数字信号处理

周治国

2023.10

第四章 快速傅里叶变换

§4-8 线性调频 Z 变换
(Chirp-Z Transform)

一、问题的提出

$$\forall x(n), n=0,1,\ldots,N-1 \longleftrightarrow X(k) \overset{\Delta}{=} DFT[x(n)] = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}kn} = X(z_k)\Big|_{z_k=e^{j\frac{2\pi}{N}k}}$$

i) N=ML，2^v → FFT算法（基-2，统一，分裂基）

$$k=0,1,\ldots,N-1$$

ii) $FFT \rightarrow X(z_k)\Big|_{z_k=e^{j\frac{2\pi}{N}k}}$, $\quad k=0,1,\ldots,N-1$

（$X(z)$在 $|z|=1$ 上等间隔取样值）

问题：

1) $\exists X(z_k)\Big|_{|z_k|\neq 1}$, $\quad k=0,1,\ldots,M-1$?

2) $\exists X(k)$, $\quad k=0,1,\ldots,M-1, M<N$?

3) $N \neq ML$（质数），$\exists X(k)$, $\quad k=0,1,\ldots,M-1$?

Chirp-Z 变换

§ 4-8 线性调频 Z 变换
(Chirp-Z Transform)

二、算法原理

$\forall x(n),\quad 0 \le n \le N-1 \leftrightarrow$

$$X(z)=\sum_{n=0}^{N-1} x(n) z^{-n}$$

令

$$z_k \overset{\Delta}{=} AW^{-k}, \qquad k=0,1,\ldots,M-1$$

$$A \overset{\Delta}{=} A_0 e^{j\theta_0}$$

$$W \overset{\Delta}{=} W_0 e^{-j\phi_0}, \qquad k=0,1,\ldots,M-1$$

$$z_k = A_0 e^{j\theta_0} \cdot W_0^{-k} \cdot e^{jk\phi_0} \quad k=0,1,\ldots,M-1$$

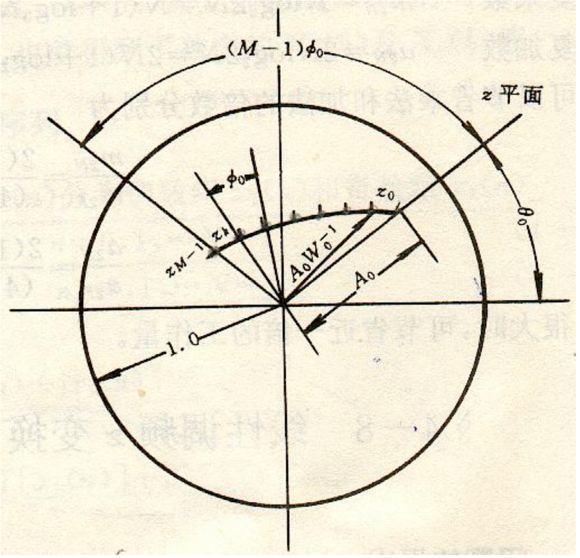
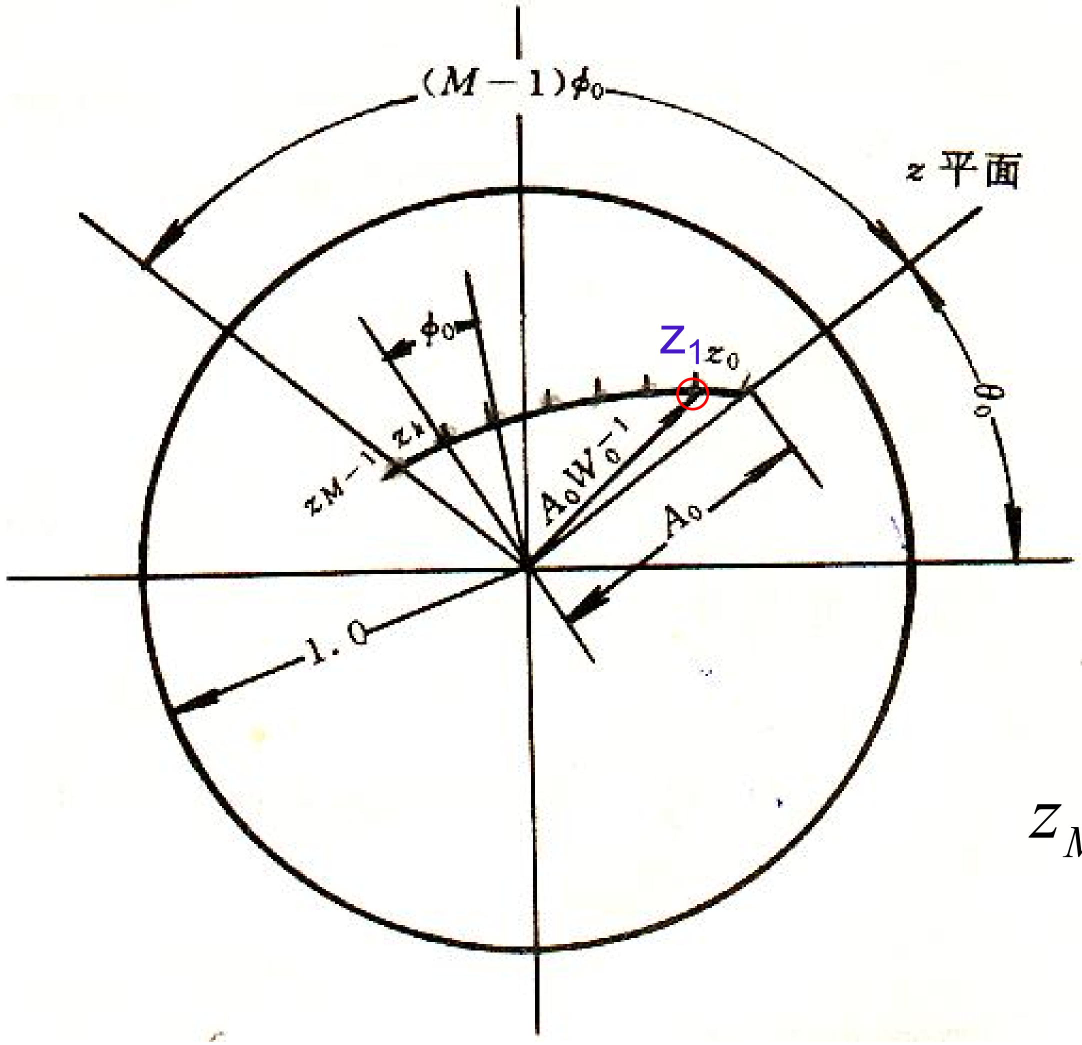


