

# ロボットコントローラ **MRC01** の開発および 設備へのロボット導入事例

近藤 大生 日下部 和明

設備の自動化による省人化を目的に、ロボットの導入が増えています。しかし、市販の産業用ロボットはサイズに制約があることや、大型のものが多いため、「ロボットの内製化」の要望も高まっています。内製したロボットを制御するためには、ロボット制御に関する知識が必要となりますが、ロボット制御の難しさが内製ロボット導入の大きな障壁となっていました。

今回開発したロボットコントローラ **MRC01** および専用プログラミングソフト **MRC Studio** は、ロボットを初めて製作する設備設計者でも、簡単にロボット制御ができるようになることを目的に開発を行いました。

ここでは、**MRC01** および **MRC Studio** の技術や機能と、実際に内製ロボットを社内設備へ導入することにより、設備立ち上げ時間やプログラム作成時間短縮が可能となった事例を紹介します。

## 1. はじめに

世界の産業用ロボットの販売台数は、近年大きく増加しています。特にハンドリング・搬送工程などの自動化しやすい工程では、省人化を目的にロボットの導入が加速しています。世界の多くの国では、生産年齢人口の減少<sup>(1)</sup>による人手不足の解消を目的として、今後もロボットによる自動化が進展すると考えられます。

ロボットの導入にあたり、市販の産業用ロボットを導入するのが一般的です。しかし、市販の産業用ロボットはサイズに制約があることや、大型のものが多いため、既存の設備への後付けは難しくなります。このため「ロボットの内製化」とそれにともない、内製ロボットの制御ができるロボットコントローラの要望が高まってきました。

本稿では、この要望に応えるため開発された、ロボットコントローラ **MRC01** および専用プログラミングソフト **MRC Studio** (図1参照) の技術や機能を説明します。

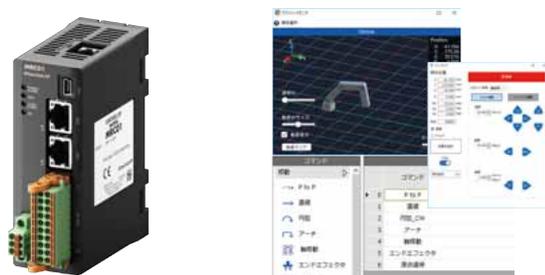


図1 MRC01 および MRC Studio

## 2. ロボット制御に必要な技術

代表的な産業用ロボットとして垂直多関節ロボット、水平多関節ロボットなどがあります(図2参照)。ロボットごとに可動範囲や速度、精度などの特性が異なるため、用途に合わせて最適なロボットを導入することが重要です。ここでは各種ロボット制御に必要な技術を説明します。

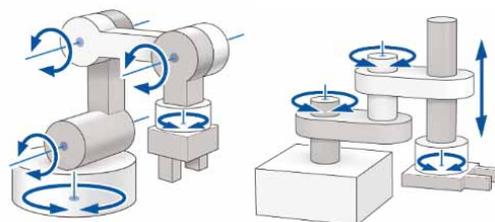


図2 代表的な産業用ロボットの構造

### 2.1. ロボットの制御とは

ロボットが実際に作業を行う手先の位置を Tool Center Point と呼びます(以下TCP)。ロボット制御とは、ロボットの各関節を駆動しているモーターの動きを個別に制御するのではなく、TCPの位置や速度、軌道を思い通りに動かすためのロボット全体の制御を指し、これがロボットコントローラの基本的な役割になります(図3参照)。

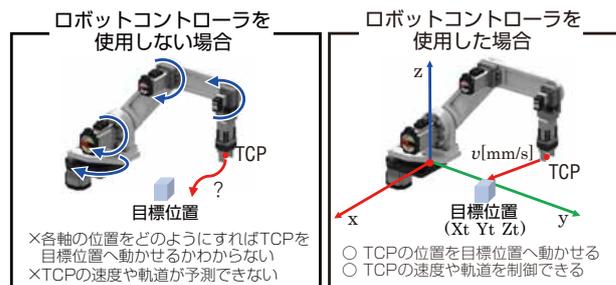
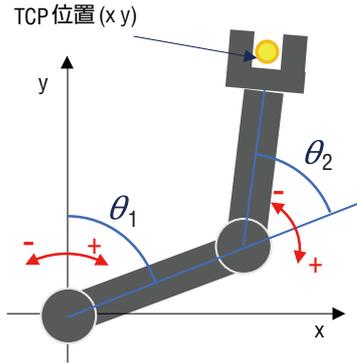


