〔展 望〕

ロボットのオフラインプログラミングの研究動向*

長 田 正**

1. はじめに

ロボットを操作して作業をさせるにはプログラミングが必要である。ロボットのプログラミングは、オンラインプログラミングとオフラインプログラミングに大別される。オンラインプログラミングは、手続きを考えたり、作業プログラム中の数値データを決めるために実際のロボットや作業環境を利用するもので、ロボットがプログラミング過程の本質的な要素として用いられる。プログラミングシステムの開発が容易であり、プログラマが実際に状況を把握でき、テストも生産ライン中のツールを利用できるので、プログラムしやすいという大きな特長を有している。しかし一方、プログラミング過程にロボットが用いられるため生産性が落ちる、プログラムに条件分岐などを取り入れにくくレベルの高いプログラムを作成し難いなどの欠点を持っている。一方オフラインプログラミングでは、タスクプログラムの記述、シミュレータの起動、データの収集・生成などを含むプログラミングを実際のロボットを利用することなく行うことができる。オフラインプログラムは、言語の改善によりプログラミングの質を高めることができる、プログラミング中にロボットが拘束されないので生産性を高めることができる、プログラミングと実行が分離できるのでオペレータや操作対象物の安全性が確保できるなどの特長を有する一方、次のような欠点が指摘されている。すなわち、シミュレータやグラフィックディスプレイなどロボット自身および作業環境を忠実に表現するためのツールが必要である。プログラムを記述するための言語の開発と修得に手間がかかりシステム開発者やプログラマの負担が重くなるなどである。

最近では、ロボットが行うべき作業内容が複雑になるにつれ、プログラミング方式は、オフラインプログラミングへと移行する方向にあり、ロボットの知能化が進むにつれこの傾向は益々進むものと思われる。また、研究

室では、オフラインプログラミングシステムの整備は、研究遂行上欠くことのできないものだと考えられている。

オフラインプログラミングの中心は言語であり、ロボット言語の名称でいろいろなプログラミングシステムが開発されているのは周知の通りである。ロボット言語ないしプログラミングシステムは、オペレータが容易にかつ自由にロボットを操作するプログラムを書くことができるか否かによって評価され、操作の目的やレベルによっていろいろに分類されている。ロボット言語のレベルが上がりプログラマの負担が軽くなるに従って、ロボット言語に要求される機能は高くなる。なかでも、障害物回避動作計画やタスクプランニング、ヒューマンロボットインターフェイスなどはこれからのロボット言語が備えるべき重要な機能だと考えられる。また、オフラインプログラミングでは、現在のロボットの代わりをするシミュレータとプログラムの実行結果をチェックするためのグラフィックディスプレイの機能も重要である。

本稿では、オフラインプログラミングに関する重要な幾つかの項目を取り上げ、私見を交えながら今後の動向を展望することにしよう。

2. ロボット言語とその周辺

ロボットが行うべき作業を記述するための言語はロボット言語と呼ばれている。ロボット言語の分類についてはさまざまな意見があるが、動作レベル、対象物レベル、作業レベルに大別し、それぞれのレベルをさらに幾つかに細分するというのが普通のように思われる¹⁾²⁾。

動作レベルはロボット（主としてマニピュレータ）の動きを中心に作業プログラムを記述するもので、ロボットの動きについての物理的なイメージを描き易い。しかし、プログラマは、マニピュレータと操作対象物あるいは作業環境との相互関係を作業の全過程にわたって正確に把握しておく必要があり、作業を完成させるに必要なマニピュレータの動きのすべてをプログラムの形で書き下さなければならない。

