JAERI-M
86-059

COMPUTER CODE ARMI FOR SOLVING THE INVERSE KINEMATICS OF A SIX-LINK MANIPULATOR ARM

March 1986

Shinobu SASAKI

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。
入手の問合わせは，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）で複写による実費頒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, Tokai-mura, Naka-gun, Ibaraki-ken 319-11, Japan.

編集兼発行　日本原子力研究所
印　　刷　㈱高野高速印刷

Computer Code ARM1 for Solving the Inverse Kinematics of a Six-Link Manipulator Arm

Shinobu SASAKI

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

(Received February 24, 1986)

ARM1 is a FORTRAN 77 program for the numerical solutions of the inverse kinematics problems. This paper is a detailed description of the current program (version 1).

The code is simple enough for practical use and besides has no particular restrictions except for the specification of memory size. Incorporated checking statements or routine are useful in identifying the accuracies of solutions obtained.

Keywords: ARM1 Computer Code, Kinematics Problem, Bairstow's Method, Manipulator Arm, Numerical Solutions Accuracies

JAERI－M 86－059

6・リンク・マニプレータ・アームの逆運動学方程式を解く計算コードＡＲＭ１

日本原子力研究所東海研究所原子炉工学部
佐々木　忍

（１９８６年2月２４日受理）

ＡＲＭ１は，逆運動学方程式の数値解を求めるためのFORTRAN ７７プログラムである。本報告は，このコード（version １）を詳細に記述したものである。コードは，実用上その取扱が極めて簡単であり，さらに記憶容量の設定以外には特に制約をもたない。

組み込まれているチェック・ルーチンは，求めた解の精度を確認する上で有効である。

東海研究所：茨城県那珂郡東海村白方白根２の４

Contents

1. Introduction 1
2. Calculational Method of the Joint Variables 4
3. Computer Program Outline 9
4. Programming Details 12
5. Conclusion 21
Acknowledgements 21
References 21
Appendices 22

目　次

1. はじめに 1
2. 関節角変数の計算方法 4
3. 計算コードの概略 9
4. プログラミングの説明 12
5. おわりに 21
謝辞 21
参考文献 21
付録 22

List of Figures

Fig.1 Manipulator and Link Coordinate Systems

Fig.2 Overall System Layout of Code ARM1

1. Introduction

ARM1(version 1) is a digital computer code, written in FORTRAN 77, which was primarily developed to make a transparent geometrical angular description of each joint while the end-point of a multi-joint manipulator(6 DOF*) moves towards the destination position along some specified trajectory in three dimensional Euclidean space. Since the program has the capability to determine all possible solutions for the so-called "inverse kinematic problems", its application is of particular interest in identifying all possible orientation of the manipulator in the workspace.

As the underlying concept, some kinematic expressions are unified into a sigle polynomial and thereby, the obtained results contain much useful information in evaluating the mechanical performance or in determining the optimum joint angle design as well as in investigating the topological situations of mechanical link moving in the space.
In other words, the benefit derived by this approach is that feasible solutions latent within kinematic relationships can be extracted as explicit ones. In this respect, the ARM1 may be viewed as a new and powerful approach to deal with the inverse kinematics problems.

Before the execution, the program needs kinematic data which completely describe the structure dimensions of the system to be analyzed — link lengths and mechanical constraints of articulated joints. (see Fig. 1 with reference to the system configuration.)

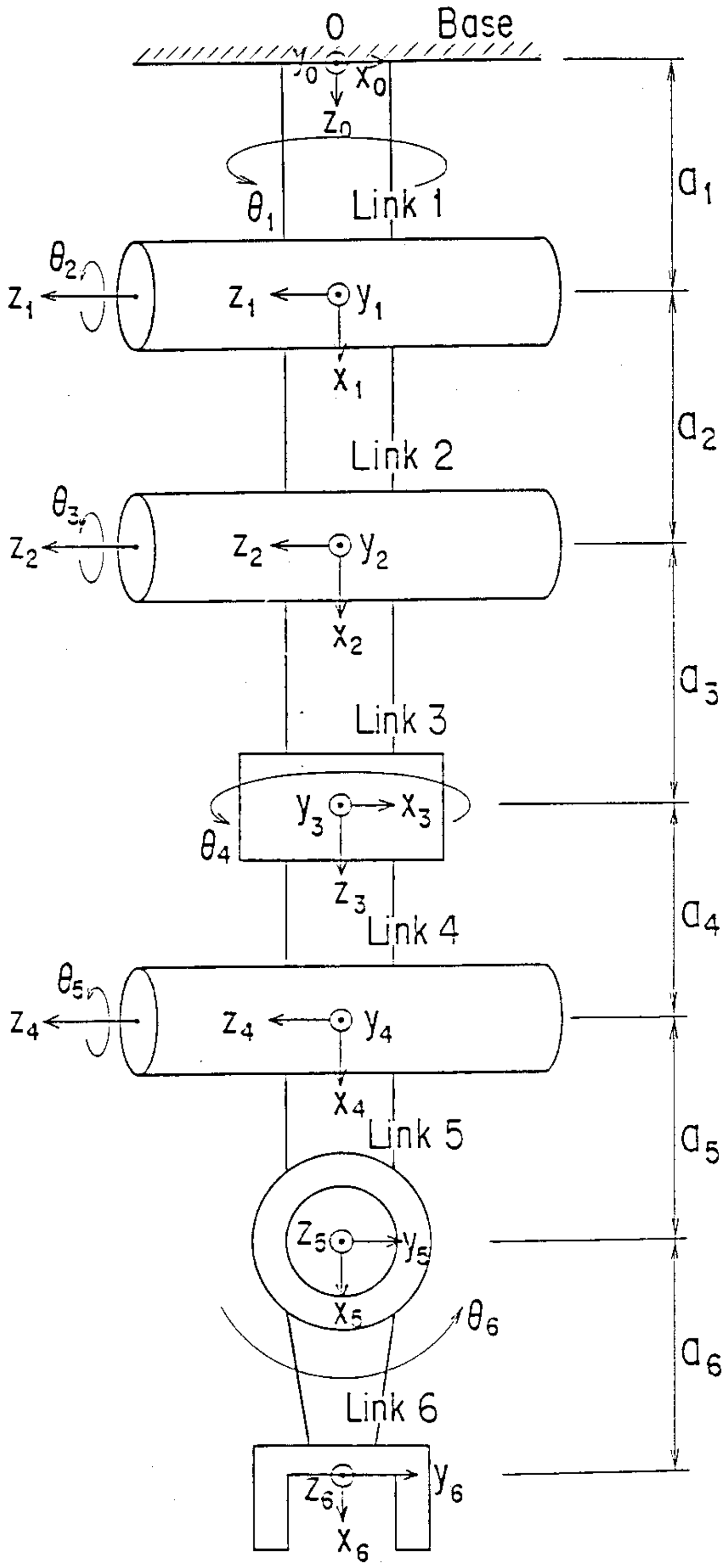
*) DOF = Degree of Freedom

In addition to this, the input data include the motion information of the finger tip describing the location and orientation at the starting point and terminal point. This means the specification of the path in the task space. Concerning the determination of solutions, the code requires two kinds of convergence criteria. One is related to the determination of a quadratic factor $x^2 + px + q$ from a polynomial of the n-th order. The other is used to check the exactness of the calculated articular angles.
Based on such input data, the program computes the individual joint angles for each position advancement on the trajectory prescribed.

The main objective of this paper is oriented to the ARM1 code user: with the intent of this, equations are merely described rather than derived. The major features of the code, programming details and user information associated with input data preparation are provided herein.
They are self-contained for execution of the computation.
A full description of analytical model is given elsewhere.[(1)*]

---

*) Number in bracket designates reference.



a_1 = 701 (mm)

a_2 = 500 (mm)

$a_3 + a_4$ = 350 (mm)

a_5 = 150 (mm)

a_6 = 280 (mm)

θ_1 ; -200 to 200 (deg)

θ_2 ; -30 to 120 (deg)

θ_3 ; -90 to 90 (deg)

θ_4 ; -180 to 180 (deg)

θ_5 ; -90 to 90 (deg)

θ_6 ; -60 to 60 (deg)

Fig.1 Manipulator and Link Coordinate Systems

2. Calculational Method of the Joint Variables

Brief descriptions of a new approach to obtain the joint solutions are herein presented to facilitate the program user.

Referring to Fig.1, the location and orientation at the end-point of the manipulator with respect to the base coordinate is represented as the components of T_6 matrix.

$$T_6 = A_1 A_2 A_3 A_4 A_5 A_6$$

$$= \begin{pmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where

$$A_1 = Rot(z_0, \theta_1) Trans(0,0,a_1) Rot(y_0, -\tfrac{\pi}{2})$$

$$A_2 = Rot(z_1, \theta_2) Trans(a_2,0,0)$$

$$A_3 = Rot(Z_2, \theta_3) Trans(a_3,0,0) Rot(y_2, \tfrac{\pi}{2})$$

$$A_4 = Rot(z_3, \theta_4) Trans(0,0,a_4) Rot(y_3, -\tfrac{\pi}{2})$$

$$A_5 = Rot(z_4, \theta_5) Trans(a_5,0,0) Rot(x_4, -\tfrac{\pi}{2})$$

$$A_6 = Rot(z_5, \theta_6) Trans(a_6,0,0)$$

$$t_{11} = -s_1(s_{23}c_5c_6 + c_{23}s_5c_4c_6 + c_{23}s_4s_6) + c_1(c_4s_6 - c_6s_4s_5)$$

$$t_{12} = s_1(s_{23}s_6c_5 + c_{23}s_5s_6c_4 - c_{23}s_4c_6) + c_1(c_4c_6 + s_4s_5s_6)$$

$$t_{13} = s_1(s_{23}s_5 - c_{23}c_4c_5) - c_1s_4c_5$$

$$\begin{aligned} t_{14} = & -s_1[a_6(s_{23}c_5c_6 + c_{23}s_5c_4c_6 + c_{23}s_4s_6) \\ & + a_5(s_{23}c_5 + c_{23}s_5c_4) + (a_3+a_4)s_{23} + a_2s_2] \\ & + c_1[a_6(c_4s_6 - s_4s_5c_6) - a_5s_4s_5] \end{aligned}$$

$$t_{21} = s_1(c_4s_6 - s_4s_5c_6) + c_1(s_{23}c_5c_6 + c_{23}c_4c_6s_5 + c_{23}s_4s_6)$$

$$t_{22} = s_1(c_4c_6 + s_4s_5s_6) - c_1(s_{23}c_5s_6 + c_{23}s_5s_6c_4 - c_{23}s_4c_6)$$

$$t_{23} = -s_1s_4c_5-c_1(s_{23}s_5-c_{23}c_4c_5)$$

$$t_{24} = s_1[a_6(c_4s_6-s_4s_5c_6)-a_5s_4s_5]+c_1[a_6(s_{23}c_5c_6 +c_{23}s_5c_4c_6+c_{23}s_4s_6)+a_5(s_{23}c_5+c_{23}s_5c_4) +(a_3+a_4)s_{23}+a_2s_2]$$

$$t_{31} = c_5c_6c_{23}-c_4c_6s_5s_{23}-s_4s_6s_{23}$$

$$t_{32} = -c_5s_6c_{23}+s_5s_6c_4s_{23}-s_4c_6s_{23}$$

$$t_{33} = -s_5c_{23}-c_4c_5s_{23}$$

$$t_{34} = a_6(c_5c_6c_{23}-s_5c_6c_4s_{23}-s_4s_6s_{23})+a_5(c_5c_{23}-s_5c_4s_{23}) +(a_3+a_4)c_{23}+a_2c_2+a_1$$

where $c_i = \cos\theta_i$; $s_i = \sin\theta_i$; $c_{ij} = \cos(\theta_i + \theta_j)$;
$s_{ij} = \sin(\theta_i + \theta_j)$

After components $t_{11} \sim t_{34}$ are represented as a trigonometric function of only c_6 and s_6, we define : $\tan\frac{\theta_6}{2} = t$.

Then $c_6 = \dfrac{1 - t^2}{1 + t^2}$, $s_6 = \dfrac{2t}{1 + t^2}$ and $\tan\theta_6 = \dfrac{2t}{1 - t^2}$.

Using this formula, the above kinematic relationships are tranformed into the following polynomial

$$\sum_{i=0}^{24} r_i t^i = 0 .$$

In other words, the problem of finding the articular angles of a manipulator is reduced to a non-linear algebraic equation with reference to $\tan(\theta_6/2)$.

In order to compute roots of this equation as exactly as possible, we used the Bairstow's method (2),(3).

Based on real root of polynomial, individual joint solutions are determined in the following.

(1) Calculation of θ_6

Once the desired solutions t are found from the algebraic equation , a joint angle θ_6 can be easily calculated.

That is, $\tan\frac{\theta_6}{2} = t$

thus, we have

$$\theta_6 = 2\tan^{-1} t\ .$$

Corresponding trigonometric function is:

$$s_6 = \sin\theta_6$$

$$c_6 = \cos\theta_6\ .$$

(2) Calculation of θ_1

Let

$$X_1 = XX + a_5 A$$

and $Y_1 = a_5 C - YY$,

where

$$XX = p_x - a_6 n_x$$

$$YY = p_y - a_5 n_y$$

From the relation $X_1 c_1 = Y_1 s_1$,

Thus, we obtain $\theta_1 = \tan^{-1}\left(\frac{X_1}{Y_1}\right)$

$$s_1 = \sin\theta_1$$

$$c_1 = \cos\theta_1$$

(3) Calculation of θ_{23}

$$\theta_{23} = \tan^{-1}\left(\frac{\pm(\psi^2+\eta^2-a)/(-2a_{34}\sqrt{\psi^2+\eta^2})}{\sqrt{1-\{(\psi^2+\eta^2-a)/(-2a_{34}\sqrt{\psi^2+\eta^2})\}^2}}\right) - \tan^{-1}\left(\frac{\psi}{\eta}\right)$$

where $a = a_2{}^2 - a_{34}{}^2$

$$\psi = a_5 c_6 (n_z - o_z \tan\theta_5) - zz$$

$$\eta = a_5 c_6\{n_y c_1 - n_x s_1 + (s_1 o_x - o_y c_1)\tan\theta_6\}$$

$$- \{-p_x s_1 + p_y c_1 + a_5(n_x s_1 - n_y c_1)\}$$

and $\quad zz = p_z - a_6 n_z - a_1.$

$$s_{23} = \sin\theta_{23}$$

$$c_{23} = \cos\theta_{23}$$

(4) <u>Calculation of θ_4</u>

$$\theta_4 = \tan^{-1}\left(\frac{s_4}{c_4}\right).$$

where $\quad c_4 = B\,c_1 + Ds_1$

$$s_4 = (-B\,s_1 + D\,c_1)/c_{23} \quad (c_{23} \fallingdotseq 0)$$

$$s_4 = -F/s_{23} \quad (c_{23} = 0)$$

$$B = n_x s_6 + o_x c_6$$

$$D = n_y s_6 + o_y c_6$$

$$F = o_z c_6 + n_z s_6$$

(5) <u>Calculation of θ_5</u>

$$\theta_5 = \tan^{-1}\left(\frac{s_5}{c_5}\right)$$

where $\quad s_5 = (Ac_1 - Cs_1)/s_4$

$$= \{c_1(-n_x c_5 + o_x s_6) - s_1(n_y c_5 - o_y s_6)\}/s_4$$

$$(s_4 \fallingdotseq 0)$$

$$s_5 = \{c_{23}(A\,s_1 + C\,c_1) - E\,s_{23}\}/c_4 \;. \quad (s_4 = 0)$$

$$c_5 = (A\,s_1 + C\,c_1)s_{23} + E\,c_{23}\;.$$

$$A = -n_x c_6 + o_x s_6$$

$$C = n_y c_6 - o_y s_6$$

$$E = n_z c_6 - o_z s_6$$

(6) Calculation of θ_2

$$c_2 = \{p_z - a_6 n_z - a_1 - a_5(n_z c_5 - o_z s_5) - a_{34} c_{23}\}/a_2$$

$$s_2 = \{-p_x s_1 + p_y c_1 + a_5(n_x s_1 - n_y c_1) - a_5(n_y c_1 - n_x s_1) c_6 - a_5(s_1 o_x - o_y c_1) s_6 - a_{34} s_{23}\}/a_2$$

Therefore, we have

$$\theta_2 = \tan^{-1} \frac{s_2}{c_2} .$$

(7) Calculation of θ_3

From the calculation of θ_{23} and θ_2, we can get

$$\theta_3 = \theta_{23} - \theta_2 .$$

3. Computer Program Outline

In this section , the overall summary of the current code is described.

(1) Program name

The program name is ARM1/mod0.

(2) Program language

The programming language is FORTRAN 77. Source deck is made for double precision and requires approximately 140 KB core storage, 2 intermediate areas of disk storage, and printer output.

(3) Computers

The code is principally designed for the FACOM-380 computer(with EBCDIC code). By an appropriate conversion routine, however, the present code is available for the other machine with ASCII code.

(4) System architecture

The system organization of the code consists of two major subprograms : one is the calculation routines, and the other is its graphic processing software.

(5) Function and range of application

The code has the capability of solving the inverse kinematics of 6 DOF robot manipulator (without mechanical redundancy and offset).

Since time dependence does not directly appear in the solution process of this approach, it is to be noted that the dynamical analysis for the manipulator motion is not handled in the ARM1.

(6) Numerical procedure for solutions

In the present algorithm, the kinematic relations from the base to end-point of manipulator are represented by a high order algebraic equation with real coefficients. Thus, all complex roots appear in the form of conjugate pairs. Each such pair corresponds to a quadratic factor $x^2 + px + q$ with real coefficients p and q.
Bairstow's method consists in finding numerically such quadratic equations, whose coefficients are determined throughout the iterative calculation of increments of Δp and Δq until the convergence condition given by input data is fulfilled.
Based on these roots of the quadric equations, each joint angle solution is obtained straightforwardly. From the nature of periodicity of the trigonometric function, every possibility for solutions is taken into consideration within the range of its mechanical constraints imposed for each joint angle.
In the computation of arctan(x), the principal value is defined as : $-\pi < \theta \leqq \pi$

For instance,
the value of $\theta = \tan^{-1}(x_1/y_1)$ is :
if $x_1 = 0$ and $y_1 < 0$, then $\theta = \pi$ (rad)
if $x_1 = 0$ and $y_1 > 0$, then $\theta = 0$ (rad)

Concerning the accuracy of solutions, the previous test results were of the order of less than 10^{-8} on the average.

(7) Program size

Based on the application of the variable dimension method, the extension of the program size is allowed resulting in the increase of core memory.

(8) Running time

Although running time is highly dependent on the problem to be solved, it was very short in our previous test cases.

(They are about 0.8 ~1.5 sec.)

(9) Auxiliary programs

Two auxiliary programs GRH51 and GRH52 developed are concerned with graphics information, which can be used to produce plots for polynomial behaviors and individual joint angle profiles, respectively.

These two programs are also operational on the FACOM 380 computer system.

The programs, written in FORTRAN 77, require about 20 KB core storage each.

4. Programming Details

As the embodiment of the algorithm stated in the previous section, in fact, much attention was directed towards the programming techniques necessary to solve the algebraic equation of high order as exactly as possible.

To this end, the decision was made to make use of the Bairstow's methodology instead of a simple Newton-Raphson numerical method. For the equation derived from the kinematic relationships, however, individual real coefficients $a_0, a_1, \ldots, a_{24}$ were ,at large , of an extremely complicated form. On further examination of the characteristics, in some cases, they are too large and in other cases too small depending on the given position and orientation of the manipulator. It is inadvisable to make a direct frontal attack against the equation with such properties. Prior to entering the concrete solution procedures, therefore, the consideration on treatments of these coefficients was required so as to suppress propagation of numerical errors as low as possible.

Now, we first take the absolute values of respective coefficients of decision equation and assume that C be the logarithm of their geometric mean. That is,

$$C = \frac{1}{n} \log_{10} |a_0||a_1| \ldots |a_n|$$

where n is a degree of polynomial f(x).

If $|a_i| = 0$, we set $C = C + 1.0$ (i.e., we use 1.0 for $\log_{10}|a_i|$ in the above formulation)

Using the value of C obtained, each coefficient is defined as:

$$a_i = \frac{a_i}{10^c} \qquad (i = 0, \cdots, n)$$

If the absolute value of new coefficient made through this procedure is less than 10^{-30}, that term is regarded as to be zero, and omitted from the present equation system.

After establishment of the algebraic equation, the next thing we must do is to find a quadratic factor $x^2 + px + q$ according to the Bairstow's method.

Concering the initial values p and q, the programming was specified as follows:

(1) Initialization of p and q

(A) First case (IR = 0)

- if $a_n = 0$, then we assume P = 1 and q = 0.
- if $a_n \neq 0$,

$$\text{then} \begin{cases} \text{(i)} \left|\dfrac{a_{n-2}}{a_n}\right| \geq 0.2, & \begin{cases} p = 0.01 & \left(\text{for } \left|\dfrac{a_{n-1}}{a_{n-2}}\right| < 10^{-5}\right) \\ q = \dfrac{a_n}{a_{n-2}} \end{cases} \\ \qquad\qquad \text{or} & \begin{cases} p = \dfrac{a_{n-1}}{a_{n-2}} & \left(\text{for } \left|\dfrac{a_{n-1}}{a_{n-2}}\right| \geq 10^{-5}\right) \\ q = \dfrac{a_n}{a_{n-2}} \end{cases} \\ \text{(ii)} \left|\dfrac{a_{n-2}}{a_n}\right| < 0.2, & p = 0.5 \text{ and } q = -1.0 \end{cases}$$

(B) Second case (IR = 1)

When the starting values of case (A) are bad and the resulting convergence of solution is not accomplished within iteration numbers required, the second option (B) is used for the initial values p and q.

- if $a_n = 0$, then we assume $p = -1.0$ and $q = 0$
- if $a_n \neq 0$,

$$
\text{then} \begin{cases}
\text{(i)} \left|\dfrac{a_{n-2}}{a_n}\right| \geq 0.2, & \begin{cases} p = -0.01 & \left(\text{for } \left|\dfrac{a_{n-1}}{a_{n-2}}\right| < 10^{-5}\right) \\ q = \dfrac{a_n}{a_{n-2}} \end{cases} \\
 & \begin{cases} p = -\dfrac{a_{n-1}}{a_{n-2}} & \left(\text{for } \left|\dfrac{a_{n-1}}{a_{n-2}}\right| \geq 10^{-5}\right) \\ q = \dfrac{a_n}{a_{n-2}} \end{cases} \\
\text{(ii)} \left|\dfrac{a_{n-2}}{a_n}\right| < 0.2, & p = -0.5 \text{ and } q = -1.0
\end{cases}
$$

Using the above initial conditions, increments $\triangle p$ and $\triangle q$ in the Bairstow's routine are computed by the elimination method. Final determination of them is made under the specified convergence condition. Following the determination of a quadratic factor, the same procedures are repeated for the polynomial with degrees (n-2,n-4,...,4,2).

Now, we should keep in mind that the existence domain of roots t is determined uniquely from the constraints of the angle θ_6 ,because we assume $t = \tan(\theta_6/2)$. Additionally, when the absolute value of imaginary part of root t is less than 10^{-3}, the code processes that root as to be real.

Once roots of the equation are determined in this manner, respective joint solutions may be obtained easily in accordance with representation described in the section 2. The problem is that accuracy. The quickest and best way to verify the validity of their solutions is to substitute the joint angle solutions into the original kinematic expressions and examine its reproducibility.

Thus, we incorporated one of perhaps the most important routines in this code, that is to say, this software is a module to compute the direct kinematic problem and to gennerate all the location in the workspace of the robot within the joint angle limits. The obtained joint solutions are fed into this routine prepared for a calculation of the arm matrix T_6,which should correspond to the T_6 matrix used to determine the joint angle. When the maximum absolute error derived by using the reproduced component values of matrix T_6 is less than some tolerance (EPS1 specified by input data),the joint solutions ($\theta_1,..,\theta_6$)are regarded as feasible ones.

Application of the direct kinematic routine will contribute to enhance the reliability of joint angle solutions.

In order to make easier for everyone the use of the ARM1, the program is described in the most simplest form.
All the data to be used are set in the main memory. The user must specify the control parameter on the size of memory, which is prepared in the main program, according as the size of problem to be solved.

The frame work of the current program ARM1 is composed of two parts : the first one is devoted to the calculation of the joint angles, whose program name is referred to as MANU.

On the other hand, the second part is , as the subsidiary for the first program, concerned with the graphical processing of the obtained solutions. The code GRH51 is a plotting program to provide a pictorial information on the solution behavior of a polynomial. Curve plotting is available within the designated range of independent variable θ_6 every position step on the trajectory given.

