

多階層制御回路を用いた高速制御システム

前川裕明[†] 田中柳一[†] 関根優年[†]

[†] 東京農工大学 工学府

〒184-8588 東京都小金井市中町2-24-16

E-mail: †{maekawa,tanaka}@sekine-lab.ei.tuat.ac.jp, ††sekinem@cc.tuat.ac.jp

あらまし ロボットには人間のような運動能力を持たせるために数多くのアクチュエータやセンサが使用されている。それらのロボットの多くはソフトウェアのみで制御されているため、ロボット全体を管理するには多くの処理時間が必要となっている。そこで、時間や負荷のかかる信号の生成や、センサの読み取りといった処理を、ハードウェアに行わせることによって、ソフトウェアにかかる負荷を軽減させる。その制御システムとして、多階層制御回路を作製した。本稿では、ソフトウェア・ハードウェアによる信号生成方法について説明をする。そして、ソフトウェアによる制御と、ハードウェアのみの制御を行い、信号生成速度や反射動作の応答速度について検討する。

キーワード 階層制御, サーボモータ, FPGA

High speed control system using Multilevel control circuit

Hiroaki MAEKAWA[†], Ryuichi TANAKA[†], and Masatoshi SEKINE^{††}

[†] The Factory of Technorogy, Tokyo University of Agriculture and Technology
2-24-16 Naka-chou, Koganei-shi, Tokyo, 184-8588 Japan

E-mail: †{maekawa,tanaka}@sekine-lab.ei.tuat.ac.jp, ††sekinem@cc.tuat.ac.jp

Abstract The robot is implemented many actuators and many sensors to get the action like Human. Almost these robots is controlled by only software, but only software control needs much processing time to manage robot action. Then we make multilevel control circuit as control system to reduce software load by performing hardware to generate signals, to sense. In this paper, we explains the method of generating the signal with software and hardware. And the control with a software level is compared with the control with the only hardware level. Moreover, examine the signal generation speed and the response speed of reflex action.

Key words hierarchy control, servo motor, sensor, FPGA

1. はじめに

1.1 研究背景

現在、ロボットに関する研究分野では、人間の持つ多種多様な運動能力をロボットに持たせるために、様々な方法が考えられている。[1] 人間のような複雑な運動と同等な制御機能をロボットに持たせるには、人間の間接や感覚神経と同等の働きをするアクチュエータやセンサが数多く必要となる。

しかし、これらの機器を単体の制御ソフトウェアで行った場合、扱うアクチュエータやセンサの数が増加していくほど、制御信号を生成するためのタイマー割り込みが頻繁に行われることになり、処理時間の増加や、使用可能な I/O 数の制限といった問題から、複数のアクチュエータやセンサを高速に制御するのは困難となる。

そこで、タイマー割り込み等を多く利用するアクチュエータ

やセンサの制御信号の生成処理などを、並列計算処理が可能で単純計算を高速に行えるハードウェアに行わせることで、ソフトウェアへの負荷を軽減させることが可能となる。これによって、ソフトウェアでロボット全体の行動計画や動作命令等を行い、ハードウェアの制御信号を管理することが可能になる。

また、ハードウェアのみで、センサからの入力データを制御信号のデータにフィードバックさせることで、同時に複数のアクチュエータを高速に制御することが可能となる。

このように、ハードウェア、ソフトウェア各々の得意とする処理を分担させることによって、高速で負荷の少ない制御システムを構築することができると考えられる。

1.2 目的

パソコンをホストコンピュータとする、複数のサーボモータ、センサで構成されるロボット(図1)の運動制御を、ハードウェア、ソフトウェアを協調動作させたシステムを用いて行う。

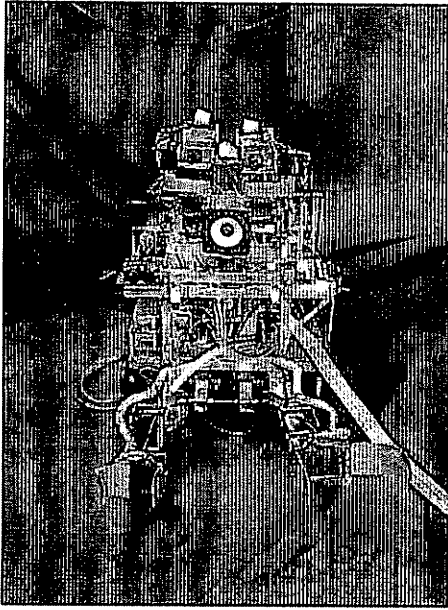


図1 統合用ロボット

このロボットには画像認識、音声認識、音声合成等の機能を統合させるため、ホストコンピュータにはそれらの負荷がかかってくる。そのため、ソフトウェアに負荷のかからないロボットの制御システムが必要となる。

