frac{1}{2\pi} \int_0^{2\pi} X(\Omega) e^{in\Omega} d\Omega = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\Omega) e^{in\Omega} d\Omega$$

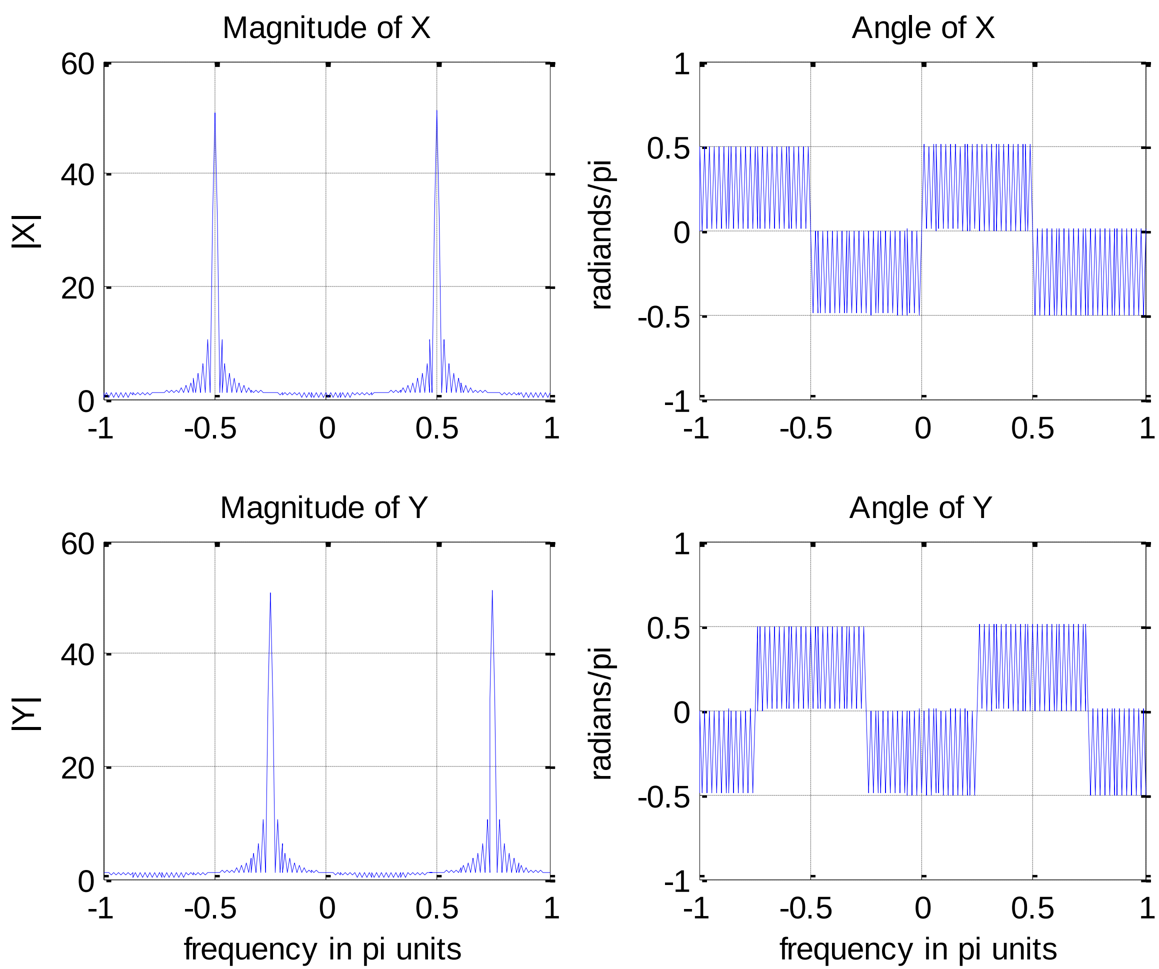
Discrete Time Fourier Transform

Discrete Time Fourier Transform: Example

$$x(n) = \cos(\pi n/2), \quad 0 \leq n \leq 100$$

$$y(n) = e^{j\pi n/4} x(n)$$

Frequency Shift Property!



Discrete Fourier Transform

From DTFT to Discrete Fourier Transform (DFT)

$$X_k = X(\Omega)\Big|_{\Omega=2\pi k/N} = X\left(\frac{2\pi k}{N}\right)$$

X_k is the frequency sampling function of $X(\Omega)$, and the sampling points are $\Omega=2\pi k/N$, $k=0, 1, \ldots, N-1$

Fast Fourier Transform (FFT)

DFT requires N multiplications and N additions for each of the N components for a total of $2N^2$ floating-point operations. If N is large, the amount of calculation is very large. In order to overcome this problem, people discovered FFT algorithms which only have computational complexity $O(N\log_2 N)$

Discrete Fourier Transform

Discrete Fourier Transform (DFT)

Supposing that the discrete time series *x*[*n*] is zero when *n*<0 or *n*≥*N*, the DFT can be defined as

$$X_k = \sum_{n=0}^{N-1} x[n] e^{-i2\pi kn/N}, \quad k = 0,1,\ldots,N-1$$

X_k is the value at f_k of Fourier transform for sampling signal $\sum x(t)\delta(t-n\Delta)$

$$X(f) = \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x(t)\delta(t-n\Delta) e^{-i2\pi ft} dt = \sum_{n=-\infty}^{\infty} x(n\Delta) e^{-i2\pi n\Delta f}$$

Discrete Fourier Transform

Discrete Fourier Transform (DFT)

Because there are only *N* sampling points, the above equation can be rewritten as

$$X(f)=\sum_{n=0}^{N-1}x(n\Delta)e^{-i2\pi n\Delta f}$$

Then, the value of *X* (*f*) at $f=f_k=\dfrac{k}{N\Delta}$ **can be rewritten as**

$$X_k=X(f_k)=\sum_{n=0}^{N-1}x[n]e^{-i2\pi kn/N}$$

Inverse DFT

$$x[n]=\frac{1}{N}\sum_{k=0}^{N-1}X_k e^{i2\pi kn/N}$$

Discrete Fourier Transform

- **Discrete Fourier Transform (DFT) : Matrix form**

DFT of $x = [x_0, \ldots, x_{n-1}]^T$ **is**

the *n*-dimensional vector $y = [y_0, \ldots, y_{n-1}]^T$

$$y_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j \omega^{jk}$$

where $\omega = e^{-i2\pi/n}$ **- twiddle factor**

Discrete Fourier Transform

- **Discrete Fourier Transform (DFT): Matrix form**

DFT of $x = [x_0, \ldots, x_{n-1}]^T$ **is**

the *n*-dimensional vector $y = [y_0, \ldots, y_{n-1}]^T$

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{bmatrix} = \boxed{\frac{1}{\sqrt{n}} \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \cdots & \omega^{n-1} \\ \omega^0 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \omega^0 & \omega^3 & \omega^6 & \cdots & \omega^{3(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{bmatrix}} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n-1} \end{bmatrix}$$

where $\omega = e^{-i2\pi/n}$

Fourier Matrix

SHANGHAI JIAO TONG UNIVERSITY

Discrete Fourier Transform

Discrete Fourier Transform (DFT): Matrix form

Let $\{y_k\}$ be the DFT of $\{x_j\}$, where the x_j are real numbers.

