音情報科学　第3部

音声の情報処理

伊藤　彰則
東北大学大学院工学研究科
aito@fw.ipsj.or.jp

スライドのダウンロード

http://www.spcom.ecei.tohoku.ac.jp/~aito/soundmedia/

講義内容

- 第１回：音声の生成と符号化その１
  - 音声の生成と特徴
  - 基本的な符号化方式：PCM,DPCM,ADPCM
- 第２回：音声の符号化その２
  - 音声の線形予測：予測係数，PARCOR係数，LSP係数
  - CELP符号化
  - オーディオ符号化
- 第３回：音声強調
  - スペクトル減算
  - マイクロホンアレイによる方法

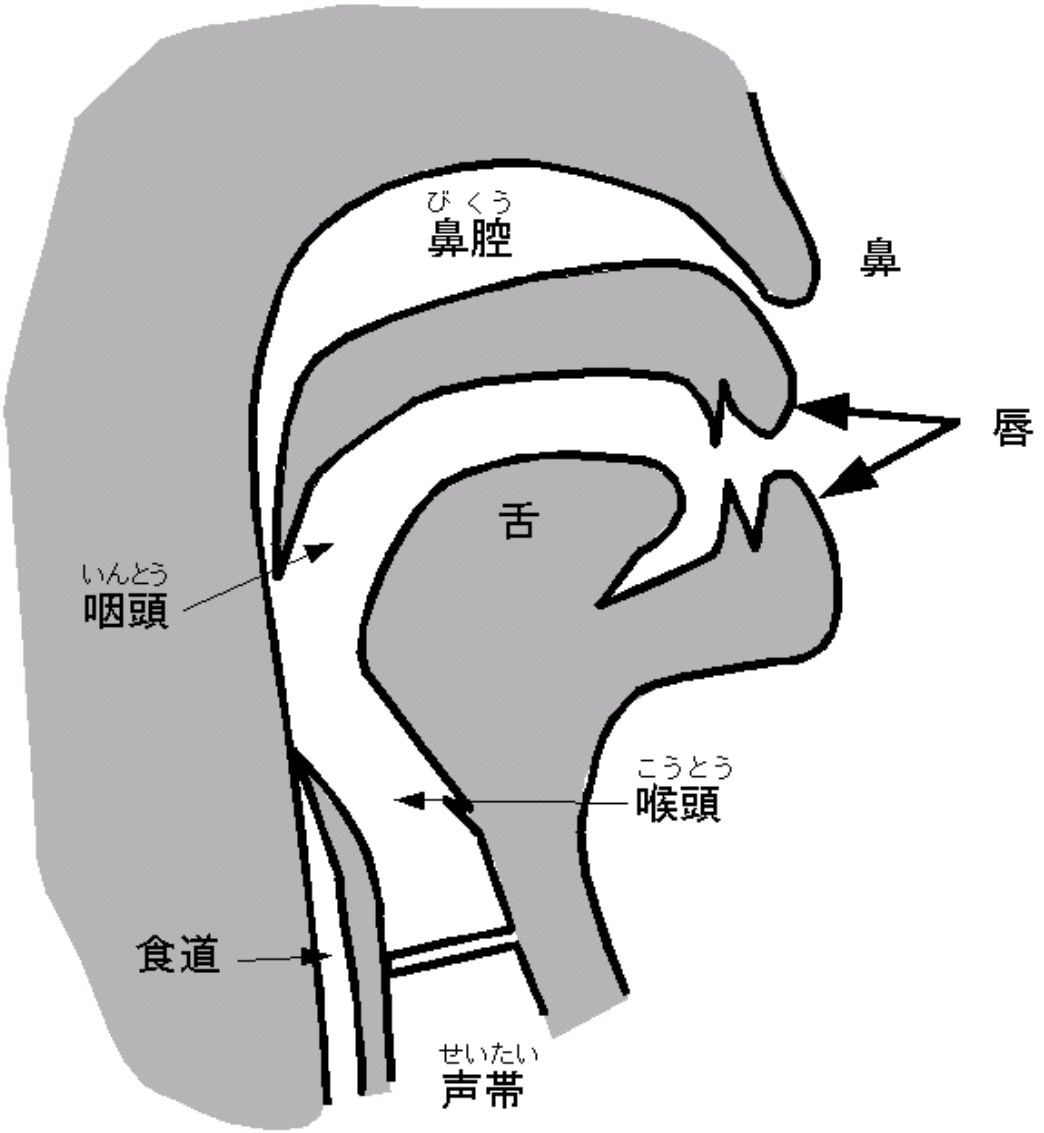
講義内容

- 第4回：音声の認識と合成
  - 音声認識の方法
  - 音声合成の方法
- 第5回：音と音楽
  - 楽器の音の特徴
  - 音楽の情報処理

音声の生成

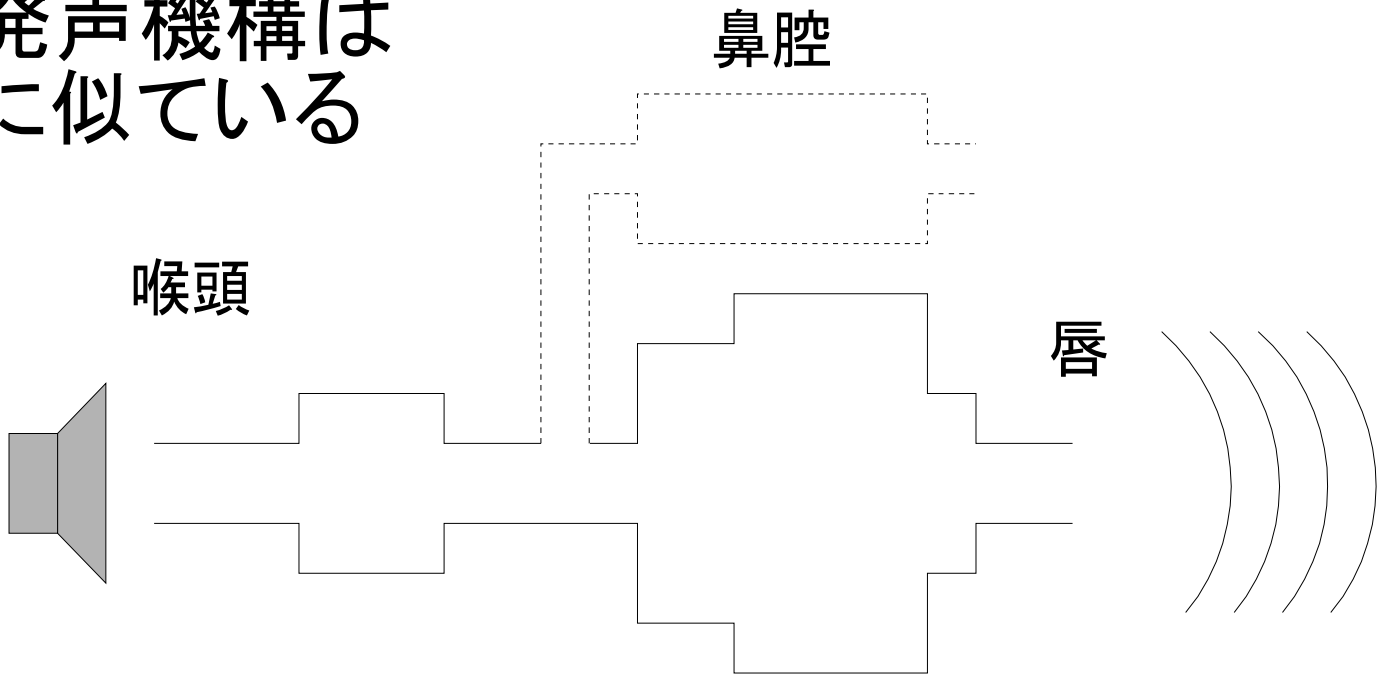
- 声を生成する器官
  - 声帯
  - 喉頭
  - 咽頭
  - 舌
  - 歯茎
  - 歯
  - 口唇
  - 鼻腔

