

ロボット制御プログラム開発支援システムの研究

田 口 幹*

Development of a Computer Aided Robot Programming System

Kan TAGUCHI

Abstract

In this paper, SimKIT or the simulator of a robot system developed on a personal computer is treated. SimKIT is a graphic simulation system with the 3D-graphic display and the object modelling process.

Recently, the off-line teach-in system for robot programming or the FA-simulation system have become essential in the production systems including robots, not only for the productivity but for the safety of the operators.

However, these systems are usually developed on main-frames or even on EWSs. Because of lacking in data capacity, computing speed, graphic capability, personal computers have not been used for such simulators.

Although the personal computer has such defects, SimKIT is realized to facilitate the operations with many considerations in software and man-machine interface without any special devices.

Using newly proposed calculation algorithm for the coordinate transformation, SimKIT can display the 3D-perspective graphics in real time. That makes the operator certify the input data immediately.

SimKIT adopts a new functional data structures for saving memories and facilitating the operations. A menu guidance method is also adopted to decrease the operations.

According to these structures and considerations in software, SimKIT has the enough capability in the simulation of a robot system including peripheral machines, even on the personal computer.

As a result, the author concludes that SimKIT can be put to practical use as a simulator for robot system on a personal computer.

1. はじめに

ロボットシステムのシュミレータは、ロボットの普及にともない、実際のロボットの稼働を妨げることなくロボット動作の教示を行うためのオフラインティーチンシステムや、総合的なFA シミュレーションシステム、あるいは設計から製品製造までを統括的に自動化するCIM (Computer Integrated Manufacturing)の一部として、教示作業者の安全確保と、生産シ

ステムの自動化に必要な不可欠のものになってきつつある。

しかし、このようなロボットシステムのシュミレータには、単にロボットのみ動作シミュレーションを行うだけでなく、ロボット自身と周辺機器などのシミュレーション対象のモデリングデータの入力や、シミュレーション結果の表示のためのグラフィック表示等、高度のデータ処理能力が要求される。そのため、コンピュータもメインフレームやEWS等を利用しなくてはならず、導入のための手間やコストが大きく、大規模な生産現場や一部の研究機関での利用にとどまっているのが現状である。

平成3年9月5日受理

* 機械システム工学科

これまでに発表されたパーソナルコンピュータによるロボットシミュレータは、メモリ容量や処理速度に難があり、各種ロボットに対応する汎用性や、グラフィック表示速度の面で、あまり実用的とは言えなかった¹⁾。

筆者は、このような現状をふまえ、最近のパーソナルコンピュータの性能の向上により、パーソナルコンピュータを利用しながらも、マンマシンインターフェースとしてのデータ入力などの操作性、及びグラフィック表示速度などを改善することで、実用に耐え得る性能の汎用のロボットシステムのシミュレータが実現可能であると考え、「SimKIT」と名付けたパーソナルコンピュータによる汎用のロボットシステムのシミュレータを開発してきた。

本報告では、この「SimKIT」について報告する。

2. SimKITの構成

ロボットシステムのシミュレータは一般に、モデリング機能、立体図形表示機能、シミュレーション機能、データ転送機能などのデータ処理の基本機能によって構成される。

SimKITでも、Fig. 1に示すように、これらの機能を

すべて含み、実用的なレベルパーソナルコンピュータ上に実現するために、主に以下に示すようなソフトウェア的な工夫を行った。

まず、基本的に計算速度を向上させるために内部表現を固定小数点方式とし、座標変換などで必要な三角関数も固定小数点のデータテーブルとする。これによって副次的にメモリの節約にもなる。固定小数点としても16 bitsの分解能があり、必要な精度は十分確保できる。

ロボットの各関節毎の運動解析は基本的に座標交換に帰着できる。一方、透視投影法による立体図形の表現も立体データからCRTに表示するグラフィックデータを生成するのも座標変換であるからロボットシミュレータでの演算は基本的にすべて座標変換行列の演算になる。さらに、座標変換演算を、整数演算だけで行う手法を用いながらも十分な精度を得る固定小数点法による計算法を使用することにより、数値演算プロセッサなどの特別な装置を追加することなく、高速な立体図形表示を実現する。

また、種々の条件に対応する汎用性と、ロボットの機構設計支援などの拡張性を考慮しながら、メモリ使用量を節約するために、データ構造化の手法を用いて内部データ分類と機能的な構造化を行う。

