

JAERI-M

86-123

マニプレータの逆運動学

—逆ヤコビアンの方法と特異点での処理—

1986年8月

鈴木 知明

日本原子力研究所
Japan Atomic Energy Research Institute

JAERI-M レポートは、日本原子力研究所が不定期に公刊している研究報告書です。
入手の問合せは、日本原子力研究所技術情報部情報資料課（〒319-11茨城県那珂郡東海村）あて、お申しこしください。なお、このほかに財団法人原子力弘済会資料センター（〒319-11 茨城県那珂郡東海村日本原子力研究所内）で複写による実費領布をおこなっております。

JAERI-M reports are issued irregularly.

Inquiries about availability of the reports should be addressed to Information Division Department of Technical Information, Japan Atomic Energy Research Institute, Tokaimura, Naka-gun, Ibaraki-ken 319-11, Japan.

©Japan Atomic Energy Research Institute, 1986

編集兼発行　　日本原子力研究所
印　　刷　　いばらき印刷株

マニプレータの逆運動学
(逆ヤコビアンの方法と特異点での処理)

日本原子力研究所東海研究所原子炉工学部
鈴 玉 知 明

(1986年7月28日受理)

マニプレータの逆運動学の解法として、その運動学方程式の線形化に基づいた逆ヤコビアンの方法が、一般的によく用いられている。本報告では、6自由度のロボット・マニプレータを指定した位置及び方向について解析し、逆ヤコビアンの方法の精度及び限界について論じる。

ヤコビアンが特異な場合、一般的には、逆ヤコビアンの方法は使えない。しかし、ヤコビアンの中で正則な小行列を見つけ出し、その逆変換を求めることが可能である。この概念に基づいて、特異点におけるマニプレータの逆変換の方法を開発した。実用性の見地から、今回得られた方法は、特異点において特異点以外で得られた計算精度とほぼ同程度の精度の逆運動学解を与えることが判明した。

On the Inverse Kinematics of a Manipulator
(The Inverse Jacobian Method and Treatment at Singular Points)

Tomoaki SUZUDO

Department of Reactor Engineering,
Tokai Research Establishment,
Japan Atomic Energy Research Institute
Tokai-mura, Naka-gun, Ibaraki-ken

(Received July 28, 1986)

The inverse Jacobian method based on the linearization of manipulator kinematics equation is generally used for obtaining its inverse kinematic solutions. In the present study, a robot manipulator having 6 degrees of freedom is analyzed for specified positions and orientations and the accuracies and limitations of the method are discussed.

Generally speaking, the inverse Jacobian method can not be applied when it is singular. However, it is possible to find a regular minor matrix and to obtain its inverse transformation. Based on this concept, a method has been developed which can be applied for obtaining the inverse transformation for manipulators at their singular points. It has been confirmed that the present method can give inverse kinematic solutions for manipulators with the same order of accuracy at singular points as those obtained at nonsingular points from a practical view point.

Keywords : Inverse Jacobian, Inverse Transformation, Jacobian,
Linearization, Manipulator, Singular Point

目 次

1. 序 論	1
2. 計算モデル	2
2.1 運動学	2
2.2 線形化による逆変換	3
2.2.1 逆変換のアルゴリズム	3
2.2.2 アルゴリズムの概要	4
2.2.3 データの入力と順変換	5
2.2.4 Gauss-Jordan 法による掃き出し	5
2.2.5 デカルト座標での補間	6
2.3 特異点	6
2.3.1 特異点の性質	6
2.3.2 特異点の処理	7
3. 結果と考察	10
3.1 特異点以外での数値実験	10
3.1.1 線形化の精度とその限界	10
3.1.2 冗長性	11
3.2 特異点での数値実験	12
4. 結 論	13
謝 辞	14
参考文献	14
記 号	15
付録 1 変換行列（並進及び回転）	30
付録 2 特異点処理のためのアルゴリズム	31

CONTENTS

1. Introduction	1
2. Calculating Model	2
2.1 Kinematics	2
2.2 Inverse Transformation in Terms of Linearization	3
2.2.1 Algorithm of the Inverse Transformation	3
2.2.2 Outline of the Algorithm	4
2.2.3 Inputting the Data and Forward Transformation	5
2.2.4 Sweeping out by Gauss-Jordan Method	5
2.2.5 Interpolation in Cartesian Coordinates	6
2.3 Singular Points	6
2.3.1 Character of Singular Points	6
2.3.2 Process at Singular Points	7
3. Results and Discussion	10
3.1 Numerical Experiments at Nonsingular Points	10
3.1.1 Accuracy and Limitation of Linearization	10
3.1.2 Redundancy	11
3.2 Numerical Experiments at Singular Points	12
4. Conclusion	13
Acknowledgments	14
References	14
Nomenclature	15
Appendix 1. Transformation Materices (Translation and Rotation) ...	30
Appendix 2. Algorithm for Process at Singular Points	31

1. 序 論

電動多関節マニピレータの操作はその各関節を動かすアクチュエータを駆動するための電気的信号によって行われる。よって、それらの信号は個々の関節の動きに関連した量となっている。ところが、マニピレータの作業対象は通常デカルト座標で表されるため、関節の動作に関連した座標からデカルト座標への変換（順変換）及びその逆変換が常に行われる必要がある。このうち順変換は 4×4 の同次変換を用いて容易にかつ一意的に求まる¹⁾。すなわち、任意の関節座標を q 、それに対応するデカルト座標を p とすれば、

$$p = f(q) \quad (1.1)$$

のように陽に p を表すことが可能である。しかし、関数 f は一般にマニピレータの回転関節に關係するため、超関数を含んだ非線形関数になる。そのため、逆変換（すなわち q を陽に表すこと）は連立非線形方程式を解くことに帰着される。特殊な場合に対するこの解法についての報告¹⁾もあるが、アームの形状に依存しない統一的方法はない。そのうえ解析解が存在しない場合もあり、これについては繰り返し計算に頼るほかはない。繰り返し計算には、アームの幾何学的形状に注目し 2 つのリンク群にわけて交互に繰り返し計算を行うもの²⁾もあるが、一般的には逆ヤコビアンによる方法が用いられている。本報告でもこの方法について検討を加える。詳しい計算法については 2 章に譲るが、この方法は数学的にはニュートン法の多次元への拡張であり、(1.1)において解の予想 q_0 を行い、非線形問題を f の導関数の q_0 における値を含むような線形問題で近似する方法である。しかし、この近似法の精度に関してはあまり明らかにされていない。また、この方法によって繰り返し計算を行い、正しい解に収束させるためには、許容される q_0 は解に対して限られている。このような逆ヤコビアンによる方法における研究の動向を鑑み、数値実験を行った結果、逆ヤコビアンによる方法の精度及び限界に関する情報を得ることができた。

また、ヤコビアンが特異な場合（ヤコビアンの行列式が 0 となる場合）にはそれに対応する逆行列（逆ヤコビアン）が求まらないので、通常の場合この方法は使えない。しかし、特異点はマニピレータの観点からは関節の自由度が退化した点であり、すべての自由度が不具になるわけではない。よって、残された自由度によって計算をすることができる³⁾。ただしこの計算には、特異なヤコビアンの中の、残された自由度に対応する正則な小行列が必要である。本研究では、この計算をソフトウェア上で実現し、マニピレータが特異点近傍にある場合でも逆ヤコビアンの方法で計算を行うことを可能にした。また、この計算法の性能を数値実験によって確認した。

2. 計算モデル

2.1 運動学¹⁾

一般に多関節マニプレータは関節によって結びついた一連のリンクから成っている。よって、このリンク機構の運動は各リンクに固定した座標系（リンク座標系）の運動を合成することにより表現される。

Fig. 2.1 に今回解析対象に選択したマニプレータの略図とそのリンク座標系を示す。リンク座標系の選び方は、最も普及している、Denavit – Hartenberg 記法⁴⁾に従った。

これらのリンク座標系間の相対的な位置姿勢は A 行列と呼ばれる 4×4 の行列によって表される。例えば Fig. 2.2 に示すような 2 つの座標系 S_1, S_2 を考える。 S_2 は S_1 の z 軸を $\pi/2$, y 軸を $\pi/2$ 回転しさらに (a, b, c) だけ並進させたものと一致する。すなわち、次式が S_1 から見た S_2 の位置姿勢である。

$$A = \text{Trans}(a, b, c) \text{Rot}(y, \pi/2) \text{Rot}(z, \pi/2) \quad (2.1)$$

ただし、Trans(a, b, c) は x, y 及び z 方向へ a, b 及び c だけ並進させる変換行列であり、Rot(k, θ) は k 軸のまわりに θ だけ回転させる変換行列である。変換行列についての詳しい説明は付録 2 で行う。 A_1 を基準座標から見た 1 番目のリンク座標系の位置姿勢とし、 A_2 を 1 番のリンク座標系から見た 2 番目のリンク座標系の位置姿勢とすれば、基準座標から見た 2 番目のリンク座標系の位置姿勢を表す変換行列は

$$T_2 = A_1 A_2 \quad (2.2)$$

となる。同様な手法で基準座標系から見た 6 番目のリンク座標系の位置姿勢は

$$T_6 = A_1 A_2 A_3 A_4 A_5 A_6 \quad (2.3)$$

で与えられる。 T_6 の要素

$$T_6 = \begin{bmatrix} a_x & o_x & n_x & p_x \\ a_y & o_y & n_y & p_y \\ a_z & o_z & n_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

において、 $p = (p_x, p_y, p_z)$ は基準座標で表した 6 番目のリンク座標系の原点位置、 $o = (o_x, o_y, o_z)$ 、 $a = (a_x, a_y, a_z)$ 、及び $n = (n_x, n_y, n_z)$ は基準座標における 6 番目のリンク座標系の x, y, 及び z 方向（すなわち x_6, y_6 , 及び z_6 方向）の単位ベクトルである。しかし入力の便宜上、マニプレータがホームポジションすなわちすべての関節角が 0° のとき、 T_6 の回転に関する部分行列

$$\begin{bmatrix} a_x & o_x & n_x \\ a_y & o_y & n_y \\ a_z & o_z & n_z \end{bmatrix}$$

が単位行列となるように次式のように補正する。

$$T_6 = A_1 A_2 A_3 A_4 A_5 A_6 \cdot \text{Rot}(y_6, \pi/2) \text{Rot}(z_6, \pi/2) \quad (2.5)$$

このように補正するとマニプレータの姿勢は基準座標系からの変位で記述される。補正後の T_6 によって変換されたリンク座標系を (x'_6, y'_6, z'_6) として Fig. 2.1 に示す。今回解析対象に選択したマニプレータ (Fig. 2.1) の各変換行列は以下のようになる。

$$\begin{aligned} A_1 &= \text{Rot}(z_0, \theta_1) \text{Trans}(0, 0, a_1) \text{Rot}(y_0, -\pi/2) \\ A_2 &= \text{Rot}(z_1, \theta_2) \text{Trans}(a_2, 0, 0) \\ A_3 &= \text{Rot}(z_2, \theta_3) \text{Trans}(a_3, 0, 0) \text{Rot}(y_2, \pi/2) \\ A_4 &= \text{Rot}(z_3, \theta_4) \text{Trans}(0, 0, a_4) \text{Rot}(y_3, -\pi/2) \\ A_5 &= \text{Rot}(z_4, \theta_5) \text{Trans}(a_5, 0, 0) \text{Rot}(x_4, -\pi/2) \\ A_6 &= \text{Rot}(z_5, \theta_6) \text{Trans}(a_6, 0, 0) \end{aligned} \quad (2.6)$$

また、補正した T_6 の要素を Table 2.1 に示す。

このように、 $\theta_1 \sim \theta_6$ が決れば T_6 の要素の式にそれらを代入し、6番目のリンク座標系の位置姿勢すなわちマニプレータの先端の位置姿勢を基準座標で表すことができ、この関係に基づいてマニプレータの運動を記述することができる。

