



Title	ロボットシミュレータの開発
Author(s)	嘉数, 侑昇; Kakazu, Yukinori; 沖野, 教郎 他
Citation	北海道大學工学部研究報告, 124, 1-12
Issue Date	1985-01-31
Doc URL	https://hdl.handle.net/2115/41900
Type	departmental bulletin paper
File Information	124_1-12.pdf



ロボットシミュレータの開発

嘉数侑昇 沖野教郎 吉村 斎* 仲村人也

(昭和59年9月29日受理)

Development of a Robot Simulator by

Yukinori KAKAZU, Norio OKINO,
Hitoshi YOSHIMURA and Hitoya NAKAMURA

(Received September 29, 1984)

Abstract

In the present days, industrial robots are playing key roles in highly automated factories. Various types of robots have been developed and used for a variety of purposes. To incorporate these robots into highly organized production systems, e. g. FMS, an approach was introduced to solve some typical problems: how to avoid dangerous collisions, how to use robots effectively and how to develop new robots some special purposes are our main objectives. The approach made here is based on a virtual robot, i. e. robot simulator: based on Geometric Modeling.

This paper describes how the new type of robot simulator was developed and the fact that the simulator consists of the following parts.

- (1) Geometric Modeling part of the virtual robot world.
- (2) Robot mechanism modeling part.
- (3) Output part of synthetic solution of robot motion.
- (4) Robot Language part.
- (5) Transformation part of the robot world geometry.
- (6) Graphic output part.

In addition, with respect to Geometric Modeling, 3D Solid Modeling method is adopted. Therefore, this robot simulator can easily be connected to 3D CAD systems.

1. 緒 言

現在、ロボットにある作業を行わせる場合、実際のロボットの存在を前提とした直接教示によるプレイバック方式やプログラミング方式が主流になりつつある。これらの方式に於いては人間のロボット作業現場への直接介入(直接教示方式)や人間のロボット作業環境や作業内容の認識(プログラミング方式)を前提としているため、作業内容の変更、環境の変化等に対処

するのに柔軟さに欠け、又、危険性をも伴う場合がある。これらの問題を解決するものとして、最近仮想ロボットを用いたオフライン・プログラミング方式のロボットシミュレータへの期待がある。

ロボットシミュレータは仮想ロボット及び仮想作業空間を3次元幾何モデルとして計算機内に設定しそれらのハンドリングが容易に行えるように構成されているために、ロボット言語によるロボット動作、作業の指示が容易となり、またその結果をディスプレイによって視覚的に把握し仮想作業空間におけるロボット作業世界を認識することが可能となる。

ロボットシミュレータは、ロボット作業のシミュレーションまでを含めるとそのカバーする領域は非常に広がるが、それらの全てに共通した核となるものは、ロボットとその作業環境の形状モデリング法、ロボットの動作状態の指示法と決定法、およびディスプレイによる効果的なロボット作業世界の表示法である。その他の種々のシミュレーションは、これらの核に付加された各問題向きシミュレータにより実行されるものである。

ここでは、3次元ソリッドモデラ TIPS-1¹⁾ を用いて上記の核を中心として構築したロボットシミュレータシステムについて述べる。

2. ロボット作業世界のモデリング

本来、ロボットシミュレータの目的は、ロボットの幾何学的運動の解析の他に、ロボット自身や被作業物体の力学的、物理的諸特性の解析など広い範囲で考えるべきであるが、ここでは幾何学的側面についてのみ考えることにする。

ロボット作業空間に存在する物体には、大別して(1)静止して動かない対象、(2)運動する対象、(3)変形する対象、の3種類があり、(1)にはロボットの作業空間内に存在する環境としての静止物体、(2)にはロボット自身およびロボットの作業によって移動させられる被作業物体、(3)にはロボットの作業によりその形が変形させられる被作業物体がそれぞれ該当する。したがって、ロボット作業世界のモデリングでは(1)～(3)を考慮する必要がある。

ロボット作業空間の形状モデリングは、一般のCADシステムにおける形状モデリングと同様に、形状に特有な幾何学的情報を形状データベースとして計算機内に構築することによりこれを行う。このために、PLACE²⁾ではワイヤフレームモデルを、AUTOPASS³⁾では多面体モデルを採用しているが、本来これらは近似的モデルであり、厳密解が要求されるシミュレーション向きではないことが指摘される。そこで、ここでは著者等が開発した厳密モデルとして知られるCSG(Constructive Solid Geometry)モデルを用いることにする。よって、ここでは、ロボット作業世界のモデリングシステムを図1のように構成することを考える。

ロボット作業世界の初期状態は、あらかじめ入力しておく必要があり、それらはロボット自体の形状、および作業環境の形状モデルデータである。本シミュレータで用いる TIPS-1 システムの場合、これらは TIPS-1 形状記述言語¹⁾により定義されてプリプロセッサ T1PRPR¹⁾によ