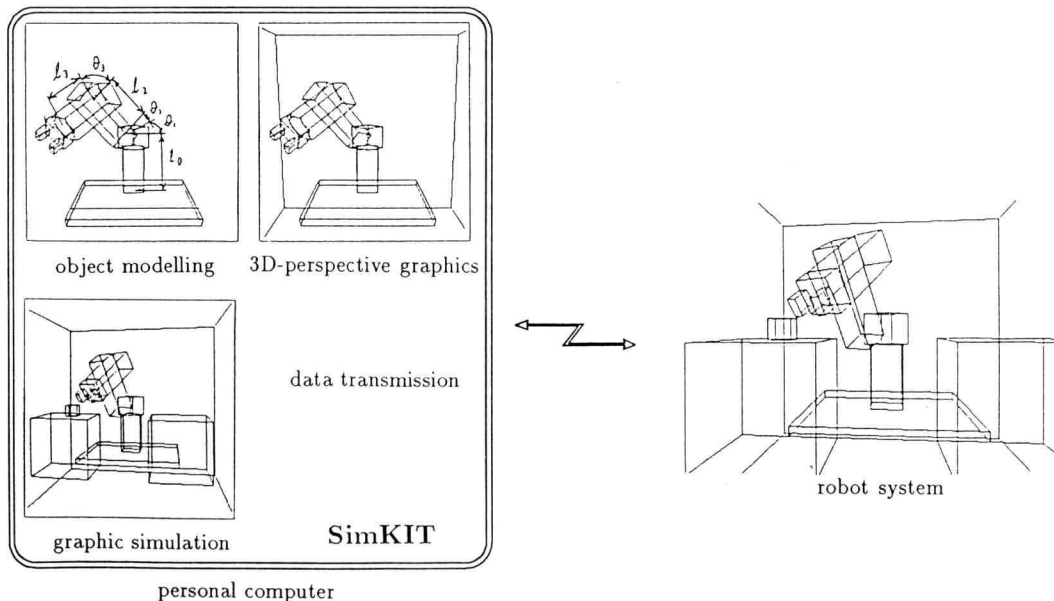


Fig. 1. Configuration of SimKIT

さらに、操作性の安定と向上を図るために、立体図形表示をデータ入力に遅延なく追従させたり、データ入力の手順に一貫性を持たせるなど、操作方法の統一化をはかる。

以下に、その詳細を述べる。

3. ソフトウェアの高速化手段

3.1 固定小数点表現

前節で述べたように、SimKIT では、演算処理の簡素化を行うために、16 bits の固定小数点演算と、座標変換に必要な三角関数の固定小数点化したデータテーブルを採用する。

多くのパーソナルコンピュータに登載されている

16 bits CPU は、最大 16 bits の整数値による四則演算を直接処理できる。乗算に関しては、16 bits 同士の乗算の積が最大 32 bits になるが、CPU 内部では乗算の結果は 32 bits に拡張しており、桁あふれなく有効に利用できる。逆に、除算は商が 16 bits の数に収まる範囲で被除数に 32 bits までの数を与えることが可能である。

固定小数点表現では、目的の小数値を整数値として扱い、乗算では積の、除算では被除数の位取りによって修正すればよい。

座標変換演算を行う場合には、座標変換行列の要素は、乗数、被乗数のどちらかが必ず正弦または余弦の値であるため絶対値が 1 以下の小数である。したがって、16 bits のうち最上位 bit を符号として、15 bits の