図3 ロボットの運転

## 2.2. 順運動学と逆運動学

ロボットの制御では、運動学計算（キネマティクス計算）が重要な役割を果たします。運動学計算とはロボットのTCPの位置と、各関節の角度変換を行う座標変換計算です。各関節角度からTCP位置を計算する「順運動学計算」と、TCP位置から各関節角度を計算する「逆運動学計算」の2種類があり、互いに逆変換の関係にあります（図4参照）。これらの計算はロボット制御に必須であり、ロボットの構造により計算方法が変化します。複雑な構造のロボットになるほど、運動学計算は高度化していきます。



TCP位置:  $r = \begin{pmatrix} x \\ y \end{pmatrix}$   
 関節角度:  $\theta = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}$   
 順運動学:  $\theta \rightarrow r$   
 逆運動学:  $r \rightarrow \theta$

図4 2リンク水平多関節ロボットの運動学

運動学計算では、まずTCP位置を表す直交座標系（ベース座標系）や各関節の原点位置、および回転方向符号を定義します。各関節の原点位置、および回転方向がこの定義からはずれると、ロボットを正しく制御できなくなります。さらに、運動学計算には、アーム長などの各部諸元情報が必要なため、正確な諸元情報を設定することも重要となります。

また、ロボット構造によっては一つのTCP位置に対して複数の姿勢（右手系/左手系）が発生する場合があります、これは逆運動学計算の解が複数存在することを意味します（図5参照）。ロボットによっては姿勢が変化すると、関節部が周囲と干渉したり、可動範囲が変化するため、作業内容に応じて適切な姿勢を選択しながら制御することが必要となります。

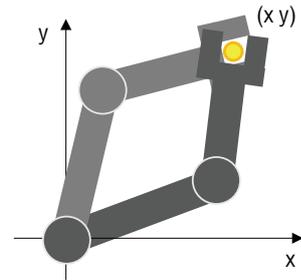


図5 逆運動学の解の重複性

## 2.3. 特異点

ロボットには特異点と呼ばれる、ロボット構造に起因する制御不可能な姿勢が存在し（図6参照）、特異点付近では一部の関節速度が急激に上昇します（図7参照）。

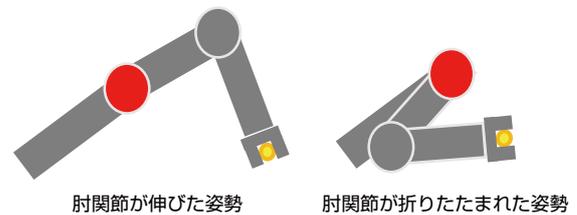


図6 代表的な特異点となる姿勢

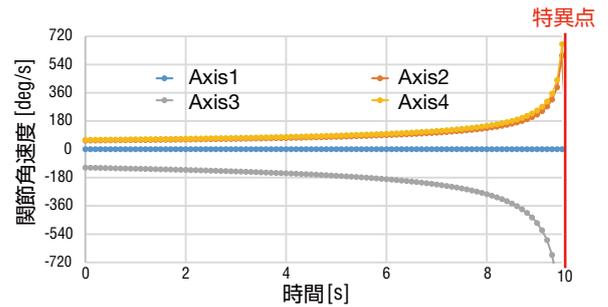


図7 4軸垂直多関節の特異点付近での関節速度変化

特異点となる姿勢は、ロボットの運動学計算で求めることが可能です。ロボットを安全に制御する上で、特異点付近の関節速度の急上昇を未然に防ぐ必要があります。

## 2.4. 軌道制御

TCP軌道の制御運転では、軌道上にある各点の直交座標系での逆運動学計算をすることで、各関節の角度や速度プロフィールを算出できます（図8参照）。

各関節の角度や速度プロフィールは原則として、時間的に変化する複雑なプロフィールになるため、このプロフィール通りに各部のモーターを同期運転させるための、高度なモーション制御技術がロボット制御には必須です。

