

Research on stereophonic acoustic echo canceller with both high sound quality and fast convergence speed

| | |
|-------|---|
| メタデータ | 言語: jpn 出版者: 公開日: 2019-12-06 キーワード (Ja): キーワード (En): 作成者: Hirano, Akihiro メールアドレス: 所属: |
| URL | https://doi.org/10.24517/00056432 |

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 International License.



高音質・高速収束ステレオエコーキャンセラに関する研究

18560371

平成 18 年度～平成 19 年度科学研究費補助金
(基盤研究 (C)) 研究成果報告書

平成 20 年 5 月

研究代表者 平野 晃宏
金沢大学大学院自然科学研究科講師

金沢大学附属図書館



1300-05172-8

<はしがき>

本報告書は、平成 18 年度～平成 19 年度の 2 年間実施した科学研究費補助金（基盤研究（C））「高音質・高速収束ステレオエコーキャンセラに関する研究」の成果をまとめたものである。

研究組織

研究代表者 平野晃宏（金沢大学大学院自然科学研究科講師）
 研究分担者 中山謙二（金沢大学大学院自然科学研究科教授）

交付決定額（配分額）

（金額単位：円）

| | 直接経費 | 間接経費 | 合計 |
|----------|-----------|---------|-----------|
| 平成 18 年度 | 1,800,000 | 0 | 1,800,000 |
| 平成 19 年度 | 1,200,000 | 360,000 | 1,560,000 |
| 総計 | 3,000,000 | 360,000 | 3,360,000 |

研究発表

(1) 雑誌論文

無し。

(2) 学会発表

- 水野 裕介, 平野 晃宏, 中山 謙二, “ステレオエコーキャンセラにおける前処理を用いない最適値推定” 電子情報通信学会 第 22 回信号処理シンポジウム, 2007 年 11 月 8 日, 仙台
- 平野 晃宏, 中山 謙二, “ステレオ音響エコーキャンセラの Intel IA-32 CPU による実現” 電子情報通信学会 第 22 回信号処理シンポジウム, 2007 年 11 月 8 日, 仙台

(3) 図 書

無し。

研究成果による産業財産権の出願・取得状況

無し。

目次

| | |
|---------------------------------|----|
| 研究成果報告書概要 | 1 |
| 2 種類の SAEC を組み合わせた高速収束アルゴリズムの開発 | 5 |
| 実時間信号処理システムの構築 | 17 |
| Intel IA32 プロセッサを用いた SAEC の実現 | 25 |
| 入力信号の白色化による高速収束アルゴリズムの検討 | 35 |
| 参考文献 | 47 |

研究成果報告書概要

目次

| | | |
|---|----------------|---|
| 1 | 研究の背景と目標 | 2 |
| 2 | 研究項目 | 3 |

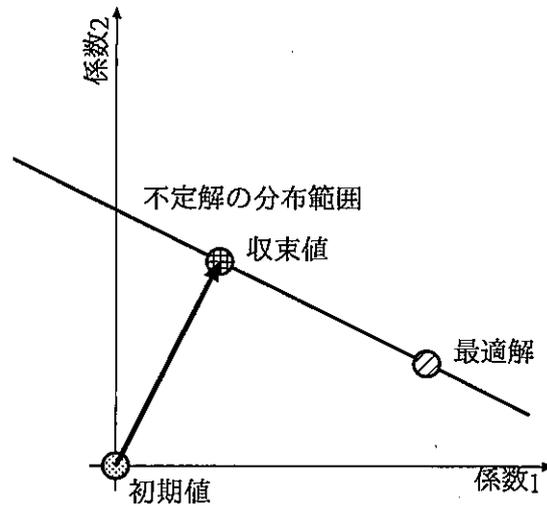


図1 係数不確定性の問題

1 研究の背景と目標

インターネットに代表される情報ネットワークの普及やコンピュータによるマルチメディア利用技術の進展により、映像、音声、データなどを統合したマルチメディア遠隔会議システム需要が高まっている。特に、通信容量や処理速度の向上にともない、臨場感のあるシステムが強く求められている。臨場感を向上させるには音声の多チャンネル化が不可欠である。そのため、多チャンネル音声を用いた遠隔会議システムが検討されている。

遠隔会議においては快適な会話の妨げになる音響エコーが問題であるため、多チャンネル音声に対応したステレオ音響エコーキャンセラ (SAEC) の研究が進められている。SAECにおける最大の問題は、係数不確定性の問題である [1, 2]。図1に示すように無数の不定解が存在し、最適値に収束する補償がない。我々は、この問題を世界に先駆けて指摘し [1]、SAECの研究が活発化させる契機となった。

この問題を解決する手法としては、図2に示すように前処理を用いてスピーカから再生される音声を変形させる方法 [2-6] が主流となっている。我々も、前処理方式の提案 [4]、解析 [7] や改良 [3] を行なっている。

これらの方法では、スピーカで再生する音声信号を前処理によって変化させるため音声品質が劣化する。特に、音楽信号における音像定位に致命的な悪影響を及ぼすことが指摘されている。そのため、多チャンネル化による臨場感向上効果は損なわれ、実用化の大きな妨げとなっている。

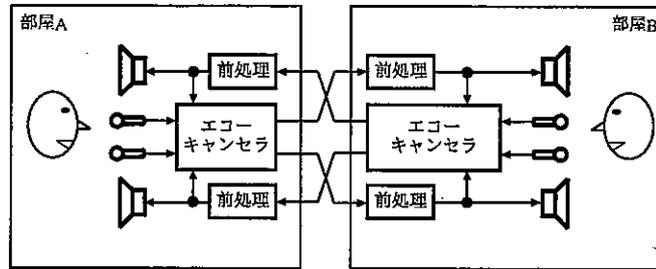


図2 前処理を用いた SAEC

そこで、我々は、前処理を用いないでフィルタ係数を最適値に収束させる分割学習法を提案した [8,9]。また、ハードウェア実現も検討している [10]。しかし、分割学習法には、学習速度が十分ではないという問題がある。

本研究の目的は、音質を劣化させる前処理を用いず、かつ、フィルタ係数が最適値に高速に収束する SAEC を開発することである。

2 研究項目

1. 2種類の SAEC を組み合わせた高速収束アルゴリズムの開発

不定解の分布解析を行った。時間領域においては、多次元空間での解析が必要であり、実用的とはいえない。周波数領域においては、2個の互いに異なる不定解を推定すれば、不定解全ての分布範囲を特定できることが判明した。4元連立方程式を解くことにより、不定解分布の理論式を特定できる。さらに、2人の話者に対する不定解が求められると、最適解を推定できることが判明した。4元連立方程式を解くことにより、周波数領域における最適解を推定できる。

2個の SAEC を用いて最適解を推定する方式を開発した。初期値の異なる2個の SAEC を用いて、2個の不定解を同時に推定する。その際、独立性の高い不定解を得るためには、適切な初期値を設定する必要があることが判明した。独立性の低い不定解の組を用いると、最適値の推定精度が劣化する。この問題を解決するために、定期的に不定解の独立性を判定し、独立性が低い場合には初期値を再設定する方式を開発した。周波数領域に変換して不定解間の距離を測定し、閾値以下である場合には初期値を再設定する。初期値としては現在の推定ベクトルと直交するものを選ぶことにより、独立性を高める。

初期値の再設定により、最適値の推定精度を高めることができた。しかし、まだ十分なエコー消去量が得られないため、さらなる改善方法を検討している。

2. 実時間信号処理システムの構築

最新のデジタルシグナルプロセッサ (Digital Signal Processor, DSP) を用いた実時間信号処理装置を導入した。この DSP に適した SAEC の実現方法を検討している。

3. Intel IA32 プロセッサを用いた SAEC の実現

PC ベースの遠隔会議システムを想定して、Intel IA32 プロセッサを用いて SAEC を実現した。ベクトル演算命令を有効に利用できるような種々の最適化を行い、最適化前のプログラムに対して約 4 倍の高速化に成功した。比較的低価格の PC でも SAEC の実時間処理が可能になるため、SAEC を用いた高品位遠隔会議システムを低価格で容易に導入可能になる。

ベクトル演算命令を用いた高速化を行う際に、通常は連続する 4 タップを 4 次元ベクトルとしてベクトル化を行う。しかし、この方式では 4 ワード境界を超えたデータアクセスが必要になり、データ配置問題を引き起こす。今回は、4 個の適応フィルタを用いるという SAEC の構成に着目し、4 個のフィルタのタップ 4 個を 4 次元ベクトルとしてベクトル化を行った。この構成では、データ配置問題は発生しない。

循環バッファの実装方法についても検討した。専用のアドレス発生器を持たない汎用プロセッサにおいては、条件分岐命令によるパイプラインの効率低下を避けるために、循環バッファサイズより大きな配列を用意することが多い。分岐予測が可能であること、分岐ユニットが並列動作することに注目すると、条件分岐の影響は少ないことが予想される。実験により、条件分岐の影響は非常に小さいことを確認した。

4. 入力信号の白色化による高速収束アルゴリズムの検討

入力信号を白色化することによって、有色信号に対する収束速度を改善する方法を検討した。ラチス形予測器を用いた方法を提案してきたが、予測係数とフィルタ係数の同期が必要であり、同期のために演算量が増加するという問題があった。白色化フィルタの位置を見直すことにより、同期が不要な構成を検討した。タップ数が少ない場合には有効であるが、タップ数が多いと大きな改善は見られないという点が課題である。

2 種類の SAEC を組み合わせた高速収束アルゴリズムの開発

目次

| | | |
|-----|---------------|----|
| 1 | ステレオエコーキャンセラ | 6 |
| 1.1 | 基本構成 | 6 |
| 1.2 | 係数不確定性の問題 | 6 |
| 2 | 周波数域における最適値推定 | 7 |
| 2.1 | 原理 | 7 |
| 2.2 | 初期値の与え方 | 10 |
| 2.3 | 話者交代の判別 | 11 |
| 3 | シミュレーション | 13 |
| 3.1 | シミュレーション条件 | 13 |
| 3.2 | シミュレーション結果 | 13 |
| 4 | まとめ | 15 |

1 ステレオエコーキャンセラ

1.1 基本構成

図3に最も一般的なSECである線形結合形 [11] の構成を示す。図3の右側の部屋 (room B) のエコーに注目し、2つのチャンネルの定義は左側の部屋 (room A) を基準として行なう。この際、roomAのタップ数を N_A 、roomBのタップ数を N_B とする。

SECは基本的には4つのエコーパスに1対1で対応する4個の適応フィルタから成る。これらは、図3の右側の部屋 (room B) の左側のスピーカーから左側のマイクに至るパス $H_{LL}(z)$ 、左側のスピーカーから右側のマイクに至るパス $H_{LR}(z)$ 、右側のスピーカーから左側のマイクに至るパス $H_{RL}(z)$ 、右側のスピーカーから右側のマイクに至るパス $H_{RR}(z)$ であり、モノラルエコーキャンセラの4倍の数の適応フィルタが必要となる。下付添字 L,R はそれぞれ "Left", "Right" チャンネルを表す。各適応フィルタはそれぞれ対応する各エコーパス特性を同定し、それによって音響エコーを除去する。

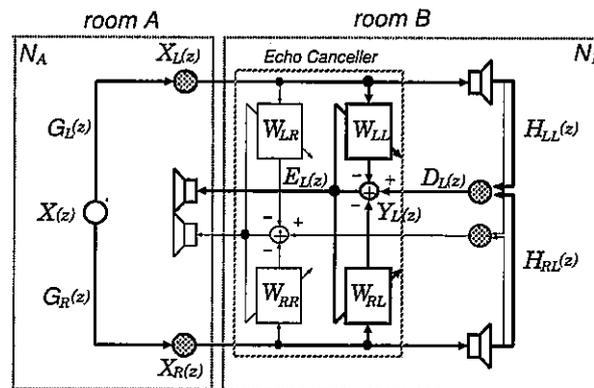


図3 ステレオエコーキャンセラシステムの構成

1.2 係数不確定性の問題

roomAで同時に2人以上が発話しないと仮定する。ここでは一つのマイク信号のエコーキャンセラだけを考慮するが、対称性より同じ議論がもう一方の経路に対してもそのまま成り立つ。

ここで入力信号の周波数特性を $X(z)$ 、roomAの伝達関数を $G_L(z), G_R(z)$ 、エコーパスの伝達関数を $H_{LL}(z), H_{RL}(z)$ 、適応フィルタの伝達関数を $W_{LL}(z), W_{RL}(z)$ とする。 $W_{LL}(z)$ は左スピーカーから左マイクに至るエコーパス $H_{LL}(z)$ を、 $W_{RL}(z)$ は右スピー

