ロボットシミュレーションのための グラフィックス技法*

木 村 文 彦**

1. はじめに

ハードウェアシステムとしてのロボットの能力が急速に高まり、産業界で実用的に利用されるようになるにつれ、ロボットを利用する技術、すなわち必要な作業を効率よくロボットに教示することが重要になってきた。そのために、近年産業用ロボットのためのオフラインプログラミングシステムが、数多く開発されている^{1,2)}。

そこでは、直接的な動作教示の代りに、プログラムによって間接的にロボットの動作あるいは作業を記述していく。一般に、ロボットは空間で複雑な運動をするので、それを一次元的な計算機のプログラムに書き下していくことは困難である。そこで、あらかじめロボット自身やその作業環境のモデルを計算機内に構築しておき、それらのモデルを操作することにより、必要なプログラムを自動生成する方式が普及してきた。

プログラムの自動生成の基礎として、ロボットの作業シミュレーションは重要である。実際のロボットで動作させる前に、計算機内のモデルを用いて、形状、運動、力学、制御など種々の観点からシミュレーションを行なう。

本稿では、ロボットのオフラインプログラミングにおける重要な問題である作業のプランニングや軌道生成などの話題には立入らずに、形状処理や運動学的な観点からロボットシミュレーションの機能を概説し、そのために必要となる基礎技術について解説する。このようなシミュレーションをグラフィックシミュレーションと呼ぶことにする（後述するように、重点はモデルに基づく動作指令の生成にあり、グラフィック表示ではないことに注意されたい）。

上記のような観点のシミュレーションにおいては、形状モデリングとディスプレイ技法が重要な基礎技術となる。形状モデリングにより、ロボットやその作業環境の形状やそれらの運動、衝突、組合せなどを詳細にシミュ

レーションする。また、その様子を視覚的に把握するために、グラフィックディスプレイ上に表示する。一般に3次元的な形状やその運動は平面上の表示では把握しにくいので、陰影や色を利用したり、表示を高速に動かしたりして、空間情報を伝えるような工夫がなされている。

仮に、ロボット作業のプランニングシステムや軌道生成システムが十分に能力の高いものであれば、人がロボットの動作をグラフィックディスプレイ上で確認する必要はなくなり、グラフィック表示はデモンストレーション以上の意味を持たなくなるであろう。将来は確実にそのようになることが予想されるが、現状では人の介入による対話的プログラミングが多くの場合不可欠であり、グラフィック表示が必要とされている。

本稿では、形状モデリングとディスプレイ技法を合わせてグラフィック技法と呼ぶことにするが、ロボットプログラミングの立場からは、形状モデルは表示のために作られるのではなく、動作生成の基礎となるものであることに十分注意する必要がある。

2. グラフィックシミュレーション

オフラインプログラミングにおけるシミュレーションの位置付けを述べ、特にグラフィックシミュレーションに要求される機能を検討する。

図1に、モデルとシミュレーションを基礎にしたオフラインプログラミングシステムの機能構成を示す³⁾。

最近良く言われるように、CADシステムとロボットオフラインプログラミングシステムとの結合は重要である。ロボットや作業環境のモデルは、予めCADシステムなどを用いてモデル化されている必要がある。考慮する処理の範囲により、形状や運動拘束、力学的条件のような種々の技術的情報など、必要な情報がモデル化されていないなければならない。表現される情報の中味や表現の枠組みは、次に述べる作業対象物の場合とほとんど同様である。

作業対象物のモデルは、指定された作業毎に、ロボットプログラミングシステムに与えなければならない。生産システムを統合的に考えると、そのようなモデルは、

原稿受付 1988年3月18日

* Geometric Technique for Robot Simulation

** 東京大学工学部

製品設計の結果として与えられるはずである。すなわち、CADシステムとの結合により、モデルデータが容易に与えられることになる。従来の多くのシステムはこのような結合を欠いていたため、モデル作成に多くの手間を要していた。もちろんロボットのプログラミングに必要な情報が全て製品設計段階で生成されているとは限らず、ロボットプログラミングシステムの中でも、補足的なモデリングを必要とする。

