

音声信号の可逆符号化における線形予測器の最適化に関する検討

A Study on Optimization of Linear Predictors for Lossless Audio Coding

川邊 博史*

松田 一朗*

伊東 晋*

Hiroshi KAWABE

Ichiro MATSUDA

Susumu ITOH

1. はじめに

近年、高品質な音響データを扱うアプリケーションを中心に、再生波形に歪を許容しない可逆符号化技術への関心が高まっている。現在実用化されている可逆符号化方式の多くは、適応線形予測とエントロピー符号化の組み合わせを基本構成としている。適応線形予測は、符号化対象信号そのものを使って予測器を設計する Forward 適応予測と、符号化済みの過去信号のみを用いて送受信両端で予測器を設計する Backward 適応予測に分類できる [1]。いずれの場合も、予測器の設計は予測誤差電力を最小化する MMSE 規範に基づいて行われるのが一般的である。しかし可逆符号化の目的である符号量の削減という観点において、MMSE 規範が最良の結果を与えとは限らない [2]。そこで本稿では、符号化レートを見積もるコスト関数の下で線形予測器を反復的に最適化する手法について検討する。

2. 符号化手順

本検討では、画像信号用に開発されたブロック適応予測 [2] に基づいた符号化手順を採用する。符号化対象は 16 bits/sample のモノラル音声信号であり、これを 65536 サンプル毎に区切ったフレーム単位で独立に符号化する。

2.1 ブロック適応予測

提案手法では、国際標準方式である MPEG-4 ALS [3] と同様に L サンプルからなるブロック単位で線形予測およびエントロピー符号化の適応化を図っている。但し、ブロック毎に Forward 適応予測に基づいて予測器を設計すると、予測係数に関する付加情報が増大してしまうため、本稿ではフレーム全体で利用できる予測器の種類を M 通りに限定し、ブロック毎に予測器の番号 $m \in \{1, 2, \dots, M\}$ を指定するものとする。ここで、時刻 t のサンプル $x(t)$ が属するブロックに対して、 m 番目の予測器が割り当てられたとき、当該サンプルにおける予測誤差は次式で表せる。

$$e(t) = x(t) - \left[\sum_{k=1}^K a_m(k) \cdot x(t-k) + 0.5 \right] \quad (1)$$

但し、 $a_m(k)$ は m 番目の予測器に関する予測の重み (予測係数)、 K は予測に用いる過去サンプルの数 (予測次数) である。

2.2 予測誤差の算術符号化

予測誤差のエントロピー符号化には、多値算術符号器の一種であるレンジコーダ [4] を採用する。実際の符号化に際して必要となる予測誤差の確率モデルは、予測器と同様フレーム全体で 16 種類用意しておき、ブロック毎にモデルの番号 $n \in \{1, 2, \dots, 16\}$ を指定する。ここで、 n はブロック内の予測誤差の分散値 σ_n^2 に対応付けられており、その分布形は次式で定義される。

$$P_n(e) = \begin{cases} \alpha_n \cdot \exp\left(-\left|\sqrt{\frac{\Gamma(3/c_n)}{\Gamma(1/c_n)}} \cdot \frac{e}{\sigma_n}\right|^{c_n}\right) & (|e| < E_n) \\ \alpha_n \cdot \delta & (\text{otherwise}) \end{cases} \quad (2)$$

但し、 $\Gamma(\cdot)$ はガンマ関数、 α_n は確率の和を 1 にする正規化係数、 c_n は分布形の急峻さを表す形状パラメータである。上式は、予測誤差の絶対値が一定範囲内 ($|e| < E_n$) のとき、その生起確率を一般化ガウス関数でモデル化し、それ以外のときは十分小さい定数 δ で近似することを意味している。本稿で扱う 16 bits/sample の音声信号では、予測誤差のとり得る範囲が ± 65535 と大きくなるため、このように生起確率が比較的高い 0 付近の範囲のみを高精度にモデル化することで、 $P_n(e)$ の値をテーブル参照する際に必要なメモリ容量の増大を抑えている。

3. 予測器の最適化

提案手法においてフレーム毎に必要な付加情報は以下の通りである。

- 予測係数 $a_m(k)$ ($M \times K$ 個)
- 予測器の選択情報 m (65536/ L 個)
- 確率モデル (分散) の選択情報 n (65536/ L 個)
- 確率モデルの形状パラメータ c_n (16 個)

予測器の設計は、上記の $a_m(k)$ の最適な組み合わせを決定することに相当する。一般に予測誤差の分布形が不変とみなせるとき、そのエントロピーは分散に対して単調増加となるため、分散の最小化、すなわち MMSE 規範によって符号化レートも最小化できる。しかし、提案手法では各予測器を設計する対象領域 (同一の m を割り当てたブロックの集合) の中に、異なる分布形のモデル ($P_n(e)$) が混在しているため、上記の前提条件は一般に成り立たない [2]。そこで提案手法では、フレーム全体の符号量を以下のコスト関数で見積ると共に、予測係数