対象物レベルのロボット言語では、対象物の間の相互

原稿受付 1989年5月8日

* Research Trends on Off-Line Programming of Robots

** 九州大学工学部

関係を念頭においてプログラムを記述するので、作業のイメージを描くのは容易であるが、各対象物の位置・姿勢および把握点や接近点、離脱点などのデータを含んだ正確な作業環境モデルを作成する必要がある。これらモデルの管理機能を言語に具備させなければならない。またそのためには、2つの対象物が密に結合しており、一方の動きが他を拘束しているか否かといった対象物間の結合状態などもプログラムとして記述する必要がある。しかし、対象物レベルのプログラムは、単体としての対象物を中心とした記述であるため、対象物が有している性質や特徴をうまく利用することでプログラムの負担を軽くできる可能性がある。このレベルの言語でも、プログラマと言語システムの作業分担をどのようにするかによって機能の差が現れる。たとえば、対象物を移動させる際の経路を前もって指定するのか、あるいはシステムに自動的に決定させるのかによっても大きな違いが現われる。特に、複雑な作業環境の場合には、障害物との衝突を回避しながら目的地まで対象物を移動させるには工夫が必要であり、この移動経路を求めることは障害物回避動作計画とよばれている。また、対象物のどの部分をどのようにして把握すべきかを決めることは把握計画といわれ、把握のためのマニピュレータの接近点あるいは把握した後の離脱点の決定と合わせて対象物レベルのプログラムを実行する上で極めて重要な問題である。このような移動経路や把握状態を自動的に決定するためのプランニングプログラムを組み込んだ言語が、このレベルの言語の今後の主流になることが予想される。なお、プログラマの負担を軽くするという観点からすれば、次に説明する作業レベルの言語が最も好ましいが、まだ未解決の問題が残されており、当分は、この対象物レベルの言語を中心にロボット言語は展開していくものと思われる。障害物回避動作計画問題については章を改めて論じることとする。

作業レベルのロボット言語は、作業目標を指示することによってロボットを動かそうというもので、最も高いレベルの、したがってプログラマが最も楽をでき、その代わりシステム側にとっての負担が重い言語である。このレベルの言語は、主として組立作業などの複雑な作業をロボットに行わせることを想定しており、作業目標の与え方によって更に細かいレベルに分類できる。ただ単に、'build a motor' とだけ指示してモータを組み立てさせる理想的なものから、そのモータを構成する部品相互の位置関係(状態)を与えるもの、組み合わせるべき部品とその順序関係を指定するものなどさまざまである。いずれにしても、何等かの方法で、作業を実行するためのマニピュレータへの動作命令を生成することができなけ

図1 モータの組み立て作業例

図2 モータの組み立ての木

ればならない。とくに、部品相互の位置関係から、ロボットの作業能力や作業環境で満たされるべき物理的制約などに関するルール(規則)を用いてロボットが実行すべき動作の系列を自動的に生成するシステムは、ロボットプロブレムソルビングシステムあるいはロボットプランニングシステムとよばれている。これらは自動プログラミングの一種と考えられ、AI(人工知能)研究の重要な一分野を形成している。しかし、ロボットの組立作業用プログラムの自動生成を一般的な推論や問題解決の研究対象として捉えるのはあまり現実的ではない。なぜならば、組立部品にはそれぞれ固有の性質があり、組み合わせの方法や操作のされ方などは前もってほぼ決まっているので、これらの知識をうまく利用すれば、作業目標から対象物レベルのソースコードを生成することがある程度は可能だと思われるからである。

簡単な例を示そう。図1は単純化されたモータの組立手順を示している。図2はそれを木構造で表現したもので、組立の木とよばれるものである。同図で先端節点を除く各節点は部品の結合状態を表している。なお、ボルトは複数個あるが1つの節点で代表させてある。もし構成部品を与えるだけでこの組立の木を自動生成することが

表 1 各部品の結合条件

部 品 名	結合の際に PIECE となる 部品	結合の際に BASE となる 部品
ARMATURE	(NIL)	(SHAFT)
SHAFT	(ARMATURE)	(HOUSING)
HOUSING	(SHAFT COVER BOLT)	(NIL)
COVER	(NIL)	(HOUSING)
BOLT	(NIL)	(HOUSING)

