A STUDY OF THE OPTIMAL ALLOCATION OF TOLERANCES AND CLEARANCES IN PLANAR LINKAGE MECHANISMS

by

O. M. A. SHARFI

A thesis submitted to the University of Newcastle upon Tyne for the Degree of Doctor of Philosophy

July, 1982

"Read - in the name of your Lord who has created - created man from a clot. Read - and your Lord is the most generous, who has taught the use of the pen - taught man that which he knew not."

Quran, XCVI, 1-5.

ACKNOWLEDGEMENTS

The author is indebted to Dr. M.R. Smith for his encouragement and guidance throughout this work. Also the author wishes to express his thanks to Professor L. Maunder for permitting the work to be carried out in the Department of Mechanical Engineering, and NUMAC Computing Centre for the use of their computing facilities. Thanks are also due to Mrs. Lynn Whiteford for her patience and accuracy in typing this thesis. Finally the author wishes to acknowledge the financial support of Kartoum Polytechnic, without which this work would not have taken place.

The work described in this thesis has not been submitted to any other University, and consists of original work by the author except where specific reference is made to the work of others.

SYNOPSIS

The work falls into two separate parts, involving respectively kinematic and dynamic aspects of planar linkage mechanisms. The first and major part reported in Part I concerns the development of a procedure for optimal allocation of tolerances and clearances in plane linkage mechanisms. The theory developed takes into account the sensitivity of the mechanism output to small deviations in the parameter dimensions and the cost-tolerance relationships for the parameters. A procedure is then derived from the theory and incorporated into a computer program to allocate tolerances to linear dimensions and angles, and clearances to the joints in the mechanism. To demonstrate the applicability of the method to a wide range of planar linkage mechanisms, a number of examples are given which include 4-, 6-, 8- and 10-bar linkages.

Part II describes the investigation of possible methods for maintaining contact in the joints of a plane four-bar mechanism by means of mass redistribution, the aim being to reduce or eliminate vibration due to impact in joints with clearance. An optimization routine is used with constraints upon the magnitude of the joint forces and the rate at which those forces change direction based on a 'no-clearance' analysis. The method was applied to several examples with little success due to inherent limitations of the analysis method used.

CONTENTS

Page

C H A P T E R O N E

GENERAL INTRODUCTION

The work reported in this thesis is a study of plane linkage mechanisms regarding: (a) the allocation of tolerances and clearances, and (b) the redistribution of link masses to avoid contact loss in the joints. The study involves theoretical analysis and development of procedures which are subsequently incorporated into computer programs.

1.1 ALLOCATION OF TOLERANCES AND CLEARANCES

In the manufacture of parts it is impossible to produce parts having dimensions exactly equal to the nominal dimensions required by a certain design. Consequently a designer dimensions parts specifying tolerable upper and lower limits. For a part which is a component of an assembly, this tolerance on the part dimension depends on the allowable deviation on the overall or output dimension of the assembly.

Part I of this study deals with the allocation of tolerances on the link arc-lengths (i.e. distances between joint centres on a link) and arc-angles (i.e. angles between arcs on the same link), and clearances in the joints, for planar multi-link mechanisms with revolute joints. The tolerances and clearances are allocated such that the deviation in the output due to the tolerances and clearances does not exceed specified limits. Since the sensitivities of the output dimension to changes in arc lengths are not the same for the different dimensions, it follows that allocating the tolerances and clearances indiscriminately would not be wise because this would lead to the tolerances on the dimensions with low sensitivity being tighter than necessary and on the dimensions with high sensitivity wider than required. Thus if all the tolerances are allocated to the tightness required for the more sensitive dimensions, the cost of manufacture would be unnecessarily high due to the extra tightness on the less sensitive dimensions. Similarly relaxing the tolerances indiscriminately would lead to the output dimension deviating beyond the specified limits. Hence for appropriate allocation of tolerances the sensitivities must be taken into account.

Assigning tolerances subject to sensitivities alone, though it leads to lower manufacturing cost, does not insure that the cost will be the minimum possible. This is because the cost-tolerance relationship is not a linear one. Hence for the distribution of the output tolerance among the component tolerances to be an optimum the total cost of the assembly must be reduced to the lowest possible by adjusting the tolerances appropriately.

This is achieved here by minimizing an objective function giving the total cost subject to the constraint that the output deviation due to the allocation of the tolerances, taking into account the sensitivities of the output to individual dimensions, does not exceed the specified limit. The solution is obtained using the Lagrange multiplier method. A procedure is then developed and incorporated into a computer program TOCALM for the allocation of optimum tolerances and clearances. A graph plotting program PSODPLOTS uses output from the above program to plot the sensitivities of the output to individual dimensions and the output deviation band resulting from the allocation of the tolerances and clearances computed against crank input position for the mechanism. Results for a number of example linkages are reported.

1.2 MAINTAINING CONTACT IN JOINTS

The existence of clearances in the joints of mechanisms is inevitable. This leads to the possibility of loss of contact between pairing elements at the joint resulting in impact when contact is remade. Consequently this leads to noise, vibration, wear and possible failure. This problem is becoming more serious due to higher speeds at which modern machinery is required to run. Since clearances are inevitable the only alternative to evade this problem is to insure that contact between the pairing elements at the joints is maintained at all times.

To maintain contact at a joint certain conditions must be satisfied, viz. that the force between the elements of the joint must not approach zero or change direction too rapidly. There are two ways by which these conditions may be satisfied, (a) addition of masses, i.e. counterweights, such that joint forces are made favourable, or (b) attaching springs between the elements at the joints so that the joint forces are modified satisfactorily. Part II of this investigation is concerned with an attempt to satisfy the conditions of maintaining contact in the joints of a four-bar mechanism by optimizing the mass distribution, i.e. method (a) above. The mechanism is analyzed as an ideal mechanism with no clearances, and an objective function is hence developed such that when it is minimized the conditions of maintaining contact are approached. The parameters of the additional masses constitute the optimization variables. The analysis is incorporated into a computer program named CONTACT and a further program named PINFORCE is written to produce polar plots of the joint forces. Results of applying the method to a number of example linkages are reported.

PART I

TOLERANCE AND CLEARANCE ALLOCATION

C H A P T E R   T W O

TOLERANCE AND CLEARANCE ALLOCATION: LITERATURE SURVEY

2.1 INTRODUCTION

Though the treatment of the subject of tolerancing dates back many years, publications on the subject have been scanty. The publications which have appeared may be divided into several categories according to the emphasis on the various aspects of the subject. The works reported here include those categories concerned with the relationship between the overall (or output) tolerance of an assembly and the tolerances on the dimensions of the component parts.

2.2 ASSEMBLY TOLERANCING

There are two philosophies as regards the accumulation of tolerances in an assembly. The first is variously termed absolute tolerancing, sure-fit, complete interchangeability and additive. As the terms imply this method assumes that parts having dimensions on the extremes of the tolerance will be assembled with other parts also at the extreme of the tolerance, at least some, or possibly all, of the time. The second method is variously termed scientific, Pythagorean and statistical. This method is based on the fact that, as shown by Rice [1], parts produced in quantity have their dimensions distributed between the limits of the tolerance zone. As a result the functional dimension of the assembly (or the output deviation) will also be distributed between the limits of the output tolerance. According to the theory of probability, since the parts are randomly selected for assembly, it is extremely rare that parts with extreme dimensions are assembled together.

Acton and Olds [2], Motrecht and Caddell [6], and Brooks [11], assuming normal distributions for the parts dimensions show that the total tolerance of an assembly is more correctly represented by the square root of the sum of squares of the individual tolerances on the part dimensions. If what Acton and Olds [2] term 'natural tolerances', i.e. $\pm 3\sigma$ (three standard deviations), are allocated then only 0.27% of the assemblies will have dimensions outside the tolerance limits. Hence they deduce that statistical tolerancing is more economical because it allows wider part tolerances for a given output tolerance or accumulated assembly tolerance.

Spotts [9] uses the probabilistic law - that the variance of a combination of variables is equal to the sum of the variances of the individual variables - and normal distribution tables to determine the percentages of assemblies which will meet specified tolerances. He gives examples for assemblies of two components, but states that the method becomes more complicated for more than two components.

The additive method of tolerance accumulation may be economical or necessary if the number of parts produced is small or when high precision is involved. Moltrecht and Caddell [5] and Tuttle [10] use this method. Tuttle [10] also gives an output error equal to the root mean square of the part errors, but does not clarify its significance. Knappe [12] also uses the additive method taking into account the sensitivity of the output to individual part dimensions. He also notes that the output dimension of the assembly of randomly assembled parts has a normal distribution even if the part dimensions have rectangular distributions. This fact was also shown by Acton and Olds [2] and Pike and Silverberg [3] to apply if the number of part dimensions involved is more than five.

A number of factors may cause the part dimension to depart from the normal distribution. Gladman [7] and Lorenz [8] report on these factors which result from the relationship between the part tolerance and the properties of the machining process such as the process tolerance and drift tendancies due to e.g. tool wear. Gladman [7] recommends adjusting the statistical tolerancing equation by a 'design factor' to safeguard against distributions that are not normal. Thoen [4] on the other hand recommends a computer simulation technique using punched cards representing frequency of occurrence of part dimensions to determine the distribution of the output dimension.

An important aspect of tolerancing assemblies is the change in dimensions due to storage, operation and wear. This is discussed by Sergeyev [25] who suggests proportioning the deviation in a part dimension into manufacturing error and age error. However this may be allowed for when specifying the assembly tolerance.

2.3 TOLERANCING FOR MINIMUM COST

Statistical tolerancing reduces cost by allowing the tolerances on the part dimensions to be wider. A further reduction is possible if the output or assembly tolerance is distributed among the individual components in a certain way.

Peters [18] reports on three methods, namely (1) distributing tolerances according to the size of each component, (2) according to the standard deviation, and (3) to achieve minimum assembly cost. A fourth method, that of distributing the tolerance equally, is implied by Sharfi and Smith [39].

Minimum cost is achieved by minimizing the total cost of an assembly subject to the constraint that output or assembly tolerance does not exceed a specified value. Pike and Silverberg [3] give a lengthy procedure to achieve minimum cost. A simple graphical method is described by Latta [13] whereby minimum cost is achieved by selecting the individual tolerances by trial and error such that the tangents to the cost-tolerance curves at the selected tolerances are parallel. This method requires cost-tolerance curves to be available and becomes tedious as the number of individual tolerances increases. An analytical equivalent of this method is given by Hillier [14] who uses a simple function to represent the cost-tolerance relationship and hence derives a simple expression for optimum tolerances. Peters [18] gives an involved procedure for assigning tolerances for assemblies with two component dimensions. He states that it becomes more involved for assemblies with more than two components and suggests that in such cases subassemblies should be considered applying the method successively. Simpler methods are reported by Speckhart [21] and Spotts [22], who use approximate cost-tolerance relationships. Speckhart [21] uses an exponential relationship, whereas Spotts [22] uses an inverse square relationship. As reported by Peat [15], cost-tolerance relationships are very complex due to the factors comprising the cost such as time, overhead, inspection, scrap percentage, etc. However, useful approximate relationships may be obtained by curve fitting. For purposes of comparison, even a relationship derived from observing the cost of producing a given tolerance is adequate, Hillier [15]. Sutherland and Roth [27] also use an exponential cost-tolerance relationship to derive an objective function which is minimized subject to a constraint equation comprising output tolerance due to structural and mechanical errors.

2.4 TOLERANCING APPLIED TO LINKAGE MECHANISMS

The problem of tolerancing linkage mechanisms is a special case of assembly tolerancing. In addition to link length tolerances, clearances in the joints of the mechanism need to be considered. Garrett and Hall [16] are perhaps the pioneers in this area-especially with respect to the effects of clearance. They use a statistical approach using a digital computer to simulate random assembly of a sample of 300 mechanisms to obtain what they termed a 'mobility band' within which a given percentage of the mechanism tolerances will lie. This is used as a check for a direct method called the 'delta method' derived from probability theory to determine the mobility band quickly. However, the tolerances and clearances are specified first and do not result from the specification of the output tolerance. Kolhatkar and Yajnik [17] examined the effect of joint clearances, treating them as 'equivalent links'. They considered four- and six-bar function generators, and found that in the four-bar the maximum output error due to clearance effects is obtained when all the clearance links are parallel to the coupler link. The method used is however deterministic. Skreiner and Ebeling [19] use a statistical approach employing Monte Carlo sampling to determine the output distribution for specified tolerances and clearances. In their analysis the clearance effects are included in the frame link length. Lakshminarayana and Narayanamurthi [20] give an analytical method and its graphical equivalent which are used to determine the effect of link length tolerances on the output in a deterministic or additive manner. The graphical method is derived from the velocity diagram and the analytical method is based on loop closure equations. The effects of clearance are not included. Coderman and Mabie [22] developed charts for predicting the maximum mechanical error for various combinations of tolerances, clearances and link length ratios of a four-bar linkage. The errors are equivalent to the maximum clockwise and anticlockwise displacement of the output for given input link position.

A departure from the above methods is that of Dhande and Chakraborty [24] who use a stochastic approach to allocate tolerances and clearances which satisfy a specified output tolerance. Tolerances and clearances are treated as optimization parameters and optimum values are allocated by dynamic programming using an iterative technique. In the analysis tolerances are included with corresponding link lengths and the pin displacement within the clearance space is included in the link length of the preceding link rather than in the more sensitive of the links meeting at the joint. Chakraborty [26] uses a similar analysis to the above, but whereas the above starts at an arbitrary input position, allocates tolerances and clearances, then analyses the mechanism to find the position giving the worst deviation and repeats the process, this method includes the input position as a parameter in the dynamic programming. Sutherland and Roth [27] determine tolerances which result in a minimum cost. An objective function derived from a cost-tolerance relationship is minimised subject to a constraint equation comprising structural error and mechanical error arising from link length tolerances, and a maximum allowable error. Hence the mechanism is synthesized and tolerances on link lengths are allocated simultaneously. This is a significant development in synthesis, however the effects of clearance are not included in the analysis. Dubowsky et al. [28] through minimizing a criterion function equal to the difference between the actual mechanism response and a mathematical model response determine the unknown parameters which comprise link length errors and joint clearances. In the model, clearances are represented by vectors joining centres of mating connection parts, i.e. 'clearance links'. In the procedure used the clearances in all the joints appear to be acting as if combined into the coupler link and it is not possible to determine the clearance in each joint.

Rao [32] outlines an iterative method whereby a mechanism is synthesized taking into account the effects of link deformation, tolerances and joint clearances. However the tolerances and clearances are specified prior to the synthesis of the mechanism. Rao and Reddy [35], using the techniques of chance constrained programming, synthesize mechanisms for minimum structural and mechanical errors treating link lengths (nominal length ± tolerance) and joint clearances as random variables. In the analysis clearances are incorporated into the link length of the link preceding the respective joint, which as noted earlier may not be the more sensitive of the two links connected by the joint.

A variation of the above methods is that used by Horie et al. [36] who use what they term 'transformation functions' (which are open chains held at their extreme positions due to the forces acting in the joints) to determine the limit positions of the output of the mechanism at given input positions taking into account both clearances and tolerances. Choubey and Rao [38] use a method where the mechanical error is treated as a deviation in the structural error. Tolerances are then allocated with reference to the position of maximum error. However, since in the analysis nominal link lengths are determined prior to tolerance allocation, there is no difference between treating the mechanical error as a deviation in either the output or the structural error.

2.5 THE PRESENT WORK

The present work involves the development of a simple method and associated computer programs for the allocation of tolerances and clearances to multi-link plane mechanisms with revolute joints in such a way that the cost due to tolerances is a minimum. It is felt that this will provide a needed tool in the computer aided design of mechanisms. To the best knowledge of the author such a tool is lacking. The program is designed to fit in a package for the design of linkage mechanisms at the Design Unit of the Department of Mechanical Engineering, University of Newcastle upon Tyne.

C H A P T E R   T H R E E

TOLERANCE AND CLEARANCE ALLOCATION: THEORY AND ANALYSIS

3.1 INTRODUCTION

The analysis presented here applies to plane one degree of freedom mechanisms with turning pairs. It does not cater for mechanisms with sliding joints because of the complexity of the error arising from play in sliding joints in the cases where the slide is longer than the guide (as depicted by Fig. 3.1). However the cases where the sliding pair is of the type shown in Fig. 3.2(a) the error due to play is comparable to that of a turning pair (Fig. 3.2(b)).

The output of a mechanism may be measured by a single coordinate - function generator mechanism , or by two co-ordinates in the case of path generators. The analysis is developed for the first case and then extended to include the second.

3.2 CASE WHERE OUTPUT IS DEFINED BY ONE COORDINATE

3.2.1 Relationship between Output Deviation and Parameter Deviations

Consider the mechanism shown schematically in Fig. 3.3. Output is measured by the coordinate ϕ which is a function of the dimensional parameters of the mechanism, and the input to the mechanism, i.e. crank position θ. The parameters of the mechanism consist of the arc lengths 'a' and arc angles 'γ'. In this context an 'arc' is defined as the centre distance of two joints on a link. For a link with more than two joints (ternary, etc.) the angle between two arcs intersecting at a joint is termed an 'arc angle'. Hence for a ternary link the relative positions of the joints are defined by two arcs and the arc angle between the two arcs.

The output ϕ_j corresponding to input position θ_j may be represented as:

$$\phi_j = f(a_{1,1}, a_{2,1} \ \ldots., a_{r,1}, \ldots.., a_{1,\ell}, a_{2,\ell}, \ldots, a_{r,\ell}, \gamma_{1,1}, \gamma_{2,1}, \ldots, \gamma_{r,1}, \ldots.., \gamma_{1,\ell}, \gamma_{2,\ell}, \ldots.., \gamma_{r,\ell}, \gamma_c, \gamma_o, \theta_j) \qquad 3.1$$

The suffixes for the arcs and arc angles refer to the number of arc in the loop and the number of the loop. According to the topology used here, (Oldham [29]) for linkages formed by the addition of one or more dyads (a dyad being a pair of links with a common joint) to a basic unit consisting of a frame and a single input link pivoted to it, the addition of each dyad results in the formation of a new independent loop. The input link pivot is taken as the starting point for each loop for numbering the arcs traversed in the loop. In the first loop the first arc is always the input arc, i.e. $a_{1,1}$. In the other loops, the first arc may be associated either with the input link or the frame. In all loops, the last arc is associated with the frame with the loop closing at the input link pivot. The remaining arcs are numbered consecutively around the loop. Two or more arcs traversed in different loops are regarded as common if they remain at a fixed angle relative to each other during the motion of the mechanism, i.e. they lie on the same link. These fixed angles are termed arc angles. An arc angle is identified by the same suffixes as the arc traversed in the latter loop.

γ_c and γ_o are the input and output reference angles respectively. 'r' is the number of arcs in the largest loop, i.e. with greatest number of arcs, and 'ℓ' is the number of independent loops in the mechanism.

To make the above expression concise, substitute a general parameter array 'ν' for the a's and γ's, hence expression 3.1 may be rewritten as

$$\phi_j = f(\nu_1, \nu_2, \ldots\ldots, \nu_n, \theta_j) \qquad 3.2$$

where $n \leq 2r\ell + 2$

The deviation of the parameters ν_i from their nominal values due to arc length and arc angle tolerances and joint clearances will cause the output to deviate from the desired value corresponding to the nominal dimensions of the parameters. This deviation of the output is known as the 'mechanical error' and is inevitable as are the tolerances and clearances. Hence it is necessary to specify a tolerable output deviation, and, based upon it, the tolerable parameter deviations may be determined.

3.2.2 Statistical Considerations

The tolerable parameter deviations are, of course, the maximum acceptable deviations when the parts are manufactured. However the actual part dimensions will be distributed within the specified tolerance zones. In an assembly, parts are selected randomly and fitted together. The error in the output dimension of the assemblies will hence be distributed within certain limits. The distribution of the output error may be derived from the distributions of the dimensions of the individual parts. The distributions of the part dimensions is determined by the behaviour of the machining process used in the manufacture rather than by the number of parts produced at any one time. Hence statistical considerations are valid even if a small number of parts is produced at a time.

The statistical parameters of the output dimension of an assembly are calculated based on the statistical parameters of the part dimensions and on how these dimensions influence the output dimension, i.e. the sensitivities of the output dimension to changes in the part dimensions. From statistical theory, Haugen [37], the relationship between the standard deviation of the output dimension, σ_T, and those of the individual part dimensions, σ_i, may be expressed as:

$$\sigma_T^2 = \sum_{i=1}^{n} P_i^2 \, \sigma_i^2 \qquad 3.3$$

where P_i is the sensitivity of the output dimension to a change in the i^{th} part dimension.

If the tolerable deviation, δ_i, on a dimension is written as a multiple of the standard deviation, i.e.

$$\delta_i = k_i \, \sigma_i \qquad 3.4$$

where k_i is a constant, then equation 3.3 may be rewritten in the form

$$\frac{\Delta^2}{k_T^2} = \sum_{i=1}^{n} P_i^2 \frac{\delta_i^2}{k_i^2} \qquad 3.5$$

where $\Delta = k_T\ \sigma_T$

Since for most manufacturing processes the distribution is 'normal', an equal confidence level (i.e. proportion of accepted parts of the total parts produced) for each of the parts and the assembly may be assumed. This means that the k's in the above equation will be equal and may be eliminated. The commonly used value is '3', corresponding to 'three standard deviations' or so called 'natural tolerance', which means that 99.73 per cent of the parts produced will be accepted. Thus equation 3.5 becomes

$$\Delta^2 = \sum_{i=1}^{n} P_i^2\ \delta_i^2 \qquad 3.6$$

For a mechanism the influence coefficients, P_i, may vary with input position. Then for any position θ_j of the input equation 3.6 will take the form

$$\Delta_j^2 = \sum_{i=1}^{n} (P_i)_j^2\ \delta_i^2 \qquad 3.7$$

The influence coefficient, $(P_i)_j$, is the sensitivity of the output dimension to changes in the part dimensions and is defined by the partial derivative $\partial\phi_j/\partial\nu_i$.

If the specified allowable output deviation is Δ_* and the maximum influence coefficient for parameter 'i' over the cycle of motion of the mechanism is P_{i*}, then the condition that

$$\Delta_*^2 = \sum_{i=1}^{n} P_{i*}^2 \, \delta_i^2 \qquad 3.8$$

will satisfy the requirement $\Delta_j \leq \Delta_*$, i.e. the output deviation at every position will be within the tolerance limit. Since the P_{i*} do not necessarily coincide at the same position in the cycle of motion of the mechanism, the δ's determined by equation 3.8 are multiplied by a scaling factor (see section 3.4).

3.2.3 <u>Simple Tolerance Allocation</u>

A simple method which readily lends itself to determine the δ's satisfying equation 3.8 is to assume that the output deviation is equally shared among the part dimensions contributing to the output dimension, i.e.

$$P_{i*} \, \delta_i = \tau = \text{constant}$$

Hence, substituting into equation 3.8

$$\Delta_*^2 = n \, \tau^2 \qquad 3.9$$

Then the tolerance for each individual part dimension is given by

$$\delta_i = \frac{\Delta_*}{\sqrt{n}\ P_{i*}} \qquad 3.10$$

Results obtained by applying this simple method are reported in Sharfi & Smith [39] and compare favourably with results obtained by other investigators using much more complex methods (see section 6.2). The method however does not take into consideration the effect of cost. As is readily obvious from equation 3.10 the dimensional tolerance δ_i is inversely proportional to the influence coefficient P_{i*} since Δ_* and n are constant. It follows that for dimensions with high influence coefficients, the tolerances allocated will be very small and therefore very expensive to achieve.

In the following sections a method which takes cost effects into account is derived and thereby tolerances are allocated such that the total cost for producing the parts is a minimum.

3.2.4 Cost Considerations

The cost of manufacturing a machine part depends, among other factors such as material, manufacturing process, etc, on the tolerances on the dimensions of the part. In effect tolerances represent a critical component of the total cost, the more tight the tolerances are the higher will be the cost of the part. Empirical cost-tolerance relationships may be derived from cost tolerance charts for different machining processes. Hillier [14] and Spotts [23] represent the cost C_i of producing a component by the relationship

$$C_i = A_i + \frac{B_i}{\delta_i^2} \tag{3.11}$$

where A_i is a constant comprising the cost of the component excluding the tolerance cost, and B_i is a constant representing the cost of producing the dimension involved to some given tolerance. While Speckhart [21] recommends the relationship

$$C_i = M_i + L_i\, e^{K_i \delta_i} \tag{3.12}$$

where M_i, L_i and K_i are constants determined by curve fitting to cost tolerance data.

Both these relationships represent reasonable approximations to the cost tolerance relationship which is a rather complex relationship owing to the many factors involved, Peat [15]. However, since the objective is one of comparing the relative costs for the parts of an assembly, either of these relationships is adequate for the purpose of the present study. The first relationship, which is simpler to manipulate will be used in the analysis which follows.

3.2.5 Optimized Tolerance Allocation

An optimized tolerance allocation of the dimensions of parts of an assembly is one which, besides satisfying the requirement that the output deviation lies within specified limits, will ensure that the total cost of the assembly will be a minimum.

Using relationship 3.11 above, the total cost C of an assembly is given by.

$$C = \sum_{i=1}^{n} \left(A_i + \frac{B_i}{\delta_i^2}\right) \qquad 3.13$$

Hence for an optimum tolerance allocation C must be made a minimum subject to the constraint given by expression 3.8. Using the Lagrange multiplier method, this condition is satisfied if

$$\frac{\partial C}{\partial \delta_i} + \lambda \frac{\partial \Delta_*}{\partial \delta_i} = 0 \qquad 3.14$$

where λ is the Lagrange multiplier.

Differentiating expression 3.8 and 3.13, and substituting in 3.14 gives

$$-\frac{2B_i}{\delta_i^3} + \lambda \frac{P_{i*}^2 \delta_i}{\Delta_*} = 0 \qquad 3.15$$

The multiplier λ is of no interest and may be eliminated as follows: rewriting equation 3.15 for any other parameter r, i.e.

$$-\frac{2B_r}{\delta_r^3} + \lambda \frac{P_{r*}^2 \delta_r}{\Delta_*} = 0 \qquad 3.16$$

and substituting for λ in 3.15 from 3.16 gives after rearranging

$$\frac{\delta_i}{\delta_r} = \left[\frac{P_{r*}^2 \; B_i}{P_{i*}^2 \; B_r} \right]^{\frac{1}{4}} \qquad 3.17$$

Substituting for δ_i in equation 3.8 gives

$$\Delta_*^2 = \sum_{i=1}^{n} P_{i*}^2 \left[\frac{P_{r*}^2 \; B_i}{P_{i*}^2 \; B_r} \right]^{\frac{1}{2}} \delta_r^2$$

$$= \frac{P_{r*} \; \delta_r^2}{\sqrt{B_r}} \sum_{i=1}^{n} P_{i*} \sqrt{B_i}$$

$$\therefore \quad \delta_r = \frac{\Delta_* \quad B_r^{\frac{1}{4}}}{\sqrt{P_{r*} \sum_{i=1}^{n} P_{i*} \sqrt{B_i}}} \qquad 3.18$$

Hence the maximum tolerable deviations δ_r ($r = 1, 2, \ldots, n$) on the part dimensions giving an optimum total cost may be computed for a given output tolerance Δ_* and tolerance cost constants B_r. The influence coefficients P_r are determined by partial differentiation of expression 3.2.

For complex multi-link mechanism, relationship 3.2 is difficult to compute analytically, and it is therefore more convenient to perform the differentiation numerically. Hence $(P_i)_j = \dfrac{\partial \phi_j}{\partial \nu_i}$ is given by

$$(P_i)_j = \frac{f(\nu_1, \nu_2, \ldots, \nu_i + \varepsilon_i, \ldots, \nu_n, \theta_j) - f(\nu_1, \nu_2, \ldots, \nu_i - \varepsilon_i, \ldots, \nu_n, \theta_j)}{2 \, \varepsilon_i}$$

3.19

where ε_i is the numerical differentiation step. To improve the accuracy and reliability of the numerical differentiation the computation using expression 3.19 is repeated once. In the first instance step values ε_i of the order of manufacturing tolerances are used. In the second instance step values equal to the allowable deviations ($\varepsilon_i = \delta_i$) obtained using the partial derivatives computed in the first instance are used. In fact the computation should be repeated continuously until the step values used and the resulting allowable deviations are equal. However this was found not be to necessary because it was observed that after the second computation the difference between successive evaluations was within $\approx 2\%$.

To safeguard against cases where the gradient $\partial\phi_j/\partial\nu_i$ changes sign across the mean value of parameter ν_i as depicted by Fig. 3.4, and since it is the absolute value of the partial derivative, as the parameter value deviates to either side of the mean, which matters in the present context, equation 3.19 may be rewritten in the form

$$(P_i)_j = \frac{\begin{array}{l} |f(\nu_1,\nu_2, \ldots, \nu_i + \varepsilon_i, \ldots, \nu_n,\theta_j) - f(\nu_1,\nu_2, \ldots, \nu_i, \ldots, \nu_n,\theta_j)| \\ + \, |f(\nu_1,\nu_2, \ldots, \nu_i, \ldots, \nu_n,\theta_j) - f(\nu_1,\nu_2, \ldots, \nu_i - \varepsilon_i, \ldots, \nu_n,\theta_j)| \end{array}}{2\varepsilon_i} \qquad 3.20$$

This in effect gives the average value of the gradients for the positive, $(P_i)_j^+$, or forward deviation from mean, and the negative, $(P_i)_j^-$, or backward deviation. In the present work tolerances are treated as symmetrical about the mean value of a dimension. For asymmetrical tolerancing, if future developments require, equation 3.20 may be separated into two components giving $(P_i)_j^+$ and $(P_i)_j^-$, and the corresponding asymmetrical tolerances δ_i^+ and δ_i^- computed.

3.3 CASE WHERE OUTPUT IS DEFINED BY TWO COORDINATES

In the path generator mechanism shown in Fig. 3.5, the output is the position of the coupler point P. To define the position of P, two coordinates are needed whether cartesian or polar coordinates are used. Hence using cartesian coordinates and an x-y reference frame the output position corresponding to input position θ_j may be defined as

$$\begin{aligned} x_j &= f_x(\nu_1, \nu_2, \ldots, \nu_n \theta_j) \\ y_j &= f_y(\nu_1, \nu_2, \ldots, \nu_n \theta_j) \end{aligned} \qquad 3.21$$

The output deviations in the x- and y-directions are given by

$$\begin{aligned} \Delta^2_{xj} &= \sum_{i=1}^{n} (P_{xi})^2_j \ \delta^2_i \\ \Delta^2_{yj} &= \sum_{i=1}^{n} (P_{yi})^2_j \ \delta^2_i \end{aligned} \qquad 3.22$$

If the allowable deviations in the x- and y-directions are Δ_{x*} and Δ_{y*} the corresponding expressions to 3.8 in the preceding case are

$$\begin{aligned} \Delta^2_{x*} &= \sum_{i=1}^{n} P^2_{xi*} \ \delta^2_i \\ \Delta^2_{y*} &= \sum_{i=1}^{n} P^2_{yi*} \ \delta^2_i \end{aligned} \qquad 3.23$$

Hence, for minimum total cost, the Lagrange multiplier method gives the condition that

$$\frac{\partial C}{\partial \delta_i} + \lambda_x \frac{\partial \Delta_{x*}}{\partial \delta_i} + {}_y \frac{\partial \Delta_{y*}}{\partial \delta_i} = 0 \qquad 3.24$$

Substituting corresponding expressions for the derivatives gives

$$\frac{-2B_i}{\delta_i^3} + \lambda_x \frac{P^2_{xi*}\ \delta_i}{\Delta_{x*}} + \lambda_y \frac{P^2_{yi*}\ \delta_i}{\Delta_{y*}} = 0 \qquad 3.25$$

This equation may only be solved by an iterative technique. The solution was attempted using NAG Library routine E04WAE (71), which incorporates an algorithm for finding a minimum of a function (which here, corresponds to the total cost given by equation 3.13) of several variables (here, the allowable deviations on the dimensions) subject to equality constraints (here, equations 3.23). The routine uses a sequential augmented Lagrangian method. However convergence to a satisfactory solution was not achieved, and moreover, being an iterative process, the solution was found to take more than fifty times that of the analytical solution for the one coordinate system case (section 3.2). The attempt was therefore abandoned in favour of an approximate method devised to convert the problem to one similar to that of the 'one coordinate' case presented in the preceding section 3.2.

Consider the expression

$$\Delta^2_{x*} \doteq \sum_{i=1}^{n} \max\,(P^2_{xi*},\ \rho^2 P^2_{yi*})\delta^2_i \qquad 3.26$$

where $\rho = \Delta_{x*}/\Delta_{y*}$. This expression satisfies both equations 3.23. Thus the two constraint equations have been reduced to a single one, and hence forth, the solution is similar to that given in the foregoing section 3.2.5.

3.4 SCALING FACTOR

In the preceeding analysis the term P_{i*} corresponding to the maximum of $(P_i)_j$ was used. However the P_{i*}'s may not necessarily coincide at one input position. Hence the allowable parameter deviations, δ_i, computed using equations 3.10 or 3.18 will be smaller than necessary. Substituting these computed deviations into equation 3.7 will give the variation of the actual output deviation, Δ_j, over the cycle of motion of the mechanism. Due to the reason mentioned above, the maximum of Δ_j may be smaller than the allowable output deviation Δ_*. Hence the parameter deviations δ_i have to be multiplied by a scaling factor c_f defined by

$$c_{f_1} = \frac{\Delta_*}{\max(\Delta_j)} \qquad 3.27$$

for the case where the output is defined by one coordinate, and

$$c_{f_2} = \text{Smaller}\left\{\frac{\Delta_{x*}}{\max(\Delta_{xj})}, \frac{\Delta_{y*}}{\max(\Delta_{yj})}\right\} \qquad 3.28$$

for the case where the output is defined by two coordinates.

C H A P T E R   F O U R

TOLERANCE AND CLEARANCE ALLOCATION: PROCEDURE FOR ALLOCATION

4.1 INTRODUCTION

The allowable deviation δ_i on a parameter dimension determined by the analysis of Chapter III is the total deviation on the parameter dimension which, for an arc length parameter, comprises the tolerance on the arc length and the associated clearances in the joints at the ends of the arc, as seen from Fig. 4.1; and for an arc angle parameter, comprises the tolerance in the arc angle and the associated clearances in the joints at the far ends of the arcs defining the arc angle as depicted in Fig. 4.2.

In this chapter a procedure is described whereby the allowable deviations δ_i are distributed between the tolerances and associated clearances. The clearance size in a joint has a distribution about a mean which is determined by the distributions of the dimensions of the pin and bore that constitute the joint pair. These dimensions have a 'normal' or Gaussian distribution, (Gladman [7]) and hence the clearance will also have a normal distribution. Hence in distributing the allowable deviation between the tolerances and clearances, the sum of squares law, i.e. the square of the allowable deviation being equal to the sum of squares of the tolerance and the associated clearances, will be applied.

4.2 ARC-LENGTH PARAMETERS

With reference to Fig. 4.1, the effective arc-length $a_{m,r}$ (i.e. arc number 'm' in loop number 'r') may be expressed as

$$a_{m,r} = a^{o}_{m,r} \pm \delta_i \qquad 4.1$$

where $a^o_{m,r}$ is the nominal dimension of the arc-length, and δ_i is the corresponding allowable deviation on this parameter (i.e. arc length $a_{m,r}$ corresponds to the general parameter ν_i - see Chapter II, section 3.2.1). Then applying the sum of squares law gives the relationship

$$\delta_i^2 = t^2_{m,r} + c^2_{\ell,m,r} + c^2_{m,n,r} \tag{4.2}$$

where $t_{m,r}$ is the tolerance on the arc length and $c_{\ell,m,r}$ and $c_{m,n,r}$ are the clearances in the joints connecting arc $a_{m,r}$ to arcs $a_{\ell,r}$ and $a_{n,r}$ respectively.

4.3 ARC-ANGLE PARAMETERS

For the ternary link shown in Fig. 4.2, the effective arc-angle $\gamma_{p,s}$ may be expressed as

$$\gamma_{p,s} = \gamma^o_{p,s} \pm \delta_j \tag{4.3}$$

where $\gamma^o_{p,s}$ is the nominal size of the arc-angle and δ_j is the corresponding allowable deviation on this parameter ($\gamma_{p,s} \equiv \nu_j$). Then, according to the sum of squares law the relationship between the allowable deviation and the tolerance and associated clearances may be expressed as

$$\delta_j^2 = \varepsilon^2_{p,s} + \left(\frac{c_{p,q,s}}{a_{p,s}} \right)^2 + \left(\frac{c_{k,\ell,r}}{a_{k,r}} \right)^2 \tag{4.4}$$

where $\varepsilon_{p,s}$ is the tolerance on the arc angle and $c_{p,q,s}$ and

$c_{k,\ell,r}$ are the clearances in the joints at the far ends of the arcs $a_{p,s}$ and $a_{k,r}$ which define the arc angle (the error in the arc angle due to a clearance being the ratio of the clearance to the respective arc).

4.4 PROCEDURE FOR EVALUATION OF TOLERANCES AND CLEARANCES

Relationships 4.2 and 4.4 above form the basis for the evaluation of the tolerances on the arc-lengths and arc-angles and the clearances in the joints of the mechanism. However to be able to use these relationships a ratio of the clearance component of the total deviation to tolerance component must be assumed, i.e. assume a ratio $\rho_a (= c_{\ell,m,r}/t_{m,r} = c_{m,n,r}/t_{m,r})$ in the case of arc-length parameters, and $\rho_g (= c_{p,q,s}/a_{p,s}\ \varepsilon_{p,s} = c_{k,\ell,r}/a_{k,r}\ \varepsilon_{p,s})$ in the case of arc-angle parameters. The designer may choose suitable values for ρ_a and ρ_g based on experience and on the machining processes used in the manufacture of the parts. Garrett and Hall [16] used a value of $\rho_a = 0.5$, for example. Their choice was perhaps based on the relative ease of achieving tolerances and clearances in a machining process: clearances are affected by tool error only, whereas tolerances are affected in addition to tool error by positioning error.

Having chosen suitable values for the clearance to tolerance ratios, tolerances and clearances are evaluated according to the following steps:

(i) Arrange the allowable arc-length deviations δ_i in an ascending order.

The effect on the output deviation of a given clearance is greatest when the clearance is aligned with the most sensitive of the dimensions with which the clearance is associated. Therefore to minimize this effect a clearance must be allocated based on the allowable deviation of the most sensitive of the dimensions with which it is associated. Any clearance is associated with at least two arc length dimensions. Some may be associated in addition with arc angle dimensions, since there are in general fewer arc angles than arc lengths. The procedure for evaluating tolerances and clearances is hence structured with the above reasoning in mind. Since not all clearances are associated with arc angles only the arc length deviations need to be arranged in a sequence according to value. Situations where a clearance is also associated with an arc length are accommodated when they are encountered, while following the arc length sequence, by checking whether the arc angle dimension is more sensitive (step (iii)) and the tolerances and clearances are allocated accordingly.

(ii) For parameter with the smallest deviation determine the arc length tolerance and associated joint clearances using relationship 4.2 and ratio ρ_a, i.e.

$$t_{m,r} = \frac{\delta_i}{\sqrt{1 + 2\rho_a^2}}$$

and 4.5

$$c_{\ell,m,r} = c_{m,n,r} = \rho_a \, t_{m,r}$$

(iii) If a clearance just computed is also associated with an arc-angle, the arc-angle and associated clearances are determined using relationship 4.4 and ratio ρ_g:

$$\varepsilon_{p,s} = \frac{\delta_j}{\sqrt{1 + 2\rho_g^2}}$$

and 4.6

$$c_{p,q,s}/a_{p,s} = c_{k,\ell,r}/a_{k,r} = \rho_g \, \varepsilon_{p,s}$$

The smaller value for the clearance in question (i.e. the value obtained using the arc-angle deviation and that obtained using the arc-length deviation) is then taken. The computation which gave the higher value is then repeated excluding the said clearance from the relationship. Thus if in the above the value for $c_{k,\ell,r}$ obtained by 4.6 was smaller than that obtained using equations corresponding to 4.5, then the equation for the arc-length deviation would be modified to

$$t_{m,r} = \frac{\delta_i}{\sqrt{1 + \rho_a^2}}$$

i.e. a clearance associated with an arc-angle deviation as well as an arc-length deviation is included with the deviation giving the smaller value for the clearance and excluded from the other.

(iv) The next arc-length deviation is then considered. If any of the associated clearances has been determined in a preceeding computation, it is excluded as above. Hence if one of the clearances has been determined equation 4.5 is adjusted to

$$t_{m,r} = \frac{\delta_i}{\sqrt{1 + \rho_a^2}}$$

and if both clearances have been determined

$$t_{m,r} = \delta_i$$

(v) Go to step (iii) and continue.

Note: In the above procedure the determination of the clearance in a joint is such that it is included in the deviation which gives the smaller value and excluded from the deviations of the other parameters with which the joint is associated. This is done so that (a) double counting is avoided, i.e. including the effect of clearance with more than one deviation, and (b) the possibility that, at worst, the 'clearance link', i.e. the vector between the pin and bore centres, may be aligned with the parameter whose deviation gives the smaller value for the clearance is accommodated.