カーから左マイクに至るエコーパス $H_{RL}(z)$ をそれぞれ同定しエコーを消去する。
左右のマイク入力信号は、

$$X_L(z) = G_L(z)X(z) \quad (1)$$

$$X_R(z) = G_R(z)X(z) \quad (2)$$

となり、受信側のマイクに入力するエコー信号は、

$$D_L(z) = H_{LL}(z)X_L(z) + H_{RL}(z)X_R(z) \quad (3)$$

である。疑似エコー信号である適応フィルタの出力は、

$$Y_L(z) = W_{LL}(z)X_L(z) + W_{RL}(z)X_R(z) \quad (4)$$

である。従って、誤差信号は、

$$\begin{aligned} E_L(z) &= D_L(z) - Y_L(z) \\ &= [H_{LL}(z) - W_{LL}(z)]X_L(z) + [H_{RL}(z) - W_{RL}(z)]X_R(z) \\ &= \{[H_{LL}(z) - W_{LL}(z)]G_L(z) + [H_{RL}(z) - W_{RL}(z)]G_R(z)\}X(z) \end{aligned} \quad (5)$$

となる。これより、エコーが完全に消去されるためには、

$$\{[H_{LL}(z) - W_{LL}(z)]G_L(z) + [H_{RL}(z) - W_{RL}(z)]G_R(z)\}X(z) = 0 \quad (6)$$

が成立しなければならない。しかし、式 (6) より分かるように、 $E_L(z)=0$ となる適応フィルタの伝達関数 $W_{LL}(z), W_{RL}(z)$ の解は無限に存在し、一意には定まらないことが分かる。すなわち、最適解である $H_{LL}(z) = W_{LL}(z)$ 及び $H_{RL}(z) = W_{RL}(z)$ は必ずしも成立しない。従って、 $W_{LL}(z)$ 及び $W_{RL}(z)$ が無限に存在する最適解以外の解 (以後不定解と呼ぶ) に収束してしまう。これが SEC における係数不確定性の問題である [1, 2, 12, 13]。

2 周波数域における最適値推定

2.1 原理

ここでは周波数領域における最適値推定の原理について説明する。2 名の話者 α, β が交互に発言するものと仮定する。

まず式 (6) を次のように変形する。

$$[H_{LL}(z) - W_{LL}(z)]G_L(z) + [H_{RL}(z) - W_{RL}(z)]G_R(z) = 0 \quad (7)$$

$G_{L\alpha}(z) \neq 0$ と仮定し、式 (7) の両辺を $G_{L\alpha}(z)$ で割ると、式 (8) と表せる。添字の α, β は話者 α, β に対するものであることを示す。

$$H_{LL}(z) - W_{LL}(z) + [H_{RL}(z) - W_{RL}(z)] \frac{G_{R\alpha}(z)}{G_{L\alpha}(z)} = 0 \quad (8)$$

次に $\frac{G_{R\alpha}(z)}{G_{L\alpha}(z)} = G_{\alpha}(z) = G_{\alpha r}(z) + jG_{\alpha i}(z)$ とすると、式 (9) のようになる。添字の r と i はそれぞれ実部と虚部を表す。

$$\begin{aligned} 0 = & [H_{LLr}(z) + jH_{LLi}(z) - W_{LLr}(z) - jW_{LLi}(z)] \\ & + [H_{RLr}(z) + jH_{RLi}(z) - W_{RLr}(z) - jW_{RLi}(z)] \\ & * [G_{\alpha r}(z) + jG_{\alpha i}(z)] \end{aligned} \quad (9)$$

これを分解すると、式 (10) となる。

$$\begin{aligned} 0 = & H_{LLr}(z) + jH_{LLi}(z) - W_{LLr}(z) - jW_{LLi}(z) \\ & + G_{\alpha r}(z)H_{RLr}(z) + jG_{\alpha r}(z)H_{RLi}(z) \\ & - G_{\alpha r}(z)W_{RLr}(z) - jG_{\alpha r}(z)W_{RLi}(z) \\ & + jG_{\alpha i}(z)H_{RLr}(z) - G_{\alpha i}(z)H_{RLi}(z) \\ & - jG_{\alpha i}(z)W_{RLr}(z) + G_{\alpha i}(z)W_{RLi}(z) \end{aligned} \quad (10)$$

これを実数部、虚数部にまとめたものが式 (11) となる。

$$\begin{aligned} 0 = & H_{LLr}(z) - W_{LLr}(z) + G_{\alpha r}(z)H_{RLr}(z) \\ & - G_{\alpha r}(z)W_{RLr}(z) - G_{\alpha i}(z)H_{RLi}(z) + G_{\alpha i}(z)W_{RLi}(z) \\ & + j(H_{LLi}(z) - W_{LLi}(z) + G_{\alpha r}(z)H_{RLi}(z) \\ & - G_{\alpha r}(z)W_{RLi}(z) + G_{\alpha i}(z)H_{RLr}(z) - G_{\alpha i}(z)W_{RLr}(z)) \end{aligned} \quad (11)$$

この式 (11) の実数部、虚数部をそれぞれ式 (12)、式 (13) とする。

$$\begin{aligned} 0 = & H_{LLr}(z) - W_{LLr}(z) + G_{\alpha r}(z)H_{RLr}(z) - G_{\alpha r}(z)W_{RLr}(z) \\ & - G_{\alpha i}(z)H_{RLi}(z) + G_{\alpha i}(z)W_{RLi}(z) \end{aligned} \quad (12)$$

$$\begin{aligned} 0 = & H_{LLi}(z) - W_{LLi}(z) + G_{\alpha r}(z)H_{RLi}(z) - G_{\alpha r}(z)W_{RLi}(z) \\ & + G_{\alpha i}(z)H_{RLr}(z) - G_{\alpha i}(z)W_{RLr}(z) \end{aligned} \quad (13)$$

式 (12), 式 (13) を変形して式 (14), 式 (15) とする。

$$\begin{aligned} W_{LLr}(z) + G_{\alpha r}(z)W_{RLr}(z) - G_{\alpha i}(z)W_{RLi}(z) \\ = H_{LLr}(z) + G_{\alpha r}(z)H_{RLr}(z) - G_{\alpha i}(z)H_{RLi}(z) \end{aligned} \quad (14)$$

$$\begin{aligned} W_{LLi}(z) + G_{\alpha r}(z)W_{RLi}(z) + G_{\alpha i}(z)W_{RLr}(z) \\ = H_{LLi}(z) + G_{\alpha r}(z)H_{RLi}(z) + G_{\alpha i}(z)H_{RLr}(z) \end{aligned} \quad (15)$$

ここで、 $G_{\alpha r}(z) = a, G_{\alpha i}(z) = b, H_{LLr}(z) + G_{\alpha r}(z)H_{RLr}(z) - G_{\alpha i}(z)H_{RLi}(z) = c, H_{LLi}(z) + G_{\alpha r}(z)H_{RLi}(z) + G_{\alpha i}(z)H_{RLr}(z) = d$ とおくと、不定解の分布は式 (16), 式 (17) となる。

$$W_{LLr}(z) + aW_{RLr}(z) - bW_{RLi}(z) = c \quad (16)$$

$$W_{LLi}(z) + aW_{RLi}(z) + bW_{RLr}(z) = d \quad (17)$$

式 (16), 式 (17) は、フィルタの数を倍にして2つの異なる初期値から学習させることで2つずつの式ができ、未知定数 a, b, c, d に対する合計4つの連立方程式を作ることができる。これを行列で表したものが式 (18) となる。添字の 1, 2 は初期値番号を表している。

$$\begin{bmatrix} W_{RLr1}(z) & -W_{RLi1}(z) & -1 & 0 \\ W_{RLi1}(z) & W_{RLr1}(z) & 0 & -1 \\ W_{RLr2}(z) & -W_{RLi2}(z) & -1 & 0 \\ W_{RLi2}(z) & W_{RLr2}(z) & 0 & -1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} -W_{LLr1}(z) \\ -W_{LLi1}(z) \\ -W_{LLr2}(z) \\ -W_{LLi2}(z) \end{bmatrix} \quad (18)$$

式 (18) の a, b, c, d は、式 (19) を解くことで求めることができる。本研究では、逆行列を用いて計算を行なった。

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} W_{RLr1}(z) & -W_{RLi1}(z) & -1 & 0 \\ W_{RLi1}(z) & W_{RLr1}(z) & 0 & -1 \\ W_{RLr2}(z) & -W_{RLi2}(z) & -1 & 0 \\ W_{RLi2}(z) & W_{RLr2}(z) & 0 & -1 \end{bmatrix}^{-1} \begin{bmatrix} -W_{LLr1}(z) \\ -W_{LLi1}(z) \\ -W_{LLr2}(z) \\ -W_{LLi2}(z) \end{bmatrix} \quad (19)$$

式 (19) より求まる a, b, c, d を話者 α, β について求めることにより、式 (16), 式 (17) は2つずつの式ができる。したがって、 $W_{LLr}, W_{LLi}, W_{RLr}, W_{RLi}$ に対する合計4つの連立方程式を作ることができる。これを行列で表したものが式 (20) となる。

$$\begin{bmatrix} 1 & 0 & a_{\alpha} & -b_{\alpha} \\ 0 & 1 & b_{\alpha} & a_{\alpha} \\ 1 & 0 & a_{\beta} & -b_{\beta} \\ 0 & 1 & b_{\beta} & a_{\beta} \end{bmatrix} \begin{bmatrix} W_{LLr}(z) \\ W_{LLi}(z) \\ W_{RLr}(z) \\ W_{RLi}(z) \end{bmatrix} = \begin{bmatrix} c_{\alpha} \\ d_{\alpha} \\ c_{\beta} \\ d_{\beta} \end{bmatrix} \quad (20)$$

式 (19) と同様に逆行列を用いて計算を行なったものが式 (21) である。

$$\begin{bmatrix} W_{LLr}(z) \\ W_{LLi}(z) \\ W_{RLr}(z) \\ W_{RLi}(z) \end{bmatrix} = \begin{bmatrix} 1 & 0 & a_\alpha & -b_\alpha \\ 0 & 1 & b_\alpha & a_\alpha \\ 1 & 0 & a_\beta & -b_\beta \\ 0 & 1 & b_\beta & a_\beta \end{bmatrix}^{-1} \begin{bmatrix} c_\alpha \\ d_\alpha \\ c_\beta \\ d_\beta \end{bmatrix} \quad (21)$$

式 (21) により $W_{LLr}(z)$, $W_{LLi}(z)$, $W_{RLr}(z)$, $W_{RLi}(z)$ を各周波数毎に求める。これを式 (22), 式 (23) のように逆フーリエ変換することで、時間域での最適値を推定する。

$$w_{LL}(n) = F^{-1}[W_{LLr}(z) + jW_{LLi}(z)] \quad (22)$$

$$w_{RL}(n) = F^{-1}[W_{RLr}(z) + jW_{RLi}(z)] \quad (23)$$

2.2 初期値の与え方

式 (16) 式 (17) に与える値は独立性の高いものでないと式 (19) が悪条件になり、推定精度が落ちてしまう。ここでは独立性の高い初期値を与える方法について説明する。

まず1つ目のフィルタをある初期値から一定期間学習させる。そしてそれを周波数領域に変換し、学習方向に垂直な位置に2つ目のフィルタの初期値をとる。そしてそれらを時間領域に戻して学習を再開し、収束させるというものである。これによって収束値を独立性の高いものにできると考えられる。

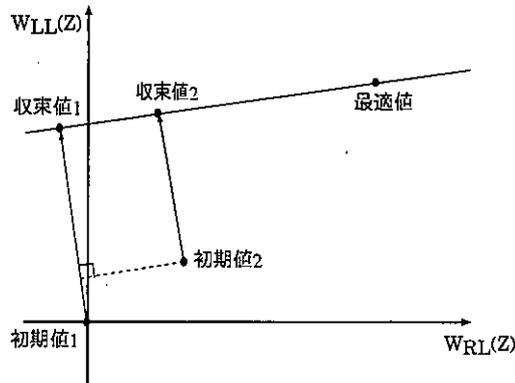


図4 フィルタの初期値設定

しかしこれだけでは学習の途中で2つの値が近くなった時に対応できないので、以下のような再初期化を行なうことにした。ある一定間隔毎に2つの値の距離を計算し、その距離が閾値より小さくなったら別の垂直方向に初期値を取り直すというものである。

2 つの適応フィルタの周波数領域における係数の値を次のようなベクトルで表す。

$$V_1 = \begin{bmatrix} W_{LLr1}(z) \\ W_{RLr1}(z) \\ W_{LLi1}(z) \\ W_{RLi1}(z) \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \quad (24)$$

$$V_2 = \begin{bmatrix} W_{LLr2}(z) \\ W_{RLr2}(z) \\ W_{LLi2}(z) \\ W_{RLi2}(z) \end{bmatrix} = \begin{bmatrix} v'_1 \\ v'_2 \\ v'_3 \\ v'_4 \end{bmatrix} \quad (25)$$

このとき、次の 3 つのベクトルの右辺第 2 項は V_1 に直交する。この 3 つの垂直方向に初期値を取り直すようにした。

$$V' = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} + \begin{bmatrix} v_2 \\ -v_1 \\ v_4 \\ -v_3 \end{bmatrix} \quad (26)$$

$$V'' = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} + \begin{bmatrix} -v_3 \\ v_4 \\ v_1 \\ -v_2 \end{bmatrix} \quad (27)$$

$$V''' = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} + \begin{bmatrix} v_4 \\ v_3 \\ -v_2 \\ -v_1 \end{bmatrix} \quad (28)$$

