

リアルタイムデジタル制御とネットワーク技術の修得を目的とした遠隔制御実験システムの検討

山田 健仁^{*1} 芳川 健^{*2}

Experimental remote-control system for aiming at acquisition of real-time digital control and network communication technology

Takehito YAMADA ^{*1} and Takesi YOSIKAWA ^{*2}

Abstract

A remote control and a digital control technology are important for recent industrial automatic control equipments. In order to master the technologies, college students have to learn much knowledge of electronics and computer. We propose a basic experiment system for remote control equipment which works under a μ TRON real-time operating system. The system is constructed from a ball-and-beam mechanism, an A/D-D/A interface and computers. The computers which have Ethernets are connected by a local-area-network and they communicate each other by a TCP/IP protocol. By using the experiment system, the students will be able to understand the basic technology of the digital control and the remote control.

Key Words : remote control, digital control, network, experimental system

1. まえがき

本論文では、コンピュータを中心に構成するデジタル制御器の修得と、LAN(Local Area Network)を利用した遠隔制御方式の基礎技術の修得のために使用する、電子・情報工学系実験用の実験システムの一形態について検討した結果を述べる。

産業機器や医療機器などにおいて、LAN や WAN (Wide Area Network) などの通信ネットワークを利用した遠隔制御機器が実用的になってきている。このような遠隔制御機器の技術を修得するためには、コンピュータによるデジタル制御技術、リアルタイム OS (Operating System) の下でのリアルタイム制御技術、ネットワーク通信技術、Web プログラミング技術など、ソフトウェアとハードウェアの複合的で広範な技術知識が必要となる。本研究は、このような遠隔制御技術

を修学者に実体験させるための電子・情報系の基礎実験システムを開発することを目的としている¹⁾。これまでも LAN を利用した遠隔操作実験装置の提案があるが²⁾、これは主にシーケンス制御を扱ったもので、デジタル制御の実験は対象としていない。

本実験システムの制御対象側装置は、ビーム上のボール位置を制御するボール&ビーム機構³⁾、デジタル制御器を実装するための A/D 変換器と D/A 変換器を内蔵したワンチップマイコンと、そのワンチップマイコンと接続した LAN 用 LSI を搭載した、ワンボードマイクロコンピュータ (以下マイコン) からなる。マイコンの OS には、リアルタイム処理に対応できるように μ ITORN⁴⁾ を採用した。これによりリアルタイムデジタル制御を実現できるようにした。デジタル制御手法は、理解が比較的容易で実用性の高いデジタル PID 制御とし、 μ ITORN 上に C 言語で記述した

*1 情報電子工学科

*2 広島大学工学部

制御プログラムとして実装した。遠隔制御のためのコンピュータ間のネットワーク通信プロトコルは、TCP/IP プロトコルとしてソケットプログラミングによりコンピュータ間通信を実現した。また、遠隔制御側のクライアントコンピュータ（以下クライアントPC）は、一般的に普及している Windows OS を搭載した PC として、クライアントプログラムを Visual C++ 6.0（Microsoft 社製）で記述した。クライアントプログラムには、制御命令の送信、制御データの受信とそのグラフ表示、そして、マイコンの制御プログラムの実行に必要な各種制御パラメータの変更機能を実装している。

本論文では、この遠隔制御基礎実験システムの詳細な構成を示し、遠隔制御技術のようなコンピュータを使った複合技術を修得するための電子・情報系工学実験用システムの一形態を提案する。

2. 実験システムの概要

図1に本論文で提案する、遠隔制御システムの実験装置の基本構成図を示す。このシステムは次の5つの要素から成っている。

- 1) 制御対象の機構（ボール&ビーム機構）
- 2) 制御対象側マイコン（サーバマイコン）
- 3) A/D、D/A変換器（ワンチップマイコンに内蔵）
- 4) ネットワーク回路（LAN用LSIはサーバマイコンに実装）
- 5) 遠隔制御側PC（クライアントPC）

