## ■ 乗算を用いない適応アルゴリズム

2022.11.1

（有）ケプストラム

乗算を用いずにNLMS(Normalized LMS)アルゴリズムに近い動作をする適応アルゴリズムをご紹介します.
ただし、適応アルゴリズムとして有効に動作する条件とその性能には制約があります.
かなり例外的な事例かもしれませんが、当社ではとあるやや特殊なアプリケーションに適用してNLMSよりも良好な処理結果が得られています.

一般的なNLMSアルゴリズムは(1)式のようになります.
x：入力信号、e：誤差信号、w：適応フィルタ係数

$$w_{n+1}[k] = w_n[k] + 2.0 \cdot \mu \cdot \frac{e[n] \cdot x[n-k]}{\overline{x^2[n]}} \qquad (1)$$

乗算を用いない適応アルゴリズムは(2)式のようになります.
(2)式中の（2.0・$\mu$）は定数になるので、適応フィルタ係数 w の更新演算に乗算は不要で、加算と符号反転のみの処理になります.

$$\begin{aligned} w_{n+1}[k] &= w_n[k] + 2.0 \cdot \mu \cdot \mathrm{sign}(e[n]) \cdot \mathrm{sign}(x[n-k]) \\ &= w_n[k] + (2.0 \cdot \mu) \cdot \mathrm{sign}(e[n]) \cdot \mathrm{sign}(x[n-k]) \qquad (2) \end{aligned}$$

(3)式の導出は非常に単純で、(1)式の一部を(3)式のように変形して得られます.

$$\frac{e[n] \cdot x[n-k]}{\overline{x^2[n]}} \approx \frac{e[n]}{|x[n]|} \cdot \frac{x[n-k]}{|x[n]|} \approx \frac{e[n]}{|x[n]|} \cdot \frac{x[n-k]}{|x[n-k]|} \approx \frac{e[n]}{|x[n]|} \cdot \mathrm{sign}(x[n-k]) \qquad (3)$$

図１のシステム同定の構成の適応フィルタを考えると、同定対象の伝達特性が H(ω) が極端な狭帯域特性を持っているのでもなければ、収束の初期状態では e[n]≒x[n] と考えられますから、e[n]/|x[n]|＝sign(e[n])となって(2)式が導かれます.

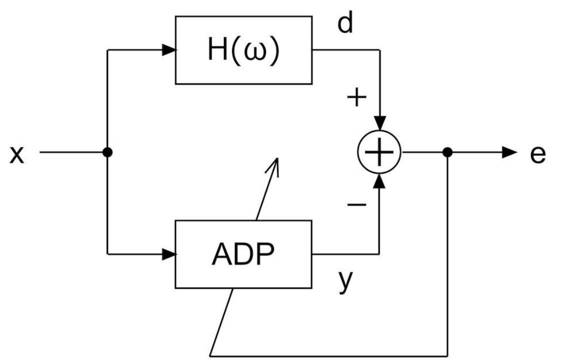
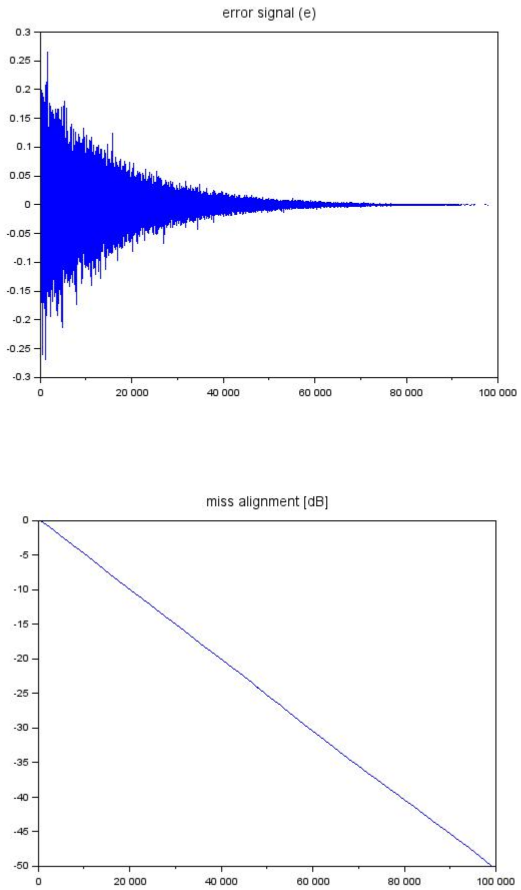