图4-26(P.152)



$$z_0 = A_0 e^{j\theta_0} \overset{\Delta}{=} A_0\angle\theta_0$$

$$z_1 = A_0 W_0^{-1} e^{j(\theta_0+\phi_0)}$$

$$z_k = A_0 W_0^{-k} e^{j(\theta_0+k\phi_0)}$$

$$z_{M-1} = A_0 W_0^{-(M-1)} e^{j[\theta_0+(M-1)\phi_0]}$$

图4-26(P.152)

参数几何意义

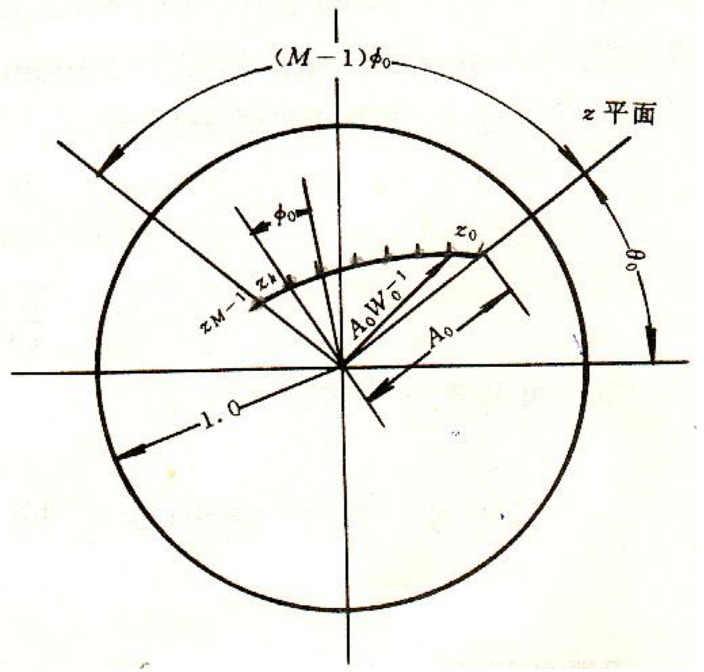
1) A_0： $|z_0|, (A \le 1)$, 取样起始点的矢量长度

2) θ_0： $\arg\{z_0\}, (>0/<0)$, 取样起始点的相角（角频率）

3) ϕ_0： 取样点z_k, z_{k+1}间的角频率差

$\phi_0 > 0$, z_k的路径为逆时针旋转

$\phi_0 < 0$, z_k的路径为顺时针旋转



4) W_0： 取值决定z_k的路径是向内/外盘旋

$W_0 < 1$, z_k的路径是向外弯曲

$W_0 > 1$, z_k的路径是向内弯曲

$W_0 = 1$, z_k的路径是半径为A_0的一段圆弧

$A_0 = 1$时，即单位圆上的一部分

1) $A_0 = 1,\ \theta_0 = 0$

2) $W_0 = 1,\ \phi_0 = \dfrac{2\pi}{N} \longrightarrow X(z_k) = X(k) = DFT[x(n)]$

3) $M = \mathrm{N}$ $\qquad k = 0,1,\ldots,N-1$

∴DFT也可视为CZT的一种特例

一般情况：

$$X(z_k) = \sum_{n=0}^{N-1} x(n) A^{-n} W^{nk} \qquad 0 \le k \le M-1 \qquad \textbf{(4-62)}$$

利用公式：

$$nk = \frac{1}{2}[n^2 + k^2 - (k-n)^2]$$

式(4-62)变为：

$$nk=\frac{1}{2}[n^2+k^2-(k-n)^2]$$

$$X(z_k)=\sum_{n=0}^{N-1}x(n)A^{-n}W^{nk}$$

$$=\sum_{n=0}^{N-1}x(n)A^{-n}W^{\frac{n^2}{2}}W^{-\frac{1}{2}(k-n)^2}W^{\frac{k^2}{2}}$$

$$=W^{\frac{k^2}{2}}\sum_{n=0}^{N-1}[x(n)A^{-n}W^{\frac{n^2}{2}}]W^{-\frac{1}{2}(k-n)^2}$$

$$=W^{\frac{k^2}{2}}\sum_{n=0}^{N-1}f(n)h(k-n) \qquad k=0,1,\ldots,M-1$$

(4-65)

