ロボットプログラミングとシミュレーション*

沖 野 教 郎**

1. ま え が き

現在、FA として指向されている工場の自動化は、一昔前の一品量産工場の自動化と違って、多種少量、中量生産、あるいはモデルチェンジへの対応可能なフレキシビリティに富んだ自動化工場である。したがって変化に柔軟に対応できる装置が要求され、ロボットはその汎用性によって FA のきわめて重要な要素としての位置を占める。しかし汎用性はソフトウェアを伴ってはじめて発揮され、しかも、そのソフトウェアはロボットのみでなく、FA システム全体の協調動作を計画するものでなければならない。

ロボットのプログラミングにはティーチング、オンラインプログラミングおよびオフラインプログラミングの3種がある。FA の進展とともにオフラインプログラミングの重要性が認識されつつあり、新しいロボット言語が次々と生れつつある^{1,2)}。またオフラインプログラミングにはシミュレータが不可欠である。ロボットは3次元空間を自在に動きまわり、機構上も NC 工作機械などと比べてはるかに複雑であるから、シミュレータなしにプログラミングすることは容易でない。ロボットのモニターあるいはシミュレータが必要とされる由縁である。ただ、ロボットのような動的対象のビジュアルシミュレーションは技術的にはかなり困難である。しかし、最近のコンピュータグラフィックス技術の急速な発展は、この方向の実用可能性に道をつけることとなり、CAD/CAM 関連企業が開発を開始しているので、間もなく続々と商用システムが現れる気配である。

本稿では、このような状況を展望するとともに、プログラミングとシミュレーションがかかえる技術的問題について述べる。

2. オフラインプログラミング

ロボットに動作を与えるに、ティーチング・プレイバック一辺倒であった時代は去り、オフラインプログラミングに関心が移ってきたのは次のような事情による。

(1) 精密なロボット動作を要求するケースが増えるに伴い、オンラインプログラミングがティーチングと共に用いられるようになり、重要な位置決めや、特定の動作をプログラミングによって与えるようになってきた。

(2) 位置決みを計算によって行うケースが増えてきた。

(3) センサからの情報によって動作を決めるロボットの知能化が進み、その情報処理がプログラミングの大きな部分になった。

(4) 複数個のロボットの協調動作や、ワークステーションなどとの連動をプログラムする必要が多くなった。

(5) ラインのフレキシビリティが高まるにつれ、頻繁な動作変更が必要となり、ラインの休止時間も短くすることが要求されているのに対して、ティーチングスタッフは十分に得られない状態になっている。

(6) 最近ではラインの設計段階で FA 全体のソフトウェア開発を行い、その中にロボットのプログラムを含むケースが増えている。

(7) CAD/CAM システムへロボットソフトウェアを結合したいとの要望の増加。

(8) 一つのロボットの動作指令を他のロボットに適用したいとの要求の増加。

以上のようにして、近い将来現場でのティーチングからオフィスでのデスクワークによるプログラミングへと主体が移って行くと考えられる。

オフラインプログラミングにおける情報の流れを考えると図1のような構成になる。これは大きく分けて次の3部に分かれる。

- ① ロボット動作の命令部
- ② ロボットおよび環境のモデル部
- ③ シミュレーションとそのディスプレイ部

原稿受付 1985 年 1 月 11 日

* Robot Programming and Simulation

** 北海道大学 工学部

図1 オフラインプログラミングシステム

このうち、①がプログラミング、②と③はシミュレータとしてまとめられる。

プログラミングのためにはロボット言語が主要な役割を果たすが、必ずしも言語によって直接プログラムを書き下すとは限らない。対話的入力やグラフィック入力の開発が進められている³⁾。しかし、プログラミングの主要な構成要素はやはり言語であり、次節に項を改めて述べる。

モデリングとシミュレーションについても、つづく節において改めて述べるが、センサの問題だけは図1の構成の中で取り扱いの困難な対象である。センサは現場で環境やロボットの現状に応じて情報を出力するものであ

