

設計データ一式を11月号DVDに収録予定

## 第1章

過渡波形を忠実再生！無限スプライン補間でFIRの宿命的雑音を原理ゼロ化

# デジタル・フィルタレス FPGA D-Aコンバータの製作

肥後 信嗣 / 小林 芳直

Nobutsugu Higo / Yoshinao Kobayashi

### 理想の過渡応答性能を追求する

#### ● 廉価版 DAC 一色のオーディオに一石投じたい

オーディオ用 D-A コンバータの主流は、低コストで作れる  $\Sigma\Delta$  型の 1 ビット D-A コンバータで、25 年以上の長きにわたってさまざまな音楽再生機に利用されています。

これらの D-A コンバータは、次の 2 つの雑音を減らすために、後ほど詳しく説明するオーバー・サンプリング処理をした後、デジタル・フィルタで馴らしています。

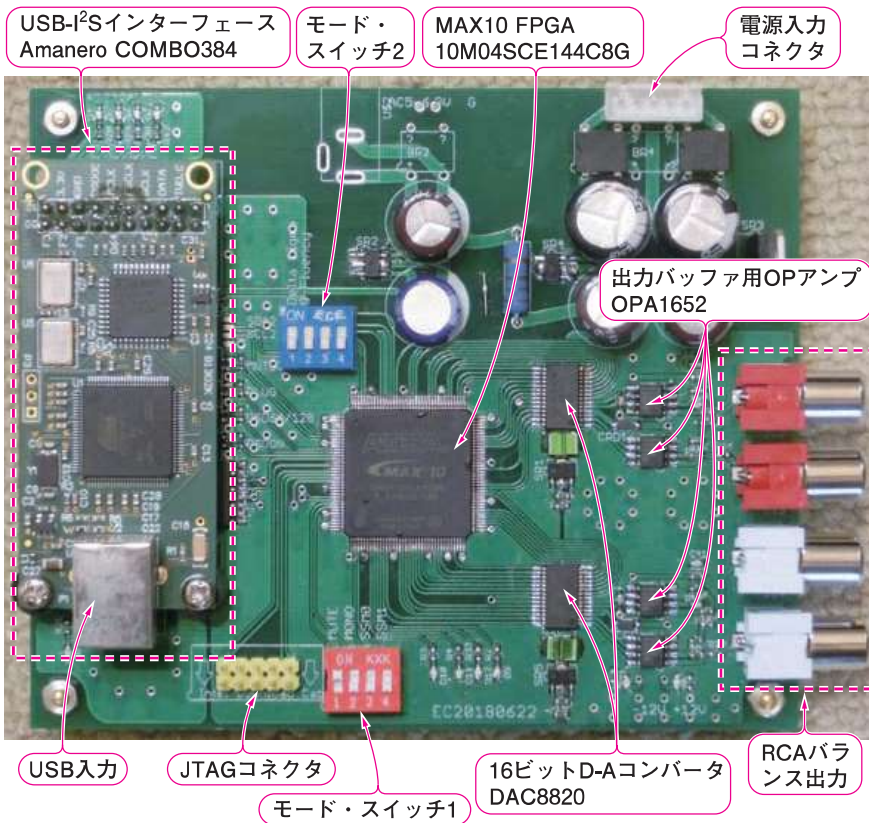
#### ● 量子化雑音 ● 折り返し雑音

このデジタル・フィルタを使った D-A コンバータは、原理的に 1 ビット・タイプでもマルチビット・タイプでも、プリ・エコー、ポスト・エコーと呼ばれる雑音を常に発生させています。

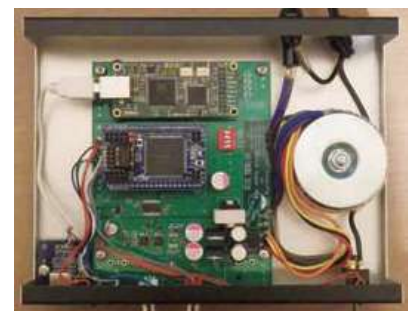
#### ● デジタル・フィルタに付き物の「エコー雑音」をなくしたい

写真 1 に示すのは、開発した エントリ FPGA 「MAX10」で構成したデジタル・フィルタレスのマルチビット D-A コンバータです。

デジタル・フィルタを使う D-A コンバータの宿命である予兆信号(プリ・エコー)のない理想的な過渡



(a) 基板



(b) ケースに収めたところ

写真1 理想的な過渡応答特性をもつデジタル・フィルタレス FPGA D-A コンバータを製作 (キットを開発中(2018年10~11月発売予定, CQ出版社))

【セミナー案内】 実習・LTspiceを活用したローノイズ・アナログ回路設計入門  
—— 微小/微弱信号回路設計者必聴! ノイズ・シミュレーションの基礎と応用, 実測との比較  
【講師】 石井 聡 氏, 9/27(木) 30,000円(税込み) <https://seminar.cqpub.co.jp/>

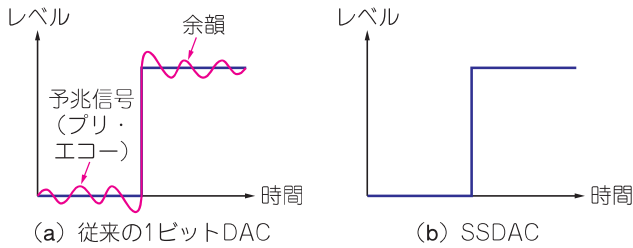


図1 SSDACはほとんどのデジタル・オーディオが出すエコー雑音が原理的に発生しない

現代のデジタル・オーディオはそのほとんどが、FIRフィルタを搭載した1ビットのオーバー・サンプリングD-Aコンバータ。FIRフィルタは宿命的に過渡応答にプリ・エコーとポスト・エコーが発生する。SSDACはこのデジタル・フィルタを使わないので、原理的にプリ・エコーとポスト・エコーが出ない

表1 製作したSSDACの基本スペック

項目	仕様
入力端子	USB, Amanero COMBO384使用
	I <sup>2</sup> S, コネクタPH1より入力
アナログ出力	RCAステレオ
	RCAステレオ反転(差動出力)
電源入力	±12V
	+5V
6つの補間モード	スーパーサンプリング(3次無限スプライン補間)
	2次補間
	1次補間
	0次ホールド
	フル・エンシ
	ΔΣ(下位1ビットだけON/OFF切り替え)

表2 1ビット・オーバー・サンプリングD-AコンバータとSSDACの性能比較

種類	特性	過渡応答	S/N	ひずみ率	高域波形(14.7 kHz以上)	動作原理
SSDAC		◎	◎	◎	△	ニュートン方程式
1ビットDAC		△	○	○	◎	フーリエ変換

応答波形を再生できます(図1). 名付けて「スーパー・サンプリングDAC(以下, SSDAC)」です。

SSDACでは、デジタル・フィルタレスなのでリリングが抑えられており、過渡特性が良好です。リリングは元データには含まれていないわけですから、これをひずみと考えれば、従来のD-Aコンバータは大きなひずみが発生しています。波形の前に発生するプリ・エコーは、デジタル・フィルタ特有の産物です。

## 本器の仕様

### 基本仕様

表1に本器の基本スペックを、表2に従来のデジタル・フィルタを使った市販のD-Aコンバータと今回製作したSSDACの特徴をまとめました。

本器は、USB-I<sup>2</sup>Sインターフェース・モジュールのCOMBO384(Amanero製)を搭載したUSB入力のD-Aコンバータで、I<sup>2</sup>Sフォーマットのオーディオ・データも入力できます(PH1端子から)。

ラズベリー・パイなどと組み合わせることもできます。シリアル・オーディオ・インターフェースS/PDIFやデジタル・ビデオ・インターフェースHDMIを接続するためには、I<sup>2</sup>Sフォーマットに変換するモジュールが別途必要です。アナログ出力端子はRCAタイプで、LチャンネルとRチャンネルの極性を反転した-Lチャンネルと-Rチャンネルも用意したので差動出力も可能です。通常はLチャンネルとRチャンネルをアンプに接続して使用します。

### ■ ブロック図

#### ● 従来のD-Aコンバータ

##### ▶ ΔΣ方式(1ビット)D-Aコンバータ

図3(a)に示すのは、現在主流のΔΣ方式1ビットD-Aコンバータのブロック図です。オーバー・サンプリングした後にΔΣ変調を行ない、1ビットD-Aコンバータでアナログ信号に変換します。

##### ▶ マルチビット方式D-Aコンバータ

図3(b)に示すのは、マルチビット方式24ビットD-Aコンバータのブロック図です。

現在入手できるオーディオ用のマルチビットD-A

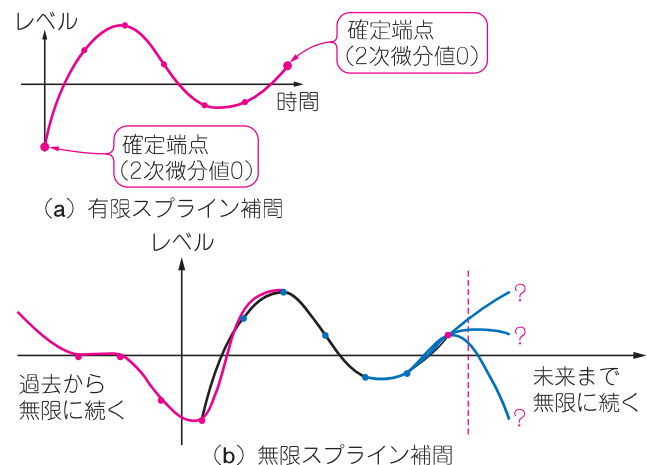


図2 SSDACが実現できたのは無限スプライン関数を解くことができたから

有限のスプラインは両端データの2次微分値を0とすれば解けるが、始点と終点のデータを取り込んでからでないで補間と再生を始められないのでは使いものにならない。遠い過去やずっと先の未来から影響を受ける特定の点と点の間のなめらかな関数を割り出すことは不可能だが、D-A変換の量子化雑音以下の影響を考慮する必要はないので有限計算は可能

コンバータは、PCM1704(テキサス・インスツルメンツ)だけです。市場価格は約5,000円と高価です。設計も古く、新規設計には不向きです。入力インターフェース、オーバー・サンプリング、デジタル・フィルタリングは、外付けのICで処理します。

## ● SSDAC

図3(c)に、SSDAC基板のブロック図を示します。

I<sup>2</sup>S信号を直接FPGA(MAX10)で受け、スプライン補間を行なったあと16ビットのマルチビットD-AコンバータDAC8820に送ってアナログ信号に変換します。16ビット・タイプを選んだ理由は、入手できるマルチビット・タイプの分解能の上限が16ビットだからです。前出のPCM1704にも対応する開発検討を進めています。

SSDACは、デジタル・データの補間を無限スプラインで行うことで、量子化雑音を理論上、4096分の1(=1/64<sup>2</sup>)に抑圧します。D-Aコンバータの量子化ノイズはオーバー・サンプリング比の2乗に逆比例します。これは正弦波における量子化雑音電力 $N_q$ が、次式で表されることに由来します。

$$N_q = \frac{E_q^2}{12}$$

ただし、 $E_q$ : 量子化デジタル値間の1振幅値

従来の標準的な8倍オーバー・サンプリングのD-Aコンバータの量子化ノイズは、オーバー・サンプリングなしのときの、1/64倍(=1/8<sup>2</sup>)にしか減少しません。これをアナログ・ロー・パス・フィルタで減衰させています。

## 新しい補間アルゴリズム

### ● 無限スプライン関数を解く

本器の開発に成功したことの背景には、無限スプ

ライン関数を解けたことと、高性能なFPGAが安価に入手できるようになったことがあります(図3)。

これまで、無限スプライン関数は解くことができないと言われてきました。しかし解析的な方法を使うことで、ついに解くことに成功しました(詳しくはAppendix)。この補間技術は、心音/呼吸音脳波など、波形による医療診断にも応用できます。

