

# デジタル信号処理 (IX)

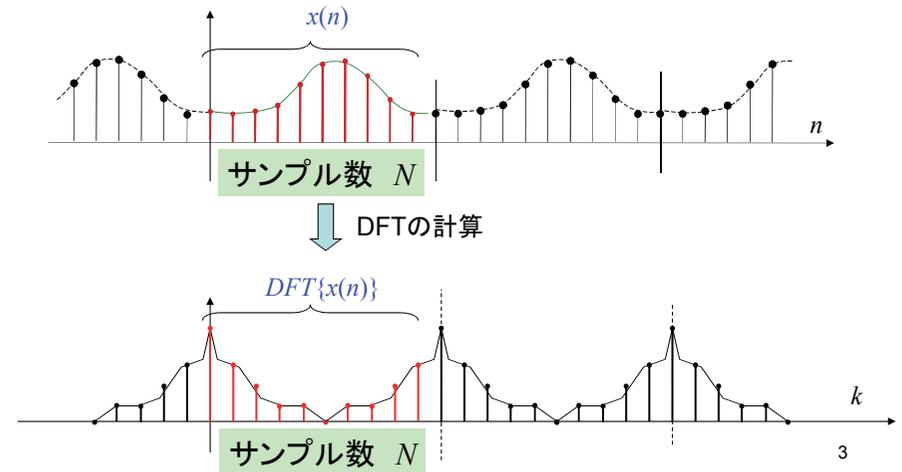
学術国際情報センター  
山口雅浩

E-mail: [yamaguchi.m.aa@m.titech.ac.jp](mailto:yamaguchi.m.aa@m.titech.ac.jp)

Web: <http://guchi.gsic.titech.ac.jp>

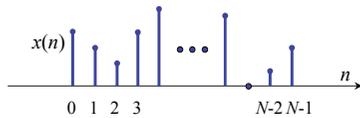
DFTは、離散時間領域で周期関数を仮定している  
→ 周波数領域で離散的・周期的となる

周期はどちらも  $N$



## 離散フーリエ変換

- 離散時間有限区間信号  $x(n)$ :  $0 \leq n < N$  に対して



DFT

$$X(k) = \mathbf{DFT}\{x(n)\} = \sum_{n=0}^{N-1} x(n) \cdot e^{-j2\pi nk/N}$$

Inverse DFT

$$x(n) = \mathbf{DFT}^{-1}\{X(k)\} = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \cdot e^{j2\pi nk/N}$$

