



# CAD/CAM システムを用いた産業用ロボットによる作業の自動化

Automation Using Industrial Robot on the Basis of CAD/CAM System/Naoki ASAKAWA

金沢大学 浅川直紀

## 1. はじめに

「ロボットを用いてさまざまな作業を自動化させたい」、「研究で新しいアクチュエータを考案したので、ロボットにもたせて作業をさせてみたい」などと考えて導入したのはよいが、実際ロボットを触りはじめてみると、どのように動作プログラムを作成するかで悩んでいるといった話をよく耳にする。特によく聞くのは、ティーチングプレイバック（後述）で使用しはじめたが、ティーチングに時間がかかる、作成した動作プログラムの修正が煩雑、などの問題である。図1に示すような現在主流の産業用ロボットは、単体ではティーチングプレイバックによって使用することを想定しているが、システム全体の自動化を進めるうちに、動作プログラムの作成もなんとか自動化できないか、と考えるのは当然であろう。

本稿では、磨き、塗装、バリ取りなどの作業を対象とし、CAD/CAM 技術を応用して上記の問題の解決について考えてみたい。その際、学術的な解説は先達に譲り、ロボットを購入後、上記の作業を自動化するシステムを自ら作成して開発や研究を行う際にどのような問題を考える必要があるのかについて焦点を絞り、なるべく簡易に概略を述べてみたい。

## 2. 作業に必要な自由度の具体例

まず、前章で挙げたような作業で使う工具にはどれくらいの自由度を与える必要があるのかを考えてみよう。工具の自由度とロボットの自由度については、本連載では機構学的観点から過去に詳しく解説されており<sup>1)</sup>、これらの作業が一般に6自由度程度を必要とするということが述べられているのでぜひ参照していただきたいが、ここではもう少し具体的に見てみよう。

図2や図3の例で示す通り、位置の指定  $P$  だけなら3自由度で十分だが、本稿で対象とするような作業の姿勢に関しては工具の構造や使い方によって、工具の主軸方向などを表すベクトル  $T$  が1本（図2）ないし、それに加えて適用方向の指定が必要な場合はその方向を表すベクトル  $D$  の2本（ $T$ ,  $D$ ）（図3）で指定をする必要がある。簡単にいうとベクトル1本の場合は5自由度が必要な作業であり、2本なら6自由度が必要な作業ということになる。ここで気を付けたいのが、「本当に必要な自由度はどれくら

いなのか」をよく考えることである。

図4に示す溶接のトーチの場合は、溶接という作業そのものは本質的には5自由度で十分だが、周囲との干渉などを考えると6自由度が必要、というような場合に相当する。しかし、干渉を考慮しても完全に6自由度が必要なわけではなく、火口の回転は多少許されるので、状況によって



図1 産業用ロボット（垂直多関節型）

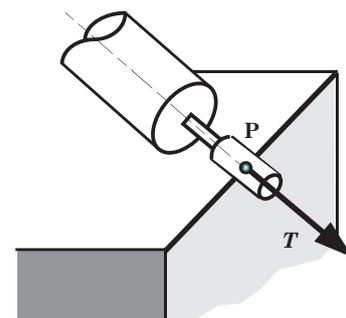


図2 ロータリーバーの自由度（位置  $P$ 、工具軸ベクトル  $T$ ）

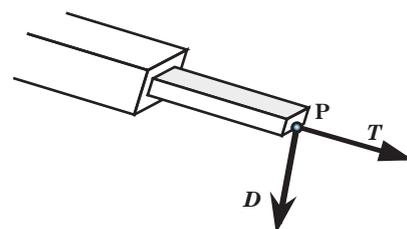


図3 スティック砥石の自由度（位置  $P$ 、工具軸ベクトル  $T$ 、工具方向ベクトル  $D$ ）

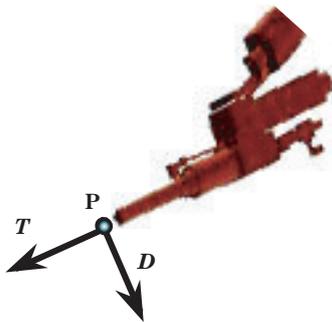


図4 溶接トーチの自由度 (位置  $P$ , 工具軸ベクトル  $T$ , 工具方向ベクトル  $D$ )