式中：

$$f(n)\stackrel{\Delta}{=}x(n)A^{-n}W^{\frac{n^2}{2}} \qquad \boxed{n=0,1,\ldots,N-1}$$

$$h(n)\stackrel{\Delta}{=}W^{-\frac{n^2}{2}}=\left(e^{j\Phi_0}\right)^{\frac{n^2}{2}} \qquad n=?$$

$W_0=1$时

$$\boxed{\begin{array}{l} n=0,1,\ldots,N-1 \\ k=0,1,\ldots,M-1 \\ -N+1\le -n\le 0 \\ -N+1\le k-n\le M-1 \end{array}}$$

角位移$\frac{n^2\Phi_0}{2}$对时间序数n的微分值为$n\Phi_0$

瞬时频率随时间成线性变化 $\rightarrow$ *Chirp Signal* $\rightarrow$ *CZT*

$$X(z_k)=\sum_{n=0}^{N-1}x(n)A^{-n}W^{nk}$$

$$=W^{\frac{k^2}{2}}\sum_{n=0}^{N-1}f(n)h(k-n)$$

$$k=0,1,\ldots,M-1$$

$$f(n)\overset{\Delta}{=}x(n)A^{-n}W^{\frac{n^2}{2}}$$

$$h(n)\overset{\Delta}{=}W^{-\frac{n^2}{2}}=\left(e^{j\Phi_0}\right)^{\frac{n^2}{2}}$$

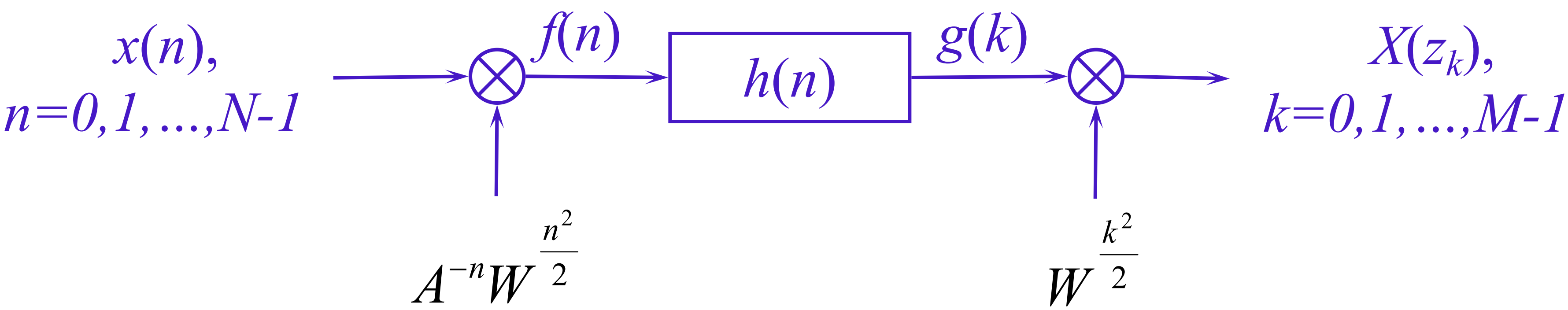


图4-27 CZT的运算流程图

注意这里的g(k)，是因为定义是$X(z_k)$，是频域序列

按常规卷积定义是x(n)*h(n)=y(n)，是时域序列

三、运算/实现步骤：

$$x(n),\ n=0,1,\ldots,N-1 \longrightarrow \bigotimes \xrightarrow{f(n)} \boxed{h(n)} \xrightarrow{g(k)} \bigotimes \longrightarrow X(z_k),\ k=0,1,\ldots,M-1$$

$$A^{-n}W^{\frac{n^2}{2}} \qquad\qquad W^{\frac{k^2}{2}}$$

(1)要求$[X(z_k)]$

$$A_0,\theta_0,W_0,\phi_0 \rightarrow A^{-n}W^{\frac{n^2}{2}},\qquad n=0,1,\ldots,N-1$$

$$W^{-\frac{n^2}{2}} \qquad\qquad \bigotimes \leftarrow x(n),\quad 0\le n\le N-1$$

$$h(n),\ \forall n \qquad\qquad f(n),\ 0\le n\le N-1 \qquad\qquad \times\text{—}N$$

