後方適応予測による
音声ロスレス符号化の研究

名古屋大学大学院 人間情報学研究科
社会情報学専攻 電子社会システム論講座

木村 敏幸

学籍番号 319802124

背景

・音声のインターネットによる普及

・音声データは莫大な量を要する
(例)CD のデータ量…約 10MB/min
↓
現在の通信網では配信に時間がかかる

・音声符号化…音声データを圧縮すること
(例) MD、MP3(MPEG1 audio layer 3)
↓
符号化によってデータの劣化が生じる

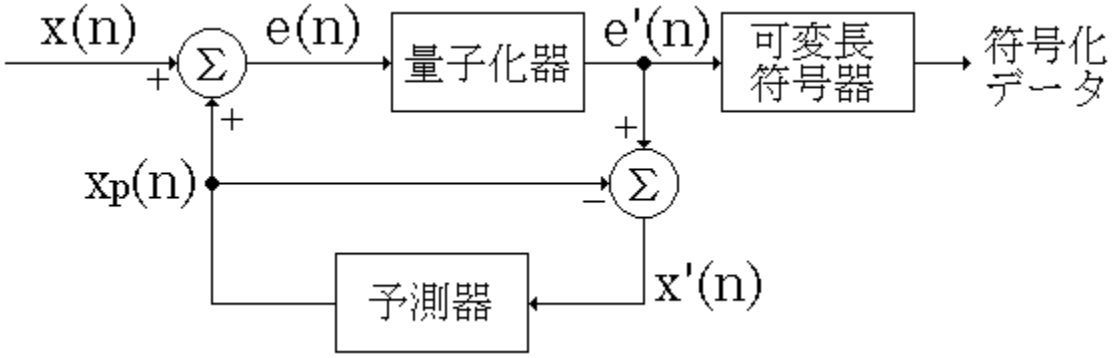
・ロスレス符号化…データの劣化が生じない
(例) LZH、ZIP など
↓
音声データには不向きである

目的

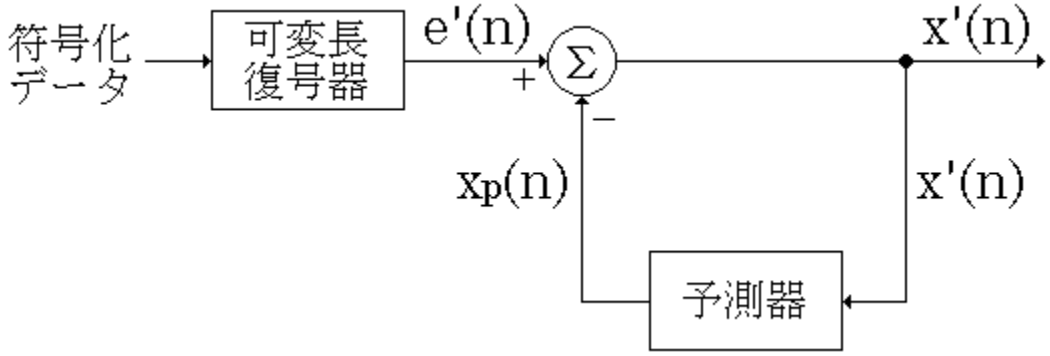
予測符号化による音声のロスレス符号化の研究

今回用いた予測符号化方式…ADPCM
(Adaptive Differential Pulse Code Modulation)

従来の ADPCM 符号化回路



(ENCODE)



(DECODE)

