信号処理論第一：第6回
窓関数・高速Fourier変換

東京大学 大学院情報理工学系研究科 システム情報学専攻
小山 翔一

講義の目的と概要

- 講義の目的
  - 信号処理の基礎を習得する。

- 講義の概要
  - 決定論的信号（確定的信号）に関する信号処理について，基礎事項を理解する。本講義では，できるだけ特定の応用分野に限らない，一般的な信号処理論について講義する。ただし，統計的信号処理については扱わず，信号処理論第二の範囲となる。

- キーワード
  - Fourier級数, Fourier変換, 離散時間Fourier変換, 離散Fourier変換, FFT, 窓Fourier変換, 線形時不変システム, サンプリング定理, インパルス応答, 伝達関数, 周波数応答, 差分方程式, ブロック線図, 因果性, 安定性, FIRフィルタ, IIRフィルタ

講義スケジュール

- Sセメスター 金曜１限 (8:30-10:15) @オンライン

- 日程（暫定版）
  - 4/ 9　第１回
  - 4/16　第２回
  - 4/23　第３回
  - 4/ 30　第４回
  - 5/ 7　第５回
  - 5/14　第６回
  - 5/21　第７回

  小山担当

  - 5/28　休講
  - 6/4　第７回
  - 6/11　第８回
  - 6/18　第９回
  - 6/25　第１０回
  - 7/ 2　第１１回
  - 7/ 9　第１２回

  堀崎担当

  - 7/30　学期末試験

オンライン講義の実施方法

- zoom（https://zoom.us/ ）を用いる。設定方法等についてはポータルサイト（ https://utelecon.github.io/oc/ ）を参照すること。
- 講義のURLは前半・後半それぞれで毎回同じにする予定。後半(6/4)からはURLが変更となるので必ずITC-LMSを確認すること。
- 質問がある場合は，チャットに書き込む，あるいはミュートをオフにして音声で質問する，のいずれかの方法で行うこと。
- もしオンラインでの受講に不都合が生じた場合は，メール等で連絡すること。

shoichi_koyama@ipc.i.u-tokyo.ac.jp

horisaki@g.ecc.u-tokyo.ac.jp

講義資料と成績評価

- 講義資料・動画
  - ITC-LMSにアップロード
  - 資料はできるだけ講義前日までにアップロードするようにします。

- 成績評価
  - 学期末試験（暫定）
  - 第7回（5/21）に小テストを実施する可能性あり

本日の目次

1. 窓関数
2. 高速Fourier変換

窓関数

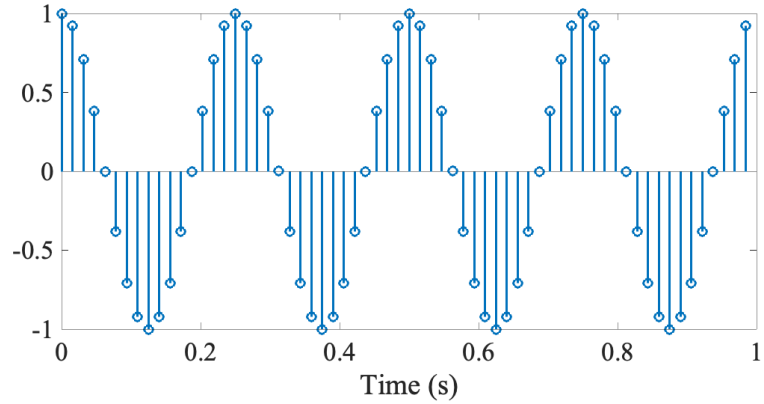
離散Fourier変換の問題

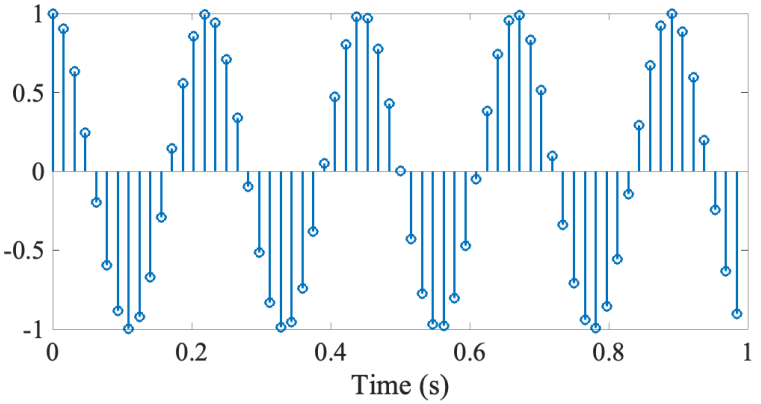
- 信号1: 4 Hz，信号2: 4.5 Hz
- Case1 – サンプリング周波数：64 Hz，信号長：1 s

信号1

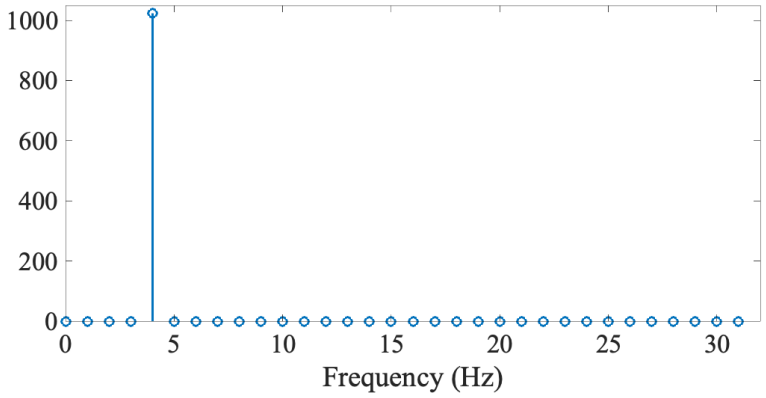
信号2

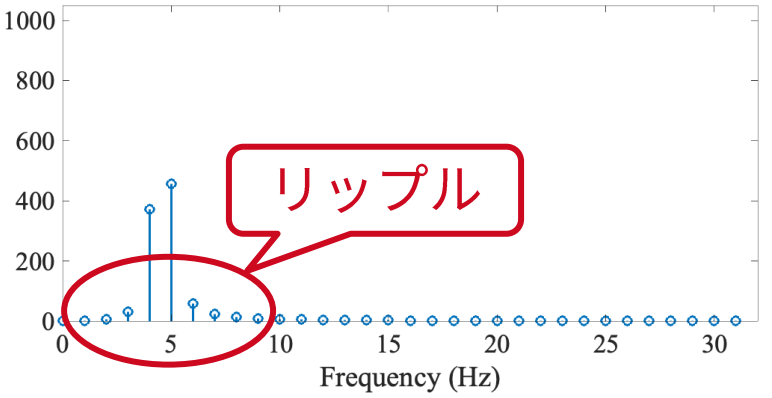
時間信号





パワースペクトル





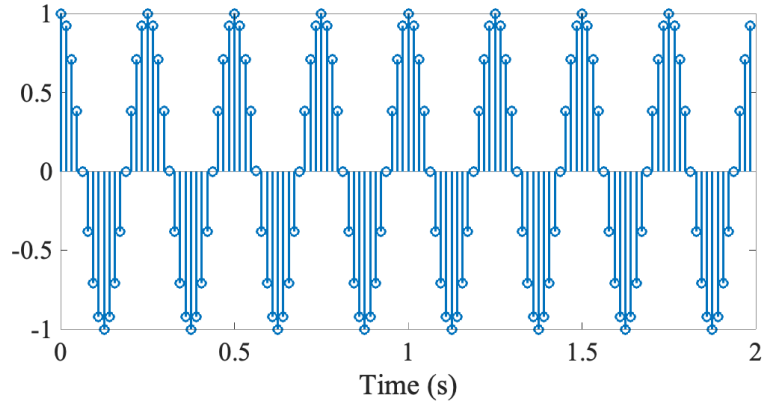
離散Fourier変換の問題

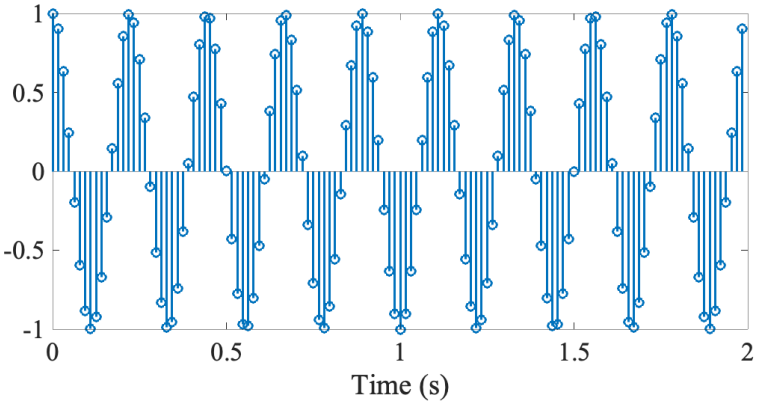
- 信号1: 4 Hz，信号2: 4.5 Hz
- Case2 – サンプリング周波数：64 Hz，信号長：2 s

信号1

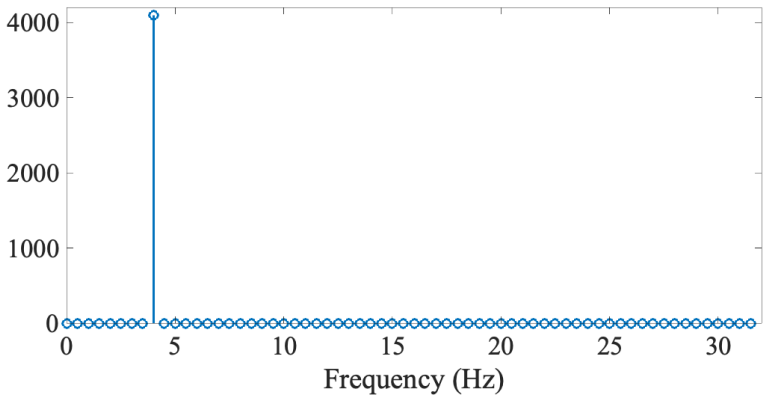
信号2

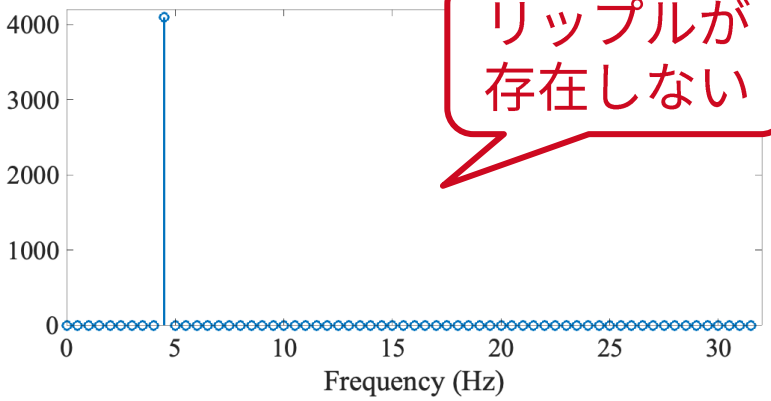
時間信号





パワースペクトル





リップルが
存在しない

離散Fourier変換の問題

- Case1:
  - 信号1：4周期分，信号2：4.5周期分
- Case2:
  - 信号1：8周期分，信号2：9周期分

- 離散Fourier変換では，時間領域，周波数領域で周期拡張されるため，切り出された信号の両端のつながり方によってはリップルの問題が生じる。

窓関数

- 信号長を解析対象に合わせて変更することは通常できないため，両端のつながりの影響を少なくするような窓関数を信号に乗算し，リップルを抑制することがよく行われる。

- 離散窓Fourier変換
  - 離散時間信号 $x[n]$ $(n \in \{0, \ldots, N-1\})$ に対し，信号区間の中心から離れるほど減衰する重み関数（窓関数）$w[n]$ を乗じて離散Fourier変換する。

