

はじめてのデータ収集

Introduction to Data Acquisition

(Received 25 November 2004)

1. はじめに

近年、パーソナル・コンピュータ (PC) の大幅な性能向上と低価格化によって、研究室の小規模プラズマ実験から大型プロジェクトに至るまで、デジタル化された実験データを PC で収集・解析するようになってきました。またデジタル計測器の普及と高性能化にも目を見張るものがあります。アナログ計測器の代表格だったチャート紙出力ペンレコーダも、今や「古くなったノートパソコンを PC レコーダにしてペーパーレスをめざす」システムに取って代わられようとしています。

今から 8 年前になりますが、「プラズマ実験におけるデータ処理」と題した講座が、本誌 72 巻 10 号～73 巻 4 号に全 85 ページにわたって連載されました。データ収集システムの基本要素とデータ利用技術、それに大・中規模核融合実験のシステム事例紹介に重点がおかれています。しかし、デジタルおよびコンピュータ関連分野ではその後も進化・変遷が続いており、新技術の登場によって今では時代遅れになってしまった解説もあります。

その一例がブロードバンド・ネットワークの登場とその応用でしょう。ギガビット・イーサネット (GbE) は PC 普及機にも搭載されるほど一般化しており、GbE の広帯域性を活かしたネットワーク接続ストレージ (NAS) も急速に利用が伸びています。大容量データ保存装置の接続形態を、コンピュータ直接接続 (DAS) から NAS へと変えてしまうほど、ブロードバンドの登場は劇的な影響をコンピュータ環境に与えています。

しかし、大学院初学年の学生さんが初めて実験に取り組むにあたって、データ処理システムの取り扱いが必修事項なのは、今も昔も同じです。本テーマへの関心の高さは 8 年前と変わらず、最新情報をふたたび提供する意義は大きいといえるでしょう。本講座では、実験データをどのようにデジタル集録し、管理・利用すればよいかについて、ハードウェアの選定から論文図面の作成まで、順序よく集中講義形式で解説していきます。対象とする読者は、核融合実験を始めたばかりの大学院学生や、これからデータ処理システムを構築・改造しようとしている入門者の皆さんです。

なお、核融合実験のデータ処理分野で重要課題となってきた

1. 定常実験での実時間データ収集と処理
2. 遠隔実験参加技術の確立
3. 巨大化するデータの管理・取扱い技術

等の話題については、今回は講座としての位置づけや紙面の制約もあり、関連する章・節で簡単に触れて参考論文を挙げる程度にとどめることになりました。本誌小特集など次の機会をご期待ください。

この講座は、中・小規模の実験装置でデータ処理システムを構築・利用するための導入編です。より理解を深めるための関連テキストや資料類は、各節の参考文献に掲示するほか、学会事務局のご協力により、学会 Web ページ (<http://www.jspf.or.jp/journal/>) から積極的に情報提供していく予定です。そちらも是非ご活用ください。

以降の連載予定と執筆者は以下のとおりです。

第 2 章 システムを選ぼう (2 月号)

- 2.1 計測したい信号の種類を分析する
板倉昭慶 (筑波大学)
- 2.2 計測信号に見合ったデジタイザを準備する
板倉昭慶 (筑波大学)
- 2.3 計測器とデジタイザの制御用コンピュータを導入する
檜垣浩之 (筑波大学)

第 3 章 データを集めよう (2 月号)

- 3.1 計測器とデジタイザを運転する
中西秀哉 (核融合科学研究所)
- 3.2 データを生成・転送する
中西秀哉 (核融合科学研究所)
- 3.3 データを保存し、共有する
奥村晴彦 (三重大学)

第 4 章 データを解析しよう (3 月号)

- 4.1 データを表示する
後藤基志 (核融合科学研究所)
- 4.2 データを解析する
大館 暁 (核融合科学研究所)

(中西秀哉)



2. システムを選ぼう

板倉昭慶, 檜垣浩之
(筑波大学プラズマ研究センター)

Let's Design and Construct a Proper Measurement System

ITAKURA Akiyosi and HIGAKI Hiroyuki

Plasma Research Center, University of Tsukuba, Tsukuba 305-8577, Japan

(Received 16 December 2004)

A method of design and construction of a measurement system is discussed. Characteristics of a signal, i.e., frequency, amplitude and so on, must be analyzed. An analog signal is digitized by an analog to digital converter. Sampling frequency must be higher than two times of the signal frequency. Control of the digitizer by a computer through several kinds of interfaces is also discussed.

Keywords:

analog signal, digitizer, ADC, sampling, clock, multiplexer, CCD, RS-232-C, GP-IB, USB, Ethernet, PCI

現在のプラズマ実験では前章の「はじめに」で述べられているように、データをデジタル的に処理することが一般的になっています。そこで、データ収集システムを構築することは信号に合わせたアナログデジタル変換器、すなわち、デジタイザを選択することが中心となります。デジタル信号の処理に関する参考書は多数あり、枚挙に暇がありませんが、さて、プラズマ実験のためにデータ収集システムを構築しようと考えたと意外とありません。既に構築されている大型装置の仕様書を入手するか、以前本誌に連載された講座[1]や[2]程度です。

ここではシステムの設計に関してメーカーの助けを借りずに進められる程度の小規模なシステムについて考えます。最近のデジタルオシロスコープはネットワークインターフェースを持っているばかりでなく、Windows OSで動作するパーソナルコンピュータを組み込んだ物が増えていて、これが最小構成と言えるでしょう。

2.1 計測したい信号の種類を分析する

2.1.1 データ収集システム

システムを設計するためにはどのような信号を処理するのかを考える必要がありますが、その前に、データ収集システムはどのように構成されているのでしょうか。

Fig. 1 にその概要を示します。プラズマに取り付けられた計測器から得られた信号は前処理回路で不要成分が除去されたり、次のデジタイザへの入力に適した信号となるように振幅などが調整されます。次にデジタイザでデジタル信号化されて制御用計算機を通してハードディスクなどの保存装置で保存されます。一般の使用者はこのデータを取り出して必要な処理を行い、表示させます。これらの計算

author's e-mail: itakura@prc.tsukuba.ac.jp, higaki@prc.tsukuba.ac.jp

機類、データ保存装置は現在ではローカルエリアネットワークを通して相互に接続されているのが一般的です。また、制御計算機のネットワーク部分はルータを通して外部のネットワークから遮断するなど制御計算機に外部から不用意なアクセスをされないように安全を確保する必要があります。デジタイザは制御用計算機により制御されますが、プラズマと同期させるため、あるいは必要な時刻にデータを収集するためにトリガ信号を入力する必要があります。そのために遅延パルス発生器などのトリガ信号発生器を使用します。制御用計算機からソフトウェア的にトリガ信号を出力する事も、デジタルオシロスコープなどではプラズマからの信号自身で内部トリガを掛けることもあります。デジタイザのクロック発生器は一般的には内蔵されていますが、適当な値のクロックがない場合、他のデジタイザと正確な同期を取る必要がある場合には外部のクロック発生器を用いたり、主となるデジタイザのクロックを取り出して分配することもあります。デジタイザについては

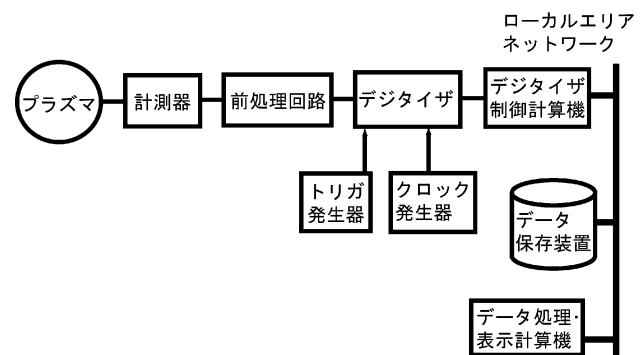


Fig. 1 データ収集装置の代表的な構成の例

2.2節で考えます。

2.1.2 信号の種類分析

それではデータ処理装置ではどのような信号を処理しているのでしょうか。プラズマ実験で得られる信号としては実験の種類に応じて次のような信号が考えられます。

- ・定常的なプラズマ生成による目に見えるようなゆっくりした変化で長時間連続する信号
- ・周期的な繰り返し放電による信号
- ・大型装置に多い周期的な繰り返しであっても夫々のパルスが独立と考えられる放電の信号
- ・実験装置の監視装置のようにプラズマ生成とは別個のモニタ信号

また、信号の形としては次のようなものが考えられます。

- ・静電探針等の微小電流
- ・これの掃引電圧等の比較的大きな電圧信号
- ・磁気探針やログスキーコイル等の微分された電圧
- ・パルスハイトアナライザのように既にデジタル化された信号
- ・CCDカメラ出力のような画像信号

計測器で測定される信号は、電圧、電流、電気量、時間、周波数、回転数等様々なものがありますが、いずれも何らかの方法で電圧、電流、パルス回数等に交換されます。さらに、電流は抵抗を通してその両端の電圧を測ったり、演算増幅器による電流—電圧変換器を通したりして電圧に変換されます。電気量もコンデンサで積分され、いずれにせよ電圧に変換されます。時間は一定のクロックパルスの数を計数してデジタル的に出力する他に一定電圧を積分して電圧に変換する時間—電圧変換器も存在します。周波数は周波数カウンタにより単純な2進数またはBCDコード(2進化10進コード, 1桁ごとに4ビットで表す)等でデジタル的に出力するのが一般的です。計算機で取り扱うためには信号をデジタル化する必要がありますが、これらのように既にデジタル化されている信号は計算機のインターフェースにデジタル入力モジュール又はデジタル入力ボードを挿入して簡単に処理できます。また、ビデオ信号も現在は一般的な計算機インターフェース用のビデオキャプチャボードが多数市販されているので容易に処理できます。ここではこれからデジタル化する必要のある信号、すなわち、アナログ信号について考えましょう。