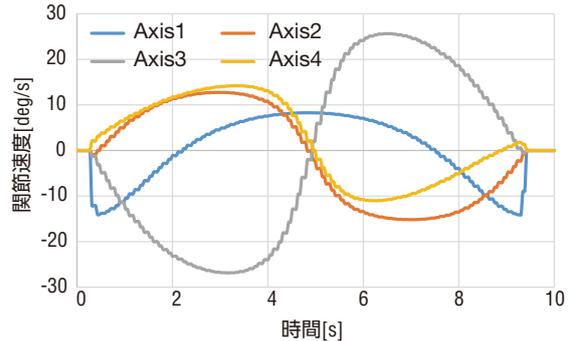
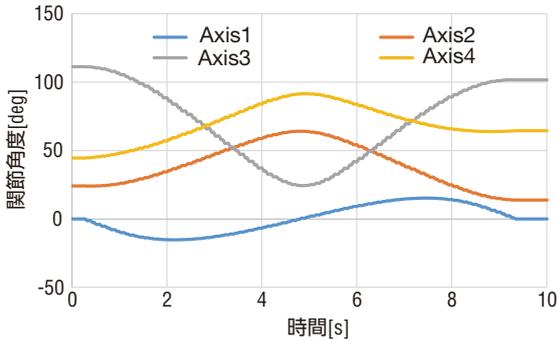
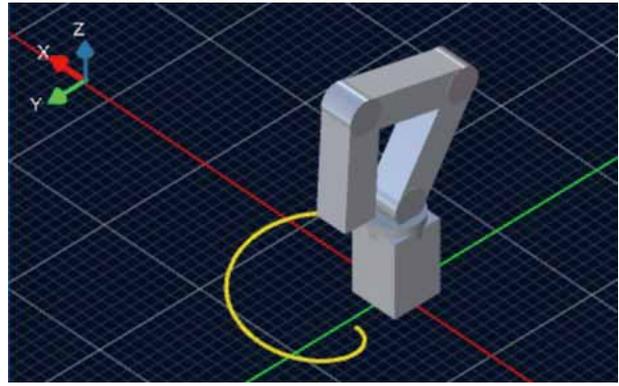
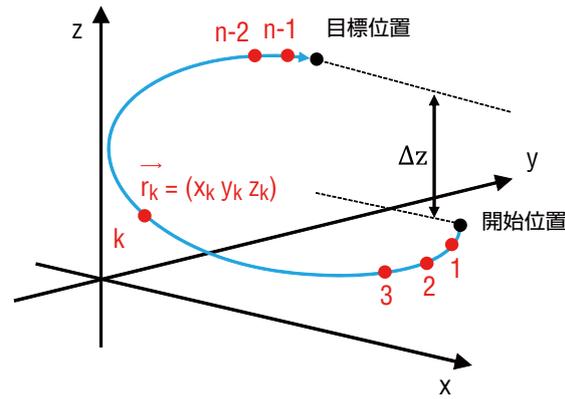


図 8 4 軸垂直多関節のヘリカル補間運動の例

### 3. ロボットコントローラ MRC01

#### 3.1. 対応するロボットの構造

MRC01 は、バッテリーレスアブソリュートセンサ<sup>(2)</sup>を搭載した **αSTEP AZ** シリーズのモーター制御用に開発された、ロボットコントローラです。ロボットの構造および機構諸元情報を MRC01 に入力することで、ロボットの構造を意識することなく共通の方法で操作できます。

ロボットの構造によってオフセットが生じる場合がありますが、MRC01 はこれらのオフセットを考慮した構造モデルにて運動学計算を行います (図 9 参照)。そのため、構造の自由度が高いロボットの内製化においても、適切な機構諸元情報を設定することで運転が可能です。さらに、エンドエフェクタ軸の制御にも対応しています。

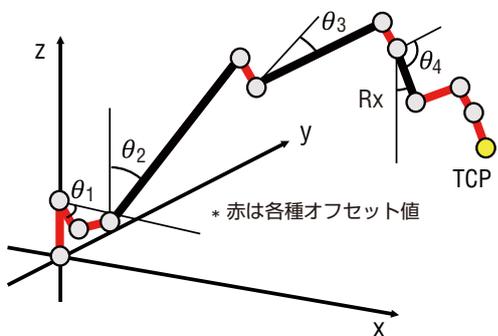


図 9 MRC01 の 4 軸垂直多関節ロボット構造モデル

#### 3.2. システム構成

MRC01 は EtherNet/IP<sup>TM</sup>(注1) に対応しています。EtherNet/IP<sup>TM</sup> の上位機器から I/O 制御や運転データを送信することで簡単に内製ロボットの制御が実現できます。<sup>(3)</sup>