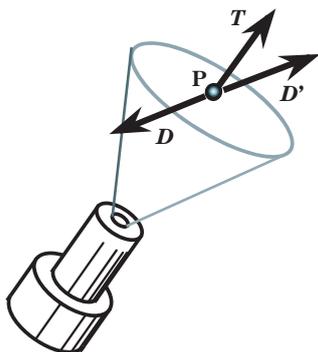


図5 スプレーガンの自由度 (位置  $P$ , 工具軸ベクトル  $T$ , 工具方向ベクトル  $D, D'$ )

必要な自由度は異なってくる。

また図5に示す塗装用スプレーの場合は、塗料が対象に付着する場合の形状(パターン)が、円形でなく楕円形であるため、一般には短軸方向にスプレーを移動して作業を行うが、短軸方向に沿っているなら正負どちらの方向でも構わない。これは概念的には5.5自由度といったところになると思われる。

一般に、例示したような作業の自動化は人間の代わりにロボットに工具をもたせるといった発想で行うことが多いため、作業に必要な自由度は人間の腕のもつ6自由度に近くなるのは当然といえる。以上からこのような作業用には6自由度、すなわち6つの関節をもつ垂直多関節型の産業用ロボットがなぜ適しているのかわかる。以後この垂直多関節型の産業用ロボットに関して話をすすめていきたい。

### 3. 一般的なプログラム生成法

一般にはロボットはどのようにプログラムされているのだろうか。ロボットを購入してマニュアルを読んでもみると、圧倒的に多いのがティーチングプレイバックという方法である。

ロボットを購入すると、ティーチングペンダントやティーチングボックスなどと呼ばれる、図6に示すような操作盤が付属していることが多い。ロボットは6つの関節をもっているが、それらをおのおの正転/逆転したりロボッ



図6 教示用操作盤



図7 教示の様子<sup>2)</sup>

トの位置と姿勢を独立して指示できるボタンがついた操作盤である。ティーチングプレイバックとは、図7に示すようにその操作盤を用いて、実際の作業に用いる工具と工作物を目前にしてロボット手先の工具先端を加工対象箇所へと誘導し、1点ずつそのポイントを教示してその動きを再生することである。工作物が目の前にあるので、操作盤の使い方に慣れてしまえば、大抵の作業を教示(ティーチング)することが可能である。現在のロボットの動作プログラムの生成法としては最も一般的であり、目にするロボットのほとんどがこの方法で教示された動作を繰り返し(プレイバック)ていると考えてよい。

長所としては、実績ある方法であるため、ロボットメーカーやロボットを含むシステムを開発するエンジニアリング会社にノウハウが蓄積されており、教示されたプログラムの管理や転送のシステムもよく整備されている、ユーザとしてもあまり難しいことを考えずに気軽に作業が自動化できる、などの点がある。