2.2 線形化による逆変換

2.2.1 逆変換のアルゴリズム

マニプレータの位置姿勢を指定するのに T_6 を用いることは上に述べたとおりである。しかしながら実際には、マニプレータの位置を指定するのに 3 自由度、姿勢を指定するのにはもう 3 自由度、合計 6 自由度が必要なだけである。 T_6 の要素のうち意味のあるものの数は 3 行目までの 12 個である。これらのうち p はマニプレータの位置を表し、マニプレータの作業領域内で何の制約もないに指定できるが、 n 、 o 、及び a についてはそれが単位長さをもち互いに直交するという 6 個の制約条件があり、それが独立ではない。よってマニプレータの姿勢を指定するには n 、 o 、及び a の計 9 要素のうち独立な 3 要素だけで十分である。今回は姿勢を指定するのに a_y 、 n_z 及び a_z を用いた。すなわちマニプレータの位置姿勢は $p = (p_x, p_y, p_z, a_y, n_z, a_z)$ で記述される。ただし、2.1 では p を (p_x, p_y, p_z) として扱ったがこれ以後 p はすべて $(p_x, p_y, p_z, a_y, n_z, a_z)$ を表すものとする。これをデカルト座標での位置姿勢の表現とする。また、6 個の関節角によってもマニプレータの位置姿勢は一意的に決定できるので $q = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ も 1 つの位置姿勢の表現である。これを関節座標での位置姿勢の表現とする。ここで便宜上

$$\begin{aligned} p &= (p_x, p_y, p_z, a_y, n_z, a_z) = (p_1, p_2, p_3, p_4, p_5, p_6) \\ q &= (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) = (q_1, q_2, q_3, q_4, q_5, q_6) \end{aligned} \quad (2.7)$$

とおく。

2.1 で述べたように q が決れば T_6 を用いて p は一意的に決る。

$$p = f(q) \quad (2.8)$$

1 章すでに述べたが、(2.8) のように q より p を求めることを、順変換または直接変換と呼ぶ。逆に p より q を求めることを逆変換と呼ぶ。逆変換は特殊なリンク構造をもつマニプレータ

以外は一般的な解析解を期待できない。しかし、マニプレータ制御にとって位置の指定は必要不可欠なことであり、解析不可能なものに対しては以下のようにして近似的に解くことが一般的である。まず、(2.8)を微分して

$$dp = J(q) \cdot dq \quad (2.9)$$

となる。Jは 6×6 ヤコビアンであり、その要素は、

$$J_{ij} = (\delta p_i / \delta q_j), i = 1, 2, \dots, 6, j = 1, 2, \dots, 6 \quad (2.10)$$

である。今回の解析対象のマニプレータについてJの全要素をTable 2.2に示す。(2.9)より、近似的に次式が成り立つ。

$$\Delta p = J(q_0) \cdot \Delta q$$

ただし

$$\Delta q = q_d - q_0, \Delta p = p_d - p_0$$

ここに q_0, p_0 はそれぞれ関節座標、デカルト座標の初期値である(ただし、 $q_0 = f(q_0)$)。 p_d はデカルト座標で与えた目標値であり、 q_d はその目標値に対応する関節座標である。上の式より

$$q_d = q_0 + J^{-1}(q_0) \cdot \Delta p \quad (2.12)$$

となる。 p と q の関係は一般的に非線形であり(2.11)及び(2.12)は Δp が十分小さい場合にのみ良い近似となりうる。このようにして関節変数の近似解を求めることができるが、特に $\Delta p, \Delta q$ の許容変位幅を知ることがここで重要な課題となる。

2.2.2 アルゴリズムの概要

(2.12)に基づく関節角の導出過程を概略的に示す。

Step 1

関節角の初期値 q_0 、及びデカルト座標で与えた目標値 p_d を入力し、 p_0 を(2.8)によって求める。Step 2へ行く。

Step 2

最初($i = 1$)の近似として $q_1 = q_0, p_1 = p_0$ と置く。Step 3へ行く。

Step 3

(2.10)で与えたヤコビアン $J(q_i)$ を計算し、さらに(2.11)における q_0, p_0 を q_i, p_i とおき、 Δq の6成分を変数とする6元連立方程式を導く。 Δq はGauss-Jordan法により求める。

$$q_{i+1} = q_i + \Delta q \quad (2.13)$$

として q_{i+1} を(2.8)に代入し p_{i+1} を求める。ここで、

$$[(p_{xi+1} - p_{xd})^2 + (p_{yi+1} - p_{yd})^2 + (p_{zi+1} - p_{zd})^2]^{1/2} < EPS$$

(EPS: 要求精度から決める収束半径)

なら、その時の q_{i+1} を解とする。そうでない時は $i = i + 1$ としてStep 3を繰り返す。ただし、 p_{xd}, p_{yd} 及び p_{zd} は p_d のx, y及びz成分、 p_{xi+1}, p_{yi+1} 及び p_{zi+1} はi回の繰り返しにより算出された p_{i+1} のx, y及びz成分である。また、後述の計算例ではEPSを0.1mmとした。

以上の手順をフローチャートにして Fig. 2.3 に示す。通常は上記の範囲で処理できるが、これら 3 つの Step だけでは処理できない場合が発生する。その一つは Step 3において (2.11) が Δq について解けない場合である。これは連立方程式の解の不定もしくは不能の問題で詳しくは 2.3 で述することにする。また、同じく Step 3において何度この Step をくりかえしても収束しない場合がある。その原因の 1 つは、 Δp が大きすぎて妥当な線形化近似の範囲を越えているためであり、もう 1 つは p_d が、各関節がすべて 2π の範囲で動作可能であると仮定した場合の作業範囲を越えていて対応する関節座標 q_d が存在しないためである。これらの場合プログラムでは収束回数を 10 以内で計算続行を打ち切ることとした。また、収束して求まった q_d の各要素のうち少なくとも 1 つが各関節の動作範囲を越えている場合は、その旨を出力し計算を中止するようにした。ただし、Fig. 2.1 のマニプレータの個々の関節の動作範囲は以下の通りである。

$$\begin{aligned}\theta_1 &: -10\pi/9 \sim 10\pi/9 \\ \theta_2 &: -\pi/6 \sim 2\pi/3 \\ \theta_3 &: -\pi/2 \sim \pi/2 \\ \theta_4 &: -\pi \sim \pi \\ \theta_5 &: -\pi/2 \sim \pi/2 \\ \theta_6 &: -\pi/3 \sim \pi/3\end{aligned}$$

ここで用いた Δq を求める操作は、直接ヤコビアンの逆行列を求めて Δp の左から掛ける操作と同等ではあるが、計算過程を減らす点で優れている。以下ではプログラムの詳細について説明する。

2.2.3 データの入力と順変換

ここでは Step 1 についてより詳しく説明する。Step 1 ではまず関節座標の初期値 q_0 を入力する必要がある。本プログラムは 6 自由度の回転関節マニプレータを対象としているので、1 軸から 6 軸までの適当な関節角を入力する。次に、入力された関節座標によって順変換を行い、そのときのデカルト座標 p_0 を算出する。

次に、マニプレータ先端の目標値の位置と姿勢をデカルト座標で入力する。位置は基準座標系における位置 (p_1, p_2, p_3) を入力する。姿勢は直接 a_y, n_z 及び a_z で入力しても良いが、実際の姿勢がわかりにくないので基準座標系におけるホームポジションからのオイラー角及び Roll - Pitch - Yaw でも入力可能である。オイラー角及び Roll - Pitch - Yaw で入力された表現は即座に a_y, n_z 及び a_z に変換され、実際の計算ではオイラー角及び Roll - Pitch - Yaw は使われない。

2.2.4 Gauss - Jordan 法による書き出し

ここでは Step 3 の Gauss - Jordan 法による書き出しについて詳述する。(2.11) を連立方程式の形式に表せば

$$\begin{aligned}J_{11} \Delta q_1 + \dots + J_{16} \Delta q_6 &= \Delta p_1 \\ \dots &\\ J_{61} \Delta q_1 + \dots + J_{66} \Delta q_6 &= \Delta p_6\end{aligned}\tag{2.14}$$

これに対する拡大係数行列を表せば、

$$\left[\begin{array}{cccccc|c} J_{11} & \cdots & J_{16} & | & \Delta p_1 \\ & \ddots & & | & \\ J_{16} & \cdots & J_{66} & | & \Delta p_1 \end{array} \right] \quad (2.15)$$

(2.15) を Gauss-Jordan 法によって掃き出せば次式のようになる。

$$\left[\begin{array}{cccccc|c} 1 & & & & a_1 \\ O & \ddots & & & \\ & O & & & \\ & & 1 & & a_6 \end{array} \right] \quad (2.16)$$

ただし、(2.16) はヤコビアンが正則である場合に適用できる。(2.16) を再度、連立方程式の形式に表せば、

$$\begin{aligned} \Delta q_1 &= a_1 \\ \Delta q_2 &= a_2 \\ &\dots \\ \Delta q_6 &= a_6 \end{aligned} \quad (2.17)$$

以上のように Δq が求まる。

2.2.5 デカルト座標での補間

前述したように 1 回の計算過程では指定された 1 つの位置姿勢に対する 1 組の関節座標しか得られない。現在位置と目標位置が大きく離れている場合に、その間の連続的な軌道に対応した関節座標の軌道を求めるには、その間での補間が必要である。すなわち、 p_0 から p_d までを適当な分割数によって分割し、 p_0 から p_d に向かって各々の隣り合った内挿点間ごとにこのプログラムを実行するようとする。分割数は任意に解析者が与える。また、解を連続的に求めれば解の健全性も検証できる。

2.3 特異点

2.3.1 特異点の性質

特異点とは J が正則でない点、すなわち $\text{rank } J < 6$ となる点である。特異点においては J^{-1} は存在せず (2.12) は成り立たない。この場合は、実際のアルゴリズムで (2.16) のように単位行列と列ベクトルの形には掃き出せない。すなわち、特異点においては 2.2 の方法では逆変換を行えない。また、マニプレータの自由度という面から次のようなことが言える。既に述べたように、マニプレータの位置を指定するには 3 自由度が必要で、姿勢にもう 3 自由度が必要である。6 リンク・マニプレータは各リンクに 1 自由度ずつ計 6 自由度を持つように設計することができ、その動作範囲内で任意の位置と姿勢を取らせることができる。しかし、特異点では 6 リンク・マニプレータが有効な 6 自由度を持たない。すなわち、自由度が退化する。1 つの関節の動作が他の関節の動作によっても実現されるような位置姿勢においては、その関節はもはや有効な 1 自由

度を担ってはいなく、結果として自由度が減り、任意の位置姿勢を取らせることができない。特異点の位置はリンク座標系の取り方とマニピレータの形状に依存していて、大部分のマニピレータにはこの特異点が存在する。Fig. 2.1 のマニピレータで図のようなリンク座標を取れば

$$\theta_1 = \theta_4 = \theta_6 = 0 \quad (2.18.a)$$

$$\theta_3 = 0 \quad (2.18.b)$$

$$\theta_2 + \theta_3 = 0, \quad \theta_5 = 0 \quad (2.18.c)$$

等の関節座標で特異点となる。特異点の一例として $q = (0, \pi/6, \pi/6, 0, 2\pi/9, 0)$ の場合を考える。その点での J は Table 2.3 のようになっている。Table 2.3において第 1 列、第 4 列、及び第 6 列の列ベクトルは $A p_x A n_z$ 平面上にある。このことは第 1 軸、第 4 軸、及び第 6 軸によるマニピレータの動作はそれぞれ他の 2 つの軸の動作の合成によって達成できることを意味する。つまり、これら 3 つ関節で自由度が実質的に 2 つしかなく全体として有効な 6 自由度を持つことができない。さらに第 4 行、第 6 行の行ベクトルが平行である。これは自由度が減ったため a_y と a_z を独立に動かすことができないことを意味し、そのためこの特異点ではマニピレータの先端位置を固定したまま手首を左右に振ることができない。