声道

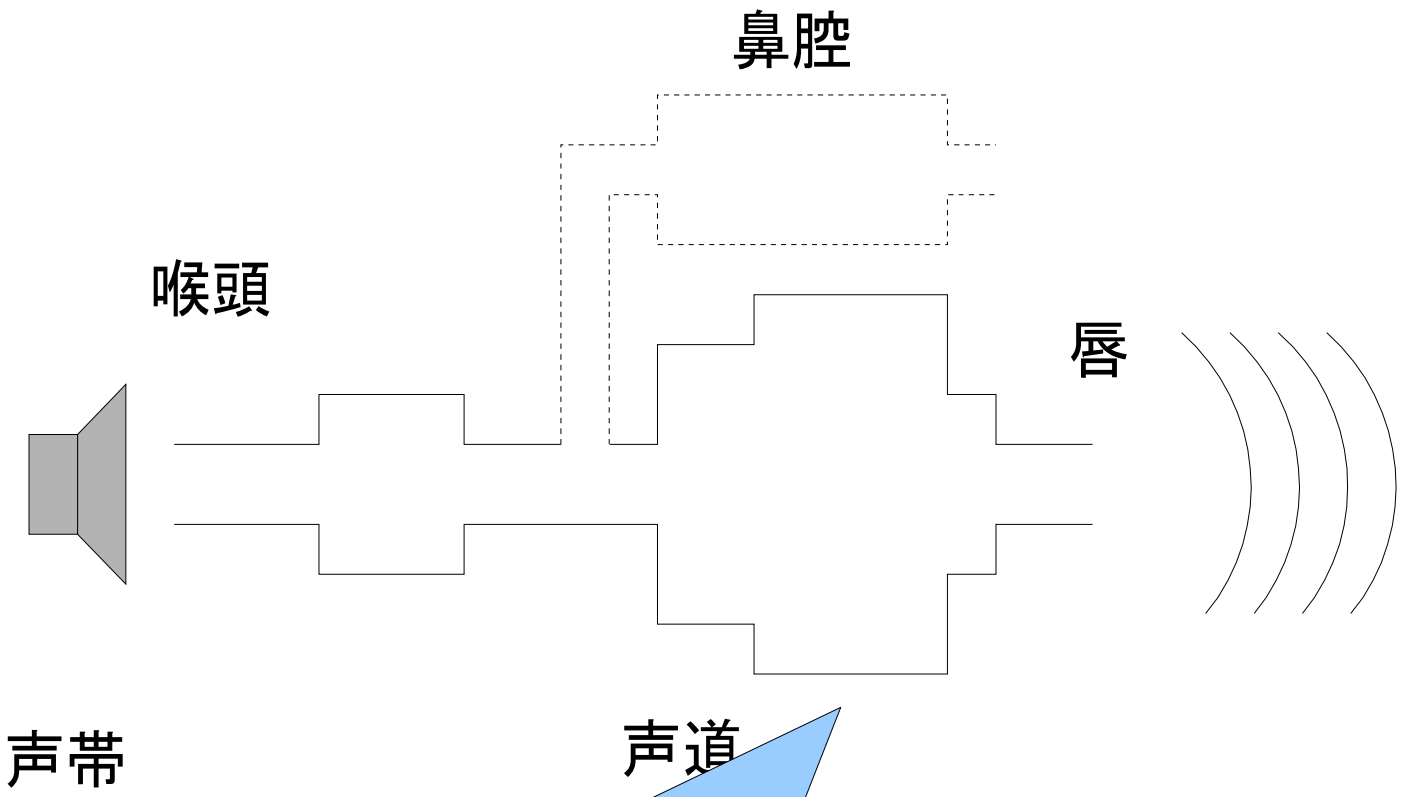


音響管モデル

- 人間の発声機構は管楽器に似ている

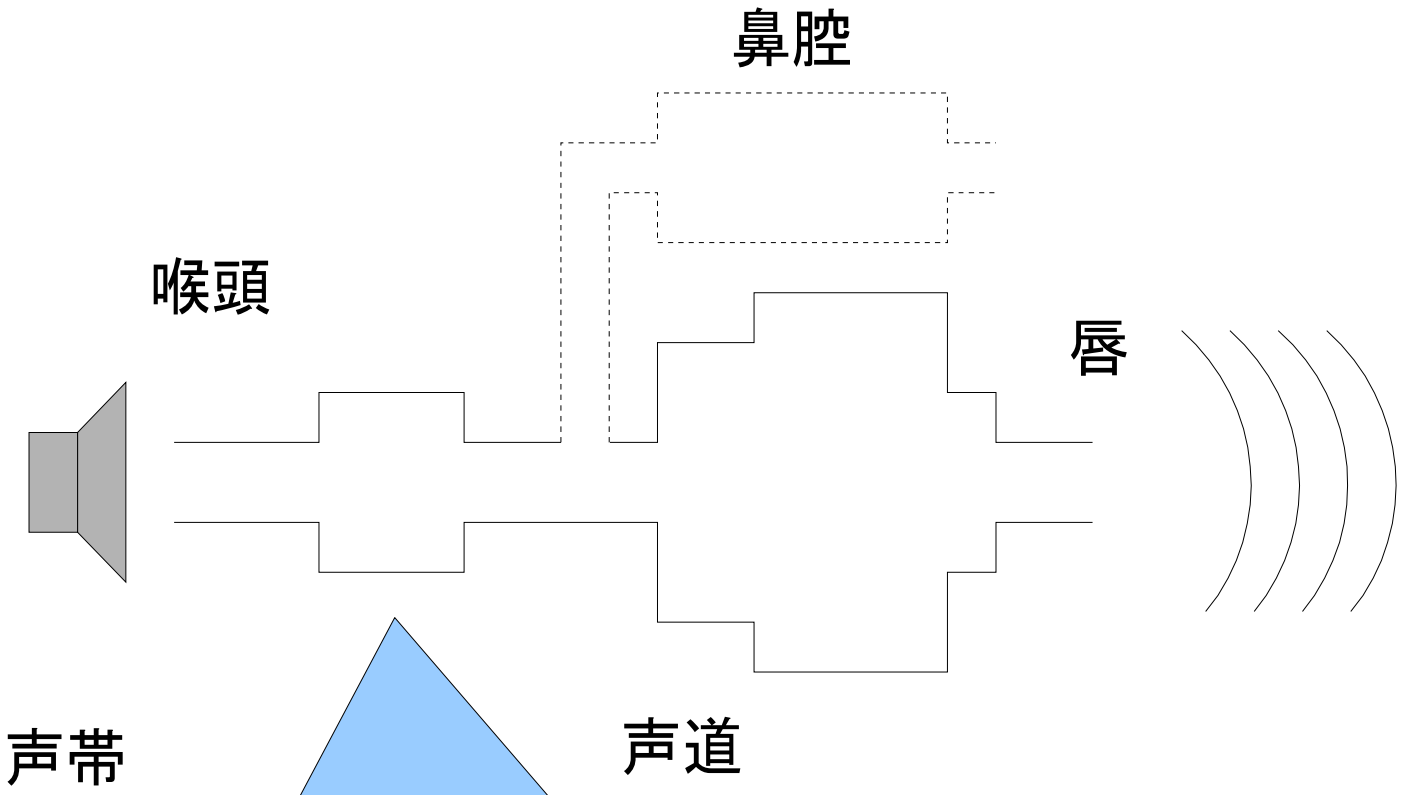


音韻性と個人性



この辺の形は
自分で制御できる

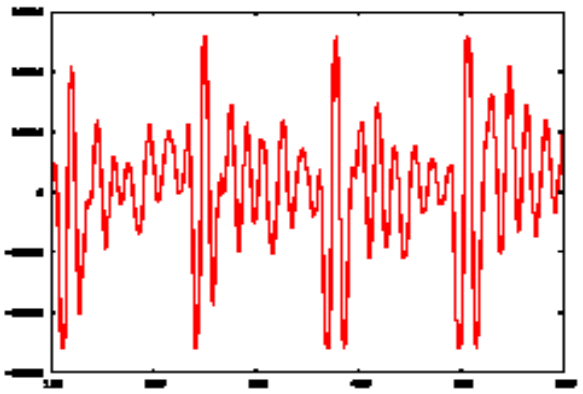
音韻性と個人性



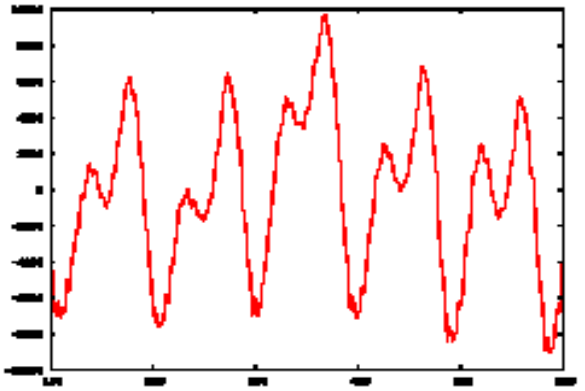
この辺の形，全体の長さ，平均的な太さなどは自分で制御できない

音声の波形

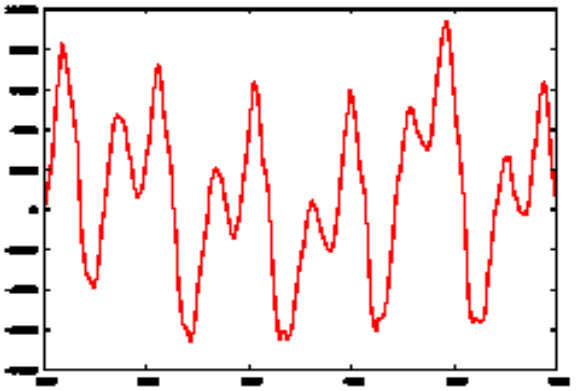
- 結構複雑です



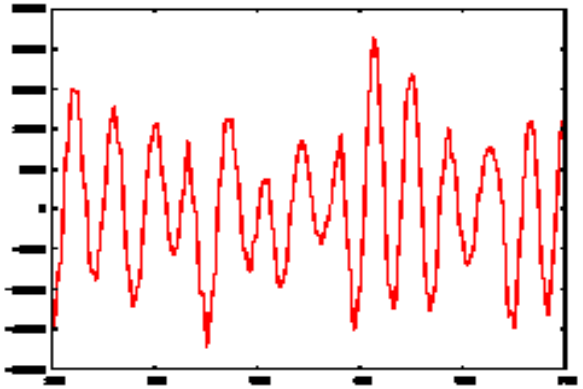
/a/



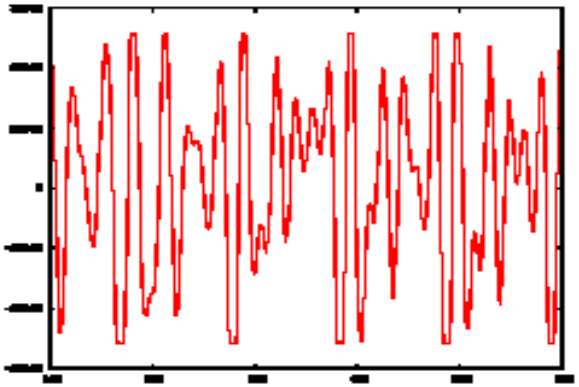
/i/



/u/



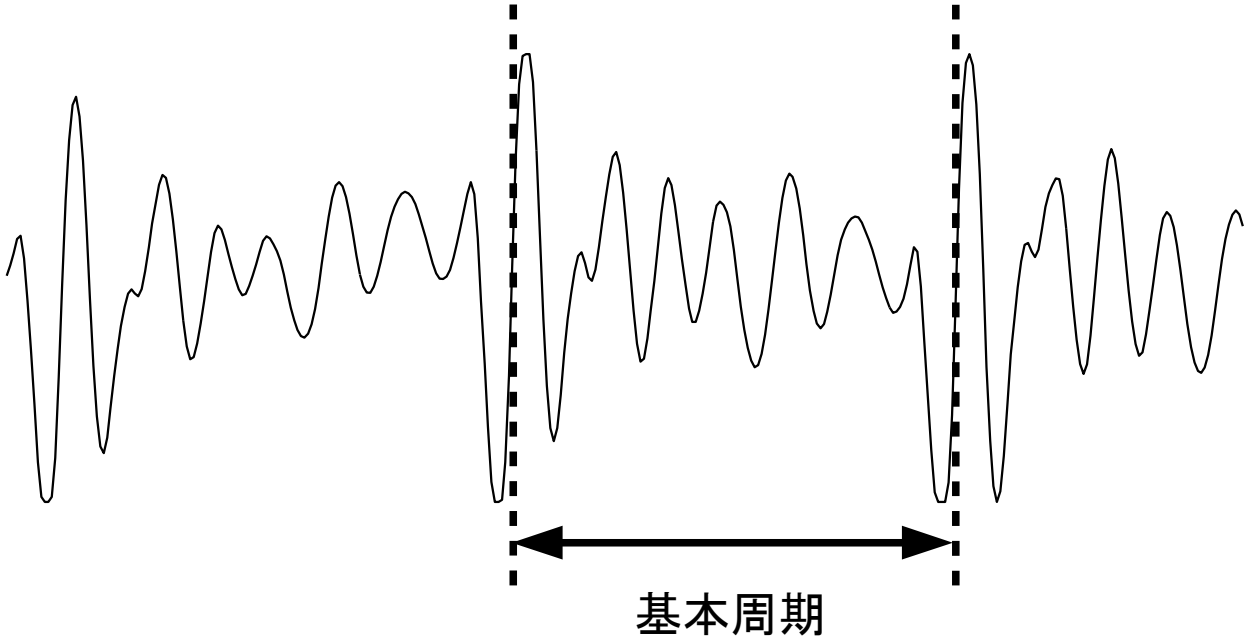
/e/



/o/

音声の波形

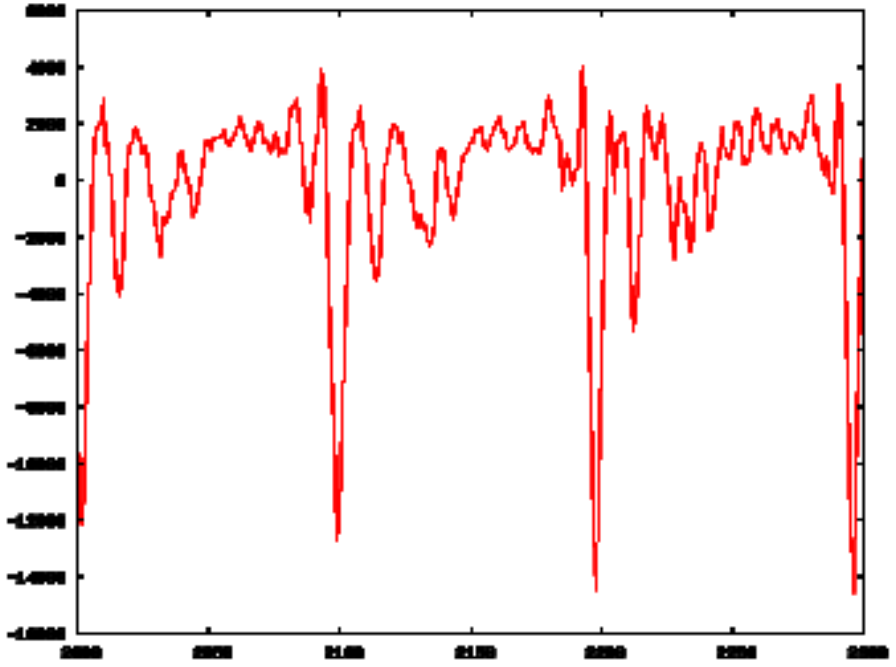
- 複雑だけどおおむね周期的

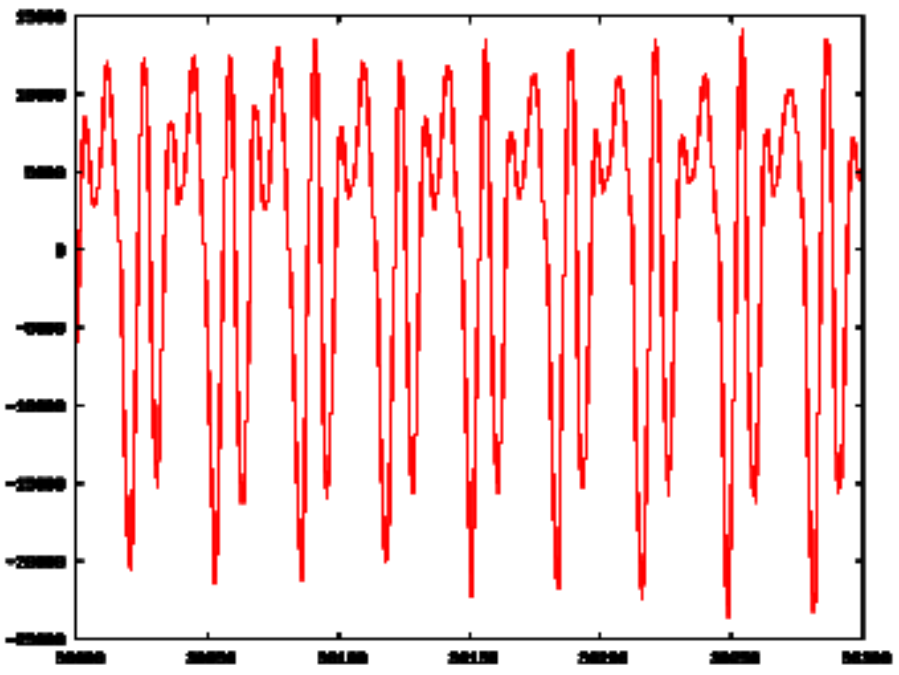


基本周期　　T [s]
基本周波数　　F_0 [Hz] $= 1/T$

いろいろな「あ」

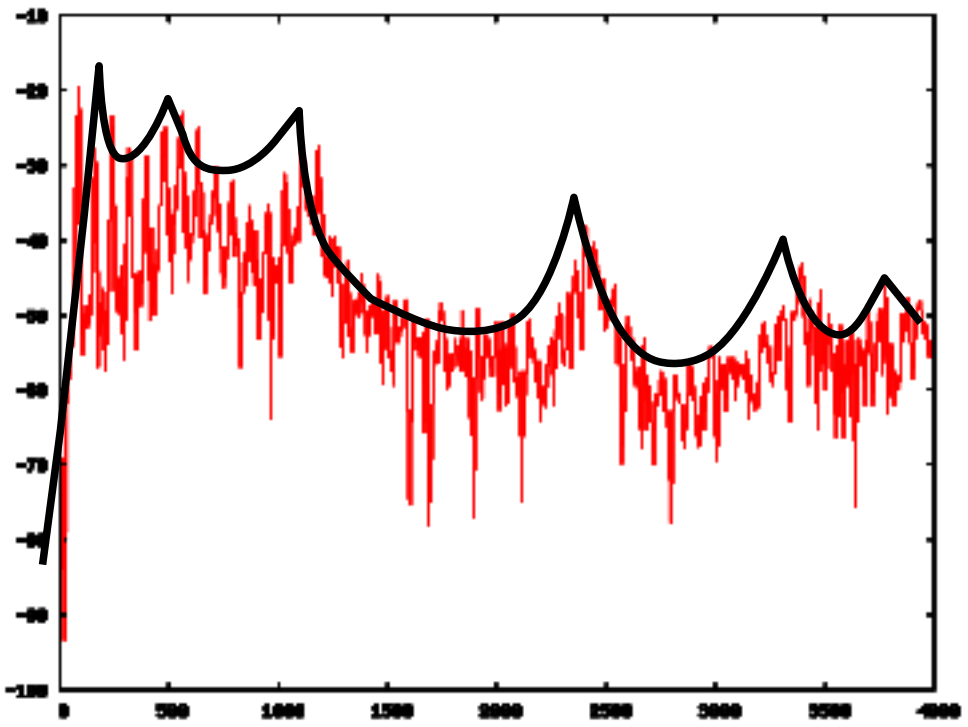
- 基本周波数の違う2つの/a/
  - 音韻としては同じ：声道の形が同じ（と思われる）
  - 波形はまったく異なる
  - 物理量の何が同じなのか？

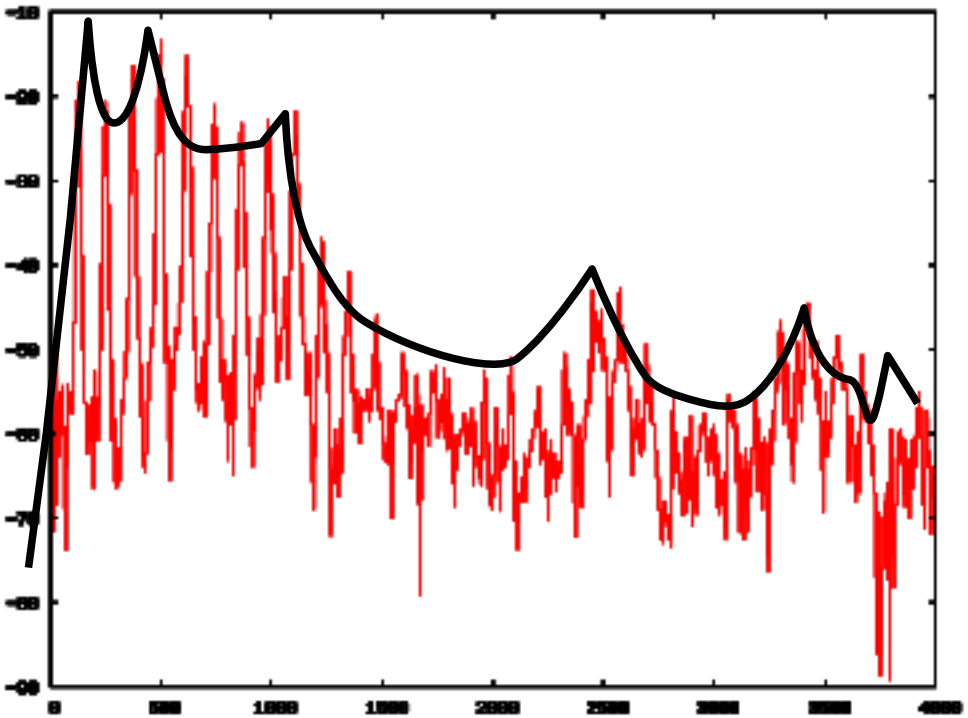




音声のスペクトル

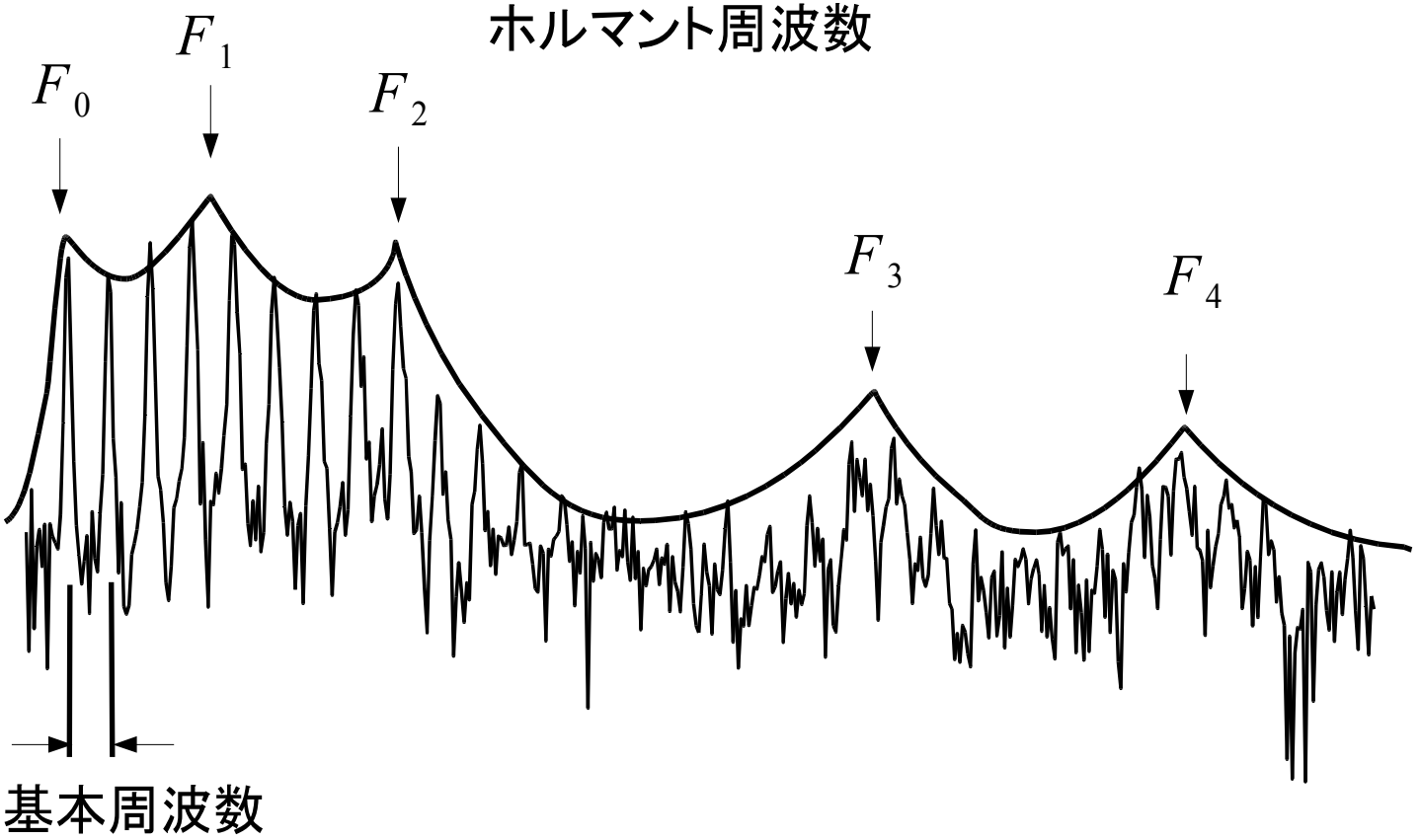
- ２つの「あ」のスペクトル
  - 大まかな形が似ている→声道形状
  - 細かいギザギザは異なる→声帯音源波の周波数





音声のスペクトルとホルマント周波数

- F_0: 基本周波数
- F_1,F_2,..: ホルマント(formant)周波数

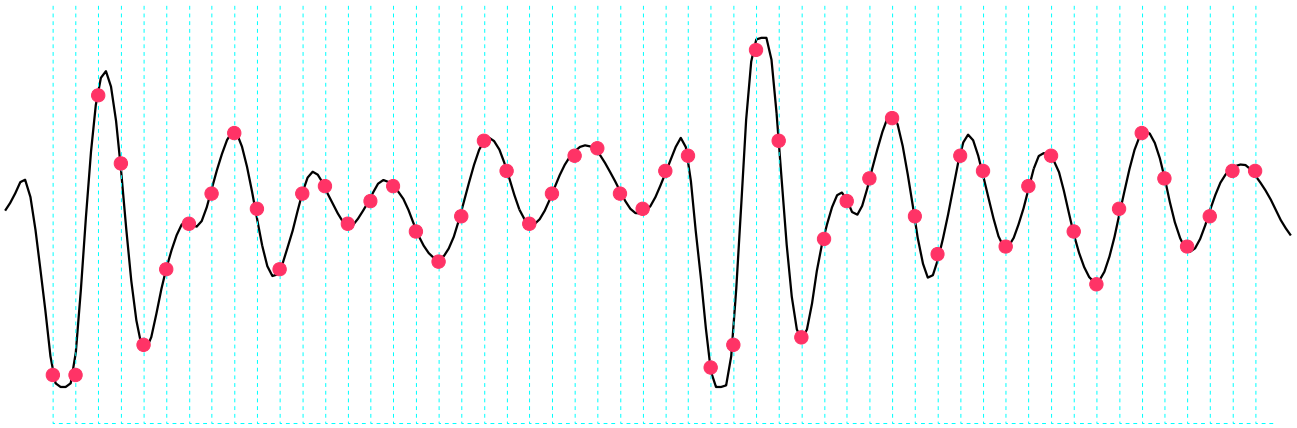


音声の符号化

- 音（アナログ量）→デジタルデータに変換
  - コンピュータで扱える
  - デジタル回線での伝送
- どうやってデジタル化するか？
  - 目標
    - アナログに戻したときの音質がよい方がいい
    - デジタルデータにしたときのビット数が少ない方がいい
  - 方法
    - 音声のいろいろな性質を利用する

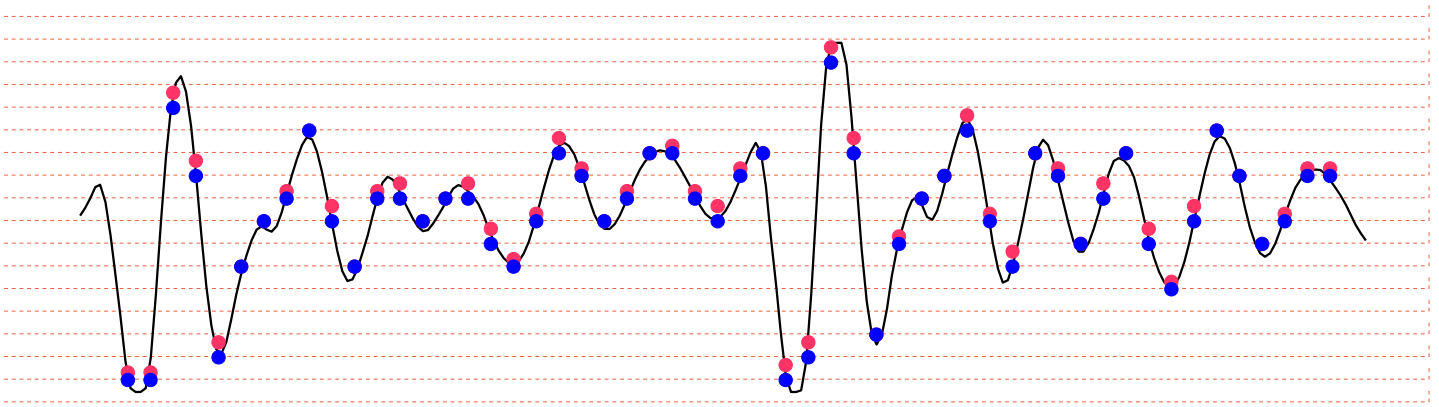
音声符号化の基本事項

- サンプリング
  - 時間的に連続する信号の「とびとび」の時間での値だけを観測する
  - 「とびとび」の速さ：サンプリング周波数 f_s
  - 入力信号が$f_s/2$以下の周波数成分だけを含む場合には，サンプリングされたデータから元の信号が復元できる（サンプリング定理）



音声符号化の基本事項

- 量子化
  - 信号の大きさを「とびとび」の値に丸める
    - 整数で値を表現できる
  - 「とびとび」の幅：量子化幅
  - 量子化した信号と元の信号の差：量子化誤差



サンプリングと量子化:どうする？

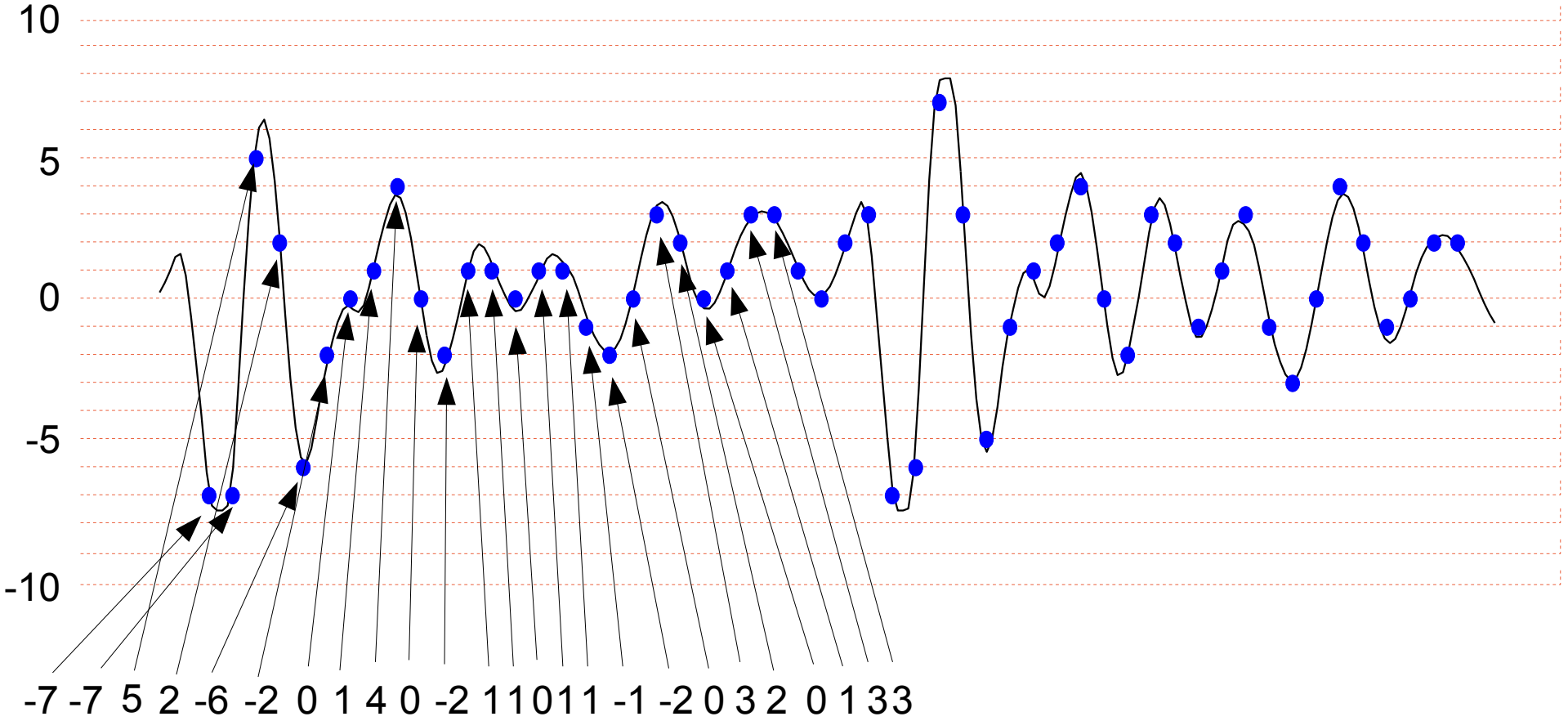
- サンプリング周波数:どの程度の高さの音まで再現するかで決まる
  - 電話:8kHz (4kHzの音まで再現)
  - 高品位音声:16kHz (8kHzの音まで再現)
  - CD:44.1kHz (22.05kHzの音まで再現)
- 量子化:どの程度の雑音まで許容するかで決まる
  - 音の符号化とはつまり「どう量子化するか」

PCM符号化

- PCM(Pulse Code Modulation)
  - 量子化した値をそのまま2進数の数字として符号化
- PCMの要素
  - 1サンプルあたり何ビット使うか
  - 量子化の間隔をどうするか
    - 等間隔に量子化：線形量子化(linear quantization)
    - 等しくない間隔で量子化：非線形量子化(nonlinear quant.)
- PCM符号の例
  - CD:16bit線形量子化
  - VoIP(G.711): 8bit 非線形量子化

