

真空ロボットへのオフラインプログラミングとロボット
言語の適用について

横山和彦・野林繁美，脇迫仁，佐藤忍
(株)安川電機製作所

真空中で動作する産業用ロボットをこのほど開発した。このロボットは、4軸構成の水平多関節構造で、真空内自動化システムをフレキシブルに構築することができる。

本稿では、真空ロボット制御に関し、主として、オフラインプログラミングシステムと、ロボット言語の適用について述べる。オフラインプログラミングは、真空という特殊な環境でのロボットの使用目的によく適合し、ロボット言語(拡張Pascal)は、作業記述性にすぐれ、広範なアプリケーションへの適応性を高める効果がある。

Application of Off-line Programming and Robot Language for Vacuum Robot.

Kazuhiko YOKOYAMA, Shigemi NOBAYASHI, Hitoshi WAKIZAKO, Shinobu SATO

Robot division, Laboratory, YASKAWA ELECTRIC MFG.CO., LTD.
2346, FUJITA, YAHATANISHI-KU, KITAKYUSHU-SHI, JAPAN

We had developed a horizontally articulated robot with 4 arms for manufacturing system in vacuum.
This paper describes an off-line programming system and a robot language for the control of the vacuum robot.
The off-line programming system is useful for operating a robot in vacuum and the robot language is convenient for handling and manufacturing.

1. はじめに

近年、半導体製造プロセスでの真空内搬送システムなど、真空内自動化システムに対するニーズが高まっている。当社は、このほど、真空内でのハンドリング、組立などの作業を行う目的で、真空ロボットを開発した。

真空内で作業するロボットには、ガスやごみの排出、吸入のないクリーンさ、真空負圧に耐え、高温時にも精度を維持すること、振動しない滑らかな動作であることなど、自然環境で動作する産業用ロボットとは異なる、技術的により困難な特性を要求される。今回開発した真空ロボットは、これらの諸特性をクリアした、真空環境に対する適応力を備えたロボットで、真空内モータをベースに、フレキシブルなライン構成ができるよう、4自由度水平多関節構造としている。

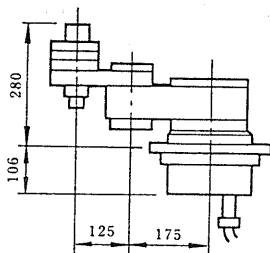
一方、この真空ロボットの制御に関しては、ロボットが人間の視界の及ばない真空チャンバ内で動作すること、環境の特殊性より、有人操作を極力減らすこと、多様な専用システムとの代替や、新規ライン構築に対し、フレキシブルに対応する機能性、作業性などが求められる。このような要求機能と問題設定に対し、種々検討の結果、ティーチボックスのKEY操作により、ロボットを直接動作させるダイレクトティーチングを主体とした方法を改善し、通常環境で実施できるオフラインプログラミングシステムの開発と、ロボットの動作状態モニタリングが、従来、目視と言語、数値情報中心であったものを、リアルタイムシミュレーション画像の表示による方法に変えることによって、マンーマシンインタフェースの改善を行った。さらに、ロボット言語(拡張Pascal)を採用し、広範なアプリケーションにフレキシブルに対応できるように、作業の記述性を高めている。

