

## 1. はじめに

パルス列入力形サーボモータは、その制御のしやすさから頻りに利用されるが、用いるモータの個数とその用途に合ったコントローラを用意しなければならない。またコントローラの制御できるモータ数(軸数)は一般に固定されており、モータの追加などの拡張性や他のシステムへの転用などの柔軟性に乏しい。

本報告では、通信機能を持った中粒度並列処理プロセッサであるトランスピュータを用いて柔軟性に富んだシステム構築ができるモーションコントローラの構成を示す。

## 2. システムの基本構成

トランスピュータは、高速シリアル転送(20Mbit/s)が可能なリンク(外部チャンネル)を4本を持った並列処理用プロセッサである。複数のトランスピュータは、このリンクを用いて多様な形態に容易に接続できる。プロセッサ間の同期はこのリンクを介したメッセージバッシングにより取られる。1台のトランスピュータで実行されるプログラムは一般に複数のプロセスから構成され、これらを並列に処理する事ができる。プロセス間はリンクと同様に内部チャンネルを介したメッセージバッシングにより同期が取られる。

このトランスピュータとその専用言語であるOCCAM2を用いて開発したモーションコントローラの基本構成を図1に示す。ルートコントローラ(以下RC)には、各サブコントローラ(以下SC)に接続されているモータ毎にその動作シーケンスを一つのプロセスとして記述しモータの数だけ並列に起動する。各プロセスから送出されるSCへのコマンドは、リンクを介して転送される。SCはすべて同じハードウェア構成であり、受けたコマンドが自分のものであればコマンドに従った駆動パルス列を自分のモータドライバへ出力する。もし他のSCのコマンドであれば次のSCへ転送する。RC-SC間およびSC-SC間はリンク接続であり組み込まれる対象システムに応じ必要な台数のSCを用意すれば簡単に接続でき、またモータの追加などの変更に柔軟に対処できる。

## 2.1 サブコントローラ

モータドライバパルス列を発生するSCは、トランスピュータ(T222、16bit、20MHz)とパルス発生用IC(PLC-3AM)から構成されている(図2)。このICは、パルス列入力形サーボモータのプリセット動作、連続動作、スローアップ・ダウンなど通常必要とする動作ができる高速パルスジェネレータ(MAX3MHz)である。専用のパルス発生用IC(以下PLC)を設けることによりT222の負荷の軽減を図った。トランスピュータとPLCは、バス接続であり、PLCへのコマンドコードはアドレスバスにより、またコマンドパラメータはデータバスによりそれぞれ設定する。また、コマンドは必要とするパラメータのバイト数により5種類に大別できる。T222は、RCからリンクを介して送られてきたこれらのコマンドが自分のものかどうか判別し、自分のコマンドであればそれを取り込みアドレスバスとデータバスを所定の値に設定しPLCを制御する。もし他のコマンドであれば次のSCに転送する。リンクのプロトコルは、コマンドの他にコマンドの識別番号としてSCのボード番号を付加し、図

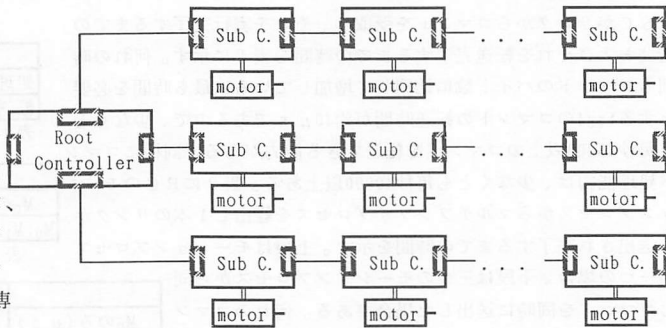


図1 基本構成

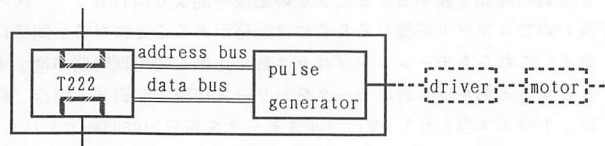


図2 サブコントローラ

からリンクを介して送られてきたこれらのコマンドが自分のものかどうか判別し、自分のコマンドであればそれを取り込みアドレスバスとデータバスを所定の値に設定しPLCを制御する。もし他のコマンドであれば次のSCに転送する。リンクのプロトコルは、コマンドの他にコマンドの識別番号としてSCのボード番号を付加し、図

3に示すように6種類設けた。SCはこのリンクにより任意台数のシリーズ接続が可能であり、コマンドの通信部とパルス発生部をそれぞれT222とPCLに分担させることにより、高速にパルスを発生中でもコマンドの授受や転送を可能にした。