## 連続と離散のフーリエ変換

	時間領域	周波数領域
連続フーリエ変換	連続	連続
離散時間フーリエ変換	離散	連続 周期関数
離散フーリエ変換	離散 周期関数	離散 周期関数

$W_N = e^{-j2\pi/N}$  を用いて表現すると

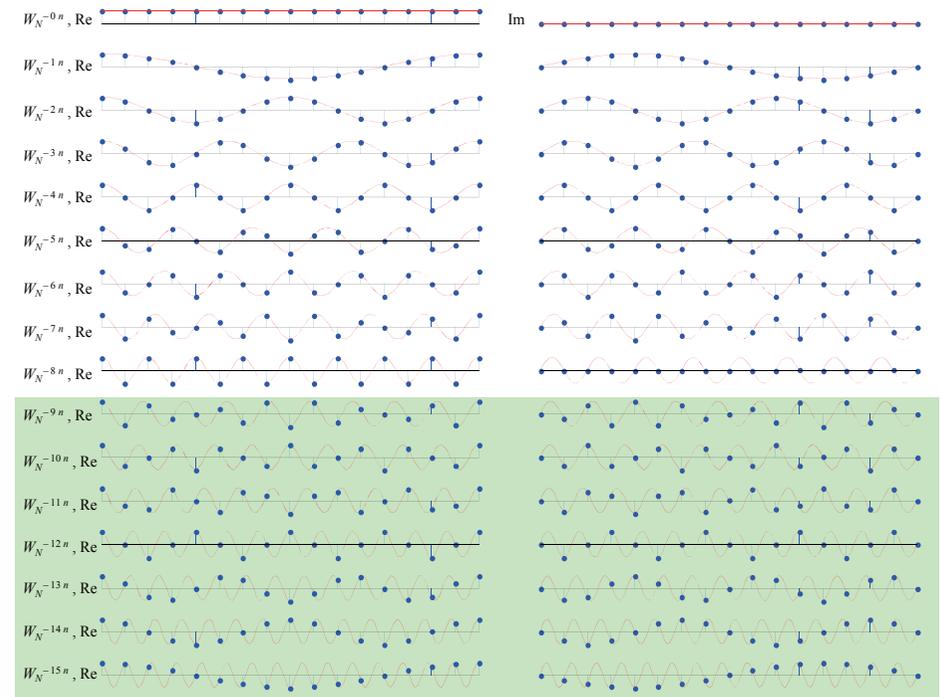
DFT

$$X(k) = \text{DFT}\{x(n)\} = \sum_{n=0}^{N-1} x(n) \cdot e^{-j2\pi nk/N} = \sum_{n=0}^{N-1} x(n) \cdot W_N^{nk}$$

Inverse DFT

$$x(n) = \text{DFT}^{-1}\{X(k)\} = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \cdot e^{j2\pi nk/N} = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \cdot W_N^{-nk}$$

※ N点DFT



$k = 0 \dots 15$  全ての  $W_N^{-nk}$  を見てみよう

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \cdot W_N^{-nk}$$

$x(n)$  は  $W_N^{-nk}$  に  $(1/N)X(k)$  の重みを付けて足し合わせたもの。

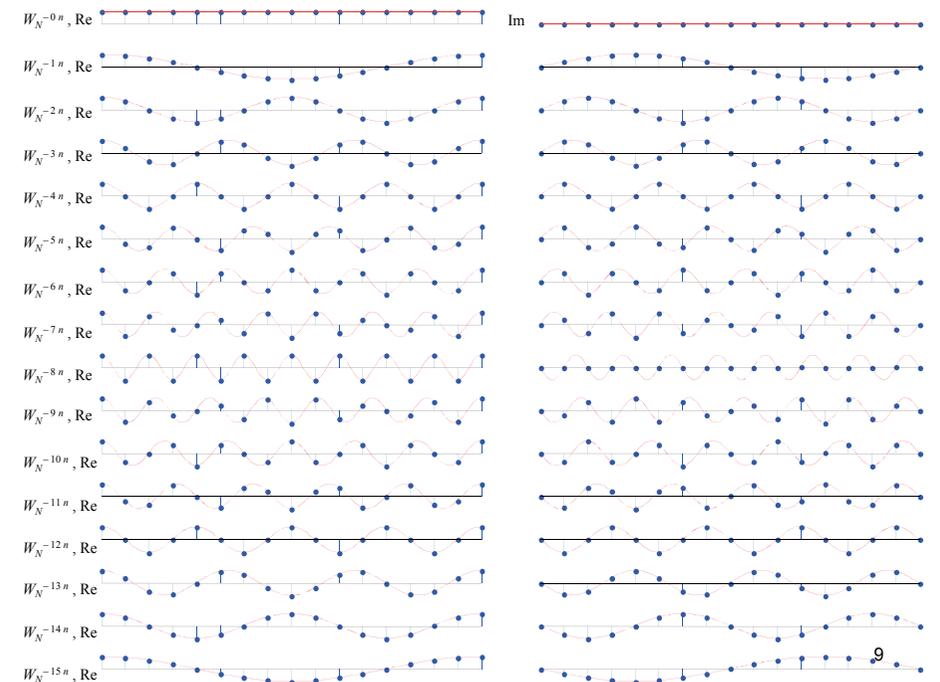
$(1/N)X(k)$  は、 $x(n)$  に  $W_N^{-nk}$  の成分がどれだけ含まれているか、を表している。

「基底関数」、「基底ベクトル」とも呼ばれる。

```

x = [0:16];
lx = [0:160];
x10 = [0:10:160];

for n = 0:15
    fc = cos(2*pi*x*n/16);
    fs = sin(2*pi*x*n/16);
    lc = cos(2*pi*x10*n/160);
    ls = sin(2*pi*x10*n/160);
    subplot(16, 2, 2*n+1);
    stem(x10, fc, 'r');
    hold on;
    plot(lx, lc, 'r');
    hold off;
    axis off;
    axis([0 160 -1 1]);
    subplot(16, 2, 2*n+2);
    stem(x10, fs, 'r');
    hold on;
    plot(lx, ls, 'r');
    hold off;
    axis off;
    axis([0 160 -1 1]);
end
    
```