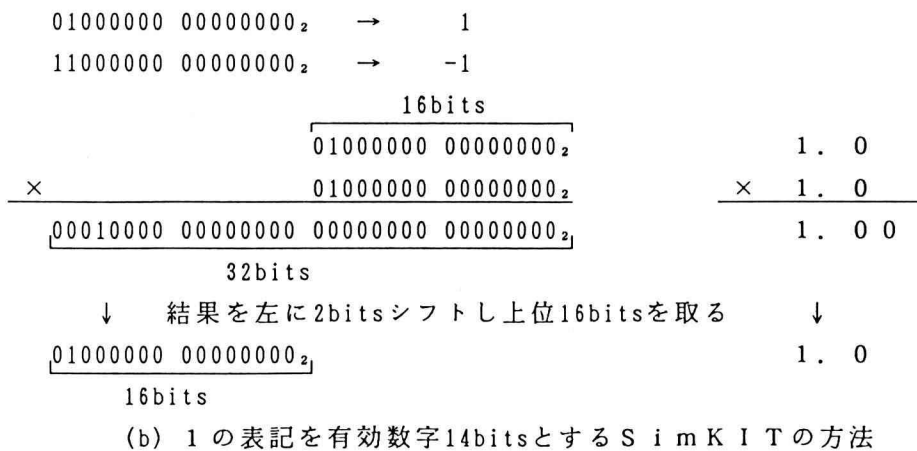
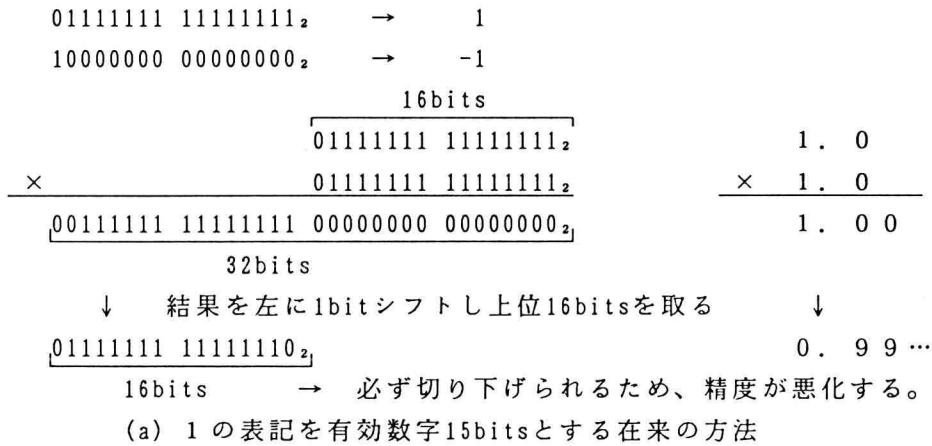


Fig. 2. Fixed-point representation

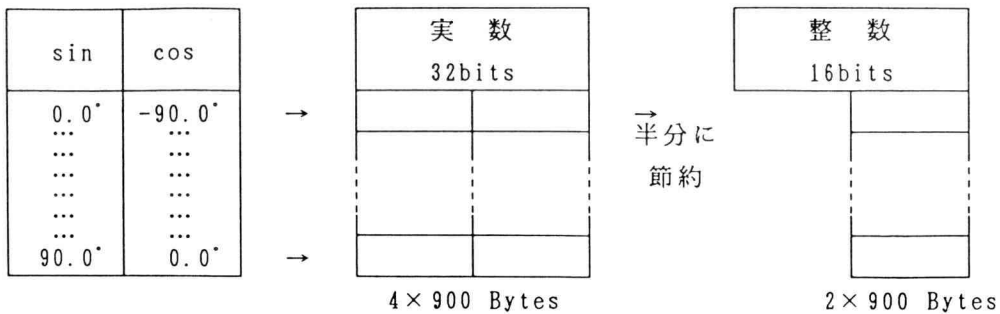


Fig. 3. Data table in fixed-point representation

最大値が1となるような固定小数点表記とすると、この方法による最大の計算精度が得られる。

これは固定小数点演算の一般的な方法である。しかし、この方法は1×1を正しく1として計算することができず、この計算が行われる度に計算精度が悪化してしまう。

SimKITでは座標変換演算を頻繁に行い、座標変換行列の要素が1のものも多い。そこで、1を正しく表現できるように、小数点以下の値の有効数字が実質最大14 bitsの有効数字となる Fig. 2のような固定小数点表記を採用する。ロボットの動作範囲内での動作精度は、通常10進数で4桁以内で表すことができるので、これでもロボットのシミュレーションには十分である。

三角関数の値をデータテーブル化する場合も、この表記方法を用いるため、データテーブルを浮動小数点の実数型のままとした場合と比べ、メモリ容量の節約ともなる (Fig. 3)。

3.2 座標変換行列による立体図形表示の高速化

ロボット動作を数学的に解析する手法にはベクトル解析を基礎とする方法と Denavit Hartenberg の方法とが主流である。この両者は各関節毎に設定する関節座標系の設定法に特徴があるが、いずれにしても行われる演算は、各関節座標系への座標変換行列の演算に帰着する。SimKITではこれらのどちらの方法でも使用できるように相互の変換を内部に用意する。

一方、ロボット動作のシミュレーション結果の確認には、立体図形表示を連続して行うアニメーションが便利である。

立体図形表示の高速化の手法は、一般には、演算時間のかかる陰影・隠線処理などを省いた、透視投影に

よるワイヤフレーム表示を採用したり、また、同じ演算処理の繰り返しを極力避けるために、座標変換行列演算を先行して、その行列によって座標ごとに変換する方法を採用するのが一般的である。これらの方法は、視認性や演算精度が多少犠牲になるが、シミュレーションには十分で、SimKITでもこれら透視投影によりワイヤフレーム表示と座標変換行列演算を先行する方法を採用する。