上記のようなモデルに含まれるべき情報を整理してみよう。グラフィックシミュレーションに必要なデータは次のようなものであろう。

●形状モデル（単一部品形状の表現）

●組合せや運動拘束関係の記述

これらについては次節で詳しく検討することにする。きちんとしたロボットの動作プログラムを生成するためには、これだけではもちろん不足であり、次のような技術的情報を必要とする。

●寸法・公差・はめあい

●表面性状

●材料

●形状特徴（穴、溝、などの形状の特徴の記述）

これらの情報に応じて、ロボットは、センサ情報なども用いながら、位置や力などを適切に制御して必要な作業を行なっていく。

以上のような情報は、CAD/CAM の分野で確立しつつあるプロダクトモデリングの概念と枠組みを用いて、表現し、操作していくことができる^{4,5)}。

本稿では、グラフィックシミュレーションのみ考えているので、以下では形状モデルとその組合せや運動拘束関係などだけを考える。

作業のプランニングや動作軌道の生成においては、上記のモデルを利用して、計算機プログラムや人が、ロボットや作業対象物の運動や状態をシミュレーションしながら、適切な情報を生成していく。この過程で、グラフィックシミュレーションに要求される形状モデリングの機能は、以下のようなものであろう。

●与えられた時間の関数として、形状モデルを操作し、運動状態をシミュレーションすること。

●拘束条件を破らないように運動させ、また他物体との衝突をきちんと検出できること。

従来の形状モデルは、主に単一の物体を扱っており、また連続的な動きを直接表現できる機能に乏しい。ロボットの応用では、複数の物体が動き回ることが特徴的であるが、現状の形状モデリングシステムは、このような状況を扱うのが苦手である。このような問題点について、次節で詳しく考える。

図1 オフラインプログラミングシステムの構成

はじめに述べたように、ロボットなどの動きをグラフィックディスプレイ上に見やすく表示することは、ここ当面の実用上の重要性はあるものの、ロボットプログラミングに固有の問題ではなく、また特に困難な技術的問題でもない。要するに、現実感のある表示が高速にできればよい。最近では、高機能なハードウェアも手軽に利用できるようになって、複雑な作業の様子をリアルタイムでシミュレートすることも夢ではなくなってきた。このような技術について第4節で考える。

3. 形状モデリング

上述のように、ロボットシミュレーションにおいて形状モデリングは重要な役割を果たす。形状モデリングについては、既に多くの解説があるが^{6,7)}、ロボットプログラミングに関連する事柄を概説し、特有な機能を考察する。

3.1 形状モデリングの分類

形状モデリングには、線（Wireframe）モデルや面（Surface）モデルなどのような簡易なモデルから、立体の性質を完全に表すような立体（Solid）モデルまで、色々なモデルが考えられる。

線モデルは、文字どおり線分のみで形状を記述していく方式であり、立体形状を完全には記述できないため、立体間の衝突を自動的に検出することなどができない。しかし、処理が簡単で応答性の良いシステムが実現できるため、会話型のロボットプログラミングシステムでは良く用いられている。作業対象物やロボットを、着目す

る部位に固定された座標系で代表させるシステムがあるが、これなども線モデルの一種とみなすこともできる。これらのシステムにおいて有意義なプログラミングをするためには、常に人が形状情報をきちんと把握しておく必要があり、線モデルは形状データの一部の保存場所としてのみ有効なのである。

上記のように、線モデルを用いる場合には人の形状認識がプログラミングの基礎となるので、特に人に対する形状視認性を良くする工夫が大事である。そのために、グラフィックハードウェアの機能により、線モデルの高速な回転や、遠近による輝度変調などを施して、3次元的情報を把握しやすくすることなどが、市販の高級システムで実現されてきた。しかし、後述するように、立体の陰影表示を高速に実現するハードウェアが急速に普及してきたので、これらは時代遅れになりつつある。線モデルベースのシステムは、むしろマイクロコンピュータなどを用いた簡易なロボットプログラミングのための道具として生残るであろう。

