

第7章

リング・バッファを活用した音声信号処理のいろいろ

オーディオ・エフェクタの製作

本章では、DSKで動くアプリケーション例としてオーディオ・エフェクタを作ります。なかなか面白い効果が得られるのですが、プログラムの中身はこれまで説明してきたリング・バッファを応用した簡単なものです。

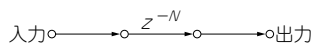
製作するオーディオ・エフェクタは、ディレイ、エコー、リバーブ、ピッチ・シフト、ビブラート、フランジャです。紹介しているサンプル・プログラムは、すべて付属CD-ROMに収録しています。

7-1 ディレイ・ラインの製作

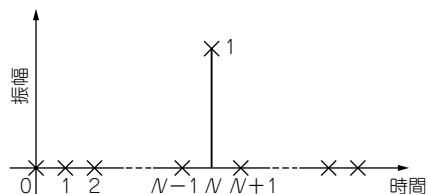
■ ディレイのシグナル・フロー

ディレイ（遅延）については、いまさら説明する必要はないでしょう。一般の電子回路の回路図に相当するシグナル・フロー・グラフを使ってディレイを表現すると、図7-1のようになります。シグナル・フロー・グラフは、ディジタル信号処理の流れを示すのによく使われるものです。

図7-1のシグナル・フロー・グラフの中央の z^{-N} という記号がついている枝は、 N サンプルの遅延、つまりディレイ・ラインを表しています。例えば、100サンプル遅延のときには $N = 100$ ですから z^{-100} と表記します。



〈図7-1〉ディレイのシグナル・フロー・グラフ
(z^{-N} : N サンプル遅延)



〈図7-2〉 N サンプル遅延のディレイ・ライン
のインパルス・レスポンス

■ ディレイの入出力関係を式で表す

式を使ってディレイの入出力の関係を表すと、次のようになります。

$$y[n] = x[n - N]$$

ただし、 x ：入力信号、 y ：出力信号、 N ：ディレイの値

ディレイ・ラインのインパルス・レスポンス、つまりインパルス(単発パルス)が入力されたときの出力は、図7-2のようになります。 N サンプルだけ遅れてインパルスが出力されます。

■ ディレイのサンプル・プログラム

ディレイ・ラインは、ソフトウェアでは、リング・バッファを使って実現できます。DSKのディレイのサンプル・プログラムはすでに第5章で紹介したthrough_intrになります。リング・バッファ長を変更すれば、メモリ(SDRAM)の容量の許す限り、長大な遅延を実現できます。

7-2 エコーの製作

■ エコーのシグナル・フロー・グラフ

ディレイのプログラムに少し手を加えるだけでエコーを実現できます。日本語に直せば、エコーは「山彦」です。エコーをシグナル・フロー・グラフで表すと、図7-3のようになります。この図には遅延を表す z^{-N} の枝の後に、ゲインを表す定数 a のついた枝があります。具体的にゲインが0.7だった場合は、 a の代わりに値0.7を図中に記します。図7-3の処理は、入力した信号に遅延を加えてゲインを調整したものと元の信号を足し合わせているだけです。

■ エコー・システムを数式で表現する

図7-3のエコー・システムの入出力の関係を式で表すと、次のようになります。

$$y[n] = x[n] + a \times x[n - N]$$

ただし、 x ：入力信号、 y ：出力信号、 a ：ゲイン、 N ：遅延の値

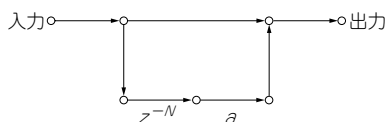
例えば、サンプリング周波数8kHzのときに、エコーの遅延を0.1s(800サンプル分の遅延)、エコー成分の振幅を入力に対して0.7倍として、具体的にパラメータに値を入れると、入出力関係は次のようになります。

$$y[n] = x[n] + 0.7 \times x[n - 800]$$

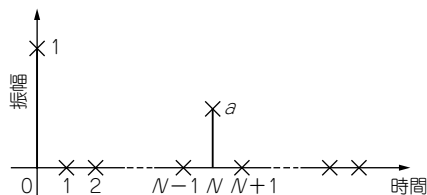
図7-3のエコー・システムのインパルス・レスポンスは図7-4のようになります。入力信号から遅れてエコー成分が出力されていることがよくわかると思います。