最初は 2 つ目の初期値を 1 つめの垂直方向 V' にとる。そして 2 つの適応フィルタに対して V_1, V_2 間の距離を求め、その距離が一定以下になったら 2 つめの垂直方向 V'' に初期値を取り直す。もう一度距離が一定以下になったら 3 つめの垂直方向 V''' に初期値を取り直す。このようにして学習の途中で 2 つの値が近くなったとき、別の垂直方向に値を取り直すようにした。

2.3 話者交代の判別

本稿における最適値推定法は、部屋 A に二人の話者 α, β を仮定している。そして、それぞれの話者に対して適応フィルタを学習させるため、二人の話者のどちらが話しているのかを判別し学習を切替える必要がある。ここではその方法について説明する。

部屋 A の 2 人の話者がそれぞれ話したとき、話者からマイクロフォンまでの伝達関数には違いがある。そのため、それぞれの話者に対して右チャンネルと左チャンネルの信号の相関を計算すると、その最大値をとる時間ずれにも違いが出る。よって各チャンネルの相互相関を計算することで話者を判別することができる。左チャンネルと右チャンネルの信号の相互相関を次のような式で計算する。 x_L が左チャンネル、 x_R が右チャンネルの信号、 m が時間ずれをあらわす。

$$r_m(n) = \frac{1}{N} \sum_{k=0}^{N-1} x_L(n-k-m)x_R(n-k) \quad (29)$$

相互相関が最大となる m を m_{max} とする。その値をみることで話者が α, β のどちらであるか判別することができる。今回は期待値の計算には次のようなリーク積分を用いた。

$$E[r_m(n)] = \gamma \cdot r_m(n-1) + (1-\gamma) \cdot r_m(n) \quad (30)$$

以上のような方法で話者 α, β を判別する。

リモートパスを通した白色信号で話者切り替え判別を行ったときの m_{max} の値を図 5 に示す。学習回数 10 万回で話者を α から β に切り替えている。 m_{max} の値をみると、話者 α から β への切り替わりで大きく値が変わっているため話者が変わっていると判別できる。

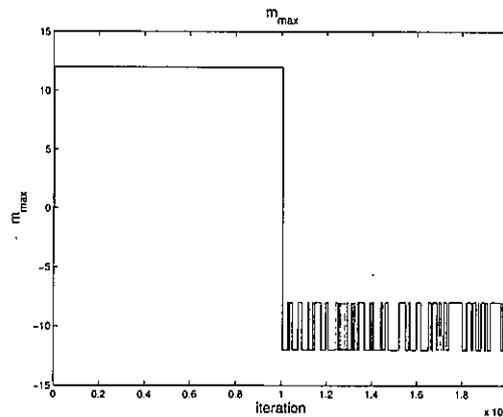


図 5 話者切り替えに対する m_{max} の変化

3 シミュレーション

3.1 シミュレーション条件

以上のような原理に基づきシミュレーションを行った。条件は表 1 の通りである。ただし、学習回数は各話者 10 万回で α, β が交互に話すものとしている。各話者に対して収束値が得られたらエコーパスの推定を行い、以降も学習を行った。適応フィルタの初期値をすべて 0 またはすべて 1 とした場合と、2 つめの初期値を学習の垂直方向にとり再設定も行う方法との比較を行った。

表 1 シミュレーション条件

| パラメータ等 | 値等 |
|--------------------|----------|
| 適応フィルタ | FIR フィルタ |
| パスのタップ数 N_A, N_B | 128, 256 |
| 適応フィルタのタップ数 N_W | 256 |
| 各話者に対する学習回数 | 10 万 |
| ステップサイズ μ | 0.1 |
| 入力信号 $s(n)$ | 白色雑音 |
| 学習アルゴリズム | NLMS |

3.2 シミュレーション結果

図 6 に初期値を全て 0 と全て 1 に設定した場合 (初期値固定) の場合、図 7 に提案法の適応フィルタの周波数領域での理想特性と推定値を示す。推定値は 40 万サンプル時点での値である。理想特性と推定値を比較すると、初期値固定の場合では 1700~2000Hz など、推定値が理想特性に一致していないところが多い。それに対して提案法では比較的推定誤差は小さくなっている。

図 8 にエコーパスの同定精度である NCEV(Normalized Coefficient Error Vector) を示す。NCEV の定義は次の通りである。

$$NCEV(n) = 10 \log_{10} \frac{\|h - w(n)\|^2}{\|h\|^2} \quad (31)$$

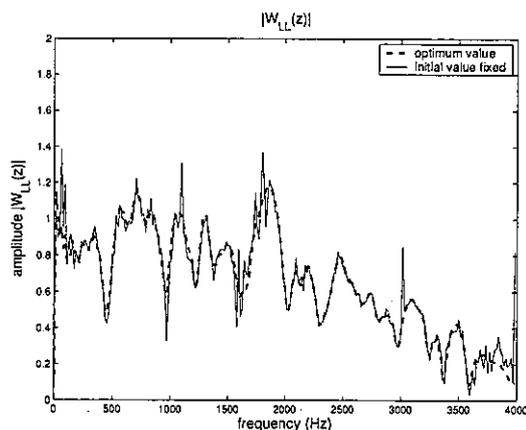


図 6 周波数領域の理想特性と推定値（初期値固定）

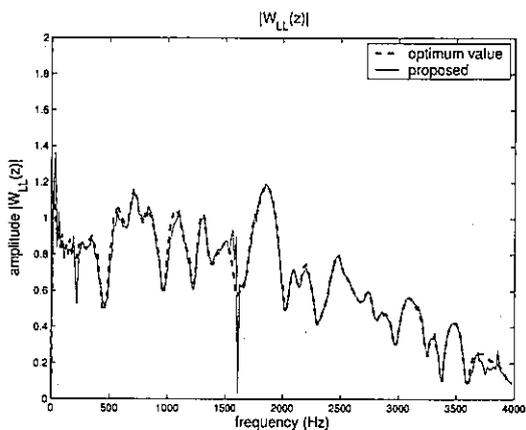


図 7 周波数領域の理想特性と推定値（提案法）

図 8 をみると、NCEV は提案法によって 40 万サンプル時点で約 2dB 下がっている。このことから提案法を用いることで式 (19) に関する悪条件を緩和し、推定精度が上がっていることが確認できる。

図 9 にエコー低減量 (Echo Return Loss Enhancement) を示す。ERLE の定義は次の通りである。 e_L が誤差信号、 d_L がエコー信号で、 $M = 256$ とした。提案法では推定後の ERLE にも若干の向上がみられた。

$$ERLE(n) = 10 \log_{10} \frac{\sum_{i=0}^{M-1} d_L(n-i)^2}{\sum_{i=0}^{M-1} e_L(n-i)^2} \quad (32)$$

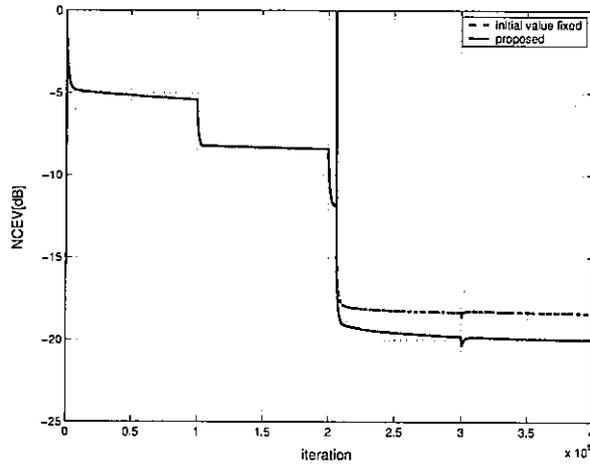


図8 エコーパスの同定精度

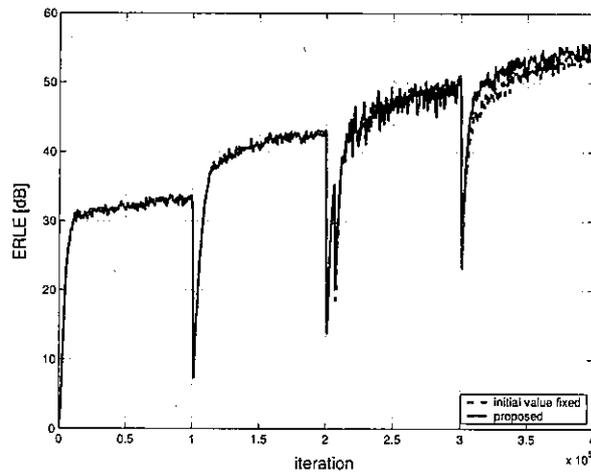


図9 エコー低減量

4 まとめ

ステレオエコーキャンセラにおける解の不確実性に対して、周波数領域から数値計算によって最適値を推定する方法を提案した。各エコーパスに対して2つの適応フィルタを別々の初期値から学習させ、その収束値から不定解の分布を得る。話者2人に対して各々の不定解の分布が得られたら、そこから最適値を数値計算によって推定するものである。また、初期値の取り方を改良することによって悪条件を緩和し、推定精度を上げる手法を提案した。また、チャンネル間の相関によって話者交代の判別を行う方法を提案した。今

後の課題としては、他方式との比較、音声入力時にも同様の結果が得られるかの確認、適応フィルタの数を減らすことができるかの検討などがあげられる。

実時間信号処理システムの構築

目次

| | | |
|-----|--|----|
| 1 | はじめに | 18 |
| 2 | 浮動小数点 DSP を用いた実時間信号処理システム | 19 |
| 2.1 | SHARC プロセッサ | 19 |
| 3 | 固定小数点 DSP と FPGA を用いた実時間信号処理システム | 22 |
| 3.1 | Blackfin プロセッサ | 22 |

1 はじめに

実時間信号処理システムを用いた性能評価を行なう準備として、複数の実時間信号処理システムを構築した。本研究においては、実現の容易性とコストの観点から、

1. 浮動小数点 DSP を用いたシステム
2. 固定小数点 DSP を用いたシステム
3. 専用 LSI を用いたシステム

の 3 種類を検討した。

浮動小数点 DSP を用いたシステムは、単精度浮動小数点演算を用いた計算機シミュレーションと同等の演算精度を確保できるため、アルゴリズムをほぼそのまま DSP に移行できる。したがって、実時間信号処理による実験の第一段階としては適している。しかし、DSP のコストが固定小数点 DSP に比べて高いため、製品化にはハードルが高い。

一方、固定小数点 DSP を用いたシステムは、アルゴリズム実現のために固定小数点演算による精度劣化の影響を評価する必要がある。アルゴリズムによっては数値的に不安定になることもある。一方、コスト的には有利であるので、製品化には適している。

専用 LSI を用いたシステムは、量産時のコストは優れているが、開発のために多くの時間とコストが発生する。しかし、Field Programmable Gate Array (FPGA) は比較的開発が容易でありながら DSP を上回る性能を得られるため、試作用途には非常に優れている。本研究では、固定小数点 DSP と FPGA を搭載したシステムを構築し、DSP と FPGA を組み合わせた実現方法を検討するための土台を整える。

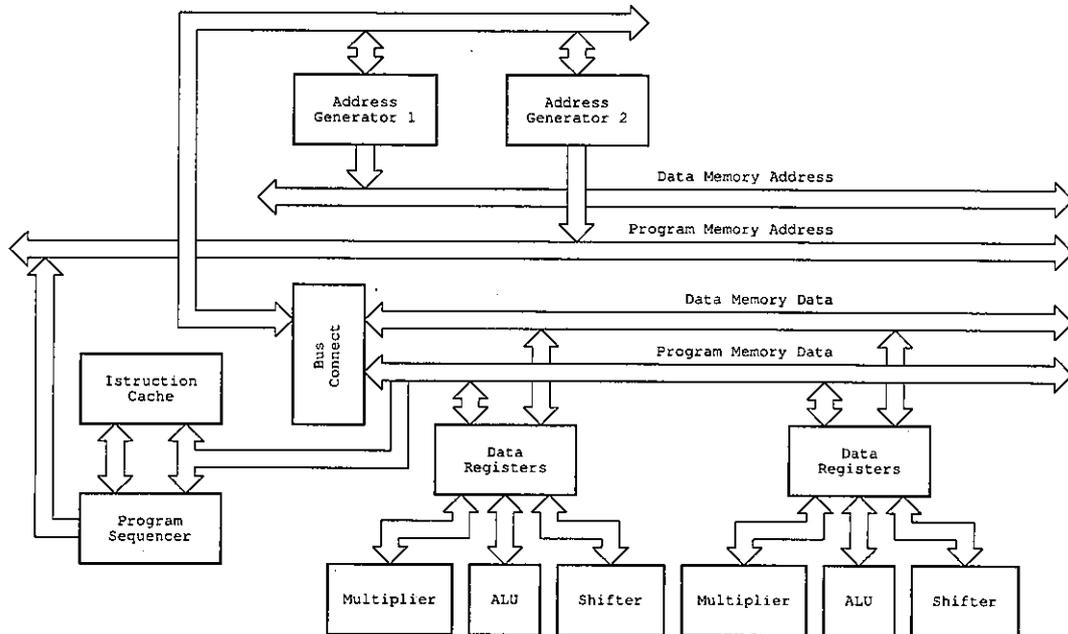


図 10 SIMD SHARC DSP コア

2 浮動小数点 DSP を用いた実時間信号処理システム

2.1 SHARC プロセッサ

浮動小数点 DSP として Analog Devices 社製の ADSP-21369 SHARC プロセッサ [14] を用いたシステムを検討する。図 10 に DSP コアのブロック図を示す。乗算器、ALU、シフタ、レジスタファイルから構成される演算ユニット (Processing Element, PE) を 2 組備えている。2 組の PE は 2 組の異なるデータに対して同じ演算を実行する。これは、SAEC のようなステレオ信号の処理に適した構成となっている。

