ディジタル信号処理
第 **13** 回
信号の変換符号化

今回の講義で取り扱う信号

- 今回の講義では, 長さ N の有限長信号

$$x = (x[0], \ldots, x[N-1]),$$

あるいは, それを周期関数に拡張したものを検討の対象とする.

変換符号化とは何か

- 信号処理では(アナログ・ディジタルともに)フーリエ変換がよく使われるが …
- なぜフーリエ変換なのかを考えてみると …
- 信号の特定の周波数成分に有用な情報が偏在している (あるいは偏在させてある) ことが多いから.

- 信号の変換とは, 広義には信号を加工して別の信号を作ることであるが, 狭義には信号をベクトルと見倣してそれに正則行列 (特にユニタリ行列) などの行列を乗ずる操作を指す.

- 信号の変換により, 変換後の信号の特定の成分に有用な情報が集中するようにできれば, 有用でない成分の削除や量子化ビット数を割り振りをの工夫したにより, データ圧縮の効率を高めることができる.
- このような目的で信号を変換してから符号化する手法を (そのままであるが) 信号の**変換符号化**と呼ぶ.

- ディジタル信号処理では, 信号の変換が多用される.
- その目的がデータ圧縮である場合もあるが, 目的が異なることも多い.
- たとえば, 連続/離散フーリエ変換や z 変換はデータ圧縮を目的とするわけではない.

教科書では「正規直交行列」「正規ユニタリ行列」などといった言葉が多用されているが, そんな数学用語はないので注意. 正しくは「直交行列」「ユニタリ行列」である.

- 以下では, 有限長信号に行列を乗じるタイプの信号の変換の代表的なものを紹介する.
- 講義の標題は教科書に合わせて信号の変換符号化としているが, 講義では, 符号化に限定せずに信号の変換について議論してゆく.
- 信号の変換には音声処理と画像処理でおおむね共通の関数等が用いられる. 今回の講義では音声を中心に述べる.

信号の変換

アフィン変換

- アフィン変換とは, ベクトルに対する一般的な演算で, ベクトルの線形変換と平行移動の組み合わせである. 有限長信号 x を有限次のベクトル $\boldsymbol{x}$ と同一視した場合, アフィン変換は $\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$ と書ける ($\boldsymbol{A}$ は N 次の行列, $\boldsymbol{b}$ は N 次のベクトル). 画像処理で多用される.

離散フーリエ変換

- 離散フーリエ変換は信号の変換のなかで最もよく用いられるもののひとつ．離散フーリエ変換についてはすでに詳しく述べているので繰り返さない．

カーネン・レーベ変換

- カーネン・レーベ変換は定常不規則信号を前提とした変換で, 統計で用いられる主成分分析と本質的に同じ.
- 信号の自己相関関数の固有ベクトルを使った変換で, 高効率符号化に適しているが, 高速アルゴリズムが存在しないため, 理論的な解析に主に用いられる.

- 以下しばらく, 有限長信号 x を N 次の縦ベクトル $\boldsymbol{x}$ と同一視する. これはエルゴード的な定常確率過程の観測値であるものとする. また, $\boldsymbol{x}$ は実ベクトルであると仮定する.
- 信号 $\boldsymbol{x}$ の自己相関関数は $\boldsymbol{R} = E[\boldsymbol{x}\boldsymbol{x}^T]$ である.
- $\boldsymbol{R}$ は非負対称行列であり, よって直交行列によって対角化される.

- $\boldsymbol{R}$ を対角化する直交行列を $\boldsymbol{Q}$ とし,

$$\boldsymbol{Q}\boldsymbol{R}\boldsymbol{Q}^T = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$$

となっているものとする.

- $\mathrm{diag}(\lambda_1, \ldots, \lambda_N)$ は $\lambda_1, \ldots, \lambda_N$ を対角に持つ対角行列. $\lambda_1 \geq \cdots \geq \lambda_N$ と仮定する.
- $\boldsymbol{y} = \boldsymbol{Q}\boldsymbol{x}$ という変換を考える(これがカーネン・レーベ変換).

- $\boldsymbol{y}$ の自己相関関数は, 以下のように対角行列になる.

$$\begin{aligned} E[\boldsymbol{y}\boldsymbol{y}^T] &= E[\boldsymbol{Q}\boldsymbol{x}\boldsymbol{y}^T\boldsymbol{Q}^T] = \boldsymbol{Q}E[\boldsymbol{x}\boldsymbol{y}^T]\boldsymbol{Q}^T \\ &= \boldsymbol{Q}\boldsymbol{R}\boldsymbol{Q}^T = \mathrm{diag}(\lambda_1, \ldots, \lambda_N) \end{aligned}$$

- 信号 $\boldsymbol{y}$ の各成分に固有値 $\lambda_1, \ldots, \lambda_N$ の大きさに応じた量子化ビット数を割り振ることで高効率の符号化が実現できる.