Another program GRH52 produces plots of each joint angle solved by code MANU. The organization of the entire software system is is presented in Fig.2. As can be seen in this figure, the calculated results are written on the disk with the unit 61 and 62. The former file is prepared for drawing the solution behavior of the polynomial composed of the coefficients computed, while the latter for demonstrating the results of joints solutions.

The graphics output of these data are processed by reading the stored contents of above dump files through the GRH51 and GRH52 from the unit 51 and 52, respectively.
Pertaining to the input or output information, the program description is as follows.

(1) input requirements included

title of problem, position and orientation vector at the end-point of manipulator, position numbers, convergence factor and so on.

(2) output information included

(A) a complete copy of input data

(B) print out of the calculation

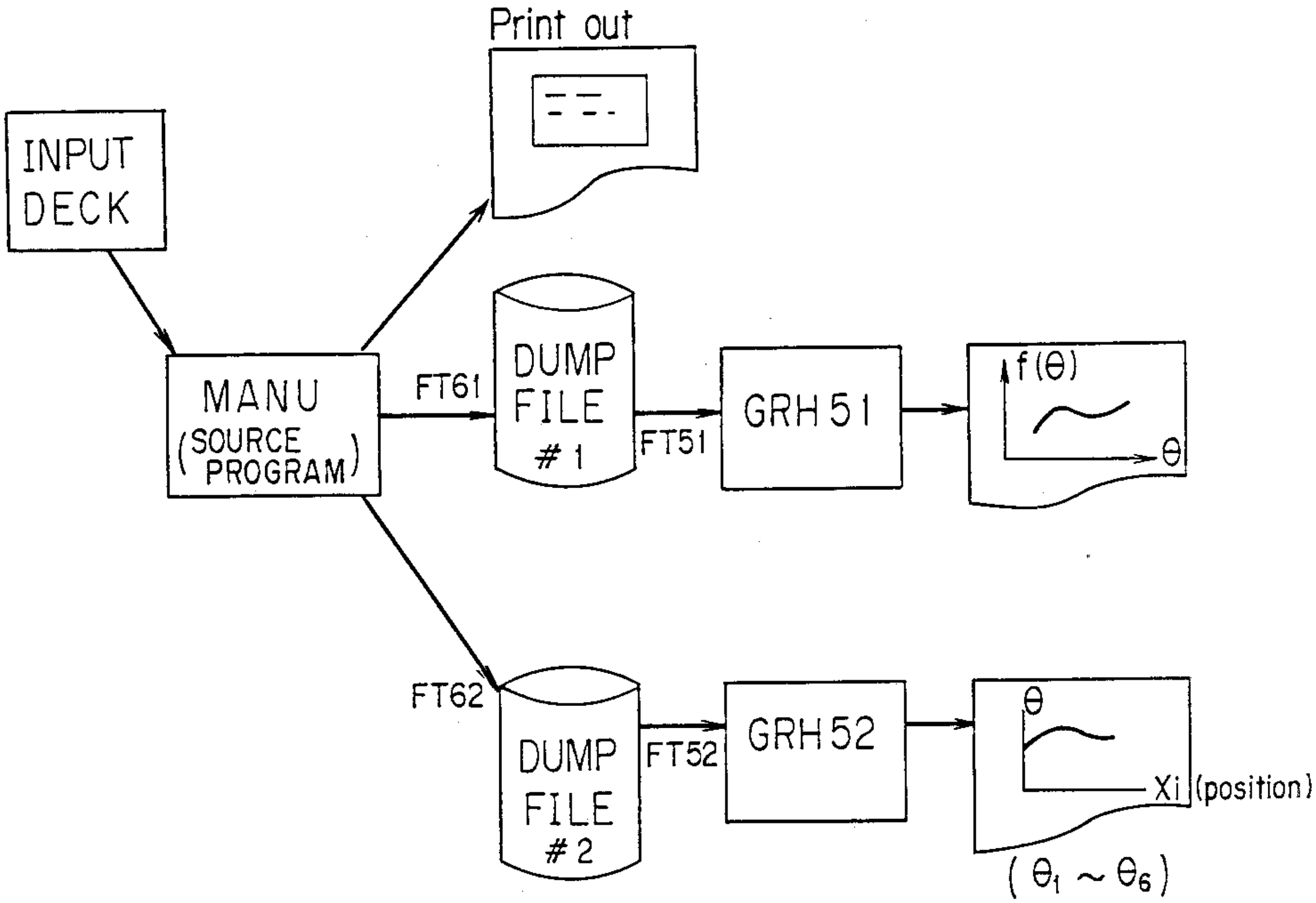


Fig.2 Overall system Layout of Code ARM1

(for each position advancement)

The output of the calculated results consists of a complete edit and a short edit.

The short edit is limited only to the joint solutions. In this edit, the following notations may appear when the calculation is abnormal.

Namely,

-999.D0 :denotes no real roots for the polynomial in question. (All the roots are complex.)

-888.D0 :denotes no desired joint solutions within the mechanical constraints.

999.D0 :denotes no convergence for Δp and Δq in determining the coefficients of the first quadratic factor. It is indicative of no joint angle solutions.

In this third indication, each value is output in the following location.

the latest value of p :is written in the location of θ_3

the latest value of q :is written in the location of θ_4

the latest value of Δp :is written in the location of θ_5

the latest value of Δq :is written in the location of θ_6

Contained in the complete edit is :

(i) real roots of polynomial

(ii) corresponding joint angle solutions

(iii) component values of T_6 matrix

(iv) absolute errors

(C) graphics

(at the end of calculation, the followings are plotted.)

(i) plots of polynomial

(ii) plots of individual joint angles

From above statements, some distinguishing characteristics of the code are summarized below.

(1) For six DOF linkage mechanism (see Fig.1),a non-linear mapping from Cartesian space into joint coordinate space is almost exactly accomplished, given the link data, mechanical constraints data and manipulator motion schemes at the end-point.
In that case, a non-linear algebraic equation transformed from the kinematic relationships is the key factor to determine the joint angle solutions. Derivation of solutions is based on the Bairstow's iterative methods with some tolerance. From the standpoint of the programming techniques, the incorporation of important check routines, double precision description and balancing treatments on the coefficients of the equation are effective to avoid excessive round-off errors and to guarantee the accuracies of joint angle solutions.

(2) On the basis of the introduction of the variable dimension technique, the detailed analysis of the inverse kinematics will be permitted.

(3) The graphics software developed is also useful in the sense presenting plot information related to the pre-

dicted joint angles or polynomial curve.

Finally, a complete description of each subroutine in ARM1 is offered in Appendix 1 together with flow charts, the program tree structures and labelled common table.
Input data definitions of the code and the sample problem are also presented in the Appendix. Included is a complete description of the problem input file that was read by the code.

5. Conclusion

We provided the user information on the code ARM1 such as model explanation, programming details, and input data preparation. The method is justified by another paper. This paper will be contributive to the user in knowing how to handle the current code.

Acknowledgements

Appreciation is expressed to Mr.Y.Shinohara, the chief of the reactor control laboratory for his continuing encouragement and useful suggestions in the developmental work of this code.

References

(1) S.Sasaki : "A Method of Solving the Inverse Kinematics of a Manipulator Arm",JAERI-M 86-018, Feb., 1986

(2) K.Isoda et al., "Handbook of Numerical Calculation", ohmusha,1971

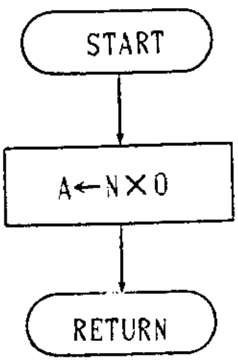
(3) H.Togawa,"Numerical Calculation of Matrix", ohmusha,1973

Appendix 1. A Complete Description of Each Subroutine

Explanatory notes

i n : denotes the input to the module.
i/o : denotes the input and output to the module.
out : denotes the output to the module.
* : denotes the real argument.
% : denotes to be defined in the data statement.
[ ] : denotes the attribute of the variable.
( ) : denotes the declaration of array.
{ } : denotes the type of the variable.

(1) Program MANU

<table>
<tr><td>Module name</td><td>AXYZ</td><td>Module type</td><td>Subroutine</td><td>Module NO.</td><td>1</td></tr>
<tr><td>Function</td><td colspan="5">calculates a cross product of a normal vector n and a sliding vector a.</td></tr>
<tr><td>Calling form</td><td colspan="5">AXYZ(DATA, L)</td></tr>
<tr><td>Call module</td><td colspan="5"></td></tr>
<tr><td>Called module</td><td colspan="5">INPUT0, INPUT1, INPUT2</td></tr>
<tr><td colspan="6">Arguments:
[i/o] DATA {R*8} (3,4,2): vector data relating to the orientation at the end-point of manipulator
[i n] L {I*4} : index of input data (“1” means the initial position and “2” the terminal position.)</td></tr>
<tr><td colspan="6">Flow chart and remarks

</td></tr>
</table>

Module name	BEA	Module type	Subroutine	Module NO.	2
Function	is the Bairstow's iterative method to find an approximation to a quadratic factor of a given polynomial $f(x)= r_0+r_1x+\cdot\cdot\cdot+r_nx^n$ (degf(x)=n, r_i=real).				
Calling form	BEA(N, A, B, C, R, S, DR, DS, LC)				
Call module	DMTX, T6Z				
Called module	DAISU				

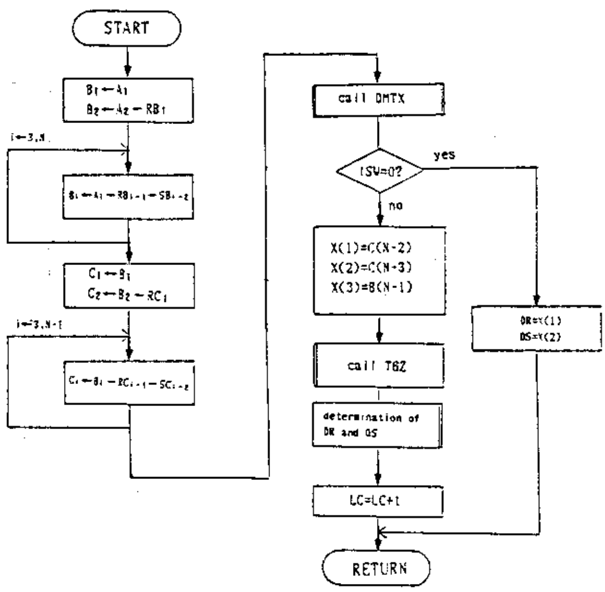
Arguments:

[i n] N {I*4} : the number of coefficients of polynomial
[i n] A {R*8} (1): real coefficients of polynomial

$$(A(i)=r_{N-i}, i=1,N), f(t) = \sum_{i=0}^{N-1} r_i t^i \quad (N=25)$$

[i/o] B {R*8} (1): work area
[i/o] C {R*8} (1): work area
[i n] R {R*8} : coefficient of a quadratic factor $x^2 + Rx + S$
[i n] S {R*8} : coefficient of a quadratic factor $x^2 + Rx + S$
[out] DR {R*8} :an increment of R
[out] DS {R*8} :an increment of S
[i/o] LC {I*4} :the number of iterations

Flow chart and remarks:

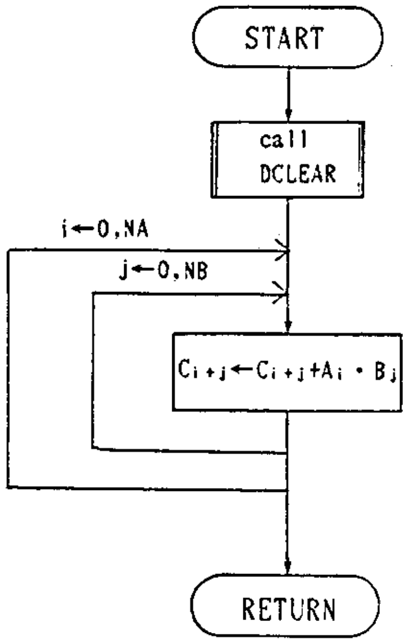


Module name	CFSET	Module type	Subroutine	Module NO.	3
Function	performs the operation $\sum_{i=0}^{NA} A_i \times \sum_{i=0}^{NB} B_i = \sum_{i=0}^{NC} C_i$.				
Calling form	CFSET(A, NA, B, NB, C, NC)				
Call module	DCLEAR				
Called module	T6				

Arguments:

[i n] A {R*8} (0:NA): variable A_i
[i n] NA {I*4} : the number of A_i
[i n] B {R*8} (0:NB): variable B_i
[i n] NB {I*4} : the number of B_i
[out] C* {R*8} (0:NC): variable C_i
[i n] NC {I*4} : the number of C_i

Flow chart and remarks



Module name	CHK	Module type	Subroutine	Module NO.	4
Function	substitutes the obtained angular solutions into the components of T_6 matrix and checks the validity of them.				
Calling form	CHK(T,ST,II,IK,IL,EPS1,ID,IT,NTHETA,ANS,ITHETA,ICNT)				
Call module	OUT				
Called module	THETA				

Arguments:

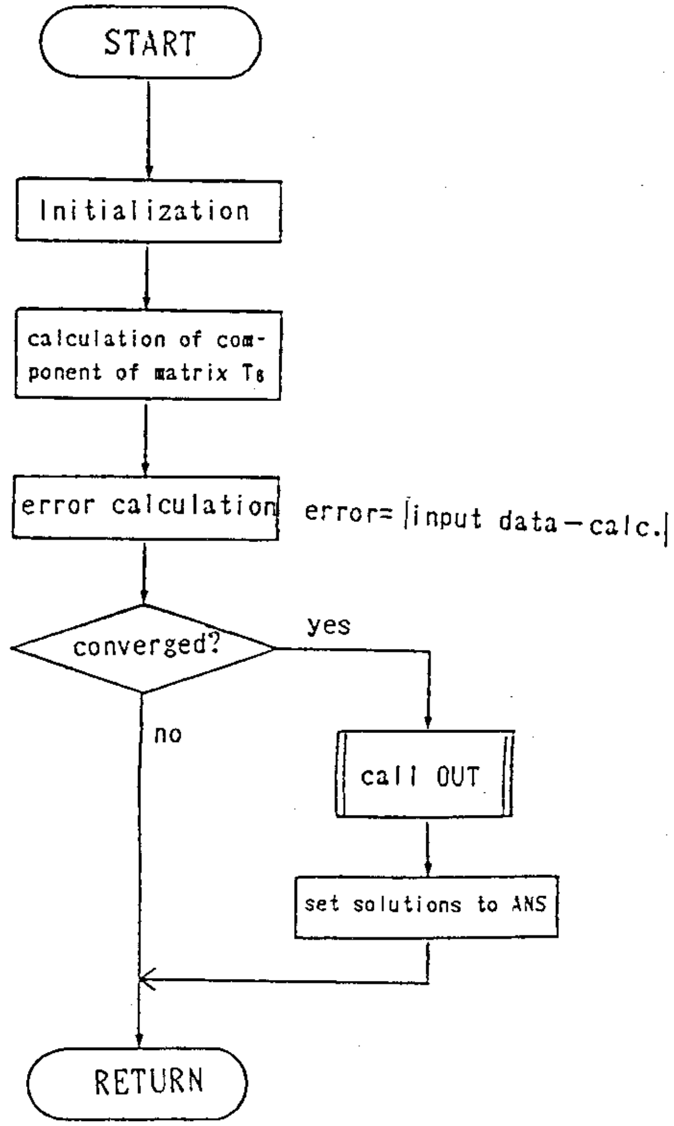
[i/o] T* {R*8} : real roots of a polynomial
[i/o] ST* {R*8} (7,3,3,3,1): solutions of joint angles
[i/o] II* {I*4} : a flag related to the angle θ_1
[i/o] IK* {I*4} : a flag related to the angle $\theta 4$
[i/o] IL* {I*4} : a flag related to the angle $\theta 5$
[i n] EPS1 {R*8} : a convergence factor in an error estimate of joint solutions
[out] ID {I*4} : indication of convergence obtained
[i/o] IT* {I*4} : a sequential position number of input data (0,NDEL)
[i/o] NTHETA {I*4} : a total number of adopted angular solutions set
[out] ANS {R*8} (7,1): the adopted angular solutions
[i/o] ITHETA {I*4} : the number of adopted joint solutions set at each position in the Cartesian space
[i/o] ICNT* {I*4} : count number per each root of polynomial (1,IA)

Flow chart and remarks

When the maximum error related to the components of the arm matrix T_6 is less than EPS1, a set of joint solutions θ_1 to θ_6 is adopted.

【common variable】

```
 COMMON /NA/  A1 [i n] ,A2 [i n] ,
              A3 [i n] ,A4 [i n] ,
              A5 [i n] ,A6 [i n]
COMMON /PAR/ ZN* [i n] ,ZO* [i n] ,
             ZA* [i n] ,ZP* [i n]
COMMON /PAI/  PP [i n]
```


START
Initialization
calculation of component of matrix T8
error calculation
error= |input data-calc.|
converged?
yes
no
call OUT
set solutions to ANS
RETURN

Module name	DAISU	Module type	Subroutine	Module NO.	5
Function	solves a high order algebraic equation using the Bairstow's method.				
Calling form	DAISU(M,A,B,C,X,IA,EPS,R,S,DR,DS)				
Call module	BEA, INDEX, RS250, T6Z				
Called module	MAIN				

Arguments:

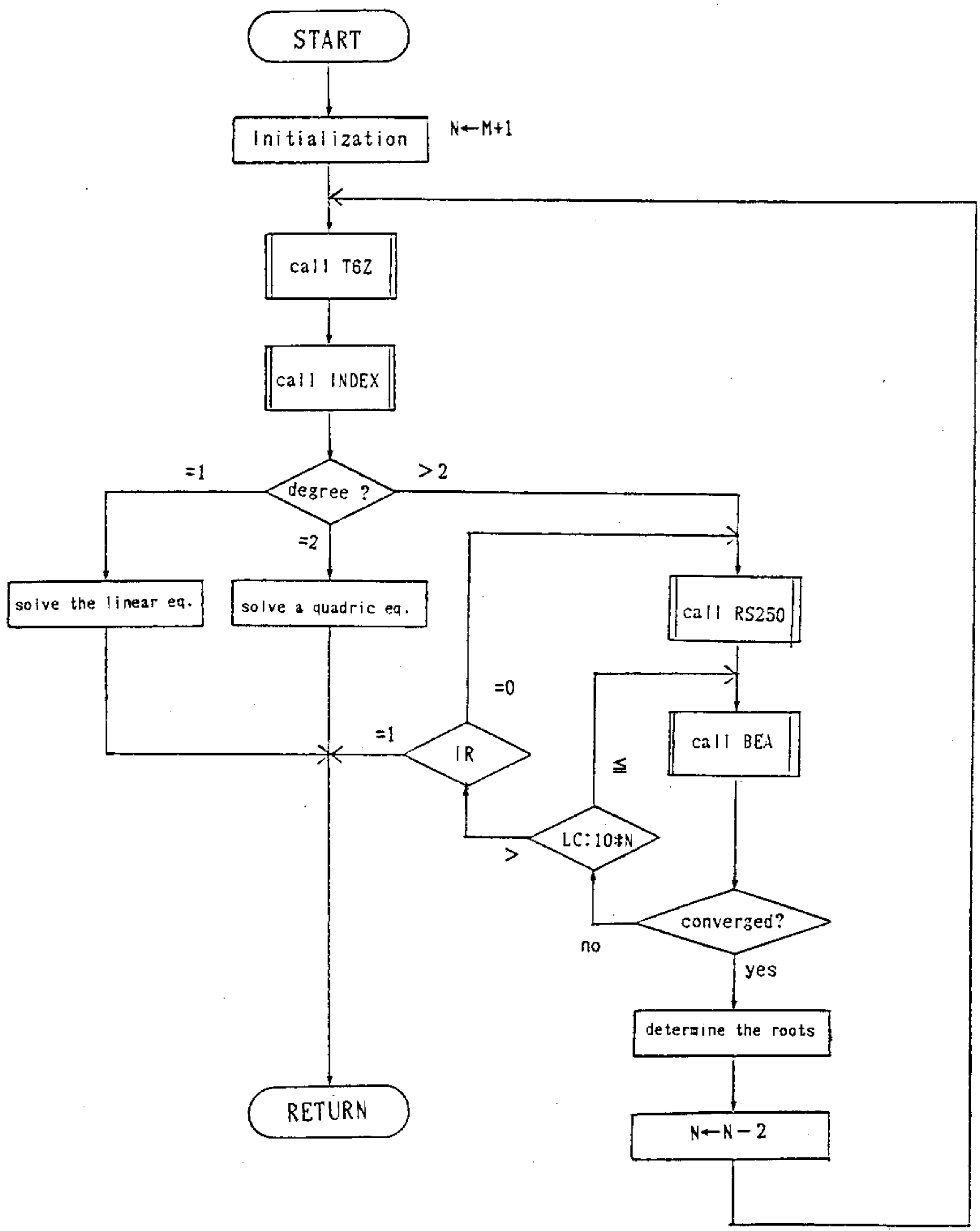
[i/o] M* {I*4} : the degree of polynomial
[i/o] A* {R*8} (1): the coefficient of polynomial
[i/o] B* {R*8} (1): work area
[i/o] C* {R*8} (1): work area
[out] X* {CPX16} (1): all roots of the equation
[i/o] IA* {I*4} : the number of complex roots obtained
[i n] EPS {R*8} : convergence condition with respect to DR and DS in determing a quadratic equation $x^2 + Rx + S$
[i/o] R* {R*8} : coefficient of a quadratic factor $x^2 + Rx + S$
[i/o] S* {R*8} : coefficient of a quadratic factor $x^2 + Rx + S$
[i/o] DR* {R*8} :increment of R
[i/o] DS* {R*8} :increment of S

Flow chart and remarks

The convergence criteria on the R and S is as follows.

If $|DR| < EPS \cdot 10^2$ or $|DS| < EPS \cdot 10^2$, then if $|DR| \leqq EPS$ & $|DS| \leqq EPS$, DR and DS will converge.

If $|DR| \leqq |R| \cdot EPS$ & $|DS| \leqq |S| \cdot EPS$, DR and DS will also converge.