ADSP-21369 は最大 300MHz で動作するので、積和演算を 1 秒間に最大 6×10^8 回実行できる。サンプリング周波数を 16kHz、学習アルゴリズムを NLMS として、線形結合型 SAEC を実装すると、最大で約 4600 タップを処理できる。これは、残響時間約 280msec に相当し、多くの環境に対応可能なものとなる。

図 11 に ADSP-21369 のブロック図を示す。ADSP-21369 は豊富な周辺回路を備えて

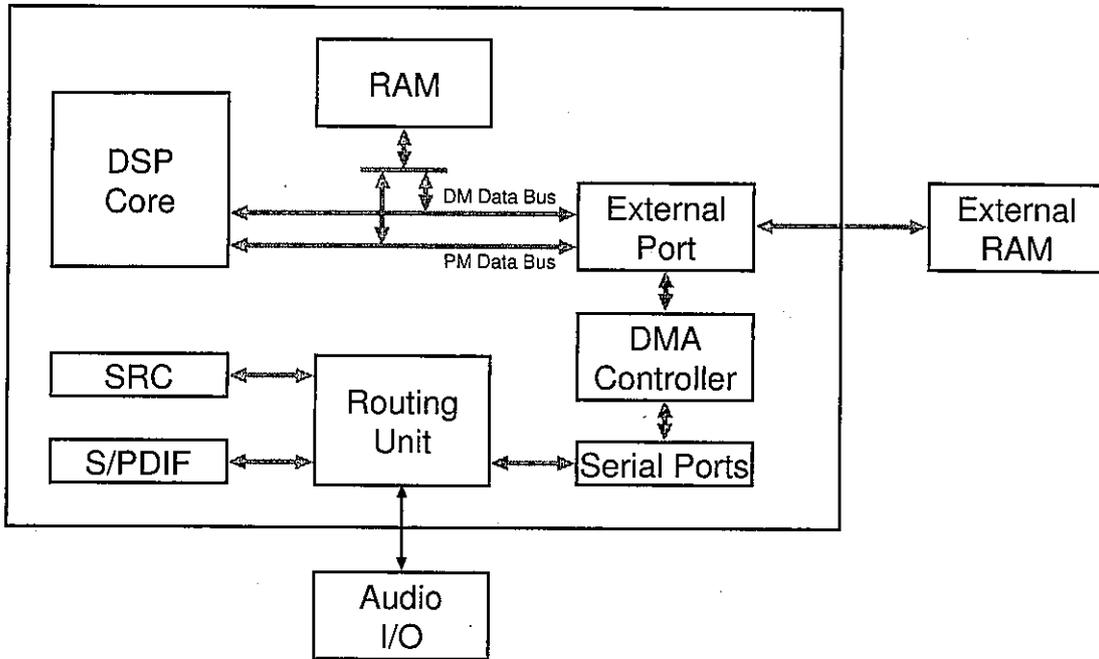


図 11 ADSP-21369 SHARC DSP

いる。SAECの実装に関連するものとしては、

- シリアルポート
- デジタルオーディオインタフェース (Sony/Philips Digital Interface, S/PDIF)
- サンプリング周波数変換器 (Sample Rate Convertor, SRC)
- Direct Memory Access (DMA) コントローラ

などが挙げられる。これらの回路間の接続は、Routing Unit の設定によって動的に変更できる。

接続するオーディオ機器によってサンプリング周波数が異なる場合や、DSP で処理する時点で異なるサンプリング周波数を選択したい場合がある。このような場合には SRC を用いることができる。DSP コアに負荷をかけることなく、高精度の変換が可能である。

入出力データを RAM に転送するためには、DMA コントローラを使用する。指定回数のデータ転送が完了した時点で DSP に割り込みを発生させる。これによって、DSP コアがデータ転送に要する負荷も低減できる。

シリアルポートは、(Analog-to-Digital Convertor, ADC) や (Digital-to-Analog Convertor, DAC) と DSP の接続に使用する。デジタルオーディオ入出力を有するマルチメディア PC 等との接続には S/PDIF が使用できる。デジタル接続することにより、

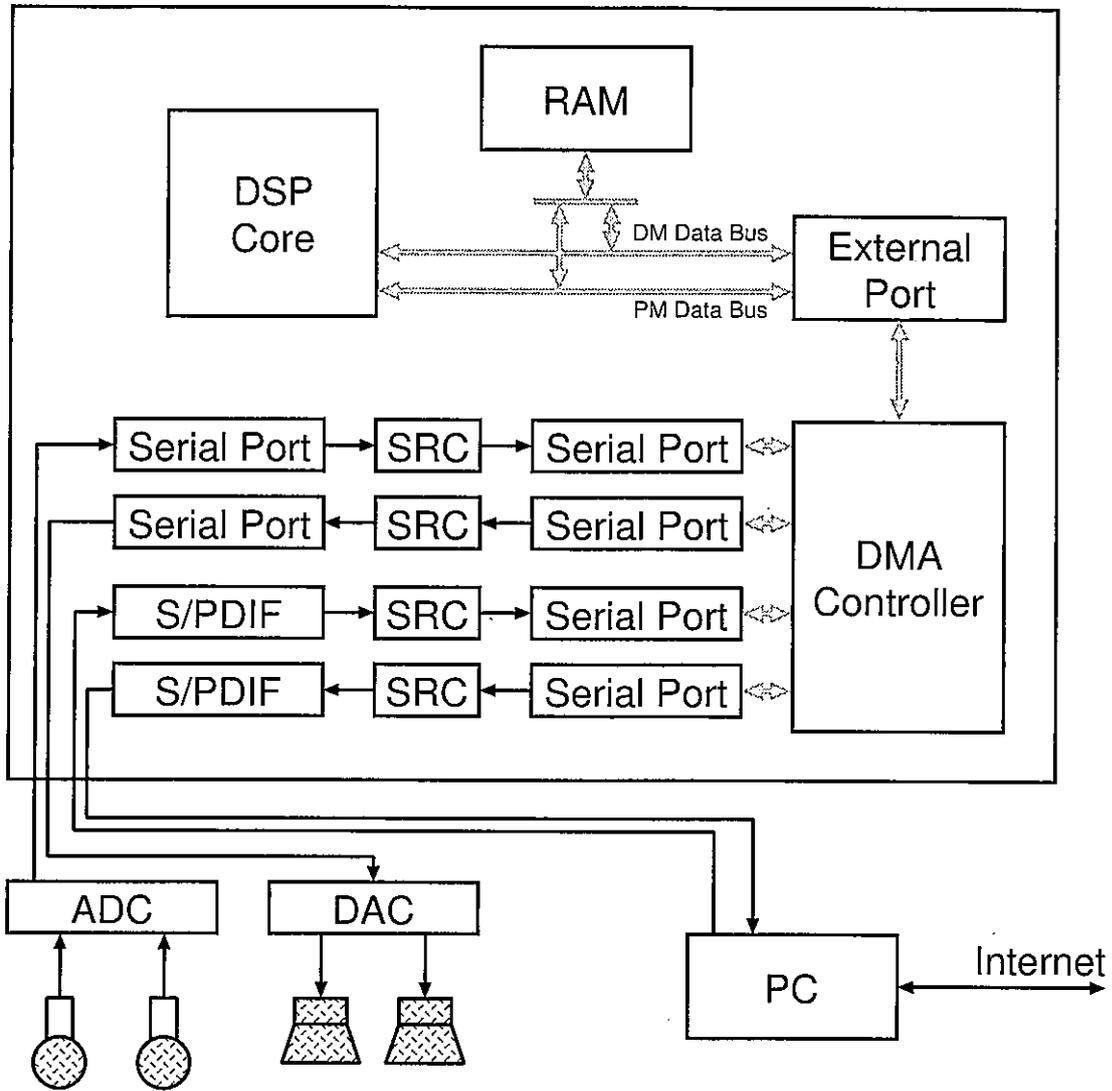


図 12 ADSP-21369 SHARC DSP による SAEC 評価装置

音質の劣化を抑えることができる。

図 12 に、ADSP-21369 を用いた SAEC 実験システムのブロック図を示す。PC と信号処理装置は S/PDIF で接続している。PC はインターネット経由で遠隔会議を行なう。

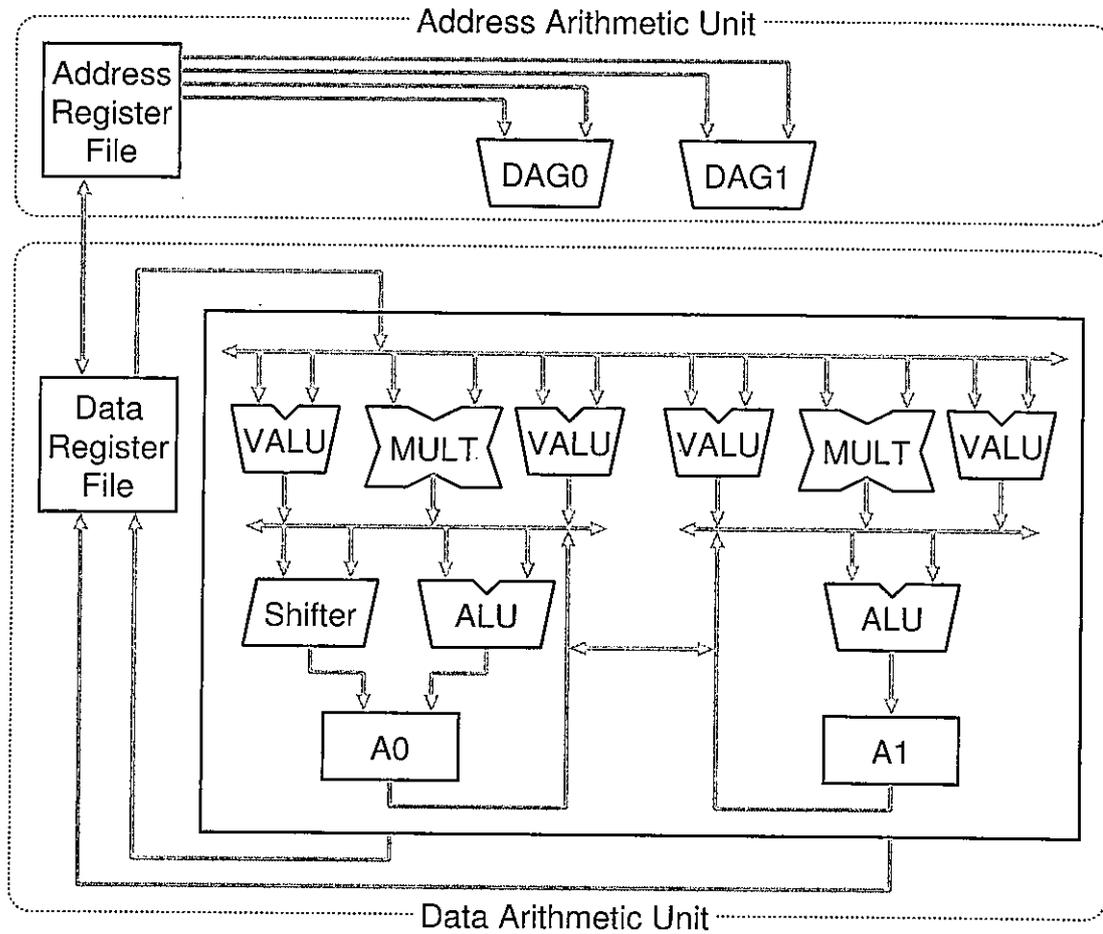


図 13 Blackfin DSP コア

3 固定小数点 DSP と FPGA を用いた実時間信号処理システム

3.1 Blackfin プロセッサ

固定小数点 DSP として Analog Devices 社製の ADSP-BF561 Blackfin プロセッサ [15] を用いたシステムを検討する。Blackfin プロセッサは Analog Devices と Intel が共同開発したもので、動画像の符号化などのマルチメディア処理を意識したプロセッサである。

図 13 に DSP コアのブロック図を示す。2 個の 16bit 乗算器 (Multiplier, MULT)、2 個の 40bit ALU、4 個の 8bit ビデオ ALU (Video ALU, VALU)、40bit シフタ、レジスタファイル、2 個のデータアドレス生成器 (Data Address Generator, DAG) を有する。