その機能を実現するために、本研究室で開発したFPGAを搭載した実装基板 hwModule ボードを使用して、hw/sw 複合体のシステムを開発してきた。

人間の脳、運動野、脊髄といった階層的な運動制御構造のように、複数の階層からなる制御システムである多階層制御回路(図5)を構築した。

そして、hwModule によって構成される階層を hwObject 化したことによって、ソフトウェアからのアクセスが容易となった。

本研究では、多数の制御信号の同時生成処理を、ハードウェアとソフトウェアを協調動作させることによって、コンピュータのリソースを消費することなく実現することを目的とする。また、下位の層である末梢レベルにおいて反射動作を行わせるなど、階層的な制御構造を持たせることによって、より人間的な動作生成が可能な制御システムを目指す。

2. 再構成可能なシステム-hwModule-

本研究で想定した多階層制御回路による効率的な制御信号の生成を実現するために、本研究室で開発した再構成可能なLSIであるFPGAが搭載されたPCIデバイス、hwModule(ハードウェアモジュール、図2)を用いてシステムの構築を行った。

2.1 hwModule の構成

hwModule はFPGA、メモリ、PCIバスコントローラにより構成される。図3にhwModuleのブロック図を示す。

hwModule は、ボード制御のPCIバスコントローラ用FPGAを1個、仮想回路用FPGAを3個搭載している。更に、処理

MPU FPGA × 3

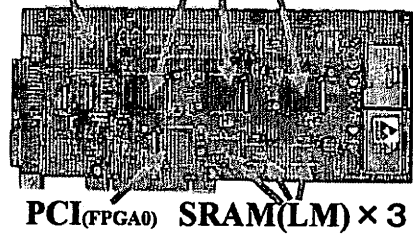


図2 hwModule

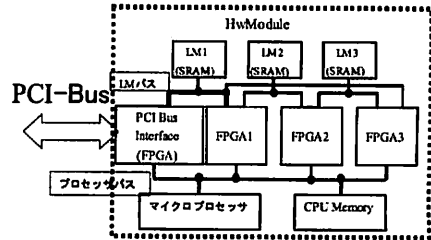


図3 hwModule の内部構成ブロック図

データを格納するためのSRAM(以下:LM(ローカルメモリ))を3個搭載している。各FPGAは2つのLMバスを持っている。また、隣り合うFPGA同士でLMを共有しているため、複数の仮想回路が動作する状況でもデータのコピーの必要がないという特徴を持つ。

2.2 hwObject

図4にhwObjectモデルの概要図を示す。hwObjectとは、複雑なハードウェアの処理をカプセル化したもので、hwModule上のFPGAに実装されるhwNetにより、目的の処理を行う。hwObjectに於けるHost上のメンバー関数は、対象の仮想回路とHostとの結合を解決して動作させるための手続きのみである。つまり、従来のC++言語によるソフトウェアの記述を用いて、容易にハードウェアが扱えるようになる。hwObjectには、ビット処理、信号処理など回路が得意とする機能を割り当てる。[2]