実験システムの制御対象機構部には、ボール&ビーム機構を採用した。ボール&ビーム機構とは、図1の制御対象の模式図に示すように、ビーム（金属梁）上を転がるボール（金属球）を、ビームの傾斜角度を制御することによって、目標とする任意の位置に位置決

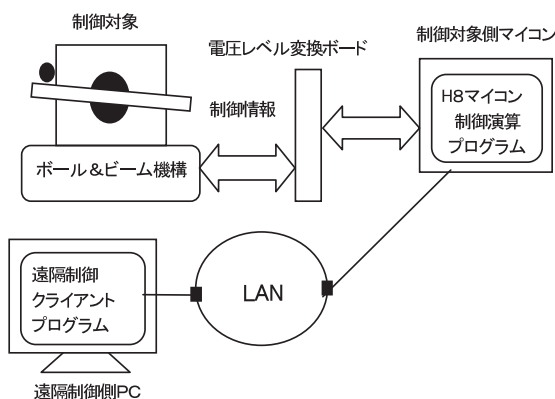


図1 遠隔制御システム実験装置の基本構成図

めする制御機構である。ビームはDCモータによってその角度を変化できるようになっている。DCモータの駆動回路は、ワンチップマイコンに内蔵されているD/A変換器からの操作量によって駆動される。ボール&ビーム機構からの出力信号は、ボール位置信号とビーム角度信号の2つの制御量である。ボール位置信号は、ビーム上のボールの位置に応じてポテンショメータの原理によって得られる電圧信号で、また、ビーム角度信号は、ビームの回転軸に取り付けられたポテンショメータから得られる電圧信号である。これらの信号は、制御対象側マイコンに内蔵されているA/D変換器により語長10bitのデジタル信号に変換され、制御量としてマイコン内部のレジスタに取り込まれる。マイコンはビーム角度とボール位置の制御量より、デジタル制御則に従った演算により操作量を求める。この操作量がマイコンに内蔵されているD/A変換器に与えられ操作電圧に変換されて、DCモータが駆動される。このようなデジタル制御演算には、リアルタイム処理が必要となるため、サーバマイコンのOSには、機器組み込み制御用のリアルタイムOSである μ ITRON4.0仕様のTOPPERS/JSPカーネル⁵⁾を採用した。また、LAN上でのネットワーク通信のためのTCP/IPプロトコルスタックとして、TOPPERSプロジェクトで提供されているTINET⁶⁾を使用した。

2.1 サーバマイコンとその周辺回路

本実験システムでは、制御対象側マイコンであるサーバマイコンに、TOPPERS/JSPカーネルがサポートされているワンチップマイコンのH8/3069F（ルネサステクノロジー社製）を使用した。H8/3069FはH8/300Hマイクロプロセッサを核にして、入出力周辺回路などを集積したワンチップマイコンである。H8/300Hは、遠隔機器制御のシステム構成に必要な周辺回路として、16ビットタイマ、8ビットタイマ、シリアルコミュニケーションインタフェース（SCI）、A/D変換器、D/A変換器、I/Oポート、DMAコントローラ（DMAC）を内蔵している。また、512KバイトフラッシュROMと16KバイトRAMが内蔵されている。

表1にH8/3069Fの主な仕様を示す。本実験システムの電子回路部には、このワンチップマイコンを実装しているAKI-H8/3069FフラッシュマイコンLANボード（秋月電子製：以下AKI-H8）を使用した。このボー

表 1 H8/3069F の主な仕様

メモリ	ROM	512K バイト	外部拡張可能
	RAM	16K バイト	外部拡張可能
I/O	ITU	16 ビットタイマ ×5CH	
	TPC	4CH バス出力	
	WDT	ウォッチドッグタイマ	インターバルタイマとして使用可
	SCI	独立 2CH	
	A/D	10 ビット分解能 ×8CH	サンプル&ホールド内蔵
	D/A	8 ビット分解能 ×2CH	
	I/Oポート	入出力端子 78本 (最大)	

