

方程式誤差法を用いた音響用エコーキャンセラの検討  
A study on an acoustic echo canceller using an equation error method

○福士達也

工藤憲昌<sup>†</sup>

田所 嘉昭<sup>‡</sup>

○Fukushi Tatsuya

Norimasa Kudoh<sup>†</sup>

Yoshiaki Tadokoro<sup>‡</sup>

八戸高専 専攻科

<sup>†</sup>八戸高専

<sup>‡</sup>豊橋技科大

Hachinohe National College of Tech. Advanced Engineering Course

<sup>†</sup>Hachinohe National College of Tech. <sup>‡</sup>Toyohashi University of Tech.

キーワード：方程式誤差法 (equation error method), エコーキャンセラ (echo canceller),  
適応フィルタ(adaptive filter)

連絡先：〒039-1192 八戸市田面木上野平 16-1 八戸高専 電気情報工学科

tel:0178-27-7281, e-mail:kudohk-e@hachinohe-ct.ac.jp

## 1. はじめに

TV 電話システムや、ハンズフリー電話といった音響的な反響路をもつ系を使用したとき、自分の声がエコーとして聞こえてくることがある。これは、スピーカにから出力された音声は壁や天井などに反響してマイクに入力された音声であり、一定の通話品質を保つためには除去する必要がある。

拡声通信系や長距離電話回線で使用される適応デジタルフィルタでは、未知系のインパルス応答長が数百ミリ秒以上と非常に長い。適応デジタルフィルタにおいては、(1)入力信号が音声信号のように有色信号である場合適応パラメータの収束速度が遅い、(2)未知系のインパルス応答長が長い場合に回路規模および適応パラメータ更新の演算量が増加する、等の解決策を検討することが課題である。(1)の課題に対しては、時間領域或いは直交変換領域で入力信号の直交化を行うことを基本にして多くの試みがなされている。(2)に対しては IIR 型で推定系を構成する検討が行われている。[1], [2]

本稿では、エコー経路を効率的に実現できる ARMA(Auto regressive Moving Average)型のエコーキャンセラについて検討を行ったので報告する。

提案法は、Fig.1 のような通話モデルにおいて生じる遅延要素の大きいエコー成分に効率的に対応するために、推定系の次数を大きくし、随時安定化を行うアルゴリズムである。

以下に本稿の構成を示す。2.ではエコーキャンセラについて概説し、2.1 では適応エコーキャンセラについて、2.2 では提案法である ARMA 型エコーキャンセラについて述べる。3.では学習システムについて、4.では全域通過フィルタを用いた安定化について述べ、5.は数値例、6.はまとめである。

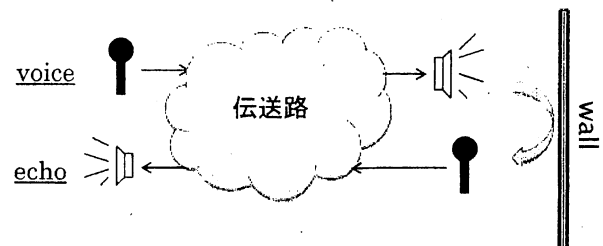


Fig.1 通話モデル

## 2. エコーキャンセラ

Fig. 2-1 に、基本的なエコーキャンセラの構成を示す。エコーを発生する経路を未知系、その出力を  $y(n)$  とし、推定系で随時その特性を学習しながら擬似エコー成分  $\hat{y}(n)$  を生成し、 $y(n)$  から減算することで除去する構成である。

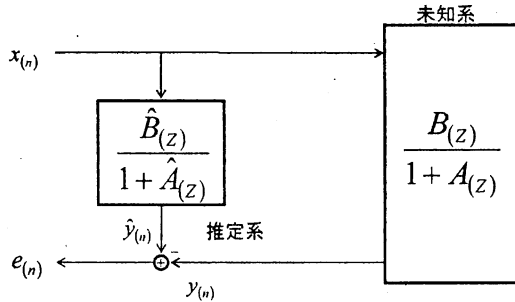
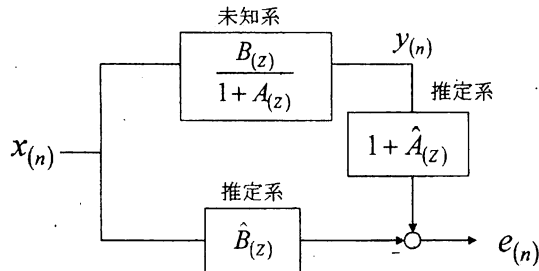


