

パソコンによるニューロン活動の記録と解析

伊藤 憲一・金子 健也・加藤 宏司

(山形大学・医学部・生理学第二講座)

電気生理学の分野において、生体から記録される信号は脳波をはじめとし心電図、筋電図などほとんどがアナログ信号である。コンピュータが導入されるまでは、これらのデータを記録・解析するためにペンレコーダあるいはポリグラフと呼ばれるアナログ式の記録装置が用いられてきた。しかし、1980年代始め頃から比較的入手しやすかつ安価なパソコンが一般に普及し始め、この研究分野でも徐々に利用されるようになった。20年近く経った現在、コンピュータは生理学の研究に不可欠の道具となっている。我々の研究室では、脳の海馬薄切標本からニューロンの電気的活動を記録し、これをコンピュータに取り込んで解析を行っている。最近では、ニューロンの電気的活動を記録・解析するためのソフトウェアが数社から市販されているが、我々は経済的理由と余分な機能がない研究目的志向のシステムを構築したいとの理由から、比較的入手し易い市販の A/D ボードとコンピュータを組み合わせ、これに適合したアプリケーションソフトウェアを作製して実験に用いている。データ収録のための一般論は他の著書[1]に譲ることにし、本稿においては我々の研究室で実際に使用しているシステムの製作過程の具体例を示し、今後読者で類似したシステムを構築するときの参考になればと考えている。

1. システムの概要

このシステムでは、ニューロンの電位変動を細胞外に設置したガラス電極で記録し、前置増幅器で増幅した後、A/D コンバータ(Analog/Digital Converter)を介してコンピュータのメモリーに読み込む。そして、その電位波形をコン

ピュータのディスプレイに表示し、ハードディスク(大容量外部記録装置)にファイルとして保存する。波形の記録と同時に波形の高さあるいは傾きを計測し、その値を画面上(ディスプレイ)にプロットして時間経過を観察する(実時間処理)。保存した波形ファイルは、実験後に呼び出し解析したり、出力装置(プリンター、プロッター)に打ち出す。この条件を満たすハードウェアおよびソフトウェア(オペレーティング・システムや言語)を選択し、プログラムを作成してシステムを構築する。

2. ハードウェアの構成

図1に全体のシステムのブロックダイアグラムを示す。図1の『ガラス電極(R_{ELE})—前置増幅器(P-Amp)—中間増幅器(IM-Amp)—オシロスコープおよびコンピュータ』の構成が信号記録系で、『コンピューター電気刺激装置(Stim)—アイソレーター—刺激電極(S_{ELE})』の部分が電気刺激系である。コンピュータには外部機器との信号のやり取りをするために2枚のインターフェースカードが付属している。

はじめに検討すべきことは、記録する電位波形の大きさと時間スケールである。細胞の電位活動を記録するには、専用の記録電極と前置増幅器が必要である。その原理と実際の回路については、実験生物学講座に詳細に記載されているので、本稿では簡単に述べることにする[2]。ニューロンの細胞外電位(フィールドポテンシャル)の大きさは、0.1 mV~20 mV(平均10 mV)で、時間スケールは5~20 msである。一般に前置増幅器は、インピーダンス変換とフィルタリングの機能を持っているが、増幅機能はないのでニューロンが発生した電位がそのまま