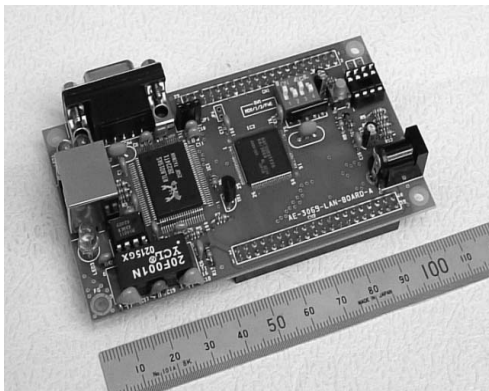


図 2 AKI-H8/3069F フラッシュマイコン LAN ボードの概観

ドは、通信インタフェースとして RS232C 用回路と Ethernet 用回路を搭載している。LAN には、ワンチップイーサネットコントローラ LSI (RTL8019AS) を介して通信接続する。この LAN 機能を使用して遠隔制御側 PC であるクライアント PC と通信する。図 2 に AKI-H8 の外観を示す。

2.2 サーバマイコンの OS (μITRON) について

サーバマイコンで使用した μITRON (Industrial The Real Time Operating System Nucleus) は、主に工業製品に搭載されることを前提としたリアルタイム OS である。

μITRON は以下の特徴を持つ。

- ・リアルタイム OS である
- ・マルチタスクに対応している
- ・リソースが少なくて良い

・OS 仕様がハードに依存しない

本実験システムに有効となる μITRON の最も大きな特徴は、マルチタスクに対応していることがあげられる。タスクは単純にプログラムを実行するだけではなく、事象の発生を待ったり、他のタスクからの情報を受け付けたりする。μITRON 仕様 OS には、複数のプログラムを管理する機能があり、複数のタスクを定義すると、OS はこれらのタスクを並列的に実行する。対象のマイコンはシングルプロセッサ構成であるから、当然、複数のタスクは完全に同時には処理できず、1 つのタスクが待ち状態になると、別のタスクが動作するという具合に交替で動作する。タスクは実行状態、待ち状態などさまざまな状態になるが、それらは OS によって管理される。本実験システムのサーバマイコンのプログラムでは、後述の並列に実行する処理が 2 つある。これらの処理をタスクとして、見かけ上同時に実行していく。

2.3 遠隔制御用 LAN システム

本実験システムの LAN システムのネットワークプロトコルスタックには TINET を使用した。これは、(社) トロン協会 ITRON 専門委員会の Embedded TCP/IP 技術委員会により 策定された「ITRON TCP/IP API 仕様 Ver. 1.00.01」に準拠して TOPPERS/JSP カーネル用に開発された TCP/IP V4 プロトコルスタックである。ITRON TCP/IP 仕様では、主に以下を必要性能としている。

- ・最小コピー回数
- ・動的メモリ管理の排除
- ・非同期インタフェース
- ・API 毎のエラーの詳細化

TINET は上記に上げた性能を満たし、組み込みシステムの制約の中で特に厳しいメモリ容量の制約への対応を優先的に行っている。実装ターゲットとして H8/3069F を用いた場合には、TCP/IP プロトコルスタックに必要なメモリ容量は、スタックと内部バッファを含めても、IPV4 で RAM が約 10K バイト、ROM が約 47K バイトと省メモリ性に優れている。

本実験システムでは、Ethernet ネットワークを使用した通信を行っており、TCP/IP プロトコルを採用している。通常、TCP/IP

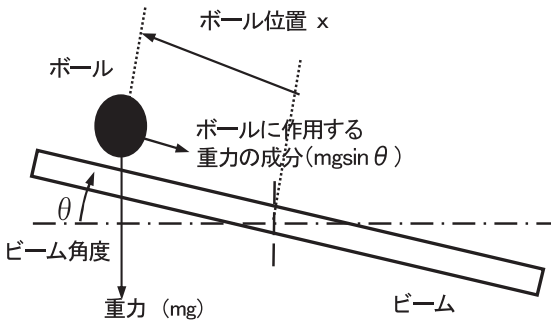


図3 ボール&ビーム機構の動作モデル

プロトコルはプログラムで記述しなければならず、プロトコルの処理手順が膨大で、学生実験でプログラミングさせるのは困難といえる。しかし、あらかじめ TCP/IP プロトコルの処理をまとめた命令集合を持つ TINET を使用することによってネットワークプログラミングが容易となった。