Then (a) y_0 is real

(b) $y_{n-k} = \overline{y}_k$ for $k = 1, \ldots, n-1$

$$F_8 \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 + ib_1 \\ a_2 + ib_2 \\ a_3 + ib_3 \\ a_4 \\ a_3 - ib_3 \\ a_2 - ib_2 \\ a_1 - ib_1 \end{bmatrix} = \begin{bmatrix} y_0 \\ \vdots \\ y_{\frac{n}{2}-1} \\ y_{\frac{n}{2}} \\ \overline{y}_{\frac{n}{2}-1} \\ \vdots \\ \overline{y}_1 \end{bmatrix}$$

Discrete Fourier Transform

Discrete Fourier Transform (DFT): Example

Find the DFT of the vector $x = [1,0,-1,0]^T$

$$\omega = e^{-i\pi/2} = \cos(\pi/2) - i\sin(\pi/2) = -i$$

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 \\ 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega^9 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix}$$

$$= \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

Discrete Fourier Transform

Discrete Fourier Transform (DFT): Example

Find the DFT of the vector $x = [1,0,-1,0]^T$

```
x = [1,0,-1,0]';
n = length(x);
omega = exp(-2*pi*i/n);
j = 0:n-1;
k = j';
F = 1/sqrt(n) * omega.^(k*j);
y = F* x;
```

```
y =
   0.0000 + 0.0000i
   1.0000 + 0.0000i
   0.0000 - 0.0000i
   1.0000 + 0.0000i
```

F'*F = ?

Discrete Fourier Transform

Inverse Discrete Fourier Transform

$$F_n = \frac{1}{\sqrt{n}} \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \cdots & \omega^{n-1} \\ \omega^0 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \omega^0 & \omega^3 & \omega^6 & \cdots & \omega^{3(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{bmatrix}$$

$$y = F_n x$$

$$F_n^{-1} = \frac{1}{\sqrt{n}} \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-(n-1)} \\ \omega^0 & \omega^{-2} & \omega^{-4} & \cdots & \omega^{-2(n-1)} \\ \omega^0 & \omega^{-3} & \omega^{-6} & \cdots & \omega^{-3(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{-(n-1)} & \omega^{-2(n-1)} & \cdots & \omega^{-(n-1)^2} \end{bmatrix}$$

$$x = F_n^{-1} y$$

Fast Fourier Transform

- **Fast Fourier Transform (FFT): Basic idea**

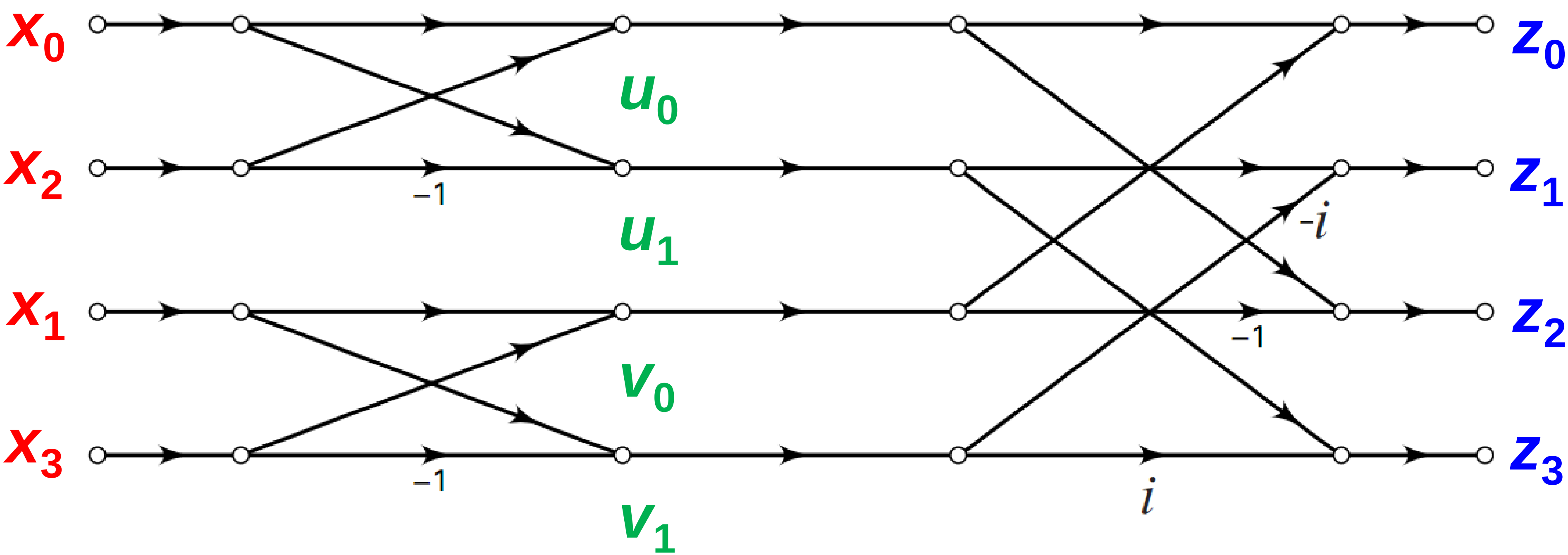
For the case *n* = 4, $\omega = e^{-i2\pi/4} = -i$

the Discrete Fourier Transform is

$$\begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \omega^3 \\ \omega^0 & \omega^2 & \omega^4 & \omega^6 \\ \omega^0 & \omega^3 & \omega^6 & \omega^9 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Fast Fourier Transform