2.2 で示した方法では、1 つもしくはそれ以上の関節が退化すると、そこを起点とした Δq が求まらず、特異点を通る軌道は計算できない。しかし実際のマニピレータでは自由度の退化に応じてある程度運動が拘束されるが制御可能である。すなわち、自由度が退化していないならばマニピレータは 6 次元空間で任意に微小変位可能であるが、自由度が退化している場合は 6 次元より少ない次元の部分空間内でのみ微小変位可能である。例えば Table 2.3 に対応するマニピレータの状態からは、姿勢を変えずに位置のみの移動は可能である。このように特異点を含んだマニピレータの軌道は存在している。よって、特異点を含んだ軌道を求めることができればマニピレータの軌道を計画するときに非常に都合が良い。以下に特異点を起点とした Δq の求め方について述べる。

2.3.2 特異点の処理

特異点においては軸が退化した数だけ自由度が減る。よって、それに合わせて関節の自由度も減らして、サイズの小さなヤコビアンを求めるには正則である³⁾。そのヤコビアンを使えば 2.2 と同様に Δq を求めることができる。Table 2.3 においては 2.3.1 で述べたように第 1 列、第 4 列及び第 6 列のうちのどれか 1 つの列、第 4 行及び第 6 行のうちどれか 1 つそれぞれ必要がない。例えば、第 6 行と第 6 列を取り去った残りの 5×5 の行列は正則であり $\Delta q_1 \sim \Delta q_5$ が求まる。また、第 6 軸は動かさないので $\Delta q_6 = 0$ である。このようにして特異点を含んだ軌道が求まる。しかし、実際の計算では視察によって行と列を削るわけにはいかない。また、Table 2.3 では比較的単純に退化しているが、一般には視察で正則な小行列を検出するのは困難である。そこで、ソフトウェアに上記のような機能をもたせねばならない。そのためのアルゴリズムの概要を以下に示す。

特異点において (2.15) を Gauss-Jordan 法によって掃き出していくと、 J が正則でないため、行交換だけではいずれは 0 でない pivot を選ぶことができなくなる。その場合、その列をそのままにして pivot の位置を同じ行の次の列に変えて掃き出しを再開する。このような操作を

加えることにより、(2.15)を階段行列に変形する。この階段行列を列交換によって整理すると次式のようになる。

$$\left[\begin{array}{cccc|c} 1 & & 0 & & a_1 \\ & & * & & a_k \\ 0 & & 1 & & a_{k+1} \\ & & & & a_6 \\ \hline & & & & \\ & O & & & \end{array} \right] \quad (2.19)$$

(2.19)中のkはヤコビアンのランクすなわち特異点での自由度を表し、6-kが軸の自由度の退化の次数を表している。(2.19)に対応する方程式は

$$\begin{aligned} \Delta q_1 + b_{1,k+1} \Delta q_{k+1} + \cdots + b_{1,6} \Delta q_6 &= a_1 \\ \Delta q_2 + b_{2,k+1} \Delta q_{k+1} + \cdots + b_{2,6} \Delta q_6 &= a_2 \\ \dots & \\ \Delta q_k + b_{k,k+1} \Delta q_{k+1} + \cdots + b_{k,6} \Delta q_6 &= a_k \\ 0 &= a_{k+1} \\ \dots & \\ 0 &= a_6 \end{aligned} \quad (2.20)$$

(b_{ij} : 掃き出し後のi行j行の要素)

となる。ただし、(2.20)中の $\Delta q_1 \sim \Delta q_6$ は列交換のため(2.14)中の $\Delta q_1 \sim \Delta q_6$ とは順番が違っている。 $a_{k+1} \sim a_6$ がすべて0ならば連立方程式の解不定の場合である。解が不定であるということは、計画された手先交換器の動作に対して関節の動作がその起点において冗長性をもつことを意味する。そこで Δq_{k+1} から Δq_6 までに0を代入する。これは上で述べた関節の自由度減らしたことに対応する。よって(2.20)は

$$\begin{aligned} \Delta q_1 &= a_1 \\ \Delta q_2 &= a_2 \\ \dots & \\ \Delta q_k &= a_k \end{aligned} \quad (2.21)$$

となり、すべての Δq の要素が求まる。得られた解は、上で行った列交換を逆に行い、との関節に対応させる。以上のような操作は前述の小行列による逆変換と同値である。

$a_{k+1} \sim a_6$ のうち少なくとも1つが0でない場合には、連立方程式が解不能の場合であり、解は存在しない。これは、自由度が退化したため、特異点からデカルト座標での目標点の方向へ微小変位する解が存在しないことに対応している。ただし、目標点の関節座標の解は存在する可能性はある。そこで、解不能であっても $a_{k+1} \sim a_6$ がある程度小さければすべて0とみなし、

上記の解不定の場合と同様に処理する。すなわち、解不定の場合の処理で得られる解で実際の解を近似してしまう。そしておいて繰り返し計算を続行し、求まった解について、解の連続性よりその健全性を確認する。これは一見無謀なようであるが、 Δp が十分小さい場合にはプログラムを弾力化でき有効な解法となる。以上特異点の処理について簡単に述べてきたが詳細なアルゴリズムについては付録 2 を参照されたい。

3. 結果と考察

3.1 特異点以外での数値実験

3.1.1 線形化の精度とその限界

まず最初に、以下の議論のために特異でない関節空間の3点A, B及びCをTable 3.1に定義しておく。以下ではこれらA, B及びCを起点とした計算を行い、それらの結果について考察する。

任意の起点よりデカルト空間のある方向へ Δp だけ変位させた目標点の関節座標を計算し、その関節座標を起点としてまた同じ方向へ Δp だけ変位させた目標点の関節座標を計算する。以下同様に順次計算していくと、マニピレータの作業範囲は限られているので、そのうちに計算が続行できなくなる。（実際のアルゴリズムでは最終的な目標点を仮想的にマニピレータの作業範囲を十分越えるような遠方にとり、起点と目標点の間にデカルト空間において多数の補間点をとりその補間点の関節座標を順次計算していく。）計算が続行不可能になる原因として次のようなことを前述した。

- 1) Δp が大きすぎて線形化の限界を越えている。
- 2) 各関節が 2π の範囲で動作したとしても目標点に対応する関節座標が存在しない。
- 3) 各関節のうち少なくとも1つの関節が2章で示した動作範囲を越えている。

既に述べたように、1), 2) は繰り返し計算が収束せず計算が続行不可能になる場合で、3) は収束した関節解が動作範囲を越えていて計算を中止する場合である。よって、起点に対してある方向の作業範囲とそこまでの関節座標の動きを計算するためには1) の理由で計算が止まらないようにする必要がある。そのためには、 Δp を十分小さくとらねばならない。このことについて予備実験を繰り返した結果、マニピレータ先端の姿勢が一定である動作については Δp が10 mm以内なら問題がないことが確かめられた。

まず、起点Aについて Δp を10 mmとしてx方向に姿勢一定で前述のような数値実験を行った結果をTable 3.2に示す。左側第1列の数字はAを原点としたときのx座標で第2列の数字は計算前の起点と目標点のデカルト空間での距離（これは当然10 mmになる）で第3列は1回の線形化によって得られた関節座標と目標点との距離であり、第4列、第5列はそれぞれ2回目、3回目の値となっている。収束条件はこの距離が0.1 mm以内に入ることとしたので、その条件を満たせばその時の関節座標を関節解としてそれ以降の計算は行わない。Table 3.2より、x座標が-50 mmと130 mmを目標点とした計算は収束に至っていない。これは前述した2) の理由によるものである。すなわち、マニピレータ先端の姿勢が一定でAを通るx方向の作業範囲の限界は-x方向で-50 mmから-40 mmの間、+x方向では120 mmから130 mmの間ということになる。ただし、以下では収束した目標点の最大幅、すなわちこの例では-40 mm～120 mmを作業範囲と呼ぶことにする。Table 3.2よりわかることは、第一に、作業範囲内において1回の線形化計算によってすべて1 mm以内に収っていることである。起点と計算の方向を変えて、同様に10

mm 巾で姿勢一定の条件で数値実験を行った結果、ほとんどの場合で 1 回の計算で 1 mm 以内の収束条件を満たすことがわかった。現在市販されているマニピレータの位置決め精度は 1 mm 程度であるのでこの計算条件においては 1 回の計算でほぼ十分である。

また、Table 3.2 より、作業範囲の両端で近似の精度が悪くなっているのがわかる。このことをより深く議論するため、求まった 1 軸～6 軸の関節座標を Fig. 3.1 に示す。この図で 2, 3 及び 5 軸が作業範囲の両端で勾配が大きくなっている。線形化の性質上このことは近似の精度と深く関連していると推測される。つまり、勾配が大きいということは同じ Δp に対して Δq の絶対値が大きい、すなわち関節解の誤差が大きくなるので近似の精度は悪くなるはずである。一般に、3) の理由により計算が止まる場合以外は、作業範囲の両端で関節解のグラフの勾配が大きくなり近似の精度も悪くなる。次に Table 3.3 に B を起点として x 方向に計算した例を示す。ただし、-x 方向については +x 方向とほぼ原点対称なので割愛した。この表では作業範囲の中心で近似の精度が良くない。(ただし、両端の傾向は Table 3.2 と一致している。) 同じようにこの計算に対応した関節解を Fig. 3.2 に表す。この図ではグラフの中心で勾配が大きくなっている。このように、関節解のグラフの勾配と近似の精度の相関は明らかである。

以上、 Δp を 10 mm に固定した数値実験の例について考察してきた。以下では、起点を固定して Δp をある方向へ徐々に 10 mm から 10 mm ずつ大きくしていく数値実験を考える。この数値実験において、予め求めておいた作業範囲まで達していないうちに計算が止った場合、1) の理由によるものと考え、そのときの Δp より 10 mm 小さな値を線形化の限界と考えることができる。例として、A, B 及び C より x, y 及び z の + 及び - 方向、すなわち 1 つの起点に対して 6 方向へ上記の計算を行った。これらの計算結果を Table 3.4 にまとめた。左端のアルファベットは起点を、表中の分母は作業範囲の限界を、分子は線形化の限界をそれぞれ表している。A の x 方向については線形化の限界がそのまま作業範囲の限界になっていて、存在するすべての関節座標へ収束させることができある。しかし、y 及び z 方向では Δp がそれぞれ 130 mm, 30 mm を越えると収束させることができない。B については逆に x 方向で線形化の限界が作業範囲の限界に比較して小さく、B から直接収束不可能な関節座標が存在している。C では y 方向がそのような性質を有している。このように、線形化の限界は作業範囲の限界と深く係わっていると同時に、場合によっては作業範囲の限界よりもかなり狭くなっている場合もありうる。そして、その値はかなりばらついていて、規則性を見出すのは困難である。以上のようなことをまとめると、ヤコビアンを用いた線形化による逆変換の計算精度は関節解の勾配に依存していて、収束が可能なデカルト空間での変位の最大値は計算を開始する起点や計算の方向によって様々である。

3.1.2 冗長性

マニピレータの関節座標からの位置姿勢の決定、すなわち順変換は一意的である。これに対して逆変換には冗長性が存在する。例えば、関節座標 A'

$$A' = (0.042, 0.169, 0.171, 0.453, 0.216, 0.074)$$

単位 : rad

を順変換したものと Table 3.1 での A を順変換したものは収束半径内で一致している。すなわちデカルト座標に対応する関節座標の決定に冗長性がある。一般に関節解はいくつかの種類にわけ