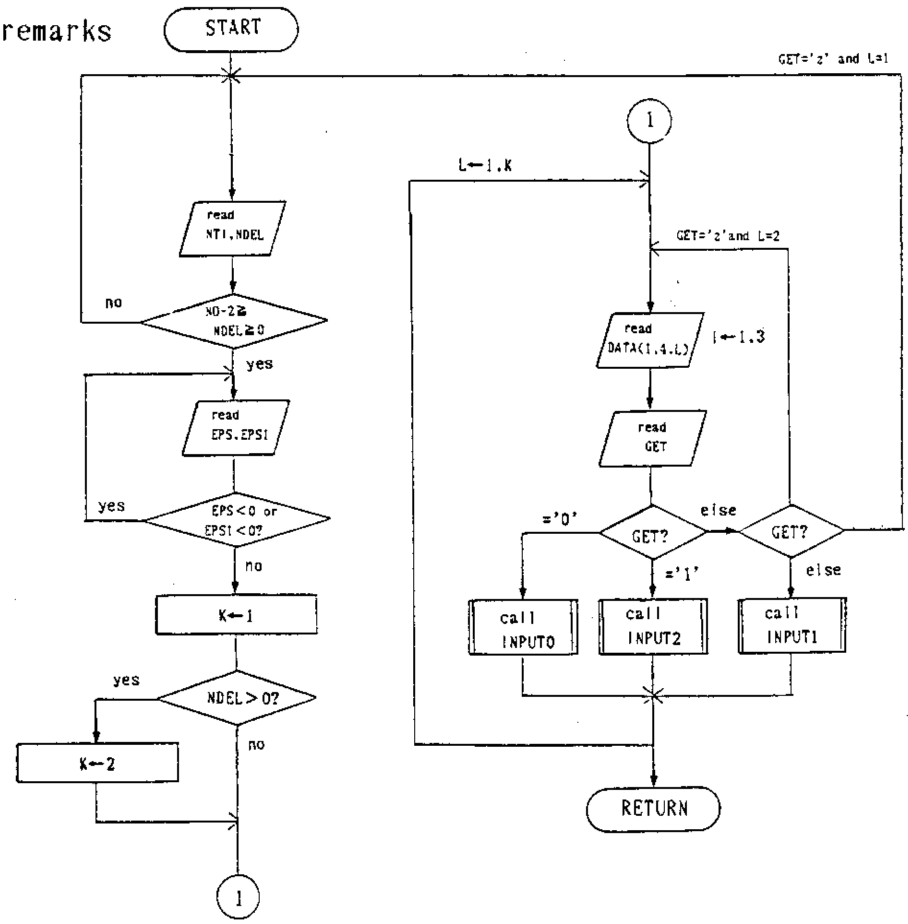

START
Initialization
N←M+1
call T6Z
call INDEX
degree ?
=1
>2
=2
solve the linear eq.
solve a quadric eq.
call RS250
=0
=1
IR
call BEA
≦
LC:10*N
>
converged?
no
yes
determine the roots
RETURN
N←N−2

Module name	DATAIN	Module type	Subroutine	Module NO.	6
Function	is a collection of input data routines.				
Calling form	DATAIN(DATA, NDEL, EPS, EPS1, NTI, NO)				
Call module	INPUT0, INPUT1, INPUT2				
Called module	MAIN				

Arguments:

[i/o] DATA* {R*8} (3,4,2) : vector data as to position and orientation
[i/o] NDEL {I*4} : position numbers between the initial and end point
[i/o] EPS {R*8} : convergence condition with respect to DR and DS
[i/o] EPS1 {R*8} : a convergence factor in an error estimate of joint solutions
[i/o] NTI {C*50} : title of problem
[i n] NO {I*4} : array size of program (variable dimension)

Flow chart and remarks



module name	DCLEAR	Module type	Subroutine	Module NO.	7
Function	sets all array elements to zero.				
Calling form	DCLEAR(W, N)				
Called module	CFSET				

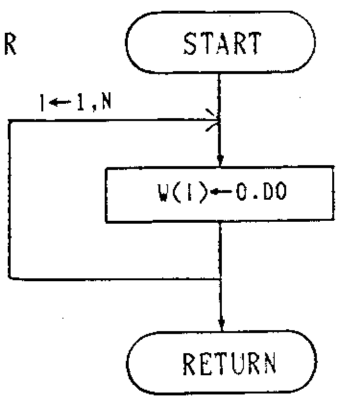
Arguments:
[out] W {R*8} (1): arrays with zero
[i n] N {I*4} : the size of array

module name	DMTX	Module type	Subroutine	Module NO.	8
Function	solves simultaneous equations by using the elimination method.				
Calling form	DMTX(N, A, B, X, ISW)				
Called module	BEA				

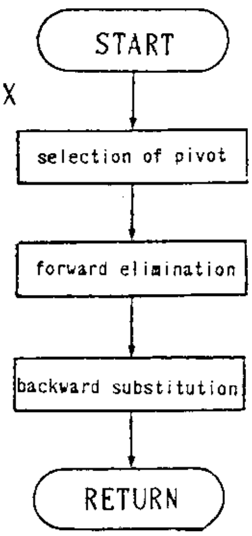
Arguments:
[i n] N {I*4} : order of a square matrix
[i n] A {R*8} (N,N): coefficient matrix
[i n] B {R*8} (N): constant vector (input)
[out] X {R*8} (N): solution vector
[out] ISW {I*4} : a flag related to solutions

Flow chart and remarks

DCLEAR



DMTX



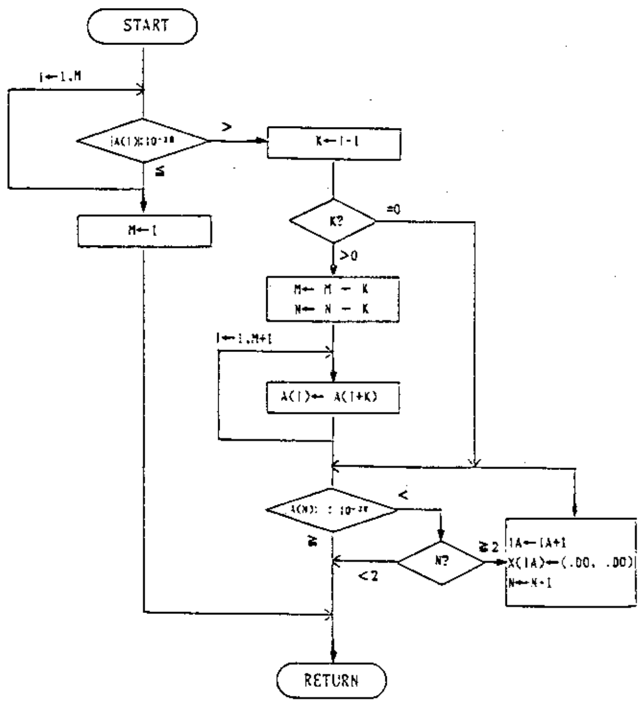
Module name	INDEX	Module type	Subroutine	Module NO.	9
Function	re-arranges the coefficients of a new polynomial with the degree lower by 2 after finding out a quadratic factor				
Calling form	INDEX(M, N, A, X, IA)				
Call module					
Called module	DAISU				

Arguments:

[i/o] M* {I*4} : the degree of a polynomial
[i/o] N* {I*4} : the number of terms in the polynomial (= M+1)
[i/o] A* {R*8} (1): the coefficient of the polynomial
[out] X* {CPX16} (1): all roots of the equation
[i/o] IA* {I*4} : the number of complex roots obtained

Flow chart and remarks

If the absolute value of new coefficient is less than 10^{-30}, that term is regarded as to be zero.



Module name	INPUTO	Module type	Subroutine	Module NO.	10
Function	requires input data as to the orientation at the finger-tip of the manipulator. (user specified option)				
Calling form	INPUTO (DATA, L)				
Call module	AXYZ				
Called module	DATAIN				

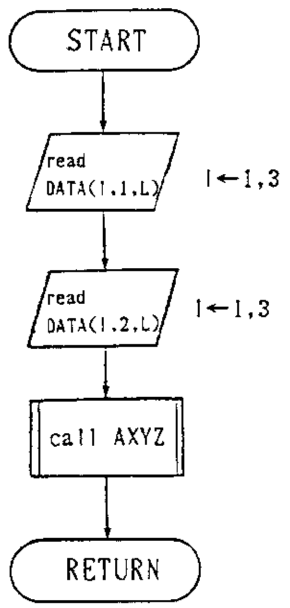
Arguments:

[out] DATA* {R*8} (3,4,2): vector data relating to the orientation at the end-point of manipulator

[i/o] L* {I*4} : index of input data ("1" means the initial position and "2" the terminal position.)

Flow chart and remarks

This module is available when the user selects the input data option GET="0".



Module name	INPUT1	Module type	Subroutine	Module NO.	11
Function	requires its input data when the user wants to specify the orientation of the manipulator with Euler angle.				
Calling form	INPUT1 (DATA, L, W1, W2, W3)				
Call module	AXYZ				
Called module	DATAIN				

Arguments:

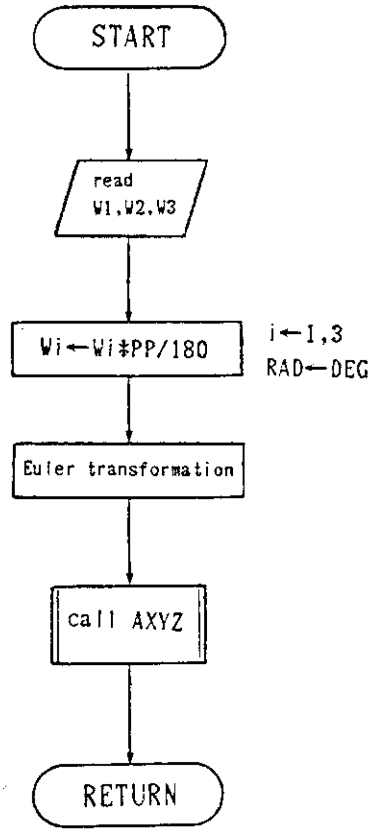
[out] DATA* {R*8} (3,4,2): vector data relating to the orientation at the end-point of manipulator
[i/o] L* {I*4} : index of input data ("1" means the initial position and "2" the terminal position.)
[i/o] W1* {R*8} : a rotation angle around the z-axis ϕ (deg)
[i/o] W2* {R*8} : a rotation angle around the y-axis θ (deg)
[i/o] W3* {R*8} : a rotation angle around the z-axis ψ (deg)

Flow chart and remarks

Euler angle transformation

$$\mathrm{Euler}(\phi, \theta, \psi) = \mathrm{Rot}(z, \phi)\mathrm{Rot}(y, \theta)\mathrm{Rot}(z, \psi)$$

[common variable]
COMMON /PA1/ PP [i n]



Module name	INPUT2	Module type	Subroutine	Module NO.	12
Function	requires its input data when the user wants to specify the orientation of the manipulator with Roll-Pitch-Yaw angle.				
Calling form	INPUT2 (DATA, L, W1, W2, W3)				
Call module	AXYZ				
Called module	DATAIN				

Arguments:

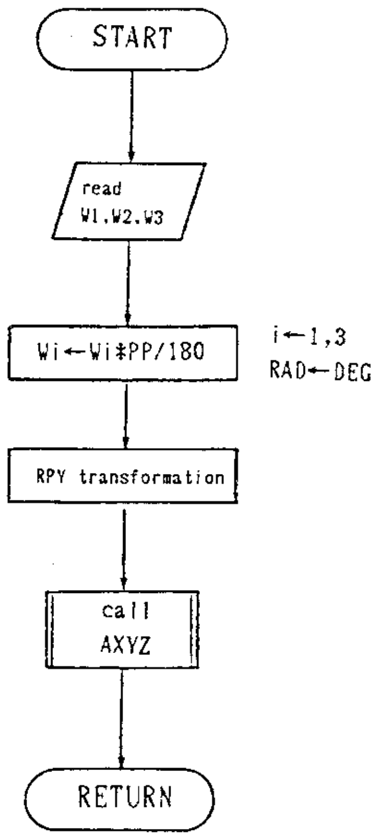
[out] DATA* {R*8} (3,4,2): vector data relating to the orientation at the end-point of manipulator
[i/o] L* {I*4} : index of input data ("1" means the initial position and "2" the terminal position.)
[i/o] W1* {R*8} : a rotation angle around the z-axis ϕ (deg)
[i/o] W2* {R*8} : a rotation angle around the y-axis θ (deg)
[i/o] W3* {R*8} : a rotation angle around the x-axis ψ (deg)

Flow chart and remarks

Roll-Pitch-Yaw angle transformation

$RPY(\phi, \theta, \psi)=Rot(z,\phi)Rot(y,\theta)Rot(x,\psi)$

【common variable】
COMMON /PAI/ PP [i n]



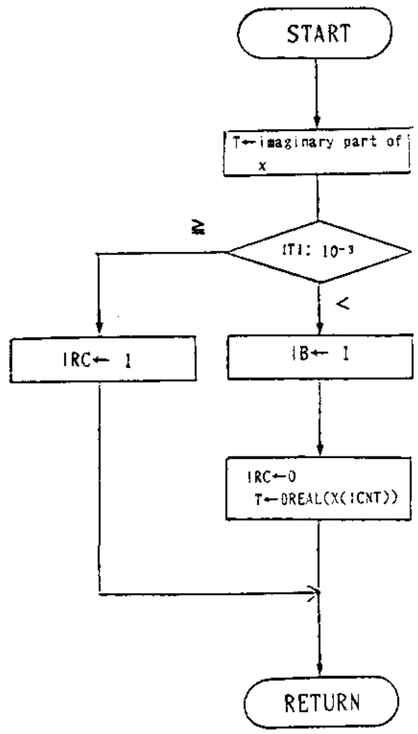
Module name	J0	Module type	Subroutine	Module NO.	13
Function	picks up complex roots in which the imaginary parts are zero or sufficiently near zero.				
Calling form	J0 (X, T, ICNT, IB, IRC)				
Call module					
Called module	MAIN				

Arguments:

[i/o] X* {CPX16} (1): all roots of the equation
[i/o] T* {R*8} : real roots solved
[i n] ICNT* {I*4} : count number per each root of polynomial (1,IA)
[out] IB {I*4} : flag as for existence of real roots
(IB=1:real/ IB=0:complex)
[i/o] IRC* {I*4} : return code (IRC=0 in case of real root)

Flow chart and remarks

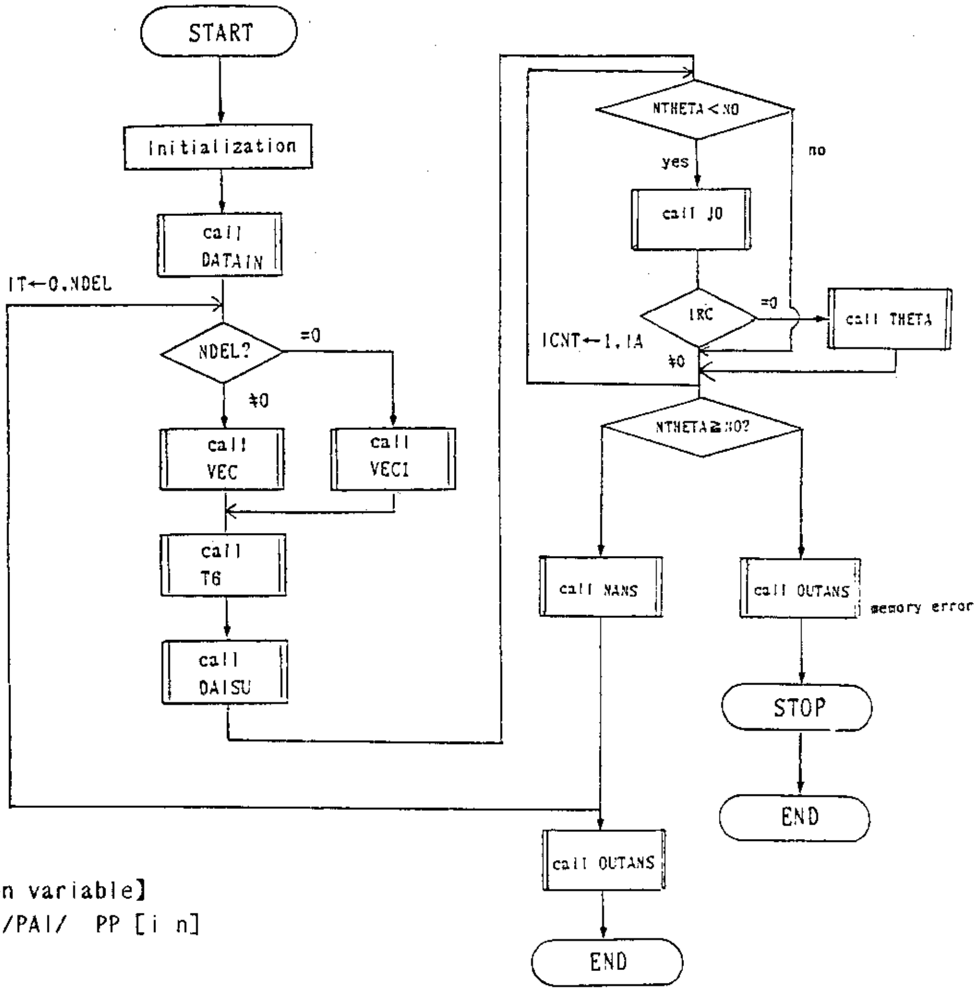
When the absolute value of imaginary part of X is less than 10^{-3}, that root is treated as to be real.



Module name	MAIN	Module type	Subroutine	Module NO.	14
Function	calls in succession individual routines.				
Call module	DAISU, DATAIN, JO, NANS, OUTANS, THETA, T6, VEC, VEC1				

Flow chart and remaks

The program dimension must be specified as the control parameter at the top of the routine together with the number of coefficients of the polynomial to be solved. If a total number of joint solutions sets are greater than NO,then memory error will cause termination of the problem.



【common variable】
COMMON /PAI/ PP [i n]

Module name	NANS	Module type	Subroutine	Module NO.	15
Function	processes the computed results when no desired real roots of the equation were obtained. The calculation is made for next position number.				
Calling form	NANS(ANS,ID,IA,IB,IT,NTHETA,R,S,DR,DS)				
Call module	NOA				
Called module	MAIN				

Arguments:

[out] ANS {R*8} (7,1): adopted joint solutions

[i n] ID {I*4} : code determined in module CHK (If ID=1, then the convergence criteria for a set of joint solutions are satisfied.)

[i n] IA* {I*4} : the number of complex roots obtained

[i n] IB {I*4} : code determined in module JO (IB=1: real, IB=0: complex)

[i n] IT* {I*4} : a sequential position number of input data (0,NDEL)

[i/o] NTHETA {I*4} : a total number of adopted angular solutions set

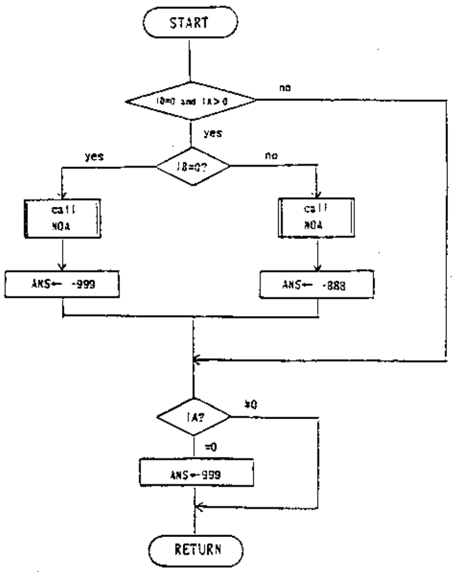
[i n] R {R*8} : coefficient of a quadratic factor $x^2 + Rx + S$

[i n] S {R*8} : coefficient of a quadratic factor $x^2 + Rx + S$

[i n] DR {R*8} :increment of R

[i n] DS {R*8} :increment of S

Flow chart and remarks

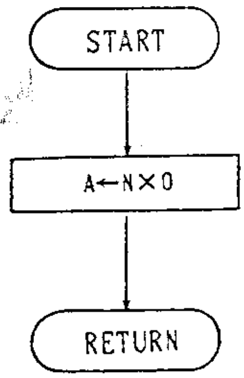


Module name	N	Module type	Subroutine	Module NO.	16
Function	calculates a cross product of vector N and vector O.				
Calling form	N				
Call module					
Called module	VEC				

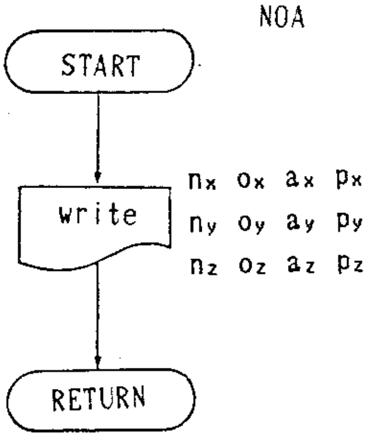
Module name	NOA	Module type	Subroutine	Module NO.	17
Function	performs complete print-out regarding to the position and orientation vector.				
Calling form	N				
Called module	VEC				

Flow chart and remarks

N



NOA



[common variable]
COMMON /PAR/ XY [i/o]

$XY(1,1)=ZN(1)=n_x$
$XY(2,1)=ZN(2)=n_y$
$XY(3,1)=ZN(3)=n_z$
$XY(1,2)=ZO(1)=o_x$
$XY(2,2)=ZO(2)=o_y$
$XY(3,2)=ZO(3)=o_z$
$XY(1,3)=ZA(1)=a_x$
$XY(2,3)=ZA(2)=a_y$
$XY(3,3)=ZA(3)=a_z$
$XY(1,4)=ZP(1)=p_x$
$XY(2,4)=ZP(2)=p_y$
$XY(3,4)=ZP(3)=p_z$

[common variable]
COMMON /PAR/ NX [in] ,NY [in] ,
NZ [in] ,OX [in] ,
OY [in] ,OZ [in] ,
AX [in] ,AY [in] ,
AZ [in] ,PX [in] ,
PY [in] ,PZ [in]

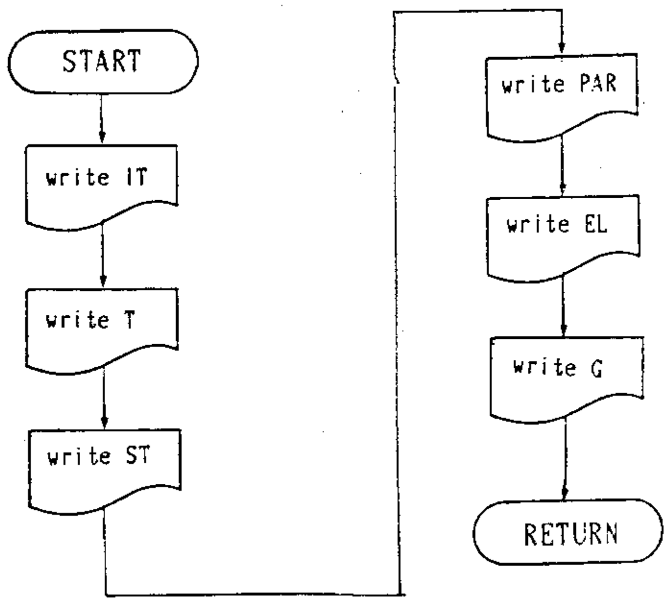
Module name	OUT	Module type	Subroutine	Module NO.	18
Function	provides the important calculated results such as the real roots of polynomial, adopted joint angles set and miscellaneous data.				
Calling form	OUT(T, ST, II, IK, IL, EL, G, IT, ICNT)				
Called module	CHK				

Arguments:

[i n] T {R*8} : real roots of a polynomial
[i n] ST {R*8} (7,3,3,3,1): solutions of joint angles
[i n] II {I*4} : flag related to the angle θ_1
[i n] IK {I*4} : flag related to the angle θ_4
[i n] IL {I*4} : flag related to the angle θ_5
[i n] EL {R*8} (3,4): the computed values of T_6 components by obtained joint angles
[i n] G {R*8} (3,4):difference between the computed values and input position and orientation vector
[i n] IT {I*4} : a sequential position number of input data (0,NDEL)
[i n] ICNT {I*4} : count number per each root of polynomial(1,IA)

Flow chart and remarks

[common variable]
COMMON /PAR/ NX [i n] ,NY [i n] ,
NZ [i n] ,OX [i n] ,
OY [i n] ,OZ [i n] ,
AX [i n] ,AY [i n] ,
AZ [i n] ,PX [i n] ,
PY [i n] ,PZ [i n]
COMMON /PA1/ PP [i n]

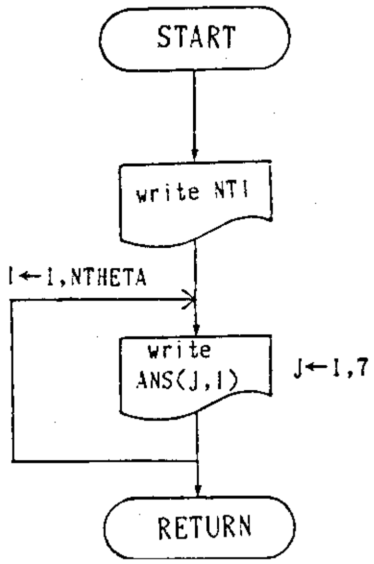


Module name	OUTANS	Module type	Subroutine	Module NO.	19
Function	writes a title and adopted angle solutions on the disk file with a unit 62.				
Calling form	OUTANS(NTI, ANS, NTHETA)				
Call module					
Called module	MAIN				

Arguments:

[i n] NTI {C*50} : title of problem
[i n] ANS* {R*8} (7,1): adopted joint solutions
[i n] NTHETA {I*4} : a total number of adopted angular solutions set

Flow chart and remarks



Module name	INP	Module type	BLOCK DATA	MODULE NO.	20
Function	stores the physical data.				