- **Fast Fourier Transform (FFT): Basic idea**



$$u_0 = \mu^0 x_0 + \mu^0 x_2$$
$$u_1 = \mu^0 x_0 + \mu^1 x_2$$
$$v_0 = \mu^0 x_1 + \mu^0 x_3$$
$$v_1 = \mu^0 x_1 + \mu^1 x_3$$

$$z_0 = u_0 + \omega^0 v_0$$
$$z_1 = u_1 + \omega^1 v_1$$
$$z_2 = u_0 + \omega^2 v_0$$
$$z_3 = u_1 + \omega^3 v_1$$

Fast Fourier Transform

Fast Fourier Transform (FFT): Basic idea

For the case *n* = 4, $\omega = e^{-i2\pi/4} = -i$

$$\begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \omega^3 \\ \omega^0 & \omega^2 & \omega^4 & \omega^6 \\ \omega^0 & \omega^3 & \omega^6 & \omega^9 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

rearrange the order

$$z_0 = \omega^0 x_0 + \omega^0 x_2 + \omega^0(\omega^0 x_1 + \omega^0 x_3)$$
$$z_1 = \omega^0 x_0 + \omega^2 x_2 + \omega^1(\omega^0 x_1 + \omega^2 x_3)$$
$$z_2 = \omega^0 x_0 + \omega^4 x_2 + \omega^2(\omega^0 x_1 + \omega^4 x_3)$$
$$z_3 = \omega^0 x_0 + \omega^6 x_2 + \omega^3(\omega^0 x_1 + \omega^6 x_3)$$

Fast Fourier Transform

- **Fast Fourier Transform (FFT): Basic idea**

For the case *n* = 4, $\omega = e^{-i2\pi/4} = -i$

$$z_0 = \omega^0 x_0 + \omega^0 x_2 + \omega^0(\omega^0 x_1 + \omega^0 x_3)$$
$$z_1 = \omega^0 x_0 + \omega^2 x_2 + \omega^1(\omega^0 x_1 + \omega^2 x_3)$$
$$z_2 = \omega^0 x_0 + \omega^4 x_2 + \omega^2(\omega^0 x_1 + \omega^4 x_3)$$
$$z_3 = \omega^0 x_0 + \omega^6 x_2 + \omega^3(\omega^0 x_1 + \omega^6 x_3)$$

$$\Downarrow \quad \omega^4 = 1$$

$$z_0 = \underline{(\omega^0 x_0 + \omega^0 x_2)} + \omega^0(\omega^0 x_1 + \omega^0 x_3)$$
$$z_1 = \underline{(\omega^0 x_0 + \omega^2 x_2)} + \omega^1(\omega^0 x_1 + \omega^2 x_3)$$
$$z_2 = \underline{(\omega^0 x_0 + \omega^0 x_2)} + \omega^2(\omega^0 x_1 + \omega^0 x_3)$$
$$z_3 = \underline{(\omega^0 x_0 + \omega^2 x_2)} + \omega^3(\omega^0 x_1 + \omega^2 x_3)$$

Fast Fourier Transform

Fast Fourier Transform (FFT): Basic idea

For the case *n* = 4, $\omega = e^{-i2\pi/4} = -i$

$$
\begin{aligned}
z_0 &= (\omega^0 x_0 + \omega^0 x_2) + \omega^0(\omega^0 x_1 + \omega^0 x_3) \\
z_1 &= (\omega^0 x_0 + \omega^2 x_2) + \omega^1(\omega^0 x_1 + \omega^2 x_3) \\
z_2 &= (\omega^0 x_0 + \omega^0 x_2) + \omega^2(\omega^0 x_1 + \omega^0 x_3) \\
z_3 &= (\omega^0 x_0 + \omega^2 x_2) + \omega^3(\omega^0 x_1 + \omega^2 x_3)
\end{aligned}
\quad\Rightarrow\quad
\begin{aligned}
z_0 &= u_0 + \omega^0 v_0 \\
z_1 &= u_1 + \omega^1 v_1 \\
z_2 &= u_0 + \omega^2 v_0 \\
z_3 &= u_1 + \omega^3 v_1
\end{aligned}
$$

define

$$
\begin{aligned}
u_0 &= \mu^0 x_0 + \mu^0 x_2 \qquad & v_0 &= \mu^0 x_1 + \mu^0 x_3 \\
u_1 &= \mu^0 x_0 + \mu^1 x_2 \qquad & v_1 &= \mu^0 x_1 + \mu^1 x_3
\end{aligned}
$$

where $\mu = \omega^2$

Fast Fourier Transform

- **Fast Fourier Transform (FFT): Basic idea**

For the case *n* = 4, $\omega = e^{-i2\pi/4} = -i$

$$u_0 = \mu^0 x_0 + \mu^0 x_2 \qquad v_0 = \mu^0 x_1 + \mu^0 x_3$$

$$u_1 = \mu^0 x_0 + \mu^1 x_2 \qquad v_1 = \mu^0 x_1 + \mu^1 x_3$$

DFTs with *n* = 2

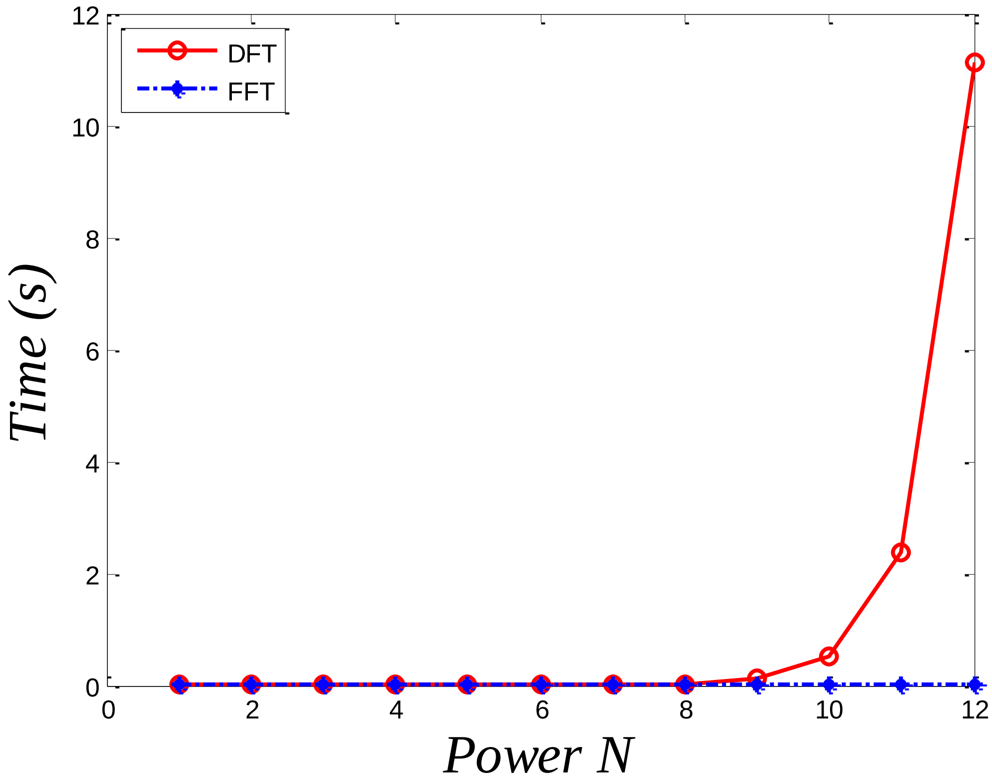
$$u = M_2 \begin{bmatrix} x_0 \\ x_2 \end{bmatrix} \qquad v = M_2 \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}$$