3. 実験システムの実装

実験システムの実装には、サーバマイコン上で動作する μ ITRON によるデジタル制御、TCP/IP のソケットによる通信、クライアント PC 上で動作する遠隔制御用クライアントプログラム、などの開発が必要となる。

3.1 デジタル制御器

本実験システムは、コンピュータ制御のため、制御器をデジタル制御器で実装する必要がある。制御器は古典制御理論で設計する PID 制御器とし、連続時間系で設計した後、双線形 S-Z 変換により、離散時間系の制御器に変換する。この際、制御帯域 1 Hz 程度に対して、サンプリング周波数を 33 Hz とした。これは、離散化で生じる問題点が、観測される現象として現れないようにすることにより、修学者にアナログ制御の延長としてのデジタル制御の基礎を理解させるためである。

制御対象であるボール&ビーム機構の動作モデルを図3に示す。ボールをビーム上の任意の位置に位置決めするためには、DC モータによりビームを一旦傾けて重力によりボールを転がし移動させる。目標位置近傍でビームを逆方向に傾けることによりボールに制動をかけ、ビームを水平にすることにより、ボールを静止させる。この時のビームの傾斜角度とボールの運動

の関係をボールの回転を無視して、ニュートンの運動方程式で表すと式(1)となる。

$$mg \sin \theta = m \frac{d^2 x}{dt^2} \quad (1)$$

(但し、 θ : ビーム角度、 x : ビーム上のボール位置、 g : 重力加速度、 m : ボールの質量)

式(1)に示すように、この系は非線形システムとなる。従って、この系には線形制御器の設計手法が適用できない。しかし、ビームの傾きが小さいとすると、 $\sin \theta \cong \theta$ と近似できるから、式(1)は式(2)のように線形化できる。

$$mg\theta = m \frac{d^2 x}{dt^2} \quad (2)$$

式(2)をラプラス変換して、ビーム角度 $\Theta(s)$ からボール位置 $X(s)$ までの伝達関数 $G(s)$ を求めると式(3)となる。

$$G(s) = \frac{X(s)}{\Theta(s)} = \frac{g}{s^2} \quad (3)$$

なお、ビーム角度の制御では、ビームの回転軸にポテンシオメータを取り付け、ボール位置から決定される基準ビーム角と実際のビーム角度が比例するようにマイナーループを構成することとする。

式(3)で示したように、制御対象の伝達関数 $G(s)$ は二重積分を含んだ二次系となっている。この系を安定化するために式(4)の形式の PID 制御器を使用する。

$$C_{PID}(s) = K \cdot \frac{1 + aTs}{1 + Ts} \cdot \frac{1 + bUs}{1 + Us} \quad (4)$$

(K : ゲイン、 a, T : 位相進み補償のパラメータ、 b, U : 位相遅れ補償のパラメータ)

図4は、ボール&ビーム装置の安定化制御システムの構造をブロック図で示したものである。DC モータへの操作量(モータ駆動電圧)とビーム角度間の伝達関数 $P(s)$ は、式(5)で表される。

$$P(s) = \frac{Gm}{s} \quad (5)$$

(但し、 Gm はシーソー機構系で決まる定数である。)従って、図4に示すマイナーループを構成することにより、PID 制御器で構成したボール位置制御器の出力で

ある基準ビーム角と実際のビーム角との間の伝達関数を等価的に1にできる. なお, 式(5)に示すようにDCサーボモータへの操作量とビーム角度間の伝達関数は一形系であるから, このマイナーループ内のビーム角度制御器は比例制御器で十分である.