表1 米、欧における主なロボット言語（順序不同）

言語名	開発者	研究商用	タイプ*	対象ロボット	特徴
AL	Stanford U.	研	C	スタンフォード シャイマンアーム	本格的ロボット言語の源流。位置、速度、力センサ、環境記述マルチアームの処理機能
AUTOPASS	IBM, Watson 研究所	研	D		マニファクチャリングタスクの記述。 平行処理、センサ処理。
AML	IBM, Watson	商	C	IBM 7565, 7535	各種データ処理、演算処理、力センサ機能。ロボット言語を越えた言語。
ANORAD	Anorad Co.	商	B	Anomatic Arm	約20のコマンドで成る比較的簡単な言語。直線、CPの動作定義機能
VAL	Unimation	商	B	Unimate, PUMA	動作要素レベル言語の範となった言語。オンライン用に開発。BASICに類似
VAL-II	同上	商	C	同上	VALの拡張
EMILY	IBM, Watson	研	B	IBM Arm	IBMの初期の言語、PTP、直線のみ。平行処理可、34命令
FUNKY	IBM, Watson	研	A	IBM Arm	PTPレベルではあるが94命令をもち、グリップコマンド、ツールコマンド、タッチセンサ機能などロボット命令豊富
PAL	Purdue U.	研	C		ALGOLに似た構造をもつ。
RCL	Rensselaer Polytechnic I.	研	B	PACS Arm	76命令から成る典型的動作要素レベルの言語
RPL	SRI	商	B	PUMA, Unimate	ロボットを含む生産システム用に開発
TEACH	Jet Prop. Lab.	研	C		協調動作処理機能
RCCL	Purdue U.	研	E	PUMA	Robot Control C Language
JARS	Jet Prop. Lab.	研	C	PUMA	PASCALにロボット言語を組み込む。
MCL	McDonnell Douglas	商	C		APTに類似した言語。ワークセルのオフラインプログラミング用
LAMA-S	IRIA	研	C		パラレルタスクの記述可
ROCOL	Leningrad P.I.	研	C	LPI-2	2本腕7自由度ロボットの制御用
ALFA	GTE Lab.	商	C		英語様言語で技術者向き
RAPT	U. Edinburgh	研	D		APTが環境を記述。タスクレベルへの実験
LAMA	MIT AI Lab.	研	C	MIT ロボット	ユーザが決めた組立法に対応。LISPに組込
ROBEX	Aachen I. T.	研	C	汎用	EXAPTの考え方に立脚
RSS	U. Illinois	研	E		サーボプロセスを定義するファンクションで構成
SIGLA	Olivetti Co.	商	B	SIGMA ロボット	VALと並ぶ典型的動作要素言語
T 3	C. Milacron	商	A	T 3 ロボット	ティーチングの補助
RAIL	Automatix Co.	商	C	Automatix	ビジョンサー情報処理を含む
HELP	DEA Co.	商	C	Allegro R.	30命令で成る構造化プログラミング言語
ARLA	ASEA	商	A	ASEA ロボット	メニューとジョイスティックによるオンライン
RFL	Siemens	商	C	Siemens	ファンクションコードでプログラミング
STORM	Olivetti	商	E		FA用スーパーバイザーシステム

* A: コマンドレベル, B: 動作要素レベル, C: 構造化プログラミングレベル, D: タスクレベル, E: その他

るから、シミュレーションのような仮想の空間すなわちモデルの世界にはなじまない。

しかし、NC などと比較してロボットのプログラミングの特殊性は、センサ情報に対応した動作をプログラムするところにあるので、この問題を避けて通ることはできない。改めて後述する。

3. ロボット言語の動向

表1は米国およびヨーロッパにおける主なロボット言語の一覧表である。我国においても最近各ロボットメーカーがそれぞれのロボット言語を開発している。

最近のロボット言語は、その機能を次第に拡張する傾向にあり、その構成は次のごとくである。

① 動作定義文：MOVE TO, MOVES などによりロボットのエンドエフェクタの位置を指定、あるいは各アームの回転角を指定する。また GRIP, RELEASE などにより作業を指示する。速度制御のための言語もある。動作定義はロボット言語の目的であるともいえる。

② 幾何定義文：POINT, LINE, CIRCLE などと幾何学的位置や運動経路を定義したり、座標系を指示する。動作定義文で用いる幾何値をあらかじめ与えるのに用いる。