従来 EWS などに上に開発されたロボットシミュレータでは、アニメーションはあらかじめ演算して用意した表示内容を、次々展開して表示する方法をとっている。これは立体図形表示用のデータ生成には、比較的演算時間のかかる陰影・隠線処理を行った上、座標変換を多用しているためと、事実上メモリ容量のことを考慮する必要がないためである。この方法では表示は高速にできるが、パーソナルコンピュータではメモリ容量が少ないので、表示内容をあまり蓄えられず、逐次演算して表示する方法をとらなくてはならない。逐次演算する方法では、多少計算時間がかかるが、立体図形表示をデータ入力に常に追従させることができ、オペレータが操作しやすくなり、入力ミスを防止できる利点もある。しかし、あまり計算時間が長くなると表示が遅くなり、かえって操作性の悪化を招くことになる。したがって、逐次演算してアニメーションとして表示するには、処理の高速化は避けられない課題である。

スムーズなアニメーションを実現するためには、単位時間内の表示コマ数を多くする必要があるが、実数演算や座標変換演算用の数値演算ハードウェア (コプロセッサ) を利用すると、乗除算などの演算そのものはソフトウェアによる実数演算と比べ非常に高速になる。しかし、数値演算コプロセッサと CPU と間のデー

タ変換に演算時間以外に時間がかかる。

これに比べ、SimKIT では固定小数点法の表現法を採用しているので、CPU が直接数値を処理するため、数値演算コプロセッサを利用するよりも高速な座標変換演算が可能となる。

4. データ表現

モデリングは、シミュレータで理論シミュレーションができるように、シミュレーション対象をモデルデータとしてコンピュータ内に取り込む処理である。ユーザによってそれらのデータの入力や保存・変更などのデータ管理ができるだけでなく、センサなどを用いた多様なロボットシステムへの対応のさせ方など、種々の条件についてユーザが容易に扱えるように考慮する必要がある。また、パーソナルコンピュータでは、シミュレーション処理の能率を考えると同時に、メモリ使用容量の節約にも気を付けなくてはならない。

SimKIT では、これらのことをふまえながら、シミュレーションに必要なデータを、処理の種類ごとに分類・抽出・整理し、一般的なデータ構造化手法を応用して、ロボットと周辺機器が同等に処理できるようなデータ構造とする。

これによって、処理の一貫性を持たせることができ、ユーザがモデリング操作を容易に行うことが可能になる。

データ構造としては、C言語で用意されている構造

体を基本単位として、配列、および連結リスト構造を採用する。

構造体は、ある決まった形式のデータのまとまりで、個々の物体形状や、座標変換ベクトルなど、一まとまりで意味をなすデータを表すのに用いる。

配列は、同じ形式のデータ・構造体を連続したメモリ領域に並べた線形リスト構造である。データ相互の関係はその並びによって表すのでデータの利用が容易であるが、データの追加は備えてあらかじめメモリ領域を多めに確保しなくてはならず、データの入らない無駄な領域も生じやすい。

一方、連結リスト構造は、データ・構造体を任意のメモリに置いて、その相互関係を示す線をつないで表すデータ構造である。データの接続情報のためにメモリ消費が増えるが、必要なだけのメモリ消費で済む。また、任意のデータの追加、削除、移動が容易で、階層構造も生成できる。したがって、このようなモデリングには、メモリを有効に利用できる連結リスト構造が、一般によく用いられる。

SimKIT では、ロボットシステムを構成する要素を、物体データ、プロセスデータ、プログラムデータという名称のモデリングデータとして3つに分類し、それぞれに適したデータ構造を設定した。

Fig. 4 に示す物体データは、ロボット、工具、周辺機器、作業対象物、あるいは周囲の環境などを、物体の、立体形状、位置、接続情報として、すべて相対的な座標変換ベクトルで積み上げる階層構造で表現する。

