

別刷

Journal of the Robotics Society of Japan

# 日本ロボット学会誌

Vol.12 No.8 1994

可搬式汎用知能アーム登場！

—オープンロボットの提案—

Mitsubishi Portable Manipulator DEBUT!  
—A Proposal of Open-Architecture Robot System—

大西典子 大西 献



社団法人 日本ロボット学会

## 解説

## 可搬式汎用知能アーム登場！

## —オープンロボットの提案—

Mitsubishi Portable Manipulator DEBUT!  
—A Proposal of Open-Architecture Robot System—

大西典子\* 大西 献\* \*三菱重工業(株)神戸造船所

Noriko Ohnishi\* and Ken Ohnishi\* \*Mitsubishi Heavy Industries, Ltd. KOBE Shipyard &amp; Machinery Works

## 1. はじめに

## 1.1 産業用ロボット開発者の悩み

ロボットが産業用として世間に出回るようになって30(?)年、1つの独立した位置決め装置だったものが、色々な周辺機能を我が身に取り込み、溶接・塗装・組立といった専用機械として分化・進化していった。

ところが産業界の進化も負けてはおらず、そのとどまる所を知らないニーズは、産業用ロボットの備えた機能だけでは満たされなくなってきている。が、これらのニーズにいちいち応えた専用ロボットを用意するのは労多くして・・・

それでは産業界は、どのような機能を産業用ロボットに望んでいるのだろうか？

## 1.2 産業用ロボット開発者のひらめき

産業用ロボットは、今やシステムの中の1パーツとして働くことを期待され、その機能を考える場合、システムを全体的な視野で眺める必要がある。すると、産業用ロボットの機能アップを図る方法としては

## ①ロボット自身の機能を直接アップする方法

例：ロボット可動範囲アップ

例：ロボット自由度アップ

だけでなく、他に

## ②周辺装置の融合により全体機能をアップする方法

例：ロボットのシステムアップ

## ③ロボットの使用が周辺装置にインパクトを与えず

相対的に全体機能がアップされる方法

例：ロボットの軽量化

例：ロボットの耐環境性能アップ

も考えることができる。

今回これら各種手法により機能アップを図った産業用ロボット「可搬式汎用知能アーム PA 10」を販売するに至った。ロボットそのもの+ロボット組込システムの設計者と

原稿受付 1994年8月8日

キーワード：Robot, Portable, Manipulator, Open-System

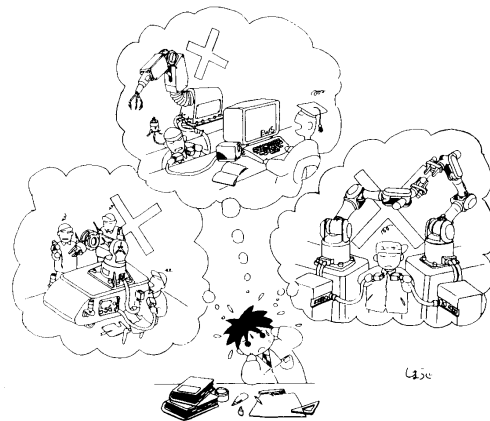
\*〒652 神戸市兵庫区和田崎町1-1-1

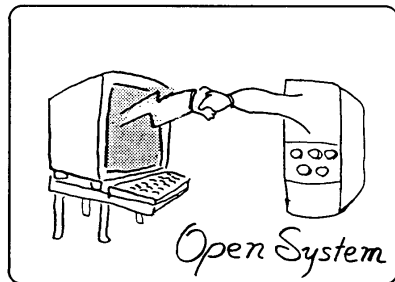
\*1-1, Wadasaki-cho 1-chome, Hyogo-ku, KOBE 652 JAPAN

として、自分たちで最も使い勝手のいい“つぶしのきく”物に仕上げたつもりである。その全体機能については、第4章の概略または詳細別資料[1]~[4]を参照頂くとして、本解説では特に上記②の手法に注目し、ロボットコントローラの構成と使い方を中心にご紹介させて頂く。