Fig. 2-1 エコーキャンセラの構成

### 2.1 適応エコーキャンセラ

ARMA 型エコーキャンセラには、係数の学習時に Fig. 2-1 に示す構成をとる出力誤差型(並列型)と、Fig. 2-2 に示す方程式誤差型(直並列型)の 2 つがある。出力誤差型は極を推定するため発振の可能性があり、また誤差局面が多峰性なので最適値に収束する保証は無い。これに対して、方程式誤差型は必ず最適値に収束する単峰性となっていることから、学習時には方程式誤差型を用いて未知系  $H(z) = B(z)/(1+A(z))$  を推定し、相手側発話時といった非学習時には Fig. 2-1 のように未知系に対し並列に動作する。ここで、 $N, M$  はそれぞれ  $B(z), A(z)$  の次数である。



$$B(z) = \sum_{n=0}^{N-1} b_n z^{-n} \quad A(z) = \sum_{m=1}^{M-1} a_m z^{-m} \quad (2-1)$$

$$e(n) = y(n) - \hat{y}(n) \\ = y(n) - \left\{ \sum_{i=0}^{N-1} \hat{b}_i x(n-i) - \sum_{i=1}^{M-1} \hat{a}_i y(n-i) \right\} \quad (2-2)$$

Fig. 2-2 方程式誤差型(直並列系)

### 2.2 安定性を保証した ARMA 型エコーキャンセラ

Fig. 2-2 に示す方程式誤差型では未知系の極の部分  $(1+A(z))$  を零点として推定しているため、システムの安定性が保証される。しかし、Fig. 2-1 の構成となる相手通話時には零点が極となるため、単位円外に極が存在した場合、安定性が保証されない。このため、零点の半径を把握する必要があるが、系の次数が 3 次以上となると因数分解による零点の判別を容易に行うことができない。以上の理由より、2 次システムの縦続接続を考慮する。

例として  $M=6$  における  $1+\hat{A}(z)$  を実現した式、系を式(2-3)、Fig. 2-3 に示す。  $1+\hat{A}(z)$  を、2 次のフィルタの縦続形で構成することにより、根の公式を用いて容易に零点を算出することができる。

$$1+\hat{A}(z) = \prod_{i=1}^{M/2-1} (1+\alpha_i z^{-1} + \beta_i z^{-2}) \quad \dots(2-3)$$