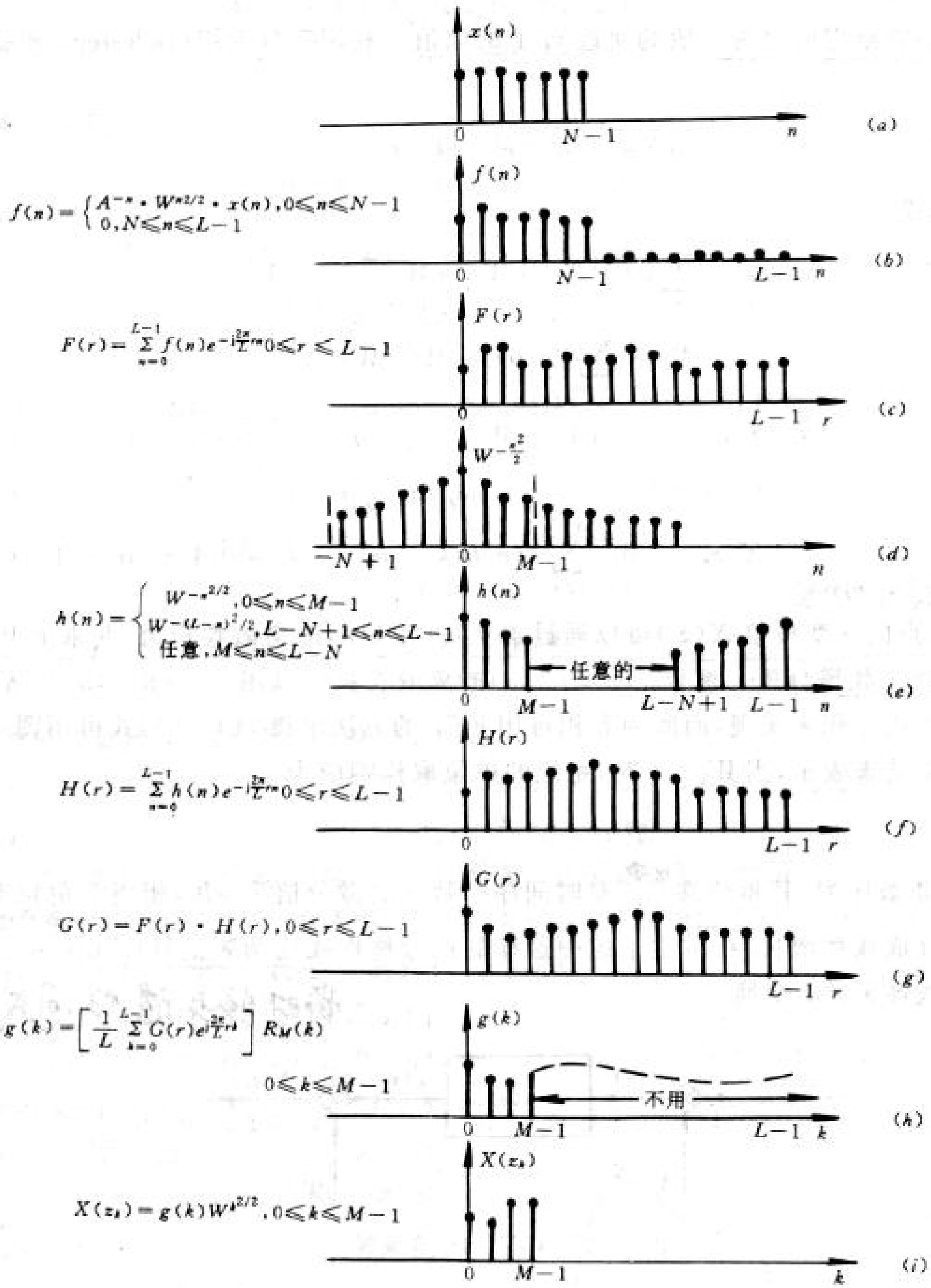
(2)计算 $f(n)*h(n)$， $n=0,1,\ldots,M-1$

$$\begin{matrix} f(n),0\le n\le N-1 \\ h(n),-(N-1)\le n\le M-1 \end{matrix} \xrightarrow[L>N+M-1]{\text{补零至}L\text{点}} \begin{matrix} f'(n) \\ h'(n) \end{matrix} \longrightarrow f(n)*h(n)$$