4.5 TRANSFORMATION OF ARC-ANGLE TOLERANCE

In some situations a designer may find it inconvenient to specify angle tolerances. In such situations the angle tolerances may be transformed into length tolerances along cartesian coordinates. With reference to Fig. 4.3 the tolerances in arc-angle $\gamma_{m,n}$ and arc-length $a_{m,n}$, i.e. $\varepsilon_{m,n}$ and $t_{m,n}$, may be transformed into tolerances in the lengths u and ν as follows:

$$\begin{aligned} u &= a_{m,n} \cos \gamma_{m,n} \\ \nu &= a_{m,n} \sin \gamma_{m,n} \end{aligned} \qquad 4.7$$

Denoting the tolerance in u by t_u and in ν by t_ν then

$$\begin{aligned} t_u &= \left| \frac{\partial u}{\partial a_{m,n}} \right| t_{m,n} + \left| \frac{\partial u}{\partial \gamma_{m,n}} \right| \varepsilon_{m,n} \\ &= \left| \cos\gamma_{m,n} \right| t_{m,n} + \left| -a_{m,n} \sin \gamma_{m,n} \right| \varepsilon_{m,n} \\ &= \frac{u}{a_{m,n}} t_{m,n} + \nu \ \varepsilon_{m,n} \end{aligned} \qquad 4.8$$

$$t_{\nu} = \left| \frac{\partial \nu}{\partial a_{m,n}} \right| t_{m,n} + \left| \frac{\partial \nu}{\partial \gamma_{m,n}} \right| \varepsilon_{m,n}$$

$$= \left| \sin \gamma_{m,n} \right| t_{m,n} + \left| a_{m,n} \cos \gamma_{m,n} \right| \varepsilon_{m,n}$$

$$= \frac{\nu}{a_{m,n}} t_{m,n} + u \, \varepsilon_{m,n} \qquad 4.9$$

As this transformation is an option the designer may or may not opt for, it has not been incorporated in the computer program.

CHAPTER FIVE

TOLERANCE AND CLEARANCE ALLOCATION: COMPUTER PROGRAMS

5.1 INTRODUCTION

Two programs are described in this chapter. Both are written in FORTRAN IV language. The first program, named TOCALM (TOlerance and Clearance Allocation in Linkage Mechanisms), incorporates the theory and analytical procedure developed in Chapters III and IV to determine tolerances and clearances for plane linkage mechanisms with revolute joints. The second, named PSODPLOTS (Parameter Sensitivities and Output Deviation PLOTS), uses the parameter sensitivities and the output deviation due to tolerance and clearance allocation calculated by the above program to produce graph plots of parameter sensitivities and output deviation against the input to the mechanism.

5.2 PROGRAM TOCALM

This program consists of a MAIN routine and twelve subroutines. Fig. 5.1 shows the flow chart for the program and a listing of the program is given in Appendix IA. Three of the subroutines, namely KALM, SETUP and SOLVE are an adaptation of Oldham's [31] program KALM (Kinematic Analysis of Linkage Mechanisms). The following sections give a description of the routines of the program and its operation.

5.2.1 Routine MAIN

This is the controlling unit of the program. The functions it performs are as follows:

(i) It reads the input data which consists of the topological description of the mechanism, dimensions of mechanism parameters, mechanism input positions, tolerance-cost constants and the mechanism output tolerance. A description of the input data and a sample input are given in Appendix IB.

(ii) It stores the topological data in pseudo arrays for use in subroutine RESET.

(iii) It examines the topological data to identify identical and non-identical common arcs. A common arc is one which is on the same link as another on a different loop. If the arc angle between the common arcs is zero and they have the same length then they are termed identical, otherwise, i.e. if the lengths are different or the angle is non-zero, they are termed non-identical.

(iv) It computes the partial derivatives and identifies their maximum values over the cycle of motion of the mechanism.

(v) It computes the allowable parameter deviations.

(vi) It controls the flow of the program and calls the subroutines at specific stages to perform the operations described in the following sections.

5.2.2 <u>Subroutine KALM</u>

This routine is called from MAIN to compute the output positions corresponding to an array of crank input positions and a given set of mechanism parameter dimensions. MAIN calls KALM, first with the dimensions set at their nominal, or mean, values, and subsequently each time a parameter dimension is altered by adding or subtracting the corresponding numerical-differentiation step. The values returned to MAIN are then used to perform the numerical differentiations.

KALM was chosen because it uses a topology capable of handling multi-link mechanisms, and also to enable the Design Unit at the Department of Mechanical Engineering in the University of Newcastle upon Tyne to build up a comprehensive package for linkage mechanism design whose components use the same input data structure.

5.2.3 Subroutine SETUP

This routine is called from KALM to convert the topological description of a linkage into an appropriate form for the next routine SOLVE.

5.2.4 Subroutine SOLVE

SOLVE is called from KALM to solve the kinematics for the linkage and hence compute the output position. In Oldham's [31] program this routine computes displacements, velocities and accelerations. Since only the output position is required for the present investigation, only the statements for computing displacements have been retained.

5.2.5 Subroutine RESET

Within the routine SETUP above, the arrays describing the topology are altered to suit routine SOLVE. Since MAIN calls KALM repeatedly it is necessary to reset the topological description to its initial form. RESET is called from MAIN, prior to subsequent calls to KALM, to reset the topological arrays using the pseudo arrays set up by MAIN.

5.2.6 Subroutine COMARC

For linkages with more than one loop it is possible that there be identical common arcs as defined above. The common arcs will have different sensitivities in the different loops in which they are traversed. Hence it follows that the same arc will have allocated to it different allowable deviations corresponding to the sensitivity values determined for it as an arc in the respective loops.

This routine is therefore called from MAIN to allocate the same allowable deviation to the common arcs. The allowable deviation allocated is of course the smallest of the allowable deviation for the arc in the different loops.

5.2.7 Subroutine CORFAC

In section 3.5 Chapter III it was shown that the allowable deviations computed need to be multiplied by a scaling factor before they are used to determine tolerances and clearances. This routine is called from MAIN to compute the scaling factor, i.e. the ratio of the allowable output deviation to the maximum output deviation resulting from the allocation of the computed allowable parameter deviations, and hence to multiply the parameter deviations by the scaling factor.

5.2.8 Subroutine DEVMAX

This routine identifies the maximum of an array, and is called from CORFAC to identify the maximum output deviation over the cycle of motion of the mechanism.

5.2.9 Subroutine CHECK

It was mentioned in Chapter III section 3.2.5, that the allowable parameter deviations computed in the first instance with partial derivatives obtained using assumed numerical differentiation steps of the order of manufacturing tolerances are recomputed a second time after replacing the assumed steps with the deviations computed in the first instance. This routine is called from MAIN to do the replacements.

5.2.10 Subroutine ORDER

The procedure for evaluating tolerances and clearances from the parameter deviations requires, as described in Chapter IV, that the arc-length parameter deviations be arranged in an ascending order. For this purpose MAIN calls this routine prior to calling RESULT.

5.2.11 Subroutine RESULT

This routine is called from MAIN to determine the arc-length and arc angle tolerances and joint clearances from the allowable parameter deviations. To do so the routine implements the tolerance and clearance allocation steps (ii) to (v) set out in section 4.4 Chapter IV. For every arc encountered a check is made as to whether the clearances in the joints at the ends of the arc have been determined in a preceeding step, and then the tolerance on the arc and the undetermined clearance(s) are computed accordingly using the relationships given in Chapter IV. If the arc has non-identical common arc(s), the joint at which the arcs intersect is identified. Considering the arcs as vectors pointing in the direction in which they are traversed in the respective loops, there are four possibilities: (1) the heads of the two vectors may meet at the common joint, (2) the tails may meet at the common joint, (3) the head of the first with the tail of the second, and (4) the tail of the first with the head of the second. Hence the arc-angle tolerance and the associated clearances are computed.

After determining all the tolerances and clearances the routine prints out the results in tabular form.

5.2.12 Subroutine JSBSCR

A joint in a linkage is identified by three numbers. the numbers of the arcs connected by the joint and the number of the loop in which the arcs are traversed. Whenever a parameter is encountered in RESULT, JSBSCR is called to determine the above mentioned identification numbers for the joint associated with the parameter. The clearance in a joint has the same identification numbers as the joint.

5.2.13 Subroutine TERNRY

This routine is called from routine RESULT whenever a non-identical common arc is encountered. This routine then determines all the other arcs which are non-identically common with the one encountered by RESULT and their arc angles. To do this the routine uses the identification arrays set up by routine MAIN for the non-identical common arcs.

5.3 PROGRAM PSODPLOTS

This program was written to produce graph plots of parameter sensitivities and output deviation against input position. It comprises a MAIN routine and five subroutines. A flow chart for the program is shown in Fig. 5.2 and Appendix IC gives the listing of the program. The following sections give a description of the various routines and the operation of the program.

5.3.1 Routine MAIN

This routine reads the input data and plots the graphs. The input data, which is produced by the preceeding program TOCALM, consists of an array of crank input positions and corresponding output deviations and parameter sensitivities. To draw the graphs MAIN calls appropriate routines from a graph plotting library called GHOST (GrapHical Output SysTem) which comprises a library of a large number of routines to draw lines, curves, characters, etc.

5.3.2 Subroutine SETPLS

This routine is called from MAIN and in turn it calls appropriate GHOST routines to draw axes for plotting the deviation bands of the mechanism output. The routine also writes the title of the plot and labels the axes.

5.3.3 Subroutine PAXES

This routine is called from MAIN and in turn it calls appropriate GHOST routines to draw axes for plotting the parameter sensitivities against crank input. In addition it writes the title for the plot and labels the axes.

5.3.4 Subroutine YSCALE

This routine is called from MAIN to determine the vertical scale for the parameter sensitivity plots. It identifies the maximum of the parameter sensitivity arrays and computes the vertical scale accordingly.

5.3.5 Subroutines SYMKEY and SYMBKY

These routines are called from MAIN to write a reference key for the parameter sensitivity plots. The curves corresponding to the parameters are plotted using a different symbol for each parameter. These routines identify the symbols with the corresponding parameters and write the key. SYMKEY is for arc-length parameters and SYMBKY for arc-angle parameters.

CHAPTER SIX

TOLERANCE AND CLEARANCE ALLOCATION: APPLICATION TO EXAMPLES

6.1 INTRODUCTION

To demonstrate the applicability of the method of tolerance and clearance analysis developed in the preceding chapters to a wide range of planar linkage mechanisms and for purposes of comparison with relevant published work, the following mechanisms were selected.

(i) Four-Bar Sine Function Generator
(ii) Slider-Crank Mechanism
(iii) Four-Bar Motor Cycle Rear Suspension
(iv) Six-Bar Sine Function Generator
(v) Eight-Bar Straight Line Generator
(vi) Ten-Bar Textile Needle Mechanism

6.2 FOUR-BAR SINE FUNCTION GENERATOR

The reason for selecting this mechanism for analysis is to compare the results of the present method with those of Dhande & Chakraborty [24],,and Chakraborty [26]. Garrett and Hall [16] also used this mechanism in their investigation, however, as pointed out in Chapter II, they studied the deviation in output due to arbitrarily specified tolerances and clearances.

The mechanism is shown schematically in Fig. 6.1, with the arc length and arc angle parameters indicated. Table 6.1 shows the results of the analysis printed out by the computer program TOCALM, and Figs. 6.2 to 6.3 show the graphs plotted by the program PSODPLOTS. Figs. 6.2a and 6.2b show the variation of the output sensitivity to parameter deviation (i.e. the partial derivative of the output with respect to each parameter) with input position. The sensitivities to the arc length parameters are approximately of the same magnitude and their maxima coincide around input position $\theta = 65^{o}$. The sensitivity to the output arc-angle γ_0 is, as expected, constant and equal to unity, whereas the sensitivity to the input crank reference position, γ_c, varies greatly with input position.

Fig. 6.3 gives the variation of the output deviation with input position, when the tolerances and clearances in Table 6.1 are allocated to the mechanism. This, in fact, represents the limits within which would lie the deviations of 99.7 per cent (assuming 3σ tolerances) of mechanisms randomly assembled from parts with the above tolerances. When this band is wrapped around the structural error it results in what Garrett and Hall [16] termed the 'mobility band'.

In Table 6.2 the results are compared with those of the references mentioned above. The column labelled 'without cost optimization' was obtained when the effect of cost was not considered and the parameter deviations were assumed to contibute equally to the output deviation, see chapter three section 3.2.3., while the column labelled 'with cost optimization' refers to the results for optimum cost as derived in section 3.2.6 and incorporated in the program TOCALM. The cost coefficients, B_i, were chosen here arbitrarily. (In practice these should be obtained from cost tolerance data.)

The arc length tolerances for the case 'without cost optimization' are approximately the same as those obtained by the quoted references; so are the clearances approximately equal to those of the second reference but equal to half of those of the first reference. Although the author of the second reference was also a co-author of the first and his work is an extension and improvement of that in the first reference, however he did not give any reasons for the difference in the results.

For the cost optimization, the coefficients B_i were chosen to be proportional to the squares of the arc lengths and as a result the tolerances are seen to have been redistributed making the tolerances on the longer arcs higher and vice versa..

A note worth mentioning is the fact that the quoted references did not account for the input reference angle, γ_c, and the output arc angle, γ_o, parameters. This is perhaps the reason for the tolerances obtained here being slightly smaller.

6.3 SLIDER-CRANK MECHANISM

The slider-crank mechanism is the most widely used linkage mechanism. It was selected as an example here to show that, despite the limitation sited in section 3.1, the program can handle linkages with sliding joints. The limitation referred to concerns the fact that the clearance in a sliding joint allows angular rotation of the sliding link relative to its guide in addition to the translational motion (see Fig. 3.1). The present program accounts for the translational motion only. However, since for most slider-crank mechanisms the coupler/

slider joint always remains within the range of the guide (Fig. 3.2), the results obtained here are useful.

Fig. 6.4 shows the mechanism schematically modelled in a form to suit the topological description used in the input data for the program.

The plots of the output sensitivity to changes in the various parameters are shown in Figs. 6.5a and 6.5b. The sensitivity to the connecting rod ($a_{2,1}$) varies very little through the input cycle while the sensitivities to the other parameters are cyclic. For the crank ($a_{1,1}$) the maxima are at 0^o and 180^o of the input position, while for the fixed arc ($a_{4,1}$) and the input reference angle (γ_c) the maxima are at 90^o and 270^o. Since here arc, $a_{4,1}$ has zero length, the allowable deviation in this link is shared by the clearances and no tolerance is allocated to the arc length as the print out of the results Table 6.3 shows.

Fig. 6.6 shows the band within which the output deviation will lie due to the allocation of the tolerances and clearances shown in Table 6.3. The positions of maximum deviations coincide with the maximum sensitivity locations for the input reference, γ_c, and the fixed arc, $a_{4,1}$, i.e. $\theta = 90^o$ & 270^o rather than with that of the crank, arc $a_{1,1}$. This is due to the relative size of the maximum sensitivities and also the cost coefficients, B_i, used which shift the emphasis. Note that the comparison between the sensitivity to the arc lengths (mm/mm) and the sensitivity to the input reference (mm/rad.) is not a direct one.

6.4 FOUR-BAR MOTOR CYCLE REAR SUSPENSION

This linkage was synthesized by Oldham and Fawcett [30] as an alternative to the conventional suspension such that a constant distance is maintained between the gearbox output sprocket and the rear wheel sprocket. The linkage is shown schematically in Fig. 6.7. The coupler point W corresponds to the axis of the rear wheel. It generates a circular path with centre at S which corresponds to the gearbox output axis.

For the purpose of analysis of the linkage, link $a_{1,1}$, is taken as the input link and point W is the output point. The output is measured in polar coordinates. With reference to Fig. 6.7, ϕ_R is the polar radius and ϕ_A is the polar angle. The results of the tolerance and clearance analysis are given in Table 6.4 and Figs. 6.8 - 6.9. The table gives the tolerances and clearances allocated for the specified tolerances of the output in the polar radius and polar angle.

Figs. 6.8a - 6.8c give the output sensitivities to parameter deviations plotted against the input position. In the above reference the authors calculated the variation in the centre distance (i.e. polar radius) when each of the moving links ($a_{1,1}$, $a_{2,1}$, and $a_{3,1}$) is shortened or lengthened by 1 mm. They found the r.m.s. error 0.74 and 0.46 mm for $a_{1,1}$, 0.92 and 0.54 for $a_{2,1}$ and 0.36 and 0.58 for $a_{3,1}$. Compared with Fig. 6.8a(i) there are large differences except for $a_{1,1}$ the difference is relatively smaller. The differences are due to the size of the parameter deviation used to calculate the sensitivity. Whereas in the above reference a parameter deviation equal to 1 mm was used, here the parameter deviations are equal to the tolerances allocated which are (from Table 6.4) 0.089 mm for $a_{1,1}$, 0.425 mm for $a_{2,1}$ and 0.183 mm for $a_{3,1}$.

The output deviation in the polar radius varies from a minimum of 0.23 mm at the initial input position to a maximum of 1.0 mm at the final input position. Judging from the sensitivity curves it is evident that it is influenced most by the tolerances allocated to the output arc angle, γ_0, and the origin arc angle γ_g.

6.5 SIX-BAR SINE FUNCTION GENERATOR

This mechanism, shown schematically in Fig. 6.10, was analysed by the Design Unit [34] and was chosen for analysis here to represent another class of mechanisms and to compare results where possible. The function generated is $\phi = k \sin\theta$, where k is a constant, $0 \leqslant \theta \leqslant 180^o$.

Figs. 6.11 show the output sensitivity curves. As expected the sensitivity for the crank, $a_{1,1}$, varies symmetrically with the input position with maximum around $\theta = 90^o$. (Fig. 6.11a). In Fig. 6.11b, arc $a_{1,2}$, which is identical common with $a_{1,1}$, gives an approximately similar curve, but the value is different since in Fig. 6.11 it is treated as an arc in the second loop. The sensitivity curves for $a_{3,1}$ and $a_{3,2}$ are similar (both being approximately constant and of the same magnitude), as would be judged intuitively due to their orientations. The sensitivity curves for $a_{2,2}$ and $a_{4,2}$ are almost identical whereas that for $a_{2,1}$, is a mirror image of the first two. The sensitivities to the fixed arcs $a_{4,1}$ and $a_{5,2}$ are also similar in shape and magnitude, perhaps due to the symmetrical locations of the fixed pivots. The sensitivities to the arc angle $\gamma_{2,2}$ and the output arc angle γ_o are constant throughout the input positions as expected, while for the input reference, γ_c, the curve is symmetrical with a minimum at $\theta = 90^o$.

Fig. 6.12 shows the output deviation band when the tolerances and clearances given in Table 6.5 are allocated to the mechanism.

The above mentioned reference gives plots of the root mean square (r.m.s.) of output error against parameter deviation. The curves are steepest for arcs $a_{1,1}$, $a_{4,2}$ and $a_{2,2}$, and arc-angles $\gamma_{5,2}$ and $\gamma_{2,2}$ (γ_o not given). The least steep is that for γ_c. There is a general similarity in the results, but a direct comparison is not possible. The reference above quotes $a_{3,1}$ and $a_{3,2}$ as the least critical arc lengths, where here, as noted earlier, their effect on the output is obviously large. The reason is that in the above reference the errors which may be eliminated by adjusting the reference angles were not accounted for, i.e. 'the reference angles were adjusted to obtain the minimum error'.

6.6 EIGHT-BAR STRAIGHT LINE GENERATOR

This mechanism was selected to represent the tolerance analysis of an eight-bar linkage. Table 6.6 gives the printout of the results. Since the mechanism is symmetrical, see Fig. 6.13, the tolerances allocated to the corresponding links are the same (cf. $a_{2,1}$ & $a_{2,2}$; $a_{3,1}$ & $a_{3,2}$). The same is true for the arcs $a_{3,3}$ and $a_{3,4}$, however because they lie in the same loop and have a common joint, and since the clearance is included with the deviation of one of the arcs having the common joint, the tolerances allocated are therefore different. In such situations, to maintain symmetric tolerances, the designer may reallocate the tolerances such that half the clearance is included with each arc.

This symmetry is also expressed by the sensitivity curves, c.f. sensitivity to $a_{2,1}$ and $a_{2,2}$ in Figs. 6.14a(i) and 6.14b(i) respectively for the x-direction and Figs. 6.14a(ii) and 6.14b(ii) for the y-direction, and similarly for the other corresponding arcs that are symmetrical (viz $a_{3,1}$ and $a_{3,2}$ in the Figs. referred to above, and $a_{3,3}$ and $a_{3,4}$ in Figs. 6.14c(i) for the x-direction and Fig. 6.14c(ii) for the y-direction).

Fig. 6.14d(i) could be misleading until one realizes that the vertical ordinates are of the order of 10^{-11}, i.e. effectively zero, as they are beyond the accuracy of the computer ($\approx 10^{-7}$ in single precision). As would be expected the mechanical error in the x-direction is independent of the crank (input) reference angle, γ_c. But in the y-direction the effect of γ_c is substantial as is evident from Fig. 6.14d(ii).

Figs 6.14(i) and 6.15(ii) show the output deviation bands, in the x- and y-directions respectively, due to the allocation of the tolerances and clearances determined in the analysis (Table 6.6). Due to the symmetry of the mechanism, referred to above, the band widths are symmetrical about the mean input position. It is evident that the mechanism output in the y-direction is much more sensitive to parameter deviations than is the output in the x-direction. This may be an advantage where the straightness of the path generated is paramount.

6.7 TEN-BAR NEEDLE MECHANISM

This is a mechanism from the textile industry and was used by Oldham [29] as a case study for dimensional optimization. It is used here to further illustrate the application of the present method of tolerance and clearance analysis to complex linkages. The mechanism is shown schematically in Fig.3.3 and Table 6.7 gives the computer print out of the results of the analysis. Figs. 6.16a to 6.16b show the sensitivity curves. The sensitivities for some of the parameters vary more sharply than others with input position, e.g. for $a_{1,2}$ the sensitivity varies from a maximum of 0.012 rad/mm to a minimum of almost zero whereas that for $a_{4,4}$ remains nearly constant throughout the cycle. The reason is that $a_{1,2}$ is the crank which rotates through 360^{o} while $a_{4,4}$ changes orientation only slightly throughout the cycle.

Fig. 6.17 shows the output deviation band due to the allocation of the tolerances and clearances determined. Comparing this with that of, say the four-bar sine function generator, Fig. 6.3, it can be seen that it has a good feature, i.e. the band width does not vary considerably over the cycle of the mechanism motion. This is due to the fact that unlike the case of the four-bar, the worst sensitivities do not coincide at or near one input position, c.f. Fig. 6.2a and Figs. 6.16 (a-h). This may be used as a design criterion at the synthesis stage of a mechanism.

6.8 COMMENTS

The examples presented give a guide to the scope of applicability of the method. The program handles planar mechanisms with rotary input but with four types of output, viz. rotary (examples (i), (iv) and (vi)), sliding (example (ii)), linear measured in polar coordinates (example (iii), and linear measured in cartesian coordinates (example (v)).

The graphical output gives a very useful picture about the performance of the mechanism and about the critical parameter dimensions. Based on this information the designer may judge whether the mechanism is satisfactory or needs to be revised.

The print-outs of the results also give the execution times for the tolerance analysis. Of course the more complex the mechanism the higher the time. If the output is measured by one coordinate, i.e. rotary or sliding output, the time will be one half of that when the output is measured by two coordinates, i.e. polar or cartesian coordinates of a point. Another factor is of course the number of input positions for which the analysis is made.

P A R T I I

MAINTAINING CONTACT IN JOINTS

C H A P T E R   S E V E N

MAINTAINING CONTACT IN JOINTS:
LITERATURE SURVEY

7.1 INTRODUCTION

The dynamic effects of clearances in the joints of a mechanism have received a growing attention from investigators. Various analytical approaches have been used to predict the response of mechanisms with clearances and a number of experimental investigations have been carried out on joint models and on mechanisms with clearance in one or more of their joints. A few investigators paid particular attention to eliminating impact in the joints through satisfying conditions for maintaining contact. In the following sections these various investigations are reviewed.

7.2 ANALYTICAL INVESTIGATIONS

In the analytical investigations two distinct methods are apparent. Either the response of an idealised one dimensional model is investigated or a specific mechanism is simulated by an analytical model. Goodman [41] replaces the actual system by an equivalent 'box-car diagram' analogous to a train of box-cars, determines the response of the equivalent system and correlates it to the actual system. The box-cars are idealised rigid bodies separated by springs and dashpots. The motion following impact is found on a piece-wise basis.

A two mass model separated by spring and dashpot similar to the box-car diagram is used by Dubowsky and Freudenstein [42]. Response of the model, termed an impact-pair, when subjected to vibrational inputs is formulated using the describing function technique. Results indicate that impact forces increase with clearances. Winfrey [45] also uses an analytical simulation for a one dimensional system with clearances using a cam-driven valve train and an impact damper as examples.

Earles and Wu [46] use Lagrangian equations to simulate a four bar mechanism with a predominant clearance in the coupler follower joint. The clearance is replaced by a massless 'clearance-link' coincident with the joint force direction. The analysis is carried out up to the point of contact loss. Dubowsky [47] gives an analytical simulation to determine the response of a slider-crank-mechanism with clearance at the crank-coupler joint. The results were found to be similar to that predicted by the 'impact-pair' model of reference [42]. Similar results were confirmed by Dubowsky [48].

A different approach which uses momentum exchange to determine impulsive reactions is used by Townsend and Mansour [51, 52] and applied to a four-bar crank rocker with clearance in the coupler-rocker joint. The results give impact spectra and agree with those of other methods. Impact intensity is found to increase with clearance size and speed of input.

To investigate the effect of link elasticity on the joint impact loads, Dubowsky and Gardner [53] studied the response of an "impact beam model" subjected to dynamic excitation perpendicular to the longitudinal axis. It was found that link flexibility reduces the impact forces in the bearings. In a later paper [57] they used Lagrangian formulation to develop dynamic equations for a general elastic clearance system in which link displacements relative to the nominal positions described by the no-clearance analysis were used to write kinetic and potential energy expressions. The analysis was applied to a scotch yoke and the results agreed qualitatively with that predicted by the impact beam model.

Townsend and Mansour [55] used a functional analysis approach to analyse the effects of clearance at the coupler-follower joint of a four-bar mechanism. The time between successive impacts was estimated. Meidema and Mansour [56] used a momentum exchange approach to develop a three mode model: free flight, impact, and 'following' or contact maintained mode. It was found that the zones of contact were compatible with that predicted by the no-clearance analysis.

Yet another approach which employs a vector-network method was used by Rogers and Andrews [58] to simulate a slider crank mechanism. In the analysis the system was treated as rigid bodies interconnected by rotational and translational springs and dashpots and bearing elements. The effects of lubrication and varying stiffness and damping were investigated.

Bahgat et al [64] developed a mathematical model of a four-bar mechanism with clearances inthe frame-crank, crank-coupler and coupler-rocker joints. The links were considered rigid bodies and contact modes were detected by iteration according to the inequalities of input torque and pin forces. The 'clearance links' were aligned with the directions of the respective pin forces and the resulting interdependent kinematic and kinetic equations were solved simultaneously to determine the input torques, pin forces and predict separation occurences.

Haines [70], adopting simplifying assumptions based on deductions from his survey [69], developed equations of motion for a revolute joint using the principle of virtual work. Conditions for loss of contact were hence derived and embodied in a design chart which is in the form of a contour map defining thresholds of contact loss.

7.3 EXPERIMENTAL INVESTIGATIONS

The experimental investigations into the effects of bearing clearances ranged from the measurement of response of mechanisms with clearances in one or more of the joints substantiating analytical models to the development and testing of simple empirical relationships for predicting contact loss in a bearing. Fawcett and Burdess [43] investigated the bearing force in the coupler-rocker joint of a four-bar mechanisms with a predominant clearance in the said joint. The positions of impact correlated to that predicted by no-clearance analysis. The results also showed that impact force increased with clearance size and decreased with compliance of the bearing bush.

Earles and Wu [50, 59-61] used experimental investigations on a mechanism with a clearance in the coupler-rocker joint to evolve and substantiate empirical relationships for predicting occurrence of contact loss and for estimating impact magnitudes. The validity of the empirical relationships was further confirmed by Grant and Fawcett [62], Grant [65], and

Earles and Kilicay [67, 68] using different experimental set ups.

Dubowsky and Young [54] investigated a simple pin connection subjected to sinusoidal input motion, a set up similar to the analytical model known as the 'impact pair' devised by Dubowsky and Freudenstein [42], and found that the experimental results confirmed the behaviour predicted by the analytical model. Such agreement is also reported by Dubowsky and Moening [63] who tested a scotch yoke mechanism and compared the results with that of the analytical model of Dubowsky and Gardner [57].

7.4 MAINTAINING CONTACT

Few investigators have paid particular attention to satisfy conditions for maintaining contact, since if this may be achieved there will be no need for the complex and lengthy analysis to determine the response of systems with clearances beyond the point of contact loss. Fawcett and Burdess [44] suggested that this may be achieved by either of two methods, viz. mass redistribution or spring loading, such that the polar force diagram shape is changed making its contour as far removed as possible from the zero force point. Fawcett [49] described a design of bearing along the principle of force-form closure whereby the bearing is spring loaded in a direction determined from a no-clearance analysis such that the polar force contour is removed far from the zero point.

Earles and Wu [60] suggested modifying the design such that the empirical criterion for maintaining contact

$\dot{\gamma}/R < 1$ (where R is the minimum joint force measured in Newtons and $\dot{\gamma}$ is the corresponding rate of change of direction of R measured in radians per second) is satisfied.

7.5 PRESENT INVESTIGATION

In the present study an attempt is made to maintain contact in the bearings of a four-bar mechanism by addition of masses to the links, i.e. by mass redistribution. An objective function is developed such that minimizing it leads to approaching the conditions for maintaining contact.

C H A P T E R   E I G H T

MAINTAINING CONTACT IN LINKAGE JOINTS: THEORY AND ANALYSIS

8.1 INTRODUCTION

For contact to be maintained between the pairing elements in a joint of a linkage mechanism, intuition suggests that a force must press the elements against each other. However, according to previous investigations [50, 61] , this condition alone will not insure maintaining contact, and another condition, that the force between the elements must not change direction rapidly, has also to be satisfied. These two conditions depend among other factors, on the distribution of the masses on the moving elements of a mechanism. In the following analysis an attempt is made to optimize the mass distribution of a plane four-bar linkage with revolute joints to satisfy the above conditions as closely as possible.

Mass redistribution is also used for balancing purposes, i.e. eliminating the shaking force and shaking moment transmitted to the frame of a mechanism. Hence, using the same method for a different purpose will probably be at the expense of the other. However, this is the designer's problem to decide which one to sacrifice in a particular case.

8.2 OBJECTIVE FUNCTION

To optimize the mass distribution of a mechanism an objective function must be derived such that when this objective function is made a minimum (or maximum, as the case may be), then the mass distribution is optimum. Our objective here is to maintain contact at the joints of a four-bar linkage. Intuition suggests that the ideal situation for maintaining contact between the elements of a revolute joint is the case where a bar rotates in a horizontal plane about a pin which fits in a hole at one end of the bar, see Fig. 8.1. In such a case, with the bar rotating at a constant speed, the force at the joint acts along the axis of the bar. It has a constant magnitude and rotates at a constant rate. Hence, this ideal is the objective to be achieved for the forces in the joints of the mechanism and the objective function will now be formulated accordingly.

Expressions for the forces at the joints of a four-bar linkage are given in Appendix IIB. For clarity these forces are represented graphically in Fig. 8.2 where the external forces have been excluded. In the following analysis the external forces will be assumed negligible in comparison to the inertia forces. The input crank will be assumed to rotate at a constant speed, $\dot{\theta}_1$.

To satisfy the condition of rotating at constant rates, the joint forces will be required to rotate at the same rate as the input crank but may have a phase shift from the crank direction. This requirement is in order since the vector heads of these forces trace a closed loop when the crank rotates through one revolution. Then the direction γ_{ij} of the force vector at any joint may be expressed as

$$\gamma_{ij} = \theta_1 + \zeta_{ij} \qquad 8.1$$

where $i = 1, 2, 3, 4$

$j = 2, 3, 4, 1$

ζ_{ij} = phase shift

Hence the cartesian coordinates of the force vectors may be expressed as

$$F_{ijx} = F_{ij}\cos(\theta_1 + \zeta_{ij})$$

$$F_{ijy} = F_{ij}\sin(\theta_1 + \zeta_{ij}) \qquad 8.2$$

Eliminating F_{ij} from equations 8.2 above gives

$$F_{ijx}\sin(\theta_1 + \zeta_{ij}) - F_{ijy}\cos(\theta_1 + \zeta_{ij}) = 0 \qquad 8.3$$

If the parameters of F_{ijx} and F_{ijy} (see Appendix IIB) and the phase shift ζ_{ij} are adjusted such that equation 8.3 is satisfied throughout the cycle of the linkage, then the condition that the force vector rotates at a constant rate will be satisfied.

The other condition requires that the magnitude of the force remains constant. Taking into account the empirical relationship of Earles and Wu [50] for maintaining contact, i.e. that $\dot{\gamma}/R \leq 1$ - where R is the minimum reaction force in a joint obtained by no clearance analysis (measured in Newtons) and $\dot{\gamma}$ is the corresponding angular velocity (in rad/sec) of the force vector R, the magnitude of the required constant force may be deduced. Allowing for the inevitable variations in the magnitude of the force and the rate at which it rotates we shall write the above relationship as

$$\frac{\dot{\theta}_1}{R_{ij}} = 0.8 \qquad 8.4$$

where R_{ij} = mean value of F_{ij}, i.e making an allowance for a variation of 20% to the limit $\dot{\gamma}/R \leq 1$. Hence the second condition will be satisfied if

$$F_{ij} - 1.25\dot{\theta}_1 = 0 \tag{8.5}$$

is satisfied throughout the cycle of motion of the mechanism.

The expressions on the left hand sides of equations 8.3 and 8.5 form the elements of the objective function. For an optimum, these expressions must have minimum absolute values for all the joints summated throughout the cycle, such that absolute values are considered, the sum of squares will be used. Hence the objective function may be expressed in the form

$$\begin{aligned} f = \sum_{r=1}^{n} & \left[F_{12x} \sin(\theta_1 + \zeta_{12}) - F_{12y} \cos(\theta_1 + \zeta_{12})\right]_r^2 + \left[F_{12} - 1.25\dot{\theta}_1\right]_r^2 \\ & + \left[F_{23x} \sin(\theta_1 + \zeta_{23}) - F_{23y} \cos(\theta_1 + \zeta_{23})\right]_r^2 + \left[F_{23} - 1.25\dot{\theta}_1\right]_r^2 \\ & + \left[F_{43x} \sin(\theta_1 + \zeta_{43}) - F_{43y} \cos(\theta_1 + \zeta_{43})\right]_r^2 + \left[F_{43} - 1.25\dot{\theta}_1\right]_r^2 \\ & + \left[F_{41x} \sin(\theta_1 + \zeta_{41}) - F_{41y} \cos(\theta_1 + \zeta_{41})\right]_r^2 + \left[F_{41} - 1.25\dot{\theta}_1\right]_r^2 \end{aligned} \tag{8.6}$$

where n = no. of positions.

8.3 OPTIMIZATION PARAMETERS

Since the aim is an optimum mass distribution, the parameters of optimization are those parameters which define the mass distribution. For each link these parameters are: the mass, the two coordinates which define the position of the centre of mass, and the moment of inertia. These mass distribution parameters - three for the crank (since joint forces are independent of the inertia of the crank, see Appendix IIB), and four per each of the coupler and follower - plus the four phase shift angles (ζ_{ij}) defining the directions of the joint forces relative to the input arc position - make a total of fifteen optimization parameters.

The mass distribution parameters may be taken as the parameters of the additional masses required to make the distribution an optimum rather than those for the total mass of a link. Fig. 8.3 shows an additional mass m_* with coordinates of its mass centre ρ_* and λ_* added on to a link of original mass m_o and mass centre coordinates ρ_o and λ_o. The mass centre of the total mass is defined by the coordinates ρ and λ. Let k_o, k_* and k define the radii of gyration about the origin, of the original mass, the additional mass and the total mass respectively. The relations between the corresponding quantities follows:

$$\text{Mass:} \qquad m = m_o + m_* \tag{8.7}$$

$$\text{Inertia:} \qquad mk^2 = m_o k_o^2 + m_* k_*^2 \tag{8.8}$$

Centre of mass coordinates:

With reference to Fig. 8.3, let $\bar{x}$ and $\bar{y}$ be the components of ρ relative to the coordinate axes x-y. Taking moments of mass about the respective axes gives:

$$\bar{x} = \frac{m_o\rho_o\cos\lambda_o + m_*\rho_*\cos\lambda_*}{m}$$

$$\bar{y} = \frac{m_o\rho_o\sin\lambda_o + m_*\rho_*\sin\lambda_*}{m}$$

Hence

Radius: $$\rho = \sqrt{\bar{x}^2 + \bar{y}^2} = \frac{1}{m}\sqrt{m_o^2\rho_o^2 + m_*^2\rho_*^2 + 2m_o\rho_o m_*\rho_*\cos(\lambda_* - \lambda_o)} \qquad 8.9$$

Angle: $$\lambda = \tan^{-1}\frac{\bar{y}}{\bar{x}} = \tan^{-1}\frac{m_o\rho_o\sin\lambda_o + m_*\rho_*\sin\lambda_*}{m_o\rho_o\cos\lambda_o + m_*\rho_*\cos\lambda_*} \qquad 8.10$$

8.4 BOUNDARY LIMITS ON OPTIMIZATION PARAMETERS

The angular parameters, i.e. orientations, λ_*, of mass centres for the additional masses on the moving links and the phase shifts, ζ_{ij}, of the joint force directions, obviously need not be bounded. However the remaining parameters need to have boundary limits imposed upon them.

The additional masses, m_*, on each of the moving links, may not be allowed to become negative nor exceed a specified upper limit. This upper limit mainly depends upon allowable forces and available space. Denoting the upper limit by m_u, the boundaries on m_* may be expressed by

$$0 \leqslant m_* \leqslant m_u \qquad 8.11$$

The centres of mass radii, ρ_*, for the additional masses need also not to exceed an absolute upper limit determined by the available space among other factors. If the upper limit is denoted by ρ_u, then

$$-\rho_u \leqslant \rho_* \leqslant \rho_u \qquad 8.12$$

The lower limit $-\rho_u$ is of course the same as the upper limit with an offset of 180^0. It may be argued that since λ_* is not bounded, setting the bounds of ρ_* as $0 \leqslant \rho_* \leqslant \rho_u$ would have been sufficient. However, it was found that setting the bounds as in 8.12 provided an additional freedom during the optimization and a safeguard against being trapped by a fictitious bound.

The radii of gyration, k_*, for the additional masses about the local origins on the links may be expressed in the form

$$k_* = \sqrt{\rho_*^2 + k_{G*}^2} \qquad 8.13$$

where k_{G*} = radius of gyration of the additional mass about its own centre of mass.

The limits on k_{G*} are determined by the allowable dimensions of the additional mass. If m_* is zero k_{G*} is of course immaterial. But with m_* non-zero what will the lower limit on k_{G*} be? This question was answered by Walker [66] . If the additional mass is attached to the link via a frictionless bearing at the centre of mass of the additional mass, then the moment of inertia of the additional mass about its centre of mass is eliminated. Hence, the limits on k_{G*} may be expressed by

$$0 \leq k_{G*} \leq k_{Gu} \qquad 8.14$$

where k_{Gu} is the upper limit.

C H A P T E R   N I N E

MAINTAINING CONTACT IN LINKAGE JOINTS: COMPUTER PROGRAMS

9.1 INTRODUCTION

Two programs are described in this chapter: the first deals with the minimization of the objective function developed in the preceding Chapter VIII, and the second is a graph plotting program for plotting the joint forces. The first program, CONTACT, comprises a MAIN routine and besides the minimization routine E04JBF from the NAG library, subroutines FUNCT, MONIT and RESULT. The NAG library is a library of computer routines which incorporate numerical algorithms for solution of problems in various subject areas. The library has been compiled by the Numerical Algorithm Group, U.K.

9.2 SELECTION OF MINIMIZATION ROUTINE

In search of a suitable optimization routine the NAG FORTRAN Library document on 'Minimizing or Maximizing a Function' [71] was consulted. The NAG Library contains a set of routines to solve a variety of problems. One set is concerned with optimization problems with each routine implementing an algorithm suitable for a certain category of problems. Each category is defined by the property of the function to be minimized- i.e. whether linear, non-linear, quadratic, or sum of squares of linear or non-linear functions - and the restrictions on the values of the optimization variables (or parameters) - i.e. constrained (simple bounds, linear or non-linear) or unconstrained.

The choice of routine depends on several factors. The category of the problem and the level of derivative information supplied are the main factors. The function to be minimized in the present problem is given by expression 8.6, Chapter VIII. It is a summation of squares of non-linear functions. The restrictions on the optimization parameters are simple lower and upper bounds (see section 8.4 Chapter VIII). The first and second derivatives are too complicated to determine analytically. Accordingly the Library routine EO4JBF was selected for the minimization of the function. A description of this routine is given in section 9.3.2.

9.3 PROGRAM CONTACT

A flow chart of this program is given in Fig. 9.1. The component routines of the program are represented by block units. The connecting paths with arrow heads indicate the flow of the program. A listing of the program is given in Appendix IIC, and the component routines are described in the following sections.

9.3.1 Routine Main

This is the controlling unit of the program. It reads in the input data which consists of the dimensional parameters of the four bar linkage, and the initial or starting values of the optimization variables and their lower and upper bounds. It sets the parameters of the optimization routine EO4JBF and prior to calling this routine, it calls Library routine Eo4HBF to supply suitable steplengths for making difference approximations to the partial derivatives of the objective function. After a successful return from the optimization routine MAIN calls routine RESULT to print out the results.

9.3.2 Library Routine E04JBF

As described in the NAG Library document [71] this routine incorporates a quasi-Newton algorithm for finding a minimum of several variables which may be either unconstrained or subject to fixed upper and/or lower bounds. It calls the supplied subroutine FUNCT to evaluate the objective function and calls a host of other library routines to evaluate first and second partial derivatives of the function with respect to the variables and implements the optimization. It also calls subroutine MONIT at a specified frequency to monitor the progress of the optimization process.