ることができ、同じ種類の解の中ではデカルト座標に変換したとき冗長性がない。すなわち、逆変換に対して2つ以上の関節解が存在している場合、それらの解はそれぞれ解の種類を異にする⁴⁾。A'に対してx方向に3.1.1と同様な計算を行った結果をFig. 3.3に示す。Fig. 3.3より作業範囲は-70 mm～80 mmである。Aのx方向の作業範囲は-40 mm～120 mmであるので、解の種類が異なると作業範囲が顕著に変化することがわかる。2つの作業範囲を重合させると-70 mm～120 mmになるがこのアルゴリズムにおいては解の種類を変化させるような連続的な軌道は計算できない。このため、すべての作業範囲にわたっての連続的な軌道の計算ができない。このことは、逆ヤコビアンによる計算法の問題点の1つである。ただし、ここで連続的と言ったのは関節座標上で連続的（すなわち $q_1 \sim q_6$ が連続的）という意味である。マニプレータ先端の軌道がデカルト空間で連続的（すなわち $p_1 \sim p_6$ が連続的）であっても、実際の操作に直結する関節座標が連続的でなければマニプレータはなめらかに動作しない。

3.2 特異点での数値実験

前述したように、特異点とは $\det J = 0$ となる点である。このような点では自由度が退化して関節の動きに冗長性が出てしまう。そのため従来のヤコビアンを用いた逆変換では特異点を含んだ軌道は計算できなかった。以下では、2.3.2のアルゴリズムを用いた特異点を含んだ軌道の計算例を示し、それについて考察する。

まず特異点の例として $D = (0, \pi/6, \pi/6, 0, \pi/6, 0)$ を挙げる。Dは(2.18.a)及び(2.18.b)を満たしている。3.1.1で行ったように、特異点Eを起点としてx方向の作業範囲の限界までの補間点を用いた計算を行った。最初の掃き出しの結果をTable 3.5に示す。掃き出した後に6行目の要素がすべて0になっていることから退化の次数が1であり、7列目が0であることから、最初の補間点が点Dに対して“不定”的位置にあることがわかる。すなわち、点Dから最初の補間点へ微小変位するのに関節の動作に冗長性がでてしまう。ここで、2.3.2で述べたような計算を行っている。この計算においては最初の1回だけが特異点での計算であり、2回目以降は特異点を脱出していて通常の計算に戻っている。収束の様子をTable 3.6に示す。Table 3.6より、特異点での計算は特異点以外での計算の精度に比べてあまり変わらないことがわかる。また、求まった関節座標をFig. 3.4に示す。このグラフは特異点Dを通るマニプレータの軌道を示している。

また、同様な計算をz方向について行った。この計算においてはすべて特異点での計算になっていて、掃き出しの結果よりすべて“不定”的計算であることがわかった。求まった関節座標をFig. 3.5に示す。この場合も、作業範囲内ではすべて1回の計算で1 mm以内に収り、近似の精度は特異点以外の計算の場合とあまり変わってはいない。

次に、目標値が初期値に対して微小変位不可能である“不能”的位置にある場合の計算例として、点Eより基準座標においてTrans (0, -100, 0) RPY (0, 0, $\pi/18$)の変換を行った点に移動させることを考える。ただし、RPYはRoll, Pitch及びYawの回転をさせる変換行列である。補間点は9点で計算回数は10回である。ここに姿勢については方向余弦 a_y , n_z 及び a_z による直線内挿とする。計算結果のうち、点Eより1つ目の補間点までの計算過程の最初の

書き出しの様子をTable 3.8に示す。同表の6行目に注目すると、7列目を除いて0になっていることがわかる。このことは点Dにとって最初の補間点は“不能”の位置にあるということである。ここで、6行7列目の要素を0とおき計算を続行させる。すなわち、この“不能”の位置を“不定”的位置で近似してしまう。計算結果を前述の様式でTable 3.8に示す。この計算例では10回の計算過程においてすべて2度の繰り返し計算で収束条件を満たしている。

無論、近似となるような“不定”的位置が存在しない場合にはもともとそのようなマニプレータの動作が無理であるので計算を行う意味がない。ここで、“不定”的計算で近似していいかどうかは、繰り返し計算によって解が収束するかどうかで判断することができる。また、この計算により求まった関節解をFig. 3.6に示す。同図において、グラフがなめらかな曲線となっていることから、3.1.2で述べたように、求まった解は実際のマニプレータの軌道として実用性があることがわかる。

このように、一般の逆ヤコビアンによる逆変換では不可能であった特異点での計算を2.3.2で述べた方法で実現することができる。そして、その計算の精度は通常の計算の精度に比べてあまり変わらない。また、目標値が“不能”的位置、すなわち特異点に対してデカルト空間で直線的に微小変位可能でない位置にある場合でも繰り返し計算によって関節解を求めることができる。

いかなるマニプレータでも、その先端の位置姿勢を指定するのには高々6自由度必要なだけである。よって、7リンク以上の冗長性マニプレータにおいては、常に1つ以上の関節が退化している。すなわち、冗長性マニプレータは常に6リンク・マニプレータにおける特異点の状態にある。しかし、2.3.2で述べたようなアルゴリズムをリンクの数に応じて拡張すれば、これまで述べてきたように、その逆変換は可能である。

4. 結論

逆ヤコビアンの方法を用いて、デカルト座標で表現されたマニプレータの位置姿勢を関節座標での表現に変換し、それを順変換してもとのデカルト座標と比較するプログラムを作成した。また、それを用いて計算を行った結果、次のようなことがわかった。

- (1) 収束半径をマニプレータに先端位置にして0.1 mm以内としたとき、初期の変位を10 mmにすれば、ほとんどの場合、2回の繰り返し計算で収束する。
- (2) 1回の計算による精度は、デカルト座標でのマニプレータ先端の位置姿勢の変位に対する関節座標の変位の大きさに依存している。
- (3) 収束が可能なデカルト座標での変位の最大値は、計算開始点における関節座標や変位の方向によって様々である。

また、この計算法において関節座標の初期値に対するヤコビアンJが $\text{rank } J < 6$ である場合は特異点となり計算不能になる。そこで、Jのなかの正則な小行列を捜しそれを用いて逆変換する操作と同等な操作をソフトウェア上で実現させた。これにより、一般の逆ヤコビアンの方法ではできなかった特異点を初期値とした逆変換を可能にした。また、このプログラムを用いて計算を行った結果、次のようなことがわかった。

書き出しの様子をTable 3.8に示す。同表の6行目に注目すると、7列目を除いて0になっていることがわかる。このことは点Dにとって最初の補間点は“不能”の位置にあるということである。ここで、6行7列目の要素を0とおき計算を続行させる。すなわち、この“不能”の位置を“不定”的位置で近似してしまう。計算結果を前述の様式でTable 3.8に示す。この計算例では10回の計算過程においてすべて2度の繰り返し計算で収束条件を満たしている。

無論、近似となるような“不定”的位置が存在しない場合にはもともとそのようなマニプレータの動作が無理であるので計算を行う意味がない。ここで、“不定”的計算で近似していいかどうかは、繰り返し計算によって解が収束するかどうかで判断することができる。また、この計算により求まった関節解をFig. 3.6に示す。同図において、グラフがなめらかな曲線となっていることから、3.1.2で述べたように、求まった解は実際のマニプレータの軌道として実用性があることがわかる。

このように、一般の逆ヤコビアンによる逆変換では不可能であった特異点での計算を2.3.2で述べた方法で実現することができる。そして、その計算の精度は通常の計算の精度に比べてあまり変わらない。また、目標値が“不能”的位置、すなわち特異点に対してデカルト空間で直線的に微小変位可能でない位置にある場合でも繰り返し計算によって関節解を求めることができる。

いかなるマニプレータでも、その先端の位置姿勢を指定するのには高々6自由度必要なだけである。よって、7リンク以上の冗長性マニプレータにおいては、常に1つ以上の関節が退化している。すなわち、冗長性マニプレータは常に6リンク・マニプレータにおける特異点の状態にある。しかし、2.3.2で述べたようなアルゴリズムをリンクの数に応じて拡張すれば、これまで述べてきたように、その逆変換は可能である。

4. 結 論

逆ヤコビアンの方法を用いて、デカルト座標で表現されたマニプレータの位置姿勢を関節座標での表現に変換し、それを順変換してもとのデカルト座標と比較するプログラムを作成した。また、それを用いて計算を行った結果、次のようなことがわかった。

- (1) 収束半径をマニプレータに先端位置にして0.1 mm以内としたとき、初期の変位を10 mmにすれば、ほとんどの場合、2回の繰り返し計算で収束する。
- (2) 1回の計算による精度は、デカルト座標でのマニプレータ先端の位置姿勢の変位に対する関節座標の変位の大きさに依存している。
- (3) 収束が可能なデカルト座標での変位の最大値は、計算開始点における関節座標や変位の方向によって様々である。

また、この計算法において関節座標の初期値に対するヤコビアンJが $\text{rank } J < 6$ である場合は特異点となり計算不能になる。そこで、Jのなかの正則な小行列を捜しそれを用いて逆変換する操作と同等な操作をソフトウェア上で実現させた。これにより、一般の逆ヤコビアンの方法ではできなかった特異点を初期値とした逆変換を可能にした。また、このプログラムを用いて計算を行った結果、次のようなことがわかった。

- (1) 特異点での逆変換の計算精度は、通常の位置姿勢での逆変換のそれに比較して、あまり変わらない。
- (2) 特異点はマニピレータの自由度が退化している点なので、その状態のマニピレータは微小変位不可能な方向を有している。特異点である初期値に対して、目標値が微小変位可能な方向にある場合は無論解を求めることができる。しかし、目標値が微小変位可能な方向にない場合でも、微小変位可能な方向に計算を繰り返すことによりその解を求めることができる場合がある。

謝 辞

本報告を作成するにあたり、原子炉制御研究室長篠原慶邦氏、同研究室佐々木忍氏から貴重な助言をいただき、ここに深く謝意を表します。また、同研究室熊谷昭雄氏にプログラム作成上御協力していただき、ここに深く謝意を表します。

参考文献

- 1) R.P. Paul : "Robot Manipulator", MIT Press (1981).
- 2) V.J. Lumelsky : "Iterative Coordinate Transformation Procedure for One Class of Robot", IEEE Trans. on SMC 14-3, 500 (1984).
- 3) D.E. Whitney : "The Mathematics of Coordinated Control of Prosthetic Arms and Manipulators", ASME Trans. J. of Dynamic Systems, Measurement and Control, 303 (1972).
- 4) J. Denavit and R.S. Hartenberg : "A Kinematics Notation for Lower Pair Mechanisms based on matrices", J. of Applied Mechanics 22, 215 (1955).
- 5) 内山勝：“人工の手の運動制御に関する研究（第一報、特異点を考慮した協調運動の計算）”，日本機械学会論文集（C編），45-391，314（1979）.

- (1) 特異点での逆変換の計算精度は、通常の位置姿勢での逆変換のそれに比較して、あまり変わらない。
- (2) 特異点はマニピレータの自由度が退化している点なので、その状態のマニピレータは微小変位不可能な方向を有している。特異点である初期値に対して、目標値が微小変位可能な方向にある場合は無論解を求めることができる。しかし、目標値が微小変位可能な方向にない場合でも、微小変位可能な方向に計算を繰り返すことによりその解を求めることができる場合がある。

謝 辞

本報告を作成するにあたり、原子炉制御研究室長篠原慶邦氏、同研究室佐々木忍氏から貴重な助言をいただき、ここに深く謝意を表します。また、同研究室熊谷昭雄氏にプログラム作成上御協力していただき、ここに深く謝意を表します。

参考文献

- 1) R.P. Paul : "Robot Manipulator", MIT Press (1981).
- 2) V.J. Lumelsky : "Iterative Coordinate Transformation Procedure for One Class of Robot", IEEE Trans. on SMC 14-3, 500 (1984).
- 3) D.E. Whitney : "The Mathematics of Coordinated Control of Prosthetic Arms and Manipulators", ASME Trans. J. of Dynamic Systems, Measurement and Control, 303 (1972).
- 4) J. Denavit and R.S. Hartenberg : "A Kinematics Notation for Lower Pair Mechanisms based on matrices", J. of Applied Mechanics 22, 215 (1955).
- 5) 内山勝：“人工の手の運動制御に関する研究（第一報、特異点を考慮した協調運動の計算）”，日本機械学会論文集（C編），45 - 391，314 (1979).