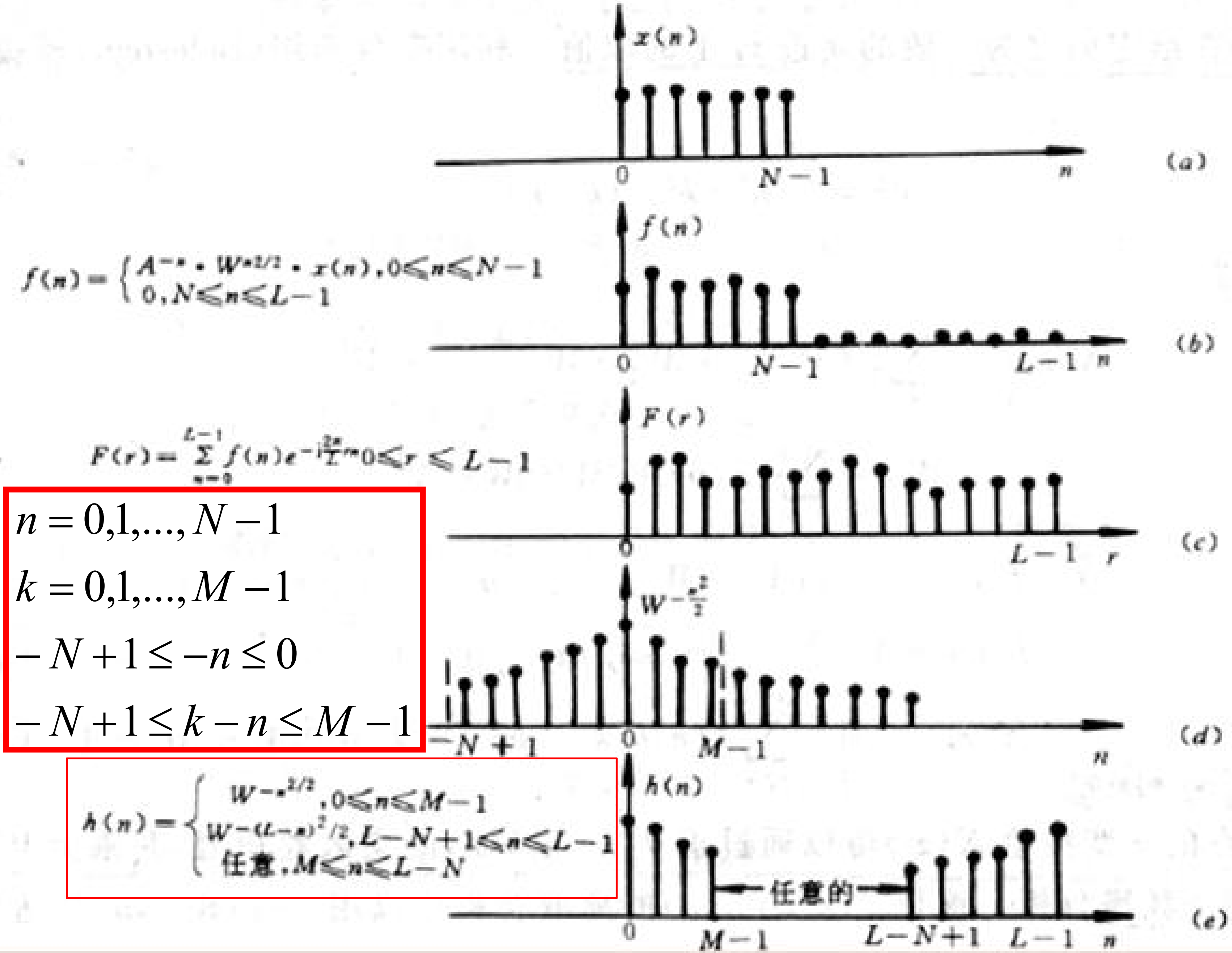
$$L=2^{\nu}$$

$$\times\text{—}3\frac{L}{2}\log_2^l+L$$

(3)计算 $F(r)H(r)$



x(n)
0
N−1
n
(a)
f(n)
f(n) = { A^{-n} · W^{n²/2} · x(n), 0≤n≤N−1 ; 0, N≤n≤L−1
L−1
(b)
F(r)
F(r) = Σ_{n=0}^{L−1} f(n) e^{−j(2π/L)rn} 0≤r≤L−1
r
(c)
W^{−n²/2}
−N+1
M−1
(d)
h(n)
h(n) = { W^{−n²/2}, 0≤n≤M−1 ; W^{−(L−n)²/2}, L−N+1≤n≤L−1 ; 任意, M≤n≤L−N
任意的
L−N+1
(e)
H(r)
H(r) = Σ_{n=0}^{L−1} h(n) e^{−j(2π/L)rn} 0≤r≤L−1
(f)
G(r)
G(r) = F(r) · H(r), 0≤r≤L−1
(g)
g(k)
g(k) = [(1/L) Σ_{k=0}^{L−1} G(r) e^{j(2π/L)rk}] R_M(k)
0≤k≤M−1
不用
k
(h)
X(z_k)
X(z_k) = g(k) W^{k²/2}, 0≤k≤M−1
(i)