The essential feature of the algorithm used - as quoted in the routine document [71] - is that the Hessian matrix $G^{(k)}$ (at iteration step 'k') for position $X^{(k)}$ (where X is the matrix of optimization variables) is used to define the search direction $p^{(k)}$. A cholesky factorization based on $G^{(k)}$ is computed to satisfy

$$L^{(k)} D^{(k)} L^{(k)^T} = G^{(k)} + E^{(k)}$$

where $L^{(k)}$ is a lower triangle matrix with unit diagonal elements, $D^{(k)}$ is a diagonal matrix with strictly positive diagonal elements, and $E^{(k)}$ is a diagonal matrix with non-negative diagonal elements. If $G^{(k)}$ is sufficiently positive definite, $E^{(k)}$ is the zero matrix. The search direction is then obtained by solving the linear system

$$L^{(k)} D^{(k)} L^{(k)^T} p^{(k)} = -g^{(k)}$$

where $g^{(k)}$ is the gradient vector, i.e. vector of first partial derivatives of the objective function $F(X^{(k)})$.

A sequence $X^{(k)}$ is then constructed satisfying

$$X^{(k+1)} = X^{(k)} + \alpha^{(k)} p^{(k)}$$

where $\alpha^{(k)}$ is a steplength chosen such that

$$F(X^{(k+1)}) < F(X^{(k)})$$

$X^{(k)}$ is then replaced by $X^{(k+1)}$ and the process is continued until the convergence criteria are satisfied.

If any variable reaches a bound during the search, it is fixed and the number of 'free' variables is reduced for the next iteration. The components of the search direction corresponding to variables currently on a bound are set to zero.

For a feasible point $\overset{*}{X}$ to be a solution, the following conditions must be satisfied:

(i) $|\bar{g}(\overset{*}{X})| = 0$

(ii) $\bar{G}(\overset{*}{X})$ is a positive definite

(iii) $g_j(\overset{*}{X}) < 0$, X_j at the upper bound

$g_j(\overset{*}{X}) > 0$, X_j at the lower bound

where $\bar{g}(\overset{*}{X})$ is the gradient with respect to the free variables, and $\bar{G}(\overset{*}{X})$ is the Hessian matrix with respect to the free variables.

9.3.3 <u>Subroutine FUNCT</u>

This routine incorporates the equations given in Appendices IIA and IIB and the equation for the objective function given in Chapter VIII. It is called from the Library routines to compute the function value. The optimization variables which are adjusted by the Library routine at each iteration are incorporated into the linkage parameters as defined by equations 8.7 to 8.10 and 8.13 in Chapter VIII. This routine is also called from MAIN after the completion of the optimization to provide the joint forces and the final mass distribution.

9.3.4 <u>Subroutine MONIT</u>

This routine is supplied to monitor the progress of the optimization process. It is called from the Library routine E04JBF at specified intervals to print out:

(i) the value of the function

(ii) the number of function evaluations that has been made so far

(iii) the values of the optimization variables, the first derivatives of the function with respect to the variables, and the status of the variables, i.e. whether free or at a bound,

(iv) the ratio of the largest and smallest elements of the matrix D mentioned in section 9.3.2, which is a good estimate of the condition number of the projected Hessian matrix, and

(v) The Euclidean norm of the projected gradient vector.

The frequency at which MONIT is called is set by the user. The setting may be none, once at the end of the optimization run, or each time a certain number of iterations have been completed.

9.3.5 Subroutine RESULT

This routine is called from MAIN at the completion of optimization to print out the final values of the optimization variables and the resulting mass distribution of the moving links. It also writes into a computer file the joint forces corresponding to an array of crank input positions for subsequent plotting by program PINFORCE.

9.4 PROGRAM PINFORCE

This program was written to produce plots of the joint forces. It consists of a MAIN routine and a subroutine SKALE, and graph plotting routines from the Library file *GHOST. Fig. 9.2 gives a flow chart for the program and a listing of the program is given in Appendix IID.

Routine MAIN reads the data stored by the preceeding program CONTACT which gives the cartesian coordinates of the joint forces corresponding to an array of input positions. For the plot at each joint MAIN calls SKALE to determine the scales of the plot to suit the extreme values of the joint force coordinates. MAIN then calls appropriate routines from *GHOST to produce the plot.

CHAPTER TEN

MAINTAINING CONTACT IN LINKAGE JOINTS: APPLICATION TO EXAMPLES

10.1 INTRODUCTION

The method of optimizing mass distribution to maintain contact in the joints of four-bar linkages developed in the preceding Chapter VIII and incorporated into the computer program, Chapter IX, was applied to three examples:

(1) a crank-rocker with a relatively poor transmission angle,

(2) a crank-rocker with a better transmission angle, and

(3) a drag-link.

The first example was chosen arbitrarily to test the validity of the method, but the other examples were selected to demonstrate the effect of (a) the transmission angle, and (b) a rotating follower as against a rocking follower. The results are presented in the following sections.

10.2 CRANK-ROCKER WITH RELATIVELY POOR TRANSMISSION ANGLE

This linkage was, as mentioned above, chosen arbitrarily, and it happened that its transmission angle - with extreme values of $\approx 29^{0}$ and 98^{0} - was relatively poor at the lower end. The parameters of this linkage are given in Table 10.1. The mass parameters are in accordance with the definitions in Fig. 8.3. Suffix 'o' refers to the original mass, i.e. before addition of mass to optimize the distribution, and '$\underline{*}$' refers to the added mass to achieve an optimised mass distribution. The moments of inertia I_0 and I_* are about the local origin on each link.

Table 10.2 gives the print out of the computer program CONTACT. It shows the initial values of the optimization variables and objective function, and the respective final values at the end of the optimization run. Figs. 10.1 and 10.2 give joint force plots produced by program PINFORCE. These are for the cases with the added mass parameters set to: (i) zero, and (ii) optimized values respectively. It is evident that the optimization has improved the joint forces both in magnitude and in the rate of rotation. This is especially true for the frame-crank and crank-coupler joints. Obviously this is due to the fact that they are the joints at the ends of the crank which rotates at a constant rate. However, at the other two joints the kinematic parameters of the mechanism did not allow much improvement. It must be noted that the optimization routine may not achieve a global minimum, but coverges to the local minimum in the vicinity of the starting point (i.e. initial values of the optimization variables). Hence the choice of the starting point is very crucial though it is subject to trial and error.

10.3 CRANK-ROCKER WITH IMPROVED TRANSMISSION ANGLE

The frame arc of the preceding example was adjusted so as to give a linkage with better extreme values for the transmission angle. The extreme values of the transmission angle for this linkage are $\approx 49^{\circ}$ and 132° (c.f: 29° & 98° for the preceding). The linkage parameters are given in Table 10.3.

The results of the optimization run are shown in Table 10.4, giving the values of the optimization variables at the starting point and the final point.

Fig. 10.3 gives the joint forces when only the original masses of the links apply. The effect of transmission angle is evident even before optimization. Though the mass parameters are identical, the joint forces are very much improved with respect to variation in magnitude and rate of change of direction. The reason for this is explained by the equations for the forces (see equations IIB.4 in Appendix IIB) and also by the vector representation of the force components in Fig. 8.2, where it is seen that two components of the force vectors have '$\sin\mu$', i.e. the sine of the transmission angle, in their denominators. Fig. 10.4 shows the joint forces after optimizing the mass distribution. The starting values for the optimization variables were the same as for the preceding example - c.f. Tables 10.2 and 10.4. Comparison of Fig. 10.4 with Fig. 10.2 for the preceding example shows that the result of optimization for the linkage with the better transmission angle is slightly better. This is especially true for the frame/crank and crank/coupler joints where the rate of rotation of the joint force is more uniform, c.f. around the points corresponding to 0^{o} crank angle. For the other two joints, though the response to the optimization is comparatively better than for the preceding example, however the optimization did not achieve the required result.

A comparison of the results print out, Tables 10.2

and 10.4, shows that for this example the function value reduced from 0.24539 to 0.18745 - a reduction of 0.05794 (23.6%), and for the preceding example from 0.27801 to 0.23297 - a reduction of 0.04504 (19.3%). The order of the actual function given in Chapter VIII equation 8.6 is 10^5 times but had to be scaled down as required by the optimization routine: that the value is of the order of unity.

It is thus obvious that the optimization failed to reduce the objective function value substantially. This is due to the limitations in the problem itself and possibly in the formulation of the objective function. These limitations will be discussed in more detail in Chapter XI.

The function values given above again emphasize that improving the transmission angle has made the response of the problem to optimization slightly better.

10.4 DRAG-LINK

This linkage is an inversion of the crank rocker with the improved transmission angle, the crank and frame have interchanged roles. It was chosen here as an example because it was felt that in the case of the crank-rocker the fact that the follower does not rotate continuously may have hampered the optimization. The parameters of the linkage are given in Table 10.5, and the result of the optimization run is given in Table 10.6. Figs. 10.5 and 10.6 show the joint-force plots prior to and after the addition of masses to optimize the mass distribution respectively.

Comparing the force plots to optimization to the corresponding ones for the crank-rocker (c.f. Figs. 10.3 and 10.5), it is evident that the drag link has a more favourable joint force behaviour. This is especially so for the coupler/ follower and follower/frame joints.

However the results of the optimization, though they have approached the objective a little further than in the previous examples, still fell short of producing satisfactory joint forces, for the coupler/follower and follower/frame joints. It is possible that a more favourable starting point may have produced a better result. Table 10.6 shows a reduction in the objective function value from 0.20688 to 0.17103, a difference of 0.03585 (17.3%). Though the minimum function value is smaller than for the preceding examples, however, the reduction achieved is also less. This again emphasizes that the optimization may have been trapped into a local minimum by an unfortunate choice of the starting values for the optimization variables.

10.5 COMMENTS

The results for the above examples show that mass distribution may be used as a tool for achieving joint force conditions for maintaining contact. However it is evident that it has several drawbacks. The magnitude of the force is increased by an order of three or more which means more wear in the bearings and possible failure. For the joints not associated with a continuously rotating link the geometric properties make it impossible to achieve a force locus comparable to that of the ideal case of a 'rotating bar'. For the other joints these effects are overcome by increasing the mass of the rotating link alas at the expense of increasing the magnitude of the force. The transmission angle has a noticeable effect, a better transmission angle results in a better joint force with respect to rate of change of direction.

Though the results presented are for one starting point for each example, several other starting points were tried and the results were worse than those presented. To continue trying alternative starting points in quest of better results is very much time consuming. It is felt that further investigation is necessary for choosing suitable starting points.

CHAPTER ELEVEN

DISCUSSION, CONCLUSIONS AND SUGGESTIONS FOR FURTHER WORK

11.1 TOLERANCE AND CLEARANCE ALLOCATION

11.1.1 Discussion

It has been possible to use the simple statistical relationship $\Delta^2 = \Sigma P_i^2 \delta_i^2$ (relating the output deviation Δ to the deviations of the component dimensions δ_i taking into account the sensitivity coefficients P_i) since the critical dimensions involved are not subject to drift in the manufacturing process. These dimensions are the joint centre distances and the machining operation is simplyboring of holes. The distributions of the component dimensions will therefore be nearly normal and since the operation is the same for all the dimensions involved, the standard deviationswill be nearly equal.

In determining the allowable deviations on the component dimensions the peak values (P_{i*}) of the sensitivity coefficients were used rather than the r.m.s. values. This was done because the concern here is the allocation of tolerances and clearances such that the output deviation does not exceed the specified limits at any time during the cycle of motion of the mechanism.

The computer program has been debugged and tested using a representative sample of the linkage mechanisms it is designed to handle, viz. planar multi-link function and path-generator mechanisms obtained by the addition of one or more dyads to a basic unit consisting of a frame and an input link pivoted to it. Though no comparable programs exist against which the efficiency could be measured, the execution times (see computer print outputs, Chapter VI) for the example problems may indicate such a measure especially in comparison with methods which use iterative optimization techniques.

The results are presented such that they may be directly related to the input data and the sketches from which the input data is compiled. It was intended to give the tolerances and clearances in terms of the ISO standard tolerance grades, however this would require, in the case of clearances, providing the nominal dimensions of the pins and bores at the joints. These dimensions are determined based on the loading on the mechanism. Such an analysis is beyond the scope of the present investigation.

The graphical output gives a very useful picture of the variations of the sensitivity coefficients and the output deviation throughout the cycle of motion of a mechanism and provides clues for possible improvements of the mechanism. It shows which parameters are critical and at which positions along the cycle.

11.1.2 Conclusions

A simple and fast method for allocating tolerances and clearances in a wide range of plane linkage mechanisms has been developed. The method has been incorporated into a computer program which will fill a gap in the computer aided design of linkage mechanisms. The program uses input data structured in the same way as that for the linkage mechanism synthesis and analysis programs at the Design Unit, University of Newcastle with a minimum of additional parameters, viz. cost-tolerance relationship constants and output tolerance specifications.

The results obtained using this program compare satisfactorily with those of other methods (see Chapter VI). These methods treated four-bar linkages only, whereas the present method is more general in that it is directly applicable to multi-link mechanisms.

11.1.3 Suggestions for Further Work

The method presented here caters for plane mechanisms with revolute joints. An area of further work is extending the method to cater for mechanisms with sliding joints where the effect of relative rotation of the sliding pair elements due to presence of clearance (see Chapter III) is accounted for.

Another area of further work, though not directly related to the present investigation, is the developing of a method for determining the shapes and dimensions (other than arc lengths) of the links of a mechanism based on the loading and other functional requirements. Availability of information produced by such a method would allow the present method to give the results in terms of standard tolerance grades.

11.2 MAINTAINING CONTACT IN JOINTS

11.2.1 Discussion

The analysis presented is simple and straightforward. The boundary limits imposed on the optimization variables plus the limitations due to arc lengths proportions which dictate the kinematic (or geometric) properties of the linkage may be among the reasons which prevent the optimization run from converging to a near ideal result, i.e. joint forces similar to that for a bar rotating about a pin (Chapter VIII) at constant angular velocity. Nevertheless the results for the examples (Chapter X) show a definite tendency to approach the ideal situation, especially for the frame/crank and crank/coupler joints. The fact that the results for the linkage with the better transmission angle and for the drag link approach the ideal more closely than does that for the linkage with the poor transmission angle supports the argument that the kinematic properties of the linkage have a great influence on the results.

The objective function used may have been too restrictive since the ideal conditions set (i.e. constant force and constant rate of rotation) are impossible to achieve. This is most severe for the case of a crank rocker where the joint forces for the coupler/follower and follower/frame joints have, by virtue of the geometric properties, restricted changes in direction. The objective function used here is therefore not suitable and need to be changed to take this into account. An alternative may be used as objective function such as

$$f = \sum_{i=1}^{m} \sum_{j=1}^{n} F_{ij}^2$$

where F_{ij} is the magnitude of the joint force, for joint 'i' at an input crank position 'j'

m is the number of joints in the mechanism

n is the number of input positions,

and minimizing the function subject to inequality constraints

$$\frac{\dot{\gamma}_{ij}}{F_{ij}} < 1$$

where $\dot{\gamma}_{ij}$ is the rate of change of direction of F_{ij}.

Since, for a given linkage, the kinematic properties may not be altered at this stage of linkage design, either of two courses of action may be taken if the result is not satisfactory. The first is to find a different linkage, e.g. a cognate linkage, which may have more favourable kinematic properties. The second is to achieve the conditions for maintaining contact by an alternative method such as the force-form closure described by Fawcett [49].

Another factor which decides the success of the optimization is the choice of the starting point (i.e. setting the initial or starting values of the optimization variables). This is crucial because it is most likely that the optimization converges to the local minimum nearest to the starting point. This means that several starting points must be tried which can be cumbersome when the number of variables is large. The choice of the starting point seems to be even more critical in the case of the angular optimization variables where the objective function is a function of the sines or cosines of these variables. Due to this the objective function will be sinusoidal and hence more prone to be trapped into a local minimum as regards the angular variables.

11.2.2 Conclusions

The results have shown that mass re-distribution may be used to approach conditions for maintaining contact in the joints of a four bar mechanism. How far these conditions are approached depends on the kinematic properties of the linkage and on the choice of the starting point.

11.2.3 Suggestions for Further Work

Further work is required to refine the method and to develop a method for choosing a suitable starting point. The objective function formulation may need to be revised. In the present method it was formulated around the ideal situation of a constant joint force R rotating at a constant rate $\dot{\gamma}$ which is impossible to achieve at the joints not associated with links rotating at constant angular velocities. This may still need to be used as a reference, but restructuring the function and/or using an alternative optimization algorithm need further investigation.

Eventually the investigation needs to be extended to cover multi-link mechanisms.

R E F E R E N C E S

REFERENCES

PART I

1. W.B. Rice
Setting Tolerances Scientifically.
Mechanical Engineering, Dec. 1944, pp. 801-803.

2. F.S. Acton and E.G. Olds
Tolerances-Additive or Pythagorean.
Industrial Quality Control, Vol. 5, Nov. 1948, pp.6-12.

3. E.W. Pike and T.R. Silverberg
Assigning Tolerances for Maximum Economy.
Machine Design, Vol. 25, Sept. 1953, pp.139-146.

4. R.L. Thoen
An Easy Approach to Statistical Tolerancing with Punched-Card Computers.
Machine Design, vol. 29, June 13, 1957, pp.121-123.

5. K.H. Moltrecht and R.M. Caddell
How to Determine Production Tolerances. Part One - Complete Interchangeability.
The Tool Engineer, Vol.39, Oct. 1957, pp.81-85.

6. K.H. Moltrecht and R.M. Caddell
How to Determine Production Tolerances. Part Two - Statistical Methods and Selective Assembly.
The Tool Engineer, Vol. 39, Nov. 1957, pp. 85-89.

7. C.A. Gladman
Techniques for Applying Probability to the Tolerancing of Machined Dimensions.
C.S.I.R.O., Aust. Nat. Stand. Lab., Tech. paper No.11

8. G. Lorenz
Dimensional Analysis of Production Processes.
C.S.I.R.O., Aust. Nat. Stand. Lab., Tech. paper No.13.

9. M.F. Spotts
An Application of Statistics to the Dimensioning of Machine Parts.
Trans. ASME, J. Eng. Ind., ser. B, Vol. 81, Nov. 1959, pp. 317-222.

10. S.B. Tuttle
Error Analysis.
Machine Design, Vol. 32, June 9, 1960, pp. 152-158.

11. K.A. Brooks
How to set up and Coordinate a Statistical Dimensioning Program.
Machine Design, Vol. 33, Sept. 14, 1961, pp. 140-145.

12. L.F. Knappe
A Technique for Analyzing Mechanism Tolerances.
Machine Design, Vol. 35, April 25, 1963, pp.155-157.

13. L.W. Latta
Least-Cost Tolerancing.
Product Engineering, Vol. 34, Sept. 16, 1963, pp. 111-113

14. M.J. Hillier
A Systematic Approach to the Cost Optimization of Tolerances in Complex Assemblies.
Bull. Mech. Eng. Educ., Vol. 5, 1966, pp. 157-161.

15. A.P. Peat
Cost Reduction Charts for Designers and Production Engineers.
The Machinery Publishing House, U.K., 1968.

16. R.E. Garrett and A.S. Hall
Effects of Tolerance and Clearance in Linkage Design.
Trans. ASME, J. Eng. Ind., ser. B, vol. 91, Feb. 1969, pp.198-202.

17. S.A. Kolhatkar and K.S. Yajnik
The Effects of Play in the Joints of Function Generating Mechanisms.
J. Mechanisms, Vol. 5, 1970, pp. 521-532.

18. J. Peters
Tolerancing the Components of an Assembly for Minimum Cost.
Trans. ASME, J. Eng. Ind., ser. B, Vol. 92, Aug. 1970, pp. 677-682.

19. M. Skreiner and D.G. Ebeling
Tolerance Analysis of Mechanisms Using PA-300 : A General Probabilistic Problem Solving Language.
ASME paper No. 70-MECH-44.

20. K. Lakshminarayana and R.G. Narayanamurthi
On the Analysis of the Effects of Tolerances in Linkages.
J. Mechanisms, Vol. 6, 1971, pp. 59-67.

21. F.H. Speckhart
Calculation of Tolerance Based on a Minimum Cost Approach.
Trans. ASME, J. Eng. Ind., ser. B, Vol. 94, May 1972, pp.447-453.

22. D.G. Coderman and H.H. Mabie
Development of Design Charts for Use in Predicting the Effect of Tolerances and Clearances on Mechanical Error in Four-Bar Linkages.
ASME Paper No. 72-MECH-7.

23. M.F. Spotts
Allocation of Tolerances to Minimize Cost of Assembly.
Trans. ASME, J. Eng. Ind., ser. B, Vol. 95, Aug. 1973, pp.762-764.

24. S.G. Dhande and J. Chakraborty
Analysis and Synthesis of Mechanical Error in Linkages - A Stochastic Approach.
Trans. ASME, J. Eng. Ind., Ser. B, Vol. 95, Aug. 1973, pp.672-676.

25. V.I. Sergeyev
Methods for Mechanism Reliability Calculation.
Mechanism and Machine Theory, Vol. 9, 1974, pp. 97-106.

26. J. Chakraborty
Synthesis of Mechanical Error in Linkages.
Mechanism and Machine Theory, Vol. 10, 1975, pp. 155-165.

27. G.H. Sutherland and B. Roth
Mechanism Design : Accounting for Manufacturing Tolerances and Costs in Function Generating Problems.
Trans. ASME, J. Eng. Ind., ser. B, Vol. 97, 1975, pp. 283-286.

28. S. Dubowsky, J. Maatuk and N.D. Perreira
A Parameter Identification Study of Kinematic Errors in Planar Mechanisms.
Trans. ASME, J. Eng. Ind., ser. B, Vol. 97, May 1975, pp. 635-642.

29. K. Oldham
The Kinematics and Vibration of Planar Linkage Mechanisms.
Ph.D. Thesis, University of Newcastle upon Tyne, 1977

30. K. Oldham and J.N. Fawcett
Computer-Aided Synthesis of Linkages - A Motorcycle Design Study.
Proc. Instn. Mech. Engrs., Vol. 190, 1977, pp. 713-720.

31. K. Oldham
KALM-Kinematic Analysis of Linkage Mechanisms.
Internal Report Ta 47, June 1977, Dept. Mech. Eng., University of Newcastle upon Tyne.

32. A.C. Rao
Improved Method for the Design of Link Mechanisms.
Indian J. Technology, Vol. 16, April 1978, pp. 145-147.

33. O. Björke
Computer-Aided Tolerancing.
Tapit Publishers, Norway, 1978.

34. Design Unit
Sine Function Generator.
Report, Sept. 1978, Design Unit, Dept. Mech. Eng., University of Newcastle upon Tyne

35. S.S. Rao and C.P. Reddy
Mechanism Design by Chance Constrained Programming Techniques.
Mechanism and Machine Theory, Vol. 14, 1979, pp. 413-424.

36. M. Horie, H. Funabashi, K. Ogawa and H. Kobayashi.
A Displacement Analysis of Plane Multi-link Mechanisms with Clearances and Tolerances.
Bull. JSME, Vol. 23, No. 183, Sep. 1980, pp. 1522-1529.

37. E.B. Haugen
Probabilistic Mechanical Design.
John Wiley and Sons, U.S.A., 1980, pp. 59.

38. M. Choubey and A.C. Rao
Synthesizing Linkages with Minimal Structural and Mechanical Error Based upon Tolerance Allocation.
Mechanism and Machine Theory, Vol. 17, 1982, pp. 91-97.

39. O.M.A. Sharfi and M.R. Smith
A Simple Method for Allocation of Appropriate Tolerances and Clearances in Linkage Mechanisms.
Mechanism and Machine Theory, accepted for publication.

PART II

40. R.C. Johnson
Impact Forces in Mechanisms.
Machine Design, Vol. 30, June 12, 1958, pp. 138-146.

41. T.P. Goodman
Dynamic Effects of Backlash.
Machine Design, Vol. 35, May 23, 1963, pp. 150-157.

42. S. Dubowsky and F. Freudenstein
Analysis of Mechanical Systems with Clearances, Part I: Formation of Dynamic Model, Part 2: Dynamic Response.
Trans. ASME, J. Eng. Ind., ser. B, Vol. 93, Feb. 1971, pp. 305-316.

43. J.N. Fawcett and J.S. Burdess
Effects of Bearing Clearance in a Four-Bar Linkage.
Proc. 3rd. World Cong. for Theory of M/Cs and Mechanisms, Kupari, Yugoslavia, Sept. 13-20, 1971, Vol. C, paper C-9, pp. 111-126.

44. J.N. Fawcett and J.S. Burdess
Control of Clearance Effects in Linkages.
Engineering Materials and Design, Sept. 1972, pp. 26-27.

45. R.C. Winfrey
Multidegree-of-Freedom Elastic Systems Having Multiple Clearances.
ASME paper No. 72.

46. S.W.E. Earles and C.L.S. Wu
Motion Analysis of a Rigid-Link Mechanism with Clearance at a Bearing, Using Lagrangian Mechanics and Digital Computation.
Proc. Instn. Mech. Engrs, Mechanisms Conf., 1972, pp. 83-87.

47. S. Dubowsky
On Predicting the Dynamic Effects of Clearances in Planar Mechanisms.
Trans. ASME, J. Eng. Ind., ser. B, Vol. 96, Feb. 1974, pp. 317-323.

48. S. Dubowsky.
On Predicting the Dynamic Effects of Clearances in One-Dimensional Closed Loop Systems.
Trans. ASME, J. Eng. Ind., ser. B, Vol. 96, Feb 1974, pp. 324-329.

49. J.N. Fawcett
Maintaining Contact Brings Rewards.
Engineering, Sept. 1975, pp. 741-743.

50. S.W.E. Earles and C.L.S. Wu
Predicting the Occurence of Contact Loss and Impact at a Bearing from a Zero-Clearance Analysis.
Proc. 4th World Cong., Th. Machines and Mechanisms, England, 1975, pp. 1013-1018.

51. W.M. Mansour and M.A. Townsend
Impact Spectra and Intensities for High-Speed Mechanisms.
Trans. ASME, J. Eng. Ind., ser. B, Vol. 97, Feb. 1975, pp. 347-353.

52. M.A. Townsend and W.M. Mansour
A Pendulating Model for Mechanisms with Clearances in the Revolutes.
Trans. ASME, J. Eng. Ind., ser. B, Vol. 97, Feb. 1975, pp. 354-358.

53. S. Dubowsky and T.N. Gardner
Dynamic Interactions of Link Elasticity and Clearance Connections in Planar Mechanical Systems.
Trans. ASME, J. Eng. Ind., ser. B, Vol. 97, May 1975, pp. 652-661.

54. S. Dubowsky and S.C. Young
An Experimental and Analytical Study of Connection Forces in High Speed Mechanisms.
Trans. ASME, J. Eng. Ind., ser. B, Vol. 97, Nov. 1975, pp. 1166-1174.

55. M.A. Townsend and W.M. Mansour
Approximating the Times Between Closely-Spaced Impacts in Mechanisms with Clearances.
Mechanism and Machine Theory, Vol. 11, 1976, pp. 259-265.

56. B. Miedema and W.M. Mansour
Mechanical Joints with Clearance: a Three-Mode Model.
Trans. ASME, J. Eng. Ind., ser. B, Vol. 98, Nov. 1976, pp. 1319-1323.

57. S. Dubowsky and T.N. Gardner
Design and Analysis of Multilink Flexible Mechanisms with Multiple Clearance Connections.
Trans. ASME, J. Eng. Ind., ser. B, Vol. 99, Feb. 1977, pp. 88-96.

58. R.J. Rogers and G.C. Andrews
Dynamic Simulation of Planar Mechanical Systems with Lubricated Bearing Clearances Using Vector-Network Methods.
Trans. ASME, J. Eng. Ind., ser. B, Vol. 99, Feb. 1977, pp. 131-137.

59. C.L.S. Wu and S.W.E. Earles
A Determination of Contact-Loss at a Bearing of a Linkage Mechanism.
Trans. ASME, J. Eng. Ind., ser. B, Vol. 99, May 1977, pp. 375-380.

60. S.W.E. Earles and C.L.S. Wu
Designing to Minimize Impacts in Bearings.
Engineering, Oct. 1977, pp. 866-867.

61. S.W.E. Earles and C.L.S. Wu
A Clearance Impact Magnitude Relationship for Plain Bearings in Oscillatory Systems.
I. Mech. E. Tribology Convention, Proc. I. Mech. Eng., 1978, pp. 63-67.

62. S.J. Grant and J.N. Fawcett
Effects of Clearance at the Coupler-Rocker Bearing of a 4-Bar Linkage.
Mechanism and Machine Theory, Vol. 14, 1979, pp. 99-110.

63. S. Dubowsky and M.F. Moening
An Experimental and Analytical Study of Impact Forces in Elastic Mechanical Systems with Clerances.
Mechanism and Machine Theory, Vol. 13, 1978, pp. 451-465.

64. B.M. Bahgat, M.O.M. Osman and T.S. Sankar
On the Effect of Bearing Clearances in the Dynamic Analysis of Planar Mechanisms.
J. Mech. Eng. Sci, Vol. 21, No. 6, 1979, pp. 429-437.

65. S.G. Grant.
Dynamic Effects of Bearing Clearance in Mechanisms.
Ph.D. Thesis, University of Newcastle upon Tyne, 1976.

66. M.J. Walker
Balancing Planar Linkage Mechanisms.
Ph.D. Thesis, University of Newcastle upon Tyne, 1979.

67. S.W.E. Earles and O. Kilicay
Predicting Impact Conditions Due to Bearing Clearances in Linkage Mechanisms.
Proc. 5th World Cong. Th. Machines and Mechanism, USA, 1979, pp. 1078-1081.

68. S.W.E. Earles and O. Kilicay
A Design Criterion for Maintaining Contact at Plain Bearings.
Proc. Instn. Mech. Engrs, Vol. 194,1980, pp. 249-258.

69. R.S. Haines
Survey: 2-Dimensional Motion and Impact at Revolute Joints.
Mechanism and Machine Theory, Vol. 15, 1980, pp. 361-370.

70. R.S. Haines
A Theory of Contact Loss at Revolute Joints with Clearance.
J. Mech. Eng. Sci., Vol. 22, No. 3, 1980, pp. 129-136.

71. E04-Minimizing or Maximizing a Function.
NAGFLIB : 1482/0 : MK6 : May 77
Numerical Algorithms Group, U.K.

TABLES

TABLE 6.1

```
******************************
***/                       /*
**/ TOCALM | OSMAN | FEB 82 /**
*/                       /***
******************************  FOUR-BAR SINE FUNCTION GENERATOR
```

LINEAR DIMENSIONS ARE IN INCHES

OUTPUT TOLERANCE = 1.000 DEGREES

TOLERANCES ON ARC LENGTHS :-

ARC NUMBER	LOOP NUMBER	ARC LENGTH	TOLERANCE
1	1	2.070	0.3570E-02
2	1	2.421	0.4364E-02
3	1	0.746	0.2092E-02
4	1	1.000	0.2497E-02

CLEARANCES IN JOINTS :-

PRECEEDING ARC NUMBER	FOLLOWING ARC NUMBER	LOOP NUMBER	CLEARANCE
1	2	1	0.1785E-02
2	3	1	0.1046E-C2
3	4	1	0.1046E-02
4	1	1	0.1248E-02

OUTPUT ARC :-

ARC ANGLE (DEGREES) = 287.148 ,   TOLERANCE ON ARC ANGLE (DEGREES) = 0.2903E+00

INPUT REFERENCE POSITION :-

REFERENCE ANGLE (DEGREES) = 296.211 ,   TOLERANCE ON ANGLE (DEGREES) = 0.3694E+00

EXECUTION TIME
0.285 CPU SEC.

TABLE 6.2

FOUR-BAR SINE FUNCTION GENERATOR: COMPARISON OF RESULTS

PARAMETER		TOLERANCES AND CLEARANCES			
NAME AND SYMBOL	NOMINAL DIMENSION	PRESENT WORK		QUOTED REFERENCES	
		WITHOUT COST OPTIMIZATION	WITH COST OPTIMIZATION	[24]	[26]
Input Ref. Angle γ_c	116.211°	$\varepsilon_c = 0.496°$	0.369°	Not Accounted For	
Output Arc Angle γ_c	287.148°	$\varepsilon_o = 0.408°$	0.290°	Not Accounted For	
Crank, $a_{1,1}$	2.06980"	$t_{1,1} = 2.47$	3.57	2.612	2.57
Coupler, $a_{2,1}$	2.42146"	$t_{2,1} = 2.41$	4.36	2.593	2.88
Follower, $a_{3,1}$	0.74613"	$t_{3,1} = 2.94$	2.09	2.846	2.88
Frame, $a_{4,1}$	1.0000	$t_{4,1} = 2.43$	2.50	2.628	2.77
		$c_{1,2,1} = 1.205$	1.785	2.556	1.26
		$c_{2,3,1} = 1.205$	1.046	2.611	1.14
		$c_{3,4,1} = 1.215$	1.046	2.952	1.40
		$c_{4,1,1} = 1.215$	1.248	2.571	1.248

Linear Tolerances and Clearances are in 10^{-3} in.

TABLE 6.3

```
******************************
***/                       /*
**/ TOCALM | OSMAN | FEB 82 /**
*/                       /***
******************************  SLIDER CRANK MECHANISM
```

LINEAR DIMENSIONS ARE IN MILLIMETERS

OUTPUT TOLERANCE (SLIDER DISPLACEMENT)=0.250E+00

TOLERANCES ON ARC LENGTHS :-

ARC NUMBER	LOOP NUMBER	ARC LENGTH	TOLERANCE
1	1	50.000	0.7742E-01
2	1	200.000	0.1669E+00

CLEARANCES IN JOINTS :-

PRECEEDING ARC NUMBER	FOLLOWING ARC NUMBER	LOOP NUMBER	CLEARANCE
1	2	1	0.3871E-01
2	3	1	0.8345E-01
3	4	1	0.1000E+00
4	1	1	0.3871E-01

INPUT REFERENCE POSITION :-

REFERENCE ANGLE = 270.000 , (DEGREES) TOLERANCE ON ANGLE = 0.3165E+00 (DEGREES)

EXECUTION TIME
0.326 CPU SEC.

T A B L E   6 . 4

```
******************************
***/                        /*
**/ TOCALM | OSMAN | FEB 82 /**
*/                        /***
******************************   FOUR-BAR MOTOR-CYCLE REAR SUSPENSION
```

LINEAR DIMENSIONS ARE IN MILLIMETERS

OUTPUT TOLERANCE (POLAR CO-ORDINATES)
RADIAL=0.100E+01   ANGULAR=0.600 DEGREES

TOLERANCES ON ARC LENGTHS :-

ARC NUMBER	LOOP NUMBER	ARC LENGTH	TOLERANCE
1	1	469.900	0.8905E-01
2	1	170.200	0.4247E+00
3	1	429.200	0.1832E+00
4	1	325.730	0.5299E+00

CLEARANCES IN JOINTS :-

PRECEEDING ARC NUMBER	FOLLOWING ARC NUMBER	LOOP NUMBER	CLEARANCE
1	2	1	0.4453E-01
2	3	1	0.4371E-01
3	4	1	0.9158E-01
4	1	1	0.4355E-01

Continued following page ....

Table 6.4 (.... continued from preceding page)

```
OUTPUT ARC :-

ARC LENGTH =   44.600 ,       TOLERANCE CN ARC LENGTH = 0.8741E-01

ARC ANGLE  =   60.000 ,       TOLERANCE CN ARC ANGLE  = 0.9088E+C0
(DEGREES)                              (DEGREES)


ORIGIN ARC :-
ARC LENGTH =  148.660 ,       TOLERANCE CN ARC LENGTH = 0.8710E-01

ARC ANGLE  =   88.246 ,       TOLERANCE CN ARC ANGLE  = 0.6880E+00
(DEGREES)                              (DEGREES)


INPUT REFERENCE POSITION :-

REFERENCE ANGLE =   91.900 ,      TOLERANCE ON ANGLE = 0.6951E+00
   (DEGREES)                            (DEGREES)


EXECUTION TIME
 0.470 CPU SEC.
```

TABLE 6.5

```
*******************************
***/                        /*
**/ TOCALM | OSMAN | FEB 82 /**
*/                        /***
*******************************   SIX-BAR SINE FUNCTION GENERATOR
```

LINEAR DIMENSIONS ARE IN MILLIMETERS

OUTPUT TOLERANCE = 0.500 DEGREES

TOLERANCES ON ARC LENGTHS :-

ARC NUMBER	LOOP NUMBER	ARC LENGTH	TOLERANCE
1	1	7.500	0.1206E-01
2	1	30.139	0.5726E-01
3	1	21.116	0.2902E-01
4	1	36.716	0.4949E-01
2	2	28.833	0.6347E-01
3	2	21.421	0.2895E-01
4	2	61.832	0.8665E-01
5	2	39.998	0.4402E-01

TOLERANCES ON ARC ANGLES :-

ARC NUMBER	LOOP NUMBER	ANGLE MAG. (DEGREES)	TOLERANCE (DEGREES)
2	2	179.990	0.3936E+00
5	2	64.537	0.4175E+00

Continued following page .....

Table 6.5 (.... continued from preceding page)

CLEARANCES IN JOINTS :-

PRECEEDING ARC NUMBER	FOLLOWING ARC NUMBER	LOOP NUMBER	CLEARANCE
1	2	1	0.6028E-02
2	3	1	0.1451E-01
3	4	1	0.1451E-01
4	1	1	0.6028E-02
2	3	2	0.1448E-01
3	4	2	0.1448E-01
4	5	2	0.2201E-01

OUTPUT ARC :-

ARC ANGLE (DEGREES) = 31.090 , TOLERANCE ON ARC ANGLE (DEGREES) = 0.3346E+00

INPUT REFERENCE POSITION :-

REFERENCE ANGLE (DEGREES) = 212.333 , TOLERANCE ON ANGLE (DEGREES) = 0.1201E+01

EXECUTION TIME
1.710 CPU SEC.

TABLE 6.6

```
*******************************
***/                        /*
**/ TOCALM | OSMAN | FEB 82 /**
*/                        /***
*******************************  EIGHT-BAR STRAIGHT LINE GENERATOR
```

LINEAR DIMENSIONS ARE IN MILLIMETERS

OUTPUT TOLERANCE (CARTESIAN CO-ORDINATES)
X-DIRECTION=0.100E+01   Y-DIRECTION=0.500E+01

TOLERANCES ON ARC LENGTHS :-

ARC NUMBER	LOOP NUMBER	ARC LENGTH	TOLERANCE
1	1	177.500	0.1465E+00
2	1	155.000	0.1177E+00
3	1	435.000	0.1467E+00
4	1	177.500	0.1820E+00
2	2	155.000	0.1177E+00
3	2	435.000	0.1467E+00
3	3	155.000	0.1478E+00
4	3	155.000	0.1652E+00

CLEARANCES IN JOINTS :-

PRECEEDING ARC NUMBER	FOLLOWING ARC NUMBER	LOOP NUMBER	CLEARANCE
1	2	1	0.5886E-01
2	3	1	0.5886E-01
3	4	1	0.7335E-01
4	1	1	0.7323E-01
1	2	2	0.5886E-01
2	3	2	0.5886E-01
3	4	2	0.7335E-01
2	3	3	0.5886E-01
3	4	3	0.7389E-01
4	5	3	0.5886E-01

INPUT REFERENCE POSITION :-

REFERENCE ANGLE = 325.000 (DEGREES)   TOLERANCE ON ANGLE = 0.1732E+01 (DEGREES)

EXECUTION TIME
1.524 CPU SEC.

TABLE 6.7

```
*****************************
***/                       /*
**/ TOCALM | OSMAN | FEB 82 /**
*/                       /***
*****************************  TEN-BAR NEEDLE MECHANISM
```

LINEAR DIMENSIONS ARE IN MILLIMETERS

OUTPUT TOLERANCE = 1.000 DEGREES

TOLERANCES ON ARC LENGTHS :-

ARC NUMBER	LOOP NUMBER	ARC LENGTH	TOLERANCE
1	1	19.000	0.5968E-01
2	1	95.000	0.3101E+00
3	1	104.500	0.4917E+00
4	1	112.468	0.1799E+00
2	2	105.000	0.1714E+00
3	2	31.000	0.7515E-01
4	2	110.064	0.2890E+00
3	3	61.000	0.1173E+00
4	3	12.500	0.4893E-01
5	3	24.000	0.8353E-01
4	4	245.500	0.2431E+00
5	4	85.000	0.3507E+00
6	4	352.716	0.3295E+00

TOLERANCES ON ARC ANGLES :-

ARC NUMBER	LOOP NUMBER	ANGLE MAG. (DEGREES)	TOLERANCE (DEGREES)
4	2	33.593	0.3463E+00
5	3	47.384	0.7960E+00
6	4	20.442	0.1945E+00

Continued following page .....

Table 6.7 (.... continued from preceding page)

CLEARANCES IN JOINTS :-

PRECEEDING ARC NUMBER	FOLLOWING ARC NUMBER	LOOP NUMBER	CLEARANCE
1	2	1	0.2984E-01
2	3	1	0.1000E+00
3	4	1	0.4177E-01
4	1	1	0.2984E-01
1	2	2	0.2984E-01
2	3	2	0.3757E-01
3	4	2	0.3757E-01
2	3	3	0.3757E-01
3	4	3	0.2446E-01
4	5	3	0.2446E-01
3	4	4	0.2446E-01
4	5	4	0.1000E+00
5	6	4	0.1000E+00

OUTPUT ARC :-

ARC ANGLE (DEGREES) = 10.048 , TOLERANCE ON ARC ANGLE (DEGREES) = 0.6377E+00

INPUT REFERENCE POSITION :-

REFERENCE ANGLE (DEGREES) = 121.641 , TOLERANCE ON ANGLE (DEGREES) = 0.2803E+01

EXECUTION TIME
4.621 CPU SEC.