本制御系を安定にするパラメータの一例は, $a = 20, T = 0.71, b = 0.3, U = 5.82$ であった. この制御器の各パラメータは, 動作中にクライアントPCから変更できるようになっている. 従って, 実験時にはボール&ビーム機構の安定性などの挙動を観測しながら, 実機に最適なパラメータを設定したり, パラメータの影響を観測したりすることができる.

式(4)の連続系の制御器をC言語で記述するプログラムとしてサーバマイコンに実装するためには, デジタル制御器に再設計する必要がある. 離散化には, 式(6)で示す連続時間系との近似度に優れている双線形S-Z変換を使用した⁷⁾.

$$s = \frac{2}{T} \cdot \frac{1 - z^{-1}}{1 + z^{-1}} \quad (T: \text{サンプリング周期}) \quad (6)$$

式(6)を使用して式(4)をZ変換した伝達関数は, 式(7)の形式となる.

$$C_{PID}(z) = \frac{a_3 + a_2 z^{-1} + a_1 z^{-2}}{1 + a_5 z^{-1} + a_4 z^{-2}} \quad (7)$$

式(7)の形式から分かるように, デジタルPID制御器の制御演算は, IIR型のデジタルフィルタ演算と同じ形式になる. この制御演算の出力方程式は, 式(8)の差分方程式で表せる.

$$y(nT) = a_1 x(nT - 2T) + a_2 x(nT - T) + a_3 x(nT) - a_4 y(nT - 2T) - a_5 y(nT - T) \quad (8)$$

($y(nT)$: 制御器出力, $x(nT)$: 制御器入力)

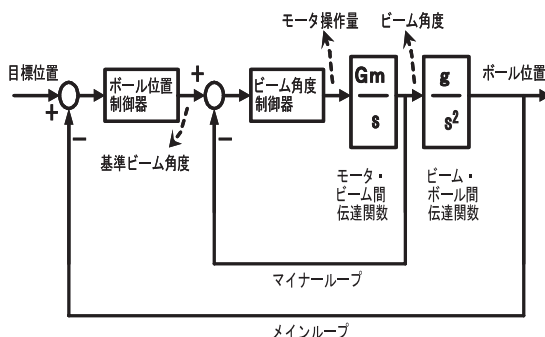


図4 ポール&ビーム装置安定化制御システムの構造

前述の制御器パラメータの一例から式(8)の各係数を求めると,

$$a_1 = 8.80, a_2 = -17.72, \\ a_3 = 8.92, a_4 = 0.97, a_5 = -1.97 \quad \text{となる.}$$

実験では, 制御対象のビームを剛体として取り扱える範囲とし, 機器仕様から制御帯域を1 Hz程度とした. また, 離散化による影響を少なくするため, サンプリング周波数を33 Hzに設定した. この時, ナイキスト周波数は約17 Hzとなり, この周波数付近で制御系の周波数特性は歪むことになる. しかし, 制御帯域は1 Hz程度であるため, この特性の歪みによる影響は, ほとんど考慮する必要はない⁸⁾.

3.2 サーバプログラム

OSはµITRON4.0仕様準拠のTOPPERS/JSPカーネル, プログラム開発言語はC言語, またCコンパイラはGNUの開発ツールを含むUNIXのさまざまなツールをWindows上で動作できるようにしたCygwinのgccを使用した.

サーバプログラムの処理はマルチタスクで行う. タスクは大きく分けて次の2つがある.

- ・遠隔制御タスク
- ・データ通信/デジタル制御タスク

遠隔制御タスクは最初に起動するタスクで, クライアントPCからの接続を待つ. クライアントPCから接続要求がくるとパスワードを要求し, それが正しいかを確認する. パスワードが正しい場合, クライアントに制御権があることを通知する. そして, クライアントが機器制御の開始を送信してくると, データ通信/デジタル制御タスクを起動する. それ以降は機器の停止命令とパラメータ設定命令を送信してくるのを待つ.

