

# 信号処理レベルアップ! 適応フィルタ

川村 新

## 6-1：システム同定

収録フォルダ：71\_SysId

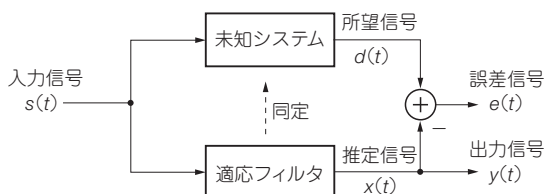


図1 未知システムと同じ構造を適応フィルタで作出す…システム同定

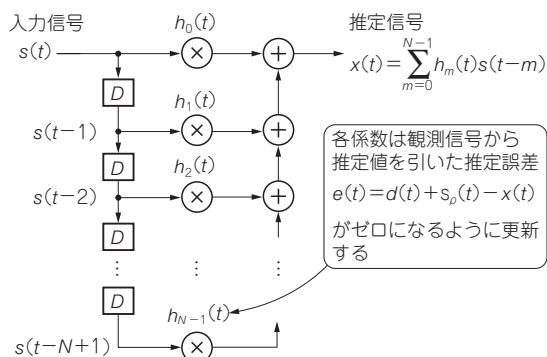
リスト1 システム同定のプログラムDD\_SysId.c (抜粋)

```
(a)冒頭の宣言部
#define MEM_SIZE 16000 // 音声メモリのサイズ
#define N 2000 // フィルタ次数

(b)信号処理用変数の宣言部
double s[MEM_SIZE+1]={0}; // 入力データ格納用変数
double y[MEM_SIZE+1]={0}; // 出力データ格納用変数
int i; // forループ用
double d; // 所望信号
double x, e; // 観測信号, 誤差信号
double g[N+1]={0}; // 未知系の係数
double h[N+1]={0}; // 適応フィルタ係数
double norm, mu; // ノルムとステップ・サイズ

(c)変数の初期設定部
mu=0.05; // ステップ・サイズ
g[0]=1.0;g[N-1]=1.0; // 未知システム(エコー)

(d)メイン・ループ内 Signal Processing 部
s[t] = input/32768.0; // 音声の最大値を1とする(正規化)
x=0,d=0,norm=0;
for(i=0;i<N;i++){
    d = d + g[i] * s[(t-i+MEM_SIZE)%MEM_SIZE]; // 所望信号作成
    x = x + h[i] * s[(t-i+MEM_SIZE)%MEM_SIZE]; // 適応フィルタ出力 = 所望信号のレプリカ
    norm=norm+s[(t-i+MEM_SIZE)%MEM_SIZE]*s[(t-i+MEM_SIZE)%MEM_SIZE]; // ノルムの計算
}
e = d - x; // 推定誤差
if(norm>0.00001){ // ノルムが一定値以上ならフィルタ係数を更新
    for(i=0;i<N;i++){ // NLMSアルゴリズムでフィルタ係数更新
        h[i] = h[i] + mu * s[(t-i+MEM_SIZE)%MEM_SIZE] * e/norm;
    }
}
y[t] = x; // 推定信号を出力とする
```



各係数は観測信号から推定値を引いた推定誤差  $e(t) = d(t) + s_p(t) - x(t)$  がゼロになるように更新する

係数更新アルゴリズムの例(NLMSアルゴリズム)

$$h_m(t+1) = h_m(t) + \mu \frac{s(t-m)e(t)}{\sum_{m=1}^{N-1} s^2(t-m)}$$

ステップ・サイズ (更新の強さを決めるパラメータ)  
( $m=0, 1, \dots, N-1$ )

図2 適応フィルタとしてFIRフィルタを用いる

適応フィルタ (Adaptive Filter；出力と所望の信号を近づけるように自動的に設計変更を行えるフィルタ)で未知システム (入力と出力だけが観測可能で自身が全く分からないシステム)を推定することをシステム同定 (System Identification) と呼びます。

### ● 仕組み

ある未知システムに信号を入力し、所望の信号が出力されたとします (図1)。例えば、カラオケなら、マイクに入る音声が入力で、スピーカから出るエコー付きの音声が入力の信号です。未知システムの中身は分かりません。

しかし、適応フィルタで所望の信号と同じ信号を作ることができれば、結果として未知システムと同じ構造を得たことになります。これがシステム同定です。

適応フィルタの構成を図2に示します。フィルタ係数を推定誤差が0に近づくように更新します。係数更新のための適応アルゴリズムとしては、NLMS