TABLE 10.1 : CRANK-ROCKER WITH POOR TRANSMISSION ANGLE

LINK	ARC LENGTH (mm)	ORIGINAL MASS PARAMETERS				ADDITIONAL MASS PARAMETERS (POST-OPTIMIZATION)			
		m_o (kg)	ρ_o (mm)	λ_o (deg)	I_o (kg mm^2)	m_* (kg)	ρ_* (mm)	λ_* (deg)	I_* (kg mm^2)
CRANK	50	0.015	25	0	18.5	0.01415	31.1	151.6	13.72
COUPLER	100	0.025	50	0	101.7	0.204	7.62	134	24.8
FOLLOWER	100	0.025	50	0	101.7	0.0	--	--	0.0
FRAME	100	-	-	-	-	-	-	-	-

TABLE 10.2

```
>
>CRANK-ROCKER WITH ARC LENGTHS  50,100,100,100
>
>
>
> 31 FUNCTION EVALUATIONS WERE NEEDED BY E04HBF
>

>ITNS     FN EVALS           FN VALUE          NORM OF PROJ GRADIENT
>  0         0                2.7801E-01                3.6750E+00

> J            X(J)               G(J)            STATUS
> 1          1.5000E-02          1.9299E-01     FREE
> 2          3.3000E-02          8.7721E-02     FREE
> 3          1.7453E+00         -5.4474E-04     FREE
> 4          2.1500E-01          3.2916E-01     FREE
> 5          2.1000E-02          3.6065E+00     FREE
> 6          1.7453E+00         -1.0730E-02     FREE
> 7          1.0000E-02          3.6142E-01     FREE
> 8          6.0000E-02          3.3745E-02     FREE
> 9          2.1000E-02          1.7129E-01     FREE
>10          1.7453E+00          3.7483E-03     FREE
>11          1.5000E-02          2.1838E-01     FREE
>12          3.1416E+00          3.6031E-01     FREE
>13          6.9813E-01          2.5872E-02     FREE
>14          5.2360E+00         -8.2803E-04     FREE
>15          3.1416E+00         -7.8676E-02     FREE

>ESTIMATED CONDITION NUMBER OF PROJECTED HESSIAN =   2.50E+05
>

>ITNS     FN EVALS           FN VALUE          NORM OF PROJ GRADIENT
> 81       1383               2.3297E-01                6.5733E-04

> J            X(J)               G(J)            STATUS
> 1          1.4154E-02         -3.3161E-04     FREE
> 2          3.1139E-02         -1.5073E-04     FREE
> 3          2.6453E+00          2.0807E-06     FREE
> 4          2.0387E-01         -3.9412E-06     FREE
> 5          7.6170E-03         -3.5815E-04     FREE
> 6          2.3383E+00         -2.0143E-06     FREE
> 7          7.9754E-03         -1.0892E-04     FREE
> 8          0.0                 1.6183E+00     LOWER BOUND
> 9         -6.0000E-02          0.0            LOWER BOUND
>10         -1.1410E+00          0.0            FREE
>11          0.0                 0.0            LOWER BOUND
>12          3.1250E+00         -1.4254E-05     FREE
>13          4.3316E-01         -3.9204E-06     FREE
>14          5.3538E+00          5.3065E-08     FREE
>15          3.1644E+00          3.9877E-04     FREE

>ESTIMATED CONDITION NUMBER OF PROJECTED HESSIAN =   1.63E+05
```

TABLE 10.3 : CRANK-ROCKER WITH GOOD TRANSMISSION ANGLE

LINK	ARC LENGTH (mm)	ORIGINAL MASS PARAMETERS				ADDITIONAL MASS PARAMETERS (POST-OPTIMIZATION)			
		m_o (kg)	ρ_o (mm)	λ_o (deg)	I_o (kg mm^2)	m_* (kg)	ρ_* (mm)	λ_* (deg)	I_* (kg mm^2)
CRANK	50	0.015	25	0	18.5	0.045	37.6	238.9	64.1
COUPLER	100	0.025	50	0	101.7	0.223	5.7	119.4	7.2
FOLLOWER	100	0.C25	50	0	101.7	0.157	17.8	242.5	49.7
FRAME	132.5	–	-	-	-	-	-	-	-

TABLE 10.4

```
>
>CRANK-ROCKER WITH ARC LENGTHS  50,100,100,132.5
>
>
>
> 31 FUNCTION EVALUATIONS WERE NEEDED BY E04HBF
>

>ITNS     FN EVALS          FN VALUE          NORM OF PROJ GRADIENT
>  0         0              2.4539E-01                4.5558E+00

> J            X(J)             G(J)            STATUS
> 1          1.5000E-02      -4.8751E-01       FREE
> 2          3.3000E-02      -2.2159E-01       FREE
> 3          1.7453E+00       1.2837E-03       FREE
> 4          2.1500E-01       3.7792E-01       FREE
> 5          2.1000E-02       4.4431E+00       FREE
> 6          1.7453E+00       8.5421E-02       FREE
> 7          1.0000E-02       7.0332E-01       FREE
> 8          6.0000E-02      -1.8041E-02       FREE
> 9          2.1000E-02      -5.2148E-02       FREE
>10          1.7453E+00      -2.6588E-03       FREE
>11          1.5000E-02      -1.7533E-03       FREE
>12          3.1416E+00       1.7322E-01       FREE
>13          6.9813E-01      -1.3623E-02       FREE
>14          5.2360E+00      -3.3516E-03       FREE
>15          3.1416E+00       2.2193E-01       FREE

>ESTIMATED CONDITION NUMBER OF PROJECTED HESSIAN =   2.78E+05
>

>ITNS     FN EVALS          FN VALUE          NORM OF PROJ GRADIENT
>120       2093             1.8745E-01                6.5441E-04

> J            X(J)             G(J)            STATUS
> 1          4.5394E-02      -3.6186E-04       FREE
> 2          3.7563E-02      -4.3591E-04       FREE
> 3         -2.1134E+00      -6.7861E-06       FREE
> 4          2.2324E-01       6.9081E-05       FREE
> 5          5.6779E-03       2.9275E-04       FREE
> 6          2.0835E+00       3.9715E-06       FREE
> 7          0.0              4.7513E-08       LOWER BOUND
> 8          1.5667E-01      -1.0097E-05       FREE
> 9          1.7810E-02       8.2714E-06       FREE
>10          4.2332E+00      -6.0208E-06       FREE
>11          1.2207E-04       1.3039E-06       FREE
>12          3.1399E+00      -1.2815E-05       FREE
>13          8.5740E-01       1.3791E-06       FREE
>14          5.0126E+00       3.3440E-06       FREE
>15          3.0270E+00      -1.2797E-04       FREE

>ESTIMATED CONDITION NUMBER OF PROJECTED HESSIAN =   1.93E+05
```

TABLE 10.5 : DRAG-LINK

LINK	ARC LENGTH (mm)	ORIGINAL MASS PARAMETERS				ADDITIONAL MASS PARAMETERS (Post-Optimization)			
		m_o (kg)	ρ_o (mm)	λ_o (deg)	I_o (kg mm^2)	m_* (kg)	ρ_* (mm)	λ_* (deg)	I_* (kg mm^2)
CRANK	132.5	0.0325	66.25	0	191.7	0	--	--	0
COUPLER	100	0.025	50	0	101.7	0.1246	28.6	195.3	101.6
FOLLOWER	100	0.025	50	0	101.7	0	--	--	0
FRAME	50	-	-	-	-	-	-	-	-

TABLE 10.6

```
>
>DRAG-LINK WITH ARC LENGTHS  132.5, 100, 100, 50
>
>
>
> 31 FUNCTION EVALUATIONS WERE NEEDED BY E04HBF
>

>ITNS     FN EVALS          FN VALUE          NORM OF PROJ GRADIENT
>  0        0              2.0688E-01                3.0911E+00

> J           X(J)               G(J)              STATUS
> 1           0.0                0.0              LOWER BOUND
> 2           0.0                0.0              FREE
> 3           3.4907E-01         0.0              FREE
> 4           1.0000E-01        -9.0944E-01       FREE
> 5           4.1000E-02        -1.0808E+00       FREE
> 6           3.3161E+00        -8.1739E-02       FREE
> 7           1.0000E-02         7.0651E-01       FREE
> 8           4.0000E-03         2.5289E+00       FREE
> 9           3.1000E-02         3.6501E-01       FREE
>10          -2.2689E-01        -1.9633E-02       FREE
>11           1.5000E-02         4.8900E-02       FREE
>12           3.1416E+00         3.4209E-01       FREE
>13          -3.4907E-01        -1.3648E-01       FREE
>14           2.7925E+00        -4.9775E-01       FREE
>15           3.1416E+00         3.7347E-01       FREE

>ESTIMATED CONDITION NUMBER OF PROJECTED HESSIAN IS MORE THAN 1.0E+6
>

>ITNS     FN EVALS          FN VALUE          NORM OF PROJ GRADIENT
> 38       673             1.7103E-01                7.0882E-04

> J           X(J)               G(J)              STATUS
> 1           0.0                2.8125E-11       LOWER BOUND
> 2           0.0                0.0              FREE
> 3           3.4907E-01         0.0              FREE
> 4           1.2457E-01         1.7631E-04       FREE
> 5           2.8551E-02         6.8371E-04       FREE
> 6           3.4094E+00        -1.4658E-06       FREE
> 7           0.0                4.2913E-08       LOWER BOUND
> 8           0.0                5.1943E-01       LOWER BOUND
> 9          -5.0851E-03         0.0              FREE
>10          -1.3523E+00         0.0              FREE
>11           0.0                0.0              LOWER BOUND
>12           3.1231E+00         4.9326E-06       FREE
>13          -3.3470E-01         5.9534E-05       FREE
>14           2.8071E+00        -1.3949E-05       FREE
>15           3.1247E+00        -1.0801E-05       FREE

>ESTIMATED CONDITION NUMBER OF PROJECTED HESSIAN IS MORE THAN 1.0E+6
```

F I G U R E S

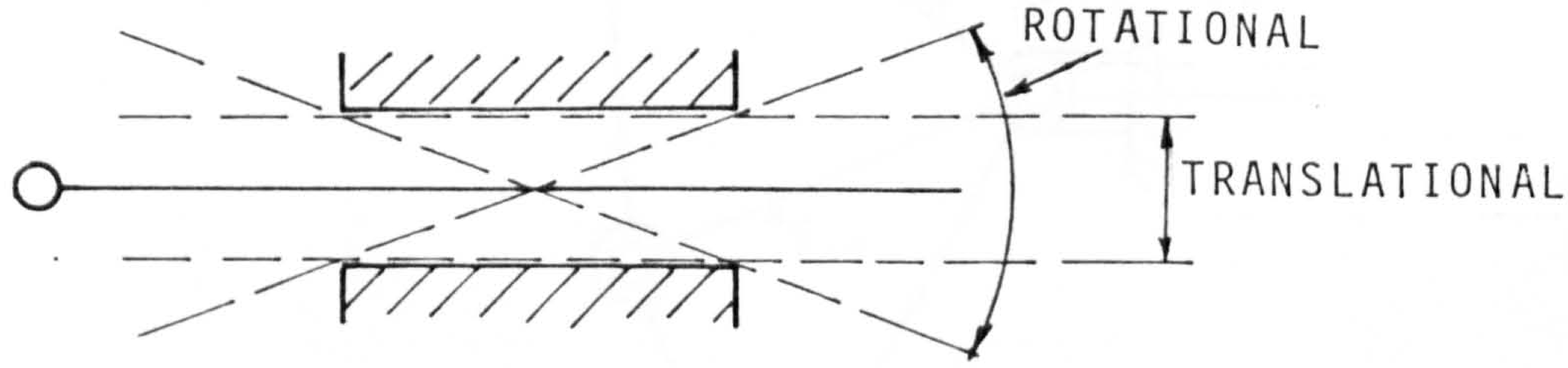
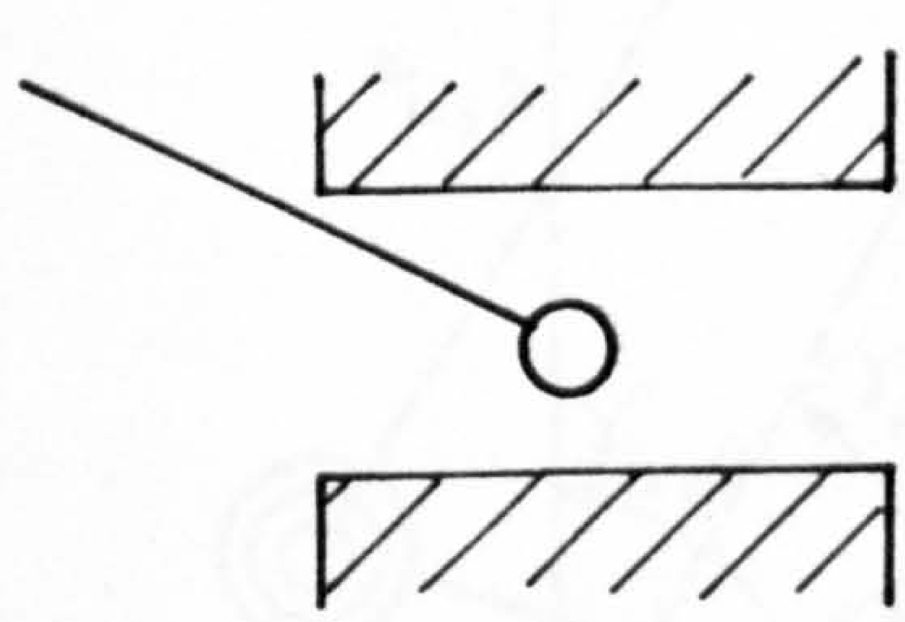
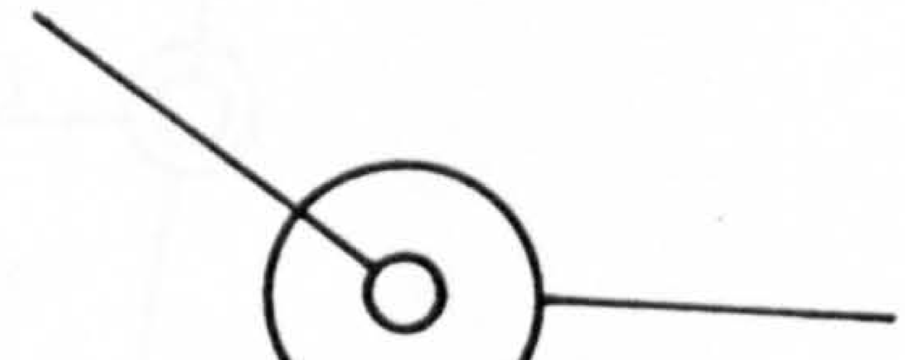