③ I/O 文：センサからの情報入力、他のロボットや、ワークステーションとの協調動作のための信号入出力を指定する。

④ プログラム文：動作位置を計算によって求めたり、GO TO 文、REPEAT 文、DO 文、WHILE 文などを用いてのプログラム制御、IF 文、比較文などによるセンサ情報の判断、あるいはそれを利用しての動作制御、そして REAL, VECTOR, AGREGATE⁴⁾ のようなデータ抽象化機能などがある。

⑤ 環境定義文：ロボットがつかむ対象やロボット自身の形状を定義する。

このうち動作定義は最も基本的なもので、T3 や FUNKY のようなコマンドレベル (Command level) のプログラムでは Point to Point の動作定義のみを記述する。これに CP (Continuous path) や速度制御など豊富な動作定義文を加え、更に幾何定義文と簡単な演算などのプログラム文によって、VAL に代表される動作要素レベル (Primitive motion level) の言語が構成される。我国のロボットメーカーの言語は大部分このレベルの言語である。

次に構造化プログラミングレベルと呼んで AML⁴⁾, MCL⁵⁾, RAIL などの言語がある。上記の動作記述に加えて、普通のプログラミング言語と同様の演算機能、制御機能、モニタ機能などを持たせる方向であり、現在の

図2 ロボット言語のレベル構造図

ロボット言語の最も一般的な発展方向と考えてよい。これには純粋にロボット言語として作成されたものと、PASCAL や LISP などの一般のプログラム言語に、動作定義文と I/O 文ほかを埋め込んでロボット言語としたものがある。

ここまではロボット側にあつて、きめ細かな動作定義を行わせる目的で発展してきたものと解することができる。次のステップは、ロボット側からタスク側への移行である。すなわちロボットと関係なく作業(タスク)を記述することに重点を置き、ロボット動作をコンピュータ側で作り出す考え方に立っている。すでにタスクレベル言語として AUTOPASS⁶⁾, RAPT⁷⁾, LAMA⁸⁾ などが提案されているが、これが実用になるにはパターン認識をはじめとする知的処理の問題が解決してからである。完全なタスクレベルは当然不可能であるといっても過言ではない。

図2に示すように、タスクレベルは構造化プログラミングレベルの延長上にあるのではなく、遠く離れて存在する。中間に人間レベルとでもいうべき過程をはさみ、人間の助けを借りて行うプロセスを経るべきであろう。

すでに構造化プログラミングレベルからは、先述の環境記述文によって組立て対象を定義し、ロボットが対象をつかんで後は対象の動きを指定するなど、よりタスクレベルへ近づける方向に発展している。

図2のオペレーションレベルは、著者が CAM-1 に提案⁹⁾した言語の考え方で、オペレーションブロックと呼ぶロボット動作とオブジェクトを一体化したマクロプログラムを、組み合わせてプログラムしようとするもので、人間にとって扱いやすいプログラミングを目ざすとともに、ブラックボックスであるオペレーションブロックの内部は、できるだけ最適化やパターン認識など高度な処理を含めるようにして、タスクレベルへ一歩進めることをねらっている。

一方、この言語は図3に示すようなブロック構成になっており、ロボットだけでなく、FA の各ワークステーションに対するプログラムをロボットブロック、NC ブロック、あるいはセンサブロックなどとして並列におき、

各々がオペレーションブロックやジオメトリブロックを低位レベルに持ち、他方上位のスケジュールブロックによって全体が結合される方式である。すなわちロボット言語から FA 言語への拡張をねらったものである。現在 CAM-1 では、この提案をもとに RFP (Request For Proposal) を行った結果、日本情報サービス、日本電気の協同提案が認められ開発を開始しているが、原案のねらいは多少縮小されている。

次に、図 1 に見るように言語で記述されたプログラムは一たん中間言語に変換される。この段階ではロボット動作とセンサ相互作用のみを残して他は処理されている。

これをポストプロセッサを通じて個別のロボットに送るシステムは、この言語が汎用のものであることを前提にしたからで、各メーカが開発しているように、特定のロボットのみの言語であればポストプロセッサは必要ないが、ロボットのコントローラの機能が大きくはないので、構造化プログラミングは中間言語に変換することが必要である。中間言語の機能はほとんど限定されるので、この部分の標準化から手をつけることは自然である。我國の産業用ロボット工業会¹⁰⁾、および米、欧の各標準化機構がこの問題に本腰を入れようとしている¹¹⁾。