アナログ信号の性質を決めるものには次の量があります。

- (1) 周波数成分、正弦波ならば簡単ですが、一般的には様々な周波数成分を含んでいて、正弦波から離れた波形となっています。フーリエ変換すると周波数成分がわかります。
- (2) 振幅
- (3) 信号の継続する時間的長さ、ここでは計測器から信号を取り出している長さ、いわばスイッチを入れてから切るまでの時間を指します。その信号の寿命と考えるとコヒーレンスの問題となり、(1)の周波数スペクトルの幅と反比例関係となります。
- (4) 信号対雑音比
雑音とは一般的には周波数帯域が広く、白色雑音等相関

のないものを指しますが、プラズマ実験の場合はコヒーレントな信号でも自分の目的に合わない不要な信号は雑音と考えます。この場合、信号と雑音とは本質的には同じ性質を持っており、単に研究者が必要とするか、しないかの違いだけです。したがって、両者を識別するのは困難な場合があります。

この他に考慮すべき事柄としては信号全体としての対地電圧があります。信号処理回路の入力対地耐電圧よりも高い場合には安全のために絶縁する必要があります。そのためには絶縁変圧器(パルストランスもこの一種と考えて構いません)や光結合器を用いたり、光ケーブルで配線したりします。

また、特にパルス実験の場合には雑音の除去、装置の安全のためにグラウンドループを切って一点接地にする必要があります。信号源のインピーダンス、同軸線で接続するような非対称信号で、片側が接地されているのか、対撚り線を使用するような対称信号で、接地から浮いているのかも考慮する必要があります。これらについては第3章で触れますが、文献[3]にも詳述されています。

2.2 計測に見合ったデジタイザを準備する

2.2.1 サンプリング

先に述べたように、計算機で処理する信号はデジタル信号である必要があります。そこで、アナログ信号を処理する為にはそれをデジタル信号に変換する、すなわち、デジタイズする必要があります。ここではアナログ信号を処理する事を中心にデジタイザについて考えます。

Fig.2に示すように計測されたアナログ入力信号(a)を一定の周期のクロック信号(b)を用いて切り出すことを考えます。クロックパルスと重なった時の信号を取り出す(サンプルする)と(c)のような波形が得られます。そしてその電圧をビット数で決まる数、すなわち分解能まで分解して、そのどこに当るか判定するのです。この判定には意外と時間を要するのです、と言っても勿論 μ 秒やそれ以下の時間です。そこでいろいろと工夫され、いくつかの方式が実用化されています。しかし、どのデジタイザがどの方式を採用しているかは一般的なカタログには記載されておらず、我々としては特に最高の性能例えば特別に高速なもの、分解能の高い物を要求しない限り方式の選択の余地はほとんどないのでここでは触れません。特に興味のある方は文献を参照してください[4,5]。

デジタイザで重要なことは入力信号波形を如何に正しく

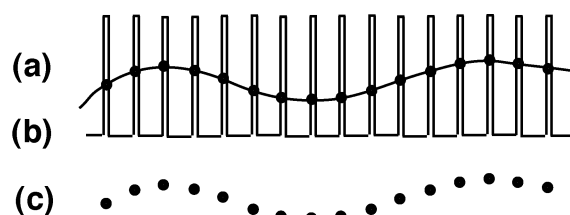


Fig.2 サンプリングの様子。(a) アナログ入力信号、(b) クロックパルス、(c) サンプルされた出力

再現できるかにあります。デジタルオシロスコープを使えばわかるように、クロック周波数の選択を誤ると正しく波形を再現出来ないことがあります。最近のオシロスコープはそのことを考慮して、前処理回路に適切な周波数特性のフィルタが設定されて、問題の起きにくくなっているものもあります。クロック周波数の選択はナイキストのサンプリング（標本化）定理として示されています。つまり、信号に含まれている周波数成分の2倍以上の周波数でサンプルする必要があるということです。もし、信号が正弦波ならば丁度2倍の周波数で半周期に1点ずつ取り出せることとなり、正しく再現されます。しかし、正弦波でなければ、もっと高い周波数成分を含んでいる事になりますから、もっと高いサンプリング周波数を必要とします。もし、信号周波数の2倍よりもサンプリング周波数が低いと、信号から取り出す点の数が信号の周期よりも不足し、少しずつ位相のずれた点をサンプルすることになり、周波数が低く見えます。この様子を Fig. 3 に示します。この現象をエイリアシング (aliasing) と言います。丁度、電燈光の下で扇風機を入り切りして回転数が変わるところを見ていると、速度により回転が止まっているように見えたり、逆回転して見えたりするストロボ効果と同じ現象です。電燈光の下で見る事は 100 Hz (東日本) または 120 Hz (西日本) でサンプルすることに相当します。この様子をコンピュータ上で体験できる WEB サイトもあります(例えば <http://mars.elcom.nitech.ac.jp/java-cai/signal/sampling2.html>)。サンプリング定理についての厳密な取り扱い文献[6]を参照してください。

2.2.2 デジタイザ

簡単化したデジタイザの構成を Fig. 4 に示します。前処理回路は先に述べたサンプリング周波数に比べて高すぎる不要周波数成分を除去したり、信号電圧を適当な値に調整します。増幅度はジャンパーピンなどでハードウェア的に変更したり、ソフトウェア的に設定したり出来るものがあります。Fig. 2 に示すように、時間的に振幅が変化する場合、変換に要している時間（後述のアーチチュア時間）の間にも入力電圧が変化していて、正確に変換できないことがあります。そこで、変換に要している時間の間電圧を一定に保つのがサンプル・ホールド回路です。マルチプレクサは電子的な切替スイッチで、必ず含まれるわけではありませんが、複数の入力信号を一定周期で順番に切替えて、1個のアナログ-デジタル変換素子で変換する場合に用いられます。変換素子の動作する周波数が信号の周波数よりも十分に速い場合に採用される方式です。この方式ではひとつの入力チャンネルに対するサンプリング周波数は変換素子のサンプリング周波数に対してチャンネル数分の1になりますから、切り替えるチャンネル数に応じて最高サンプリング周波数が変わることになります。信号の入力数よりも変換素子の数を少なくできますから、変換素子が高価だった時代にはよく使われました。最近では少なくなってきました。マルチプレクサを使用している場合、各チャンネルを時間的に切替えていますから、全く同時刻にサンプルすることができません。これは位相や時間の同時

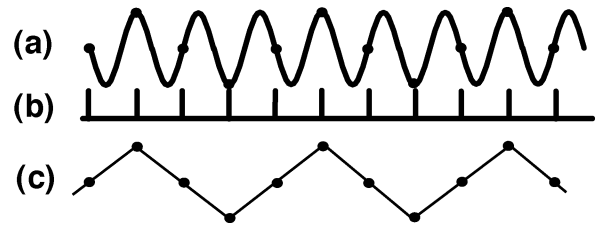


Fig. 3 エイリアシングの例。(a)アナログ入力信号, (b)クロックパルス, (c)サンプルされた出力。クロック周波数が信号周波数の2倍より低いと正しく再現されない。

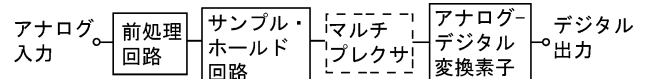


Fig. 4 デジタイザの構成例

性を見る時には致命的ですが、サンプル・ホールド回路により擬似的に同時刻にサンプルすることができます。厳密さを要求する時はマルチプレクサを使用せず、チャンネル毎にデジタイザ素子を持っている物を使用します。

上記のことを念頭において、デジタイザを選択するのに必要な項目を考えましょう。

- (1)サンプリング周波数（或いはクロック周波数、クロック時間）
- (2)出力ビット数
- (3)入力処理回路
 - 入力信号範囲
 - 入力インピーダンス
- (4)アーチチュア時間
- (5)メモリ量（接続時間=サンプリング時間×メモリ量）
- (6)データ転送所要時間
- (7)トリガモード
- (8)マルチプレクサの使用, 不使用, サンプル・ホールド回路等—チャンネル間同時性
- (9)その他（入力チャンネル数, トリガ, クロック等の入出力等）

以下これまでに触れていない事柄についてももう少し解説しましょう。

(1)サンプリング時間（クロック信号）

一般的にはハードウェア的、ソフトウェア的に周波数が何段階か変えられます。安価な製品には内部クロックは1種類だけというものもあります。また、ほとんどの機種には内部クロック周波数と同じ程度の範囲で外部からのクロック信号を入力できたり、逆に外部へクロック信号を取り出すこともできます。これは他のデジタイザと同期を取る時などに必要になり、外部クロックで一斉に制御したり、主となるデジタイザから他のデジタイザへ分配したりします。

(2)ビット数

縦軸の分解能はビット数で表されます。例えば12ビットのものでは、入力電圧、例えば+/-5Vがデジタル化されて4095に分割され、分解能は約2.4mVとなります。出力信号の値は単純なバイナリコード（オフセットバイナリ）や