Fig. 3.1 PLAY DUE TO CLEARANCE IN A SLIDING JOINT



(a) SLIDING JOINT



(b) Revolute Joint

Fig. 3.2 JOINTS WITH COMPARABLE PLAY DUE TO CLEARANCE

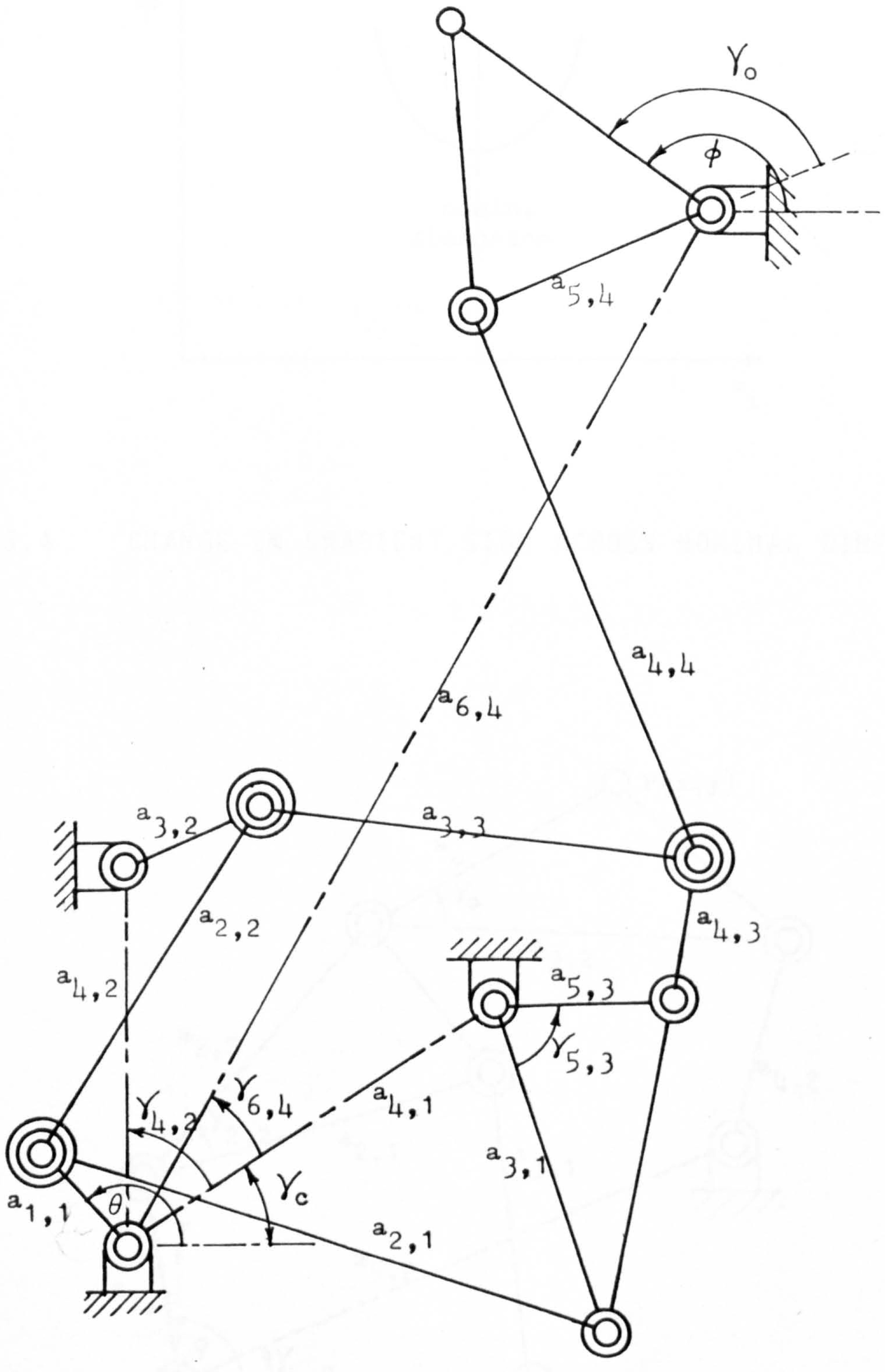


Fig. 3.3 TEN-BAR PLANAR LINKAGE MECHANISM

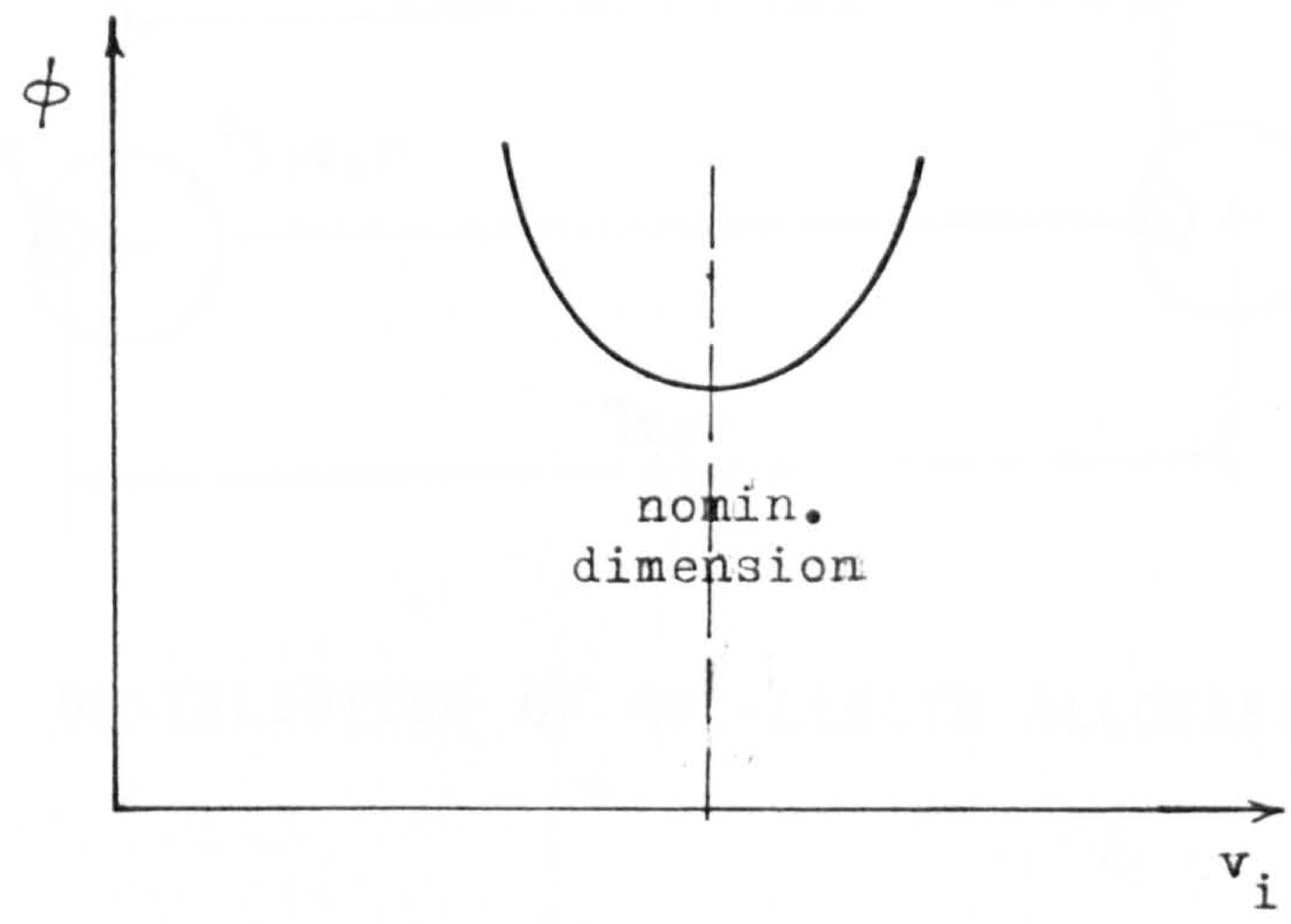


Fig. 3.4 CHANGE IN GRADIENT SIGN ACROSS NOMINAL DIMENSION

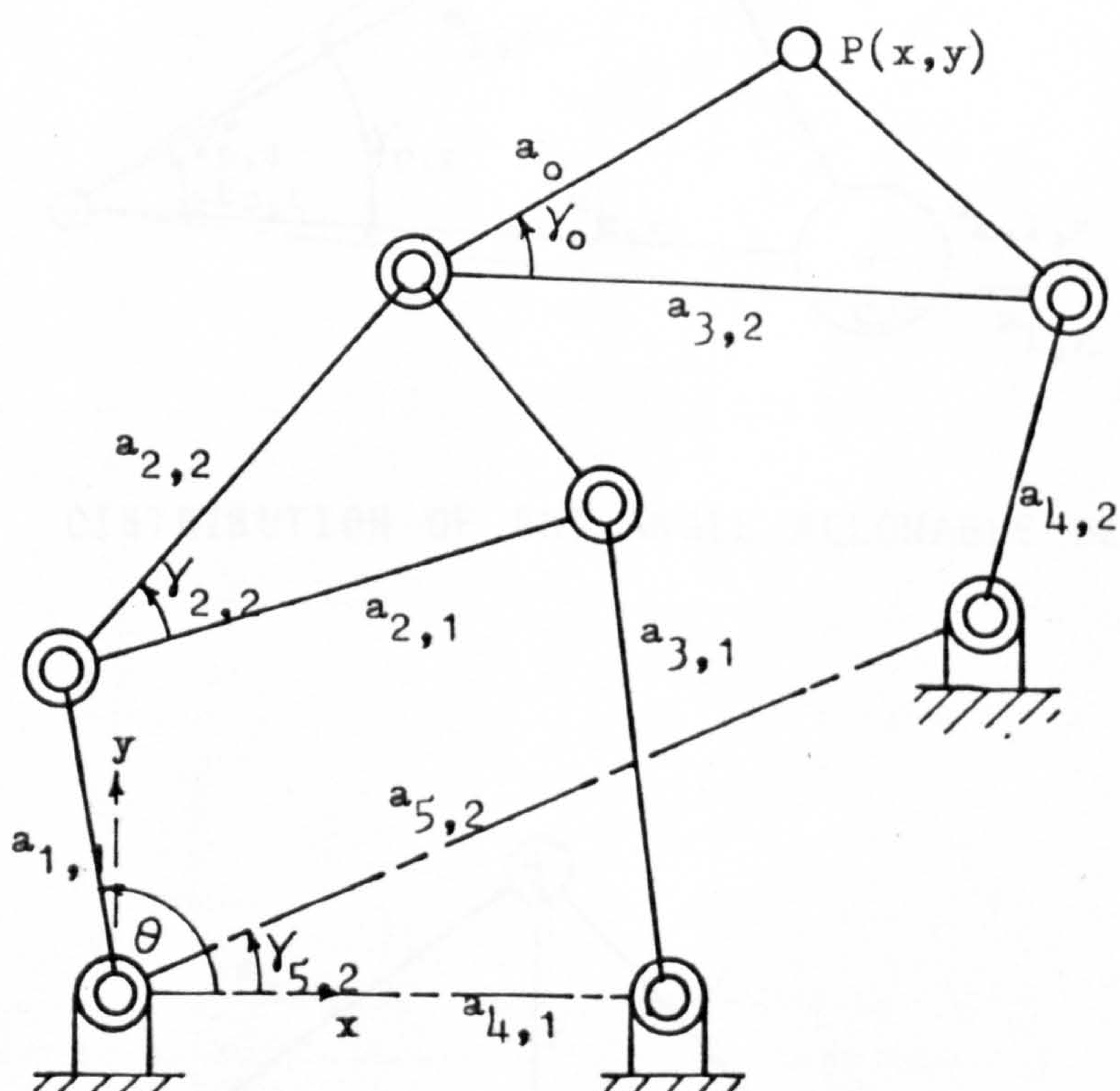


Fig. 3.5 SIX-BAR PATH GENERATOR LINKAGE

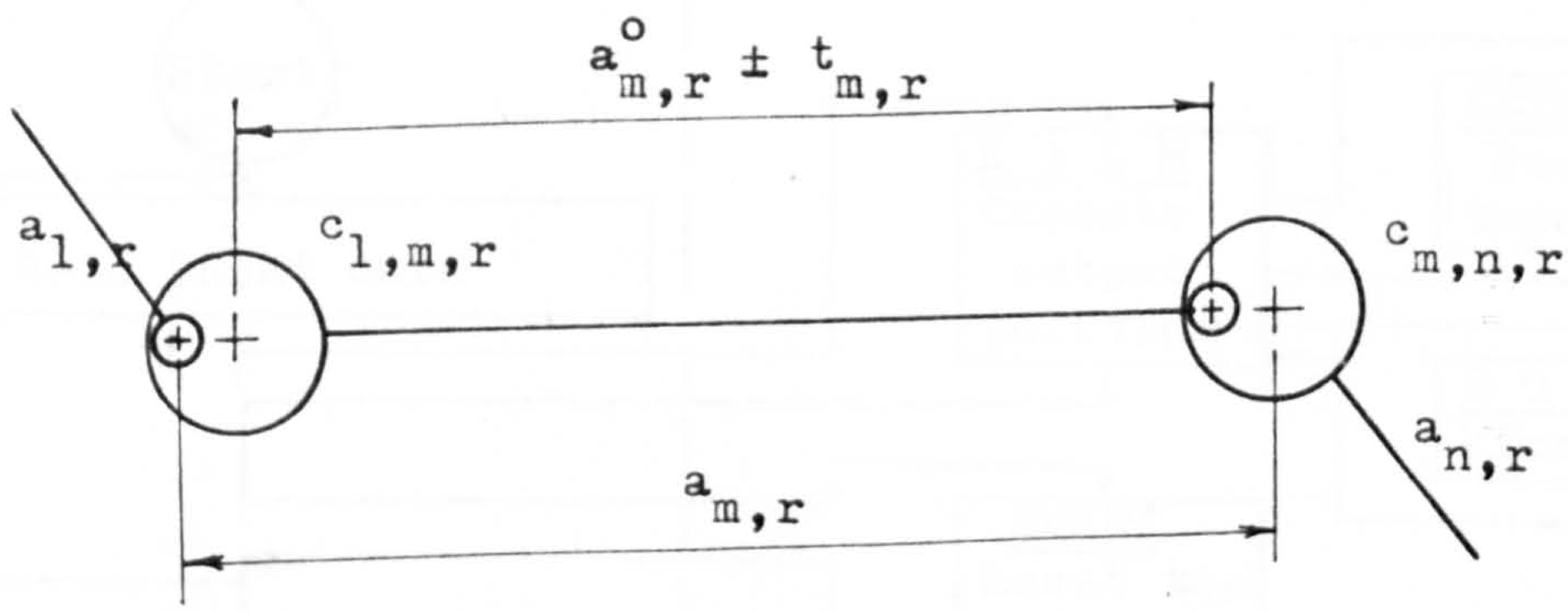


Fig. 4.1 DISTRIBUTION OF ARC-LENGTH ALLOWABLE DEVIATION

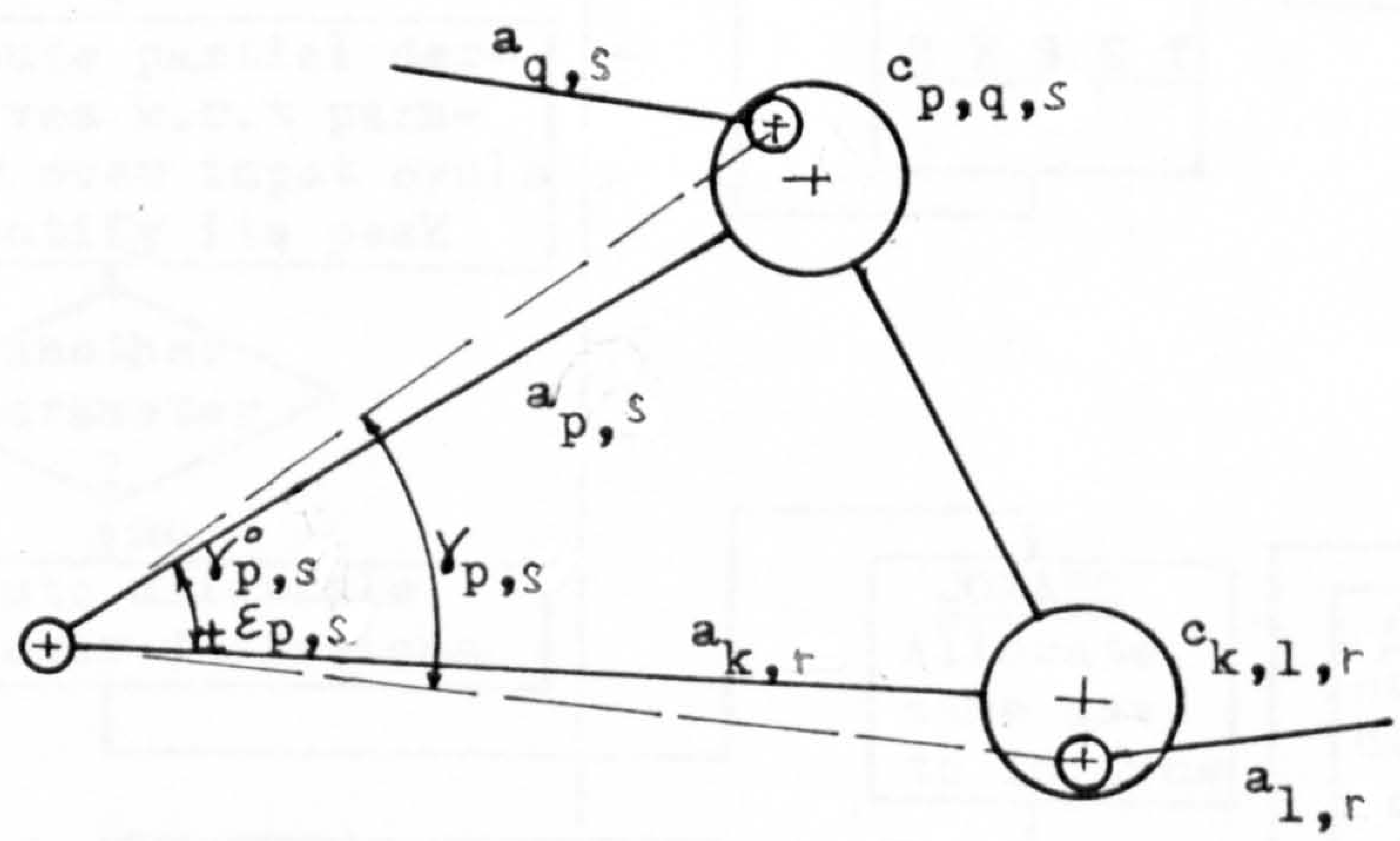


Fig. 4.2 DISTRIBUTION OF ARC-ANGLE ALLOWABLE DEVIATION

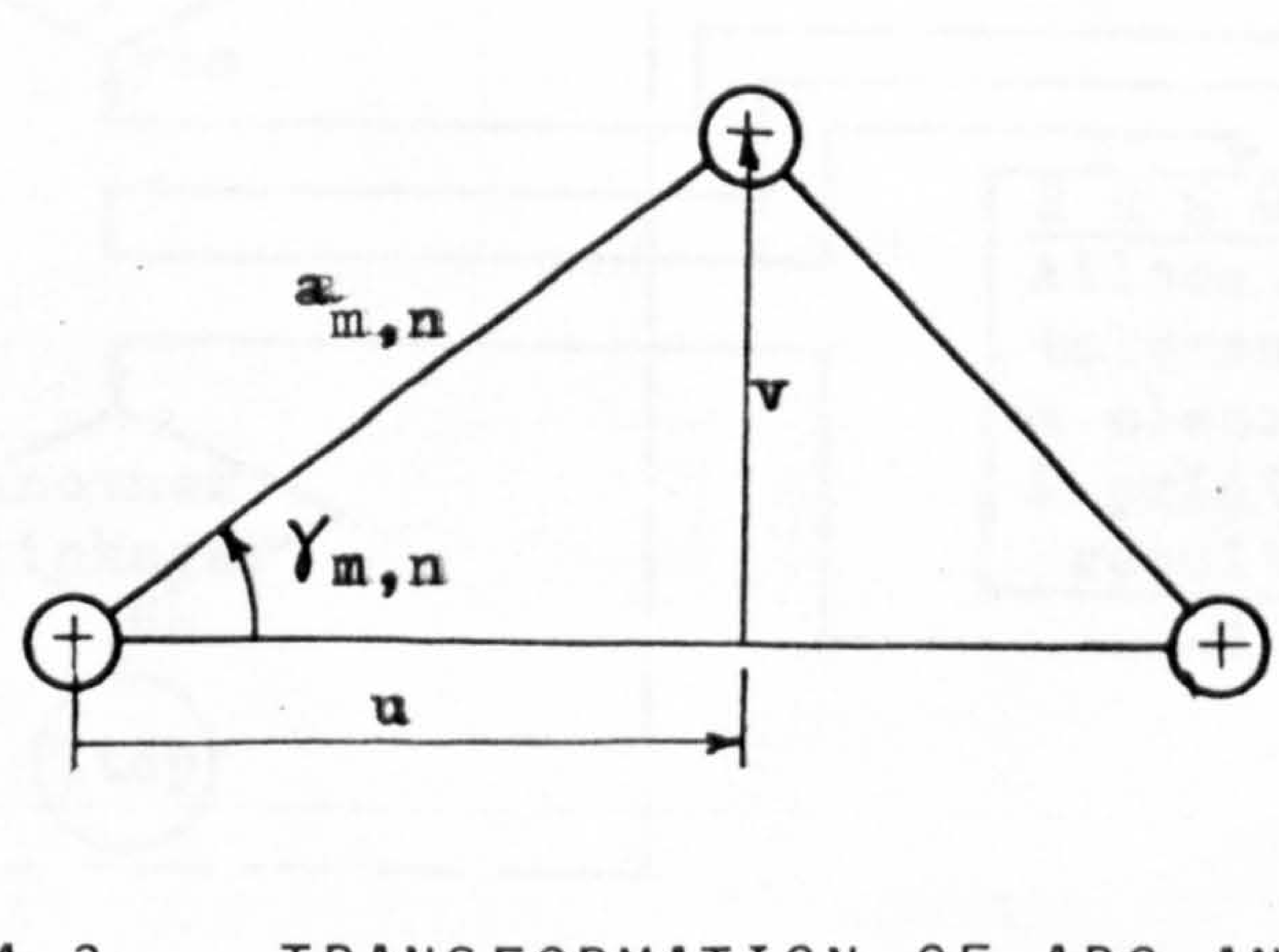


Fig. 4.3 TRANSFORMATION OF ARC-ANGLE

Fig. 5.1 FLOW CHART OF PROGRAM TOCALM

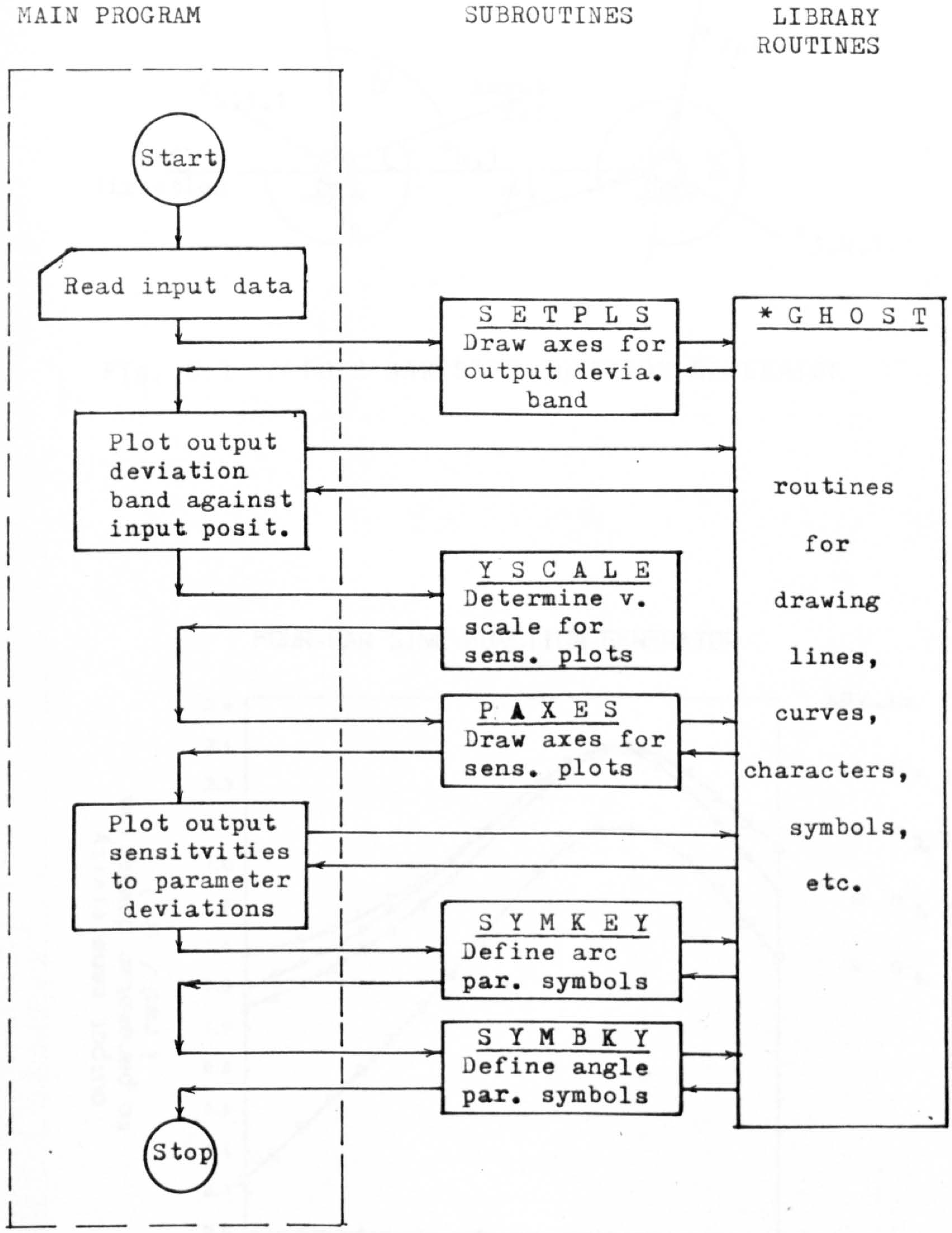


Fig. 5.2 FLOW CHART OF PROGRAM <u>PSODPLOTS</u>

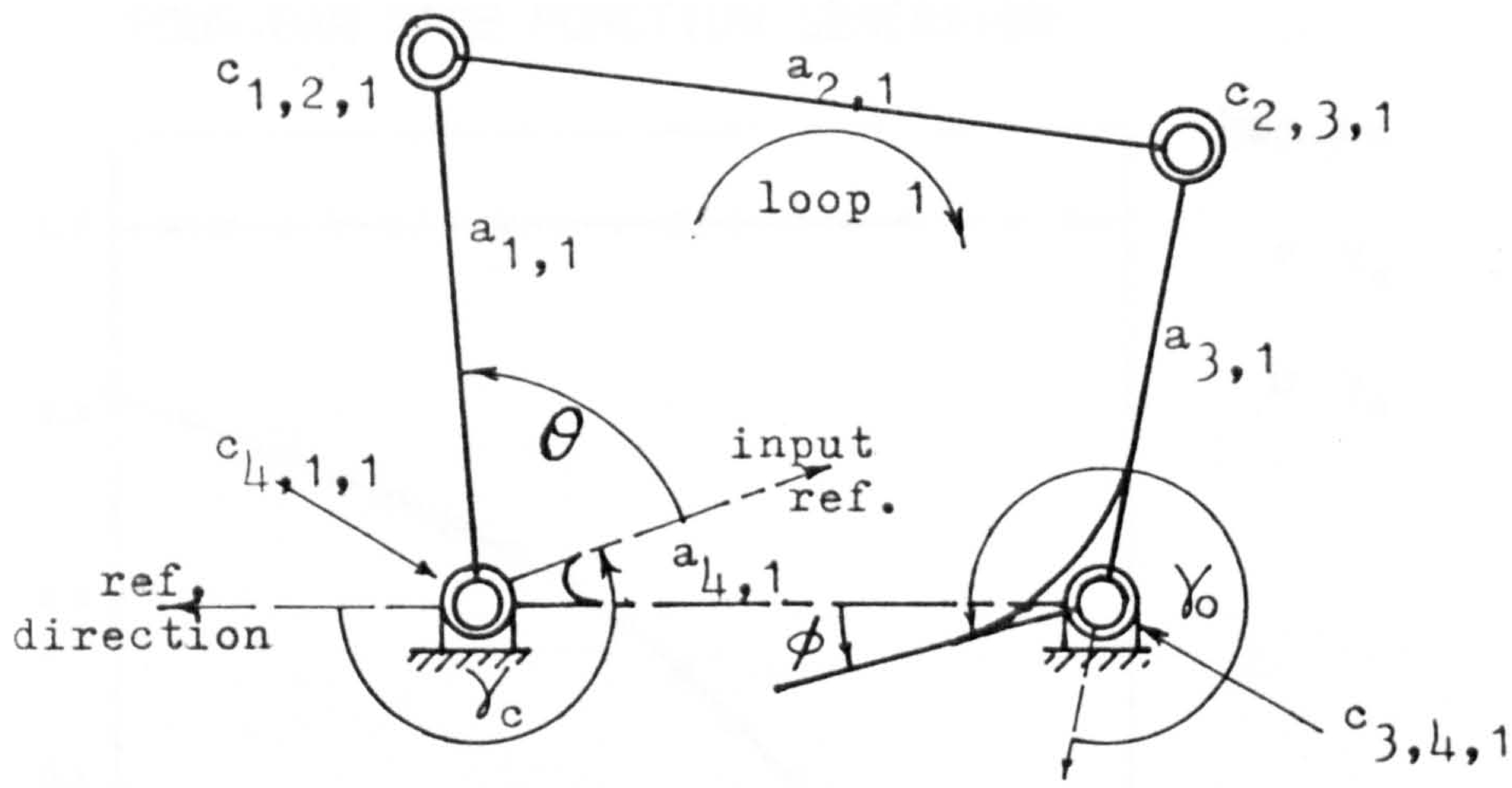


Fig. 6.1 FOUR-BAR SINE FUNCTION GENERATOR

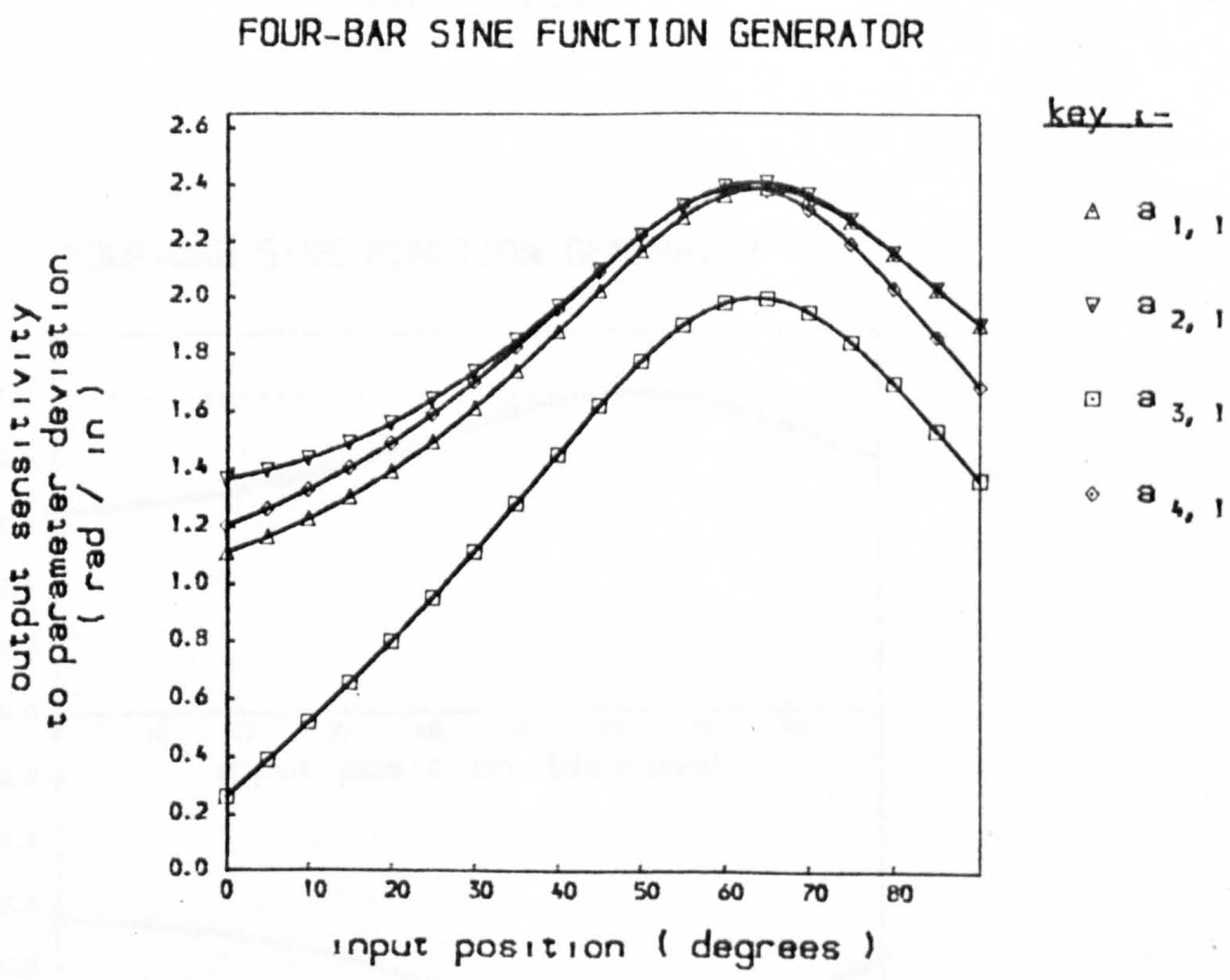


Fig. 6.2a

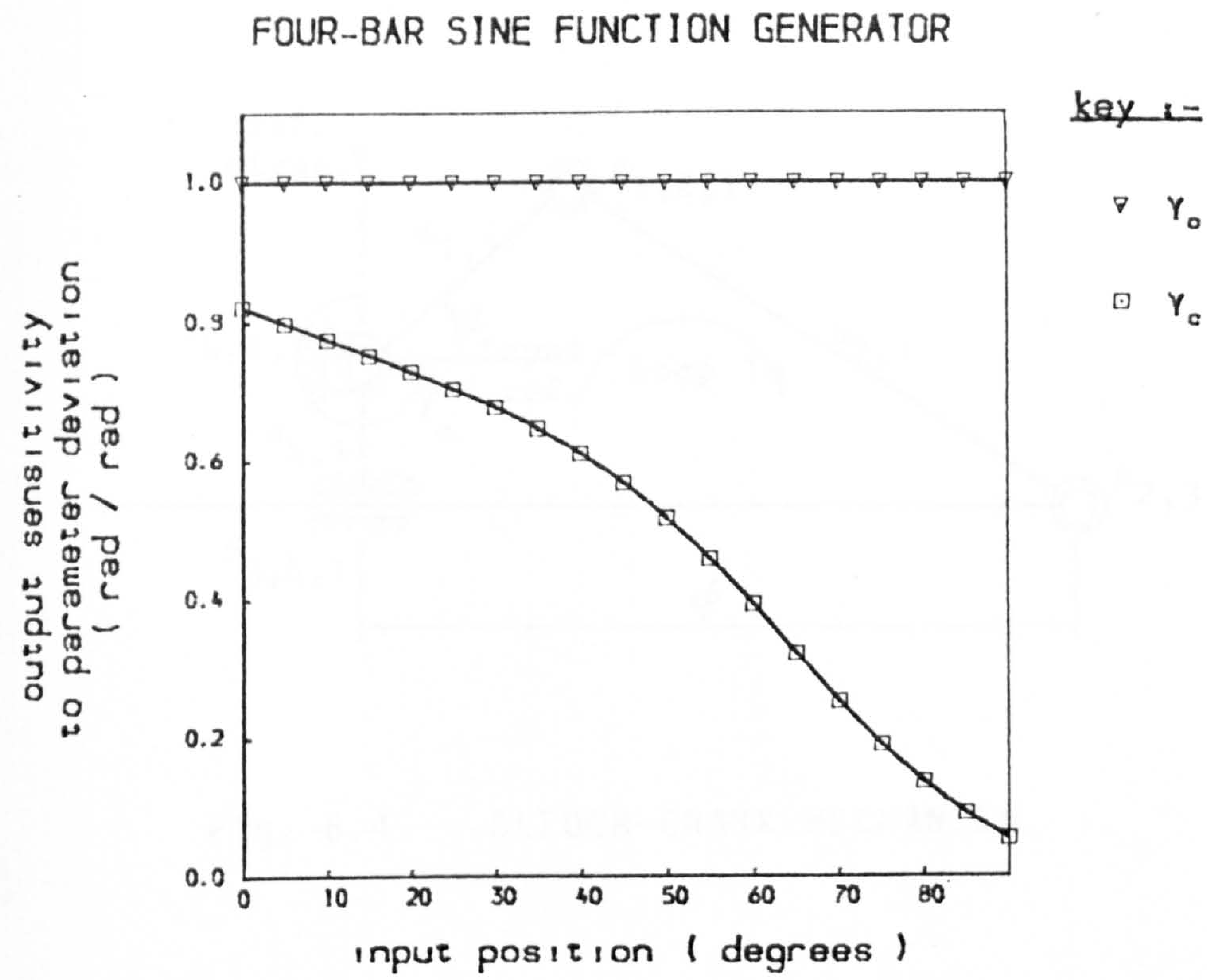


Fig. 6.2b

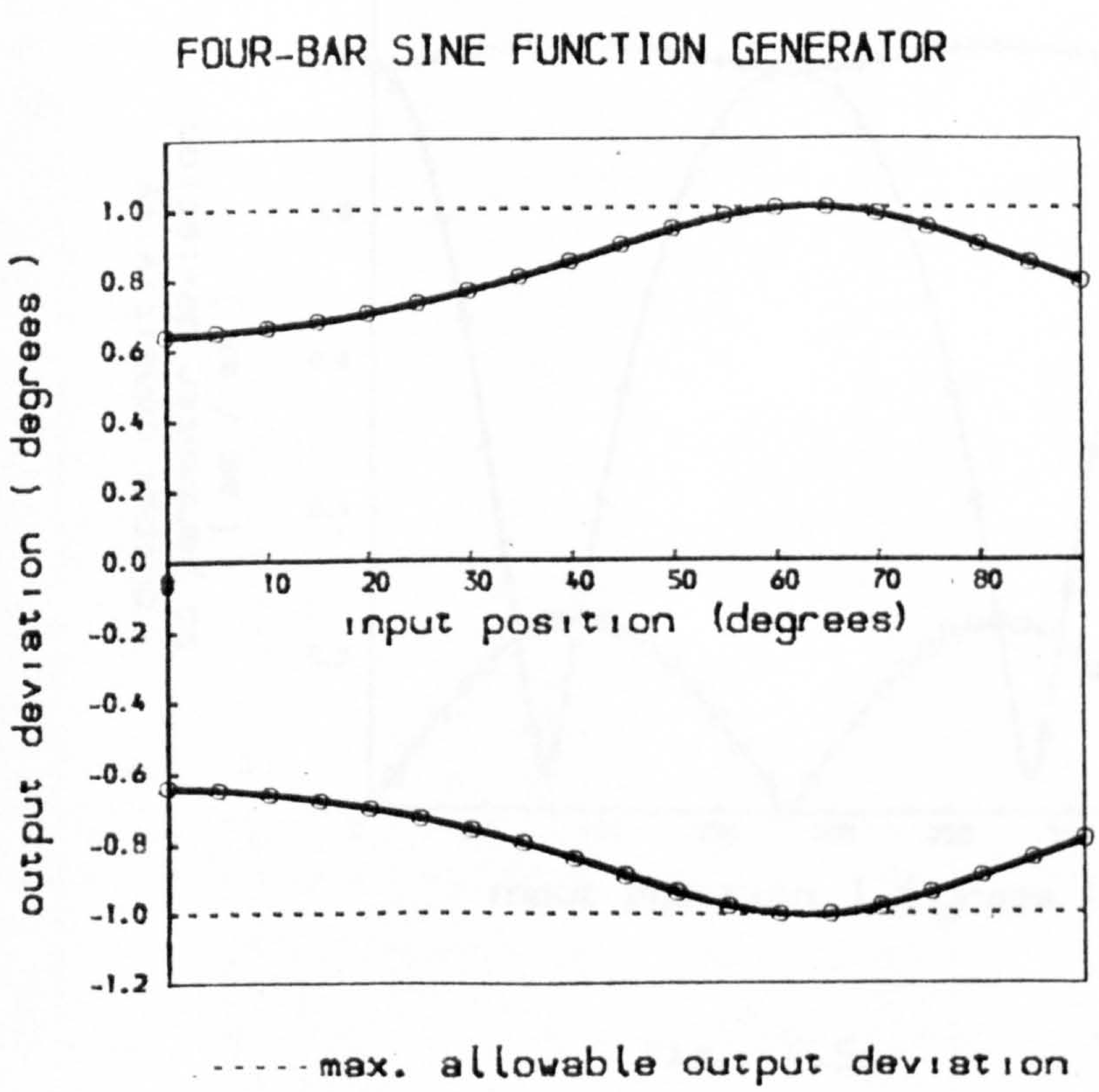


Fig. 6.3

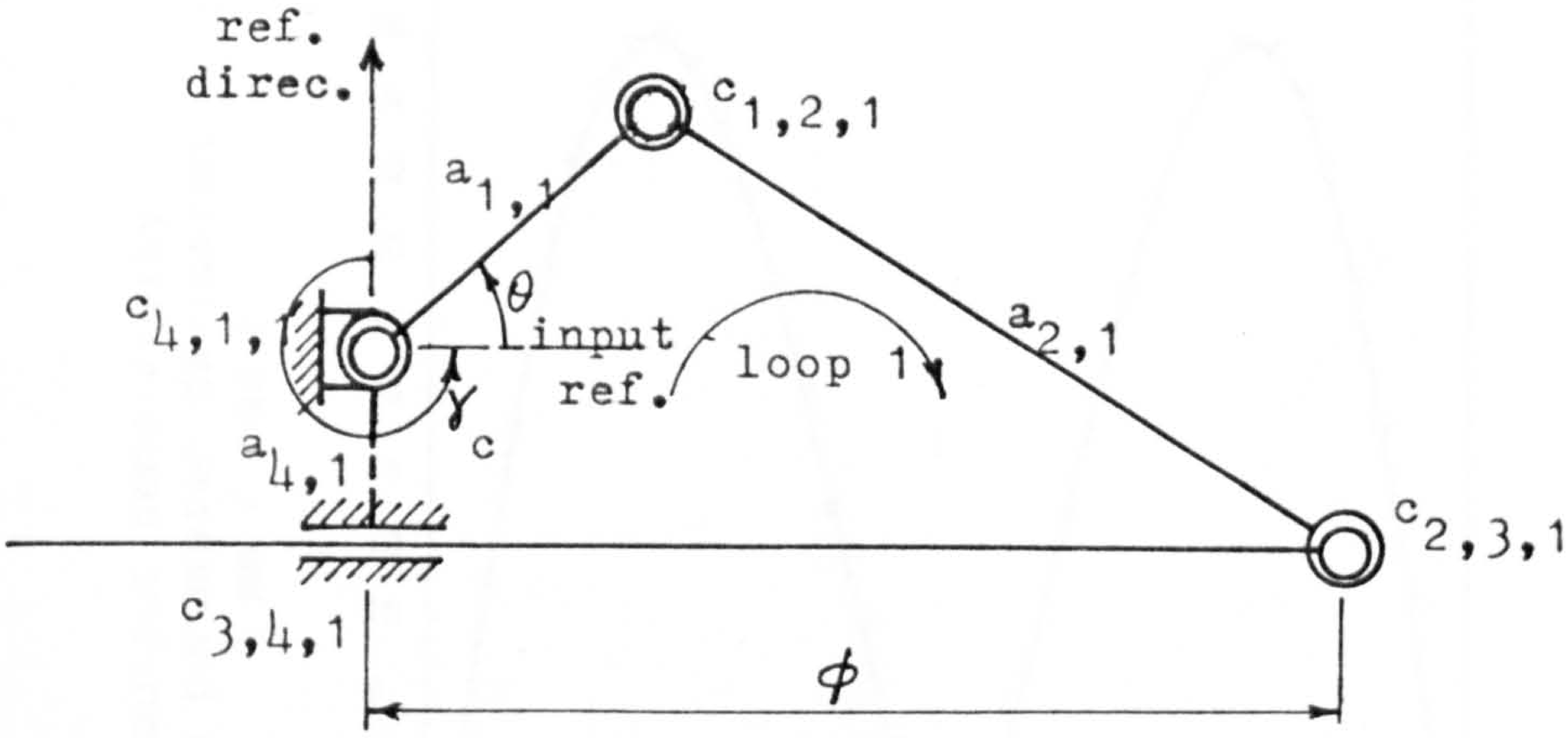


Fig. 6.4 SLIDER-CRANK MECHANISM

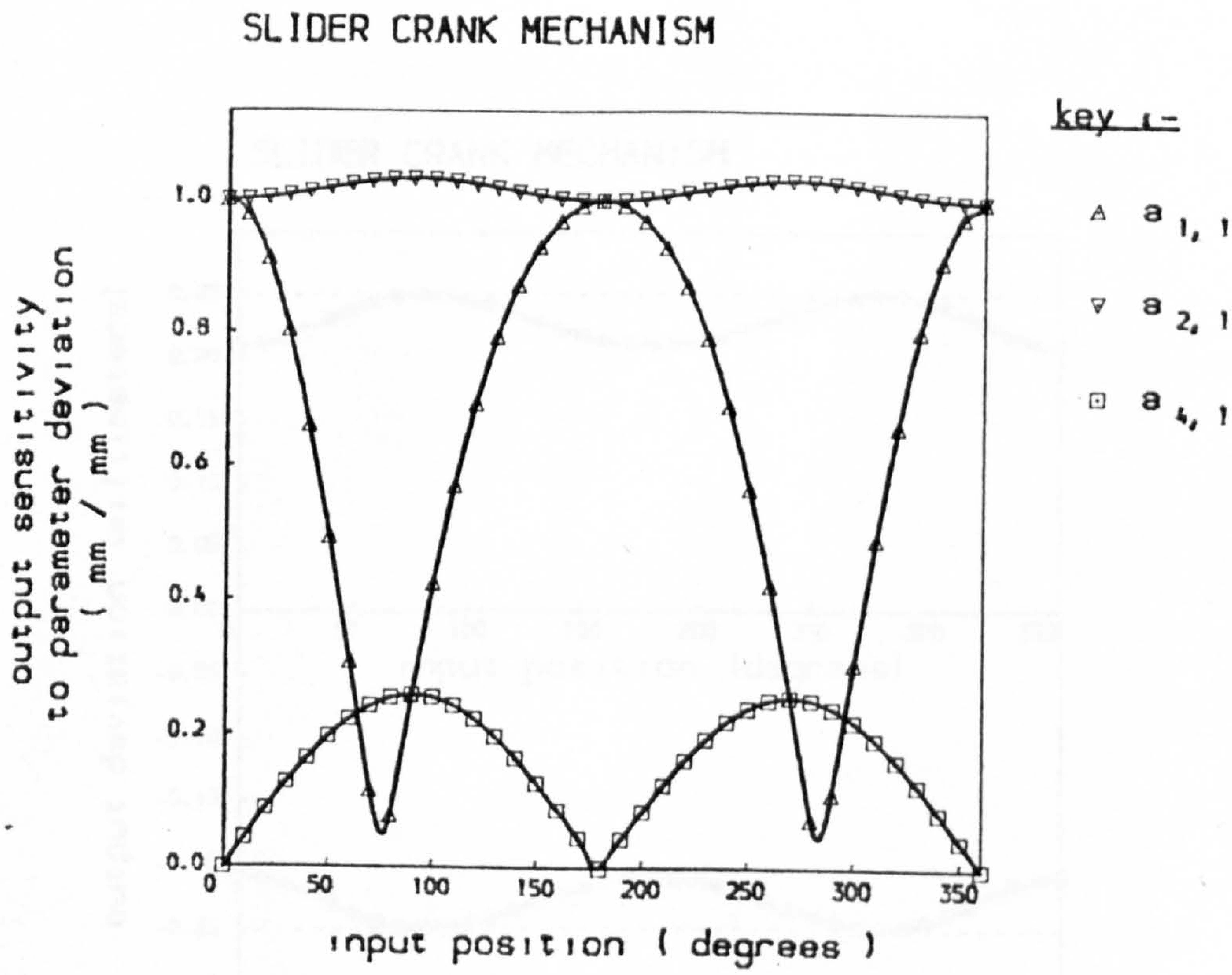


Fig. 6.5a

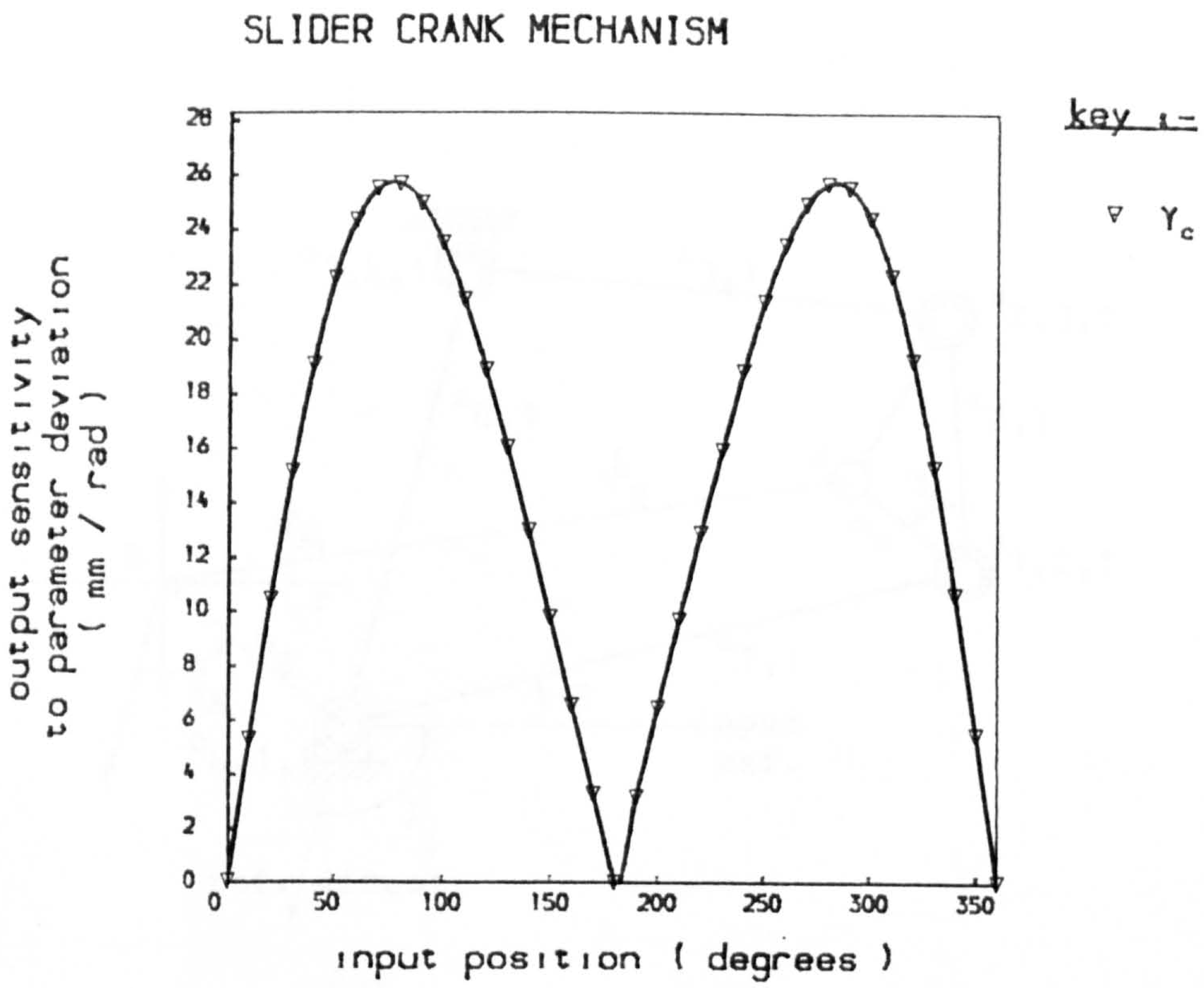


Fig. 6.5b

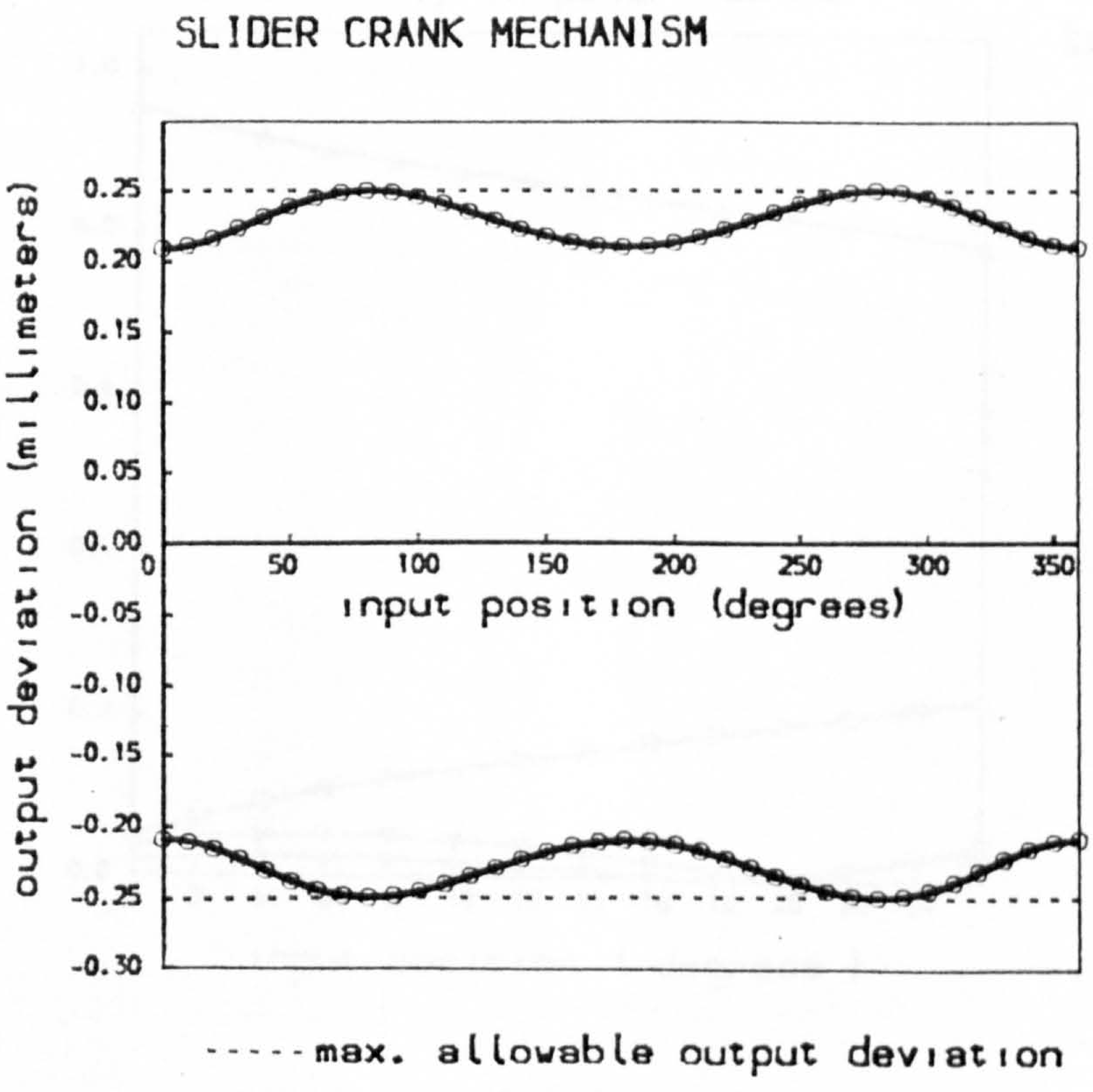


Fig. 6.6

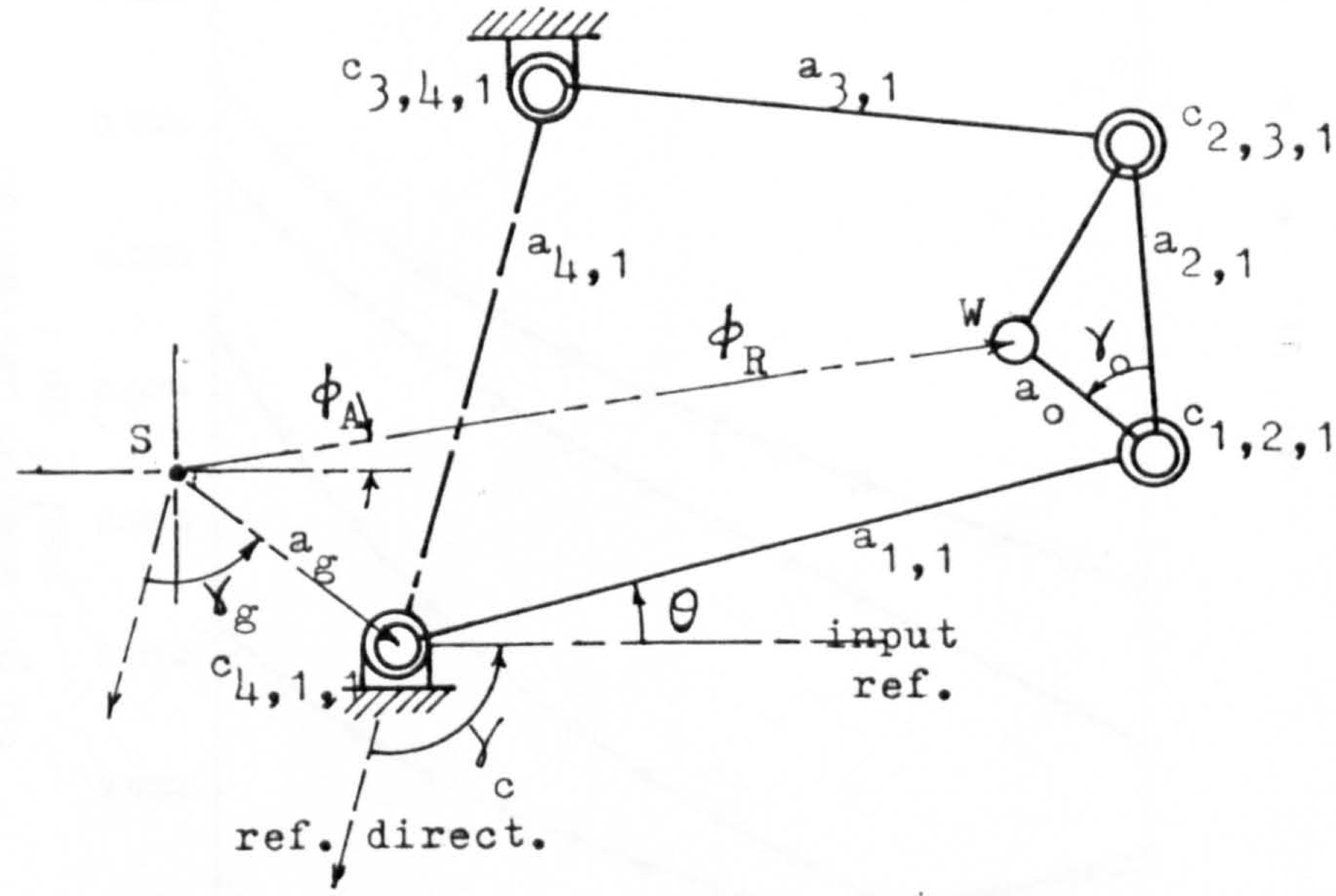


Fig. 6.7 FOUR-BAR MOTOR CYCLE REAR SUSPENSION

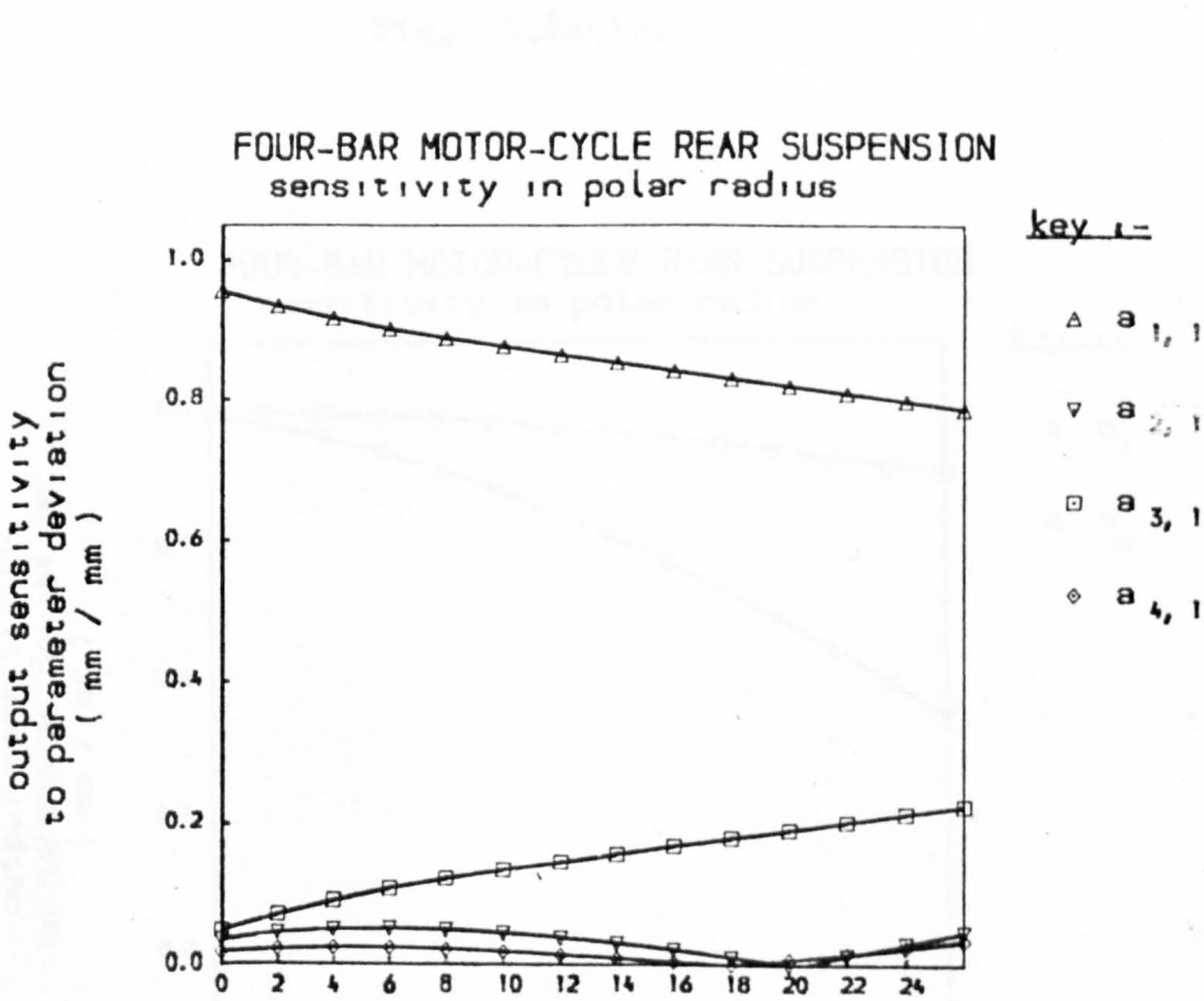


Fig. 6.8a(i)

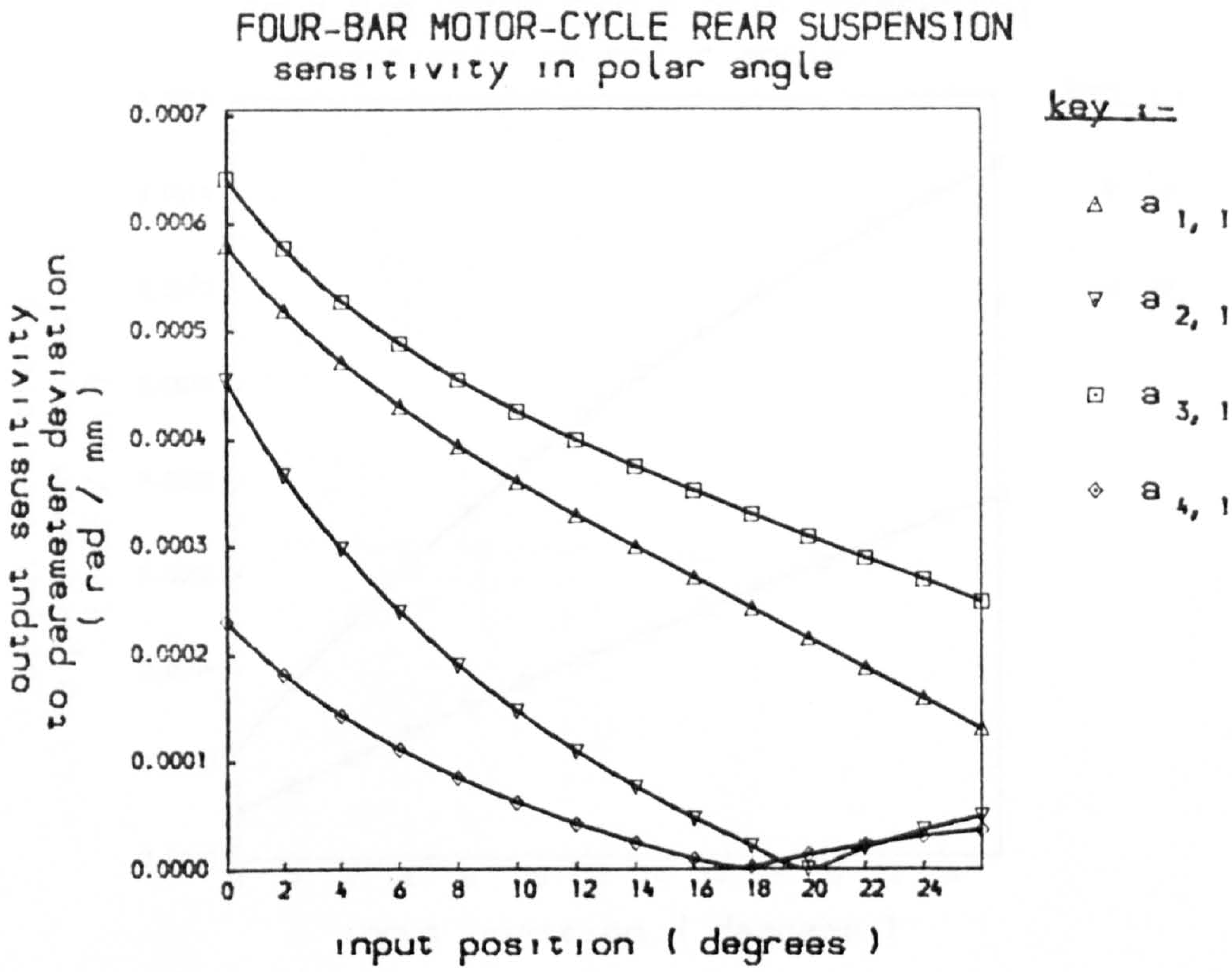


Fig. 6.8a(ii)