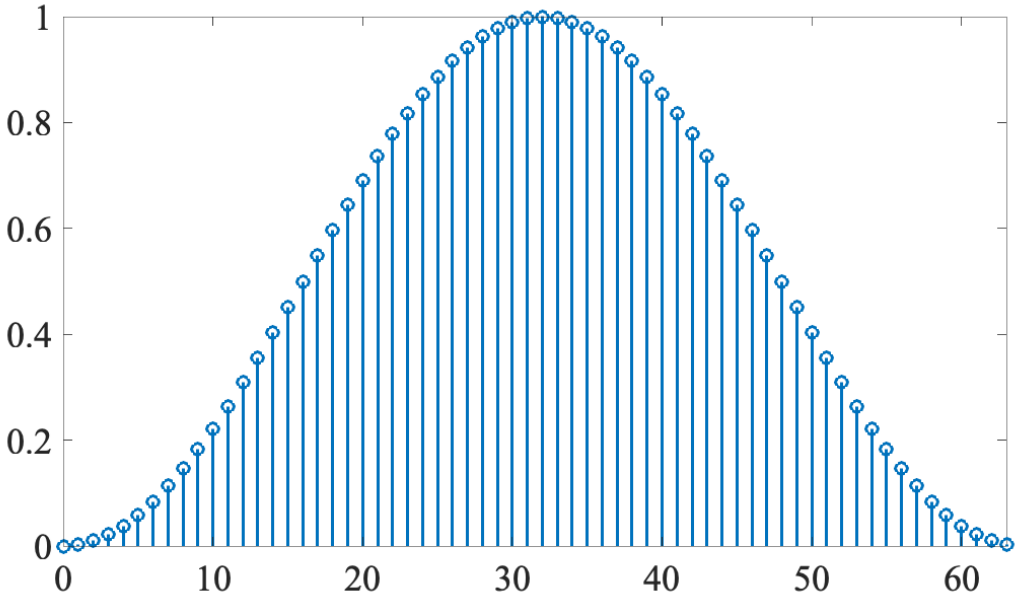
$$X[k] = \sum_{n=0}^{N-1} \underbrace{x[n]w[n]}_{\text{窓関数の乗算}} \exp\left(-\mathrm{j}\frac{2\pi kn}{N}\right)$$

  - 次スライド以降でいくつかの代表的な窓関数を挙げる。

窓関数の例

- Hanning窓

$$w[n] = \begin{cases} 0.5 - 0.5\cos\left(\frac{2\pi n}{N}\right), & 0 \le n \le N \\ 0, & \text{otherwise} \end{cases}$$

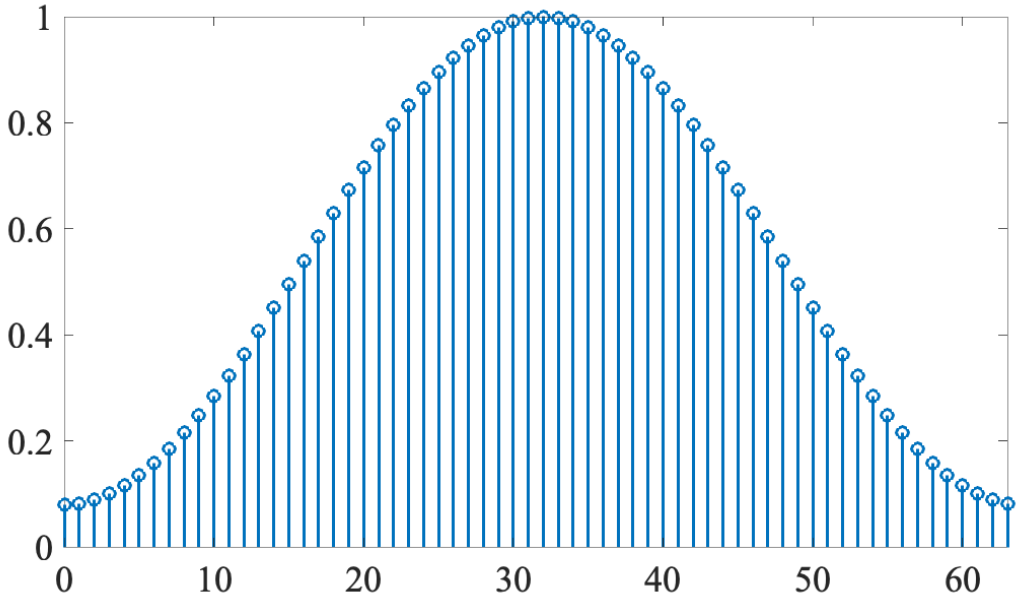


$N = 64$ の場合

窓関数の例

- Hamming窓

$$w[n] = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{N}\right), & 0 \le n \le N \\ 0, & \text{otherwise} \end{cases}$$

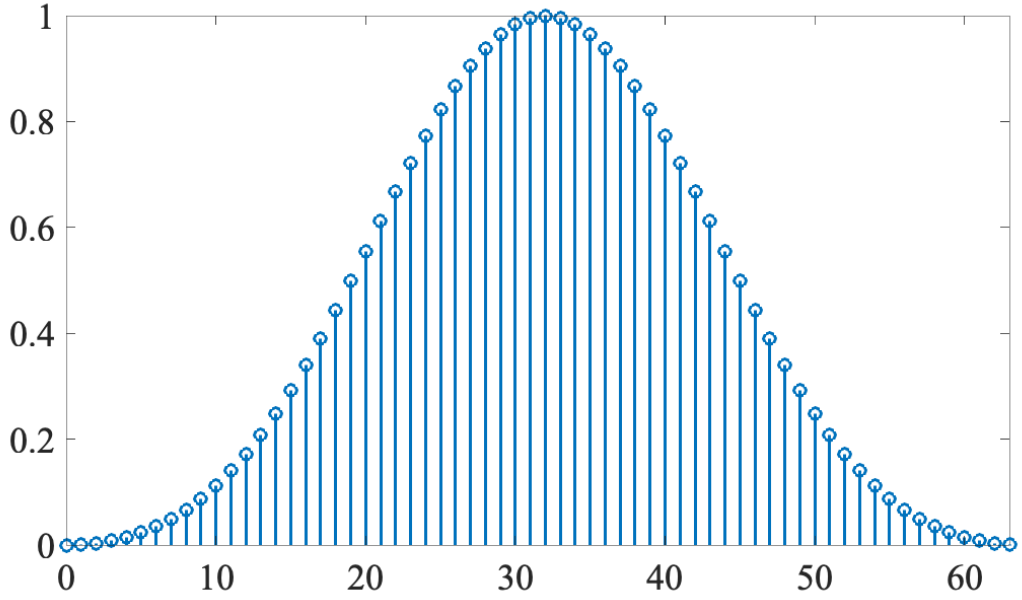


$N = 64$ の場合

窓関数の例

- Blackman窓

$$w[n] = \begin{cases} 0.42 - 0.5\cos\left(\frac{2\pi n}{N}\right) + 0.08\cos\left(\frac{4\pi n}{N}\right), & 0 \leq n \leq N \\ 0, & \text{otherwise} \end{cases}$$

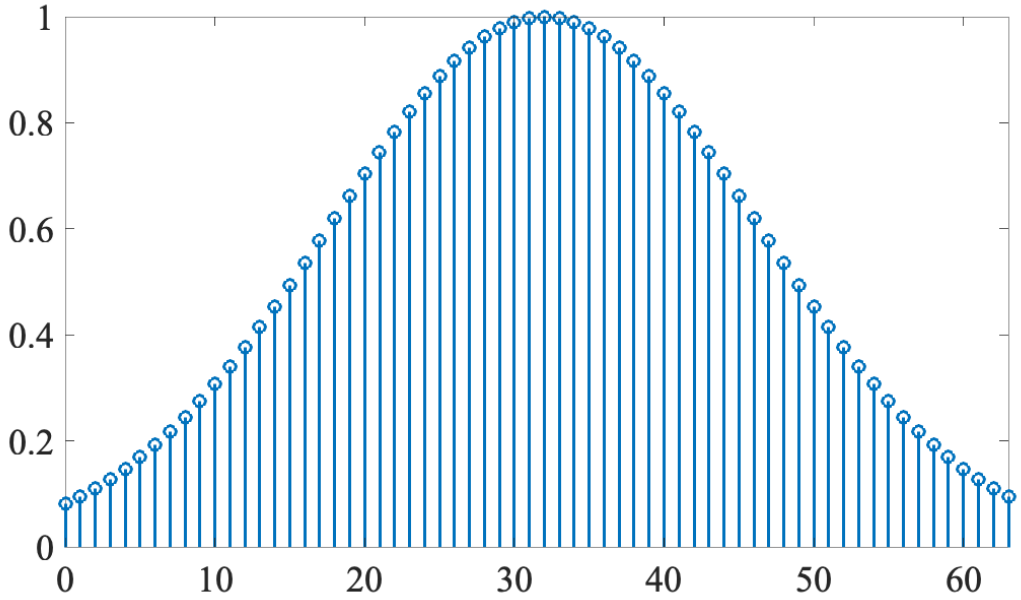


$N = 64$ の場合

窓関数の例

- Gauss窓

$$w[n] = \begin{cases} \exp\left(-\frac{1}{\sigma^2}\frac{(n-\frac{N}{2})^2}{N^2}\right), & 0 \le n \le N \\ 0, & \text{otherwise} \end{cases}$$

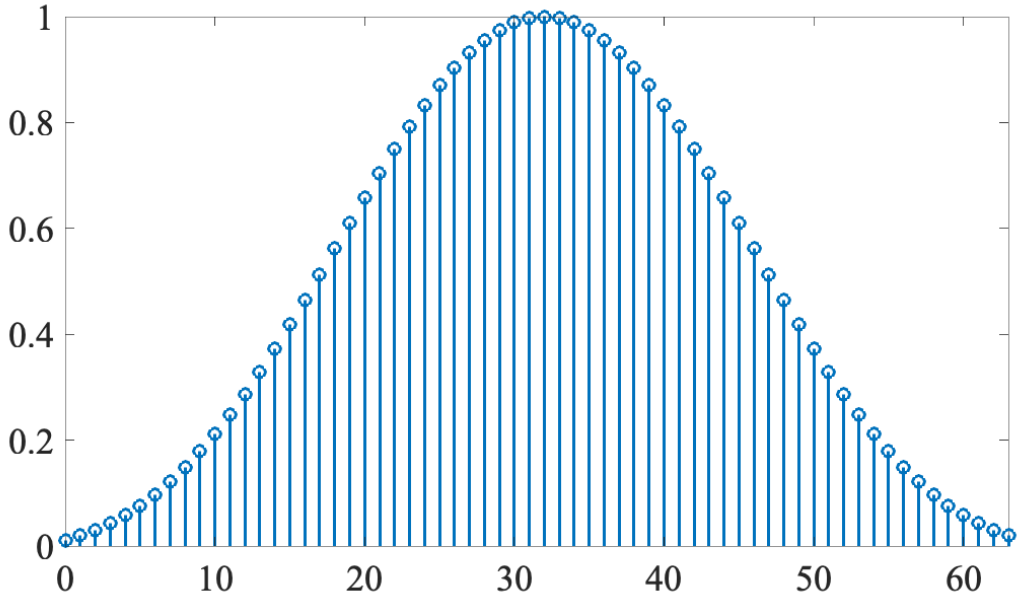


$N = 64,\ \sigma^2 = 0.1$ の場合

窓関数の例

- Kaiser窓

$$w[n] = \begin{cases} \dfrac{I_0\left(\pi\alpha\sqrt{1-\left(\frac{2n}{N}-1\right)^2}\right)}{I_0(\pi\alpha)}, & 0 \le n \le N \\ 0, & \text{otherwise} \end{cases}$$