3. 多階層制御回路の構成

多階層制御回路は、ソフトウェアレベル、hwModuleレベル、末梢FPGAレベル、回路レベルの4つの階層に分かれている(図5)。

3.1 制御対象

図6に制御対象となるロボットのサーボモータの接続やセンサの位置を示す。

今回の実験では、左右の腕を構成するサーボモータ8個分の制御信号を出力する。

3.2 ソフトウェアレベル

ソフトウェアレベルは多階層制御回路の最上位階層である。この階層は、ロボットの行動計画及び動作命令を行う。また、センサデータのグラフ表示を行っている。

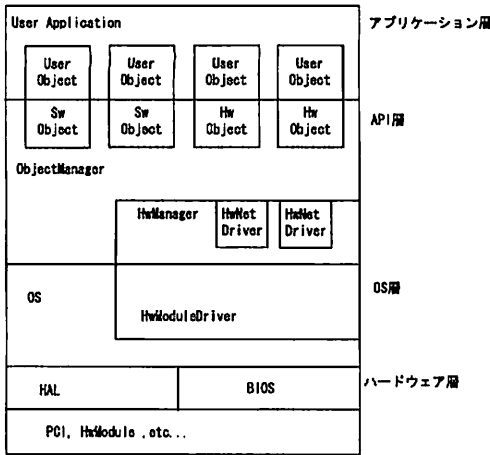


図 4 hwObject

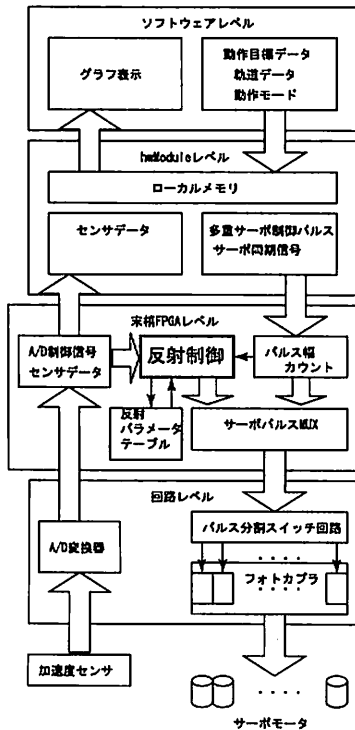


図 5 システム構成図

3.2.1 パターン動作

ロボットの行動にはいくつかの動作をパターンとしてホストコンピュータに保存しておき、操縦者がそのパターンの中から動作を選択し、実行する。

パターン動作には、大まかな動作目標を示すデータである MacroData と、目標までの細かな動作軌道を示す MicroData のふたつのデータを組み合わせて生成する。これらのデータはコンピュータ上のメモリーやディスクに記憶しておき、実行す

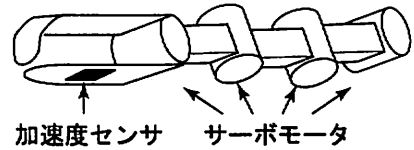


図 6 ロボットアームの構成

るパターン動作に応じて呼び出して利用する。

このように、2種類のデータを用いて制御データを生成することで、動作の開始時と終了時はゆっくり、中間ではすばやく移動するといった動作を、効率的に作り出すことができる。

3.2.2 MacroData と MicroData

・MacroData

動作の目標位置を示すデータであり、モータ 1 個につき 8bit の数値データで表される。今回は 8 個のモータを同時に動作させるため、MacroData を 8 個用意し、合計 64bit のデータとして用いる。

・MicroData

目標位置までの軌道を示すデータであり、モータ 1 個につき 1bit の正負と 3bit の数値のデータで表す。MicroData をモータの信号入力周期ごとに読み出し、現在の信号データに加算、減算を行うことで次の動作位置を示す信号データを生成する。

・MicroAddress

8bit のアドレスデータであり、パターン動作を実行する際に使用する MicroData の FPGA 内部の SRAM の開始アドレスを示す。

3.2.3 センサデータのグラフ表示

末梢 FPGA レベルで取得したセンサの情報が hwModule レベルの LM に入力されていくので、50ms ごとにソフトウェアレベルで読み込むように設定している。読み込んだデータはアプリケーション GUI 上にグラフ表示する。

3.2.4 hwModule レベルへの出力

呼び出された MicroData はあらかじめ hwModule レベルの LM へと保存しておく。MicroData は、MacroData に比べて短い時間間隔で呼び出され、同時に複数のサーボモータの制御データを計算する必要があるため、ソフトウェアから頻繁にアクセスをするよりも、ハードウェアで並列処理を行う方が効率が良いためである。そして、目標値である MacroData を動作開始信号とともに hwModule レベルへ出力し、動作を開始させる。