## 2. ロボットのシステムアップ

システムアップと言っても、そのニーズたるや千差万別、少なくともロボット言語とRS 232C、数点のデジタルI/Oだけでは荷が重そうである。そこでロボットを階層化、各層間のインタフェースを公開し、どこからでもロボットにアクセスできるようにする、即ちオープンなシステム構成にすれば、この入り組んだニーズに応えられるのではないかと無責任に考える。ところがそのインタフェース仕様は、ハードウェア的にもソフトウェア的にも必要かつ十分に簡素化されたものでなければ、本当に使い勝手の良いシステムにはならない。





### 3. 可搬式汎用知能アーム PA-10 のオープンシステム

#### 3.1 第1階層 ロボット本体

オープンなロボットシステムの最も原始的(?)な層を構成するのがロボット本体ハードウェア(いわゆるメカ)である。耐環境性を考慮し、ACサーボモータ・電磁ブレーキ・位置検出器(ブラシレスレゾルバ)の最もシンプルな構成で、冗長自由度を有する7関節アームとした(図1)。

#### 3.2 第2階層 サーボドライバ

第1階層ロボット本体の各関節を速度/トルク制御する階層であり、我々メーカーとしてもこれに適した2軸1体型サーボドライバを用意した。

##### 1) サーボドライバへのアクセスハードウェア

サーボドライバと上位コントローラとは、アークネット(ARCNET=データポイント社の登録商標)トークンバス方式LANで接続し、上位コントローラ1つ、サーボドライバ4つ、計5つのノード間で通信する。

アークネット接続の利点は

- ①十分な通信速度(5[Mbps])を有する
- ②接続が同軸ケーブルまたはツイストペア線1本で済む
- ③通信ボード(VME, PC 98用)が市販されている

アークネットの採用により、ユーザのワークステーションからでも、ケーブル1本でアームのすべての機能・情報がコントロールできる(図2)。

さらに高速な指令を必要とする場合には、コントローラ側に複数個のノード(=複数枚の通信ボード)を用意し、サーボドライバと1対1で接続すればよい。1つのサーボドライバに対する通信時間は約0.2[msec]である。

##### 2) サーボドライバへのアクセスソフトウェア

サーボドライバ/上位コントローラ間の情報は、すべて“コマンド+パラメータ”プロトコルで通信される。例えば通常制御中は、“C”コマンドと共に、上位コントローラから各関節への指令(速度または電流)指令が、逆にサーボドライバから各関節情報(関節角度、モータ電流値およびエラー情報等)が双方向に通信される。