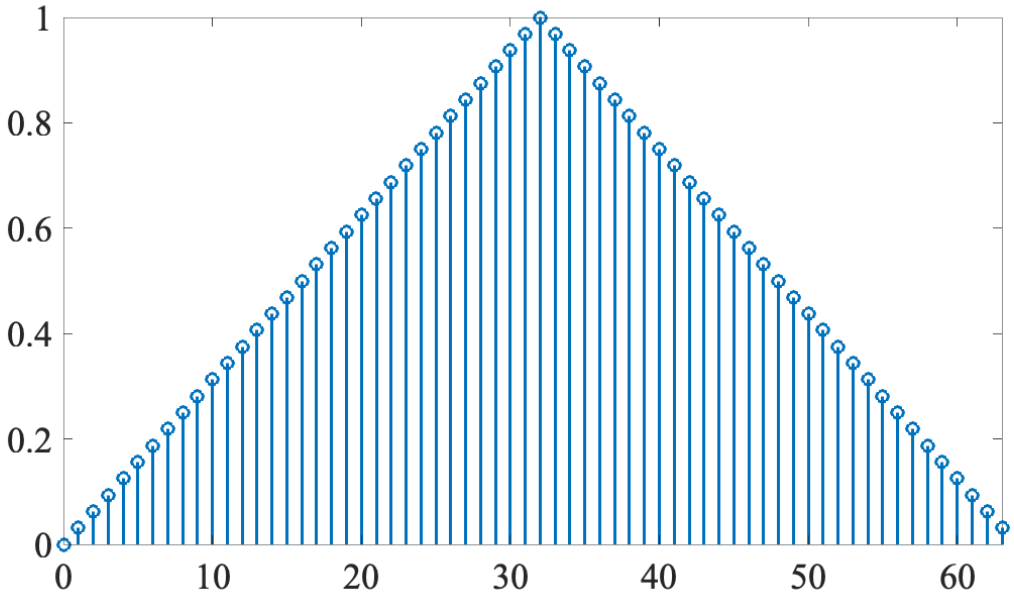


$N = 64,\ \alpha = 2$ の場合

窓関数の例

- 三角窓（Bartlett窓）

$$w[n] = \begin{cases} \frac{2n}{N}, & 0 \le n \le \frac{N}{2} \\ 2 - \frac{2n}{N}, & \frac{N}{2} < n \le N \\ 0, & \text{otherwise} \end{cases}$$

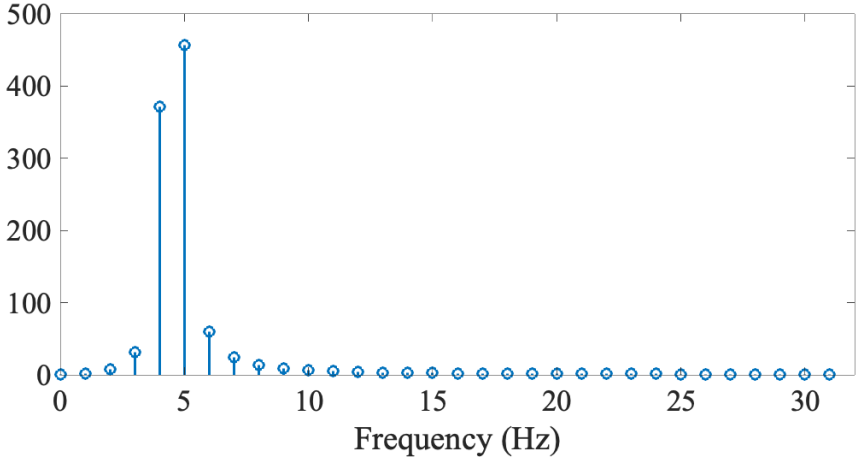


$N = 64$ の場合

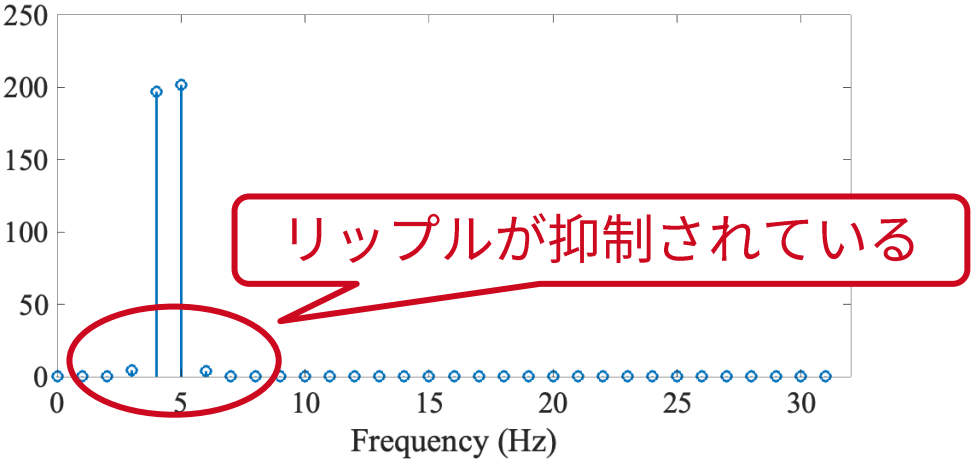
窓関数の例

- 先程のCase1における4.5 Hzの正弦波にHamming窓関数を適用した例。

矩形窓の場合



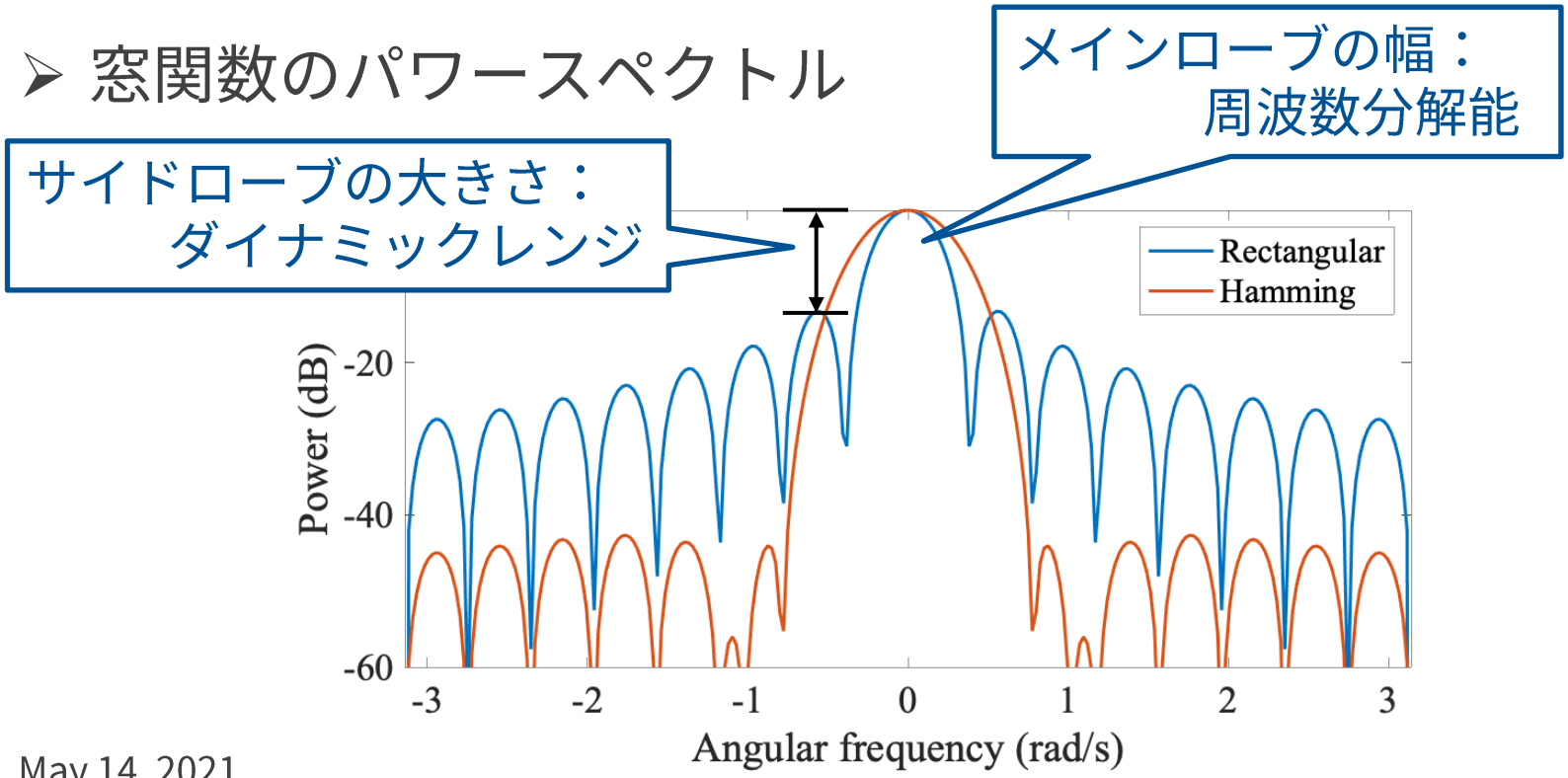
Hamming窓の場合



窓関数の性能

- 窓関数をかけることで元の信号が変わってしまうのでは？
  - 窓関数の性能は周波数分解能とダイナミックレンジによって決まる。
  - これらはトレードオフの関係にあるため，目的に合わせて選択することが必要。

- 窓関数のパワースペクトル

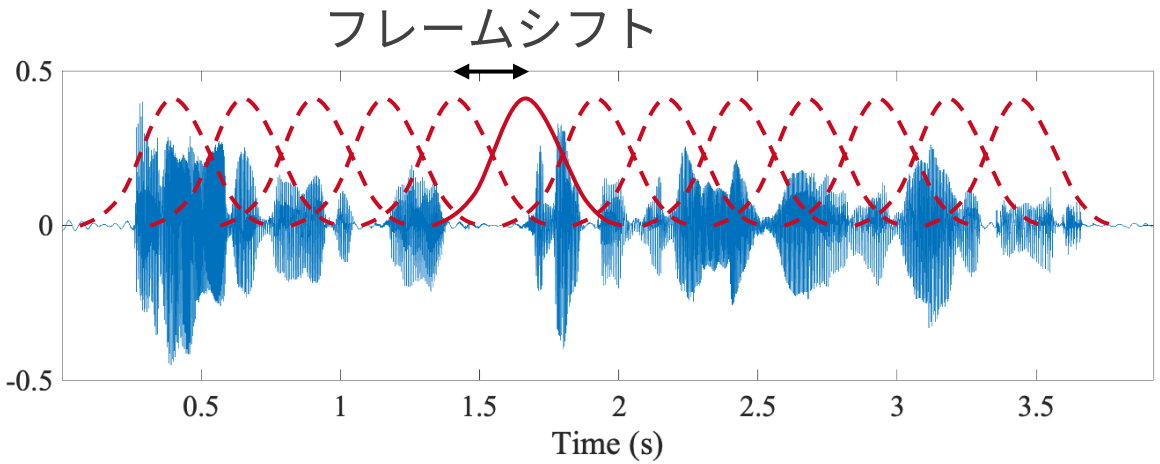


統計的信号処理におけるスペクトル推定

- 実際は無限に長い時系列信号から一部を切り出し，スペクトルを推定する問題。
  - 無限長時系列信号の真のスペクトルと切り出した有限区間の信号から算出されるスペクトルは通常一致しない。
  - 信号を確率的な変動（不規則信号）とみなし，観測される有限区間の信号から潜在的なパラメータであるスペクトルを推定することをスペクトル推定と呼ぶ。
  - スペクトル推定は統計的信号処理における重要なトピックの一つ。

短時間Fourier変換

- 長い時間区間の時系列信号では，スペクトルの時間変化に意味がある場合が多い。（時間区間全体のスペクトルはあまり意味を持たない。）
- 短時間Fourier変換 (short-time Fourier transform: STFT)
  - 短時間の時間区間ごとに窓関数を乗じてFourier変換



- 連続系での数式表現

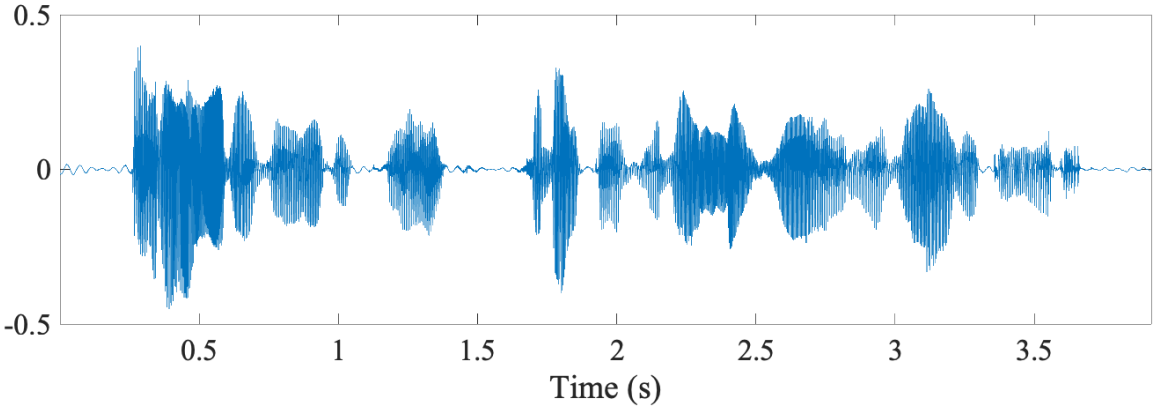
$$X(\omega, t) = \int_{-\infty}^{\infty} x(\tau) w(\tau - t) \exp(-\mathrm{j}\omega\tau) \mathrm{d}\tau$$