$$n=0,1,\ldots,N-1$$

$$k=0,1,\ldots,M-1$$

$$-N+1\le -n\le 0$$

$$-N+1\le k-n\le M-1$$

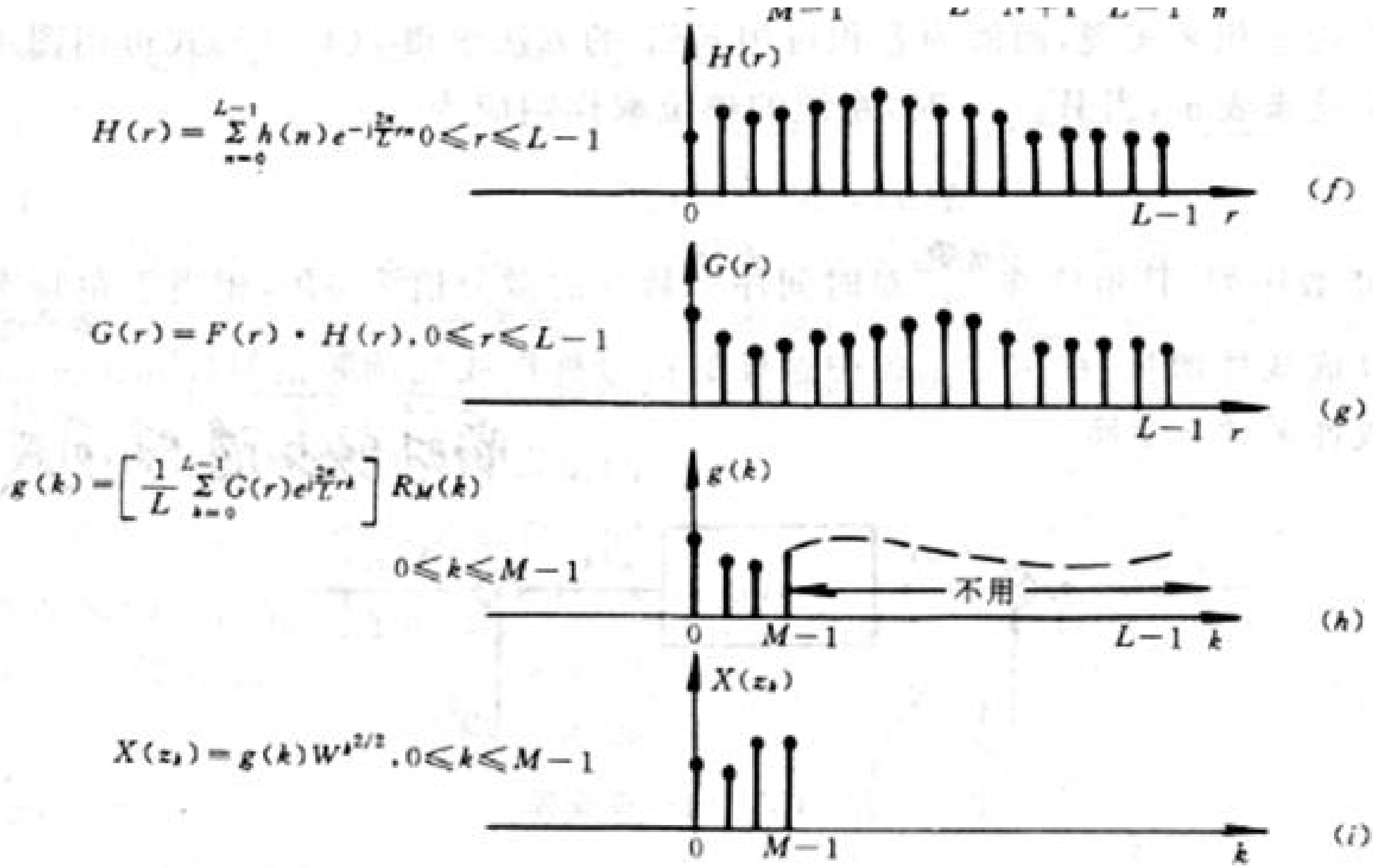


图 4—28 CZT 的波形图

四、运算量估算

$$*: \frac{3}{2}L\log_2^L + N + L + M$$

(M,N>50→CZT优于直接计算)

五、CZT算法的特点

1) $\forall\, x(n), \quad 0 \le n \le N-1$

$\exists\, X(z_k), \quad 0 \le k \le M-1$

$M \ne N,\ |z_k| \ne 1$

2) N,M均可为质数 → 任意情况

3) 取样起始点z_0任选：

$X(z_k), \quad k = 0,1,\dots,M-1$

$z_0 \ne A_0 \to \theta_0 \ne 0$

4) ϕ_0 可任意取值

↓

z_k, z_{k+1} 的角间隔（频率）任意

↓

频率分辨率可变

→ 进行窄带高分辨分析

$\phi_0 = \dfrac{2\pi}{M}, \theta_0 \ne 0$

$M > N, A_0 = 1, W_0 = 1$

$M \ne pq$

5) $A = 1, W = e^{j\frac{2\pi}{N}}, \forall N$

$$CZT \to X(z_k) = DFT[x(n)], \quad \forall M$$

DFT的推广

$(N \neq pl)$ $\quad k = 0, 1, \ldots, M-1$

§4-10 FFT的应用

一、利用FFT求卷积——快速卷积

结合CZT

注意观察卷积对移位的定义

$$\forall\ x(n) \qquad 0 \le n \le N_1 - 1$$

$$h(n) \qquad 0 \le n \le N_2 - 1$$

$$\exists\ x(n) * h(n) = \sum_{l=0}^{N_1-1} x(l)h(n-l) = \sum_{l=0}^{N_2-1} h(l)x(n-l)$$

$$\begin{matrix} x(n) \\ h(n) \end{matrix} \xrightarrow{\text{补零}} \begin{matrix} x'(n) \\ h'(n) \end{matrix} \xrightarrow{FFT} X'(k)H'(k) \xrightarrow{IFFT}$$

1. $N_1 \approx N_2$
2. $N_1 >> N_2$ 分段卷积
3. $x(n) = x^*(n), \quad h(n) = h^*(n)$

运算量比较：

1.直接卷积：N^2

2.快速卷积：$3N\log_2 N$

思考：补零会造成卷积计算误差吗？

一、利用FFT求卷积——快速卷积计算步骤

$(1) x(n)\ N_1\ h(n)\ N_2$

$y(n) = x(n) * h(n)$

(2)补零$N \geq N_1 + N_2 - 1 \quad N = 2^{v}$

$x'(n) \quad h'(n)$

$$y'(n) = x'(n) \otimes h'(n) = \sum_{k=0}^{N-1} x'(k) h'((n-k))_N R_N(n)$$

$(3) FFT : x'(n) \to X'(k) \quad h'(n) \to H'(k)$

$(4) Y'(k) = X'(k) H'(k)$

$$(5) IFFT : \boxed{y'(n)} = \sum_{k=0}^{N-1} [\frac{1}{N} Y'(k)] W_N^{-nk} = \left[\sum_{k=0}^{N-1} [\frac{1}{N} Y'^{*}(k)] W_N^{nk} \right]^{*}$$