スプライン曲線とは、通りたい点(制御点)をすべて通る、なめらかに連続した曲線です。弾性をもつ薄くしなやかな長い板バネを制御点に沿わせていくイメージです。点と点をつなぐ曲線を描くとき、自在定規や雲形定規を使います。また、パソコンのCADや描画ソフトウェアにもスプライン曲線の機能がついています。

図2(a)に示すように、有限スプライン関数は、両端の点の2次微分値をゼロにすれば解くことができますが、この方法は、再生前に楽曲データのすべてをいったん大容量メモリに蓄え、補間計算を終えておく必要があります。計算が終了するまで音楽再生を始めることができません。

リアルタイムに補間計算を行うためには、無限スプライン関数を解く必要があります。

ある区間のスプライン関数を確定するためには、無限の過去から無限の未来までのデータの影響を加味する必要があります。これは、曲のある部分だけを聴くことで、その未来にある音が予測できるか?ということですが、これは無理そうです。図2(b)は、中央の区間のスプライン曲線が、過去と未来のデータから影響を受けていることを示しています。

スプライン関数で補間するには最低3点のサンプル点が必要なため、サンプリング周波数44.1 kHzの信号は、14.7 kHz(=44.1 kHz/3)が正確に波形が再現できる上限周波数です。本器は、96 kHzサンプリングの音源データを入力でき、32 kHz(=96 kHz/3)までの

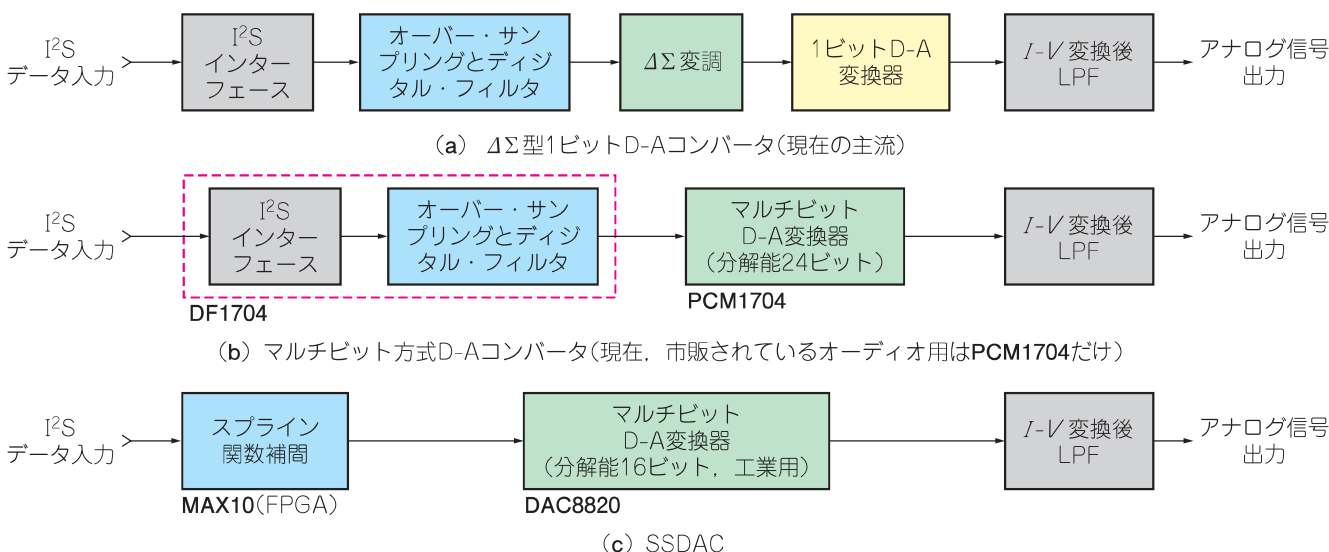


図3 従来のD-AコンバータとSSDACの構成

【セミナー案内】 実習・GNU Radioで始めるSDR入門[教材基板付き]

—— スペアナ、FMラジオ、TVの自作とキーレスエントリの解析で学ぶ

【講師】 猿渡 俊介 氏, 9/29(土) 26,000円(税込み) <https://seminar.cqpub.co.jp/>



波形が正確に再現できます。

### ● 過去と未来のデータの影響

過去と未来のデータは、スプライン関数を計算する区間から遠ざかるほど影響力が低下します。影響力の低下する割合は、1サンプル遠ざかるごとに、 $\alpha$ 、 $\alpha^2$ 、 $\alpha^3$  ( $\alpha < 1$ ) のように等比級数になります。もし影響力がデータのビット分解能よりも小さければ、その先のデータは無視できます。

図4は、補完したい領域(ⓐ-ⓑ間)への前後データの影響力を示しています。

スプライン演算区間ⓐ-ⓑ(本器は63点で補間)を中心に、過去と未来にそれぞれサンプリング・データ点

ⓐ、ⓐ'、ⓑ、ⓑ'があります。図4(a)を見てください。過去と未来のデータが、ⓐとⓑの場合とⓐ'とⓑ'の場合では、求まるスプライン関数に $\Delta S$ の差が出ます。 $\Delta S$ は、前後のサンプリング・データがスプライン関数に影響する度合いを表していて、データが遠ざかるほど小さくなります。この $\Delta S$ が、使用するD-Aコンバータの分解能以下になったとき、それよりも離れたデータは考慮しても意味がなく無視することができます。

### ● FPGAの高い演算能力を利用して24ビット、96 kHzデータを64倍補間

本器に入力できるオーディオ・データは、24ビット、

## D-Aコンバータの分解能を上げる信号処理技術「オーバー・サンプリング」

### ● シンプルな回路で高速動作させて分解能を上げる「オーバー・サンプリング技術」

D-Aコンバータの目的は、デジタル・データをアナログ値に変換して元の波形を再現することですが、サンプル点のデジタル・データをアナログ・データに変換するだけでは十分ではありません。サンプル点とサンプル点の間を適切な補間値で埋めてからD-A変換することが広く行われています。これを**オーバー・サンプリング**といいます。

オーバー・サンプルする倍数のことを**オーバー・サンプル比**といい**OVR**と略されます。**OVRの2乗でD-Aコンバータの分解能が上がる**という理論があり、ビット数の少ないD-Aコンバータを高速動作させることで高精度のD-A変換ができます。

1ビットのD-Aコンバータならサンプリング周波数44.1 kHzに対して動作周波数はその256倍の11.2896 MHz。つまり1つのサンプル点に対して256個の01のビット列なのでビット密度からいって8ビットの精度しかないだろうというのが古典的なパルス密度変調の考え方です。

これに対して、OVRの2乗の精度を加算して16ビットを名乗るのが、現代風のD-Aコンバータです。オーバー・サンプリングD-Aコンバータは、簡単なハードウェアを高速動作させて精度を出すという、現在のIC技術にマッチした技術です。

### ● オーバー・サンプリングの補間には計算パワーが要る

オーバー・サンプリングの適切な補間値を得るとするのはかなりの難題です。

主流は、1つのサンプル点についてsinc関数(ジंकと読む)という次のような関数を発生させることです。

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$$
$$\text{sinc}(x) = 1 \quad \text{when } x = 0,$$
$$\text{else } 0 \quad \text{when } x = \pm n$$

sinc関数は、 $x = 0$ のサンプル点だけ1になり、ほかのサンプル点では0になります。つまり他のサンプル点に影響を与えません。これは**1つのサンプル点の特殊なインパルス応答**ということです。

sinc関数を各サンプル点で発生させて、それをすべて加算すれば、単なる加算だけでサンプル点とサンプル点の間をなめらかに補間できます。

理論は美しいのですが、現実の計算はたいへんです。sinc関数の収束が悪いので、補間値の計算には遠くの多くのサンプル点の計算が必要です。回路規模が大きくなり、D-Aコンバータの応答を遅らせます。映画のように画像と音声がある場合は、音声だけ遅れるのはよくありません。

### ● アップ・サンプリングとオーバー・サンプリングの違い

デジタル信号処理において、アップ・サンプリングとはデータ補間を行なうことによってサンプリング・レートを増す処理のことを指します。たとえばオーディオ・データのサンプリング周波数を44.1 kHzから96 kHzに変換することも、D-Aコンバータにおいていわゆるオーバー・サンプリングを行なうことも、どちらもアップ・サンプリングです。しかし、オーディオの業界では、44.1 kHz→96 kHzのようにデータのフォーマットをコンバートすることを**アップ・サンプリング**、D-Aコンバータにおいてデータの補間を行なうことを**オーバー・サンプリング**という場合が多いです。

(小林 芳直)

【セミナー案内】 波形で実演! ワイヤレス通信におけるデジタル変復調の基礎 [講師による実験実演付き]

—— 基本的な無線データ伝送からOFDMまで、SPICEシミュレータで波形を確認

【講師】 石井 聡 氏, 9/30(日) 19,000円(税込み) <https://seminar.cqpub.co.jp/>



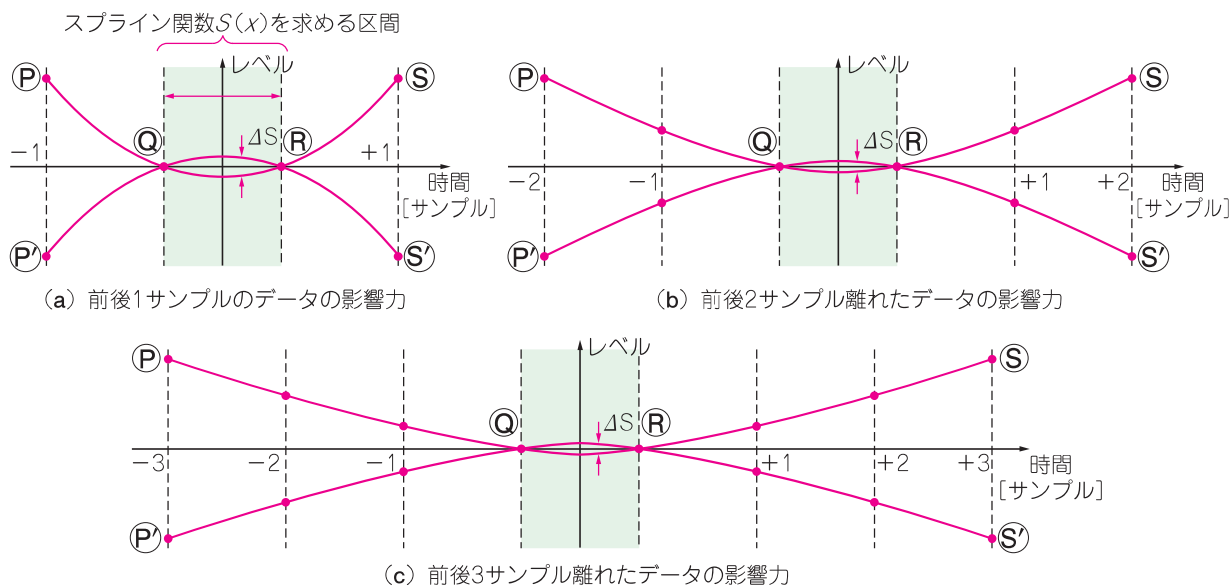


図4 補完したい領域へ前(過去)/後(未来)データの影響力は遠ざかるほど小さくなる

96 kHz サンプリングが上限で、これを無限スプライン関数によって、**64倍にデータ補間計算**します。このときD-Aコンバータには6.144 MHz (= 96 kHz × 64)の動作周波数が求められます。採用したDAC8820の最高動作周波数は10 MHzで、入手できた最高速デバイスです。したがって**現状のSSDAC基板の性能がDAC8820では上限**と考えています。

サンプリング周波数が48 kHzの音源データであれば、DAC8820では128倍まで対応できる可能性があります。FPGAの性能だけを考えるなら、本器の動作クロックは最大41.34 MHzなので、**最大430倍(= 41.34 MHz/96 kHz)まで、つまり256倍まで余裕で補間できる計算**です。

\*

本SSDAC基板は、USBインターフェースを備えており、パソコンやボード・コンピュータと組み合わせ利用できます。頒布キットを開発中で10月~11月発売を目指しています。