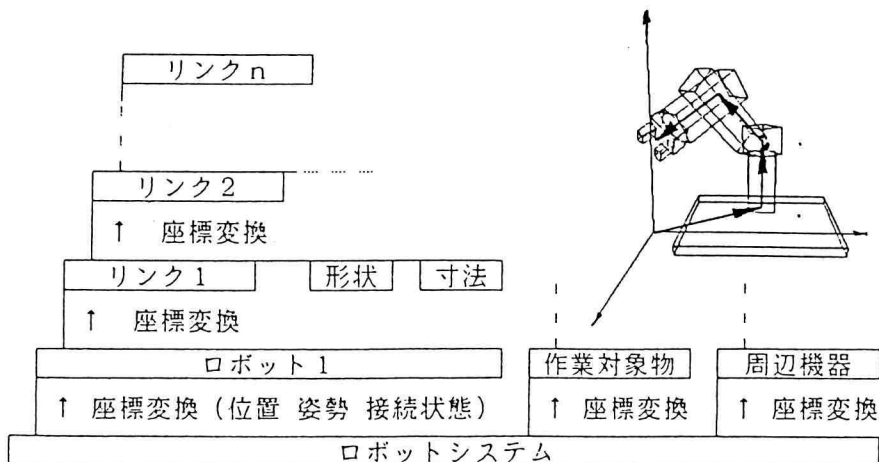


Fig. 4. Structure of the object data

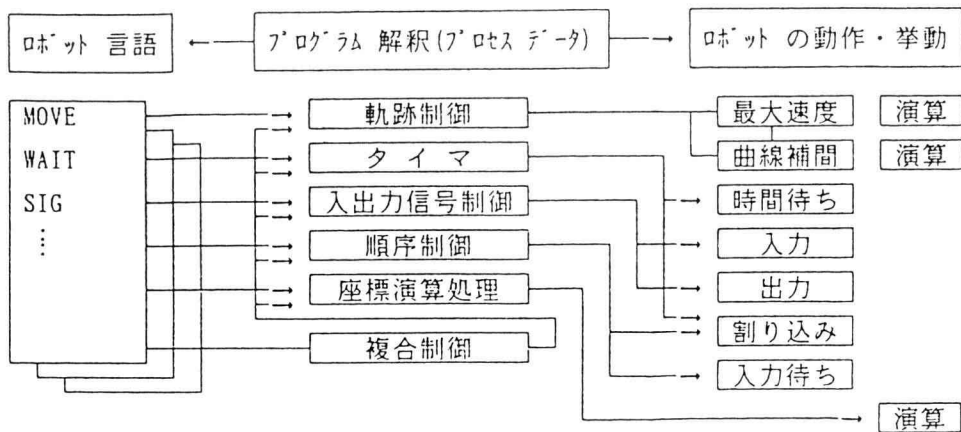


Fig. 5. Structure of the process data

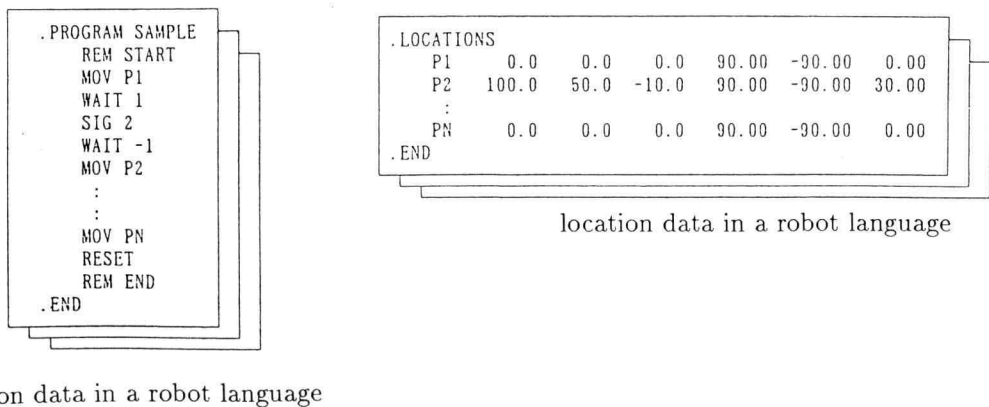
形状に関しては、よく用いられる、基本立体（プリミティブ）図形の組合せで表し、各物体に座標系を設定して、その位置関係は相対的な座標変換ベクトルで表す。すると、物体同士の衝突・干渉や、動作可能範囲の確認などを、一括して扱うことができる。可動軸（ジョイント）で接続される場合には、座標変換ベクトルの1パラメータを可変とする。質量、動的特性などの情報も同時に与えれば、動力学的なシミュレーションにも対応できる。

Fig. 5 に示すプロセスデータは、各機材の操作手順と、その操作による実際の機材の動作や挙動とを関係付けるデータで、データ構造は対応表形式とする。たとえば、ロボット言語は、各命令語と、ロボットの動

