

I C O T o n e ~
メロディ抽出プログラム A c k I

2H-5

阿久津 達也, 小池 汎平, 田中 英彦, 大須賀 節雄
(東京大学 工学部)

1. はじめに

Ack-Iは、ICOToneシステムにおいて、マイク、CDプレーヤ等より入力したアナログのオーディオ信号から、その曲の楽譜を作り出す為のツールである。但し、Ack-Iは、楽譜を作り出すまで全ての作業を行うのではなく、入力したアナログ信号から、その音程、音長、音の強弱、音の出だしのタイミング、ビブラートやトレモロの有無等の情報を分離し、MIDIコードに変換し、それをFSI (ICOToneの採譜プログラム) ([7])に提供するというものである。Ack-Iの最終的な目標は、FSIと協力し、オーケストラのような多くの楽器による演奏から、すべてのパートの楽譜を生成する事である。しかしながら、その目標の達成は非常に困難なものであるので、簡単なものから初めて、徐々にレベルアップしていくつもりである。

ICOToneは、現在Sunワークステーション、及び、PrologマシンPSI上にインプリメントされているが、Ack-Iはハードウェアの関係上、PC-9801上にインプリメントしてある。しかし、将来的には、SunやPSI上に移植しICOTone本体と結合し、更にウェーブ・メモリやFFTプロセッサ等と結合し、高速化、大容量化をはかる予定である。なお、現在の段階では、単音による演奏の認識機能のみしかサポートしていない。

2. 方法の検討

実際に演奏した楽曲からの自動採譜の研究は、色々[1],[2],[3],[4])となされているが、多くの場合について採用されている方法は、

- (i) 入力信号をA/Dコンバータにより、デジタル信号に変換する。
- (ii) FFT等の方法によりその周波数スペクトル、基本周波数等を得る。
- (iii)(ii)で得られた情報を主体として、音程、音長、タイミング等の情報を得る。
- (iv) (iii)で得られた情報を元に、楽譜を構成する。

というものである。但し、場合によっては、このままストレートに処理するのではなく適当に前のステップに戻ったりしてフィードバックをかけて処理する事もある。

このうち、(i)の部分については議論の必要はなく、(iv)の部分についてはFSIの担当する部分であるので、(ii)と(iii)の部分について議論したい。なお、以下の議論では単音の認識に限る事にする。

まず、(ii)についてであるが、本システムにおいてもFFTを使用している。MEM(最大エントロピー法)の使用も考慮したが、計算時間を考え、FFTを採用する事にした。問題は、基本周波数の抽出である。単純にFFTを用いて音程を正しく同定する為には、周波数分解能が高くなければならないが、その為には、1回分のFFTの為の測定時間を長くとらなければならない。しかし、その測定時間を長くすると、今度は、細かい音の移り変わりの情報を得る事ができなくなり、音長やタイミングなどを正確に知る事が難しくなる。そこで、従来から、短時間で高い周波数精度を得る為の方法として、ケプストラム法、位相補償法([1])、複素スペクトルを利用する方法([2])等が研究されてきた。しかしながら、Ack-Iでは、現在のところ、単純で正確な音源からの信号しか分析していないので、高調波を利用する単純な方法を用いている。その方法は、

- (1) いくつかのピーク値を検出し、それらを基本周波数の候補とする。
- (2) (1)で求めた候補のうち、低い周波数のものから順にその整数倍の周波数付近にもある程度のピーク(高調波成分)がでていのかどうかを調べ、そうであれば、その周波数を基本周波数とする。
- (3) 更に、(2)で求めた基本周波数の整数倍の周波数付近でそのピーク位置を調べて高調波周波数を求め、それを用いて基本周波数を修正し、より高い精度を得る。

というものである。

(iii)については、まず、(ii)の基本周波数の時間変動を一旦フィルタリング(平滑化)し、誤って検出した基本周波数を取り除いている。その後、その基本周波数の時間変動のブロック化を行い、音長、タイミングを抽出している。