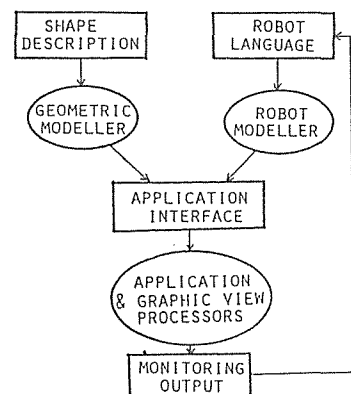


図1 基本構成

って標準形状フォーマットに変換されて、形状データベース TILIST³⁾ に格納されることになる。したがって、(1)の静止物体の場合、通常の CAD システムに於ける物体と同一に扱えるので特に問題が発生しないが、ただ一般の CAD システムの場合、その対象を部品一個に限っているのに対し、ここで扱う作業環境に於いては複数の部品を同時に扱う事になることに注意を要する。(2)および(3)に含まれる物体に関しては、与えられたロボット言語によるプログラムの実行によって運動や変形を生じ、時々刻々と変化していく。このようなケースに対しては CSG を構築するのは一般に困難であるが、ここではその問題をも解決している。具体的には、形状モデルを構成する各 CSG プリミティブに関し、その構成半空間群に変換マトリクスを掛けて変換後の CSG プリミティブの再構成を行う。変換マトリクスの各要素は、与えられたロボット言語を翻訳し、必要な 1 命令毎に抽出し、計算を実行することにより与えられる。すなわち、ロボット作業世界の形状モデルは 3 次元ソリッドモデルとして、時間関数に従って時々刻々変化するように構築されることになる。

3. システム構成

図 2 に構築したプロトタイプのロボットシミュレータのシステム構成を示す。

システムを構成する各部は以下に示す通りである。

① マニピュレータ機構データ入力部

コード化されているマニピュレータの関節の種類、リンクの長さを入力し、プリプロセッサを通して、ロボット機構モデル用ファイルに保存する。

② 幾何形状定義部

マニピュレータおよびロボット作業空間内の環境の形状情報に関するモデリング部である。前述したごとく、形状定義には TIPS-1 の形状モデリング法に従う。ただし、マニピュレータの形状定義には①で定義した機構モデルとの対応がとられる。

③ 操作言語システム部

操作言語を用いて環境の設定を行い、マニピュレータの動作命令を入力し、操作データを作成する。ここでは操作言語(いわゆる狭義のロボット言語)として VAL⁴⁾-like なロボット言語を開発している。解釈された操作データは、動作テーブルおよび位置姿勢テーブルと呼ばれるアレイに格納される。

④ 動作発生部

先に決められているマニピュレータの機構データと操作データを入力として、開発したマニピュレータ動作プロセッサにより、マニピュレータとその作業によって移動、変化する物体のある時刻(指定された)での状態データ(Present state data)を作成する部分である。

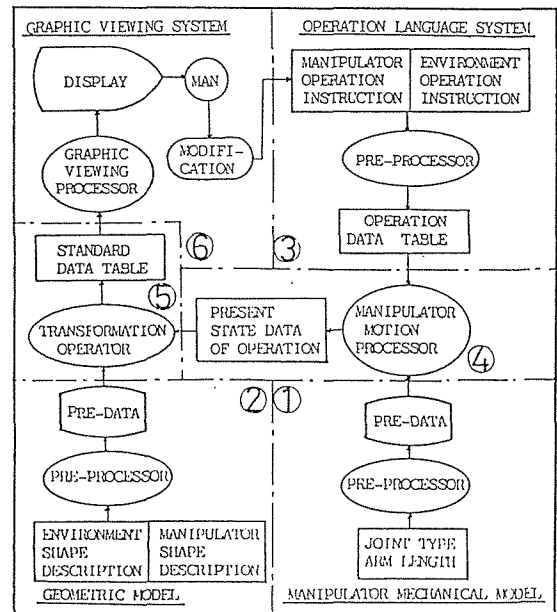


図 2 システム構成

⑤内部データ自動変換部

定義されている形状モデルおよび現時点での操作状態データを入力として、内部データ自動変換プロセッサ (Translation Operator) を通して、形状情報に関するインターフェイスとしての標準形状データテーブルを作成する。このテーブルは TIPS-1 では T1LIST と呼ばれる。T1LIST は CAD における種々のアプリケーション・ルーチンへのインターフェイスともなり得るので、ロボットシミュレータは、直接それら呼び込んで利用することができる。

⑥グラフィックス出力部

標準形状データテーブルをもとに、すでに開発されている各種グラフィックルーチン群やアプリケーション・ルーチン群を用いて、ロボット作業空間の状態表示を行う。これらの処理は対話的に実行されて、次の動作命令の実行や修正、変更を繰り返すことにより、ロボットの動作状態を評価する。

以上は換言すれば、①、③および④はマニピュレータの幾何学的動作の指示に関係する部分であり、②、⑤および⑥は幾何形状モデリングおよびその出力に関する部分となる。処理手順からみると、最初に前処理①および②を行った後、③から⑥までの操作を順次繰り返すことにより操作プログラムを作成し処理することになる。

4. ロボット機構数式モデル

一般に、マニピュレータの関節には回転関節、旋回関節、直動関節等があり、これらを組み合わせることによりマニピュレータを構成することができる。また、3次元空間内にある物体の位置を指示するのに3自由度、その姿勢を指示するのに3自由度が必要となるため、マニピュレータの手先 (エンドエフェクタ) が任意の位置で任意の姿勢をとるためには、マニピュレータには最低6自由度が要求されることになる。

本シミュレータにおいては、一例として図3に示すような回転及び旋回関節で構成された6関節を持つ6自由度の関節型マニピュレータを用いた。そのベクトル図は図4に示す通りである。

このマニピュレータに関する特性方程式は以下に示す通りである。

$$P = C_1^k + E^{k\theta_1} (C_2^k + E^{j\theta_2} (C_3^i + E^{j\theta_3} (C_4^i + E^{i\theta_4} (C_5^i + E^{j\theta_5} (C_6^i + E^{i\theta_6} (G_7^i)))))) \dots (1)$$