$(6) \boxed{y(n)}$

一、利用FFT求卷积——高效的FFT卷积

$\forall$实序列 $g(n), s(n), h(n) \quad 0 \le n \le N-1$

$$G(k), S(k), H(k) \ 0 \le k \le N-1$$

用一次FFT实现两个卷积运算

$$\begin{cases} y_1(n) = g(n) \otimes h(n) \\ y_2(n) = s(n) \otimes h(n) \end{cases}$$

合成： $p(n) = g(n) + js(n)$

$$\Rightarrow DFT[p(n)] = P(k) = \boxed{G(k) + jS(k)}$$

$$\Rightarrow Y(k) = H(k)P(k)$$

$$\Rightarrow y(n) = IFFT[Y(k)] = p(n) \otimes h(n)$$

$$= [g(n) + js(n)] \otimes h(n) = \boxed{g(n) \otimes h(n) + js(n) \otimes h(n)}$$

因此：$\begin{cases} y_1(n) = g(n) \otimes h(n) = \mathrm{Re}[y(n)] \\ y_2(n) = s(n) \otimes h(n) = \mathrm{Im}[y(n)] \end{cases}$

§4-10 FFT的应用

一、利用FFT求卷积——高效的FFT卷积

应用：

(1)一个系统同时通过两种输入信号

(2)一个系统同时处理长序列分段过滤中的两个片段

(3)一个信号同时通过两个系统

二、利用FFT求相关——快速相关

$$\forall \quad x(n) \qquad 0 \le n \le N_1 - 1$$

$$y(n) \qquad 0 \le n \le N_2 - 1$$

$$\exists \quad z(n) = \sum_{l=0}^{N_1-1} x^*(l) y(n+l) = \sum_{l=0}^{N_2-1} y^*(l) x(n+l)$$

$$\begin{matrix} x(n) \\ y(n) \end{matrix} \xrightarrow{\text{补零}} \begin{matrix} x'(n) \\ y'(n) \end{matrix} \xrightarrow{FFT} X'^*(k) Y'(k) \xrightarrow{IFFT} z(n)$$

1. $N_1 \approx N_2$

2. $N_1 >> N_2$

3. $x(n) = x^*(n), \quad y(n) = y^*(n)$

$x(n) = y(n) \rightarrow$ 自相关

1. $N_1 \approx N_2$

2. $N_1 >> N_2$

3. $x(n) = x^*(n)$

二、利用FFT求相关——快速相关计算步骤

$(1) x(n)\ N_1\ \ y(n)\ N_2$

$$z(n)=\sum_{k=0}^{N_1} x^*(k)y(n+k)$$

(2)补零$N \ge N_1 + N_2 - 1 \quad N = 2^v$

$x'(n) \quad y'(n)$

$(3) FFT: x'(n) \to X'(k) \quad y'(n) \to Y'(k)$

$(4) Z(k) = X'^*(k)Y'(k)$

$$(5) IFFT: z'(n) = \sum_{k=0}^{N-1}[\frac{1}{N}Z(k)]W_N^{-nk} = \left[\sum_{k=0}^{N-1}[\frac{1}{N}Z^*(k)]W_N^{nk}\right]^*$$

$(6) z(n)$

往年真题：

4、设有两个有限长实序列，试给出用基-**2 FFT**计算其线性卷积的方法步骤（要求尽量减少乘法运算次数），并与用线性卷积定义直接计算时的运算量做以比较。

往年真题：

5、已知实序列**x(n)**和**y(n)**，长度分别为**N**和**M**，试给出仅用基**-2 FFT**正变换快速计算其线性卷积的方法步骤，要求尽量减少乘法运算次数。

§ 4-11 2-D DFT/FFT 算法

一、2-D DFT

$\forall x(m,n)$

$$0 \le m \le M-1 \qquad 0 \le n \le N-1$$

$$M = 2^{\nu_1} \qquad N = 2^{\nu_2}$$

$$X(k,l) \overset{\Delta}{=} DFT[x(m,n)] = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} x(m,n) W_M^{km} W_N^{ln} \qquad \begin{array}{l} 0 \le k \le M-1 \\ 0 \le l \le N-1 \end{array}$$

$$x(m,n) \overset{\Delta}{=} \frac{1}{MN} \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} X(k,l) W_M^{-km} W_N^{-ln} \qquad \begin{array}{l} 0 \le m \le M-1 \\ 0 \le n \le N-1 \end{array}$$

二、2-D FFT

$$1.\ \forall n, \qquad n = 0,1,\ldots,N-1$$

$$A(k,n) = \sum_{m=0}^{M-1} x(m,n) W_M^{km}, \qquad k = 0,1,\ldots,M-1$$ 列

$$* — N \times \frac{M}{2} \log_2^M$$

$$2.\ \forall k, \qquad k = 0,1,\ldots,M-1$$

$$X(k,l) = \sum_{n=0}^{N-1} A(k,n) W_N^{ln}, \qquad l = 0,1,\ldots,N-1$$ 行

$$* — M \times \frac{N}{2} \log_2^N$$

三、运算量估算

1、2-D FFT

$$M \times \frac{N}{2}\log_2^N + N \times \frac{M}{2}\log_2^M = \frac{MN}{2}(\log_2^N + \log_2^M) = \frac{MN}{2}\log_2^{MN}$$