- (1) 特異点での逆変換の計算精度は、通常の位置姿勢での逆変換のそれに比較して、あまり変わらない。
- (2) 特異点はマニピレータの自由度が退化している点なので、その状態のマニピレータは微小変位不可能な方向を有している。特異点である初期値に対して、目標値が微小変位可能な方向にある場合は無論解を求めることができる。しかし、目標値が微小変位可能な方向にない場合でも、微小変位可能な方向に計算を繰り返すことによりその解を求めることができる場合がある。

謝 辞

本報告を作成するにあたり、原子炉制御研究室長篠原慶邦氏、同研究室佐々木忍氏から貴重な助言をいただき、ここに深く謝意を表します。また、同研究室熊谷昭雄氏にプログラム作成上御協力していただき、ここに深く謝意を表します。

参考文献

- 1) R.P. Paul : "Robot Manipulator", MIT Press (1981).
- 2) V.J. Lumelsky : "Iterative Coordinate Transformation Procedure for One Class of Robot", IEEE Trans. on SMC 14-3, 500 (1984).
- 3) D.E. Whitney : "The Mathematics of Coordinated Control of Prosthetic Arms and Manipulators", ASME Trans. J. of Dynamic Systems, Measurement and Control, 303 (1972).
- 4) J. Denavit and R.S. Hartenberg : "A Kinematics Notation for Lower Pair Mechanisms based on matrices", J. of Applied Mechanics 22, 215 (1955).
- 5) 内山勝： “人工の手の運動制御に関する研究（第一報、特異点を考慮した協調運動の計算）”，日本機械学会論文集（C編），45 - 391，314 (1979).

Nomenclature

A or A_i	Homogeneous transformation matrix
C_i	$\cos \theta_i$
C_{ij}	$\cos(\theta_i + \theta_j)$
J	Jacobian of the transformation of a joint coordinate system into a cartesian coordinate system
n,o,a	Unit vectors defining the orientation of the end effector
P	Position vector of the end effector (in 2.1) The vector of cartesian coordinates (except 2.1)
q	The vector of joint coordinates
Rot	Rotation transformation matrix
S_i	$\sin \theta_i$
S_{ij}	$\sin(\theta_i + \theta_j)$
Trans	Translation transformation matrix
T_i	The position and orientation of the i-th link with respect to reference coordinate frame

Table 2.1 The elements of T_6 -matrix

n_x	$S_1(S_{23}S_6C_5+C_{23}S_5S_6C_4-C_{23}S_4C_6)+C_1(C_4C_6+S_4S_5S_6)$
n_y	$S_1(C_4S_6-S_4S_5S_6)-C_1(S_{23}C_5S_6+C_{23}S_5S_6C_4-C_{23}S_4C_6)$
n_z	$-C_5S_6C_{23}+S_5S_6C_4S_{23}-S_4C_6S_{23}$
o_x	$S_1(S_{23}S_5-C_{23}C_4C_5)-C_1S_4C_5$
o_y	$-S_1S_4C_5-C_1(S_{23}S_5-S_{23}C_4C_5)$
o_z	$-S_5C_{23}-C_4C_5S_{23}$
a_x	$-S_1(S_{23}C_5C_6+C_{23}S_5C_4C_6C_{23}S_4S_6)+C_1(C_4S_6-C_6S_4S_5)$
a_y	$S_1(C_4S_6-S_4S_5C_6)+C_1(S_{23}C_5C_6+C_{23}C_4C_6S_6+C_{23}S_4S_6)$
a_z	$C_5C_6C_{23}-C_4C_6S_5S_{23}-S_4S_6S_{23}$
p_x	$-S_1[a_6(S_{23}C_5C_6+C_{23}S_5C_4C_6+C_{23}S_4S_6)+a_6(S_{23}C_5+C_{23}S_5C_4)+(a_3+a_4)S_{23}+a_2S_2]+C_1[a_6(C_4S_6-S_4S_5C_6)-a_5S_4S_5]$
p_y	$S_1[a_6(C_4S_6-S_4S_5C_6)-a_5S_4S_5]+C_1[a_6(S_{23}C_5C_6+C_{23}S_5C_4C_6+C_{23}S_4S_6)+a_5(S_{23}C_5+C_{23}S_5C_4)+(a_3+a_4)S_{23}+a_2S_2]$
p_z	$a_6(C_5C_6C_{23}-S_5C_6S_{23}-S_4S_6S_{23})+a_5(C_5C_{23}-S_5C_4S_{23})+(a_3+a_4)C_{23}+a_2C_2+a_1$

$$S_i = \sin \theta_i, \quad C_i = \cos \theta_i$$

$$S_{ij} = \sin(\theta_i + \theta_j), \quad C_{ij} = \cos(\theta_i + \theta_j)$$

Table 2.2 The elements of J-matrix

J ₁₁	$S_1(a_6S_4S_5C_6+a_5S_5S_4-a_6S_6C_4)$ $-C_1(a_6S_2_3C_5C_6+a_5C_5S_2_3+a_6S_5C_6C_4C_2_3+a_5S_5C_4C_2_3+a_6S_6S_4C_2_3+a_3S_2_3+a_2S_2)$
J ₁₂	$S_1(-a_6C_2_3C_5C_6-a_5C_5C_2_3+a_6S_5C_6C_4S_2_3+a_5S_5C_4S_2_3+a_6S_6S_4S_2_3-a_3C_2_3-a_2C_2)$
J ₁₃	$J_{12}+a_2S_1C_2$
J ₁₄	$S_1(a_6S_5C_6S_4C_2_3+a_5S_5S_4C_2_3-a_8S_6C_4C_2_3)-C_1(a_6C_4S_5C_6+a_5S_5C_4+a_6S_6S_4)$
J ₁₅	$S_1(a_6S_2_3S_5C_6+a_5S_5S_2_3-a_6C_5C_6C_4C_2_3-a_5C_5C_4C_2_3)-C_1(a_6S_4C_5C_6+a_5C_5S_4)$
J ₁₆	$S_1(a_6S_2_3C_5S_6+a_6S_5S_6C_4C_2_3)+C_1(a_6S_4S_5S_6+a_6C_6C_4-a_6C_6S_4C_2_3)$
J ₂₁	$-S_1(a_6C_5C_6S_2_3+a_5C_5S_2_3+a_6S_5C_6C_4C_2_3+a_5S_5C_4C_2_3+a_6S_6S_4C_2_3+a_3S_2_3+a_2S_2)$ $+C_1(-a_6S_4S_5C_6-a_5S_5S_4+a_6S_6C_4)$
J ₂₂	$C_1(a_6C_5C_6C_2_3+a_5C_5C_2_3-a_6S_5C_6C_4S_2_3-a_5S_5C_4S_2_3-a_6S_6S_4S_2_3+a_3C_2_3+a_2C_2)$
J ₂₃	$J_{22}-a_2C_2C_1$
J ₂₄	$S_1(-a_6C_4S_5C_6-a_5S_5C_4-a_6S_6S_4+a_6S_6C_4C_2_3)-C_1(a_6S_5C_6S_4C_2_3+a_5S_5S_4C_2_3)$
J ₂₅	$-S_1(a_6S_4C_5C_6+a_5C_5S_4)+C_1(-a_6S_5C_6S_2_3-a_5S_5S_2_3+a_6C_5C_6C_4C_2_3+a_5C_5C_4C_2_3)$
J ₂₆	$S_1(a_6S_4S_5S_6+a_6C_6C_4)+C_1(-a_6C_5S_6S_2_3-a_6S_5S_6C_4C_2_3+a_6C_6S_4C_2_3)$
J ₃₁	0
J ₃₂	$-S_{23}(a_6C_5C_6+a_5C_5+a_3)+C_{23}(-a_6S_5C_4C_6-a_5S_5C_4-a_6S_4S_6)-a_2S_2$
J ₃₃	$J_{32}+a_2S_2$
J ₃₄	$S_{23}(a_6S_5S_4C_6+a_5S_5S_4-a_6C_4S_6)$
J ₃₅	$S_{23}(-a_6C_5C_4C_6-a_5C_5C_4)-C_{23}(a_6S_5C_6+a_5S_5)$
J ₃₆	$S_{23}(a_6S_5C_4S_6-a_6S_4C_6)-C_{23}(a_6C_5S_6)$

Table 2.2 (cont.)

J41	$S_1(-C_5C_6S_{23}-C_4C_{23}S_5C_6-S_4C_{23}S_6)+C_1(-S_4S_5C_6+C_4S_6)$
J42	$C_1(C_5C_6C_{23}-C_4S_{23}S_5C_6-S_4S_{23}S_6)$
J43	J42
J44	$-S_1(C_4S_5C_6+S_4S_6)+C_1(-S_4C_{23}S_5C_6+C_4C_{23}S_6)$
J45	$-S_1(S_4C_5C_6)+C_1(-S_5C_6S_{23}+C_4C_{23}C_5C_6)$
J46	$S_1(S_4S_5S_6+C_4C_6)+C_1(-C_5S_6S_{23}-C_4C_{23}S_5S_6+S_4C_{23}C_6)$
J51	0
J52	$S_{23}(C_5S_6)+C_{23}(C_4S_5S_6-C_6S_4)$
J53	J52
J54	$-S_{23}(S_4S_5S_6+C_6C_4)$
J55	$S_{23}C_4C_5S_6+C_{23}S_5S_6$
J56	$S_{23}(C_4S_5C_6+S_6S_4)-C_{23}C_5C_6$
J61	0
J62	$-S_{23}C_5C_6-C_{23}(C_4C_6S_5-S_4S_6)$
J63	J62
J64	$S_{23}(S_4C_6S_5-C_4S_6)$
J65	$-S_{23}C_4C_5C_6-C_{23}S_5C_6$
J66	$S_{23}(C_4S_6S_5-S_4C_6)-C_5S_6C_{23}$

Table 2.3 An example of Jacobian J at singular point

-0.9760E+3	0	0	0.1471E+3	0	0.2800E+3
0	0.6827E+3	0.2497E+3	0	0.7467E+2	0
0	-0.9766E+3	-0.7266E+3	0	-0.4253E+3	0
0	0.1736E+0	0.1736E+0	0	0.1736E+0	0
0	0	0	0.8660E+0	0	0.1736E+0
0	0.9848E+0	0.9848E+0	0	0.9848E+0	0

Table 3.1 The start points of the calculation without singularity

	θ_1 (rad)	θ_2 (rad)	θ_3 (rad)	θ_4 (rad)	θ_5 (rad)	θ_6 (rad)
A	$\frac{\pi}{18}$	$\frac{\pi}{18}$	$\frac{\pi}{18}$	$\frac{\pi}{18}$	$\frac{\pi}{18}$	$\frac{\pi}{18}$
B	$\frac{\pi}{180}$	$\frac{\pi}{9}$	$\frac{5\pi}{18}$	$\frac{\pi}{180}$	$\frac{\pi}{9}$	$\frac{\pi}{180}$
C	$\frac{\pi}{180}$	$\frac{89\pi}{180}$	$\frac{\pi}{9}$	$\frac{\pi}{180}$	$\frac{\pi}{6}$	$\frac{\pi}{180}$

Table 3.2 Convergent character with start point A, orientation x

Destination (mm)	Distance from Destination after n-th Iteration (mm)				Convergence (Y/N)
	0	1	2	3	
-50	10.00	0.403	0.108	0.462	N
-40	10.00	0.261	0.005		Y
-30	10.00	0.236	0.001		Y
-20	10.00	0.235	0.001		Y
-10	10.00	0.240	0.000		Y
0					
10	10.00	0.240	0.000		Y
20	10.00	0.247	0.000		Y
30	10.00	0.255	0.000		Y
40	10.00	0.262	0.000		Y
50	10.00	0.269	0.000		Y
60	10.00	0.275	0.000		Y
70	10.00	0.281	0.000		Y
80	10.00	0.287	0.000		Y
90	10.00	0.293	0.000		Y
100	10.00	0.303	0.001		Y
110	10.00	0.320	0.002		Y
120	10.00	0.366	0.008		Y
130	10.00	0.589	0.455	0.325	N