本稿では、真空ロボットの制御という立場から、主として、オフラインプログラミングシステムとロボット言語の概要について報告する。

2. システムの概要

第1図に真空ロボットの機械的仕様を示す。

ロボットは、水平回転軸が3軸、昇降軸が1軸の4軸構成の水平多関節型で、フランジ面より上のアーム部が真空々間で稼働する。



真空ロボット

● 真空ロボット

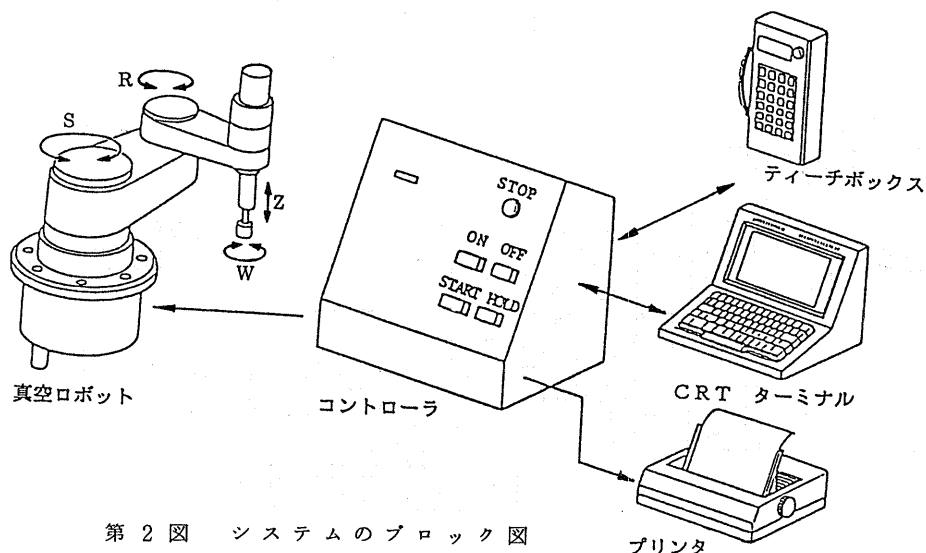
動作形態		水平多関節形
自由度		4
可搬重量		500g
動作範囲	第1アーム	360°
	第2アーム	270°
	手首ひねり	360°
	手首上下	40mm
合成	最大速度	470mm/s

第1図 真空ロボットの機械的仕様

一方，第2図には，システムの概要を示している。

真空ロボットを制御するロボットコントローラには，ティーチボックス及びターミナルがRS-232Cで接続される。

ティーチボックスはオフラインプログラミングで生成した位置データファイルのポジション補正用，ターミナルは，ロボット言語による作業プログラム作成を主とするマンマシンインタフェースである。プリンタは，生成された作業プログラムや各種データファイルの出力用である。



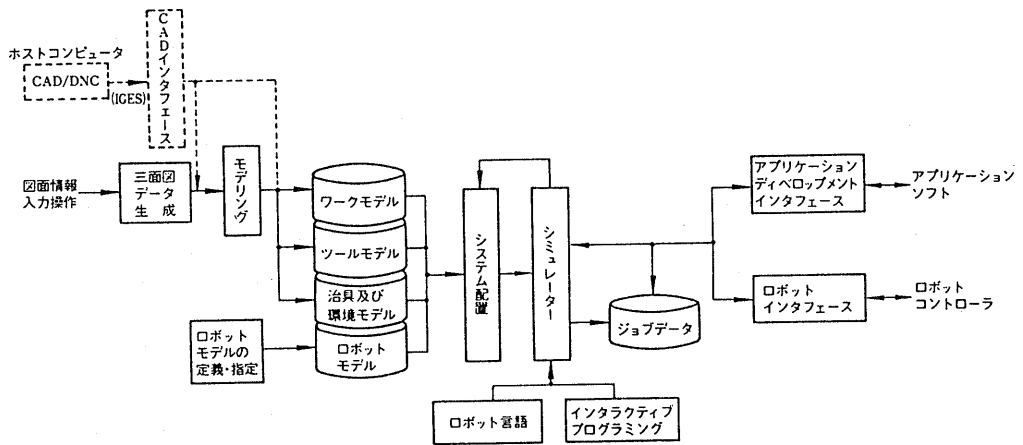
第2図 システムのブロック図

2-1 オフラインプログラミングシステムとの結合

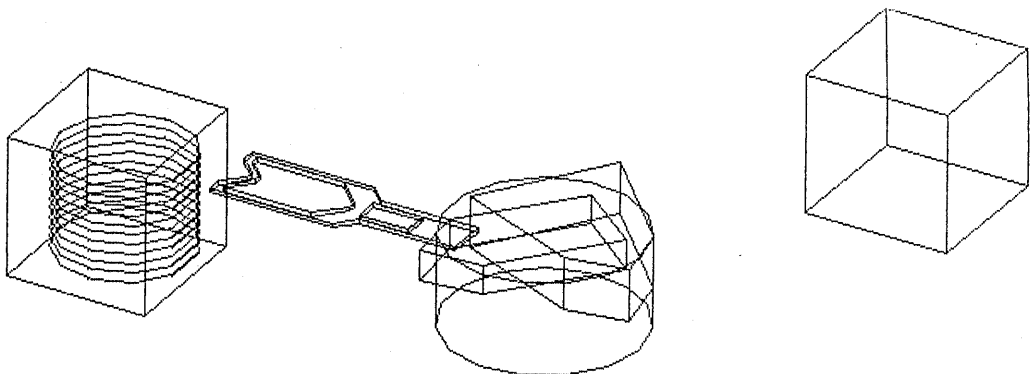
オフラインプログラミングとは，第3図に示すように，三面図などの幾何情報をベースに，コンピュータモデリングによる各対象モデルの生成，各モデル間の相対位置を決定するシステム配置，その結果完成したシミュレーションモデルへのインタラクティブ操作により生成される位置データファイルと，プログラミング言語（ロボット言語と同一）によって作成される作業プログラムとのシミュレーションを経て，実際のロボット用教示データファイルを生成させるもので，コンピュータ，エイデッドな仮想システムである。

