

フィルタのいろいろな特性と その分類

ここでは、アナログでもデジタルでも共通な信号処理システムとしての“フィルタ”とはどういうものであるのかというイメージの組み立てから始めて、その性質を明らかにする。簡単な定義からスタートして、直感的な理解ができるように説明のステップを進めていく。 (筆者)

1 周波数選択性による分類

一般に、『入力信号の中から必要な周波数成分だけを通過させ、不要な周波数成分を除去して出力する』という働き、すなわち**周波数選択性**を有するシステムを**フィルタ**(filter)と言います(図1)。たとえば、家庭におけるゴミの分別処理といったところでしょうか。フィルタにはさまざまな定義のしかたがあります。そこでまず、**周波数選択性**に基づく分類の説明から始めましょう。

フィルタのもつ周波数選択性

本論に入る前に、フィルタのもつ周波数選択性の雰囲気をつかみましょう。まず、Scilabで記述された[プログラム例1]をエディタで注意深く作成します。ただし、『//』で始まる行()は注釈コメントなので、入力しなくてもかまいません。なお、使いなれた日本語テキスト・エディタ(たとえば、メモ帳)を利用して入力するときは、バック・スラッシュ(\)は円マーク(¥)となるので注意が必要です。

また、[プログラム例]は本誌付属のCD-ROMにはあえて収録していません。各自で入力してもらうことを前提にしています。なぜなら、実際に入力してみることによって、Scilabプログラムの記述方法が学べると思うからです。なお、Scilabの文

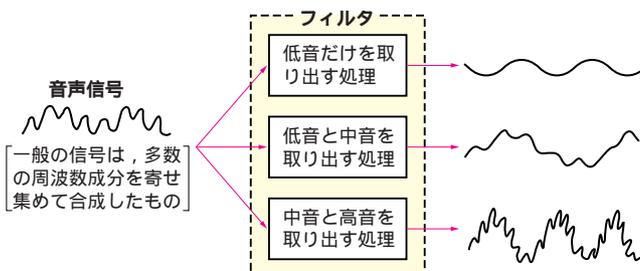


図1 フィルタの働き

法の詳細はヘルプ・ファイルを参考にしてください。

プログラム例1(周波数選択性)

```
//frequency characteristic ***** .....

global sfrq,tmax; .....
T = 0.002; .....
sfrq = 1/T; .....
tmax = 0.5/T; .....
t = 0:T:(0.5-T); .....

while(1) .....
printf('\n'); .....
frq = input('input frequency(
                                0-50 [Hz]) '); ...
if frq == [] then break end; .....

amp = 1/sqrt(1+(frq/20)^2); .....
pha = arctan(1/(1+%i*(frq/20))); .....
xin = cos(2*pi*frq*t); .....
yout = amp*cos(2*pi*frq*t+pha); .....

aoscg(xin,yout,0); .....
disp(max(abs(yout)), 'ymax = '); .....
end .....
// ***** .....
```

ところで、Scilabプログラムの記述内容を大ざっぱにでもつかんでおくと、Scilabプログラミングの理解の助けになるので、まずは簡単に説明しておきます。

[プログラム例1の説明]

利用する関数にまたがって変数を共通で利用するために、グローバル変数(sfrq, tmax)を定義する

グラフ表示の時間間隔(変数 T)を 0.002[秒]に設定する
 1秒間の分割数(変数 $sfrq$)を設定する
 グラフ表示する横軸(時間軸)の最大値(変数 $tmax$)を設定する
 分割した時刻(変数 t)を設定する
 ~ while(1)とendではさまれるプログラムを繰り返し実行する
 改行する
 入力信号の周波数(変数 frq)の入力待ちになる
 周波数を入力せず、キーボードから [Enter] キーを押したときに、プログラムの実行が終了する。なお、ほかのプログラムを実行するには、かならず実行中のプログラムを終わらせてからにしなければならない
 フィルタの利得(変数 amp)を計算する。利得は、入力信号が何倍で出力されるかを表すパラメータであり、詳細は「2. 利得特性による分類」で後述する
 フィルタの位相(変数 pha)を計算する。位相は、入力信号がどれくらいずれて出力されるかを表すパラメータであり、詳細は「3. 位相特性による分類」で後述する
 入力信号(変数 xin)として周波数(変数 frq)の cos 波形の信号値を設定する
 利得と位相を考慮して、出力信号(変数 $yout$)を計算する
 入力および出力信号を 2 現象オシロスコープ表示する。
 関数コマンド `aoscg` は、

```
aoscg(信号変数1, 信号変数2, ウィンドウ番号)
```