2の補数がよく用いられています。中には電圧の大小と出力値の大小が逆になっているものもあり、データの処理には注意が必要です。数10kHz程度よりも低速では14ビット、16ビット等もっと高分解能の物、数10MHz程度よりも高速では8ビットのものが多いようですが、次第にビット数は大きくなっていきます。8ビットのデジタイザでは二個のデータをつないで16ビットとして出力するものがあります。この場合は計算機で処理する時にデータを分離する必要があります。気をつけなくてはならないのは入力電圧の範囲で、実際の信号電圧が小さすぎるとせっかく分解能の高い物を使用してもその一部だけを利用することとなり、実際の分解能は必ずしもよくなるのではありません。周波数特性とともにFig.4の前処理回路が重要になります。

(3)前処理回路

入力されたアナログ信号はまずデジタイザで処理しやすい振幅になるよう調整されます。前項で述べたように、良い結果を得るには適当な振幅にする必要があります。デジタイザによっては入力電圧範囲、両極性(例えば-5V~+5Vのように+/-に変化するもの)、単極性(例えば0~+10Vのように片方の符号の電圧のみ)が切替えられます。また、同軸ケーブルを使う非対称入力の物や、対振り線を使う対称入力の物が選べるようになっていきます。入力インピーダンスは現在ではほとんどの製品が100kΩ以上であり、オシロスコープ同様の注意が必要です。

(4)アパーチャ時間

ひとつのクロックで、デジタイザが開いている時間です。つまり、信号がその間に変化した場合にはその平均値を出力することとなります。この時間が長いと変換は容易になりますがその間の変化を平均することとなり、不正確となります。短いと変換そのものが不正確となります。適当な時間を選択する必要がありますが、なかなか自由には選べません。

(5)接続時間(収集時間)

クロック時間×メモリ容量で決まります。以前はメモリ素子が高価であったので、高速で長時間データを収集することは困難でしたが、現在は安価となったので、十分な時間が収集できるようになりました。

(6)データ転送速度(データ転送所要時間)

これには二つの意味があります。デジタイザ内の、データの処理および出力に要する時間とデジタイザから制御用計算機またはデータ保存用計算機に転送する時間です。

(7)トリガ

プラズマと同期してデータ収集を制御するためには何らかのトリガを入力する必要があります。データ転送やトリガについては第3章の「データを集めよう」のところで解説されています。

(8)リアルタイムデータ収集

従来のデジタイザはデータの処理および出力に時間がかかったため、アナログ信号をデジタイズすると同時に出力することができず、一旦メモリに貯めてから吐き出すトランジェントレコーダ型が主流でした。これでは定常プラ

ズマなど長時間続くプラズマではデータを収集している間はデータを見ることはできません。そこで、長時間継続する現象を観測するのにCAMACインターフェース(文献[1]参照)などでは2台のデジタイザを組み合わせ、データ収集と出力を交互に繰り返させるなどの工夫が必要でした。現在では収集して処理し、出力する速度が上がり、横河電機WE7000シリーズなどリアルタイムでデータ収集、観測のできるものも開発されました。

また、CAMACはデジタイザがモジュール化されていて、電源を兼ねたクレートと言う筐体に組込むことができ使いやすかったものの制御用のコンピュータを別に必要とし、高価で大型装置向きでした。現在ではデジタルオシロスコープ同様に直接ネットワークに接続できたり、コンピュータを組み込んだ製品も市販され、ある程度チャンネル数を必要とする場合には手軽に使えるようになりました。

この節の終わりにあたって、具体的にどのデジタイザを選ぶかは既存の機器をなるべく活用できるように考慮しながら、次節のインターフェースを参考にして、速度のバランスから選択することになります。ただ、計算機、その周辺機器の日進月歩振りは目覚しく、いつまでも既存の財産に囚われているとみすみす到達できる性能を逃してしまい、苦しいところです。(板倉昭慶)

2.3 計測器とデジタイザの制御用コンピュータを導入する

冒頭に説明があったように、本講座では安価で手軽に機器の制御を行うための道筋を初心者具体的に紹介することを目的としていますが、本節ではハードウェアを制御するコンピュータを導入するにあたって考慮すべき点についてお話しします。別の言い方をすると、どのような性能のパソコン(PC)を使ってどのような方法で計測器類(例えばオシロスコープ、CCDカメラ、前節にでてきたADC等)を制御するのかということになります。

前節に必要なスペックを持った機器を選定したわけですが、当然のことながらこれらの機器はPCと何らかの形で通信する機能を持っていないわけではなりません。安価なデジタルオシロスコープやCCDカメラを手に入れてもPCと接続できなければPC制御による自動データ収集はできません。そこでどのようにして機器とPCを接続するかを決める必要があります。具体的な通信方法の例としてはRS-232-C、GP-IB、USB、Ethernet(LAN)等が挙げられるほか、PCのPCIバス等に直接差すようなプラグインタイプの機器を使う可能性もあります。どのような通信方法を使うのが一番良いかは、何台の計測器類をどのような条件下で制御しどのくらいのデータ量をやりとりする必要があるのか等によります。そこで

- ・単にデータを自動的に保存するだけなのか? データ処理まで行うのか? データを転送する必要はないのか?
- ・どのくらいのデータ量をやりとりするのか?
- ・放射線等の理由により実験室から離れたところで機器を遠隔制御する必要があるのか?
- ・高電圧のかかるところにある計測器を制御する必要がある

あるのか？

・数多くの計測器を制御する必要があるのか？

といったことのうち重要度の高い要求により構築しようとする計測制御システムが決まります。

先に挙げたPCとの通信手段のうち、RS-232-C, GP-IB, CAMAC, VME等については9年前の講座[7]にも記述がありますし、3章でも最近のUSBやIEEE1394等の記述があるので詳細はそちらを参照していただきたいと思えます。いずれの通信手段をとるにしても、**制御したい機器がどのような通信機能を持っているかを確認し、PCにも同じ規格の通信機能があることを確認する必要があります。**

次には、PCのOSを決める必要があります。最近では計測器制御、データ収集用のソフトウェアとしてLabVIEW[8], MATLAB[9]等もかなり利用されています。それらを導入する場合、基本的にはWindows, Linux, Macintoshいずれでもかまわないということになります。もし、Visual C++, や Visual BASIC等を使いたいということであれば基本的にはWindowsを採用することになります。様々なメーカーの継続的広範なサポート、システム導入の容易さということを考慮すると現状ではWindows PCが有利かもしれません。

制御用PCのメモリサイズは、その制御用PCで何をするのかに強く依存します。計測データの自動収集だけでなくデータの転送または処理もさせる場合、どのようなプログラムでどのくらいのデータ量を処理するかによってPCのメモリ量を増やしておく必要があります。例えば、8MBのデータを使って周波数解析をする場合ですと、プログラムにもよりますがWindows PCの256MBのRAMでは不十分となる可能性があります。大量の画像データ等、処理するデータ量が数MBよりも大きい場合はPCのメモリを可能な限り大きめにしておく、もしくは計測器制御用(ローカルなデータ収集用)PCとデータ保存用サーバー、データ処理用PCは独立にした方が将来性があります。

PCと呼ぶと違和感がありますが、最近ではOSの搭載された制御用コンピュータ内蔵でLAN制御可能なCAMAC, PCI, CompactPCI, PXIコントローラ等も様々なモジュールとともに各社から供給されています。もっとも、このようなモジュール化された計測システムは比較的高価なものになるので、新規に導入する場合は費用対効果を考慮する必要があります。また、オシロスコープやスペクトラムアナライザ等、一部の高性能機器ではLANによる遠隔制御可能な機器が主流になりつつありますが、多くの場合、これらの機器のシステムはWindows OSで制御されており、PCとしての機能を備えています。

さらに、USB=GP-IB, USB=RS-232-C, Ethernet=RS-232-C等のアダプタも市販されているので古い機器を最近の性能の良いPCで制御することも可能です(もちろん通信速度は速くなりません)。

このような状況を踏まえた上で、具体的にいくつかの例を見てみることにします。

2.3.1 オシロスコープのGP-IB, LAN 制御

我々のグループでは数十kHz以下のプラズマ中の低周波

揺動を計測するために静電プローブからの信号を2台の4ch デジタルオシロスコープ(Yokogawa DL1540)に取り込み、これをPC98でGP-IB制御して自動データ収集をしていました。ここでは8ビットの計測データを1chあたり200 ks/sで40kサンプル(200ms相当)で取り込んでいるので1回のプラズマ放電での全データ量は1MB以下ですが、このPC98は計測データを保存するだけのもので、データ処理はこれとは別のPCで行っていました。最近ではパソコンの性能が年々良くなっているため、データ処理も同時に行いたいというデータ処理効率化の要望や将来的な拡張性も考慮してパソコンをPentium4, 2.0 GHz, メモリ512MB, 内蔵Hard Disk (HD) 40GBのものに置き換えそのPCIバスにGP-IBボード(National Instruments: Windows, Mac, Unix等に対応)をさしてWindows OS上のLabVIEWで制御することにしました(Fig. 5)。ここでは、既存のオシロスコープをGP-IB制御する必要と1MB程度のデータを処理する必要からシステムの概要が決まっています。オシロスコープとの通信はGP-IBですが、データ量が多くないのでPCとのデータ転送時間はほとんど気になりませんし、その後のデータ処理(FFT, peak detection, 位相差測定等の周波数解析)も上記のPCの性能で十分足りているのでまだ計測ch数を増やす余裕があります。