複雑な曲面形状を扱うような面モデルは、ロボットプログラミングではあまり現れなかった。作業対象物として、自動車のボディなどを扱う時に必要となる。この際に面モデルに対して必要となる操作は、面間の交線計算、面の法線などの幾何学的性質の算出などが主なものであり、下に述べる運動制御の問題を除いて、ロボットプログラミングに特有なものは少ない。

複雑な曲面形状は、通常 Bezier 式や B-spline 式などのパラメトリックな自由曲面式により表されており、曲面のパラメータと実空間における位置や距離との対応は一般的に複雑でわかりにくい。自由曲面上をロボットの先端が指定された速度で移動するように制御することなどを考える場合には、パラメータの変化に対する実空間での移動量を算出するような比較的面倒な計算を必要とする。

上記のような面モデルの利用はまだあまり進展していないが、今後生産システムへの応用が盛んになるにつれ、重要になるものと思われる。

立体モデルは、立体としての完全な情報を計算機的に取扱おうとするもので、ロボットのプログラミングにとって最も重要なものである。線モデルなどと異なり、形状の認識を計算機が行なうことができるため、衝突検出などの操作を自動的に行なうことができ、プログラミングの自動化のレベルを上げることができる。反面、処理が複雑となり、良いモデリングシステムがなかなか得られなかったため、立体モデリングベースの実用的なロボットプログラミングシステムは数少ない⁷⁾。

内部データ表現と処理について、立体モデリングの方

式を分類してみると、次の3種に大別される⁷⁾。

●Decomposition Model :

空間を予め定められた基本要素の集りで分割し、立体の占める部分をマーク付けすることにより立体を表す。空間を一樣に分割する方式、形状に応じて分割寸法を可変にする方式、形状を構成する基本セルを用意する方式など、各種の方式が知られている。基本演算が単純で、一樣、理論的に明快であるが、一般に形状表現は近似となり、データ量が多くなる。ハードウェアで高速処理できる見通しがある。

●Constructive Model :

基本立体あるいは基本曲面を、種々の形状構成操作で結合して、立体を表す。いわゆる CSG(Constructive Solid Geometry) モデルである。表現が簡潔かつ厳密で、他モデルへの変換が可能であるが、一般に、必要なデータを導出するまでに時間がかかる。

●Boundary Model :

立体を規定する境界により表現する。立体の頂点、稜線、面などの関係を表すデータ構造を持つ、いわゆる B-rep (Boundary Representation) モデルである。形状構成操作や応用に応じて、種々のデータ構造が提案され、利害得失が論じられている。表示データや応用で必要となる形状データをほとんど直接的に得ることができるが、一般に処理は複雑である。

CAD/CAM システムへの応用においては、モデル形状を人の要求に従って生成し、変更していく操作が主となるため、上記のどのモデルを用いても、単一の方式では要求を満足させることは難しく、これらの方式を組合せて用いることが多い。実用システムでは、Constructive Model を基礎として、Boundary Model や Decomposition Model を生成するものが見られる。

ロボットプログラミングにおいては、形状生成や変更は補助的であり、立体の運動や衝突検出が重要となる。運動物体に対して Decomposition Model の方式を適用することは一般には困難なので、これらは運動しない作業環境表現として補助的に用いられ、必要な計算を効率的に実行するのに役立てられる。主な表現形式としては、Boundary Model か Constructive Model のどちらかを用いるものが多い。精密な作業に関連する部分(例えば、精密はめあい部分など)を除けば、加工で必要とされるような正確な形状情報は必要ないことが多いので、いかにして簡略な表現を得て、処理を高速化するか、という点が重要になる。