離散コサイン変換

- **離散コサイン変換 (Discrete Cosine Transform，DCT)** は，余弦関数を用いた，離散フーリエ変換と類似した変換.
- 離散フーリエ変換は, 第 $(k+1, n+1)$ 成分が

$$\exp\left[-j\frac{2\pi}{N}kn\right], \quad 0 \leq k, n \leq N-1$$

となっている行列を使った変換であった.

- DCT では, 指数関数のかわりに余弦関数を使う. これは, コンセプトとしては, 変換したい信号を偶関数に拡張してから離散フーリエ変換を適用することに相当する.
- 指数関数のかわりに余弦関数を用いることができる理由は, 偶関数をフーリエ級数展開するときに正弦関数の項が無視できることと同じで, 偶関数への拡張の部分が本質的.

- 偶関数への拡張のしかたと端点の処理には合計 8 通りの手法があることが知られており, 対応して 8 種類の DCT が存在する.
- 8 種類の中で, 文献に頻出するものは **DCT-I**, **DCT-II**, **DCT-III**, **DCT-IV** と呼ばれるものの 4 種.
- 応用上よく使われるのは, DCT-II と, その逆変換である DCT-III.

- 各種 DCT の定義には, 行列の大きさやベクトルの正規化のしかたにバリエーションがあり, 文献によって記述がばらばらなことが多い.
- 以下の記述は, DCT-I については A. V. Oppenheim and R. W. Schafer による. DCT-II, DCT-III, DCT-IV は教科書と同じである.

- 変換に用いられる行列の第 $(k+1, n+1)$ 成分を $T(k,n)$ と書く $(0 \leq k, n \leq N-1)$.
- $C(k)$ と $\alpha(k)$ を次のように定義する.

$$C(k) = \begin{cases} \frac{1}{\sqrt{2}}, & k = 0 \\ 1, & \text{それ以外} \end{cases},$$

$$\alpha(k) = \begin{cases} \frac{1}{2}, & k = 0 \text{ または } k = N-1 \\ 1, & \text{それ以外} \end{cases}$$

$$\textbf{DCT-I} \quad T(k,n) = 2\alpha(n)\cos\frac{\pi k n}{N-1}$$

$$\textbf{DCT-II} \quad T(k,n) = \sqrt{\frac{2}{N}}C(k)\cos\frac{\left(n+\frac{1}{2}\right)k\pi}{N}$$

$$\textbf{DCT-III} \quad T(k,n) = \sqrt{\frac{2}{N}}C(n)\cos\frac{\left(k+\frac{1}{2}\right)n\pi}{N}$$

$$\textbf{DCT-IV} \quad T(k,n) = \sqrt{\frac{2}{N}}\cos\frac{\left(k+\frac{1}{2}\right)\left(n+\frac{1}{2}\right)\pi}{N}$$

- DCT-II と DCT-III は互いに逆変換.
- DCT-IV の逆変換は DCT-IV に一致する.
- DCT-I の逆変換の行列の第 $(k+1, n+1)$ 成分は以下の通り.

$$\frac{1}{N-1}\alpha(n)\cos\frac{\pi k n}{N-1}$$

- DCT-I に対応する行列の定義を

$$T(k,n) = \sqrt{\frac{2}{N-1}}\alpha(n)\cos\frac{\pi kn}{N-1}$$

に変更すれば, DCT-I の逆変換を DCT-I に一致させることができる (この種の書き換えが, 亜種が多く発生する理由のひとつ).

修正離散コサイン変換

- 無限長あるいは十分に長い信号 x_0 を変換する場合には, これを長さ N のブロックに区切り, ブロックごとに変換することが一般的.

$$
\begin{array}{ll}
\text{ブロック } 0 & (x_0[0], \dots, x_0[N-1]) \\
\text{ブロック } 1 & (x_0[N], \dots, x_0(2N-1)) \\
\text{ブロック } 2 & (x_0[2N], \dots, x_0(3N-1)) \\
 & \cdots
\end{array}
$$

- このとき, ブロックごとに圧縮符号化をおこなうと, 隣接するブロックの端点でデータの不連続性が発生し, 人間には視覚あるいは聴覚上不快な雑音として知覚される.
- これを防ぐためには, 窓関数を利用し, 隣接ブロック間で信号に重複を持たせることが一般的.