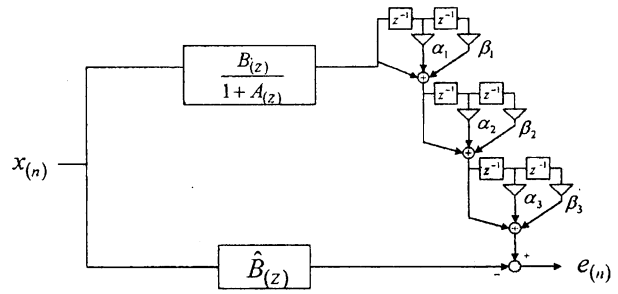


Fig. 2-3 縦続システムを用いた構成

(2-3)式を展開し、(2-1)式の  $1+\hat{A}(z)$  の係数を対応させると、

$$\begin{aligned} \hat{a}_1 &= \alpha_1 + \alpha_2 + \alpha_3 \\ \hat{a}_2 &= \alpha_1 \alpha_2 + \alpha_1 \alpha_3 + \alpha_2 \alpha_3 + \beta_1 + \beta_2 + \beta_3 \\ \hat{a}_3 &= \alpha_1 \alpha_2 \alpha_3 + (\alpha_2 + \alpha_3) \beta_1 \\ &\quad + (\alpha_1 + \alpha_3) \beta_2 + (\alpha_1 + \alpha_2) \beta_3 \\ \hat{a}_4 &= \alpha_1 \alpha_2 \beta_3 + \alpha_1 \alpha_3 \beta_2 + \alpha_2 \alpha_3 \beta_1 \\ &\quad + \beta_1 \beta_3 + \beta_2 \beta_3 + \beta_1 \beta_2 \\ \hat{a}_5 &= \alpha_1 \beta_2 \beta_3 + \alpha_2 \beta_1 \beta_3 + \alpha_3 \beta_1 \beta_2 \\ \hat{a}_6 &= \beta_1 \beta_2 \beta_3 \end{aligned} \quad \dots(2-4)$$

となる。この係数の関係式は縦続系の段数の増加に伴って長くなり、複雑になる。

### 3. 係数更新システム

#### 3.1 LMS 法

適応アルゴリズムとして, LMS (Least Mean Square) を用い, 入力信号  $y(n)$  と推定信号  $\hat{y}(n)$  の誤差  $e(n)$  の二乗値が最小になるよう, 推定系のフィルタ係数を逐次更新してゆく. 更新には二乗誤差の瞬時勾配を用い, 推定系は未知系の次数以上の次数とし, 係数はそれぞれ  $\hat{b}_n, \hat{a}_n$  とする. (3-1)式に Fig.2-2 における二乗誤差の瞬時勾配を示す.

$$\begin{aligned}\hat{\nabla} a_i(n) &= \frac{\partial}{\partial \hat{a}_i} e^2(n) = 2e(n) \frac{\partial e(n)}{\partial \hat{a}_i} \\ &= 2e(n)y(n-i) \quad \dots(3-1-a)\end{aligned}$$

$$\begin{aligned}\hat{\nabla} b_i(n) &= \frac{\partial}{\partial \hat{b}_i} e^2(n) = 2e(n) \frac{\partial e(n)}{\partial \hat{b}_i} \\ &= -2e(n)x(n-i) \quad \dots(3-1-b)\end{aligned}$$

これより更新式は, (3-2)式で与えられる. ここで,  $\mu$  は学習のステップサイズである.

$$\begin{aligned}\hat{a}_i(n+1) &= \hat{a}_i(n) - \mu \nabla \hat{a}_i(n) \\ &= \hat{a}_i(n) - \mu e(n)y(n-i) \quad \dots(3-2-a)\end{aligned}$$

$$\begin{aligned}\hat{b}_i(n+1) &= \hat{b}_i(n) - \mu \nabla \hat{b}_i(n) \\ &= \hat{b}_i(n) + \mu e(n)x(n-i) \quad \dots(3-2-b)\end{aligned}$$

#### 3.2 縦続系における係数更新

Fig.2-3 に示す縦続系の係数更新においても二乗誤差の瞬時勾配を用いる. 縦続系の瞬時勾配は(3-3)式に示すように, 誤差を(2-6)式で表される各係数で偏微分した値の和で表される.

$$\begin{aligned}\hat{\nabla} \alpha_i(n) &= \frac{\partial e(n)^2}{\partial \alpha_i} = 2e(n) \frac{\partial e(n)}{\partial \alpha_i} \\ &= 2e(n) \sum_{k=1}^6 \frac{\partial e(n)}{\partial \alpha_k} \frac{\partial \alpha_k}{\partial \alpha_i} \quad \dots(3-3-a)\end{aligned}$$

$$\begin{aligned}\hat{\nabla} \beta_i(n) &= \frac{\partial e(n)^2}{\partial \beta_i} = 2e(n) \frac{\partial e(n)}{\partial \beta_i} \\ &= 2e(n) \sum_{k=1}^6 \frac{\partial e(n)}{\partial \alpha_k} \frac{\partial \alpha_k}{\partial \beta_i} \quad \dots(3-3-b)\end{aligned}$$

(3-3)式の瞬時勾配を用いて導出した更新式の1つを(3-4)式に示す.