Table 3.3 Convergent character with start point B, orientation x

Destination (mm)	Distance from Destination after n-th Iteration (mm)				Convergence (Y/N)
	0	1	2	3	
10	10.00	1.661	0.005		Y
20	10.00	1.745	0.004		Y
30	10.00	1.935	0.008		Y
40	10.00	2.251	0.008		Y
50	10.00	2.730	0.014		Y
60	10.00	3.371	0.020		Y
70	10.00	3.925	0.018		Y
80	10.00	3.587	0.022		Y
90	10.00	2.022	0.028		Y
100	10.00	0.881	0.021		Y
110	10.00	0.479	0.007		Y
120	10.00	0.298	0.002		Y
130	10.00	0.205	0.001		Y
140	10.00	0.153	0.000		Y
150	10.00	0.122	0.000		Y
160	10.00	0.103	0.000		Y
170	10.00	0.091			Y
180	10.00	0.176	0.000		Y
190	10.00	0.081			Y
200	10.00	0.161	0.000		Y
210	10.00	0.080			Y
220	10.00	0.016	0.000		Y
230	10.00	0.083			Y
240	10.00	0.168	0.000		Y
250	10.00	0.089			Y
260	10.00	0.181	0.000		Y
270	10.00	0.098			Y
280	10.00	0.201	0.000		Y
290	10.00	0.111	0.000		Y
300	10.00	0.120	0.000		Y
310	10.00	0.132	0.000		Y
320	10.00	0.146	0.000		Y
330	10.00	0.166	0.000		Y
340	10.00	0.193	0.001		Y
350	10.00	0.233	0.001		Y
360	10.00	0.298	0.002		Y
370	10.00	0.416	0.007		Y
380	10.00	0.707	0.055		Y
390	10.00	2.368	27.89	7.438	N

Table 3.4 Limits of convergence

Start Points	Limits of Linearization and Limits of Workarea from Start Points(mm)					
	+x	-x	+y	-y	+z	-z
A	120/120	40/40	10/10	130/230	0/0	30/240
B	30/380	30/380	340/340	60/60	70/70	100/100
C	170/170	140/140	10/10	190/230	280/280	70/70

Table 3.5 Results of Gauss-Jordan method at indefinite point

Before Sweeping out

-0.983E+3	0	0	-0.215E+3	0	0.280E+3	0.100E+2
0	0.608E+3	0.175E+3	0	0	0	0
0	-0.983E+3	-0.733E+3	0	-0.430E+0	0	0
0	0	0	0	0	0	0
0	0	0	-0.866E+0	0	0	0
0	-1	-1	0	-1	0	0

After Sweeping out

1	0	0	0	0	0.285E+0	0.102E-1
0	1	0	0	0	0	0
0	0	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	1	0	0
0	0	0	0	0	0	0

Table 3.6 Convergent character with start point D, orientation x

Destination (mm)	Distance from Destination after n-th Iteration (mm)				Convergence (Y/N)
	0	1	2	3	
10	10.00	0.051			Y
20	10.00	0.122	0.005		Y
30	10.00	0.076			Y
40	10.00	0.148	0.001		Y
50	10.00	0.073			Y
60	10.00	0.146	0.000		Y
70	10.00	0.074			Y
80	10.00	0.148	0.000		Y
90	10.00	0.076			Y
100	10.00	0.153	0.000		Y
110	10.00	0.079			Y
120	10.00	0.159	0.000		Y
130	10.00	0.083			Y
140	10.00	0.169	0.000		Y
150	10.00	0.090			Y
160	10.00	0.184	0.000		Y
170	10.00	0.100	0.000		Y
180	10.00	0.108	0.000		Y
190	10.00	0.119	0.000		Y
200	10.00	0.134	0.000		Y
210	10.00	0.156	0.001		Y
220	10.00	0.194	0.002		Y
230	10.00	0.266	0.005		Y
240	10.00	0.452	0.043		Y
250	10.00	1.735	4.597	1.745	N

Table 3.7 Results of Gauss-Jordan method at impossible point

Before Sweeping out

-0.983E+3	0	0	-0.215E+3	0	0.280E+3	0
0	0.608E+3	0.175E+3	0	0	0	-0.100E+2
0	-0.983E+3	-0.733E+3	0	-0.430E+0	0	0
0	0	0	0	0	0	-0.150E-2
0	0	0	-0.866E+0	0	0	0
0	-1	-1	0	-1	0	0.174E-1

After Sweeping out

1	0	0	0	0	0.285E+0	0
0	1	0	0	0	0	0.496E-1
0	0	1	0	0	0	0.115E+0
0	0	0	1	0	0	0
0	0	0	0	1	0	-0.829E-1
0	0	0	0	0	0	-0.150E-2

Table 3.8 Convergent Character with start point D, orientation y and Yaw

Destination (mm)	Distance from Destination after n-th Iteration (mm)				Convergence (Y/N)
	0	1	2	3	
10	10.00	1.361	0.011		Y
20	10.00	0.955	0.004		Y
30	10.00	0.728	0.002		Y
40	10.00	0.589	0.001		Y
50	10.00	0.496	0.001		Y
60	10.00	0.429	0.000		Y
70	10.00	0.378	0.000		Y
80	10.00	0.339	0.000		Y
90	10.00	0.308	0.000		Y
100	10.00	0.283	0.000		Y