できれば、モータ組立に関する対象物レベルのソースコードを生成することが可能である。組立の木が二進木であることから明らかなように、一般に、組立作業は2つの部品の結合動作のシーケンスと考えることができる。そこで、他の部品が組み付けられるべき部品をベース (base)、ベースに組み付けられるべき部品をピース (piece) とよぶことにする。部品には、ベースだけ、あるいはピースだけにしかなり得ないものもあれば、両者の役割を演じるものもある。部品は前もって自分の役割はわかっているし、また対となるべき相手の部品の候補もわかっ

ていることが多い。これらは、いわば各部品の結合に関する局所的な知識であり、例の場合には表1のように整理される。さらに、把握点、動作軸 (結合動作時の移動軸) などの属性や結合に必要な条件などもわかっている。これらを部品に関する知識として記述し、各部品を1つのクラスにまとめておく、このような部品に関する知識や制約条件を基に、図2に示す組立の木の自動生成が可能であり、対象物レベルのソースプログラムを作り出すことができる³⁾。このシステムは、KCL上にインプリメントされた並行オブジェクト指向型ロボット言語 THERA⁴⁾⁵⁾ で記述されており、生成された対象物レベルのプログラム

[ARMATURE WOBBLE SHAFT]
[SHAFT INSERT HOUSING]
[COVER PUT-ON HOUSING]
[BOLT FASTEN COVER]

は、組立作業システム⁶⁾により動作レベルのソースプログラムに展開される。

図 3 モータ組み立て作業におけるオブジェクトの階層構造

参考のために、本システムにおけるオブジェクトの階層構造を図3に示す。

本システムは、ピースをベースに組み付けて新しいオブジェクトを生成する時にそのオブジェクトの属性やメソッドをどうしたらよいかという問題を残しているがある程度まで組立の木の自動生成が可能なことを示唆している。なお、ここで取り上げた組立の木の自動生成プロセスは言語処理系の構文木の生成のそれとよく似ており、組立てに関する知識は言語処理系における構文規則(文法)に対応していると考えられる。木構造を利用した自動組立てに関する研究はかにも発表されており⁷⁾⁸⁾⁹⁾、実用的な作業レベル言語を実現するための1つの方法として有望だと思われる。

オフラインプログラミングでは、ロボットやラインを実際に動かすことはできないので、プログラムの正しさを検証し、具体的なロボットの動きを確認するためには、ロボットや環境のモデルに基づくシミュレータやグラフィックディスプレイが必須の道具立てとなる¹⁰⁾¹¹⁾。ロボットや環境のモデルに関しては、運動学および幾何学モデルとしての機能の他に、動力学モデルの働きを備えていなければならない。マニピュレータの動力学モデルとしては種々のものが提案¹²⁾されており、実時間的なモデルの構成も可能となりつつある。一方、マニピュレータの動きや環境の変化を実時間で表示するためには、高速のグラフィックディスプレイが必要である。最近では、高速表示可能な専用のグラフィック用ワークステーションも開発されているが、物体同士の衝突検出のための干渉チェックなどはまだ完全とはいえないようである。また、もし、力センサのモデリング機能を備え、衝突検出と同時に反力検出を実時間レベルで行うことのできるシミュレータを開発することができたら、各種の拘束面からの反力を利用した微細運動計画プログラム¹³⁾¹⁴⁾をオフラインプログラミングで検証することが可能となる。