$$\begin{aligned}\alpha_{1(n+1)} &= \alpha_{1(n)} - \mu e(n) [y_{(n-1)} + y_{(n-2)} \{\alpha_{2(n)} + \alpha_{3(n)}\} \\ &\quad + y_{(n-3)} \{\alpha_{2(n)} \alpha_{3(n)} + \beta_{2(n)} + \beta_{3(n)}\} \\ &\quad + y_{(n-4)} \{\alpha_{2(n)} \beta_{3(n)} + \alpha_{3(n)} \beta_{2(n)}\} + y_{(n-5)} \beta_{2(n)} \beta_{3(n)}] \\ &\quad \dots(3-4)\end{aligned}$$

#### 3.3 縦続系の係数更新アルゴリズム

これまで, (2-4)式の係数計算式と, 偏微分を用いた各  $\alpha, \beta$  の更新式導出は手計算により導出し, プログラムに実装していた. しかし, 段数の増加と共に畳込みの式の計算量が増加, 複雑化してしまい, 高次数化が困難となっていたのに加え, C 言語で実現しているため, プログラム上での文字式, 偏微分計算といった数式処理が実現不可能であった.

そこで, プログラム上において, 段数と次数を入力することで係数値, 更新値計算をするアルゴリズムについて検討した. 係数値の計算が畳込みで計算できることを利用して段数ごとに次数に沿って計算していく. 式(3-5)は  $n$  段目,  $i$  次の係数値を表している.

$$a_{n-i} = a_{(n-1)-i} + a_{(n-1)-(i-1)} \alpha_n + a_{(n-1)-(i-2)} \beta_n \quad (3-5)$$

式(2-5)より, 前段の係数値に, 前段の計数值と現在の段の  $\alpha, \beta$  との積を加算していく形となっている. この式の前段の係数値を用いて, で偏微分すると

$$\frac{\partial a_i}{\partial \alpha_n} = a_{(n-1)-(i-1)} \quad (3-6)$$

$$\frac{\partial a_i}{\partial \beta_n} = a_{(n-1)-(i-2)}$$

のように偏微分値を前段の係数値で表すことができる. 最終段目以外の  $\alpha, \beta$  については計算順序を変更することにより同様に計算できる.

#### 4. 全域通過フィルタ(APF)による安定化

相手側発話時に適応される Fig.2-1 の構成で, 未知系の極が単位円に近く, 適応エコーキャンセラの推定系の零点が単位円の極半径 1 を超えてしまうと, フィルタが不安定になってしまう. この不安定な根を判別し, 全域通過フィルタ(APF: All Pass Filter)を通す, という操作を前項の縦続系の計算に追加することによって安定性を確保する.

今, Fig.2-3 の構成の 1 段目の 2 次のフィルタを考える. この零点を, 根の公式よりそれぞれ  $Z_{1,2}$  とすると,

$$1 + \alpha_1 z^{-1} + \beta_1 z^{-2} = 0 \quad \dots(4-1)$$

より,

$$z_{1,2} = \frac{-\alpha_1 \pm \sqrt{\alpha_1^2 - 4\beta_1}}{2} \quad \dots(4-2)$$

と表される. このとき算出された零点の大きさが 1 を超えた場合に, 全域通過フィルタを通

して零点の大きさを単位円内の相反な位置に折り返す処理を追加する。

全域通過フィルタのシステム関数を,(4-3)式に示す。

$$H_{op} = \frac{Z^{-1} - Z_1^*}{1 - Z_1 Z^{-1}} \quad \dots(4-3)$$

$$\ast Z_1 = a_1 e^{j\theta} \quad \frac{1}{Z_1^*} = \frac{1}{a_1} e^{j\theta} = Z_2$$

この式の  $Z_1$  と  $Z_2$  は相反な位置にあり, それぞれ極と零点となっている. 不安定な零点が, APF の極によって打ち消しあい, APF の安定な零点が残る, つまり単位円を基準として相反な位置に折り返される形となる。