次に磁気プローブを用いた周波数の高いプラズマの揺動計測の例です。これまでは100 Ms/s, 8 MB/chのADC(NI PXI-5112)をPXIコントローラ(NI PXI-8175: Pentium3, 866 MHz, メモリ512MB, 内蔵HD14GB)で制御していた

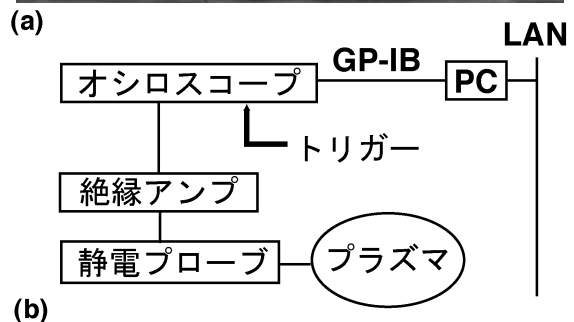


Fig. 5 (a)Windows PC(左)のPCIスロットにGP-IBボードをさしてGP-IBケーブルでオシロスコープ(右)と接続したところ。デスクトップPCでもPCIスロットがついているとは限りません。(b)Fig. 1と比較すると、この例では計測器—静電プローブ、前処理回路—絶縁アンプ、デジタルオシロスコープとなっています。

のですが、これだとFFTの結果50 MHzまでの揺動しか計測できません。より高い周波数の揺動を計測する必要性から、最近になって5 Gs/s, 12 MB/chのオシロスコープ(Le-Croy 6051)を導入しました。これは先に述べたWindows OSで制御されたPC内蔵(Celeron, 1.7 GHz, メモリ 512 MB, 内蔵HD40 GB)のデジタルオシロスコープなので実際にはオシロスコープと別にPCを用意する必要はありません。LANによる遠隔制御用のプログラムも供給されていますし、自動データ収集をしたければ、この機種の場合WindowsのActiveX対応のアプリケーション(Excel, Visual BASIC, MATLAB, LabVIEW, 等)によって独自の制御をすることが可能です。いまのところ、1回のプラズマ放電で約20 MBのデータを保存していますが、周波数解析まで含めてオシロスコープ内蔵のコンピュータで済ませています(少し重い)。放電間隔が短くなるとこれでは対応しきれなくなるので、新たにデータ解析用のPCを導入してデータを転送する等の工夫が必要になります。

2.3.2 CCDカメラの制御

次に画像入力の例としてCCDカメラの制御について見てみます。一口にCCDカメラといってもデジタル-アナログ、カラー-モノクロ、画像の伝送方式、等、様々な性能、規格のものがああります。ここで注意しなくてはいけないのは、そのカメラに対応した画像収集ボード(コントローラまたはフレームグラバ)を使う必要があるということです。特殊な用途を持った高性能のCCDカメラは多くの場合、専用のコントローラが供給されているので、ここでは汎用のアナログCCDについて見てみます。モノクロの場合RS-170とCCIRという信号方式が主流で、カラーの場合NTSCとPALが普及しており、いずれも前者の場合30フレーム/sec、後者の場合25フレーム/secが標準となっています。これらはすべてインターレース走査といい、1枚の画像を2回に分けて走査しています。これに対して、プログレッシブ走査は1枚の画像を1回で走査するので画素数が同じであれば、プログレッシブ走査の方が画像が鮮明になり、移動する物を撮像するのに適しているという特徴があります[10]。また、外部信号に同期して画像を取り込む必要がある場合、外部トリガー機能の有無、さらにシャッタースピードも確認しておく必要があります。Edmund等ではCCDカメラとコントローラをセットで供給しているので、特殊な用途でない場合はそのようなシステムを導入するのが簡単な方法だと思います。一例としては、38万画素のカラーCCD(HITACHI KP-D20:NTSC, インターレース走査)をWindows PCのPCIバスにさす画像収集ボード(National Instruments PCI-1411)を使って制御することが可能です。最近では数百万画素のCCDも普通に手に入りますが、この場合、モノクロ8ビットでもメモリ使用量としては数MBになりますし、カラーの場合、10MBを超えてしまいます。これを1回のプラズマ放電で何フレームも取り込むということになると先のオシロスコープの例と比べてもさらに多くのメモリが必要になることがわかります。多数の画像処理をする必要があるときはPCのメモリも1GBのオーダーで用意しておいた方がよいでしょう。

学会誌では複雑なシステムの開発例も紹介されていますので興味のある方は参照してみてください[11]。

もう一つ、画像制御の例として挙げられるのが監視カメラ(またはプラズマモニタ)としての用途です。市販のデジタルカメラからの出力(IEEE1394)をPCや専用のコントロールサーバに取り込んでEthernet経由で遠隔地に転送するといったことが可能です[12]。

以上、紙面の都合上非常に簡単で一面的ではありますが、PCを使った自動データ収集システムの例を紹介しました。(檜垣浩之)

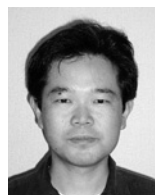
参考文献

- [1] T. Matsuda, *et al.*, J. Plasma Fusion Res. **72**, (1996) No.10 -**73**, (1997) No.4.
- [2] A. Itakura, *et al.*, J. Plasma Fusion Res. **74**, 616 (1998).
- [3] 伏見康治編: 実験物理学講座30 プラズマ・核融合(共立出版, 1979).
- [4] 相良岩男: A/D・D/A変換回路入門 第2版(日刊工業新聞社, 2003).
- [5] A-D変換の方式と設計技法 トランジスタ技術 2月号(CQ出版, 東京, 1996).
- [6] 磯部 孝編: 実験物理学講座3 情報処理技術(共立出版, 1985).
- [7] H. Nakanishi, M. Kojima and LABCOM Group, J. Plasma Fusion Res. **72** 1061 (1996).
- [8] <http://www.ni.com/labview/>
<http://www.ni.com/labview/ja/>
R. H. Bishop, *LabVIEW 6i Student Edition* (Prentice Hall, New Jersey, 2001).
- [9] <http://www.mathworks.com/products/matlab/>
<http://www.cybernet.co.jp/matlab/>
- [10] トランジスタ技術 2月号(CQ出版, 東京, 1999).
- [11] S. Kado, H. Nakanishi, K. Ida, M. Kojima, CHS group and LHD group, J. Plasma Fusion Res. **76** 1266 (2000).
- [12] トランジスタ技術 9月号(CQ出版, 東京, 2001).



いたくら あきよし
板倉 昭慶

筑波大学プラズマ研究センター勤務(物理学系)。東京教育大学理学部物理学科を卒業して移動縞、電子ビームとプラズマの相互作用による非線形現象を研究、理学博士。現在はマイクロ波によるプラズマ診断、電源の制御、データ収集を担当。趣味はデータばかりでなく様々なものを収集することと今年は実行出来なかった山歩き。



ひがき ひろゆき
檜垣 浩之

筑波大学大学院数理物質科学研究科講師。1998年京都大学大学院人間・環境学研究科博士課程修了。東京大学大学院総合文化研究科での日本学術振興会特別研究員を経て2001年より現職。電磁場中での荷電粒子の閉じ込め、ICRFによるプラズマの生成加熱、及びプラズマ中の揺動計測の実験研究を主に行っています。



3. データを集めよう

中西秀哉, 奥村晴彦¹⁾
(核融合科学研究所, ¹⁾三重大学教育学部)

Let's Acquire Data!

NAKANISHI Hideya and OKUMURA Haruhiko¹⁾

National Institute for Fusion Science, 322-6 Oroshi-cho, Toki 509-5292, Japan

¹⁾Mie University, 1515 Kamihama-cho, Tsu 514-8507, Japan

(Received 25 November 2004)

In fusion experiments, diagnostic control and logging devices are usually connected through the field bus, e.g. GP-IB. Internet technologies are often applied for their remote operation. All equipment and digitizers are driven by pre-programmed sequences, in which clocks and triggers give the essential timing for data acquisition. Data production rate and amount must be checked in comparison with the transfer and store rates. To store binary raw data safely, journaling file systems are preferably used with redundant disks (RAID) or mirroring mechanism, such as "rsync". A proper choice of the data compression method not only reduces the storage size but also improves the I/O throughputs. DBMS is even applicable to quick search or security around the table data.