作制御方法や外部信号との対応を定義付けるものである。これを産業用ロボットのプロセスデータとして扱うこととする。各種ロボット言語への対応は、このプロセスデータの差し替えによって行うことができる。

また、周辺装置の制御は、ロボット、あるいは別の制御装置から電気信号などで制御することになるため、信号の操作手順と、周辺機器の挙動との対応を、プロセスデータとして扱う。この、プロセスデータをロボットシステム内のあらゆる変化に対応させることによって、ロボットシステム全体の働きを統括してシミュレーションすることができる。

プログラムデータは、Fig. 6 に示すように、ロボット



motion data in a robot language

location data in a robot language

Fig. 6. Robot language and program data

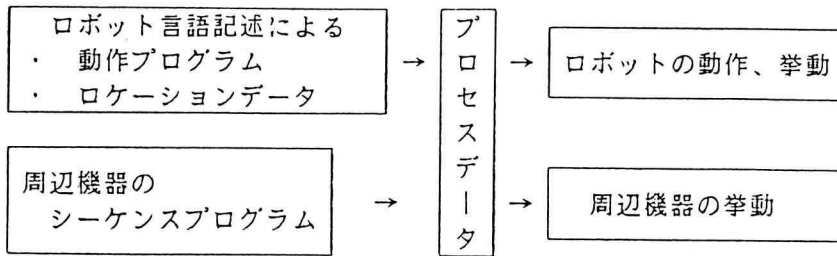


Fig. 7. The data flow in SimKIT

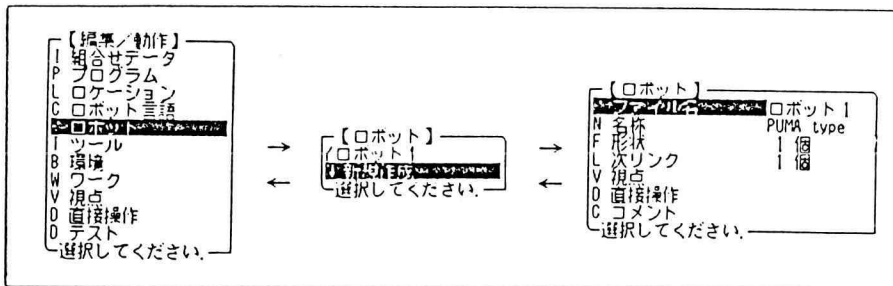
システムの行程を制御するために、各機材を操作する順序を示すデータである。ロボット言語を記述した動作プログラムや動作位置情報、およびシーケンスのシーケンスデータなどを、このプログラムデータとして扱う。

このようにして分類したデータの処理は、Fig. 7に示すように、プログラムデータをプロセスデータによって変換し、物体データ中の可動パラメータに与え、ロボットシステムの動きをシミュレーションする、という流れで表すことができる。

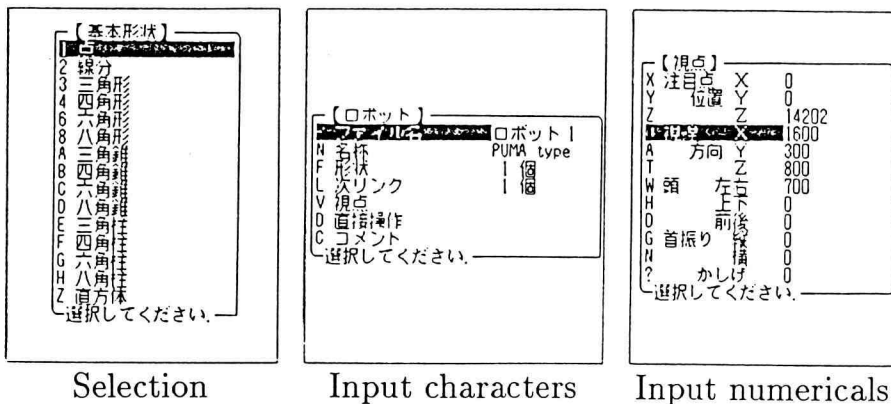
5. シミュレータの操作性

以上までに述べてきたように、高速な立体図形表示と、機能的なモデリング処理を実現したので、さらにシミュレータとしての実用性を高めるために、オペレータの操作方法の工夫を行い、操作性の向上を図る。