PCM線形量子化

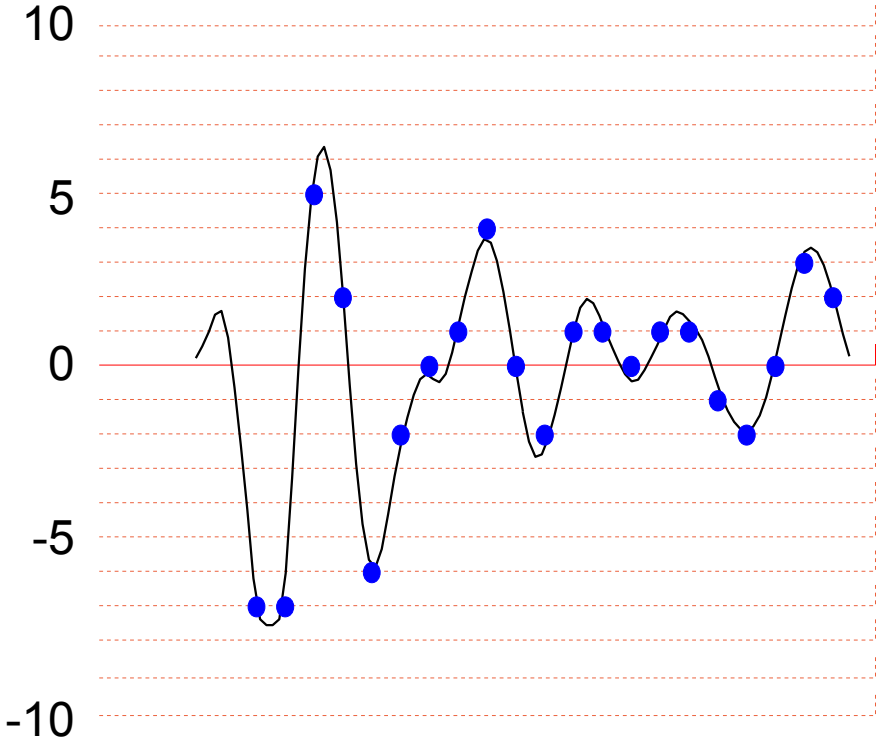
- 特に面倒なことはありません

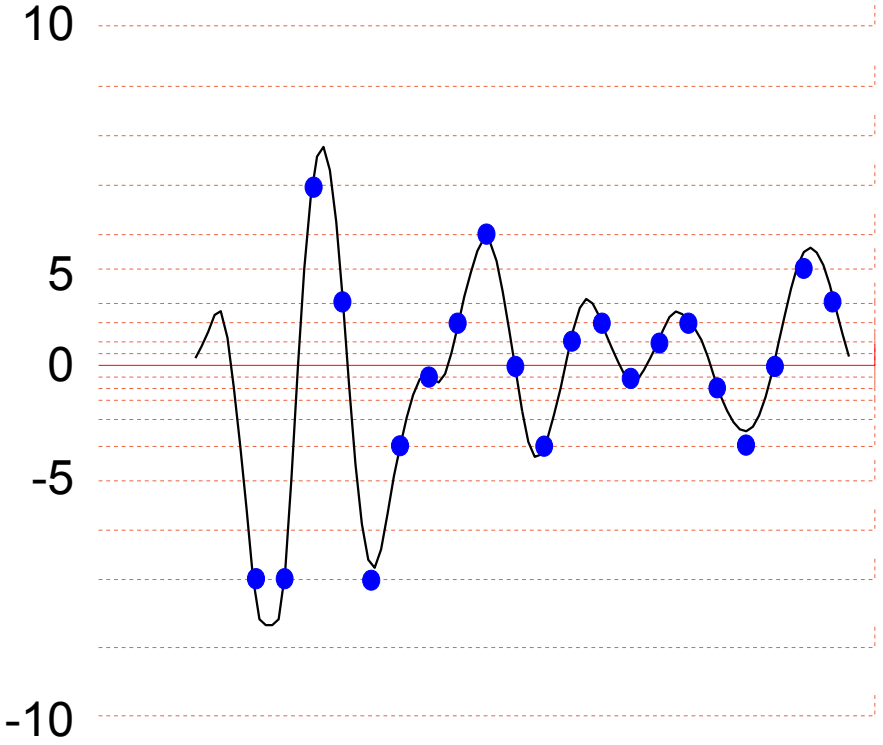


CDでは16bit(-32768～+32767)で量子化

非線形量子化

- 多くのサンプルは0に近い値を持つ
  →0付近を細かく量子化すれば全体の誤差が減らせる

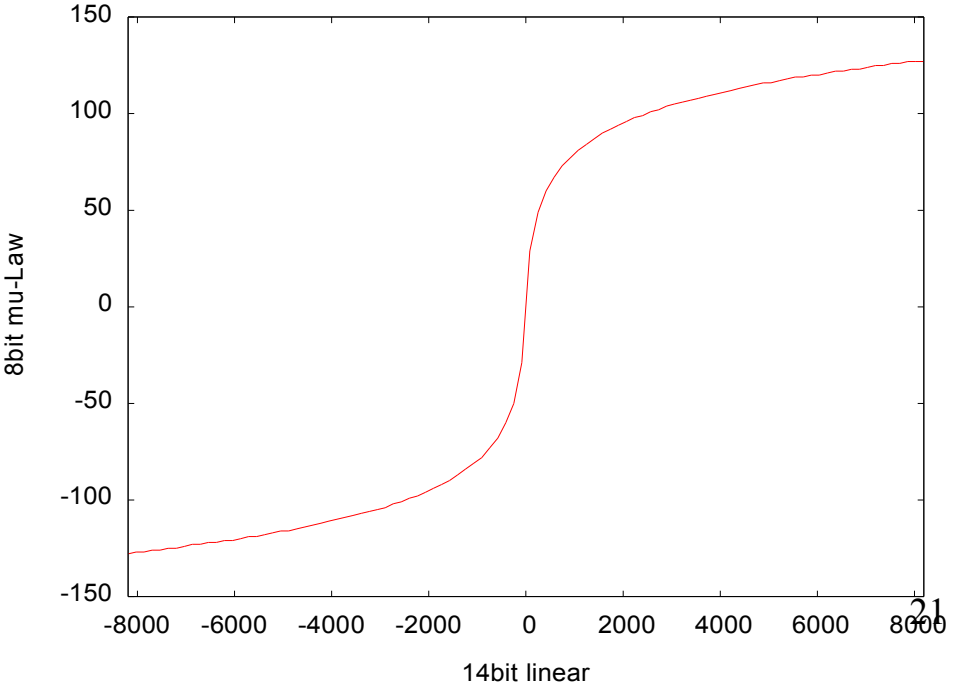




非線形量子化の例：G.711

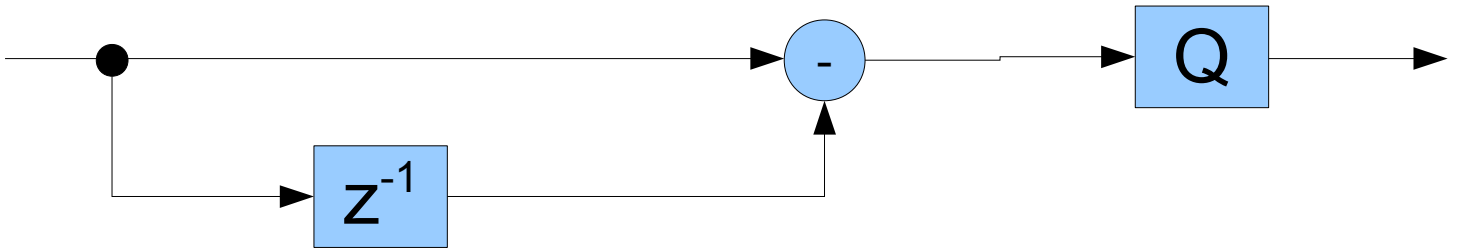
- 64kbit/sのデジタル回線での音声通信用
  - 8kHzサンプリング，8bit非線形量子化
  - μ-Law（日本，アメリカ），A-Law（ヨーロッパ）
  - μ-Law: 14bit線形量子化の値→8bit非線形量子化の値

$$Y = 128\, sign(X) \frac{\log\left(1 + \frac{255|X|}{8192}\right)}{\log 256}$$



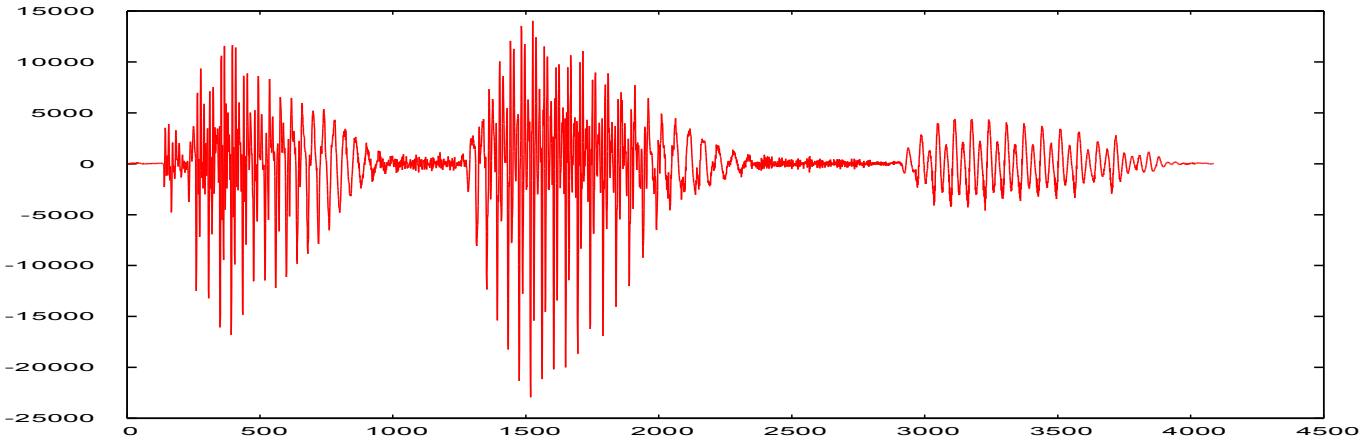
差分PCM (DPCM)

- 通常の音声信号では隣り合うサンプルの数値はそれほど大きく違わない
→差分だけを量子化して送ればビット数がケチれる

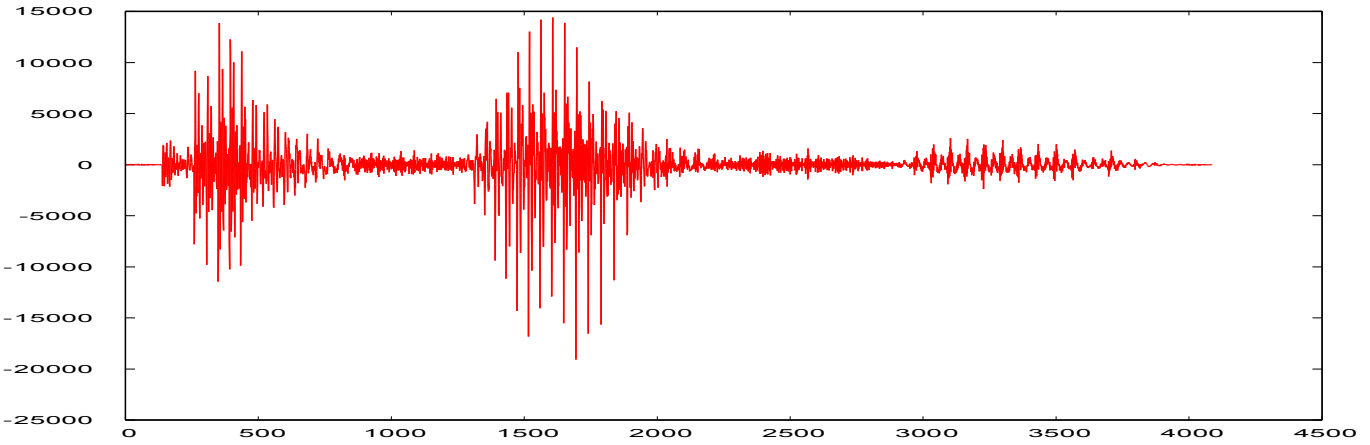


差分PCM(DPCM)

- 元の波形



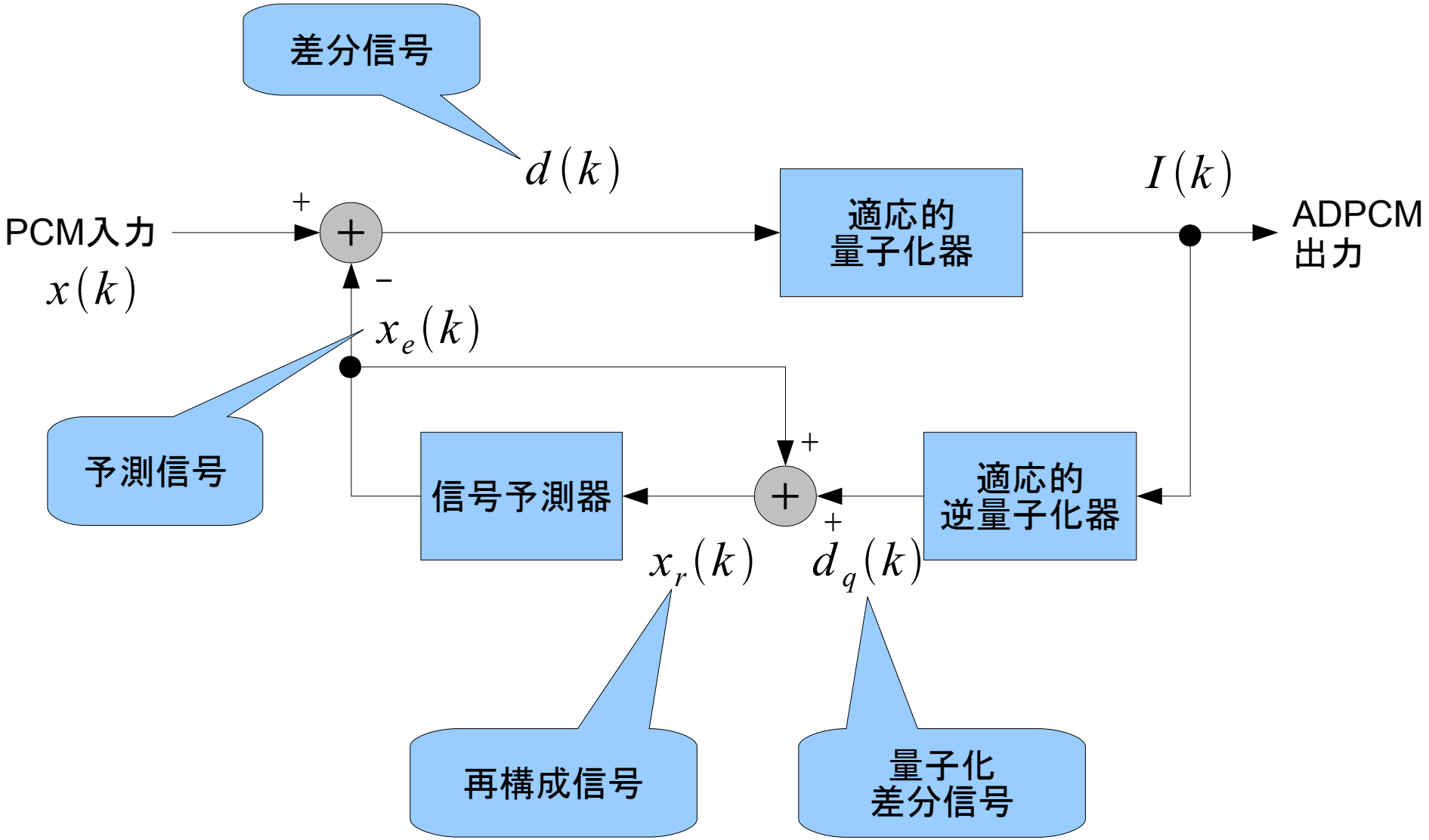
- 差分の波形



適応差分PCM(ADPCM)

- DPCMの効率をさらに上げるため
  - 連続するサンプル間の単純な差ではなく，フィルタによる予測値を使う
  - 量子化の幅を適応的に変化させる
    - 変化が大きい（すなわち，音が大きい）時点では，次の変化も大きい可能性が高い
    - 変化が小さい（すなわち，音が小さい）時点では，次の変化も小さい可能性が高い

ADPCMのブロック図



ADPCMの計算アルゴリズム

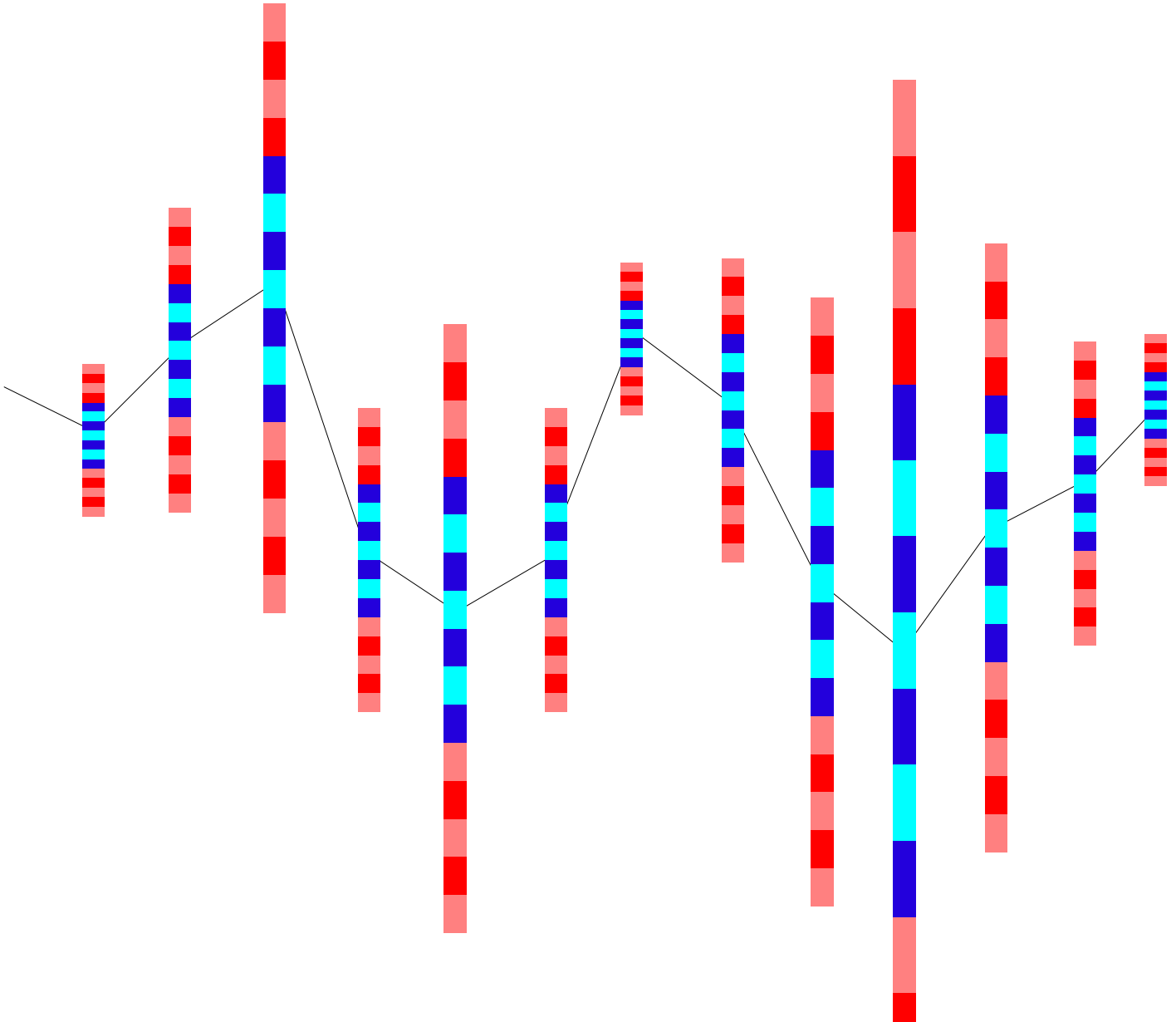
1. 予測信号を計算 $x_e(k)$
2. 差分信号を計算 $d(k)=x_e(k)-x(k)$
3. 量子化する(ADPCM出力) $I(k)=Q\left(d(k)\right)$
4. 逆量子化する $d_q(k)=Q^{-1}\left(I(k)\right)$
5. 再構成信号を計算 $x_r(k)=x_e(k)+d_q(k)$
6. 次の予測信号を計算 $x_e(k+1)=pred\left(x_r(k),d_q(k),\ldots\right)$

音声の予測

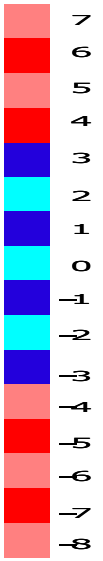
- ADPCMでは音声の予測値と実測値との差分を量子化する
- 音声をどう予測するか
  - DPCMの場合　$x_e(k)=x_r(k-1)$
  - もう少し賢く　$x_e(k)=2x_r(k-1)-x_r(k-2)$
  - G.726

$$x_e(k)=\sum_{i=1}^{2} a_i x_r(k-i)+\sum_{i=1}^{6} b_i d_q(k-i)$$

適応的に量子化幅を決める(例)



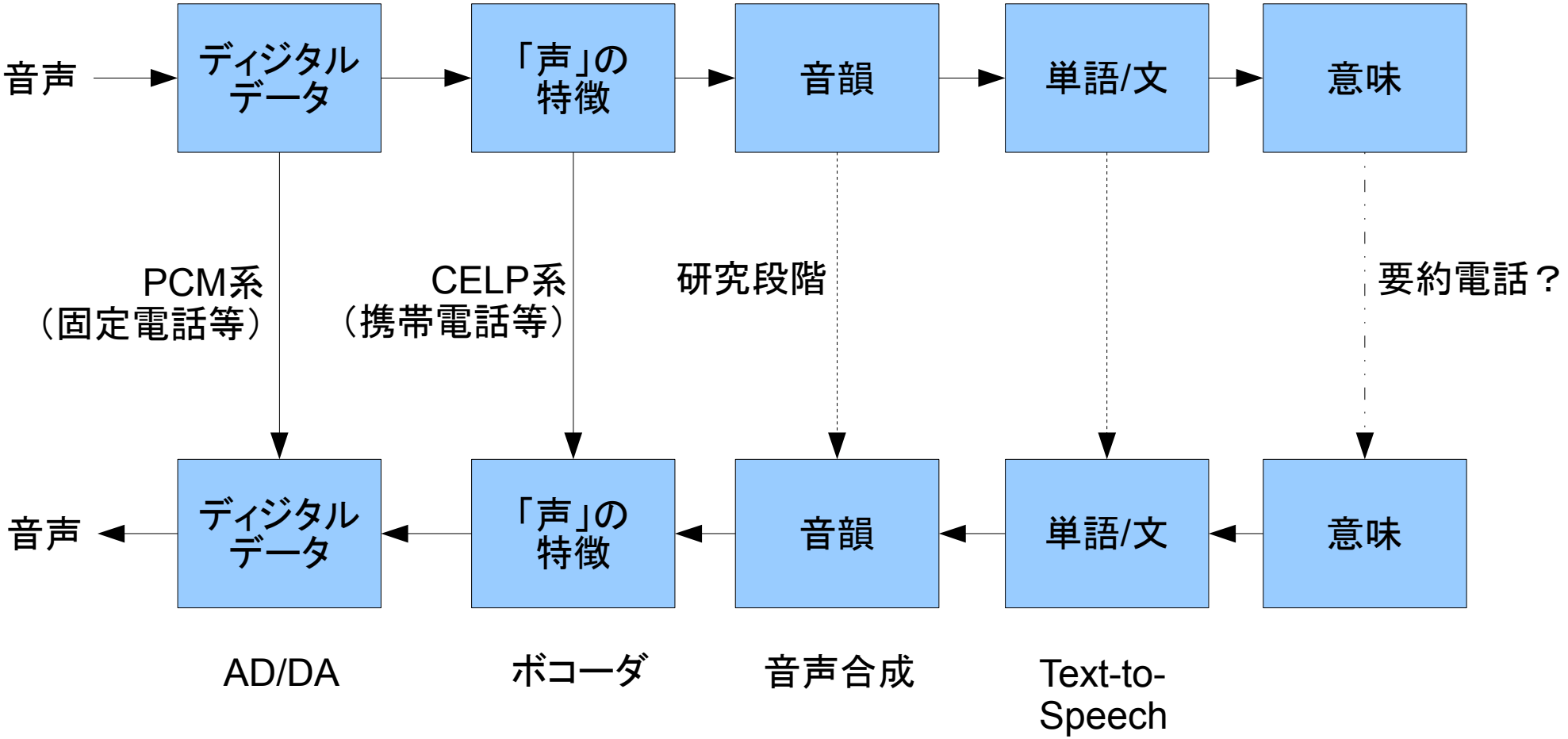
- 前との差分を物差しで計る.
- 値が青なら次の物差しを半分にする.
- 値が赤なら次の物差しを倍にする.



