ディジタル信号処理 (X)

学術国際情報センター
山口雅浩

高速フーリエ変換 (FFT)

- N点DFTの計算

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

  - N^2回の複素乗算
  - $N(N\text{-}1)$回の複素加算

- 高速フーリエ変換 (Fast Fourier Transform: FFT)
  - DFTの性質を利用して乗算回数を削減
    - Cooley and Tukey のアルゴリズム
  - $N=2^p$のときが多く用いられる
  - → 乗算回数は $(N/2)(\log_2 N - 1)$

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

N^2

n を偶数と奇数に分ける
$l = 0, 1, \dots N/2-1$ を用いて

$$X(k) = \sum_{l=0}^{N/2-1} x(2l) \cdot W_N^{2lk} + \sum_{l=0}^{N/2-1} x(2l+1) \cdot W_N^{(2l+1)k}$$

$$W_N^{2lk} = e^{-j\frac{2\pi}{N}(2lk)} = e^{-j\frac{2\pi}{N/2}lk} = W_{N/2}^{lk}$$ と変形できるので、

$$X(k) = \underbrace{\sum_{l=0}^{N/2-1} x(2l) \cdot W_{N/2}^{lk}}_{G(k)} + W_N^k \underbrace{\sum_{l=0}^{N/2-1} x(2l+1) \cdot W_{N/2}^{lk}}_{H(k)}$$

$(N/2)$点DFTの式と同じだが、$k = 0, 1, \dots N-1$

$$G(k) = \sum_{l=0}^{N/2-1} x(2l) \cdot W_{N/2}^{lk}$$

$$H(k) = \sum_{l=0}^{N/2-1} x(2l+1) \cdot W_{N/2}^{lk}$$

を $k = 0, 1, \dots N-1$ について計算する必要がある

$\frac{N}{2} \times N \times 2$

しかし、$N/2 \le k \le N-1$ では、$k' = k - N/2$ として

$$W_{N/2}^{lk} = W_{N/2}^{l(k'+N/2)} = W_{N/2}^{lk'+(N/2)l} = W_{N/2}^{lk'}$$

が成り立つことから、$k = N/2, N/2+1, \dots N-1$ に対しては

$$G(k) = G(k - N/2)$$
$$H(k) = H(k - N/2)$$

であり、$G(k), H(k)$ の計算は、$k = 0, 1, \dots N/2-1$ だけ行えばよい

$(N/2)$点DFT × 2

さらに、同様に、$N/2 \le k \le N-1$ では　$k' = k - N/2$ として

$$W_N^k = W_N^{k'+N/2} = W_N^{k'} W_N^{N/2} = W_N^{k'} \cdot e^{-j\frac{2\pi}{N}\cdot\frac{N}{2}} = W_N^{k'} \cdot e^{-j\pi} = -W_N^{k'}$$

であることから、　W_N^k の計算も $k = 0, 1, \ldots N/2-1$ まででよく、

以下のようにして計算できる

$k = 0, 1, \ldots N/2-1$

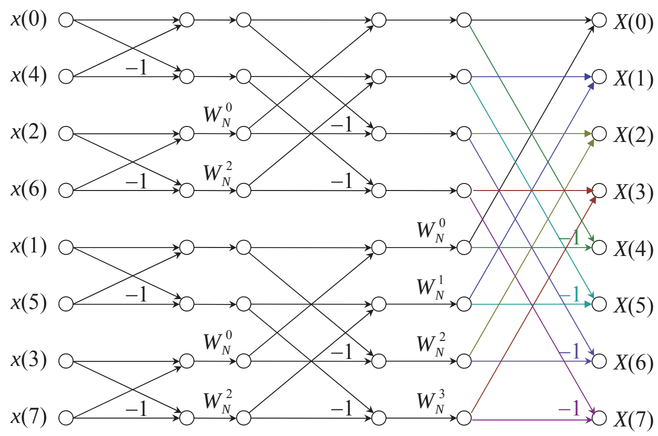
$$X(k) = G(k) + W_N^k H(k)$$

$k = N/2, N/2+1, \ldots N-1$
$(k' = 0, 1, \ldots N/2-1)$

$$X(k) = G(k') - W_N^{k'} H(k')$$

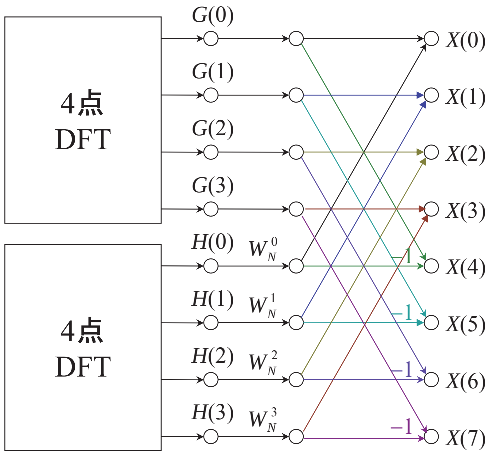
$$\left(\frac{N}{2}\right)^2 \times 2 + \frac{N}{2}$$

シグナルフローグラフ



→　バタフライ演算

シグナルフローグラフ



計算量について（乗算回数）

2点DFTの計算

$$x(l) + x(l + \frac{N}{2})$$

$$x(l) - x(l + \frac{N}{2})$$

・・・乗算回数は0

各ステップにおいて

$W_N^0 \ldots W_N^{N/2}$

の複素乗算　→　$(N/2)$

（ただしN=2のときは不要）

$\log_2 N$ ステップ　➡　$\frac{N}{2}(\log_2 N - 1)$ or　$O\{N \log_2 N\}$

留意点

- FFT, DFTのプログラムでは、あらかじめ cos, sin は計算して
  テーブルとして持っておく
  （cos, sinの計算には時間が掛かるため）
- IFFTは

  X(k) → [FFT] → [順番を反転] → [1/N]

  により計算できる。
- 実数であることがわかっている信号のFFTでは、$X(0)$ ... $X(N/2)$ まで計算し、$X(N/2+1)$ ... $X(N-1)$ を$X(N-k+1)$の複素共役として求めてもよい。

  ※ただし一般に$X(k)$は複素数なのでIFFTには使えない
- FFTのプログラムは書籍・webなどから比較的容易に入手可能。
  （FFTは信号処理以外でも幅広く用いられている）

さらに深く習いたい人は

- ここで紹介したFFTアルゴリズムは「時間間引き形」と呼ばれるもの。他に「周波数間引き形」のアルゴリズムもある。
  ref. 「ディジタル信号処理の基礎」、辻井重男監修、電子情報通信学会編など
- Dが2^nでは無い場合のアルゴリズムもある。
- 時間信号の周波数解析を行う場合、DFTは有限長の時間信号を周期関数として扱っていることに留意する必要がある。端部の影響を避けるために、「窓関数」がよく用いられる。有名なものにハミング窓、ハニング窓などがある。
- 画像などの2次元信号に合わせたFFTもある。計算時間がそれほど問題にならない場合には、1次元FFTを各行に適用し、その結果の各列に対して再度1次元FFTを適用することで2次元信号のDFTを行うことができる。
- DFTは離散系列に対する直交変換の一種である。DFT以外の直交変換も多数ある。離散コサイン変換（DCT）はJPEGなどの画像圧縮に用いられる。他に、アダマール変換、ハール変換、KL変換などがある。