3.3 hwModule レベル

hwModule レベルはソフトウェアレベルの下位に位置し、hwModule ボード上の FPGA(Spartan2, XC2S200) に仮想回路として実装される。さらに、これを hwObject にすることでソフトウェアから容易にアクセスできるようにした。

hwModule レベルではソフトウェアレベルから入力された MacroData, MicroAddress から、サーボモータ制御パルスを生成する。また、末梢 FPGA レベルからシリアル通信で送信されてくるセンサデータを受け取り、ソフトウェアレベルへ渡すために LM へと書き込む処理も同時に行う。

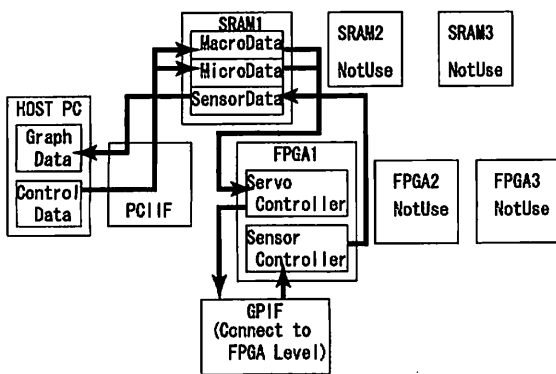


図 7 hwModule の回路配置とメモリの割り当て

3.3.1 内部構成

hwModule 上の回路配置を、図 7 に示す。FPGA1 にサーボモータの制御信号生成回路 (ServoController) と、センサデータを LM へ書き込む回路 (SensorController) をダウンロードする。

ローカルメモリの割り当ては、図 7 の上部の LM の部分に記載する。

上位のソフトウェアレベルとは PCIIF を通じて PCI バスによって接続されており、下位の末梢 FPGA レベルとは GPIF によって接続されている。

3.3.2 サーボモータ制御用パルス生成方法

サーボモータの制御用信号を生成するために、ソフトウェアレベルから MacroData, MicroAddress, Start 信号が入力される。そして、現在のサーボモータの回転角度を示すパルス幅データを保持するためのレジスタとして、PWMDATA を用意する。

PWMDATA は 8bit のレジスタであり、制御するモータと同じ数だけ用意する。本研究では 8 個のモータを使用するため 8x8bit のレジスタとなり、それぞれ 1 個のサーボモータの PWM パルス幅を保持する。このデータを信号生成部に入力し、カウンタを用いて計測することで、出力する PWM パルスの幅を増減させる。

パターン動作で使用する MicroData (軌道データ) は FPGA 内部の BlockRAM に保存されており、動作開始命令と共に入力する MicroAddress によって、読み込み開始位置を指定して読み込む。

1 回の MacroData による動作 (Macro 動作) では、目標位置である MacroData に対して、動作軌道を補完する MicroData は 8 回呼び出す。MicroData を 1 回読み出すたびに PWMDATA に加減算することで、次の軌道位置を示す PWMDATA として出力する。

また、サーボ制御パルス信号は時間多重化しており、8 個分のサーボモータの信号を制御信号、同期信号の 2 本の信号で出力する。

以上の信号生成の方法は、目標角度を入力することで、あらかじめ記憶されているパラメータから、制御信号を生成する逆モデル [3] による運動制御をハードウェアによって模擬した制御

| address | 32bit | | | |
|---------|-----------------|-----------------|-----------------|-----------------|
| 0x0000 | MacroData[0][3] | MacroData[0][2] | MacroData[0][1] | MacroData[0][0] |
| 0x0001 | MacroData[0][7] | MacroData[0][6] | MacroData[0][5] | MacroData[0][4] |
| 0x0002 | | | | MicroAddress |
| 0x0003 | MacroData[1][3] | MacroData[1][2] | MacroData[1][1] | MacroData[1][0] |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 0x0018 | MacroData[7][3] | MacroData[7][2] | MacroData[7][1] | MacroData[7][0] |
| 0x0019 | MacroData[7][7] | MacroData[7][6] | MacroData[7][5] | MacroData[7][4] |
| 0x0020 | | | | MicroAddress |

図 8 MacroData の SRAM 内メモリアドレス設定

方法である。

3.3.3 センサデータの管理

本実験では 8ch の 8bit A/D 変換器を使用しているため最大 8 個のアナログセンサの値を読み込むことができる。データは 4ch 分を順番に配置し、それぞれ 50 個保存する。

