

# 双腕ロボットの動作制御システムの開発\*

外山 真也\*<sup>1</sup>・比良 貴浩\*<sup>2</sup>・塚田 美由紀\*<sup>1</sup>

## Development of System for Double Robot Arm Control

Masaya TOYAMA, Takahiro HIRA and Miyuki TSUKADA

株式会社 ホンダロックは、FANUC 製双腕ロボットと MVTec 製画像解析ソフトを導入し、カメラと画像解析装置により物体の位置を認識し、ロボットアームでその物体を把持し、移動させ複数の部品と組み合わせて指定した位置に配置する一連の動作を制御するシステムの開発に取り組んでいる。

このようなシステムを独自に開発することにより、開発コストを低減し、技術的ノウハウを蓄積し、かつ今後の応用発展的開発に活用できると考える。

今回の開発では、ロボットも画像処理装置においても、ユーザー側でソフト開発が可能な機能が備えられており、それらの機能を利用して開発を進めた。

キーワード：生産工程の省力化、ロボット、画像処理

### 1 はじめに

双腕ロボットとカメラ(位置認識)の全体像を図1に示す。双腕ロボットは通常アームロボットとして利用される据え置き型のロボットアームをベースの左右に取り付けた格好になっている。

そのため、右手系のロボットは右手方向がZ軸、上方向がY軸、前方向がX軸の+方向となる。また、左手系のロボットは、左手方向がZ軸、下方向がY軸、前方向がX軸の+方向となる(図2参照)。

カメラも独自の座標系があり、カメラで認識した物体の位置を、ロボットの左右の座標系における位置として求めるためには、共通の座標系を設定する必要があった。

そこで、開発手順としては、まず、1)座標系の設定を行い、共通座標と各座標系との座標変換機能の開発、次に2)動作制御機能の開発を行い、物体の位置を指定し、ロボットアームを移動させ、物体を把持する動作制御機能の開発、3)指定された位置に物体を置く動作制御機能の開発の順序で実施した。



図1 双腕ロボットの外観

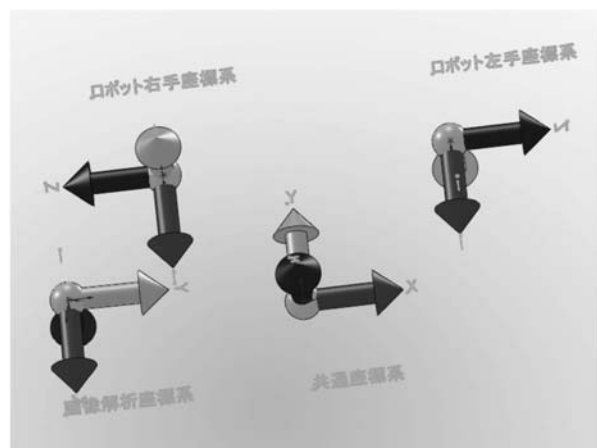


図2 座標系の関係の概念図

\* 共同研究(第1報)

\*<sup>1</sup> 機械電子部

\*<sup>2</sup> 株式会社 ホンダロック

以下、開発手順に従って詳細を記述する。

## 2 開発方法

### 2-1 共通座標系の設定と座標変換

共通の座標系はロボットの基礎台の位置を基準にとり、ロボットの右手系、左手系、カメラの座標系のそれぞれにアフィン変換のデータを用意し、プログラム開発を行った。開発した座標変換ソフトを図3に示す。

このソフトでは、共通座標系から各座標系へのアフィン変換により各座標系における位置座標が求められる。また、その逆も可能である。

さらに、ロボットアームの所定の位置から指定された位置までロボットアームを移動させる場合に、移動時のZ座標を指定することで、退避Z座標まで上昇、XY移動、把持位置までの4地点の位置座標を求められる機能を備えた。この機能は、メニュー「DataList」に「Cvt MoveList」として追加した。この機能については図4に示す。

