

ソフトウェアシンセサイザの開発 — FPD の開発 —

門田 暁人†, 黒田 久泰††, 藤井 秀樹

† 奈良先端科学技術大学院大学 情報科学研究科
†† 京都大学 工学部 応用システム科学教室

あらまし

現在、計算機を利用した音楽の演奏および創作の主な手段として、MIDI 音源などのシンセサイザを用いる方法と内蔵音源を用いる方法があるが、シンセサイザには普及率や互換性の問題があり、内蔵音源は現状では、シンセサイザを利用する場合に比べて限定された機能を持ったツールしか存在しない。我々は、これらの問題を解決する方法として、民生のシンセサイザが備えている主な機能（シーケンサ、マルチティンバ音源、サンブラ、エフェクタとしての機能など）を計算機内蔵の PCM 音源とソフトウェアにより実現した。これをソフトウェアシンセサイザと呼ぶことにする。本稿では、ソフトウェアシンセサイザ“FPD”の設計および評価について述べる。

Development of Software Synthesizer “FPD”

Akito MONDEN †, Hisayasu KURODA †, Hideki FUJII

† Graduate School of Information Science, Nara Institute of Science and Technology
†† Department of Applied Systems Science, Faculty of Engineering, Kyoto University

Abstract

Recently, most of personal computers have a PCM sound set in it, but there are no tools which support overall functions of synthesizer. We have developed software synthesizer “FPD”, which provides most of synthesizer functions, as a multi timbre module, as a sampling machine, as a sequencer, as effects, etc. This paper shows the design and estimation of “FPD”.

1 はじめに

現在、計算機を利用した音楽の演奏および創作の主な手段として、MIDI を利用し、計算機に接続された MIDI 音源などのシンセサイザを用いる方法と、計算機に内蔵された音源を用いる方法がある。

MIDI 音源は計算機に比べると普及率が低く、音色の互換性の問題がある。したがって、演奏データを聴く場合には、特定の音源のために作られたデータを対象とすることしかできない。

一方、計算機内蔵の音源としては、現在は PCM 音源が主流となり、音楽 CD と同等の録音・再生周波数と量子化ビット数を実現している。しかし、シンセサイザの機能を全般的に実現するようなソフトはこれまで存在せず、1 度録音したものを再生するという使われ方が多かった。

これらの問題を解決する方法として、民生のシンセサイザが備えている主な機能を、計算機内蔵の PCM 音源とソフトウェアにより実現することが考えられる。このようなシステムは、計算機内蔵の PCM 音源を用いるので、シンセサイザのように、普及率や互換性といった問題はない。

我々は、このようなシステムを、ソフトウェアによってシンセサイザの機能を実現する意味で、ソフトウェアシンセサイザと呼ぶことにする。

シンセサイザは本来、音響を合成して任意の音色の音を発生する楽器を指すが、本研究では民生のシンセサイザが備えている機能(シーケンサ、マルチティンパ音源、サンブラ、エフェクタとしての機能など)を全般的に実現することを目的とする。

ソフトウェアシンセサイザの開発における大きな問題点の 1 つに、複数音色、多重和音での音響の出力が難しいということがある。これは、計算機には音響処理用の DSP などの専用の演算装置が搭載されていないということによる。しかし、近年の計算機速度の飛躍的な向上により、DSP を用いなくても複数の音を合成して音響出力することができるようになってきた。これにより、ソフトウェアシンセサイザの実現が可能となった。

現在、ソフトウェアシンセサイザの一例として、Macintosh で動作する QUICK TIME 2.0 に実装されているシステムが挙げられる。このシステムは、ソフトウェアによる仮想的な MIDI インターフェースと音源を実現したものであったが [1]、音色数が少ない、新たに自分で音色を増やすことができない、エフェクタの機能が無いなどの問題点があり、GM(General MIDI) への対応も完全ではなかった。我々はこれらの問題点の解決も含め、より多機能なソフトウェアシンセサイザ“FPD”を開発した。本稿は、FPD の開発の紹介である。