$x(n)$：入力信号、$x'(n)$：出力信号

$e(n)$：予測誤差、$e'(n)$：量子化予測誤差

$q(n)$：量子化誤差 $= e(n) - e'(n) = x(n) - x'(n)$

$x_p(n)$：線形予測値 $= \sum_{i=1}^{L} a_i x'(n-i)$、$a_i$：予測係数

量子化誤差を 0 にする

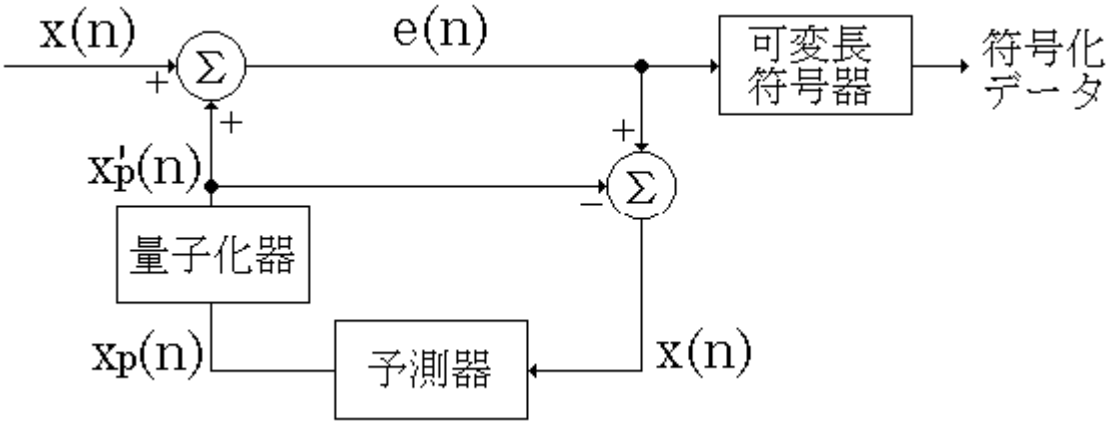
↓

予測係数の値が制限される

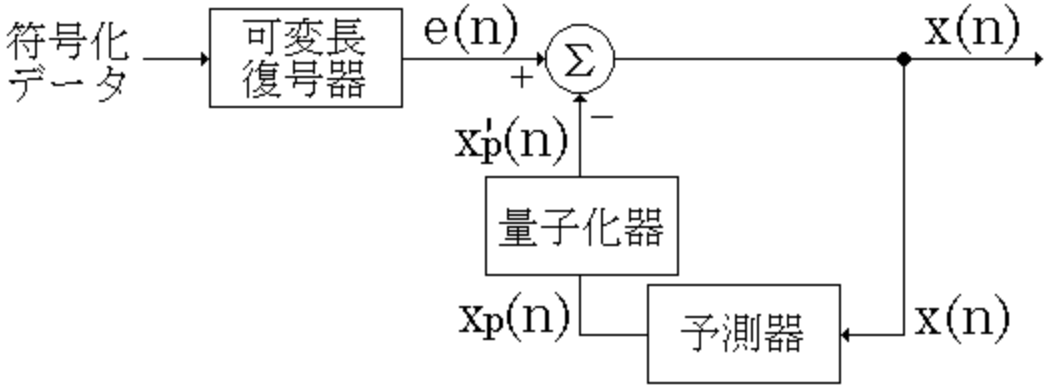
↓

圧縮効率が悪くなる

今回用いたロスレス符号化回路



(ENCODE)



(DECODE)

$x_p(n)$：線形予測値 $= \sum_{i=1}^{L} a_i x(n-i)$

$x'_p(n)$：量子化線形予測値 $= \lfloor x_p(n) + 0.5 \rfloor$

$\lfloor x \rfloor = x$以下の最も大きな整数

長所

予測係数の最適値を用いることができる

↓

より良い圧縮効率を見込むことができる

予測係数の更新

予測係数の最適値は以下の連立方程式で算出

$$\begin{pmatrix} R(0) & R(1) & \cdots & R(L-1) \\ R(1) & R(0) & \cdots & R(L-2) \\ \vdots & \vdots & \ddots & \vdots \\ R(L-1) & R(L-2) & \cdots & R(0) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_L \end{pmatrix} = -\begin{pmatrix} R(1) \\ R(2) \\ \vdots \\ R(L) \end{pmatrix}$$

$$R(\tau):\text{自己相関関数} = E[x(n)x(n+\tau)]$$

Durbin 法…連立方程式の高速計算法

$$(i=1)\begin{cases} w^{(1)} = r(1) \\ a_1^{(1)} = -w^{(1)} \\ u^{(1)} = 1-\{w^{(1)}\}^2 \end{cases}$$

$$(i=2\sim L)\begin{cases} w^{(i)} = \dfrac{r(i)+\sum_{j=1}^{i-1} a_j^{(i-1)} r(i-j)}{u^{(i-1)}} \\ a_j^{(i)} = a_j^{(i-1)} - w^{(i)} \times a_{(i-j)}^{(i-1)} \quad (j=1,\cdots,i-1) \\ a_i^{(i)} = -w^{(i)} \\ u^{(i)} = (1-\{w^{(i)}\}^2)u^{(i-1)} \end{cases}$$