時刻tを中心とした窓関数

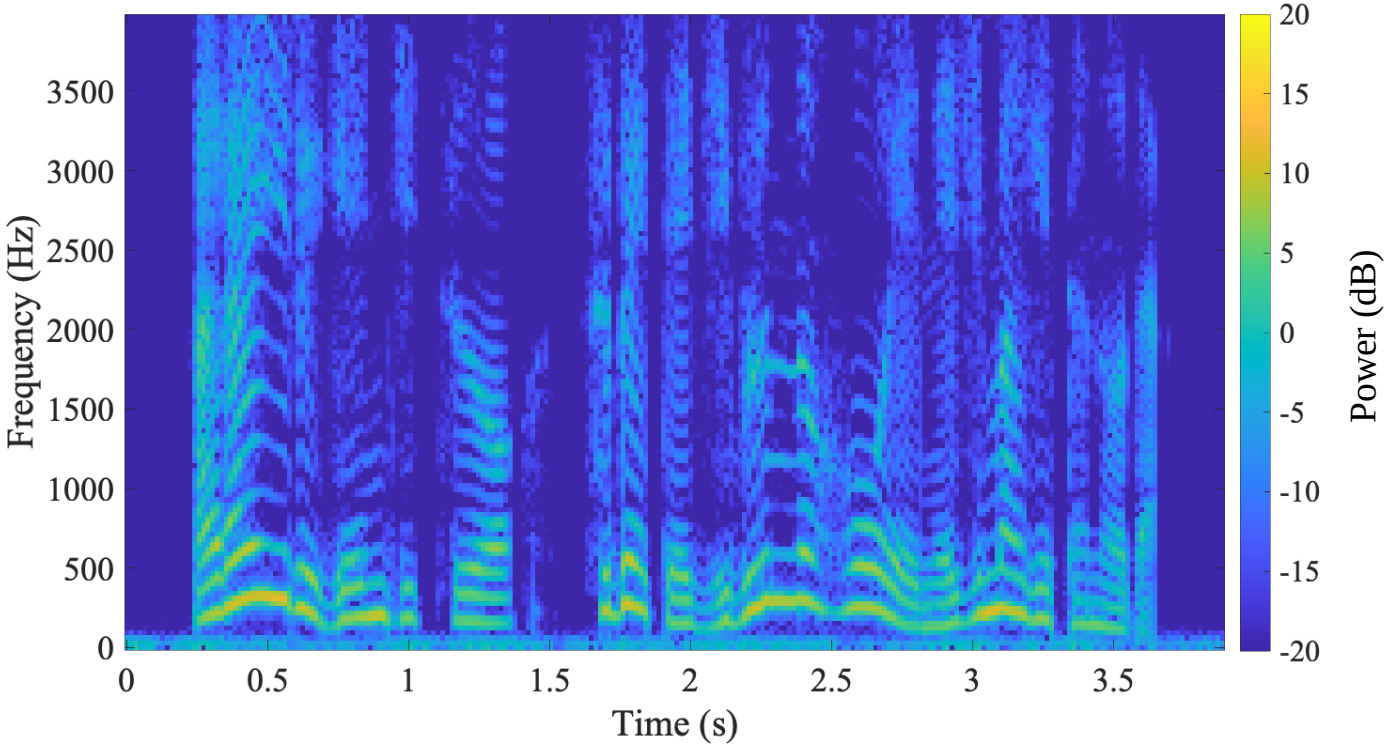
短時間Fourier変換

- スペクトログラム：時間–周波数と信号パワーとの関係

時間信号



スペクトログラム



不確定性原理

- 短時間Fourier変換では，時間分解能と周波数分解能との間にトレードオフの関係がある。
  - 時間分解能を高くするためにフレーム長を短くすると，周波数分解能が低下
  - 周波数分解能を高くするためにフレーム長を長くすると，時間分解能が低下

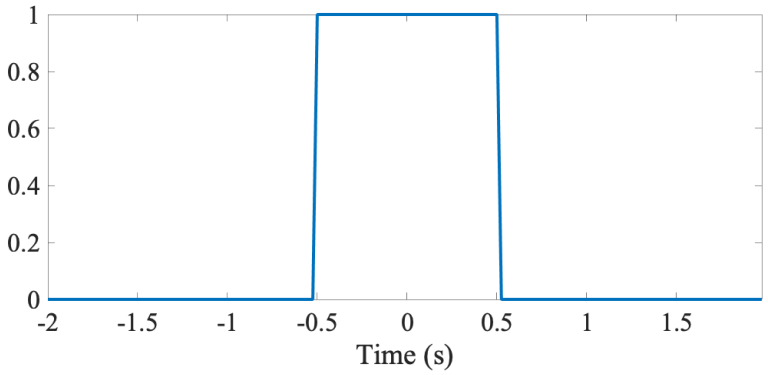
- このような関係を時間–周波数解析における不確定性原理と呼び，時刻の不確定さ Δt と周波数の不確定さ $\Delta\omega$（信号の持続時間と帯域幅の標準偏差）の間に以下のような関係がある。

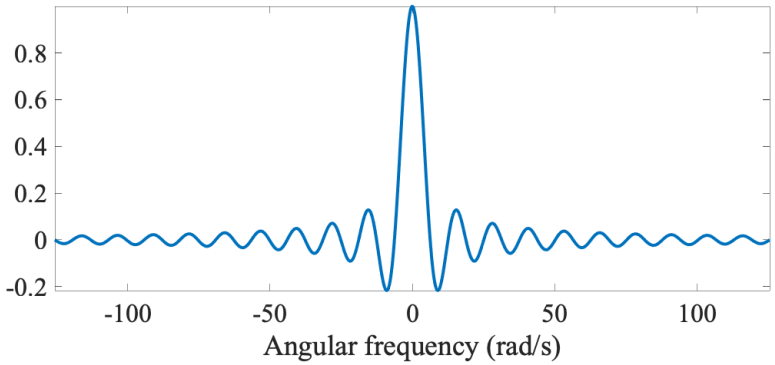
$$\Delta t \Delta\omega \geq \frac{1}{2}$$

不確定性原理

- 矩形波パルスのFourier変換を思い出すと，

$$f(t) = \begin{cases} A, & |t| \le T/2 \\ 0, & |t| > T/2 \end{cases} \quad \overset{\mathrm{FT}}{\longleftrightarrow} \quad F(\omega) = \int_{-T/2}^{T/2} A \exp(-\mathrm{j}\omega t)\mathrm{d}t = AT\frac{\sin \omega T/2}{\omega T/2}$$





➡ 矩形波パルスを例に不確定性原理を考える

不確定性原理

- $F(\omega)$ を $-\omega' \leq \omega \leq \omega'$ に帯域制限した信号のエネルギーと，全エネルギーとの比を求めると，

$$\frac{\int_{-\omega'}^{\omega'} |F(\omega)|^2 \mathrm{d}\omega}{\int_{-\infty}^{\infty} |F(\omega)|^2 \mathrm{d}\omega} = \frac{1}{\pi} \int_{-\omega'}^{\omega'} \left| \frac{\sin \omega T/2}{\omega T/2} \right|^2 \mathrm{d}\left(\frac{\omega T}{2}\right)$$

  - これが1に近いほど信号をよく表現できているとみなせる。

- 矩形波パルス幅 T が小さいほど，信号を表現するために必要な帯域幅が広くなる
- 矩形波パルス幅 T が大きいほど，狭い帯域幅でも信号を表現できる

➡ $\Delta t \Delta \omega \geq \frac{1}{2}$ の証明は信号処理論第二で・・・

高速FOURIER変換

高速Fourier変換

- 高速Fourier変換 (Fast Fourier transform: FFT)は離散Fourier変換 (DFT)の高速なアルゴリズム
  - 特別なFourier変換ではないことに注意

準備

- 離散Fourier変換 (DFT)の復習。
  - 定義

$$X[k] = \sum_{n=0}^{N-1} x[n] \exp\left(-\mathrm{j}\frac{2\pi kn}{N}\right)$$

  - 行列表現

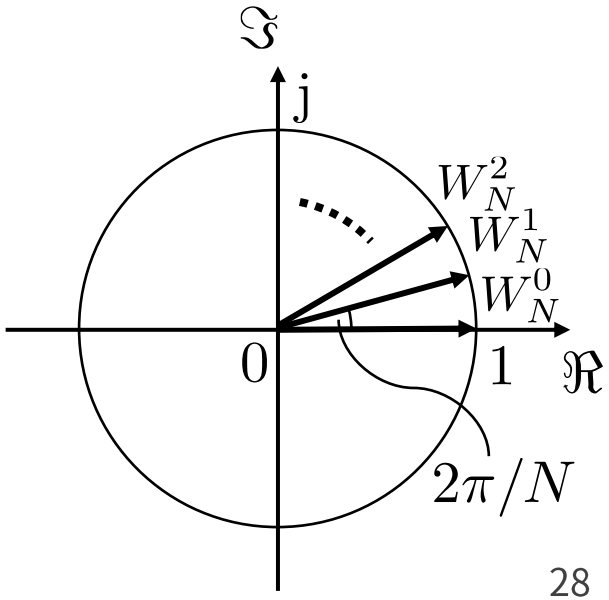
$$\boldsymbol{X} = \boldsymbol{W}_N^{\mathsf{H}} \boldsymbol{x}$$

$\boldsymbol{X} = [X[0], \ldots, X[N-1]]^{\mathsf{T}}$
$\boldsymbol{x} = [x[0], \ldots, x[N-1]]^{\mathsf{T}}$

$$(\boldsymbol{W}_N)_{nk} = W_N^{kn} = \exp\left(\mathrm{j}\frac{2\pi kn}{N}\right)$$

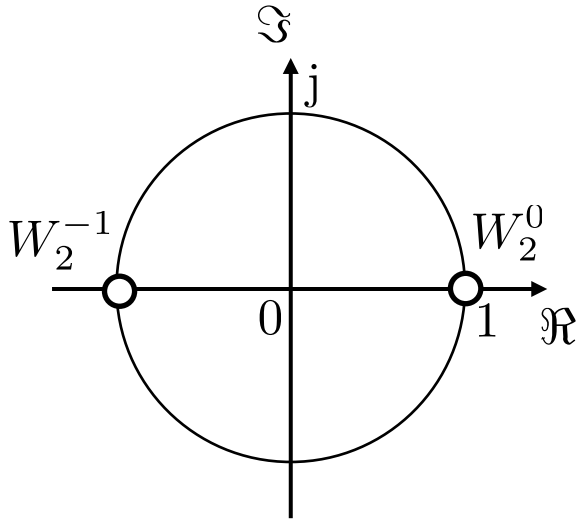
$\boldsymbol{W}_N$の(n,k)番目の要素

$$\left(W_N = \exp\left(\mathrm{j}\frac{2\pi}{N}\right) \right)$$

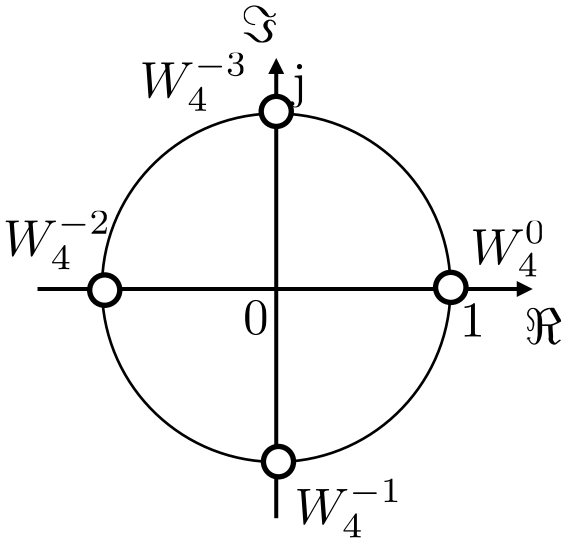


準備

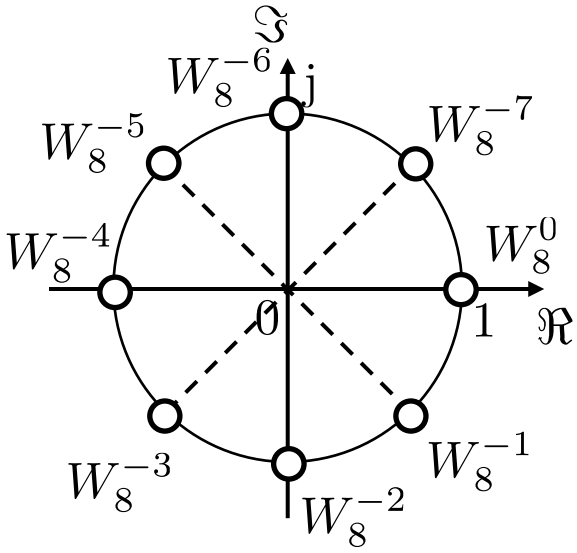
- DFT行列 $\boldsymbol{W}_N$の対称性
  - $W_N^n = W_N^{n+N}$
  - Nが偶数のとき $W_N^n = -W_N^{n+\frac{N}{2}}$