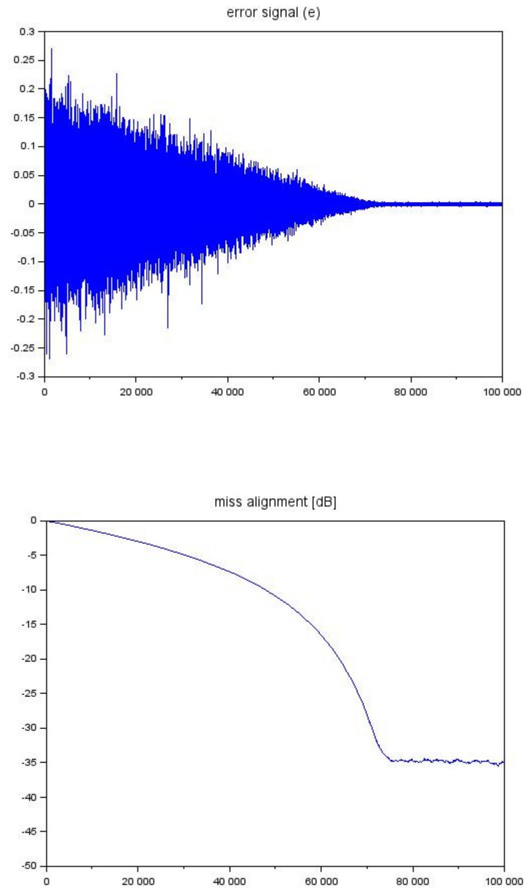

図１　システム同定の構成の適応フィルタ

そんな怪しげな説明を信じられるものか！まともな証明になってない！と怒られるかもしれませんが、実際にシミュレーションをしてみると、ちゃんと適応アルゴリズムとして動作することを確認できます.