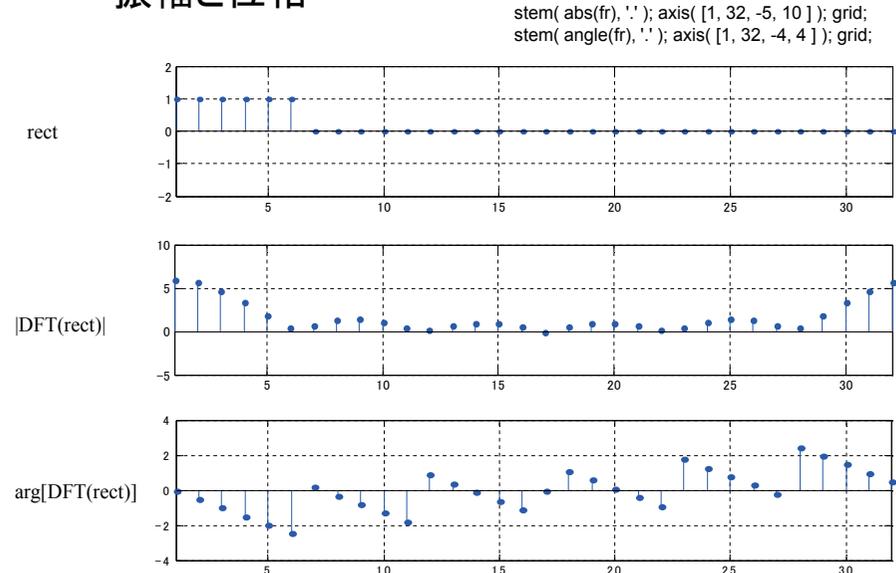


# 離散フーリエ変換・逆変換の行列表記

$$\begin{pmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \end{pmatrix} = \begin{pmatrix} e^0 & e^0 & e^0 & e^0 \\ e^0 & e^{-j\frac{2\pi}{4}} & e^{-j\frac{4\pi}{4}} & e^{-j\frac{6\pi}{4}} \\ e^0 & e^{-j\frac{4\pi}{4}} & e^{-j\frac{8\pi}{4}} & e^{-j\frac{12\pi}{4}} \\ e^0 & e^{-j\frac{6\pi}{4}} & e^{-j\frac{12\pi}{4}} & e^{-j\frac{18\pi}{4}} \end{pmatrix} \begin{pmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & W_4^1 & W_4^2 & W_4^3 \\ 1 & W_4^2 & W_4^4 & W_4^6 \\ 1 & W_4^3 & W_4^6 & W_4^9 \end{pmatrix} \begin{pmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{pmatrix}$$

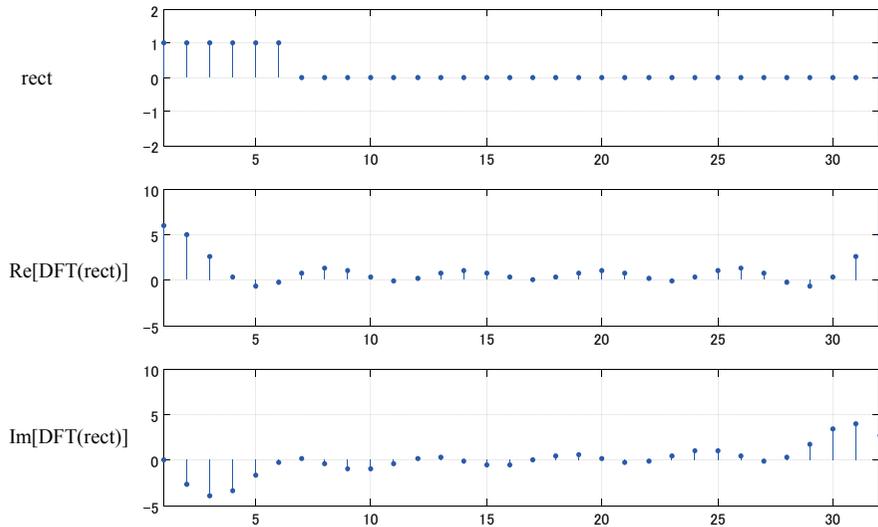
$$\begin{pmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{pmatrix} = \frac{1}{4} \begin{pmatrix} e^0 & e^0 & e^0 & e^0 \\ e^0 & e^{j\frac{2\pi}{4}} & e^{j\frac{4\pi}{4}} & e^{j\frac{6\pi}{4}} \\ e^0 & e^{j\frac{4\pi}{4}} & e^{j\frac{8\pi}{4}} & e^{j\frac{12\pi}{4}} \\ e^0 & e^{j\frac{6\pi}{4}} & e^{j\frac{12\pi}{4}} & e^{j\frac{18\pi}{4}} \end{pmatrix} \begin{pmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \end{pmatrix} = \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & W_4^{-1} & W_4^{-2} & W_4^{-3} \\ 1 & W_4^{-2} & W_4^{-4} & W_4^{-6} \\ 1 & W_4^{-3} & W_4^{-6} & W_4^{-9} \end{pmatrix} \begin{pmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \end{pmatrix}$$