## デジタル・フィルタは過渡的に変化する信号を大きくひずませる

● これまでのオーディオ用D-Aコンバータがデジタル・フィルタを使ってきた理由

量子化雑音や折り返し雑音が発生するのは、D-A変換の宿命です。

量子化雑音とは、簡単にいうとD-A変換後の波形に現れるサンプリングの階段状のギザギザ波形です。このギザギザが全帯域にわたる雑音となるため、どうやってこれを取り除くかというのがオーディオCDが発表された当初からの課題でした。

データとデータの中にサンプル点を挿入した後、デジタル・フィルタで馴らすと、この量子化雑音を広帯域に分散させることができます。可聴帯域を超えた部分に分散した信号は、D-Aコンバータの後段に接続するアナログ・ロー・パス・フィルタで減衰させます。このオーバー・サンプリング用のデジタル・フィルタの作り方によって、再生されるアナログ信号の波形が大きく変わります。

● デジタル・フィルタは原理的に元信号に雑音を加える

図3(a)に示すように、デジタル・フィルタに採用されるFIRフィルタを使うと、必ず**プリ・エコーとポスト・エコー**という過渡現象が発生します(コラム2参照)。

プリ・エコーとポスト・エコーは、元の信号にはなかったものですから、**デジタル・フィルタが過渡的に変化する信号に付加したひずみ**です。静寂の中でシンバルが鳴る場面を想定すると、シンバルが鳴る前に何か信号が再生されるというわけです。

ポスト・エコーも問題ですが、必要な信号が出た後のことで、出た信号にマスクされるので、プリ・エコーほど深刻ではなさそうです。

**プリ・エコーとポスト・エコーはナイキスト周波数で発生します。**サンプリング周波数が44.1 kHzのCDの場合は、22.05 kHzです。

この周波数(22.05 kHz)は、単体の信号としては聞き取ることはできません。しかし、楽器音や音声のように連続的ではなく過渡的に変化する複数の信号が同時に入力されると、可聴帯域の信号や別のエコーと干渉して、混変調(うなり)を起こします。

【セミナー案内】電子回路シミュレータLTspice 実習セミナー[トランジスタ技術8月号 特集 連動企画] —— LTspiceを使用して電子回路を体験学習  
【講師】小川 敦 氏, 10/4(木) 22,000円(税込み) <https://seminar.cqpub.co.jp/>

## デジタル・フィルタを 使わずにすむ補間技術を考案

### ● 従来のオーバー・サンプリング・デジタル・フィルタ

図5(a)に示すのは、D-A変換された後のデジタル・データです。

a, b, c, dの連続する4つのデータが出力されており、1つデータが出力されてから次のデータが出力されるまで値がホールドされています。これを0次ホールドといいます。

図5(b)に示すのは、図5(a)の各点の間に、0を3つ加えたところです。サンプル点が増えるので4倍オーバー・サンプリングといいます。

元のサンプル点と追加した補間点は、デジタル・フィルタによってなめらかな曲線になります(図5(c))。ギザギザは細分化され、量子化雑音が広帯域に分散されて、必要なオーディオ帯域の雑音レベルが低下します。さらに補間データを増すと、8倍、16倍、32倍…という具合に、より雑音が広帯域に広がります。

### ● 無限スプライン関数で補間する

図6は、SSDACの補間方法を示しています。図5(b)のように0を加えるのではなく、a, b, c, dの点と点の間をスプライン曲線でつないで補間します。

製作したSSDACは、a~d各点の値からスプライン関数で算出した63個の値を追加しています。つまり64倍サンプリングです。

このデータをそのまま16ビットのマルチビットD-Aコンバータに送り、アナログ信号を再生します。デジタル・フィルタを使わないので、プリ・エコーやポスト・エコーが発生することは原理的にありません。

## キー・パーツ

### ● USB-I<sup>2</sup>Sインターフェース・モジュール

USBから入力された音楽信号は、USB-I<sup>2</sup>Sインターフェース Amanero Combo384でI<sup>2</sup>Sに変換されます。I<sup>2</sup>S信号は、外来雑音の侵入を防ぐためデジタル・アイソレータを介してMAX10に入力されます。I<sup>2</sup>S信号以外のI<sup>2</sup>C信号やMUTE, PLUGなどの各信号も、必要に応じてデジタル・アイソレータ(図7)やフォトカプラでアイソレートしています。

### ● FPGA

ロジック・エレメント数4032個のMAX10 FPGA(10M04SCE144C8G)を使用しました。使用したエレメントの数は3090個/4032個(使用率77%)です。動作クロックは42.75 MHzです。

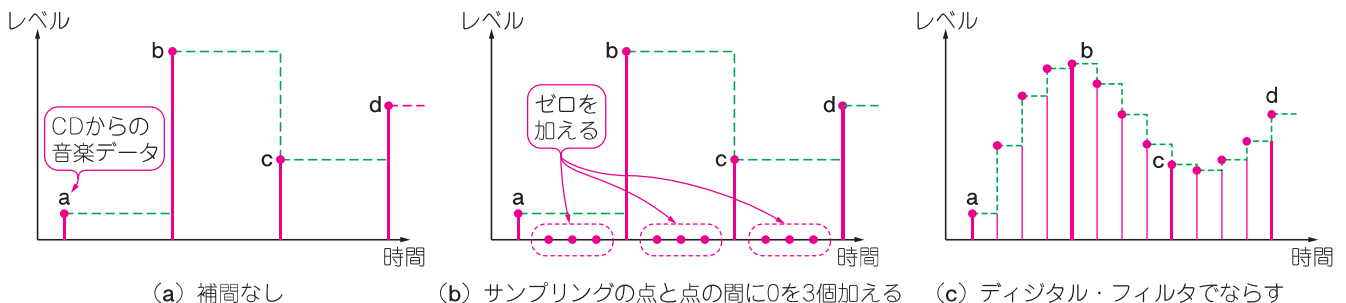


図5 従来のD-Aコンバータのオーバー・サンプリング処理の流れ

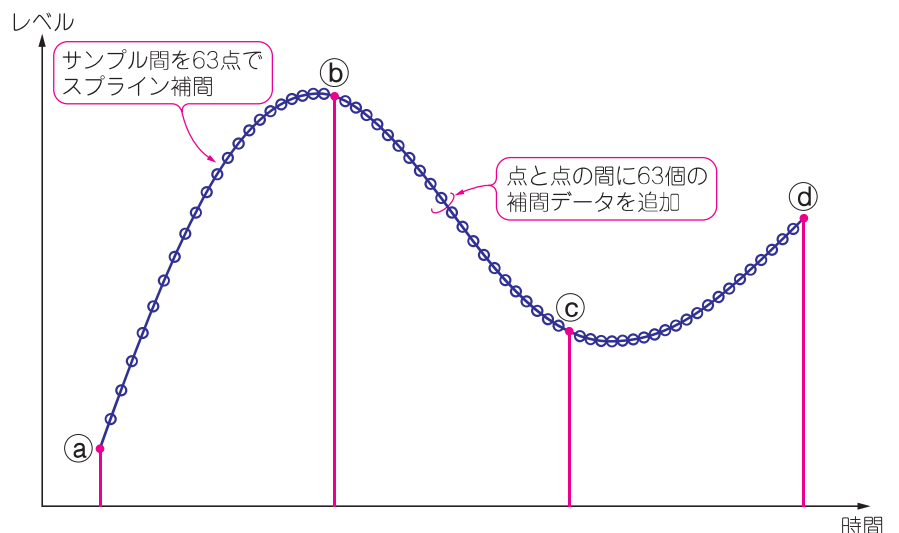


図6 SSDACはスプライン関数でCDの再生データの点と点を64倍補完する

## ● デジタル・フィルタレスの16マルチビットD-Aコンバータ

MAX10に入力されたI<sup>2</sup>Sデータに、スプライン補間演算処理を施し、16ビットD-AコンバータDAC8820(テキサス・インスツルメンツ)に送ります。図8にDAC8820の内部ブロック図を示します。

DAC8820は、16ビットのデジタル・データと±10Vのアナログ・データを積算してくれます。D-Aコンバータのリニアリティ誤差はそのままスプリアス放射になるので、最後の1ビットまで厳しく正確に作られています。D-Aコンバータは16ビットなので、残りの8ビットはΔΣまたは四捨五入で出しています。

## ● I-V変換アンプとアナログ・フィルタ

図9に示すように、DAC8820が出力する電流をI-V変換アンプとロー・パス・フィルタに加えます。OPアンプは低雑音低ひずみのFET入力タイプ(OPA1652)です。

I-V変換出力の信号振幅は、DAC8820のR<sub>OFS</sub>とR<sub>1</sub>端子に加えるV<sub>ref</sub>で決まります。V<sub>ref</sub>は2.5Vで、シャント・レギュレータ(TA76431F)で生成しました。

ロー・パス・フィルタは2次多重帰還型ベッセルで、

カットオフ周波数は約276 kHzです。

最大出力振幅は2.5V<sub>0-peak</sub>です。LチャンネルとRチャンネルの信号電流がグラウンドに流れて、互いが干渉しないバランス出力にも対応しました。

## ● 電源

電源は、出力OPアンプ用のアナログ回路用の±12Vとデジタル回路用の3.3Vの2系統です。別々の入力がありますが、3.3Vは+12Vからも生成できるようにしました。+12Vから3.3Vを電源レギュレータだけで生成するには損失が大きいため、R<sub>0</sub>(68Ω)

DAC8820ICDB(TI, 16ビット, マルチビット)

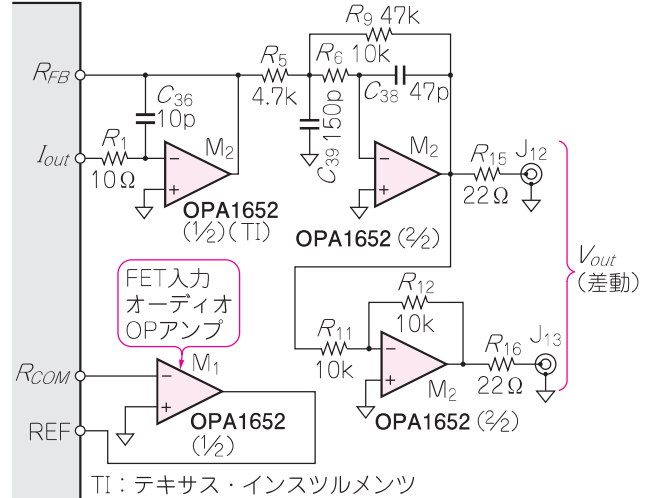
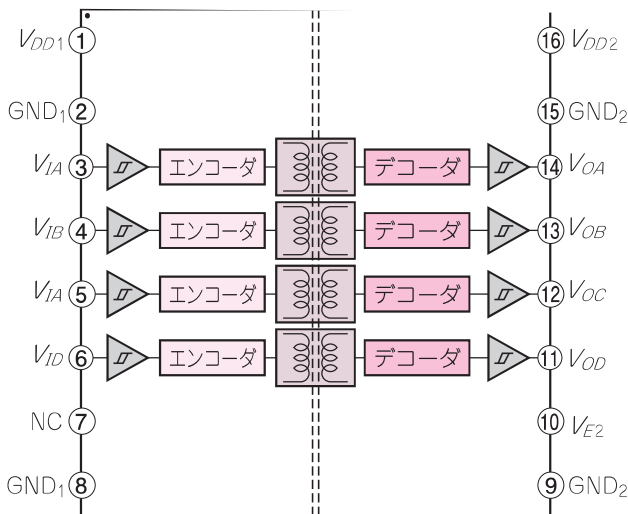
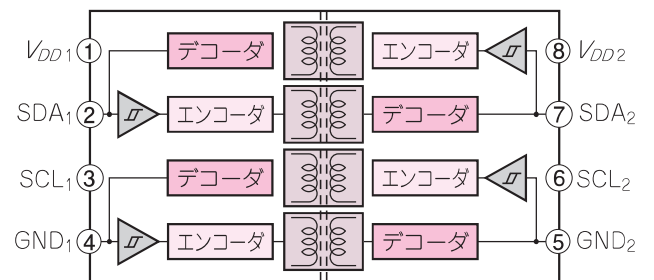