立体モデリングシステムは、通常個別の立体を扱い、立体の組合せ関係や運動拘束関係を扱う手段を十分に持っていない。しかし、多関節型のロボットの運動を計算

する程度であれば、立体間の関係をトリーやグラフ構造で表し、通常のロボットプログラミング言語に備っているような座標変換計算をそれらの構造に対応させて行なえるようにすれば、大体うまく処理することができる。実用的な立体モデリングシステムは、上述のような機能を備えているのが普通である。

機械組み立て品の構造を分析して、組み立て方向を検出したり、運動の拘束条件より、運動の自由度を抽出したりするためには、より完備した関係記述法を必要とする。数式処理による方法⁹⁾や、述語論理記述を基にする方法⁸⁾など、種々の方法が試みられてきているが、未だ、完全なものにはほど遠い。これらの問題も、CAD/CAMにおけるプロダクトモデリング研究の一環として盛んに研究されており、将来の成果が期待されている。

3.2 シミュレーションの機能

ロボットプログラミングにおいて重要となるシミュレーションの機能として、立体の運動と衝突検出について、手法を展望し、問題点を探る。

ロボット動作の幾何学的シミュレーションに必要な計算は、単純な運動学的計算であり、計算機化するのに特になにも問題はない（ロボットの先端の位置や姿勢を指定して、各関節のパラメータを求めることや、軌道生成などの問題は考えていない。これらは、シミュレーションの以前の問題である）。実用的に現れる程度の複雑なモデルであっても、十分に高速に、指定された位置、姿勢に次々とモデルを移動させていくができる。但し、人がその動きを見られるように、運動と同期させて表示していくことは、必ずしも簡単ではない。隠れ線を消去しない稜線による表示ならば難しくないが、視認性が良くない。表示の問題については、次節で考える。

運動の問題というよりは、軌道生成の問題ではあるが、曲線軌道上の運動について考えてみる。一般に、曲線上を指定された姿勢を保ちながら、指定された速度で移動するような運動を生成することは、意外に良く扱われていないようである。複雑な曲線は、パラメトリックな自由曲線で表されることが多いが、曲線のパラメータを単純に時間とみなして計算したのでは、うまくいかないことは良く知られている。曲線長を計算し、与えられた速度分布に従って時間を計算し、それをパラメータと関係付ける、といった計算が必要となる。

次の衝突検出の問題とも関連するが、形状モデリングにおいて、時間あるいは状態の変化を一般的かつ効率良く扱う方法はまだ確立しておらず、今後の研究が必要である。特に、時間が連続的に変化していく時の扱いは難しい。

立体間の衝突検出は、ロボットのグラフィックシミュ

レーションにおいて、最も重要で困難な問題である。空間内に静止している立体間の干渉検出も面倒な問題であるが、ロボットの問題においては、複数の立体が動き回るので、一層複雑となる。

3次元空間の位置関係は、いかに工夫しても、2次元のディスプレイ画面上で人が認識することは難しい。複数の動物体の衝突をもれなく見つけることは不可能に近い。従って、たとえ会話型プログラミング方式をとっていても、立体間の衝突検出は、是非とも完全に計算機による自動処理を実現したいものである。

上のような状況から、計算機による衝突検出においては、指定された精度の範囲内で、起り得る衝突を決して見逃さないことが要求される。通常の機械の組み立て作業において、例えば 0.1mm のクリアランスで衝突を回避しても、計算機のモデルと実際の状況との誤差を考えれば、実際的でない。クリアランスの指定がきちんとしてできることが望ましい。また、衝突が起きる時にその回避計画を立てられるように、衝突の起きる時刻や場所を同定できることが必要である¹⁰⁾。

今までに試みられた手法を以下に展望してみよう。一般に、計算を高速化するために、形状は多面体で近似されるのが普通である。上に述べたように、現実と計算機内のモデルとは完全に対応がとれていないので、衝突の観点から安全側に、一定の近似度で形状を多面体近似することは極めて妥当であると考えられる。