# 例: 矩形パルスのDFT 振幅と位相



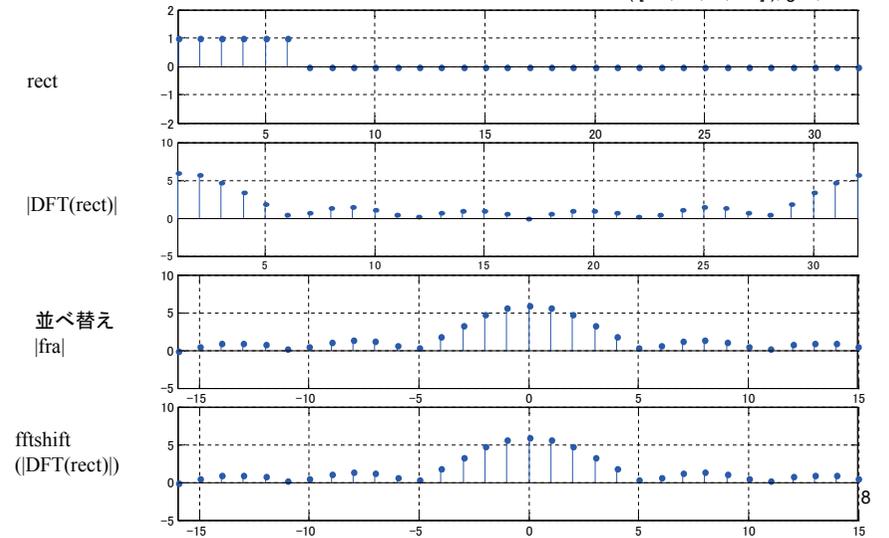
# 例: 矩形パルスのDFT

```
rect([1:32]) = 0;
rect([1:6]) = 1;
stem( rect, '' ); axis( [1, 32, -2, 2 ] ); grid;
fr = fft(rect);
stem( real(fr), '' ); axis( [1, 32, -5, 10 ] ); grid;
stem( imag(fr), '' ); axis( [1, 32, -5, 10 ] ); grid;
```