短所としては、教示作業中は生産ラインを占有してしまう、対象製品が変わればまた教示を1からやり直す必要が

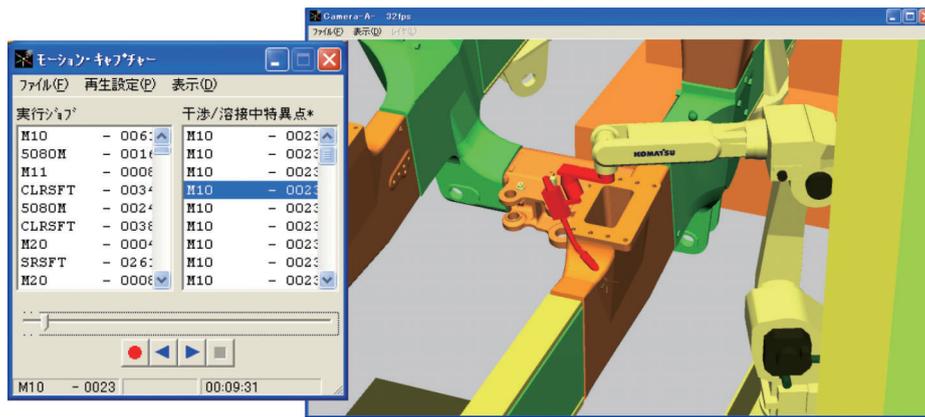


図8 オフラインティーチングの例<sup>3)</sup>

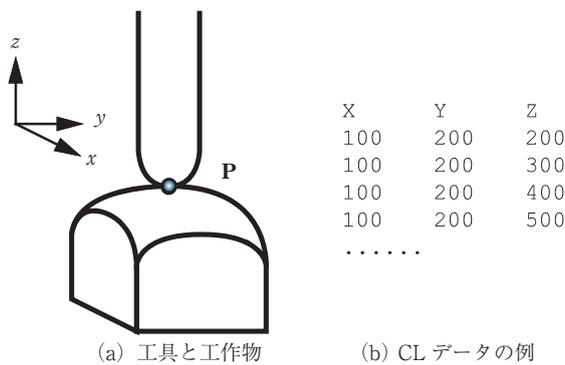


図9 3軸MC用CAMメインプロセッサ

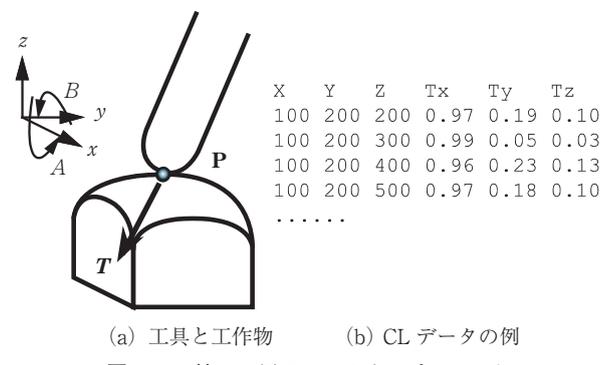


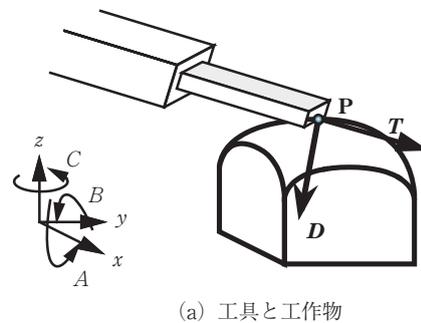
図10 5軸MC用CAMメインプロセッサ

ある、簡単とはいってもある程度教示に対してのスキルを要求される、などがある。

ライン占有の問題に対しては図8に示すような、PCを使用してオフラインで教示できる製品もあり、シミュレータとともによく導入されている。教示を簡略化する機能や、プログラム管理の機能などもあり、これらの導入でラインを占有せず効率化を図ることはできる。しかし最終的に問題となるのは、このティーチングにかかる時間と手間であり、これらを解決しない限りシステムの自動化を進めるのは難しい。

#### 4. 工作機械用 CAD/CAM 技術の応用

プログラムの自動生成といえばMC（マシニングセンタ）用のCAMが本誌読者にはおなじみであろう。本連載では、CAx技術の一つとして、他の3次元CAD应用技术と併せてCAMについて解説されている<sup>4)</sup>ので参照いただきたいが、工作物のCADデータが存在する場合は、それを利用して工具の位置や姿勢を生成するという意味では大変よく似たシステムとなる。よって、MC用CAMをベースとしてシステムを開発するのは理にかなっているといえる。一般にMC用CAMはメインプロセッサとポストプロセッサの2つのプログラムで構成されているので、以下ロボット用CAMでもそれに沿ってMC用CAMと比