1995 年 2 月現在、FPD はフリーソフトとして入手することができ、PC-9801 互換機において MS-DOS および MS-Windows 3.1 上で動作する。音色データは KORG の音源 M3R,03R/W,WAVESTATION SR からサンプリングしたものを加工した。

本システムの利用者としては、演奏データを作成する人、音色データのサンプリングをする人、音楽を観賞する人、の 3 者を想定する。したがって、本研究では、リアルタイム性を考慮した音響出力、GM および SMF への対応、演奏データの入力システムの構築、サンプリングツールの構築、などを考慮した。また、実際にもっとも多く出回っていると思われる GS 規格に対応した MIDI ファイルへの対応も考慮した。

以降、第 2 章ではシステム全体の設計の概要を説明する。第 3 章では、シンセサイザの機能のうち、実現可能な機能についての考察を行い、FPD が個々の機能をどのように実現したかを述べる。第 4 章ではシステムの評価を示す。第 5 章はまとめと今後の課題である。

2 設計の概要

システムは、図 1 に示すように、波形加工部、波形合成部、SMF (Standard MIDI File) 変換部、MML(Music Macro Language) 変換部、に分けられる。

波形加工部は、シンセサイザのサンブラにあたる部分で、楽器の音をサンプリング・量子化し、不

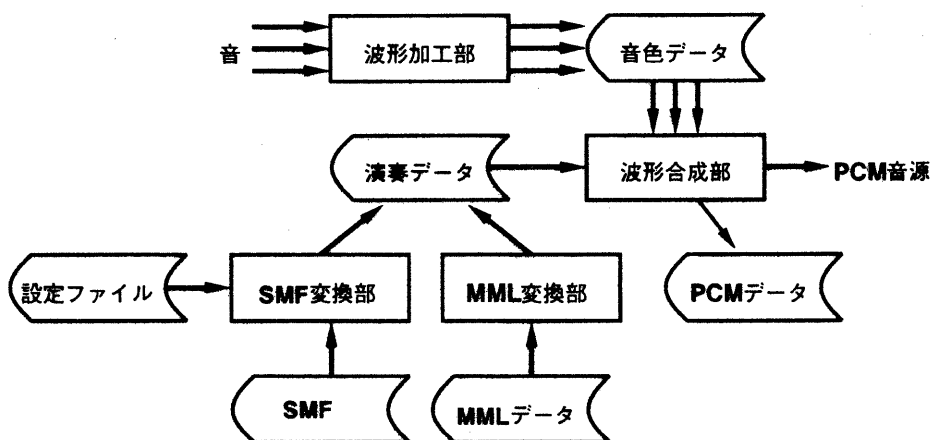


図 1: 設計の概要

要な部分の削除、ルーピング、ボリューム・エンベロープ (以下、エンベロープ) の設定を行う部分のことで、これらの作業によりできあがった波形をファイルとして出力する部分である。(以下、このファイルを音色データと呼ぶ。)

波形合成部は、演奏データと音色データを元に、音響を合成して PCM 音源に出力する部分である。ここでは、音程、音量、エンベロープ、LFO (Low Frequency Oscillator)、エフェクトなどの制御を行う。また、1 曲まるごとの PCM データを出力することもできるが、ファイルサイズは巨大なものになる。演奏データはシンセサイザの MIDI ファイルにあたる部分で、音階、時間、音量、音色データの識別子など、音楽の演奏に必要な情報が記述されたファイルのことであり、SMF 変換部または MML 変換部により出力される。

SMF 変換部は、SMF を演奏データに変換する部分である。変換の際に設定ファイルの情報に従って変換を行う。このことにより、様々な音源に対応した SMF を変換することができる。MML 変換部は、MML データを演奏データに変換する部分である。MML データの編集は、テキストエディタで行う。