$N = 2$



$N = 4$



$N = 8$

高速Fourier変換（N=8の場合）

- $N = 8$ のときのDFT

$$
\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \\ X[4] \\ X[5] \\ X[6] \\ X[7] \end{bmatrix} = \begin{bmatrix} W_8^{-0\cdot0} & W_8^{-1\cdot0} & W_8^{-2\cdot0} & W_8^{-3\cdot0} & W_8^{-4\cdot0} & W_8^{-5\cdot0} & W_8^{-6\cdot0} & W_8^{-7\cdot0} \\ W_8^{-0\cdot1} & W_8^{-1\cdot1} & W_8^{-2\cdot1} & W_8^{-3\cdot1} & W_8^{-4\cdot1} & W_8^{-5\cdot1} & W_8^{-6\cdot1} & W_8^{-7\cdot1} \\ W_8^{-0\cdot2} & W_8^{-1\cdot2} & W_8^{-2\cdot2} & W_8^{-3\cdot2} & W_8^{-4\cdot2} & W_8^{-5\cdot2} & W_8^{-6\cdot2} & W_8^{-7\cdot2} \\ W_8^{-0\cdot3} & W_8^{-1\cdot3} & W_8^{-2\cdot3} & W_8^{-3\cdot3} & W_8^{-4\cdot3} & W_8^{-5\cdot3} & W_8^{-6\cdot3} & W_8^{-7\cdot3} \\ W_8^{-0\cdot4} & W_8^{-1\cdot4} & W_8^{-2\cdot4} & W_8^{-3\cdot4} & W_8^{-4\cdot4} & W_8^{-5\cdot4} & W_8^{-6\cdot4} & W_8^{-7\cdot4} \\ W_8^{-0\cdot5} & W_8^{-1\cdot5} & W_8^{-2\cdot5} & W_8^{-3\cdot5} & W_8^{-4\cdot5} & W_8^{-5\cdot5} & W_8^{-6\cdot5} & W_8^{-7\cdot5} \\ W_8^{-0\cdot6} & W_8^{-1\cdot6} & W_8^{-2\cdot6} & W_8^{-3\cdot6} & W_8^{-4\cdot6} & W_8^{-5\cdot6} & W_8^{-6\cdot6} & W_8^{-7\cdot6} \\ W_8^{-0\cdot7} & W_8^{-1\cdot7} & W_8^{-2\cdot7} & W_8^{-3\cdot7} & W_8^{-4\cdot7} & W_8^{-5\cdot7} & W_8^{-6\cdot7} & W_8^{-7\cdot7} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \\ x[4] \\ x[5] \\ x[6] \\ x[7] \end{bmatrix}
$$

そのままだと $O(N^2)$ の計算量

高速Fourier変換（N=8の場合）

- 対称性 ($W_N^n = W_N^{n+N}$)を利用

$$
\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \\ X[4] \\ X[5] \\ X[6] \\ X[7] \end{bmatrix} = \begin{bmatrix} W_8^{-0} & W_8^{-0} & W_8^{-0} & W_8^{-0} & W_8^{-0} & W_8^{-0} & W_8^{-0} & W_8^{-0} \\ W_8^{-0} & W_8^{-1} & W_8^{-2} & W_8^{-3} & W_8^{-4} & W_8^{-5} & W_8^{-6} & W_8^{-7} \\ W_8^{-0} & W_8^{-2} & W_8^{-4} & W_8^{-6} & W_8^{-8} & W_8^{-10} & W_8^{-12} & W_8^{-14} \\ W_8^{-0} & W_8^{-3} & W_8^{-6} & W_8^{-9} & W_8^{-12} & W_8^{-15} & W_8^{-18} & W_8^{-21} \\ W_8^{-0} & W_8^{-4} & W_8^{-8} & W_8^{-12} & W_8^{-16} & W_8^{-20} & W_8^{-24} & W_8^{-28} \\ W_8^{-0} & W_8^{-5} & W_8^{-10} & W_8^{-15} & W_8^{-20} & W_8^{-25} & W_8^{-30} & W_8^{-35} \\ W_8^{-0} & W_8^{-6} & W_8^{-12} & W_8^{-18} & W_8^{-24} & W_8^{-30} & W_8^{-36} & W_8^{-42} \\ W_8^{-0} & W_8^{-7} & W_8^{-14} & W_8^{-21} & W_8^{-28} & W_8^{-35} & W_8^{-42} & W_8^{-49} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \\ x[4] \\ x[5] \\ x[6] \\ x[7] \end{bmatrix}
$$

$$
= \left[\begin{array}{cccc|cccc} W_8^{-0} & W_8^{-0} & W_8^{-0} & W_8^{-0} & W_8^{-0} & W_8^{-0} & W_8^{-0} & W_8^{-0} \\ W_8^{-0} & W_8^{-1} & W_8^{-2} & W_8^{-3} & W_8^{-4} & W_8^{-5} & W_8^{-6} & W_8^{-7} \\ W_8^{-0} & W_8^{-2} & W_8^{-4} & W_8^{-6} & W_8^{-0} & W_8^{-2} & W_8^{-4} & W_8^{-6} \\ W_8^{-0} & W_8^{-3} & W_8^{-6} & W_8^{-1} & W_8^{-4} & W_8^{-7} & W_8^{-2} & W_8^{-5} \\ W_8^{-0} & W_8^{-4} & W_8^{-0} & W_8^{-4} & W_8^{-0} & W_8^{-4} & W_8^{-0} & W_8^{-4} \\ W_8^{-0} & W_8^{-5} & W_8^{-2} & W_8^{-7} & W_8^{-4} & W_8^{-1} & W_8^{-6} & W_8^{-3} \\ W_8^{-0} & W_8^{-6} & W_8^{-4} & W_8^{-2} & W_8^{-0} & W_8^{-6} & W_8^{-4} & W_8^{-2} \\ W_8^{-0} & W_8^{-7} & W_8^{-6} & W_8^{-5} & W_8^{-4} & W_8^{-3} & W_8^{-2} & W_8^{-1} \end{array}\right] \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \\ x[4] \\ x[5] \\ x[6] \\ x[7] \end{bmatrix}
$$

- 偶数番目の要素は半分に分けたときの左右の要素が同じ
- 奇数番目の要素は半周期 (W_8^{-4})ずれたもの

高速Fourier変換（N=8の場合）

- $X[k]$の順序を並べ替えて分割

$$
\begin{bmatrix} X[0] \\ X[2] \\ X[4] \\ X[6] \\ X[1] \\ X[3] \\ X[5] \\ X[7] \end{bmatrix} = \left[\begin{array}{cccc:cccc} W_8^{-0} & W_8^{-0} & W_8^{-0} & W_8^{-0} & W_8^{-0} & W_8^{-0} & W_8^{-0} & W_8^{-0} \\ W_8^{-0} & W_8^{-2} & W_8^{-4} & W_8^{-6} & W_8^{-0} & W_8^{-2} & W_8^{-4} & W_8^{-6} \\ W_8^{-0} & W_8^{-4} & W_8^{-0} & W_8^{-4} & W_8^{-0} & W_8^{-4} & W_8^{-0} & W_8^{-4} \\ W_8^{-0} & W_8^{-6} & W_8^{-4} & W_8^{-2} & W_8^{-0} & W_8^{-6} & W_8^{-4} & W_8^{-2} \\ \hdashline W_8^{-0} & W_8^{-1} & W_8^{-2} & W_8^{-3} & W_8^{-4} & W_8^{-5} & W_8^{-6} & W_8^{-7} \\ W_8^{-0} & W_8^{-3} & W_8^{-6} & W_8^{-1} & W_8^{-4} & W_8^{-7} & W_8^{-2} & W_8^{-5} \\ W_8^{-0} & W_8^{-5} & W_8^{-2} & W_8^{-7} & W_8^{-4} & W_8^{-1} & W_8^{-6} & W_8^{-3} \\ W_8^{-0} & W_8^{-7} & W_8^{-6} & W_8^{-5} & W_8^{-4} & W_8^{-3} & W_8^{-2} & W_8^{-1} \end{array}\right] \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \\ x[4] \\ x[5] \\ x[6] \\ x[7] \end{bmatrix}
$$