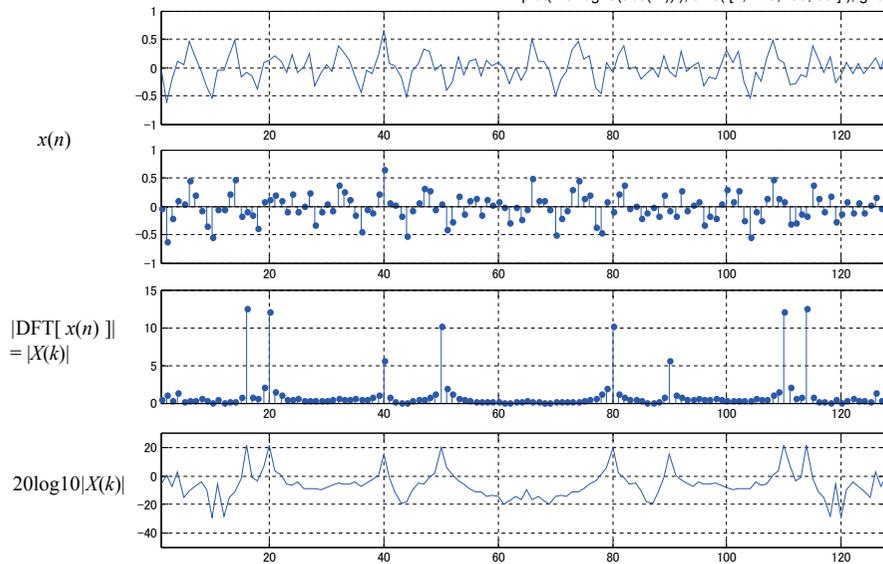
# 例: 矩形パルスのDFT 周波数成分の並べ替え

```
stem( rect, '' ); axis( [1, 32, -2, 2 ] ); grid;
fra = [fr([17:32]),fr([1:16])];
x32 = [-16:15];
stem( x32, abs(fftshift(fr)), '' );
axis( [-16, 15, -5, 10 ] ); grid;
stem( x32, abs(fra), '' );
axis( [-16, 15, -5, 10 ] ); grid;
```



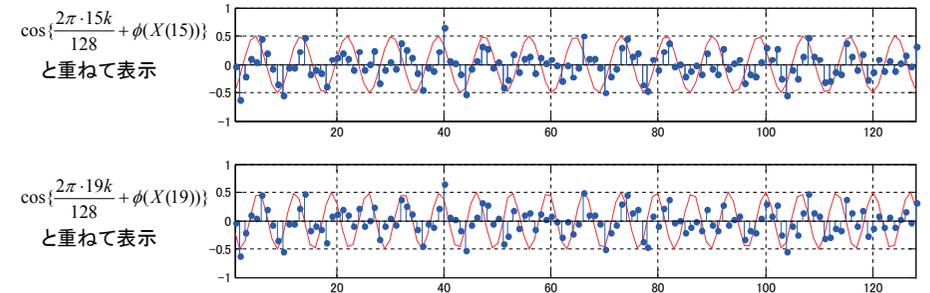
## 例2: 音声信号のDFT

```
ring = wavread('ring.wav');
ri = ring(1:128);
plot(ri); axis([1, 128, -1, 1]); grid;
stem(ri, 'b'); axis([1, 128, -1, 1]); grid;
fri = fft(ri);
stem(abs(fri), 'b'); axis([1, 128, 0, 15]); grid;
plot(20*log10(abs(fri))); axis([1, 128, -50, 30]); grid;
```



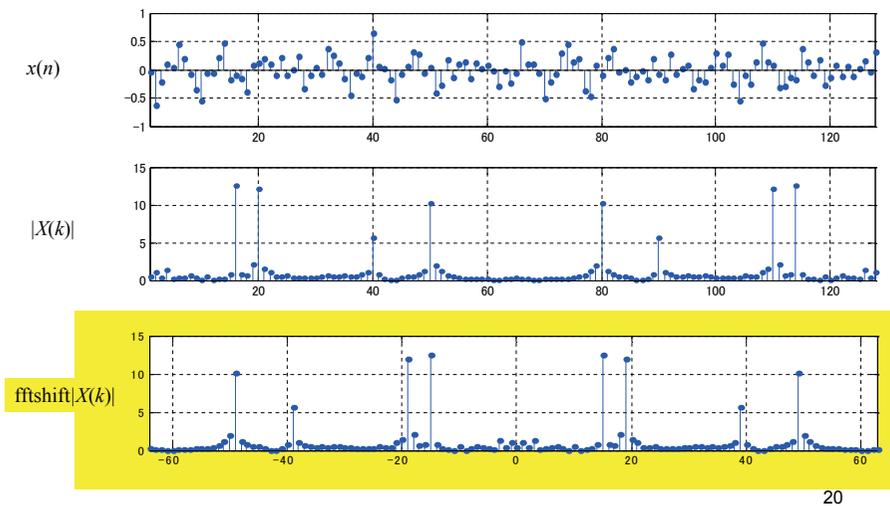
## 例2: 音声信号のDFT 強い周波数成分

```
>> angle(fri(16))
ans =
    2.7592
>> c16a = cos(2 * pi * 15 / 128 * [1:128] + 2.7592) * 0.5;
>> plot(c16a, 'r');
>> hold on;
>> stem(ri, 'b'); axis([1, 128, -1, 1]); grid;
>> hold off;
>> angle(fri(20))
ans =
    1.1151
>> c20a = cos(2 * pi * 19 / 128 * [1:128] + 1.1151) * 0.5;
>> plot(c20a, 'r');
>> hold on;
>> stem(ri, 'b'); axis([1, 128, -1, 1]); grid;
```