2、2-D DFT

$$N^2 M^2 = (MN)^2$$

§4-12 FFT的其它形式

Winograd Fourier Transform Algorithm (WFTA):

维络格勒德

N — DFT

2 —

3 —

4 —

5 —

⋮

7 —

8 —

9 —

16 —

$DFT \rightarrow X(k), \quad 0 \le k \le N\text{-}1$

例：$16 \times 9 \times 7 \times 5 = 5040 - DFT$

2 4 8　3 3

算法步骤：

1.利用下标映射，把大N点的DFT化成互素的小N点的DFT；

2.把小N点的DFT化成循环卷积；

3.找出计算小N点卷积的快速算法，从而得到小N点的DFT的快速算法；

4.由小N点的DFT求出大N点的DFT。

x(n)= {x(0), x(1), x(2)…x(14)} = {0,1,2,…,14}

FFT

X(k)= {

1.0e+002 *

1.0500	-0.0750 + 0.3528i	-0.0750 + 0.1685i	-0.0750 + 0.1032i	-0.0750 + 0.0675i
-0.0750 + 0.0433i	-0.0750 + 0.0244i	-0.0750 + 0.0079i	-0.0750 - 0.0079i	-0.0750 - 0.0244i
-0.0750 - 0.0433i	-0.0750 - 0.0675i	-0.0750 - 0.1032i	-0.0750 - 0.1685i	-0.0750 - 0.3528i

}

2D-FFT

x(n)= {

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14

}

X(k)= {

1.0e+002 *

1.0500	-0.0750 + 0.1032i	-0.0750 + 0.0244i	-0.0750 - 0.0244i	-0.0750 - 0.1032i
-0.3750 + 0.2165i	0	0	0	0
-0.3750 - 0.2165i	0	0	0	0

}

x(n)= {x(0), x(1), x(2)…x(14)} = {0,1,2,…,14}

FFT

X(k)= {

1.0e+002 *

1.0500	-0.0750 + 0.3528i	-0.0750 + 0.1685i	-0.0750 + 0.1032i	-0.0750 + 0.0675i
-0.0750 + 0.0433i	-0.0750 + 0.0244i	-0.0750 + 0.0079i	-0.0750 - 0.0079i	-0.0750 - 0.0244i
-0.0750 - 0.0433i	-0.0750 - 0.0675i	-0.0750 - 0.1032i	-0.0750 - 0.1685i	-0.0750 - 0.3528i

}

x(n)= {

x(0)	**x(3)**	**x(6)**	**x(9)**	**x(12)**
x(5)	**x(8)**	**x(11)**	**x(14)**	**x(2)**
x(10)	**x(13)**	**x(1)**	**x(4)**	**x(7)**

}

X(k)= {

X(0)	**X(6)**	**X(12)**	**X(3)**	**X(9)**
X(10)	**X(1)**	**X(7)**	**X(13)**	**X(4)**
X(5)	**X(11)**	**X(2)**	**X(8)**	**X(14)**

}

2D-FFT

X(k)= {

1.0e+002 *

1.0500	-0.0750 + 0.0244i	-0.0750 - 0.1032i	-0.0750 + 0.1032i	-0.0750 - 0.0244i
-0.0750 - 0.0433i	-0.0750 + 0.3528i	-0.0750 + 0.0079i	-0.0750 - 0.1685i	-0.0750 + 0.0675i
-0.0750 + 0.0433i	-0.0750 - 0.0675i	-0.0750 + 0.1685i	-0.0750 - 0.0079i	-0.0750 - 0.3528i

}

本章回顾：

1.基-2 DIT
2.基-2 DIF
3.统一复合数
4.基-4 DIF/DIT
5.分裂基
6.实序列FFT
7.Chirp Z变换

算法原理
时抽频抽
蝶形流图
复乘复加
算法特点
变换卷积

FFT理解：逻辑

途径：

- 利用旋转因子W_N的对称性+周期性
- 将长序列分为短序列

逻辑：

- 合并同类项，可以减少x(n)与旋转因子W_N乘法
- 希望计算量N^2变$N^2/2$，但需要保持短序列仍然符合DFT表达，所以通过奇偶抽来“凑”

FFT理解：因果

- 算法：DIT, DIF 时抽、频抽

通过将x(n)按时间轴分为r段短序列，然后按"r点蝶形合成的"短序列进行DFT，那么X(k)必然要随着合成的短序列来"奇偶抽取"保持"表达式"改变序号，结果导致了X(k)逆序输出。

- 分 break　　分则乱　　---序号
- 抽 decimatiom　　奇偶抽　　---序号
- 合 combine　　前后合　　---(0~N/2-1)（N/2~N-1)
- 蝶 butterfly　　旋转乘　　---(twiddle factors)

FFT理解：艺术

- 算法：DIT, DIF，复合数，分裂基

- 分、抽、合、蝶 → 算法推导，破蛹(茧)成蝶

- 蛹→DFT：信息不损失
- 蝶→FFT：我想飞得更快