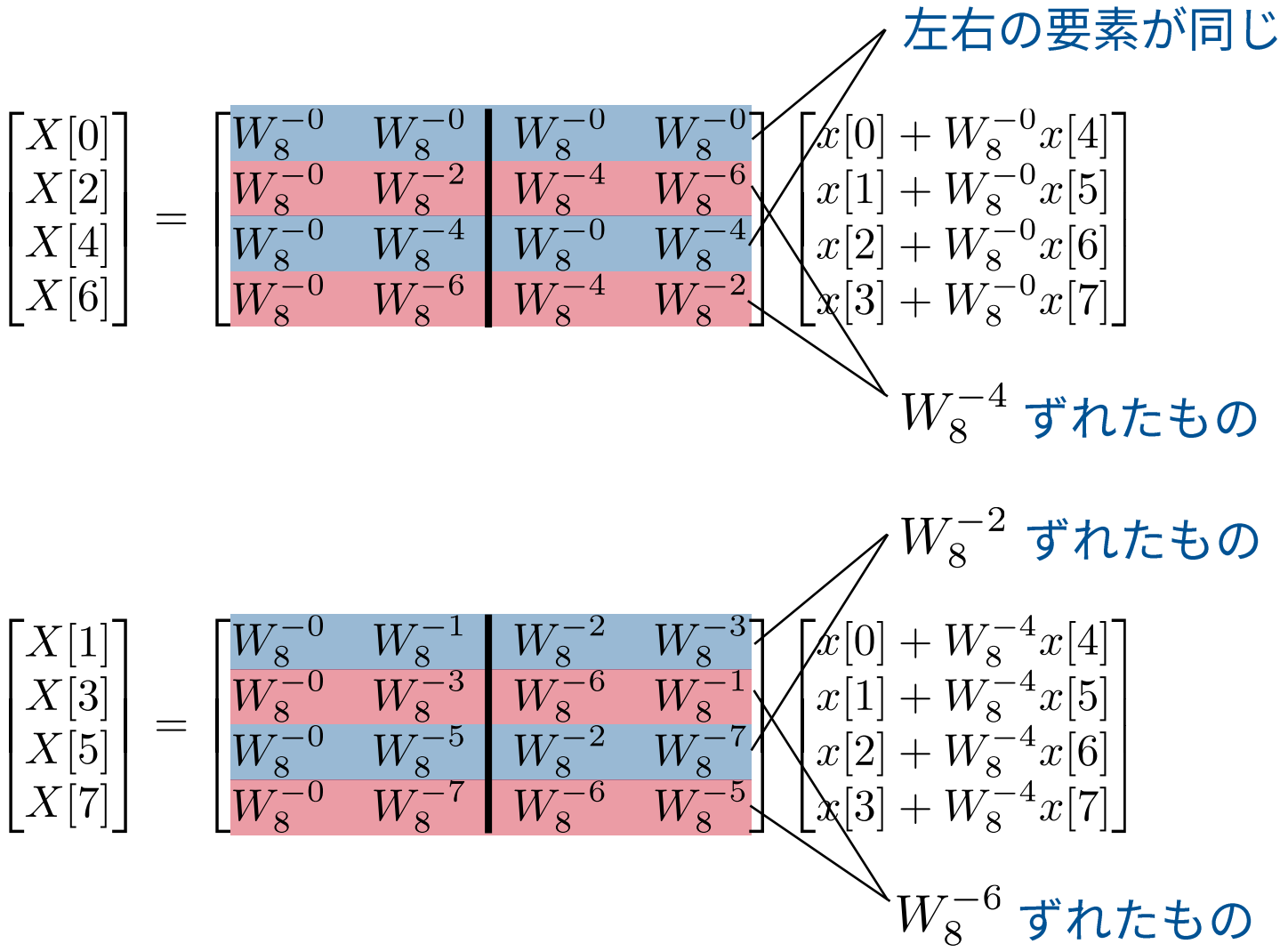
分割

$$
\begin{bmatrix} X[0] \\ X[2] \\ X[4] \\ X[6] \end{bmatrix} = \begin{bmatrix} W_8^{-0} & W_8^{-0} & W_8^{-0} & W_8^{-0} \\ W_8^{-0} & W_8^{-2} & W_8^{-4} & W_8^{-6} \\ W_8^{-0} & W_8^{-4} & W_8^{-0} & W_8^{-4} \\ W_8^{-0} & W_8^{-6} & W_8^{-4} & W_8^{-2} \end{bmatrix} \begin{bmatrix} x[0] + W_8^{-0}x[4] \\ x[1] + W_8^{-0}x[5] \\ x[2] + W_8^{-0}x[6] \\ x[3] + W_8^{-0}x[7] \end{bmatrix}
$$

$$
\begin{bmatrix} X[1] \\ X[3] \\ X[5] \\ X[7] \end{bmatrix} = \begin{bmatrix} W_8^{-0} & W_8^{-1} & W_8^{-2} & W_8^{-3} \\ W_8^{-0} & W_8^{-3} & W_8^{-6} & W_8^{-1} \\ W_8^{-0} & W_8^{-5} & W_8^{-2} & W_8^{-7} \\ W_8^{-0} & W_8^{-7} & W_8^{-6} & W_8^{-5} \end{bmatrix} \begin{bmatrix} x[0] + W_8^{-4}x[4] \\ x[1] + W_8^{-4}x[5] \\ x[2] + W_8^{-4}x[6] \\ x[3] + W_8^{-4}x[7] \end{bmatrix}
$$

高速Fourier変換（N=8の場合）

- 再び行列の左右の関係を見てみる。



高速Fourier変換（N=8の場合）

- 再び並べ替えて分割（一つ目）

$$\begin{bmatrix} X[0] \\ X[4] \\ X[2] \\ X[6] \end{bmatrix} = \left[\begin{array}{cc:cc} W_8^{-0} & W_8^{-0} & W_8^{-0} & W_8^{-0} \\ W_8^{-0} & W_8^{-4} & W_8^{-0} & W_8^{-4} \\ \hdashline W_8^{-0} & W_8^{-2} & W_8^{-4} & W_8^{-6} \\ W_8^{-0} & W_8^{-6} & W_8^{-4} & W_8^{-2} \end{array}\right] \begin{bmatrix} x[0] + W_8^{-0}x[4] \\ x[1] + W_8^{-0}x[5] \\ x[2] + W_8^{-0}x[6] \\ x[3] + W_8^{-0}x[7] \end{bmatrix}$$

分割

$$\begin{bmatrix} X[0] \\ X[4] \end{bmatrix} = \begin{bmatrix} W_8^{-0} & W_8^{-0} \\ W_8^{-0} & W_8^{-4} \end{bmatrix} \begin{bmatrix} x[0] + W_8^{-0}x[4] + W_8^{-0}(x[2] + W_8^{-0}x[6]) \\ x[1] + W_8^{-0}x[5] + W_8^{-0}(x[3] + W_8^{-0}x[7]) \end{bmatrix}$$

$$\begin{bmatrix} X[2] \\ X[6] \end{bmatrix} = \begin{bmatrix} W_8^{-0} & W_8^{-2} \\ W_8^{-0} & W_8^{-6} \end{bmatrix} \begin{bmatrix} x[0] + W_8^{-0}x[4] + W_8^{-4}(x[2] + W_8^{-0}x[6]) \\ x[1] + W_8^{-0}x[5] + W_8^{-4}(x[3] + W_8^{-0}x[7]) \end{bmatrix}$$

高速Fourier変換（N=8の場合）

- 再び並べ替えて分割（二つ目）

$$\begin{bmatrix} X[1] \\ X[5] \\ X[3] \\ X[7] \end{bmatrix} = \left[\begin{array}{cc:cc} W_8^{-0} & W_8^{-1} & W_8^{-2} & W_8^{-3} \\ W_8^{-0} & W_8^{-5} & W_8^{-2} & W_8^{-7} \\ \hdashline W_8^{-0} & W_8^{-3} & W_8^{-6} & W_8^{-1} \\ W_8^{-0} & W_8^{-7} & W_8^{-6} & W_8^{-5} \end{array}\right] \begin{bmatrix} x[0] + W_8^{-4}x[4] \\ x[1] + W_8^{-4}x[5] \\ x[2] + W_8^{-4}x[6] \\ x[3] + W_8^{-4}x[7] \end{bmatrix}$$

分割

$$\begin{bmatrix} X[1] \\ X[5] \end{bmatrix} = \begin{bmatrix} W_8^{-0} & W_8^{-1} \\ W_8^{-0} & W_8^{-5} \end{bmatrix} \begin{bmatrix} x[0] + W_8^{-4}x[4] + W_8^{-2}(x[2] + W_8^{-4}x[6]) \\ x[1] + W_8^{-4}x[5] + W_8^{-2}(x[3] + W_8^{-4}x[7]) \end{bmatrix}$$

$$\begin{bmatrix} X[3] \\ X[7] \end{bmatrix} = \begin{bmatrix} W_8^{-0} & W_8^{-3} \\ W_8^{-0} & W_8^{-7} \end{bmatrix} \begin{bmatrix} x[0] + W_8^{-4}x[4] + W_8^{-6}(x[2] + W_8^{-4}x[6]) \\ x[1] + W_8^{-4}x[5] + W_8^{-6}(x[3] + W_8^{-4}x[7]) \end{bmatrix}$$

高速Fourier変換（N=8の場合）

- 分割したものを全て書くと，

$$\begin{bmatrix} X[0] \\ X[4] \end{bmatrix} = \begin{bmatrix} W_8^{-0} & W_8^{-0} \\ W_8^{-0} & W_8^{-4} \end{bmatrix} \begin{bmatrix} x[0] + W_8^{-0}x[4] + W_8^{-0}(x[2] + W_8^{-0}x[6]) \\ x[1] + W_8^{-0}x[5] + W_8^{-0}(x[3] + W_8^{-0}x[7]) \end{bmatrix}$$

$$\begin{bmatrix} X[2] \\ X[6] \end{bmatrix} = \begin{bmatrix} W_8^{-0} & W_8^{-2} \\ W_8^{-0} & W_8^{-6} \end{bmatrix} \begin{bmatrix} x[0] + W_8^{-0}x[4] + W_8^{-4}(x[2] + W_8^{-0}x[6]) \\ x[1] + W_8^{-0}x[5] + W_8^{-4}(x[3] + W_8^{-0}x[7]) \end{bmatrix}$$

$$\begin{bmatrix} X[1] \\ X[5] \end{bmatrix} = \begin{bmatrix} W_8^{-0} & W_8^{-1} \\ W_8^{-0} & W_8^{-5} \end{bmatrix} \begin{bmatrix} x[0] + W_8^{-4}x[4] + W_8^{-2}(x[2] + W_8^{-4}x[6]) \\ x[1] + W_8^{-4}x[5] + W_8^{-2}(x[3] + W_8^{-4}x[7]) \end{bmatrix}$$

$$\begin{bmatrix} X[3] \\ X[7] \end{bmatrix} = \begin{bmatrix} W_8^{-0} & W_8^{-3} \\ W_8^{-0} & W_8^{-7} \end{bmatrix} \begin{bmatrix} x[0] + W_8^{-4}x[4] + W_8^{-6}(x[2] + W_8^{-4}x[6]) \\ x[1] + W_8^{-4}x[5] + W_8^{-6}(x[3] + W_8^{-4}x[7]) \end{bmatrix}$$

高速Fourier変換（N=8の場合）

- 対称性 $W_N^n = -W_N^{n+\frac{N}{2}}$ を用いて書き換えると，

$$\begin{bmatrix} X[0] \\ X[4] \end{bmatrix} = \begin{bmatrix} W_8^{-0} & W_8^{-0} \\ W_8^{-0} & -W_8^{-0} \end{bmatrix} \begin{bmatrix} x[0] + W_8^{-0}x[4] + W_8^{-0}(x[2] + W_8^{-0}x[6]) \\ x[1] + W_8^{-0}x[5] + W_8^{-0}(x[3] + W_8^{-0}x[7]) \end{bmatrix}$$

$$\begin{bmatrix} X[2] \\ X[6] \end{bmatrix} = \begin{bmatrix} W_8^{-0} & W_8^{-2} \\ W_8^{-0} & -W_8^{-2} \end{bmatrix} \begin{bmatrix} x[0] + W_8^{-0}x[4] - W_8^{-0}(x[2] + W_8^{-0}x[6]) \\ x[1] + W_8^{-0}x[5] - W_8^{-0}(x[3] + W_8^{-0}x[7]) \end{bmatrix}$$

$$\begin{bmatrix} X[1] \\ X[5] \end{bmatrix} = \begin{bmatrix} W_8^{-0} & W_8^{-1} \\ W_8^{-0} & -W_8^{-1} \end{bmatrix} \begin{bmatrix} x[0] - W_8^{-0}x[4] + W_8^{-2}(x[2] - W_8^{-0}x[6]) \\ x[1] - W_8^{-0}x[5] + W_8^{-2}(x[3] - W_8^{-0}x[7]) \end{bmatrix}$$

$$\begin{bmatrix} X[3] \\ X[7] \end{bmatrix} = \begin{bmatrix} W_8^{-0} & W_8^{-3} \\ W_8^{-0} & -W_8^{-3} \end{bmatrix} \begin{bmatrix} x[0] - W_8^{-0}x[4] - W_8^{-2}(x[2] - W_8^{-0}x[6]) \\ x[1] - W_8^{-0}x[5] - W_8^{-2}(x[3] - W_8^{-0}x[7]) \end{bmatrix}$$

高速Fourier変換（N=8の場合）

- 最初の式について計算手順を書き下すと，

$$\begin{bmatrix} X[0] \\ X[4] \end{bmatrix} = \begin{bmatrix} W_8^{-0} & W_8^{-0} \\ W_8^{-0} & -W_8^{-0} \end{bmatrix} \begin{bmatrix} x[0] + W_8^{-0}x[4] + W_8^{-0}(x[2] + W_8^{-0}x[6]) \\ x[1] + W_8^{-0}x[5] + W_8^{-0}(x[3] + W_8^{-0}x[7]) \end{bmatrix}$$

$$a_0 = \begin{bmatrix} 1 & W_8^{-0} \end{bmatrix} \begin{bmatrix} x[0] \\ x[4] \end{bmatrix}$$

$$a_1 = \begin{bmatrix} 1 & W_8^{-0} \end{bmatrix} \begin{bmatrix} x[2] \\ x[6] \end{bmatrix}$$

$$a_2 = \begin{bmatrix} 1 & W_8^{-0} \end{bmatrix} \begin{bmatrix} x[1] \\ x[5] \end{bmatrix}$$

$$a_3 = \begin{bmatrix} 1 & W_8^{-0} \end{bmatrix} \begin{bmatrix} x[3] \\ x[7] \end{bmatrix}$$

$$b_0 = \begin{bmatrix} 1 & W_8^{-0} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix}$$

$$b_1 = \begin{bmatrix} 1 & W_8^{-0} \end{bmatrix} \begin{bmatrix} a_2 \\ a_3 \end{bmatrix}$$