LM に各チャンネル 50 回ずつ保存が完了した後に終了信号をアサートし、ソフトウェアレベルにデータが入力し終わったことを知らせる。

そして、ソフトウェアレベルでデータの取得を完了すると、次のデータを入力し始めるようクリア信号が入力される。A/D 変換器は 100us で 1 回の変換が可能であり、8ch 全て読み込むために各チャンネル 1kHz のサンプリングとした。よって、各チャンネルごとに 50ms 分のデータを保存している。

3.4 末梢 FPGA レベル

末梢 FPGA レベルは、Spartan3 の XC3S400 を用いる。

この階層では、hwModule レベルから入力されるサーボ制御用パルス信号を回路レベルへと出力する際に、各サーボのパルス幅を測定する。このデータはセンサの情報から人間の反射動作のような制御を行う際に利用する。

さらに、この階層では回路レベルに実装される A/D 変換器の信号制御を行う。また、センサデータをパラレルシリアル変換を行い、hwModule レベルへとシリアル通信によって送信する。

3.4.1 反射動作

反射動作は人間の運動における脊髄反射を模擬したものである。ソフトウェアレベルを介することなく、末梢 FPGA レベルのみでサーボモータを制御することによって、高速な応答を行う。

3.4.2 反射パラメータテーブル

反射動作は、反射応答に使用するデータ (反射パラメータ) をあらかじめ FPGA 内部の BlockRAM に保存し、センサの入力からアドレスを算出し、該当する反射パラメータを読み出して制御信号を生成する。

3.4.3 反射動作の生成方法

反射動作を行うか否かの判定のために、反射パラメータテーブルに 8bit レジスタを付属させる。センサデータの値によって呼び出された反射パラメータがこの値を超えた場合、危険と判断し反射的に回避行動を行わせる。

反射制御信号の生成方法は、各センサに対応する反射パラメータテーブルから反射パラメータを取り出し、現在の制御パルス

表 1 HwNetCluster : 2s200fg456-6

| | 使用数 | 最大数 | 使用率% |
|----------------------------|------|------|------|
| Number of Slices | 1738 | 2352 | 73% |
| Number of Slice Flip Flops | 2110 | 4704 | 44% |
| Number of 4 input LUTs | 2901 | 4704 | 61% |
| Number of BRAMs | 6 | 14 | 42% |

表 2 末梢 FPGA レベル : 3s400tq144-4

| | 使用数 | 最大数 | 使用率% |
|----------------------------|------|------|------|
| Number of Slices | 1272 | 3584 | 35% |
| Number of Slice Flip Flops | 1448 | 7168 | 20% |
| Number of 4 input LUTs | 2017 | 7168 | 28% |
| Number of BRAMs | 3 | 16 | 18% |

幅を測定したデータに加減算を行い、パルス幅データとする。

3.4.4 センサデータの送信

送信するデータは、8ch の 8bit A/D 変換器を使用しているため、アドレス 3bit、データ 8bit に、スタートビット、エンドビットを追加した、合計 13bit のデータを、4MHz のデータクロックを用いて送信する。

3.5 回路レベル

回路レベルの構成は、CPLD、バスタランシーバ、フォトカプラ、A/D 変換器等によって構成されており、ハードウェアレベルから送られてくる信号の整形と分割、センサ出力の A/D 変換や、電圧レベルの変換、回路とサーボモータの電源分離等を行っている。

3.6 各ハードウェアレベルの回路規模

3.6.1 hwModule レベル

hwModule レベルの ServoController, SensorController 及び周辺回路 (LM や PCI への接続) を含めた、hwModule に搭載されている FPGA1 全体の回路 (HwNetCluster) の回路規模を表 1 に示す。また、hwModule は 33MHz のクロックで動作させる。

制御用のモジュールの回路規模は、ServoControl が 29%、SensorControl が 4% となっている。

3.6.2 末梢 FPGA レベルの回路規模

次に末梢 FPGA レベルの使用状況を表 2 に示す。FPGA の動作クロックは、hwModule レベルから入力される 8MHz のクロックを DCM を用いて 4 倍し、hwModule と同じ 33MHz にして動作させている。