- 隣接ブロックの重複分を加味した信号の変換を**重複直交変換 (Lapped Orthological Transform; LOT)** と呼び, **修正離散コサイン変換 (Modified Discrete Cosine Transform; MDCT)** はその一種.
- 修正離散コサイン変換の逆変換を, **逆修正離散コサイン変換 (Inverse Modified Discrete Cosine Transform; IMDCT)** と呼ぶ.

- 修正離散コサイン変換は, 長さ $2N$ の信号から長さ N の信号を生成する．逆修正離散コサイン変換は, 長さ N の信号から長さ $2N$ の信号を生成する.
- 変換の式は次のページの通り.

- MDCT: $0 \le k \le 2N-1$ に対し,

$$X(k) = \sum_{n=0}^{2N-1} x(n) \cos \frac{(2n+N+1)(2k+1)\pi}{4N}$$

- IMDCT: $0 \le k \le N-1$ に対し,

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \cos \frac{(2n+N+1)(2k+1)\pi}{4N}$$

正弦的ユニタリ変換

- 離散コサイン変換における余弦関数を正弦関数で置き換えると, 離散サイン変換 (Discrete Sine Transform, DST) と呼ばれる変換が定義できる. この変換は信号の直流成分を表現することができないので, これを使うためには, 信号の直流成分を除去する前処理が必要になる.

- 離散サイン変換には離散コサイン変換と同様に 8 種類のバリエーションがあるが, 詳細は略す (文献参照).
- 離散フーリエ変換, 離散コサイン変換および離散サイン変換は, 正弦的ユニタリ変換と呼ばれる変換の一種だが, 正弦的ユニタリ変換の一般論が必要になる機会は稀であると思われるので, この講義では名前の紹介に留める.

ウォルシュ・アダマール変換

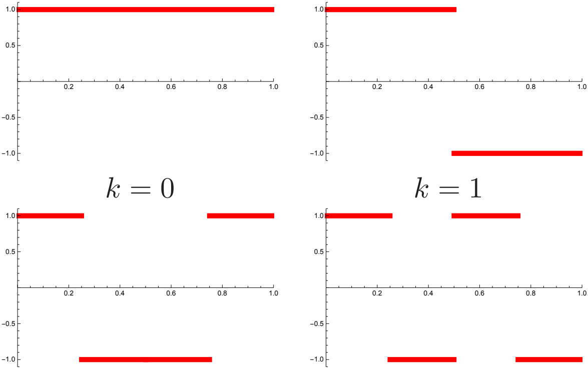
- ウォルシュ･アダマール変換は, 値が -1 か 1 の 2 値を取るウォルシュ関数と呼ばれる関数を使った変換. 高速演算が可能で, 用途は限定されるが, 心電図の処理や Code Division Multiple Access(CDMA) 通信などに使われる.

- ウォルシュ関数は次の式によって帰納的に定義される関数.

$$W(0,t/T) = \begin{cases} 1, & 0 \le t < T, \\ 0, & \text{それ以外} \end{cases}$$

$$W(2k+q) = W(k,2t/T) + (-1)^{k+q} W(k,2t/T-1),$$
$$k \in \mathbb{N}, q = 0,1$$

$T = 1$ としたときのウォルシュ関数の様子



$k = 0$　$k = 1$

$k = 2$　$k = 3$

- ウォルシュ·アダマール変換は, 第 $(k+1, n+1)$ 成分が

$$W[k, n/N], \quad 0 \leq k, n \leq N-1$$

となっている行列を使った変換.

- k が 2 のべきのとき, この行列はアダマール行列と呼ばれる行列に対応するため, ウォルシュ・アダマール変換と呼ばれる.

その他の変換

- これ以外にハール変換, ハートレー変換, スラント変換, M 変換, ウェーブレット変換など様々な変換があるが, この講義では名前を挙げるのみに留める.
- ウェーブレット変換は教科書では画像処理のみについて紹介されているが, 音声信号にも適用されるので注意.

MPEG オーディオ

- 教科書で述べられている MPEG オーディオの解説は MPEG-1 Audio Layer III(MP3) なのだが …
- 今日では AAC(MPEG-2 Advanced Audio Coding あるいは MPEG-4 Advanced Audio Coding) 形式が MP3 形式に徐々に置き換わりつつある.

- MPEG4-AAC は MPEG2-AAC の拡張版.
- MP3 と AAC の処理の内容には共通点が多いが, 相異点もある.
- MP3 の符号化の手順は基本的にはフィルタバンク及び MDCT と心理聴覚分析の組み合わせ. これらの後で量子化をおこなう. 復号には IMDCT とフィルタバンクを用いる.