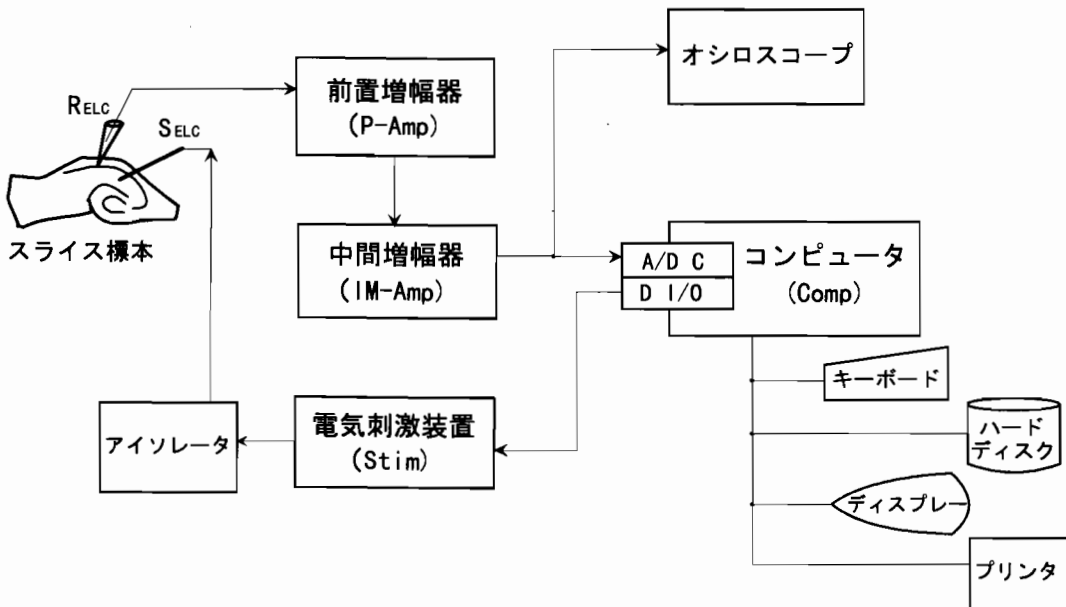


図1. ブロックダイアグラム

矢印が、信号の流れを示す。A/D C は A/D コンバータ、D I/O はデジタル入出力カード。
 $RELC$ は記録用ガラス電極、 $SELC$ は刺激用金属電極。

の大きさで前置増幅器から出力される。ところが、コンピュータで読み込める最大電圧は 5 V なので、信号の最大値が 5 V になるように増幅する必要がある。これが中間増幅器で、250 倍から 1000 倍までステップ的に可変できる増幅機能を備えていれば良い。

2.1 アナログ・デジタル(A/D)コンバータ

中間増幅器からの信号をコンピュータに取り込むには、アナログ信号をデジタル信号に変換する必要がある。このデータの変換装置が A/D コンバータ (Analog to Digital Converter, A/D C) である。この A/D コンバータを選択する場合、以下のいくつかの条件を満たしている必要がある。一つは入力電圧から計算される最小量子化幅とダイナミックレンジで、もう一つは入力信号に含まれる最大周波数をもらさず記録するためのサンプリング時間である。

入力信号として雑音と分離したい最小の電位 (分解能) を V_{min} 、最大反応の電位を V_{max} とするとダイナミックレンジ (D) は、 $D = \log_2$

(V_{max} / V_{min}) で表される。実際のニューロンのフィールドポテンシャルの記録から V_{min} は $10 \mu V$ 、 V_{max} は 20 mV 程度なので、ダイナミックレンジ: $D = 11.0$ となり、12ビットの A/D コンバータが必要となる。この時の量子化による誤差 (e) は、 $0 \leq e \leq V_{min}$ の条件が成り立つので、 $10 \mu V$ 以下となる。

次に、横軸の量子化を行うサンプリング時間を求める場合は、元の波形情報が失われないようにするためにサンプリング定理を適用してサンプリング時間を決定する。時系列データのサンプリング定理によれば『波形データの含む最大周波数を f_c (Hz) とすると、サンプリング時間 Δt が $\Delta t \leq 1/(2 \cdot f_c)$ の条件を満たせば、波形情報を全て得ることができる。』従って、フィールドポテンシャルの最大周波数は 10 KHz 程度なので、サンプリング時間は $1/(2 \cdot 10 \text{ KHz}) = 50 \mu sec$ 以下であれば良いことになる。

さらに A/D コンバータを選択する場合、これら以外の検討すべき機能としてはサンプル・アンド・ホールド機能である。これは複数のチャンネルから信号を同時に記録する場合、A/D

ボードには A/D コンバータの IC チップが一個しか装備されていないので、マルチプレクササーでチャンネルを切り替えてデータの変換を行なう必要がある。そのため、同時サンプリングしようとしても、物理的制約として1チャンネルの A/D 変換時間とマルチプレクササーのチャンネルの切り替え時間分だけチャンネル間で遅れが生じてしまう。これを解決してくれるのがサンプル・アンド・ホールド機能で、A/D 変換の開始信号を受け取ると全チャンネル同時にアナログ信号を読み取り、それを A/D 変換されるまでその電圧を保持する回路である。