と記述し、信号変数 1 はピンク色、信号変数 2 は緑色で、アナログ信号を表す連続した曲線として画面表示する。

なお、関数コマンドと称するものは、本特集のために自作した Scilab プログラムであり、これまでの命令と記したもの(Scilab の組み込み関数)とは区別できるように表記してある。

出力信号の最大振幅値(変数 y_{max})を求め、画面上に表示する

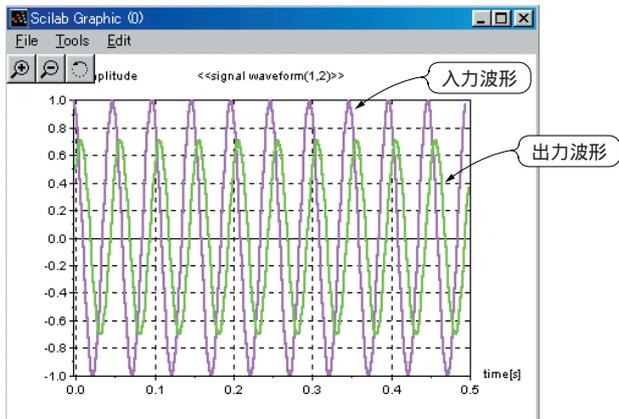


図2 実行例1のグラフ表示例(入力周波数 30Hz)

次に、作成したプログラムに名前(たとえば、ファイル名「prog21.sce」)を付けて、フォルダ `CQfilter` に保存します。なお、保存するファイル名にはかならず拡張子「.sce」を付けることが肝要です。

入力が完了したら、Scilab 画面(第1章の**実行例 32**を参照、p.52 ページ)において、メニュー・アイコンの中から [File] [Exec...] と操作し、ファイル `prog21.sce` を選択したあと、開く(O) ボタンを左クリックして実行します。作成した Scilab プログラムに誤りがなければ、すぐに実行され、

```
実行例 1
->exec('C:\CQfilter\prog21.sce');
      disp('exec done');
input frequency(0-50 [Hz]) -->
```

と画面表示されます。誤りがある場合は、エラー・メッセージが表示されるので、入力ミスや文法上の誤りなどをチェックし、適切に修正し、最終的に誤りがなくなれば、正常に実行されます。

試しに、30 と入力して [Enter] キーを押してみると、30[Hz] の cos 波形(最大振幅が 1)の入力信号に対する出力信号が表示されます(図2)。図2は2現象オシロスコープであり、上段(紫色)が入力波形、下段(緑色)が出力波形に相当します。

[プログラム例1]は、周波数選択性を調べるための実験回路(図3)を Scilab プログラムとして書いたもので、入力信号の最大振幅 1(入力周波数によらず一定値を採る)に対する出力信号の変化がわかります。そこで、周波数を 0 ~ 50[Hz] の範囲で入力し、出力される信号波形の最大振幅値(y_{max})を読み取り、横軸に周波数、縦軸に最大振幅値をとってグラフを描いてみましょう(図4)。

図4からわかるように、周波数が高くなるほど出力が小さくなっています。このように周波数の高低に応じて出力をコント

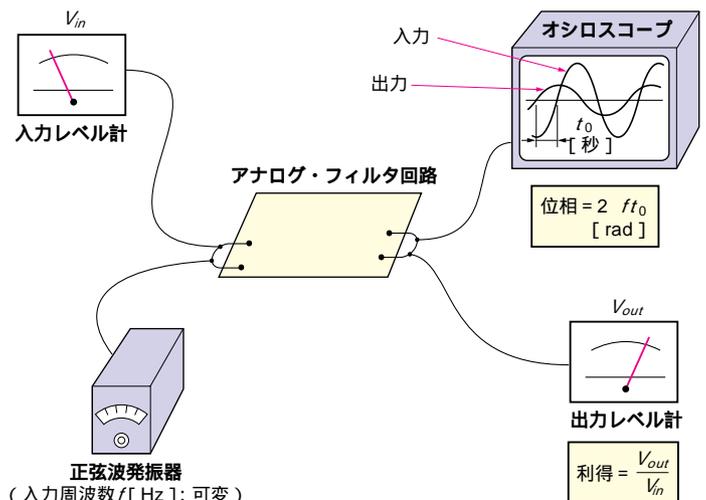


図3 周波数特性の測定