ところで、ロボット言語を構築するには、仕様を決め全く新しく設計し製作する方法と、既存の言語の上に作り上げていく方法とがある¹⁵⁾。最近では、後者の方法が多いと思われるが、その場合、基本言語を何にするか選択に迷うところであろう。また、基本言語のサブセットの形でロボット言語を作るか、あるいは既存の言語の機能の一部を利用し、新しい機能を付け加えながらロボット言語をつくる方法¹⁵⁾もある。ロボット言語に主として持たせるべき機能の内容によって、基本言語としては、C, Ada, Lisp, Prolog, Smalltalk¹⁵⁾¹⁶⁾などが選ばれる。言語製作者の好みにも依存するが、融通性がある、作り易い、手ごろで利用し易いなどの理由から、Lisp系言語にロボット向きのオブジェクト指向の機能を持たせる

というのが比較的多いように思われる。特に、オブジェクト指向の機能は、ロボット言語のレベルを高める際に必要となる知識の記述や処理の機能を実現し易く、しかも、複数台ロボットの協調作業をシミュレートするに必要な並行性を実現し易いなどの特徴があり、有用だと考えられる。

今後特に重要となると思われるのは、ヒューマンロボットインターフェイスである。これは、言語の各レベルにおいてプログラムを記述する際、およびモデルの構築の時に必要となる。高レベルのロボット言語では、命令自体が重要な意味を持つようになるため、自然言語処理的なインターフェイス機能が必要となろう。また低レベルのロボット言語の場合にも、具体的な数値データを用意してMOVE命令を書くより、言葉で動かす方が便利な場合がある。たとえば、ファジイの概念を導入し、グラフィックディスプレイを見ながら‘もう少し先’、‘ちょっと右’といった命令で比較的容易にマニピュレータを動かすプログラムを作成することが可能である¹⁷⁾。また、グラフィックディスプレイとアイコンでマニピュレータを操り、作業プログラムを作ることも十分可能だと思われる¹⁸⁾。今後の研究を期待したい。

3. マニピュレータの障害物回避動作計画

マニピュレータが作業環境内の障害物との衝突を回避しながら目標地点まで移動するための経路を求めることは障害物回避動作計画あるいは障害物回避経路の自動生成といわれる。この計画では、三次元作業空間において障害物の幾何学的なモデルが与えられると、それを移動物体の位置・姿勢を一般的に表現するためのコンフィグレーション空間(関節型マニピュレータでは一般に関節角空間が選ばれる)でのモデルに引き直し、その空間内で、障害物との衝突を避けながら目標点に到達するための点列として経路を求めるのが普通である。関節型マニピュレータでは、コンフィグレーション空間の次元が関節の数だけあるため、衝突回避のための経路を求めるのが非常に複雑となる。そこで、マニピュレータをアーム部とハンド部に分け、次数を下げた空間での問題として解くことが多い。具体的な経路は、目標点までの距離と障害物までの距離の関数として適切なポテンシャル関数を定義し、その極値を求めるか、マニピュレータが移動可能な空間(自由空間)を探索して最適な移動経路を求めるか、あるいは、両者を混合するかのいずれかの方法によって得られる。しかし、ポテンシャル関数法はポテンシャル関数の決め方が難しく、また極値問題の常として局所解を求めてしまう可能性がある。一方、自由空間探索法は、自由空間を適切な方法で分割し、しかるべき

図 4 多数台ロボットの衝突回避経路生成ならびに制御のための階層的システム構成図²⁴⁾

評価関数の下でのグラフあるいは本の探索問題に置き換えることが多いが、探索空間が広いとたちまち組合せの爆発を起こしてしまう危険性がある。ポテンシャル関数法に関しては、合理的なポテンシャル関数の設定法や局所解を避けるための種々の研究がなされており、自由空間探索法についても、状況に応じて探索領域を制限したり、何らかのヒューリスティクスを導入する方法などが研究されている。しかし、両者ともに汎用性のある手法はまだ確立されていないようである。障害物回避動作計画に関しては、尾崎¹⁹⁾、長谷川²⁰⁾らの解説もあるし、ここではその詳細にはふれず、問題点並びに今後の課題について述べることにしよう。