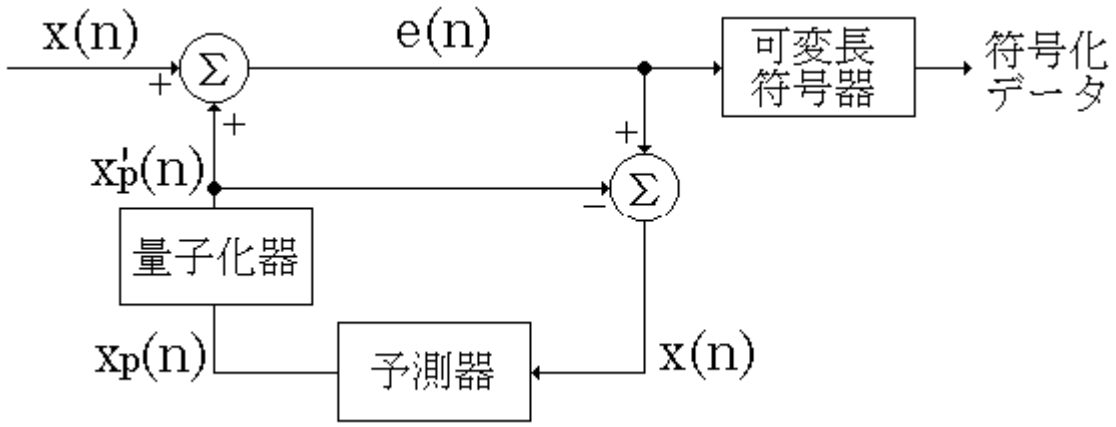
$$r(\tau):\text{自己相関係数} = R(\tau)/R(0)$$

$$u:\text{二乗平均誤差} = E[\{e(n)\}^2]$$

$$w:\text{PARCOR係数}(|w|<1\text{の時安定})$$

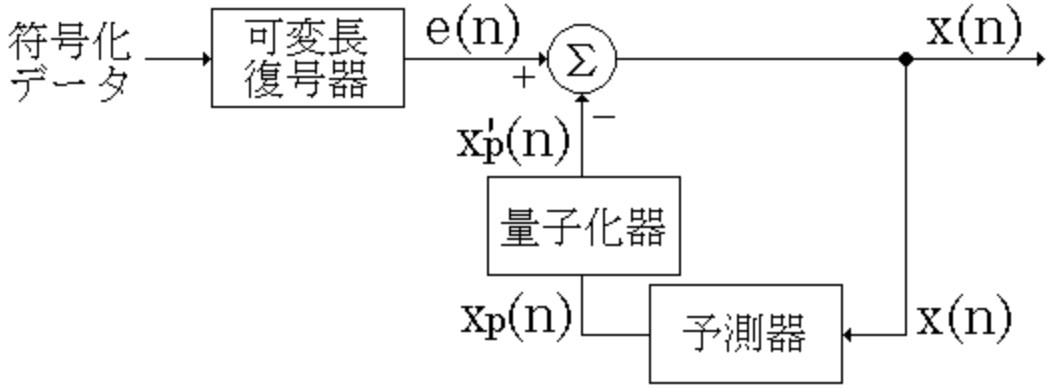
後方適応予測の採用

(ENCODE、i=1～L)



① 過去の出力信号(=入力信号)より r(i)を計算
② Durbin 法によって a_i を算出
③ x(n-i)と a_i を元に $x_p(n)$を予測
④ $x_p(n)$を丸め込み、$x'_p(n)$を生成
⑤ x(n)を入力
⑥ x(n)＋$x'_p(n)$によって e(n)を計算
⑦ e(n)を符号化データに変換し、送信
⑧ 時間をシフトし、①から繰り返す。

(DECODE、i=1～L)



① 過去の出力信号より r(i)を計算
② Durbin 法によって a_i を算出
③ x(n-i)と a_i を元に $x_p(n)$を予測
④ $x_p(n)$を丸め込み、$x'_p(n)$を生成
⑤ 符号化データを e(n)に変換
⑥ $e(n)-x'_p(n)$によって x(n)を計算
⑦ x(n)を出力
⑧ 時間をシフトし、①から繰り返す。