オペレータに対する操作性の良さは、操作の分かりやすさ、操作量の少なさ、操作方法の一貫性が大事であると考えられるので、SimKIT では、Fig. 8 に示すように、全機能を誘導的なメニュー形式で選択すること



hierarchical menu



Selection

Input characters

Input numerals

Fig. 8. Examples of the menus in SimKIT

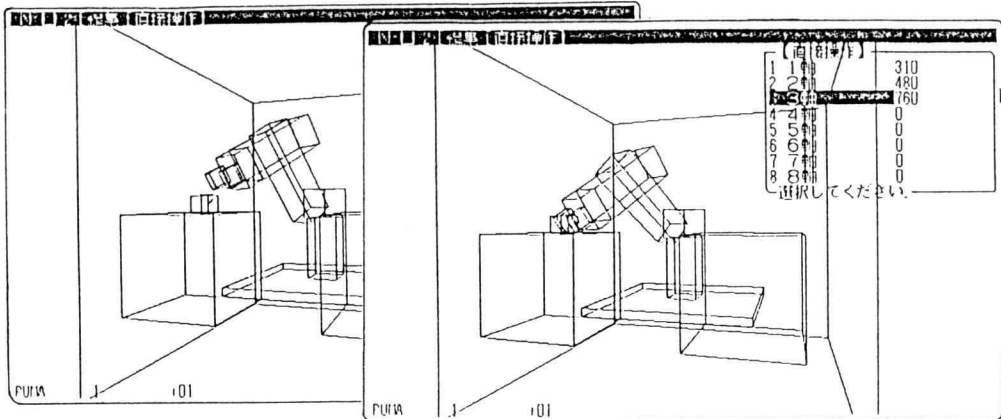


Fig. 9. Examples of the execution of SimKIT

により実行できる方式を採用する。

メニューには、機能選択項目とデータ入力項目とがあり、選択方法は、カーソル移動による方法か、その項目を示す記号を入力する方法のうち、操作量の少ない方法が選べる。データ入力には、複数の項目から一つ選ぶ選択データ、データの名称、動作プログラムなどを入力する文字データ、物体の形状寸法、ロボットの移動位置などを入力する数値データと分類して、それぞれの入力方法を統一した。

特に数値データは、直接数字を打ち込む方法の他に、カーソルキーなどでアナログ的に増減して、立体図形表示の変化を見ながら入力できるような方法も用意したが、表示が速やかに反応するので、操作がしやすくなっている。

Fig. 9はPUMA型のロボットについてSimKITでシミュレーションを行っている画面の例である。

6. おわりに

SimKITは、パーソナルコンピュータを利用するロボットシステムのシミュレータにもかかわらず、処理全体が高速で、操作性の良いものである。

SimKITでは、以下のようなソフトウェア上の工夫によって手法の採用によって、特別な装置を追加することなく、高速な立体図形表示を実現した。

- (1) 乗・除算に16bitsの固定小数点演算を使用する。
- (2) 主な三角関数をデータテーブル化する。
- (3) 透視投影法によるワイヤーフレーム表示を採用

する。

(4) グラフィックデータの生成とロボットの運動解析とを座標変換行列演算に統一する。

高速な立体図形表示により、データ入力にともない遅延なく立体図形の表示ができるようになった。また、パーソナルコンピュータにおいても立体視を用いたコマ数の多いスムーズなアニメーションが実現でき、作業の流れを見渡すことのできるシミュレーションが可能となった。

もちろん、あまり大きな生産システムを対象にするのは困難であるが、周辺機器の挙動なども表現できるため、1つの作業行程全体をシミュレーションすることが可能である。

特別な装置を必要とせず、携帯用のパーソナルコンピュータも利用できるのも、導入コストが安く、場所を選ばないのがSimKITの最大のメリットであろう。

本研究は、本学大学院学生矢茸直文君の協力によってなされた。記して感謝を表す。

参考文献

- 1) 『日本ロボット学会誌』 Vol. 7, no. 4
小特集—ロボットのオフラインプログラミング,
日本ロボット学会 (1989)
- 2) 『ロボット』 No. 73
特集—シミュレーション・オフラインプログラミング,
日本産業ロボット工業会 (1990)
- 3) 「ロボットと機構の運動シミュレータの基礎と応用」,
日本機械学会講習会 [No. 900-76] 資料

- (1990)
- 4) 田口 幹, 矢葺直文:「ロボット制御プログラム開発支援システムの研究」, 第7回日本ロボット学会 学術講演会予稿集, pp.267-268 (1989)
- 5) 田口 幹, 矢葺直文:「ロボット制御プログラム開発支援システムの研究」, 第1回日本ロボット学会 ロボットシンポジウム予稿集, pp.229-234 (1991).