The calculation of the DFT(4) has been reduced to a pair of DFT(2)s plus some extra multiplications and additions.

Fast Fourier Transform

- **Fast Fourier Transform (FFT): Comparison**

```
testTimes = 12;
power_List = 1:testTimes;
N_List = 2.^power_List;
TimeList = zeros(testTimes,2);
for k = power_List
    N = N_List(k);
    x = rand(N,1);
    tic;
    y = DFT_original(x);
    TimeList(k,1) = toc;
    tic;
    y_fft = fft(x)/sqrt(N);
    TimeList(k,2) = toc;
end
```



Fast Fourier Transform

- **[Ref. 2, P.475] Operation Count for FFT**

Let n be a power of 2. Then the Fast Fourier Transform of size n can be completed in $n(2\log_2 n - 1) + 1$ additions and multiplications, plus a division by $\sqrt{n}$.

Fast Fourier Transform

- **For example Touch-Tone Dialing**

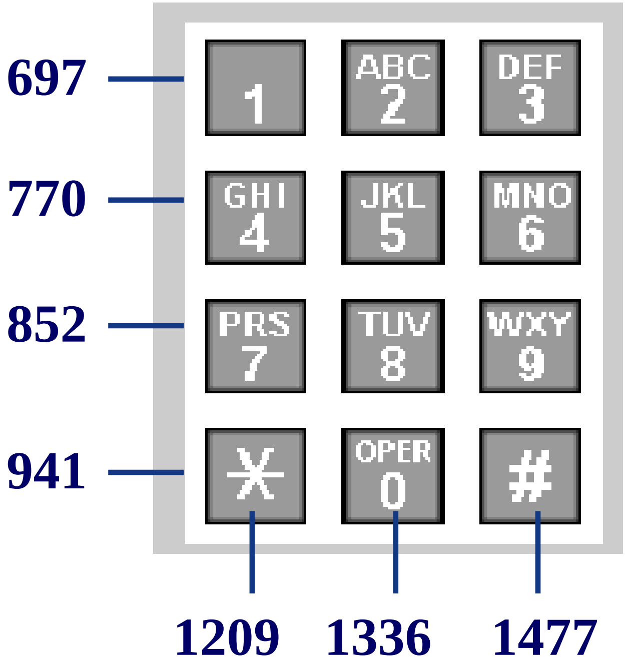


Fig.9-6

Touch-tone dialing is an example of everyday use of Fourier transform. The basis for touch-tone dialing is the Dual Tone Multi-Frequency system. The telephone dialing pad acts as a 4-by-3 matrix. Associated with each row and column is a frequency. These basic frequencies are

Fast Fourier Transform

- **For example Touch-Tone Dialing**

$$f_r = [697 \;\; 770 \;\; 852 \;\; 941]$$

$$f_c = [1209 \;\; 1336 \;\; 1477]$$

The tone generated by the button in position (*k*, *j*) is obtained by superimposing the two fundamental tones with frequencies $f_r(k)$ and $f_c(j)$.

$$y_1 = \sin(2\pi f_r(k)t), \quad y_2 = \sin(2\pi f_c(j)t)$$

$$y = (y_1 + y_2)/2$$

Fig.9-6 is the display produced by touchtone for the '1' button.