ただし

P: 位置ベクトル

C: 定数ベクトル

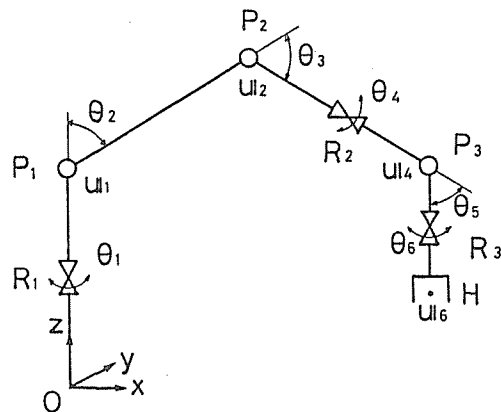


図3 マニピュレータの構成

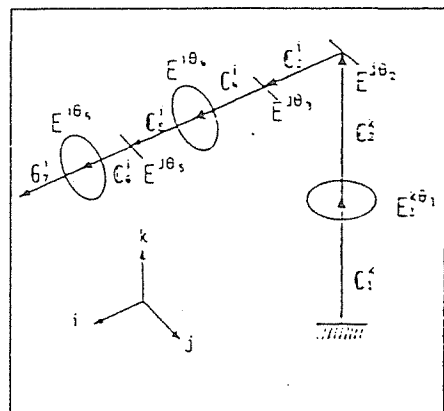


図4 ベクトル図

E: 回転テンソル

G: グリップベクトル

マニピュレータのエンドエフェクタにおける把握点の位置および姿勢が与えられると、それに対応する各関節の関節変数の値 (変位置) $\Theta = \{\theta_i\}, i = 1 \sim 6$ をシンセシス解として求めることができる。シンセシス解の求め方を以下に概略して述べる。図3に於いて以下の定義を行う。

$$\overline{OP_1} \equiv l_1, \overline{P_1P_2} \equiv l_2, \overline{P_2P_3} \equiv l_3, \overline{P_3H} \equiv l_4$$

一般に任意のリンク i とそれに続くリンク $i+1$ の関係を表わす同次変換マトリクスは A マトリクス⁵⁾ と呼ばれるが、回転/旋回関節に関して A マトリクスはそれぞれ式(2)および(3)で表わされる。

$$A_{i+1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & C_i & -S_i & 0 \\ 0 & S_i & C_i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \dots\dots\dots (2)$$

$$A_{i+1} = \begin{pmatrix} C_i & 0 & S_i & l_i C_i \\ 0 & 1 & 0 & 0 \\ -S_i & 0 & C_i & -l_i S_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \dots\dots\dots (3)$$

ただし式 (2), (3) に於いて, $C_i \equiv \cos(\theta_i), S_i \equiv \sin(\theta_i)$ とする。

絶対座標系に対するエンドエフェクタの位置姿勢は式 (4) のごとく各関節における A_i マトリクスの積で表わされ、一般にこれは T マトリクス⁴⁾ と表記される。

$$T = A_1 A_2 A_3 A_4 A_5 A_6$$
$$= \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \dots\dots\dots (4)$$

ここで n, o, a, p は各々、エンドエフェクタにおける絶対座標系に対する垂直ベクトル、把持ベクトル、接近ベクトルおよび位置ベクトルを示すものとする。これらの関係式を使いながら以下各関節変数値を求める。

まず、図3における U_1, U_6 は既知ゆえ、 $U_4 = (x_4, y_4, z_4, 1)^T$ は式 (5) によって求める。

$$U_4 = U_6 + T_6 (-l_4 \ 0 \ 0 \ 0)^T \dots\dots\dots (5)$$

また、点 O, P_1, P_2, P_3 は同一平面上にあることから

$$\theta_1 = \tan^{-1}(y_4/x_4) \dots\dots\dots (6)$$

$$\theta_2 = (\tan^{-1}(z_1 - z_4) / ((x_4 - x_1)^2 + (y_4 - y_1)^2)^{1/2}) \pm B \dots\dots\dots (7)$$

ここで

$$B = \cos^{-1} (|U_1 - U_4|^2 + l_2^2 - l_3^2) / 2l_2 \cdot |U_1 - U_4| .$$

θ_1, θ_2 が求まったので A_1, A_2 マトリクスは既知となる。次に関係式, 式 (8) を用いてその回転部分の成分を比較することにより θ_3 を求めることができる。

$$A_3[1\ 0\ 0\ 0]^T = (A_1 A_2)^{-1} (1/\ell_3) [x_4 - x_2\ y_4 - y_2\ z_4 - z_2\ 0]^T \quad \dots\dots\dots (8)$$

$$\begin{pmatrix} C_3 \\ O \\ -S_3 \end{pmatrix} = \begin{pmatrix} C_1 C_2 & S_1 C_2 & -S_2 \\ -S_1 & C_1 & O \\ C_1 S_2 & S_1 S_2 & C_2 \end{pmatrix} \frac{1}{\ell_3} \begin{pmatrix} x_4 - x_2 \\ y_4 - y_2 \\ z_4 - z_2 \end{pmatrix} \equiv \begin{pmatrix} e_{13} \\ e_{23} \\ e_{33} \end{pmatrix} \quad \dots\dots\dots (9)$$