[common variable]
COMMON /NA/ A% [out]
COMMON /MO/ MO% [out]
COMMON /MS/ MS% [out]

Module name	RS250	Module type	Subroutine	Module NO.	21
Function	sets the initial values of R and S in order to determine coefficients R and S in a quadratic equation .				
Calling form	RS250(N, A, R, S, IR)				
Called module	DAISU				

Arguments:

[i n] N {I*4} : the number of terms in the polynomial (= M+1)
[i n] A {R*8} (1): the coefficient of the polynomial
[out] R {R*8} : coefficient of a quadratic factor x^2 + Rx + S
[out] S {R*8} : coefficient of a quadratic factor x^2 + Rx + S
[i n] IR {I*4} : flag to change the initial condition

Flow chart and memarks

(1) Initialization of p and q

(A) First case (IR = 0)

- if $a_n = 0$, then we assume p = 1 and q = 0.
- if $a_n \neq 0$, then

  (i) $\left|\frac{a_{n-2}}{a_n}\right| \geq 0.2$, $\begin{cases} p = 0.01 \left(\text{for } \left|\frac{a_{n-1}}{a_{n-2}}\right| < 10^{-3}\right) \\ q = \frac{a_n}{a_{n-2}} \end{cases}$ or $\begin{cases} p = \frac{a_{n-1}}{a_{n-2}} \left(\text{for } \left|\frac{a_{n-1}}{a_{n-2}}\right| \geq 10^{-3}\right) \\ q = \frac{a_n}{a_{n-2}} \end{cases}$

  (ii) $\left|\frac{a_{n-2}}{a_n}\right| < 0.2$, p = 0.5 and q = −1.0

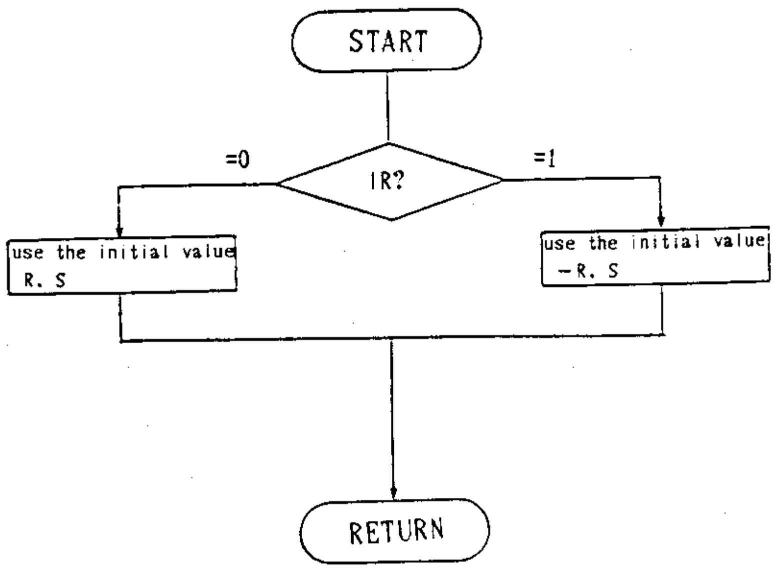
(B) Second case (IR = 1)

- if $a_n = 0$, then we assume p = −1.0 and q = 0
- if $a_n \neq 0$, then

  (i) $\left|\frac{a_{n-2}}{a_n}\right| \geq 0.2$, $\begin{cases} p = -0.01 \left(\text{for } \left|\frac{a_{n-1}}{a_{n-2}}\right| < 10^{-3}\right) \\ q = \frac{a_n}{a_{n-2}} \end{cases}$ $\begin{cases} p = -\frac{a_{n-1}}{a_{n-2}} \left(\text{for } \left|\frac{a_{n-1}}{a_{n-2}}\right| \geq 10^{-3}\right) \\ q = \frac{a_n}{a_{n-2}} \end{cases}$

  (ii) $\left|\frac{a_{n-2}}{a_n}\right| < 0.2$, p = −0.5 and q = −1.0

**The values p and q correspond R and S in flow diagram, respectively.



Module name	SS23	Module type	Subroutine	Module NO.	22
Function	calculates the joint angle θ_{23}.				
Calling form	SS23(ST, II, IS23)				
Called module	THETA				

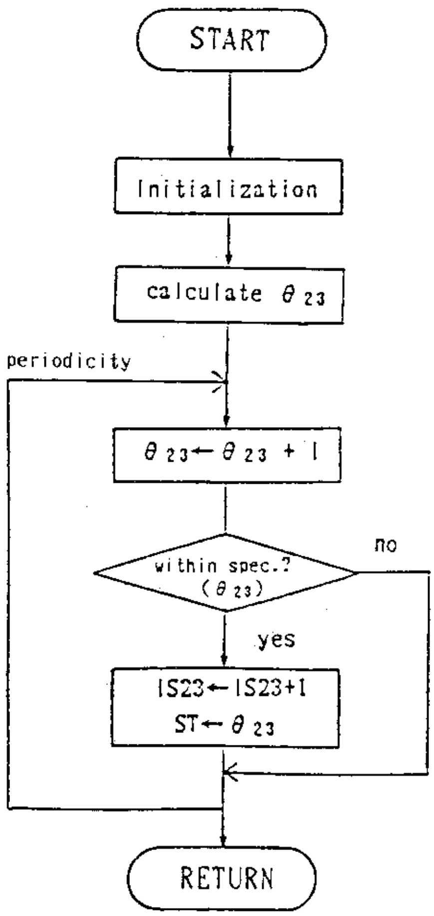
Arguments:

[i/o] ST {R*8} (7,3,3,3,1): adopted solutions of joint angles
[i n] II {I*4} : a flag related to the angle θ_1
[i/o] IS23 {I*4} : the number of the angle θ_{23}-solutions obtained

Flow chart and remarks

【common variable】

```
COMMON /PAR/  NX [i n] ,NY [i n] ,
              NZ [i n] ,OX [i n] ,
              OY [i n] ,OZ [i n] ,
              PX [i n] ,
              PY [i n] ,PZ [i n]
COMMON /PAI/  PP [i n]
COMMON /NA/   A2 [i n] ,A3 [i n] ,
              A4 [i n] ,A5 [i n] ,
              A6 [i n]
COMMON /MO/   MO2 [i n] ,MO3 [i n]
COMMON /MS/   MS2 [i n] ,MS3 [i n]
COMMON /AF/   VZZ [i n]
```



Module name	S1	Module type	Subroutine	Module NO.	23
Function	calculates the joint angle θ_1.				
Calling form	S1(ST, IS1)				
Call module					
Called module	THETA				

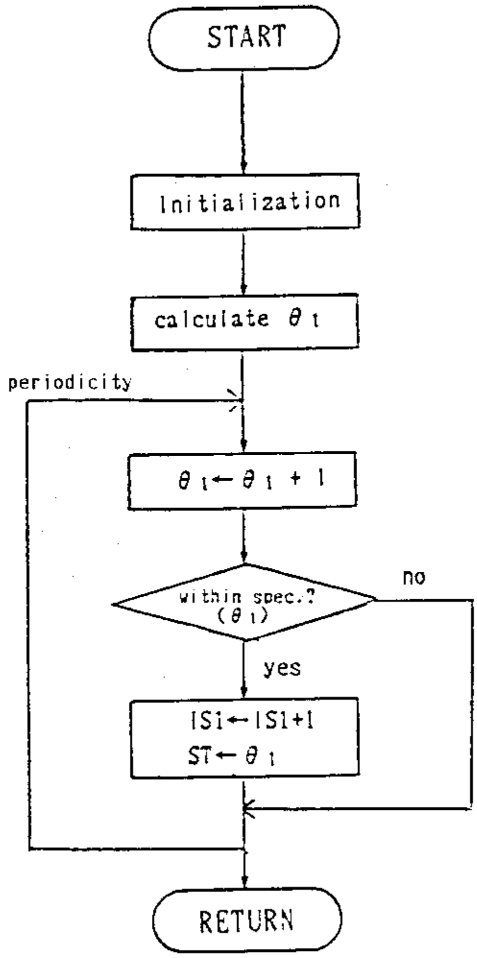
Arguments:

[out] ST {R*8} (7,3,3,3,1): adopted solutions of joint angles
[i/o] IS1 {I*4} : the number of the angle θ_1-solutions obtained

Flow chart and remarks

【common variable】

COMMON /AF/ VA [i n] ,VC [i n] ,
VXX [i n] ,VYY [i n]
COMMON /PAI/ PP [i n]
COMMON /NA/ A5 [i n]
COMMON /MO/ MO1 [i n]
COMMON /MS/ MS1 [i n]



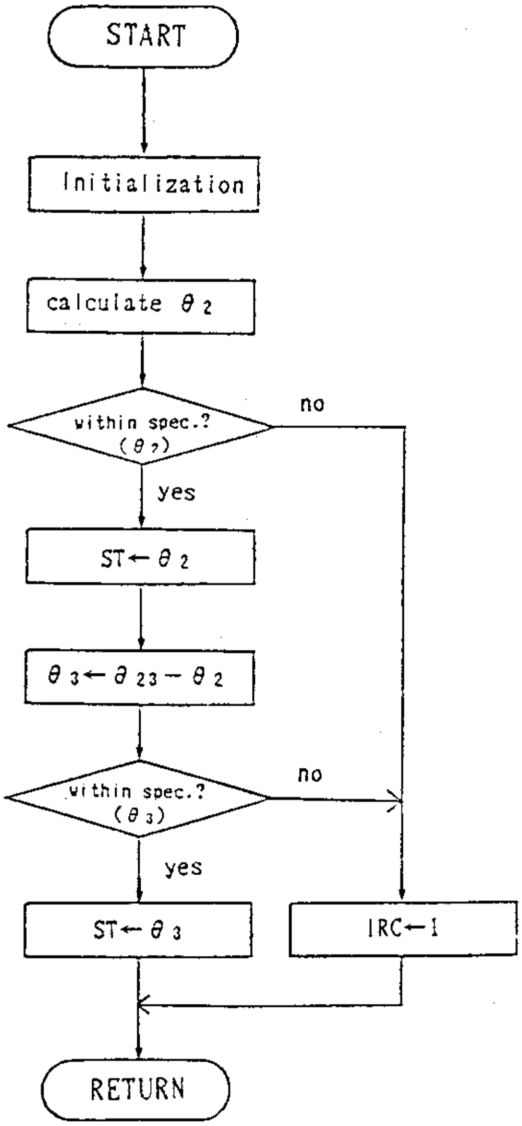
Module name	S2S3	Module type	Subroutine	Module NO.	24
Function	calculates the joint angles θ_2 and θ_3 .				
Calling form	S2S3(ST, II, IJ, IRC)				
Called module	THETA				

Arguments:

[i/o] ST {R*8} (7,3,3,3,1): adopted solutions of joint angles
[i n] II {I*4} : a flag related to the angle θ_1
[i n] IJ {I*4} : a flag related to the angle θ_{23}
[out] IRC {I*4} : return code

Flow chart and remarks

【common variable】
COMMON /PAR/ NX [i n] ,NY [i n] ,
PX [i n] ,PY [i n] ,
COMMON /PAI/ PP [i n]
COMMON /NA/ A2 [i n] ,A3 [i n] ,
A4 [i n] ,A5 [i n] ,
A6 [i n]
COMMON /MO/ MO2 [i n] ,MO3 [i n]
COMMON /MS/ MS2 [i n] ,MS3 [i n]
COMMON /AF/ VA [i n] , VC [i n]
VE [i n] ,VZZ [i n]



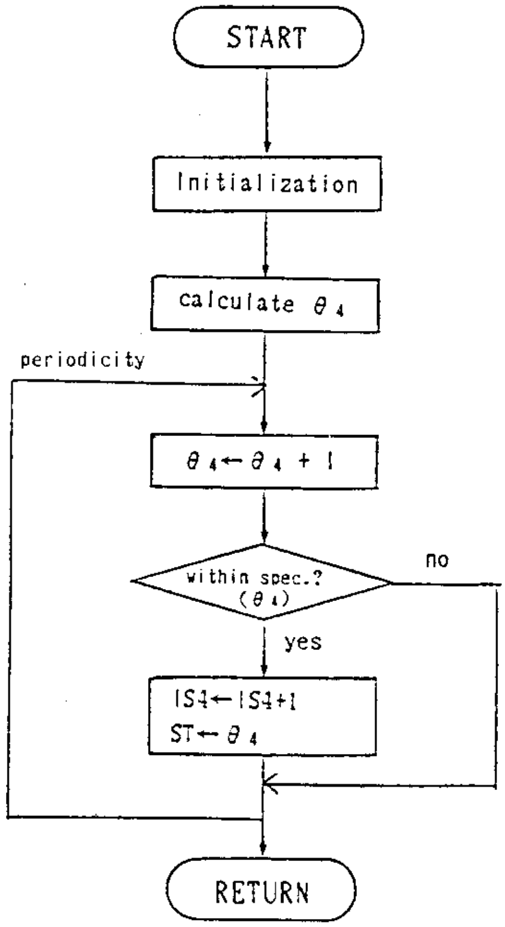
Module name	S4	Module type	Subroutine	Module NO.	25
Function	calculates the joint angle θ_4.				
Calling form	S4(ST, II, IJ, IS4)				
Called module	THETA				

Arguments:

[i/o] ST {R*8} (7,3,3,3,1): adopted solutions of joint angles
[i n] II {I*4} : a flag related to the angle θ_1
[i n] IJ {I*4} : a flag related to the angle θ_{23}
[i/o] IS4 {I*4} : the number of the angle θ_4-solutions obtained

Flow chart and remarks

【common variable】
COMMON /PAI/ PP [i n]
COMMON /MO/ MO4 [i n]
COMMON /MS/ MS4 [i n]
COMMON /AF/ VB [i n] , VD [i n] ,
VF [i n]



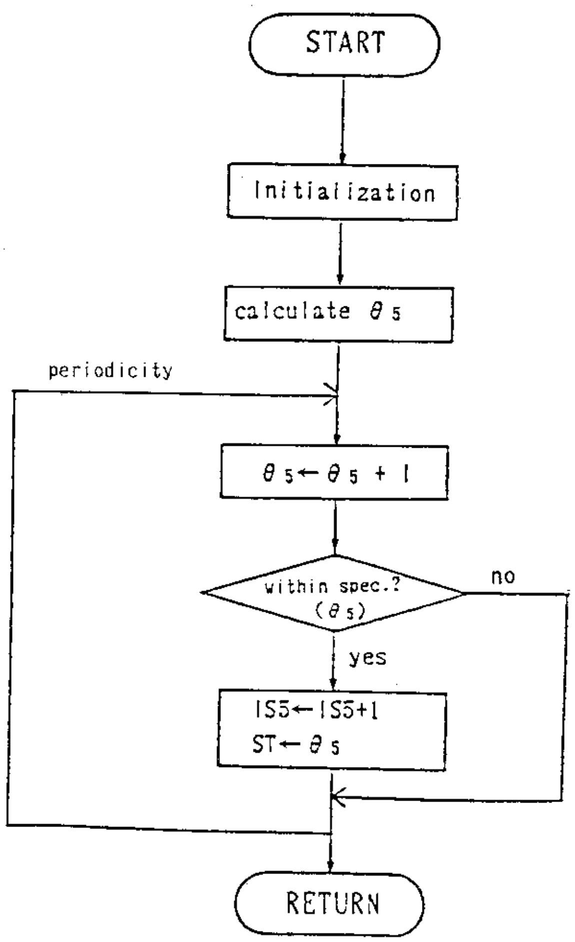
Module name	S5	Module type	Subroutine	Module NO.	26
Function	calculates the joint angle θ_5.				
Calling form	S5(ST, II, IJ, IK, IS5)				
Called module	THETA				

Arguments:

[i/o] ST {R*8} (7,3,3,3,1): adopted solutions of joint angles
[i n] II {I*4} : a flag related to the angle θ_1
[i n] IJ {I*4} : a flag related to the angle θ_{23}
[i n] IK {I*4} : a flag related to the angle θ_4
[i/o] IS5 {I*4} : the number of the angle θ_5-solutions obtained

Flow chart and remarks

[common variable]
COMMON /PAI/ PP [i n]
COMMON /MO/ MO5 [i n]
COMMON /MS/ MS5 [i n]
COMMON /AF/ VA [i n] , VC [i n] ,
VE [i n]



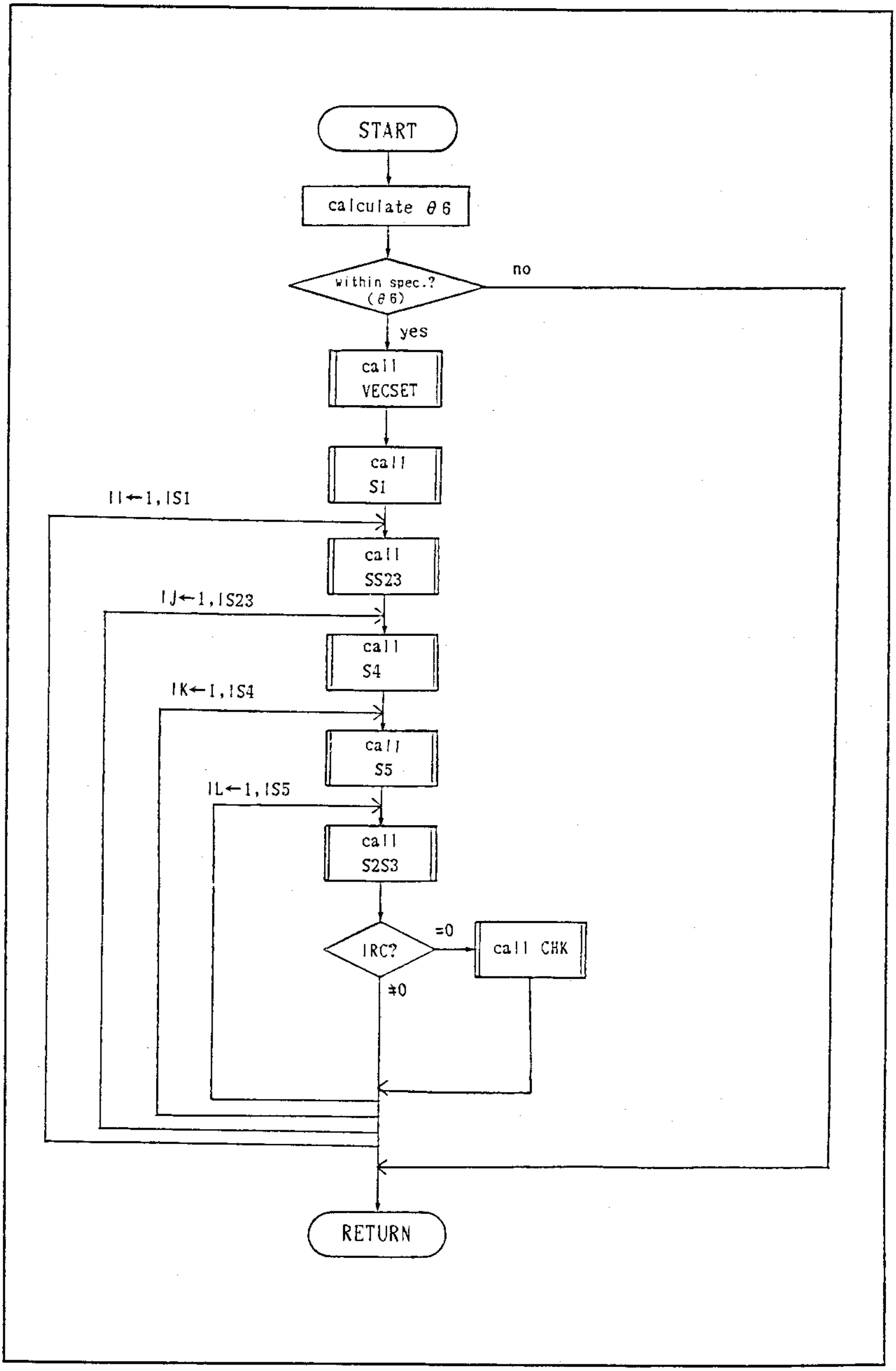
Module name	THETA	Module type	Subroutine	Module NO.	27
Function	calls individual modules to determine rotation angles.				
Calling form	THETA(T,ST,EPS1,ID,IT,NTHETA,ANS,ITHETA,ICNT,NO)				
Call module	CHK, SS23, S1, S2S3, S4, S5, VECSET				
Called module	MAIN				

Arguments:

[i/o] T* {R*8} : real roots of a polynomial
[i/o] ST* {R*8} (7,3,3,3,1): solutions of joint angles
[i/o] EPS1* {R*8} : a convergence factor in an error estimate of joint solutions
[i/o] ID* {I*4} : indication of convergence obtained
[i/o] IT* {I*4} : a sequential position number of input data (0,NDEL)
[i/o] NTHETA* {I*4} : a total number of adopted angular solutions set
[i/o] ANS* {R*8} (7,1): the adopted angular solutions
[i/o] ITHETA* {I*4} : the number of adopted joint solutions set at each position in the Cartesian space
[i/o] ICNT* {I*4} : count number per each root of polynomial (1,IA)
[i n] NO {I*4} : array size of the program (variable dimension)

Flow chart and remarks

[common variable]
COMMON /PAI/ PP [i n]
COMMON /MO/ MO6 [i n]
COMMON /MS/ MS6 [i n]


START
calculate θ6
within spec.?
(θ6)
no
yes
call
VECSET
call
S1
II←1,IS1
call
SS23
IJ←1,IS23
call
S4
IK←1,IS4
call
S5
IL←1,IS5
call
S2S3
IRC?
=0
call CHK
≠0
RETURN

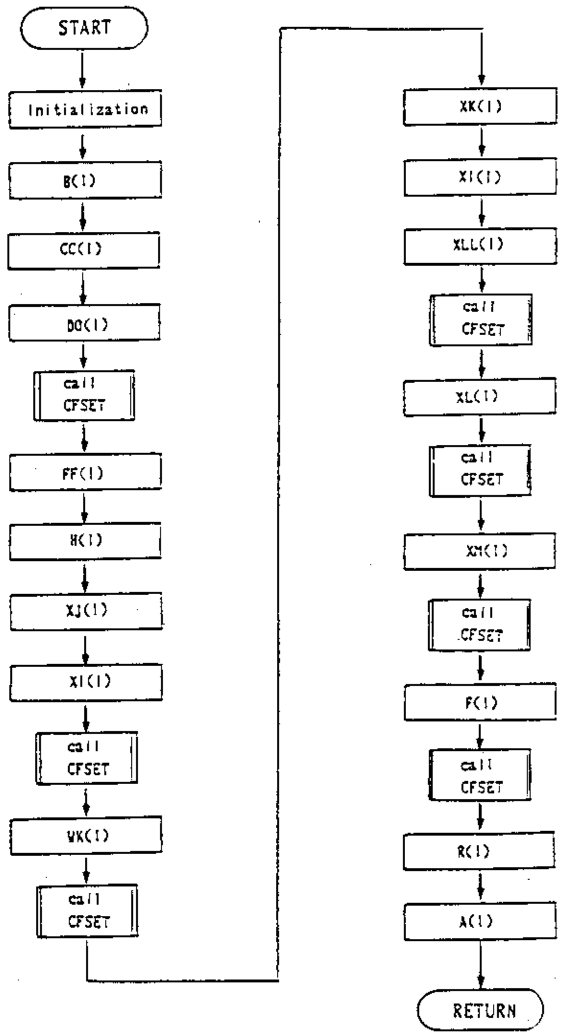
Module name	T6	Module type	Subroutine	Module NO.	28
Function	transforms the kinematic relationships of manipulator represented in T6 matrix into a high order algebraic equation with a single variable.				
Calling form	T6(A)				
Call module	CFSET				
Called module	MAIN				

Arguments:

[out] A {R*8} (25): real coefficients of polynomial

Flow chart and remarks

[common variable]
COMMON /NA/ A1 [i n] ,A2 [i n] ,
A3 [i n] ,A4 [i n] ,
A5 [i n] ,A6 [i n]
COMMON /PAR/ ZN [i n] ,Z0 [i n] ,
ZP [i n]

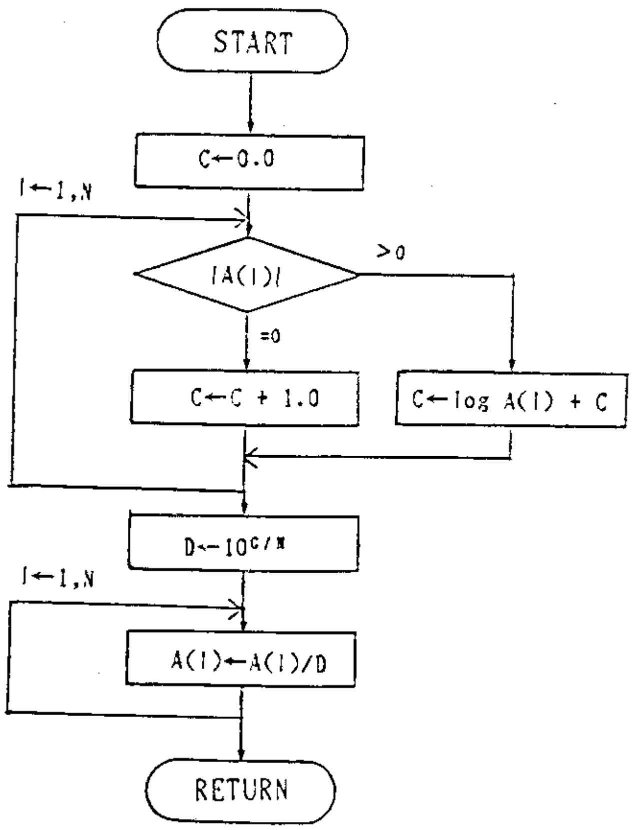


Module name	T6Z	Module type	Subroutine	Module NO.	29
Function	balances the magnitude of coefficients appeared in individual terms in order to prohibit the propagation of numerical errors				
Calling form	T6Z(N ,A)				
Call module					
Called module	DAISU				

Arguments:

[i n] N* {I*4} : the number of terms in a polynomial
[i n] A* {R*8} (1): coefficients of polynomial