(a) 工具と工作物

(b) CLデータの例

図11 ロボット用CAMメインプロセッサ

較して説明する。

#### 5. メインプロセッサ

メインプロセッサとは、工作物のCADデータに基づいて、対象となる機器の軸構成にかかわらず工作物の加工対象点における工具の位置や姿勢を生成するソフトウェアである。MCで代表的なのは3軸MCと5軸MCであるが、

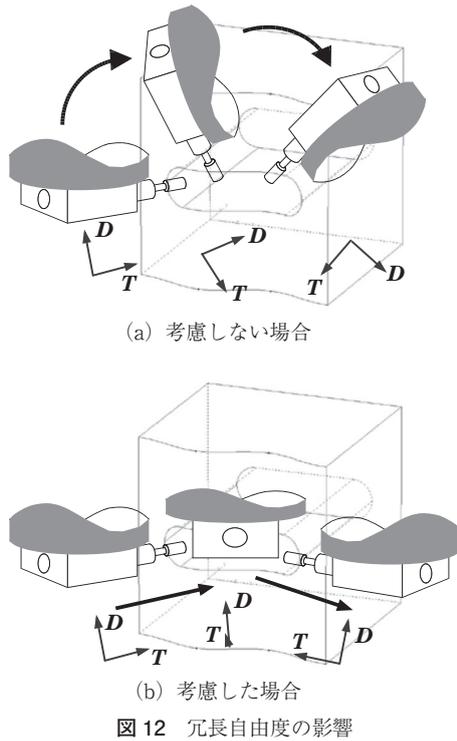


図 12 冗長自由度の影響

まずはそれらに対する CAM システムについて見てみよう。

3 軸 MC は 3 自由度をもつため、図 9 (a) に示すように、CL (Cutting Location : 工具の位置などを表す) データは位置を表す  $P$  のみとなり、例えば図 9 (b) に示すようになる。

5 軸 MC は 5 自由度をもつため、図 10 (a) に示すように、CL データは位置を表す  $P$  に加え、工具軸を表すベクトル  $T$  が加わり、例えば図 10 (b) に示すようになる。

一方、本稿で対象としているようなロボットによる作業の場合は、第 2 章で示したように最大 6 自由度必要ということになり、図 11 (a) に示すように、CL データは位置を表す  $P$ 、工具軸を表すベクトル  $T$ 、さらに工具の方向を表すベクトル  $D$  が加わり、例えば図 11 (b) に示すようになる。したがって、一般的な MC 用 CAM の流用は難しいことがわかる。

ここで、市販の CAM 製品を購入する場合でも、自前で開発する場合でも、第 2 章で示したような作業が必要としている自由度によって、実際の工具の形状と  $T$  や  $D$  をどう関連付けるかが重要である。それは、行おうとしている作業をよく分析し、その「工具の使い方」を反映することであり、工具のどこを  $P$  にし、何を  $T$ 、 $D$  に割り当てるかが、一般的なエンドミルによる切削加工よりも大変重要である。

例として、原理的には 5 自由度の指定のみで使用可能なロータリーバーのような工具を用いて面取りを行う場合<sup>5)</sup>を挙げる。5 自由度分の指定で良いからと工具軸方向を表すベクトル  $T$  のみに着目し、工具の方向を決定する 2 本目のベクトル  $D$  を何も考えずに決定してしまうと図 12