広く行なわれている方法は、立体の連続的な動きを、一定の（空間距離的あるいは時間的）間隔でサンプルされた離散的な空間位置で近似する方法である。そして、各位置で立体間の干渉を検出する。この方法は、立体間の静的な干渉検出手法があれば、容易に実現できる。しかし、元々膨大な計算を要する干渉検出を多数回繰返すことになるので計算時間がかかる、運動をサンプルする時の間隔を大きく取りすぎると起こり得る衝突を見逃す可能性がある、などの欠点がある。

静的な立体間の干渉検出は、立体間の集合演算による形状構成操作などに必須の機能で、完備した立体モデラならば、基本機能として必ず備えているべき機能である。これの実現手法については、立体モデリングの教科書などに詳しい⁷⁾。計算を効率的に進めるために、干渉しそうな所を概略計算により素速く求め、しかる後に必要な所だけ、詳しく計算するのが普通である。立体形状を、球、円柱、角柱、凸形状やそれらの集合として安全側に近似することがよく行なわれる。また、Decomposition Model 的な考え方をを用いて、一定の大きさに仕切られた空間セルと立体との干渉を効率よく検出していく方法も有効である。

運動を離散的に近似しないで、運動立体が空間を掃引してできる掃引立体を求め、それらの立体の干渉を静的に検出することも行なわれる。一般に掃引立体の表面をなす運動の包絡面は、単純な運動でなければ複雑な面となり、掃引立体の厳密な構成や干渉検出は容易ではない。また、干渉時の時刻などを同定するのも面倒になる場合がある。この手法は、比較的単純な運動の場合に有効なので、数値制御工作機械における工具と工作物との衝突検出などに用いられている例が多い。

上記の計算を簡略化するために、2次元で知られている手法¹¹⁾を拡張して、次のような試みも行なわれている。一般にロボットが動きまわり、作業対象物は静止している。ロボットは、通常は、折れ曲った棒のような形状をしている。そこで、ロボットを、太さのない折れ線で近似し、太さ分の寸法だけ作業対象物の寸法を拡大する。そして、折れ線が運動して構成する掃引面と拡大された静止立体との干渉を検出しようというものである。この場合には、面と立体の干渉計算になるので、計算が楽になる。

最近では、立体同士が衝突する時の様子を、面と頂点、および稜線と稜線との交わりとして分類し、直接算出することの有効性が報告されている¹⁰⁾。

立体間の衝突検出問題は、基本的問題であるにもかかわらず、まだ理論的な解明の余地を残しており、今後の研究が必要である。なお、作業プログラミングなどの研究の一環として、衝突を起こさない軌道の生成法（例えば、configuration space による方法¹¹⁾）などが論じられているが、それらの基礎としても、ここで述べた事柄は重要である。

4. ディスプレイ技法

最近のコンピュータグラフィックスのソフトウェア、ハードウェアの進歩は目覚ましい。高解像度のカラー表示面を持つグラフィックワークステーションが手頃な価格で手に入るようになってきた。表示を高速に行なうためのグラフィックハードウェアの機能も急速に向上している。

ロボットシミュレーションにおけるグラフィック表示において要求されることは、高速、高精度な表示、特に3次元的な動きのある立体のリアルタイム表示である。ここでリアルタイムとは、実ロボットの動作と同程度の速度で動きが表示できることを指す。

純ソフトウェア的に、上記の事を達成するのは容易ではない。最近では、立体の面情報を考慮して、立体の陰影表示を高速に実行するグラフィックハードウェアが容易に入手可能となってきたので、これらを利用すること

は有効である。

これらのハードウェアは、通常凸多角形程度の単純な多角形の集合として規定される多面体を処理の対象としている。そこで、ロボットシミュレーションに用いるためには、対象となる立体のデータをハードウェアに適合するように変換する必要がある。また、運動に従ってデータを逐一更新していく必要がある。現在のハードウェアは、このような目的には十分に強力ではなく、実用的な複雑さのモデルの動きをリアルタイム表示することはまだ困難である。今後は、より強力なハードウェアを出現させると共に、運動物体の表示に適合した表示アルゴリズムの開発にも努める必要がある。