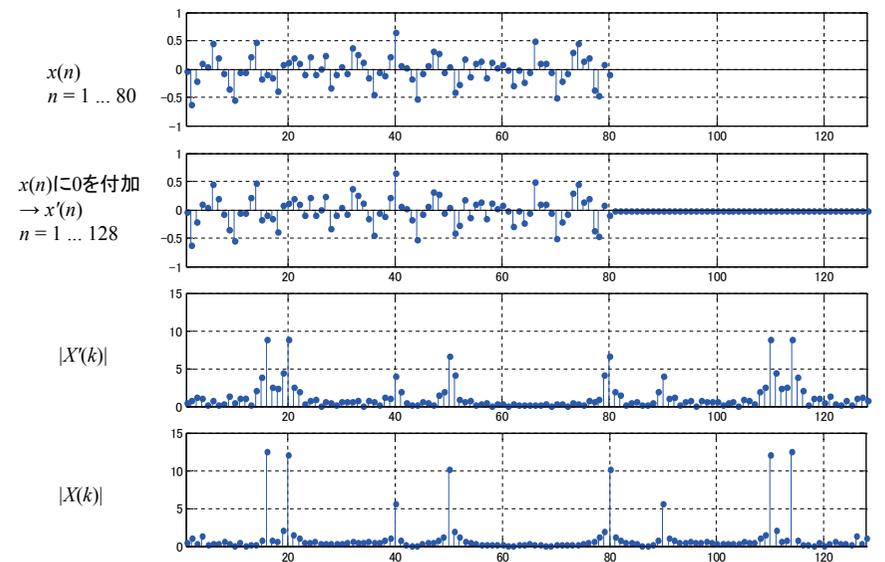
## 例2: 音声信号のDFT 周波数成分の並べ替え

```
x = [-64:63];
stem(x, fftshift(abs(fri)), 'b');
axis([-64, 63, 0, 15]); grid;
```



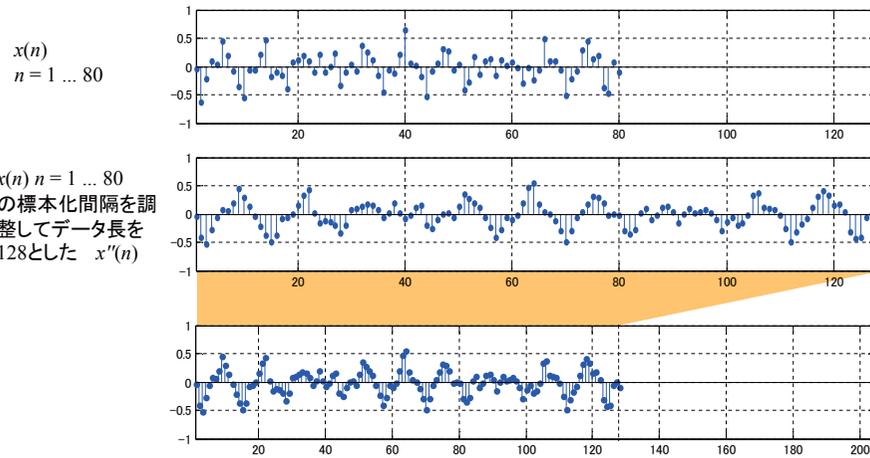
## 例2: 音声信号のDFT 短いデータに0を挿入

```
ri80 = ri(1:80);
stem(ri80, 'b'); axis([1, 128, -1, 1]); grid;
ri80([81:128]) = 0;
stem(ri80, 'b'); axis([1, 128, -1, 1]); grid;
fri80 = fft(ri80);
stem(abs(fri80), 'b'); axis([1, 128, 0, 15]); grid;
```