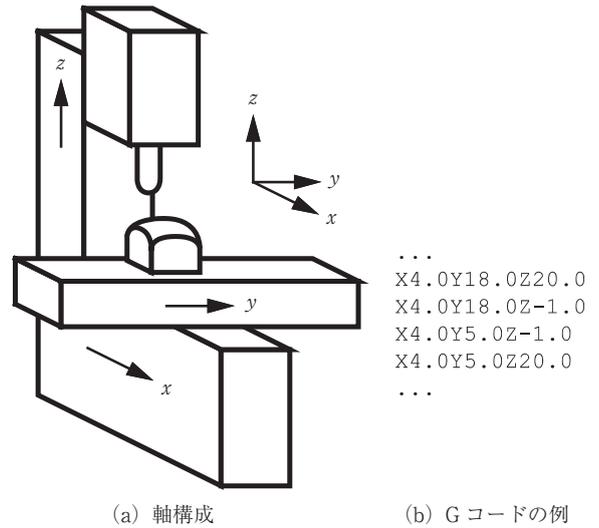


図 13 3 軸 MC 用ポストプロセッサ

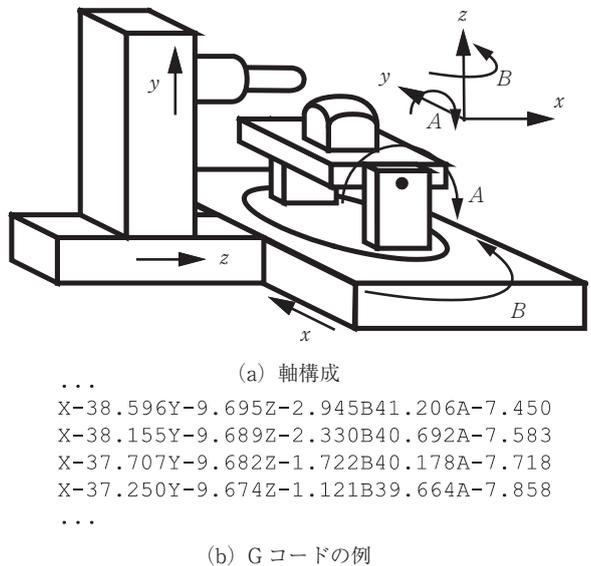


図 14 5 軸 MC 用ポストプロセッサ

(a) に示すように、不要なロボットの姿勢変化が生じて動作範囲逸脱エラーを起こしてしまうことがある。それに対し、 $D$  を適切に生成してやることで、図 12 (b) に示すように、効率よく加工を行うことができるようになる。つまり、ロボットの場合、冗長な自由度をうまく利用できるように、作業の本質的な理解がより重要ということである。

## 6. ポストプロセッサ

MC 用 CAM ではポストプロセッサとは、CL データを G コード (NC プログラム) へ変換するソフトウェアを指す。例えば 3 軸 MC 用 CAM では、図 13 (a) に示すような装置の各軸の変位量 ( $X$ ,  $Y$ ,  $Z$ ) に対応する図 13 (b) に示すような G コードが生成される。また、5 軸 MC 用 CAM では、図 14 (a) に示すような装置の各軸の変位量 ( $X$ ,  $Y$ ,  $Z$ ,  $A$ ,  $B$ ) に対応する図 14 (b) に示すような G

```

/JOB
//NAME sample1
//POS
///NPOS 87,0,0,0,0,0
///TOOL 0
///POSTYPE PULSE
///PULSE
C00000=8309,20001,-8247,-49,15840,-2079
C00001=8424,18841,-12848,-52,18115,-2103
C00002=8312,20682,-10636,-51,17893,-2073
...
  J1  J2  J3  J4  J5  J6
//INST
///ATTR SC,RW
///GROUP1 RB1
NOP
MOVL C00000 V=11.0
MOVL C00001 V=11.0
MOVL C00002 V=11.0
...
END

```

6つの軸の回転変位が  
記述されている

(a) 各軸角度での記述

```

/JOB
//NAME sample2
//POS
///NPOS 5,0,0,0,0,0
///TOOL 0
///POSTYPE BASE
///RECTAN
///RCONF 0,0,0,0,0,0,0,0
C00000=811.400,2.529,288.424,180.00,0.00,0.00
C00001=811.387,232.534,288.423,180.00,0.00,0.00
C00002=1001.382,232.529,288.420,180.00,0.00,0.00
...
  X    Y    Z    Rx  Ry  Rz
//INST
///ATTR SC,RW,RJ
///FRAME BASE
///GROUP1 RB1
NOP
MOVL C00000
MOVL C00001 PL=0
MOVL C00002 PL=4
...
END

