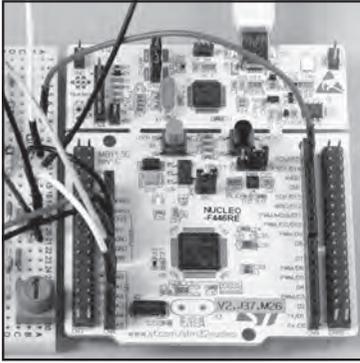


短期連載



インストール不要でどこからでも使える今どき Keil Studio Cloud 入門

モダン Arm 開発環境ではじめる音声信号処理実験

第3回 周波数シフタを作る方法その1… Weaver 変調器

三上 直樹 Naoki Mikami

本短期連載では、Armマイコンのモダン開発環境 Keil Studio Cloud を使いながら音声信号処理実験を行っています。今回は登録方法(コラム)を紹介しつつ、もう少しだけ音声信号処理実験を続けます。

周波数シフタは、入力信号に含まれるすべての周波数成分に対して、それらの周波数を同じだけシフトさせます。ピッチ・シフタとは異なります。

周波数シフタは、アマチュア無線でよく使われる SSB 変調器・復調器でも使われます。今回はマイコンで音声などのオーディオ信号を対象とする周波数シフタを作ります。

周波数シフタを作る方法はいくつかありますが、デジタル信号処理方式で作る場合、代表的なのが、次の2つです。

1. Weaver 変調器を使う方法
2. Hilbert 変換器による位相シフタを使う方法

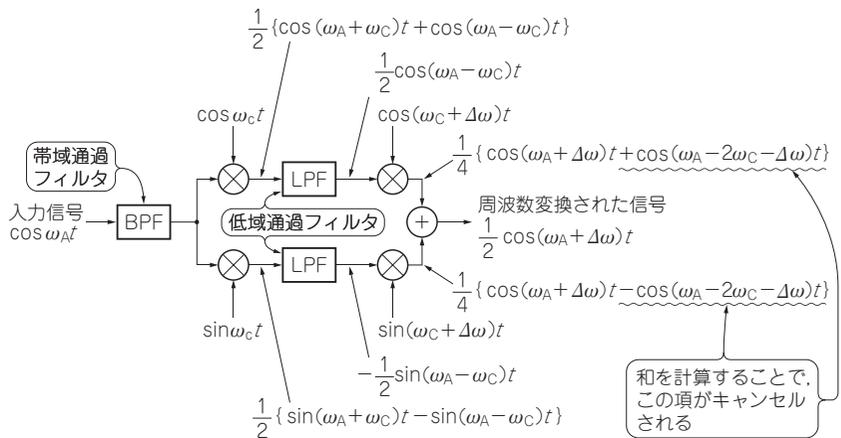
今回は、1の Weaver 変調器を使う方法でプログラムを作り、次回は2の Hilbert 変換器による位相シフタを使う方法でプログラムを作ります。

Weaver 変調器

● 原理

図1に Weaver 変調器のブロック図と各部分の信号

ω_C : 入力信号の帯域の中央の角周波数
 (例) 帯域上端: 19.5kHz, 帯域下端: 0.1kHzとすると,
 $\omega_C/2\pi = (19.5+0.1)/2 = 9.8[\text{kHz}]$
 このとき、低域通過フィルタの遮断周波数は $(19.5-0.1)/2 = 9.7[\text{kHz}]$ とする。
 $\Delta\omega$: シフトする角周波数



を表す式を示します。角周波数 ω_A の正弦波^{注1}を角周波数 $\Delta\omega$ だけ高域側へシフトし、角周波数 $\omega_A + \Delta\omega$ の正弦波にするようすを示しています。また、実際にプログラムで扱う信号は標準化された信号(離散時間信号)ですが、変数を離散的な表現にすると式が見にくくなるので、変数は t としています^{注2}。

なお、実際に入力信号にはいろいろな周波数の成分が含まれていますが、ここでは ω_A という角周波数の成分で代表させて表現しています。 $\omega_A > 0$ としておきます。

図1に示している式は、フィルタが理想的なもの^{注3}で、信号がフィルタを通過する際の時間遅れは0であると仮定した場合です。実際にはフィルタを通過すると時間遅れを生じますが、この図の上の経路と下の経路に入っている2つの低域通過フィルタの時間遅れの特性がそろっていれば問題はありません。ここではデジタル・フィルタを使うので、時間遅れの特性が正

注1 図1では入力に COS 波だが、 SIN 波と COS 波の違いは単に位相が違うだけで本質的には同じものであり、 SIN 波も COS 波も広い意味では正弦波と呼んでもかまわない。

注2 たとえば、 $\text{COS } \omega_A T n$ (T : 標準化間隔, n : 整数)と表現する代わりに、 $\text{COS } \omega_A t$ と表現する。

注3 通過域の利得が1で、阻止域の利得が0倍という特性のフィルタ。

図1 Weaver 変調器による周波数シフタ