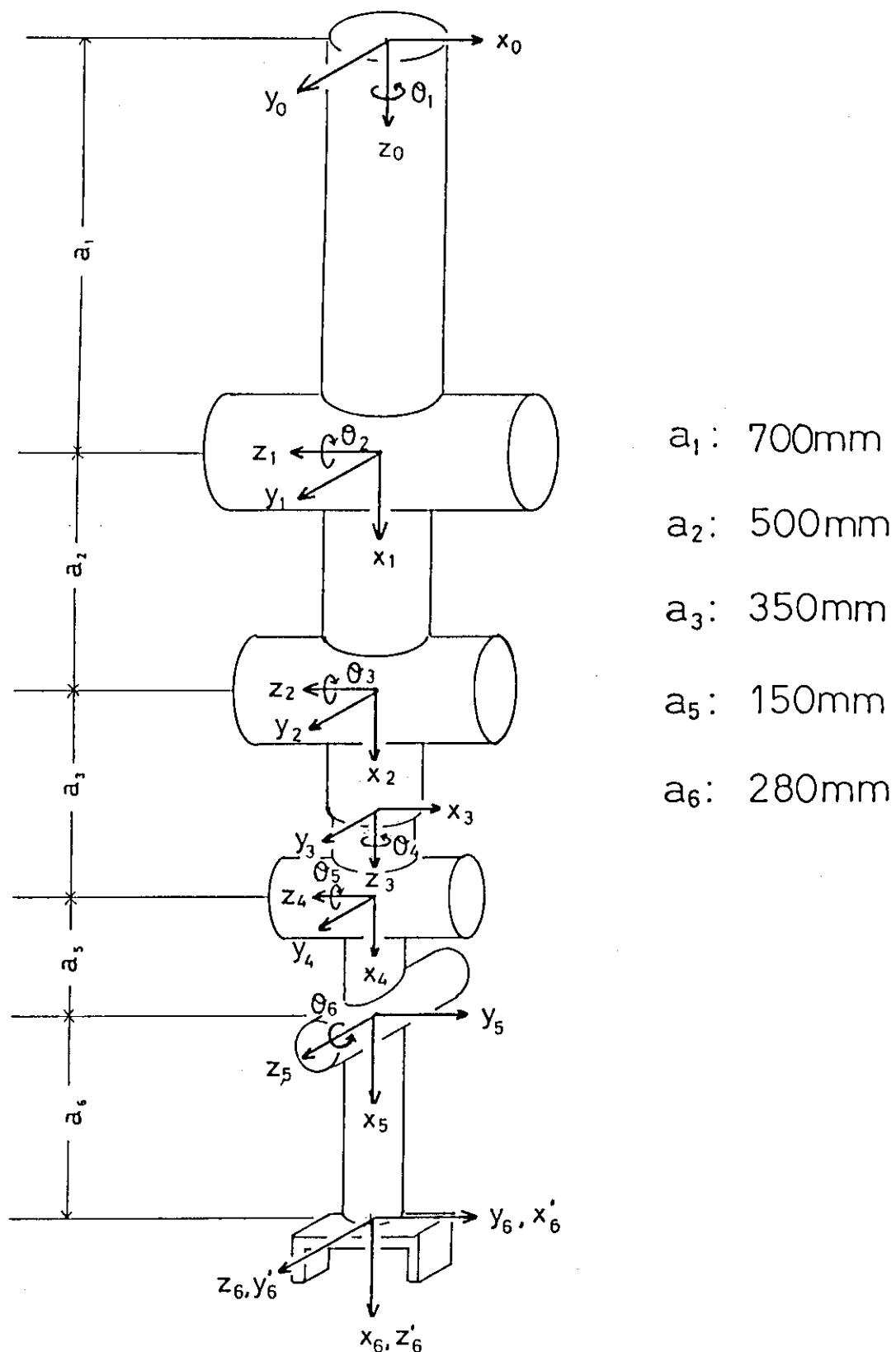


Fig. 2.1 Manipulator and link coordinate systems

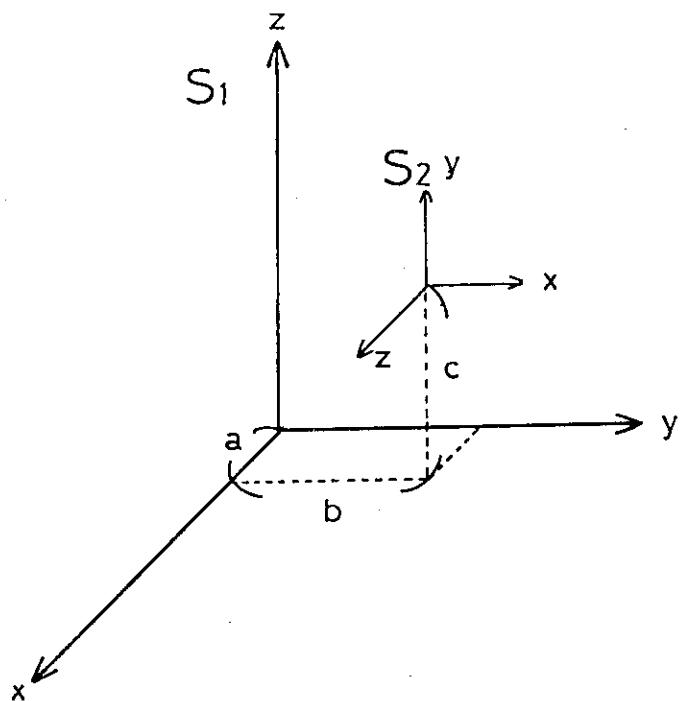


Fig. 2.2 Transformation of coordinate systems

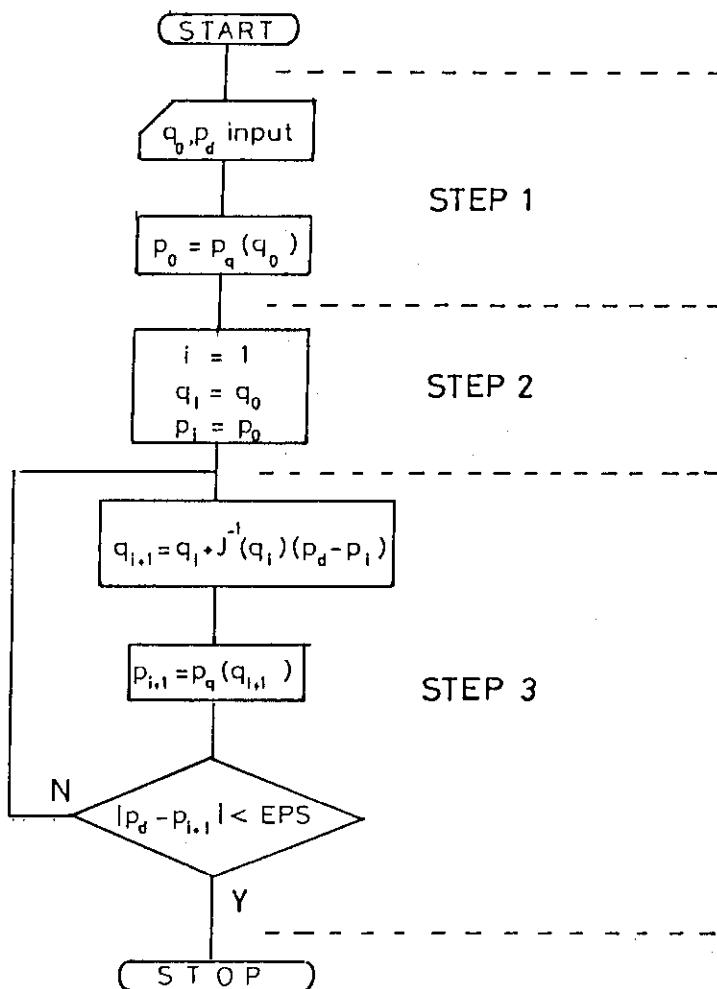


Fig. 2.3 Flowchart of inverse kinematic algorithm

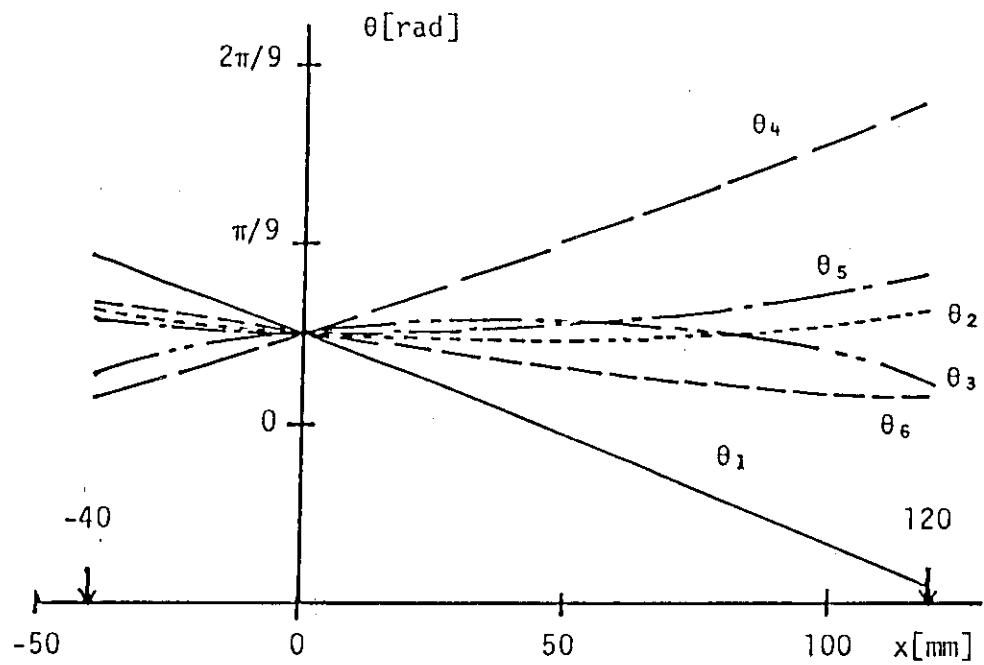


Fig. 3.1 Solution of joint angles with start point A, orientation x

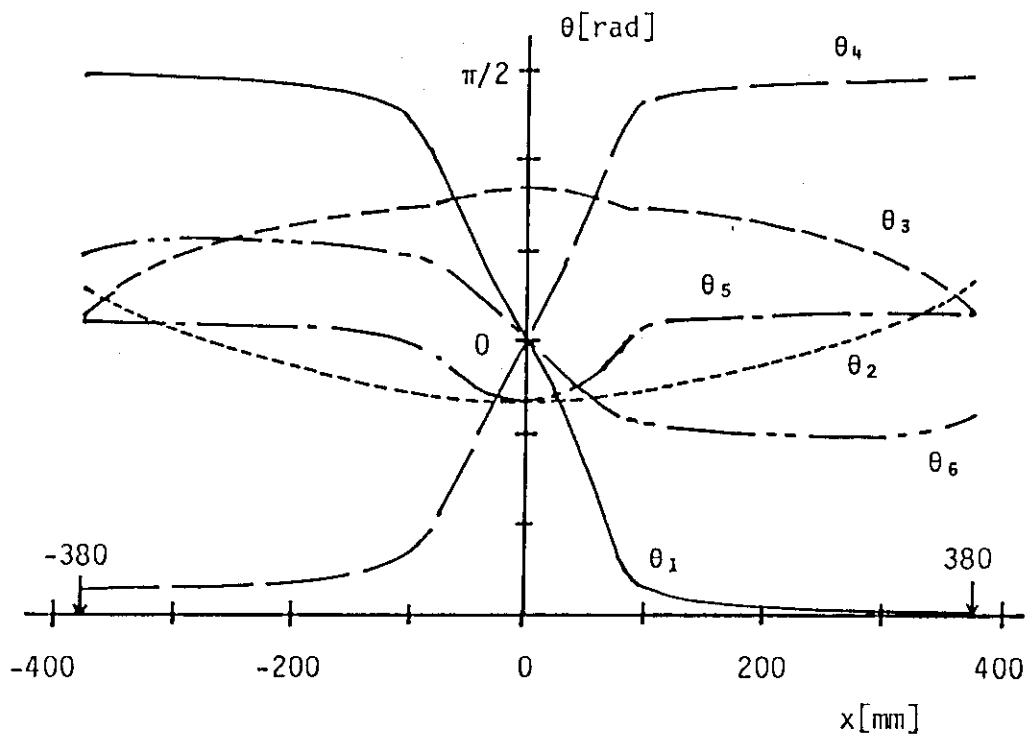


Fig. 3.2 Solution of joint angles with start point B, orientation x

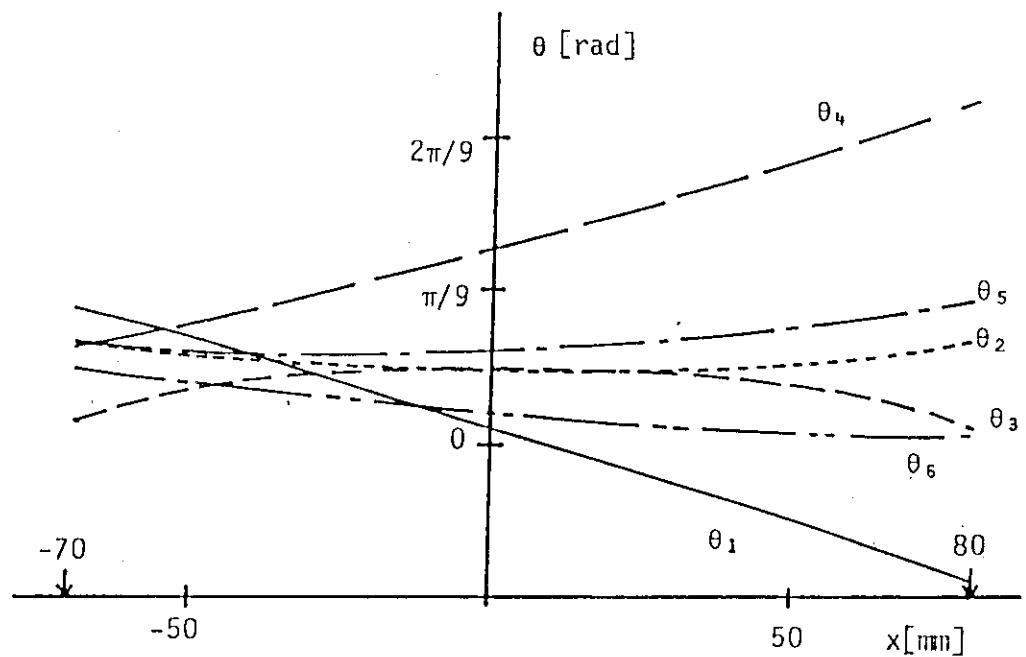


Fig. 3.3 Solution of joint angles with start point A', orientation x

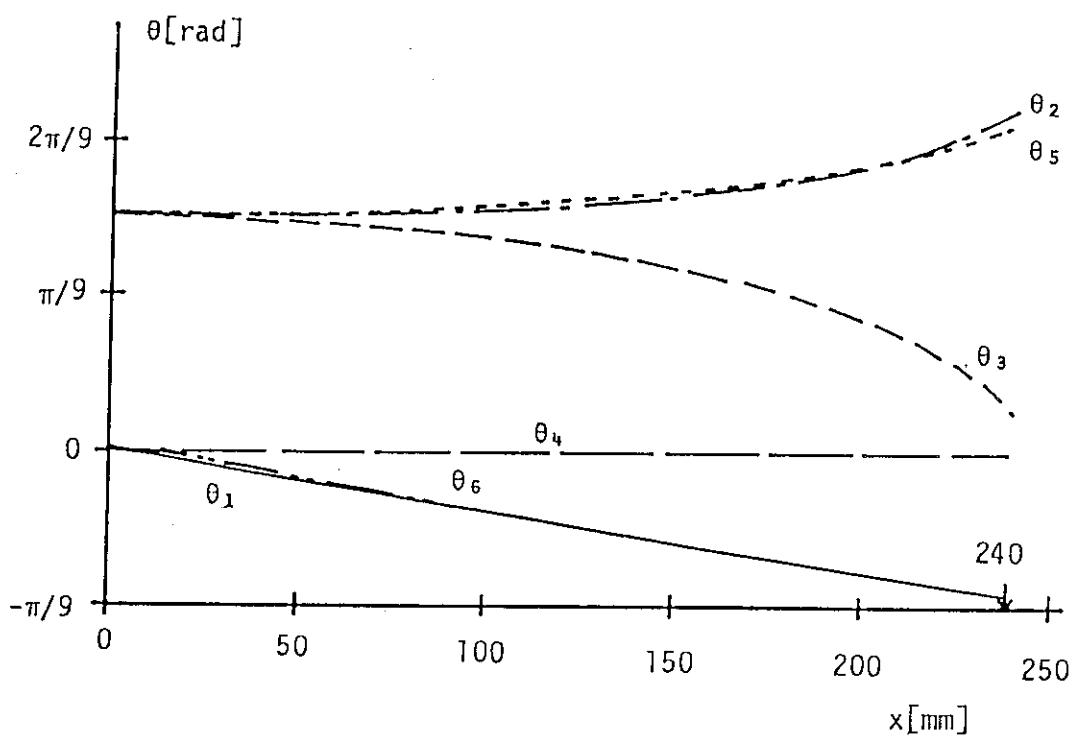


Fig. 3.4 Solution of joint angles with start point D, orientation x

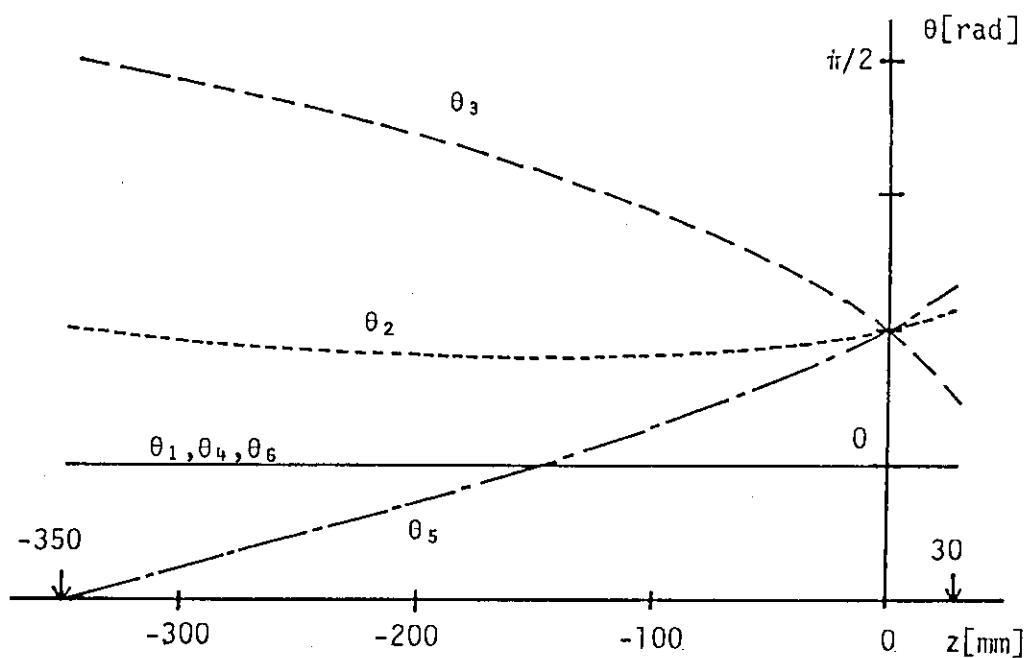


Fig. 3.5 Solution of joint angles with start point D, orientation z

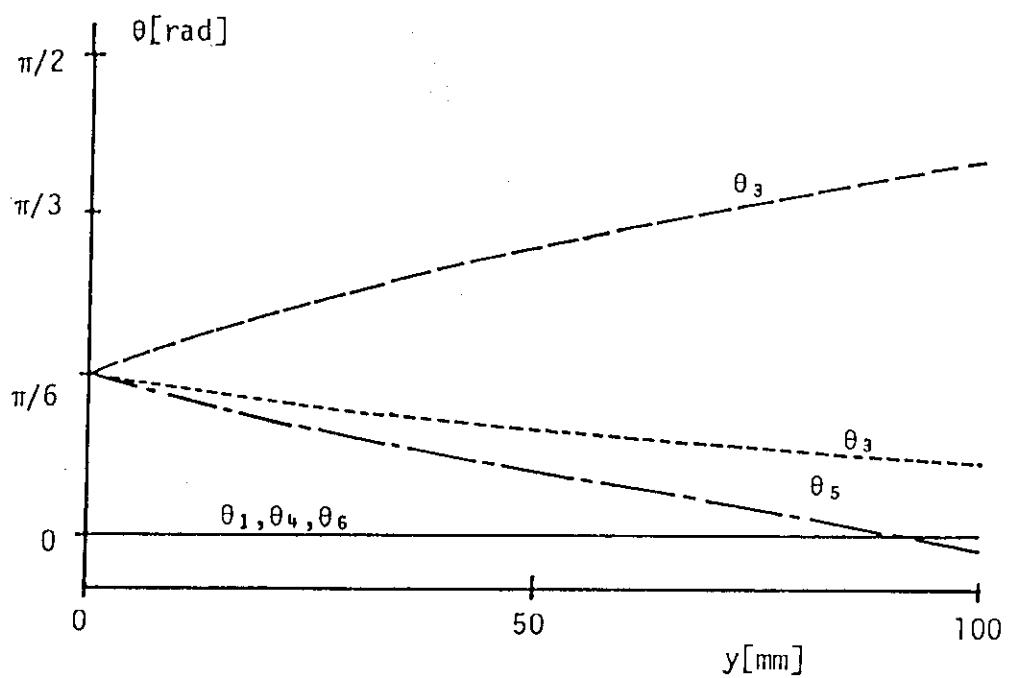


Fig. 3.6 Solution of joint angles with start point D, orientation y and Yaw

付録 1 変換行列（並進及び回転）

$\text{Trans}(a, b, c)$ は x, y, z 方向へ a, b 及び c だけ並進させる変換行列であり、 $\text{Rot}(k, \theta)$ は k 軸のまわりに θ だけ回転させる変換行列である。計算に使用した Trans および Rot の要素は次のように表現される。