真空ロボットと，このオフラインプログラミングとは，環境の特殊性より，とくに密接な関係にある。即ち，真空ロボットは，真空チャンバ内で動作するため，目視することが困難で，ティーチボックスによる直接の教示作業自体が不可能なことがある。また，ライン稼働前であっても，発塵問題もあり，人の出入りは極力避けたいところである。このような条件に対し，ライン外の通常環境でプログラミングできるオフラインプログラミングシステムは有効な手段である。また，真空環境に通常のテレビカメラを持ち込むことはできないため，通常のロボットコントローラのディスプレイのように，言語情報，数値情報によるモニタ機能だ

けでは、情報不足である。これに対し、オフラインプログラミングシステムでは、実際のロボットの動きに同期して、対応するシミュレーション画像を表示することができる。オフラインプログラミングシステムはロボットコントローラからの同期情報、現在値情報、ステータス情報をRS-232C経由で受け取ると、シミュレータを駆動し、対応する画像を生成し、ディスプレイすることによって、リアルタイムディスプレイモニターとなることができる。この機能は、緊急発生時、貴重な作業対象物を損傷させないで、復旧作業を行う場合の空間的位置関係の確認に役立つ。



第 3 図 オフライン・プログラミングのブロック図



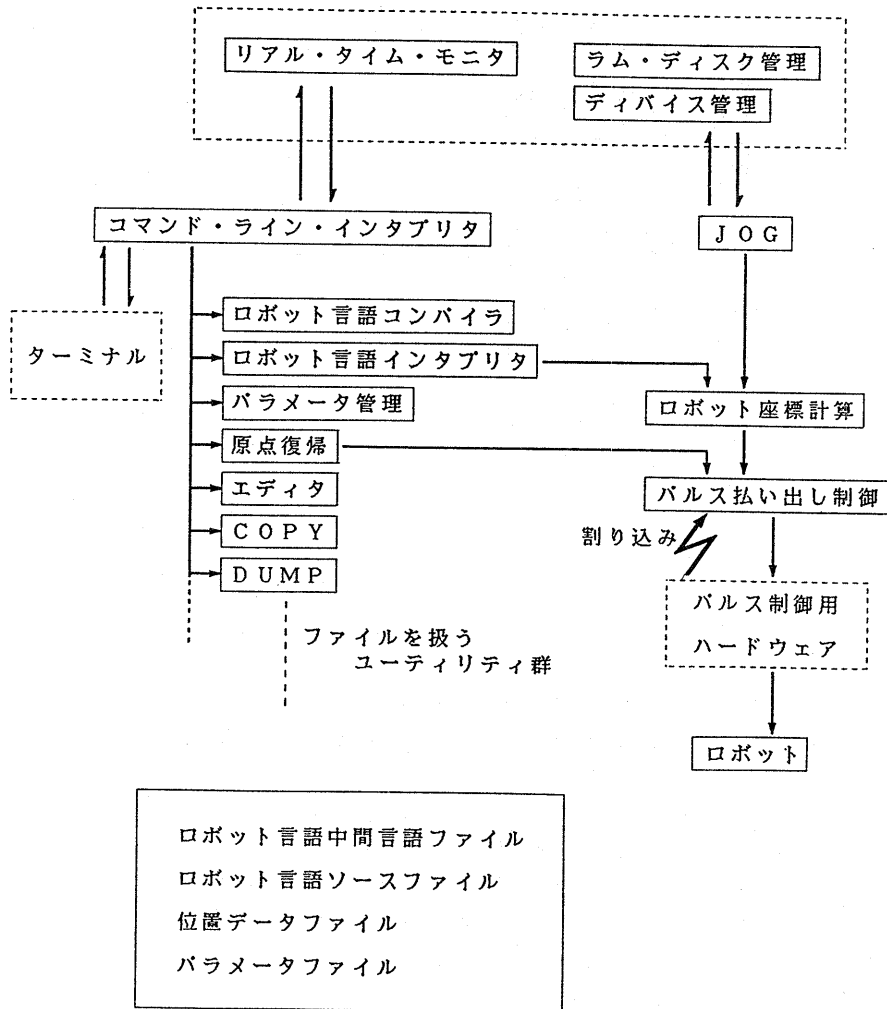