3 各機能の実現について

シンセサイザの種々の機能を計算機単体で実現するには、DSP などの専用の演算装置が無いという制約がある。本章では、2 章で説明したシステムの各構成部分について、どのような機能が実現可能かを検討し、FPD においてそれぞれの機能をどのように実現したかを述べる。

3.1 波形加工部

FPD では、サンプリング周波数は最大 44.1 kHz で、7 通りの周波数を選択できる。これは、現在出回っている計算機内蔵の PCM 音源の多くが、複数の周波数での録音・再生に対応していることによる。

実在の楽器の音に近い音色を出すためには、いくつかの音階でサンプリングポイントを設けて、1 つの音色につき複数の音色データを作成することが望ましいが、音色データの量が膨大になってしまうことと、サンプリングポイントでの滑らかな音色の移り変わりを実現することが難しいため、FPD では 1 つの音色で 1 つの音色データしか作成していない。

また、実際のシンセサイザでは、音色ごとにローパスフィルタなどのフィルタを通してから発音する機種が多い。このような機種では、ペロシティの変化に応じて、フィルタのカットオフ周波数を変化させている。しかし、計算機上で演奏時にフィルタリングを行うには処理速度の問題がある。そこで、1つの音色データに対して、フィルタのカットオフ周波数を変えてフィルタリングしたものを複数個保存しておき、ペロシティの値に応じて呼び出すことが考えられる。これにより、ソフトウェアシンセサイザにおいてもフィルタリングが可能となる。

3.2 波形合成部

この部分での大まかな処理の流れは以下のようになっている。

1. 演奏データから必要な情報を抽出する
2. 音色データから波形のサンプルを取得
3. 音程変更
4. 音量変更
5. エフェクトの処理

3.2.1 音程変更

PCM 音源には、可変再生レート、定再生レートの2通りの方式があり、計算機内蔵のPCM音源の多くは定再生レート方式である。

定再生レート方式において音程変更を行うには、サンプリングされた音色波形のサンプル列を、鳴らしたい音程のサンプル列に変換(リサンプル)する必要がある[2]。具体的には、サンプル列を補間し、鳴らしたい音程に応じた時間の間隔でサンプルを抽出する。例を図2に示す。

FPDでは、音程変更の際にデータ間を補うのに1次補間を使用している。処理速度の点では、0次補間(保留)の方が有利であるが、実験の結果、0次補間では音程変更した場合に著しく音色が変化してしまい、実用的ではないと判断した。また、1次補間を用いた場合と2次補間を用いた場合の音程変更後の音色を比べた結果、前者は再生周波数

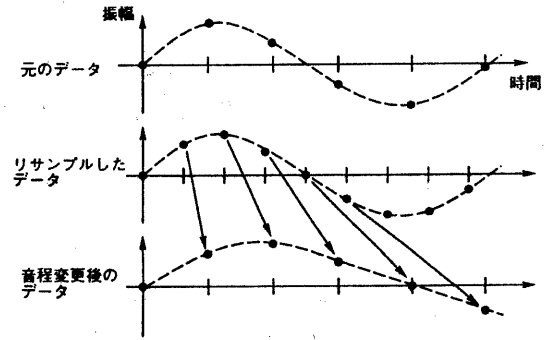


図 2: 音程変更の例

が低い場合に高音部で少し音割れする感じを受けたが、後者に比べて処理速度の点で圧倒的に有利であるので、1次補間を用いることにした。

ピッチベンドやLFOの処理は、サンプル抽出の時間の間隔をずらすことにより実現している。

3.2.2 音量変更

一般に、音量を表す値 V と音色波形の振幅 X は、単純な比例関係にはならない。これは、振幅を半分にしても、人間の感覚では音量が半分になったとは捉えられず、少ししか音量が下がっていないような感覚を受けるためである。そこで、 V が n 下がると X は半分になるという仮定のもとに、以下の式により音量計算を行った。 X_0 が音量変更後の値、 V_{max} は音量の最大値である。

$$X_0 = \lambda X, \quad \lambda = \left(\frac{1}{2}\right)^{\frac{V_{max}-V}{n}}$$