Keywords:

data acquisition, sequence control, field bus, remote participation, clock, trigger, meta-data, block transfer, batch processing, journaling file-system, RAID, rsync, NAS, data compression, DBMS

3.1 計測器とデジタイザを運転する

第2章までのお話で、計測器から得られるアナログ信号をA-D変換するデジタイザ、それを運転・制御するコンピュータが一式そろいました。ここからは、そうした装置を実際に動作させ、本来の目的である計測データを集める手順へと話を進めていきます。

3.1.1 計測器とデジタイザの結線

計測データが流れる経路は、基本的に、

計測器

↓ … アナログ信号

デジタイザ (A-D変換)

↓ … デジタル信号

データ収集コンピュータ

となっています。しかし規模の小さいシステムでは、計測器とデジタイザが一体になったデジタル・ビデオ・カメラの利用や、A-D変換ボードをデータ収集PCの拡張バスに挿す形態も一般的です。

こうした結線で最も注意すべき点が電氣的絶縁です。高速=広帯域なデジタル信号線やコンピュータは、微弱なアナログ計測信号にとっては強力なノイズ源です。GND線の接続や筐体の接触などによるデジタルノイズのアナログ系への回り込みは、細心の注意で避けなければいけません。

信号線の絶縁には、アナログの場合は絶縁アンプ、デ

authors' e-mail: nakanisi@nifs.ac.jp, okumura@edu.mie-u.ac.jp

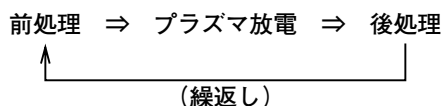
タルの場合は光メディア変換器などが必要になります。LANの標準規格であるEthernet用の光変換器は比較的安価に入手できますが、広帯域のアナログ絶縁アンプ等は大変高価です。導入コスト低減のためにも、電氣的絶縁が本当に必要か、どの部位で絶縁をとるべきかをよく吟味しましょう。

アナログ信号線はT型コネクタにより容易に分岐できますが、末端からの反射が重畳するなど信号波形の崩れる要因にもなります。デジタル信号線も同じなのですが、高周波の矩形信号波形が崩れて'H' = 1が'L' = 0と誤認識されると、伝送データは全く意味を失います。

そのためデジタイザ⇒PC間のデジタル伝送では、信号線そのものを分岐することはほとんどありません。そのかわり同一線路上に複数の装置をつなげられるバス接続[1]を用います。計測用バスの詳細は改めて後節でも触れますが、GP-IB[2]やSCSI[3,4], USB[5], Ethernet[6]など、分岐方法はバス規格毎に規定されています。よく確認して接続してください。規格外の接続をすると、バスの電氣的条件を逸脱してしまい、伝送トラブルに見舞われる危険性が高くなります。

3.1.2 自動制御と実験シーケンス

核融合実験ではほとんどの場合、電源準備等から始まる一連の前準備、実際のプラズマ放電、後処理、を一つの単位として、連続的に多数回繰り返す運転をします。



このようにあらかじめ決められた順序で一連の処理を進行させる自動制御法をシーケンス制御と呼びます。様々な前後処理のタイミングを与える(実験)シーケンスは、通称シーケンサと呼ばれる装置で生成・管理されます。核融合実験は典型的なシーケンス制御ですので、計測器を含めた全装置はシーケンサが出すタイミングで動作します。

古くはシーケンサにはリレー制御盤が用いられていましたが、現在ではプログラマブル・コントローラ(PC)*1とよばれるソリッドステート製品が使われます。いずれにせよ信号の授受は、リレー回路と同じ接点のON/OFF入出力が基本です。接点ON/OFFの時間精度は～数十ms程度と悪いため、次の3.2節で述べるように、トリガなど高精度なタイミング信号としては使えません。

大中規模の装置ではシーケンス制御以外にも、プラズマの密度・温度・位置などの自動制御にフィードバック制御が用いられます。計測器の信号によってアクチュエータ(actuator)への制御出力を変化させ、プラズマの安定保持を図るなどで使われます。

本講座の対象から外れるので、これ以上の詳説はここでは避けませんが、計測⇒制御の信号処理にはニューラルネットを応用したものなど、様々な発展形が研究されています。テキストも多数出版されているので興味があれば参照してみてください。

3.1.3 計測器の制御

最近の計測器は多くがインテリジェント化されており、ホストコンピュータとの間でコマンドを授受しながら色々な制御を行うことができます。こうした通信で用いられる伝送路を一般にフィールドバスと呼びます。オープンなフィールドバス規格の代表がGP-IBであり、自動車の車載計装でよく使われるCANバスです。その他、シリアルバス規格であるUSB, IEEE1394*[7], Ethernet等も普及しており、100 Mbpsを超える高速通信も可能になっています。

その一方で、RS-232-Cに代表される従来からのシリアル通信も、高速通信が必要ない装置で変わらず使われていま

す。いずれにしても計測器との通信路は標準規格であることが多く、WindowsやLinux等OSの標準ドライバから利用可能です。インターネット検索で容易にサンプル・プログラムも入手できるでしょう。また、GP-IBカード等には必ずドライバ・ライブラリが添付されており、サンプル・コードも含まれています。

まずは簡単なコマンドを送るプログラムを作って試してみましょう。実験シーケンスに則った複雑な制御も、1コマンドを送って戻り値をもらう処理の繰り返しの過ぎません。

3.1.4 監視とロギング

計測器やデジタイザは、運転中の機器その他の状態を24時間監視し続ける**運転監視系**と、観測事象が存続している間だけ稼動する**物理計測系**とに大別できます。前者には真空計、後者にはコイル電源、プラズマ波形などが相当します。両者は動作タイミングが全く異なるので、データ保管・取出しを除き、別々の系統に分けたほうが、プログラム開発や障害切り分け等が容易になります。

運転監視系でモニターしている観測値や、機器に加えられた操作を履歴として逐次保存することを**ロギング**と呼びます。真空計のロギングは、長らくペンレコーダで連票紙出力という簡便なアナログ集録でした。しかし小規模な実験でもデータをデジタル化してPCに取り込むのが一般化している今日、ロギングデータだけがオンラインで参照できないのはやはり不便です。計測器のコントローラ類は、フィールドバス経由の制御・データ集録が可能になっているか、あるいは大抵そのオプションがあります。やはり早期にコンピュータ集録へ移行し、他の物理計測データ等と一緒に保管するのが、将来的にも、また共同研究等の利用でも望ましいでしょう。

運転モニタのデータは～1 Hz内外の粗いサンプリングですし、機器の操作履歴はさらに散発的です。このためロギングには、**各サンプルごとに正確な時刻情報を伴っている**必要があります。また一般に、物理計測系が動作する実験シーケンスは標準時刻とは独立しているため、**実験タイミングと標準時刻との対応情報**も忘れずに収集・保存しておかなければいけません。

3.1.5 遠隔操作

Ethernetに代表されるLANが社会全体に普及した現在、

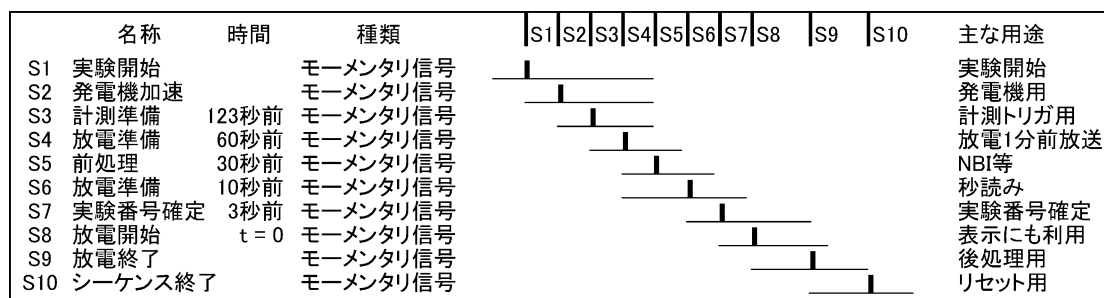


Fig. 1 LHDにおける実験シーケンスの例。主制御信号と呼ばれるS1～S10が、実験の進行タイミングを規定します。

* 1 別称、プログラマブル・ロジック・コントローラ(PLC)です。
 * 2 FireWire, iLINK, DV 端子とも呼ばれます。

実験の各種装置もネットワーク経由で遠隔操作するのが一般的になっています。3.1.1節の電氣的絶縁に関連して触れたとおり、光 Ethernet 規格^{*3}や GP-IB, RS-232-C の光トランシーバを用いれば、光ファイバで距離を大幅に延長できます。利用する光モジュールの種類にもよりますが、300 m 程度の製品が多く、大抵の実験環境には十分な長さです。

LAN (Ethernet) で延長するか、GP-IB や RS-232-C で延長するかは、制御用コンピュータの設置場所やシステム構成にも影響を与えます。前者の場合は制御用 PC を近くにおいて接続し、それをネットワーク経由で別の PC から遠隔操作することになります。これには Windows のターミナルサービス (リモートデスクトップ) やフリーソフトの VNC[8] といった、コンソール画面を遠方に飛ばすツールが便利です。

機器との接続に、専用 PCI カードとケーブルが必須で、光ファイバが使えない場合もあります。付属の専用ユーティリティでないと動かせない機器もあるでしょう。そういう場合にも LAN+リモートコンソールが役立ちます。

便利な反面、**多対多 (P 2 P; Peer-to-Peer) 接続**が基本である LAN では、他の通信による影響を皆無にはできません。外乱による思わぬ通信遅れや、セキュリティ問題を常に念頭においておく必要があります。フィールドバスを延長する方法では、1 台あるいは少数の接続機器で閉じた回線を専有するので、この心配はあまりありません。

能動機器では誤操作による事故や破壊・破損などの危険性が常にあり、遠隔操作にも高い信頼性が求められます。Ethernet でもスイッチングハブを用いて通信帯域の専有性を上げたり、クロスケーブルで対向接続する手もあります。要は利便性と信頼性のバランスが重要です。

インターネット経由でサイト外の遠隔地から操作をする場合、通信品質は LAN から較べて格段に劣るため、通信エラーを前提としたシステム構成が必須です。**クライアント/サーバ (C/S) モデル**[9]では、ハードウェア制御などの機能を提供するサーバと、依頼を出すクライアントとを明確に分離します。

自律的な保護管理のもとに、サーバ側が全依頼の健全性・正当性をチェックすることで、無許可のアクセスや不正操作を拒否できます。依頼内容が不完全であれば破棄するなど、通信エラーや不意の接続断にも強い仕組みが実現できます。