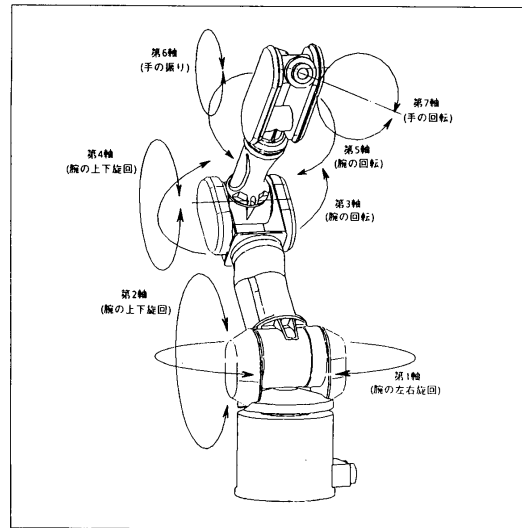


図1 ロボット本体

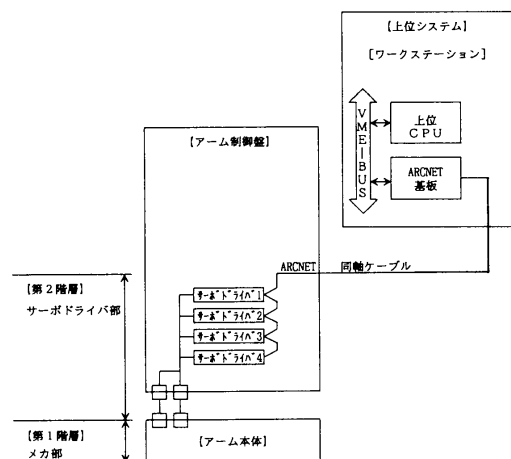


図2 サーボドライバへのアクセス

サーボドライバだけでも独立した階層を構成できるよう、サーボドライバ自身がアーム固有の情報(関節角度リミタなど)を管理できる設計とした。

#### 3.3 第3階層 運動制御コントローラ

運動制御コントローラは、逆キネマティクスや軌道補間計算を行い第2階層(サーボドライバ)への指令を生成する最もCPUパフォーマンスを要する階層である。我々メーカーも第3階層用に汎用のCPU(i80486)とDSPを用い、サンプリングタイム10[msec]を実現するVMEバス/ダブルハイトボードを用意した。運動制御コントローラ

表1 運動制御コントローラのイベント例

ParamSelEvent	8	上位計算機(マンマシン)から制御パラメータのダウンロードを行う
PBTopStartEvent	14	順方向に連続でプレイバックを行う。
ReturnFPEvent	60	基本姿勢に各軸制御で復帰する。
RMRCStartEvent1	131	ロボット座標系での位置偏差 [mm] ( $\Delta X, \Delta Y, \Delta Z$ ) が指定され、RMR C制御で動作する。(動作速度はパラメータで指定し、台形速度パターンを持つ)
RMRCStartEvent2	132	手先座標系での位置偏差 [mm] ( $\Delta X, \Delta Y, \Delta Z$ ) が指定され、RMR C制御で動作する。(動作速度はパラメータで指定)

表2 ライブラリの内部構造(例)

```

/*****
RMRC: 0k ッ座標 位置偏差動作
引数 armno 7-ムNo.
X 0k ッ座標 x方向 位置偏差
Y 0k ッ座標 y方向 位置偏差
Z 0k ッ座標 z方向 位置偏差
wfp 動作が完了まで待つかどうかの指定

戻値 ERR_OK 正常終了
ERR_ARM 7-ムNo. が間違っている。
ERR_INT 割り込みが受信されなかった。
以上の他、7-ムからの15-があります。
*****/
ERR_pa_mov_XYZ (ARM armno, REAL32 X, REAL32 Y, REAL32 Z, WFP wfp)
{
ERR err;
if (ERR_OK != (err=pa_map_ctl (armno))) return err;
_comuni->hd.Header.Event =RMRCStartEvent1;
_comuni->hd.Header.ComMode =1;
_comuni->hd.Header.Dummy[0] =2;
_comuni->hd.HighDown.Down.Axis =0x7f;
_comuni->hd.HighDown.Down.Position[0]=;
_comuni->hd.HighDown.Down.Position[1]=;
_comuni->hd.HighDown.Down.Position[2]=;

if (ERR_OK != (err=pa_req_ctl (armno, RETRY_COUNT)) ) return err;
pa_exe_wfp (armno, wfp);
return pa_get_err ();
}
    
```