Flow chart and remarks



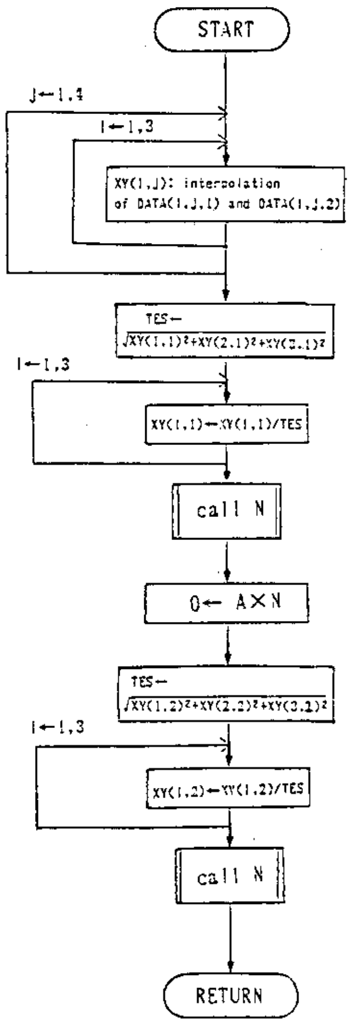
Module name	VEC	Module type	Subroutine	Module NO.	30
Function	makes a linear interpolation calculation with reference to the position and orientation vector.				
Calling form	VEC(DATA, NDEL, IT)				
Call module	N				
Called module	MAIN				

Arguments:

[i n] DATA {R*8} (3,4,2) : vector data as to position and orientation
[i n] NDEL* {I*4} : position numbers between the initial and end point
[i n] IT* {I*4} : a sequential position number of input data (0,NDEL)

Flow chart and remarks

[common variable]
COMMON /PAR/ XY [i/o]



Module name	VECSET	Module type	Subroutine	Module NO.	31
Function	collects some important relationships described as the function of θ_6, which are frequently used in the program.				
Calling form	VECSET(ST)				
Called module	THETA				

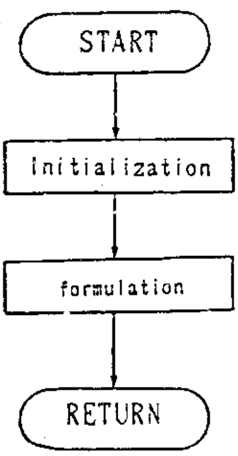
Arguments:
[i n] ST {R*8} (7,3,3,3,1): adopted solutions of joint angles

Module name	VEC1	Module type	Subroutine	Module NO.	32
Function	replaces the vector data with two dimensional array.				
Calling form	VEC1(DATA)				
Called module	MAIN				

Arguments:
[i n] DATA {R*8} (3,4,2): vector data as to position and orientation

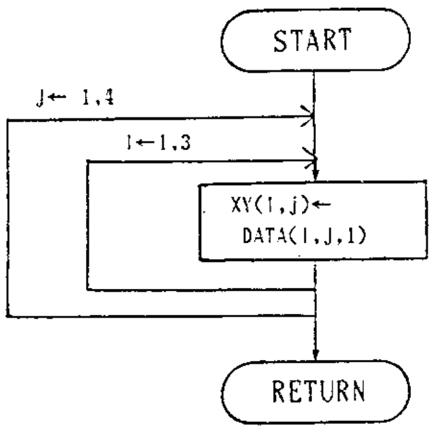
Flow chart and remarks

VECSET



[common variable]
COMMON /PAR/ NX [i n] ,NY [i n] ,
NZ [i n] ,OX [i n] ,
OY [i n] ,OZ [i n] ,
AX [i n] ,AY [i n] ,
AZ [i n] ,PX [i n] ,
PY [i n] ,PZ [i n]
COMMON /NA/ A1 [i n] ,A6 [i n]
COMMON /AF/ VA [out] , VB [out] ,
VC [out] , VD [out] ,
VE [out] , VF [out] ,
VXX [out] ,VYY [out] ,
VZZ [out]

VEC1



[common variable]
COMMON /PAR/ XY [out]

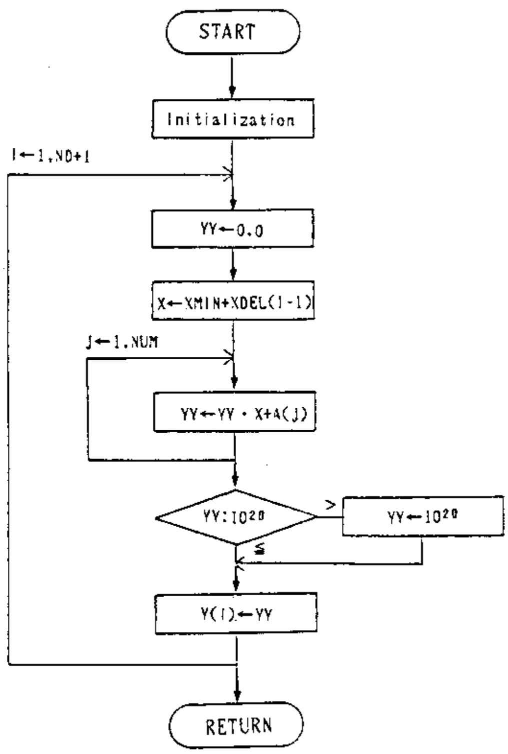
(2) Program GRH51

module name	CLC	Module type	Subroutine	Module NO.	1
Function	makes the calculation of $f(\theta)=\sum_{i=0}^{24} r_i \tan(\theta/2)^i$				
Calling form	CLC(ND, NUM, A, Y, TH)				
Call module					
Called module	MAIN				

Arguments:

[i n] ND* {I*4} : point numbers between upper and lower limits of θ
[i n] NUM {I*4} : the number of terms of a polynomial
[i n] A {R*4} (1): the coefficients of a polynomial
[out] Y {R*4} (1): the calculated result of $f(\theta)$
[i n] TH {R*4} : ±(upper limit of the joint angle θ (deg))

Flow chart and remarks

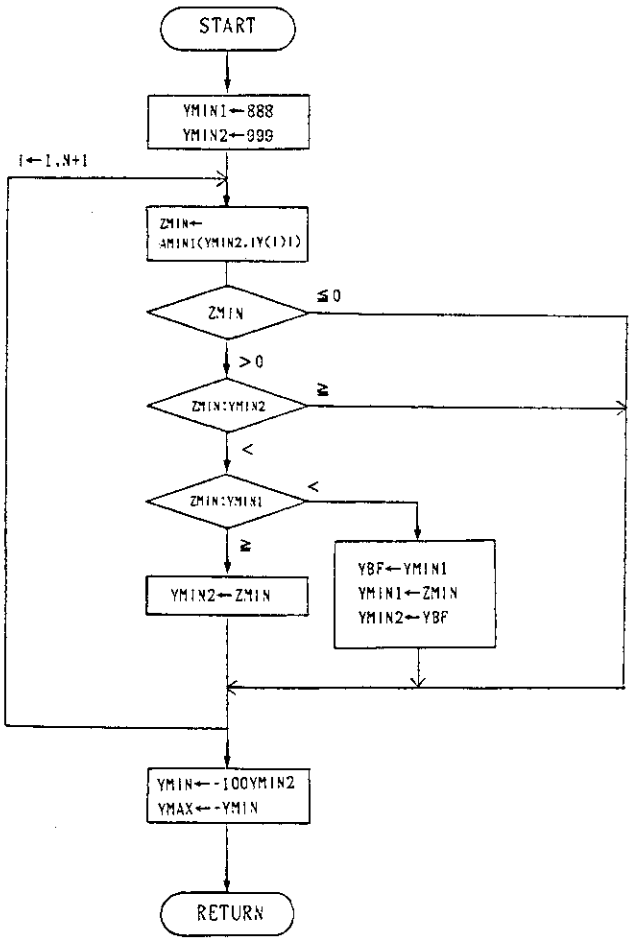


module name	EARY01	Module type	Subroutine	Module NO.	2
Function	calculates YMIN for scaling.				
Calling form	EARY01(N, Y, YMAX, YMIN)				
Call module					
Called module	MAIN				

Arguments:

[i n] N {I*4} : point numbers between upper and lower limits of θ
[i n] Y* {R*4} (1): calculated result of function f(θ)
[out] YMAX {R*4} :-YMIN (absolute of f(θ)$_{min}$)
[i/o] YMIN {R*4} : Y_{min} (-(absolute of f(θ)$_{min}$))

Flow chart and remarks

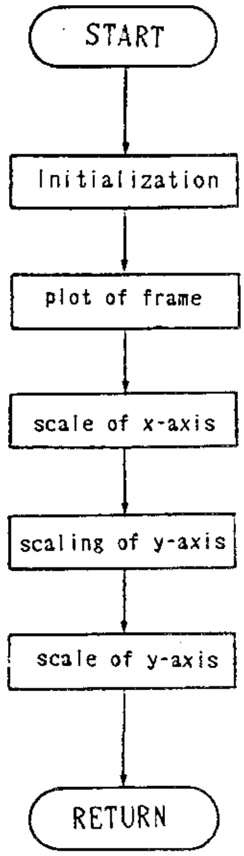


module name	EARY02	Module type	Subroutine	Module NO.	3
Function	draws the frame and label after scaling of Y-axis.				
Calling form	EARY02(GX, GY, TH, YMAX, YMIN, ZMAX, ZMIN, ICNT)				
Call module	NUMBER, PLOT, SYMBOL				
Called module	MAIN				

Arguments:

[i/o] GX* {R*4} : the actual length of x-axis(mm)
[i/o] GY* {R*4} : the actual length of y-axis(mm)
[i n] TH* {R*4} : ±(upper limit of the joint angle θ (deg))
[i n] YMAX {R*4} :the value to be used in scaling
[i n] YMIN {R*4} :the value to be used in scaling
[i/o] ZMAX {R*4} :scaled Y-axis value(max)
[i/o] ZMIN {R*4} :scaled Y-axis value(min)
[i n] ICNT {I*4} :position number

Flow chart and remarks



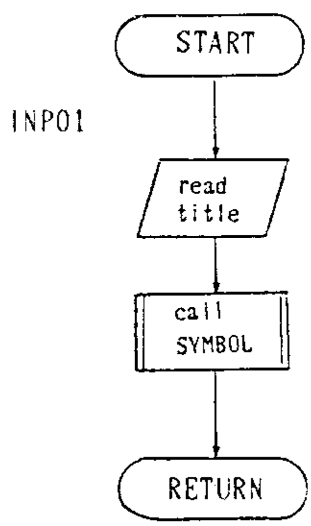
module name	INP01	Module type	Subroutine	Module NO.	4
Function	reads the title of problem and draws its title.				
Calling form	INP01(TITLE)				
Call module	PLOT, SYMBOL				
Called module	MAIN				

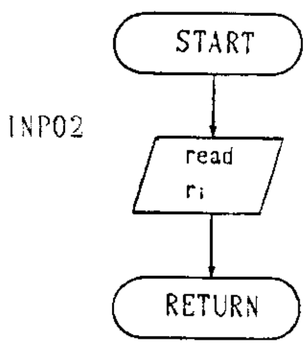
Arguments:
[i/o] TITLE* {C*50} : title of problem

module name	INP02	Module type	Subroutine	Module NO.	5
Function	reads the coefficients of the polynomial.				
Calling form	INP02(A, IRC)				
Call module					
Called module	MAIN				

Arguments:
[i/o] A {R*4} (1): coefficients of polynomial
[out] IRC {I*4} : return code

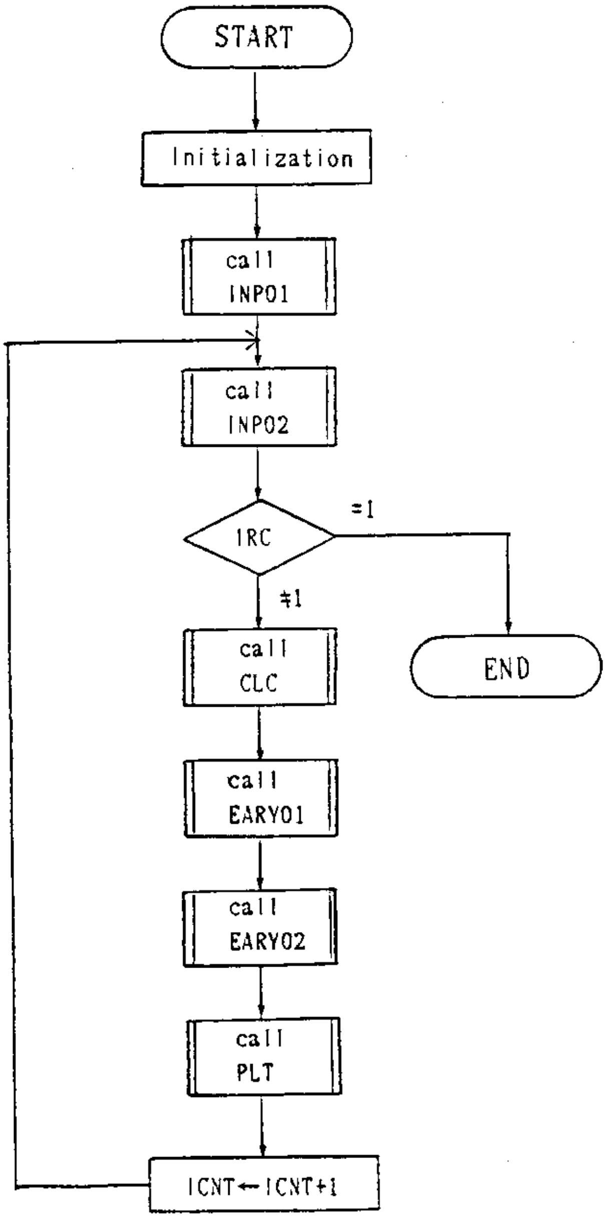
Flow chart and remarks





module name	MAIN	Module type	Subroutine	Module NO.	6
Function	calls individual subroutines after initialization				
Calling form					
Call module	CLC, EARY01, EARY02, INP01, INP02, PLOT, PLOTS, PLT				
Called module					

Flow chart and remarks

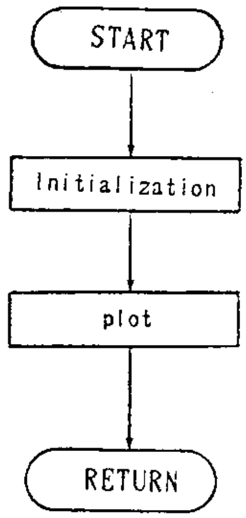


module name	PLT	Module type	Subroutine	Module NO.	7
Function	provides plots of computed results.				
Calling form	PLT(N, Y, GX, GY, TH, ZMAX, ZMIN)				
Call module	PLOT				
Called module	MAIN				

Arguments:

[i n] N {I*4} : point numbers between upper and lower limits of θ
[i n] Y {R*4} (1): calculated result of function f(θ)
[i n] GX {R*4} : the actual length of x-axis(mm)
[i n] GY {R*4} : the actual length of y-axis(mm)
[i n] TH {R*4} : ±(upper limit of the joint angle θ (deg))
[i n] ZMAX {R*4} :scaled Y-axis value (max)
[i n] ZMIN {R*4} :scaled Y-axis value (min)

Flow chart and remarks



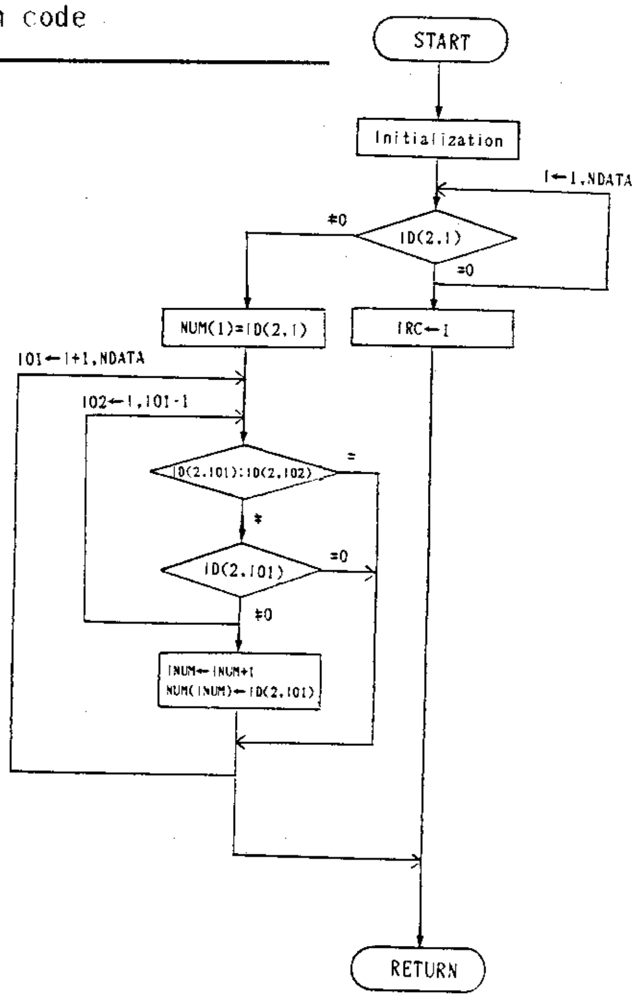
(3) Program GRH52

Module name	EARY01	Module type	Subroutine	Module NO.	1
Function	divides T-number (i.e.numbering of roots) into some groups				
Calling form	EARY01(ID, NUM, NDATA, INUM, IRC)				
Call module					
Called module	MAIN				

Arguments:

[i n] ID {I*4} (2,1) : position number and T-number
ID(1,NDATA);position number,
ID(2,NDATA);numbering of roots
[out] NUM {I*4} (1) : T-number
[i n] NDATA {I*4} : a total number of data
[i/o] INUM {I*4} : the number of real roots
[out] IRC {I*4} : return code

Flow chart and remarks

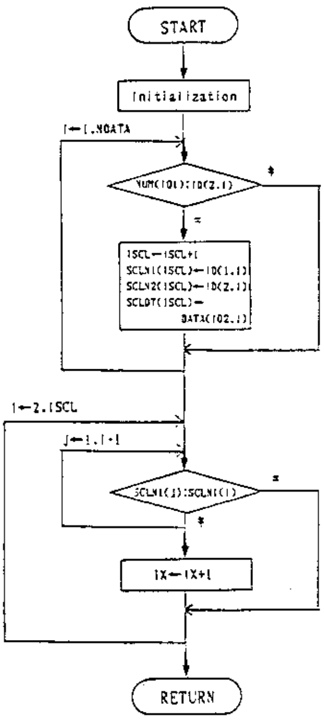


Module name	EARY11	Module type	Subroutine	Module NO.	2
Function	selects the same T-number (the same group of roots).				
Calling form	EARY11(DATA,ID,NUM,SCLDT,SCLN1,SCLN2,IX,ISCL,NDATA,I01,I02)				
Call module					
Called module	MAIN				

Arguments:

[i n] DATA {R*4} (6,1): rotation angles (θ_1 to θ_6)
[i n] ID {I*4} (2,1) : position number and number of real roots
[i n] NUM {I*4} (1) : numbering of real roots(T-number)
[out] SCLDT {R*4} (1) :rotation angle corresponding to the same T-number
[i/o] SCLN1 {R*4} (1) :position no. corresponding to the same T-number
[out] SCLN2 {R*4} (1) : selected T-number
[i/o] IX {I*4} : the position numbers
[i/o] ISCL {I*4} : the number of data in the same T-number
[i n] NDATA {I*4} : a total number of data
[i n] I01 {I*4} : variable of NUM
[i n] I02 {I*4} : variable of angles θ_1 to θ_6

Flow chart and remarks

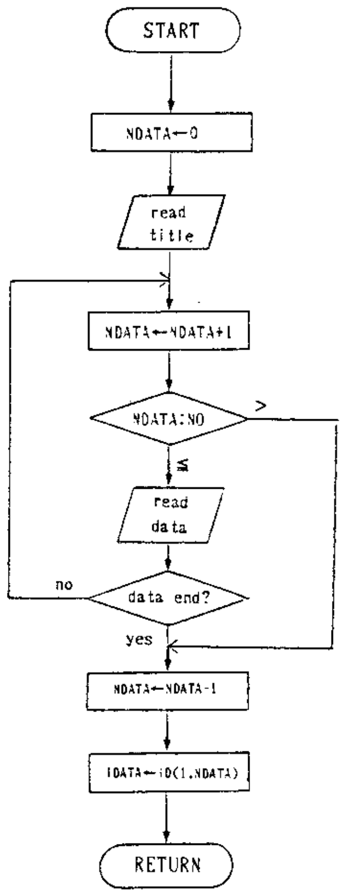


Module name	INP	Module type	Subroutine	Module NO.	3
Function	reads the data to be plotted.				
Calling form	INP(NO, DATA, ID, NDATA, TITLE, IDATA)				
Call module					
Called module	MAIN				

Arguments:

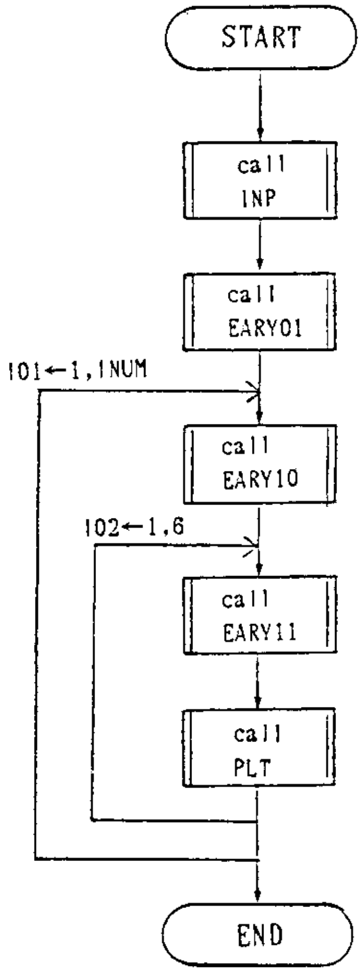
[i n] NO {I*4} : the array size
[i/o] DATA {R*4} (6,1): joint angles θ_1 to θ_6
[i/o] ID {I*4} (2,1) : position number and T-number
[i/o] NDATA {I*4} : a total number of data
[out] TITLE {C*50} : title of problem
[out] IDATA {I*4} : posotion numbers

Flow chart and remarks



Module name	MAIN	Module type	Subroutine	Module NO.	4
Function	calls individual subroutines.				
Calling form					
Call module	EARY01,EARY10,EARY11,INP,PLOT,PLOTS,PLT,SCALE,SYMBOL				
Called module					

Arguments:

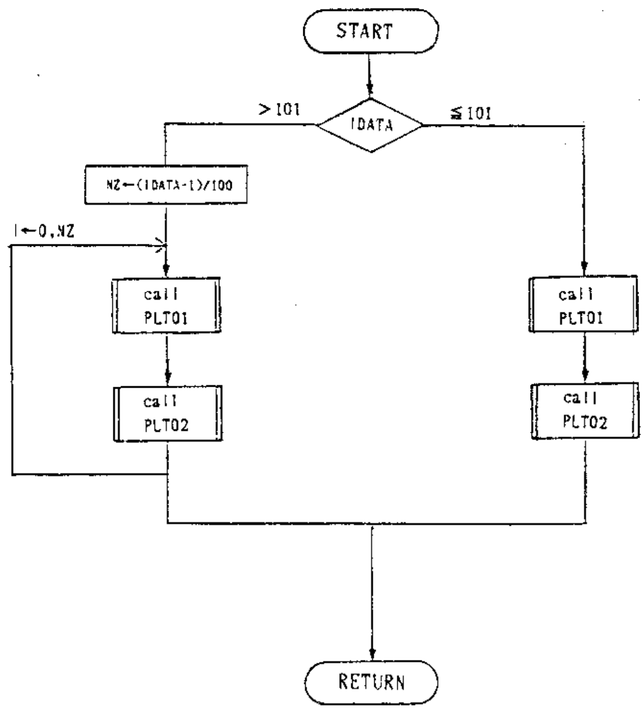


Module name	PLT	Module type	Subroutine	Module NO.	5
Function	makes plots of the individual joints angles data				
Calling form	PLT(SCLDT,SCLN1,SCLN2,SCLX,NDATA,GX,GY,IDATA,ISCL,IO2,NUMO2)				
Call module	PLOT, PLTO1, PLTO2, SCALE				
Called module	MAIN				

Arguments:

[i/o] SCLDT* {R*4} (1):rotation angle corresponding to the same T-number
[i/o] SCLN1* {R*4} (1):position no. corresponding to the same T-number
[i/o] SCLN2* {R*4} (1): selected T-number
[out] SCLX* {R*4} (1) : scale data of x-axis(min,max)
[i/o] NDATA* {I*4} : a total number of data
[i/o] GX* {R*4} : the actual length of x-axis (mm)
[i/o] GY* {R*4} : the actual length of y-axis (mm)
[i n] IDATA* {I*4} : position numbers
[i/o] ISCL* {I*4} : the number of data in the same T-number
[i/o] IO2* {I*4} : variable of the angles θ_1 to θ_6
[i/o] NUMO2* {I*4} : T-number

Flow chart and remarks



Module name	EARY10	Module type	Subroutine	Module NO.	6
Function	makes a plot of character " T-number ".				
Calling form	EARY10(NUM, I01, NUM02, T1)				
Call module	PLOT, SYMBOL				
Called module	MAIN				