インターネットで実質的標準となっている TCP/IP[10]では、遠距離通信に付き物である通信エラーの検知や再送などのエラー回復処理が自動化されています。セキュリティにせよ通信エラーにせよ、こうしたミドルウェアとよばれる API をうまく選定し活用することで、プログラム開発の負荷が低減されます。

3.1.6 デジタイザの制御と運転

3.1.4節で運転監視系から得られるデータのロギングについて触れたので、ここからは狭義のデータ収集、即ち物

理計測系のデジタイザ運転について考えていきましょう。

デジタイザには各種設定パラメータが多数存在しています。多チャンネルのものではチャンネルごとに設定できる場合が多く、また、新しい製品は一般的に多機能で設定値も多くなります。デジタイザを動作させる前には、それらをすべて運転したい値に設定しなければならないので、前処理はそれだけ長手順になり時間もかかります。

高サンプリング率のデジタイザでは、通常、集録開始のトリガを受ける前から A-D 変換動作を開始しています。トリガ直後の第 1 サンプルを取り落とすことのないようにです。つまり高速デジタイザの前処理には、以下の動作タイミングがあることとなります。

1. デジタイザの内部メモリ、レジスタ等を消去 (初期化) して運転パラメータを設定する。
2. トリガ待ち状態にする。この時点で A-D 変換は開始される。
3. トリガを受けて、それ以降の、あるいは前後の A-D 変換結果を集録する。

2. 以降、A-D 変換結果は内部バッファメモリに書き込まれますが、いわゆる上書きモードで延々と書き換えられています。3. のトリガを受けると、それ以降の上書きを止めて既定時間経過後に A-D 変換を終了します。

こうしたトリガの利用は **STOP トリガ動作** (Fig. 2 参照) と呼ばれます。STOP トリガではトリガ入力直前の A-D 変換結果も保存できるので、遷移現象が起こった前後の状態を記録することが可能です。遷移現象の信号自体をトリガとして A-D 変換を駆動する動作を、特に **イベントトリガ動作** といいます。定常化実験で重要となる技術の一つです。

通常、デジタイザの設定には既定値 (default) があるので、実験前に毎回全ての設定値を与える必要はなく、既定値以外の設定のみで十分動作します。しかし次節でも述べるとおり、デジタイザの動作パラメータは、サンプリング

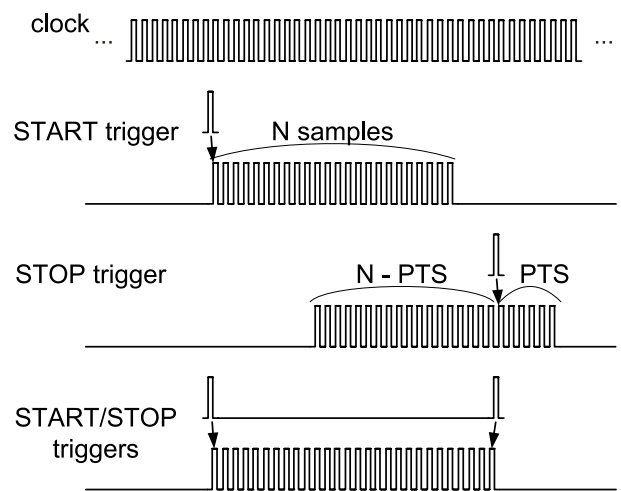


Fig. 2 START, STOP トリガ動作の違い。N はバッファサイズ、PTS は Post Trigger Samples の略、N=PTS にすると START トリガ動作と STOP トリガ動作は同じ結果になります。START/STOP トリガ動作では予めサンプル数は決まりません。

* 3 10/100BASE-FL や 1000BASE-SX/-LX 等が規定されています。

速度やフルスケール、サンプル数など、計測信号を復元するための重要データです。A-D 変換結果の配列データを補完するのですから、実験毎に必ずセットで保存しなければなりません。デジタイザの設定が正しく行われているかの確認のためにも、上記 1. の手順でプログラムから全設定値を明示的に設定し、デジタイザから再度設定を読み取って比較検証するのが望ましいでしょう。

デジタイザの後処理動作については次節で取り扱います。

3.2 データを生成し転送する

3.2.1 データとは何か。どう取り扱うのか

実験によって得られるデータといえば、計測器から出力される信号波形や画像などの生データがまず頭に浮かびます。しかしデータとして忘れてはならないのが、本体実験装置や計測器などを運転したときの各種設定パラメータです。

デジタイザが一括して生成する生データとは異なり、機器の運転パラメータはあちこちで分散して与えられます。また、その形態もしばしば多岐にわたるため、運転情報の収集システムを後から開発するのはとても大変です。増設・増強が随時可能な計測器やデジタイザとは異なり、**運転パラメータの収集・保存方法は、システム構築の最初からよく検討しておく必要があります。**

核融合の実験では、通常、実験回数が数万～数十万回と多くなります。こういった実験が過去に行われたのかを把握するためには、機器の運転パラメータが系統的に保存され、検索・閲覧がスムーズにできるようになっている必要があります。こうした目的には、データベース管理システム (DBMS) の活用が好適です。これについては 3.3 節にて解説することになります。

実験データの主役的存在である計測データも、デジタイザが生成する 1 次元あるいは 2 次元の整数配列だけでは用をなしません。密度や温度といった物理量に変換するためには、当然、各種の変換係数が必要です。デジタイザの運転パラメータの他に、プリアンプのゲインやセンサーの較正值などがこれに含まれます。これらは生データの**属性**、あるいはデータに対する**メタデータ**と呼ばれ、生の配列データと同じくらい重要です。

3.2.2 トリガとクロック

真空計など 24 時間連続で各種状態を監視する常時モニター装置を除き、計測器やデジタイザを運転するには、それらを動作させるタイミングを外部から与えてやる必要があります。最も重要といえるのがデジタイザに A-D 変換開始を指示する**トリガ (trigger)** です。

核融合プラズマの挙動には一般に固有の特性周波数 (あるいは周期) があることがよく知られており、電磁流体力学 (MHD) 的な揺動は数 kHz～数百 kHz といわれます。こうした挙動を計測するには、当然それ以上の時間精度が必要です。つまり一般的な核融合計測の場合、データ集録開始トリガ以降の経過時間は 1 MHz (1 μ s) 以上の分解能で測られる必要があります。この周期的な時間間隔を精度よく

与えるのが**クロック (clock)** と呼ばれるデジタル信号です。一般に ADC ではクロックに同期してデータがサンプリングされますので、デジタルデータの出力は 1 クロックにつき 1 サンプルとなります。

数 kHz を超える高速 ADC モジュールの多くは、内部に周波数可変のサンプリング・クロックを内蔵しており、外部/内部クロックを切り替えて利用できるようになっています。内部クロックを用いるとクロック配線の手間が省けて便利ですが、複数モジュールを使って多チャンネルの信号を同時計測する場合、同じクロック設定をしても、モジュール間でクロック位相がずれます。周波数が低くなるほど位相差による時間のずれは大きくなりますから、全モジュール共通に単一の外部クロックを与えてサンプリングを同期させる等が必要になります。

他方、コンピュータ内蔵のいわゆるシステムクロックには、通常 100 Hz～1 kHz が使われており、プラズマ計測で用いるサンプリング・クロックよりかなり低い周波数です。このため PC のシステム時計は、A-D 変換開始等の時間精度が必要なタイミング制御には使えません。ご注意ください。

3.2.3 データ転送の開始タイミング

計測対象となるプラズマの持続時間は、核融合実験の歴史の中で長らく 1 秒未満でした。今でも 1 分を超える保持が可能なのは、超伝導コイルを用いている TRIAM[11]、LHD[12] 等、一部装置の定常化実験に限られています。小型装置では 1 ms 以下の場合もあります。計測用 ADC としては、電圧信号波形を時系列デジタル・データに逐次変換する**トランジェント・レコーダ型**がよく使われます。観測したいプラズマ挙動の周波数帯、例えば流体力学的揺動などにあわせてサンプリング速度は kS/s～MS/s で、kS～MS 程度のデータ・バッファを内部に備えているタイプです。

工場プラントなどの計測制御用途よりかなり高いデータ生成率ですから、データ転送では、通信帯域幅を最大限利用できる方式が求められます。データのある程度の大きさにまとめて一気に送る**ブロック転送**がそれです。逆に、生成されたデータを 1 サンプルずつ即座に送る方式をとれば、データ到着のリアルタイム性は向上します。しかし通信に占めるオーバーヘッドの割合も大きくなるので、単位時間に送れるデータ量は大幅に下がってしまいます。そのため、実験時間の短かった核融合実験では、実験終了後にデータ転送および解析・表示を一括して行う**バッチ処理**動作が定着しました。

しかし 21 世紀に入ってようやく、計測器やデジタイザでも Ethernet や USB に代表される安価で高速なデータ伝送路が利用可能になってきました。これまではリアルタイムな連続処理かバッチ処理かによって処理システムを分離して構築・運転していましたが、実時間性と高速性を両立した新たなデータ収集系の模索も始まっています[13]。

さて、放電終了後、データ転送開始タイミングを知る/知らせるにはいくつか方法があります。すでに A-D 変換が終了してデジタイザ中に蓄えられている観測データの転送

開始ですから、基本的に時間精度は必要ありません。

1. 放電終了/データ生成完了の信号を受ける (ハードウェア割込み)
2. データが生成されたかどうかを繰り返し確認し続ける (ポーリング)
3. 一定時間経過後 (タイマ)
4. ネットワーク経由で放電終了情報を頒布 (メッセージ・パッシング)