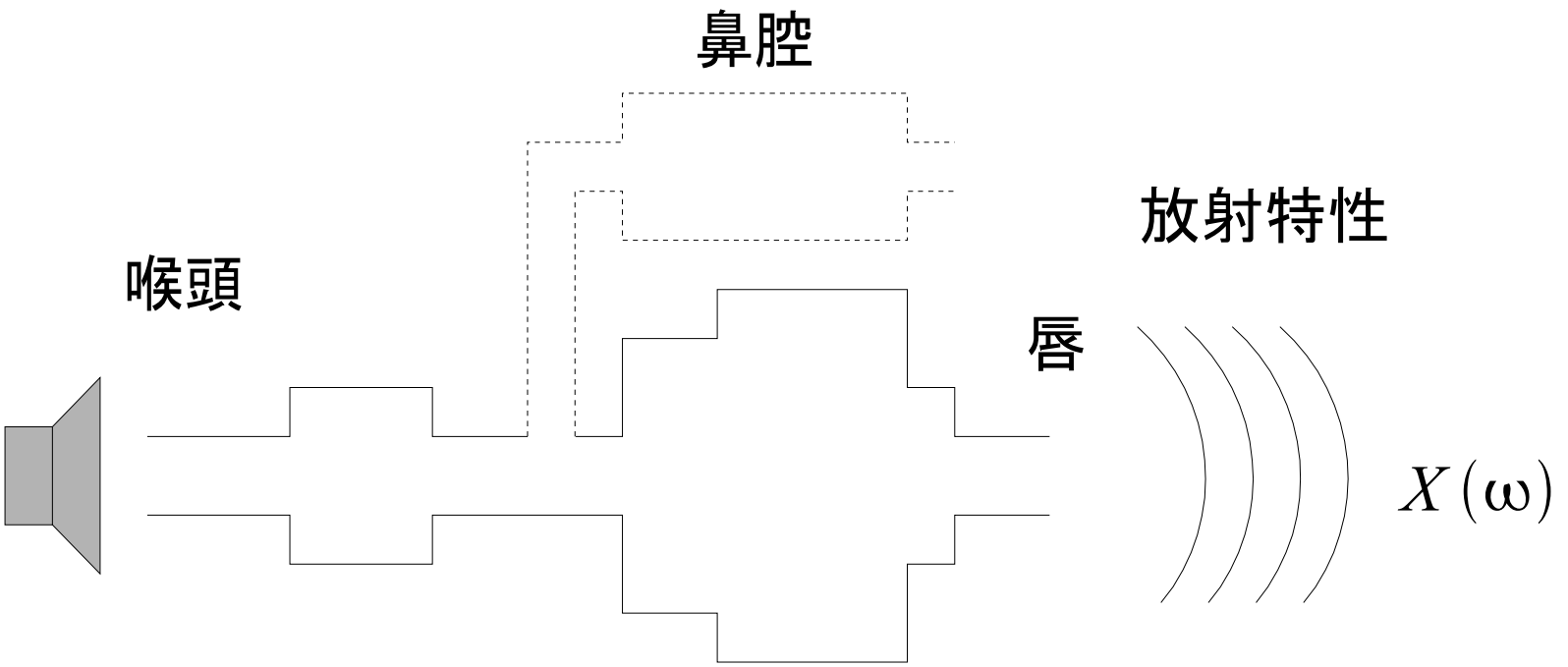
効率の高い音声符号化のために

- PCM, DPCM, ADPCM は一般の音信号を符号化する方法
  - DPCM, ADPCMは信号の性質を一部利用
- 人間の音声は音信号のごく一部
  →「声」として必要最小限の部分だけを使えば効率が上がる
- 「声」の必要最小限とは？

高レベル音声符号化

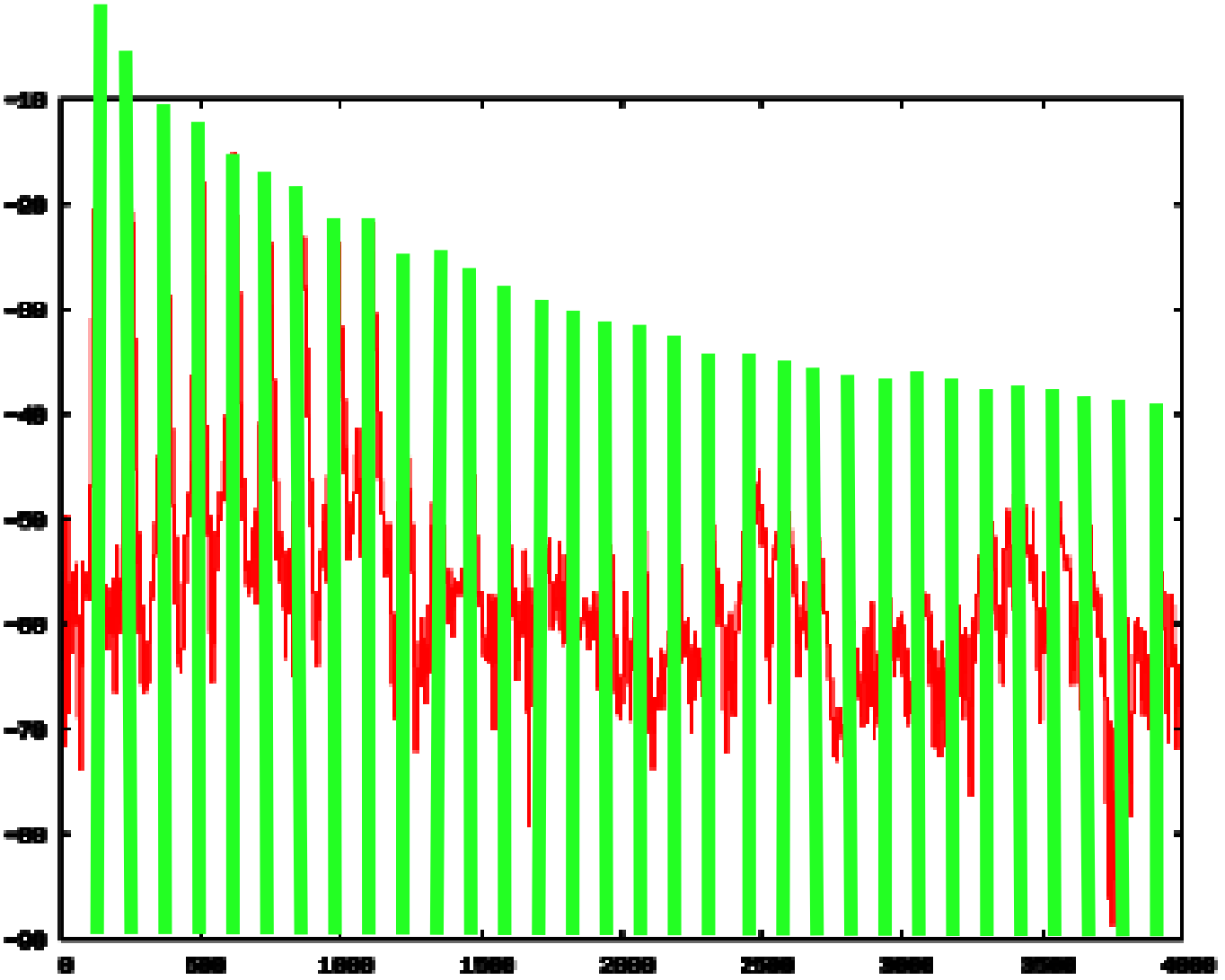

音声
ディジタル
データ
「声」の
特徴
音韻
単語/文
意味
PCM系
（固定電話等）
CELP系
（携帯電話等）
研究段階
要約電話？
音声
ディジタル
データ
「声」の
特徴
音韻
単語/文
意味
AD/DA
ボコーダ
音声合成
Text-to-
Speech

音声の生成モデル



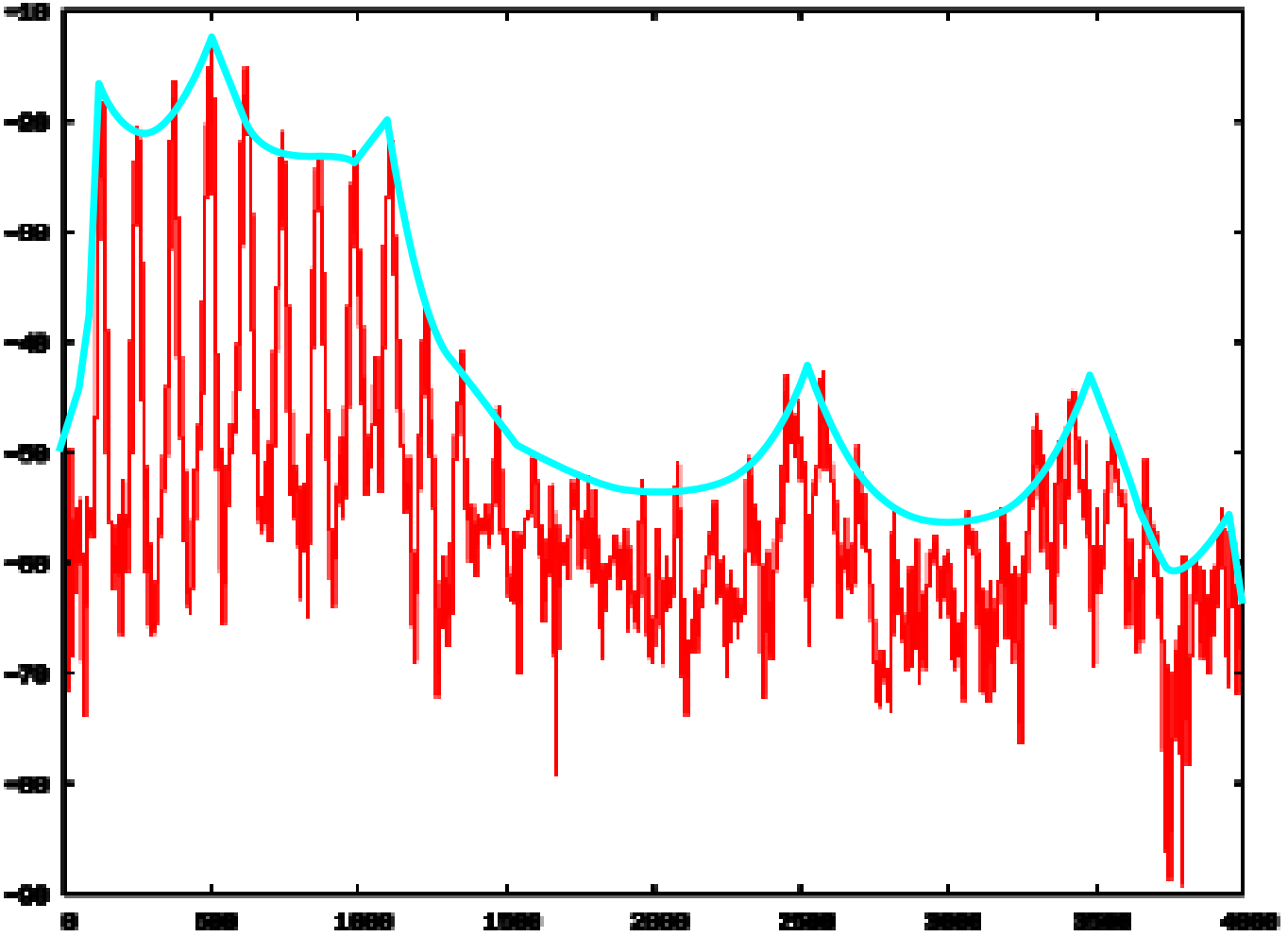
$$X(\omega)=S(\omega)T(\omega)R(\omega)$$

音声の生成モデル



$S(\omega)$

音声の生成モデル



$T(\omega)R(\omega)$

$S(\omega)$

出力からのシステム推定

- $X(\omega)$を観測しただけで，$S(\omega)$, $T(\omega)$が推定できるか？
  - 一般的には不可能
  - 何らかの仮定を置けば可能
- 仮定
  - $S(\omega)$はできるだけ白色雑音に近い (スペクトルが平坦) ※実際は違う
- 手法
  - 線形予測(LPC)分析

パラメータによる音声のモデル化

- 線形予測係数(LPC)によるモデル化
  - スペクトル概形：線形予測フィルタの特性
  - 声帯音源波：予測残差

$$x(k) = -\sum_{i=1}^{p} a_i x(k-i) + e(k)$$

残差を最小にするように係数を推定

  - スペクトル領域では

$$X(\omega) = \frac{E(\omega)}{1 + \sum_{n=1}^{p} a_n e^{ni\omega}} = E(\omega) H(\omega)$$

$S(\omega)$ $T(\omega) R(\omega)$

LPCによる分析と伝送

- 送信すべき情報
  - LPC係数 a_i と予測残差 $e(k)$
- どうやって送るか？
  - ある長さのサンプル(ブロック)全体に対してa_i を推定
  - 推定したa_i を使って$e(k)$を計算
  - ブロックに対してa_i と $e(k)$を伝送
- 受け取ったらどうするか
  - LPCの式に従って信号を復元

$$x(k) = -\sum_{i=1}^{p} a_i x(k-i) + e(k)$$

LPC係数の推定

- $x(1)...x(k)$からp次のLPC係数を推定するには？
  - 連立方程式(Yule-Walker方程式）を解く
    →誤差を最小にするLPC係数が求まる
  - より高速な解法もある
    (Levinson-Durbinのアルゴリズム)
- LPCの式

$$
\begin{aligned}
-a_p x(1) - a_{p-1} x(2) - \cdots - a_1 x(p) + e(p+1) &= x(p+1) \\
-a_p x(2) - a_{p-1} x(3) - \cdots - a_1 x(p+1) + e(p+2) &= x(p+2) \\
-a_p x(3) - a_{p-1} x(4) - \cdots - a_1 x(p+2) + e(p+3) &= x(p+3) \\
&\vdots \\
-a_p x(k-p) - a_{p-1} x(k-p+1) - \cdots - a_1 x(k-1) + e(k) &= x(k)
\end{aligned}
$$

LPC係数の推定

- LPCの式

$$-\begin{pmatrix} x(k-1) & x(k-2) & \cdots & x(k-p) \\ x(k-2) & x(k-3) & \cdots & x(k-p-1) \\ \vdots & \vdots & \ddots & \vdots \\ x(p) & x(p-1) & \cdots & x(1) \end{pmatrix}\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{pmatrix} = \begin{pmatrix} x(k) \\ x(k-1) \\ \vdots \\ x(p+1) \end{pmatrix} + \begin{pmatrix} e(k) \\ e(k-1) \\ \vdots \\ e(p+1) \end{pmatrix}$$

$$-FA = V + E$$

$$F = \{ x(k-i-j+1) \} \qquad (1 \le i \le k-p,\ 1 \le j \le p)$$

LPC係数の推定

- 最小二乗解: $|\mathrm{E}|^2$を最小化→$|\mathrm{FA+V}|^2$を最小化
- （式展開略）解くべき式は次のとおり

$$F^T F A = -F^T V$$

$F^T F = (\phi_{ij})$, $F^T V = (\phi_{0j})$ とすると

$$\begin{pmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1p} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{p1} & \phi_{p2} & \cdots & \phi_{pp} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{pmatrix} = - \begin{pmatrix} \phi_{01} \\ \phi_{02} \\ \vdots \\ \phi_{0p} \end{pmatrix}$$

Yule-Walker方程式
（正規方程式）

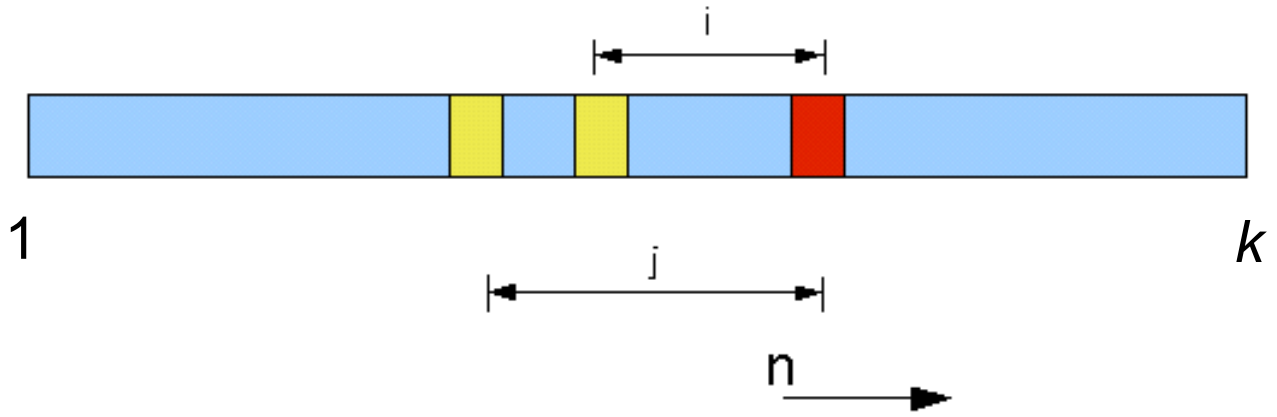
LPC係数の推定

- Yule-Walker方程式の要素

$$\phi_{ij} = \sum_{m=1}^{k-p} x(k-m-i+1)\,x(k-m-j+1) \qquad n = k-m+1 \quad \text{とおくと}$$

$$= \sum_{n=p+1}^{k} x(n-i)\,x(n-j)$$

- これをそのまま解く→共分散法

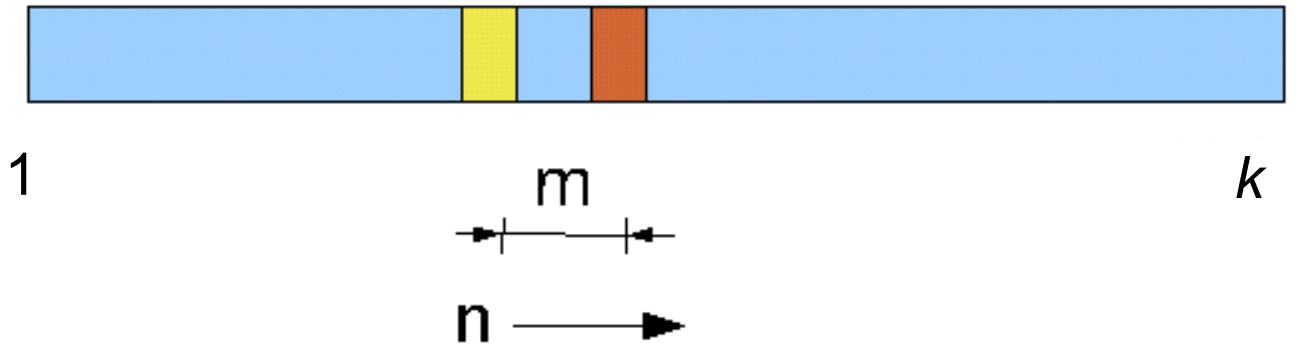


LPC係数の推定

- 高速解法(自己相関法) $k>>p$

$$\phi_{ij} \approx r(|i-j|) = \sum_{n=0}^{k-|i-j|-1} x(n)x(n+|i-j|)$$

  - 行列が特殊な形（対称Toeplitz行列）になる
  - 特別な解法(Levinson-Durbin法)で高速に解ける



LPCによる分析と伝送

- 問題点
  - LPCの式による復元は不安定になりやすい
    $\rightarrow a_i$ に量子化誤差があると発振することがある
- 解決方法
  - LPC係数と相互に変換可能で，量子化誤差に対して安定な量を使う
    - PARCOR係数
    - LSP係数

LPC係数とPARCOR係数

- PARCOR（偏自己相関）係数

$$k_i = \frac{\sum_{n=-\infty}^{\infty} \xi_{i-1}(n)\eta_{i-1}(n)}{\sqrt{\sum_{n=-\infty}^{\infty} \xi_{i-1}^2(n) \sum_{n=-\infty}^{\infty} \eta_{i-1}^2(n)}}$$

PARCOR係数は
前向き予測誤差と
後向き予測誤差の
相関係数に等しい

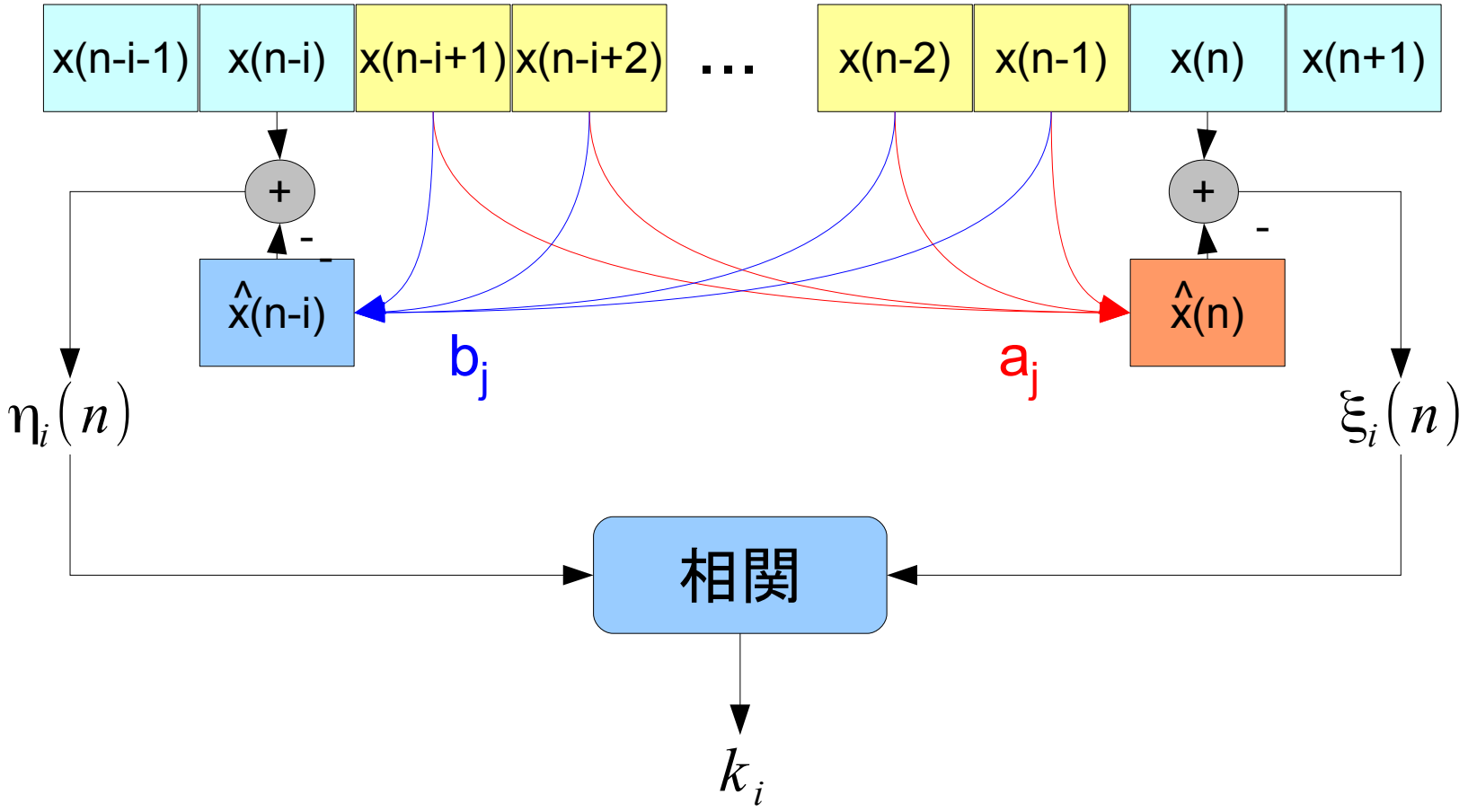
$$\xi_{i-1}(n) = x(n) + \sum_{j=1}^{i-1} a_j^{(i-1)} x(n-j)$$