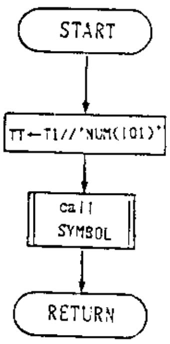
Arguments:
- [i n] NUM {I*4} (1) : T-number(numbering of real roots)
- [i n] I01 {I*4} : variable of NUM
- [i/o] NUM02 {I*4} : T-number(work area)
- [i n] T1 {C*11} : character constant (title of "T-NUMBER")

Module name	PLT01	Module type	Subroutine	Module NO.	7
Function	plots the co-ordinate axis.				
Calling form	PLT01(SCLDT, SCLX, ISCL, GX, GY, I02)				
Call module	AXIS, PLOT				
Called module	PLT				

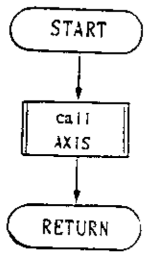
Arguments:
- [i/o] SCLDT* {R*4} (1): rotation angle corresponding to the same T-number
- [i/o] SCLX* {R*4} (1): scaled x-axis value
- [i n] ISCL {I*4} : the number of data in the same T-number
- [i/o] GX* {R*4} : the actual length of x-axis (mm)
- [i/o] GY* {R*4} : the actual length of y-axis (mm)
- [i n] I02 {I*4} : variable of the angles θ_1 to θ_6

Flow chart and remarks

EARY10



PLT01

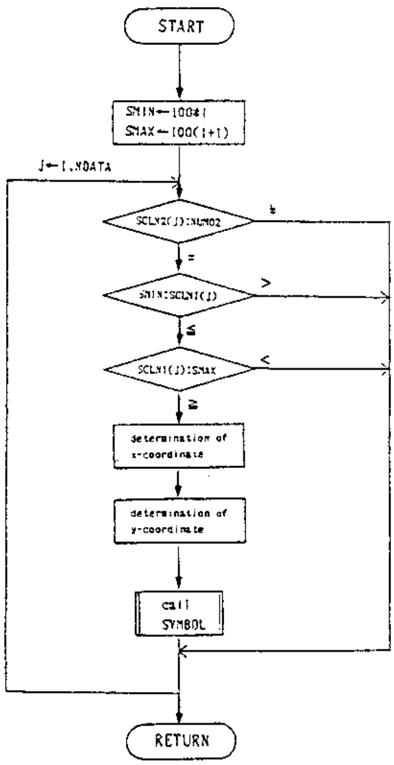


Module name	PLT02	Module type	Subroutine	Module NO.	8
Function	makes plots of joints data.				
Calling form	PLT02(SCLDT,SCLN1,SCLN2,SCLX,NDATA,ISCL,NUM02,I,I02)				
Call module	SYMBOL				
Called module	PLT				

Arguments:

[i n] SCLDT {R*4} (1) :rotation angle corresponding to the same T-number
[i n] SCLN1 {R*4} (1) :position no. corresponding to the same T-number
[i n] SCLN2 {R*4} (1) : selected T-number
[i n] SCLX {R*4} (1) : scaled x-axis value
[i n] NDATA {I*4} : a total number of data
[i n] ISCL {I*4} : the number of data in the same T-number
[i n] NUM02 {I*4} : T-number
[i n] I {I*4} : variable related to the range to be plotted
[i n] I02 {I*4} : variable of the angles θ_1 to θ_6

Flow chart and remarks



```
=ANALYSIS/77= ** TREE STRUCTURE **  ENTRY POINT = MAIN

MAIN  ----*DACOS
        +--DATAIN-----INPUT0-----AXYZ
        I          +--INPUT2----*DCOS
        I          I          +-*DSIN
        I          I          +--AXYZ
        I          +--INPUT1----*DCOS
        I                     +-*DSIN
        I                     +--AXYZ
        +--VEC1
        +--VEC   ----*DBLE
        I          +-*DSQRT
        I          +--N
        +--T6    -----CFSET -----DCLEAR
        +--DAISU -----T6Z   ----*DABS
        I          I          +-*DLOG10
        I          I          +-*DNINT
        I          I          +-*DBLE
        I          +--INDEX ----*DABS
        I          +--RS250 ----*DABS
        I          +--BEA   -----DMTX  ----*DABS
        I          I          +--T6Z   ----*DABS
        I          I                     +-*DLOG10
        I          I                     +-*DNINT
        I          I                     +-*DBLE
        I          +-*DABS
        I          +-*DCMPLX
        I          +-*CDSQRT
        +--JO    ----*DIMAG
        I          +-*DABS
        I          +-*DREAL
        +--THETA ----*DATAN
        I          +--VECSET----*DCOS
        I          I          +-*DSIN
        I          +--S1    ----*DATAN2
        I          +--SS23  ----*DCOS
        I          I          +-*DSIN
        I          I          +-*DTAN
        I          I          +-*DSQRT
        I          I          +-*DATAN2
        I          +--S4    ----*DCOS
        I          I          +-*DSIN
        I          I          +-*DABS
        I          I          +-*DATAN2
        I          +--S5    ----*DSIN
        I          I          +-*DCOS
        I          I          +-*DABS
        I          I          +-*DATAN2
        I          +--S2S3  ----*DSIN
        I          I          +-*DCOS
        I          I          +-*DATAN2
        I          +--CHK   ----*DSIN
        I                     +-*DCOS
        I                     +-*DABS
        I                     +--OUT   ----*SNGL
        I                     +-*DBLE
        +--OUTANS----*IDINT
        I          +-*SNGL
        +--NANS  -----NOA
```

Program Tree Structure of Code MANU

```
MAIN   ----*PLOTS
         +--INP01 -----SYMBOL----*SYMB4
         I            +-*PLOT
         +--INP02
         +--CLC   ----*ACOS
         I            +-*TAN
         I            +-*FLOAT
         +--EARY01----*AMIN1
         I            +-*ABS
         +--EARY02----*PLOT
         I            +--SYMBOL----*SYMB4
         I            +-*IFIX
         I            +-*FLOAT
         I            +-*NUMBER
         I            +-*ALOG10
         I            +-*ABS
         +--PLT   ----*FLOAT
         I            +-*PLOT
         +-*PLOT
```

Program Tree Structure of Code GRH51

```
MAIN   -----INP
         +--EARY01
         +-*PLOTS
         +--SYMBOL----*SYMB4
         +-*PLOT
         +--EARY10-----SYMBOL----*SYMB4
         I            +-*PLOT
         +--EARY11
         +-*SCALE
         +--PLT   ----*FLOAT
                      +-*SCALE
                      +--PLT01 ----*AXIS
                      I            +-*PLOT
                      +--PLT02 -----SYMBOL----*SYMB4
                      +-*PLOT
```

Program Tree Structure of Code GRH52

Common Table of ARM1

```
=ANALYSIS/77=      COMMON SPECIFICATION PATTERN                              DATE  1986/02/14(FRIDAY)     TIME 14:35:48  PAGE 0003

+----------------------------------------------------------------------------------------------------------------------------+
I  AF                                                                                                                         I
I        COMMON/AF/VA,VB,VC,VD,VE,VF,VXX,VYY,VZZ                               .....SS23     S1       S2S3     S4      S5     I
I                                                                                   VECSET                                    I
+----------------------------------------------------------------------------------------------------------------------------+
I  MO                                                                                                                         I
I        COMMON/MO/MO1,MO2,MO3,MO4,MO5,MO6                                     .....SS23     S1       S2S3     S4      S5     I
I                                                                                   THETA                                     I
I        COMMON/MO/MO(6)                                                       .....INP                                       I
+----------------------------------------------------------------------------------------------------------------------------+
I  MS                                                                                                                         I
I        COMMON/MS/MS1,MS2,MS3,MS4,MS5,MS6                                     .....SS23     S1       S2S3     S4      S5     I
I                                                                                   THETA                                     I
I        COMMON/MS/MS(6)                                                       .....INP                                       I
+----------------------------------------------------------------------------------------------------------------------------+
I  NA                                                                                                                         I
I        COMMON/NA/A1,A2,A3,A4,A5,A6                                           .....CHK      SS23     S1       S2S3    T6     I
I                                                                                   VECSET                                    I
I        COMMON/NA/A(6)                                                        .....INP                                       I
+----------------------------------------------------------------------------------------------------------------------------+
I  PAI                                                                                                                        I
I        COMMON/PAI/PP                                                         .....CHK      CHKA     INPUT1   INPUT2  MAIN   I
I                                                                                   OUT      SS23     S1       S2S3    S4     I
I                                                                                   S5       THETA                            I
+----------------------------------------------------------------------------------------------------------------------------+
I  PAR                                                                                                                        I
I        COMMON/PAR/ZN(3),ZO(3),ZA(3),ZP(3)                                    .....CHK      T6                               I
I        COMMON/PAR/XY(3,4)                                                    .....N        VEC      VEC1                    I
I        COMMON/PAR/NX,NY,NZ,OX,OY,OZ,AX,AY,AZ,PX,PY,PZ                        .....NOA      OUT      SS23     S2S3    VECSET I
+----------------------------------------------------------------------------------------------------------------------------+
```

Appendix 2. Input Data Requirements

In this section , the ARM1 input data requirements are presented in the form necessary for computer execution from TSS terminals.

Record 1 TITLE(A50)* = Title of Problem
Any alphanumeric or special characters may be input in column 1 through 50.

Record 2 NDEL(I*4) = Position numbers between initial and
($NDEL \geq 0$) terminal position.
If NDEL = 0, Record No.8 through No.11 are neglected, because a point interpolation is not used.

Record 3 EPS(R*8) = Convergence condition related to the determination of a quadratic factor ($x^2 + px + q$)

EPS1(R*8) = Check of the validity of the calculated articular angles.
Joint angles solutions are substituted into the components of the T_6 matrix and compared with the given data.

Record 4 ~ Record 6 ---- Initial point data ----

Record 4 DATA(1,4,1) = P_x(R*8) = Initial position of x-direction of the manipulator hand (m)

DATA(2,4,1) = P_y(R*8) = Initial position of y-direction of the manipulator hand (m)

---

*) ( ) denotes the type of variable.

DATA(3,4,1) = P_z(R*8) = Initial position of z-direction of the manipulator hand

Record 5 GET('A1') = Option of orientation calculation

= '0': user specified

= '1': Roll-Pitch-Yaw transformation

= 'Z': Return to the initial stage Record 1 (Re-trial of input data).

= excepting the above letter (default)

: Euler transformation

Record 6

(i) If GET = '0', then

DATA(1,1,1) = NX(R*8) = x-component of normal vector n

DATA(2,1,1) = NY(R*8) = y-component of normal vector n

DATA(3,1,1) = NZ(R*8) = z-component of normal vector n

(ii) If GET = '1', then

w_1(R*8) = Rotation angle about the z-axis (Deg)

w_2(R*8) = Rotation angle about the y-axis (Deg)

w_3(R*8) = Rotation angle about the x-axis (Deg)

The direction cosines are calculated as follows.

DATA(1,1,1) = $\cos w_1 \cdot \cos w_2$

DATA(2,1,1) = $\sin w_1 \cdot \cos w_2$

DATA(3,1,1) = $-\sin w_2$

DATA(1,2,1) = $\cos w_1 \cdot \sin w_2 \cdot \sin w_3 - \sin w_1 \cdot \cos w_3$

DATA(2,2,1) = $\sin w_1 \cdot \sin w_2 \cdot \sin w_3 + \cos w_1 \cdot \cos w_3$

DATA(3,2,1) = $\cos w_2 \cdot \sin w_3$

(iii) If GET = Euler option,

w_1(R*8) = Rotation angle about the z-axis (Deg)

w_2(R*8) = Rotation angle about the y-axis (Deg)

w_3(R*8) = Rotation angle about the z-axis (Deg)

The direction cosines are:

DATA(1,1,1) = $\cos w_1 \cdot \cos w_2 \cdot \cos w_3 - \sin w_1 \cdot \sin w_3$

DATA(2,1,1) = $\sin w_1 \cdot \cos w_2 \cdot \cos w_3 + \cos w_1 \cdot \sin w_3$

DATA(3,1,1) = $-\sin W_2 \cdot \cos W_3$

DATA(1.2,1) = $-\cos w_1 \cdot \cos w_2 \cdot \sin w_3 - \sin w_1 \cdot \cos w_3$

DATA(2,2,1) = $-\sin w_1 \cdot \cos W_2 \cdot \sin W_3 + \cos w_1 \cdot \cos w_3$

DATA(3,2,1) = $\sin w_2 \cdot \sin w_3$

Record 7 If GET = '0', then

DATA(1,2,1) = OX(R*8) = x-component of sliding vector O

DATA(2,2,1) = OY(R*8) = y-component of sliding vector O

DATA(3,2,1) = OZ(R*8) = z-component of sliding vector O

Record 8 ~ Record 11 ---- Terminal point data

Record 8 DATA(1,4,2) = P_x(R*8) = Terminal position of x-direction of the manipulator hand (m)

DATA(2,4,2) = P_y(R*8) = Terminal position of y-direction of the manipulator hand (m)

DATA(3,4,2) = P_z(R*8) = Terminal position of z-direction of the manipulator hand (m)

Record 9 GET('A1') = Option of orientation calculation

= '0': user specified

= '1': Roll-Pitch-Yaw transformation

= 'Z': Return to Record 4

(Retrial of end-point data)

= excepting the above letter (default)

: Euler transformation

Record 10

(i) If GET = '0', then

DATA(1,1,2) = NX(R*8) = x-component of normal vector n

DATA(2,1,2) = NY(R*8) = y-component of normal vector n

DATA(3,1,2) = NZ(R*8) = z-component of normal vector n

(ii) If GET = '1', then

W_1(R*8) = Rotation angle about the z-axis (Deg)

W_2(R*8) = Rotation angle about the y-axis (Deg)

W_3(R*8) = Rotation angle about the x-axis (Deg)

The direction cosines are calculated as follows.

DATA(1,1,2) = $\cos w_1 \cdot \cos w_2$

DATA(2,1,2) = $\sin w_1 \cdot \cos w_2$

DATA(3,1,2) = $-\sin w_2$

DATA(1,2,2) = $\cos w_1 \cdot \sin w_2 \cdot \sin w_3 - \sin w_1 \cdot \cos w_3$

DATA(2,2,2) = $\sin w_1 \cdot \sin w_2 \cdot \sin w_3 + \cos w_1 \cdot \cos w_3$

DATA(3,2,2) = $\cos w_2 \cdot \sin w_3$

(ii) If GET = Euler option,

W_1(R*8) = Rotation angle about the z-axis (Deg)

W_2(R*8) = Rotation angle about the y-axis (Deg)

W_3(R*8) = Rotation angle about the z-axis (Deg)

The direction cosines are:

DATA(1,1,2) = $\cos w_1 \cdot \cos w_2 \cdot \cos w_3 - \sin w_1 \cdot \sin w_3$

DATA(2,1,2) = $\sin w_1 \cdot \cos w_2 \cdot \cos w_3 + \cos w_1 \cdot \sin w_3$

DATA(3,1,2) = $-\sin w_2 \cdot \cos w_3$

DATA(1,2,2) = $-\cos w_1 \cdot \cos w_2 \cdot \sin w_3 - \sin w_1 \cdot \cos w_3$

DATA(2,2,2) = $-\sin w_1 \cdot \cos w_2 \cdot \sin w_3 + \cos w_1 \cdot \cos w_3$

DATA(3,2,2) = $\sin w_2 \cdot \sin w_3$

Record 11 If GET = '0', then

DATA(1,2,2) = OX(R*8) = x-component of sliding vector O

DATA(2,2,2) = OY(R*8) = y-component of sliding vector O

DATA(3,2,2) = OZ(R*8) = z-component of sliding vector O

Appendix 3. Sample Problem

(1) Schematic description of sample problem

- Position co-ordinate of the initial point A =

  (-100,350,1631) (mm in unit)

- Position co-ordinate of the terminal point B =

  (100,350,1631) (mm in unit)

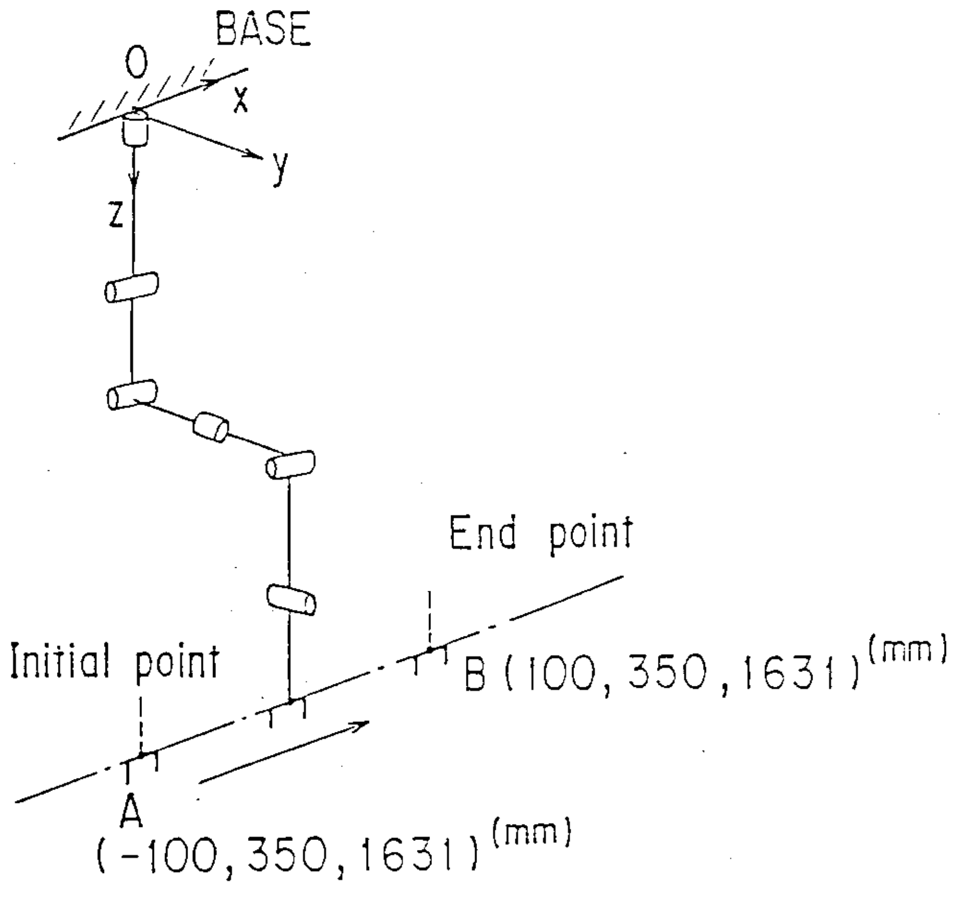
- Number of points (position numbers) = 41

- Direction cosines

NX = 0.0	:	OX = 1.0
NY = 0.0	:	OY = 0.0
NZ = 1.0	:	OZ = 0.0

- Convergence condition

  EPS = 10^{-4} , EPSL = 10^{-3}



Straight line motion from point A to B

(2) List of calculated joint angle solutions (short edit)

+++ BENCH MARK 2 +++
THETA (DEG)

NO. --	---	1	2	3	4	5	6
0	1	-177.2467	-0.1502	-85.7259	146.2277	-85.0428	33.6731
0	1	2.7533	0.1502	85.7259	-33.7722	-85.0428	33.6731
0	1	182.7533	-0.1502	-85.7259	146.2277	-85.0428	33.6731
0	2	18.0896	70.5747	-87.7473	-18.8751	16.2995	-5.4810
1	1	-177.5756	-0.1250	-86.0568	147.5513	-85.4780	32.3679
1	1	2.4244	0.1250	86.0568	-32.4487	-85.4780	32.3679
1	1	182.4244	-0.1250	-86.0568	147.5513	-85.4780	32.3679
1	2	17.3077	70.5241	-87.9459	-18.0868	16.6095	-5.3335
2	1	-177.8831	-0.1028	-86.3825	148.9123	-85.8978	31.0228
2	1	2.1169	0.1028	86.3825	-31.0877	-85.8978	31.0228
2	1	182.1169	-0.1028	-86.3825	148.9123	-85.8978	31.0228
2	2	16.5135	70.4751	-88.1371	-17.2828	16.9110	-5.1715
3	1	-178.1690	-0.0836	-86.7021	150.3110	-86.3012	29.6376
3	1	1.8310	0.0836	86.7021	-29.6890	-86.3012	29.6376
3	1	181.8310	-0.0836	-86.7021	150.3110	-86.3012	29.6376
3	2	15.7069	70.4277	-88.3205	-16.4630	17.2034	-4.9951
4	1	-178.4327	-0.0670	-87.0145	151.7475	-86.6876	28.2126
4	1	1.5673	0.0670	87.0145	-28.2524	-86.6876	28.2126
4	1	181.5672	-0.0670	-87.0145	151.7475	-86.6876	28.2126

+++ BENCH MARK 2 +++
THETA (DEG)

NO. --	---	1	2	3	4	5	6
4	2	14.8882	70.3821	-88.4961	-15.6275	17.4859	-4.8043
5	1	-178.6742	-0.0529	-87.3186	153.2218	-87.0563	26.7478
5	1	1.3258	0.0529	87.3186	-26.7782	-87.0563	26.7478
5	1	181.3258	-0.0529	-87.3186	153.2218	-87.0563	26.7478
5	2	14.0575	70.3384	-88.6635	-14.7764	17.7577	-4.5994
6	1	-178.8933	-0.0411	-87.6132	154.7336	-87.4065	25.2438
6	1	1.1067	0.0411	87.6132	-25.2664	-87.4065	25.2438
6	1	181.1067	-0.0411	-87.6132	154.7336	-87.4065	25.2438
6	2	13.2151	70.2967	-88.8223	-13.9099	18.0179	-4.3806
7	1	-179.0901	-0.0313	-87.8971	156.2823	-87.7375	23.7012
7	1	0.9099	0.0313	87.8971	-23.7177	-87.7375	23.7012
7	1	180.9099	-0.0313	-87.8971	156.2823	-87.7375	23.7012
7	2	12.3611	70.2572	-88.9725	-13.0283	18.2659	-4.1480
8	1	-179.2649	-0.0233	-88.1689	157.8673	-88.0485	22.1211
8	1	0.7351	0.0233	88.1689	-22.1327	-88.0485	22.1211
8	1	180.7351	-0.0233	-88.1689	157.8673	-88.0485	22.1211
8	2	11.4961	70.2199	-89.1135	-12.1319	18.5006	-3.9022
9	1	-179.4180	-0.0169	-88.4272	159.4876	-88.3388	20.5045
9	1	0.5820	0.0169	88.4272	-20.5124	-88.3388	20.5045
9	1	180.5820	-0.0169	-88.4272	159.4876	-88.3388	20.5045

+++ BENCH MARK 2 +++
THETA (DEG)

NO.	1	2	3	4	5	6
NO. -- 9 --- 2	10.6204	70.1851	-89.2452	-11.2212	18.7214	-3.6435
NO. -- 10 --- 1	-179.5500	-0.0118	-88.6707	161.1419	-88.6078	18.8529
NO. -- 10 --- 1	0.4499	0.0118	88.6707	-18.8580	-88.6078	18.8529
NO. -- 10 --- 1	180.4499	-0.0118	-88.6707	161.1419	-88.6078	18.8529
NO. -- 10 --- 2	9.7345	70.1527	-89.3673	-10.2968	18.9273	-3.3725
NO. -- 11 --- 1	-179.6619	-0.0080	-88.8978	162.8287	-88.8548	17.1680
NO. -- 11 --- 1	0.3381	0.0080	88.8978	-17.1712	-88.8548	17.1680
NO. -- 11 --- 1	180.3381	-0.0080	-88.8978	162.8287	-88.8548	17.1680
NO. -- 11 --- 2	8.8390	70.1229	-89.4794	-9.3594	19.1176	-3.0898
NO. -- 12 --- 1	-179.7546	-0.0052	-89.1072	164.5462	-89.0791	15.4519
NO. -- 12 --- 1	0.2454	0.0052	89.1072	-15.4538	-89.0791	15.4519
NO. -- 12 --- 1	180.2454	-0.0052	-89.1072	164.5462	-89.0791	15.4519
NO. -- 12 --- 2	7.9345	70.0958	-89.5812	-8.4098	19.2914	-2.7963
NO. -- 13 --- 1	-179.8294	-0.0032	-89.2975	166.2920	-89.2802	13.7069
NO. -- 13 --- 1	0.1706	0.0032	89.2975	-13.7080	-89.2802	13.7069
NO. -- 13 --- 1	180.1706	-0.0032	-89.2975	166.2920	-89.2802	13.7069
NO. -- 13 --- 2	7.0217	70.0714	-89.6725	-7.4489	19.4481	-2.4926
NO. -- 14 --- 1	-179.8878	-0.0018	-89.4673	168.0639	-89.4574	11.9356
NO. -- 14 --- 1	0.1122	0.0018	89.4673	-11.9361	-89.4574	11.9356
NO. -- 14 --- 1	180.1122	-0.0018	-89.4673	168.0639	-89.4574	11.9356
NO. -- 14 --- 2	6.1014	70.0500	-89.7531	-6.4777	19.5869	-2.1798