実際には、 λ をあらかじめ計算しておき、テーブルとして利用するので、音量計算にはかけ算を1回だけ要する。

パンポットの処理は、 X_0 を増減して、PCM音源のステレオの左右の出力に対して別々の値を送ることにより実現している。エンベロープの処理は、 X_0 をエンベロープのパラメータに応じて増減させることによって実現している。

3.2.3 エフェクト

最近のシンセサイザにはデジタル・マルチ・エフェクタを内蔵しているものが多く、エフェクタは音場の制御・音色の加工のために、必須の機能となっている。

FPDでは、音場を制御するためのエフェクタとして、大きな効果を期待できると思われるディレイ系のエフェクタを実現した。ディレイ系のエフェクタの基本構成を図3に示す。

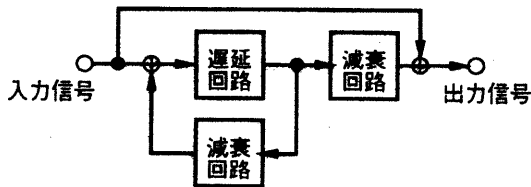


図3: ディレイ系エフェクタの基本構成

FPDでは遅延回路をディレイバッファとしてメモリ空間上に実現した。この場合、バッファの大きさが遅延時間になる。

ディレイ系のエフェクタは、バッファからの出力を入力信号に加算するとともに、出力をバッファへの入力にフィードバックすることにより実現できる。バッファからの出力に周波数変調をかけたリ、左右逆位相で入力信号に加算したりすることで、ステレオ、コーラス、フランジャ、アンサンブル等のエフェクトが得られる。ただし、より高い効果を得るためには、複数のバッファが必要となる[3]。複数のバッファを設けたエフェクタを実現するには処理速度の問題があるので、FPDでは、複数のバッファを用いた特殊な場合として、1つのバッファを用いて遅延時間に応じてバッファの出口を複数設けるような構成のエフェクタを実現した。これを図4に示す。

バッファからのフィードバックは、1つの出力に対してのみ行われる。これは処理速度との妥協点であると判断した。加工部では、信号に対しての周波数変調および減衰を行う。位相制御部では、場合に応じて位相を反転する。

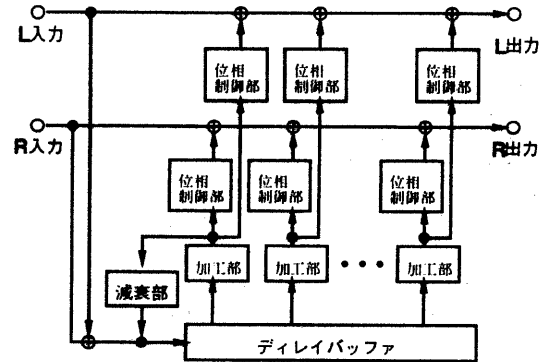


図4: FPDにおけるエフェクタの構成

FPDではこのエフェクタを、直列もしくは並列に最大で4つまで接続することができる。

3.3 SMF 変換部

SMF 変換部では、設定ファイルの情報を元にSMFを演奏データに変換する。このことにより、既存のシーケンスソフトを利用してSMFを作成し、FPDで演奏することが可能である。

設定ファイルでは、1つの音色番号に対して、音色データの識別子、エンベロープ、音量バランス、ビブラートのかかりぐあい、ビブラートのかかり始める時間、などが記述されており、現状ではRolandのGS音源に対応した情報が記述されている。設定ファイルを変更することにより、様々な音源に対応したSMFを変換することができる。

3.4 MML 変換部

MMLは演奏データを記述するための言語で、主に計算機内蔵の音源のための演奏データの記述に使用されていた。MMLによる入力方式は、ステップ入力などの他の方法と比べて、比較的短時間で入力ができるという利点があるが、その反面、緻密な演奏データの作成に不向きだという欠点がある。