前向き予測誤差

$$\eta_{i-1}(n) = x(n-i) + \sum_{j=1}^{i-1} b_j^{(i-1)} x(n-j)$$

後向き予測誤差

PARCOR係数



PARCOR係数と線形予測

$$k_1 = \frac{\sum_{n=-\infty}^{\infty} \xi_0(n)\eta_0(n)}{\sqrt{\sum_{n=-\infty}^{\infty} \xi_0^2(n) \sum_{n=-\infty}^{\infty} \eta_0^2(n)}} \qquad \xi_0(n) = x(n) \quad \eta_0 = x(n-1)$$

k_1は$x(n\text{-}1)$と$x(n)$の相関係数

$x(n\text{-}1)$と$x(n)$は分散が同じで平均が0なので

$$\hat{x}^{(1)}(n) = k_1 x(n-1) \quad \rightarrow a_1^{(1)} = -k_1$$

$$\hat{x}^{(1)}(n-1) = k_1 x(n) \quad \rightarrow b_1^{(1)} = -k_1$$

PARCOR係数と線形予測

$$k_2=\frac{\sum_{n=-\infty}^{\infty}\xi_1(n)\eta_1(n)}{\sqrt{\sum_{n=-\infty}^{\infty}\xi_1^2(n)\sum_{n=-\infty}^{\infty}\eta_1^2(n)}}$$

$$\xi_1(n)=x(n)-k_1x(n-1)$$

$$\eta_1(n)=x(n-2)-k_1x(n-1)$$

$$\hat{\xi}_1^{(2)}(n)=k_2\eta_1(n-1)=-k_1k_2x(n-1)+k_2x(n-2)$$

ここで $\xi_1(n)=x(n)-k_1x(n-1)$ より

$$\hat{x}^{(2)}(n)=k_1(1-k_2)x(n-1)+k_2x(n-2)$$

$$\rightarrow a_1^{(2)}=-k_1(1-k_2)=a_1^{(1)}-k_2b_1^{(1)}$$

$$a_2^{(2)}=-k_2$$

同様に

$$b_2^{(2)}=b_1^{(1)}-k_2a_1^{(1)} \qquad b_1^{(2)}=-k_2$$

PARCOR係数と線形予測

一般に

$$a_j^{(i)} = a_j^{(i-1)} - k_i b_j^{(i-1)} \quad a_0^{(i-1)} = 0$$

$$b_j^{(i)} = b_{j-1}^{(i-1)} - k_i a_{j-1}^{(i-1)} \quad b_0^{(i-1)} = 0$$

この漸化式により，順次線形予測係数をPARCOR係数から求めることができる

LPC係数とLSP

- LSP（線スペクトル対）とは
  - 線形予測式のz領域での表現

$$x(k)+\sum_{i=1}^{p} a_i x(k-i)=e(k) \Rightarrow X(z)\underbrace{\left(1+\sum_{i=1}^{p} a_i z^{-i}\right)}_{A^{(p)}(z)}=E(z)$$

  - $A(z)$を$P(z)$と$Q(z)$に分解

$$P(z)=A^{(p)}(z)-z^{-(p+1)}A^{(p)}(z^{-1})$$
$$Q(z)=A^{(p)}(z)+z^{-(p+1)}A^{(p)}(z^{-1})$$

$$A^{(p)}(z)=\frac{P(z)+Q(z)}{2}$$

LSPパラメータ

- $P(z)=0$と$Q(z)=0$の根は周波数軸上(z領域では単位円上)にある
- $P(z)$と$Q(z)$は次のように書ける

$$0 < \theta_1 < \omega_1 < \cdots < \theta_{p/2} < \omega_{p/2} < \pi$$

$$\omega_1, \omega_2, \ldots, \omega_{p/2}$$

- $P(z)=0$と$Q(z)=0$の根　$z = \cos\omega_i \pm i \sin\omega_i$
- LSPパラメータ　$\omega_1, \omega_2, \ldots, \omega_p$

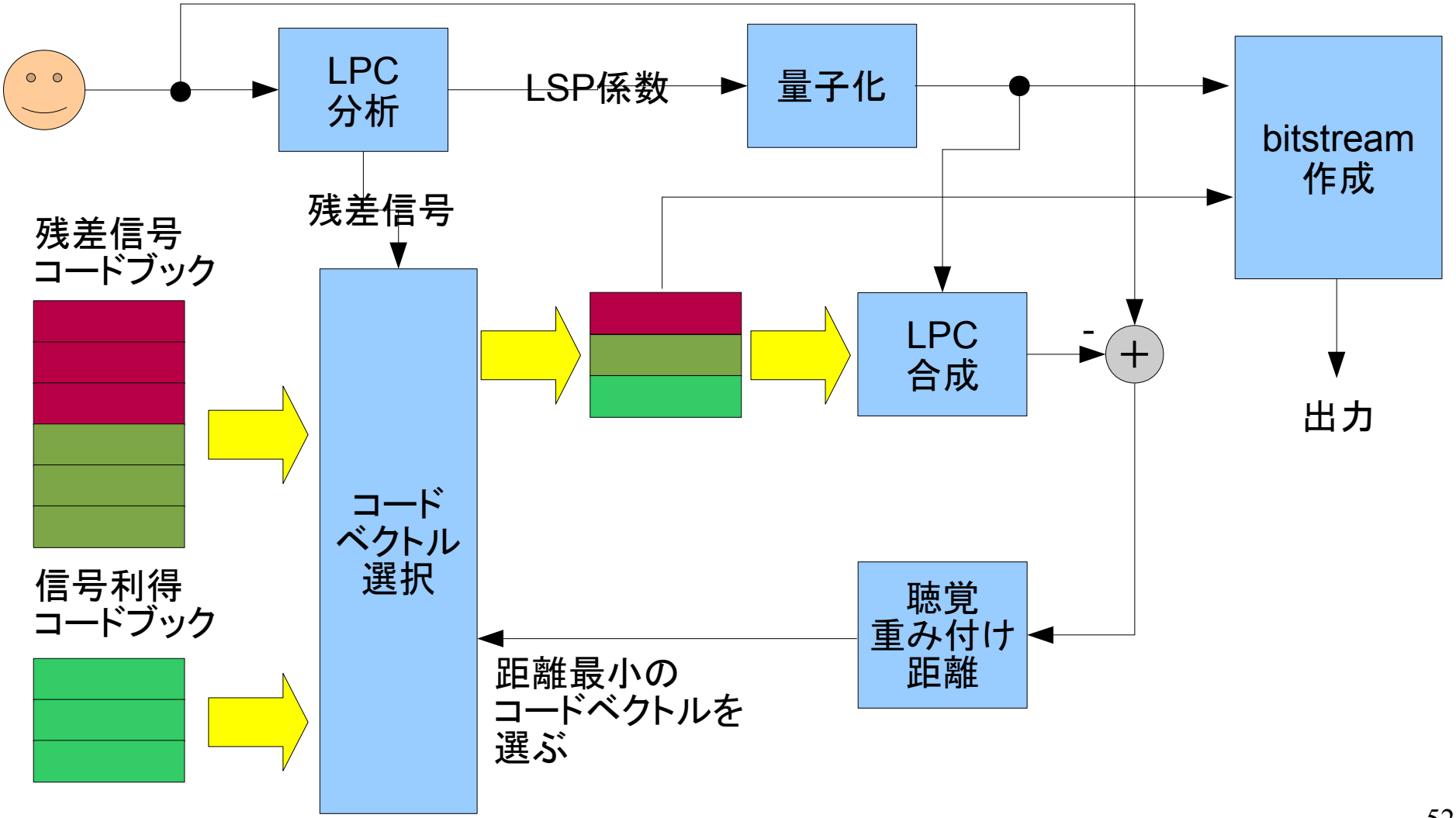
LSPパラメータの特徴

- 周波数軸上のパラメータである
- 数値解法によって求める
- 安定性の判別　$0 < \omega_1 < \omega_2 < \cdots < \omega_p < \pi$
- PARCORよりも量子化・補完に強い
- PARCORより推定・合成の処理量が多い
- 現在の音声符号化の主流

CELP符号化

- CELP(Code-Excitation Linear Prediction)符号化
  - 携帯電話での音声符号化の基本方式
  - LPCを基本とした分析合成
  - 予測残差を符号化して送る

CELPの構成


LPC
分析
LSP係数
量子化
bitstream
作成
残差信号
残差信号
コードブック
コード
ベクトル
選択
LPC
合成
-
+
出力
信号利得
コードブック
聴覚
重み付け
距離
距離最小の
コードベクトルを
選ぶ

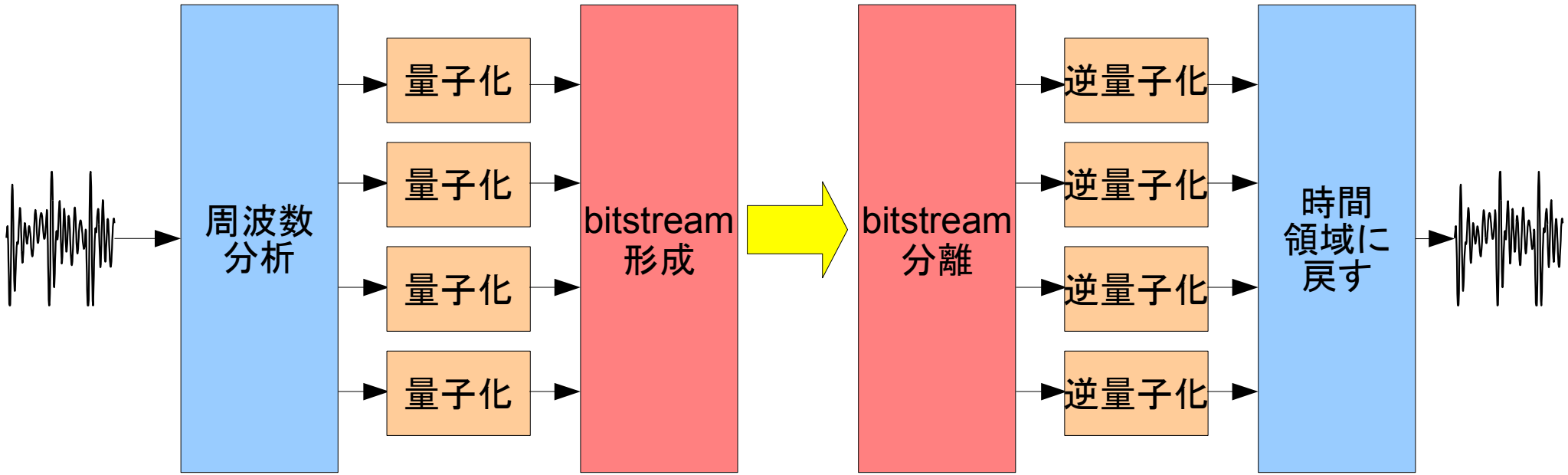
CELPに基づく各種符号化

- LD-CELP (Low-Delay CELP)
  - G.728　16kbit/s
- CS-ACELP (Conjugate Structure Algebraic CELP)
  - G.729　8kbit/s
- RPE-LTP (Regular Pulse Excitation with Long Term Prediction)
  - GSM標準　13kbit/s
- VSELP (Vector Sum Excitation LP)
  - PDC標準　6.7kbit/s
- PSI-CELP (Pitch Synchronous Innovation CELP)
  - PDCハーフレート標準 3.45kbit/s
- ACELP (Algebraic CELP)
  - GSM改訂標準 7.4kbit/s

オーディオ符号化

- 一般の音信号，音楽信号の符号化方式
  - 入力がどういう音か仮定が（あまり）できない
    - 一般には高域パワーが低域に比べて小さいことを仮定
  - 音声符号化のようにモデルに基づく方法は使えない
- 音の帯域に応じた符号化
  - 入力を低音～高音に分ける
    - フィルタを使った分離
    - 周波数分析(MDCT)
  - 音の高さに応じて量子化を変える
    - パワーの小さい部分は粗く量子化
    - よく聞こえない部分は粗く量子化

オーディオ符号化の基本的枠組み



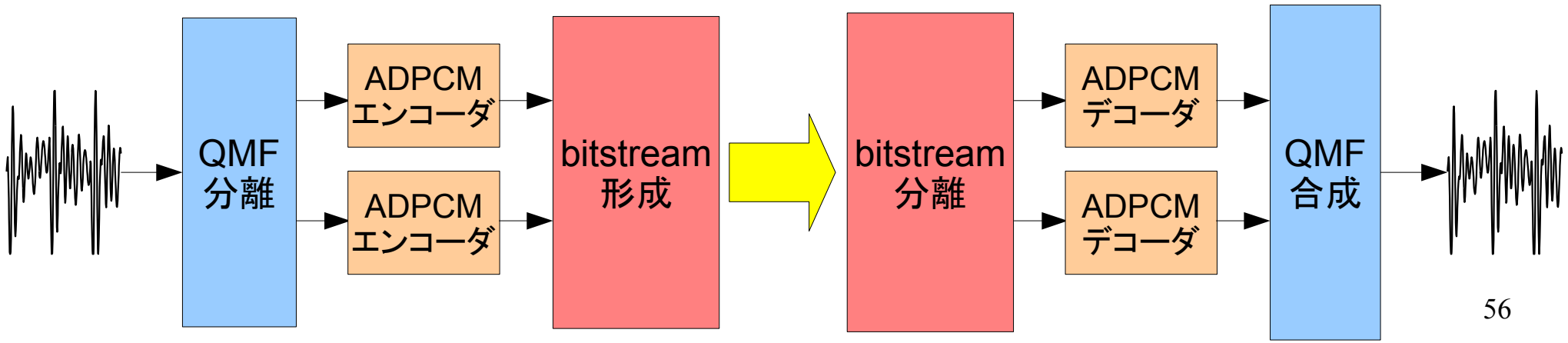
QMF
MDCT
Wavelet等

聴覚特性を考慮（最小可聴曲線，マスキング等）

エントロピー圧縮の利用(Huffman, 算術圧縮等)

SB-ADPCM

- 16kHz 高品質音声・中品質オーディオ符号化方式 (G.722)
  - Sub-Band ADPCM の略
  - 信号を高域と低域に分離し，それぞれADPCM符号化
    - 共役ミラーフィルタ(QMF)による帯域分離・合成
    - ADPCM符号化：高域2bit，低域4～6ビット (48～64kbit/s)

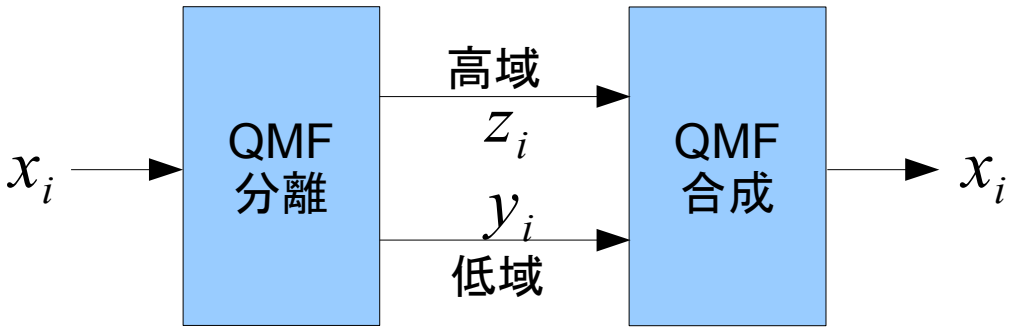


共役ミラーフィルタ

- Quadrature Mirror Filter (QMF)
- 入力信号を，同じデータ量のまま低域と高域の情報を含む信号に分離する
- 分離した情報を合わせることで，元の信号が完全に復元される
- 簡単なQMFの例 (Haar Wavelet)

$$y_i = \frac{x_{2i} + x_{2i+1}}{2}$$

$$z_i = \frac{x_{2i} - x_{2i+1}}{2}$$

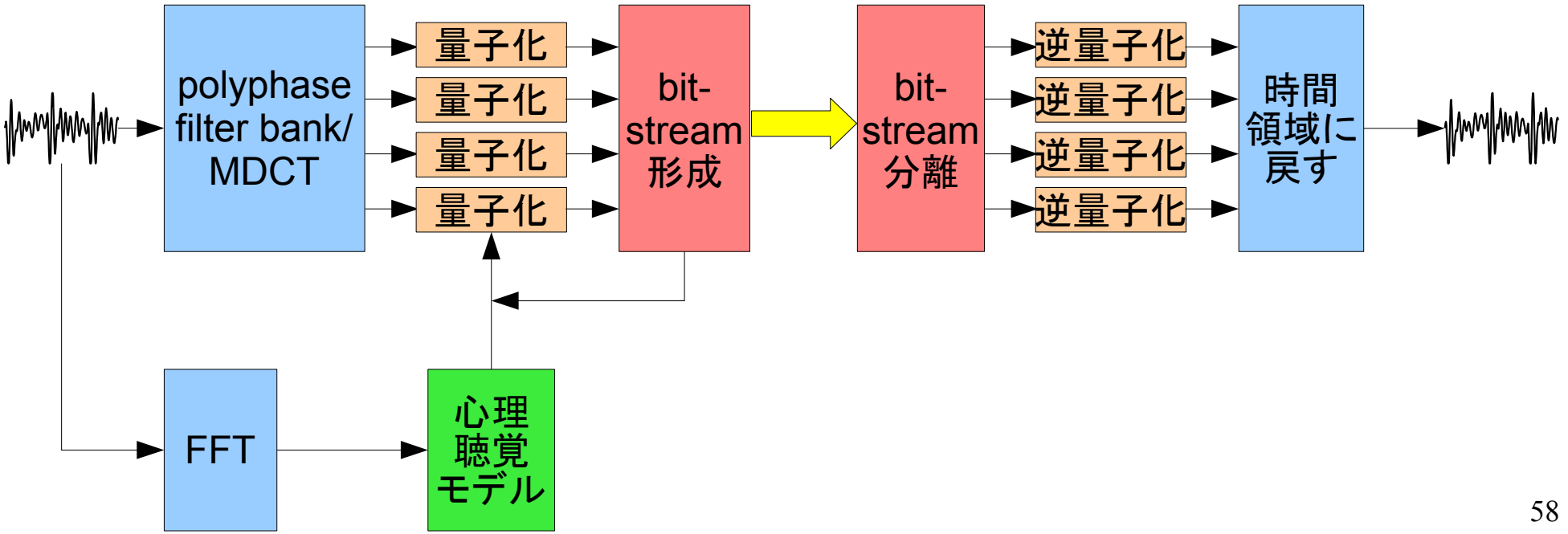


$$x_{2i} = y_i + z_i$$

$$x_{2i+1} = y_i - z_i$$

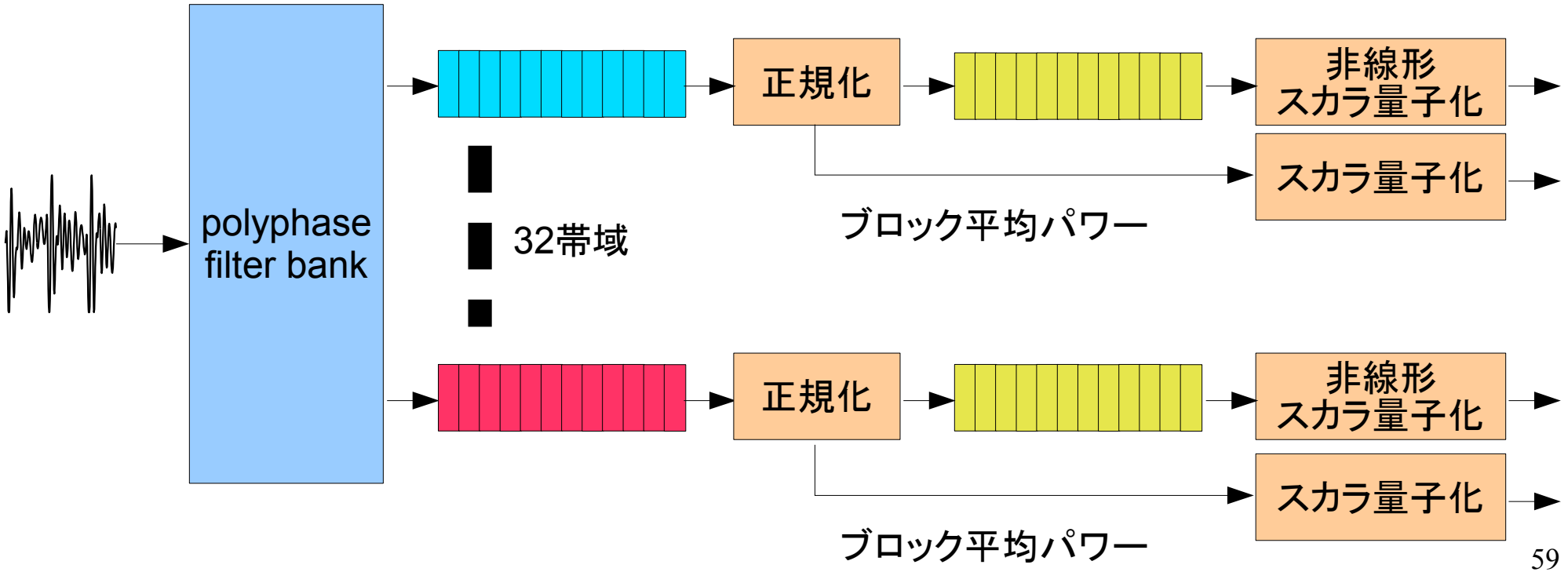
MPEG1 audio

- 映像符号化規格MPEGの音声部分
  - Layer 1 (MP1), layer 2 (MP2), layer 3 (MP3)がある
  - 周波数分析，心理聴覚モデル，非線形スカラ量子化



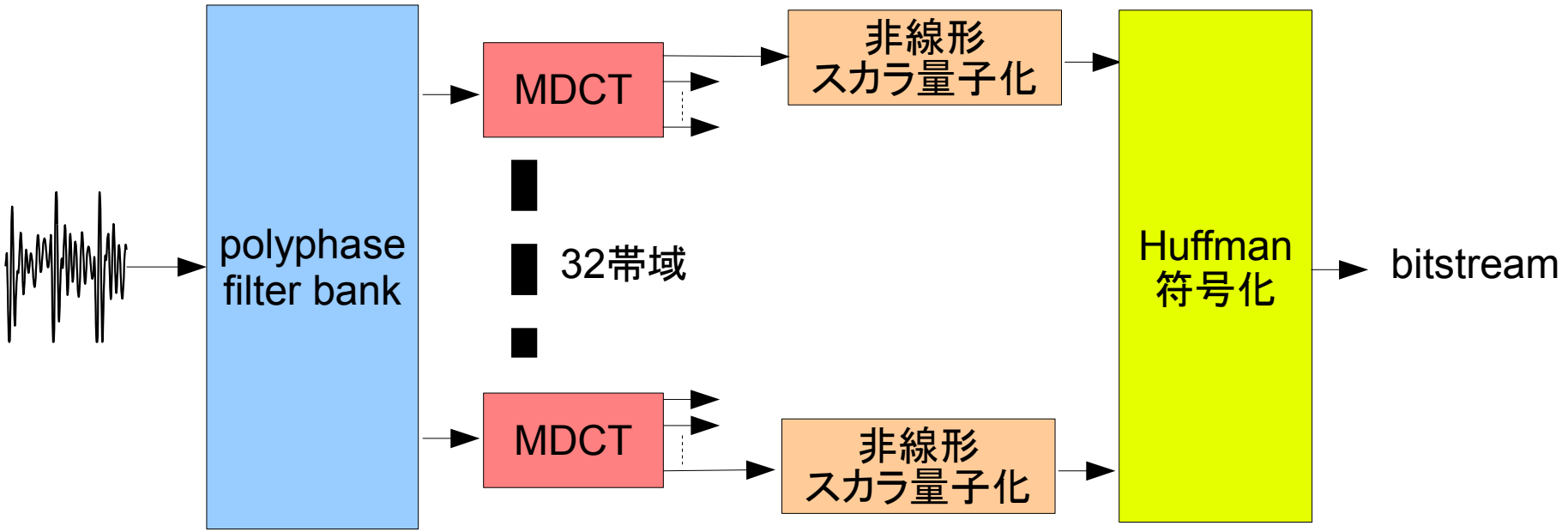
MP1

- MPEG1 audio layer 1
  - ポリフェーズフィルタバンクによる周波数分析
  - 12サンプルごとに正規化＋スカラ量子化



MP3

- ポリフェーズフィルタバンク＋MDCTによる分析
- 可変フレーム長（標準18点）
- エントロピー符号化



修正離散コサイン変換(MDCT)