図9 SSDAC基板のD-Aコンバータの出力回路(I-V変換アンプとアナログ・フィルタ)



(a) ADuM1400の機能ブロック図



(b) I<sup>2</sup>Cアイソレータ

図7 SSDACはI<sup>2</sup>SやI<sup>2</sup>Cなどのシリアル・インターフェースやミュートなどの制御信号をアイソレータで絶縁している

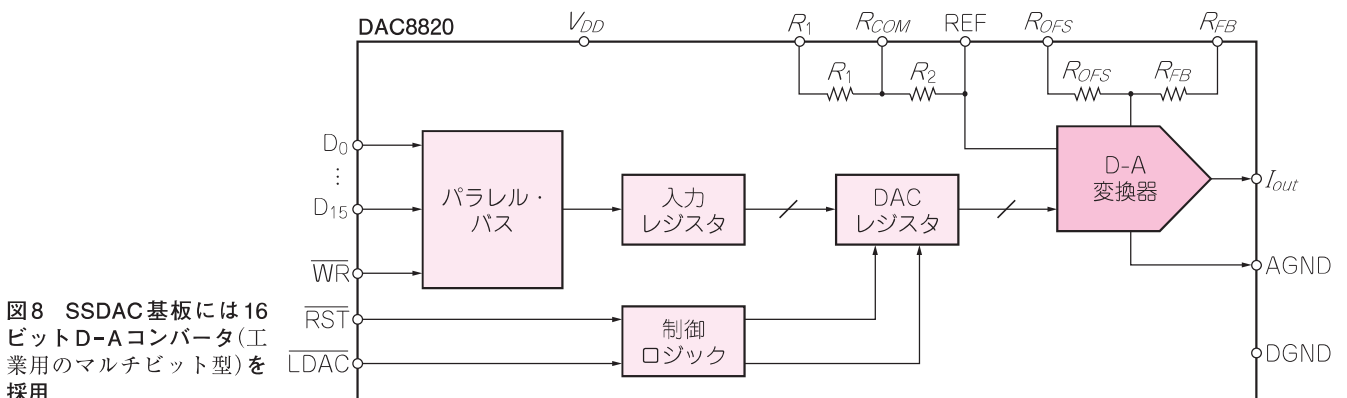


図8 SSDAC基板には16ビットD-Aコンバータ(工業用のマルチビット型)を採用

【セミナー案内】各種二次電池充電回路入門[講師による実演付き]  
 ——鉛蓄電池やニッケル水素、リチウム・イオン二次電池の充電回路の基礎と設計(カリキュラム・リニューアル)  
 【講師】梅前 尚氏, 9/12(水) 19,000円(税込み) <https://seminar.cqpub.co.jp/>



を挿入して損失を抑えています。3.3Vの消費電流は約85mAです。

● オプション

入力信号をUSB入力とせずに、直接I<sup>2</sup>S信号を入力したい場合は、コネクタPH1からMCLK, BCLK, LRCLK, DATAを入力できるようにしました。

コネクタPH2は信号出力用で、液晶表示用などの信号を出力するために設けました。コネクタAは

FPGA書き込み用です。USB Blasterや「完全版FPGA電子工作オールインワン・キット(CQ出版)」に付属しているMAX10-JB基板を接続し、パソコンからUSBを介してFPGAの書き込みを行ないます。

● 動作モード切り換え

基板上的DIPスイッチで次の設定が可能です。

- 補間モード(補間なし/1次直線補間/2次補間/3次スプライン補間)

FIRフィルタが原音にない雑音を付け足すメカニズム

● FIRフィルタの動作原理

FIRフィルタは、どうしてプリ・エコーやポスト・エコーを出すのでしょうか？

図Aに示すのはFIR型のLPFです。Xから入ってくるデジタル・オーディオ信号は遅延素子(D)を経るごとに、1クロックずつ昔のデータになります。a<sub>0</sub>~a<sub>n</sub>はその各データへの重み付けです。出力Yは、過去n個分のデータに重み付けを行なった値を加算した結果です。

パソコンを使って一定間隔で計測データを取り込んで、データを1つ取り込むごとに最新の10個のデータの平均を算出したいとします。もちろんすべてのデータを記録しておいて、あとで平均することはできますが、これをリアルタイムで行なうとしたらどうしたらいいのでしょうか？取り込んだデータをその場で1/10倍して、10個分加算すればいいですね。これがごくおおざっぱなFIRフィルタの原理です。

● 標準的なFIRフィルタ(対称係数型)の周波数応答と波形

図Bに示すように、このFIRフィルタの周波数特性(フーリエ級数展開)は、図中に示す式のようにsinc関数によって表現され、ω<sub>C</sub>で急峻にカットオフします<sup>(2)</sup>。

図Cはsinc関数の波形を示しています。nを無限数にして、各点の値を図AのFIRフィルタのタップ係数に入れると、図Bに示すような完全な矩形の周波数特性のFIRフィルタになります。しかし無限個のタップをもつFIRフィルタを作ることは不可能です。実際は有限次数で打ち切ることになり、ゲイン

の周波数特性は図Dのように波打ちます。このFIRフィルタにトーン・バーストや方形波などの過渡的な信号を通過させると、プリ・エコーやポスト・エコーが現れます。sinc関数で表わされるFIRフィルタの係数が波形に現れます。

図CのFIRフィルタは、

$$a_0 = a_n, a_1 = a_{n-1}, a_2 = a_{n-2} \dots$$

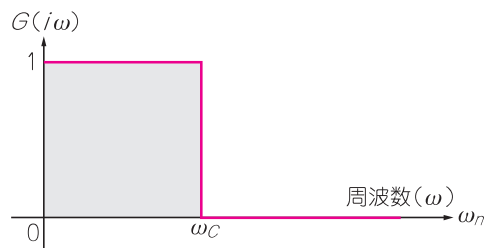
というふうに、外側どうしタップ係数が等しいので対称係数型といいます。このタイプのFIRフィルタには次のような特徴があります。

- 周波数によらず遅延時間が一定(線形位相特性)
- 高調波を含む波形がひずまない

▶ 実際のエコー雑音波形

対称係数型FIRフィルタは、図E(a)のようにプリ・エコーとポスト・エコーの波形が同じようになります。

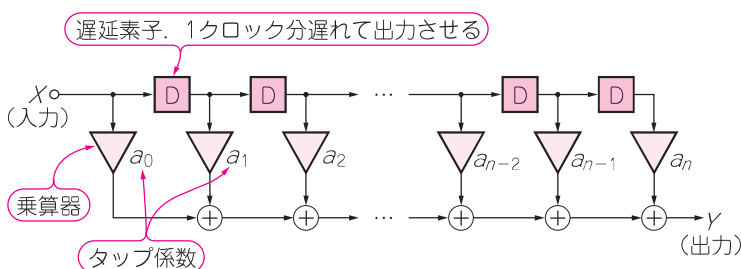
図E(a)は、市販の2種類のD-Aコンバータで方形波を再生したときの立ち上がり応答です。波形が立ち上がる前後にリングングが出ています。立ち上がった後に出るリングングはよく知られた現象ですが、立ち上がる前にリングングが出るのは現実には起こり得ない奇妙な現象です。この現象はFIRフィルタの宿命です。



$$G(e^{j\omega T}) = \frac{\omega_C}{\omega_n} \sum_{n=-\infty}^{\infty} \frac{1}{n\pi} \text{sinc}\left(n \frac{\omega_C}{\omega_n}\right) e^{-nj\omega T}$$

図B sinc関数の無限個のタップ係数をもつFIRフィルタの周波数特性

周波数ω<sub>C</sub>で完全にカットオフする理想LPF。この周波数特性を逆フーリエ変換して、時間領域で表現しなおすとsinc関数が得られる。この理想的なFIRフィルタはエコーが出ない



図A FIRフィルタが入力信号にエコー雑音を付け加えるメカニズムを研究

- $\Delta\Sigma$  モードとフル・エンシ・モードのON/OFF
- ミュートのONとOFF

## SSDACの再生能力

### ● 補完モードと波形再現性

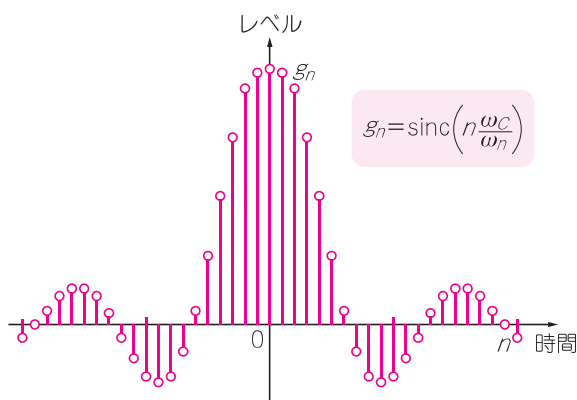
図10に示すのは、SSDACに10 kHzの正弦波を入力して、補完機能をON/OFFしたときの波形です。

### ● プリ・エコーだけが小さくなる非対称係数FIRフィルタ

図E(b)に示すのは、プリ・エコーがポスト・エコーよりも小さくなるように設計されたD-Aコンバータの方形波応答です。これは、FIRフィルタのタップ係数を非対称にしているようです。ただし線形位相は確保していません。偶然にも、このD-Aコンバータを搭載しているサウンド・カードは、長らく使っていたお気に入りでした。

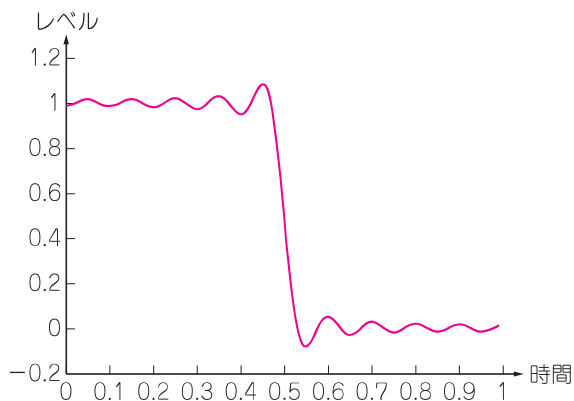
### ● sinc関数係数をもつデジタル・フィルタは自然音再生に向かない？

sinc関数を使ったデジタル・フィルタには原理



図C 係数が対称的なsinc関数の時間変化

$n$ が無限ならFIRフィルタは理想的な周波数特性になりエコーも出ない



図D  $n$ が有限の現実のFIRフィルタの周波数特性は波打つ

図10(a)に示すのは補完機能をOFFしたときです。サンプル点から次のサンプル点まで前者の値を保持するため階段状の0次ホールド波形です。図10(b)は1次補間をONしたときの1次ホールド波形です。サンプル点から次のサンプル点の間を直線で補間します。図10(c)は、2次補間をONしたときの波形です。サンプル点とサンプル点を2次スプライン関数で補間しています。波形が少しひずんでいます。図10(d)は、

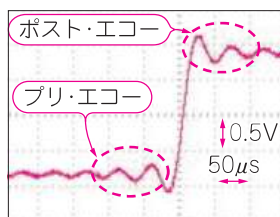
的な欠陥があります。

音声の遅れを解決しようと、sinc関数の計算を簡略化して応答を速めると、今度は単発パルス周辺に余分な信号(プリ・エコーやポスト・エコー)が出ます。完全な解決は簡単ではありません。

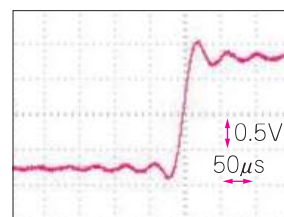
サンプル点のデータにサンプル周波数の半分の周波数成分があれば、これは自然数の逆数の和になり発散します。サンプル周波数の半分の周波数をナイキスト周波数といいます。これがデジタル・フィルタにナイキスト周波数の信号を入れてはいけない理由であり、プリ・エコーやポスト・エコーに含まれる周波数成分がナイキスト周波数と一致する理由でもあります。デジタル・フィルタの鬼門はナイキスト周波数です。