また、把持動作において、把持する直前に把持方向を考慮し、物体の手前(例えば5cm)で一時的停止し、把持しようとする方向へ移動する機能を設定しているが、回転方向により危険な動作となるため、今回の開発においては回避することとした。今後の検討が必要である。

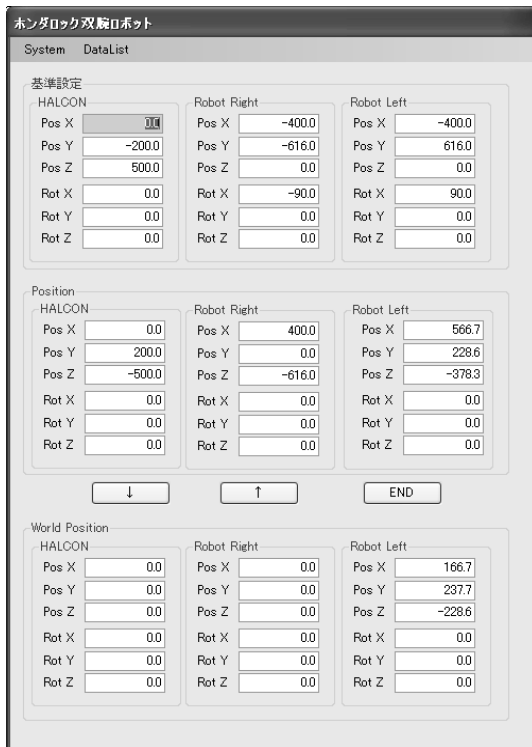


図3 座標変換ソフトの起動状況

### 2-2 ロボットの制御機能の開発

ロボットの制御プログラムに関してはFANUCよりサンプルプログラムが提供されており、それを元に開発を進めた。

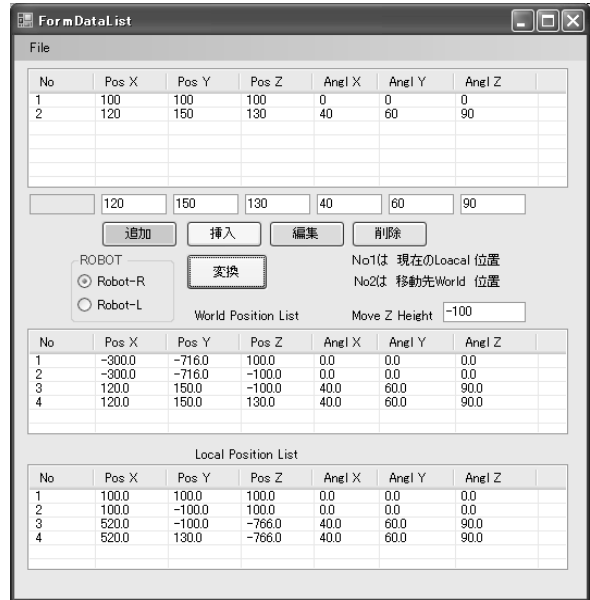


図4 指定した位置までの移動座標計算例

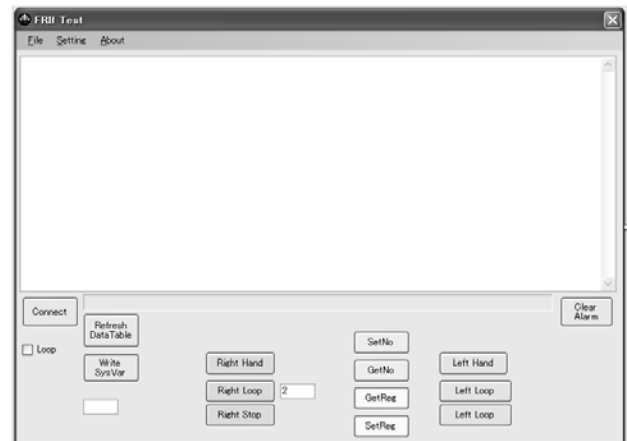


図5 ロボット制御ソフトの起動状況

ロボットの動作については、シーケンス制御と同様なプログラム方式であり、位置レジスタに保存された座標を読み込んで、その位置へロボットを移動させる制御などが可能である。