白色雑音を入力信号とした場合のシミュレーション結果は下図のようになります. （図１の構成）

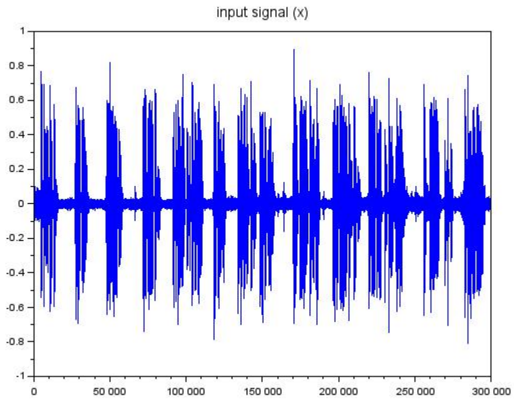



NLMS (1)式

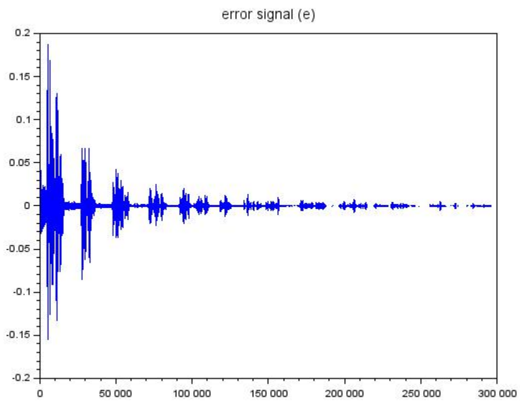


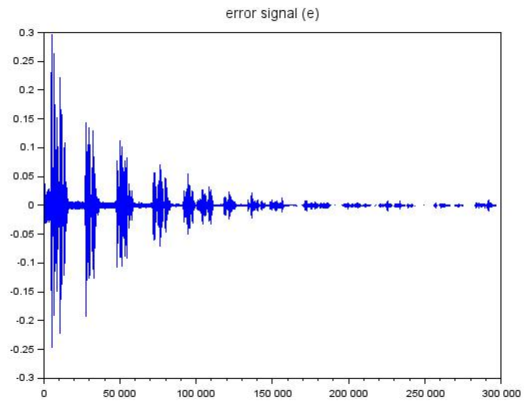

乗算なしの適応アルゴリズム (2)式

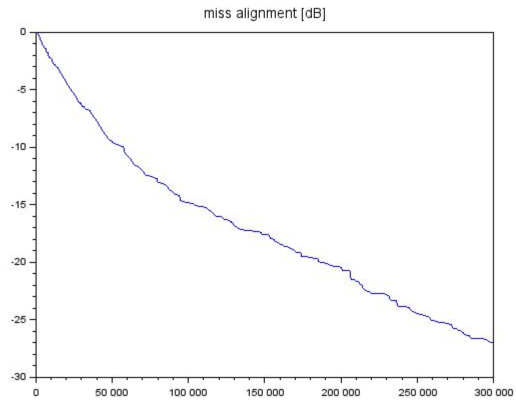
音声を入力信号とした場合のシミュレーション結果は下記のようになります．（図１の構成）

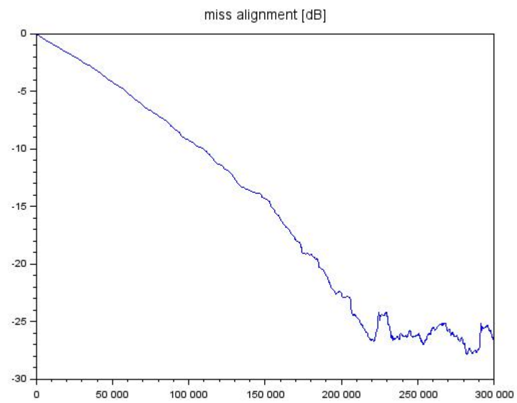



入力信号 x（音声）




NLMS (1)式

乗算なしの適応アルゴリズム (2)式

シミュレーション結果を見ると、適応フィルタとしての性能、収束特性にはかなり微妙なところがありますが、まったく実用にはならないというほどでは無いと思います.

厳密な解析は出来ませんが、おおむね以下のような条件を満たしていれば確実に収束するのではないかと推測されます

1. 図１のシステム同定の構成の場合、同定対象 H(ω) は極端な狭帯域特性では無い.
2. 同様に H(ω) の直流利得はゼロである.
3. 入力信号は直流成分を含まない交流信号である.

一方、以下のような応用では実用にはならない可能性が高いです.

1. いわゆるダブルトーク状態が生じるエコーキャンセラ
2. ノイズキャンセラの構成の適応フィルタ

当社ではとあるやや特殊なアプリケーションに乗算を用いない適応アルゴリズムを適用して、NLMSよりも良好な処理結果が得られました.　なぜ良好な結果が得られたかというと、Normailzeのための平均パワー（二乗ノルム）の計算が不要、したがって平均・積分操作のためのフィルタも不要で、平均・積分フィルタの処理遅延・位相特性がシステム全体の特性には影響を及ぼさないことがプラスに作用したからではないかと推測されます.

別の表現の仕方をすると以下のようになります.

1. NLMS(Normalized LMS)アルゴリズムはステップサイズ・パラメータμで定まる時定数を持っているが、実はnormailzeのためのパワー計算（二乗ノルム計算）のための平均・積分フィルタの時定数も含んでいる.
2. 乗算を用いない適応アルゴリズムにステップサイズ・パラメータμ以外の時定数は無い.

何よりも乗算を用いない適応アルゴリズムの良いところは演算が簡単になることではなくて、不規則信号を処理対象とした場合に、最適な平均・積分フィルタを選択する必要が無いことかもしれません.

使い方を適切に選べば、乗算を用いない適応アルゴリズムは他にも有効な利用方法があるのではないかと思われます.