長所

e(n)しか送信する必要がない

↓

より良い圧縮効率を見込むことができる

可変長符号化

e(n)の統計分布はラプラシアン分布に近似される

ラプラシアン分布の一般式

$$p(x) = \frac{1}{\sqrt{2\sigma^2}} \exp\left(-\sqrt{\frac{2}{\sigma^2}} |x| \right)$$

$$E[\ ] = \int_{-\infty}^{\infty} p(x)dx = 1$$

$$E[x] = \int_{-\infty}^{\infty} xp(x)dx = 0$$

$$E[x^2] = \int_{-\infty}^{\infty} x^2 p(x)dx = \sigma^2$$

符号長は$-\log_2 p(x)$に収束する

$$-\log_2 p(x) = -\log_2 \left\{ \frac{1}{\sqrt{2\sigma^2}} \exp\left(-\sqrt{\frac{2}{\sigma^2}} |x| \right) \right\}$$

$$= \sqrt{\frac{2}{\sigma^2}} \cdot \frac{|x|}{\ln 2} + \log_2 \sqrt{2\sigma^2}$$ ：y軸対称の直線

↓

符号長が直線的に変化する
コード表を作成することが必要

今回用いたコード表

SHORTEN のコード表の改良版

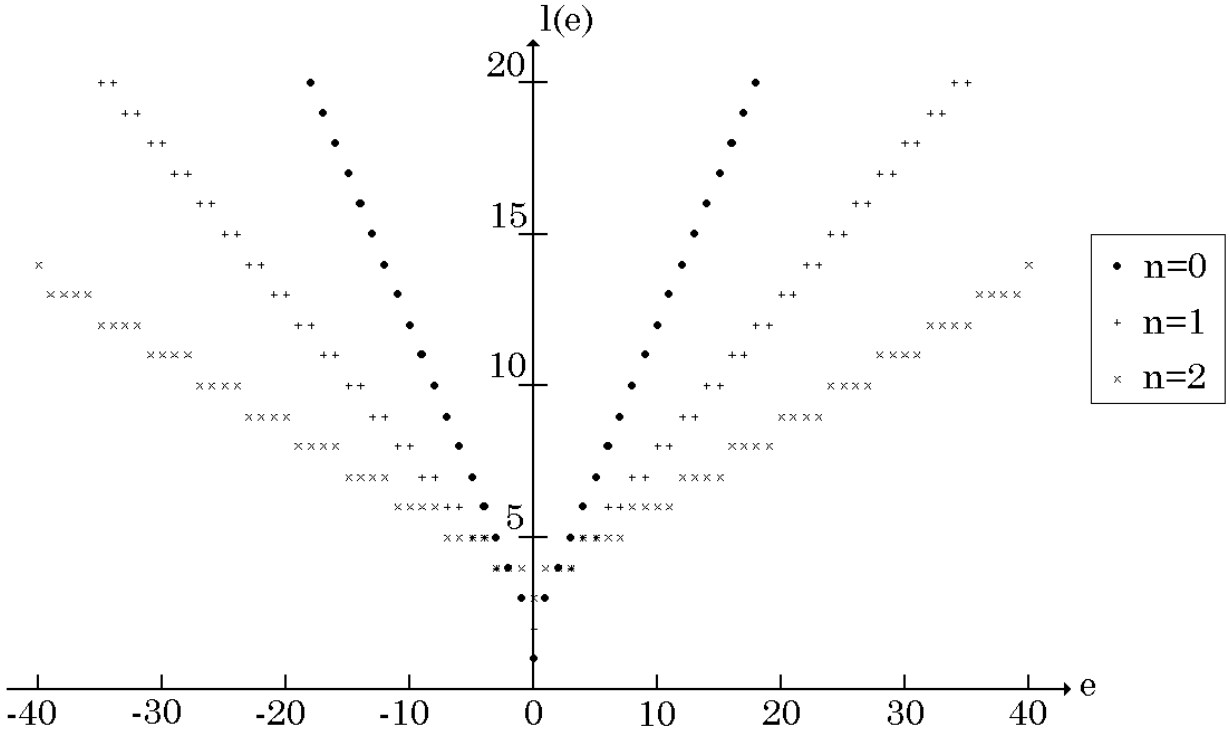
codeword (e)=

upper bit	number of '0's	end bit	sign bit
$\|e\| \bmod 2^n$ を２進表記	$\|e\| ¥ 2^n$の数だけ0を連ねる	1	0(e>0) 1(e<0)
n	$\|e\| ¥ 2^n$	1	0 or 1