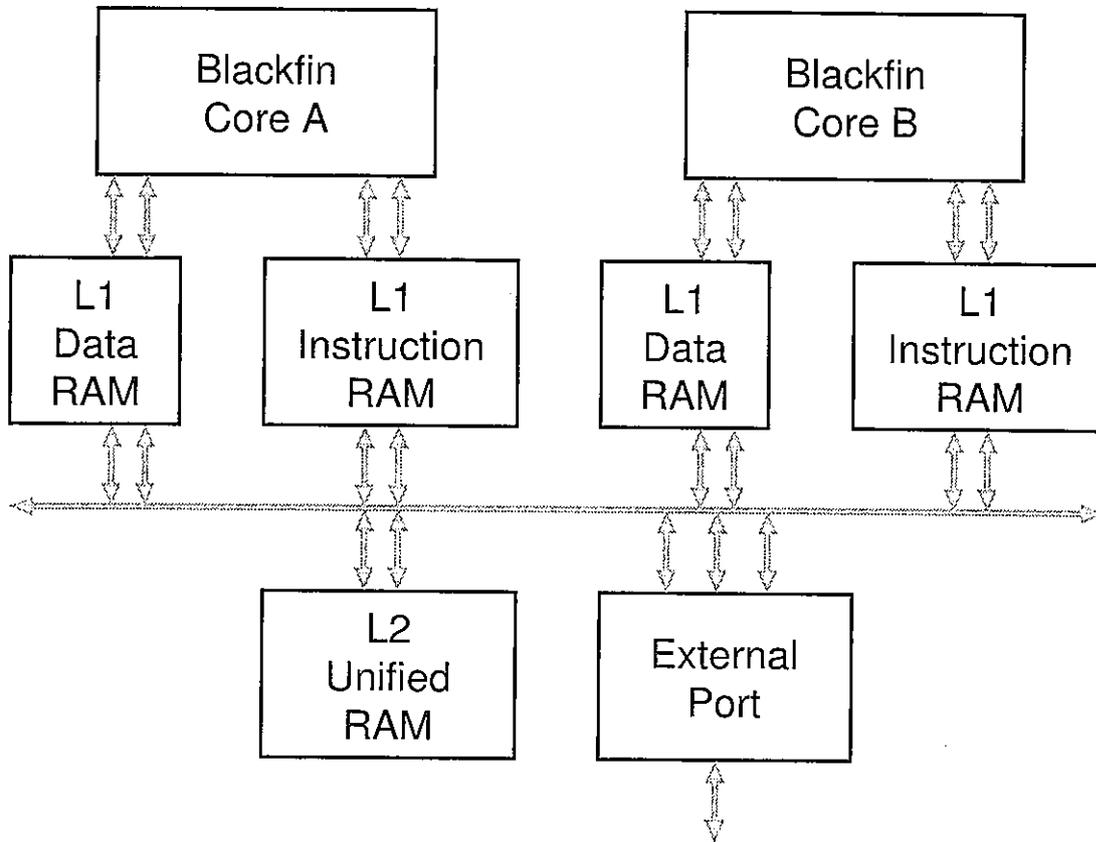


図 14 Blackfin DSP

VALU は動画像のフレーム間予測符号化における動き推定を高速に処理する。乗算器と ALU を 2 組備えており、SAEC のようなステレオ信号の処理にも適した構成となっている。

図 14 に ADSP-BF561 Blackfin プロセッサのブロック図を示す。図 13 に示した DSP コアを 2 組有している。最大 600MHz で動作するので、積和演算を 1 秒間に最大 2.4×10^9 回実行できる。固定小数点演算であることに起因する精度劣化対策のために演算量が増加することを考慮しても、残響時間数百 msec に対応する SAEC を十分に実装可能であると考えられる。

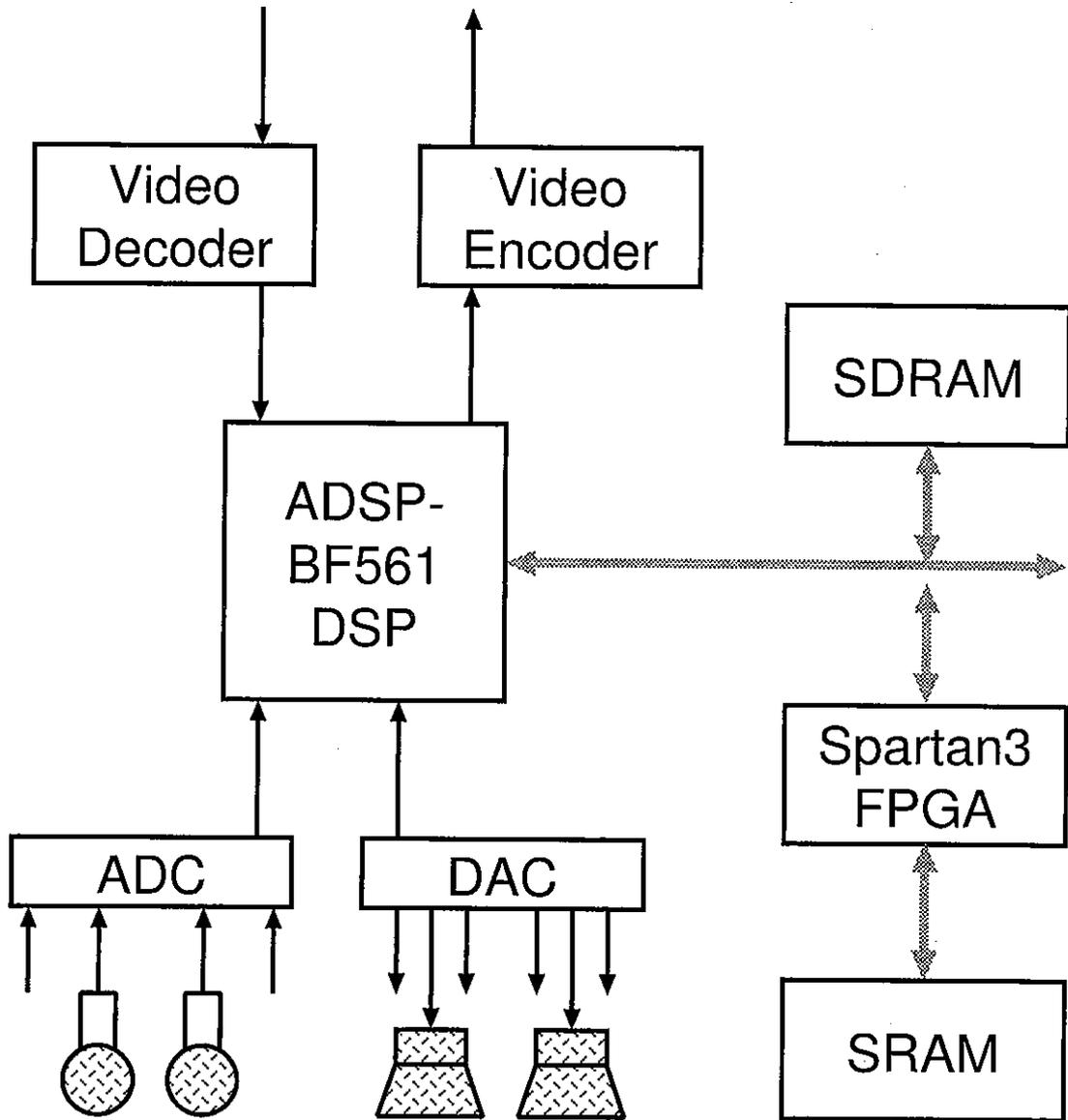


図 15 Blackfin DSP による SAEC 評価装置

図 15 に、ADSP-BF561 を用いた SAEC 実験システムのブロック図を示す。Xilinx 社の Spartan3 FPGA [16]、ビデオ入出力などを備えている。

Intel IA32 プロセッサを用いた SAEC の実現

目次

| | | |
|---|------------------------|----|
| 1 | ステレオ音響エコーキャンセラ | 26 |
| 2 | Intel IA-32 プロセッサ [17] | 27 |
| 3 | 最適化における問題点 | 28 |
| 4 | SAEC の高速化 | 28 |
| 5 | 性能評価 | 32 |
| 6 | まとめ | 33 |

1 ステレオ音響エコーキャンセラ

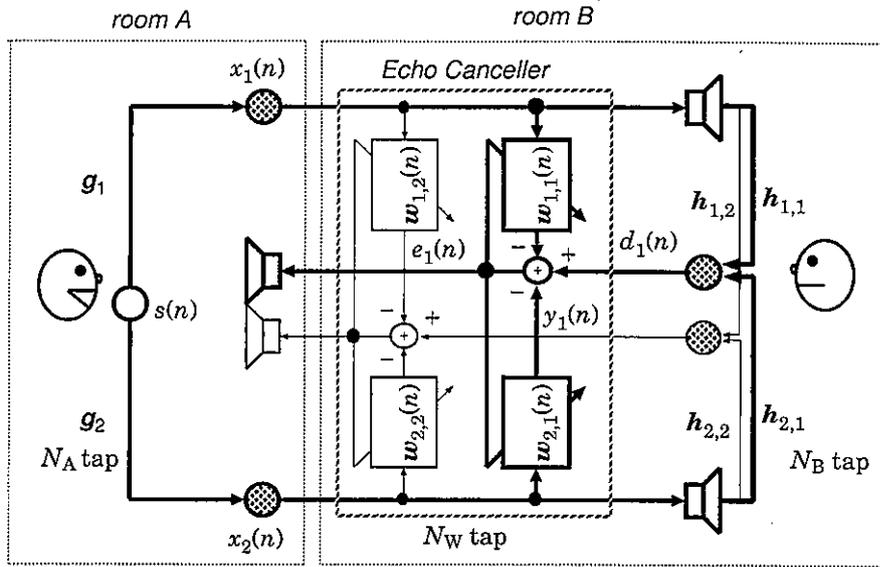


図 16 ステレオ音響エコーキャンセラを用いた遠隔会議

図 16 にステレオ音響エコーキャンセラ (Stereophonic Acoustic Echo Canceller, SAEC) を用いた遠隔会議を示す。このエコーキャンセラは 4 個のエコーパスと 1 対 1 に対応する 4 個の適応フィルタを使用する。各適応フィルタは対応するエコーパスを推定する。

時刻 n における第 i チャンネルの遠端信号 $x_i(n)$ は、話者の音声 $s(n)$ が部屋 A のインパルス応答 g_i を経て生成される。 $x_i(n)$ は i 番目のスピーカから j 番目のマイクロホンに至るエコーパス $h_{i,j}$ を経てマイクロホンに到達し、エコー $d_j(n)$ となる。同様に、適応フィルタ $w_{i,j}(n)$ はエコーレプリカ $y_j(n)$ を生成する。 $w_{i,j}(n)$ は残留エコー $e_j(n)$ を最小にするように制御される。

SAEC はエコーレプリカ $y_j(n)$ を

$$y_j(n) = w_{1,j}^T(n)x_1(n) + w_{2,j}^T(n)x_2(n) \quad (1)$$

で計算する。残留エコー $e_j(n)$ は

$$e_j(n) = d_j(n) - y_j(n). \quad (2)$$

となる。Normalized Least Mean Squares (NLMS) アルゴリズム [18] を仮定すると、

フィルタ係数 $w_{i,j}(n)$ は

$$w_{i,j}(n+1) = w_{i,j}(n) + \frac{\mu e_j(n) x_i(n)}{|x_i(n)|^2} \quad (3)$$

で更新される。ここで、正の定数 μ はステップサイズと呼ばれ、学習を制御する。

2 Intel IA-32 プロセッサ [17]

Intel Core2 プロセッサを想定して、演算量の削減を行なった。Core2 プロセッサの主な特徴は下記の通り。

- マルチメディア向け拡張命令セット MMX と SSE
- 最大 14 ステージのパイプラインが 5 本
 - ALU, FP/MMX/SSE Move, Branch
 - ALU, FP/MMX/SSE Add
 - ALU, FP/MMX/SSE Multiply
 - Load
 - Store
- 1 サイクル当たり最大 5 命令を実行可能
 - 1 サイクル当たり最大 3 個の ALU 演算
 - 1 サイクル当たり最大 3 個の SSE 命令
- 分岐予測
- 32kB の命令用と 32kB のデータ用 L1 キャッシュ
- 2MB~6MB の L2 キャッシュ
- ハードウェアによるキャッシュのプリフェッチ

データアクセスのパターンを予測し、事前に、かつ、自動的に外部メモリからキャッシュにデータを読み込む。
- 8 個の汎用整数レジスタ
- 8 個の浮動小数点レジスタ
- 8 個の SSE レジスタ

MMX (Multi Media eXtension) と SSE (Streaming Single-instruction multiple-data Extension) 命令 [19, 20] はベクトル演算命令を含んでいる。32 ビット単精度浮動小数点データに対しては、4 組のデータに対して同時に同じ演算を行なう。したがって、十分なデータを演算器に供給可能であれば、最大 4 倍の高速化を見込める。

Pentium4 プロセッサとの性能比較も行なった。Core2 と Pentium4 との大きな違いの一つは SSE 命令のスループットである。Core2 はほとんどの SSE 命令を 1 命令 1 サイク