*東京理科大学 理工学部 電気電子情報工学科

表 1 符号化レートの比較 [bits/sample]

ファイル名	提案手法	MMSE	FLAC	ALS (default)	ALS (RLSLMS)
41_30sec	10.112	10.275	10.394	10.312	9.919
ATrain	8.194	8.272	8.819	8.528	7.788
Blackwater	9.549	9.679	9.943	9.637	9.037
FloorEssence	10.660	10.831	10.956	10.882	10.455
Layla	10.047	10.135	10.405	10.265	9.734
MidnightVoyage	9.459	9.554	9.894	9.714	9.178
TheSouece	9.446	9.575	9.743	9.623	9.255
Waiting	10.733	10.833	11.099	10.967	10.493
Average	9.775	9.894	10.157	9.991	9.482

表 2 復号時間の比較 [sec.]

提案手法	平均復号時間
提案手法	4.22
FLAC	0.06
ALS (default)	0.12
ALS (RLSLMS)	23.70

だけでなく上に列挙した4種類全てのパラメータについて、このコスト関数に基づいて繰り返し最適化を図る。

$$J = - \sum_t \log_2 P_n(e(t)) + B_{side} \quad (3)$$

以下に具体的な処理手順を示す [2]。

- (1) MMSE 規範に基づいて予測係数 $a_m(k)$ と予測器の割り当て m の初期値を決定する。
- (2) 乱数で選んだ2つの予測係数 $a_m(i)$, $a_m(j)$ に対し、微小な修正項 Δa_i , Δa_j を加えた時のコスト J を評価する。この処理を各予測器 (m) 毎に一定回数繰り返し、 J の値が減少する方向に予測係数を徐々に修正する。
- (3) ブロック毎に最適な確率モデル (n) を選択する。
- (4) 各確率モデル $P_n(e)$ について、最適な形状パラメータ c_n を16通りの値 $\{0.2, 0.4, \dots, 3.2\}$ の中から選択する。
- (5) ブロック毎に最適な予測器の割り当て (m) を決定する。
- (6) (2) から (5) までの手順を、コスト関数 J の値が収束するか、または反復回数の上限 (本稿では100回) に達するまで繰り返す。

4. 特性評価

ベンチマークサイト <http://www.rjamorim.com/test/> からダウンロードした音楽ファイルの1チャンネル分 (サンプリング周波数 44.1 kHz, 19~30 秒) を対象に符号化シミュレーションを実施した。但し、ブロックサイズおよび予測器の数と予測次数についてはそれぞれ $L = 64$, $M = 9$, $K = 40$ と設定した。ここで、「MMSE」は提案手法において3.で述べたパラメータの最適化手順のみをMMSE 規範に基づいて実行した場合の特性を示している。一方「FLAC」はオープンソフトウェアのロスレス符号化方式 [5]、「ALS (default)」および「ALS (RLSLMS)」は、それぞれ国際標準方式である MPEG-4 ALS [3] の参照ソフトウェア (RM17) [6] をデフォルトの設定および Backward 適応予測に基づいた RLSLMS モード (オプ

ション -z3 -p -b) で実行した場合の符号化結果を表している。なお、実験には AMD Athlon プロセッサ 2.2GHz、メモリ 512MB を搭載した計算機を使用した。

表 1 において、提案手法は「MMSE」よりも平均で 0.02 bits/sample 低い符号化レートを示しており、可逆符号化において MMSE 規範が最適ではないことを実証する結果となっている。一方、提案手法は Forward 適応予測に基づいた「FLAC」、「ALS (default)」よりも低い符号化レートを達成しているものの、Backward 適応予測に基づいた「ALS (RLSLMS)」の符号化効率には劣る結果となった。しかし、Backward 適応予測では復号器においても予測器の設計を行う必要があるため、表 2 に示すように「ALS (RLSLMS)」による実時間の復号処理は、実験環境において困難であることが判明している。今後は適応予測の精度を改善すると共に、付加情報の効率的な符号化手法について検討を加え、更なる性能の向上を図る予定である。

【参考文献】

- [1] C.D. Giurcaneanu et al.: Proc. 12th International Conference on Control Systems and Computer Science (CSCS-12), pp. 396-401, May 1999.
- [2] 松田 他: 信学論 (D-II), Vol. J88-D-II, No.9, pp. 1798-1807, Sep. 2005.
- [3] T. Liebchen et al.: Proc. IEEE Data Compression Conference (DCC 2004), pp. 439-448, Mar. 2004.
- [4] 奥村晴彦: C Magazine, Vol.14, No.7, pp.13-35, July 2002.
- [5] <http://flac.sourceforge.net/>
- [6] <http://www.nue.tu-berlin.de/research/projects/lossless/mp4als.html>