4. ロボットシミュレータの開発

はじめに述べたように、プログラムはロボットの動作をシミュレータで確認しながら進める。すなわちビジュアルシミュレーションである。したがって CRT ディスプレイ上にロボットが動作する様子をできるだけ正確に表現しなければならない。しかし、その目的の中で干渉を避けることにかかなりの重点が置かれているので、別に準備した干渉チェックのためのプログラムをシミュレータに含めるのが普通である。

なおロボットシミュレータには、CAD 用とロボットプログラム用の 2 種があるが、ここでは後者のみを対象にしている。

現在、ロボットシミュレータ開発段階から漸く実用段階が始まろうとしている。McAUTO 社の PLACE はすでに発売されているし、Nottingham 大学で開発された GRASP も商用に供されつつある。CAD メーカもその応用プログラムの一つとしてロボットシミュレータをとり上げ、続々と開発をはじめているので、間もなく多数の実用システムが現れると期待されている。しかし、ロボットシミュレータを完全なものにするためには、解決しなければならない幾つかの問題点をかかえている。逆にそれらは開発のための要点でもある。以下に簡単に述べる。

4.1 モデリングの問題

(1) ロボットおよび環境の形状

シミュレーションにモデリングは不可欠であり、ビジュアルシミュレーションのためには、ロボットと環境の形状モデル (Geometric Model) が必要であることはいうまでもないが、すでに CAD で発展し

図 3 ロボットプランナー

ているモデリング技術を適用して、ワイヤモデルあるいはソリッドモデルとしてそのまま適用できる。ただし、時間 t をパラメータとして導入し、ロボット機構を考慮した動的モデルとして構成する必要がある。この場合、環境モデルは CAD データベースを介して取り出し、CAD/CAM の結合をはかる方向にある。

ロボットのモデルは、例えその作成に時間を要しても以後長く使えるものであるから問題はないが、そのデータがロボットメーカにより供給されねばならないところに多少の問題がある。

(2) コントローラモデル

ロボットの制御装置のモデルであるが、厳密にはロボットの機構と力学モデルとの相互作用を考慮しなければならない。しかし現実には、多くの制御装置が線型のスローアップ、スローダウン特性をロボットに与えるよう制御しているので、厳密さを要求しなければモデルは容易に得られる。

(3) センサモデル

センサによって現実にかかるロボットとの相互作用をモデル化することは不可能であり、その意味でのモデリングはできない。しかしロボットシミュレーションの中で、センサ関連のプログラムを検証することの意味は、起こり得る状態のすべてに対応するセンサ出力を系統的に発生させ、これに対応してロボットがどのように動作するかを確認することである。そのようなプログラムを作ることは、もちろん容易でないにしても不可能ではない。

4.2 ディスプレイの問題

(1) ワイヤフレームディスプレイ

形状モデルがワイヤフレームの場合は、そのままソリッドモデルであればワイヤフレームに変換してそれぞれ表示する。ワイヤフレームのコンピュータグラフィックス技術はほとんど完成しており、ハードウェア化された

変換マトリクスをローカルターミナルに持たせることによって、10 万ベクトル程度の画面を瞬時に拡大、縮小、移動、回転などができる。これによってロボットの運動表示はほとんど支障なく可能である。ただし隠線消去プログラムのハードウェア化はまだ完成していないので、かなり見にくい画面になり、かつワイヤフレームは大きな近似であるため、シミュレータとしての機能は低い。

(2) 濃淡画面ディスプレイ

近年のコンピュータグラフィックス技術の発展は、濃淡画面の瞬時作成の道を開き、リアルなカラー写真のような図のアニメーションを、スローモーション程度ではあるがオンラインで得ることができるようになった(図 4)。

フライトシミュレータのような高価な CRT ディスプレイでなく、広く用いられる端末においてこれが可能になった意義は大きい。ロボットシミュレータとして利用可能な装置として z-Buffer (Depth Buffer ともいう)を持ち、これを瞬間的に書き換えるスムーズシェーディング用ハードウェアを持つディスプレイをあげることができる。各ピクセル毎に手前の面を z-Buffer 上に記憶させることによって陰面消去を行わせ、z-Buffer へのデータのセットは、多面体の一面毎に線形補間によって行うのが普通である。したがって、この装置を利用するには多面体化したモデルでなければならない。多面体はその境界によって決まるので、ワイヤフレーム処理と同じ機能を利用できる利点がある。ただし、ソリッドモデルからはその多面体変換を必要とする。このことは多面体近似以上の精度は望めないことを意味している。