## 例2: 音声信号のDFT 標本化周期を調整

```
xi = [0:127] * 79 / 127 + 1;
ipr = interp1(ri80([1:80]), xi);
stem( ipr, '.' ); axis( [1, 128, -1, 1] ); grid;
axis( [1, 80, 0, 15] );
```



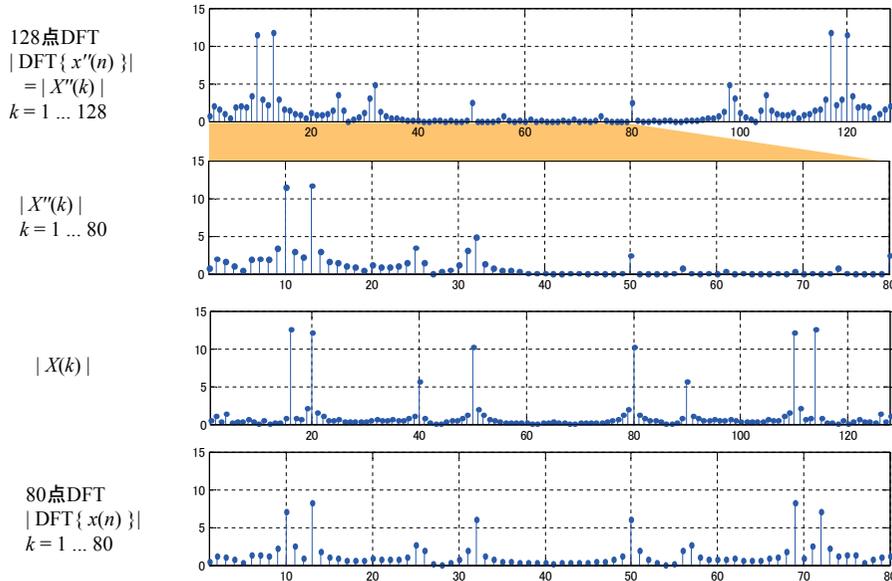
$x(n)$   
 $n = 1 \dots 80$

$x(n)$   $n = 1 \dots 80$   
の標本化間隔を調整してデータ長を128とした  $x''(n)$

23

## 例2: 音声信号のDFT 標本化周期を調整

```
fipr = fft( ipr );
stem( abs(fipr), '.' ); axis( [1, 128, 0, 15] ); grid;
axis( [1, 80, 0, 15] );
fri80 = fft( ri80([1:80]) );
stem( abs(fri80), '.' ); axis( [1, 80, 0, 15] ); grid;
```



128点DFT  
 $| \text{DFT} \{ x''(n) \} |$   
 $= | X''(k) |$   
 $k = 1 \dots 128$