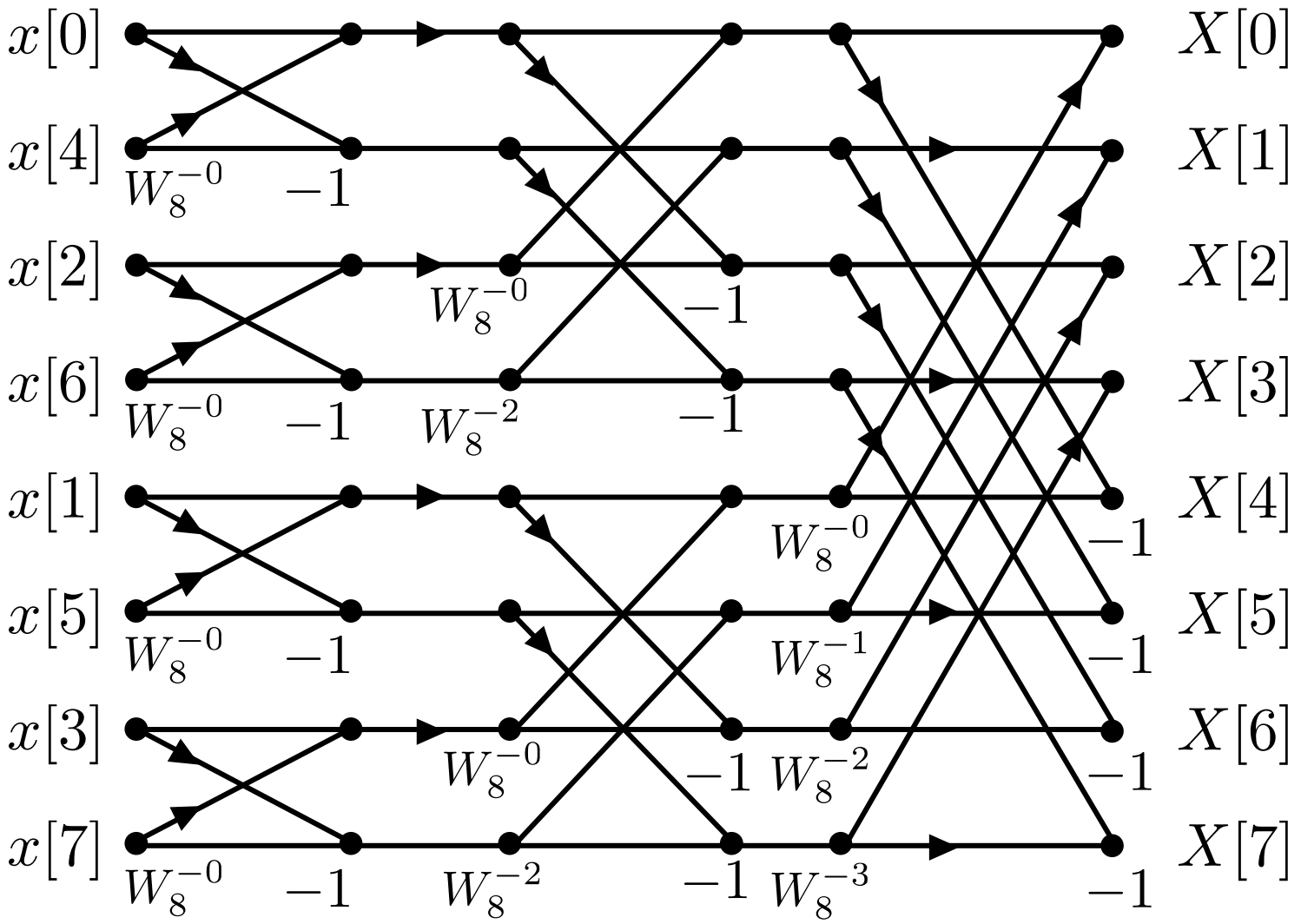
$$X[0] = \begin{bmatrix} W_8^{-0} & W_8^{-0} \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$$

$$X[4] = \begin{bmatrix} W_8^{-0} & -W_8^{-0} \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$$