第 4 図 リアルタイムディスプレイモニターの画像

このオフラインプログラミングシステムは、パーソナルコンピュータ（PC 9801もしくはPC/AT）上で実行する。主な機器構成として、PC本体のほかに、HD、FDD X 2、RAM 500KB以上、マウスもしくはディジタイザなどが必要であるが、殆んど標準的機器構成でよい。

第4図にリアルタイムディスプレイモニタの画像のサンプルを示す。

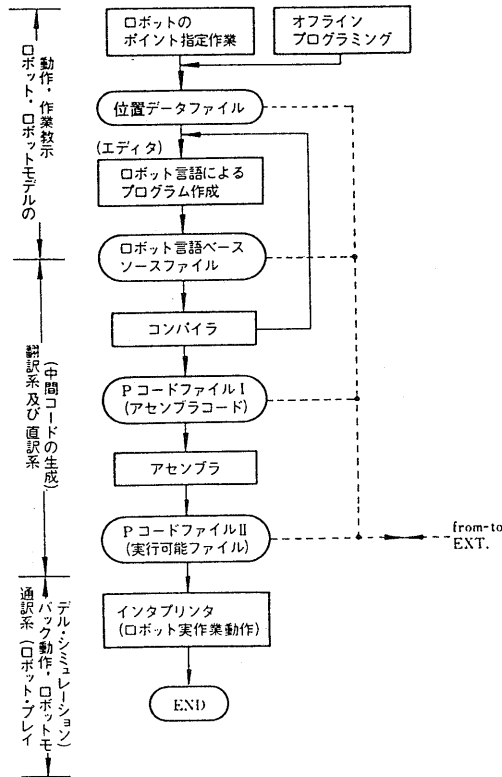
2-2 システムプログラムとロボット言語

ロボットコントローラのシステムプログラムのブロック図を第5図に示す。リアルタイムモニタの管理下で、コマンドラインインタプリタとJOGは同時併行動作をさせることができる。コマンドラインインタプリタはターミナルからコマンド行を読み込み解釈実行する。JOGはティーチボックスの操作により、ロボットのアーム姿勢を任意に移動させ、位置情報を位置データファイルに記憶する。



第5図 システムプログラムのブロック図

ロボット言語コンパイラ，ロボット言語インタプリタなど，ロボット言語処理に関するフローチャートの詳細を第6図に示す。本図は，ロボットコントローラ，オフラインプログラミング併用図で，両者は，表面言語（拡張Pascal），中間Pコードとも全く同一なコンパイラ／インタプリタ方式を採用している。