4.3 干渉チェックの問題

数学的な干渉チェック問題は、曲面の交わり問題として軽やかたづけられ勝ちであるが、実際にはかなり困難な問題である。中でもロボットシミュレータにおける干渉チェックはまだ未解決である。先述の多面体近似されたモデル間の静止状態での干渉は、面同志の交わりの問題に過ぎないが、面同志の接触、面と稜線、稜線と稜線、面と頂点などの接触の場合には、丸め誤差ともからんで面倒な問題になる。もともとが近似された面であればなおさらである。

厳密な面はその多くが平面と 2 次曲面であり、これらの計算幾何学としての研究は進んでいるが、未解決部分も多い。

さらに、静止問題はまだ扱いやすいが、ロボットの場合は互いに運動する対象の動的干渉問題である。静止する環境と運動するロボットとの間の干渉は、運動する側の軌跡をモデリングすることによって静止問題に帰着させられる。ただし、軌跡のモデリングは容易ではない。

図 4 ロボットの動作シミュレーション

互に運動する物体の干渉は、時間を細かく区切って静止問題に帰着させる以外に方法はない。

5. あとがき

以上を要約して、ロボットプログラミングとシミュレータの動向ベクトルとしては次のことがいえるであろう。

- (1) ティーチングからオフラインプログラミングへ
の方向
- (2) シミュレーションを伴うプログラミングへの方
向
- (3) ロボットの制御言語から FA の制御言語への
方向
- (4) 言語から新しい入力方法（例えばグラフィック
インプットなど）へ変化する方向
- (5) センサ情報の知的処理を含む総合 FA シス
テムへの方向

最近の技術の進歩は速く、これらは近い将来実現すると考えられる。

参 考 文 献

- 1) R. B. Thornhill, "CAM-I Baseline Robotic Software Survey", CAM-I R-84-RSP-03, 1984
- 2) A. Rovettu, "同上 European Survey", CAM-I R-84-RSP-05, 1984
- 3) "Minutes of The Robotic Software Meeting", CAM-I M-84-RSP-03, 1984
- 4) R. H. Taylor, P. D. Summers, J. M. Meyer, "AML : A Manufacturing Language", The International Journal of Robotic Research Vol.1, No.3, 1982
- 5) L. A. Oldroyd, "MCL : An APT Approach to Robotic Manufacturing", CAM-I APTC Meeting 15-17 Sept. 1981
- 6) L. I. Lieberman, M. A. Wesley, "AUTOPASS : An Automated Programming System for Computer Controlled Mechanical Assembly", IBM J. of R & D, 19 Co nt 21-4, pp. 321-333, 1977
- 7) R. J. Propplestone et al., "RAPT : A Language for Describing Assemblies", Industrial Robot, 5-3, pp. 131-137, 1978
- 8) T. Lozano-Perez, P. H. Winstone, "LAMA : A Language for Automatic Mechanical Assembly", Proc. of 5 th IJCAI, pp. 710-716, 1977
- 9) N. Okino, "A Work Statement Proposal for CAM-I Robot Planner", CAM-I Robotics Software Project M-84-RSP-01, pp. 11-164, 1984
- 10) "ロボット言語の標準化研究報告書", 日本機械工業連合会, 日本産業用ロボット工業会, 1983
- 11) 新井, "ロボット言語の標準化", 日本ロボット学会誌 2 巻 2 号, pp. 136-141, 1984

沖野教郎 (Norio OKINO)

昭和 8 年 10 月 13 日生れ。36 年京都大学大学院工学研究科機械工学専攻博士課程修了。工学博士。36 年京都大学助手, 38 年同助教授, 42 年北海道大学助教授, 43 年同教授, 現在に至る。CAD/CAM の研究に従事。精機学会, 日本機械学会, 情報処理学会, 自動制御学会等の会員。なおロボット関係については 1983~1984 の期間 CAM-I Robotics Software Project 議長としてシミュレータを伴うロボットプログラミングシステムのプロジェクトを推進。