Z 平面における全域通過フィルタの適用の様子を Fig.4 に示す。

この全域通過フィルタは, 振幅特性は変化しないが, 位相特性が変化してしまう問題点があるが, 今後の課題である。

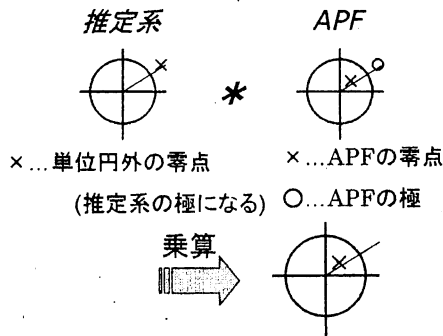


Fig. 4 Z 平面における全域通過フィルタの適用

## 5. 数値例

B(z)に20次のLPF, (1+A(z))に6個の極を用いた未知システムを用いてシミュレーションを行った. この未知システムの特性を Fig5-1 に示す. 入力には分散が1の白色雑音を, 音声(母音)を模擬したフィルタを通して有色化した信号を用いた. 各パラメータは以下のように設定した。

ステップサイズ  $\mu = 0.01$

繰り返し回数  $n = 10000$

観測雑音分散  $\sigma_y^2 = 0.1$

Fig5-2にERLE(Echo Return Loss Enhancement)の結果を示す. ERLEとはエコーキャンセラの性能指標であり, 式(5-1)で定義される. なお, ERLEは, 20回の独立した試行の集合平均をとっている。

$$ERLE(n) = 10 \log_{10} \left( \frac{y(n)^2}{e(n)^2 + 10^{-8}} \right) [dB] \quad \dots(5-1)$$

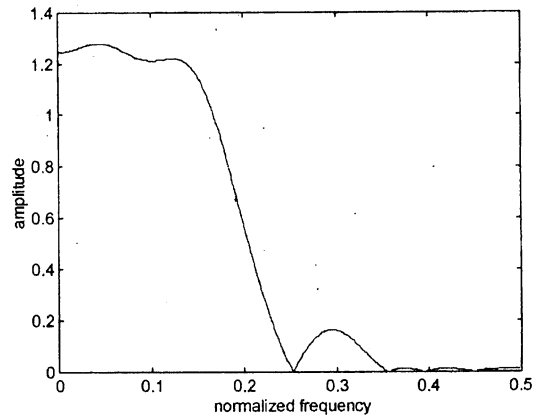


Fig. 5-1 未知システムの例

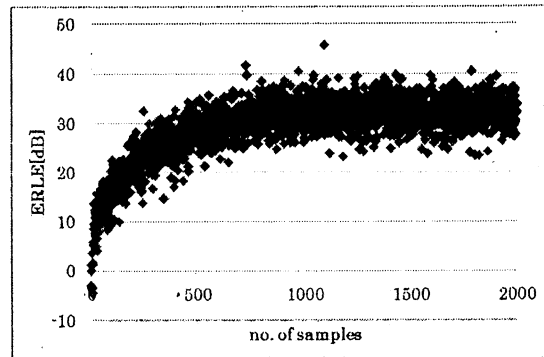


Fig. 5-2 ERLE

## 6. まとめ

ERLEの結果より, 十分な性能値である40[dB]に近い値に到達していることがわかる. 今後は, DSPに実装し, 音声のような有色信号を用いた学習の検討, ダブルトーク検出等の実環境に沿った構成の実現を行っていき, 全域通過フィルタの位相特性変化という問題にも位相補償等の方法を用いて対処する予定である。

## 7. 参考文献

- [1] 小林 "直交関数系を用いたエコーキャンセラ", 信学論 B
- [2] 工藤, 田所 "位相補償器を縦続接続したIIR適応フィルタの検討", SICE講演論文集 (1996)
- [3] B.widrow, S.D.Sterns, "Adaptive signal Processing", Prentice-Hall.(1985)
- [4] 福士, 工藤, 田所 "方程式誤差法を用いたVoIP用エコーキャンセラの検討", 電気関係学会東北支部連合大会 講演論文集 2F09 (2010)