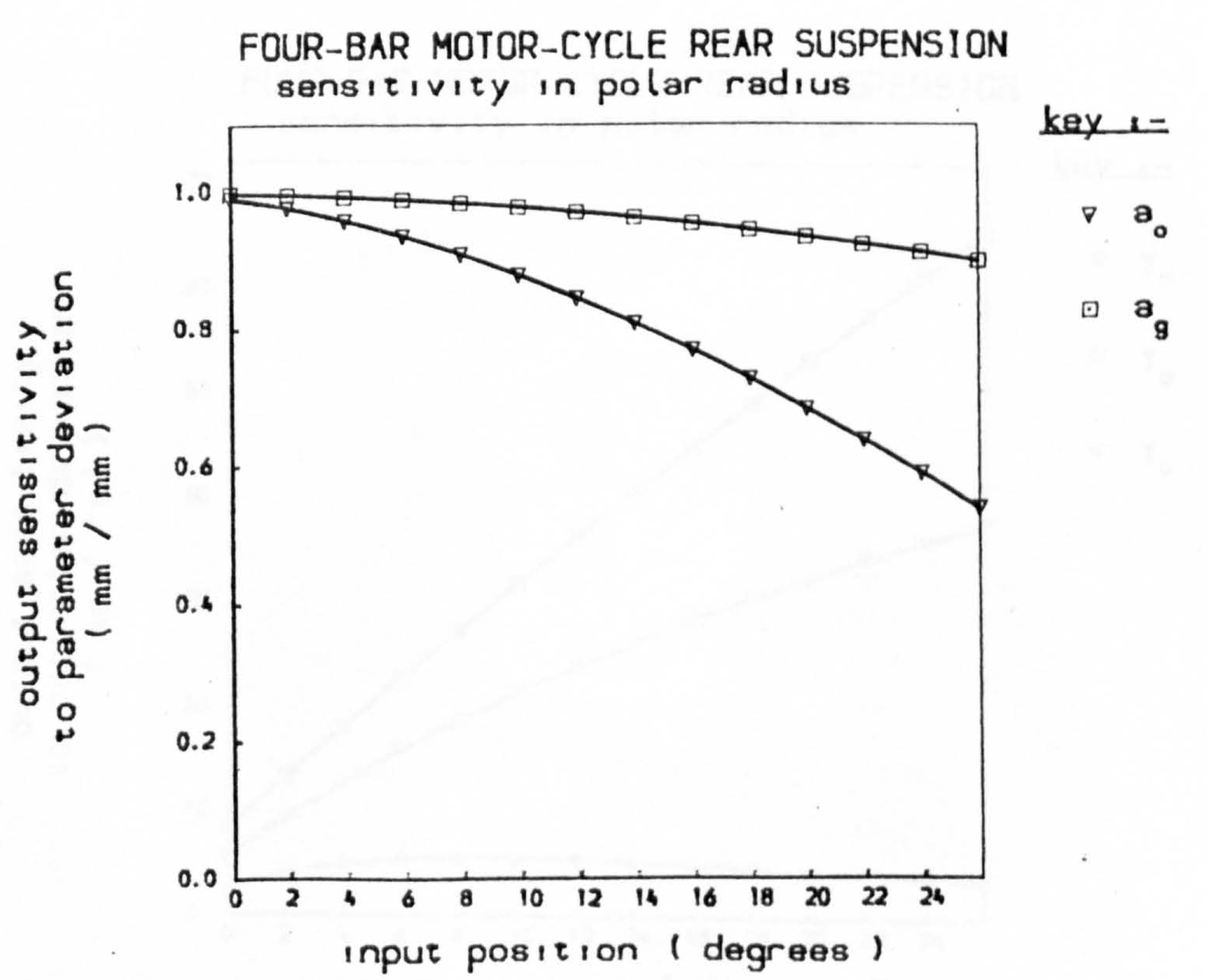


Fig. 6.8b(i)

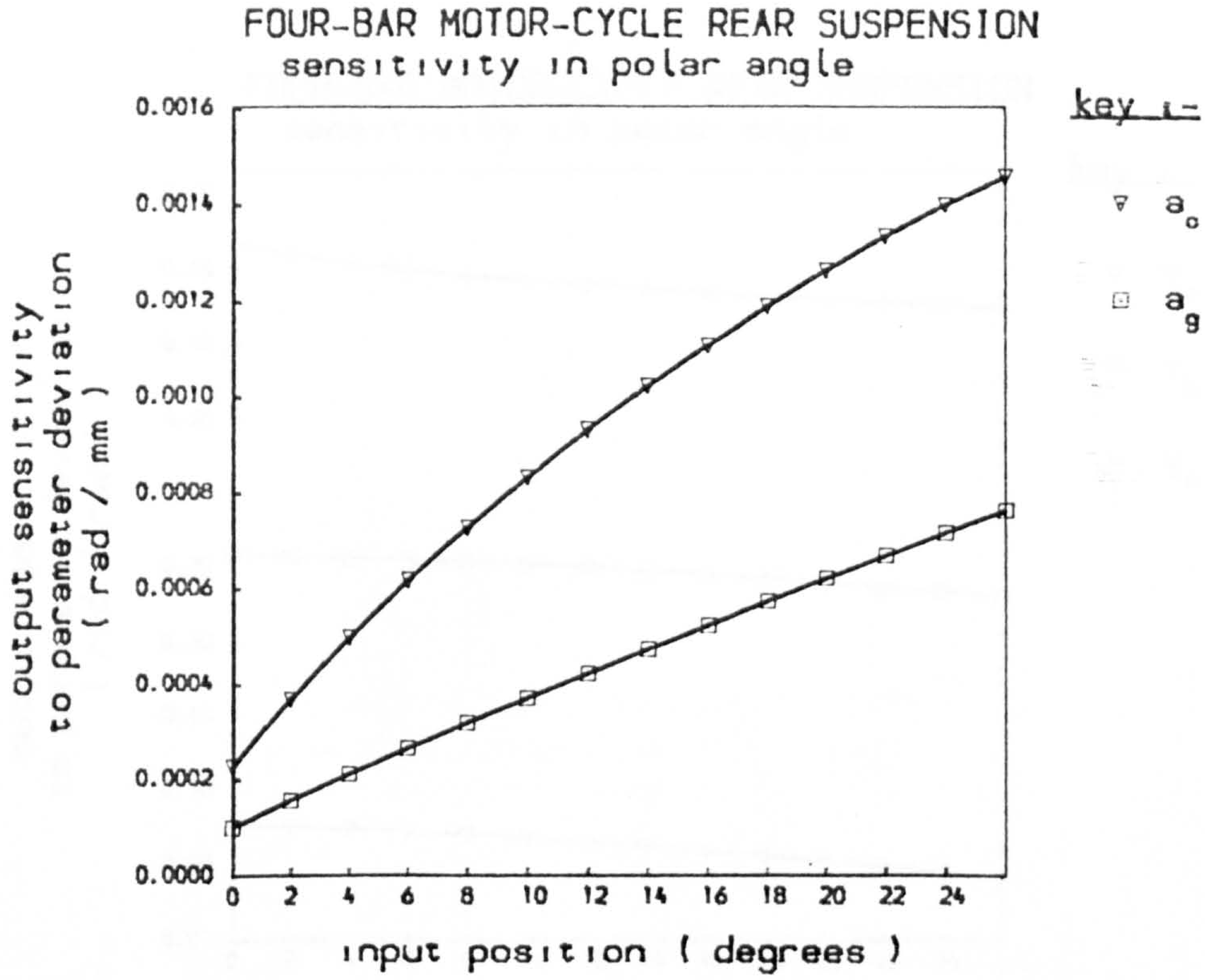


Fig. 6.8b(ii)

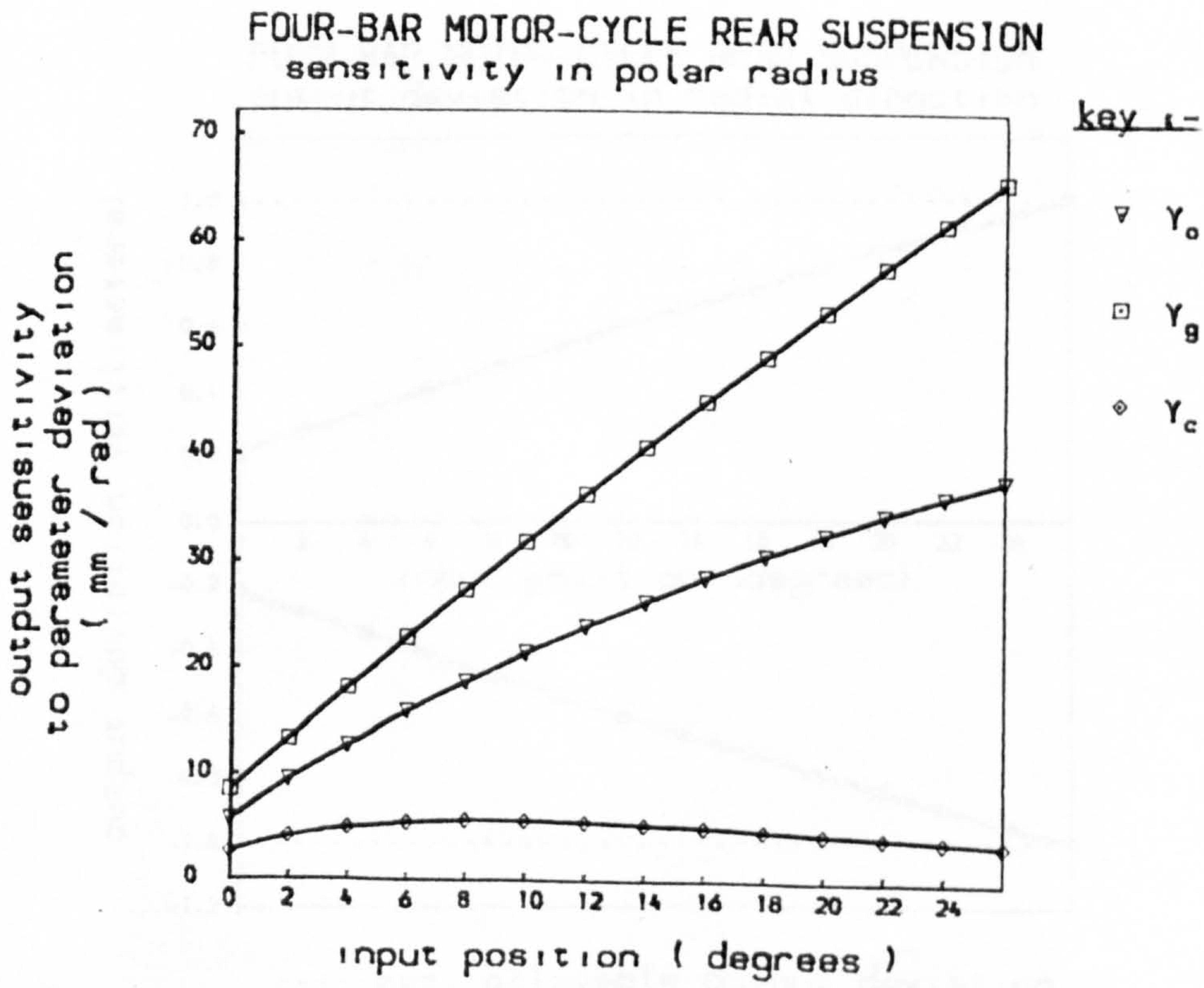


Fig. 6.8c(i)

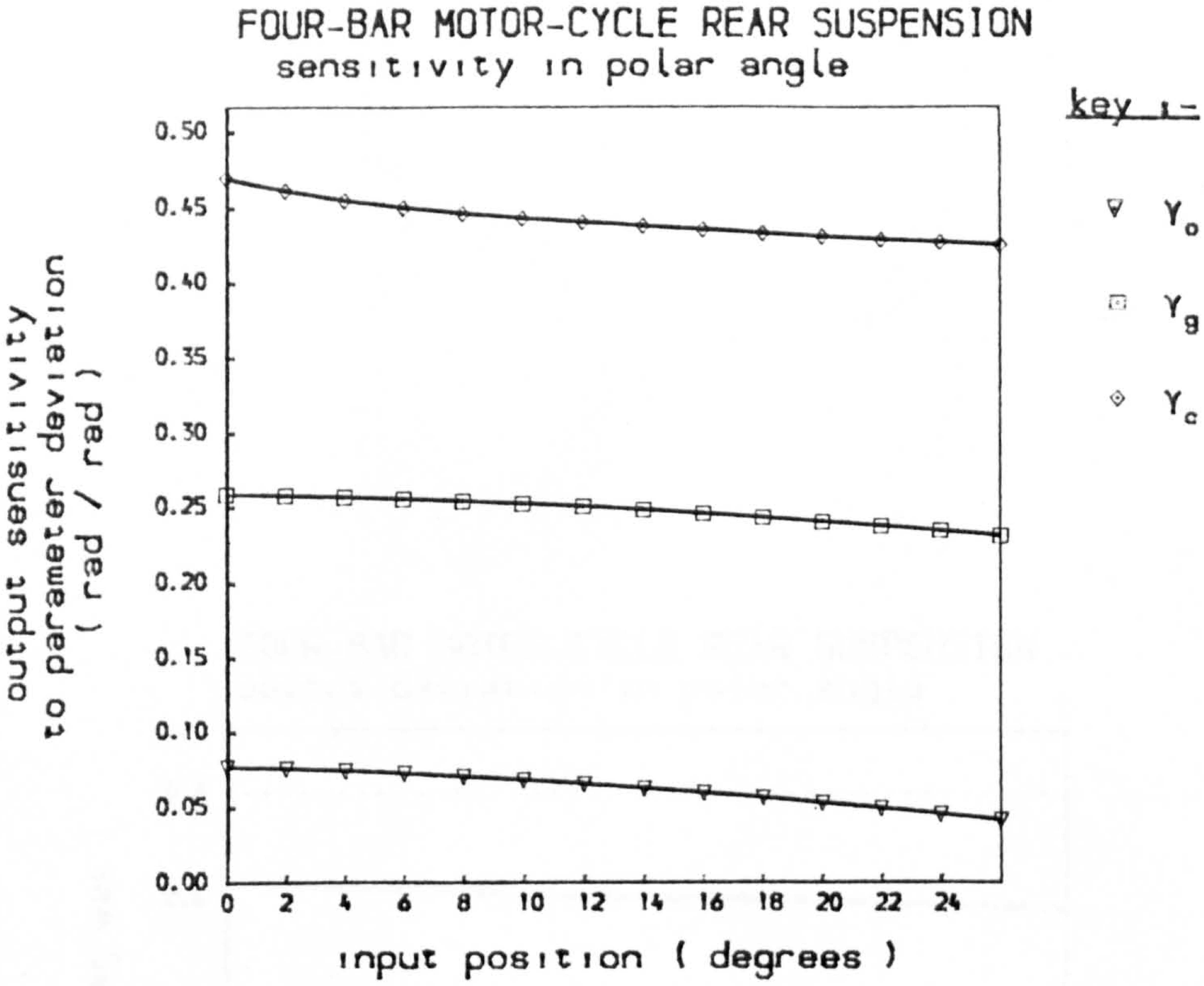


Fig. 6.8c(ii)

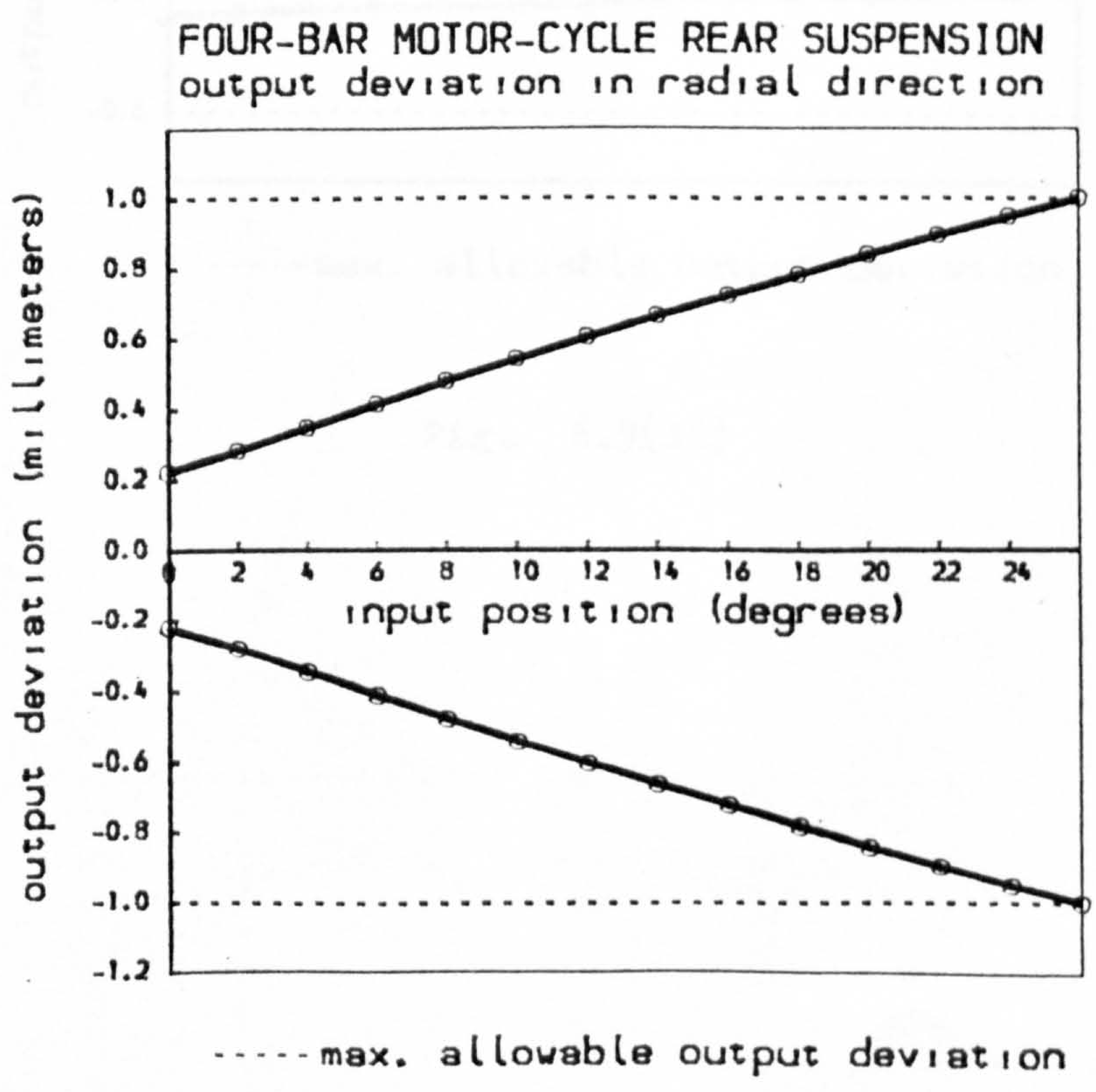
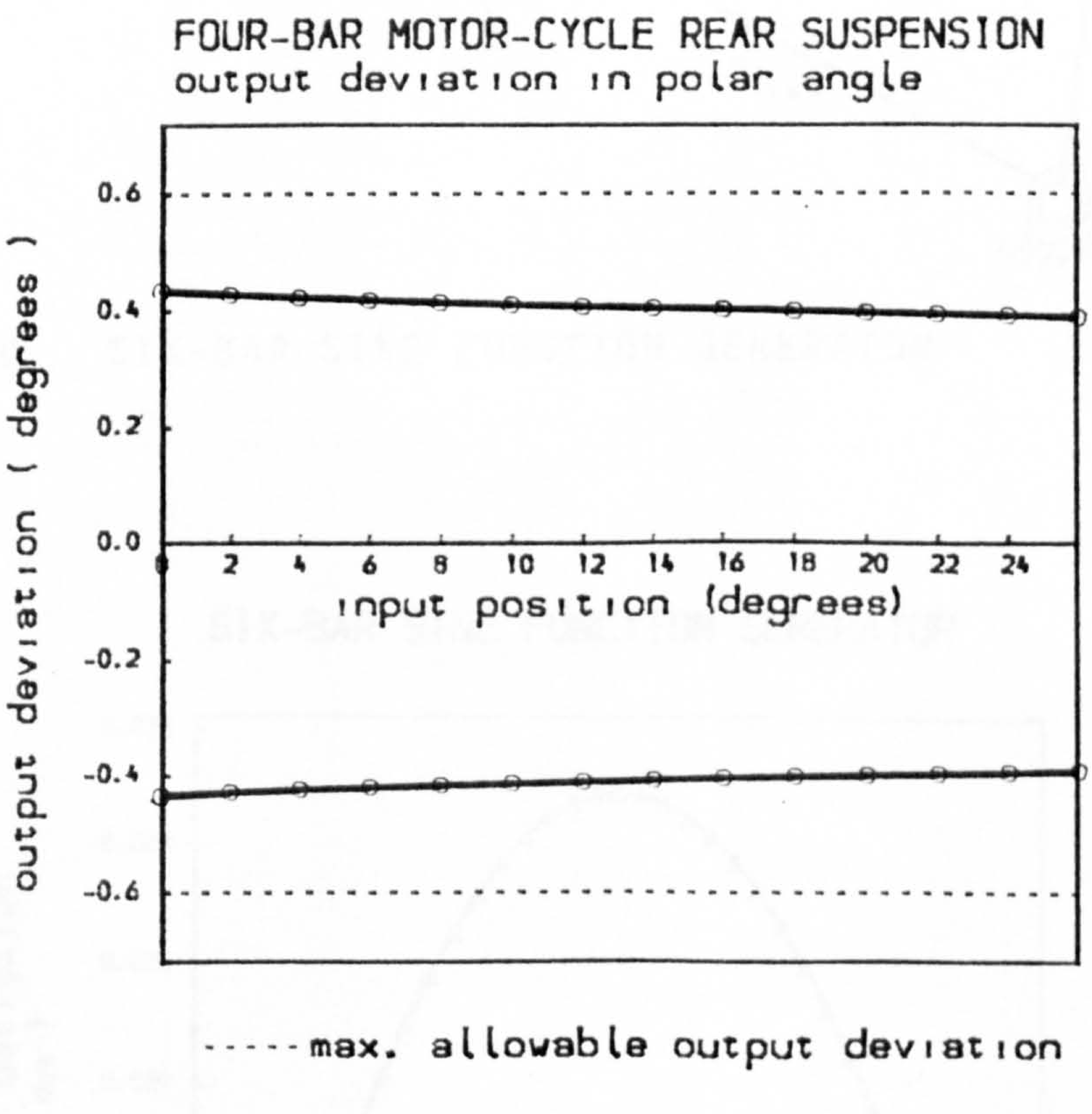


Fig. 6.9(i)



FOUR-BAR MOTOR-CYCLE REAR SUSPENSION
output deviation in polar angle
output deviation (degrees)
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
0
2
4
6
8
10
12
14
16
18
20
22
24
input position (degrees)
max. allowable output deviation

Fig. 6.9(ii)

