

移動平均 / 共振器 / ノッチ…基本フィルタをアプリを使わずに作ってみる

三上 直樹

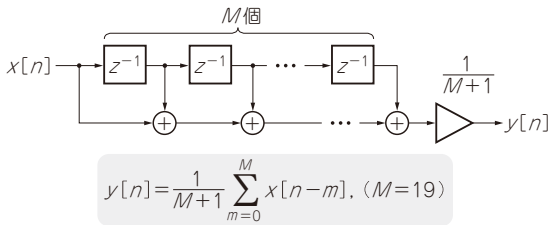


図1 マイコンで作る移動平均のブロック図

この章では、第1部第4章で説明したデジタル・フィルタのうち、

- ・移動平均
- ・共振器
- ・ノッチ・フィルタ

のプログラムを作りマイコン・ボードで動かします。これらのフィルタは、フィルタの係数を求めるための設計アプリケーションを使わずにプログラムを作れます。

本章で説明する共振器のプログラムにはフィルタの係数を計算する部分が含まれています。係数を自分で算出できれば、プログラムの外部から共振周波数をコントロールする機能を付けるような場合に役に立ちます。

マイコンのプログラムでは、フィルタの部分をクリックで実現することにします。クラスで作ると、他の処理と簡単に区別できるので、どこまでの処理が、そのフィルタの処理に本質的な部分かということが分かりやすいと思います。

希望する振幅特性を実現できているかどうかは、FFTアナライザ(第3部第1章のコラム1を参照)⁽¹⁾を使って確かめます。

● マイコンで動かすフィルタは float を使う

STM32F446マイコンを使った場合、浮動小数点数の計算を double 型で行うと処理にかなり時間がかかります。そのため、基本的には float 型で計算するようにプログラムを作る必要があります。

▶ float 型であることを明示する記述方法

プログラムで直接数値を記述する場合、例えば 1.0f というように数値の後ろに float 型を表すサ

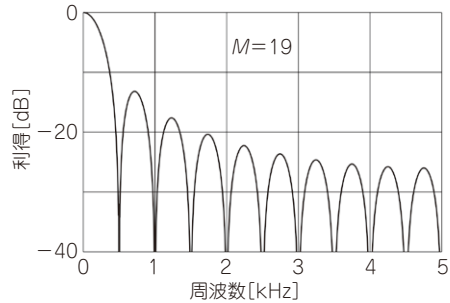


図2 マイコンで作る移動平均フィルタの振幅特性

フィックス f を付けます。このサフィックスを付けずに例えば 1.0 と書いた場合、その数値は double 型だとみなされます。

▶ Keil Studio には float 型の算術演算関数が用意されている

Keil Studio が提供する標準の算術関数ライブラリでは、float 型に対応する関数も提供されています。例えば、cos() 関数は引数と戻り値が double 型です。そのため、これから示すプログラムの中では、cos() の代わりに、引数と戻り値が float 型の cosf() を使います^{注1}。

フィルタ 1：移動平均

マイコン用プログラムに対応する移動平均のブロック図と差分方程式を図1に示します。図2には、M=19 の場合の周波数特性を示します。

● プログラムの解説

移動平均のプロジェクト (DSP_MovingAverage)

注1: Keil Studio Cloud がサポートする C++ の標準の算術関数を計算するためのライブラリでは、関数はオーバーロードされているので、引数に float 型のものを与えると、float 型のバージョンの関数に対応するコードが生成されます。しかし、ここでは float 型に対応する関数を使っていることを強調したいので、あえてプログラムでは、例えば cos() を使わずに cosf() を使うように書いています。