3. インプリメント及び結果

システムのインプリメントは、PC-9801において、FFTの主要部、及び、A/D変換の為のインターバル・タイムによる割り込み処理の部分をアセンブラで記述している他は、C言語で行っている。アナログ信号のサンプリング間隔は、A/Dコンバータ、及び、メモリ容量の制限等から80 μ sec前後である。FFT等の解析処理は、リアルタイムではおこなわず、一旦、メモリ上にサンプリングした信号をため込んでから、解析する事にしている。最終的にはリアルタイムで行えるのが望ましいが、現在は方法の検討、実験段階であり、解析にかかる時間はあまり重視していない。FFTは1回につき1024点で、サンプリング位置が512点進むに従いFFTを行っている。つまり、512点ずつオーバーラップしているわけである。これは、基本周波数の時間変動を細かく得る為である。しかし、これでも細かい時間変動を調べる事ができないので、今後、2節で挙げた方法等を検討していきたい。

また、現在の時点では、ICOTone本体と接続されていずFSIと結合していないので、MIDIコードを出力する代わりに、直接、音程、音長データ等から音符に変換して、簡単な楽譜をPC-9801上の画面に表示している。

現在、実験に用いている音源は、まずは安定した音源から始めようという事で、ヤマハのボータ・サウンドを用いて行っている。音程の認識率は、音源が安定しているせいもあって非常に良く、ピアノ音モード等では、100%近い認識率を得ている。しかしながら、ハーブシコード音モード等を選択すると、やや認識率は低下して80%を切る事もある。音長やタイミングの認識は、演奏者の技量のせいもあるかもしれないが、あまり、良くなく、50~90%程度である。

4. おわりに

現在の時点では、安定した音源からの単音の認識がある程度できただけであり、実際の道具として使えるまでにはいたっていない。しかしながら、今後の研究への足掛りをつかむ事はできたと思う。

さて、Ack-Iの今後の予定としては、まず、IC

OTone本体と結合する事が第一である。その際には、PC-9801を使わずに、直接、Sun、もしくは、PSIにA/Dコンバータを接続したい。特に、FFT以後の解析には、多くのヒューリスティックスが必要となるので、PSIによるPrologの利用は、有用であろう。そして、Ack-Iの次の段階としては、

- (i) 人間の歌声等のように不安定な音源からの採譜。
- (ii) キーボード等において伴奏つきで演奏した場合のメロディパートと伴奏パートの採譜。

を達成したい。このうち、(i)については、まだ、その波形や周波数スペクトルを調べていないので、何とも言えないが、(ii)については、その周波数スペクトルを見ると、二音くらいであれば、各々のパートの音程の基本周波数付近にピークがある事が確認できるので、可能性は非常に高いと思われる。

謝辞

本研究の為に高速FFTサブルーチンを提供して下さいました大須賀研の吉沢 康介氏に感謝の意を表したい。

<<参考文献>>

- [1] 井口 征士：音楽情報の処理—電算機を用いた自動採譜—、計測と制御、Vol.19, No.3, 1980.
- [2] 原、井口：複素スペクトルを用いた周波数同定、計測自動制御学会論文集、Vol.19, No.9, 1983.
- [3] 北垣、清水、末武：コンピュータによる楽音認識の試み、日本音響学会誌、Vol.33, No.9, 1977.
- [4] Special Issue on Computer Music, ACM Computing Surveys, Vol. 17, No. 2, 1985.
- [5] 平田、他：新世代音楽システムICOTone—立派な和音LPC86、日本ソフトウェア科学会第3回大会論文集、1986.
- [6] 平田、他：新世代音楽システムICOToneのユーティリティ体系、本大会 2H-1.
- [7] 青柳、他：ICOTone~採譜プログラム FSI on PSI、本大会 2H-2.