■ 複数のエコー成分でリアリティを出す

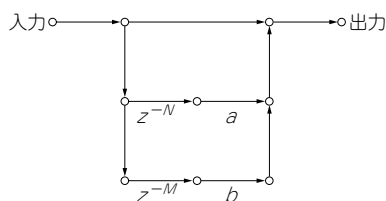
図7-3に示す処理で実際にエコーをかけた音を聞いてみると、いささか自然さが欠けた単調な音色



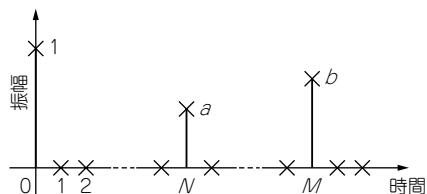
〈図7-3〉エコーのシグナル・フロー・グラフ (z^{-N} : N サンプル遅延, a : ゲイン)



〈図7-4〉エコーにインパルス・レスポンス (単発パルスを入力したときの出力)



〈図7-5〉エコー成分が二つあるエコーのシグナル・フロー・グラフ



〈図7-6〉図7-5のシステムのインパルス・レスポンス

に聞こえます。プログラムを改良してエコー成分を複数付け加えれば、よりリアリティのある音になります。具体的に、エコー成分を二つ加える場合の処理は図5-5のようになります。

● 二つのエコー成分のインパルス応答と数式

図7-5のシステムのインパルス・レスポンス(図7-6)を見ると、入力信号に対してエコー成分が二つ加わっていることがわかります。数式で入出力の関係を表すと、次のようになります。

$$y[n] = x[n] + a \times x[n - N] + b \times x[n - M]$$

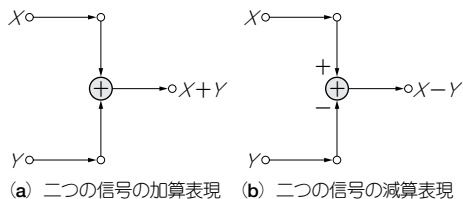
さらに、図7-5を改良して、エコー成分を三つ以上に増やすこともできます。

Column 7-1

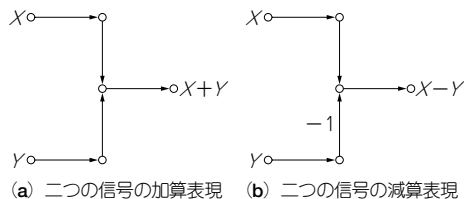
シグナル・フロー・グラフでの加減算記述

一般的なフローチャートなどでは二つの信号の加算(または減算)を図7-Aのように表しますが、シグナル・フローチャートでは図7-Bのように表

現します。図7-3や図7-5のエコーのシグナル・フローでも、図7-Bと同じ形式で入力信号とエコー成分の加算を表現しています。



〈図7-A〉一般のフローチャート



〈図7-B〉シグナル・フローチャート

■ エコー・プログラムecho1について

DSK用のエコー・プログラムはecho1になります。このプログラムは、第3章で取り上げたスルー・プログラム(ADC入力をそのままDAC出力)through1を元に作ってあります。この章のプログラムはすべてモノラル・マイク入力です。出力はステレオですが、エフェクトをかけた同じデータを両方のチャンネルに出力するようになっています。