「そもそもsinc関数はインパルス応答として適切か？」という根本的な疑問もあります。自然現象ならインパルス応答は時間遅れの方向にだけ伝わるはずであり、未来方向に延々と余波が伝わるのはおかしいことです。自然現象なら、減衰振動は等比級数で振幅が小さくなるはずなのに、 $1/x$ という減衰の仕方は納得いきません。

プリ・エコーやポスト・エコーはひずみですが、元の波形を飾る効果もあります。CDの音が百花繚乱たる華やかで軽い音になる傾向があるのはその効果です。華やかでいいという人もいるし、自然でないと嫌う人もいます。  
(肥後 信嗣)



(a) プリ・エコーとポスト・エコーのレベルが等しいタイプ



(b) プリ・エコーをポスト・エコーより小さく抑え込んでいるタイプ

図E 係数が非対称なFIRフィルタを採用したD-Aコンバータはプリ・エコーとポスト・エコーの出方が違う

【セミナー案内】 実習・高精度A/D変換の極意

—— アナログ信号を正確に数値化するための関連知識強化セミナー

【講師】 中村 黄三 氏, 9/19(水) 22,000円(税込み) <https://seminar.cqpub.co.jp/>

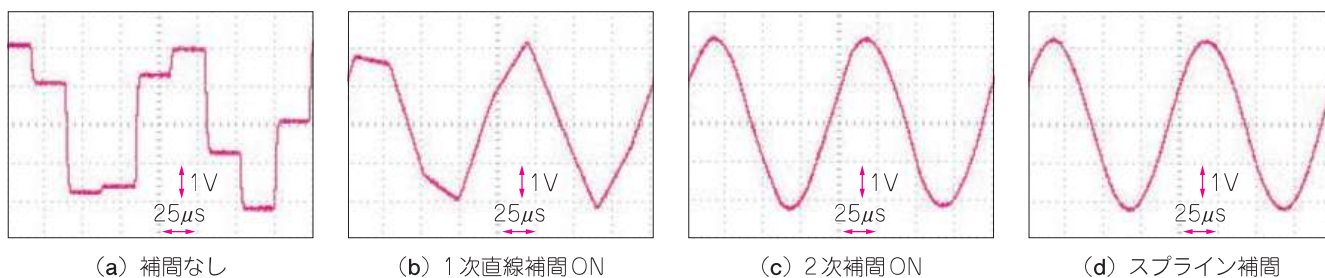


図10 SSDACに10 kHzの正弦波を入力して、補間機能をON/OFFしたときの波形

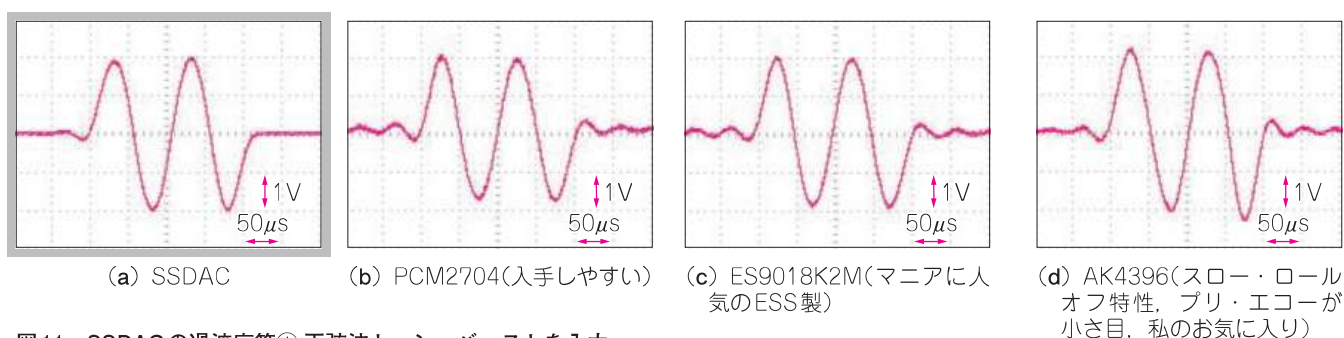


図11 SSDACの過渡応答① 正弦波トーン・バーストを入力

SSDACは波形前後のプリ・エコーとポスト・エコーが抑えられている

3次スプライン補間をONしたときの波形です。きれいに正弦波が再生されています。

● 過渡応答① 正弦波トーン・バーストを入力

図11に正弦波トーン・バーストの波形を示します。SSDACではリングングがひと山出ています。他のD-Aコンバータにおいては1~数周期にわたってリングングが発生しています。このリングングのうち、波形本体の直前に出ているのがプリ・エコー、直後に出ているのがポスト・エコーです。

● 過渡応答② 矩形波とのごり波を入力

図12に矩形波の立ち上がり部の波形を、図13にのごり波の波形を示します。リングングの出方がさらに顕著で、SSDACでは1周期、他のD-Aコンバータでは数周期にわたって発生しています。

過渡応答において、従来のデジタル・フィルタを使ったD-Aコンバータは、現在のデータから1データずつ遠ざかるにつれて、

$$1/2, 1/3, 1/4, 1/5 \dots 1/n$$

というレートで影響力が薄れます(これがsinc関数)。

SSDACは、現在のデータから1データずつ遠ざかるにつれて、

$$\alpha, \alpha^2, \alpha^3 \dots \alpha^n$$

$$\text{ただし、} \alpha = -2 + \sqrt{3} \approx -0.27$$

というふうな等比級数で影響が薄れるため、リングングの収束が非常に早く、1山から2山しか観測されません。

● 高域周波数の波形再現性

図14に、SSDACの弱点である高域の波形再現性を示します。

図14(a)は、14 kHzの正弦波を入力したときのスプ

ライン補間波形はきれいですが、17 kHz[図14(b)]と20 kHz[図14(c)]の波形は崩れています。図15に示すのは、ES9018K2Mを使用したD-Aコンバータでそれぞれ17 kHz、20 kHzを再生した波形ですが、どちらも波形が再現されています。

SSDACは、3次スプライン関数で補間するため、波形1周期に対し最低3点のサンプリングが必要です。CDのサンプリング周波数44.1 kHzの信号の波形を正しく再現できるのは、最高で14.7 kHz(=44.1 kHz/3)までです。SSDACが処理できる最高サンプリング周波数96 kHzの音源であれば、32 kHzまで再現できます。  
オーバー・サンプリングとデジタル・フィルタリ

SSDAC誕生のきっかけ…汎用ロジック製ディスクリットD-Aコンバータ

2004年「手作りアンプの会D-Aコンバータ分科会」にて、上野 智弘氏がR=2RラダーD-Aコンバータ(写真A)を発表しました(5)。

図Fに示すように、汎用CMOSロジックIC7個とラダー抵抗だけで構成された、補間もロー・パス・フィルタもない16ビットD-Aコンバータでした。高域での音の粗さが目立つものの、低域のアタックがとても心地良く、一音一音雑味がなく、歯切れと分離のよい音に感じました。現在主流の1ビット型D-Aコンバータでは出せない過渡応答性能によるものだと直感したのです。(小林 芳直)

【セミナー案内】わかる！トランジスタ回路の設計[演習付き]

—— 実験を通してバイポーラ・トランジスタの動作イメージを理解する

【講師】鈴木 雅臣氏, 9/22(土) 19,000円(税込み) <https://seminar.cqpub.co.jp/>





ング処理をすること、プリ・エコーとポスト・エコーを減らすことはトレードオフの関係です。2011年にこの問題点を指摘している文献があります<sup>(1)</sup>。この文献には「プリ・エコーはすべてのデジタル・フィルタに存在するものであり、これを否定することは、オーバー・サンプリング・デジタル・フィルタを使わないか、特別な応答特性で設計されたものを用いることになる」と指摘しています。

### ● ひずみ率とSN比

表3にひずみ率とSN比の測定結果を示します。SSDACはメーカ製と遜色ありません。

測定には、efu氏開発によるWaveGeneとWaveSpectraを使用しました。測定機材は次のとお

りです。

- ノート・パソコンEPCX101CH, Windows XP SP3
- SoundBlaster PremiumHD  
測定条件は次のとおりです。
- 1 kHz, 正弦波, 1 V<sub>0-peak</sub>
- 窓関数: ハニング
- サンプル数131072個(96 kSPS, 24ビット)

測定時には観測波形がクリップしない最大振幅となるように調整し、被測定D-Aコンバータごとに測定結果が最良になるように行ないました。

### ◆参考・引用\*文献◆

- (1)\* 河合 一; デジタル・オーディオの基本と応用, 2011年12月, 誠文堂新光社。

## 1 サンプリング間に239回積和計算! SSDACはFPGAだからこそ実現できた

### ● DSPやマイコンは非力すぎ

サンプル点とサンプル点を無限スプライン関数でつなぐ、つまりサンプル点ごとの音圧変化を初期値として音波の方程式を解くスプライン関数は、2次微分まで連続の3次式です(Appendix参照)。これを解くには、無限次元の3重対角行列を解かないといけません。これまで「そんなの絶対無理」と思われていたのですが、無限級数として解くことに成功しました。

級数の収束が速いから24ビット・データでも前後13サンプル近傍の計算で十分です。「なんだ簡単じゃん!」と思うかもしれませんが、**本64倍SSDACは、1サンプリングの間に、219回の積和演算をしています。**この演算量はDSPでは処理できません。

DSPで済むならDSPのほうが開発環境揃っているし、プログラム変更も簡単なので、まずはDSPを考えるべきでしょう。しかし、プロセッサにはノイマンのボトルネックという基本的な構造欠陥があり、**演算1に対して100の命令実行が伴います。**つまり、計算1に対して100の発熱をします。さらに演算精度が一定で、高精度、短精度の混在が苦手です。ビット・ハンドリングは効率が落ちるのです。外部との同期も苦手です。

FPGAなら計算量同じなら、発熱は1/10です。複数のプロセスが同時に走るので、外部との同期も得意です。計算精度は必要十分な値に自由に設定できます。

〈小林 芳直〉

### ● SSDACに使うFPGAの最大演算能力は1313 MFLOPS

製作したSSDACのロジック回路のサイズは、デジタル・フィルタ以下で、4000 LE(Logic Element)のMAX FPGAに十分収まりました。この方法はマルチビットD-Aコンバータに適していますが、1ビット型D-Aコンバータに応用することも可能です。

CDの44.1 kHz, 16ビット用のSSDACを作る場合、演算能力が**約95 MFLOPS**のCPUが必要です。

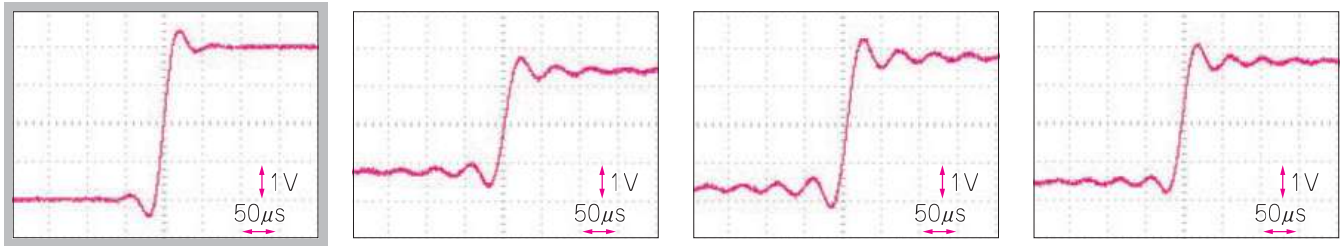
FLOPSはFloating-point Operations Per Secondの略です。

今回製作したSSDACは、**96 kHz, 24ビットまで対応**しています。この演算に必要なCPU性能は、**328 MFLOPS**です。今回使ったMAX10 FPGAの能力があれば、**192 kHz, 24ビット, 4チャンネルを処理することも可能**です。このときの演算性能は、**約1313 MFLOPS**です。