$$\therefore \theta_3 = \tan^{-1}(-e_{33}/e_{13}) \quad \dots\dots\dots (10)$$

同様に関係式, 式 (11) に A_4, A_5 を適用して θ_4, θ_5 を求めることができる。

$$A_i A_{i+1} [1\ 0\ 0\ 0]^T = (A_1 A_2 \dots\dots\dots A_{i-1})^{-1} \frac{U_{i+1} - U_i}{|U_{i+1} - U_i|} \quad \dots\dots\dots (11)$$

$$A_4 A_5 [1\ 0\ 0\ 0]^T = (A_1 A_2 A_3)^{-1} [n_x\ n_y\ n_z]^T \quad \dots\dots\dots (12)$$

$$\begin{pmatrix} C_5 \\ S_4\ S_5 \\ -C_4 S_5 \end{pmatrix} = \begin{pmatrix} C_1 \cdot C_{23} \cdot C_2 \cdot C_\beta + S_1 \cdot C_{23} \cdot S_\alpha \cdot C_\beta + S_{23} \cdot S_\beta \\ -S_1 \cdot C_\alpha \cdot C_\beta + C_1 \cdot S_2 \cdot C_\beta \\ C_1 \cdot S_{23} \cdot S_\alpha \cdot C_\beta + S_1 \cdot S_{23} \cdot S_\alpha \cdot C_\beta - C_{23} \cdot C_\beta \end{pmatrix} \equiv \begin{pmatrix} e_{15} \\ e_{25} \\ e_{35} \end{pmatrix} \quad \dots\dots (13)$$

$$\theta_4 = \tan^{-1}(e_{25}/-e_{35}) \quad \dots\dots\dots (14)$$

$$\theta_5 = \tan^{-1}(-e_{25}/e_{13} \cdot C_4) \quad \dots\dots\dots (15)$$

式 (13) に於いて $C_{23} = \cos(\theta_2 + \theta_3)$, $S_{23} = \sin(\theta_2 + \theta_3)$, $Cq = \cos q$, $Sq = \sin q$ (ただし $q = \{\alpha, \beta, \gamma\}$: α, β, γ はエンドエフェクタのオイラー角) とする。

最後に式 (16) を用いて前と同様に式 (18) の如く θ_6 を定めることができる。

$$A_6[0\ 0\ 0\ 1]^T = (A_1 A_2 A_3 A_4 A_5)^{-1} [a_x\ a_y\ a_z\ 0]^T \quad \dots\dots\dots (16)$$

$$\begin{pmatrix} O \\ -S_6 \\ C_6 \end{pmatrix} = \begin{pmatrix} C_1 \cdot C_{23} \cdot C_5 - S_1 \cdot S_4 \cdot S_5 - C_1 \cdot S_{23} \cdot C_4 \cdot S_5 & S_1 \cdot C_{23} \cdot C_5 + C_1 \cdot S_4 \cdot S_5 - S_1 \cdot S_{23} \cdot C_4 \cdot S_5 \\ -S_1 \cdot C_4 + C_1 \cdot S_{23} \cdot S_4 & C_1 \cdot C_4 + S_1 \cdot S_{23} \cdot S_4 \\ C_1 \cdot C_{23} \cdot S_5 + S_1 \cdot S_4 \cdot C_5 + C_1 \cdot S_{23} \cdot C_4 \cdot C_5 & S_1 \cdot C_{23} \cdot S_5 - C_1 \cdot S_4 \cdot C_5 + S_1 \cdot S_{23} \cdot C_4 \cdot S_5 \\ C_4 \cdot S_5 & -S_{23} \cdot C_5 - C_{23} \cdot C_4 \cdot C_5 \\ C_{23} \cdot S_4 & \\ C_4 \cdot C_5 & -S_{23} \cdot S_5 + C_{23} \cdot C_4 \cdot C_5 \end{pmatrix} \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \equiv \begin{pmatrix} e_{16} \\ e_{26} \\ e_{36} \end{pmatrix} \quad \dots\dots\dots (17)$$

$$\theta_6 = \tan^{-1}(-e_{26}/e_{36}) \quad \dots\dots\dots (18)$$

以上により必要な全ての関節変数 $\theta = \{|\theta_i|\}, i=1 \sim 6$ が求まることになる。

なお, 他の関節タイプや自由度が6以上の場合についても一部シンセシス解, その他は擬似逆行列を用いてこれを求めることができる。

5. ロボット言語システム

ロボットシミュレータによる作業シミュレーションを実行するために様々なレベルの言語が開発されているが, 言語レベルとしては高度な仕事記述言語や作業記述言語も, 基本的には単純な動作記述言語の組み合わせで実現可能である。作業記述言語としては RAPT⁶⁾ や AUTO-

PASS等が知られているが、一般にこの種の言語はまだ問題向きレベルで研究開発段階であり、汎用的なものはまだ開発されていない。本研究に於いては、マニピュレータの動きの基本となる動作機能を確保するために VAL-like な動作レベルの言語を開発した。また、幾何モデルを取扱う必要上、VAL には存在しないような FEED (部品供給)、HAND (物体を把握したまま移動を可能にする)、RELEASE (HAND 命令の解除) 等を新しく付加してある。また、エディタコマンドは、プログラムステートメントの挿入、変更、削除、終了 (ファイル保存) を可能にする。表 1 に本システムで開発したコマンド一覧を示す。