データ通信/デジタル制御タスクは, 遠隔制御タスクから起動させられる. 起動するとデータ通信用の通信路の接続を待つ. クライアントPCと接続したら, 機器の制御を開始して制御情報をクライアントPCに送信する. データ通信/デジタル制御タスクは, 機器の制御を行っているため, 必ず決まった周期で制御量(ビーム角度, ボール位置)のサンプリングをし, 操作量の出力をしなければならない. しかし, 上記2つのタスクは同じ優先度になっており, そのため通常一方の

タスクが実行されている場合は待ち状態を得て実行可能状態に移り、CPU が空くまで待ち続けてしまう。これでは正確な周期での制御情報をサンプリングすることができない。そこで強制的にタスクを切り替える `irotd_rdq` 命令を周期的に実行することで、2 つのタスクの処理数が均等になるようにした。

`μITRON` では、C 言語で作成するソースファイルは、一般的な `main` 関数から始まるわけではなく、タスクの概念がとられている。タスクの設定はシステムコンフィギュレーションファイルで行う。このファイルでアクティブに設定されているタスクは、AKI-H8 マイコンの電源が ON になった場合すぐに実行される。すなわちシステムは、システムコンフィギュレーションファイルより生成された初期化情報に基づいて初期化を実行する。組み込みシステムでは資源の動的生成を嫌うため、生成されることがわかっている資源をあらかじめ生成しておき ROM に書き込んでおくことが推奨されている。タスクやセマフォ、周期ハンドラ等がそれにあたる。そこで `μITRON4.0` 仕様ではこのシステム初期化処理の手順をモデル化している。システムコンフィギュレーションファイルでは静的 API を使用して記述していく。実際に使用した API とその機能を以下に記す。

・CRE_TSK

タスクの生成を行う。タスクの属性、拡張情報、起動番地、起動時優先度、スタック領域のサイズ、スタック領域の先頭番地を設定する。タスクの実行に必要なとなる。

・CRE_SEM

セマフォの生成を行う。セマフォ属性、資源数初期値、最大資源数を設定する。通信のノンブロッキングコールに使用する。

・CRE_CYC

周期ハンドラの生成を行う。周期ハンドラ属性、拡張情報、起動番地、起動周期、起動位相を設定する。タスク切り替え命令、`irotd_rdq` はここで実行する。

・DEF_EXC

CPU 例外ハンドラの定義を行う。CPU 例外ハンドラ属性、起動番地を設定する。

このようにして作成されたプログラムは、`μITRON` 下で実行されるが、実行環境としてデバッグモードとリリースモードが用意されている。デバッグモードでは、モニタプログラム上で実行プログラムを動作させることができる。この場合、RAM 上にプログラムを書き込むことになるので、何回でもプログラムの書き換

えと実行テストができる。AKI-H8 のフラッシュ ROM への書き込みは 100 回までしか保障されていないので、学生実験では、基本的に何回でもプログラムの書き直しのできるデバッグモードでの実行で開発を行うことになる。内蔵フラッシュ ROM へプログラムを書き込み実行する場合は、リリースモードにする。

プログラムは、RS232C 通信ポートを介して AKI-H8 マイコンへ書き込む。デバッグモード時では、RAM 領域へ書き込まれ、モニタプログラム上で作成したプログラムを実行する。この場合、実行プログラムは `μITRON` の下で動いているモニタプログラムの上で動作することになる。リリースモード時では、フラッシュ ROM 領域で書き込まれ、`μITRON` の下で作成したプログラムを実行する。デバッグモード時のクライアント PC 上の実行画面を図 5 に示す。