2.2 デジタル入出力(D I/O)

D I/O カードは、コンピュータと外部機器を接続するもう一つのインターフェースカードである。これは、プログラムによってコンピュータ内部で発生した信号で電気刺激装置を駆動し、ニューロンに電気刺激を加えるものである。D I/O カードにはたくさんの入出力用のチャンネルがあるが、刺激装置をトリガーするには1チャンネルの出力信号があれば良い。出力信号のレベルには TTL (Transister-Transister-Logic) レベル、リレー出力、オープンコレクタレベルなどいくつか用意されているので、刺激装置の外部トリガー入力の場合にあったものを選ばなければならない。一般的には TTL レベル(0~5V)の入出力端子をもった機器が多い。デジタル出力信号の遅延時間は、プログラムの実行から実際にコンピュータから信号が出力されるまで約 1~2 μ sec と極めて短いので、システム設計上はほとんど問題にはならない。

2.3 コンピュータ

コンピュータの選択に関してはいろいろ議論もあるが、安価でコンピュータ内部が公開されており、かつその情報に関する書籍もたくさん出版されているという点から DOS/V 機を選択した。コンピュータの速度は、インテル社の CPU チップで 80486, 33 MHz 相当以上の能力のあるコンピュータであれば、この計測システ

ムには十分である。メモリー容量は、走らせるオペレーティング・システム(OS)と接続する周辺機器によるが、MS-DOS であれば 16 MB 以上、Windows 95 なら 24 MB 以上であれば良い。

3. ソフトウェアの構成

ハードウェアの準備ができれば、次はソフトウェアの構築である。プログラミングを開始する前にどんなオペレーティング・システム(OS)上で計測プログラムを開発するかを決めなければならない。OS を選択する際、A/D および D I/O ボードのメーカーがこれらの OS に対応したドライバソフトウェアを供給しているかが重要なポイントである。何故ならドライバソフトウェアとは、実際のボードと OS にあってデータをやり取りするソフトウェアで、もしメーカーから提供されていなければアセンブリ言語を使ってドライバに相当するプログラムを書くことになるが、これはプログラムの専門家でなければかなり困難であるからである。OS の選択の要点はこの他にもいくつかあるが、例えば、記録したデータの2次処理にどのようなアプリケーション・ソフトウェアを用いるか、どんなプログラミング言語を使って計測プログラムを開発するか、などである。

一般的にオペレーティング・システムにはシングル・タスク形式とマルチ・タスク形式の2つがある。前者の代表例として DOS (Disk Operating System) があり、後者の例として Windows 3.1 や Windows 95 が挙げられる。前者のシングルタスク形式は、同時に実行できるプログラムは一つだけであるのに対し、マルチタスクは多数のプログラムを同時に実行できる環境をもつ OS である。従って、シングルタスク形式の OS のほうがシステムとしては単純であり、マルチタスク形式のように計測プログラムが他のアプリケーションと競合することがないので計測プログラムの動作が安定している。しかしながら、DOS は時代の流れからすると徐々に過去の遺物となりつつあることは否

めない。しかしながら、データ計測という性質から考えて動作の安定性が最も重要であり、我々は、この計測システムの OS としてシングルタスク形式の DOS を選んだ。

次に、プログラミング言語の選択である。プログラミング言語にはたくさんの種類があるが、その中でも FORTRAN は、技術計測用プログラミング言語として過去何年にもわたってその代表的存在であった。最近では構造化言語という新しいプログラミング言語がいろいろなアプリケーションの開発に用いられるようになってきた。この言語はプログラムの作成と保守が容易なことから、プログラムの生産性を格段に向上させた。現在は、OS をはじめとし多くのアプリケーションは、この構造化プログラミング言語によって開発されている。それは、構造化プログラミング言語の処理の流れが『連結、反復、選択』の3つの基本構造のみで表現される単純な技法だからである。この構造化プログラミング言語として、ALGOL, PASCAL, C, QUICK-BASIC などがある。QUICK-BASIC は、初級者向けに開発された BASIC の手軽さ(インタープリタ方式)と高級水準言語である構造化プログラミング言語の2つの特長を併せ持った言語で、プログラムの経験のないパソコンユーザでも比較的抵抗なくプログラムを作成することができる。従って、今回の計測プログラムの開発には QUICK BASIC 言語を用いることにした。