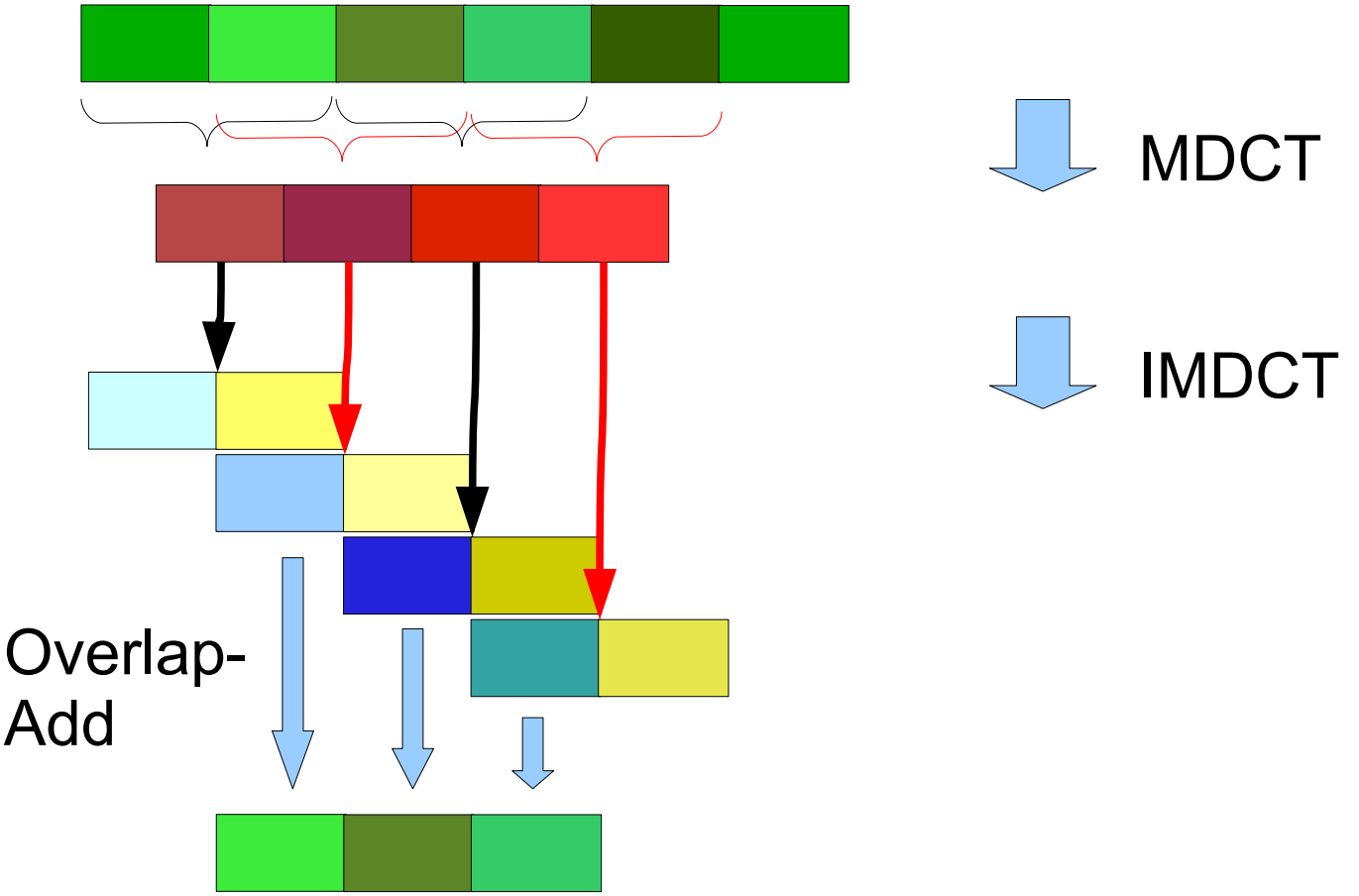
- Modified Discrete Cosine Transformの略
- n点の時間領域信号をn/2点の周波数領域信号に変換する
- n/2点オーバーラップ窓を利用する

$$X(m)=\sum_{k=0}^{n-1} f(k)\,x(k)\cos\left\{\frac{\pi}{2n}\left(2k+1+\frac{n}{2}\right)(2m+1)\right\}$$

$$x(k)=\frac{4f(k)}{n}\sum_{m=0}^{n/2-1} X(m)\cos\left\{\frac{\pi}{2n}\left(2k+1+\frac{n}{2}\right)(2m+1)\right\}$$

Overlap-Addによる復元

- 時間的に連続する復元信号を重ね合わせることで原信号が復元できる



音声強調

- 音声強調(Speech Enhancement)とは？
  - 雑音などの混入した音声信号から特定の音声だけを抜き出す
  - 一般には難しい：何らかの仮定が必要
- 主要な方法
  - 1チャンネルの場合
    - 線形な方法：ウィーナーフィルタ
    - 非線形な方法：スペクトル減算
  - 多チャンネルの場合
    - 線形な方法：マイクロホンアレイ（遅延和アレイ等）
    - 非線形な方法:多チャンネルスペクトル減算

1チャンネルの場合

- 音声信号x，雑音信号n，観測信号y

$$y(t)=x(t)+n(t)$$
$$Y(\omega)=X(\omega)+N(\omega)$$

- 目標：yからxを推定する（xもnも未知）
- このままでは不可能→前提が必要
  - nの振幅スペクトル（またはパワースペクトル）が既知（nの波形が既知，は考えにくい）

線形な手法:ウィーナフィルタ

- 誤差を最小にするフィルタWを考える

$$\hat{X}(\omega)=W(\omega)Y(\omega)=W(\omega)\left(X(\omega)+N(\omega)\right)$$

$$\int_0^{\pi}\left|\hat{X}(\omega)-X(\omega)\right|^2 d\omega\rightarrow\min$$

- スペクトルを離散化，時間周波数領域で考える

$$\sum_t\sum_{i=0}^{N-1}\left|X_i(t)-\hat{X}_i(t)\right|^2=$$

$$\sum_t\sum_{i=0}^{N-1}\left|X_i(t)-W_i\left(X_i(t)+N_i(t)\right)\right|^2\rightarrow\min$$

線形な手法:ウィーナフィルタ

- 前の式を微分して0とおく

$$\frac{\partial}{\partial W_i}\sum_t\sum_{i=0}^{N-1}\left|X_i(t)-W_i\left(X_i(t)+N_i(t)\right)\right|^2=0$$

$$\sum_t\left|-2X_i(t)\left(X_i(t)+N_i(t)\right)+2W_i\left(X_i(t)+N_i(t)\right)\right|^2=0$$

$X(t)$と$N(t)$が無相関と仮定すれば$\sum_t\left|X_i(t)N_i(t)\right|\approx 0$ より

$$W_i=\frac{\sum_t\left|X_i(t)\right|^2}{\sum_t\left|X_i(t)\right|^2+\sum_t\left|N_i(t)\right|^2}$$

ウィーナフィルタ

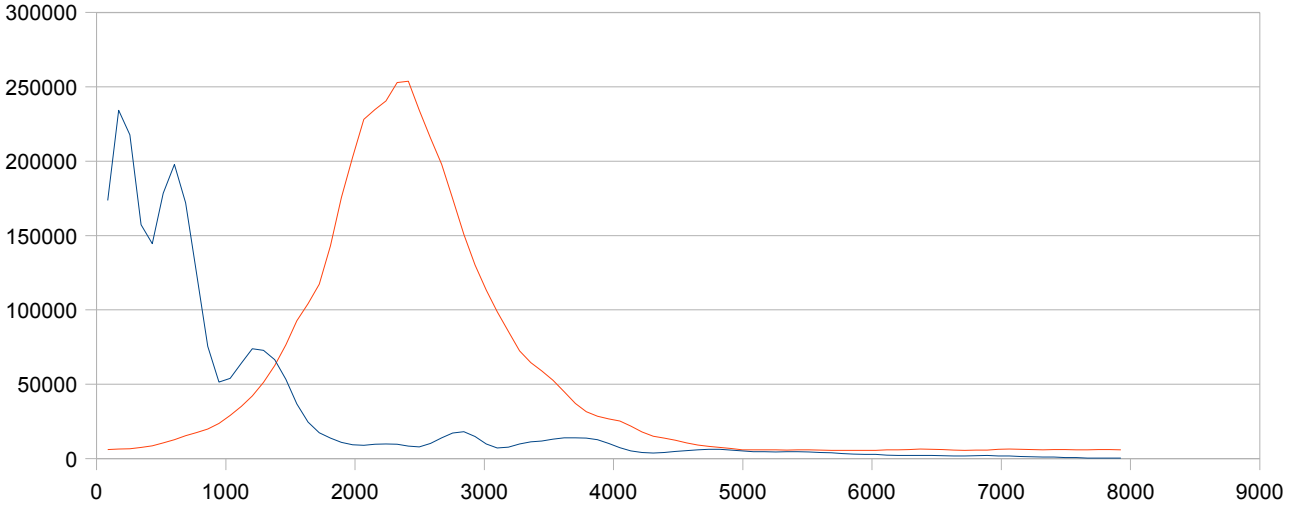
ウィーナフィルタ

- 適用条件
  - 信号および雑音の平均的パワースペクトルが既知
    - 雑音はともかく信号については厳しい仮定
  - 信号と雑音が無相関
- ウィーナフィルタの意味
  - 雑音のパワーが大きい帯域を抑圧する
    - 出力の各帯域におけるパワーは平均的に$E[X_i^2]$に一致

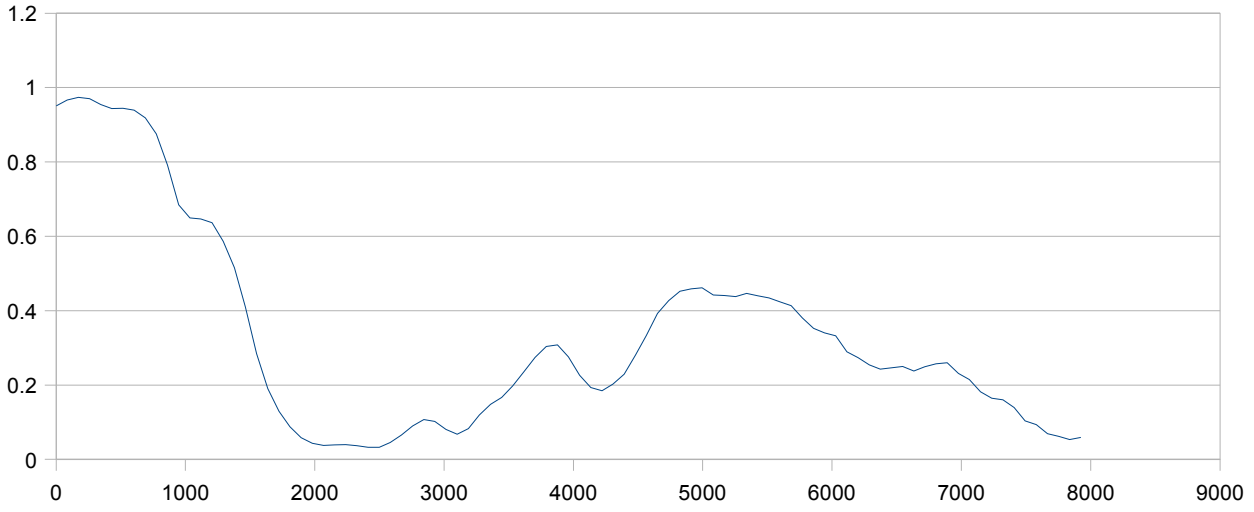
$$W_i\left(X_i(t)+N_i(t)\right)=\frac{\sum_t \left|X_i(t)\right|^2}{\sum_t \left|X_i(t)\right|^2+\sum_t \left|N_i(t)\right|^2}\cdot\left(X_i(t)+N_i(t)\right)$$

ウィーナフィルタ:適用例

音声と雑音



フィルタ特性



スペクトル減算

- 非線形処理によって雑音を抑圧する
- 雑音が定常だと仮定
- 観測信号のスペクトルから雑音のスペクトルを引き去る
- 準備
  - 信号　　$X_i(t)$
  - 雑音　　$N_i(t)$
  - 観測信号　　$Y_i(t)=X_i(t)+N_i(t)$

スペクトル減算

- スペクトル減算の原理
  - 観測信号のパワースペクトル

$$|Y_i(t)|^2 = |X_i(t) + N_i(t)|^2 \leq |X_i(t)|^2 + 2|X_i(t)\,N_i(t)| + |N_i(t)|^2$$

  - 信号と雑音が無相関と仮定

$$|X_i(t)\,N_i(t)| \ll |X_i(t)|^2 + |N_i(t)|^2$$

$$\Rightarrow \quad |X_i(t)|^2 \approx |Y_i(t)|^2 - |N_i(t)|^2$$

  - 音声区間で雑音の定常性を仮定 $|N_i(t)|^2 = N_i^2$

$$\Rightarrow \quad |X_i(t)|^2 \approx |Y_i(t)|^2 - N_i^2$$

スペクトル減算

- 雑音スペクトルの推定
  - あらかじめ用意しておく
  - 発話区間の直前（無音区間)などから推定
- パワーでないスペクトルを推定するには

$$X_i(t) \approx \sqrt{\frac{\overline{|Y_i(t)|^2 - |N_i(t)|^2}}{|Y_i(t)|^2}}\, Y_i(t)$$

実用上の問題と工夫

- 引きすぎ：パワースペクトルの推定値が負になる
  - フロアリングによる対処

$$|X_i(t)|^2 \approx \begin{cases} |Y_i(t)|^2 - N_i^2 & \text{if} \quad |Y_i(t)|^2 > N_i^2 \\ \beta |Y_i(t)|^2 & \text{otherwise} \end{cases} \qquad (0 < \beta \ll 1)$$

- 過剰推定：雑音を多めに見積もることで性能向上

$$|X_i(t)|^2 \approx \begin{cases} |Y_i(t)|^2 - \alpha N_i^2 & \text{if} \quad |Y_i(t)|^2 > \alpha N_i^2 \\ \beta |Y_i(t)|^2 & \text{otherwise} \end{cases} \qquad (\alpha > 1)$$

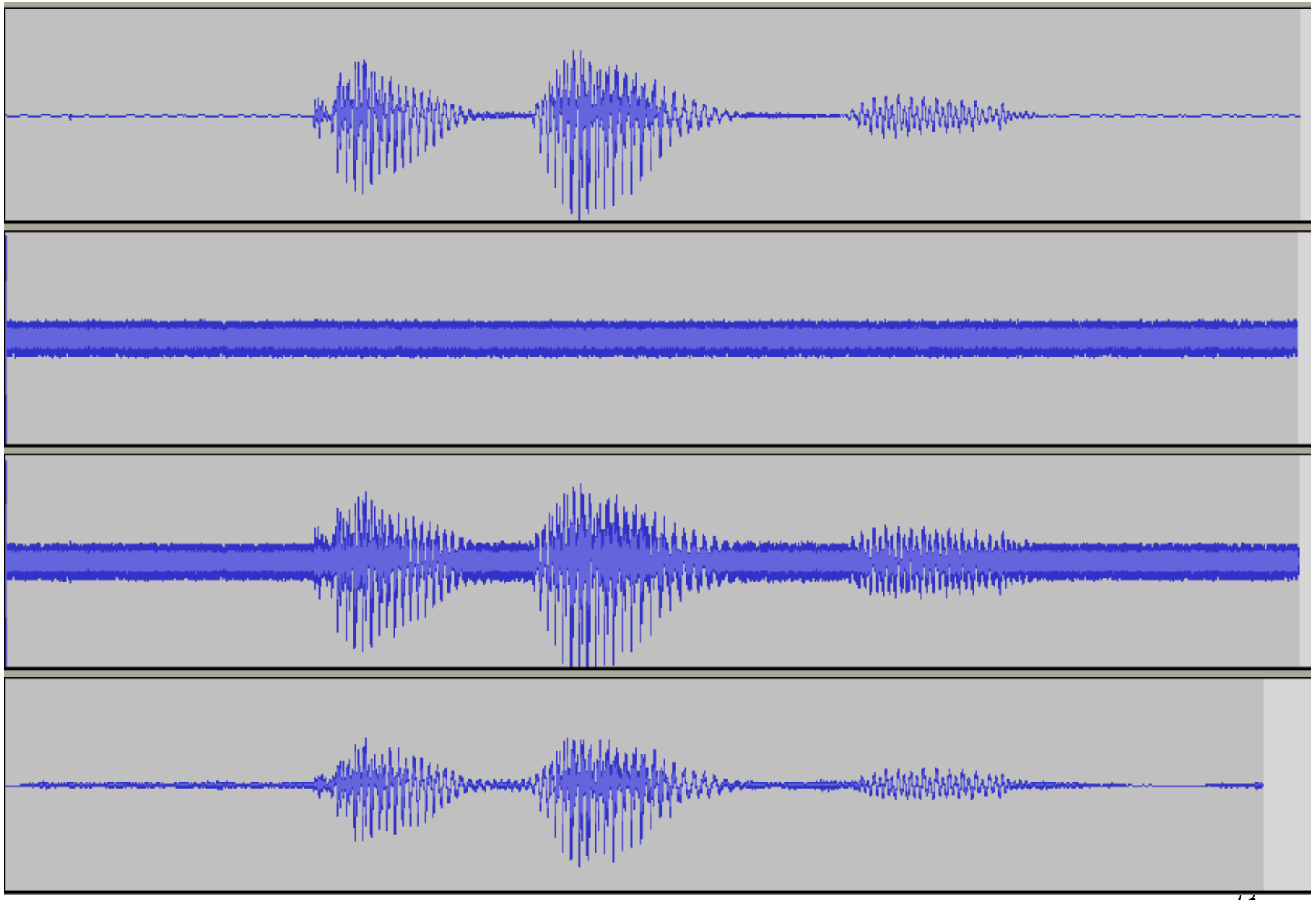
適用例(波形)

音声

雑音

雑音入り

雑音除去



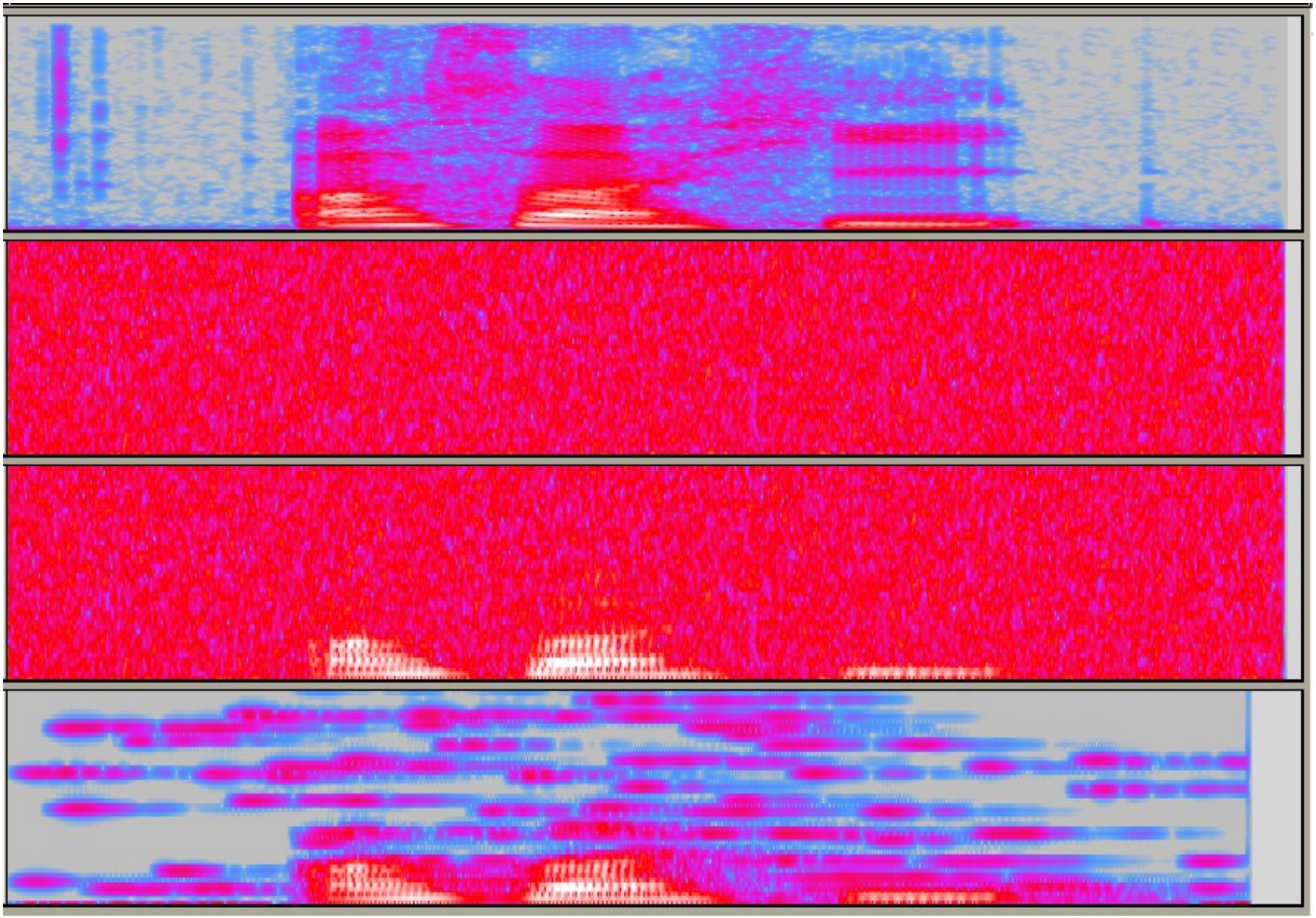
適用例(スペクトル)