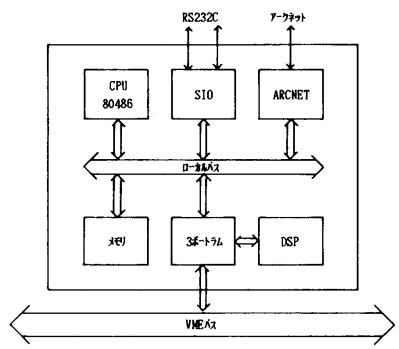


図3 運動制御コントローラの構成

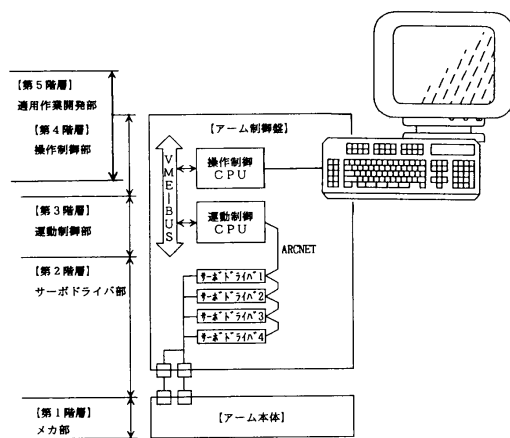


図5 操作制御コントローラへのアクセス

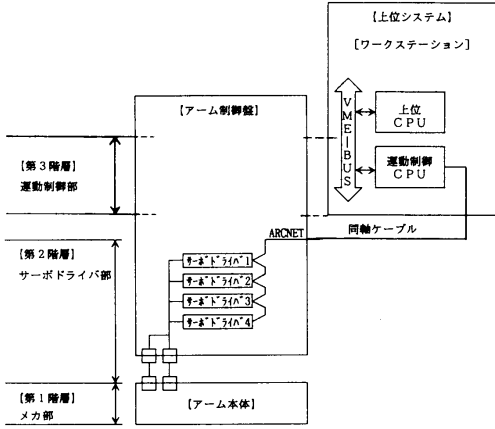


図4 運動制御コントローラへのアクセス

ラの概略ブロックを図3に示す。

- 1) 運動制御コントローラへのアクセスソフトウェア
 

運動制御コントローラへの指令(イベント)は、VMEバス上のアドレス空間に割り当てられた3ポートメモリへの“イベントナンバ+パラメータ”書き込み+割込み発生に

よってなされる。イベントはロボットの動作に関わるもの他に、パラメータやデータ操作・エラー情報に関するものなど多岐に渡っており、このイベントを直接操作すれば、プレイバック動作とプログラミング動作を混在させたり、外部操作によるアーム動作を加えたりといった自由な制御が可能となる。イベントの一例を表1に示す。

- 2) 運動制御コントローラへのアクセスハードウェア
 

標準バスの採用により、ロボットコントローラ内のVMEバスに市販のCPUボードを追加したり、運動制御コントローラをロボット制御盤からユーザのワークステーション空スロットに移設してロボットをコントロールすることができる(図4)。この構成は複数台のロボットにも1台のワークステーションで対応でき、アークネットにより接続は容易である。

### 3.4 第4階層 操作制御コントローラ

操作制御コントローラは、マンマシン I/F と第3階層への指令（イベント発生）を行う階層である。我々メーカーでは第3階層と同じVMEバス/ダブルハイトボードのAT互換DOS/Vマシンを用意した。それ自身がパーソナルコンピュータである利点を生かすなら、操作制御コントローラに直接キーボードとCRTを接続したシステム（図5）が最もコンパクト。操作制御コントローラ上のマンマシン

表3 ライブラリを用いたプログラミング例

```

/*-----*/
/*      CtlArm -- メイン処理      */
/*-----*/
int CtlArm(ARM arm)
{
    ERR  err;
    int  k;
    int  doExecute=1;

    while (doExecute) {
        if (kbhit()) {
            if ((err=pa_exe_hom(arm, WM_WAIT)) <= ERR_STOPARM)
                ErrStop(arm, err);
            pa_stp_arm(arm, WM_WAIT);
            return(1);
        }
        if ((err=pa_exe_hom(arm, WM_WAIT)) <= ERR_STOPARM)
            ErrStop(arm, err);
        if ((err=pa_exe_saf(arm, WM_WAIT)) <= ERR_STOPARM)
            ErrStop(arm, err);
        if ((err=pa_mov_XYZ(arm, 0.0, 400.0, 0.0, WM_WAIT)) <= ERR_STOPARM)
            ErrStop(arm, err);
    }
}

```

ンソフトウェアを通してのロボット操縦はもちろん、若干のメモリの増設とソフトウェアのインストールによって、それ自身をアプリケーション開発環境にも利用することができる。

実は、操作制御コントローラにRS 232Cを通してティーチングボックスを接続したのが、一般的な産業用ロボットと同じシステム形態である。

### 3.5 第5階層 アプリケーション開発部

アプリケーションの開発は、ユーザのDOS/Vマシン上でも行うことができる。ロボット言語ライクなC言語ライブラリを用意したので、これを利用してロボットの動作をオフライン記述してもよい。ライブラリの一例を表2に示す。ライブラリは第3階層へのイベント発生+エラー/タイミング処理のC言語で書かれた関数であり、ソースも公開しているので、第4階層プログラミングのサンプルとしても利用できる。ライブラリを用いたプログラミング例を表3に示す。標準のC言語で、ロボット動作以外のプログラミングも自在。オフライン開発した実行形式ファイルやデータはフロッピーディスク/ICカードで操作制御コントローラへ渡して実行する（図6）。