ルのスループットで実行できる一方、Pentium4 は 2 サイクルに 1 命令となる [21]。

3 最適化における問題点

Intel Core2 プロセッサのような SIMD (Single-Instruction Multiple-Data) 命令を備える汎用プロセッサを用いて信号処理システムを実現する際には、多くの考慮すべき問題がある。以下にその一例を示す。

- 効率の良いベクトル化
- 並列ロードストアに適したデータ配置
- タップ付き遅延線の実装方法
- メモリ構成、特に低速な外部メモリ
- メモリからのデータ読み出しの長いレイテンシ

Core2 プロセッサは L1 キャッシュからの読み出しでも 7 サイクルを要する

- 少ないデータレジスタ

ベクトル化とデータ配置の問題は、デジタルシグナルプロセッサ (Digital Signal Processors DSP's) と共通する。一方、他の問題点は汎用プロセッサに特有となる。多くの DSP はタップ付き遅延線用のアドレス発生器や複数の低遅延メモリを持つ。データレジスタが少ないことは、Intel IA-32 プロセッサに固有の問題といえる。

4 要素のベクトル演算を用いて FIR フィルタを実現する際には、 k 番目から $k + 3$ 番目までの 4 タップを同時に演算することが多い。しかし、このベクトル化はデータ配置問題を引き起こす。これは、4 ワード境界を超えた同時アクセスができないことに起因する。データ配置に関する問題とその解決方法は、[22] 等で検討されている。データ配置問題を解決するために、位置をずらした複数個のデータメモリを用意する方法が提案されている。しかし、4 要素のベクトル演算には 4 倍のメモリが必要となる。

長いレイテンシに対しては、ループアンローリング [23] が有効である。しかし、この手法は多くのレジスタを必要とする。 n サイクルのレイテンシを隠蔽するためには、通常 n 個のレジスタが必要となる。レジスタが 8 個というのは極めて少ない。

4 SAEC の高速化

今回の実装においては、複数のタップ番号ではなく、複数のチャンネルに対するベクトル化を行なった。適応フィルタ 4 個における第 k 番目のタップを同時に計算する。図 17 にデータの配置を示す。この配置では、データ配置問題は発生しない。

メモリアクセスの長いレイテンシによる性能低下を避けるには、メモリアクセスを削

| | | | |
|------------------|------------------|------------------|------------------|
| ⋮ | ⋮ | ⋮ | ⋮ |
| $w_{2,2,k+1}(n)$ | $w_{1,2,k+1}(n)$ | $w_{2,1,k+1}(n)$ | $w_{1,1,k+1}(n)$ |
| $w_{2,2,k}(n)$ | $w_{1,2,k}(n)$ | $w_{2,1,k}(n)$ | $w_{1,1,k}(n)$ |
| ⋮ | ⋮ | ⋮ | ⋮ |

(a) フィルタ係数

| | |
|--------------|--------------|
| ⋮ | ⋮ |
| $x_2(n-i-1)$ | $x_1(n-i-1)$ |
| $x_2(n-i)$ | $x_1(n-i)$ |
| ⋮ | ⋮ |

(b) タップ付き遅延線

図 17 データ配置

減することが有効となる。これは、ロードストア型 DSP においても有効である [24]。式 (1) と (5) の順番を変更することによって、データ読み出しの回数を削減できる。具体的には、

$$w_{i,j,k}(n) = w_{i,j,k}(n-1) + \delta_j(n-1)x_i(n-k-1) \quad (4)$$

$$sum_j(n) = sum_j(n) + w_{i,j,k}(n)x_i(n-k) \quad (5)$$

の 2 式をタップ番号 k が減ずる方向に反復計算する。これによって、 $w_{i,j,k}(n)$ と $x_i(n-k)$ のロード回数を削減できる。(4) において、 $\delta_j(n-1)$ は

$$\delta_j(n-1) = \frac{\mu e_j(n-1)}{|x_i(n-1)|^2} \quad (6)$$

で定義される。 $w_{i,j,k}(n)$ は $w_{i,j}(n)$ の第 k 要素である。 $x_i(n-k)$ のロード回数は、(5) 式における $x_i(n-k)$ を $k = k-1$ における (4) 式で再利用することにより削減する。

図 18 および図 21 に、係数更新と畳み込みのデータフローを示す。図 18 における $x_i(n-k-1)$ は、図 21 の $k = k+1$ に対する $x_i(n-k)$ を再利用する。図 18 における $w_{i,j,k}(n)$ は図 21 でも利用する。したがって、 $w_{i,j,k}(n)$ に対しては N_W 回のロードと N_W 回のストアが必要となる。 $x_i(n-k)$ に対しては N_W 回のロードを行なう。

レイテンシを隠蔽するために、2 個のベクトルを 1 つのループで処理している。1 ループで処理するベクトル数は、実行効率によるものではなく、データレジスタ数によって制

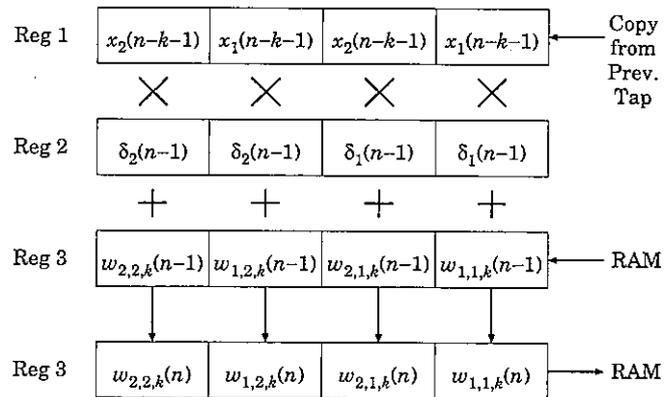


図 18 係数更新のデータフロー

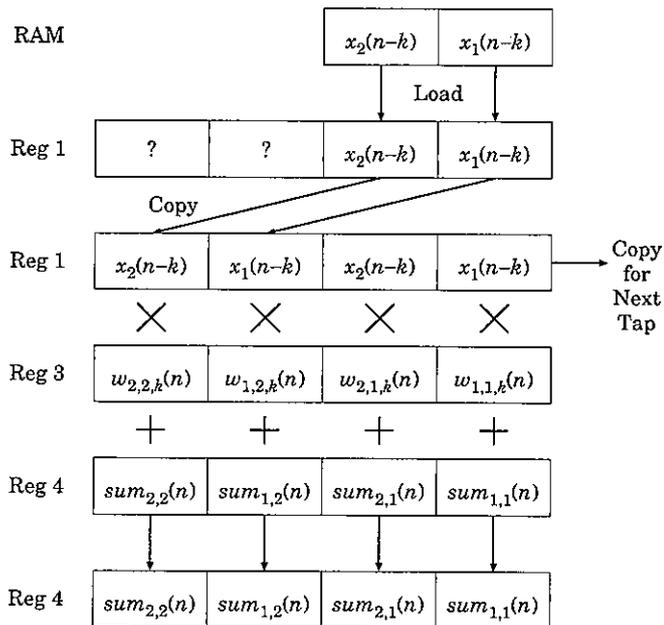


図 19 畳み込み演算のデータフロー

限された。今回使用した命令でレイテンシが最長なのは 7 サイクルなので、効率の観点からは 1 ループ 8 ベクトルが適切と考えられる。しかし、それには 4 倍のレジスタが必要となる。

タップ付き遅延線の実装には、循環バッファが用いられる。専用のアドレス計算命令を持たない場合には、1 タップにつき 1 回の条件分岐を行なうか、余分なメモリを割り当てる必要がある。通常条件分岐はパイプラインの性能を低下させるが、分岐予測機能を有する分岐ユニットが ALU や SSE 演算器と並列に動作するため、性能劣化が非常に低く抑

えられる可能性がある。分岐予測の効果は性能評価の際に検討する。

5 性能評価

表 2 SAEC の仕様

| | |
|--------------------|-----------------------|
| Sampling frequency | 16kHz |
| Number of taps | 3200 |
| Reverberation time | 200ms |
| Adaptation | NLMS |
| Precision | 32-bit floating point |

表 3 PC の仕様

| | Core2 Duo | Pentium4 |
|----------------|------------|------------|
| Type | E6600 | 2.8GHz |
| Core Clock | 2.4GHz | 2.8GHz |
| FSB Clock | 1033MHz | 800MHz |
| L1 Data Cache | 32kB | 8kB |
| L1 Inst. Cache | 32kB | 12kB |
| L2 Cache | 4MB | 512kB |
| Chipset | Intel G965 | Intel 865G |

線形結合形 SAEC を実現し、2 種類の PC で性能評価を行なった。実装した SAEC の仕様を表 2 に示す。表 3 は PC の仕様である。SAEC の最も演算量が多い部分、すなわち (4) と (5) を含むループは SSE 命令を使用してアセンブリ言語で記述した。Core2 プロセッサで実効速度が最大となるように最適化を行なっている。実効速度を FPU を使用する C 言語プログラムと比較した。なお、アセンブラで SSE 命令を記述した場合と C 言語の `xmmintrinsic` を利用して SSE 化した場合と比較して、約 10% 程度の高速化に留まっている。したがって、アセンブラと C 言語による違いはそれほど影響していないと考えられる。

また、循環バッファの実装方法による影響もした。余分なメモリを使用して条件分岐を行なわない方式と、条件分岐を行なう方式を実装し、実行時間を比較した。

表 4 性能評価

| CPU | Unit | Delay Ex | Branch | Time |
|----------|------|----------|--------|-------|
| Core2 | FPU | Yes | No | 77.50 |
| | SSE | Yes | No | 15.65 |
| | SSE | No | Yes | 15.84 |
| Pentium4 | FPU | Yes | No | 98.71 |
| | SSE | Yes | No | 20.08 |
| | SSE | No | Yes | 20.28 |

表 4 に比較結果を示す。120 秒分の入力信号を処理するために要した時間を比較した。120 秒間のデータを約 15~20 秒で処理している。したがって、2.4GHz の Core 2 または 2.8GHz の Pentium4 を使用した場合、1/6~1/8 の CPU パワーで SAEC を処理できることになる。

SSE 命令によるベクトル化によって、4 倍以上の高速化に成功している。また、タップ付き遅延線の実装方法による影響はほとんど見られない。このことから、分岐予測は有効に機能していると考えられる。

Core 2 Duo と Pentium4 の比較結果から、演算速度を規定するのは CPU コアのクロック周波数でも SSE 演算のスループットでもないと考えられる。メモリ配置やロード命令の最適化を行なっても、なおもデータ転送がボトルネックになっていると思われる。

6 まとめ

PC ベースの遠隔会議システムを想定して、Intel IA32 プロセッサを用いて SAEC を実現した。ベクトル演算命令を有効に利用できるような種々の最適化を行い、最適化前のプログラムに対して約 4 倍の高速化に成功した。比較的低価格の PC でも SAEC の実時間処理が可能になるため、SAEC を用いた高品位遠隔会議システムを低価格で容易に導入可能になる。

入力信号の白色化による高速収束アルゴリズムの 検討

目次

| | | |
|-----|-------------------------|----|
| 1 | FIR 形予測誤差フィルタを含む適応フィルタ | 36 |
| 1.1 | 有色信号の白色化 | 36 |
| 1.2 | フィルタ構成 | 37 |
| 1.3 | 演算量 | 39 |
| 2 | 計算機シミュレーション | 39 |
| 2.1 | 入力信号が定常有色信号の時の出力誤差の収束特性 | 39 |
| 2.2 | 入力信号が音声信号の場合の出力誤差の収束特性 | 43 |
| 2.3 | 考察 | 45 |
| 3 | まとめ | 45 |

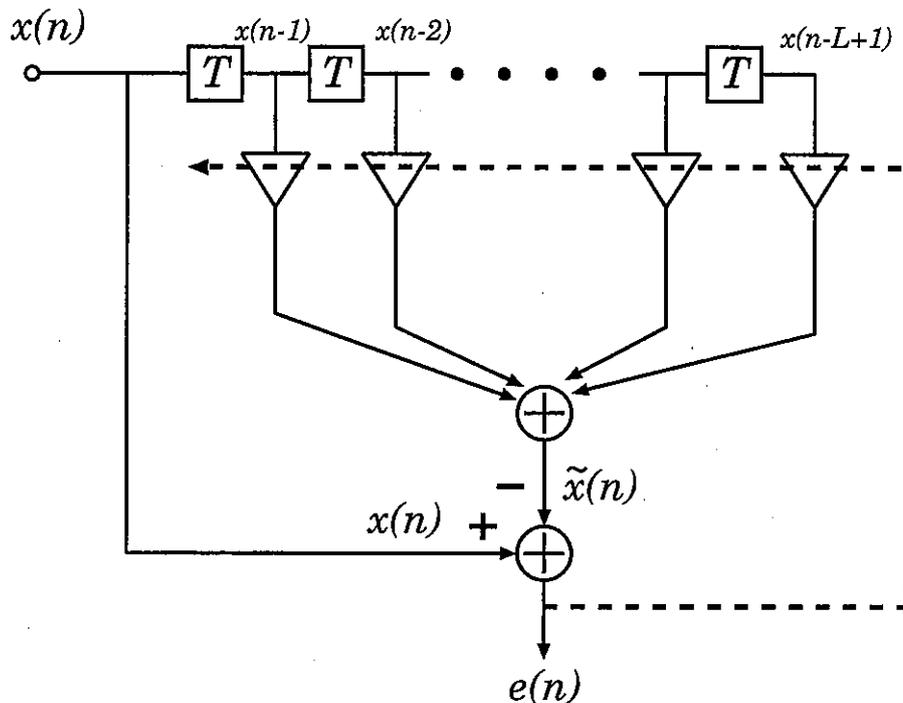


図 20 予測誤差フィルタ

1 FIR 形予測誤差フィルタを含む適応フィルタ

1.1 有色信号の白色化

適応フィルタの入力信号が有色である場合には、白色信号に比べて固有値広がりが大きいため、収束速度が遅くなる。そこで、有色信号を白色化することにより適応フィルタの収束速度の改善を図る。図 20 に示す FIR 形予測誤差フィルタによる白色化を考える。