音声

雑音

雑音入り

雑音除去

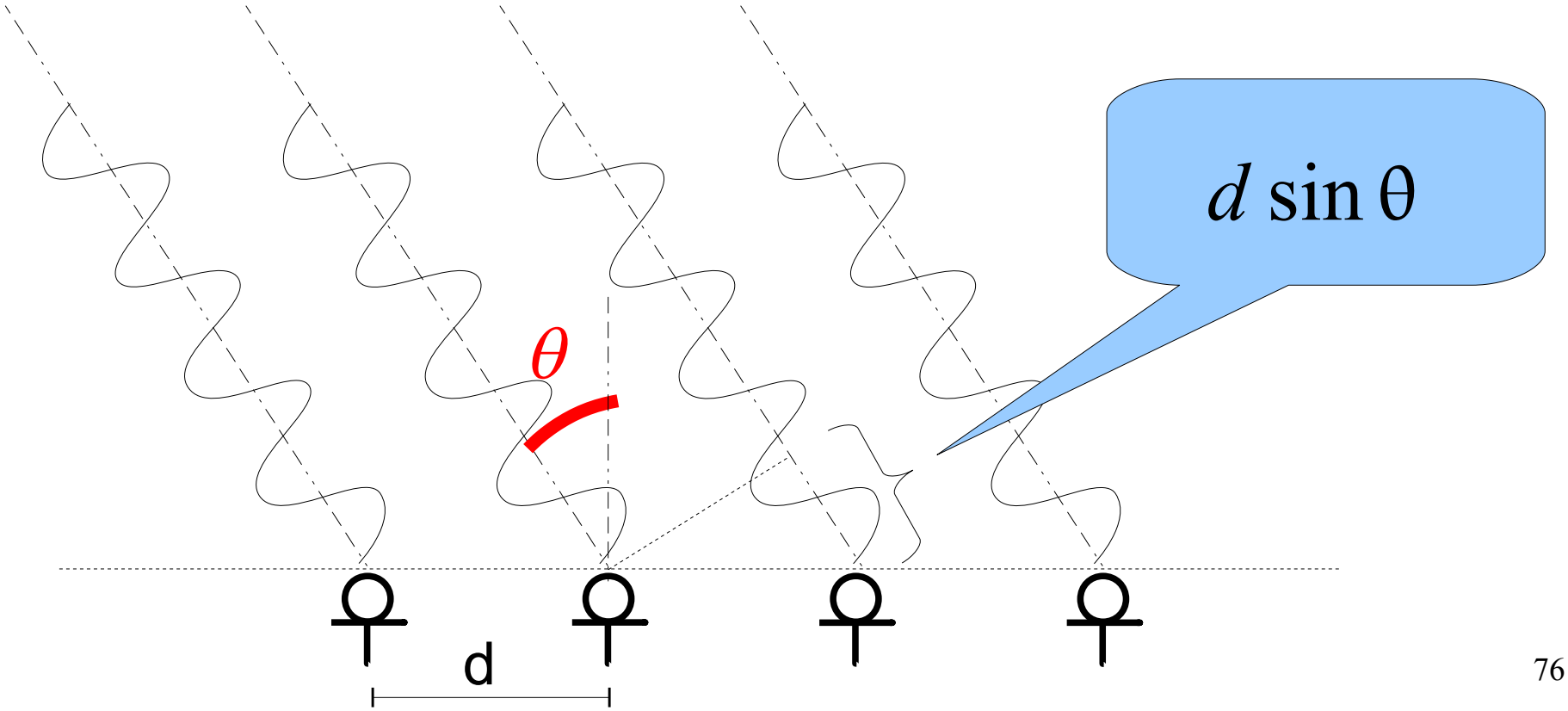


複数マイクロホンによる方法

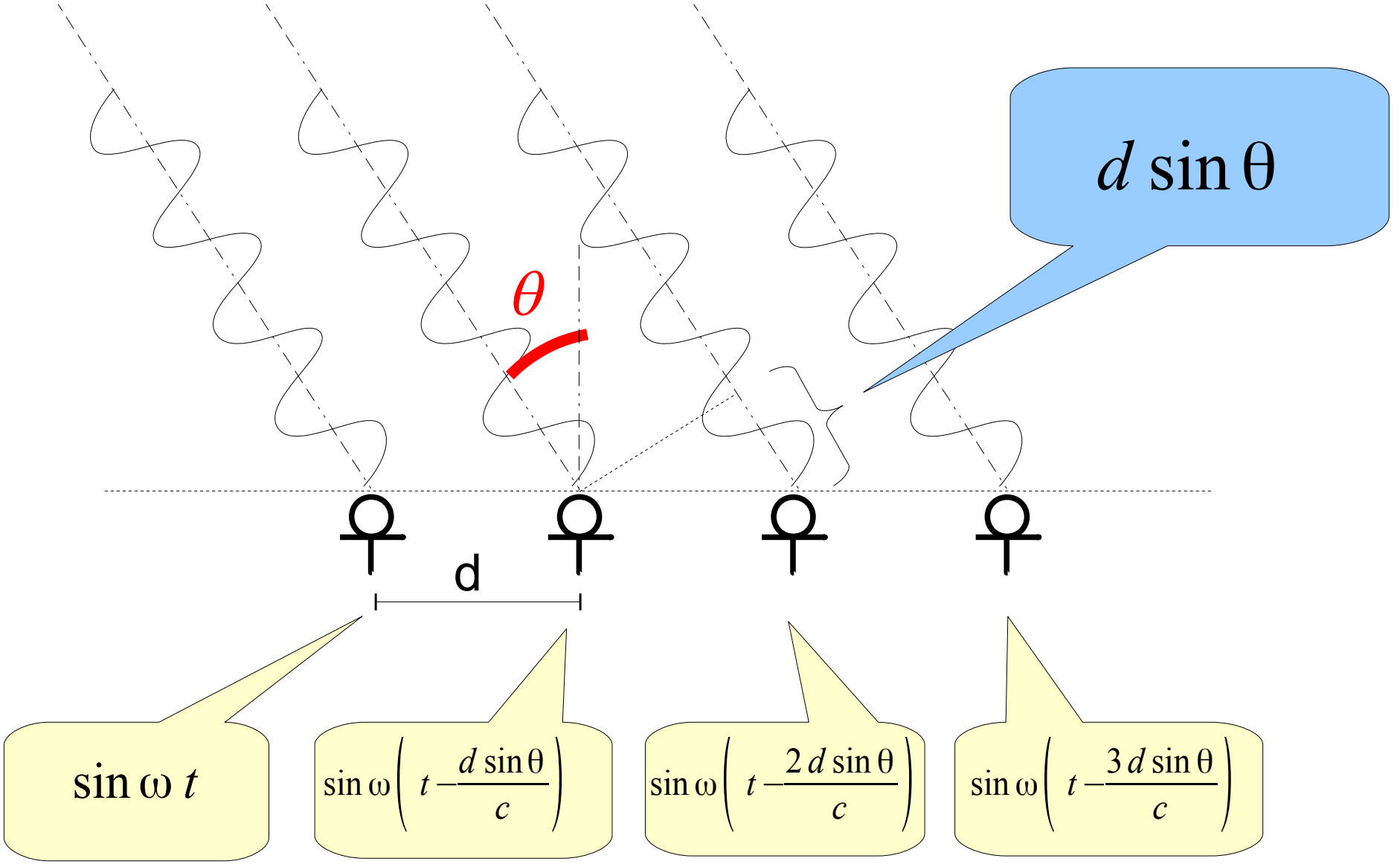
- 2本以上のマイクロホンによる集音を利用
  - 空間的な情報が利用できる
    - 音声と雑音を個別に拾う
    - 指向性を形成
- さまざまな方法がある
  - 線形処理
    - 遅延和アレイ（超指向性を形成）
    - 適応アレイ
  - 非線形処理
    - マルチチャネルスペクトル減算

遅延和アレイ

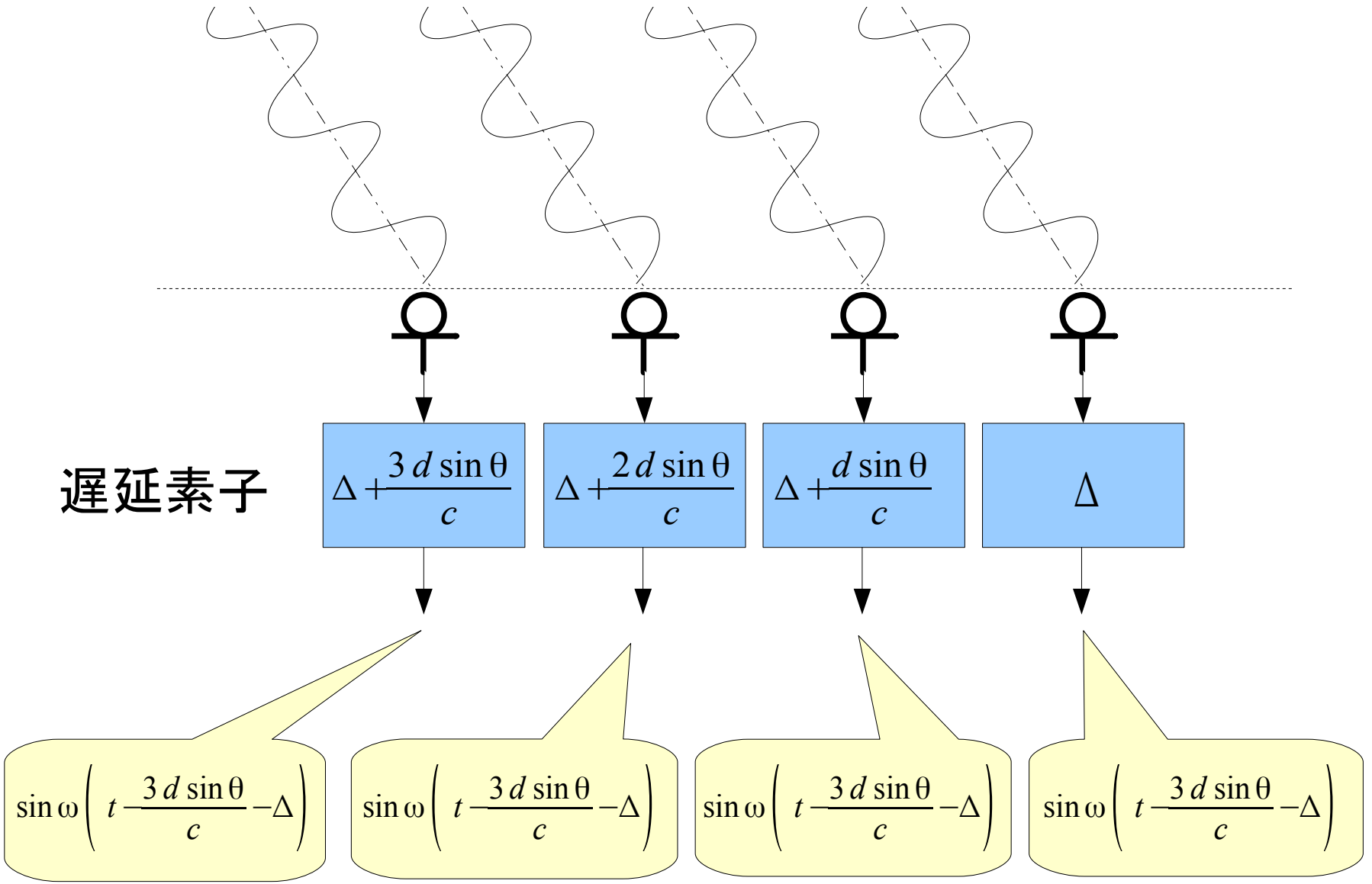
- 特定の角度から到来した音を複数のマイクロホンで拾う
  - 平面波を仮定



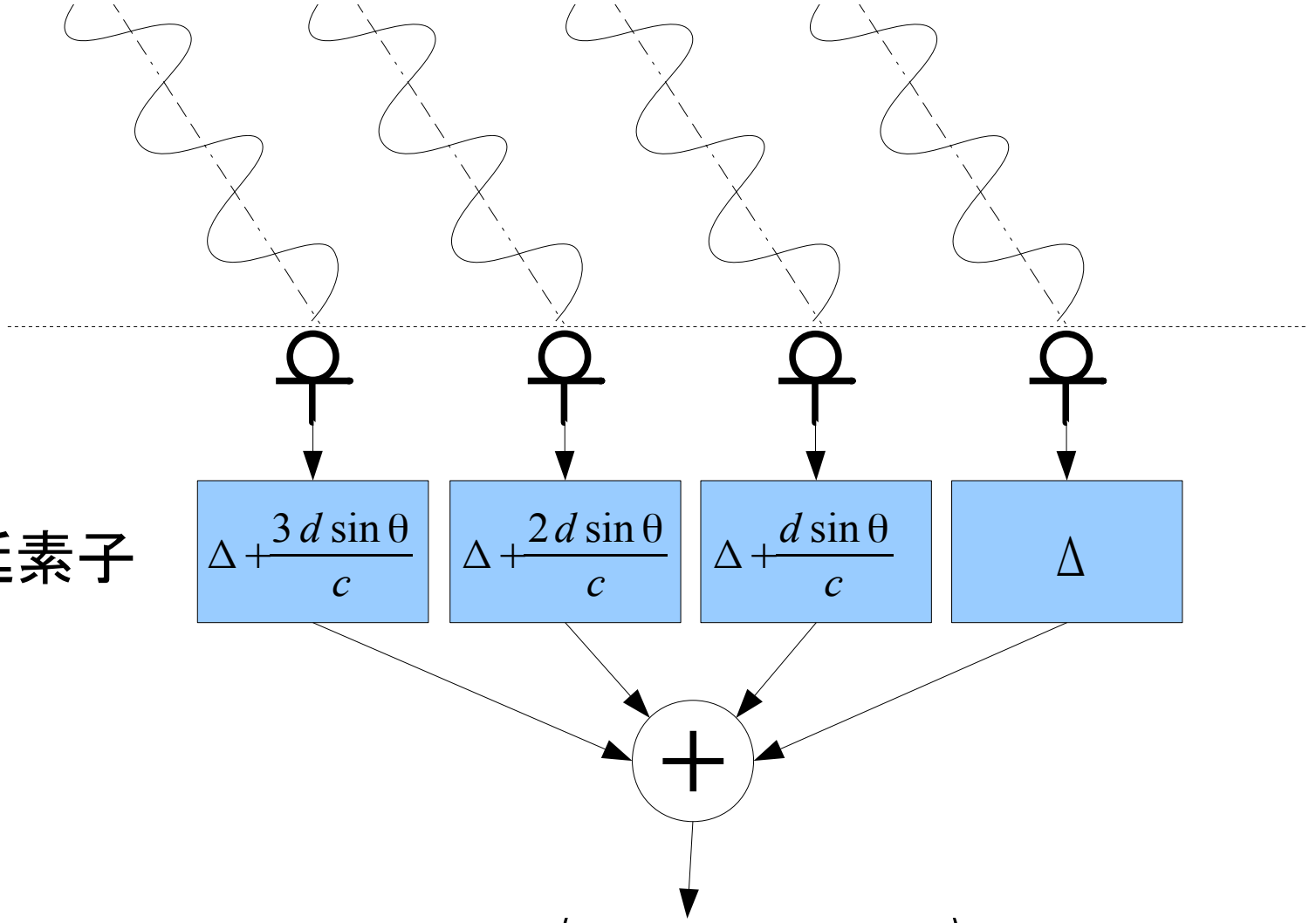
遅延和アレイ


$d \sin\theta$
θ
d
$\sin \omega t$
$\sin \omega \left(t - \frac{d \sin\theta}{c} \right)$
$\sin \omega \left(t - \frac{2 d \sin\theta}{c} \right)$
$\sin \omega \left(t - \frac{3 d \sin\theta}{c} \right)$

遅延和アレイ

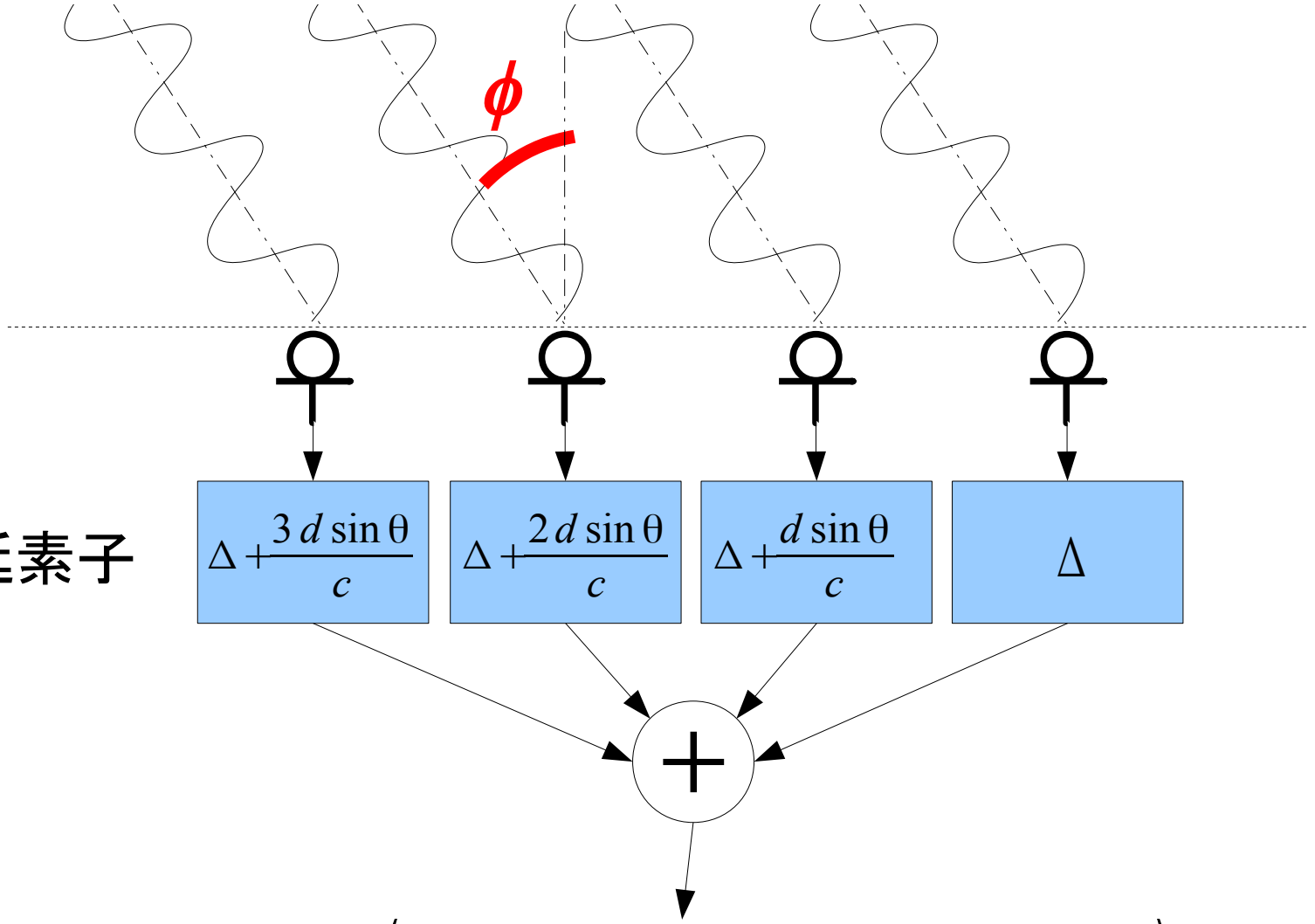

遅延素子
$\Delta + \frac{3d\sin\theta}{c}$
$\Delta + \frac{2d\sin\theta}{c}$
$\Delta + \frac{d\sin\theta}{c}$
Δ
$\sin\omega\left(t - \frac{3d\sin\theta}{c} - \Delta\right)$
$\sin\omega\left(t - \frac{3d\sin\theta}{c} - \Delta\right)$
$\sin\omega\left(t - \frac{3d\sin\theta}{c} - \Delta\right)$
$\sin\omega\left(t - \frac{3d\sin\theta}{c} - \Delta\right)$

遅延和アレイ



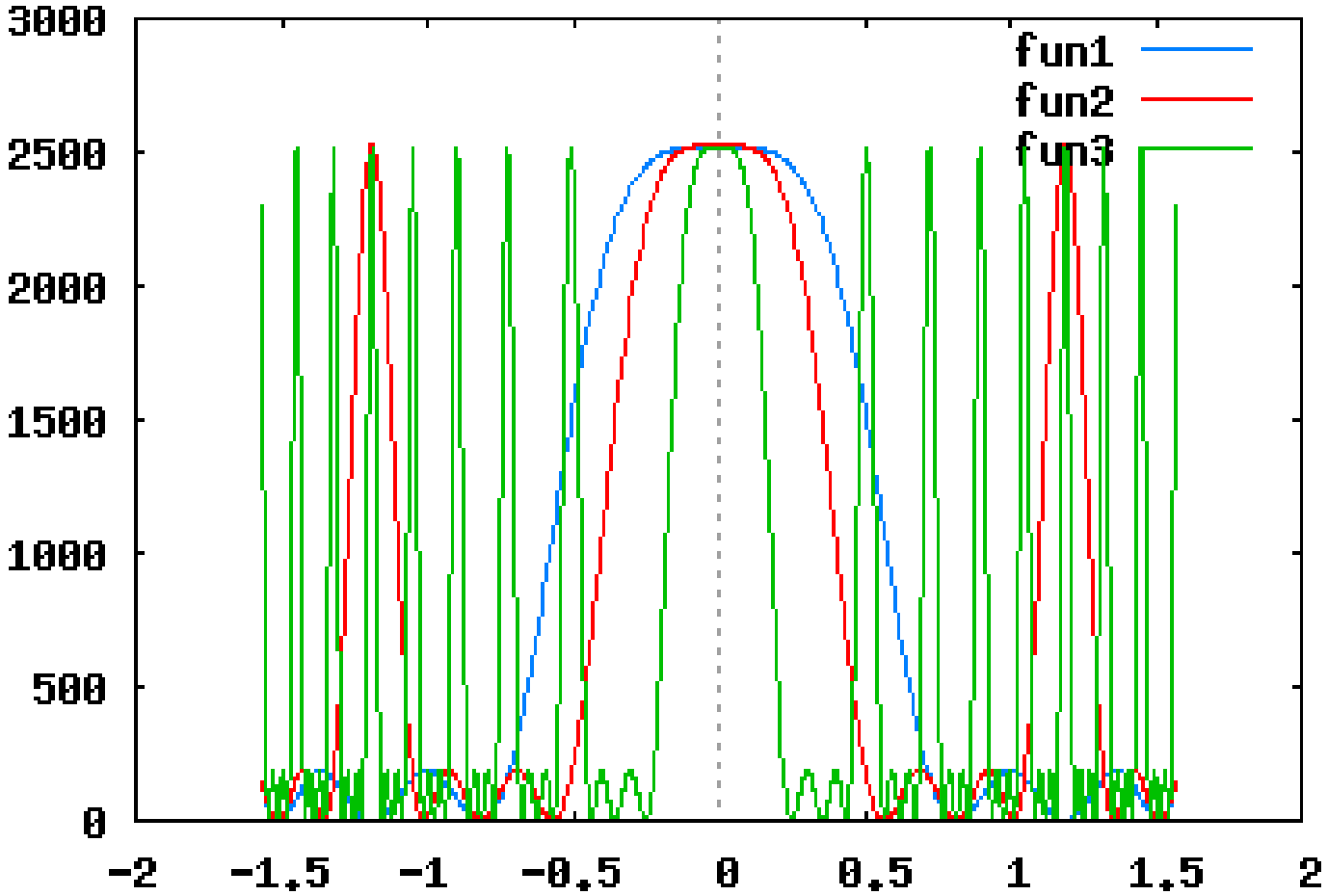
$$4\sin\omega\left(t\frac{3d\sin\theta}{c}-\Delta\right)$$

遅延和アレイ



$$\sum_{n=0}^{3}\sin\omega\left(t-\frac{nd\sin\phi+(3-n)d\sin\theta}{c}-\Delta\right)$$