ここまでに選択したプログラム開発環境をハード、ソフトの両面からまとめると

- インテル社の80486, 33 MHz 相当以上の CPU チップを載せた DOS/V 機,
- メモリー容量は 16 MB 以上,
- 12ビットの分解能と最小サンプリング時間が $50 \mu\text{sec}$ 以下の A/D コンバータ,
- TTL レベルで 1 ビット以上の出力をもつ D I/O ボード,
- インターフェースのドライバーが用意されているシングルタスク形式の DOS,
- 構造化プログラミング言語の仕様を備えた

QUICK-BASIC 言語、となる。

以上の開発環境で作成するプログラムは、プログラムを実行するメモリー領域の物理的制限から、フィールド・ポテンシャルの読み込み、保存用の計測プログラムとオフライン解析プログラムの2本立てで構成される。

3.1 計測プログラム

このプログラムでは、予め設定された時間間隔で刺激装置をトリガーしてニューロンに刺激を与える。それと同時にガラス電極で記録される電位を A/D コンバータで読み込み、その波形を画面に表示する。計測モードになっていれば波高値ならびに波形の傾きを自動的に計測し、画面にプロットする。さらに、波形保存モード(ストアモード)になっていれば、ファイル名を日付とスライス番号とセル番号から自動的に構築して、そのファイル名で波形をハードディスク上に保存する。計測プログラムの操作は、キーボードからのコマンド入力によって操作する。その際、操作に必要なコマンドはディスプレイ画面にメニュー形式で表示する。さらに、画面には計測した波形とその波形の計測値を常時モニターできるように表示する。

以上の条件に基づいて設計した画面が図2である。プログラムのフローチャート設計と画面設計が、システムを構築する際の作業工程の約1/3に相当する。さらに1/3はプログラムをフローチャートに基づいてプログラミング言語の文法に従ってコンピュータに入力してゆくコーディング(Coding)の作業で、残り1/3は通常“虫取り”と呼ばれるプログラムの修正作業であるデバッグ(Debugging)である。

図2のAにはフィールドポテンシャルの波形を読み込む毎に表示し、常に新しい波形に更新してゆく。縦軸、横軸の値は軸範囲変更コマンドで変更する。横軸の変更は、サンプリング時間とサンプリング数によって行なう。通常、波形は2チャンネルで記録するので、もう一つの波形を表示させるにはコマンドで波形窓を切り替

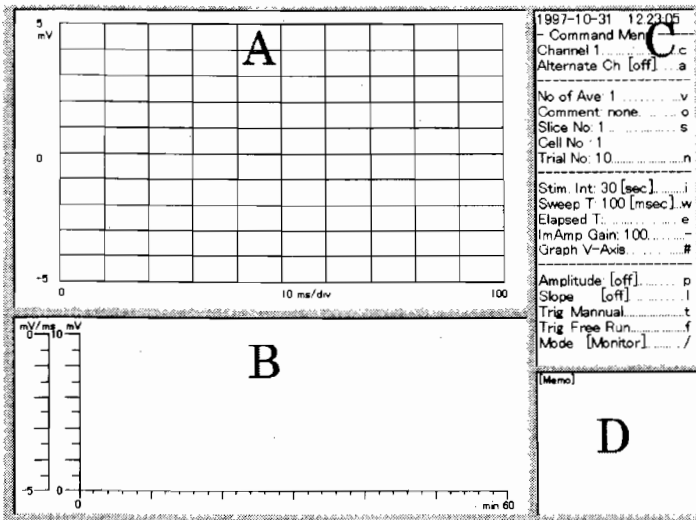


図2. 画面構成

A: 波形表示 B: 波形計測値 C: コマンドメニュー D: メモボード

えて行なう。また、ある部分の波形を詳細に観察するために時間軸方向の拡大ができる。

図2のBは、波形の波高値と傾きの時間経過を観察するためのデータをプロットした散布図である。この散布図の軸の範囲もコマンドによって変えることができる。また、波形記録中に薬物の灌流などを行なったときのイベントをマークするために、ラインをグラフ上に引くことができる。