+++ BENCH MARK 2 +++
THETA (DEG)

NO.	1	2	3	4	5	6
NO. -- 15 --- 1	-179.9314	-0.0009	-89.6155	169.8590	-89.6103	10.1408
NO. -- 15 --- 1	0.0686	0.0009	89.6155	-10.1410	-89.6103	10.1408
NO. -- 15 --- 1	180.0686	-0.0009	-89.6155	169.8590	-89.6103	10.1408
NO. -- 15 --- 2	5.1744	70.0314	-89.8226	-5.4973	19.7072	-1.8588
NO. -- 16 --- 1	-179.9621	-0.0004	-89.7408	171.6743	-89.7384	8.3256
NO. -- 16 --- 1	0.0379	0.0004	89.7408	-8.3257	-89.7384	8.3256
NO. -- 16 --- 1	180.0379	-0.0004	-89.7408	171.6743	-89.7384	8.3256
NO. -- 16 --- 2	4.2417	70.0158	-89.8809	-4.5090	19.8084	-1.5308
NO. -- 17 --- 1	-179.9821	-0.0002	-89.8423	173.5064	-89.8414	6.4935
NO. -- 17 --- 1	0.0179	0.0002	89.8423	-6.4935	-89.8414	6.4935
NO. -- 17 --- 1	180.0179	-0.0002	-89.8423	173.5064	-89.8414	6.4935
NO. -- 17 --- 2	3.3041	70.0033	-89.9278	-3.5140	19.8900	-1.1968
NO. -- 18 --- 1	-179.9934	-0.0000	-89.9192	175.3520	-89.9190	4.6480
NO. -- 18 --- 1	0.0066	0.0000	89.9192	-4.6480	-89.9190	4.6480
NO. -- 18 --- 1	180.0066	-0.0000	-89.9192	175.3520	-89.9190	4.6480
NO. -- 18 --- 2	2.3628	69.9939	-89.9631	-2.5138	19.9515	-0.8583
NO. -- 19 --- 1	-179.9986	-0.0000	-89.9708	177.2073	-89.9708	2.7927
NO. -- 19 --- 1	0.0014	0.0000	89.9708	-2.7927	-89.9708	2.7927
NO. -- 19 --- 1	180.0014	-0.0000	-89.9708	177.2073	-89.9708	2.7927
NO. -- 19 --- 2	1.4188	69.9876	-89.9867	-1.5098	19.9927	-0.5163

+++ BENCH MARK 2 +++
THETA (DEG)

	1	2	3	4	5	6
NO. -- 20 --- 1	-179.9999	-0.0000	-89.9967	179.0684	-89.9967	0.9316
NO. -- 20 --- 1	0.0001	0.0000	89.9967	-0.9316	-89.9967	0.9316
NO. -- 20 --- 1	180.0000	-0.0000	-89.9967	179.0684	-89.9967	0.9316
NO. -- 20 --- 2	0.4731	69.9844	-89.9985	-0.5035	20.0134	-0.1723
NO. -- 21 --- 1	179.9999	-0.0000	-89.9967	-179.0684	-89.9967	-0.9316
NO. -- 21 --- 1	-0.0001	0.0000	89.9967	0.9316	-89.9967	-0.9316
NO. -- 21 --- 1	-180.0000	-0.0000	-89.9967	-179.0684	-89.9967	-0.9316
NO. -- 21 --- 2	-0.4731	69.9844	-89.9985	0.5035	20.0134	0.1723
NO. -- 22 --- 1	179.9986	-0.0000	-89.9708	-177.2073	-89.9708	-2.7927
NO. -- 22 --- 1	-0.0014	0.0000	89.9708	2.7927	-89.9708	-2.7927
NO. -- 22 --- 1	-180.0014	-0.0000	-89.9708	-177.2073	-89.9708	-2.7927
NO. -- 22 --- 2	-1.4188	69.9876	-89.9867	1.5098	19.9927	0.5163
NO. -- 23 --- 1	179.9934	-0.0000	-89.9192	-175.3520	-89.9190	-4.6480
NO. -- 23 --- 1	-0.0066	0.0000	89.9192	4.6480	-89.9190	-4.6480
NO. -- 23 --- 1	-180.0066	-0.0000	-89.9192	-175.3520	-89.9190	-4.6480
NO. -- 23 --- 2	-2.3628	69.9939	-89.9631	2.5138	19.9515	0.8583
NO. -- 24 --- 1	179.9821	-0.0002	-89.8423	-173.5064	-89.8414	-6.4935
NO. -- 24 --- 1	-0.0179	0.0002	89.8423	6.4935	-89.8414	-6.4935
NO. -- 24 --- 1	-180.0179	-0.0002	-89.8423	-173.5064	-89.8414	-6.4935
NO. -- 24 --- 2	-3.3041	70.0033	-89.9278	3.5140	19.8900	1.1968
NO. -- 25 --- 1	179.9621	-0.0004	-89.7408	-171.6743	-89.7384	-8.3256

+++ BENCH MARK 2 +++
THETA (DEG)

	1	2	3	4	5	6
NO. -- 25 --- 1	-0.0379	0.0004	89.7408	8.3257	-89.7384	-8.3256
NO. -- 25 --- 1	-180.0379	-0.0004	-89.7408	-171.6743	-89.7384	-8.3256
NO. -- 25 --- 2	-4.2417	70.0158	-89.8809	4.5090	19.8084	1.5308
NO. -- 26 --- 1	179.9314	-0.0009	-89.6155	-169.8590	-89.6103	-10.1408
NO. -- 26 --- 1	-0.0686	0.0009	89.6155	10.1410	-89.6103	-10.1408
NO. -- 26 --- 1	-180.0686	-0.0009	-89.6155	-169.8590	-89.6103	-10.1408
NO. -- 26 --- 2	-5.1744	70.0314	-89.8226	5.4973	19.7072	1.8588
NO. -- 27 --- 1	179.8878	-0.0018	-89.4673	-168.0639	-89.4574	-11.9356
NO. -- 27 --- 1	-0.1122	0.0018	89.4673	11.9361	-89.4574	-11.9356
NO. -- 27 --- 1	-180.1122	-0.0018	-89.4673	-168.0639	-89.4574	-11.9356
NO. -- 27 --- 2	-6.1014	70.0500	-89.7531	6.4777	19.5869	2.1798
NO. -- 28 --- 1	179.8294	-0.0032	-89.2975	-166.2920	-89.2802	-13.7069
NO. -- 28 --- 1	-0.1706	0.0032	89.2975	13.7080	-89.2802	-13.7069
NO. -- 28 --- 1	-180.1706	-0.0032	-89.2975	-166.2920	-89.2802	-13.7069
NO. -- 28 --- 2	-7.0217	70.0714	-89.6725	7.4489	19.4481	2.4926
NO. -- 29 --- 1	179.7546	-0.0052	-89.1072	-164.5462	-89.0791	-15.4519
NO. -- 29 --- 1	-0.2454	0.0052	89.1072	15.4538	-89.0791	-15.4519
NO. -- 29 --- 1	-180.2454	-0.0052	-89.1072	-164.5462	-89.0791	-15.4519
NO. -- 29 --- 2	-7.9345	70.0958	-89.5812	8.4098	19.2914	2.7963
NO. -- 30 --- 1	179.6619	-0.0080	-88.8978	-162.8287	-88.8548	-17.1680
NO. -- 30 --- 1	-0.3381	0.0080	88.8978	17.1712	-88.8548	-17.1680

+++ BENCH MARK 2 +++
THETA (DEG)

	1	2	3	4	5	6
NO. -- 30 --- 1	-180.3381	-0.0080	-88.8978	-162.8287	-88.8548	-17.1680
NO. -- 30 --- 2	-8.8390	70.1229	-89.4794	9.3594	19.1176	3.0898
NO. -- 31 --- 1	179.5500	-0.0118	-88.6707	-161.1419	-88.6078	-18.8529
NO. -- 31 --- 1	-0.4499	0.0118	88.6707	18.8580	-88.6078	-18.8529
NO. -- 31 --- 1	-180.4499	-0.0118	-88.6707	-161.1419	-88.6078	-18.8529
NO. -- 31 --- 2	-9.7345	70.1527	-89.3673	10.2968	18.9273	3.3725
NO. -- 32 --- 1	179.4180	-0.0169	-88.4272	-159.4876	-88.3388	-20.5045
NO. -- 32 --- 1	-0.5820	0.0169	88.4272	20.5124	-88.3388	-20.5045
NO. -- 32 --- 1	-180.5820	-0.0169	-88.4272	-159.4876	-88.3388	-20.5045
NO. -- 32 --- 2	-10.6204	70.1851	-89.2452	11.2212	18.7214	3.6435
NO. -- 33 --- 1	179.2649	-0.0233	-88.1689	-157.8673	-88.0485	-22.1211
NO. -- 33 --- 1	-0.7351	0.0233	88.1689	22.1327	-88.0485	-22.1211
NO. -- 33 --- 1	-180.7351	-0.0233	-88.1689	-157.8673	-88.0485	-22.1211
NO. -- 33 --- 2	-11.4961	70.2199	-89.1135	12.1319	18.5006	3.9022
NO. -- 34 --- 1	179.0901	-0.0313	-87.8971	-156.2823	-87.7375	-23.7012
NO. -- 34 --- 1	-0.9099	0.0313	87.8971	23.7177	-87.7375	-23.7012
NO. -- 34 --- 1	-180.9099	-0.0313	-87.8971	-156.2823	-87.7375	-23.7012
NO. -- 34 --- 2	-12.3611	70.2572	-88.9725	13.0283	18.2659	4.1480
NO. -- 35 --- 1	178.8933	-0.0411	-87.6132	-154.7336	-87.4065	-25.2438
NO. -- 35 --- 1	-1.1067	0.0411	87.6132	25.2664	-87.4065	-25.2438

+++ BENCH MARK 2 +++
THETA (DEG)

	1	2	3	4	5	6
NO. -- 35 --- 1	-181.1067	-0.0411	-87.6132	-154.7336	-87.4065	-25.2438
NO. -- 35 --- 2	-13.2151	70.2967	-88.8223	13.9099	18.0179	4.3806
NO. -- 36 --- 1	178.6742	-0.0529	-87.3186	-153.2218	-87.0563	-26.7478
NO. -- 36 --- 1	-1.3258	0.0529	87.3186	26.7782	-87.0563	-26.7478
NO. -- 36 --- 1	-181.3258	-0.0529	-87.3186	-153.2218	-87.0563	-26.7478
NO. -- 36 --- 2	-14.0575	70.3384	-88.6635	14.7764	17.7577	4.5994
NO. -- 37 --- 1	178.4327	-0.0670	-87.0145	-151.7475	-86.6876	-28.2126
NO. -- 37 --- 1	-1.5673	0.0670	87.0145	28.2524	-86.6876	-28.2126
NO. -- 37 --- 1	-181.5672	-0.0670	-87.0145	-151.7475	-86.6876	-28.2126
NO. -- 37 --- 2	-14.8882	70.3821	-88.4961	15.6275	17.4859	4.8043
NO. -- 38 --- 1	178.1690	-0.0836	-86.7021	-150.3110	-86.3012	-29.6376
NO. -- 38 --- 1	-1.8310	0.0836	86.7021	29.6890	-86.3012	-29.6376
NO. -- 38 --- 1	-181.8310	-0.0836	-86.7021	-150.3110	-86.3012	-29.6376
NO. -- 38 --- 2	-15.7069	70.4277	-88.3205	16.4630	17.2034	4.9951
NO. -- 39 --- 1	177.8831	-0.1028	-86.3825	-148.9123	-85.8978	-31.0228
NO. -- 39 --- 1	-2.1169	0.1028	86.3825	31.0877	-85.8978	-31.0228
NO. -- 39 --- 1	-182.1169	-0.1028	-86.3825	-148.9123	-85.8978	-31.0228
NO. -- 39 --- 2	-16.5135	70.4751	-88.1371	17.2828	16.9110	5.1715
NO. -- 40 --- 1	177.5756	-0.1250	-86.0568	-147.5513	-85.4780	-32.3679
NO. -- 40 --- 1	-2.4244	0.1250	86.0568	32.4487	-85.4780	-32.3679
NO. -- 40 --- 1	-182.4244	-0.1250	-86.0568	-147.5513	-85.4780	-32.3679

+++ BENCH MARK 2 +++
THETA (DEG)

1	2	3	4	5	6
NO. -- 40	--- 2				
-17.3077	70.5241	-87.9459	18.0868	16.6095	5.3335
NO. -- 41	--- 1				
177.2467	-0.1502	-85.7259	-146.2277	-85.0428	-33.6731
NO. -- 41	--- 1				
-2.7533	0.1502	85.7259	33.7722	-85.0428	-33.6731
NO. -- 41	--- 1				
-182.7533	-0.1502	-85.7259	-146.2277	-85.0428	-33.6731
NO. -- 41	--- 2				
-18.0896	70.5747	-87.7473	18.8751	16.2995	5.4810

(3) List of calculated results

(complete edit ....extraction)

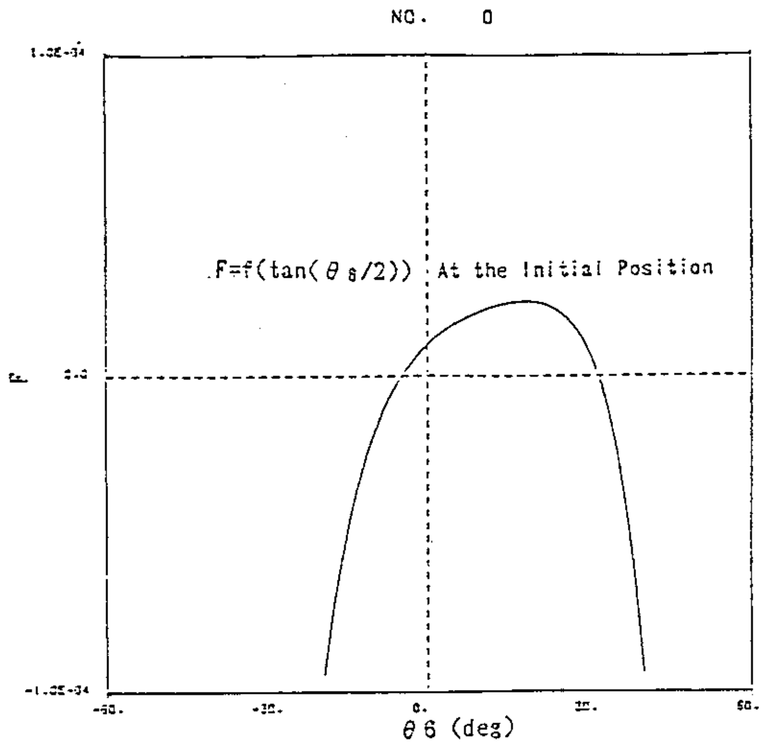
```
****************************************************************************
NO. -------------------------  0  ---  1

ANSWER OF POLYNOMIAL ---------  T  =  0.30261D+00  < T = TAN(THETA6/2) >

 ADOPT ANSWERS (DEG) ----------

    THETA1 = -177.2467     THETA2 =   -0.1502     THETA3 =  -85.7259
    THETA4 =  146.2277     THETA5 =  -85.0428     THETA6 =   33.6731

INPUT                  0.0          :  1.00000D+00 :  0.0          : -1.00000D-01
         VALUES ---    0.0          :  0.0         :  1.00000D+00 :  3.50000D-01
                       1.00000D+00 :  0.0         :  0.0          :  1.63100D+00

CALCULATED             1.09379D-08 :  1.00000D+00 : -7.83633D-10 : -1.00000D-01
         VALUES ---   -1.17620D-09 :  7.83633D-10 :  1.00000D+00 :  3.50000D-01
                       1.00000D+00 : -1.09379D-08 :  1.17620D-09 :  1.63100D+00

ABSOLUTE               1.09379D-08 :  1.11022D-16 :  7.83633D-10 :  4.42800D-09
         ERRORS ---    1.17620D-09 :  7.83633D-10 :  0.0          :  5.41339D-10
                       8.32667D-17 :  1.09379D-08 :  1.17620D-09 :  9.09681D-10

****************************************************************************
NO. -------------------------  0  ---  1

ANSWER OF POLYNOMIAL ---------  T  =  0.30261D+00  < T = TAN(THETA6/2) >

 ADOPT ANSWERS (DEG) ----------

    THETA1 =    2.7533     THETA2 =    0.1502     THETA3 =   85.7259
    THETA4 =  -33.7722     THETA5 =  -85.0428     THETA6 =   33.6731

INPUT                  0.0          :  1.00000D+00 :  0.0          : -1.00000D-01
         VALUES ---    0.0          :  0.0         :  1.00000D+00 :  3.50000D-01
                       1.00000D+00 :  0.0         :  0.0          :  1.63100D+00

CALCULATED             1.09379D-08 :  1.00000D+00 : -7.83633D-10 : -1.00000D-01
         VALUES ---   -1.17620D-09 :  7.83633D-10 :  1.00000D+00 :  3.50000D-01
                       1.00000D+00 : -1.09379D-08 :  1.17620D-09 :  1.63100D+00

ABSOLUTE               1.09379D-08 :  9.71445D-17 :  7.83633D-10 :  4.42800D-09
         ERRORS ---    1.17620D-09 :  7.83633D-10 :  0.0          :  5.41339D-10
                       1.11022D-16 :  1.09379D-08 :  1.17620D-09 :  9.09682D-10

****************************************************************************
NO. -------------------------  0  ---  1

ANSWER OF POLYNOMIAL ---------  T  =  0.30261D+00  < T = TAN(THETA6/2) >

 ADOPT ANSWERS (DEG) ----------

    THETA1 =  182.7533     THETA2 =   -0.1502     THETA3 =  -85.7259
    THETA4 =  146.2277     THETA5 =  -85.0428     THETA6 =   33.6731

INPUT                  0.0          :  1.00000D+00 :  0.0          : -1.00000D-01

         VALUES ---    0.0          :  0.0         :  1.00000D+00 :  3.50000D-01
                       1.00000D+00 :  0.0         :  0.0          :  1.63100D+00

CALCULATED             1.09379D-08 :  1.00000D+00 : -7.83633D-10 : -1.00000D-01
         VALUES ---   -1.17620D-09 :  7.83633D-10 :  1.00000D+00 :  3.50000D-01
                       1.00000D+00 : -1.09379D-08 :  1.17620D-09 :  1.63100D+00

ABSOLUTE               1.09379D-08 :  9.71445D-17 :  7.83633D-10 :  4.42800D-09
         ERRORS ---    1.17620D-09 :  7.83633D-10 :  0.0          :  5.41339D-10
                       1.11022D-16 :  1.09379D-08 :  1.17620D-09 :  9.09681D-10
```

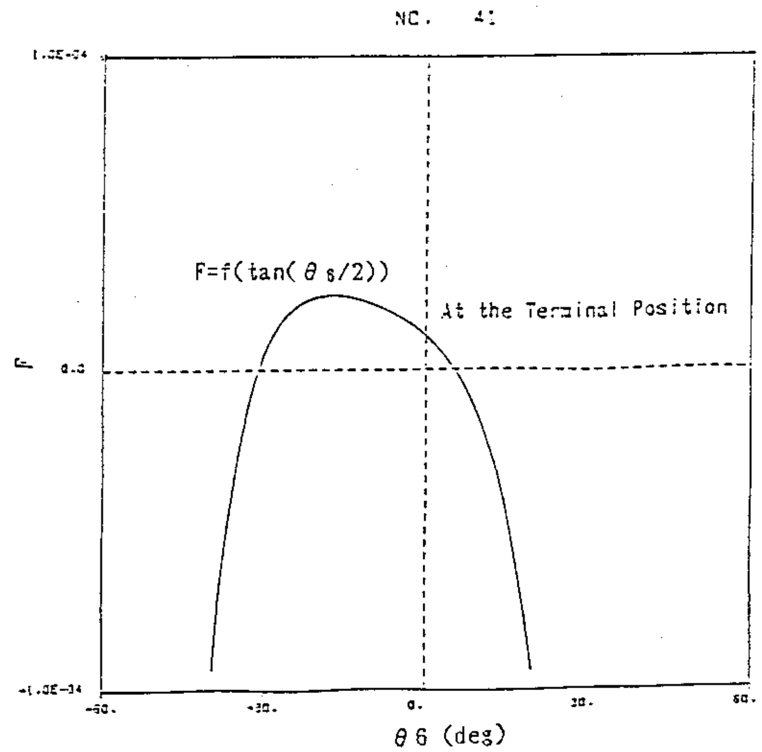
```
*****************************************************************************

NO. -------------------------   0  ---   2

ANSWER OF POLYNOMIAL ---------   T    = -0.47868D-01   < T = TAN(THETA6/2) >

 ADOPT ANSWERS (DEG) ----------

    THETA1 =   18.0896      THETA2 =    70.5747      THETA3 =  -87.7473
    THETA4 =  -18.8751      THETA5 =    16.2995      THETA6 =   -5.4810


INPUT                     0.0            :  1.00000D+00 :  0.0           : -1.00000D-01
          VALUES ---      0.0            :  0.0         :  1.00000D+00   :  3.50000D-01
                          1.00000D+00    :  0.0         :  0.0           :  1.63100D+00

CALCULATED                7.17967D-13    :  1.00000D+00 :  1.92555D-13   : -1.00000D-01
          VALUES ---     -2.00653D-12    : -1.92523D-13 :  1.00000D+00   :  3.50000D-01
                          1.00000D+00    : -7.17995D-13 :  2.00655D-12   :  1.63100D+00

ABSOLUTE                  7.17967D-13    :  2.77556D-17 :  1.92555D-13   :  3.08115D-13
          ERRORS ---      2.00653D-12    :  1.92523D-13 :  0.0           :  8.64267D-13
                          4.16334D-17    :  7.17995D-13 :  2.00655D-12   :  1.08802D-14

*****************************************************************************

NO. -------------------------   1  ---   1

ANSWER OF POLYNOMIAL ---------   T    =  0.29022D+00   < T = TAN(THETA6/2) >

 ADOPT ANSWERS (DEG) ----------

    THETA1 = -177.5756      THETA2 =    -0.1250      THETA3 =  -86.0568
    THETA4 =  147.5513      THETA5 =   -85.4780      THETA6 =   32.3679


INPUT                     0.0            :  1.00000D+00 :  0.0           : -9.51220D-02
          VALUES ---      0.0            :  0.0         :  1.00000D+00   :  3.50000D-01
                          1.00000D+00    :  0.0         :  0.0           :  1.63100D+00

CALCULATED                9.14610D-10    :  1.00000D+00 : -6.07156D-11   : -9.51220D-02
          VALUES ---     -9.57911D-11    :  6.07156D-11 :  1.00000D+00   :  3.50000D-01
                          1.00000D+00    : -9.14610D-10 :  9.57911D-11   :  1.63100D+00

ABSOLUTE                  9.14610D-10    :  5.55112D-17 :  6.07156D-11   :  3.71966D-10
          ERRORS ---      9.57911D-11    :  6.07156D-11 :  0.0           :  4.38334D-11
                          2.77556D-17    :  9.14610D-10 :  9.57911D-11   :  7.34455D-11

*****************************************************************************

NO. -------------------------   1  ---   1

ANSWER OF POLYNOMIAL ---------   T    =  0.29022D+00   < T = TAN(THETA6/2) >

 ADOPT ANSWERS (DEG) ----------

    THETA1 =    2.4244      THETA2 =     0.1250      THETA3 =   86.0568
    THETA4 =  -32.4487      THETA5 =   -85.4780      THETA6 =   32.3679


INPUT                     0.0            :  1.00000D+00 :  0.0           : -9.51220D-02
          VALUES ---      0.0            :  0.0         :  1.00000D+00   :  3.50000D-01
                          1.00000D+00    :  0.0         :  0.0           :  1.63100D+00

CALCULATED                9.14609D-10    :  1.00000D+00 : -6.07155D-11   : -9.51220D-02
          VALUES ---     -9.57909D-11    :  6.07155D-11 :  1.00000D+00   :  3.50000D-01
                          1.00000D+00    : -9.14609D-10 :  9.57910D-11   :  1.63100D+00

ABSOLUTE                  9.14609D-10    :  4.16334D-17 :  6.07155D-11   :  3.71966D-10
          ERRORS ---      9.57909D-11    :  6.07155D-11 :  0.0           :  4.38333D-11
                          2.77556D-17    :  9.14609D-10 :  9.57910D-11   :  7.34452D-11
```

```
******************************************************************************
NO. -------------------------   1  ---   1

ANSWER OF POLYNOMIAL ---------   T   =  0.29022D+00   < T = TAN(THETA6/2) >

 ADOPT ANSWERS (DEG) ----------

    THETA1 =  182.4244      THETA2 =   -0.1250      THETA3 =  -86.0568
    THETA4 =  147.5513      THETA5 =  -85.4780      THETA6 =   32.3679


INPUT                    0.0          :  1.00000D+00 :  0.0          : -9.51220D-02
         VALUES ---      0.0          :  0.0         :  1.00000D+00 :  3.50000D-01
                         1.00000D+00 :  0.0         :  0.0          :  1.63100D+00

CALCULATED               9.14608D-10 :  1.00000D+00 : -6.07154D-11 : -9.51220D-02
         VALUES ---     -9.57908D-11 :  6.07154D-11 :  1.00000D+00 :  3.50000D-01
                         1.00000D+00 : -9.14608D-10 :  9.57908D-11 :  1.63100D+00

ABSOLUTE                 9.14608D-10 :  5.55112D-17 :  6.07154D-11 :  3.71965D-10
         ERRORS ---      9.57908D-11 :  6.07154D-11 :  1.38778D-17 :  4.38333D-11
                         5.55112D-17 :  9.14608D-10 :  9.57908D-11 :  7.34452D-11

******************************************************************************
NO. -------------------------   1  ---   2

ANSWER OF POLYNOMIAL ---------   T   = -0.46577D-01   < T = TAN(THETA6/2) >

 ADOPT ANSWERS (DEG) ----------

    THETA1 =   17.3077      THETA2 =   70.5241      THETA3 =  -87.9459
    THETA4 =  -18.0868      THETA5 =   16.6095      THETA6 =   -5.3335


INPUT                    0.0          :  1.00000D+00 :  0.0          : -9.51220D-02
         VALUES ---      0.0          :  0.0         :  1.00000D+00 :  3.50000D-01
                         1.00000D+00 :  0.0         :  0.0          :  1.63100D+00

CALCULATED               5.84810D-14 :  1.00000D+00 :  1.59326D-14 : -9.51220D-02
         VALUES ---     -1.70985D-13 : -1.59069D-14 :  1.00000D+00 :  3.50000D-01
                         1.00000D+00 : -5.85088D-14 :  1.70974D-13 :  1.63100D+00

ABSOLUTE                 5.84810D-14 :  1.38778D-17 :  1.59326D-14 :  2.49939D-14
         ERRORS ---      1.70985D-13 :  1.59069D-14 :  5.55112D-17 :  7.37188D-14
                         4.16334D-17 :  5.85088D-14 :  1.70974D-13 :  1.33227D-15

******************************************************************************
NO. -------------------------   2  ---   1

ANSWER OF POLYNOMIAL ---------   T   =  0.27754D+00   < T = TAN(THETA6/2) >

 ADOPT ANSWERS (DEG) ----------

    THETA1 = -177.8831      THETA2 =   -0.1028      THETA3 =  -86.3825
    THETA4 =  148.9123      THETA5 =  -85.8978      THETA6 =   31.0228


INPUT                    0.0          :  1.00000D+00 :  0.0          : -9.02439D-02
         VALUES ---      0.0          :  0.0         :  1.00000D+00 :  3.500000D-01
                         1.00000D+00 :  0.0         :  0.0          :  1.63100D+00

CALCULATED               6.44081D-11 :  1.00000D+00 : -3.93843D-12 : -9.02439D-02
         VALUES ---     -6.54880D-12 :  3.93843D-12 :  1.00000D+00 :  3.50000D-01
                         1.00000D+00 : -6.44081D-11 :  6.54880D-12 :  1.63100D+00

ABSOLUTE                 6.44081D-11 :  2.77556D-17 :  3.93843D-12 :  2.63135D-11
         ERRORS ---      6.54880D-12 :  3.93843D-12 :  2.77556D-17 :  2.98002D-12
                         1.38778D-17 :  6.44081D-11 :  6.54880D-12 :  4.97868D-12
```