また、レジスタを利用し、あるレジスタが「1」となったら動作を開始する、あるいは「0」の場合は待機するなどの制御が可能となっている。

そのため、ロボットの一連の動作をロボットの

制御装置においてプログラムしておき、システム全体を制御する PC においては、タイミングに応じて、指定されたレジスタの数値を変更することで、ロボットの制御を実現できる。

開発した制御ソフトの起動状況を図 5 に示す。このプログラムにおいて、スレッド処理を行い、動作中での中断処理を実現した。

また、ロボットの動作制御の概要を List1 に示す。

Robot 動作流れ	PC の流れ
初期設定	PCが保有する変数は
物体の個数設定 n=10	n
終了レジスタ Sw02 =0	Ct
実行数Ct =0	Pos
Loop 100	
If 終了レジスタ Sw02=0 then Loop1 0	
終了レジスタ Sw02 =0に設定	Sw02.Set(0)
物体のPosを読み込んで レジスタに設定	Pos.Get & Push Register
Posを設定したレジスタSw01=1	Sw01.Set(1)
ロボットハンド'物体の位置の手前まで移動	do
ロボットハンド'物体の法線方向(Vector)に対応して回転	Sw02.GetRegister()
ロボットハンド'物体の把持位置まで移動	while Sw02==0
ロボットハンド'物体を把持	
ロボットハンド'物体を持ち上げて移動する	
処理	
ロボットハンド'物体を配置位置の手前まで移動	
ロボットハンド'物体を配置位置まで移動	
ロボットハンド'物体を離す	
終了処理	
終了レジスタSw02=1	Sw01.Set(0)
Posを設定したレジスタSw01=0	Ct = Ct+1
実行数Ct = Ct+1	
If Ct < n Then Loop 100	If Ct < n Then Loop 100
END	
コメント	
Sw01 は 位置座標をレジスタに設定したとき =1	
Sw02 は ロボットが動作完了したとき =1	
Sw03 は 位置座標がファイル保存されたとき =1	
Sw04 は ロボットの動作制御 待機するとき =0	
ループ実行 =1	
終了 =2	

List1 ロボット制御の概要

### 3 結果および考察

今回の研究開発において、ロボットの動作制御に関する PC のソフトウェアを開発し、指定した位置への移動、物体の把持などの一連の動作制御を実現した。ただし、手首の回転については、まだ十分な動作確認が取れていない状況である。特に物体を把持する動作において、ベクトル方向を考慮して把持方向を求め、手首を回転させる場合、回転方向により危険な動作となる場合について考慮が必要である。

このような場合、把持する手首の方向におい

て、ある程度の範囲を設定して、回転の制御を実施することの検討が必要である。

また、カメラおよび位置認識については、位置精度があまり正確ではなく、画像解析ソフトとの連携が実現できていない。

今後、三次元座標系における位置認識ではなく、簡易的な二次元平面での位置認識を実現させ、システムとしての活用を実現したい。

### 4 まとめ

今回の研究開発において、ロボット制御に関するソフトウェアの開発を行い、その手法などについて理解を深めることができた。ロボットの動作は、各動作ごとにコマンド化し、動作の開始と終了をレジスタに設定することで、一連の動作制御が実現できることが分かった。

画像処理ソフトの方も、ユーザー側でコマンド開発可能な環境が用意されているので、双方を連携させて制御することは可能であると考えられる。

ただし、それぞれに用意されている関数の新旧バージョンの相違が、開発の難易度に影響するので注意が必要であることがわかった。