実力が90 MFLOPS前後だと思われるラズベリー・パイ2Bでは到底無理な水準です。マイコンでは遠く及びません。私のCore i5のパソコンは、ベンチマークでの実測値が1880 MFLOPSでしたので、数字上は1313 MFLOPSであれば処理できそうに見えますが、おそらく遅延時間が大きい**リアルタイムの処理は困難**でしょう。

図Gに示すのは、SSDACに搭載したMAX FPGAのフローチャートです。 $c_j$ の演算後に左右のチャンネルに分岐してそれぞれアキュムレータ $b_j$ ,  $c_j$ ,  $s_j$ の演算を並列で行なっています。こういった処理はまさにFPGAの得意とするところ

〈肥後 信嗣〉



(a) SSDAC

(b) PCM2704

(c) ES9018K2M

(d) AK4396

図12 SSDACの過渡応答① 矩形波を入力

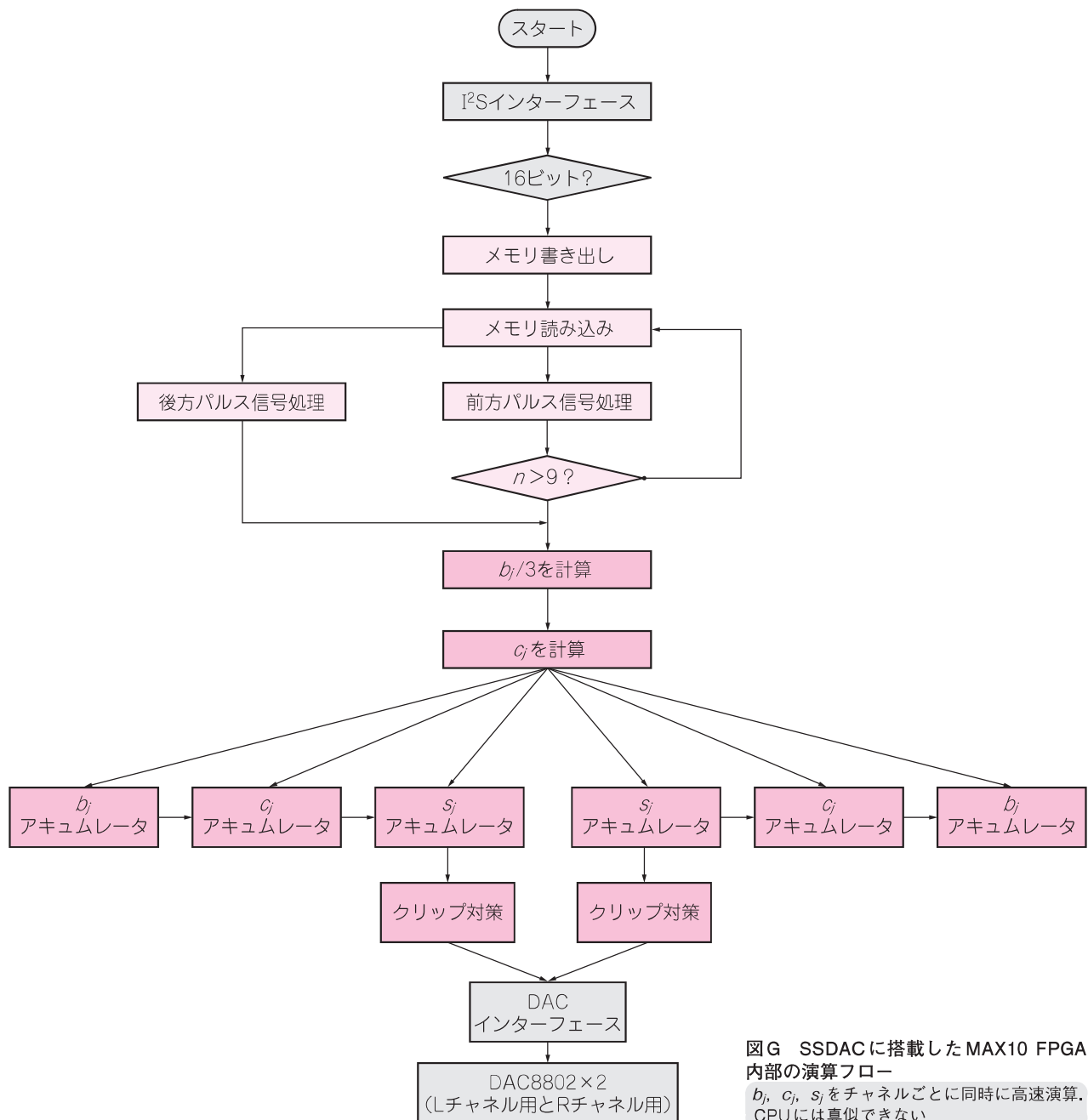
SSDAC以外のD-Aコンバータは数周期にわたってリングングが発生

- (2) \* 勝本 信吾；電子回路論 第15回(最終回), 東京大学理学部・理学系研究科(物性研究所).
- (3) 和田 成夫；よくわかる信号処理, 森北出版.
- (4) new\_western\_elec；アップ・サンプリングの方法.

<http://nw-electric.way-nifty.com/blog/2013/10/post-38c3.html>

(5) \* 上野 智弘ホームページ.

<http://tomozon.sakura.ne.jp/>



図G SSDACに搭載したMAX10 FPGA内部の演算フロー  
 $b_j$ ,  $c_j$ ,  $s_j$ をチャンネルごとに同時に高速演算。CPUには真似できない

【セミナー案内】 実習・LTspiceを活用したローノイズ・アナログ回路設計入門  
 —— 微小/微弱信号回路設計者必聴! ノイズ・シミュレーションの基礎と応用, 実測との比較  
 【講師】 石井 聡 氏, 9/27(木) 30,000円(税込み) <https://seminar.cqpub.co.jp/>



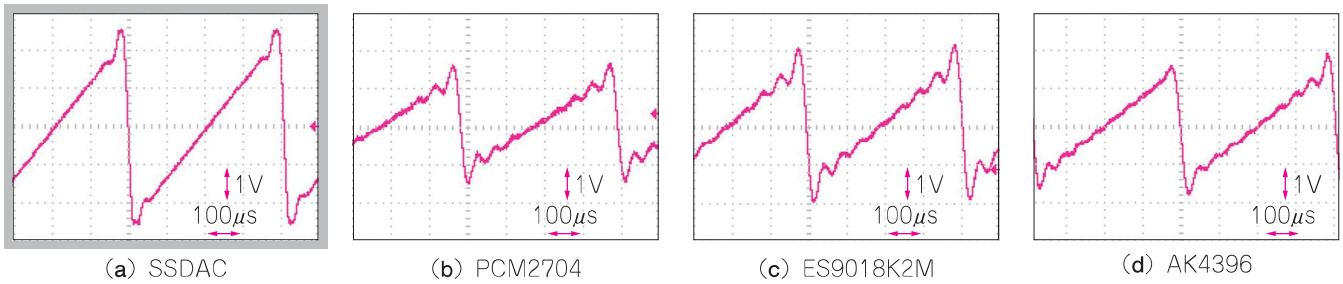


図13 SSDACの過渡応答①のござり波を入力  
SSDAC以外のD-Aコンバータは数周期にわたってリングングが発生

図14 SSDACは高域の波形の再現が苦手

SSDACは波形1周期に対し最低3点のサンプリング・データが必要なので、44.1kHzのCDのデータを再生できるのは最高14.7kHz、96kHzの音源を入力すれば32kHzまで再生できる

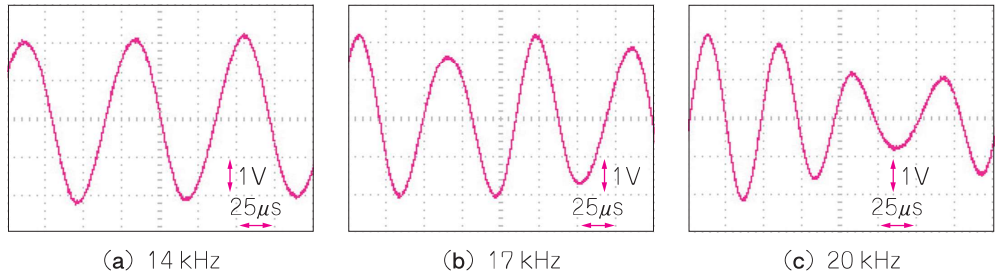


図15 従来のオーバー・サンプリングD-Aコンバータ(ES9018K2M)は20kHzをきれいに再生する

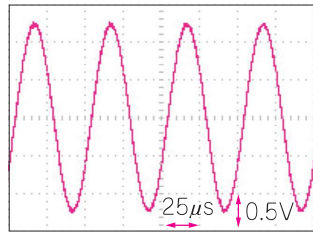


表3 SSDACと従来のD-Aコンバータのひずみ率とSN比

型名	PCM2704	ES9018K2M	AK4396	SSDAC
性能				
THD	0.011 %	0.015 %	0.0083 %	0.0047 %
SNR	59.85 dB	69.39 dB	67.95 dB	78.32 dB

## オーバー・サンプリングの倍率が高いほど外付けアナログ・フィルタが作りやすい

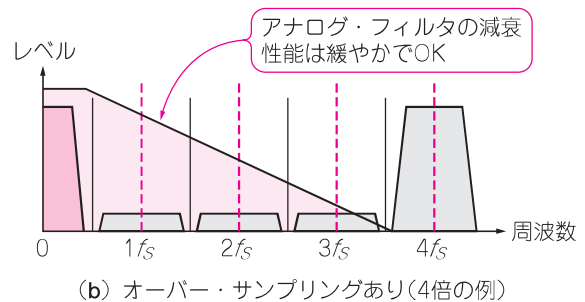
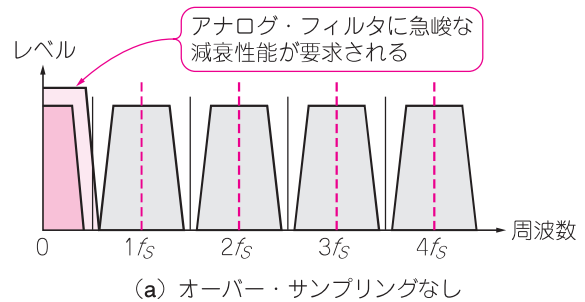
D-Aコンバータは、サンプリング周波数とアナログ信号周波数の差分が、雑音として必ず現れます。これは宿命であり、アナログ・フィルタできれいに取り除かなければなりません。

44.1kHzサンプリングの音楽データで10kHzの音を再生しようとするとき、34.1kHz(=44.1kHz-10kHz)の雑音が発生します。同じく、44.1kHzサンプリングの音源に30kHzの音声データが含まれていると、14.1kHz(=44.1kHz-30kHz)の雑音が発生します。これをエイリアシング雑音(折り返し雑音)といいます。

図H(a)に示すのは、オーバー・サンプリングなしの状態での折り返し雑音の様子です。fsはサンプリング周波数です。D-Aコンバータの出力に接続するアナログ・フィルタには、とても急峻な減衰特性が求められます。音楽CDなら、20kHz以下はゲインがフラットな通過特性を示し、24.1kHz以上は大きく減衰させるという、たいへん厳しいもので。

図H(b)は、4倍オーバー・サンプリングしてから、FIRフィルタを通した信号の折り返し雑音の発生の様子です。折り返し雑音は1~3fs付近ではなく、4fsのとても高い周波数で発生しています。アナロ

グ・フィルタは156.4kHz以上を減衰させればよいのですから設計は楽です。  
(肥後 信嗣)



図H オーバー・サンプリングの倍率が高いほど折り返し雑音の周波数が高く、減衰特性のゆるやかなアナログ・フィルタですむ

### もっと知りたい人へ! 無限スプライン関数の計算アルゴリズム

SSDACの補間に使うスプライン関数は3次スプラインといって、3次関数曲線で補間するものです。

有限区間のスプライン関数は、区間の両端が定まっていることから、始点と終点の2次微分を0とする境界条件からスプライン関数を決定できます(本文図2(a)参照)。しかし音楽信号は過去から未来へ無限に続いている信号ですから、この境界条件は使えず、解くことができません。このようすは本文の図2(b)が表しています。本稿では、SSDACがどのようにして、この課題に取り込んだのかその過程を説明します。