図 10 MRC01 のシステム構成

また、MRC01 には入力 8 点、出力 8 点のダイレクト入出力を搭載しています。信号の入力のみで運転の起動や停止、周辺機器の制御などのさまざまな制御が可能です。

(注1) EtherNet/IP<sup>TM</sup> は ODVA の商標です。

### 3.3. MRC01 が対応する座標系

ロボットのティーチングをする際は、動作に応じて適切な座標系を選択することで、複雑な構造のロボットでも直感的にティーチングを行うことができます。ロボットのもつ動きの自由度を最大限に生かした、多彩な動作プログラム作成が可能です。**MRC01**によるティーチングで選択可能な座標系および、定義を説明します。

#### 1) 直交座標系(XYZ) [単位: mm]

最も使われるのが直交座標系です。原点位置をロボットのベース取付面中央に固定する「ベース座標系」と、任意に原点位置を変更可能な「ユーザー座標系」の二つがあります(図11参照)。

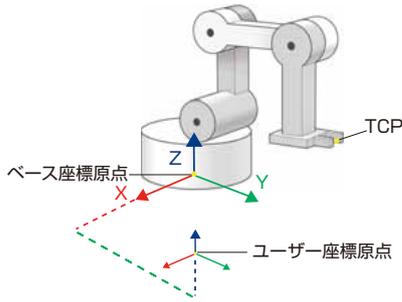


図 11 直交座標系

#### 2) 手先角度座標系(RxRyRz) [単位: deg]

垂直多関節や水平多関節ロボットで設定可能な、アーム先端の手先の角度を制御するための座標系です(図12参照)。ロボット姿勢によらず、手先角度を一定に保ったり、さまざまな角度からワークを掴む動作が実現できます。

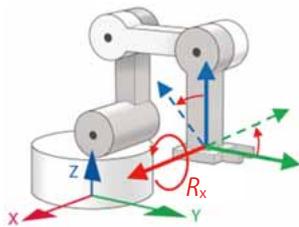


図 12 手先角度座標系

#### 3) ツール座標系(TxTyTz) [単位: mm]

ロボットの手先位置を原点とする直交座標系です(図13参照)。各座標の向きは手先角度によって変化します。手先角度を維持したまま、向いている方向へ前進、後退の動作が可能です。

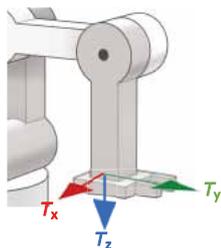


図 13 ツール座標系

#### 4) 軸座標系(J1 J2 J3 ...) [単位: deg or mm]

ロボットの各関節の回転角度(直動機構の場合は移動量)で表す座標系です(図14参照)。特定の軸のみを動作させたい場合に使用します。

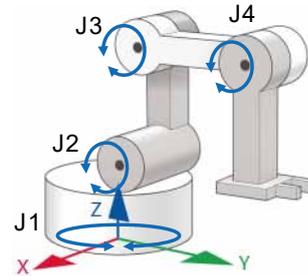


図 14 軸座標系

### 3.4. 運転機能

**MRC01**の運転機能を表1に示します。

表 1 運転機能一覧

| 運転の種類                 | 説明   | 対応 I/F                  |
|-----------------------|--|-------------------------|
| プログラム運転               | ・MRC Studio で作成・書き込みしたプログラムを実行<br>・決まった運転を実行する場合に使用上位のプログラムの負担を軽減できる | EtherNet/IP™<br>ダイレクト入力 |
| ダイレクトデータ運転            | ・上位機器から運転データを書き換えながら行う運転<br>・状況に応じた指令を与えたい場合に有効                      | EtherNet/IP™            |
| JOG/ インチング / ZHOME 運転 | ・特定の信号を ON するだけで実行できる運転<br>(JOG、インチング、ZHOME)                         | EtherNet/IP™<br>ダイレクト入力 |

ロボットに装着されたカメラで目標位置を検出して動作させる場合(図15A参照)は、ダイレクトデータ運転を使用し、所定位置への搬送のような決まった動作の場合(図15B参照)は、プログラム運転を使用することで、上位側機器のプログラム作成の負担を減らすことができます。