$$\text{Trans}(a, b, c) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Rot}(x, \theta) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Rot}(y, \theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Rot}(z, \theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

付録 2 特異点処理のためのアルゴリズム

本報告で用いたプログラムの大筋はFig. 2.3 のフローチャートで示した。ここでは、特異点の処理の詳細なアルゴリズム及びそれとメインルーチン (Fig. 2.3)との関係について述べる。

Fig. 2.3 のステップ 3 では J^{-1} を使って

$$q_{i+1} = q_i + J^{-1} (q_i) (p_d - p_i) \quad (A 2.1)$$

と表されているが、実際には逆行列を求めて解く代りに連立方程式

$$\Delta p = J (q_i) \cdot \Delta q \quad (A 2.2)$$

ただし、 $\Delta p = p_d - p_i$

$$\Delta q = q_{i+1} - q_i$$

から Δq を求めている。(このことはすでに 2 章で述べた。) すなわち、ステップ 3 の最初のブロックには掃き出し法により連立方程式を解くプログラムが含まれており、また特異点を処理するプログラムはこのプログラムの中に含まれている。Fig. A. 1 に示したフローチャートはメインルーチンからの引数 $J (q_i)$ 及び Δp を行列表示にして掃き出し、特異点かどうかの判定を行い、解 (すなわち Δq) をメインルーチンに引き渡すサブルーチンである。フローチャート中の NSTOP は特異点判定のフラグであり、NSTOP = 1 は通常の計算、NSTOP = 2 はあるいは 3 は連立方程式の不定、不能の場合を示す。特に、NSTOP = 3 の場合は Δq は求めずに、計算を終了させる。

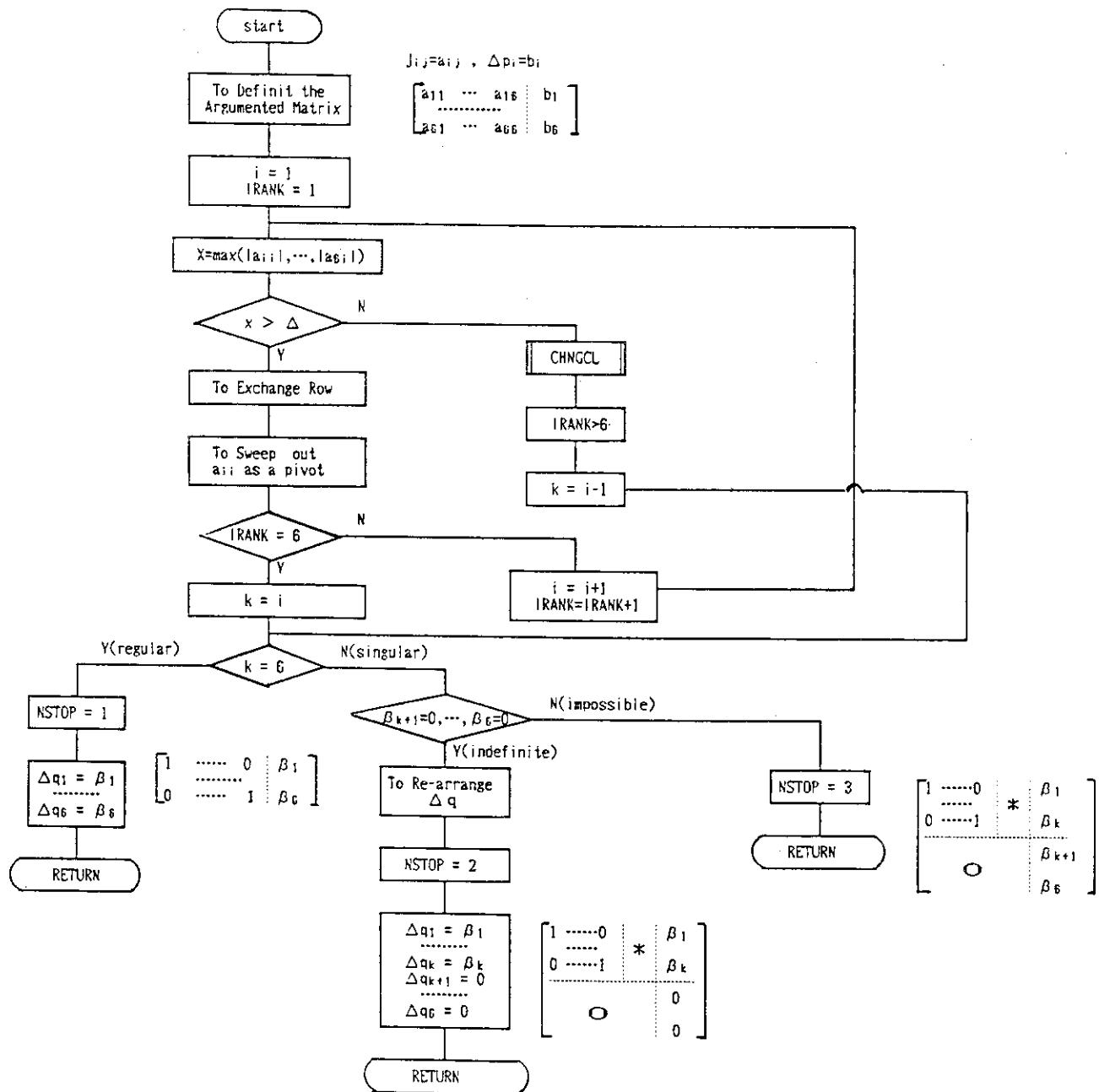


Fig. A.1 Flowchart of algorithm for process at singular points

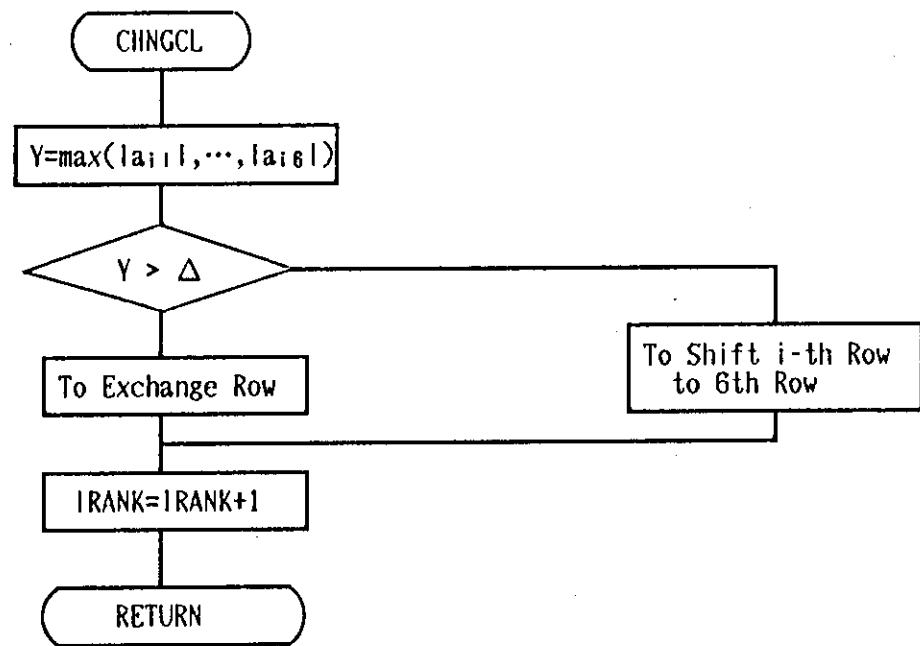


Fig. A.1 (cont.)