● 4つの音楽データがD-Aコンバータに入ってきたとする

図1に示すのは、CDからD-Aコンバータに4つの音楽データ(P, Q, R, S)が入ってきたところです。知りたいのは、Q-Rのデータ間を結ぶ3次関数です。

$a_j - 1$ や $c_j + 1$ などの係数はすべて不明で、その値はPより過去のデータやSより未来のデータの影響を受けます。 $x_P$ はP-Q間の時間で、 $x_P = 0$ のとき $y_P = d_{j-1}$ です。現在( $x_Q = 0$ )のときに得られたデータを $y_Q$ とすると、区間P-Q, Q-R, R-Sのスプライン関数はそれぞれ次のように表すことができます。

▶ 区間P-Q

$$a_j - 1x^3 + b_j - 1x^2 + c_j - 1x + d_{j-1} \dots (1)$$

▶ 区間Q-R

$$a_j x^3 + b_j x^2 + c_j x + d_j \dots (2)$$

▶ 区間R-S

$$a_{j+1} x^3 + b_{j+1} x^2 + c_{j+1} x + d_{j+1} \dots (3)$$

● 点と点をなめらかにつなぐ条件から3次スプライン関数の係数の公式を導ける

次の3つの条件を満足すると、各区間の境界点(QとR)で隣合う関数がなめらかにつながります。

① 境界の値が等しくなるようにする

$$a_j + b_j + c_j + d_j = d_{j+1} \dots (4)$$

② 境界の1次微分値が等しくなるようにする

$$3a_j + 2b_j + c_j = c_{j+1} \dots (5)$$

③ 境界の2次微分値が等しくなるようにする

$$6a_j + 2b_j = 2b_{j+1} \dots (6)$$

式(6)から、

$$a_j = (b_{j+1} - b_j) / 3 \dots (7)$$

式(4)～式(6)から、

$$c_j = d_{j+1} - d_j - 2b_j / 3 - b_{j+1} / 3 \dots (8)$$

式(4)、式(7)、式(8)から次式が得られます。

$$b_{j-1} + 4b_j + b_{j+1} = 3(d_{j-1} - 2d_j + d_{j+1}) \dots (9)$$

式(9)は3次スプライン関数の係数の公式で、よく知られています。

● これだけの条件ではQ-R間の関数を求められない  
式(9)の右辺の $d_{j-1}$ ,  $d_j$ ,  $d_{j+1}$ は、再生用の元データなので値は決まっています。

$b_j$ がわかれば、式(7)と式(8)を使って $a_j$ と $c_j$ を求めることができ、Q-R間の3次スプライン関数が求まります。しかし、式(9)は連立3元1次方程式なので解くことができません。これが理由で、スプライン関数補間はあきらめられていたのです。

あきらめずに、Pより過去やSより未来のデータが、Q-Rの関数(波形)にどのくらい影響を与えるのかを考えてみます。すると、P点やS点より時間が離れるほど影響が小さくなるのが想像できます。波形への影響が使うD-Aコンバータの分解能(ここでは16ビット)の量子化雑音より小さい未来と過去のデータは無視できます。その境界を見つければ、有限条件で計算でき、 $b_j$ 、つまりQ-R間のスプライン関数を求められそうです。

結果からいうと、16ビットのD-Aコンバータを使う場合、前後9クロック分のデータから求まるQ-R間3次スプライン関数の誤差は、16ビットの量子化雑音以下に収まります(後述)。

● 未来と過去データがQ-R関数に与える影響の計算方法、を確立する

Q-R間の3次スプライン関数への影響が量子化雑音以下になる未来と過去の境界が、現在から何サンプルリング離れているのかはわからないので、試しに現在

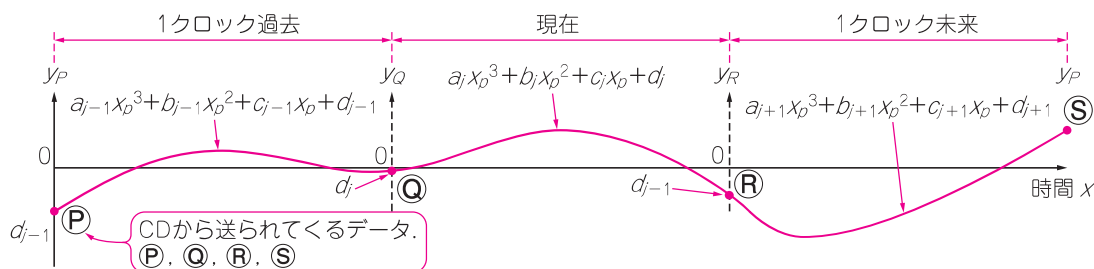


図1 SSDACはオーディオ・サンプル・データP, Q, R, S間を3次スプライン関数で補完する

【セミナー案内】 実習・GNU Radioで始めるSDR入門[教材基板付き]  
—— スペアナ, FMラジオ, TVの自作とキーレスエントリの解析で学ぶ  
【講師】 猿渡 俊介 氏, 9/29(土) 26,000円(税込み) <https://seminar.cqpub.co.jp/>

のサンプル値を含む区間を中心に前後5区間を抜き出し、何か特徴的な傾向がないか調べます。

式(9)と同じように、全9区間の係数 $b$ とサンプル値 $d$ の間に次式が成り立ちます。

●5サンプリング前

$$b_{j-6} + 4b_{j-5} + b_{j-4} = 3(d_{j-6} - 2d_{j-5} + d_{j-4}) \cdots (10-5)$$

●4サンプリング前

$$b_{j-5} + 4b_{j-4} + b_{j-3} = 3(d_{j-5} - 2d_{j-4} + d_{j-3}) \cdots (10-4)$$

●3サンプリング前

$$b_{j-4} + 4b_{j-3} + b_{j-2} = 3(d_{j-4} - 2d_{j-3} + d_{j-2}) \cdots (10-3)$$

●2サンプリング前

$$b_{j-3} + 4b_{j-2} + b_{j-1} = 3(d_{j-3} - 2d_{j-2} + d_{j-1}) \cdots (10-2)$$

●1サンプリング前

$$b_{j-2} + 4b_{j-1} + b_j = 3(d_{j-2} - 2d_{j-1} + d_j) \cdots (10-1)$$

●現在

$$b_{j-1} + 4b_j + b_{j+1} = 3(d_{j-1} - 2d_j + d_{j+1}) \cdots (10_0)$$

●1サンプリング後

$$b_j + 4b_{j+1} + b_{j+2} = 3(d_j - 2d_{j+1} + d_{j+2}) \cdots (10_{+1})$$

●2サンプリング後

$$b_{j+1} + 4b_{j+2} + b_{j+3} = 3(d_{j+1} - 2d_{j+2} + d_{j+3}) \cdots (10_{+2})$$

●3サンプリング後

$$b_{j+2} + 4b_{j+3} + b_{j+4} = 3(d_{j+2} - 2d_{j+3} + d_{j+4}) \cdots (10_{+3})$$

●4サンプリング後

$$b_{j+3} + 4b_{j+4} + b_{j+5} = 3(d_{j+3} - 2d_{j+4} + d_{j+5}) \cdots (10_{+4})$$

●5サンプリング後

$$b_{j+4} + 4b_{j+5} + b_{j+6} = 3(d_{j+4} - 2d_{j+5} + d_{j+6}) \cdots (10_{+5})$$

知りたいのは $b_j$ なので、9連立方程式から $b_j$ 以外の係数( $b_{j-1}$ や $b_{j-3}$ )を何とかして消します。まず式(10-5)を、

$$\alpha_0 = -1/4 \text{ 倍}$$

して式(10-4)に加えます。つまり次の計算をします。

$$\text{式}(10-4) + \text{式}(10-5) \times (-1/4) = \text{式}(10-4)'$$

すると、5クロック前の2次係数 $b_{j-5}$ が消えます。

次に、式(10-4)を、

$$\alpha_1 = \frac{-1}{4+\alpha_0} \text{ 倍} = \frac{-1}{4-1/4} \text{ 倍}$$

して式(10-3)に加えます。つまり、

$$\text{式}(10-3) + \text{式}(10-4)' \times \{-1/(4-1/4)\} = \text{式}(10-3)'$$

すると、4クロック前の2次係数 $b_{j-4}$ が消えます。

このお話し計算結果から、ひとつ上の式を定数( $\alpha_k$ )

倍して、ひとつ下の式に加える作業を行えば、係数が1つ消えることがわかりました。式(10)にたどり着いたときに、 $b_j$ 以外はすべて消え( $b_j$ だけが残り)ます。同じように、一番下の式(10+5)から同様の作業を上方向に行くと、式(10)にたどり着きます。

▶過去未来影響定数 $\alpha$ は、わずか4サンプリングで一定値 $-2 + \sqrt{3}$ に収束する

$\alpha_k$ は、式(10)の方程式で $b_j$ 、つまりスプライン関数を解くかぎを握ります。 $\alpha_k$ は次式を満たす数列です。

$$\begin{aligned} 4\alpha_0 + 1 = 0, \quad \alpha_1(4 + \alpha_0) + 1 = 0 \\ \alpha_k(4 + \alpha_{k-1}) + 1 = 0 \\ \vdots \\ \alpha_\infty(4 + \alpha_\infty) + 1 = 0 \cdots \cdots \cdots (11) \end{aligned}$$

$k = 0, 1, 2, 3 \cdots$ と増やしていくと、次のように $\alpha_\infty = -2 + \sqrt{3}$ (約-0.268)に素早く収束します。

$$\begin{aligned} \alpha_0 = -0.25, \quad \alpha_1 = -0.266667 \\ \alpha_2 = -0.267857, \quad \alpha_3 = -0.267943 \\ \alpha_4 = -0.267949, \quad \alpha_5 = -0.267949 \cdots \\ \alpha_\infty = -2 + \sqrt{3} = -0.267949 \cdots \end{aligned}$$

この値( $-2 + \sqrt{3}$ )をIS(Infinite Spline Constant)定数と命名し、以降単に $\alpha$ と表します。 $\alpha_k$ と $\alpha$ との誤差はいずれ、使うD-Aコンバータの最小量子化雑音(16ビットなら $2^{-16}$ )以下になり、いちいち計算する意味はなくなります。FPGAやCPUのリソースを食うだけです。

$\alpha_k$ がどのくらいの時間で収束するのかを計算してみます。 $\alpha_k$ と収束値 $\alpha_\infty = -2 + \sqrt{3}$ との差分を $\delta_k$ とおきます。すると $\alpha_k$ は次式で表されます。

$$\begin{aligned} \alpha_k = \alpha + \delta_k \\ (\alpha + \delta_k)(4 + \alpha + \delta_{k-1}) + 1 = 0 \\ \delta_k = -\alpha - \frac{1}{4 + \alpha + \delta_{k-1}} = \frac{\alpha^2 \delta_{k-1}}{1 - \alpha \delta_{k-1}} \end{aligned}$$

次の近似で式を簡単すると、次のようになります。

$$1 - \alpha \delta_{k-1} \geq 1 \cdots \cdots \cdots (12)$$

$$\alpha = -0.267949 \doteq -2^{-2} \cdots \cdots \cdots (13)$$

$$\begin{aligned} \delta_0 = \alpha_0 - \alpha = (-0.25) - (-2 + \sqrt{3}) \\ = 0.01794912 \doteq 2^{-6} \cdots \cdots \cdots (14) \end{aligned}$$

式(12)~式(13)から、収束値 $\alpha_\infty = -2 + \sqrt{3}$ との差分 $\delta_k$ について次の大小関係が成立し、 $\alpha^{2k} \delta_0$ が一番大きいことがわかります。

$$\begin{aligned} 0 \leq \delta_k \leq \alpha^2 \delta_{k-1} \leq \alpha^4 \delta_{k-2} \leq \cdots \leq \alpha^{2k} \delta_0 \\ \doteq (-2^{-2})^{2k} 2^{-6} \doteq 2^{-4k} 2^{-6} \cdots \cdots \cdots (15) \end{aligned}$$