適用例



n=4, d=1

fun1: ω=5

fun2: ω=10

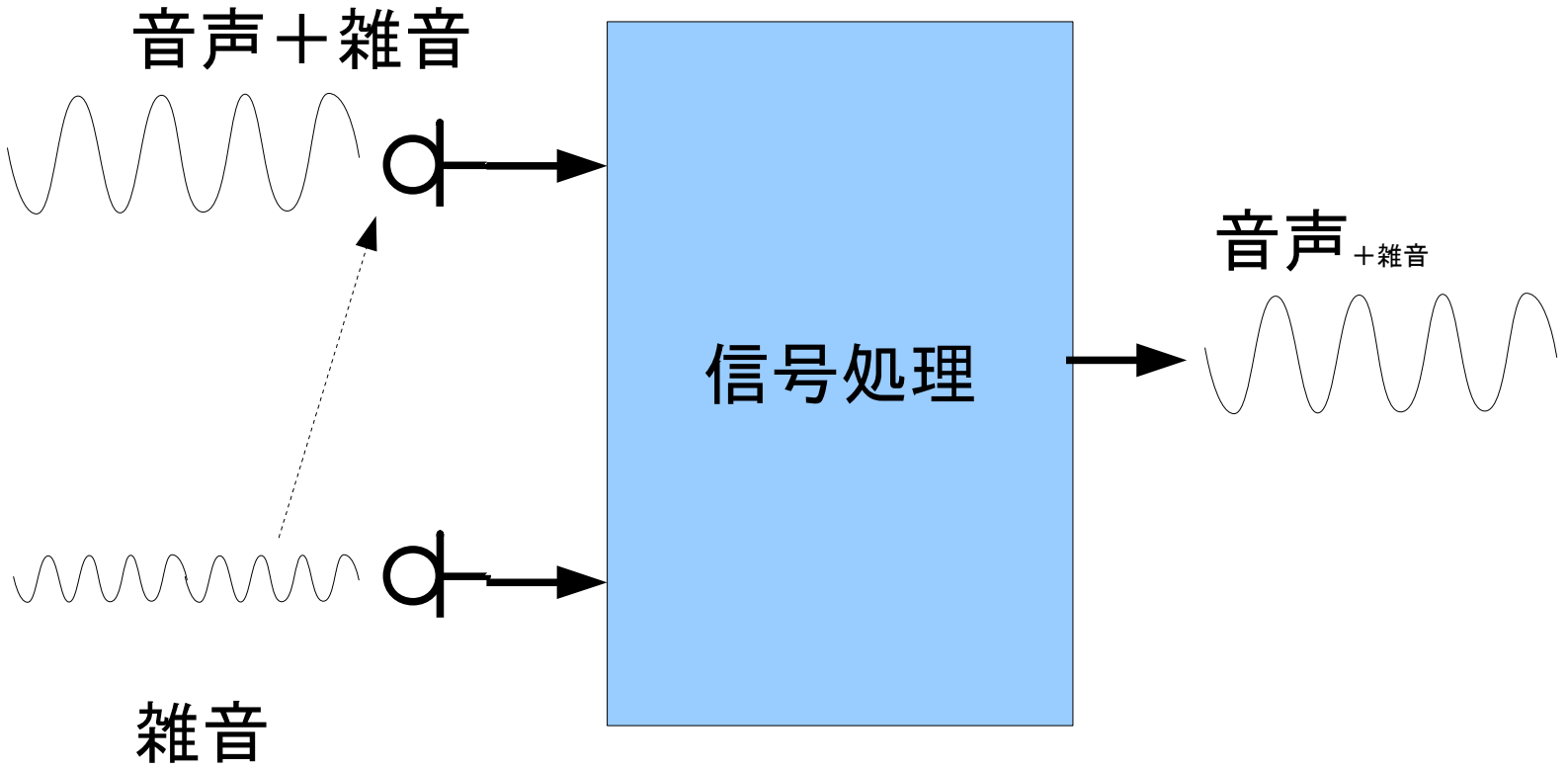
fun3: ω=50

遅延和アレイの特徴

- 処理が単純
  - 高速
  - ハードウェア化が容易
- 指向性
  - メインローブ幅
    - 素子数が多いほど狭い
    - 周波数が高いほど狭い
    - マイクロホン間隔が広いほど狭い
  - 空間的折り返し歪み
    - 折り返しが発生しない条件 $d<c/2f$

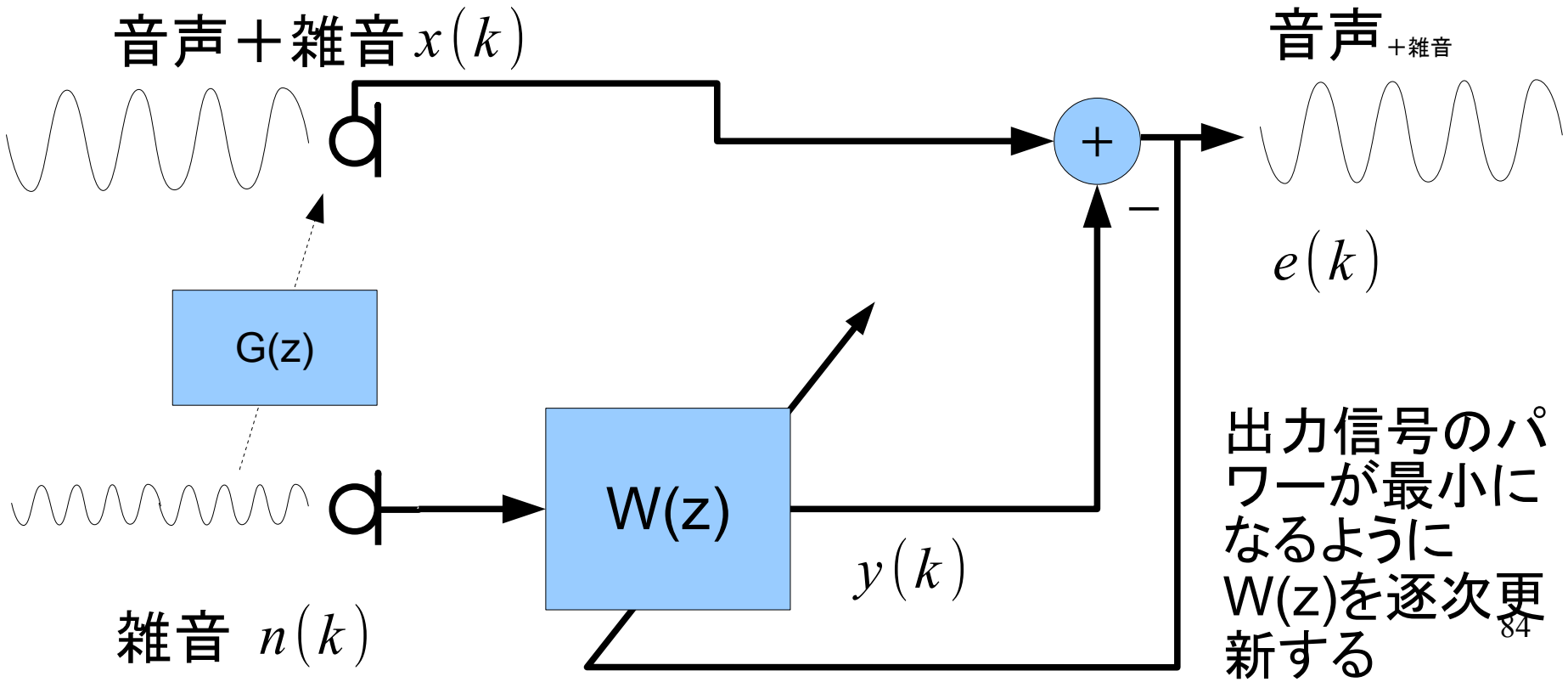
適応雑音抑圧

- 雑音をマイクで拾える場合
- 線形フィルタ処理



適応雑音抑圧

- 適応フィルタによる処理
  - n(k)はx(k)に混入している雑音そのものではないので直接減算はできない→フィルタW(z)を利用



適応フィルタ

- $W(z)$としてFIRフィルタを考える

$$y(k)=\sum_{i=1}^{p} w_i(k)\,n(k-i)$$

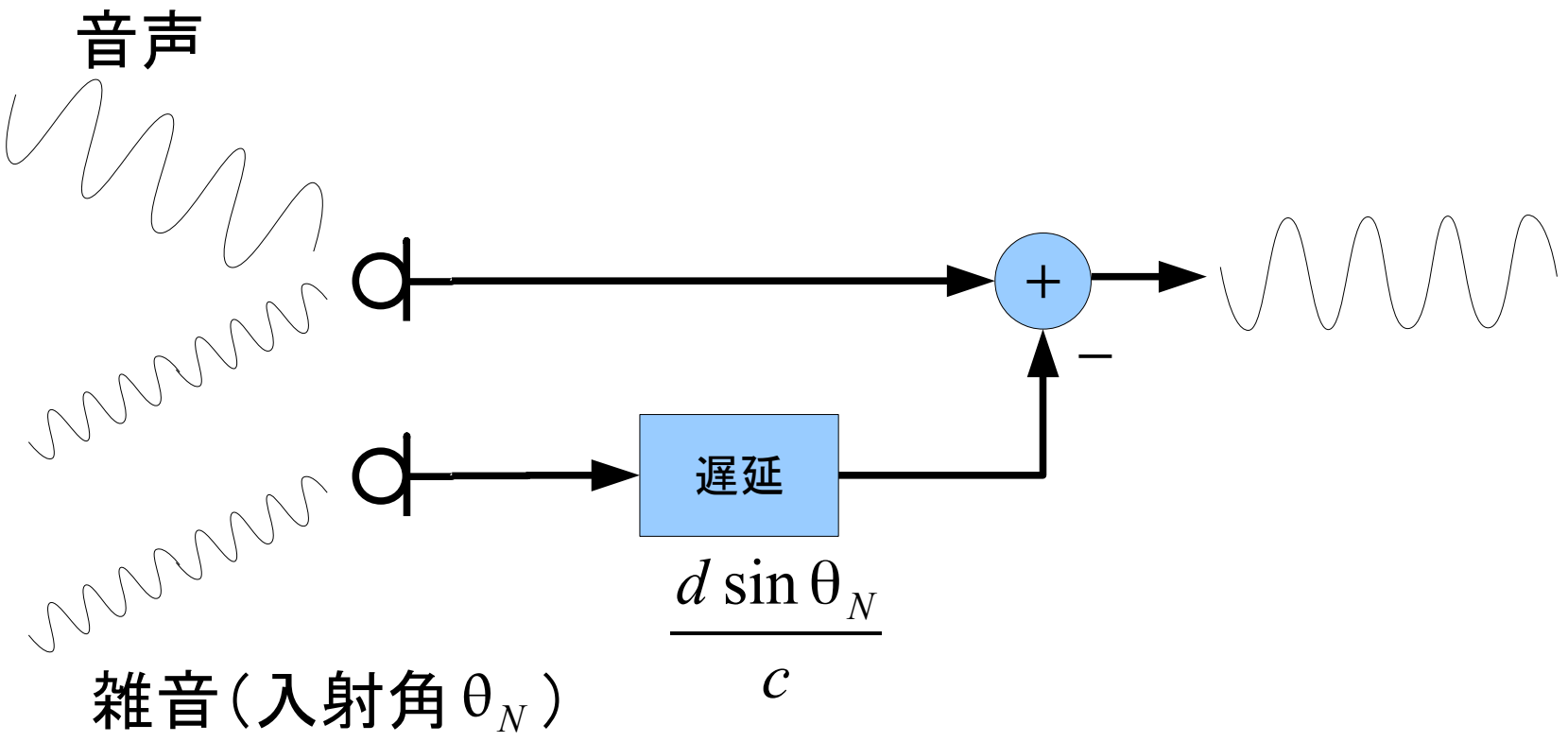
  - $w_i(k)$: 時刻kにおけるフィルタのi番目の係数
- LMSアルゴリズムによる係数の更新

$$w_i(k+1)=w_i(k)+2\mu e(k)\,n(k-i+1)$$

（そのほかにも多くの係数更新アルゴリズムがある）

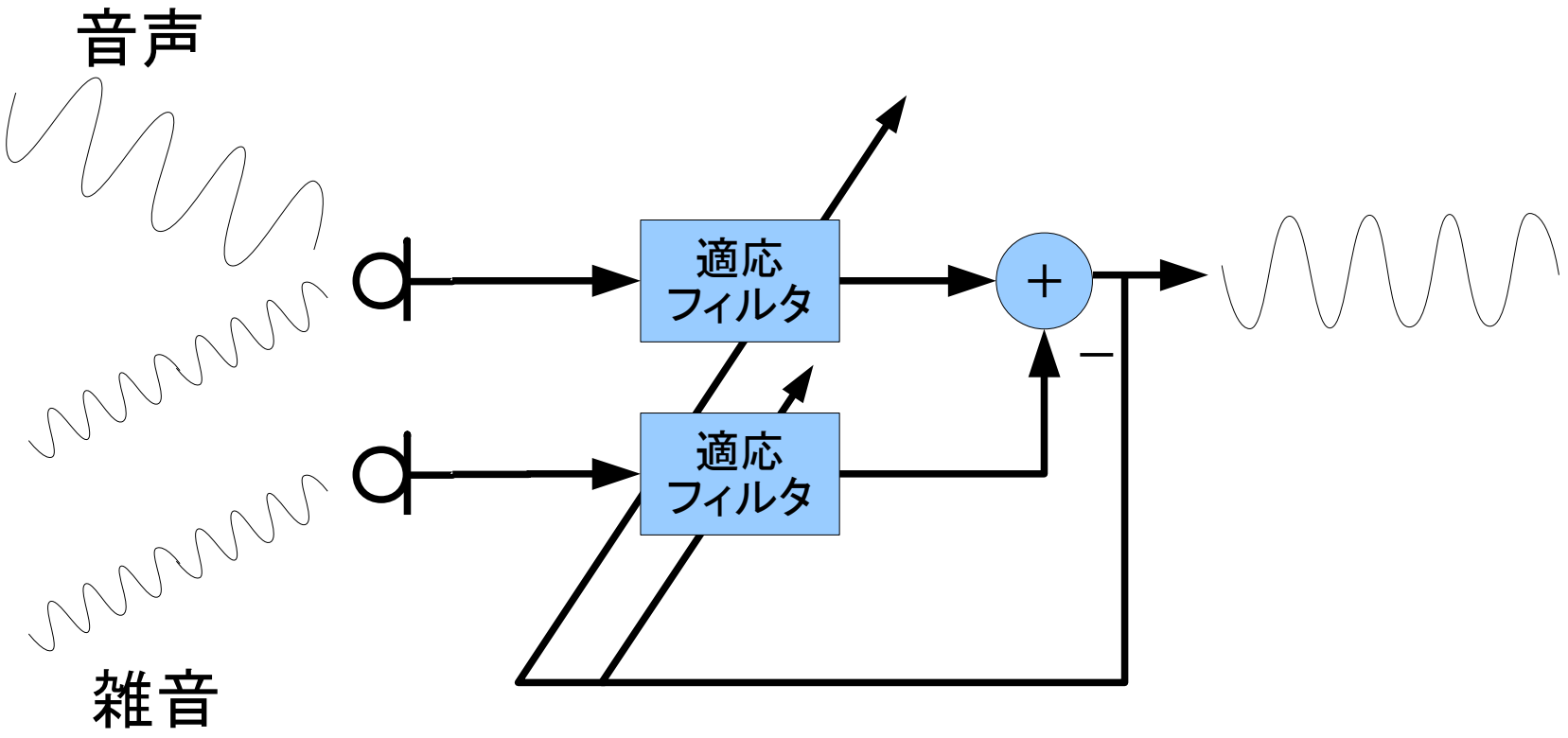
減算アレイ

- マイクロホンアレイによって雑音を除去する（雑音の到来方向が既知の場合）



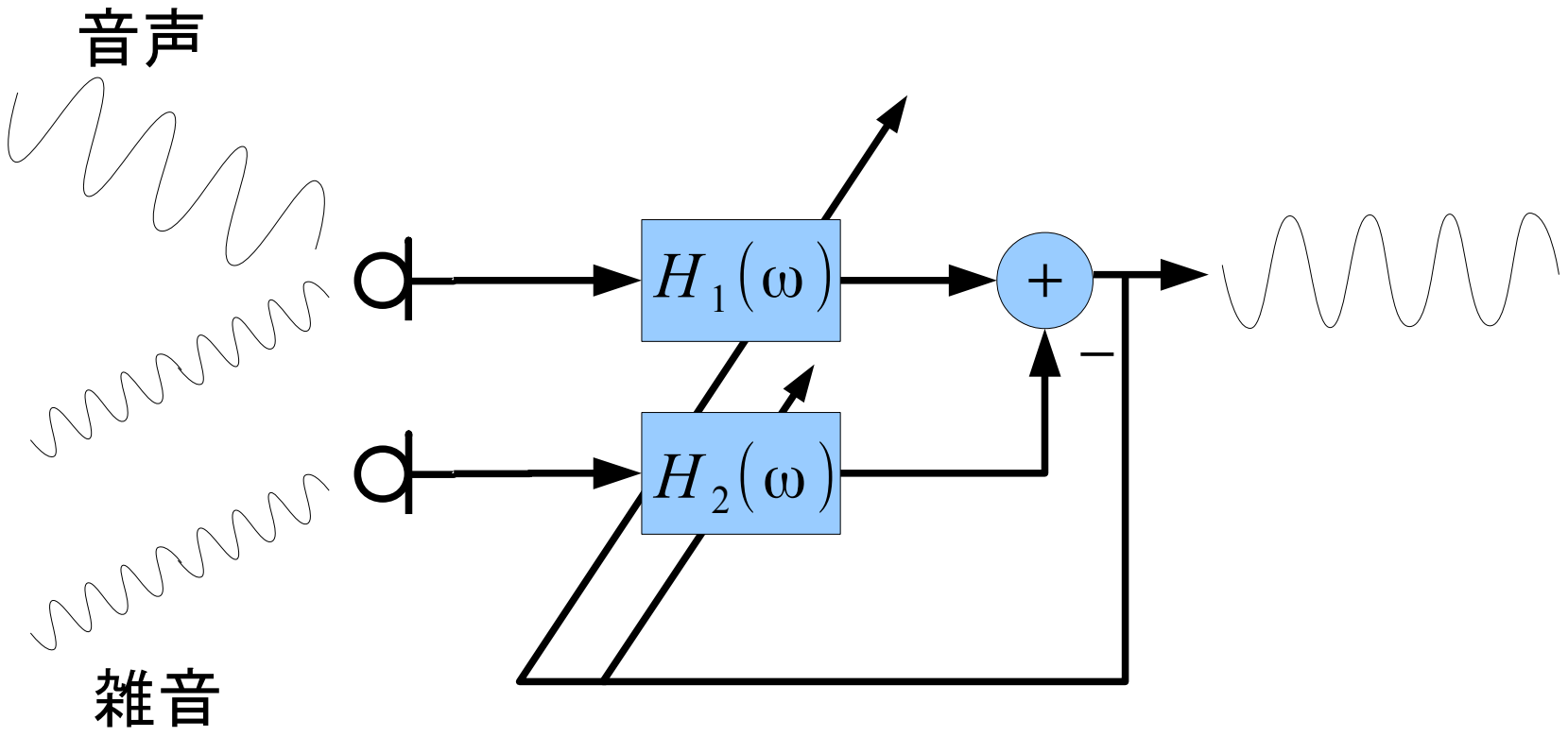
適応減算アレイ

- 雑音の到来方向が未知の場合
  - 適応フィルタによって雑音抑圧



適応減算アレイ

- フィルタに拘束条件が必要
  - そのままだと出力が単に0になる



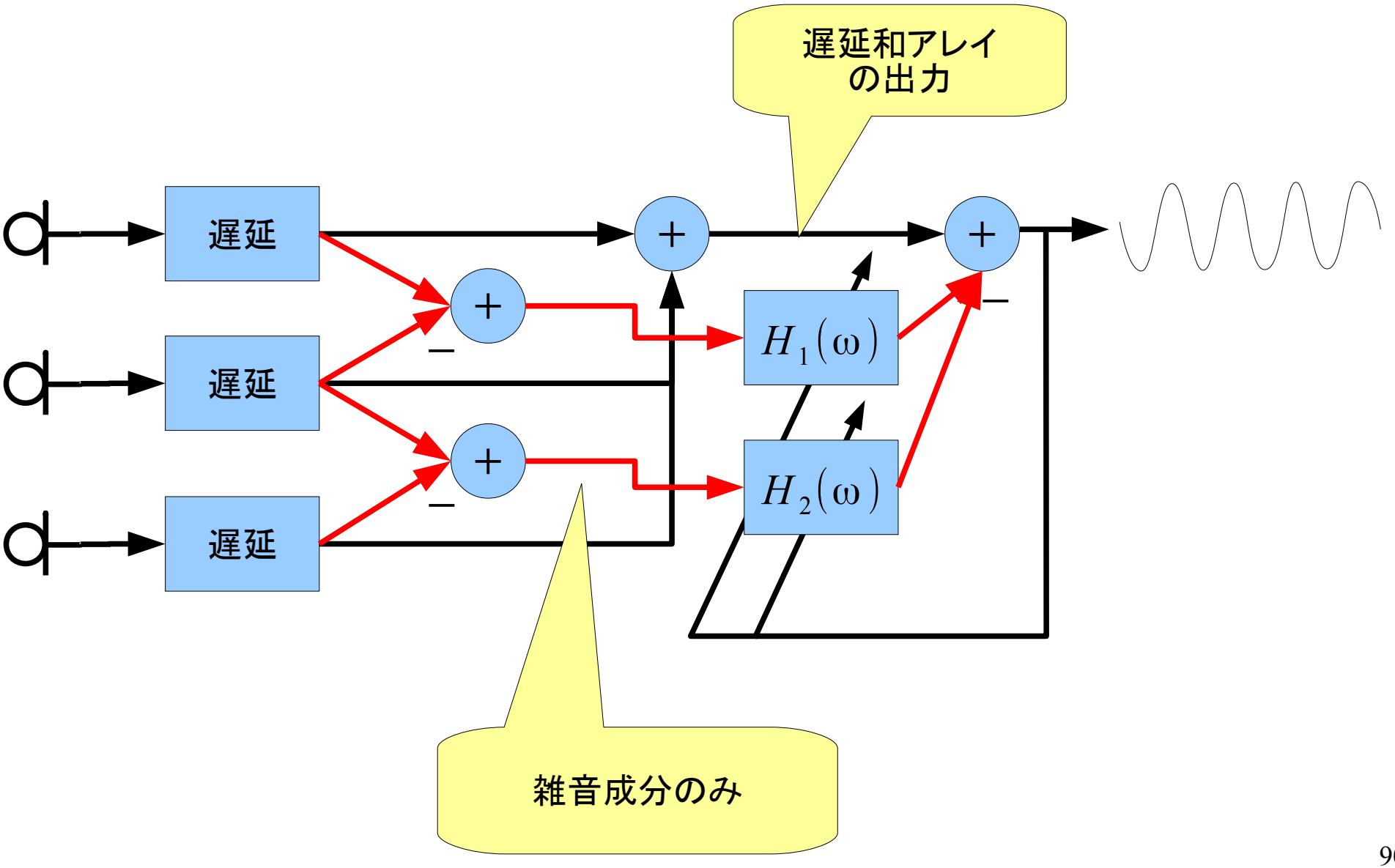
適応減算アレイ

- 適応フィルタ $H_i(\omega)$
- 音声からマイクロホンまでの伝達関数 $G_i(\omega)$
- フィルタの拘束条件

$$F(\omega)=\sum_i G_i(\omega)H_i(\omega)=1$$

- 実現例
  - Griffith-Jim型アレイ

Griffith-Jim型アレイ



２chスペクトル減算

- マイクロホンアレイによるスペクトル減算
  - 目的信号を除去して雑音信号を推定
  - 雑音信号のパワースペクトルを目的信号のパワースペクトルから減算
- 特徴
  - 非線形処理（適応フィルタは線形処理）
  - 雑音スペクトルをあらかじめ推定しておく必要がない
  - 時間的に変化する雑音に対応できる

2chスペクトル減算