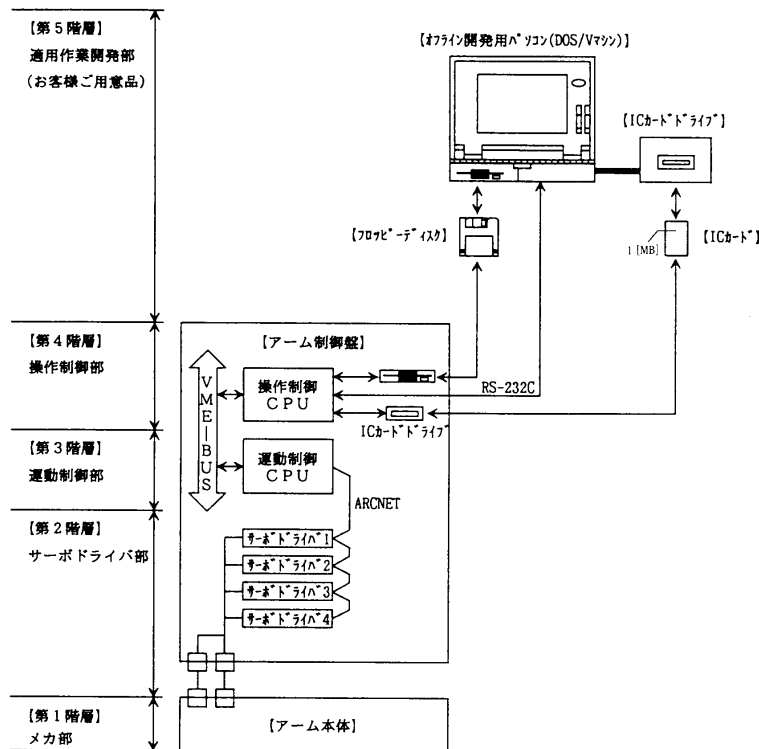


図6 フルシステム構成

#### 4. おわりに

以上のようにPA-10のコントローラは、その形態を問わない。一例として我々メーカーが標準で用意したものを図7に示す。

電源+第4階層までが専用ラックにコンパクト収納でき、どこへでも持ち運んで使用できるよう、専用のキャスタを用意した

最後にPA-10の持つその他の特徴について紹介させて頂いて本解説を終えたい。PA-10の全体外観・仕様を

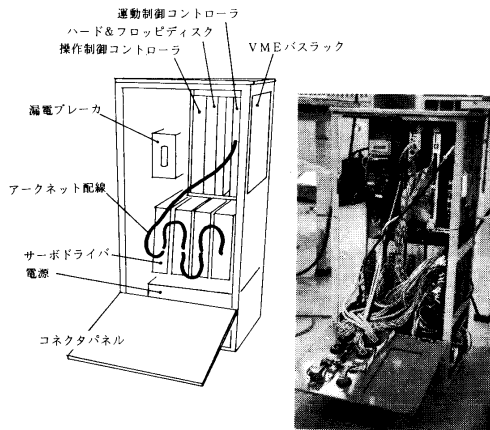


図7 標準コントローラ

図8・表4に示す。冗長自由度を持つ7軸構成、可搬重量10[Kgf]/自重35[Kgf]の超軽量設計、人間とほぼ同じリーチと防塵・防滴性能。これらはすべて、土木・建築・造

表4 仕様

形式	PA 10
アーム本体重量	323 [N] (33 [kgf])
先端取扱重量	98 [N] (10 [kgf])
関節数	7
駆動方式	DC ブラシレスサーボモータ
手先合成最大速度	1550 [mm/sec]
位置繰り返し精度	±0.1 [mm]
アーム長	950 [mm] (関節間距離)
構造	防塵、防滴多関節型
コントローラ重量	294 [N] (30 [kgf])
経路制御方式	PTP(点間)制御、直線補間、円弧補間 CP(連続軌跡)制御
システム構成	オープンアーキテクチャ
電源	AC 220 [V] ±10% 50/60 [Hz] 3 [kVA] 以下
周囲温度	0°C~50°C(アーム本体)