INSTRUCTION	OPERAND & EXPLANATION OF COMMAND		
INSERT	(LINE NO.) Insert a line.	APPROS	(LOC.) (DISTANCE) Move from the point specified by the variable (LOC.) to the point offset by the given distance (DISTANCE) along a straight-line path.
CHANGE	(LINE NO.) Change a line.	MOVE	(LOC.) Move to the location and orientation specified by the variable (LOC.). Intermediate set points between the initial and final robot locations are computed by interpolating between the initial and final joint-variable values.
DELETE	(LINE NO.) Delete a line.	MOVES	(LOC.) Move to the location and orientation specified by the variable along a straight-line path.
SAVE	(Y OR N) Save a program.	DRAW	(DX) (DY) (DZ) Move along a straight line a distance (DX) in the X direction, (DY) in the Y direction and (DZ) in the Z direction. The hand's orientation is maintained during this motion.
STOP	(Y OR N) Stop EDIT mode.	DRIVE	(JOINT NO.) (CHANGE) Operate the single specified joint (JOINT NO.), changing its joint variable by (CHANGE) units.
TYPEA	(LINE NO.) (LINE NO.) Display the motion table.	OPENI	(DISTANCE) Open the hand to (DISTANCE).
TYPEB	(LINE NO.) (LINE NO.) Display the location and orientation table.	CLOSEI	(DISTANCE) Close the hand to (DISTANCE).
TYPEC	(LINE NO.) (LINE NO.) Display a source program.	READY	(Y OR N) Move the robot to the READY location about the work space.
TYPED	(Y OR N) Display a part name.	DEPARTS	(DISTANCE) Move the tool to given distance along the current Z axis of the hand.
POINT	(LOC.) (X) (Y) (Z) (ALPHA) (BETA) (GAMMA) Set the value of the variable (LOC.) equal to the location (X), (Y), (Z) and the orientation (ALPHA), (BETA), (GAMMA).		
HERE	(LOC.) Set the value of the variable (LOC.) equal to the current manipulator location and orientation.		
WHERE	(Y OR N) Display the current robot location.		
ABOVE	(Y OR N) Request a change in robot configuration so that the elbow of the robot is pointed up.		
BELOW	(Y OR N) Request a change in robot configuration so that the elbow of the robot is pointed down.		
FEED	(PART NAME) Feed a part.		
RELEAS	(Y OR N) Release the HAND instruction.		
REMARK	(CHARACTERS) Provide a "comment" line in a program.		
END	(Y OR N) Specify the end of a program.		

表 1 コマンドとその説明

実例を図 5 に示すように、ロボット言語によるプログラミングは、プログラム編集機能を持つエディタコマンドを使用して、必要とされるコマンドを対話的に逐次入力することにより実行される。先に述べたごとく、対話的に入力されたプログラムから、自動的に内部で動作テーブル、位置姿勢テーブルが作成される (図 6)。動作テーブルには動作命令がコード化されてストアされ、動作命令に続くオペランドも同様にストアされる。同様に、位置姿勢テーブルには定義された位置姿勢情報がコード化されてストアされている。

ロボットモデラは、まず最初にこれらのテーブルを用いて必要なマニピュレータの各関節変位量を第 4 章にて展開された各式に従って決定していく。すなわち、まず動作テーブルの命令を一個づつ読み込み、それを順次実行していく。この時に動作命令によって与えられたエン

ドエフェクタ座標情報から、シセンシス解としての各関節の変位量を求めることになる。

ロボットモデラは次に求めた各関節変位量に対応して、実際に変化するマニピュレータの各パートや、ロボット作業によって移動する。あるいは変化する被作業物体に対して、先に定義されているそれらの幾何情報を逐次変更していく。これらの作業は標準形状データテーブルである T1 LIST 上にて常に行われる。

6. ロボット作業空間の幾何モデル

ロボットシミュレータは仮想ロボット作業空間を扱うのでその核となる幾何形状データは必須である。幾何形状データの作成には前述のごとく3次元ソリッドモデラ TIPS-1 を用い

```

-----
| MOVE      A      MOVES   B      DRAW    C      APPROX  D      |
| DEPARTS  E      DRIVE   F      OPENI   G      CLOSEI  H      |
| HAND     I      RELEASE J      READY   K      POINT  L      |
| HERE     M      FEED    N      ABOVE   T      BELOW  U      |
| REMARK   U      END     Z      WHERE   Y      TYPER  O      |
| TYPEB   P      TYPEC   Q      TYPED   R      INSERT 1      |
| CHANGE  Z      DELETE  3      SAVE    4      STOP   4      |
-----
**PLEASE KEY IN COMMAND(A1)?
L
-----
| 10 POINT < LOC. > < X,Y,Z > < D,A,T > |
|-----|
| LOC. |.....X.....Y.....Z.....|
READY 670. 0. 950.
|-----|
|.....O.....A.....T.....|
| 0. 0. 0. |
**PLEASE KEY IN COMMAND(A1)?
L
-----
| 20 POINT < LOC. > < X,Y,Z > < D,A,T > |
|-----|
| LOC. |.....X.....Y.....Z.....|
LOC1 700. 0. 280.
|-----|
|.....O.....A.....T.....|
| 0. 90. 0. |
**PLEASE KEY IN COMMAND(A1)?
L
-----
| 30 POINT < LOC. > < X,Y,Z > < D,A,T > |
|-----|
| LOC. |.....X.....Y.....Z.....|
LOC2 450. 450. 370.
|-----|
|.....O.....A.....T.....|
| 30. 40. 50. |
**PLEASE KEY IN COMMAND(A1)?
L
-----
| 40 POINT < LOC. > < X,Y,Z > < D,A,T > |
|-----|
| LOC. |.....X.....Y.....Z.....|
LOC3 630. 200. 760.
|-----|
|.....O.....A.....T.....|
| -30. -10. -10. |
**PLEASE KEY IN COMMAND(A1)?
X
-----
| 50 READY |
|-----|
LN= 1
OK ? YES(0) NO(1)
0
**PLEASE KEY IN COMMAND(A1)?
A
-----
| 60 MOVE < LOCATION NAME (AB) > |
|-----|
LOC1
-----
B
-----
| 70 MOVES < LOCATION NAME (AB) > |
|-----|
LOC2
|-----|
**PLEASE KEY IN COMMAND(A1)?
D
-----
| 80 APPROX < LOC. > < DISTANCE > |
|-----|
LOC3 30.
|-----|
**PLEASE KEY IN COMMAND(A1)?
C
-----
| 90 DRAW < X > < Y > < Z > |
|-----|
| -30. -40. -50. |
|-----|
**PLEASE KEY IN COMMAND(A1)?
F
-----
| 100 DRIVE < JT > < CHANGE > |
|-----|
| 2 30. |
|-----|
**PLEASE KEY IN COMMAND(A1)?
G
-----
| 110 OPENI < DISTANCE > |
|-----|
| 50. |
|-----|
**PLEASE KEY IN COMMAND(A1)?
H
-----
| 120 CLOSEI < DISTANCE > |
|-----|
| 10. |
|-----|
**PLEASE KEY IN COMMAND(A1)?
Z
-----
| 130 END |
|-----|
OK ? YES(0) NO(1)
0

