

7Q-10

テンポ情報を基にした
MIDI/PCM同期システム佐々木 宏 飯島 重喜 鈴木 孝
東京工業高等専門学校 情報工学科

1 はじめに

現在の情報処理と音楽を結び付けるものの代表格にMIDIとPCMがある。

MIDIは技術者の目から見れば通信プロトコルの一種にすぎないが、一般的な人(例えばミュージシャン)にとっては、容易に高度な情報通信ができるものとして、現代音楽の重要な位置にある。

PCMはアナログ信号をデジタル信号に変換して扱う信号技術である。アナログ信号として「音」をマイクから入力すれば、デジタル信号に変換することによってより複雑な音声信号処理ができる。

今回は、この異なる特質を持つ2種類の信号を、統一的に扱うための重要なポイントとして時間軸に着目し、時間軸を可変的な「テンポ」として扱うための処理方法を述べる。

2 相対時間系と絶対時間系

ここでは、それぞれの信号がどのような時間軸で成り立っているかを述べる。

MIDIは、電子楽器の制御を前提に設計されたものである。時間軸は音楽に密着するように考えられている。当然、テンポという概念はあるが、時間軸の基本となる単位はMIDIクロック(タイミングクロック)によって決まる。MIDIクロックは4分音符あたり24の割合で送出され、これを使って複数のMIDI機器(例えばMIDIシーケンサーとリズムマシン)の同期を行う。しかし、4分音符は音符の長さとしては一定の長さということになるが、テンポによって発音している時間や次に発音するためのタイミングが変わるので、テンポが変わればMIDIクロックの間隔が変わることになる。これを用いることにより楽譜の表記にある「rit」(だんだんゆっくり)など、テンポを変化させ表現ができるのである。しかし、5秒後にはどの音を鳴らしているのかということは、テンポによって変わってしまうので、簡単に決めることができない。このような時間軸を相対時間系の時間軸と言われている。

PCMの時間軸については、PCMそのものがどのようなものかを考えればよい。例えばサンプリング周波数48kHzで、ある音(曲)をサンプリング(録音)したとする。この時に作られたデータを始めから5秒後の位置から再生することは容易にできる。しかし、録音したデータを再生するとき、テンポを変えるためにサンプリング周波数を変更すると、再生された音は、ピッチが変わってしまう、元の曲とは違ったものに聞こえてしまうので、PCMはテンポを変化させることは不向きであるといえる。つまり、音楽の再生にはサンプリング周波数を変えることが許されなくなり、固定された時間軸になる。このような時間軸を絶対時間系の時間軸と言われている。PCMのほか、映像(動画)なども、この時間

系に属する。しかし、最近は音声用DSP(Digital signal processor)を用いた信号処理により、この問題が解決できるようになっている。

3 MIDIとPCMの同期再生系

では、MIDIとPCMを統一的に扱えるようにするためには、どのような時間軸を用いれば良いかということになる。これは、2種類の信号をどのような目的で扱うかで決まることになる。例えば、無人の場所でPCMを使って図や物を説明しながら、MIDIでBGMを流す、というような使い方をするならば、MIDIの再生開始だけをPCMの時間軸に合わせて位置付ければよく、MIDIで演奏する曲のテンポをPCMのクロックに合わせる必要がない。このような時は、PCMの時間軸、絶対時間系の時間軸にすればよい。しかし、今回は可変的なテンポを扱えるようにするため、音楽のために作られたMIDIの時間軸を用いるべきである。PCMの再生開始位置をMIDIの時間軸、相対時間系の時間軸に合わせる様にして、テンポを変えても、楽譜上では同じ位置からPCMのデータを再生開始できるようにする。MIDIの時間軸をこのシステムのテンポ用クロックとする。

以上のことを踏まえて、PCMの再生位置をテンポ用クロックに合わせる方法は3通り考えられる。(Fig.1(a), (b), (c) 参照)

Fig.1(a)はPCMの再生開始位置とMIDIの再生開始位置だけを合わせる方法である。再生中はそれぞれ独自の時間軸で進むことになる。

Fig.1(b)は一定のテンポで生演奏した音をサンプリングしたデータを小節単位に切り分け、それをテンポ用クロックの小節の先頭に再生位置を合わせ同期をとる方法である。この方法ではテンポを遅くするとPCMのデータ間が再生されないことになる。テンポを速くすると、データの再生中に別のデータを再生しなければならない状況になる。小節内ではPCMの再生音とMIDIの演奏のテンポがずれる。以上のような問題が生じてしまう。この問題を解決する方法としては(1) ピッチを変えても演奏上大きな影響を与えない音(リズムパターンなど)の場合、サンプリング周波数を変更してPCMの再生時間を変化させることにより、前に上げた問題が生じないようにする。(2) 音声用DSPを使う。と、いうような方法で解決することができるので有効な方法の1つと考えられる。