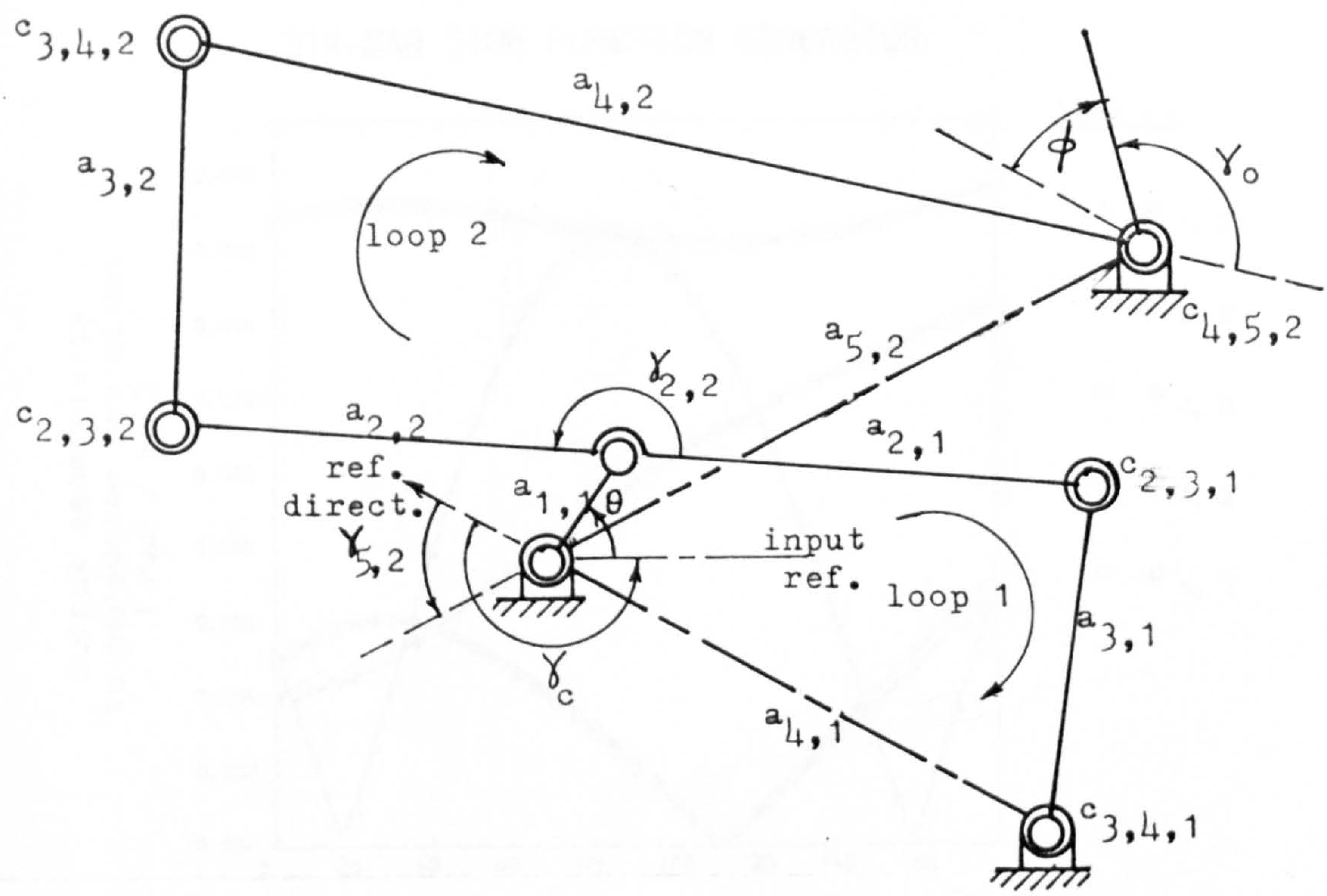


Fig. 6.10 SIX-BAR SINE FUNCTION GENERATOR

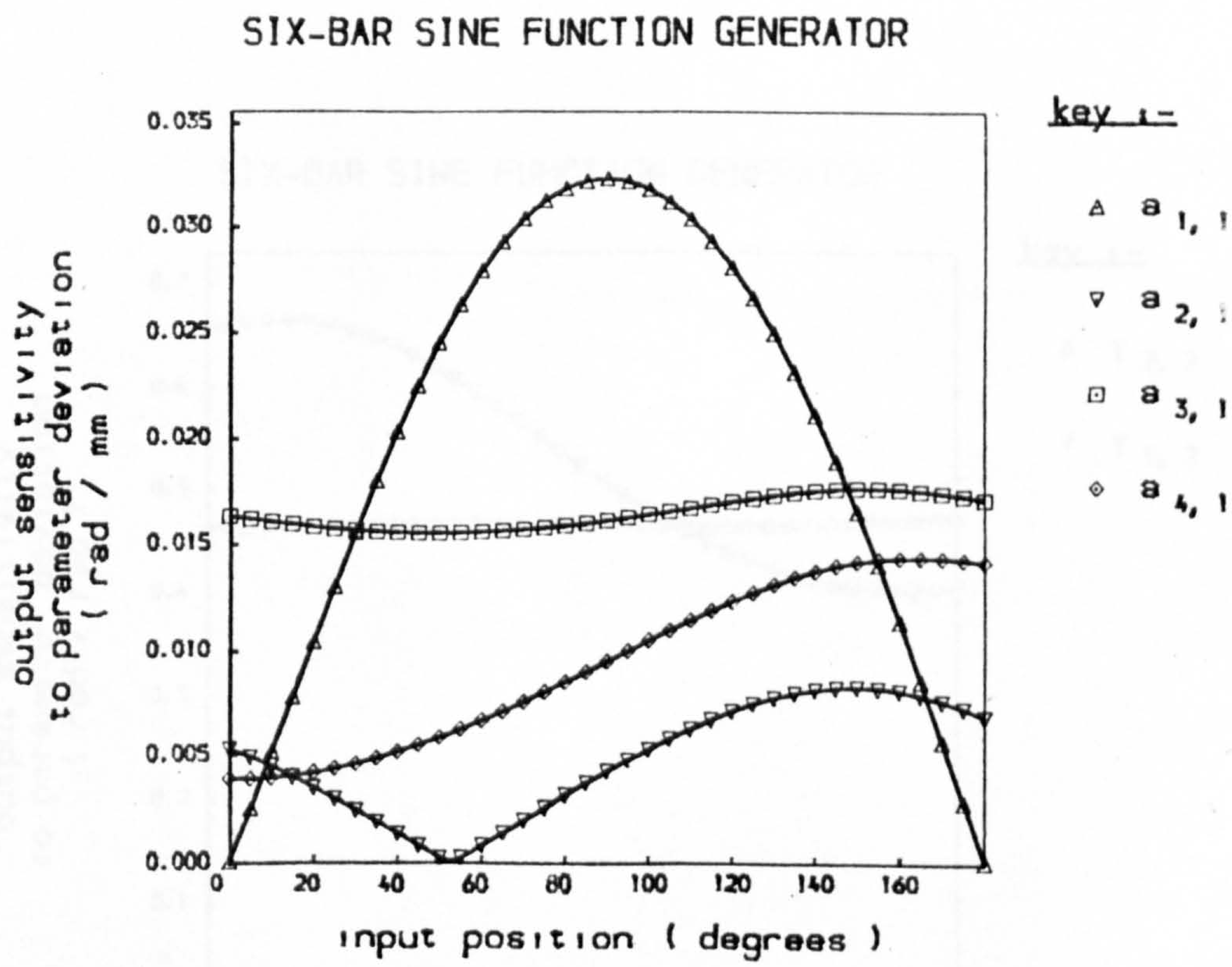


Fig. 6.11a

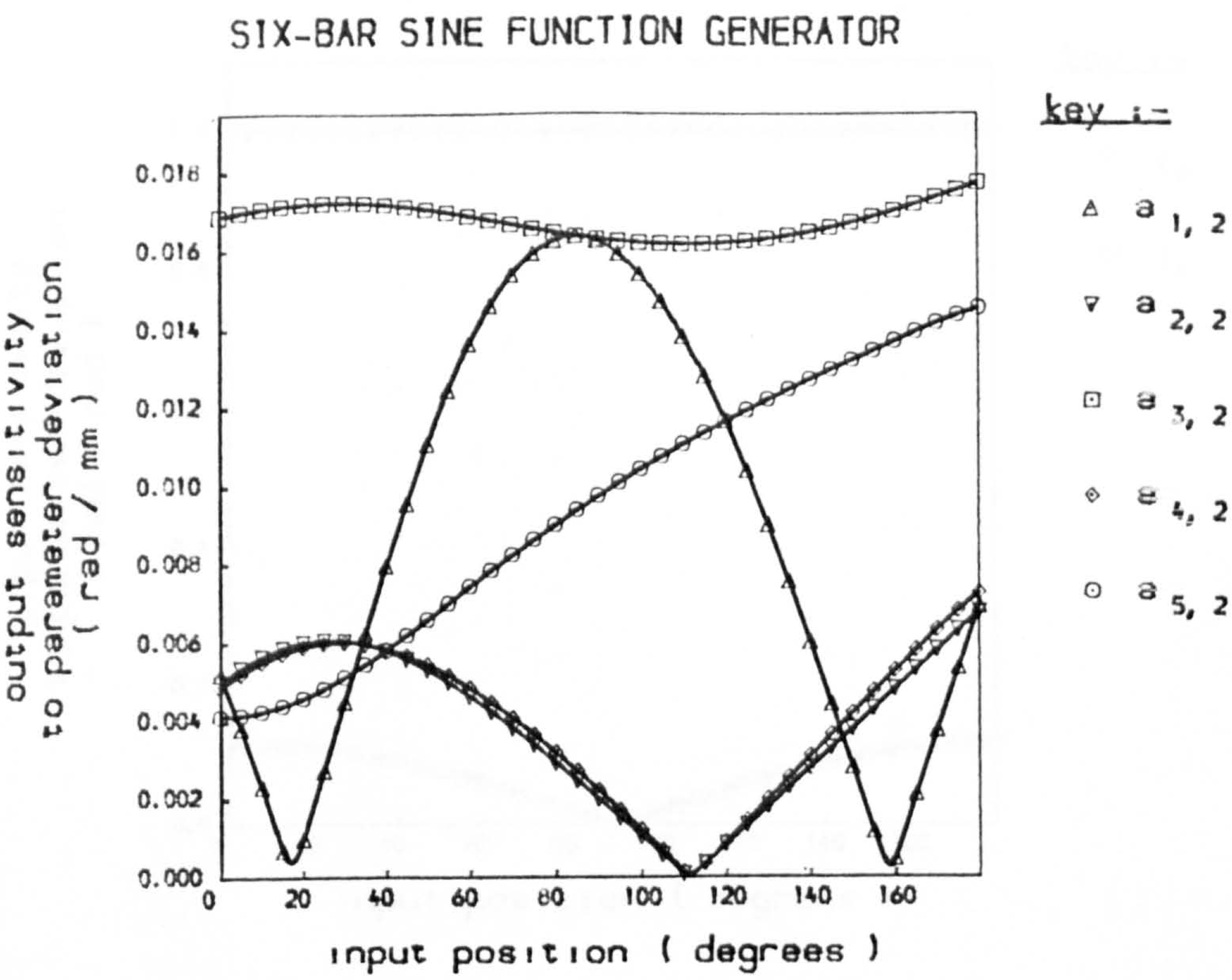


Fig. 6.11b

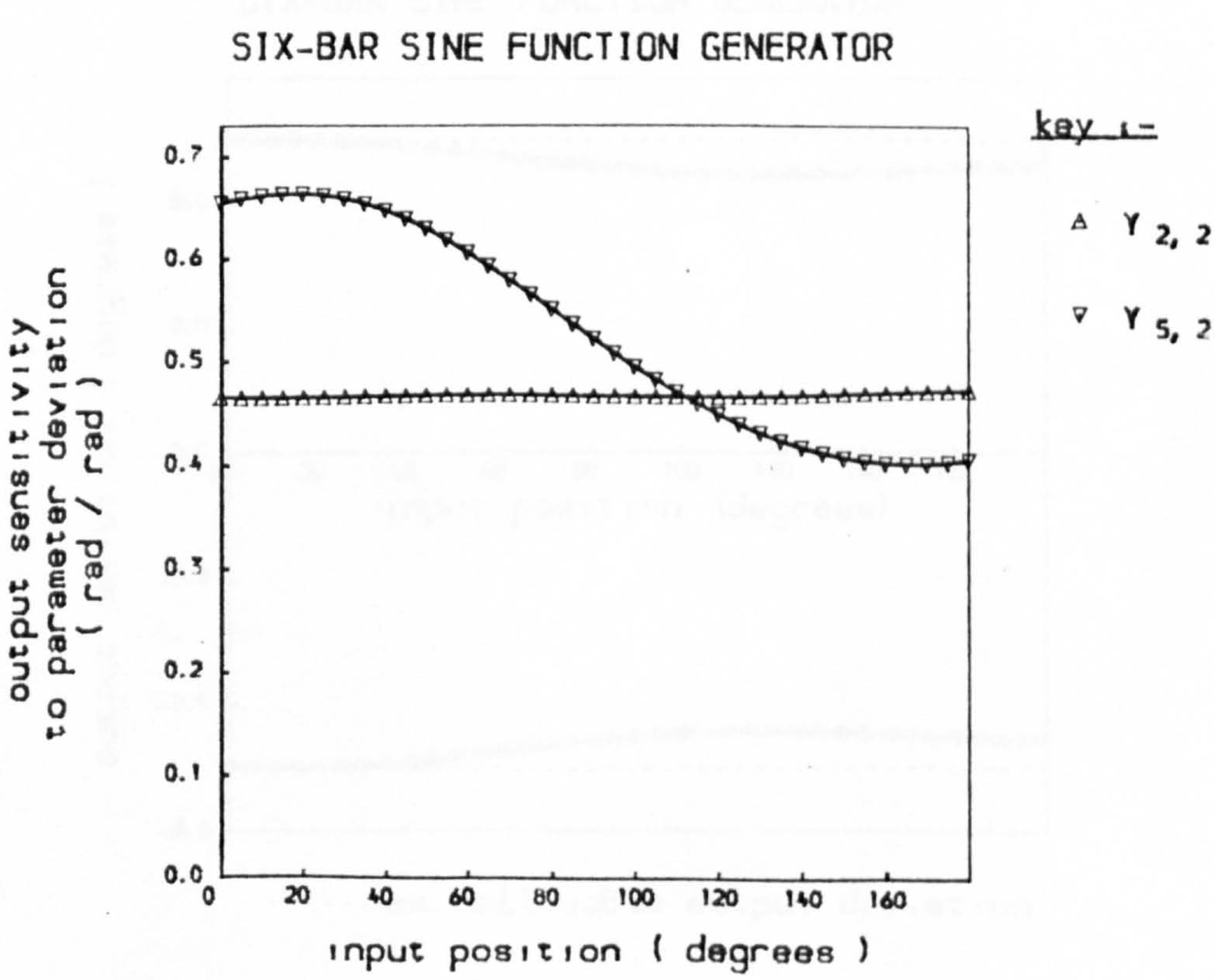


Fig. 6.11c

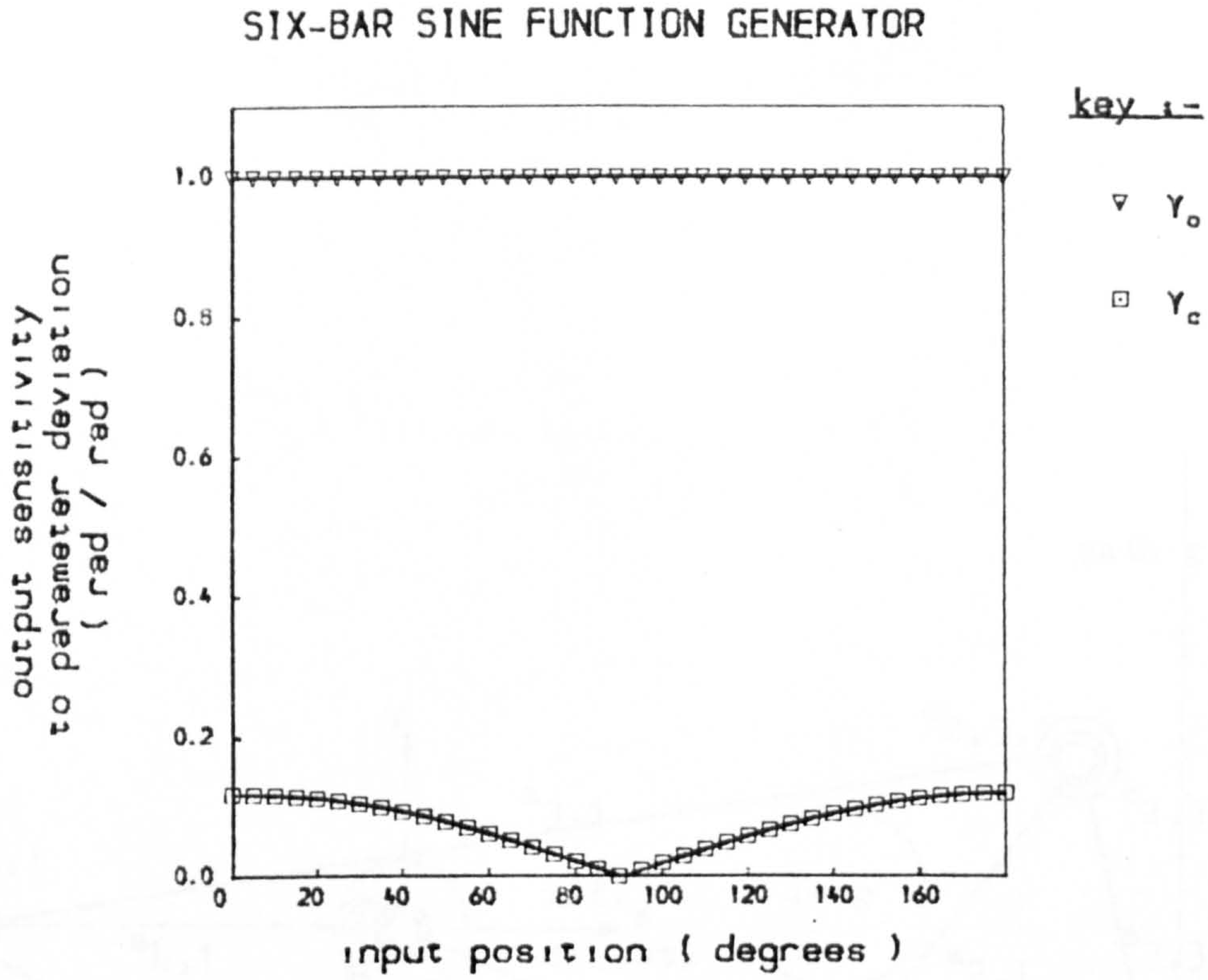


Fig. 6.11d

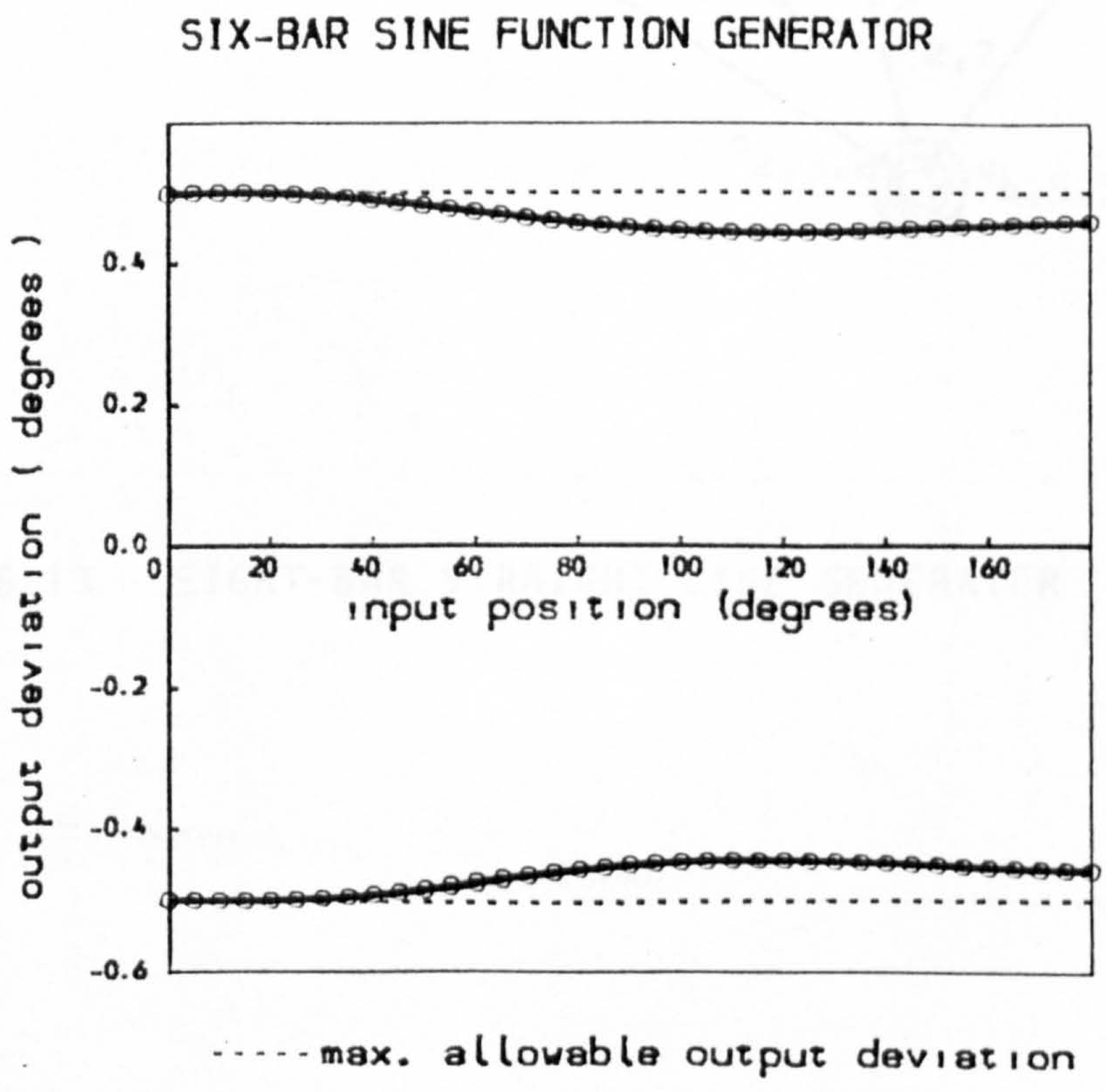


Fig. 6.12

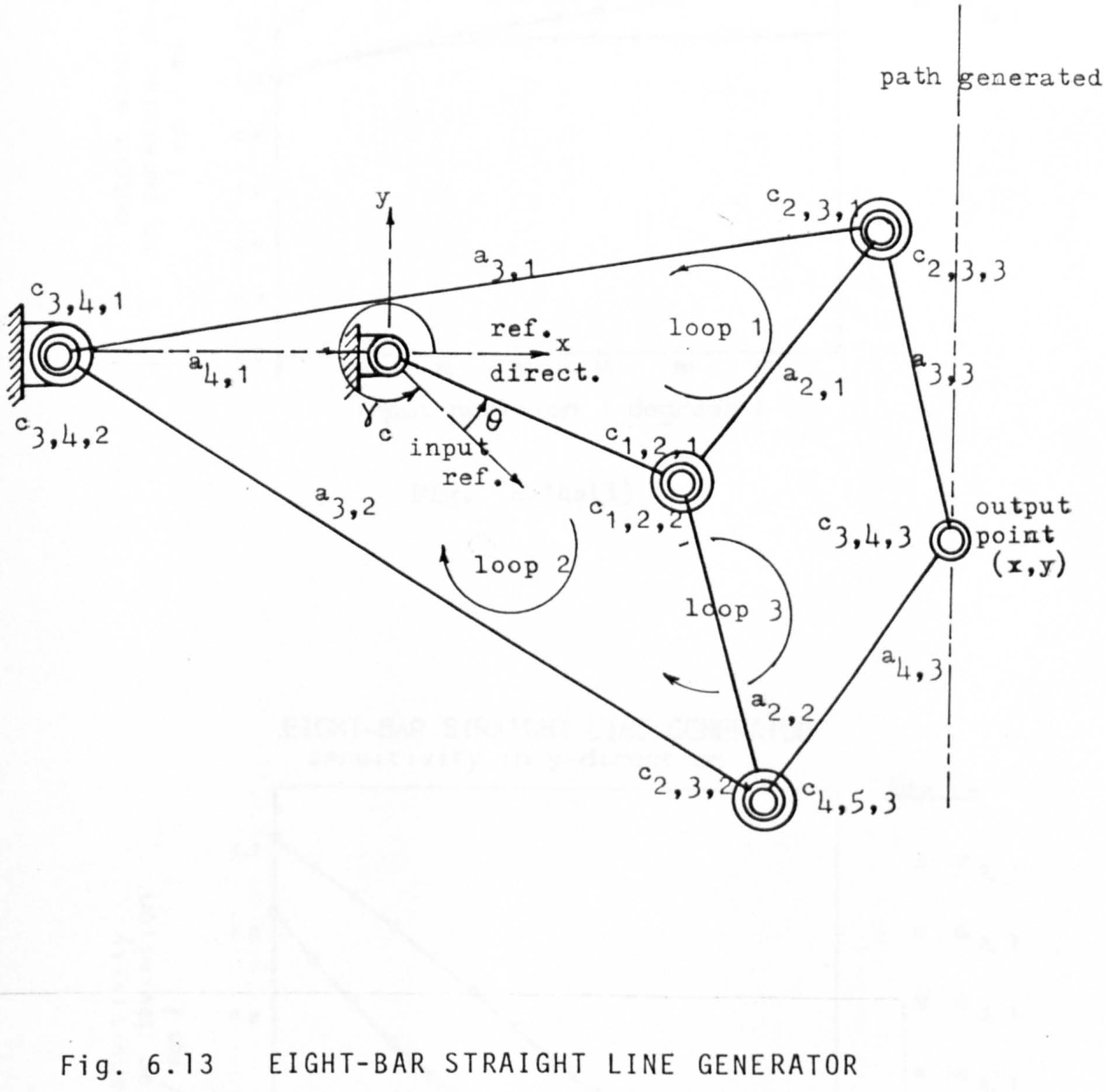


Fig. 6.13 EIGHT-BAR STRAIGHT LINE GENERATOR

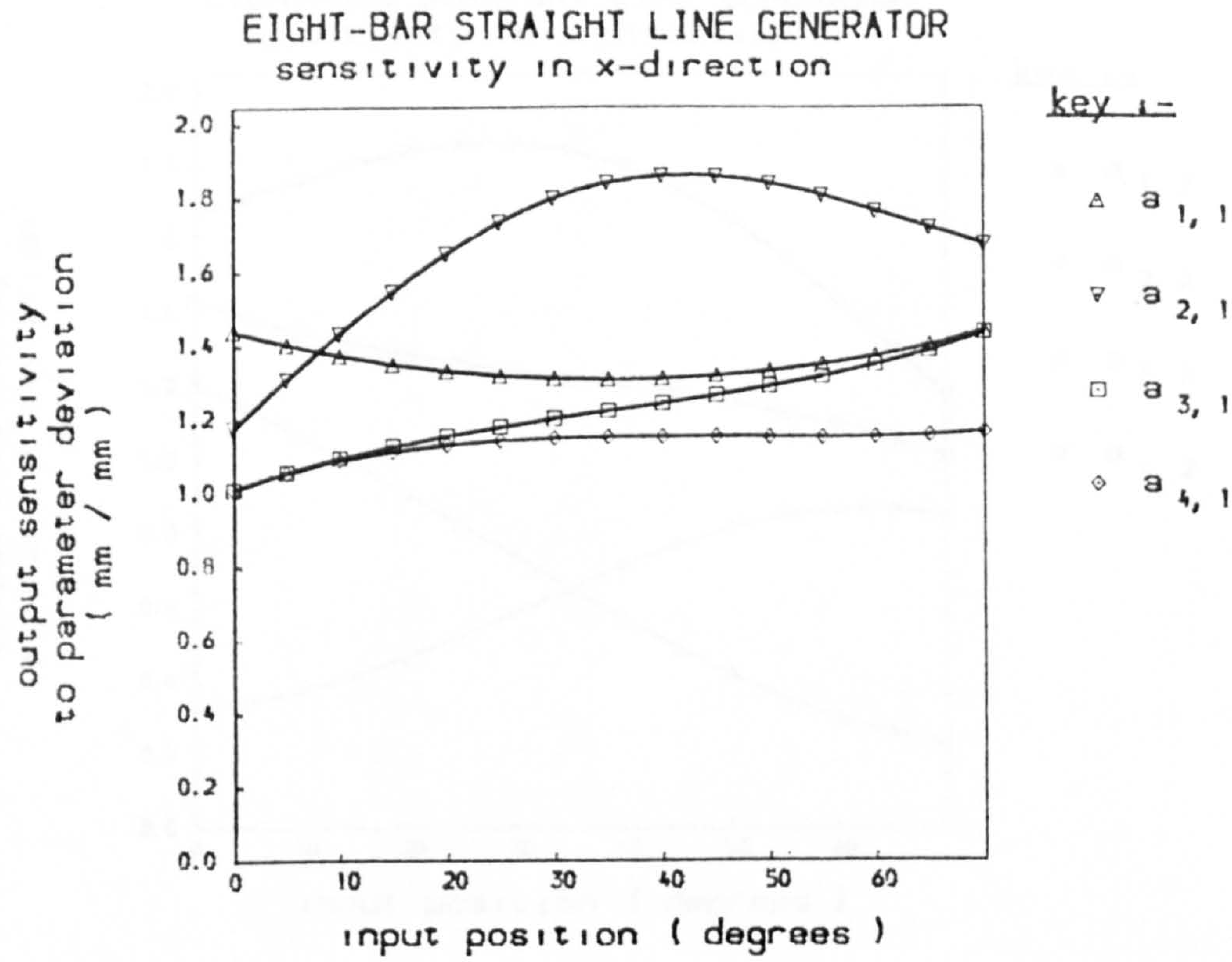


Fig. 6.14a(i)

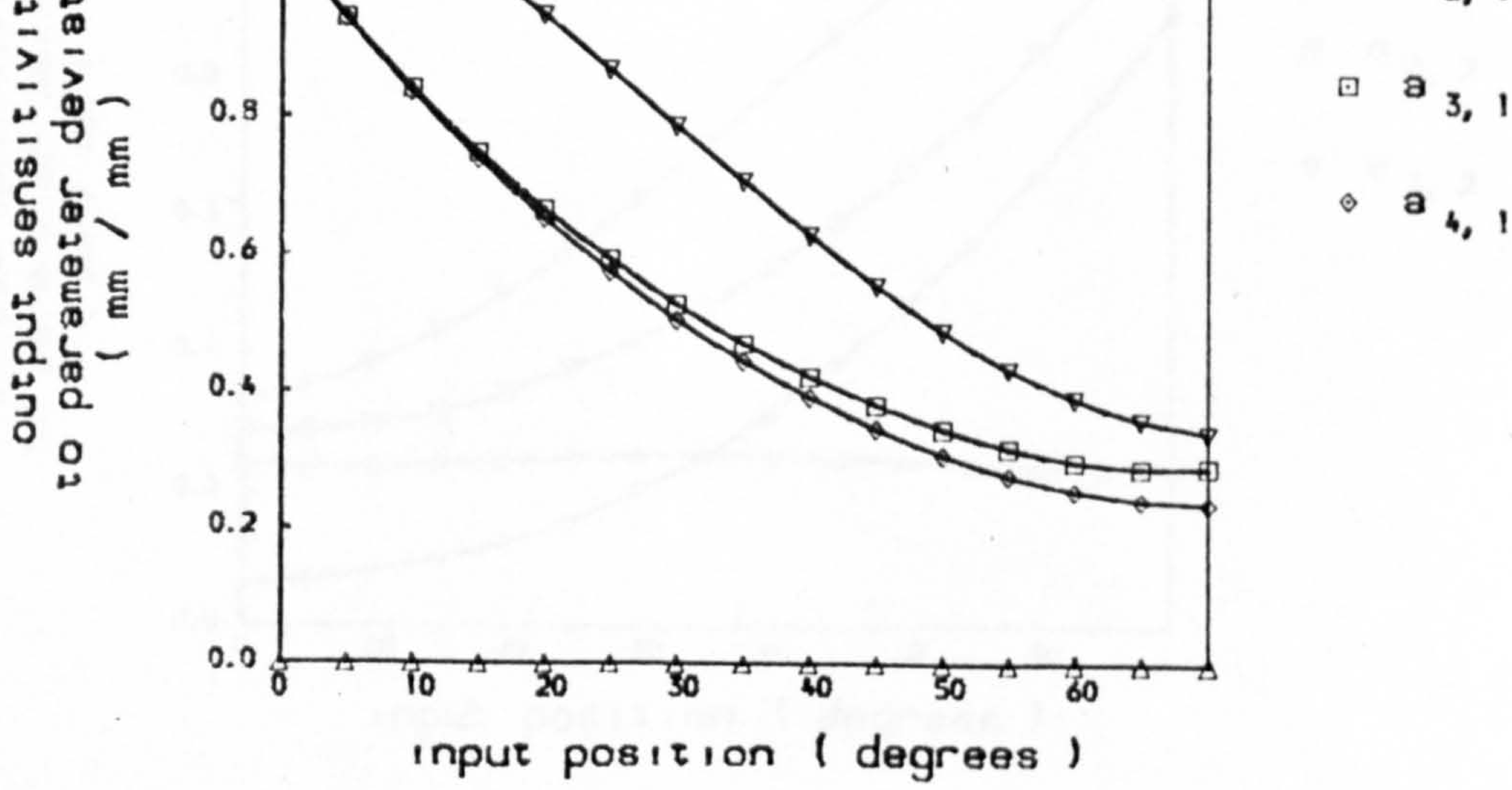


Fig. 6.14a(ii)

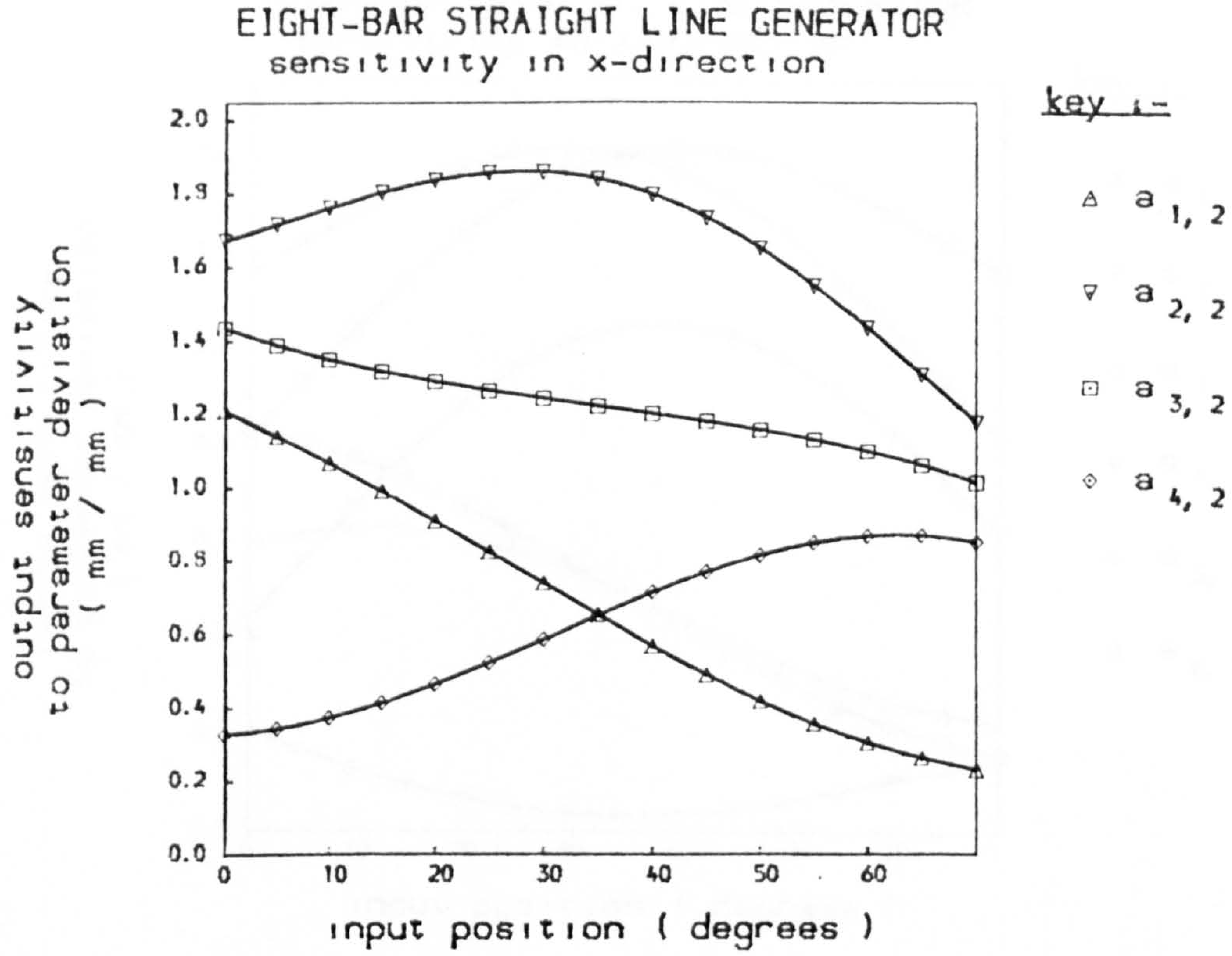


Fig. 6.14b(i)

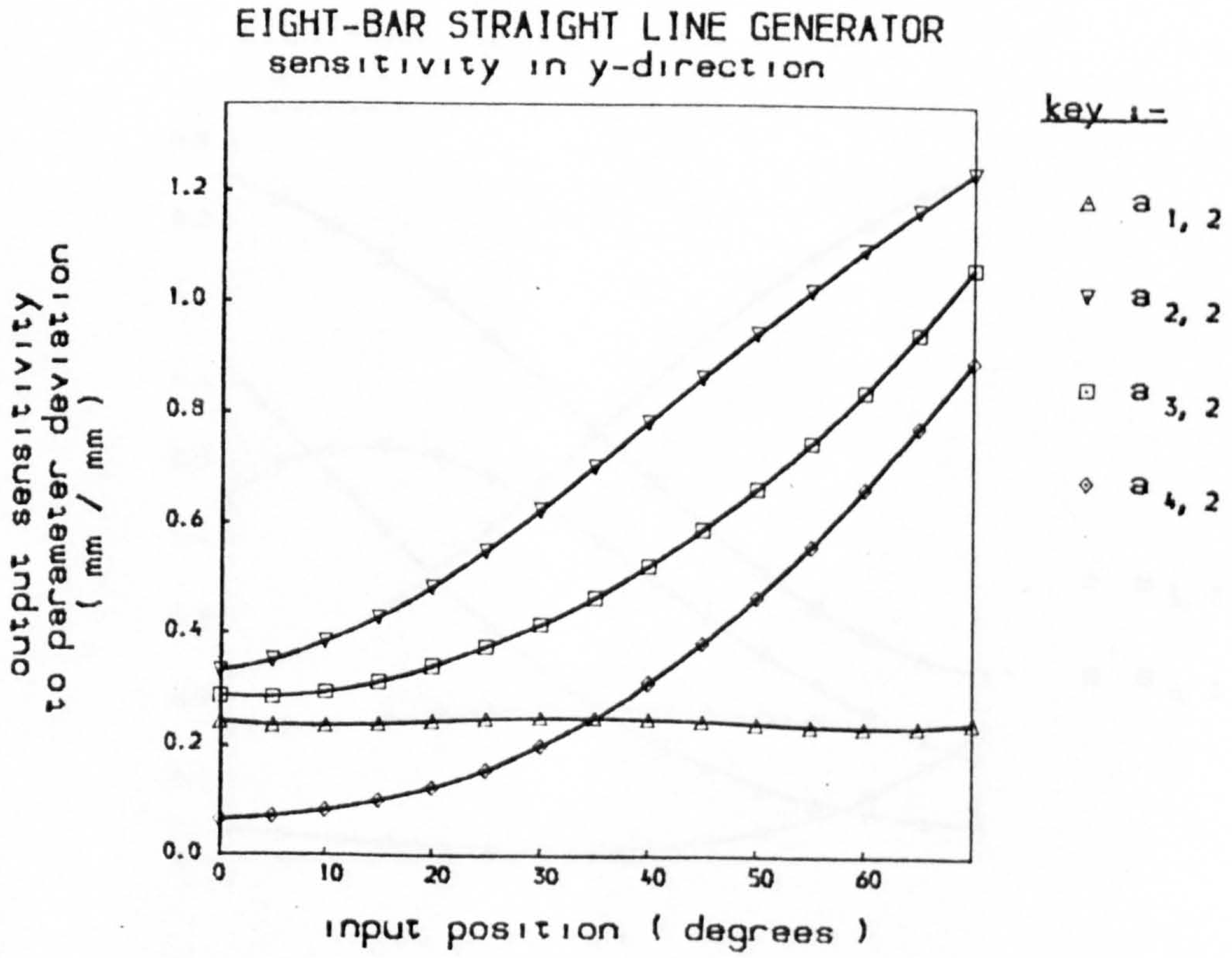


Fig. 6.14b(ii)

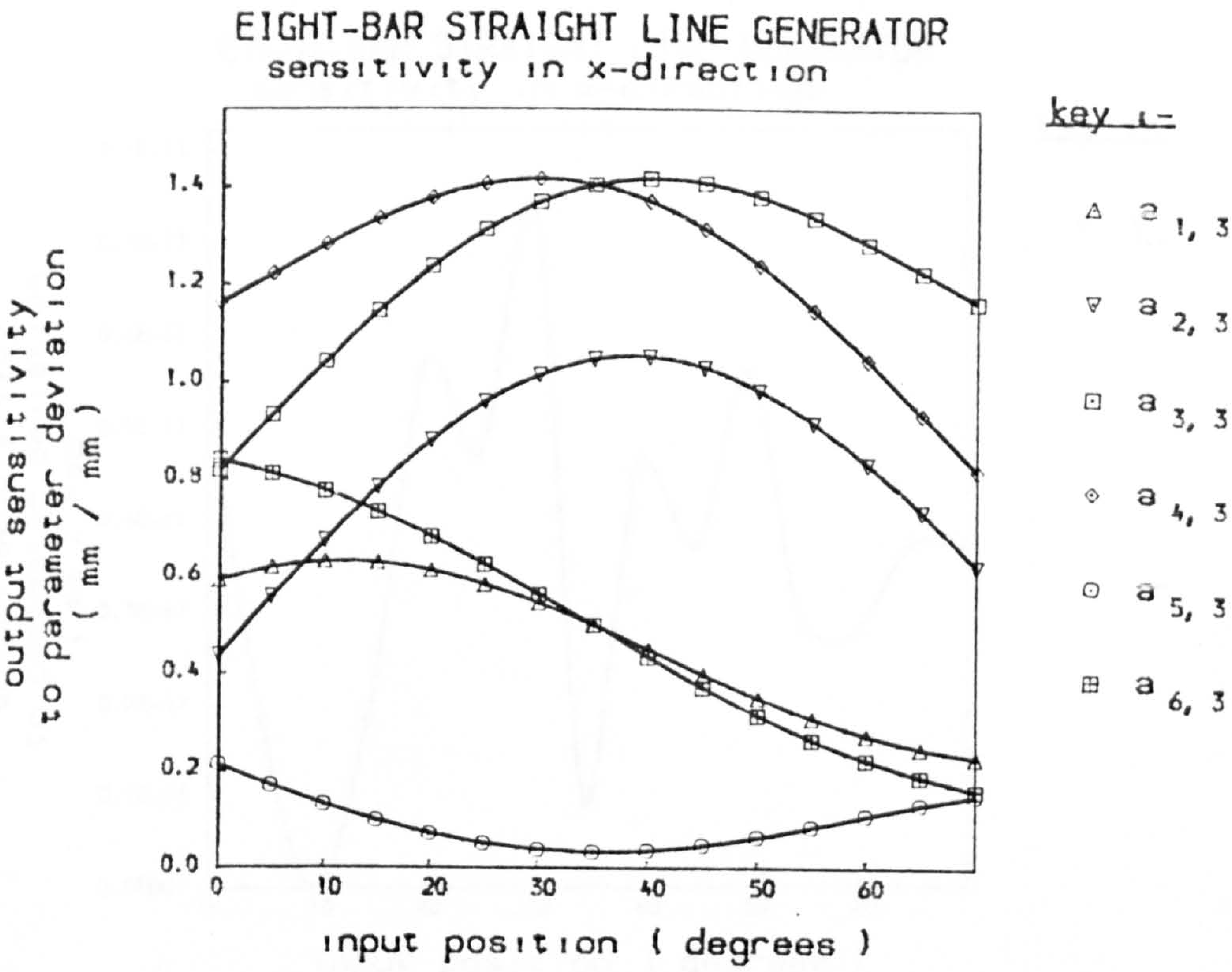


Fig. 6.14c(i)

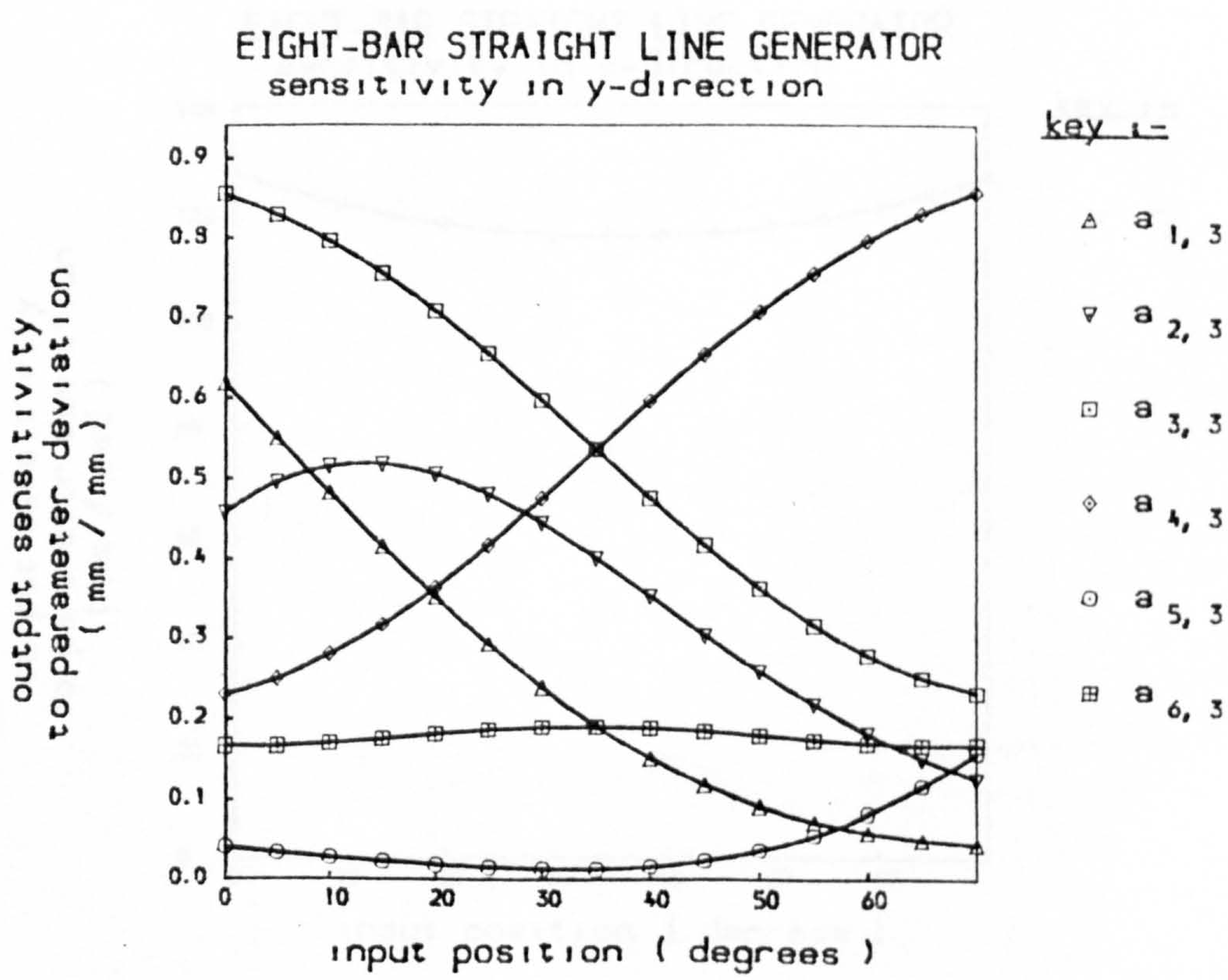


Fig. 6.14c(ii)

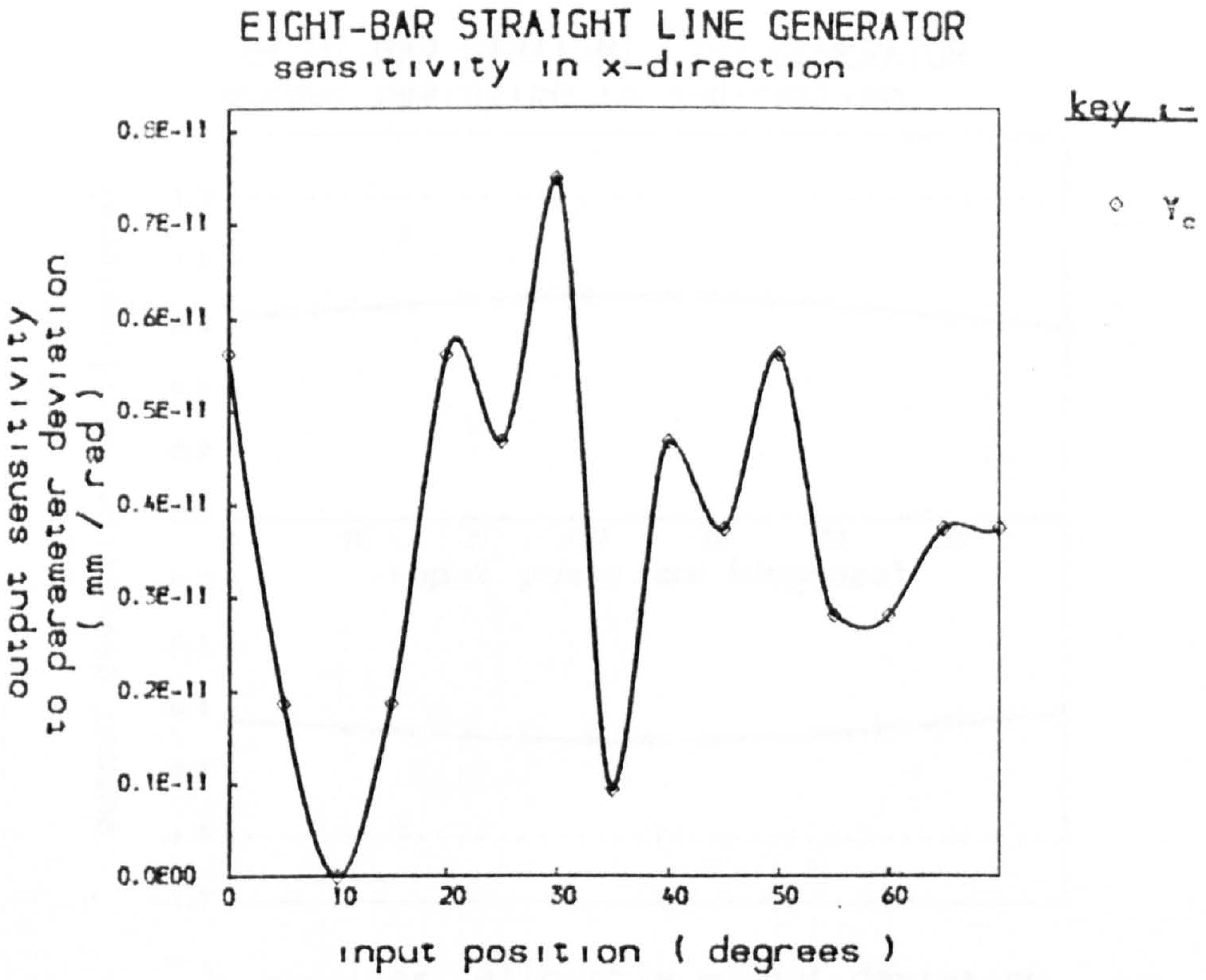


Fig. 6.14d(i)

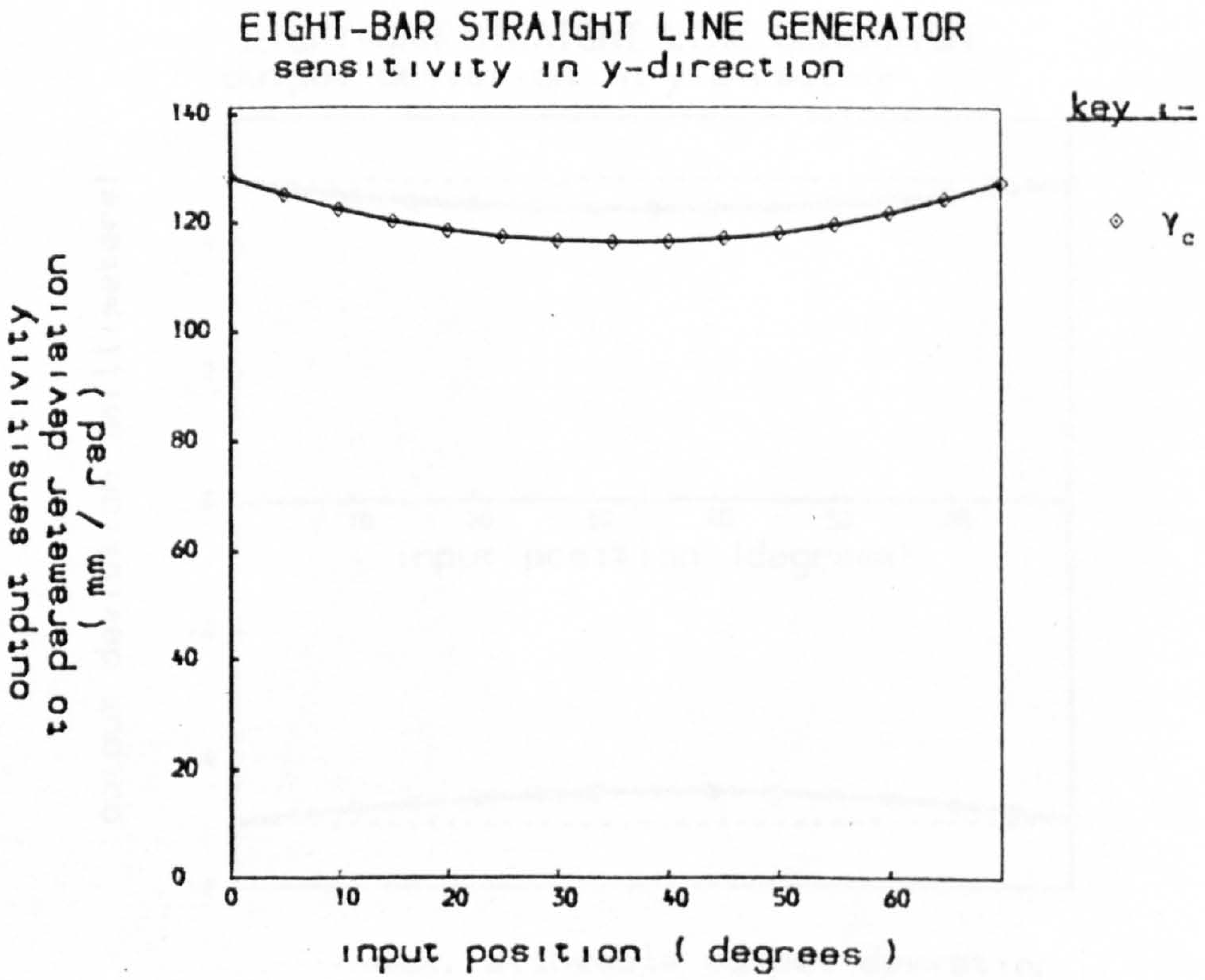


Fig. 6.14d(ii)

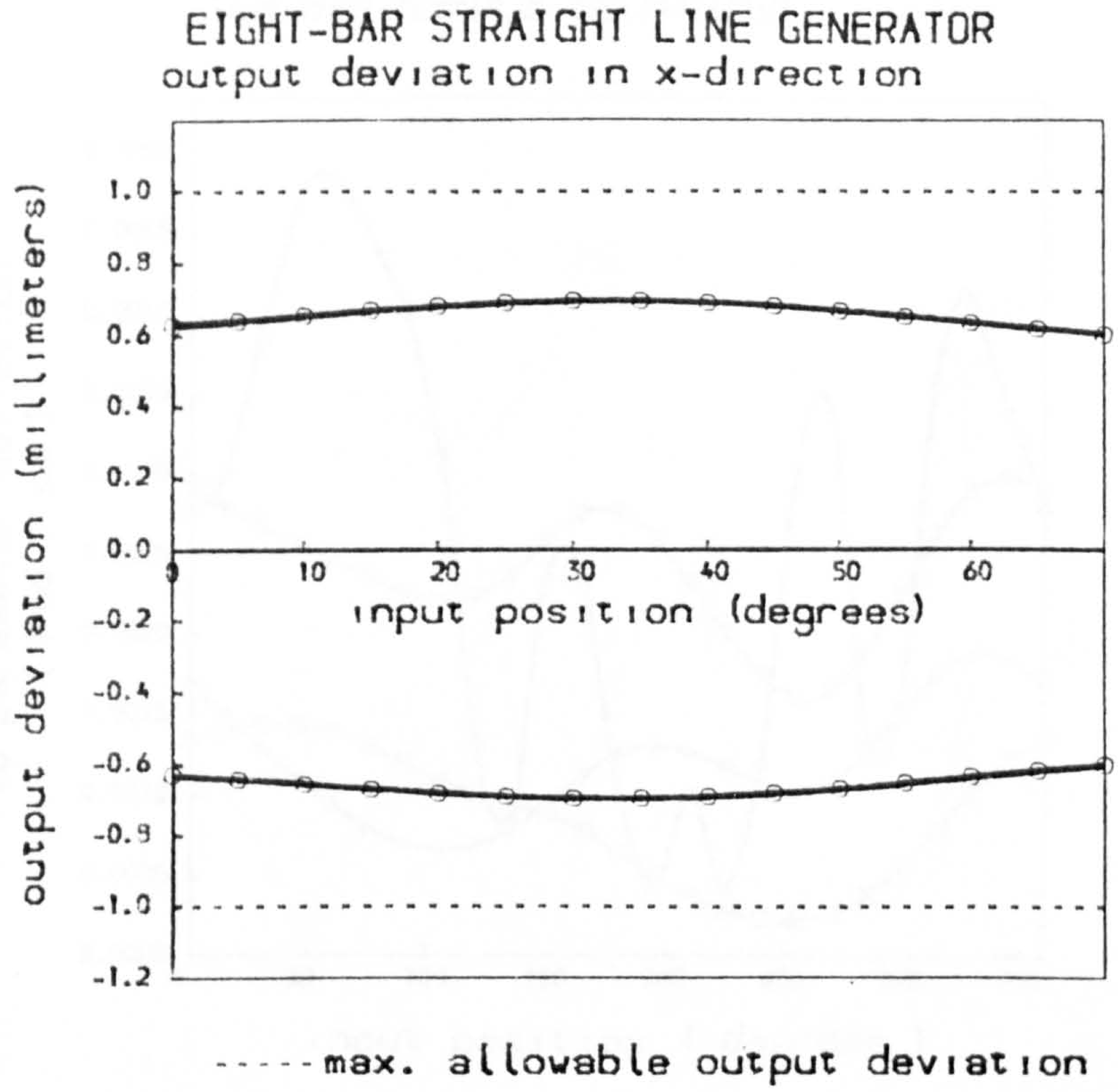


Fig. 6.15(i)

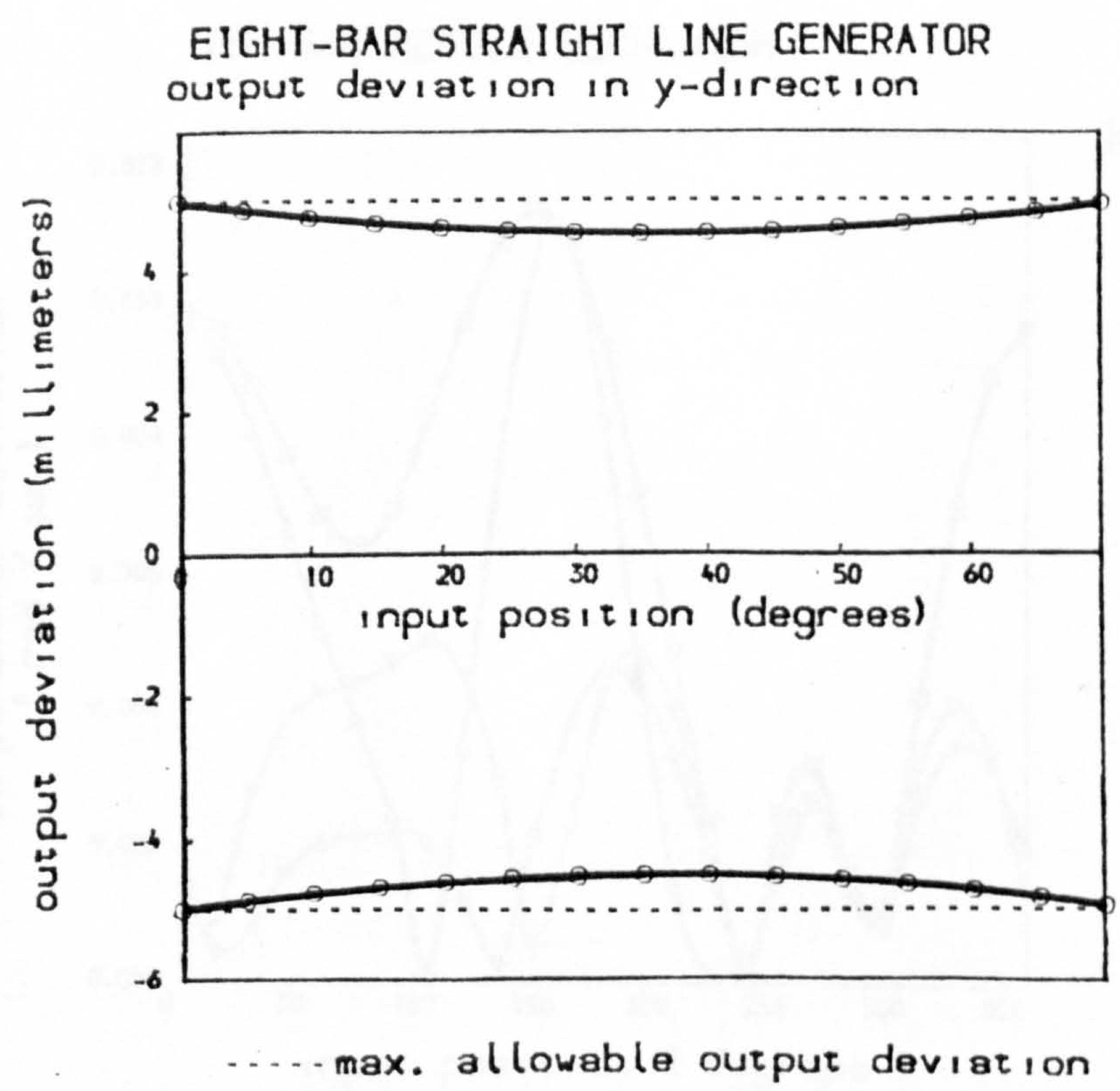


Fig. 6.15(ii)