3.3 クライアントプログラム

クライアント PC 上のクライアントプログラムには以下の機能を必要とする。

- ・制御ポート、データ通信ポートの接続
- ・制御パラメータの設定、送信
- ・受信した制御情報のグラフ表示

クライアントプログラムの開発言語には Microsoft Visual C++ 6.0 を使用し、OS は Windows とした。本システムでは、サーバが AKI-H8 マイコンであるため、クライアント側に JavaApplet を使用するような本格的なサーバの機能を持たせることは難しい。この問題を回避するためにクライアント側のプログラムは Microsoft Visual C++ 6.0 で組み、各々のクライアント PC にクライアントで処理すべきプログラムを持たせる構成とした。

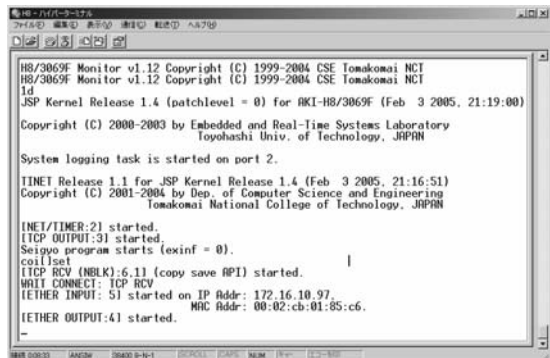


図 5 サーバ実行画面 (Windows のハイパーターミナルを使用)

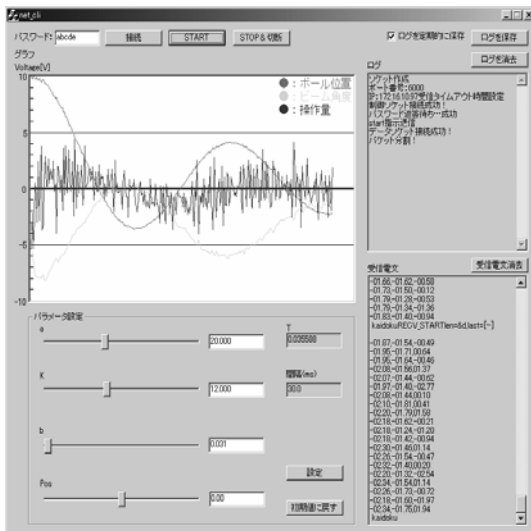


図6 クライアントPC上の実行画面

クライアントの実行画面は図6のようになる。制御実行結果として、ビーム角度、ボール位置、モータ操作量が表示されている。また、実行中にパラメータの設定を受け付けることができ、制御演算に反映できるようにになっている。

4. 動作評価

データ通信/デジタル制御タスクではボール位置、ビーム角度のサンプリング、モータ操作量の計算、操作量の出力、制御量の送信を行う。これらの処理を制御周期の30[ms]以内に行う必要があるが、実際はマルチタスクのため同時に遠隔制御タスクも実行されており、制御周期に影響を与える。デバッグモードにおいて、制御周期の一周期にかかる時間を測定すると図7のようになった。サンプル数は2000である。

図7に示すように一周期が30[ms]で実行されている場合が全体の3割、 30 ± 1 [ms]以内では7割となった。しかし、残りの3割は30[ms]を大きく超えている。これでは正確な制御周期が確保できないため、デジタル制御系を不安定とする要因となる。

前述のように、本システムは学生実験への適用を目的としているので何回もやり直しのできるデバッグモードでの実行を基本としている。デバッグモードでは、ROMにモニタプログラムを書き込み、実際のプログラムは書き込み回数の限りのないRAMに書き

込む。そしてモニタプログラム上で実際のプログラムを実行することになる。このため処理速度が遅くなることが考えられる。

処理時間を短縮するためにリリースモードでの実行を試みた。デバッグモードで実行していたプログラムをリリースモードで実行した結果を図8に示す。このように処理時間が短縮し、処理時間のばらつきも減った。制御周期にかかる時間は、30[ms]が全体の4割で、 30 ± 1 [ms]以内に実行周期の9割が収まった。