A￥B：A÷Bの商

A mod B：A÷Bの余り

符号長の変化グラフ



n を決定するための条件

$$
\begin{aligned}
P &= \int_{-2^n}^{2^n} p(x)dx \\
&= \int_{-2^n}^{2^n} \frac{1}{\sqrt{2\sigma^2}} \exp\left(-\sqrt{\frac{2}{\sigma^2}} |x| \right) dx \\
&= -\exp\left(-\sqrt{\frac{2}{\sigma^2}} 2^n \right) + 1
\end{aligned}
$$

P：閾値$(0<P<1)$

ゆえに、

$$
n = \log_2 \left(-\ln(1-P) \sqrt{\frac{\sigma^2}{2}} \right)
$$

n の変更の手順

① $n = \log_2 \left(-\ln(1-P) \sqrt{\frac{\sigma^2}{2}} \right)$を計算

② nを四捨五入

③ $n < 0$の時には$n = 0$

実験概要

WAV ファイルを圧縮するプログラムを作成
(DOS 形式)

・コンピュータ:GATEWAY2000 G6-233
　CPU：Pentium　Ⅱ(233MHz)
　Memory：64MB
　OS：Windows 95

・プログラム言語:C 言語(ANSI 準拠)

・コンパイラ:Microsoft Visual C++ 6.0

・サンプル:Windows OS のサウンドファイル
- Chimes.wav (15932 Byte, 15876 Sample)
- Chord.wav (24994 Byte, 24938 Sample)
- Ding.wav (11586 Byte, 11530 Sample)
- Tada.wav (27516 Byte, 27549 Sample)

・サウンドフォーマット(すべてのサンプル)
- サンプリング周波数 22.05kHz
- 量子化ビット 8 bit
- チャンネル mono

圧縮率に関係するパラメータ

・タップ数
　線形予測値 $x_p(n)$を計算する時に用いる過去の出力信号列の数

・ブロック数
　自己相関係数 $r(\tau)$を計算する時に用いる過去の出力信号列の数

・コード表の閾値
　コード表の n の値を変化させるときに用いる

・更新間隔
　予測係数を更新し直すまでに入力される入力信号の数

今回用いたパラメータの値

	Type 1	Type 2	Type 3
タップ数	8	16	24
ブロック数	512	768	1024
コード表の閾値	0.2	0.2	0.2
更新間隔	20	20	20

実験結果

・圧縮率

圧縮率の比較表(%単位)

	Type 1	Type 2	Type 3	Shorten	LTAC
Chimes	34.17	32.41	31.42	42.41	32.26
Chord	29.74	30.63	30.89	36.20	31.54
Ding	28.08	27.18	26.79	41.27	27.46
Tada	51.84	51.91	49.76	54.40	50.01
Average	37.98	37.80	36.89	44.43	37.44

Type 1,2,3 とも Shorten より圧縮率が良い
Type 3 は LTAC よりも圧縮率が良い

・符号化時間

符号化時間の比較表(sec 単位)

	Time	Type 1	Type 2	Type 3
Chimes	0.72	0.22	0.33	0.49
Chord	1.13	0.38	0.54	0.83
Ding	0.52	0.22	0.32	0.39
Tada	1.25	0.44	0.77	0.93
Total	3.62	1.26	1.96	2.64

サンプルの演奏時間よりも符号化時間が短い
→リアルタイム符号化及び復号化が可能

まとめ

後方適応予測による音声ロスレス符号化の実行
↓
従来の符号化ツールより良い性能が得られた

リアルタイム符号化及び復号化が可能
↓
将来的な実用化が可能

今後の展望

・復号化プログラムの作成

・他の音声フォーマットでの検証

CD の音声フォーマット
- サンプリング周波数 44.1kHz
- 量子化ビット16 bit
- チャンネル stereo

・実用的な符号化及び復号化ツールの作成