



第4章 周波数シフト, フィルタリング, サンプリング・レート変換

マイコンのソフトウェア開発③ 送信変調の前後処理

小川 一朗 OjisanKoubou

第3章では、SDR-3のRF信号処理の中核である「変調処理ブロック」のCプログラムを紹介しました。本章では、A-Dコンバータが出力信号を変調処理ブロックに渡すまでと、変調処理ブロックの出力信号をD-Aコンバータに渡すまでのCプログラムについて解説します。

周波数シフト [第3章 図1の⑥]

● +12 kHz($f_s/4$)を選んだ4つの理由

第3章 図1 周波数変換ブロック⑥では、サンプリング周波数48 kHzのLSB信号のキャリア周波数を0 Hzから+12 kHzシフトしています。

周波数を0 Hzからシフトする理由の1つは、オーディオ・コーデックICがDC成分(0 Hz)を通さないからです。

2つ目の理由は、LSB信号がI/Q信号(複素信号)なので、キャリア周波数±数kHzの帯域が必要だからです。3つ目の理由は、ナイキスト周波数($f_s/2 = 48 \text{ kHz}/2 = 24 \text{ kHz}$)に近すぎると、D-Aコンバータ内でゲインが落ちたり、エイリアシングが出やすくなります。第3章で紹介した第4の方式では、作りにくい急峻なフィルタが必要になったりもします。

SDRの多くが、オーディオ・キャリア周波数として $f_s/4 (= 12 \text{ kHz})$ を選んでいるのは、以上のような理由からでしょう。

▶ 決定的な4つ目の理由

SDR-3特有の決定的な4つ目の理由があります。マイコンで信号処理するとき、 $f_s/4 (12 \text{ kHz})$ だとしても効率がいいのです。

変調処理ブロックでは、0 Hzの信号を処理します(ゼロ周波数方式)。上記のような理由からキャリア周波数を上げる処理が必要です。周波数を変換する演算は、具体的には $\exp(j2\pi ft)$ を掛けることです。 $\exp(j2\pi ft)$ は次式で表されます。

$$\exp(j2\pi ft) = \cos(2\pi ft) + j\sin(2\pi ft)$$

ここで $f = f_s/4$ に設定すると、 $\cos(2\pi ft)$ は+1, 0, -1, 0, $\sin(2\pi ft)$ は0, +1, 0, -1という4値だけをとります。その結果 $\exp(j2\pi ft)$ を信号に掛ける処理は、

- 信号をそのまま出力する(+1)
- 信号を出力しない(0)
- 信号の符号を反転する(-1)

というシンプルな操作で片付きます。三角関数を使った難しい計算は不要です。

昔のマイコンは、乗算器が高く計算に数クロックかかったので、このように符号を変えるプログラムが多かったようですが、最近のマイコンは1クロックで乗算が終わるので、sin関数の値を収めたテーブルからデータを引いて、乗算するほうが結果的に高速です。cosはsinと位相が90°ずれているだけなので、同じsin関数テーブルから計算で得ます。FPGAなどのハードウェアの場合は、符号を変える処理のほうが簡単で、資源を使わずに済みます。

リスト1 +12 kHz周波数シフトのCプログラム(STM32F405マイコン用)

第3章 図1 変調ブロック④の信号処理を行う

```
//
// 周波数変換 +12kHz
//
static void freqconv_tx( float *isig, float *qsig ) {
    static int cnt=0;
    static const float sintbl[5] = { 0, 1, 0, -1, 0 };
    float sin = sintbl[cnt];
    float cos = sintbl[++cnt];
    if( cnt==4 ) cnt=0;
    float i = (*isig)*cos - (*qsig)*sin;
    float q = (*isig)*sin + (*qsig)*cos;
    *isig = i;
    *qsig = q;
}
```

I/Q信号への
ポインタを渡す

$\exp(j\omega t)$ を掛けて
周波数変換を実行

周波数変換した
結果を返す

三角関数テーブル、1つ
余分にデータをもっている
のでcosの値を得るのに
条件判断がいらない

周波数変換処理は、 $\exp(j\omega t)$ を掛けることで実行できる。入力信号、つまり、 $i_{sig} + jq_{sig}$ に $\exp(j\omega t) = \cos(\omega t) + j\sin(\omega t)$ を掛けると、
Real part : $i_{sig}\cos(\omega t) - q_{sig}\sin(\omega t)$
Imag part : $i_{sig}\sin(\omega t) + q_{sig}\cos(\omega t)$
になる。プログラムはポインタになっているためわかりにくいかもしれないが、やっていることは同じ

【 세미나案内 】 スイッチング電源トランス&コイル設計
—— コアの選択から各種トランス&コイルの定数の計算まで

【講師】 戸川 治朗 氏, 9/5(水) 18,000円(税込み) <http://seminar.cqpub.co.jp/>