## 2.2 ルートコントローラ

RCのトランスビュータは、接続されている全てのモータへのコマンドを発生しなければならず負荷の集中が想定されるのでT800(32bit、20MHz)を採用した。リンク4本の内の1本は、ホストコンピュータ(PC-9801)との入出力に使い、残りの3本のリンクにSCが接続される。図4に示すように、このトランスビュータは、各モータの動作を記述した各モーションプロセスとこれらから送出される各モータへのコマンドを適時、リンクへ出力するマルチプレクサプロセスの二種類のプロセスを実行する。

モーションプロセス内にモータの動作を直接PCLのコマンドコードで記述することは、多大な労力を必要とするので、PCLのコマンドを発生するプロシージャを開発しライブラリ化した。各プロシージャは図5に例を示すように利用者が理解しやすい名称で作られており、各モータの動作記述は、各モーションプロセス毎に該当モータの動作シーケンスに従って単にこのライブラリを利用するだけになる。プロシージャはPCLのコマンドを発生しボード番号と共にマルチプレクサプロセスへ送出する。マルチプレクサプロセスは送られてくるコマンドを早い者順にリンクへ出力する。これらのプロセスはリンク毎に同様に設けられ、全て並列に処理される。

## 3. 処理能力

SCがリンクからコマンドを受取り、それを実行完了するまでの時間およびそれを転送完了するまでの時間を表1に示す。何れの時間もコマンドのバイト数に比例して増加している。最も時間を必要とするtag4のコマンドの転送時間が約40 $\mu$ sであるので、少なくとも毎秒25000以上のコマンドを転送できる能力がある。同様にコマンド処理能力は、少なくとも毎秒40000以上ある。表2にRCのモーションプロセスからマルチプレクサプロセスを経由し1本のリンクから送出され完了するまでの時間を示す。上段はモーションプロセスが一つの場合、下段は三つのモーションプロセスから同じコマンドを同時に送出した場合である。何れのコマンドもモーションプロセスが三つの場合は、一つの場合の3倍弱の時間を要する。またSCの処理時間より何れも

長いのでコマンドを受けるSCには余裕があることになる。同様に表3はRCのプロシージャの処理時間の例である。これらもモーションプロセス数に比例して処理時間が増加する。各プロシージャの処理時間の最大を83 $\mu$ sとすると例えば3台のモータをシリーズ接続で制御する場合、最大遅れ時間は約250 $\mu$ sとなる。この時間は、1パルス当たり3.6度のパルスモータを毎分2400回転で1パルス以内に同時に制御できる時間である。

## 4. おわりに

通信機能を持った並列処理用プロセッサであるトランスビュータを用いて必要なモータ数に合わせて柔軟にシステム構築ができるモーションコントローラを開発した。

現在、モータの動作記述用プロシージャの標準化やフィードバック信号の取り込み等を検討中である。

### PROTOCOL CMDSTREAM CASE

```
tag0;boardnum
tag1;boardnum;cmdcode
tag2;boardnum;cmdcode;ldata
tag3;boardnum;cmdcode;ldata;mdata
tag4;boardnum;cmdcode;ldata;mdata;hdata
tag5;boardnum;cmdcode;hdata
```

図3 リンクのプロトコル

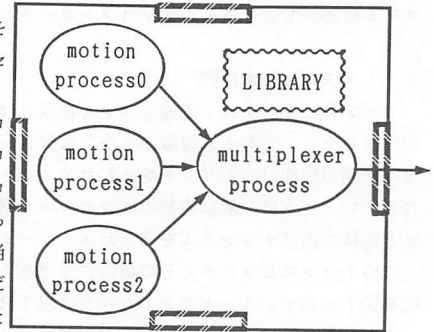


図4 ルートコントローラ

```
setfh(linknum, boardnum, counter2)
start. fh(linknum, boardnum, direction)
stop. fl(linknum, boardnum, direction)
change. fl(linknum, boardnum, direction)
```

図5 プロシージャの例

	tag1	tag2	tag3	tag4
処理時間( $\mu$ s)	11.1	15.3	19.7	23.5
転送時間( $\mu$ s)	19.9	26.9	35.5	40.8

表1 サブコントローラの処理能力

	tag1	tag2	tag3	tag4
M <sub>0</sub> のみ( $\mu$ s)	33.3	41.6	46.1	54.3
M <sub>0</sub> , M <sub>1</sub> , M <sub>2</sub> ( $\mu$ s)	98.0	120.9	135.0	156.8

表2 ルートコントローラのコマンド送出時間

	setR7	set. contmode	change. slowfl
M <sub>0</sub> のみ( $\mu$ s)	66.6	70.7	41.8
M <sub>0</sub> , M <sub>1</sub> , M <sub>2</sub> ( $\mu$ s)	194.5	207.8	120.3

表3 ルートコントローラのプロシージャの処理時間