Fig.1(c)はPCMのデータの内容を1音(例えばピアノのCの音)に限定して、クロックの分解能に合わせてPCMの再生位置を任意の位置で合わせられるようにする。確かに、この方法ならばテンポを変えても、発音するタイミングはテンポ用クロックに合わせているので問題はない。また、この方法はFig.1(2)の拡張版と考えられる。

3通りの方法のうち、Fig.1(a)はPCMが再生開

始以外はPCM自身の時間軸（絶対時間系）で管理されるため、テンポの変更ができなくなる。Fig.1 (3) はハードディスクへのアクセス回数が多くなるため、ファイルへのアクセス時間が発音開始の遅れになる可能性がある。よって、今回はFig.1(b)の方法で試作することにした。

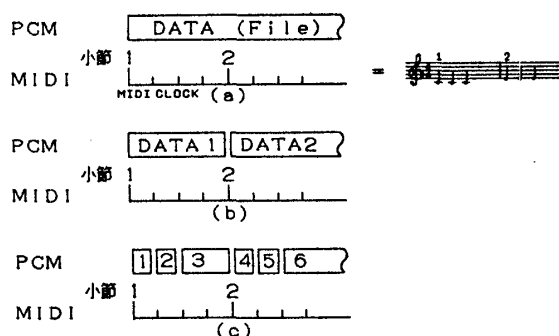


Fig. 1 MIDI/PCM同期方法

4 同期システム 構成

Fig.2 にMIDI/PCM同期システムの構成を示す。

このシステムは主に、PCMモジュール（音声録音再生ボード、カノーブス製サウンドマスター使用）、MIDIインタフェース（ローランド製MPU-PC98使用）、PC（パーソナルコンピュータ、NEC製PC-9801FS使用）、ハードディスクから成り立っている。

PCMモジュールは、主にA/Dコンバータ、D/Aコンバータ、制御用CPU、LPF、FIFOから構成されている。今回はPCMデータはメインメモリ上には置かず、直接リアルタイムでハードディスクに書き込み、読み出しを行う「ハードディスクレコーディング」という方法をとっている。この方法の特徴は、PCMデータをファイルとして管理でき、補助記憶媒体の容量が許す限りの長時間の録音・再生ができる。そのためには高速・大容量のハードディスクが必要になる。

MIDIインタフェースは外部のシンセサイザなどのMIDI機器を制御するほか、このインタフェース上のCPUは、テンポ制御の機能を持つため、この機能を使って本システムのテンポ制御、つまりテンポ用クロックの発生を行っている。

PCは、ホストの役割を持ち、主に各装置の制御、及び装置間の通信制御、受信したテンポ用クロックに対する制御の割当、テンポをリアルタイムで変更するためにMIDIインタフェースに指示を与えるなどの処理を行う。また、リアルタイムでのテンポ情報の変更はキーボードからの入力で行えるようにしている。将来は他の入力装置を用いて楽器のように扱える予定である。

このシステムの特徴はハードディスクレコーディングによるメモリの有効活用と、ホストとテンポ用

のCPUを独立させることによりホスト側CPUの負担の軽減である。このため、ホスト側には新たな機能を拡張させることができる。

実際にこのシステムを制作してPCMのデータとして1小節のリズムパターンを使ってMIDI/PCMの同期を行ってみたが問題なく再生できた。

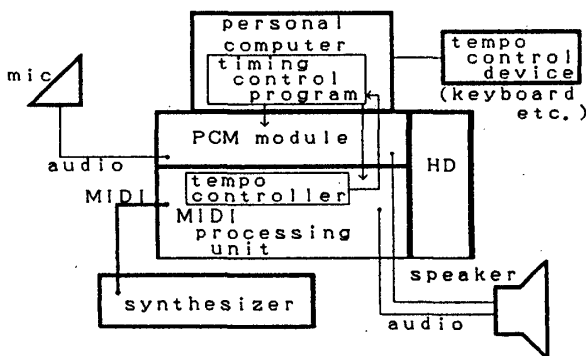


Fig. 2 MIDI/PCM同期システムの構成図

5 今後の機能拡張

テンポの変更ができるシステムになっているので、これを使い、リアルタイムで不自然なくテンポを変える。たとえばオーケストラの指揮者に合わせて自由にテンポを変えたりできるなど、楽器のように扱えるようにする。

また、自然なゆらぎのあるテンポをつけるなど、機械的ではなくヒューマンライクなテンポの変更を行えるようにして、より自由な音楽制作をユーザーに提供できるようにする。

ほかにもDSPによるエフェクトや、各種アナログ・デジタルインターフェイス、映像など他の媒体と複合させ、トータルとしてマルチメディアシステムへの拡張が考えられる。

6 おわりに

テンポと時間軸を中心にMIDI/PCM同期システムの基本的な構造について述べた。

今後MIDI/PCMのそれぞれの機能拡張と共にユーザーインタフェースの設計を行っていく予定である。

文献

- (1) CANOPUS Electronics Co., Ltd.:
Sound Master V USER'S GUIDE
- (2) Roland:
MPU-PC98 テクニカル・リファレンス・マニュアル