この方法では、有色信号が生成されたシステムのモデル化を考える。システムの現在の出力 $x(n)$ は過去の信号の値となんらかの相関があるので、過去の N 個の出力の線形結合で予測できると考えられる。時刻 n におけるシステムの出力 $x(n)$ の予測値 $\tilde{x}(n)$ は

$$\tilde{x}(n) = -w_1(n)x(n-1) - w_2(n)x(n-2) - \cdots - w_{N-1}(n)x(n-N+1) \quad (1)$$

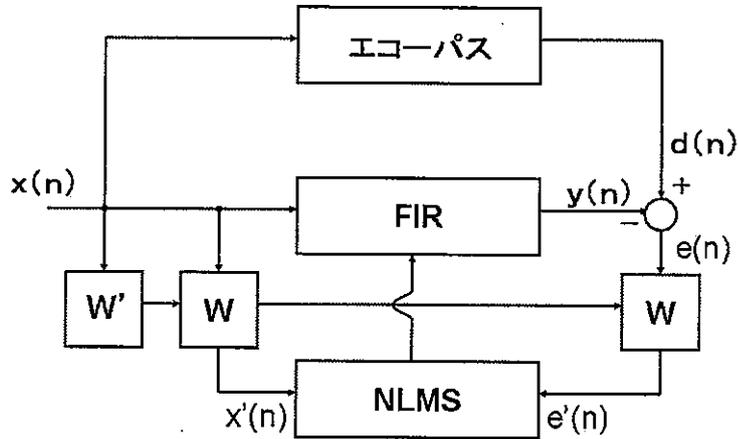


図 21 フィルタの構成

と定義する。このとき予測誤差 $e(n)$ は

$$e(n) = x(n) - \tilde{x}(n) = x(n) + \sum_{i=1}^{N-1} w_i x(n-i) \quad (2)$$

となり、 $e(n)$ の二乗期待値

$$J(n) = E[e(n)^2] \quad (3)$$

が最小になるように係数を適応的に更新していくことで、高精度の予測が行なえる。その結果、現在の信号から過去の信号との相関分を差し引くことで無相関な成分が出力される。つまりは白色化された信号が出力されることになる。このとき更新アルゴリズムには NLMS アルゴリズムを用いた。この方法では一つの白色化出力を得るために必要な計算量は $O(N)$ であり、比較的少ない演算量で済む。

1.2 フィルタ構成

FIR 形予測誤差フィルタを適応フィルタに適用する。そのブロック図を図 21 に示す。

学習アルゴリズムには NLMS アルゴリズムを用いている。この際、フィルタ係数 $w_i(n)$ の更新に用いるフィルタ入力 $x'(n)$ と誤差信号 $e'(n)$ を同じフィルタ w で処理する。また、フィルタ出力計算には入力 $x(n)$ をそのまま入力することによって、従来法で行なっていた同期学習が不用となる。また、このことにより、白色化フィルタの遅延が FIR の

出力を遅延させることもない。すると、フィルタ出力計算式と、フィルタ係数 $w_i(n)$ の更新式をそれぞれ (4) 式、(5) 式のようにあらわされる。

$$y(n) = \mathbf{w}^T(n)\mathbf{x}(n) \quad (4)$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \frac{\mathbf{x}'(n)}{\|\mathbf{x}'(n)\|^2} e'(n) \quad (5)$$

1.3 演算量

次に提案法の演算量について説明する。適応フィルタのタップ数を L_1 、白色化フィルタのタップ数を L_w とすると、演算量は表 5 のようになる。

表 5 演算量

| | |
|-------------|--|
| 適応フィルタの演算量 | $2 \times L_1 + \alpha$ |
| 白色化フィルタの演算量 | $3 \times L_w + \beta$ |
| 合計の演算量 | $2 \times L_1 + 3 \times L_w + \alpha + \beta$ |

ここで、 α は L_1 に依存しない定数、 β は L_w に依存しない定数とする。 $L_1 \gg L_w$ のとき、提案法の演算量は NLMS と同程度 (若干多い) となる。

2 計算機シミュレーション

2.1 入力信号が定常有色信号の時の出力誤差の収束特性

まず、入力信号が定常有色信号の場合のシミュレーションを行ない、出力誤差の収束特性を調べる。エコーパスとしては自動車内の実測データを用いた。定常有色信号は 2 次の AR 回路より生成する。

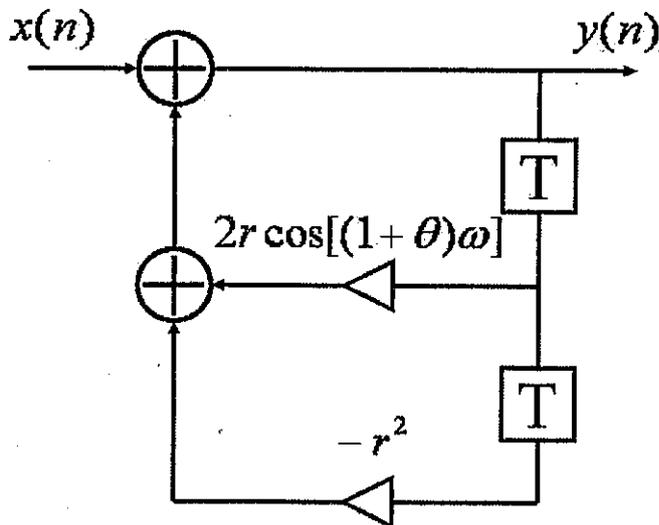


図 22 2次 AR モデル

定常有色信号の生成に用いる2次ARモデルを図22に示す。このときの出力は式(6)のようになる。

$$y(n) = x(n) + (2r \cos \varphi_1)y(n-1) - r^2y(n-2) \quad (6)$$

そして、極の位置は図23のようになる。今回用いる定常有色信号では、図22において $r = 0.9$ 、 $\theta = \frac{2}{\pi}$ とした定常有色信号を用いた。

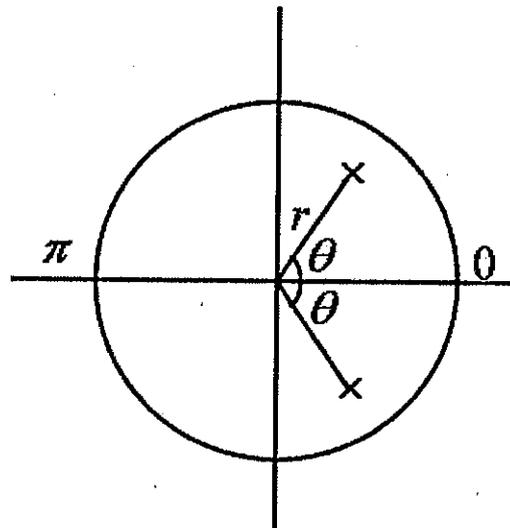


図23 極の位置

表6の条件下でのシミュレーション結果を図24に示す。白色化により収束速度が大幅に高速化されている。出力誤差が-20dBに達するまでに要する時間は約1/2になっている。

表7の条件下でのシミュレーション結果を図25に示す。学習開始後約30,000サンプルにおいては、出力誤差を最大約1dB改善している。しかし、50,000サンプル以降はほぼ同等の特性となっている。

表 6 シミュレーション条件 1

| | | |
|-----|--------------------------|------|
| 示す。 | 適応フィルタのタップ数 L_1 | 25 |
| | 適応フィルタのステップサイズ μ | 0.01 |
| | 予測誤差フィルタのタップ数 L_w | 25 |
| | 予測誤差フィルタのステップサイズ μ_0 | 0.01 |

表 7 シミュレーション条件 2

| | |
|--------------------------|------|
| 適応フィルタのタップ数 L_1 | 512 |
| 適応フィルタのステップサイズ μ | 0.01 |
| 予測誤差フィルタのタップ数 L_w | 20 |
| 予測誤差フィルタのステップサイズ μ_0 | 0.01 |

