ご購入はこちら

第2章 はじめの一歩！フィルタがわかれば何でもできる

基本フィルタ

川村 新

2-1：素通しフィルタ

収録フォルダ：00_Through

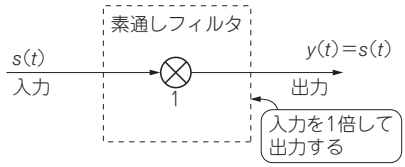

図1　素通しシステムの仕組み

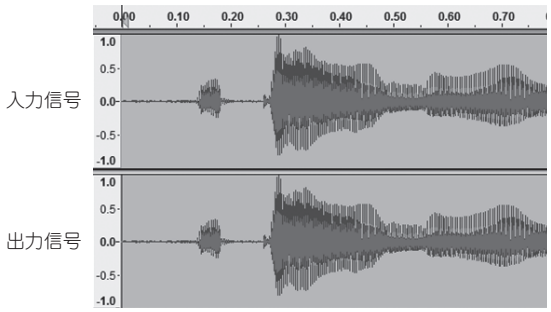

図2　入力信号がそのまま出力されている

最も簡単な音声処理の例として，あるwavファイルの信号を1倍して（つまりそのまま）出力します．出力wavファイルと入力wavファイルは一致します．

また，マイクから入力を得て，そのままスピーカから出力する素通しのリアルタイム・プログラムを作成します．

● 仕組み

入力をそのまま出力する素通しシステムを**図1**に示します．破線で囲まれた部分が素通しフィルタです．

中央の×マークは乗算器であり，矢印で入力される信号を定数倍して出力する役割を果たします．

● プログラム

素通しフィルタのプログラムを**リスト1**に示します．音声の入出力ファイルの取り扱いに関する説明は省きます．

変数宣言部において，時刻はtで管理し，0からスタートします．メイン・ループ内では，各時刻の処理が終了すると1ずつ増加させ，MEM_SIZEまで増加すると，再び0に戻るようにしています．入力データと出力データを格納する配列はMEM_SIZE＋1まで用意します．

メイン・ループ内では，入力として得たs[t]をそのまま出力y[t]に代入しています．これが各時刻における処理です．

リアルタイム・プログラム（RT_Through.c）でも入力データと出力データを同じ変数で定義していますので，宣言する変数と，メイン・ループ内の処理は全く同じです．

リスト1　素通しフィルタのプログラム DD_Through.c（抜粋）

(a) 信号処理用変数の宣言部のプログラム

```
int       t       = 0;          // 時刻の変数
long int  t_out   = 0;          // 終了時刻計測用の変数
int       add_len= 0;           // 出力信号を延長するサンプル数
short     input, output;        // 読み込み変数と書き出し変数
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
```

(b) メイン・ループ内 Signal Processing 部のプログラム

```
y[t]=s[t];                // 出力=入力
```

● 実行結果

実行結果を**図2**に示します．入力波形と出力波形が同一であることが確認できます．

リアルタイム・プログラムでは，マイク入力がそのまま出力されます．ハウリングを避けるために，必ずイヤホンを使用してください．