第6図 ロボット言語処理及び実行のフローチャート

Pascalをロボット言語に採用することによって，Pascalの特長をロボット制御に活用することができる。その効果は次の通りである。

1. Pascalがもつ構文チェック機能を生かせるため，ユーザプロテクションが万全で，より安全で確実な作業プログラムを作成することができる。
2. モジュール（サブルーチン）化が容易で，プログラムの階層化，構造化が可能であるため，ロボットの動作記述，作業記述が簡単で一覧性がよい。このモジュールは，使い方に制約がないため，繁雑になりがちなプログラム作成の負担を軽減することができる。
3. リカーシブルな情報処理を可能にするため，教示点群に対する平行シフト，スケールシフト，回転シフトなどと，それらの組み合わせによる，規則

```

(*$c+*)
(* Handling test program by linear gauge sensor *)
program base(input, output);

const LOOP      = 5;
      A_CHANNEL = 0;
      WAIT_TIMER = 0.5;
      v_coef    = 1.7;
      pai       = 3.141592;
      sens_on   = 1;
      sens_off  = 0;

var i : integer;
      alpha, beta, x_coef, y_coef, z_coef : real;

procedure initial;
begin
  write('Alpha ? : ');      readln(alpha);
  write('Beta ? : ');      readln(beta);

  alpha := alpha * pai / 180.0;
  beta  := beta * pai / 180.0;
  x_coef := cos(alpha) * cos(beta);
  y_coef := sin(alpha) * cos(beta);
  z_coef := sin(beta);

end;

procedure get_work(sens_flag, tool_no, work_no, distance, speed : integer);
var dx, dy, dz, length : real;
      volt : integer;
begin
  movedir(-distance, 0, speed);
  ref(tool_no, work_no, speed);
  if sens_flag = sens_on then
    begin
      volt := inp_a/d(A_CHANNEL, WAIT_TIMER);
      if (volt > 1) and (volt < 128) then
        begin
          length := v_coef * volt;
          dx := x_coef * length;
          dy := y_coef * length;
          dz := z_coef * length;
          movedlt(-dx, -dy, -dz, speed);
        end
      end;

  close(1);
  movedir(distance, 0, speed);
  bend(1);
end;

procedure put_work(tool_no, work_no, distance, speed : integer);
begin
  ref(tool_no, work_no, speed);
  movedir(-distance, 0, speed);
  open(1);
  bend(1);
end;

begin (* main routine *)

  initial;

  for i:=1 to LOOP do
    begin
      init(1);

      get_work(sens_off, 2, 2, 100, 300);

      put_work(2, 3, 50, 300);

      get_work(sens_on, 1, 1, 100, 300);

      put_work(1, 3, 30, 300);
    end;
  end.

```

第 7 図 ロボット言語によるセンサコントロールの事例

性を有するポジション群の生成が容易で、「ハノイの塔」的な作業記述も比較的簡単なため、真空中でのハンドリング、組み立て作業などに適している。

4. P a s c a l がもつ組み込み函数が豊富であるため、演算による任意軌跡の生成、ユーザ定義の座標系と、ロボット座標への座標変換を、ユーザが作成することができる。このため、ユーザが独自にセンサベースロボットコントロールシステムを構築することができるよう、アナログ入力、アナログ出力の制御をロボット言語で、実行できるように準備している。

また、真空内自動化システムを構成する周辺システムとの信号の受け渡しなどのシーケンスコントロールに関し、P a s c a l がもつ論理処理機能は直接のビット操作をなすものではないが、外部 I / O 制御に役立つよう、ロボット言語による R E L A Y - I N / O U T 制御ができるため、フレキシブルなシステム構築を可能にする。

P a s c a l のユーティリティは別の面での効果も予測させる。例えば、1日の作業結果を集計し、場合によってはそれらに統計処理をほどこして、プリントアウトするといった作業の総括も可能にする。

以上の如く、ロボット言語は、真空中作業システムのよりフレキシブルで、よりユーザオリエンテッドなシステムを提供するだろう。

第7図に、ロボット言語による真空ロボットのセンサコントロールの事例を示している。

3. おわりに

真空ロボットは、フレキシブルな真空中自動化システムの実用的なツールとして、呱呱の声をあげたばかりである。今後ユーザ各位のご指導を賜わりつつ、なお一層のレベルアップをはかるべく、研鑽を重ねる所存である。

ロボット言語は、見かけ上のとっつきにくさにもかかわらず、ユーザメリットが大きいと信じる。今後強まるであろうロボット言語の標準化の動きに注目し、かつ、対応しつつ、体系的深耕をはかっていきたい。

オフラインプログラミングシステムは、尚一層の使い易さとともに、実機との誤差修正手段を確立することが求められている。同様に、真空中作業に関する独自機能を豊富にすることによって、専用化の道を歩ませたいと考えている。

最後になったが、紙数の関係で、真空ロボット、コントローラのハードウェアの詳細、ならびに、実機制御に関する記述を行うことができなかつたことをおわびする。

参考文献

- 〔1〕 P a s c a l の言語処理系 (P a s c a l - P 4) ; スティーブソン, ペンバートン, マーチン, ダニエルズ共著 武内・木村訳 近代科学社
〔2〕 P a s c a l の言語処理系 (翻訳系・直訳系 / 通訳系) ; スティーブソン, ペンバートン, マーチン, ダニエルズ共著, 武内, 木村訳 近代科学社
〔3〕 ソリッドモデリング; 千代倉弘明著 工業調査会
〔4〕 ロボット言語 A L ; S ・ ムシュタバ, R ・ ゴールドマン共著 金山・坪内共訳 総研出版