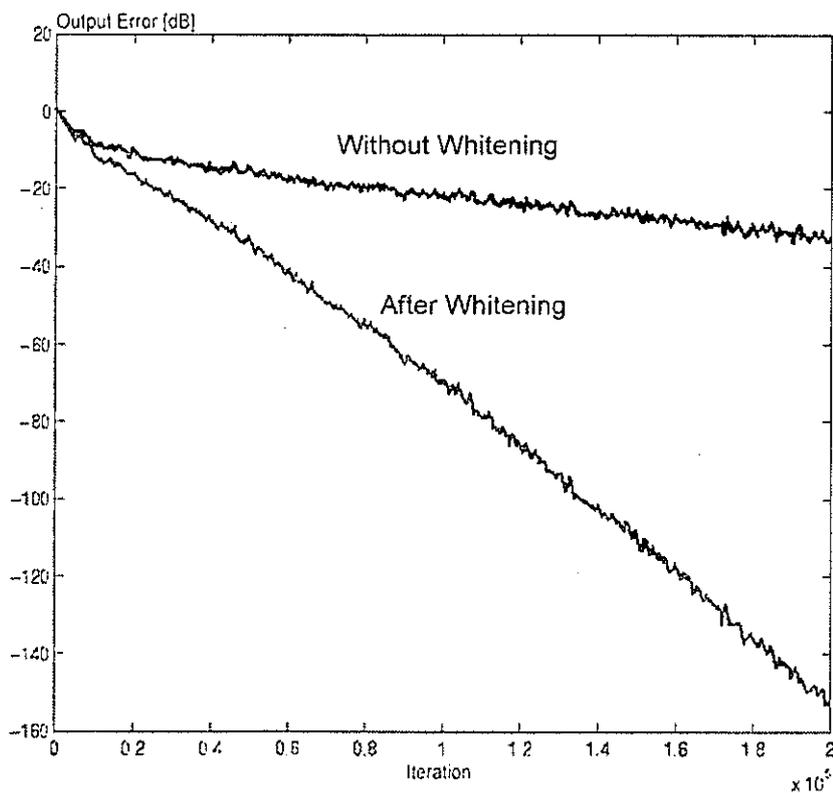


図 24 条件 1 の時の出力誤差の収束特性

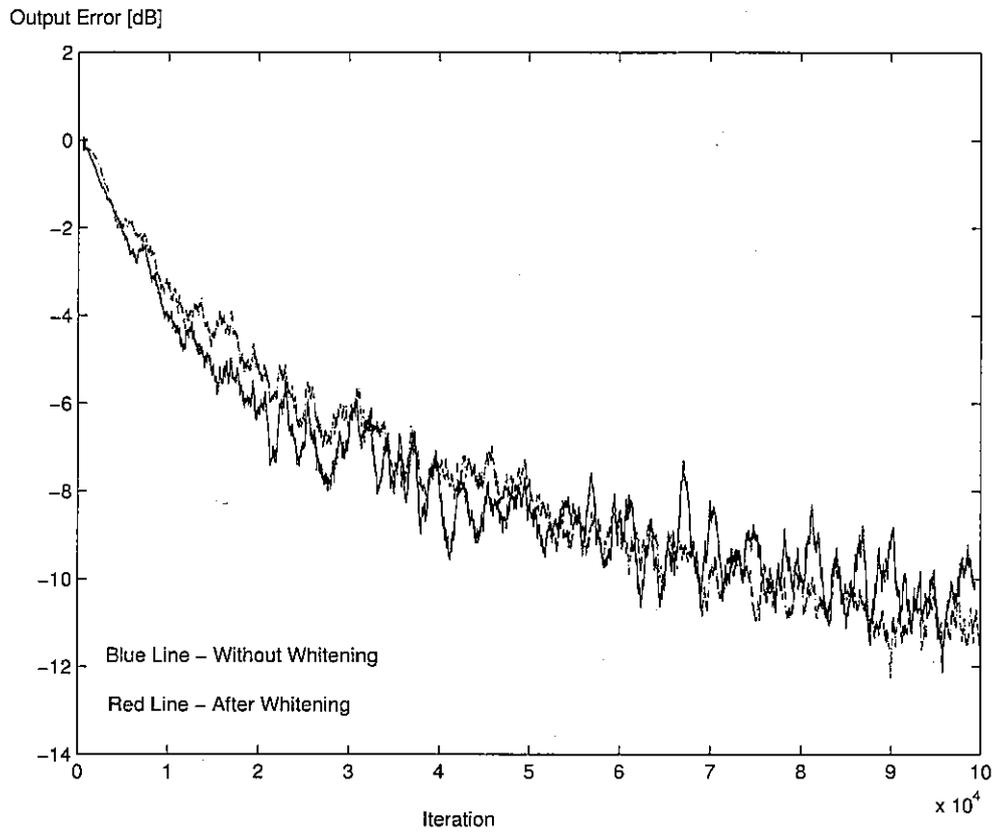


図 25 条件 2 の時の出力誤差の収束特性

2.2 入力信号が音声信号の場合の出力誤差の収束特性

今回シミュレーションに用いた女性の音声信号を図 26 に示す。

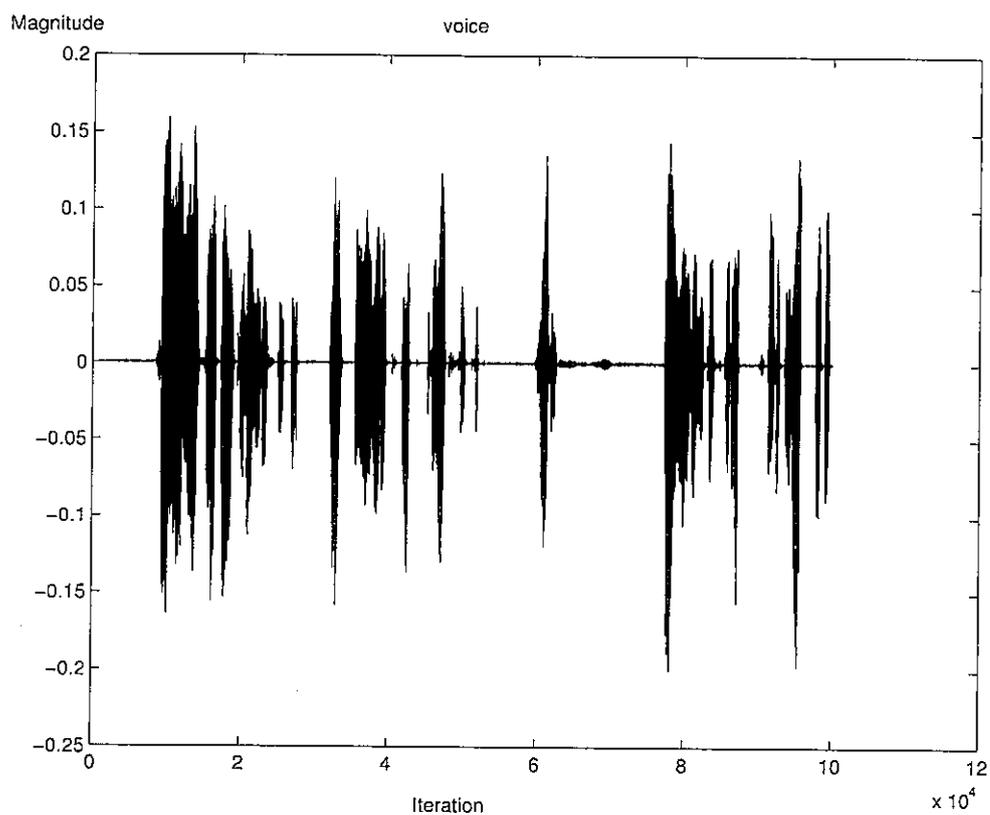


図 26 音声信号

次に、入力信号に音声信号を用いて、表 8 に示す条件下でシミュレーションを行なった結果を図 27 に示す。学習開始後約 15,000 サンプルにおいては、出力誤差を最大約 5dB 改善している。しかし、30,000 サンプル以降はほぼ同等の特性となっている。

表 8 シミュレーション条件 3

| 音声信号 | 女性の声 |
|--------------------------|-------|
| 適応フィルタのタップ数 L_1 | 512 |
| 適応フィルタのステップサイズ μ | 0.01 |
| 予測誤差フィルタのタップ数 L_w | 20 |
| 予測誤差フィルタのステップサイズ μ_0 | 0.001 |

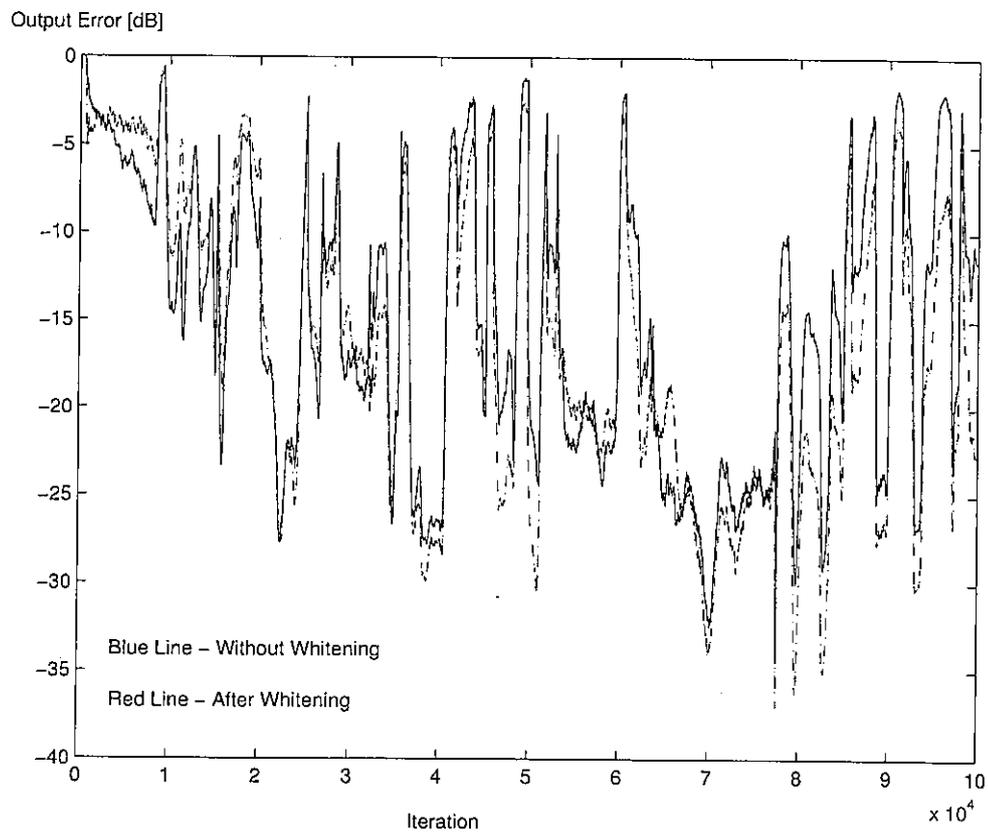


図 27 条件 3 の時の出力誤差の収束特性

2.3 考察

シミュレーション結果より、入力信号が定常有色信号の際に適応フィルタのタップ数が少ない場合は出力誤差の収束特性に改善がみられる。しかし、適応フィルタのタップ数が多い場合は若干の改善しかみられない。また、入力信号が音声信号の場合は、出力誤差の収束特性は初期は若干の改善がみられるが、定常信号の時ほどの効果はみられないことが分かる。

図 28 に、 $L_1 = 512$ タップである場合における白色化信号 $u'(n)$ 計算部のブロック図を示す。

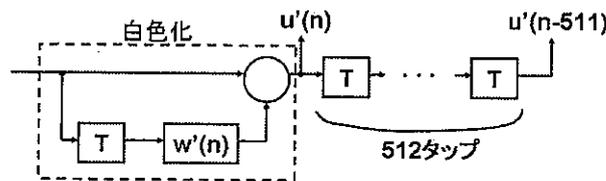


図 28 適応フィルタのタップ数が 512 タップの時

図 28 において、 $u'(n)$ の計算には $w'(n)$ を用いているが、 $u'(n-511)$ の計算には $w'(n-511)$ を用いている。したがって、 $u'(n)$ と $u'(n-511)$ は異なる白色化係数で計算されている。そのため、適応フィルタのタップ数が多い場合や、入力信号が時変信号の場合は影響を受けやすいと考えられる。

3 まとめ

本研究では、エコーキャンセラにおいて入力信号の白色化を行なうことによって、変色信号に対する収束速度を改善する手法について検討した。入力信号が定常有色信号でタップ数が少ない場合には、大きな改善がみられた。しかし、タップ数が多い場合や、入力信号が非定常である場合には、大きな改善はみられなかった。入力信号が音声信号の場合や、タップ数が大きい場合に対する、誤差出力の収束特性が改善されるようなアルゴリズムを検討する必要がある。

参考文献

目次

参考文献

49

参考文献

- [1] A. Hirano and A. Sugiyama, "Convergence characteristics of a multi-channel echo canceller with strongly cross-correlated input signals - analytical results -," *Proc. of 6th DSP Symposium*, pp. 144–149, 11 1991.
- [2] A. Sugiyama, A. Hirano, and K. Nakayama, "Acoustic echo cancellation for conference systems," *Proc. of EUSIPCO 2004*, pp. 17–20, 9 2004.
- [3] A. Hirano, K. Nakayama, and K. Takebe, "Stereophonic acoustic echo canceller with pre-processing - second-order pre-processing filter and its convergence -," *Proc. of 8-th IWAENC*, pp. 63–66, 9 2003.
- [4] A. Sugiyama, Y. Joncour, and A. Hirano, "A stereo echo canceler with correct echo-path identification based on an input-sliding technique," *IEEE Trans. SP*, vol. 49, no. 11, pp. 2577–2587, Nov. 2001.
- [5] J. Benesty, D. Morgan, and M. Sondhi, "A better understanding and an improved solution to the specific problems of stereophonic acoustic echo cancellation," *IEEE Trans. Speech and Audio*, vol. 6, no. 2, pp. 156–165, Mar. 1998.
- [6] M. Ali, "Stereophonic acoustic echo cancellation system using time-varying all-path filtering for signal decorrelation," *Proc. of ICASSP'98*, pp. 3689–3692, May 1998.
- [7] A. Hirano, K. Nakayama, and K. Watanabe, "Convergence analysis of stereophonic echo canceller with pre-processing — relation between pre-processing and convergence —," *Proc. of ICASSP '99*, pp. 861–864, Mar. 1999.
- [8] A. Hirano, K. Nakayama, D. Someda, and M. Tanaka, "Alternative learning algorithm for stereophonic acoustic echo canceller without pre-processing," *IEICE Trans. Fundamentals*, vol. E87-A, no. 8, pp. 1958–1964, 8 2004.
- [9] A. Hirano, K. Nakayama, and J. Mizuno, "RLs algorithm and its stabilization in alternative learning algorithm for stereophonic acoustic echo canceller," *Proc. of 19-th SIP symposium*, 11 2004.

-
- [10] 平野晃宏, 中山謙二, 染田大輔, and 田中正彦, “分割学習法に基づくステレオエコーキャンセラの SIMD マルチプロセッサ実現に適したメモリとタスクの割当,” 2004年電子情報通信学会総合大会講演論文集, pp. S-22-23, 3 2004.
- [11] T. Fujii and S. Shimada, “A note on multi-channel echo cancellers,” *Technical Reports of IEICE on CS*, pp. 7-14, 3 1985.
- [12] A. Hirano and A. Sugiyama, “A compact multi-channel echo canceller with a single adaptive filter per channel,” *Proc. of IEEE 1992 ISCAS*, pp. 1922-1925, 5 1992.
- [13] M. M. Sondhi and D. R. Morgan, “Stereophonic acoustic echo cancellation — an overview of the fundamental problem,” *IEEE SP Letters*, vol. 2, no. 8, pp. 148-151, Aug. 1995.
- [14] “ADSP-2136x SHARC processor hardware reference,” May 2006.
- [15] “ADSP-BF561 Blackfin processor hardware reference,” July 2005.
- [16] “Spartan-3 FPGA family: Complete data sheet,” April 2006.
- [17] “Intel 64 and IA-32 architectures software developer’s manual volume 1: Basic architecture,” May 2007.
- [18] J. Nagumo and A. Noda, “A learning method for system identification,” *IEEE Trans. AC*, vol. 12, no. 3, pp. 282-287, Mar 1967.
- [19] “Intel 64 and IA-32 architectures software developer’s manual volume 2a: Instruction set reference, a-m,” May 2007.
- [20] “Intel 64 and IA-32 architectures software developer’s manual volume 2b: Instruction set reference, n-z,” May 2007.
- [21] “Intel 64 and IA-32 architectures optimization reference manual,” May 2007.
- [22] B. Juurlink A. Shahbahrani and S. Vassiliadis, “Performance impact of misaligned accesses in SIMD extensions,” *Proc. of ProRISC 2006*, pp. 334-342, 2006.
- [23] David A. Patterson and John L. Hennessy, “Computer organization and design,”
- [24] A. Sugiyama A. Hirano and S. Ikeda, “DSP implementation and performance evaluation of sparse-tap adaptive fir filters with tap-position control,” *Proc. of ICASSP '96*, vol. 3, pp. 1295-1298, May 1996.