Fast Fourier Transform

- **For example Touch-Tone Dialing**

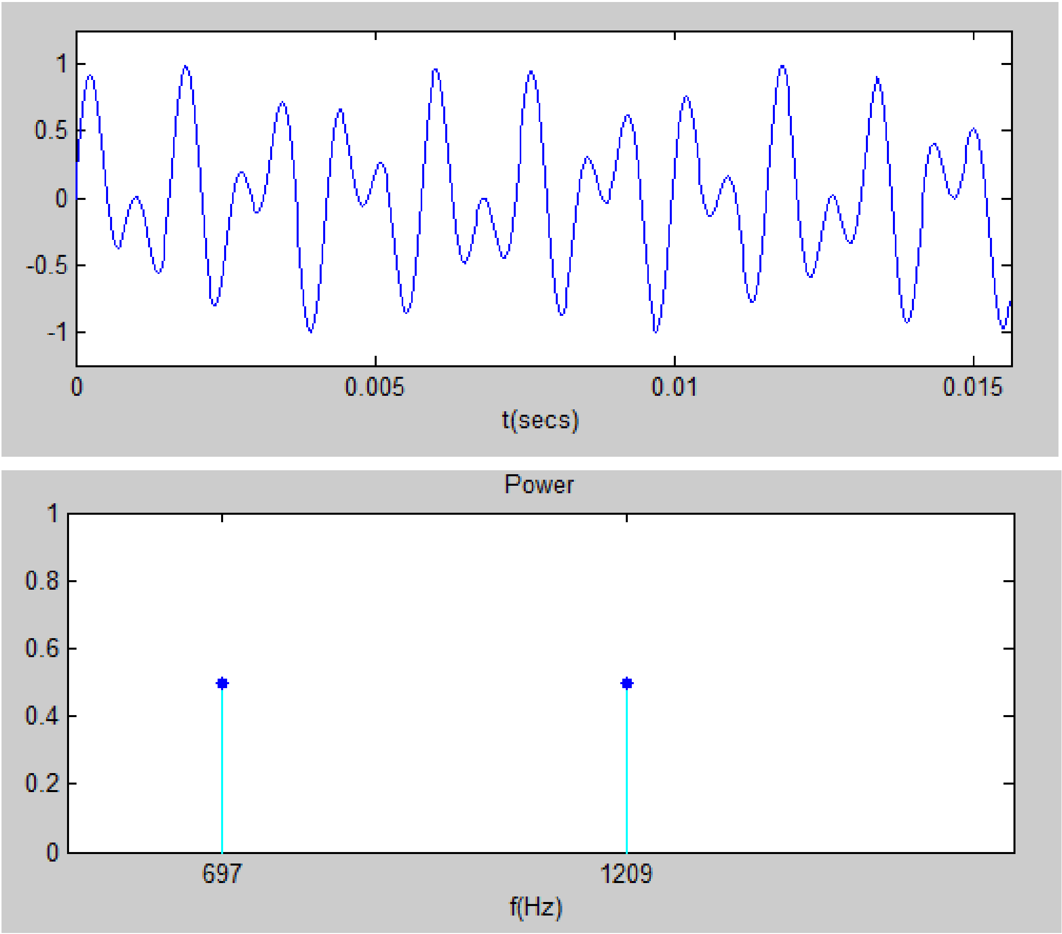
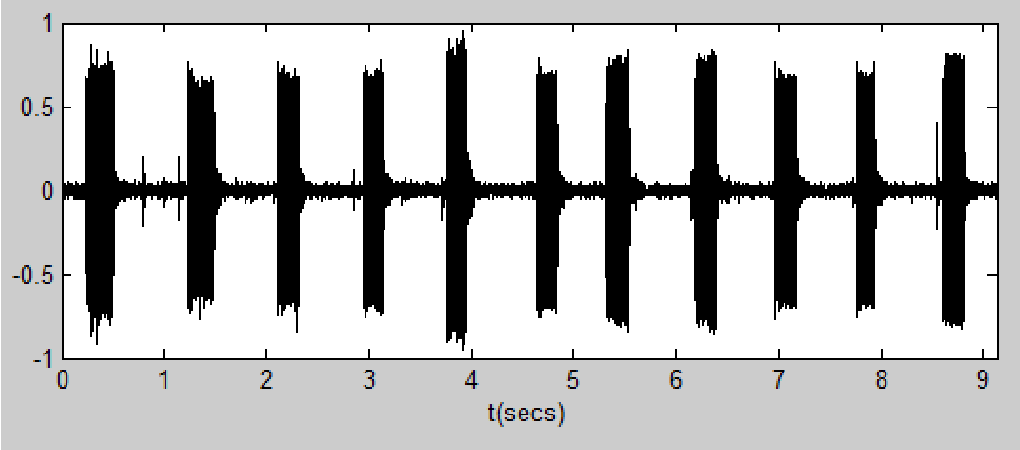


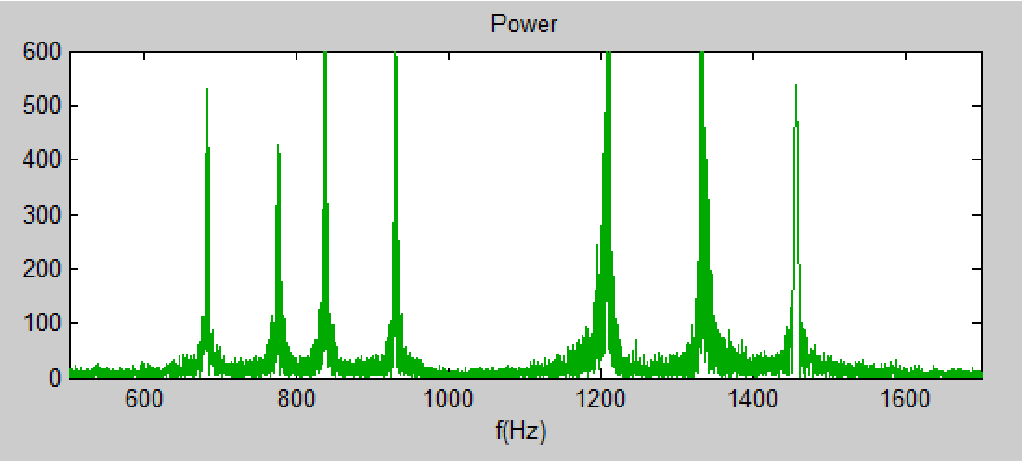
Fig.9-6 The tone generated by the 1 button

Fast Fourier Transform

- **For example Touch-Tone Dialing**

Fig. 9-6 is a plot of the entire signal.



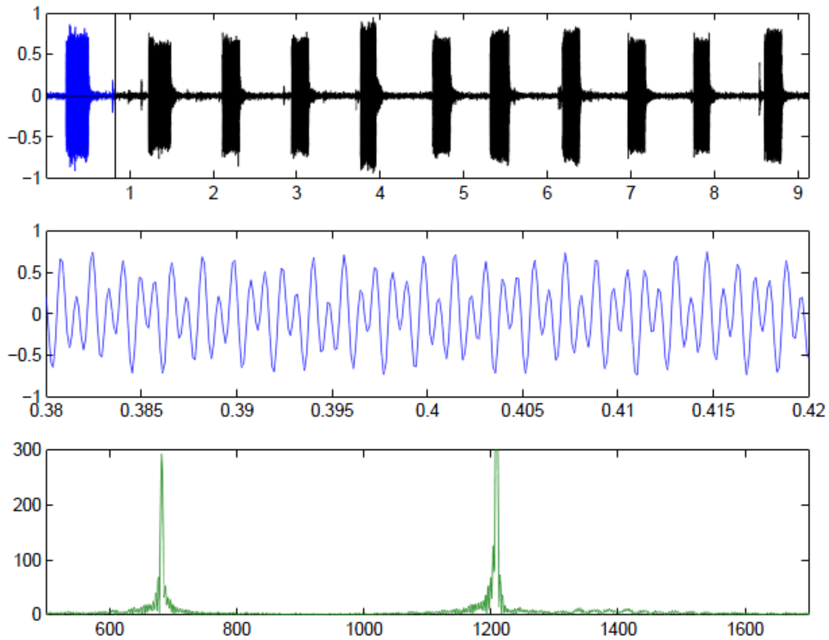


It is easy to see that eleven digits were dialed, but on this scale, it is impossible to determine the specific digits.

Fast Fourier Transform

For example Touch-Tone Dialing

Break the signal into eleven equal segments and analyze each segment separately. Fig.9-6 is the display of the first segment.



only two peaks, and indicate that only two of the basic frequencies come from the '1' button.