コマンドメニュー画面(図2C)で、カーソルキーあるいはショートカットキーでコマンドの選択を行なう。最上段には日時と時間が1秒毎に表示されるので、実験中のプロトコルを記録するときの役に立つ。3, 4段目はチャンネルの切り替えに関するコマンドである。5段目は、S/N比を稼ぐために行なう加算の回数の設定で、波形は加算されたものが表示され、ファイルにも加算された波形が保存される。6~8段目は波形と一緒に記録しておく簡単なメモ設定で20文字以内ならば自由に書くことができる。次の Trial No は、刺激回数を示し、この実験を通して何回刺激を加えたかを示している。10段目は刺激間隔時間の設定で、何秒毎に刺激を繰り返すかを設定するためのコマンドである。11~14段目までが波形とプロット画面の縦・横

軸の範囲の変更コマンドである。次の2つは計測モードの設定コマンドで、これが設定されていなければ波形を読み込んでも波高値等の測定は行なわない。『Trig. Manual』が単発波形の読み込みコマンドで、『Trig. Free Run』は一定の時間間隔で連続的に波形を読み込むコマンドである。最後の段は、波形をファイルとして保存するかどうかを切り替えるコマンドで、『ストアモード』と『モニターモード』のトグル・スイッチになっている。メニューには表示されていないが、この他にも隠れコマンドがある。図2Dは、メモボードで設定値の入力やファイル名の表示などに用いられる。

波形および波高値と傾きのデータをファイルに記録する際、そのファイルの構成についても設計しておく必要がある。波形に関しては、記録したときの日時・時間、サンプリング条件、コメントなどのメモ、中間アンプのゲイン、波形を画面上に表示するためのスケールファクターなどその時の波形記録に関する情報をひとまとめにしてファイルヘッダーとして定義し、各波形ファイルの先頭に波形データと一緒に記録する。これは、オフラインで波形を解析する際に必要な情報である。プロットデータのファイルについても同様にファイルヘッダーを

駆動し、すぐに A/D ボードの 1 チャンネルから電位を読み取り、次に 2 チャンネルから電位を読み取る。この交互の処理をサンプリング数だけ繰り返し、A/D コンバータからの読み込みを終了後に波形を画面上に描き出す。計測モードが設定されていれば波形を計測し、ストアモードが設定されていればファイル名を自動的に発生し、ヘッダーと共にハードディスク上に書き込む。連続波形の読み込みコマンド『f』がタイプインされると、次に示すタイマーイベントのステートメントが実行される。

```

...
TIMER ON
ON TIMER (StimInterval) GOSUB ADConversion
...
    
```

このステートメントは StimInterval によって設定された時間になると、現在行なっている処理を一時中断して『ADConversion』というプロセッサにコントロールを移し、データの読み込みを優先的に行ない、この処理が終了すると再び中断していた処理を継続する。A/D コンバータから読み込むためのステートメント

は：

```

Dim Wave% (8000)
...
...
err% = Lab. ISCAN. Op (board%, Num.
    Channel%, LabPCGain%, Wave% ( ), 2*
    SampleNo%, Sample. Rate#, 0, Final.
    Order% ( ))
    
```

(ここで使用しているパラメータの意味は下記のとおりに)

- board% : A/D ボードを複数使用しているときのボードの識別のための番号,
- Num. Channel% : 16のチャンネルのうち使用するチャンネル数,
- LabPCGain% : A/D ボードのもっている増幅機能のゲインの指定,
- Wave% () : 波形データを格納するメモリー領域,
- SampleNo% : サンプルする点数,
- Sample. Rate# : A/D 変換時間,
- Final. Order% () : 16チャンネル中のサンプリングして行く順序.)