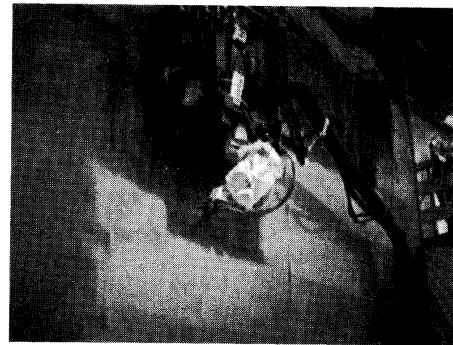


図9 壁面台車搭載による塗装作業

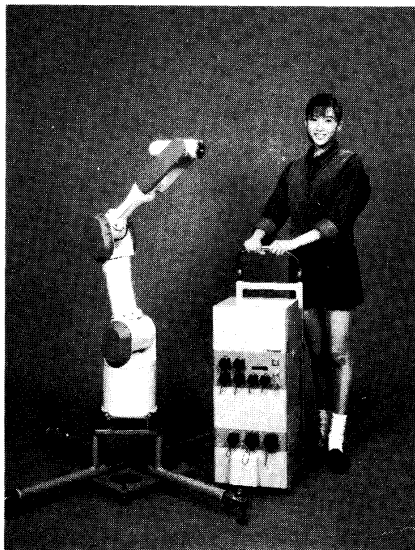


図8 可搬式汎用知能アーム PA-10

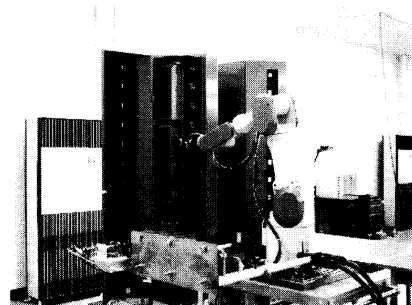


図10 視覚センサによる盤の組立作業

船など、従来マニピュレータを持ち込みにくかった現場作業へのニーズから生まれた。

軽さを生かして台車搭載 (図 9)、オープンシステムを利用して周辺装置とシステムアップ (図 10) や研究用にと、産業用ロボットにはまだまだ色々な使い道があるように思える。

## 5. 余 談

PA-10 のコントローラをオープン化して市販しようと計画してから約 2 年、エラー処理やインタロックの壁に何度となく阻まれながらようやくここまでこぎつけた感がある。が、よくよく評価してみると、セミ (?) オープンシステムである。途中の階層だけを抜き出すことができず、

下位の階層から積み上げた上位側のみオープン可能なシステムなのである。これらオープンのは非はもちろん、冗長軸のより効果的な利用方法など、読者の皆様の聡明なご意見をお待ちし申し上げている次第である。

## 参 考 文 献

- [1] 時岡：“可搬式汎用知能アームについて”，電気学会産業応用部門全国大会，1993.
- [2] 大西：“New Industrial World created by Portable Manipulator.” 24 th ISIR, 1993.
- [3] 濱田他：“可搬式汎用知能アームの開発”，自動化技術第 26 巻第 1 号，1994.
- [4] 大西：“三菱重工の提案する汎用ロボットの新しい展開”，日本産業用ロボット工業会誌，no. 98, 1994.



大西典子 (Noriko Ohnishi)

1962 年 1 月 30 日生。1984 年より三菱重工業 (株) 神戸造船所にて原子力プラント自動検査装置等の開発に従事。1986 年 JPDR 廃炉解体ロボット，1990 年新型原子炉容器探傷ロボット (A-UT マシン) などの主としてソフトウェア設計・製作に携わる。



大西 献 (Ken Ohnishi)

1962 年 10 月 30 日生。1986 年東京大学工学部精密機械工学科卒。同年 4 月より三菱重工業 (株) 神戸造船所にて原子力プラント自動検査装置等の開発に従事。1987 年通産省大型プロジェクト「極限作業ロボット」多本指双腕マニピュレータの開発に携わる。日本ロボット工業会，バイオメカニズム学会会員。  
(日本ロボット学会正会員)