- AAC の符号化ではフィルタバンクを用いず直接 MDCT を適用する．また, MDCT 係数に対して線形予測を適用する.
- MP3, AAC の双方で, 心理聴覚分析によりデータを削減する.

心理聴覚分析

- 人間の耳に聞こえる音の大きさの下限は周波数によって異なり (教科書図 13.6a), 可聴域を下回る周波数成分は削除できる.
- 人間の聴覚ではマスキング現象と呼ばれる現象が発生する． これには, 周波数マスキングと経時マスキングの 2 種類がある.

- 周波数マスキングとは, 耳がある周波数の音に反応しているとき, それに近い高周波側の領域において耳の感度が低下する現象をさす. 実用上は, 低周波の音の成分によって, それより少し高い音の成分が聞こえなくなるということが多い. 聞こえる (抑制する) 方の音をマスカ (masker), 聞こえなくなる (抑制される) 方の音をマスキ (maskee) と呼ぶ. マスカとマスキが同時刻に存在するときに発生する.

- 経時マスキングは, 耳がある時刻で大きな音に反応したとき, 時間的に．その直前および直後で耳の感度が低下する現象をさす．大きな音の直後の音が知覚されにくくなる順方向マスキングと大きな音の直前の音が知覚されにくくなる逆方向マスキングがある．

ディジタルオーディオのフォーマット

- 今日流通しているディジタルオーディオのフォーマットには,
  - ▷ 無圧縮: WAV, AIFF
  - ▷ 可逆圧縮: FLAC, ALAC
  - ▷ 非可逆圧縮: MQA, MP3, AAC, Ogg Vorbis, WMA

  があり, 今後どれが主流になるかは不明. DSD という PCM の系統とはまったく異なる形式による音楽配信もおこなわれている.

窓関数 (第 5 回の補遺)

- 信号 x の長さが無限長あるいは極めて長い場合を考える.
- コンピュータの能力の関係で, 信号全体を一挙に変換することは必ずしも現実的ではないので, 信号を短い連続した時間領域で区切り. 短い有限長信号を処理することが多い.

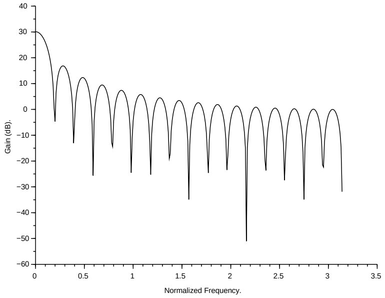
- 信号 $x = (x[n])_{n\in\mathbb{Z}}$ から長さ N の信号を切り出すことを考える.
- 信号 x は l_1 に属し, 離散時間フーリエ変換可能であると仮定する.
- 信号の切り出しについてはどの部分を考えても議論は変わらないので, 簡単のため, $(x[0], \dots, x[N-1])$ を切り出す状況を考える.

- 信号 x の周波数特性と $(x[0],\dots,x[N-1])$(これを一時的に $x^{(N)}$ と書くことにする) の周波数特性の関係を知りたい (これらの相異が大きいと, 信号の切り出しによって信号に歪みが発生することになる).
- $x^{(N)}$ は, 信号 x と, 次の信号 w_{sq} との積.

$$w_{\mathrm{sq}}[n] = \begin{cases} 1, & 0 \le n \le N-1, \\ 0, & \text{それ以外} \end{cases}$$

- w_{sq} を方形窓あるいは矩形窓と呼ぶ.
- 周波数領域では, 原信号の周波数特性と窓関数の周波数特性の畳み込みが実行されることになる. $N = 32$ とした場合に, 矩形窓の周波数特性を次ページに示す.

矩形窓の周波数特性



40
30
20
10
0
−10
−20
−30
−40
−50
−60
Gain (dB).
0
0.5
1
1.5
2
2.5
3
3.5
Normalized Frequency.

- 図からわかるように, 周波数軸の原点以外に, 複数のピークがあらわれることがある.
- 周波数軸の原点を中心とする窓関数のピークをメインローブと呼ぶ.
- メインローブ以外の周波数応答のピークをサイドローブと呼ぶ.

- 窓関数を使うときには, なるべく原信号を残したいという要求がある一方で, 周波数特性の歪みを減らすという観点からは, メインローブの幅が広く, サイドローブが速やかに減衰することが望ましい. 矩形窓は, 原信号に忠実という観点からは好ましいが, 周波数特性の歪みという観点からは好ましいとは言えない.