MMLによる入力方式を採用した理由は、実装のしやすさと、既存のエディタを利用できることである。MMLデータの例を図5に示す。

```

! Happy Song

$k @152 o4 v125 _6 ;Kick
$s @157 o4 v127 _2 ;Snare
$h @160 o4 p2 v97 _0 ;Closed HiHat
$o @169 o4 p-2 v105 _0 ;Open HiHat

A #R70,140,250,0 ;Reverb set
A #G0,0,-1,12,0,8,3 ;Chorus set
B M8,1,32,8 ;Pitch Modulation

? t120 116 q3 _-1 _v3
A o3v123 @34 c8re8rg8f8ra8r>c8(g8rb8r>d8<c8rc
B o4v118 @82 rr>cc<gg88aara&a4rrgrarbr>ccrc
C o4v116 @63 r8.cr8cr r4.cr r8.dr8dr.ec<re
D o4v116 @63 r8.er8er r4.fr r8.gr8gr >c<grc
E #1 [$krc8c$sc8$kr]3 $sccrc
F #1 $h[cr8ccr:rc r8.ccrer]2cr $ocrc

```

図 5: MML データの例

4 評価

本章では、シンセサイザの機能を評価する項目として、再生周波数と最大同時発音数について述べる。

FPDでは、5.52, 8.27, 11.03, 16.54, 22.05, 33.08, 44.10kHzの7種類の再生周波数が選択できる。最大同時発音数は、主にCPUの速度と再生周波数によって決まる。Intel i486 CPUにおいて、CPUクロック周波数25MHz、50MHzについて再生周波数と最大同時発音数の関係を調べた結果を図6に示す。25MHzで約14MIPS、50MHzで約28MIPSである。

GM対応のSMFを演奏するには、30程度の最大同時発音数があれば十分だと思われる。FPDでは、i486,50MHzのCPUならば再生周波数16kHzで最大同時発音数29音を実現できる。なお、この数値はステレオ再生の場合であり、モノラル再生の場合には約1.5倍の最大同時発音数を実現することができる。また、この実験結果はエフェクトを用いない場合のものであり、エフェクトをかける強さによって最大同時発音数は変化する。

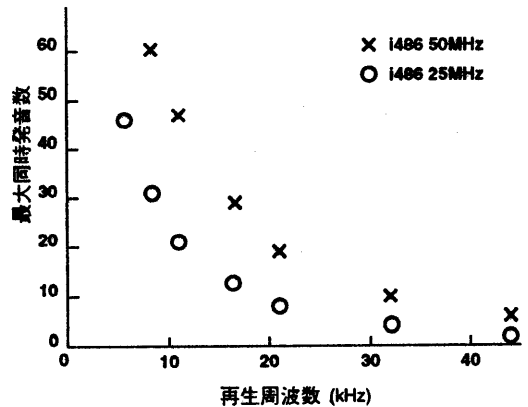


図 6: 再生周波数と最大同時発音数の関係

5 おわりに

本稿では、ソフトウェアシンセサイザ“FPD”の開発について紹介した。

今後、計算機の実行速度の一層の向上が見込まれ、ソフトウェアシンセサイザの利用価値がますます高まっていくと思われる。FPDの今後の課題は、ディレイ系以外のエフェクトの実現、文字や画像との同期などである。

謝辞

シンセサイザからのサンプリングの許可を頂きました株式会社KORGに深く感謝致します。

参考文献

- [1] Drucker, D.: “QuickTime turns 2.0”, CD-ROM Prof., Vol.7, No.4, pp.27-30 (1994).
- [2] 岸恒行: “電子楽器の成り立ちとその変遷”, トランジスタ技術, pp.413-427, CQ 出版社 (1990-2).
- [3] 松村南: “マルチ・エフェクタとサラウンド・プロセッサの製作”, トランジスタ技術 SPECIAL, pp.56-75, CQ 出版社 (1990).