以上の動作評価より、学生実験では全体のプログラムはデバッグモードで開発を行い、最後の微調整をリリースモードで行えばよいと考えられる。

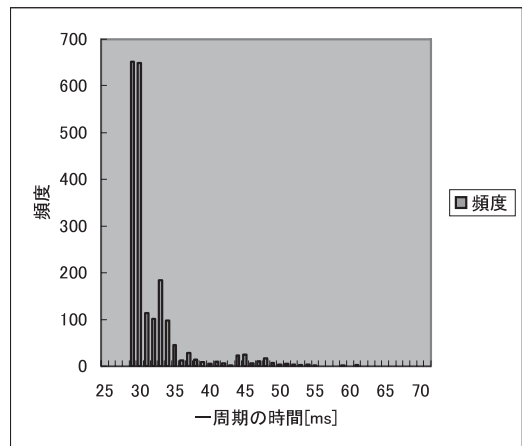


図7 デバッグモード実行時の一周期にかかる時間

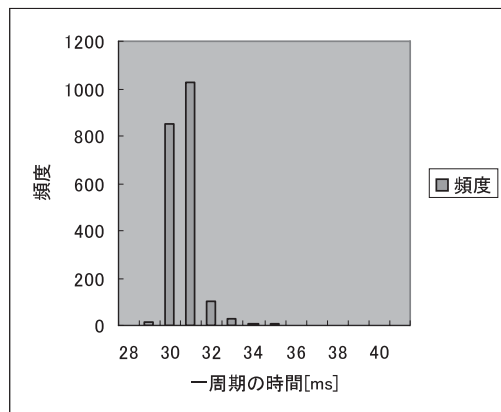


図8 リリースモード実行時の一周期にかかる時間

5. あとがき

本論文では、リアルタイム OS の μ ITRON を利用したデジタル制御と遠隔制御のための基礎実験システムの構成を報告した。この実験システムを利用することにより、ソフトウェアによりコンピュータによる機器制御やネットワークを利用した遠隔制御が実現できることが体験できる。これは、修学者にコンピュータ応用の実際的なイメージを把握させることに有効であると考えられる。

当然のことながら、この実験システムを理解し、プログラムを書けるようになるためには、Cプログラミング、ネットワーク通信工学、OS、制御など多くの事を先修科目として習得しておく必要がある。実際、今回のプログラムの多くは卒業研究生が作成したが、動作を理解して作成するのに半期程度の時間がかかっている。今後は、これらの技術を効率よく修得するために、授業プログラム全体をどのように構成するかを検討していく必要がある。

本論文での試みが、デジタル制御や遠隔制御技術を修得するための工学実験用システム構築に関心のある方々の参考になれば幸いである。

謝辞

本実験システムの開発にあたり、ご協力をいただいた、情報電子工学科第25期生の松田敏之君、第26期生の藤澤亨君に感謝いたします。

文献

- 1) 山田健仁, 百田正広: ネットワークを利用した遠隔制御システムの検討, 徳山高専研究紀要, 第25号, pp.7 - 12 (2001)
- 2) 岡, 小川: LANを利用した遠隔操作実験, 論文集「高専教育」, 第21号, pp.265-271 (1998)
- 3) CE106 ボール&ビーム制御実験装置取扱説明書, TecQuipement社 (1993)
- 4) 社団法人トロン協会編集, 「 μ ITRON4.0 標準ガイドブック」, パーソナルメディア (2001)
- 5) TOPPERS プロジェクト: <http://www.toppers.jp/>
- 6) ITRON TCP/IP API 仕様 (Ver.1.00.01) : <http://www.sakamura-lab.org/TRON/ITRON/SPEC/tcpip-j.html>
- 7) P. Katz: Digital control using microprocessors, Prentice-Hall International inc. (1981)
- 8) 中村, 香田, 久良: サーボ系による連続経路制御のサンプル値制御における適正サンプル周波数の決定法, 計測自動制御学会論文集, Vol.28, No.5, pp.649-651 (1992)

(2006. 9. 05 受理)