Fig.9-6 The first segment and its FFT

Fourier Analysis

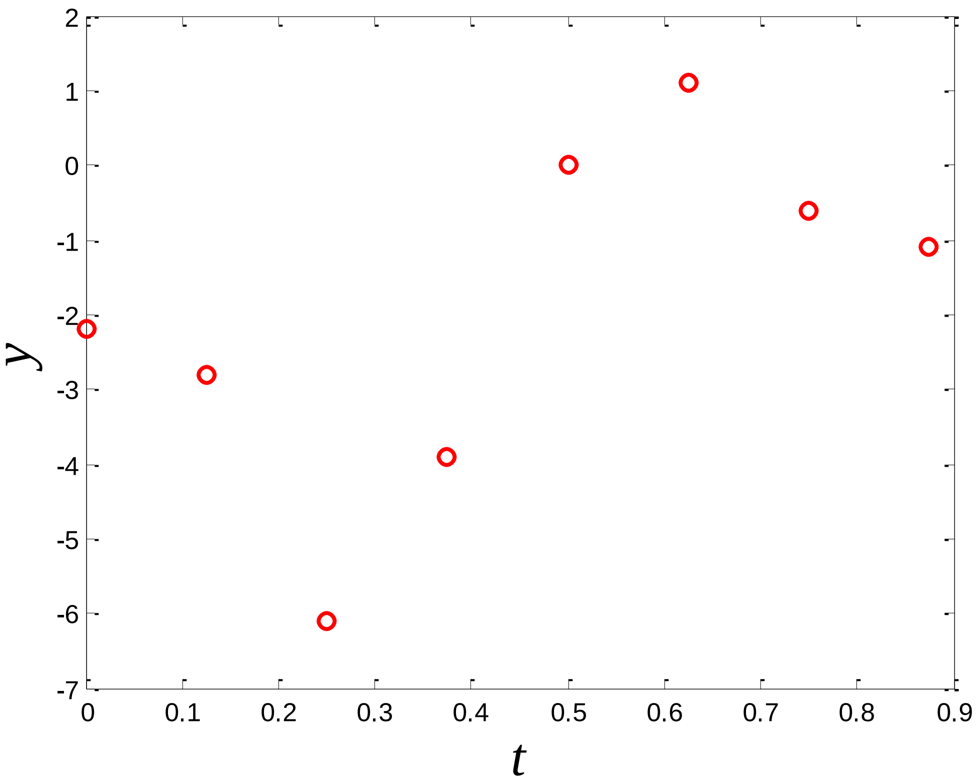
- **Fourier Analysis**
  - **Fourier Series**
  - **Fourier Transform**
  - **Discrete Time Fourier Transform (DTFT)**
  - **Discrete Fourier Transform (DFT)**
  - **Fast Fourier Transform (FFT)**
- **Applications**
  - **DFT Interpolation**
  - **Least Squares Fitting**

DFT Interpolation

DFT Interpolation: Example

Fit the recorded temperatures in a city, as listed in the following table, to a periodic model:

time of day	t	temp (C)
12 mid.	0	-2.2
3 am	$\frac{1}{8}$	-2.8
6 am	$\frac{1}{4}$	-6.1
9 am	$\frac{3}{8}$	-3.9
12 noon	$\frac{1}{2}$	0.0
3 pm	$\frac{5}{8}$	1.1
6 pm	$\frac{3}{4}$	-0.6
9 pm	$\frac{7}{8}$	-1.1



DFT Interpolation

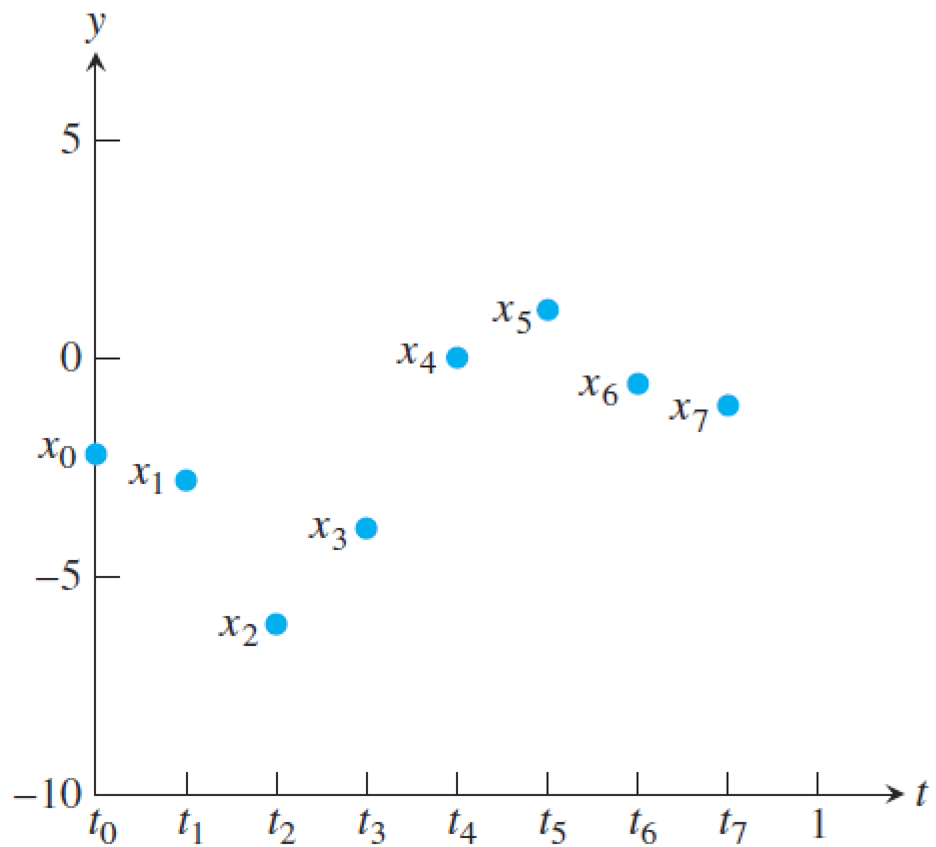
DFT Interpolation Method

Let [c,d] be an interval and let *n* be a positive integer.