付録1. ロボットプログラム開発支援システム「SimKIT」の仕様

ハードウェア構成

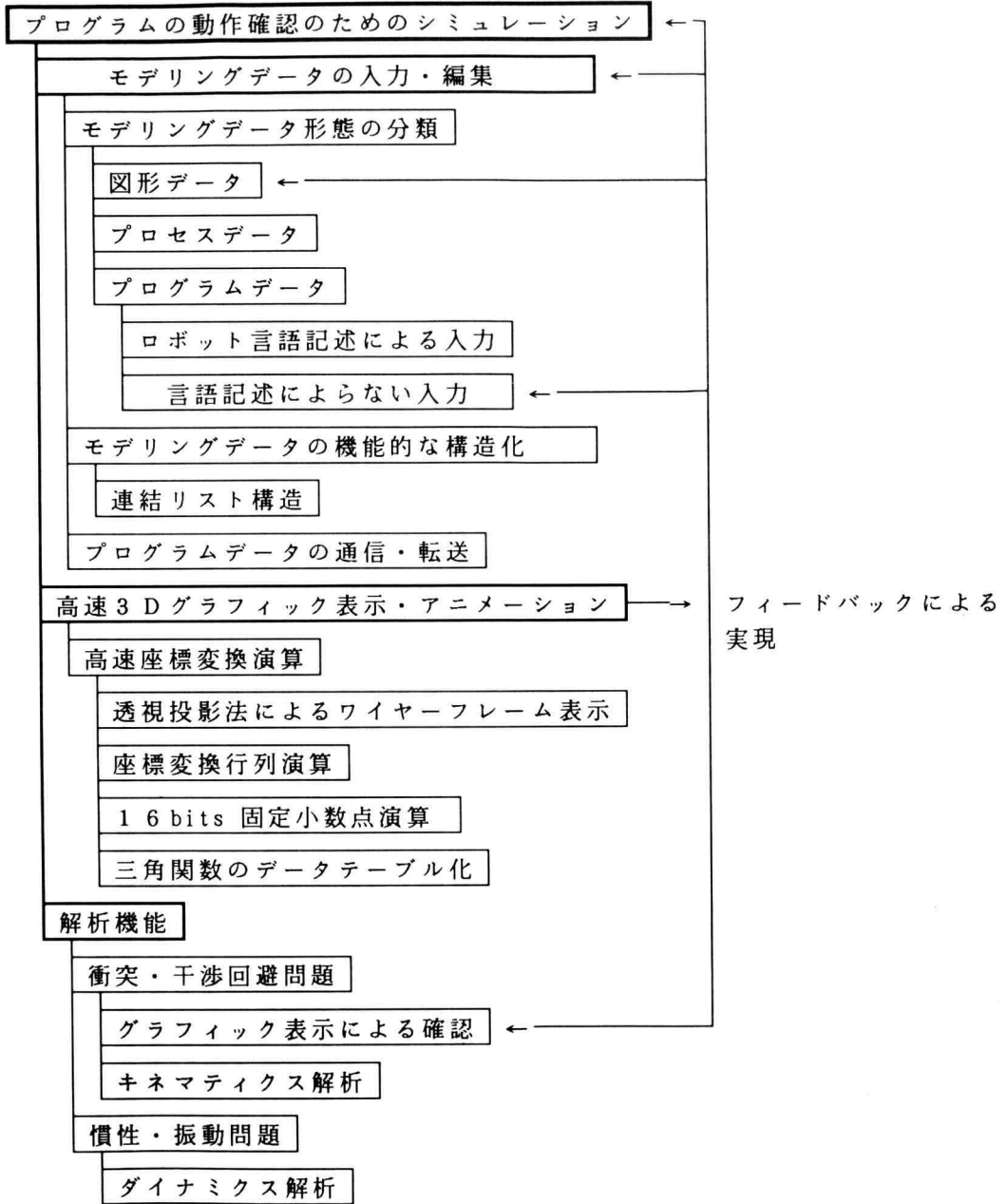
CPU	NEC PC-9800/EPSON PC-286 シリーズ
メモリ	384KB 以上
入力装置	キーボード, マウス
外部記憶装置	FDD

ロボットとのデータ交換シリアル, パラレル I/O, または FD

ソフトウェア構成

OS	MS-DOS Ver. 2.11 以上
記述言語	C 言語, 一部アセンブリ言語
開発部品	マンマシンインターフェースモジュール 座標変換演算モジュール グラフィック表示モジュール データベースモジュール

付録2 SimKIT の機能構成概念図



付録3 ロボットシステムと SimKIT の結合概念図