```

図5 コマンド入力例

** MOTION TABLE (TM) **							
1	2	3	4	5	6		
MC	AC	HC	LC	HO, JN DX, DI	DY	DZ	H, D
0	1	0	1	9999.996	9999.996	9999.996	9999.996
1	1	0	2	9999.996	9999.996	9999.996	9999.996
3	1	0	3	9999.996	9999.996	9999.996	9999.996
6	1	0	4	30.000	9999.996	9999.996	9999.996
4	1	0	99	-30.000	-40.000	-50.000	9999.996
8	1	0	99	2.000	30.000	9999.996	9999.996
19	1	0	99	9999.996	9999.996	9999.996	50.000
20	1	0	99	9999.996	9999.996	9999.996	10.000

LOCATION AND ORIENTATION TABLE (TL)										
1	2	3	4	5	6	7	8	9	10	
LN	ISF	RNM	PC	RNO	X	Y	Z	O	A	T
READY	0	99	1	99	670.000	0.0	950.000	0.0	0.0	0.0
LOC1	0	99	1	99	700.000	0.0	280.000	0.0	90.000	0.0
LOC2	0	99	1	99	450.000	450.000	370.000	30.000	40.000	50.000
LOC3	0	99	1	99	630.000	200.000	760.000	-30.000	-10.000	-10.000

図6 動作テーブル及び位置姿勢テーブル

る。説明の詳細は省くが、TIPS-1ではいわゆる CSG としての基本形状要素；セグメントとしてのプリミティブと呼ばれる3次元形状要素があり、これらの和，積，差集合操作により3次元形状をモデリングする。

今、図7のようにロボットおよびロボット作業空間の環境が仮想的に設定された時、これらを TIPS-1 のプリミティブ群を用いて図8のごとく形状記述することができる。すなわち、これはシステムへの初期幾

何形状入力データとなる。

入力された初期幾何形状データは、翻訳されて標準幾何形状データテーブル T1LIST に格納される (図9)。この時点ではロボット・モデラからの影響はまだ及んでいない。

次にロボットモデラがロボット言語から作成された動作テーブル、位置姿勢テーブルを参照することによって実際にロボットに動作を与えていく、すなわち、T1LISTの内容は次々とロボットモデラからの指令により変換マトリクスを掛けられてその内容を変更されていく。(図10)

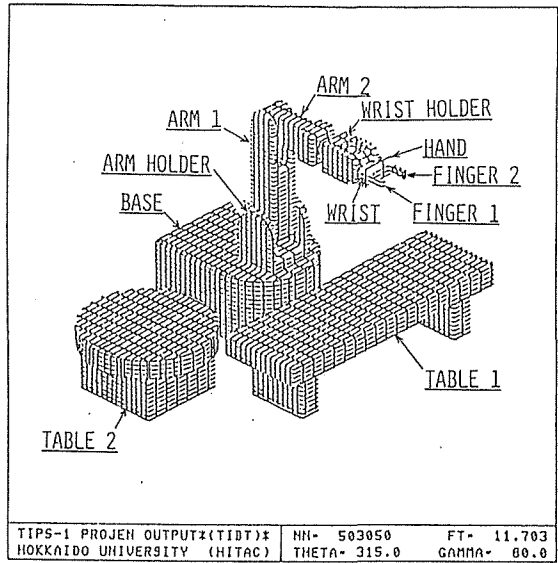


図7 ロボットおよびロボット作業空間

7. 実験

以上、展開してきた方法に従って、北大大型計算機センター、HITAC M280H上でFORTRAN言語によりシステムを構築した。構築したシステムを使って、2、3の実験を行ったので以下に示す。

●前処理データの入力

予め、マニピュレータの機構を定義しておく必要がある。機構モデルの指定の方法は文献⁷⁾に詳述してあるが、ここではその機構を図3に示したものと同一とし、ロボットの腕の長さを次の通りとした。すなわち