$$\Delta t = (d - c)/n$$

$$t_j = c + j\Delta t$$

for $j = 0, \ldots, n-1$



DFT Interpolation

DFT Interpolation Method

$$F_n = \frac{1}{\sqrt{n}} \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \cdots & \omega^{n-1} \\ \omega^0 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \omega^0 & \omega^3 & \omega^6 & \cdots & \omega^{3(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{bmatrix}$$

$$y = F_n x$$

$$F_n^{-1} = \frac{1}{\sqrt{n}} \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-(n-1)} \\ \omega^0 & \omega^{-2} & \omega^{-4} & \cdots & \omega^{-2(n-1)} \\ \omega^0 & \omega^{-3} & \omega^{-6} & \cdots & \omega^{-3(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{-(n-1)} & \omega^{-2(n-1)} & \cdots & \omega^{-(n-1)^2} \end{bmatrix}$$

$$\boxed{x = F_n^{-1} y}$$

上海交通大學 SHANGHAI JIAO TONG UNIVERSITY

DFT Interpolation

DFT Interpolation Method

Let y = F_nx be the DFT of x. $\omega = e^{-i2\pi/n}$

$$\Delta t = (d - c)/n$$

$$t_j = c + j\Delta t$$

$$\text{for } j = 0, \ldots, n-1$$

$$x_j = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} y_k (\omega^{-k})^j$$

$$= \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} y_k e^{i2\pi kj/n}$$

$$= \sum_{k=0}^{n-1} y_k \frac{e^{\frac{i2\pi k(t_j - c)}{d-c}}}{\sqrt{n}}$$

Interpolation of the points (t_j , x_j)!!!

DFT Interpolation

DFT Interpolation Theorem.

Given an interval [c,d] and positive integer n, let $t_j = c + j(d - c)/n$ for $j = 0,\ldots,n - 1$, and let $\mathbf{x} = (x_0, \ldots, x_{n-1})$ denote a vector of n numbers.

Define $\vec{a} + \vec{b}i = F_n x$, where F_n is the Discrete Fourier Transform matrix. Then the complex function

$$Q(t) = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} (a_k + i b_k) e^{i2\pi k(t-c)/(d-c)}$$

satisfies

$$Q(t_j) = x_j \text{ for } j = 0, \ldots, n-1$$

DFT Interpolation

- **DFT Interpolation Theorem.**

Given an interval [c,d] and positive integer n, let $t_j = c + j(d - c)/n$ for $j = 0,\ldots,n - 1$, and let $\mathbf{x} = (x_0, \ldots, x_{n-1})$ denote a vector of n numbers.

Define $\vec{a} + \vec{b}i = F_n x$, where F_n is the Discrete Fourier Transform matrix.

If the x_j are real, the real function

$$P(t) = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} \left(a_k \cos \frac{2\pi k(t - c)}{d - c} - b_k \sin \frac{2\pi k(t - c)}{d - c} \right)$$

satisfies

$$P(t_j) = x_j \text{ for } j = 0, \ldots, n - 1$$

DFT Interpolation

- **DFT Interpolation Theorem: simplified version**

Given an interval [c,d] and an even integer n, let $t_j = c + j(d - c)/n$ for $j = 0,\ldots,n-1$, and let $\mathbf{x} = (x_0, \ldots, x_{n-1})$ denote a vector of n numbers.

Define $\vec{a} + \vec{b}i = F_n x$, where F_n is the Discrete Fourier Transform matrix.

If the x_j are real, the real function

$$P_n(t) = \frac{a_0}{\sqrt{n}} + \frac{2}{\sqrt{n}} \sum_{k=1}^{n/2-1} \left(a_k \cos \frac{2k\pi(t-c)}{d-c} - b_k \sin \frac{2k\pi(t-c)}{d-c} \right) + \frac{a_{n/2}}{\sqrt{n}} \cos \frac{n\pi(t-c)}{d-c}$$

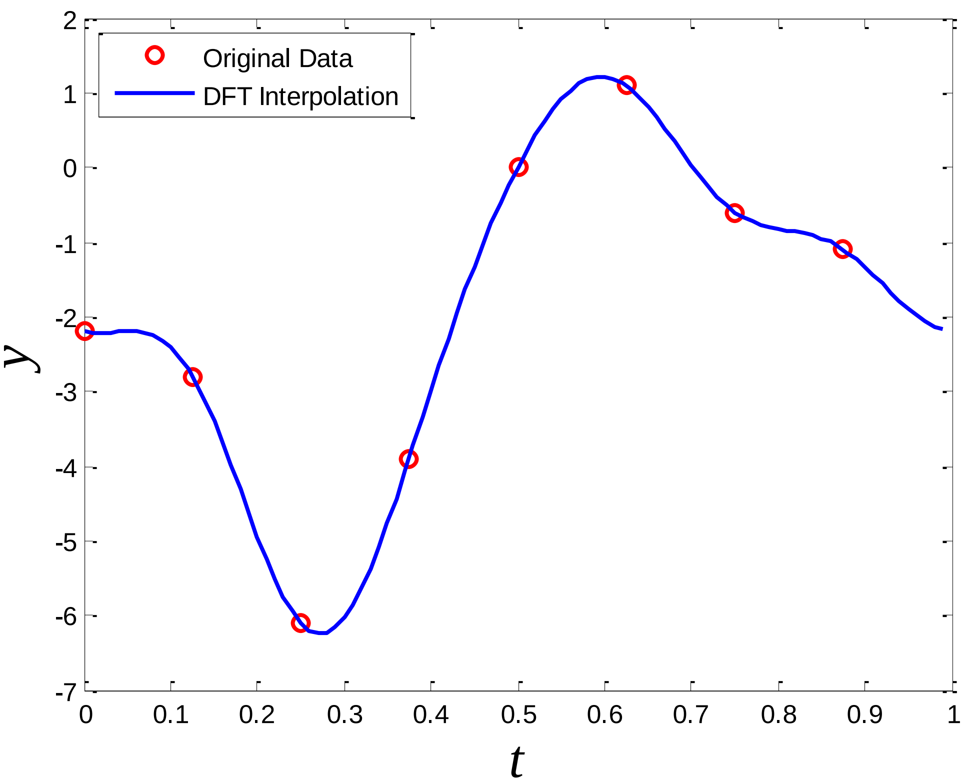
satisfies $P_n(t_j) = x_j$ for $j = 0, \ldots, n-1$.

DFT Interpolation

- **DFT Interpolation: Example**

Fit the recorded temperatures in a city, as listed in the following table, to a periodic model:

time of day	t	temp (C)
12 mid.	0	-2.2
3 am	$\frac{1}{8}$	-2.8
6 am	$\frac{1}{4}$	-6.1
9 am	$\frac{3}{8}$	-3.9
12 noon	$\frac{1}{2}$	0.0
3 pm	$\frac{5}{8}$	1.1
6 pm	$\frac{3}{4}$	-0.6
9 pm	$\frac{7}{8}$	-1.1



Least Squares Fitting

Least Squares Approximation Theorem

Let [c,d] be an interval, let $m < n$ be even positive integers, $\boldsymbol{x}=(x_0, \ldots, x_{n-1})$ a vector of n real numbers, and let $t_j = c + j(d - c)/n$ for $j = 0,\ldots,n - 1$.