2や3の収集プログラムは比較的簡単に作ることはできますが、2ではシステム負荷が不必要に上がる、3では実験タイミングの変更に柔軟に対応できない等の欠点もあります。プログラム開発負荷も考慮して、ふさわしい手段を選択しましょう。

3.2.4 データの生成量

トランジェント・レコーダ型 ADC の場合、単位時間当たりのデータ生成量は単純にサンプリング速度(周波数)に比例します。1サンプルあたりの分解能によっても変わりますが、よく用いられる10~16ビット分解能のADCでは、1サンプルのデータ長は2バイトになります。

例をあげて計算してみましょう。100 kHz サンプリングのADCが生成するデータ量は、毎秒

$$2 \text{ byte/sample} \times 10^5 \text{ sample/s} = 0.2 \times 10^6 \text{ byte/s}$$

と1チャンネルあたり0.2 MB/sです。モジュール型のデジタル・フロントエンド (DFE) を使って1筐体に50チャンネル入れると10 MB/sとなります。CCDなどのカメラ計測ではどうでしょう。画面がVGA (幅640高さ480ピクセル) で分解能がモノクロ (グレースケール) 10 bit, プログレッシブ・スキャン (60 fps)*4の場合、

$$2 \text{ byte} \times (640 \times 480) \times 60 / \text{s} = 36.864 \times 10^6 \text{ byte/s}$$

と、カメラ1台で約36.864 MB/sになってしまいます。

このようにデータの生成率がわかれば、それに計測時間をかけることで生成量が得られます。上の例にならってデータの生成率と生成量を、各計測器あるいはデジタル群ごとに計算してみましょう。システム動作への理解が深まるほか、次の増設、設計見直しへの大きな助けになります。

3.2.5 どれくらいまで転送できるか

デジタルの種類によっては、データ生成率がPCへのデータ転送速度を超えるような運転も一時的には可能です。しかし、転送できないデータが徐々にバッファ内に溜まっていき、いつかはバッファが溢れてしまいますので、この状態を長く続けることはできません。特に、リアルタイム収集を行う場合は1系統あたり厳密に、

$$\text{データ生成率} \leq \text{データ転送速度}$$

の条件を満たしている必要があります。データ量は無限に大きくなり得ますので、次節で扱う保存の面でも注意が必要です。

これに対してバッチ収集の場合、A-D変換結果は内部バッファに蓄えられ、データ転送開始は変換完了後になります。生成率は転送速度からは独立した条件となりますが、データ量はバッファ・サイズが上限となります。ADCの各チャンネルで

$$\begin{aligned} \text{データ生成率} \times \text{変換時間} &= \text{データ生成量} \\ &\leq \text{バッファ・サイズ} \end{aligned}$$

となります。転送時間に余裕があるバッチ収集も、当然、次の実験が始まるまでの3分~5分の有限の時間内に処理を完了させなければなりません。核融合実験ではこれをよくショット間処理と呼びます。

ショット間では、データ転送から解析・表示まで全ての処理が必要になります。それを考えると、時間的に必ずしも余裕があるわけではなく、特にデータ量が多くなった場合は非常に厳しい時間配分を求められます。計測データは全て、

デジタル⇒データ伝送路⇒収集プログラム

と転送されます。伝送路にはいろんな種類があり速度も様々ですが、伝送路の通信帯域を最大限有効利用するためには、収集プログラムの処理がデータ転送の足を引っ張らないようにする必要があります。

では具体的にどれぐらいのデータ転送が可能なのか簡単に計算してみましょう。3分周期の実験を想定すると、ショット間処理で使える時間は余裕を見て120秒程度でしょう。計測器I/Fとして一般的なGP-IBは、最大伝送性能こそ1 MB/sながら実行速度は数十 kB/sです[1]。仮に30 kB/sならGP-IB経由でデジタルからデータ収集できる上限は、

$$30 \text{ kB/s} \times 120 \text{ s} = 3.6 \text{ MB}$$

1つのGP-IB系統で約3.6 MBになります。これは切れ目なく120秒間データを送り続けた場合の数値です。もしアプリケーションが1チャンネルごとに、転送⇒解析⇒表示⇒保存の処理を順次行っていたりすると、転送の時間帯以外はデータが流れません。収集できるデータ量もそれだけ減ってしまいます。

(中西秀哉)

3.3 データの保存と共有

保存・共有すべきデータとは、一般に生データと呼ばれるバイナリーデータと、検索・統計処理の対象となるパラメータ類です。後者はExcel表、あるいはCSVテキストのファイル形式で、実験番号ごとにレコード(行)を蓄積していくことが多いでしょう。こうしたファイル保有は簡便な反面、コピーを繰り返している間に複数バージョンが出来てしまう問題点があります。本節の最後で触れますが、データベースを利用することでこれは解決できます。

データサイズ的には表データは生データに比べてはるか

* 4 fps: frame per second. プログレッシブ・スキャンはノンインターレース・モードともいわれる。通常のTVで使われるのは偶数奇数の走査線を2回に分けるインターレース・モードでNTSC 30 fps, PAL 25 fps.

に少量です。また全実験にまたがるデータを1ファイル中にまとめることで統計処理などの利用価値も上がります。これに対して生データは個々のサイズが大きく、生成される日時もバラバラですので、どうしても多数のファイル群になってしまいます。これらを安全に管理するには信頼性のあるファイルシステムと、冗長性のある保存・複製装置が重要になります。

3.3.1 ジャーナリングファイルシステム

実験室環境ではUPS(無停電電源装置)を使っているにもかかわらず、種々のトラブルや人的ミスで不意の電源断が生じることがあります。2004年10月23日の新潟県中越地震でも、停電が繰り返し起こったため、小千谷市のサーバがUPSのバッテリー切れで自動シャットダウンできずクラッシュしたと報じられています*5。

従来のファイルシステムは電源断で壊れやすく、しかも再起動時にスキャンディスクやfsck(UNIX系OSのファイルシステムのチェック)に時間がかかりました。しかし、今はWindowsならNTFS, Linuxならext3やReiserFS, XFS, JFS, Reiser 4といったジャーナリングファイルシステムがあります。電源断があっても、記録をロールバックし、ファイルシステムの整合性を保ってくれます。Solaris 7以降やMac OS X 10.2.2以降でもジャーナリングを設定できます。

最近のLinuxでよく使われているext3は従来のext2との互換性が売りですが、ジャーナリングなしのext2に比べて書き込み速度が若干遅いようです。高速性が要求される場合は他の選択肢を考慮するといいいでしょう。なお、ディスクI/Oはメモリにキャッシュされますので、見かけの速度はメモリの量によって大きく変わります。

3.3.2 RAID

複数のディスクを用いて全体としての信頼性や速度を上げるRAID(Redundant Array of Independent Disks, レイド)と呼ばれるディスクアレイ技術がよく使われます[14]。

RAIDには0~5のレベルがありますが、0, 1, 5がよく使われます。

RAID 0は、2台のHDDにデータを分割して書き込みます(ストライピング)。速度は2倍に近づきますが、信頼性が増すわけではありません。

RAID 1は、2台のHDDに同じ内容を書き込みます(ミラーリング)。速度はやや落ちますが、信頼性が増します。

RAID 5は、 n (≥ 3) 台のHDDを使い、データを $n-1$ 台に分割して書き込み、残りの1台にパリティ情報を書き込みます。どの1台にパリティ情報を書き込むかは常に変化します。ストライピングによって速度が改善されるだけでなく、どれか1台が壊れてもデータが失われません。

また、RAID 0+1などと称して、RAID 0をさらにミラー化することもあります。

RAIDはさらにソフトウェアRAIDとハードウェアRAIDに分類されます。一般に、ハードウェアRAIDのほうが高速で、電源を切らずにディスクを入れ替えるホット

スワップができるため便利です。ただ、RAIDコントローラが故障して代品が得られない場合は、データがまったく読み出せなくなる可能性もあります。この点ではソフトウェアRAIDのほうが安心です。速度についても、CPUが速くなれば両者の差は縮まります。

3.3.3 S.M.A.R.T.

ディスクのクラッシュには、異常なシーク音やリトライによる速度低下といった前兆を伴うことがあります。

最近のHDDが備えるS.M.A.R.T.(Self-Monitoring, Analysis and Reporting Technology)というモニタ機能を使えば前兆を捉えることがある程度可能です。温度センサを備えたディスクなら、この機能で温度を調べることができます。

UNIX系OSではsmartmontools*6を使って異状が見つかれば管理者にメールが届くように設定しておきましょう。

3.3.4 rsyncを使った同期

本格的なミラーリングでなくても、実験ごと、あるいはUNIX系OSが標準で備えるcronを使って定時に別ディスクにミラーすることができます。具体的には、rsyncコマンドを使って

```
rsync -auv /usr/local/data /opt/local/
```

とすれば/usr/local/dataディレクトリをそのまま/opt/local/dataにコピーします。すでに同じファイルがあれば、新しいものだけが送られますので、単なるcpコマンドより無駄がありません。

データ量が膨大でなければ、ネットワーク経由で別マシンにミラーしておけば、さらに安心です。

```
rsync -auv /usr/local/data foo:/usr/local/
```