echo1 の C ソース・プログラム (echo1.c) 中のエコー処理部分だけを抜き出したものをリスト 5-1

〈リスト7-1〉DSK用のエコー・プログラムecho1.c(抜粋)

```
// simple echo (mic in, fs=48kHz)
//
//                                     +-----+
//      in ---->|OUT_GAIN+----> out
//               |                     +-----+
//               +-----+             +-----+
//               |delay+--->|ECHO_GAIN+---+
//               +-----+             +-----+
//               BUF_LEN
//
#include "c6713dsk.h"

#define UINT        unsigned int
#define USHORT     unsigned short

#define BUF_LEN    4800
#define ECHO_GAIN   0.8
#define OUT_GAIN    0.5

#pragma DATA_SECTION(buf, "sdram");
far float buf[BUF_LEN];
unsigned int ix_buf=0;

.....

interrupt void mcbbsp1_rx_intr(void) {
    UINT temp;
    short l_in, r_in, l_out, r_out;
    float buf_out;

    temp=(UINT *)McBSP1_DRR; // read A/D
    l_in=temp>>16;           // l_in : left ch. in
    r_in=temp&0xffff;         // r_in : right ch. in } モノラル・マイク入力なので後の処理では左チャンネル側のデータのみを使用している

    buf_out=buf[ix_buf]; // read ring buffer
    buf[ix_buf]=l_in; // write ring buffer
    ix_buf=(ix_buf+1)%BUF_LEN; // update buffer index } リング・バッファ実現のために剰余演算子(modulo演算子)%を使っている
    l_out=l_in+ECHO_GAIN*buf_out; // add echo
    l_out=OUT_GAIN*l_out; // output gain trim
    r_out=l_out; // right ch = left ch } 左右チャネルに同じエコーを出力

    temp=l_out<<16; // l_out : left ch. out
    temp=temp|(r_out&0xffff); // r_out : right ch. out
    *(UINT *)McBSP1_DXR=temp; // write D/A
}
```

に示します。このプログラムではリング・バッファを使ったディレイを用いています。

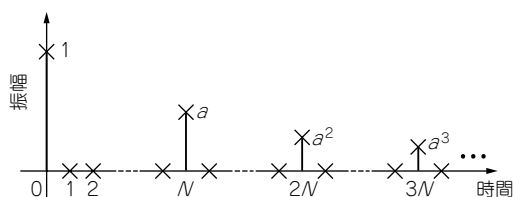
7-3 リバーブ効果の製作

■ エコー成分が無限に続くリバーブ

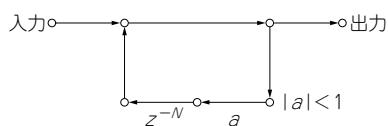
リバーブは、先に説明したエコーのプログラムに少し手を加えるだけで実現できます。エレキ・ギターなどの電子楽器の演奏やカラオケを趣味にしている人は、エコーとリバーブの違いをよく知っていると思いますが、あまりその違いを意識したことのない人も多いかもしれません。図7-7のリバーブのインパルス・レスポンスを図7-4、図7-6のエコーのレスポンスと比較すれば、その違いは明瞭です。有限の数のエコー成分しかないエコーと比較すると、図7-7のリバーブではエコー成分が少しずつレベルを下げながらも無限に続いています。

■ 出力から入力側にフィードバックをかける

このように無限にインパルス・レスポンスが続くリバーブは、図7-8のようなシグナル・フローで実現可能です。図7-3に示したエコーのシグナル・フローと比較すると、エコーではディレイ z^{-N} 成



〈図7-7〉 リバーブにインパルスを入力したときの出力信号



〈図7-8〉 リバーブのシグナル・フロー・グラフ

Column 7-2

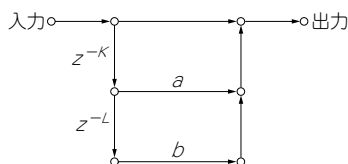
同一のシステムを複数の等価シグナル・フローで表す

同一のインパルス・レスポンスをもつシステムのシグナル・フロー・グラフ表現は、ただ一つとは限りません。同一のシステムを複数の等価のシグナル・フローで表すことも可能です。

例えば、図7-5の二つのエコー成分をもつエコー・システムは、図7-Cのように表すことも可能です。

遅延の値を $N = K$, $M = K + L$ とすれば二つの

システムのインパルス・レスポンスは同一となります。



〈図7-C〉 二つのエコー成分があるエコーのシグナル・フロー・グラフ (図7-5とは異なる表現)