Let $\{a_0, a_1, b_1, a_2, b_2, \ldots, a_{n/2-1}, b_{n/2-1}, a_{n/2}\} = \boldsymbol{F}_n\boldsymbol{x}$ be the interpolating coefficients for $\boldsymbol{x}$ so that

$$x_j = P_n(t_j) = \frac{a_0}{\sqrt{n}} + \frac{2}{\sqrt{n}} \sum_{k=1}^{\frac{n}{2}-1} \left(a_k \cos \frac{2k\pi(t_j - c)}{d - c} - b_k \sin \frac{2k\pi(t_j - c)}{d - c} \right) + \frac{a_{\frac{n}{2}}}{\sqrt{n}} \cos \frac{n\pi(t_j - c)}{d - c}$$

for $j = 0, \ldots, n - 1$.

上海交通大学 SHANGHAI JIAO TONG UNIVERSITY

Least Squares Fitting

Least Squares Approximation Theorem

Let [c,d] be an interval, let $m < n$ be even positive integers, $\boldsymbol{x}=(x_0, \ldots, x_{n-1})$ a vector of n real numbers, and let $t_j = c + j(d - c)/n$ for $j = 0,\ldots,n - 1$.

Let $\{a_0, a_1, b_1, a_2, b_2, \ldots, a_{n/2-1}, b_{n/2-1}, a_{n/2}\} = \boldsymbol{F}_n \boldsymbol{x}$ be the interpolating coefficients for $\boldsymbol{x}$ so that

$$P_m(t) = \frac{a_0}{\sqrt{n}} + \frac{2}{\sqrt{n}} \sum_{k=1}^{\frac{m}{2}-1} \left(a_k \cos \frac{2k\pi(t-c)}{d-c} - b_k \sin \frac{2k\pi(t-c)}{d-c} \right) + \frac{2a_{\frac{m}{2}}}{\sqrt{n}} \cos \frac{n\pi(t-c)}{d-c}$$

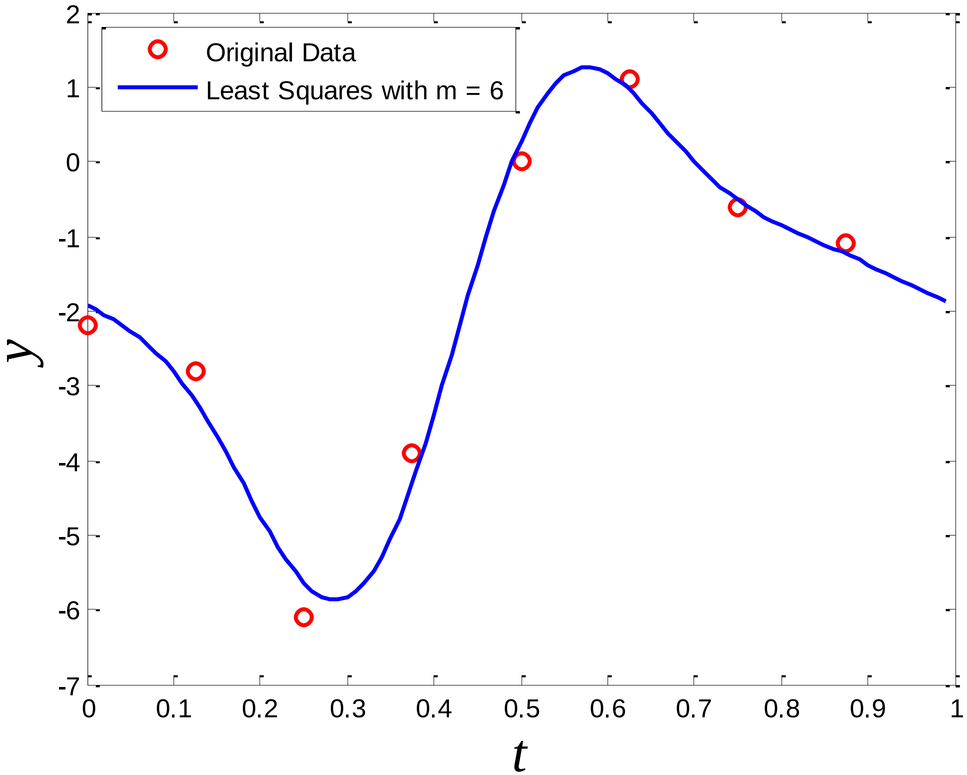
is the best least squares fit of order m to the data.

Least Squares Fitting

DFT Least Squares: Example

Fit the recorded temperatures in Washington, D.C., on January 1, 2001, as listed in the following table, to a periodic model:

time of day	t	temp (C)
12 mid.	0	-2.2
3 am	$\frac{1}{8}$	-2.8
6 am	$\frac{1}{4}$	-6.1
9 am	$\frac{3}{8}$	-3.9
12 noon	$\frac{1}{2}$	0.0
3 pm	$\frac{5}{8}$	1.1
6 pm	$\frac{3}{4}$	-0.6
9 pm	$\frac{7}{8}$	-1.1



MATLAB Built-in Functions

MATLAB Built-in Functions for FFT

***Y* = fft(*X*)** **returns the discrete Fourier transform (DFT) of vector *X*, computed with a fast Fourier transform (FFT) algorithm.**

***Y* = fft(*X*, *n*)** **returns the *n*-point DFT. If the length of *X* is less than *n*, *X* is padded with trailing zeros to length *n*. If the length of *X* is greater than *n*, the sequence *X* is truncated.**

***X*= ifft(Y)** **returns the inverse discrete Fourier transform (DFT) of vector *X*, computed with a fast Fourier transform (FFT) algorithm.**

***X* = ifft(*Y*, *n*)** **returns the n-point inverse DFT of vector *X*.**

Summary

- **Fourier Analysis**
  - ✓ Fourier Series
  - ✓ Fourier Transform
  - ✓ Discrete Time Fourier Transform (DTFT)
  - ✓ Discrete Fourier Transform (DFT)
  - ✓ Fast Fourier Transform (FFT)
- **Applications**
  - ✓ DFT Interpolation
  - ✓ Least Squares Fitting

Thank You !