(4) Graphics output 1

(output of code GRH51)

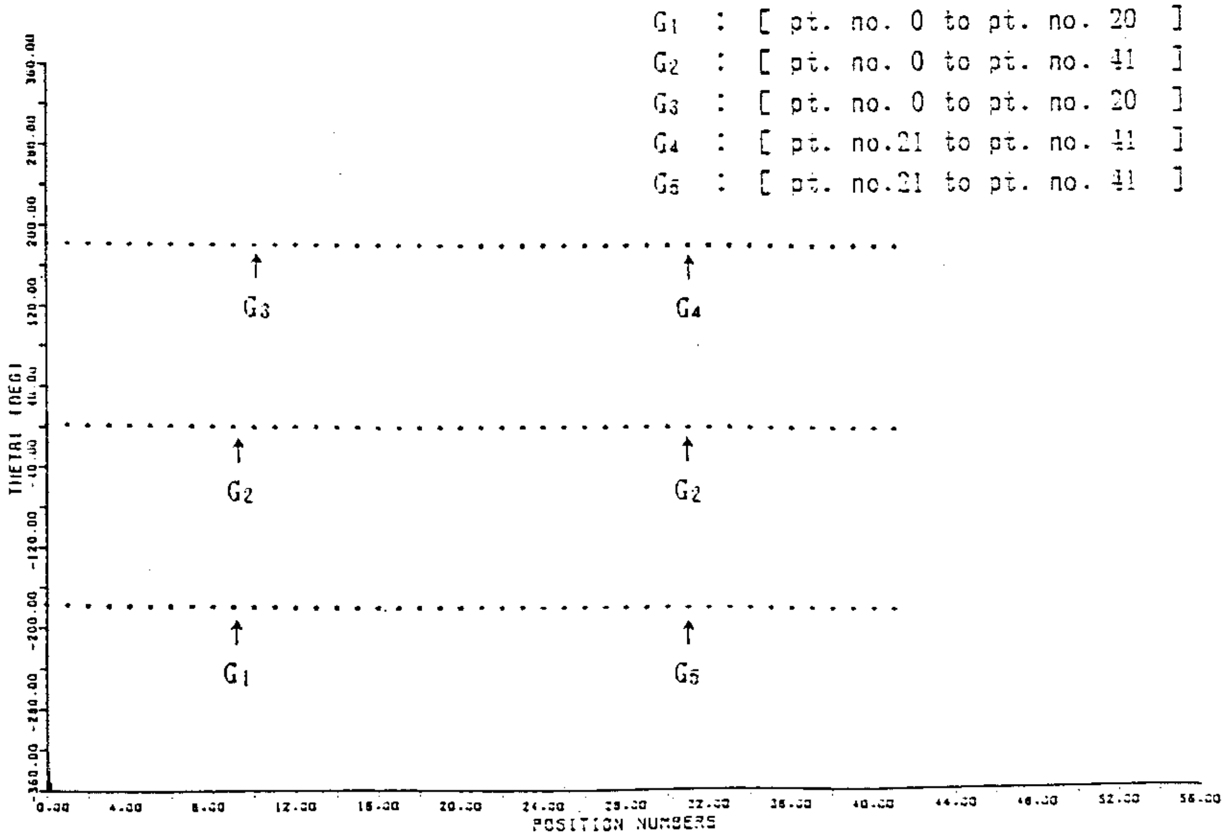




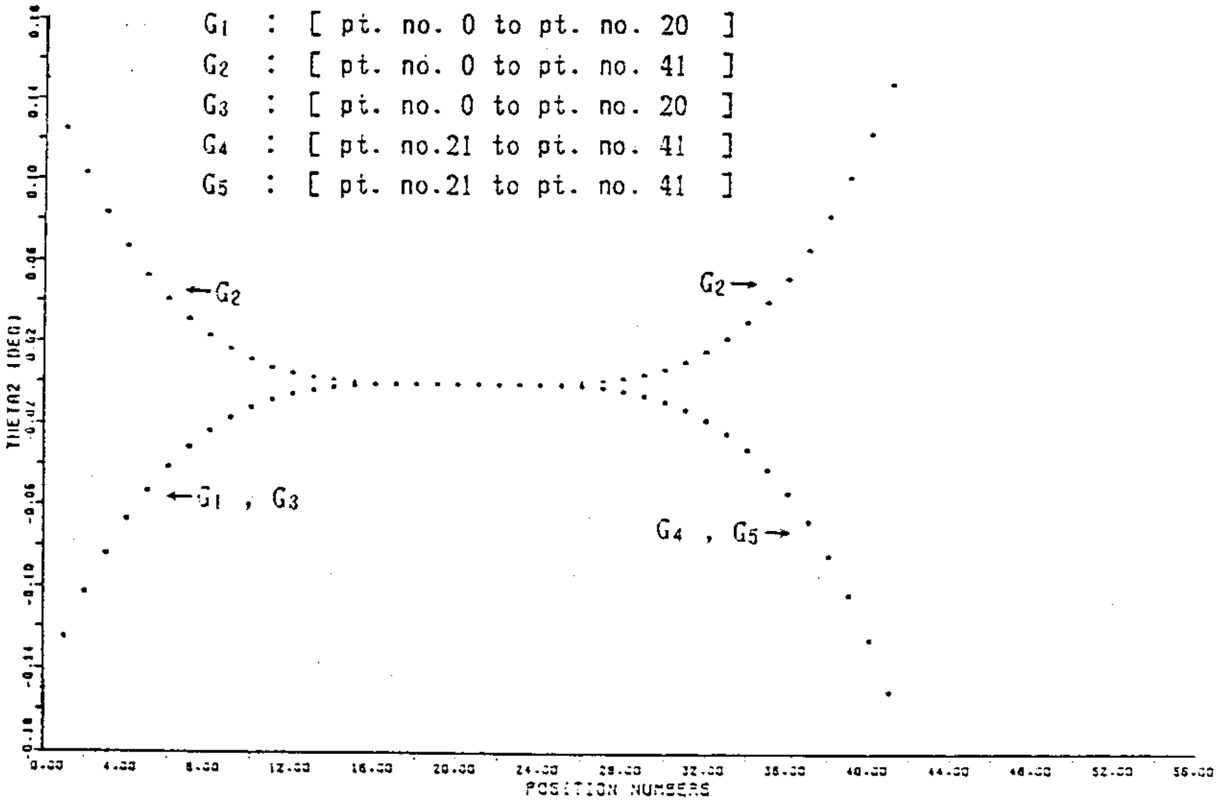
Solution behaviors of Polynomial

(5) Graphics output 2

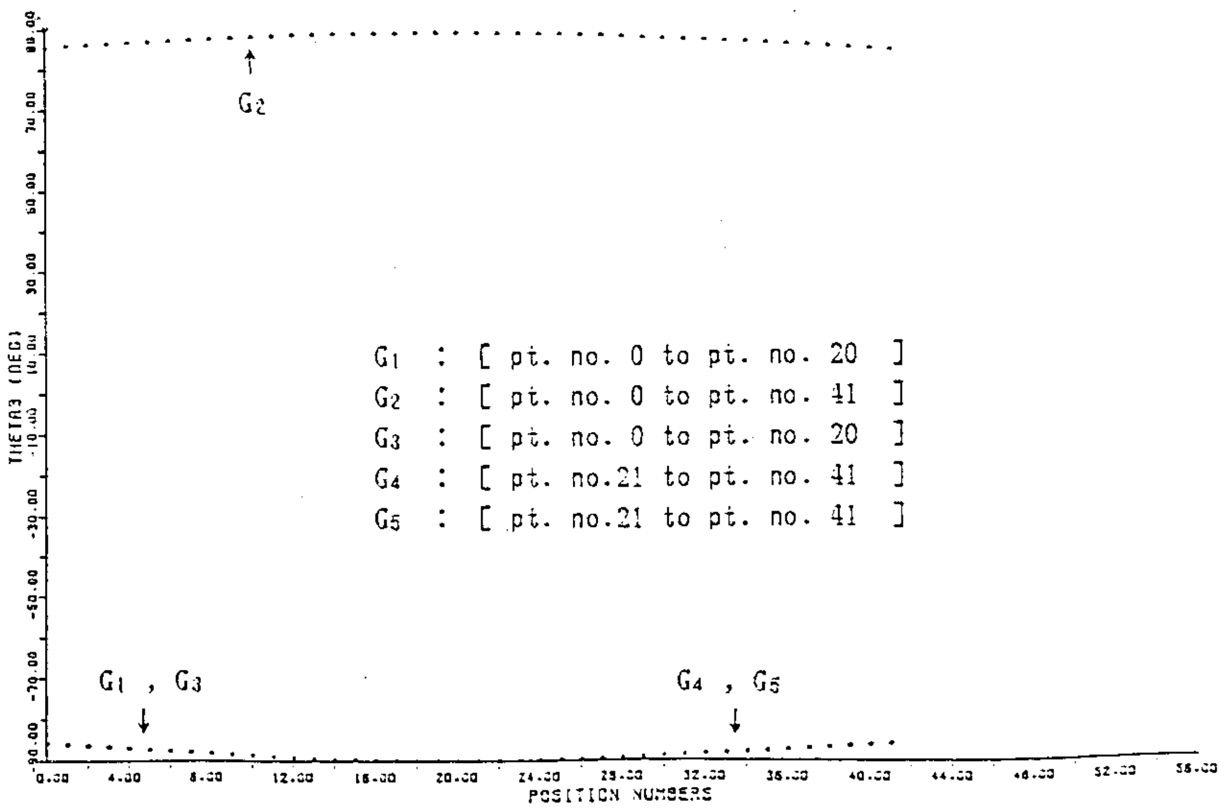
(output of code GRH52)



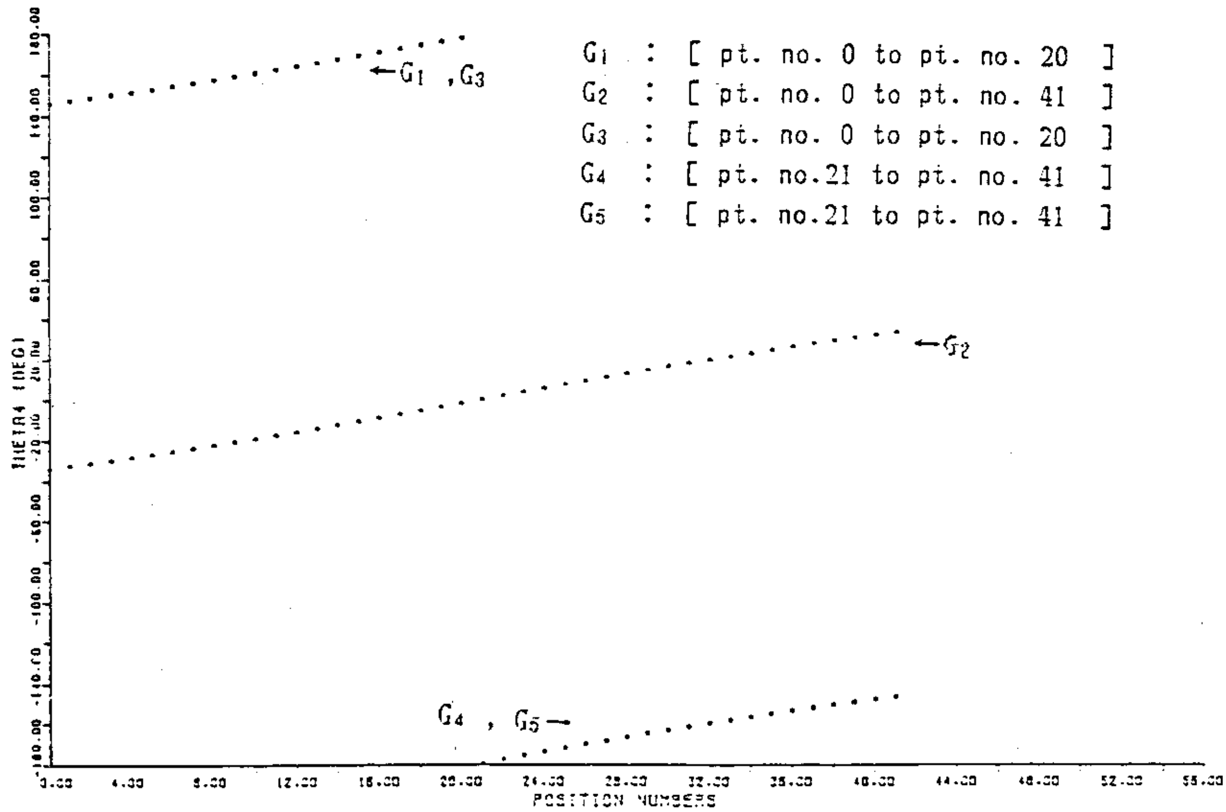
Calculated Results of Joint Angle theta 1



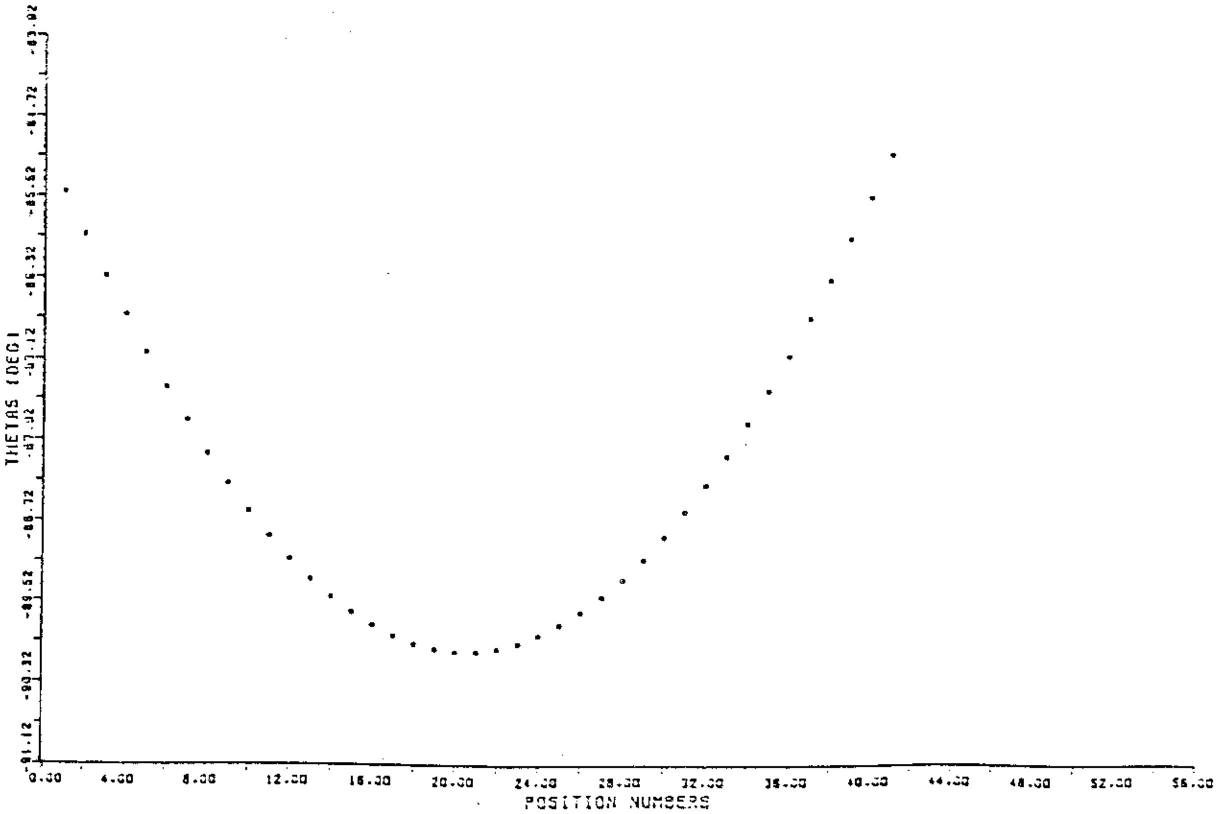
Calculated Results of Joint Angle theta 2



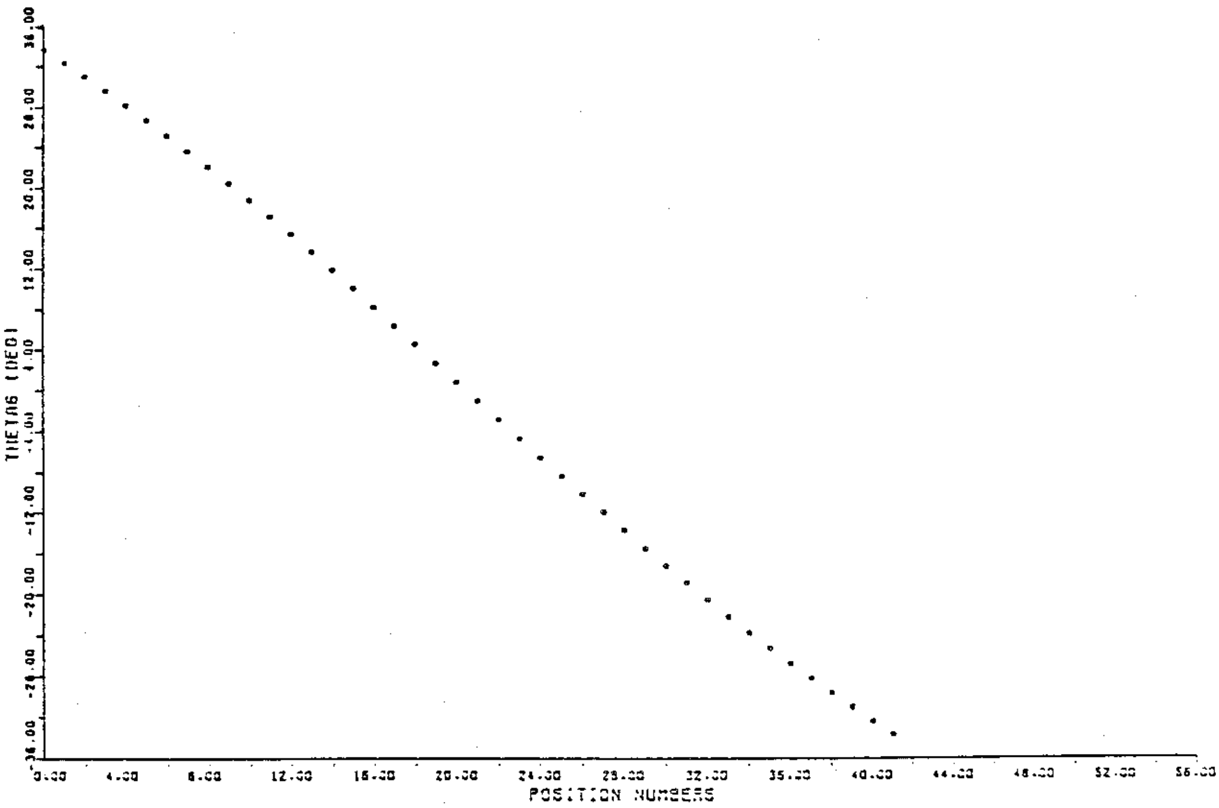
Calculated Results of Joint Angle theta 3



Calculated Results of Joint Angle theta 4



Calculated Results of Joint Angle theta 5



Calculated Results of Joint Angle theta 6

(6) List of input data

```
 INPUT --- TITLE
++++++++++++++++++ T  I  T  L  E ++++++++++++++++++++
+                                                   +
+ +++  BENCH  MARK  2  +++                          +
+                                                   +
+++++++++++++++++++++++++++++++++++++++++++++++++++++

 INPUT --- N

 N      ===>   41

 INPUT --- EPS , EPS1

 EPS    ===>  1.000000D-04    EPS1    ===>  1.000000D-03

---------- INITIAL   POINT ----------


 INPUT --- PX : PY :PZ

 PX ==> -1.000000D-01   PY ==>  3.500000D-01   PZ ==>  1.63100D+00 ( M )


 KEYIN       0 : USER SPECIFIED
             1 : RPY
             2 : EARLY STAGES
     DEFAULTS : EULER

 INPUT --- NX , NY , NZ

 NX ==>  0.0             NY ==>  0.0            NZ ==>  1.00000D+00

 INPUT --- OX , OY , OZ

 OX ==>  1.00000D+00   OY ==>  0.0            OZ ==>  0.0

---------- TERMINAL  POINT ----------


 INPUT --- PX : PY :PZ

 PX ==>  1.00000D-01   PY ==>  3.500000D-01   PZ ==>  1.63100D+00 ( M )


 KEYIN       0 : USER SPECIFIED
             1 : RPY
             2 : EARLY STAGES
     DEFAULTS : EULER

 INPUT --- NX , NY , NZ

 NX ==>  0.0             NY ==>  0.0            NZ ==>  1.00000D+00

 INPUT --- OX , OY , OZ

 OX ==>  1.00000D+00   OY ==>  0.0            OZ ==>  0.0
```