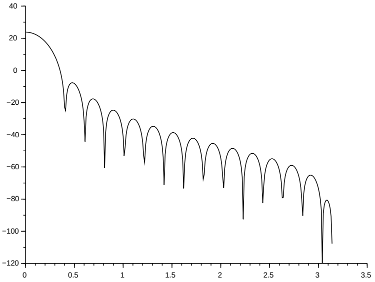
- このため, 時間波形の歪みを許容した上で, 周波数特性の歪みを減らすための窓関数が複数提案されている．この講義では, 代表的なもののみ紹介する．

- ハニング窓は, 信号から長さ N のフレームを切り出すために,

$$w[n] = 0.5 - 0.5\cos(2\pi n/(N-1))$$

という関数を使う窓. J. von Hann による. よく使われる.

ハニング窓の周波数特性

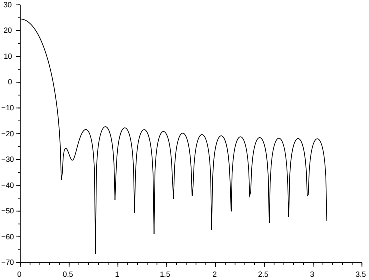


- ハミング窓は, 信号から長さ N のフレームを切り出すために,

$$w[n] = 0.54 - 0.46\cos(2\pi n/(N-1))$$

という関数を使う窓. R. W. Hamming による. これもよく使われる.

ハミング窓の周波数特性



- ハミング窓とハニング窓は一般化ハミング窓と呼ばれるものの特別な場合になっている. 一般化ハミング窓は, 次のように定義される.

$$w[n] = \alpha - (1-\alpha)\cos(2\pi n/(N-1))$$

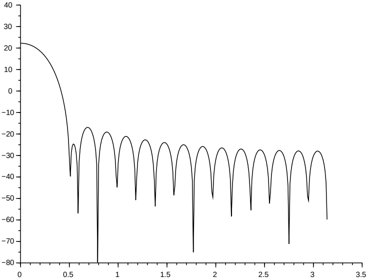
$\alpha = 0.5$ としたものがハニング窓, $\alpha = 0.54$ としたものがハミング窓である.

- ブラックマン窓は, 信号から長さ N のフレームを切り出すために,

$$w[n] = 0.42 - 0.54\cos(2\pi n/(N-1)) + 0.08\cos(4\pi n/(N-1))$$

という関数を使う窓. R. B Blackmann による.

ブラックマン窓の周波数特性



- これらの窓関数が提案されたのは 19 世紀から 20 世紀で, 古くから知られた技法.
- 非因果的な処理 (時間軸 $[-N, N]$ の範囲で演算をおこなう) をおこなうこともあるが, この講義では因果的な場合を紹介した.
- 窓関数は非定常な信号の解析にも用いられる.

- 上記以外では, 正規分布の関数を利用するガウス窓, 三角形の波形を利用する三角窓あるいはバートレット窓, 第一種ベッセル関数を利用するカイザー窓などがあるが (MATLAB は 17 種類の窓関数をサポートしている), この講義では網羅的な紹介はおこなわない.

https://jp.mathworks.com/help/signal/windows.html

- 窓関数を用いて信号解析をおこなうときには, 矩形窓以外を利用する場合, 信号を長さ N のブロック (フレーム) に重複なく切り分けてから窓関数を適用するのではなく, 分割するブロック (フレーム) に重複を持たせることが普通.
- 先に述べた MDCT でも, そのようになっている.

今回の参考文献

電子情報通信学会 知識ベース

ディジタル信号処理ハンドブック

小野, 渡辺, 国際標準画像符号化の基礎技術, コロナ社, 1998

A. V. Oppenheim and R. W. Schafer, Discrete-time signal processing, 3/e, Pearson, 2010.

尾知 (監修), ディジタル音声&画像の圧縮/伸張/加工技術, CQ 出版, 2013.

樋口, 川又, MATLAB 対応ディジタル信号処理, 昭晃堂, 2000.

府川, ディジタル信号処理, 培風館, 2009

荻原, ディジタル信号処理, 第 2 版, 森北出版, 2014

橋本, Scilab で学ぶ統計・スペクトル解析と同定, オーム社, 2008.

https://jp.mathworks.com/

https://www.vtv.co.jp/intro/glossary/mpeg4aac.html

https://www.onosokki.co.jp/HP-WK/nakaniwa/rekishi/keisoku.htm

https://av.watch.impress.co.jp/docs/topic/1068831.html