である。Lab. ISCAN の実行形式のオブジェクト

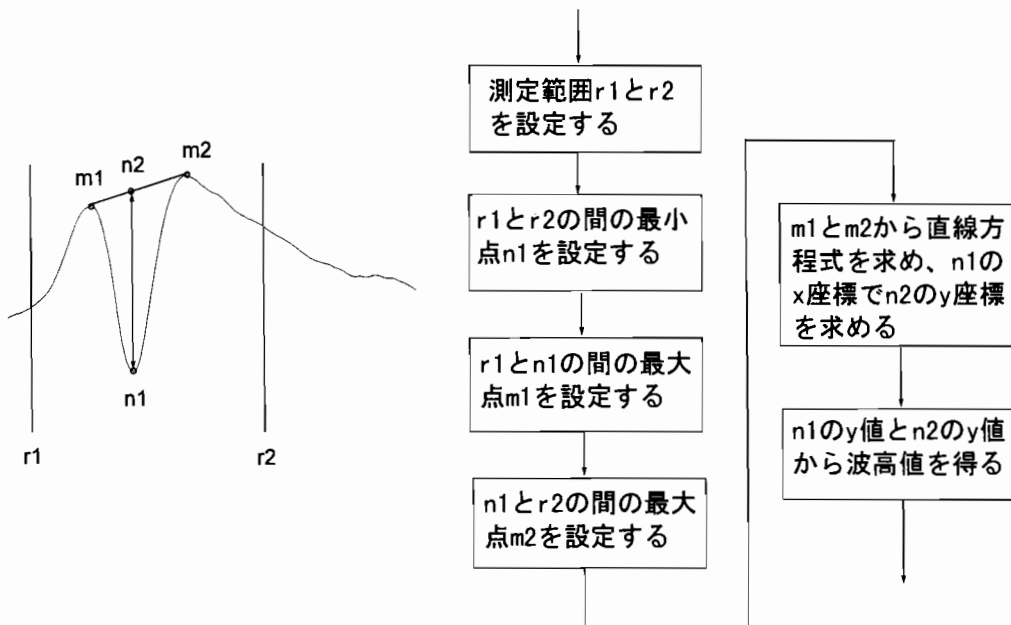


図 4. 波高値を自動計測するアルゴリズム

トは、A/D ボードのメーカーがサポートするライブラリーに登録されているので、計測プログラムをコンパイルして実行形式のプログラムを作成する際、サポート・ライブラリーをリンクする必要がある。これが、このプログラムの中枢である。

その他の機能として、チャンネルの切り替え、コメント・スライス番号・セル番号の入力、軸の変更、測定モードの設定、波形保存モードの設定、波形の時間軸拡大表示、イベント・ラインの挿入、刺激間隔時間の変更、終了時の処理などがある。これら機能の中でこのプログラムために開発したアルゴリズムは、波高値の自動計測である。その実際の計測概念とフローチャートを図4に示す。そして、その実際のコードは、

```
...
Call SetRange (r1, r2)
Call Min (r1, r2, n1x, n1y)
Call Max (r1, n1x, m1x, m1y)
Call Max (n1, r2, m2x, m2y)
Slope = (m2y - m1y) / (m2x - m1x)
Offset = m1y - Slope * m1x
n2y = Slope * m1x + Offset
Amplitude = n2y - n1y
...
```

となり、設定した範囲内で波高値 (Amplitude) を計算することができる。『SetRange』は計測範囲の x 座標を決定するプロセッジャー、『Min』と『Max』は指定された範囲内の最大/最小値の x, y 座標を与えるプロセッジャーである。

3. 1. 3 終了セクション

プログラムを終了するときには、計測データの確保と使用したメモリーの解放とインターフェースカードの設定解除を行わなければならない。実験中、波形を記録しながらその波高値と傾きを計測しているので、必要ならばそのデータをファイルとしてハードディスクに保存する。波形データの一時的格納領域として確保したメモリーエリアを、プログラム終了時には開放しなければならない。また、A/D コンバー

タと D I/O ボードは次の使用のために設定値を解除する。

この計測プログラムは、ひとつのメイン・プロセッジャーと62個のサブ・プロセッジャーから構成されている。ステートメント数にすると約2,000行になり、ひとつのプログラムとして扱えば、プログラムの修正段階 (Debugging Process) でかなり時間がかかると予想される。しかしながら、今回のプログラミングでは機能単位にプロセッジャーを設定したので、ひとつのプロセッジャーが約30行程度の短いサブ・プロセッジャーとなり、デバッキングの作業も効率よく行なえた。さらに、QUICK BASIC 言語では各プロセッジャーの変数の独立性が保たれているので、偶然に同一の変数名を設定してしまうなどの変数の設定によるプロセッジャー間で発生するトラブルを最小限にすることができるとも、効率の良いプログラム作成に寄与していると思われる。