運転プログラムは最大64個保存が可能で、各運転プログラムは最大128個のコマンドによるシーケンスを組むことができます。



図 15 ロボットとカメラの連携

その他に特定の信号入力のみで実行可能な運転として、JOG運転、インチング運転(一定量送り運転)、ZHOME運転(高速原点復帰運転)があります。

**MRC01** のプログラム運転および、ダイレクトデータ運転で実行できる運転コマンドを表2に示します。TCPの軌道を制御する「直線」や「円弧/ヘリカル」、ピック&プレイス用途を想定した「アーチ」などの代表的な補間運転に対応しています。また、「P to P」はTCPの軌道を保証しない運転ですが、直線補間により高速運転が可能です。「P to P」では一部ロボットのみ、右手系/左手系の姿勢切り替えが可能です。

表2 運転コマンド一覧

| コマンド                | 説明   |
|---------------------|--|
| P to P              | ・目標位置まで各モーターが最短位置決めを行う<br>・軌道は保証しないが、直線補間よりも速く移動可能<br>・スカルロボットは右手/左手の変更が可能 |
| 直線                  | ・目標位置まで直線補間で移動   |
| 円弧/ヘリカル             | ・目標位置まで円弧(X-Y座標上)補間で移動<br>・Z座標を設定することでヘリカル補間も可能                            |
| アーチ                 | ・アーチ状の軌道で移動<br>・速度を落とさずに上昇⇒水平移動⇒下降動作が可能<br>・ピック&プレイスの動きがアーチコマンドのみで可能       |
| パレット<br>PtoP/直線/アーチ | ・パレタイジング用コマンド<br>・専用パラメータで設定したパレタイジング動作を実行                                 |
| 軸移動                 | ・指定した軸(モーター)のみを動かす   |
| エンドエフェクタ            | ・エンドエフェクタのみを動かす<br>・押し当て運転が実行可   |
| 原点復帰                | ・TCPをユーザー座標系の原点位置に移動   |

さらに、パレタイジングが簡単に実行できる「パレットコマンド」、特定の軸のみ動作させる「軸移動」コマンド、エンドエフェクタの押し当て運転によりワークの把持動作が可能な「エンドエフェクタ」コマンドに対応しています。

プログラム運転で使用可能な制御コマンドを表3に示します。これらのコマンドを使用することで上位機器が無くても**MRC01**のI/Oから周辺機器を制御したり、簡易シーケンスを組むことが可能です。

表3 制御コマンド一覧

| コマンド       | 説明                               |
|------------|----------------------------------|
| 待機(時間)     | 指定時間、プログラムの実行を止める                |
| 待機(信号)     | 指定の信号がONもしくはOFFするまでプログラムの実行を止める  |
| ループ(始点/終点) | 始点コマンドと終点コマンドの間にあるシーケンスを指定回数繰り返す |
| 信号出力       | 指定の信号をONもしくはOFFする                |

## 4. 保護機能

**MRC01**には操作や設定の誤りによる、ロボットの意図しない動作や、誤ってロボットが人、周辺機器などと接触したときのリスクを低減する各種機能を搭載しています。

### 4.1. 特異点保護

垂直多関節ロボットや水平多関節ロボットのような多関節ロボットにはすべて特異点が存在します。

**MRC01**にはロボットが特異点に近づき関節速度が急加速する危険動作になる前に、運転停止してアラームで知らせる「特異点保護機能」を搭載しています。ロボットティーチング中の軸速度の急激な上昇などのリスクを低減できます。

### 4.2. スリップモード

ロボット停止時に、いずれかの軸に外力により大きな負荷がかかった場合に、元の位置に戻らないようにすることができる機能です。

作業人や周辺機器などが誤ってロボットに接触すると、外力によりモーターに大きな位置偏差が生じます。この場合、一般的なモーター制御ではより大きなトルクで元の位置に戻そうとする動きとなりますが、状況によっては、新たに別の接触を引き起こす可能性があります。