現在のグラフィックハードウェアの多くは、表示面の各要素毎に奥行方向のデータも持たせ、視点により近いデータが入る度に、表示データを更新していくという、Z-buffer 法、あるいは depth-buffer 法と呼ばれる手法を用いている。この方法は、画面上で隣り合うデータ間の、あるいは運動データで合い前後するデータ間の関連を考えなければ、実現は単純である。処理能率を良くするため、できるだけ上記のような関連を考慮するような工夫がなされつつある。

生成された軌道を見て確認する、という目的のためには、リアリスティックな2次元的な表示が適しているとは必ずしも言えない。軌道がよくわかるように、その表示法を工夫するとか、場合によっては、立体視などの手法も有効であるかもしれない。これは、計算機による表示法の研究というより、人の認識能力に関する問題であり、マンマシンインタフェースの研究が一層必要とされている。

5. お わ り に

ロボットのオフラインプログラミング、特にシミュレーションなどに関して、形状モデリングやそのディスプレイ技法について考えてきた。これらの技術については、既に機械系の CAD/CAM の分野で膨大な研究がなされており、それらの成果を十分吸収して、利用すべきである。しかし、空間的な運動の扱いについては、ロボット固有の問題があり、今後追及されるべきであろう。将来的には、プロダクトモデルというような形式を通じて、CAD/CAM とのより緊密な統合化を考えていくことが必要であろう。

参 考 文 献

- 1) A. Storr & J.F. McWaters (eds): Off-line Programming of Industrial Robots, North-Holland, Amsterdam, 1986
- 2) 沖野教郎: ロボットプログラミングとシミュレーション, ロボット学会誌, Vol.3, No.2, pp.34-39, 1985.

- 3) T. Sata, F. Kimura, H. Hiraoka, H. Suzuki & T. Fujita: Comprehensive Modelling of a Machine Assembly for Off-line Programming of Industrial Robots, in 1), pp. 19-33, 1986
- 4) 木村文彦: CAD/CAM 統合化のためのプロダクトモデリングと技術情報処理, 日本機械学会誌, Vol.89, No. 815, pp. 1157-1163, 1986
- 5) 伊藤公俊: 対象の計算機モデリングと設計, ロボット学会誌, Vol.4, No.4, pp.101-108, 1986
- 6) 木村文彦: 形状モデリングと CAD/CAM, 精密工学会誌, Vol.53, No.2, pp. 361-364, 1987
- 7) M. Mantyla: An Introduction to Solid Modeling, Computer Science Press, 1987
- 8) 嘉数侑昇: CAD/CAM とロボットの接続, ロボット学会誌, Vol.5, No.3, pp. 64-69, 1987
- 9) R. Popplestone, A. Ambler & I. Bellos: An Interpreter for a Language for Describing Assemblies, Artificial Intelligence, Vol.14, No.1, pp. 79-107, 1980
- 10) 岡野彰, 川辺真嗣, 嶋田憲司: シミュレーションによる移動物体間の衝突チェック, ロボット学会誌, Vol.6, No.1, pp. 35-41, 1988
- 11) T. Lozano-Perez: Task Planning, in "Robot Motion: Planning and Control", MIT Press, 1982

木村文彦 (Fumihiko KIMURA)

昭和 20 年生. 昭和 49 年東京大学大学院博士課程修了. 同年電子技術総合研究所パターン情報部入所. 昭和 54 年より東京大学工学部精密機械工学科に勤務, 昭和 62 年同教授. マン・マシン・システム, コンピュータ・グラフィックス, 形状モデリング, CAD/CAM などの研究に従事. 工学博士. IFIP-WG 5. 2-5.3 委員. ACM, 通信学会, 精密工学会, 情報処理学会などの各会員.
(日本ロボット学会正会員)