障害物回避経路探索は、計算幾何学の対象でもあり、経路は、一般にコンフィグレーション空間での座標の関数として求められる。しかし、マニピュレータは動特性を有しており、種々の制約条件を考慮したうえで、この経路を通るためのマニピュレータの操作量が求められなければならない。また、求められた経路は、マニピュレータが比較的自由に動ける部分と障害物の間をすり抜けるためにハンド部の姿勢が拘束をうける部分とから成ることが多い。さらに、後者においては、触覚センサの情報に基づく経路の修正が現実には必要だと思われる。動きにくい所はゆっくりと慎重に、動き易いところは高速でというのは常識的であろうし、作業の内容や障害物の

性質によっては十分な余裕をもって移動しなければならない部分とある程度の危険性を冒しても差し支えない部分とが存在することもある。すなわち、経路はマニピュレータの動特性や作業内容、環境を考慮にいれ、コンフィグレーション変数およびその速度、加速度の関数として与えられるべきで、経路探索は、マニピュレータの動的制御の立場から捉えることが望ましい。マニピュレータの動特性を考慮した経路探索は部分的には取り上げられているが²¹⁾²²⁾²³⁾、上述のことをも考慮した総合的観点からの研究はまだ少ないようである。

次に、可変環境における経路探索問題がある。作業空間内の障害物は必ずしも固定されているとは限らない。また、複数台マニピュレータが同一作業空間で動いている場合には、お互いに他の可動障害物となる恐れがある。このような変化する作業環境における経路探索問題は、重要であるにも拘わらずあまり研究が進んでいない。複雑で一般的な解を求めることが非常に困難であること、センサ情報が必要でオフラインプログラミングには適していないことなどのため、研究は直交座標型²⁴⁾や円筒座標型²⁵⁾の様な簡単な場合に限定されているようである。

図 4 は、Y. P. Chien らにより示された多数台ロボットの衝突回避経路生成ならびに制御のための階層的なシステム構成図である。図はオンラインプログラミングを前提とした構成図であるが、並行プログラミングシステム

の整備とシミュレータ機能の拡充によりオフラインプログラミングシステムとしてもある程度は実現可能だと思われる。

4. お わ り に

ロボット言語と障害物回避動作計画を中心にオフラインプログラミングの最近の動向について述べた。他の解説との重複を避ける意図もあり、筆者の周辺の研究指向に基づく狭い範囲での話題に限定しており、独断と偏見のそしりはまぬがれないかも知れない。

オフラインプログラミングは数多くの特長を有し、プログラミングシステムの主流であると考えられているにも拘わらず、現場での普及率はまだ低いといわれている。その大きな理由としては、ロボット言語が使いにくいこと、およびシミュレータやグラフィックディスプレイの性能に不満があることなどがあげられている。さまざまな立場からのアプローチにより、優れたオフラインプログラミング方式が一日も早く確立されることを期待したい。