スリップモードを設定することで、このような新たな接触リスクを低減することができます(図16参照)。また、スリップ後の位置から運転を再開することができます。

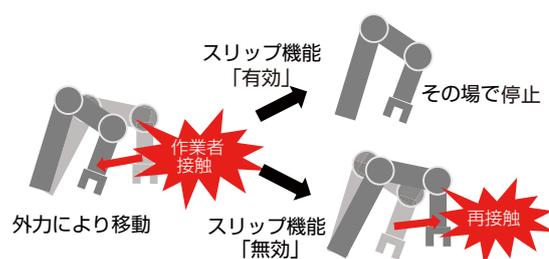


図16 スリップモード設定時の動作

### 4.3. 運転時軸異常検出による停止

ロボットの運転中にいずれかの軸で異常が発生し、運転不可能となった場合、全軸を停止してロボットの動作を停止させます。ロボットの運転はどれか1軸でも停止したり、コントローラの指令に追従できなくなると、直線や円弧のような軌道の制御ができず、予測不能な動きをすることがあります。それを避けるため、すべての軸を停止させてアラームで知らせる機能を搭載しています。

### 4.4. 位置リミット、速度リミット

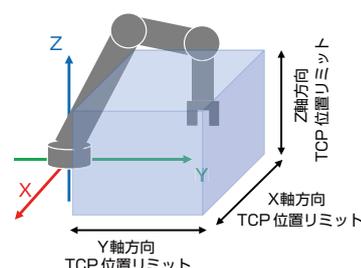


図17 TCP位置リミットの例

**MRC01**にはロボットのTCPと各モーターの位置リミットおよび速度リミット機能を搭載しています(図17参照)。操作ミスなどによる意図しない速度や動作を未然に防止することで、周辺機器との接触や破損リスクを低減することができます。

## 5. 生産設備へのMRC01と内製ロボットの導入事例

本章では、**MRC01**を用いた社内製造ラインの自動化事例を解説します。

### 5.1. 自動化ライン概要

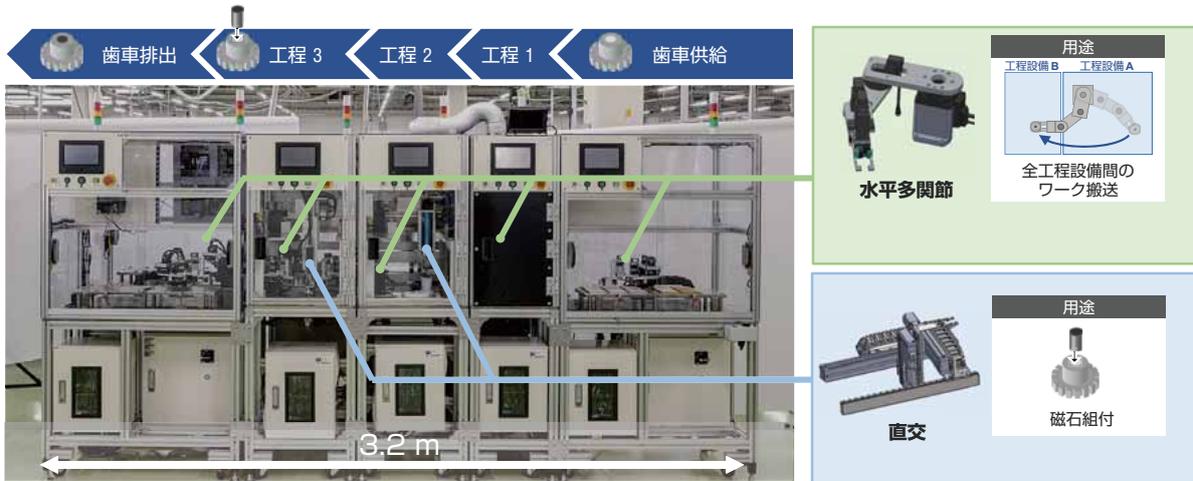


図 18 自動化ライン概要

図18は、歯車に磁石を組み付ける自動化ラインです。各工程の設備を標準化し、製造工程にあわせて各設備を連結しています。これにより以下の利点があります。

- ・ 工程ごとに設備設計や、動作検証が可能
- ・ 工程の変更(設備の入れ替えや、追加、削除など)がしやすい
- ・ 標準化設備により、各設備の他製造ラインへの転用が容易

本設備では工程間の歯車搬送および、工程2と工程3の磁石組付部で**MRC01**を用いた内製ロボットを採用しています。市販のロボットではなく、設備に合わせたサイズのロボットを使用することで、設備の小型化を図りました。システム構成を図19に示します。搬送部には水平多関節ロボット、磁石組付部には直交ロボットを使用し、各ロボットを制御する**MRC01**と、設備内および工程全体を管理するPLCが接続されています。