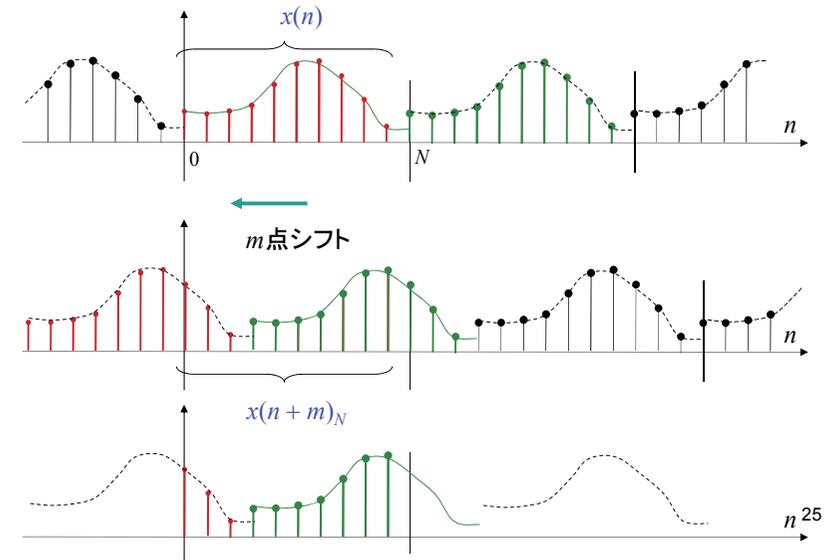
$| X''(k) |$   
 $k = 1 \dots 80$

$| X(k) |$

80点DFT  
 $| \text{DFT} \{ x(n) \} |$   
 $k = 1 \dots 80$

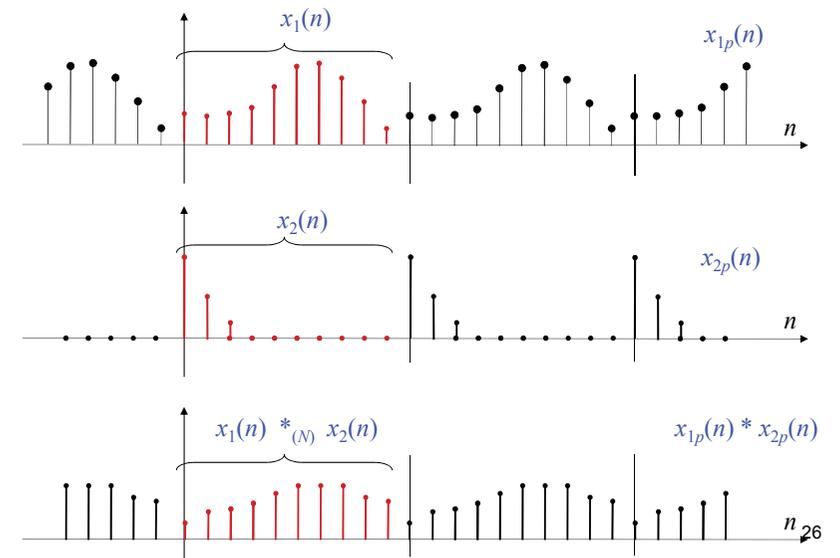
## 循環推移

循環推移定理  $\text{DFT} \{ x(n+m)_N \} = W_N^{-km} \text{DFT} \{ x(n) \}$



$n^{25}$

## 循環畳込み



$n^{26}$

## 循環畳込み定理

- 系列長 $N$ の信号  $x_1(n)$  と  $x_2(n)$  の循環畳込みを以下の式で定義する

$$y(n) = x_1(n) *_{(N)} x_2(n) = \left\{ \sum_{l=0}^{N-1} x_{1p}(l) x_{2p}(n-l) \right\} P_N(n)$$

ただし

$$P_N(n) = \begin{cases} 1 & 0 \leq n < N \\ 0 & \text{otherwise} \end{cases}$$

- このとき

$$DFT\{y(n)\} = DFT\{x_1(n) *_{(N)} x_2(n)\} = DFT\{x_1(n)\} \cdot DFT\{x_2(n)\}$$

→ 2つの離散信号の循環畳込みのDFTは、それぞれのDFTの積に一致する

27

## まとめ: 離散フーリエ変換と連続信号

- 連続時間領域でのサンプリング間隔  $T$
  - サンプル数  $N$
  - 連続系の周波数領域でのサンプリング間隔  $\omega_p$
- 
- 連続時間領域で周期  $T_p$  の周期関数として扱われる
  - $T_p = NT$
  - $f(n) = f(nT)$   
 $F(k) = F(k\omega_p)$
  - $\omega_p = 2\pi / T_p$
  - 周波数のサンプリング間隔  $f_p = 1/T_p = 1 / (NT)$
  - 最高周波数  $N/2$  は  
 $(N/2)(1/T_p) = 1/(2T)$  Hz cf. サンプリング定理

29

## 循環畳込み定理(周波数領域)

- 系列長 $N$ の信号  $x_1(n)$  と  $x_2(n)$  の積のDFTは、

$$Y(k) = DFT\{x_1(n) \cdot x_2(n)\} = \frac{1}{N} \left\{ \sum_{m=0}^{N-1} X_{1p}(m) X_{2p}(k-m) \right\} P_N(k)$$

ただし  $X_{qp}(k)$  は  $X_q(k)$  を周期 $D$ で並べて周期関数としたもの ( $q = 1, 2$ ) であり、

$$X_q(k) = DFT\{x_q(n)\}$$

→ 2つの離散信号の積のDFTは、それぞれのDFTの循環畳込みに一致する

28