参 考 文 献

- 1) T. Lozano-Perez, "Robot Programming", Proc. IEEE 71-1, pp. 821-841, 1983.
- 2) 井上: "ロボット言語の研究課題", 日本ロボット学会誌, Vol.1, No.2, pp. 87-90, 1984.
- 3) 長田, 原: "オブジェクト指向の概念を用いた組立行業プログラムの実現", 第28回計測自動制御学会学術講演会, 1989.
- 4) 長田, 橋元: "並列オブジェクト指向型言語 THERAを用いたロボットシミュレーション", 第6回日本ロボット学会学術講演会, 1609, 1988.
- 5) 橋元: "並列オブジェクト指向型言語 THERAを用いたロボットシミュレータ", 九州大学修士論文, 1989.
- 6) 小川: "オブジェクト指向的アプローチを用いた組立行業システムの開発", 九州大学修士論文, 1988.
- 7) L. S. Homem de Mello and A. C. Sanderson, "AND/OR Graph Representation of Assembly Plans", Proc. 5th National Conf. on Artif. Intell., pp. 1113-1119,
- 8) L. S. Homem de Mello and A. C. Sanderson, "Planning Repair Sequences Using the AND/OR Graph Representation of Assembly Plan", Proc. IEEE Int. Conf. on Robotics and Automation, pp. 1861-1862, 1988.
- 9) K. H. Chang and W. G. Wee, "A Knowledge-Based Planning System for Mechanical Assembly Using Robots", IEEE Expert, pp. 18-30, 1988.
- 10) 沖野: "ロボットプログラミングとシミュレーション", 日本ロボット学会誌, 3巻, 2号, pp. 118-123, 1985.
- 11) A. Naylor et al, "PROGRESS- A Graphical Robot Programming System", Proc. IEEE Int. Conf. on Robotics and Automation, pp. 1282-1291, 1987.
- 12) 川崎: "ロボットアームの動力学計算法", 計測と制御, Vol.25, No.1, pp. 23-29, 1986.
- 13) T. Lozano-Perez et al, "Automatic Synthesis of Fine Motion Strategies for Robot", Int. J. of Robotics Research, Vol.3-1, pp. 3-24, 1984.
- 14) M. Erdmann, "Using Backprojection for Fine Motion Planning with Uncertainty", Int. J. of Robotics Research, Vol.5-1, pp. 19-45, 1986.
- 15) S. T. Rock, "Developing robot programming language using an existing language as a base-A viewpoint", Robotica, Vol.7, pp. 71-77, 1989.
- 16) W. R. Lalonde et al, "Smalltalk As A Programming Language For Robotics?", IEEE Int. Conf. on Robotics and Automation, pp. 1456-1461, 1987.
- 17) 長田, 三苦: "日本語入力によるマニピュレータの制御", 第7回計測自動制御学会九州支部学術講演会, 214, 1988.
- 18) A. Troncy et al, "Modular Robots-Graphical Interactive Programming", IEEE Int. Conf. on Robotics and Automation, pp. 1739-1742, 1988.
- 19) 尾崎: "マニピュレータの障害物回避", 日本ロボット学会誌, 2巻, 6号, pp. 580-586, 1984.
- 20) 長谷川: "マニピュレータの障害物回避動作の自動計画", 日本ロボット学会誌, 5巻, 6号, pp. 475-479, 1987.
- 21) B. K. Kim and K. G. Shin, "Minimum-Time Path Planning for Robot Arms and Their Dynamics", IEEE Trans. on System, Man, and Cybernetics, Vol. SMC-15, No.2, pp. 213-223, 1985.
- 22) 尾崎他: "空間経路が指定されるマニピュレータ動作の時間短縮", 計測自動制御学会論文集, Vol.22, No.10, pp. 1074-1080, 1986.
- 23) E. G. Gilbert and D. W. Johnson, "Distance Functions and Their Application to Robot Path Planning in the Presence of Obstacles", IEEE J. of Robotics and Automation, Vol. RA-1, No.1, pp. 21-30, 1985.
- 24) Y. P. Chien et al, "On-Line Generation of Collision Free Trajectories for Multiple Robots", IEEE Int. Conf. on Robotics and Automation, pp. 209-214, 1988.
- 25) T. Nagata et al, "Multirobot Plan Generation in a Continuous Domain: Planning by Use of Plan Graph and Avoiding Collisions Among Robots", IEEE J. of Robotics and Automation, Vol.4, No.1, pp. 2-13, 1988.

長 田 正 (Tadashi NAGATA)

1958 年九州大学大学院修士課程修了, 59 年同大学院博士課程中退, 同年電子技術総合研究所入所, リレー制御, 統計的制御, 知能ロボットの研究に従事, 1980 年より九州大学工学部教授, コンピュータビジョン, ロボティックスなどの研究に従事, 電気学会, 情報処理学会, 計測自動制御学会, IEEE などの会員 (日本ロボット学会正会員)