とすればローカルマシンの/usr/local/dataディレクトリをリモートマシンfooの同じ場所にコピーします。逆に

```
rsync -auv foo:/usr/local/data /usr/local/
```

とすればリモートマシンfooから更新部分を取り寄せます。これにより二つのマシンの同期をとることができます。

ネットワーク的に遠いマシンの場合は、オプション-zを付ければ、圧縮して送り、リモートマシンで伸長します。圧縮ライブラリzlibが使われますが、これは後述のように必ずしもバイナリデータの圧縮に適切な方式ではありません。

rsyncは無指定ではリモートマシンでリモートシェルrshを起動しますが、最近ではセキュリティ上の理由でこれを標準では許可していないはずです。その場合は、オプション-e sshを追加するか、環境変数RSYNC_RSHにsshを設定すれば、安全なシェルssh経由で(暗号化して)送られます。より性能を上げるにはrsyncをサーバとして起動しておきます。詳細はman rsyncでマニュアルをご覧ください。

cronコマンドを使った定時バックアップについては、

* 5 <http://itpro.nikkeibp.co.jp/free/NCC/NEWS/20041106/152236/>

* 6 <http://smartmontools.sourceforge.net/>

cron や crontab について、やはり man コマンドで調べてください。

Windows の UNIX 互換ツール群 Cygwin にも rsync や ssh コマンドが含まれていますが、本物の UNIX 互換 OS より性能が出ません。Windows 同士なら単にファイル共有を使うのが簡単で高速です。Windows・UNIX 間なら、UNIX 側で Samba を立ち上げてファイル共有するか、あるいは FTP や SCP, WebDAV 等を使います。逆に、Windows 側の共有フォルダを UNIX 側から smbclient で使うこともできます。

3.3.5 FTP, SCP, WebDAV

FTP は昔から使われてきたファイル転送プロトコルです。ダウンロードには今では HTTP のほうが便利ですが、アップロードにはまだよく使われます。FTP では認証パスワードが暗号化されずにネットに流れるので、安全でない場所を通過する場合は ssh (scp) を使いましょう。Windows では WinSCP がよく使われます。

最近ではアップロードにも HTTP プロトコルを使う WebDAV が流行しています。シェア No.1 の Apache (アパッチ) Web サーバも、2.x では標準で WebDAV をサポートしています。SSL を併用すればパスワードもデータも暗号化されますが、実験データでは SSL は大袈裟かもしれません。やはり Apache 2.x で標準サポートされた Digest 認証を使えば、パスワードだけ安全な形でやりとりされます。

3.3.6 NAS

ファイル共有用に別マシンを立ち上げなくても、今なら NAS (Network Attached Storage, ナス) を使うのが手取り早いでしょう。数万円程度でもギガビット LAN に対応したものがあありますが、TCP/IP や SMB プロトコルのオーバヘッドのため、単純な USB 2.0 接続の外付けディスクの速さにはなかなか及びません。非 TCP/IP の NDAS (Network Direct Attached Storage) プロトコルを使ったものは、より高速になります。

より高価なものは、RAID, ホットスワップに対応しています。

3.3.7 データ圧縮

データ圧縮は単にディスク容量を節約するためだけのものではありません。CPU に余力があれば、データ圧縮をすることにより、ディスクやネットワークの見かけのスループットが上がります。要は、CPU と I/O のバランスの問題です [15]。

例えば、12ビットの ADC 出力を 2 バイトとして扱わず、二つで 3 バイトとするようなトリビアルな圧縮でも、データ量が 3/4 になり、見かけの I/O のスループットは 33% 向上します。

こういった ADC 出力は、Zip や gzip や zlib ライブラリといった汎用ツール [16] で圧縮しても、CPU にかかる負荷の割には、あまり縮みません。その理由は、これらのツールが、繰り返し現れるバイト列を見つけるために大部分の時間を割いているからです。ADC の生出力に同じデータ列が繰り返し現れることは、人間の書いた文章ほどは期待できません。また、汎用ツールはバイト単位でデータを見て

いきますので、バイト単位でないデータは不利になります。

あまり動きの激しくないチャンネルについては、時間軸に沿って差分を取れば、0 に近い値が増えるために、縮みやよくなります。このとき、例えば 16 ビット値 (-32768 ~ 32767) 同士を引き算すれば範囲は 17 ビット (-65535 ~ 65535) になるように誤解されるかもしれませんが、引き算は mod 65536 ですればいいので、差分も 16 ビットに収まります。

こういった方法をさらに洗練したものが、音楽データ (16 ビット) のロスレス (lossless, 可逆) 圧縮ツールとして配布されています。MP3 などはロスレスでないので、実験データの圧縮には使えません。

2次元の画像データについては、JPEG はロスレスではありませんので、PNG や JPEG LS などロスレスの圧縮形式を選びます。JPEG LS は PNG よりずっと軽いロスレス圧縮方式としてもっと使われてしかるべきです。ロスレスでなくていいなら、JPEG よりノイズが目立ちにくい JPEG 2000 をお勧めします。

3.3.8 データベースの利用

ADC から取得した大量の生データなどは、通常のバイナリファイルとして保存すればいいでしょう。

実験のパラメータなどのメタデータをどう保存するべきかは、ケースバイケースです。管理データはテキストファイルで保存するのが UNIX の伝統であり、そのほうが grep や Perl などで簡単に扱えますし、万が一ファイルが壊れても、生きているデータを拾い出せる可能性があります。逆に、不馴れな人にテキストエディタで編集させると必ず壊されてしまうのも事実です。専用のツールでないと編集できないようにして、編集時に入力チェックを十分するようにしておくほうが安全です。

データ入力ツールを作る際は、ファイル入出力を自前で行うより、データベース管理システム (DBMS) を使うほうが楽です。同時に更新をかけてもデータが壊れたりしませんし、大量のデータでもインデックスを生成することにより高速に検索できます。

よく使われるオープンソースの DBMS には、機能が豊富な PostgreSQL [17], 高速な MySQL [18], 簡便な SQLite [19] などがあります。SQLite はまだ知名度が低いかもしれませんが、PHP 5 に同梱されて有名になりました。大規模システムには使えませんが、バックグラウンドでプロセスを動かす必要がなく、単なるファイルを扱う気軽さで使えるので便利です。

(奥村晴彦)

参考文献

- [1] H. Nakanishi, M. Kojima *et al.*, J. Plasma Fusion Res. **72**, 1062 (1996).
- [2] 岡村勉夫: 「標準デジタル・バス (IEEE-488) とその応用」(CQ 出版社, 東京, 1981).
- [3] インターフェース編集部 (編): 「最新 SCSI マニュアル」(CQ 出版社, 東京, 1989).

- [4] 大島啓孝他：「SCSI 完璧リファレンス」オープンデザイン, No.1 (CQ 出版社, 東京, 1994).
- [5] 桑野雅彦他：「USB 機器&ドライバ実践開発手法」インターフェース, No.1 in 2001, p.51 (CQ 出版社, 東京, 2001).
- [6] 好川哲人：「LAN 技術総合入門」オープンデザイン, No.12 (CQ 出版社, 東京, 1996).
- [7] 北山洋幸他：「IEEE1394 のハード&ソフト入門」インターフェース, No.7 in 1999, p.51 (CQ 出版社, 東京, 1999).
- [8] VNC <http://oku.edu.mie-u.ac.jp/~okumura/linux/?VNC> (2004).
- [9] 吉岡隆一, 山本哲夫, 松尾紀彦, 須藤隆成：「C/S ネットワーキング」(日経 BP 出版センター, 東京, 1995).
- [10] D.E. Comer and D.L. Stevens：「TCP/IP によるネットワーク構築 Vol. III」(共立出版, 東京, 2003).
- [11] E. Jotaki and S. Itoh, J. Plasma Fusion Res. **73**, 330 (1997).
- [12] H. Nakanishi, S. Hidekuma, M. Kojima *et al.*, J. Plasma Fusion Res. **72**, 1362 (1996).
- [13] H. Nakanishi, M. Kojima and LABCOM group, Fusion Eng. Des. **56-57**, 1011 (2001).
- [14] 日高亜友：「ローダブル・カーネル・モジュールを使った計測データの蓄積テクニック」インターフェース, No. 11 in 2004, p.96 (CQ 出版社, 東京, 2004).
- [15] 奥村晴彦：「データ圧縮の基礎から応用まで」C Magazine, No. 7 in 2002, p.13 (ソフトバンク, 東京, 2002).
- [16] 奥村晴彦, 山崎敏：「LHA と ZIP：圧縮アルゴリズム×プログラミング入門」(ソフトバンク, 東京, 2003).
- [17] PostgreSQL <http://www.postgresql.org/>
- [18] MySQL <http://www.mysql.com/>
- [19] SQLite <http://www.sqlite.org/>



おくむら はる ひこ
奥村 晴彦

物理と計算機を行ったり来たりしていましたが、2004年4月からは家から自転車で通える三重大学の教育学部情報教育課程に移り、なんとなく情報教育が専門になりました。



なかにし ひで や
中西 秀哉

核融合科学研究所助手。1995年東京大学大学院原子力工学専攻博士課程単位取得退学。博士(工学)。主な研究分野は、実験情報システム、特にデータ収集・処理、計測制御工学等です。素粒子や天文のように、核融合分野からも新たな計算機技術を生み出せないかと日々模索していますが、当面の目標はSLACのデータ量をLHDで抜くことです。ITERはLHCを超えられるでしょうか。