式(15)は、差分 $\delta_k$ は、 $k$ が1つ増す、つまり1サンプリング未来または過去に遠ざかるごとに、 $2^{-4}$ 倍ずつ小さくなり0に収束すると言っています。この差分 $\delta_k$ が、D-Aコンバータの最小分解能(量子化雑音)以下に達したら、 $\alpha_k$ は収束値( $-2 + \sqrt{3}$ )一定値として扱うことができます。

D-Aコンバータの分解能が16ビットのときは $k =$

【セミナー案内】波形で実演！ワイヤレス通信におけるデジタル変復調の基礎[講師による実験実演付き]  
—— 基本的な無線データ伝送からOFDMまで、SPICEシミュレータで波形を確認  
【講師】石井 聡 氏, 9/30(日) 19,000円(税込み) <https://seminar.cqpub.co.jp/>



4です。これは4サンプリングより未来と過去のデータについて方程式を解くときは、 $\alpha = -2 + \sqrt{3}$ の定数で計算できるということです。24ビットD-Aコンバータの場合は $k=6$ 、32ビットの場合は $k=8$ です。これで、式(10)の方程式を解くかぎになるIS定数 $\alpha$ の性質が整理できました。

本器のFPGAでは、D-A変換に先行して $k$ 個のサンプルを処理するので、 $\alpha_0 = -0.25$ などの数値は使わず、 $\alpha = -2 + \sqrt{3}$ の一定値として計算しています。

●  $\alpha = -2 + \sqrt{3}$ 固定値として式(10)を解く

スプライン関数の係数計算にもどりましょう。

$\alpha = -2 + \sqrt{3}$ 固定値として、式(10)の無限過去と無限未来の式から、 $b_j$ 以外の係数が消える計算を繰り返します。すると、式(10)は次のようになります。

$$(4 + \alpha + \alpha) b_j = 3(d_{j-1} - 2d_j + d_{j+1}) - X \dots \quad (15)$$

過去 未来

式(15)の $X$ の部分は、 $d_{j-6} \sim d_{j+6}$ が組み合わされた複雑な式です。手間がかかりますが、式(15)を $b_j$ について整理すると次のようになります。

$$b_j = -3(\sqrt{3}-1)d_j + \sum_{k=-\infty}^{j-1} 3\sqrt{3}\alpha^{j-k}d_k + \sum_{l=j+1}^{+\infty} 3\sqrt{3}\alpha^{l-j}d_l = -3(\sqrt{3}-1)d_j + 3\sqrt{3}\alpha(d_{j+1} + \alpha(d_{j+2} + \alpha(d_{j+3} + \alpha(d_{j+4} \dots + 3\sqrt{3}\alpha(d_{j-1} + \alpha(d_{j-2} + \alpha(d_{j-3} + \alpha(d_{j-4} \dots \dots \dots \quad (16)$$

$b_j$ は $d_j$ の無限級数の形で表され、解けそうな雰囲気になってきました。次の漸化式(インパルス関数という)を利用して式(16)をシンプル化します。

$$e_n = d_n + \alpha e_{n-1} \dots \dots \dots \quad (17)$$

$$f_n = d_n + \alpha f_{n+1} \dots \dots \dots \quad (18)$$

式(16)は次のように簡単になります。

$$b_j = -3(\sqrt{3}-1)d_j + 3(2\sqrt{3}-3)(f_{j+1} + e_{j-1}) \dots \dots \dots \quad (19)$$

式(19)において、どこまで未来と過去をスプライン関数計算の対象にするかを決めると、 $b_j$ が求まります。

$n$ サンプリングの過去と未来までをスプライン関数算出用に利用すると決めて、式(17)の $e_n$ を与えると、

1サンプリング前の $e_{n-1}$ が決まり、 $e_{n-1}$ が決まると $e_{n-2}$ が決まり…というふうに、自動的に $e_{n-1} \sim e_1$ のすべての値が求まります。式(18)も同様です。例えば、5サンプリングで打ち切ると決めて、 $f_5$ に値 $d_5$ を与えると、同じ計算を繰り返すことで、次のように $f_4, f_3, f_2, f_1$ がパラパラと決まります。

$$f_4 = d_4 + \alpha f_5, f_3 = d_3 + \alpha f_4$$

$$f_2 = d_2 + \alpha f_3, f_1 = d_1 + \alpha f_2$$

式(19)に $f_5$ と $e_5$ に値を与えて、インパルス関数を4回計算すると、 $b_j$ が求まります。インパルス関数にデータを与える順序を換えれば、 $e_j$ と $f_j$ は同一の回路で計算できます。

● 過去と未来の何サンプリングをスプライン関数計算の対象とするか

無限級数を有限項で打ち切ったときに生じる誤差が、D-Aコンバータの量子化雑音以下になるように決めなければなりません。無限級数はインパルス関数を1回計算するごとに、 $\alpha = -2 + \sqrt{3}$ を乗じます。 $n$ 項で打ち切ると次式で求まる誤差が発生します。

$$\sum_{k=0}^{\infty} |\alpha|^k - \sum_{k=0}^n |\alpha|^k = \sum_{k=n+1}^{\infty} |\alpha|^k = \frac{|\alpha|^{n+1}}{1-|\alpha|} \dots \dots \dots \quad (20)$$

分解能が $D$ [ビット]のD-Aコンバータを使う場合、式(20)の誤差は最小分解能の半分以下( $2^{-D-1}$ )になるように $n$ を決めます。つまり次式が成り立つように $n$ を決めます。

$$\frac{|\alpha|^{n+1}}{1-|\alpha|} = (2-\sqrt{3})^{n+1} \times (\sqrt{3}+1) / 2 \leq 2^{-(D+1)} \dots \dots \dots \quad (21)$$

分解能が16, 24, 32ビットのとき、図1のQ-R間のスプライン関数を求めるために必要な過去と未来のサンプリング・エリア $n$ は次のように求まります。

- ▶  $D = 16$ のとき  
 $n \geq 8.184282 \rightarrow n = 9$
- ▶  $D = 24$ のとき  
 $n \geq 12.39484 \rightarrow n = 13$
- ▶  $D = 32$ のとき  
 $n \geq 16.6054 \rightarrow n = 17$

特集の見どころ

最新プロセッサ

① DAC

FPGAオーディオ製作

② I/Fコンバータ

① パワー・アンプ  
ハイ・パフォーマン  
ス・アナログ

② リニア電源

③ スピーカ

オーディオ製作の素

ボード・コンピュータ・シリーズ 好評発売中

## Web接続, ネットワーク構成からスマホ制御, ハイレゾ再生まで コンピュータ搭載! Linuxオーディオの作り方

Interface編集部 編 B5判 192ページ 定価: 本体2,600円+税

本書は、ラズベリー・パイやBeagleBone BlackといったLinuxボードをオーディオ専用マシンとして使用し、ネットワーク構成や「USB-DAC」の自作、専用デバイス・ドライバの作り方まで解説します。

**CQ出版社** http://shop.cqpub.co.jp/

分解能が16ビットの場合は、サンプル点を中心に過去/未来の9個のサンプリング・データ、つまり19個のサンプリング・データが必要です。24ビットの場合は27データ、32ビットの場合は35データ必要です。

\*

$b_j$ が求まったら、式(7)と式(8)を使って $a_j$ と $c_j$ が計算でき、Q-R間のスプライン関数が求まります。

● SSDACの実力分解能は22ビット

本器は、内部演算をすべて24ビットで行なっており、D-Aコンバータが16ビットなので、表現できなかつ

た下位8ビットについては四捨五入によって16ビットD-Aコンバータの最下位ビットに反映されます。

16ビットのデータを64回出すので、平均値として22ビットの精度が出ていると考えることができます。詳しくは、電気学会電子回路研究会論文 ECT-17-058「スーパーサンプリングDAC  $\alpha = -2 + \sqrt{3}$ 」および電子情報通信学会論文ICD2017-28「スーパーサンプリングDAC  $\alpha = -2 + \sqrt{3}$ 」を参照ください。

〈肥後 信嗣〉

## もう1つのデジタル・フィルタレス・アルゴリズム「フルーエンシ」

● sinc関数を収束のよい2次関数でうまく具合に丸め込んだ感じ

sinc関数の欠点を解析的に解決しようと挑戦したのがフルーエンシ理論です。オーディオの世界からはやや遠ざかりましたが、画像処理の世界で利用されています。これはsinc関数を収束のよい2次関数で近似するというアルゴリズムです。再生波形は0次と1次が連続になります。過渡応答が良く発散しません。代わりに周波数特性が悪くなります。ときどきおかしな結果を出すこともあります。

下記は特許からの抜粋です。

標本化関数 $H(t)$ を区分多項式の形で求めると、

$$(-t_2 - 4t - 4)/4 \quad -2 \leq t < -3/2$$

$$(3t_2 + 8t + 5)/4 \quad -3/2 \leq t < -1$$

$$(5t_2 + 12t + 7)/4 \quad -1 \leq t < -1/2$$

$$(-7t_2 + 4)/4 \quad -1/2 \leq t < 1/2$$

$$(5t_2 - 12t + 7)/4 \quad 1/2 \leq t < 1$$

$$(3t_2 - 8t + 5)/4 \quad 1 \leq t < 3/2$$

$$(-t_2 + 4t - 4)/4 \quad 3/2 \leq t \leq 2$$

と表すことができる。

フルーエンシ関数の応答波形をExcelで計算してみると、図1(a)のようになります。1つのサンプル点の値が前後の4領域に広がります。sinc関数によるインパルス応答は図1(b)に示すとおりで、収束性がよくありません。

一般的なD-Aコンバータが搭載するデジタル・フィルタは、サンプル点の前後の約200点を元に計算します。この200点は、オーバー・サンプリングのサンプル点も含むので、元データのサンプル点は1/8の約25個しかなく、メーカーは、プリ・エコーやリングング、ポスト・エコーの発生を甘受しているのだろうと想像します。最近評価の高いHUGO製のD-Aコンバータは、デジタル・フィルタ計算用のサンプル点を10~20倍に増やしています。た

だし、音が出始めるまでに少し遅延があるようです。

● 1次微分は連続するけれど、2次微分は連続しない  
フルーエンシ理論は、図1(a)のインパルス応答が前後2区間で収まるように、特殊な2次関数を定義しています。1つのサンプル点は、前後2点のサンプル点の値を使って次式で表すことができます。

$$f(x) = ax^2 + bx + c$$

$$a = \frac{-d_{j-2} + 5d_{j-1} - 7d_j + 3d_{j+1}}{4} \quad -0.5 \leq x \leq 0$$

$$a = \frac{3d_{j-1} - 7d_j + 5d_{j+1} - d_{j+2}}{4} \quad 0 \leq x \leq +0.5$$

$$b = -0.5d_{j-1} + 0.5d_{j+1} \quad -0.5 \leq x \leq +0.5$$

$$c = d_j \quad -0.5 \leq x \leq +0.5$$

$$x = 0$$

2次係数が変わるので、2次微分は連続しません。1次微分は連続します。サンプル点の前後2点の値が同じなら、完全な直線になります。斜線も出ます。実は、sinc関数を使ったデジタル・フィルタは、この単純な処理が苦手です。

図1は、Excelの描画機能を利用して作成しました。この機能は、0次、1次、2次が連続したスプライン関数を利用して曲線を描きます。この関数を使って補間値を求めたのがSSDACです。 〈小林 芳直〉

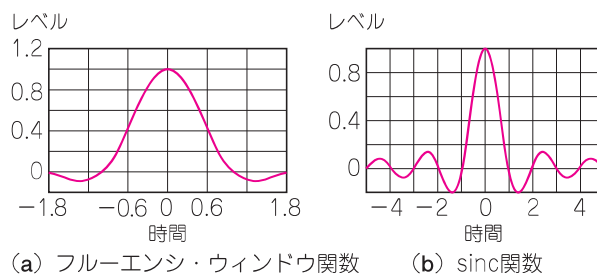


図1 フルーエンシは、sinc関数を特殊な2次関数で近似して高い収束性を実現するというアルゴリズム