- $l_1 = 450$ (mm)
- $l_2 = 500$ (mm)
- $l_3 = 450$ (mm)
- $l_4 = 170$ (mm)

また、必要なロボット作業空間の幾何形状定義文は図8に示してある。

●シミュレーションの実行

モニタ・コマンドやロボットへ

```

00010 #F -450, 1000, -1050, 800, 0, 1500,
00020 #P B ( ) -450, 100, -250, 250, 0, 201;
00030 #1/P, CUBE, /1, 1-450, -80, -250, 250, 0, 201;
00040 #2/P, CIRCLE, /1, 250(-80, 0), 150, 190, -250, 250, 0, 201;
00050 #E
00060 #E
00070 #P J ( ) -101, 101, -120, 120, 0, 201;
00080 #1/P, CUBE, /1, 1-100, 100, -120, 120, 0, 201;
00090 #2/P, CUBE, /1, 1-100, 100, -120, -80, 0, 150;
00100 #3/P, CUBE, /1, 1-100, 100, 80, 120, 0, 150;
00110 #4/P, CIRCLE, /1, 100(0, 150)-101, 101, -120, -80, 143, 201;
00120 #5/P, CIRCLE, /1, 100(0, 150)-101, 101, 101, 120, 143, 201;
00130 #E
00140 #E
00150 #P AR ( ) -71, 571, -30, 30, -71, 71;
00160 #1/P, CIRCLE, /1, 70(0, 0)-71, 71, -30, 30, -71, 71;
00170 #2/P, CUBE, /1, 10, 200, -70, 70, -70, 70;
00180 #3/P, CUBE, /1, 123, 500, -70, -40, -70, 70;
00190 #4/P, CUBE, /1, 123, 500, 40, 70, -70, 70;
00200 #5/P, CIRCLE, /1, 70(150, 0)143, 221, -70, -40, -71, 71;
00210 #6/P, CIRCLE, /1, 70(150, 0)143, 221, 40, 70, -71, 71;
00220 #E
00230 #E
00240 #P BF ( ) -71, 571, -70, 70, -71, 71;
00250 #1/P, CIRCLE, /1, 50(0, 0)-70, 70, -39, 39, -70, 70;
00260 #2/P, CUBE, /1, 10, 211, -39, 39, -70, 70;
00270 #3/P, CUBE, /1, 1210, 200, -70, 70, -70, 70;
00280 #4/P, CIRCLE, /1, 40(0, 0)123, 551, -41, 41, -41, 41;
00290 #E
00300 #E
00310 #P CP ( ) 0, 221, -70, 70, -71, 71;
00320 #1/P, CUBE, /1, 10, 70, -70, 70, -70, 70;
00330 #2/P, CUBE, /1, 163, 150, -70, -40, -70, 70;
00340 #3/P, CUBE, /1, 163, 150, 40, 70, -70, 70;
00350 #4/P, CIRCLE, /1, 70(150, 0)143, 221, -70, -40, -71, 71;
00360 #5/P, CIRCLE, /1, 70(150, 0)143, 221, 40, 70, -71, 71;
00370 #E
00380 #E
00390 #P DF ( ) -51, 102, -50, 50, -51, 51;
00400 #1/P, CIRCLE, /1, 50(0, 0)-51, 51, -50, 50, -51, 51;
00410 #2/P, CIRCLE, /1, 50(0, 0)0, 102, -51, 51, -51, 51;
00420 #E
00430 #E
00440 #P G1 ( ) 0, 110, -80, 80, -50, 50;
00450 #1/P, CUBE, /1, 10, 20, -80, 80, -50, 50;
00460 #E
00470 #E
00480 #P G2 ( ) 0, 110, 0, 20, -50, 50;
00490 #2/P, CUBE, /1, 123, 110, -20, 0, -20, 20;
00500 #E
00510 #E
00520 #P G3 ( ) 0, 110, 0, 20, -50, 50;
00530 #3/P, CUBE, /1, 123, 110, 0, 20, -20, 20;
00540 #E
00550 #E
00560 #P TR1 ( ) -300, 200, -1050, -450, 0, 200;
00570 #1/P, CUBE, /1, 1-200, 200, -360, -560, 0, 210;
00580 #2/P, CIRCLE, /1, 200(0, -260, 1)-200, 200, -1050, -450, 200, 200;
00590 #E
00600 #E
00610 #P TR2 ( ) 200, 200, -600, 600, 0, 200;
00620 #1/P, CUBE, /1, 1200, 200, -600, 600, 0, 200;
00630 #2/P, CUBE, /1, 1250, 620, -500, -400, 0, 210;
00640 #3/P, CUBE, /1, 1350, 620, 400, 500, 0, 210;
00650 #E
00660 #E

```

図8 形状の記述例

力図、図13はいわゆる積層図と呼ばれるものであり、図14はラスターキャンニングによる濃淡面画出力である。なお、ロボットの動きそのものを実時間的にみるための図15に示すようなスケルトン図出力用のルーチンも用意されていて、マニピュレータの動きのシーケンスを見ることができるので種々の応用に結びつけられよう。