4. アプリケーション

4.1 GUI

センサのグラフ表示及びサーボモータのコントロールを行うために、ソフトウェアレベルとなる GUI を持つアプリケーションを作成した。

サーボモータの制御は、トラックバーをスライドさせることによって 1 個ずつ動作させる方法と、パターン動作ボタンを使用することで保存されたパターンに従って動作させる方法、マップを使い、あらかじめ設定されている角度へ移動する方法がある。加速度センサからのデータはグラフに表示される (図 9)。

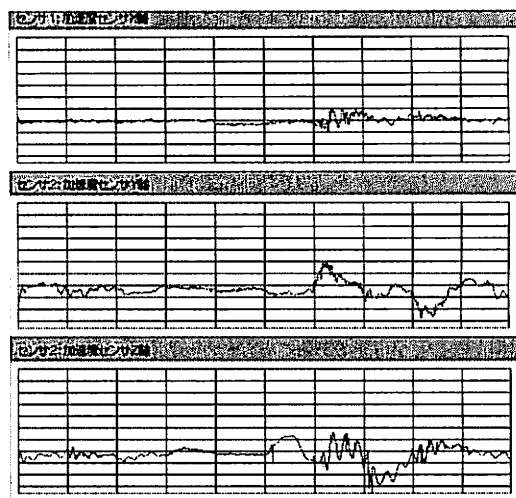


図 9 センサデータのグラフ

表 3 実験に使用した PC のスペック

| | |
|--------|-----------------------|
| OS | Windows 2000 SP4 |
| CPU | Intel Pentium4 3.0GHz |
| Memory | DDR SDRAM 3200 1GB |

表 4 実験に使用した PC のスペック

| | | |
|--------|------------------------|-------------|
| サーボモータ | KONDO | KRS-2346ICS |
| 加速度センサ | FreeScale | MMA7260Q |
| AD 変換器 | National Semiconductor | ADC0808 |

5. 制御システムの検証

多階層制御回路を用いた制御システムの動作検証を行った。検証ではロボット制御の要素部分である、ソフトウェアからの制御とハードウェアのみの反射制御に関して行った。

5.1 検証環境

検証環境は次のようになっている。

5.1.1 ホストコンピュータ

ソフトウェアレベルに使用するコンピュータの性能を以下に示す。

5.1.2 制御対象

実験用ロボットアームに使用した部品を表 4 に示す。

5.2 ソフトウェアレベルの動作速度

この階層の実験では、Windows 上のアプリケーションを用いて、動作命令を入力してから、動作が完了するまでの時間を計測した結果を示す。

5.2.1 MacroData を用いた動作速度

1 回の Macro 動作 (160ms) を行い、出力終了を知らせる信号が返ってくるまでの時間を計測した。

その結果、GUI の動作ボタンをクリックしてから終了信号が帰ってくるまでの時間は 170ms であった。hwObject の関数を呼び出してから、実行されるまでの遅延時間が 10ms かかっていた。

5.2.2 MicroDataTable の書き換え速度

ハードディスクに保存してある MicroData の配列 (32×256bit) をメインメモリーに読み出し、hwModule レベルの LM に書き込み、動作が終了するまでの時間を計測したところ、平均約 22ms であった。

ディスクからデータを読み込み、MicroDataTable の形式に変換するのに約 10ms かかり、hwObject の呼び出しに約 10ms、データの書き込みは 0.5ms 程度であった。

この実験から、1 回の Macro 動作の終わりに 40ms 程度の MicroDataTable を利用しない時間中に、内部のパラメータを書き換えることが可能であることが分かった。よって、多くの軌道パラメータをディスクに保存しておき、必要に応じて動作中に書き換えることが可能であると考えられる。

5.2.3 センサデータの読み込み速度

この実験では、50ms の間に取得するデータ (50 回 × 8ch) を hwModule の LM から読み出し、グラフに表示するのに必要な時間を計測した。

計測した結果は約 65ms であった。このうち、50ms はデータを溜め込む時間であり約 5ms はデータを読み出し、グラフを描画するまでの時間である。

5.2.4 ホストコンピュータにかかる負荷

ソフトウェアレベルにおける各実験中にかかる CPU の負荷は、MacroData を LM に書き込み、ロボットに動作させる命令を実行したとき、MicroDataTable の書き換えを実行した瞬間のみ 5~8% 程度の負荷が発生していた。