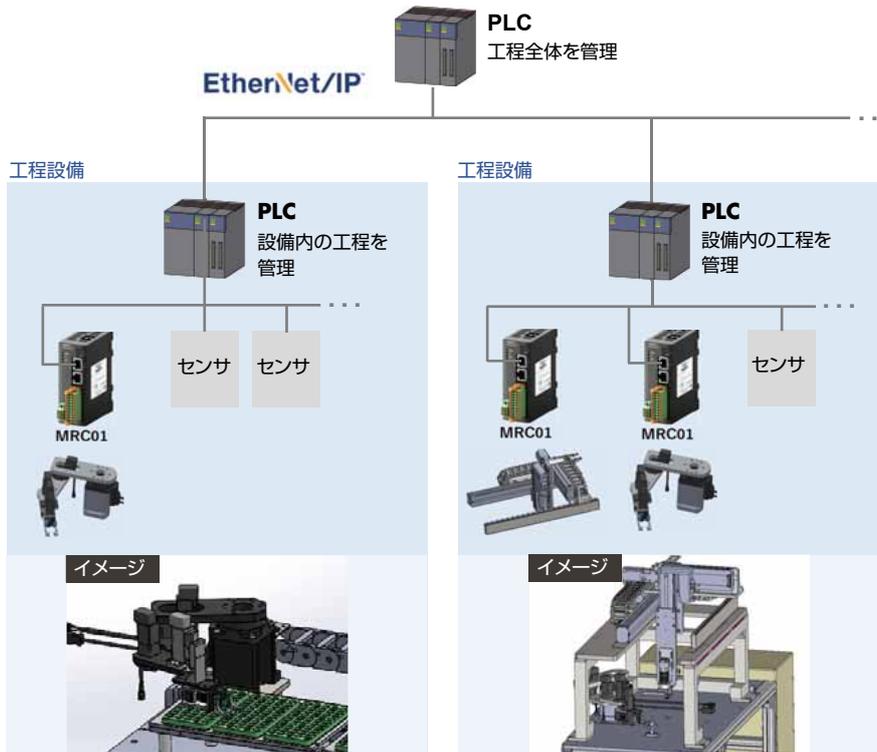


図 19 システム構成イメージ

## 5.2. 水平多関節ロボットの内製化

本設備では設備の標準化・小型化を達成するため、ロボットは狭い設備内に組み込める大きさであることや、限られたスペースでも自由度が高く、高精度な位置決め動作が求められました。そこで、搬送用ロボットには可動域に対して設置面積が小さく、姿勢の自由度が高い水平多関節ロボットを採用しました。また、市販ロボットでは本設備に適した大きさのものが無いため、**MRC01**による内製ロボット設計を実施しました。これにより、市販ロボットでは難しかった、限られた範囲内の高い自由度の確保と、高精度位置決め動作の両立を実現しました。(図20、21参照)。



ロボットタイプ  
4軸 水平多関節  
(先端軸昇降+Rz軸)  
最大リーチ  
約 450 mm

図 20 水平多関節ロボット (内製)

### 水平多関節の特徴

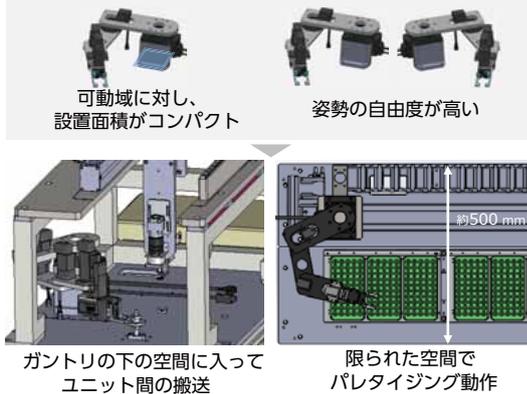


図 21 水平多関節ロボットの特徴と使用実例

## 6. MRC01の設備導入効果

ロボット動作を**MRC01**で管理することで、設備立ち上げ、動作変更、他設備へのロボットの流用など、あらゆるシーンで作業効率を向上することができます。

### 6.1. 内製多関節ロボットを容易に操作

内製した多関節ロボットを思い通り操作するには、2章で記述した通り各モーターに対して複雑な制御が必要になります。しかし、本設備では**MRC01**を使用することでその複雑さを意識せず、多関節ロボットを直交ロボットと同様の感覚で、直交座標系で操作・ティーチングすることができました(図22参照)。設備設計初心者でも、1時間ほどで動作可能なプログラムの作成を行うことができました。