➡ 内積の計算の繰り返しによって最終的な結果が得られる

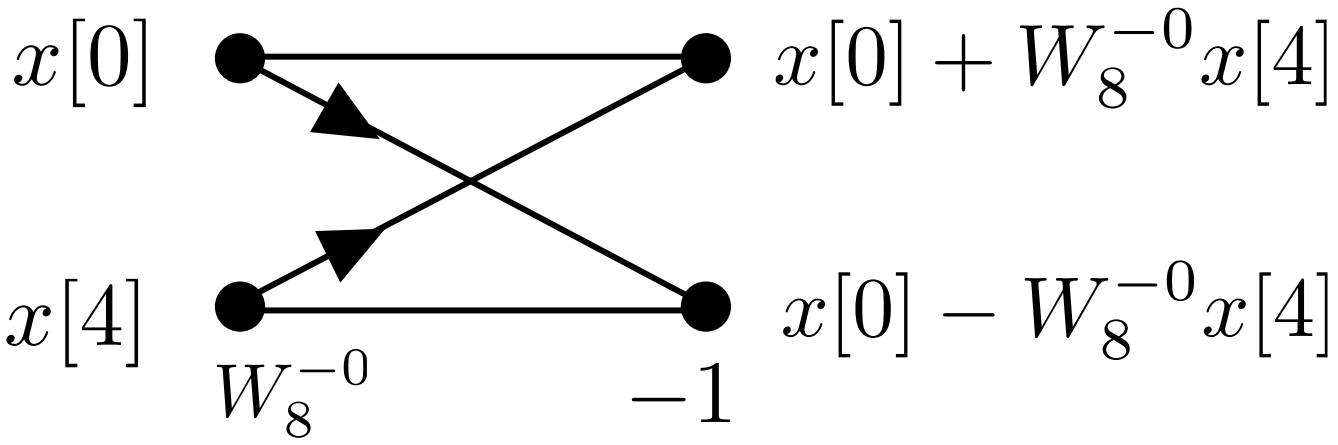
高速Fourier変換（N=8の場合）

- 全体の演算を図で表す（黒丸で数値を乗算後に加算）



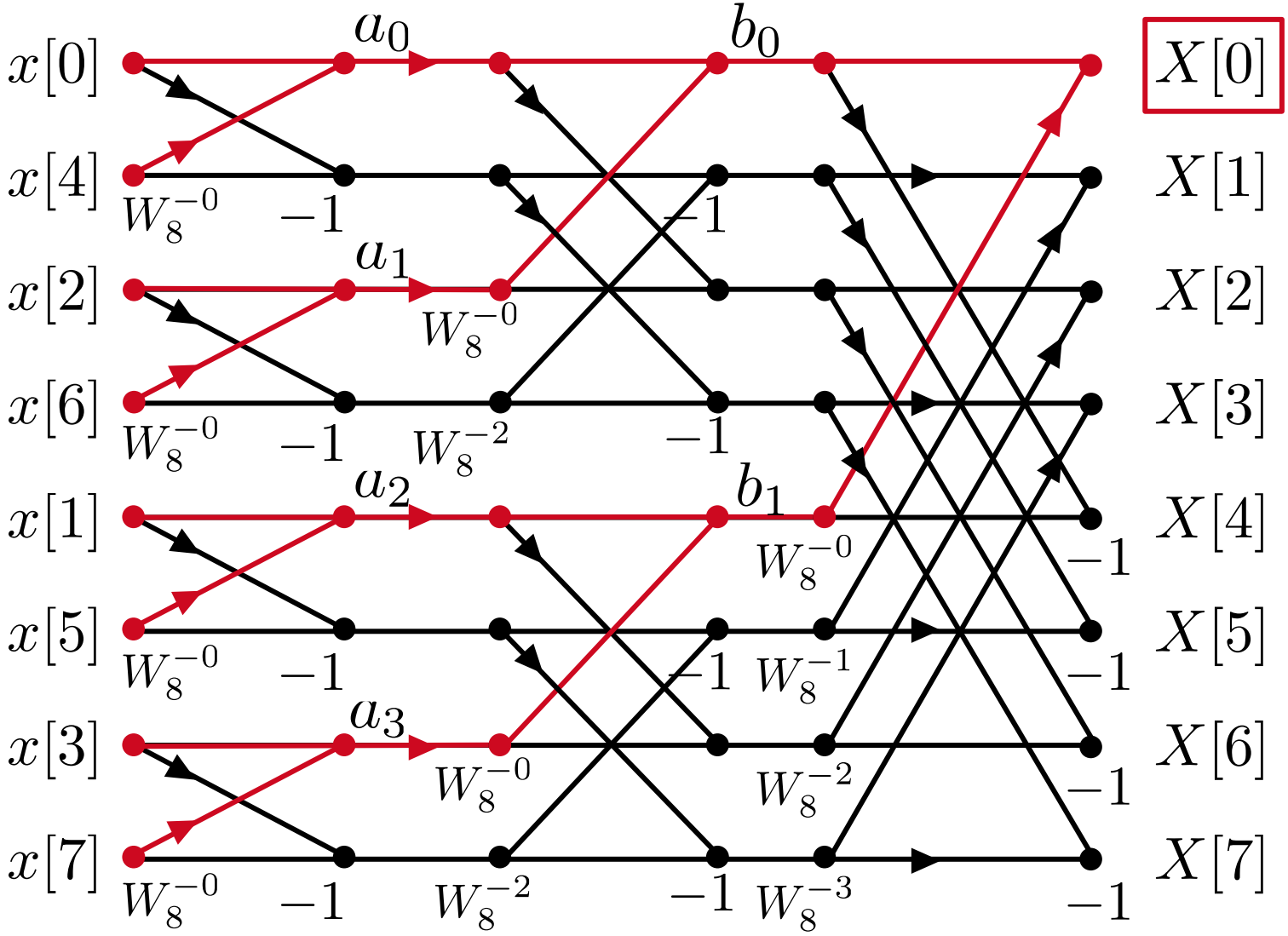
高速Fourier変換（N=8の場合）

- FFTは複素加算2回，複素乗算1回のバタフライ演算を基本演算とし，これの組み合わせで構成される。
- Nが2のべき乗であればこの例と同じ手順で計算が可能。



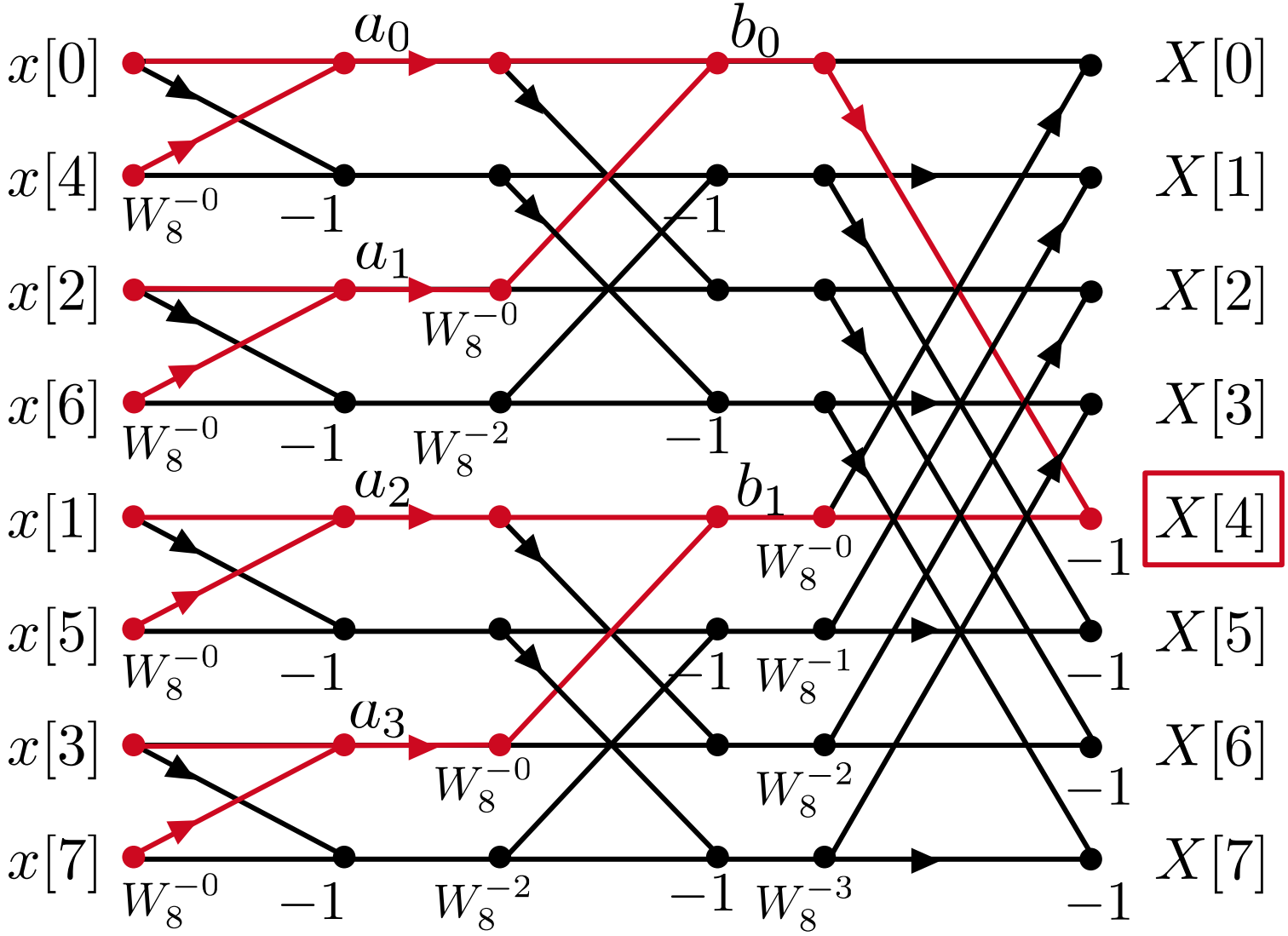
高速Fourier変換（N=8の場合）

- 全体の演算を図で表す（黒丸で数値を乗算後に加算）



高速Fourier変換（N=8の場合）

- 全体の演算を図で表す（黒丸で数値を乗算後に加算）



高速Fourier変換（N=8の場合）

- バタフライ演算を基本演算とするFFTはNが2のべき乗の場合に適用可能であり，その総数は $N/2\log_2 N$ となる。
- N=8の例での計算量
  - DFT: 複素加算 56回，複素乗算 64回
  - FFT: 複素加算 24回，複素乗算 12回
- 一般の場合の計算量
  - DFT: 複素加算 $N(N-1)$，複素乗算 N^2
  - FFT: 複素加算 $N\log_2 N$ ，複素乗算 $N/2\log_2 N$
- 計算オーダ
  - DFT: $O(N^2)$
  - FFT: $O(N\log_2 N)$

ビットリバース

- FFTでは信号の要素を並び替える必要が生じるが，これはビットを逆順にすることで簡単に実現できる。

2進数			ビットリバース	
0	000	→	000	0
1	001	→	100	4
2	010	→	010	2
3	011	→	110	6
4	100	→	001	1
5	101	→	101	5
6	110	→	011	3
7	111	→	111	7

FFTの畳み込み演算への適用

- FFTはDFTを高速に処理するためだけでなく，畳み込みや相関関数の計算を高速化することにも利用できる。
- DFTの巡回畳み込みと乗算の性質

$$\mathrm{DFT}\left[x[n] \circledast y[n]\right] = X[k]Y[k]$$

- 実用上は巡回畳み込みではなく，有限区間の離散時間信号間の畳み込みを計算したい場合が多い。

$$x[n] * y[n]$$

➡ これをFFTを使って計算するには，巡回畳み込みによって畳み込み演算を表現することが必要。

零詰め

- 有限長の２つの信号の畳み込みを計算したい。

$$x[n]\ (n \in \{0, \ldots, N-1\})$$

$$y[n]\ (n \in \{0, \ldots, M-1\})$$

畳み込み演算の定義

$$x[n] * y[n] = \sum_{m=-\infty}^{\infty} x[m]y[n-m]$$

- 零詰め (zeropadding)

$$x'[n] = \begin{cases} x[n], & n \in [0, N-1] \\ 0, & \text{otherwise} \end{cases}$$

$$y'[n] = \begin{cases} y[n], & n \in [0, M-1] \\ 0, & \text{otherwise} \end{cases}$$

零詰め

- 長さ$N+M-1$とする零詰め

$$x'[n] = \begin{cases} x[n], & n \in [0, N-1] \\ 0, & n \in [N, N+M-2] \end{cases}$$

$$y'[n] = \begin{cases} y[n], & n \in [0, M-1] \\ 0, & n \in [M, N+M-2] \end{cases}$$

- 零詰めした信号の周期的拡張

$$\tilde{x}'[n] = [\ldots, x'[n], x'[n], \ldots]$$

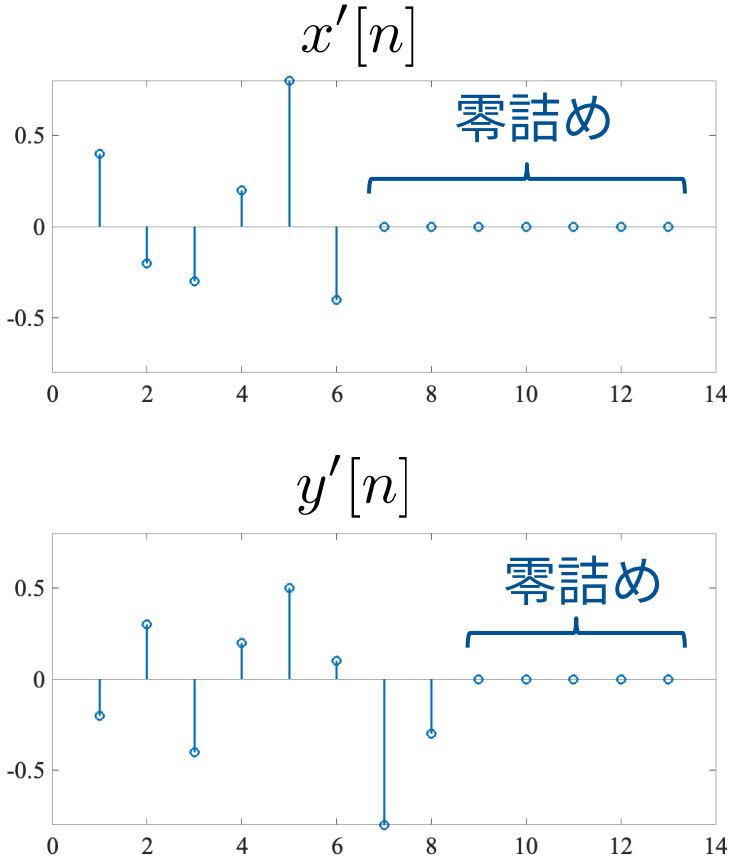
$$\tilde{y}'[n] = [\ldots, y'[n], y'[n], \ldots]$$

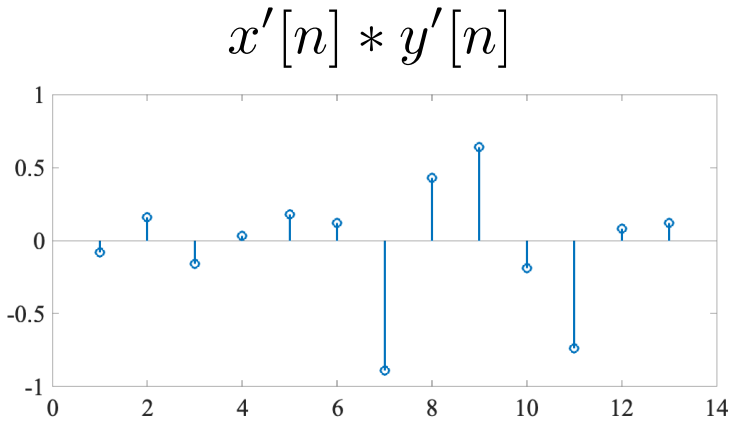
有限長の信号間の畳み込み

- N点の信号とM点の信号の畳み込みは，

$$x'[n] * y'[n]$$

の自明に0とならない部分を利用する。





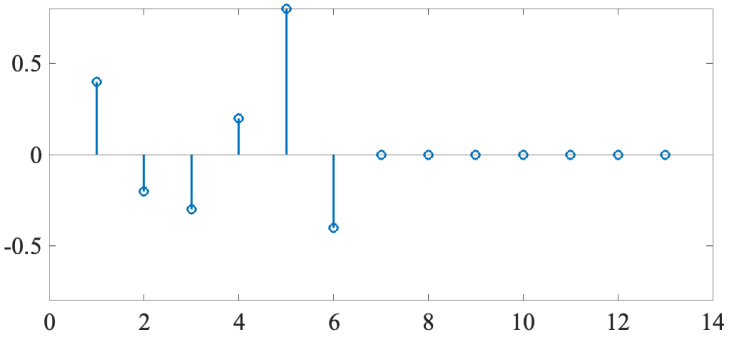
信号長は$N + M - 1$となる

零詰めした信号の巡回畳み込み

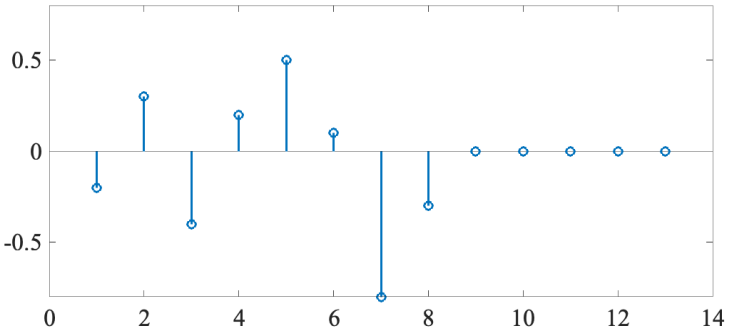
- $x'[n] \circledast y'[n]$ は$x'[n]$と$y'[n]$の周期的拡張 $\tilde{x}'[n]$と$\tilde{y}'[n]$の畳み込みに相当する。

$$x'[n] \circledast y'[n] = \sum_{m=0}^{N+M-2} x'[m]y'[n-m]$$

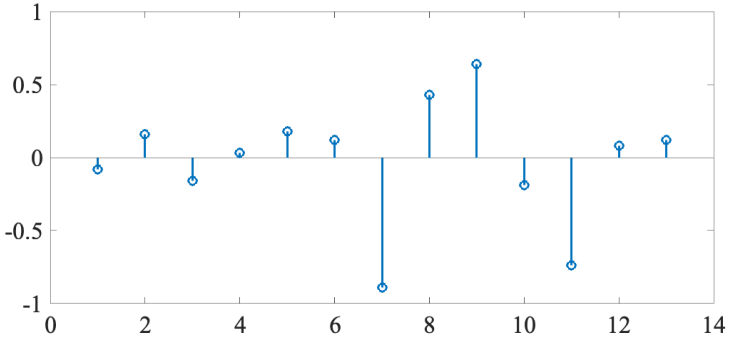
$x'[n]$



$y'[n]$



$x'[n] \circledast y'[n]$



$x'[n] * y'[n]$ と同じ結果となる。

FFTの畳み込み演算への適用

- 巡回畳み込みを有限長の信号間の畳み込みにするため，信号の長さがN+M-1となるように零詰め。
- この零詰めの長さは必要最小限の数であり，これより多くても構わない。
- 信号の長さが2のべき乗個となるまで冗長に零詰めすれば，FFTが利用できることになる。

- Pythonでの実例：
  - https://colab.research.google.com/drive/1BP7G5DosAhtkcgSSSkS_EpJimNjEbgLD?usp=sharing

参考文献

1. 眞溪歩, "ディジタル信号処理工学," 昭晃堂, 2004.
2. M. Vetterli, J. Kovačević, and V. K. Goyal, "Foundations of Signal Processing," Cambridge University Press, 2014.
3. L. コーエン, “時間-周波数解析,” 朝倉書店, 1998.