50ms ごとに実行されているセンサデータの取得と、グラフの描画による CPU 負荷は、処理開始の瞬間以降は 0% であった。

5.3 末梢 FPGA レベルにおける反射動作の応答速度

末梢 FPGA レベルでの反射動作の応答速度の実験結果について述べる。

5.3.1 シミュレーション結果

FPGA 内の仮想回路のシミュレーションでは、A/D 変換器からの変換終了信号入力から反射部へデータ出力までに 20 クロック (約 600ns)、反射部で計算を行い制御信号データが作成されるまでが 32 クロック (約 1000ns) 程度となっている。

よって、センサデータが入力されてから反射動作信号の生成までは約 1.6 μ s かかると考えられる。

5.3.2 実験結果

反射動作の実験は、静止しているロボットアームに外力を加え、回避行動を行うまでのセンサからのデータを、ソフトウェアレベルに表示されるグラフから読み取り、時間を計測した。

その結果、反射判定しきい値を超えたデータが入力されてから約 30ms~50ms 程度で、回避動作による加速度を検出していた。

シミュレーション結果に対して、この様に応答速度が遅い理由として、センサの入力に対する出力の遅延、サーボモータの信号入力から駆動までの遅延が考えられる。

サーボモータの制御パルスは、上位層から入力される制御信号とタイミングを同期させているため、反射動作を行うモータのパルス信号を出力し終わった直後に反射動作が実行される最悪のケースの場合には、20ms 後のパルスを待つ必要があり、応

答速度に 20ms 程度加算されるので 50ms になる。

5.4 ソフトウェアでの反射動作の予測値

これまでの実験結果から、ソフトウェアレベルで反射動作を行う場合、hwObject を呼び出すたびに 10ms の遅延が発生するため、最低でも 30ms 程度の遅れが必ず発生することになる。さらに、モータのパルス入力周期のタイミングにより最大 20ms の遅れが発生する。そして、センサとモータの応答速度から 30ms 程度の遅れが加わることが考えられる。そのため、末梢 FPGA レベルのみによる応答に比べ、最速でも 2 倍以上遅くなることが予想される。

5.5 人間の反射速度との比較

人間の手先における脊髄反射の応答速度は約 30ms といわれている [4]。

今回使用したサーボモータとセンサの構成では、最も早いタイミングでの応答速度が 30ms であり、これは人間の反射動作と同等の速度であるといえる。

6. むすび

本報告では、多階層制御回路、MacroData、MicroData を用いた動作軌道の生成、ハードウェアによる反射動作、それらを用いたロボット制御システムの提案と、その検証結果について述べた。

検証では、ソフトウェアとハードウェアの処理速度の比較を行った。そして、頻繁にアクセスが必要な処理や、高速度が求められる処理をハードウェアで行わせることで、ソフトウェアに殆ど負荷がかからないシステムが構築できた。

また、反射動作の検証結果から、末梢 FPGA レベルのみで反射を行わせることによって、人間の脊髄反射と同程度の応答速度が得られることがわかった。

今後は、ソフトウェアレベルで動作軌道データを作成し、ロボットを運動させながら次々に MicroDataTable を書き換えていくことで、様々な運動をスムーズに行えるようなシステムを目指す。

7. 謝 辞

大学発ベンチャー創出推進「1811 動的な仮想回路による超高速 Hw/Sw 複合システムの研究開発」の資金による

文 献

- [1] インテリジェンス・ダイナミクス 1 脳・身体性・ロボット知能の創発をめざして、土井利忠、藤田雅博、下村秀樹編、シュプリング・フェアラーク東京株式会社
- [2] 工藤健慈、今中昭記、志賀裕介、関根優年、" hw/sw 混載システムにおける hwObject モデルとその制御手法"、情報科学技術フォーラム一般講演論文集第 1 分冊 FT2002、C-3, pp.193-194, 2002/9
- [3] 脳の計算理論、川入光男、産業図書
- [4] Matthews PB, Farmer SF and Ingram DA (1990) On the localization of the stretch reflex of intrinsic hand muscles in a patient with mirror movements. J Physiol 428: 561-577.