### 6.2. MRC01とPLCプログラム作業効率の向上

ロボットを含めたすべての機器をPLCで制御する場合、キネマティクスの設定やロボット動作、その他周辺機器の動作プログラムを一度に作成する必要があり、PLC担当者の負担が大きくなります。

一方、**MRC01**は**MRC Studio**を使うことで、PLCを介さず単独でロボット動作プログラムの作成、テストをすることができます。

PLCのプログラムとは別にロボット動作を作りこむことが可能なため、複数の担当で作業分担することで、設備立ち上げの効率化が可能となりました。

また、後からロボット動作の変更が必要な場合もPLCのプログラムはそのままに、**MRC Studio**の編集のみでよいので、少ない工数で変更ができます。

### ティーチング画面

### プログラム画面

### グラフィックモニタ

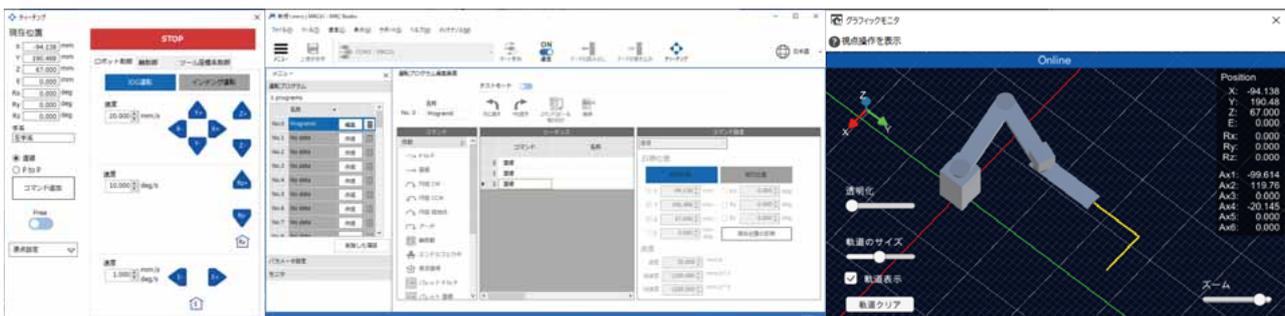


図 22 MRC Studio ティーチングイメージ

### 6.3. 各種ロボット操作の習得時間短縮

用途に合わせて各種市販ロボットを使用する場合は、複数メーカーのロボット操作を習得する必要があり、操作習得に長い時間がかかります。しかし、**MRC01**は直交、水平多関節、垂直多関節などのロボットタイプに関わらず、

**MRC Studio**の操作方法や、PLCから**MRC01**への接続、通信方法は変わりません。そのため、一度ロボット担当者が操作方法を習得すれば、別のロボットを立ち上げる場合でも、新たに操作方法の習得は必要ありません。本設備の事例では、直交ロボットで**MRC01**の操作を習得することで、水平多関節ロボットも操作することが可能となり、立ち上げ時間の大幅短縮が可能となりました。

### 7. まとめ

**MRC01**は、さまざまな構造の、内製ロボットの制御を簡単に行うためのロボットコントローラです。

生産年齢人口の減少傾向にある現在、増加していくロボット導入のニーズにお応えできる製品として開発しました。

今後も機能拡張や対応ロボットの追加、使い勝手や安全性の向上を実施し、幅広いお客様のご要望に対応できる製品の開発を行っていきます。

### 参考文献

- (1) 総務省 ホームページ「人口減少の現状」  
<https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h30/html/nd101100.html>  
(2022/3/18)
- (2) 根岸 徳行「バッテリー不要多回転アブソリュートセンサの開発」,  
RENGA, No.179, (2014), pp.1-3
- (3) ODVA ホームページ, ODVA(Open DeviceNet Vender Association),  
<http://www.odva.org/>, (2022/3/2)

### 筆者



近藤 大生

制御機器システム事業部



日下部 和明

制御機器システム事業部