```

手先の位置と姿勢を表す  
角度が記述されている

(手先の位置と姿勢)

(b) 位置+姿勢角度での記述

図 15 ロボット動作プログラムの例

コードが生成される。5軸 MC 用 CAM では G コードに回転変位量  $A$ ,  $B$  が含まれているものの、リニアガイドや回転テーブルなどの実際の装置の位置決め機構に対する指令に準じた値を与えることに違いはない。

一方でロボットの場合、G コードに対応するものはロボット言語で記述されたロボット動作プログラムということになる。ロボットメーカーによってその文法はさまざまだが、G コードに相当する表現ということになると、図 15 (a) に示すように、6つの回転軸のおのの角度を記述することになる。実際にロボットを動作させる際には本連載のロボット工学の基礎に関する記事<sup>6)</sup>で解説されたように、何らかの方法で逆運動学を解いて各軸への指令をサーボコントローラに送ってはいらぬ。しかし、ユーザ自身が希

望する作業の手先の位置と姿勢から逆運動学を解いて各軸への指令を算出し、プログラムに記述するのはあまりにも煩雑なので、現在市販の多くのロボットは、ロボット手先の位置と姿勢をそのままプログラムに記述できるものが多い。ただし、姿勢に関してはベクトルで記述できるものは少なく、図 15 (b) に示すように、姿勢角度表現で記述するものが多い。つまり、フォーマットとして手先の位置をロボット言語に変換するだけでなく、メインプロセッサで生成した姿勢を表す 2 本のベクトルを姿勢角度表現に変換して記述する必要がある。

姿勢角度の表現にはオイラー、固定角などがあるが、メーカーによって採用している形式が異なる。どのように姿勢を表現しているかはマニュアルなどで説明されているが、「オイラー」、「固定角」などの表現を明確に記述していない場合も多く、自分でベクトル表現を姿勢角表現に変換するシステムを構築しようとする場合には、注意が必要である。

よって、ロボット CAM 用のポストプロセッサの場合には、それぞれのロボットに合わせたロボット言語のフォーマットを理解していることと、姿勢の表現をどのように行っているのかを理解していることが肝心である。

## 7. おわりに

CAD/CAM 技術を応用して各種作業を産業用ロボットで自動化するという課題に関し、実際のシステム構築に必要と思われる問題について、工作機械用の CAM システムと比較することにより解説を試みた。自前でシステムを構築しようとする場合はもちろん、近年販売されているロボット CAM 製品を購入して使用する際にも基本的な問題は変わらないと思われるので、多少なりとも本誌読者の方々のお役に立てば幸いである。

## 参考文献

- 1) 嶋田憲司：はじめての精密工学 製品設計と製造における「自由度」、精密工学会誌, **80**, 2 (2014) 162.
- 2) 溶接ロボットティーチング作業, 東大阪高等職業技術専門校, <http://www.pref.osaka.lg.jp/tc-hiosaka/top/kyuuzin.html>
- 3) 山中伸好：製品紹介 オフラインティーチングシステム「ティーチモア」の紹介, Komatsu technical report, **51**, 1 (2005) 17.
- 4) 加瀬究：はじめての精密工学 CAD/CAE/CAM/CAT 通論 (3), 精密工学会誌, **79**, 4 (2013) 309.
- 5) 戸田健司, 浅川直紀, 竹内芳美：産業用ロボットによる面取り作業の自動化 (円筒面形状への穴加工の場合), 日本機械学会論文集 (C 編), **65**, 631 (1999) 1288.
- 6) 大隅久：はじめての精密工学 ロボット工学の基礎, 精密工学会誌, **73**, 10 (2007) 1123.