3. 2 解析プログラム

解析プログラムでは、ファイルに保存してある波形/グラフを読み出し、印刷したりあるいは波形の比較などを行なう。研究者がどのように解析を進めるかによってまったく異なり、ほとんどが研究者に依存するプログラムである。その意味でこの部分を計測プログラムのように詳しく述べることはせず、参考程度に簡単に紹介する。

図5は我々が使用している解析プログラムのフローチャートである。ハードディスクに保存してある波形ファイルを読み出し、その波形を詳細に検討するために波形を拡大したり、縮小して画面上で観察するための機能や、ノイズ成分が大きく S/N 比が悪いときにはいくつかの波形を加算したり、あるいは2つの波形の差分をとって変化分を観察したりする機能が盛り込まれている。そして、それらの波形をプリンターに打ち出して紙面上で2次的な解析を行なう。また、オンライン中に測定した波高値/傾きの再評価が必要と思われる波形について再度測定

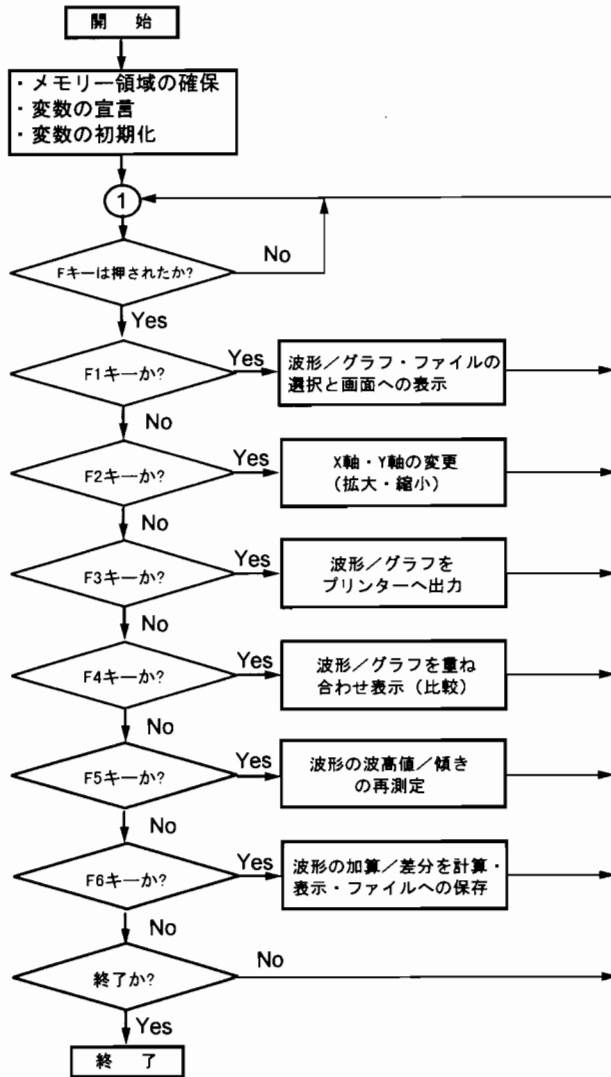


図5. 波形解析のフローチャート

し直す。また、フローチャートには載せていないが、他のアプリケーションでスライド作成などのためにファイルのデータ変換(バイナリデータからアスキーデータへの変換)を行なう機能や Paired Pulse Facilitation を評価するプログラムといった極めて特殊な解析などもこれには含まれている。これらの解析プログラムは、その時々プロジェクトによって必要とされる新たな解析方法を加えてゆくのでかなり流動的である。

以上が我々の研究室で実際に使用している計測・解析プログラムの概要である。しかしながら、コンピュータ/インターフェースカードが日々進歩するのにあわせて、このプログラムも効率の良いものに更新されている。従って、実際のコードはすぐに陳腐化してしまうので本稿ではその考え方だけを記述した。

参 考 書

- 1) 科学計測のための波形データ処理(計測システム

- におけるマイコン(パソコン活用技術), 南 茂
夫編者, CQ 出版社, 1986年.
- 2) 実験生物学講座 5 電氣的測定法, 平本幸男, 竹
中敏文編者, 丸善, 1982年.
- 3) Microsoft Quick BASIC Programming Guide, マ
イクロソフト社, 1990年.