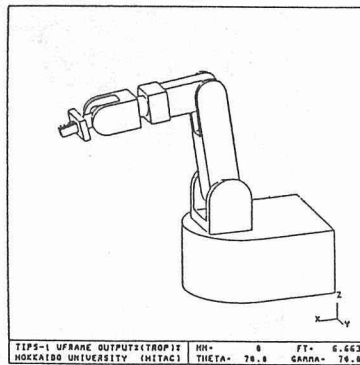
なお、図16はグラフィックス出力とは関係ないがTIPS-1の持つ他のアプリケーションルーチンである干涉認識用ルーチンを通して見た、マニピュレータの作業台との干涉の認識結果の出力図である。

8. 結言

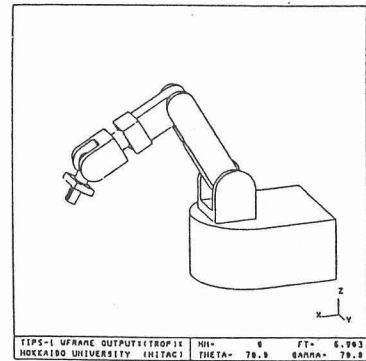
以上をまとめると次のように結言できる。

- 3Dソリッドモデルとしてのロボット作業空間の幾何形状モデルを構築した。

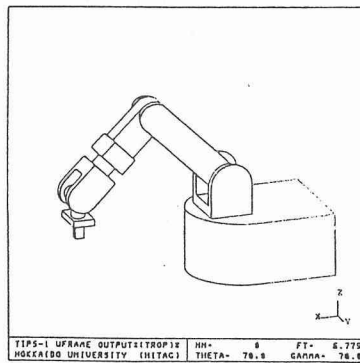
- ロボット機構モデルのシンセシス解を示した。
- ロボット言語システムを開発した。
- ロボットシミュレータシステムと3次元CADシステムの直結を幾何形状データテーブルをインターフェイスとすることにより実現した。



SCENE NO.1



SCENE NO.2



SCENE NO.3

図12 ワイヤフレーム出力

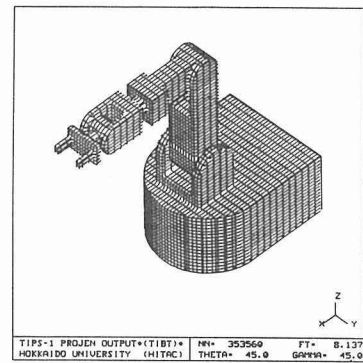


図13 積層図出力

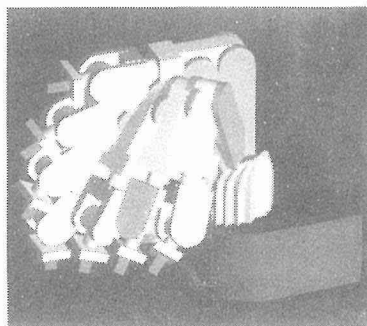


図14 濃淡面画出力

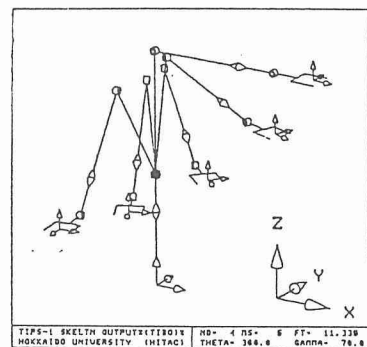


図15 スケルトン出力

- 各種グラフィックス出力による仮想ロボット作業空間の可視化を可能にした。

なお、ここで構築したロボットシミュレータは、ここで述べた以外のシミュレーションを実行する際の核になり得るものであることを改めて強調したい。

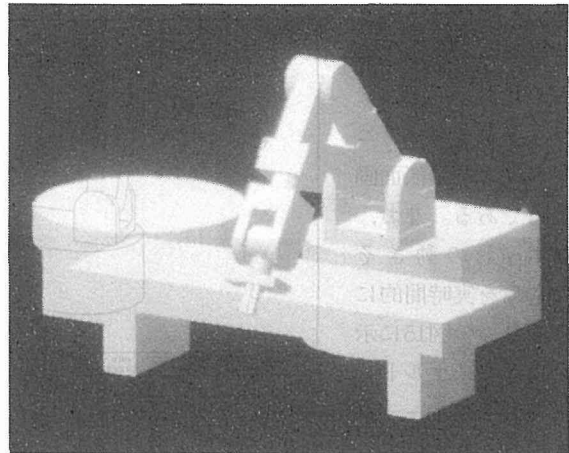


図16 干渉チェック出力例

参 考 文 献

- 1) N. OKINO, Y.KAKAZU and H. KUBO; TIPS-1; Technical Information Processing System for design, drawing and manufacturing; Proceedings of '73 PROLAMAT, Budapest Hungary (1973)
- 2) McAuto, PLACE; The Virtual Robot Simulator, McAuto Inc (1983)
- 3) L. I. Lieberman, M. A. Westey; AUTOPASS; An Automatic Programming System for Computer Controlled Mechanical Assembly, IBM J. RES. Develop (1977)
- 4) Unimation; The User's Guide to VAL, Unimation Inc (June 1980)
- 5) R. P. Paul; Robot Manipulators; Mathematics, Programming and Control, The MIT Press (1981)
- 6) R. J. Popplestone, A. P. Ambler and I. B. Bollos; An Interpreter for a language for Describing Assemblies, Artificial Intelligence vol. 14 pp. 79~107 (1980)
- 7) 嘉数, 沖野, 吉村, 仲村, 冗長型マニピュレータのシンセシス解——ロボット・シミュレータのための基礎研究——
北海道大学工学部研究報告 第120号 PP.75~85, (昭和59年3月)