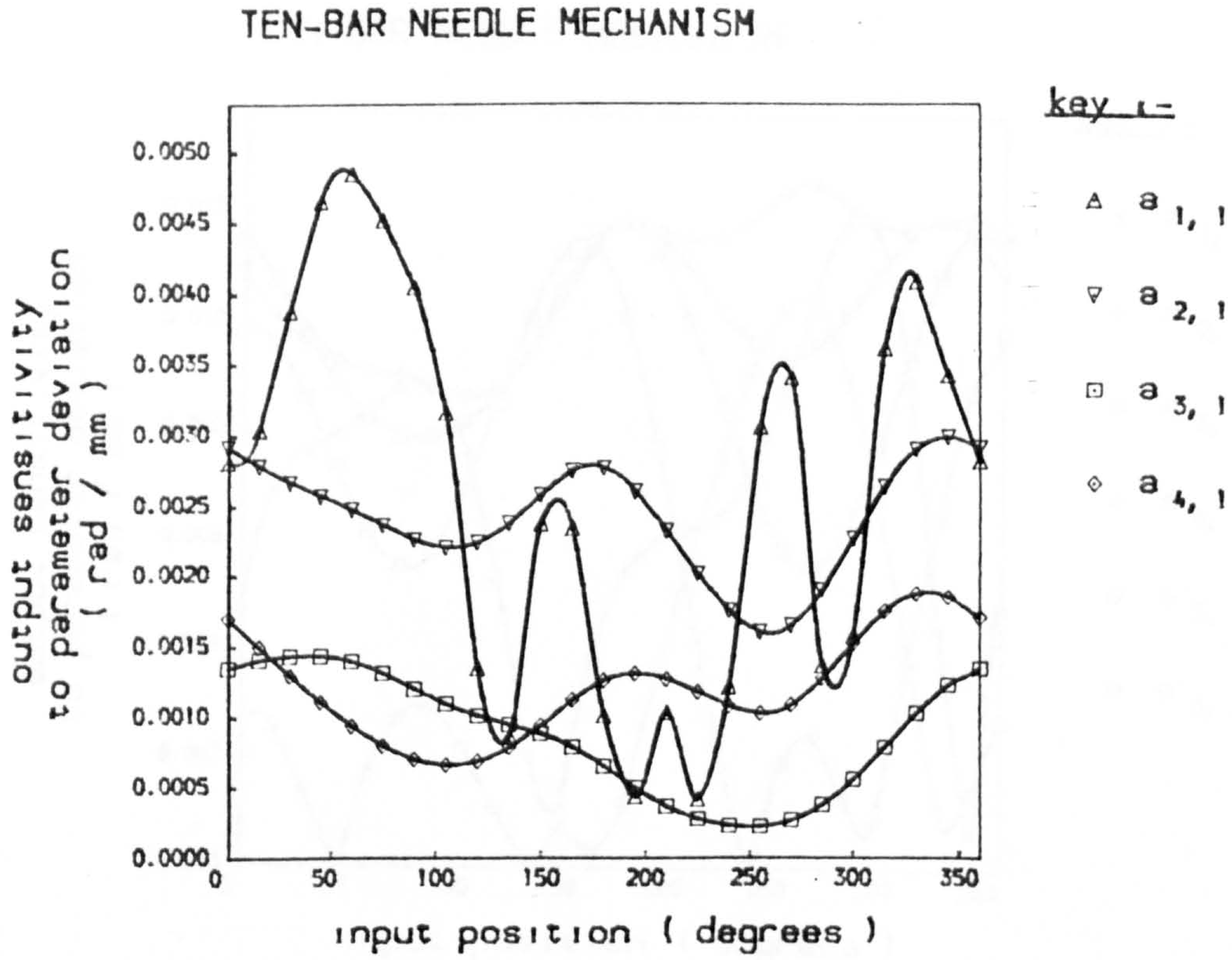


Fig. 6.16a

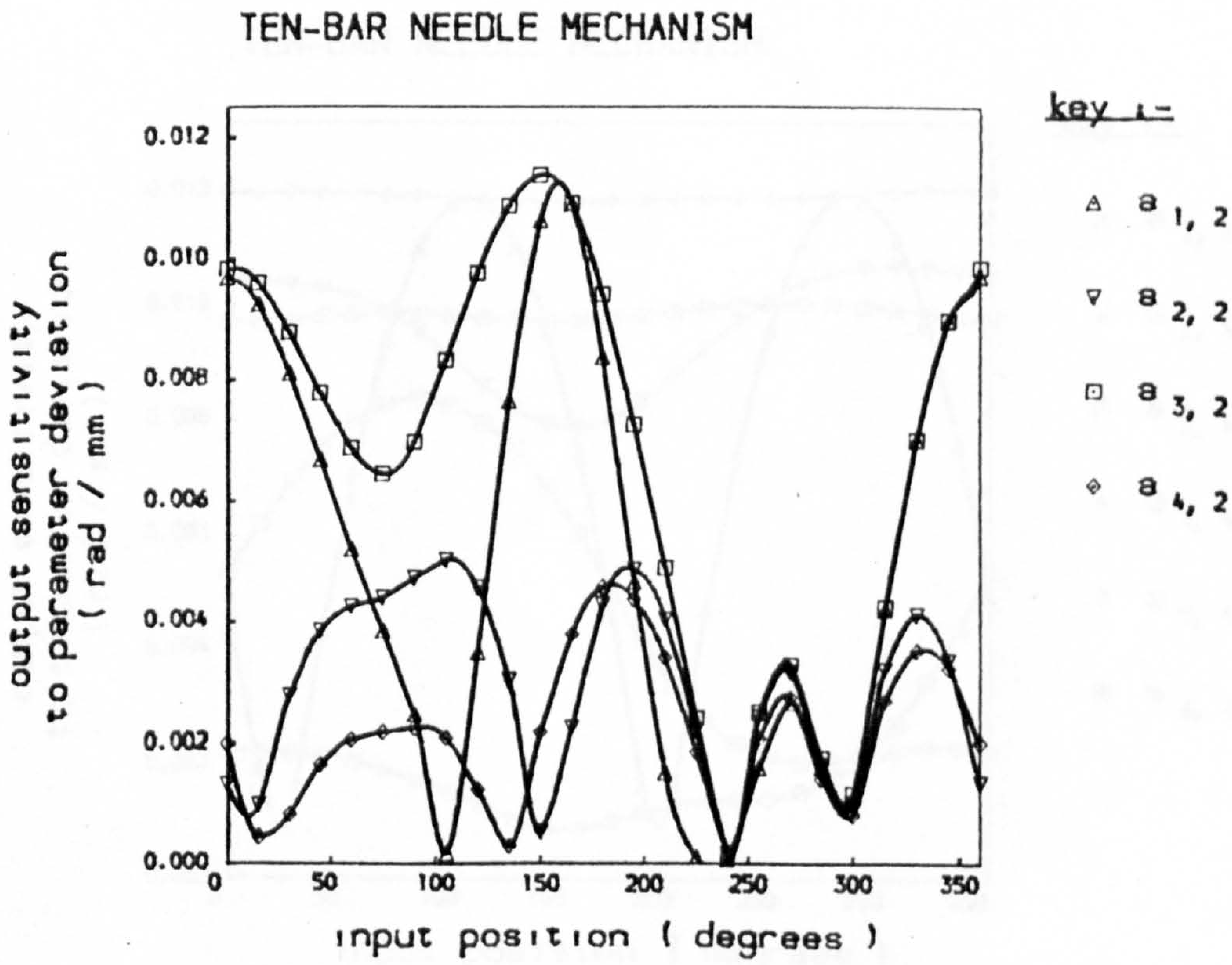


Fig. 6.16b

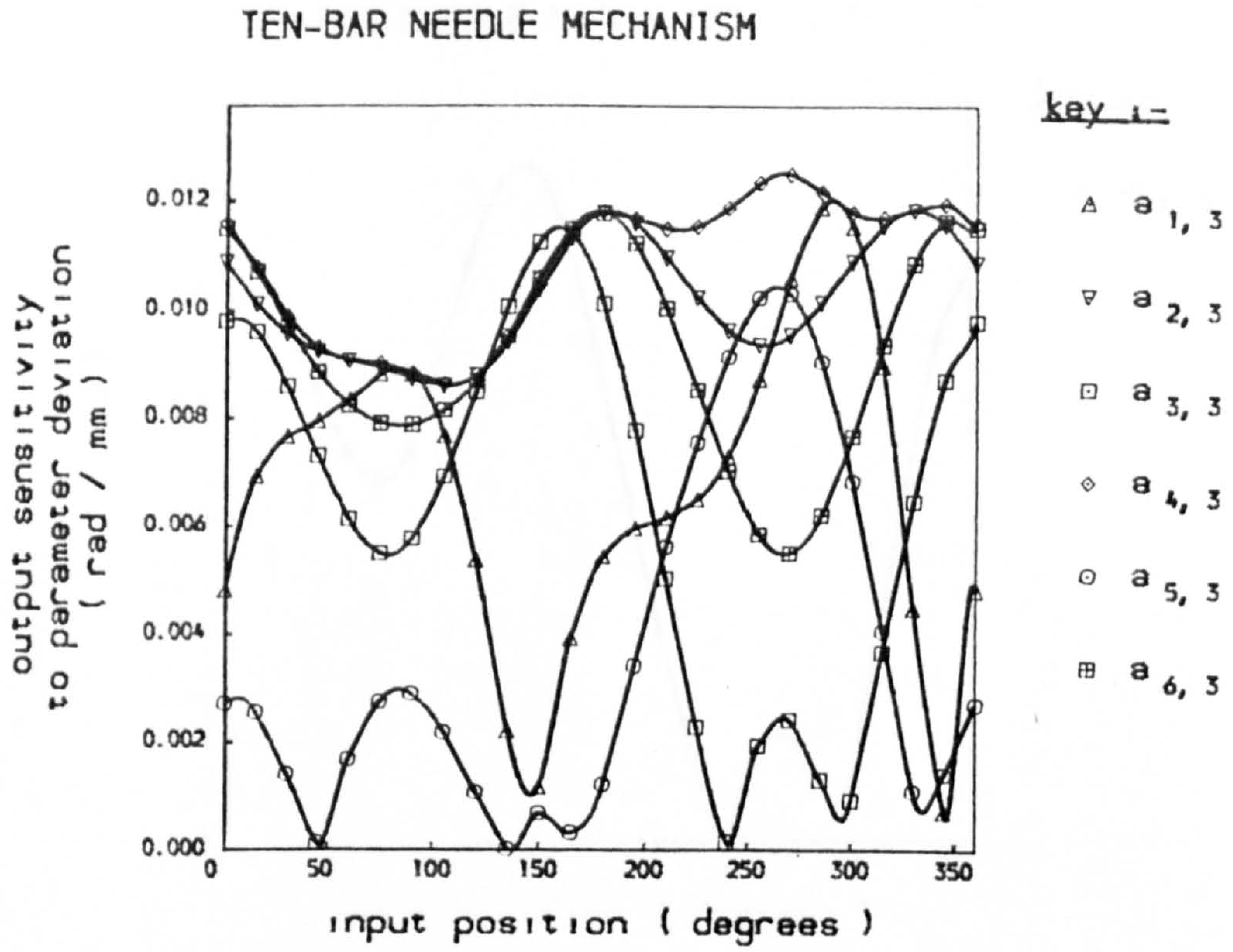


Fig. 6.16c

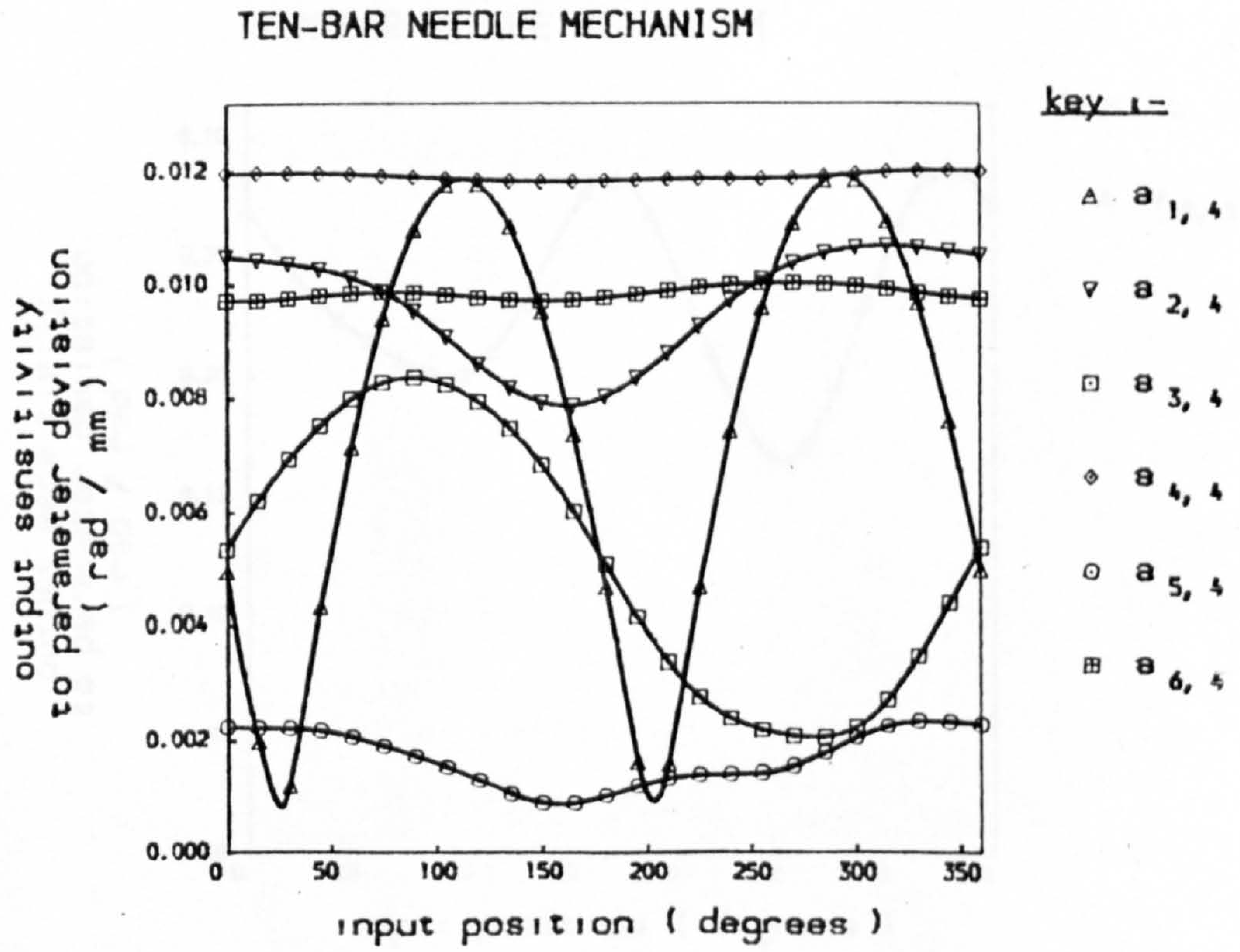


Fig. 6.16d

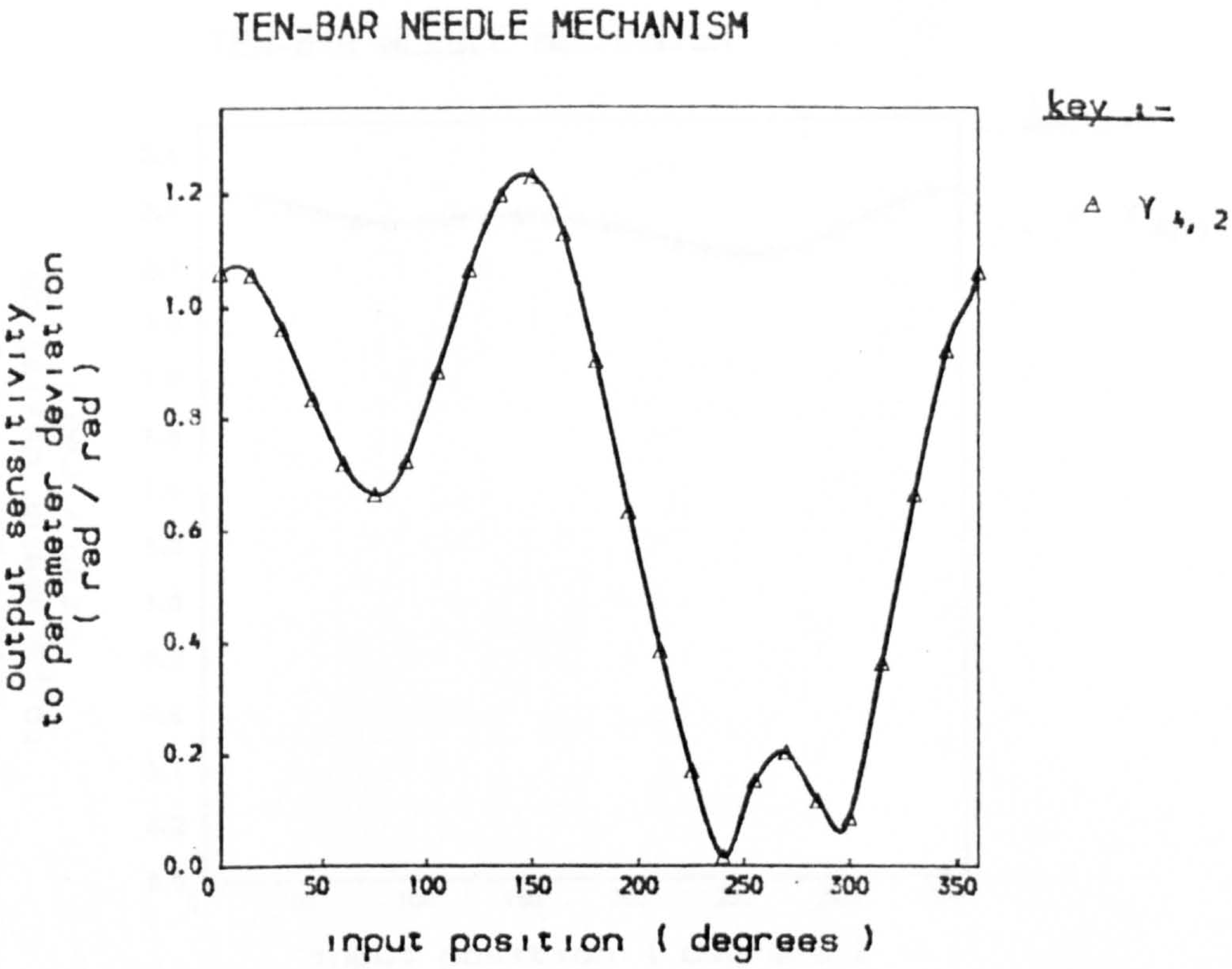


Fig. 6.16e

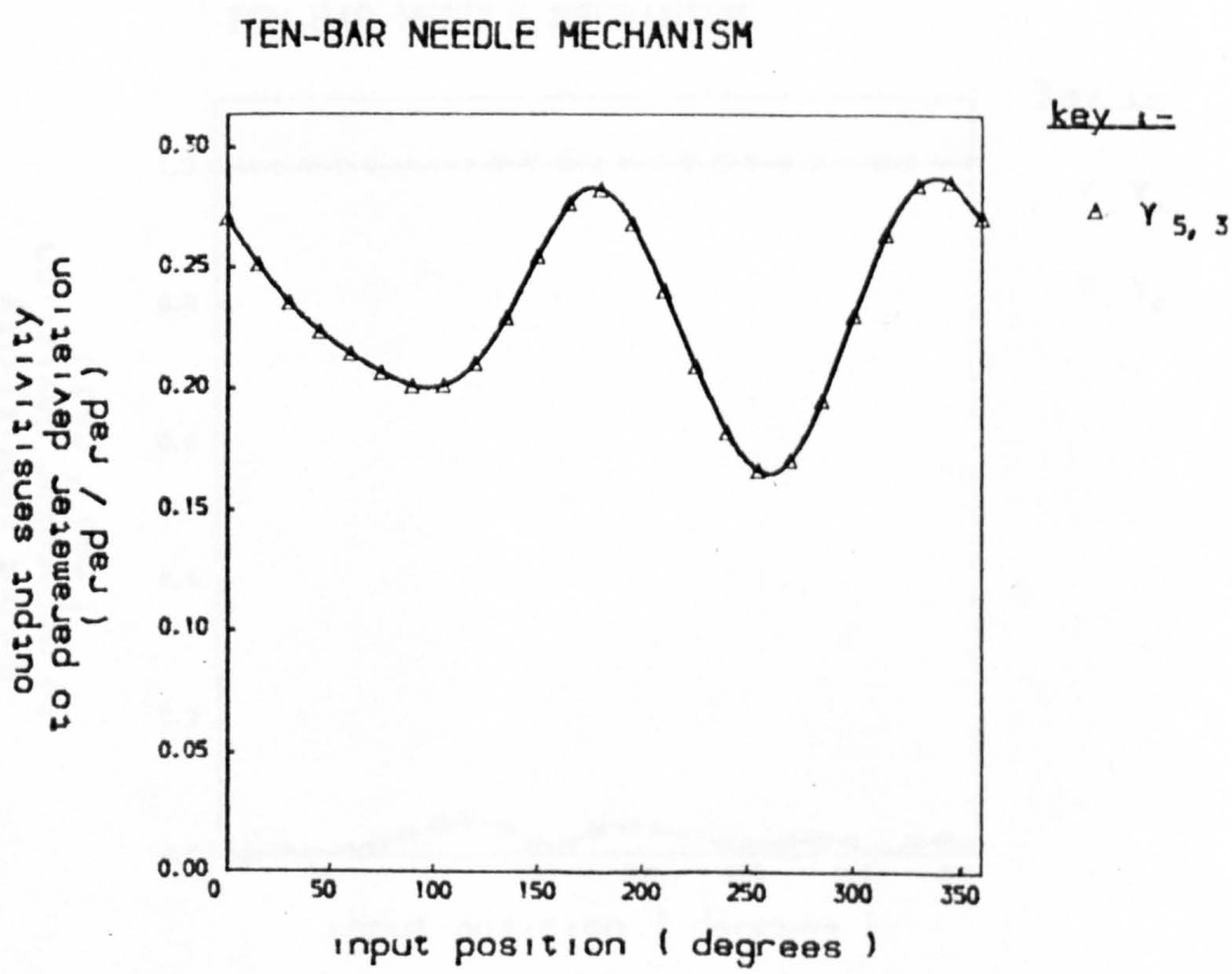


Fig. 6.16f

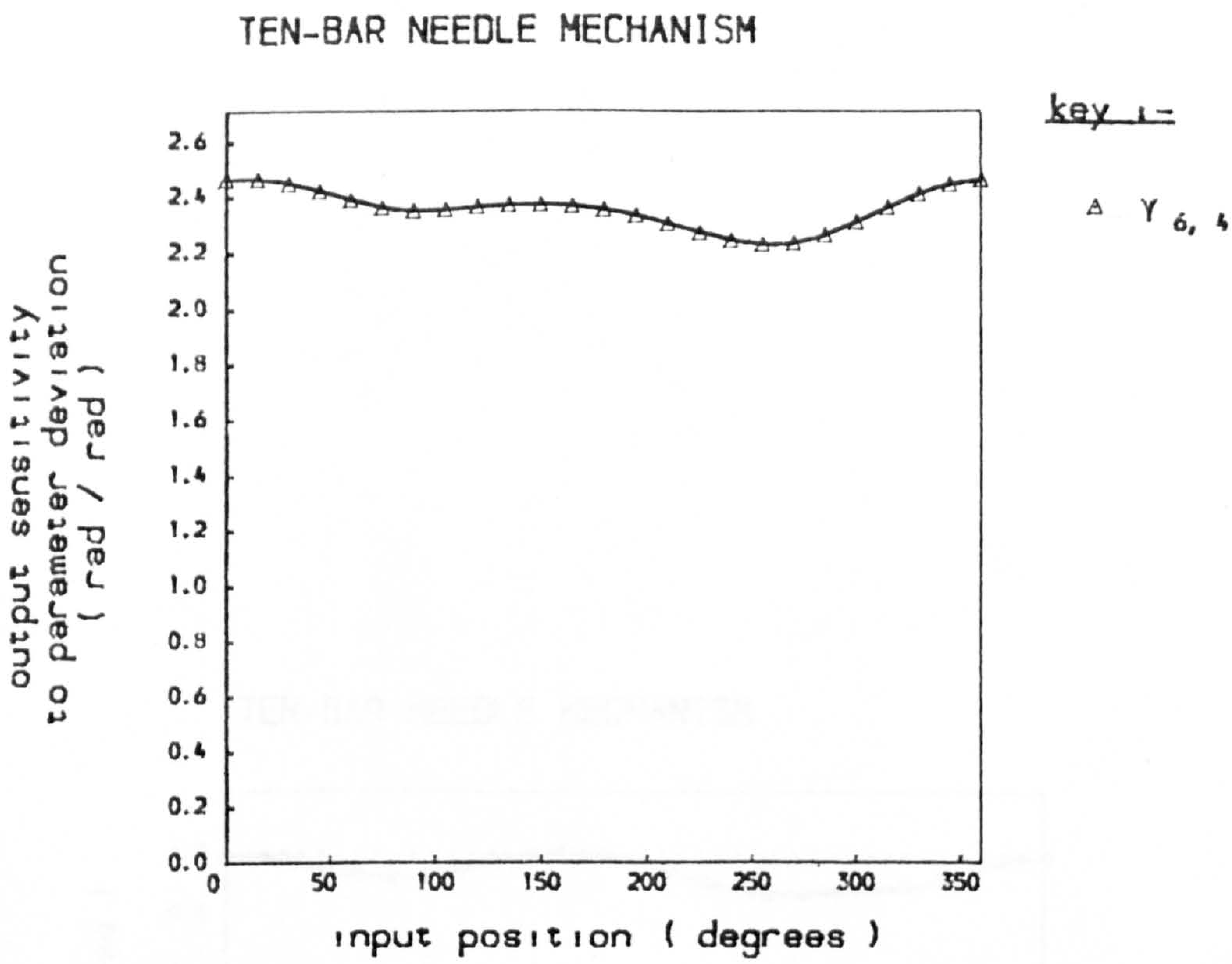


Fig. 6.16g

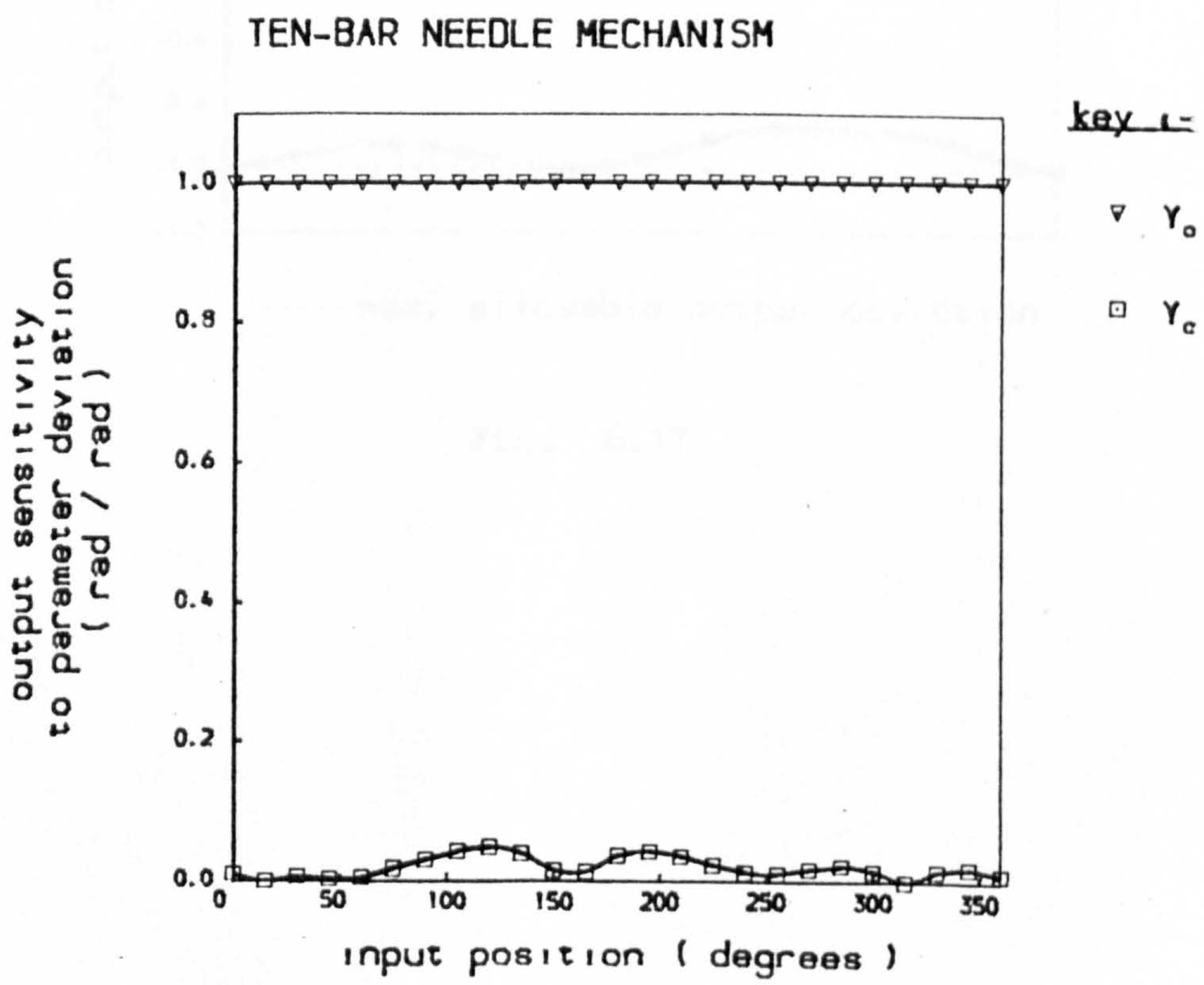


Fig. 6.16h

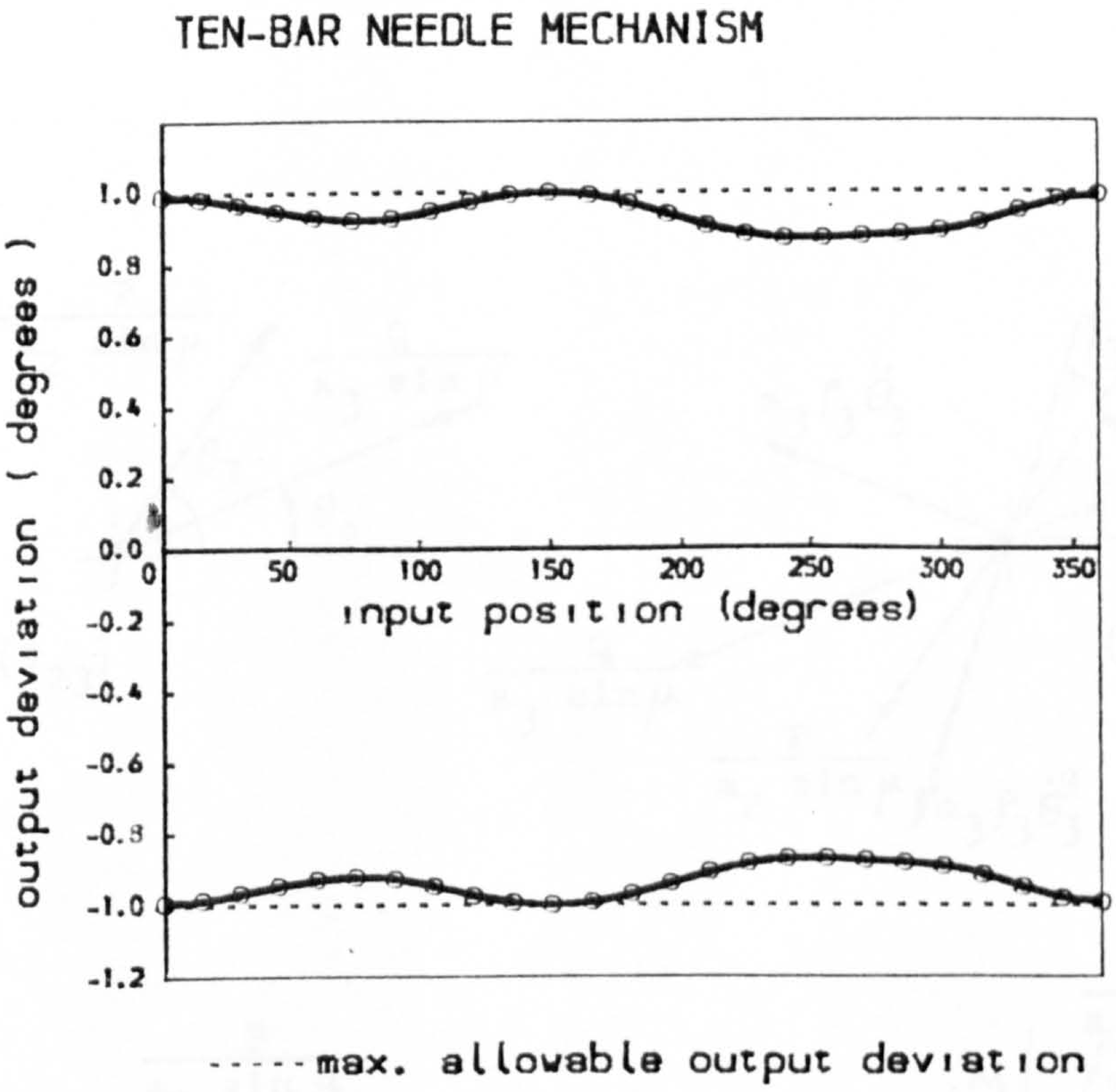

TEN-BAR NEEDLE MECHANISM
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
-1.2
0
50
100
150
200
250
300
350
input position (degrees)
output deviation (degrees)
-----max. allowable output deviation

Fig. 6.17

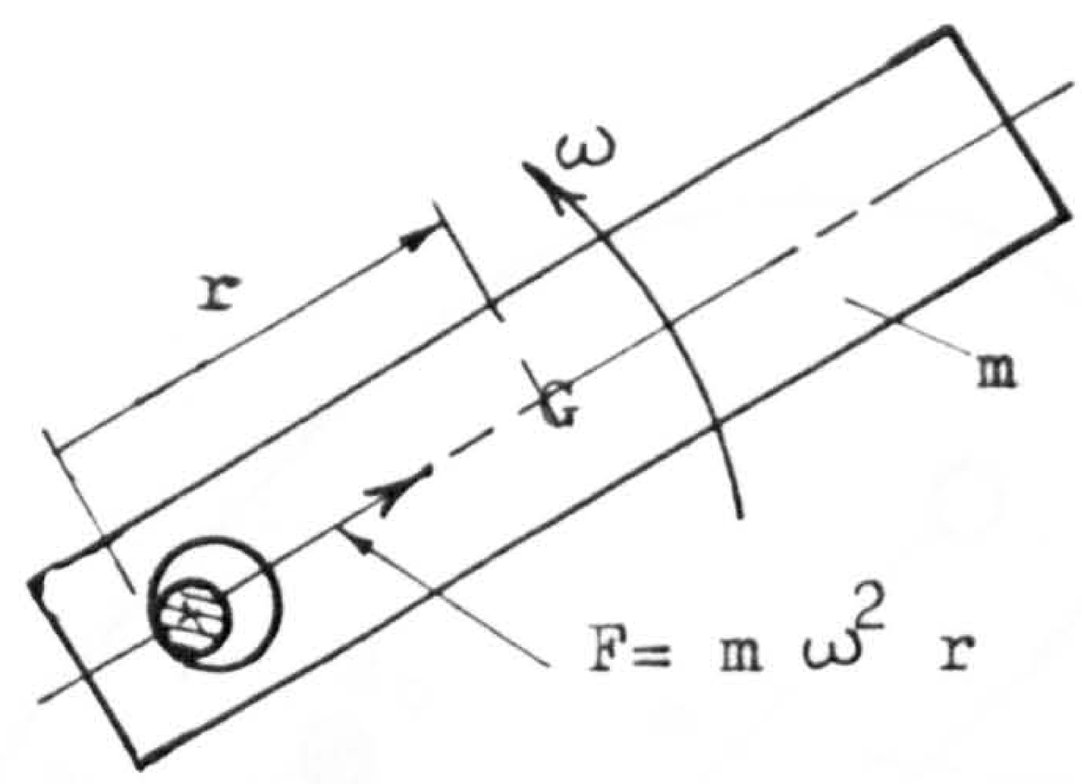


Fig. 8.1 BAR ROTATING IN A HORIZONTAL PLANE

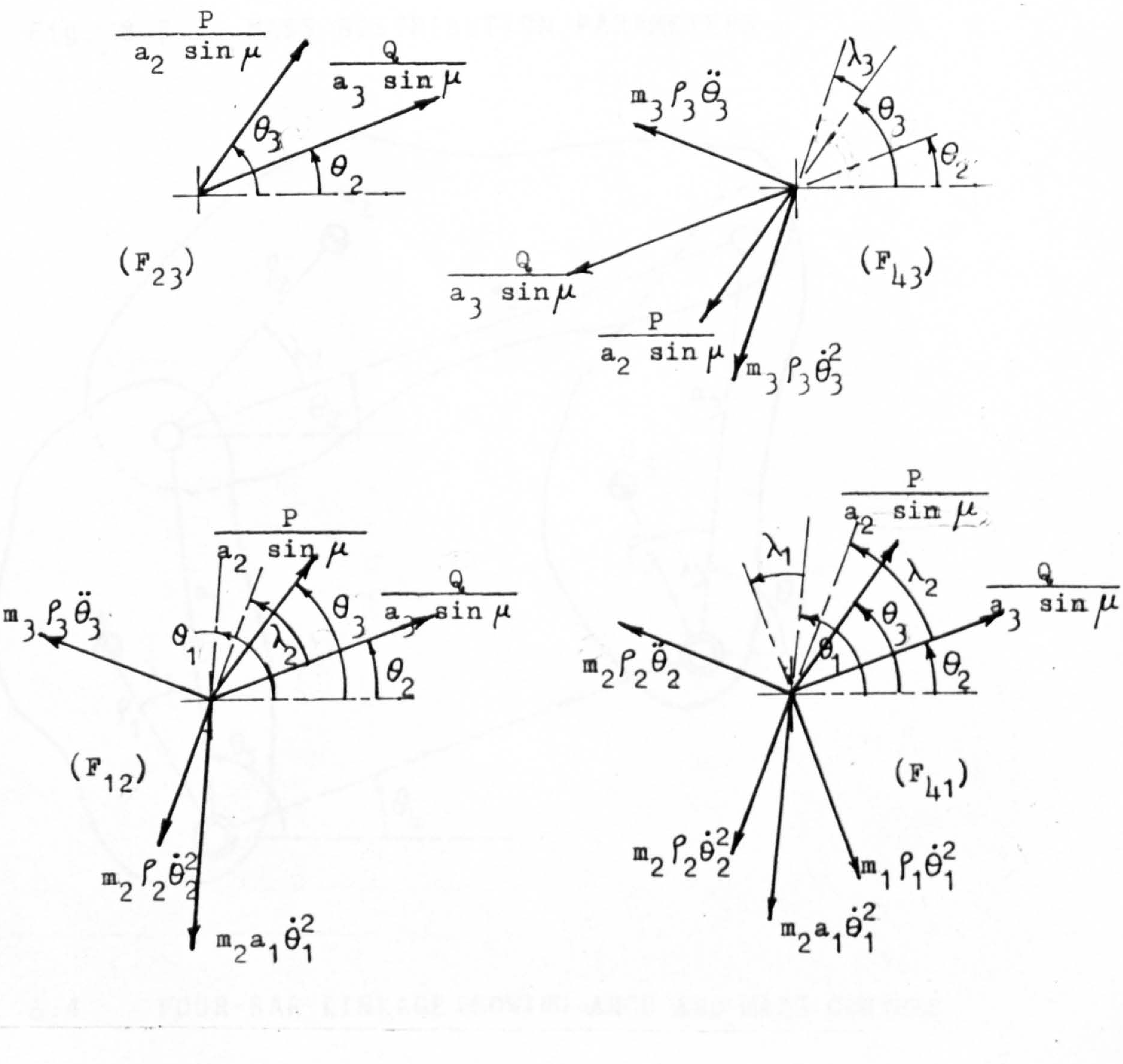


Fig. 8.2 COMPONENT VECTORS OF JOINT FORCES

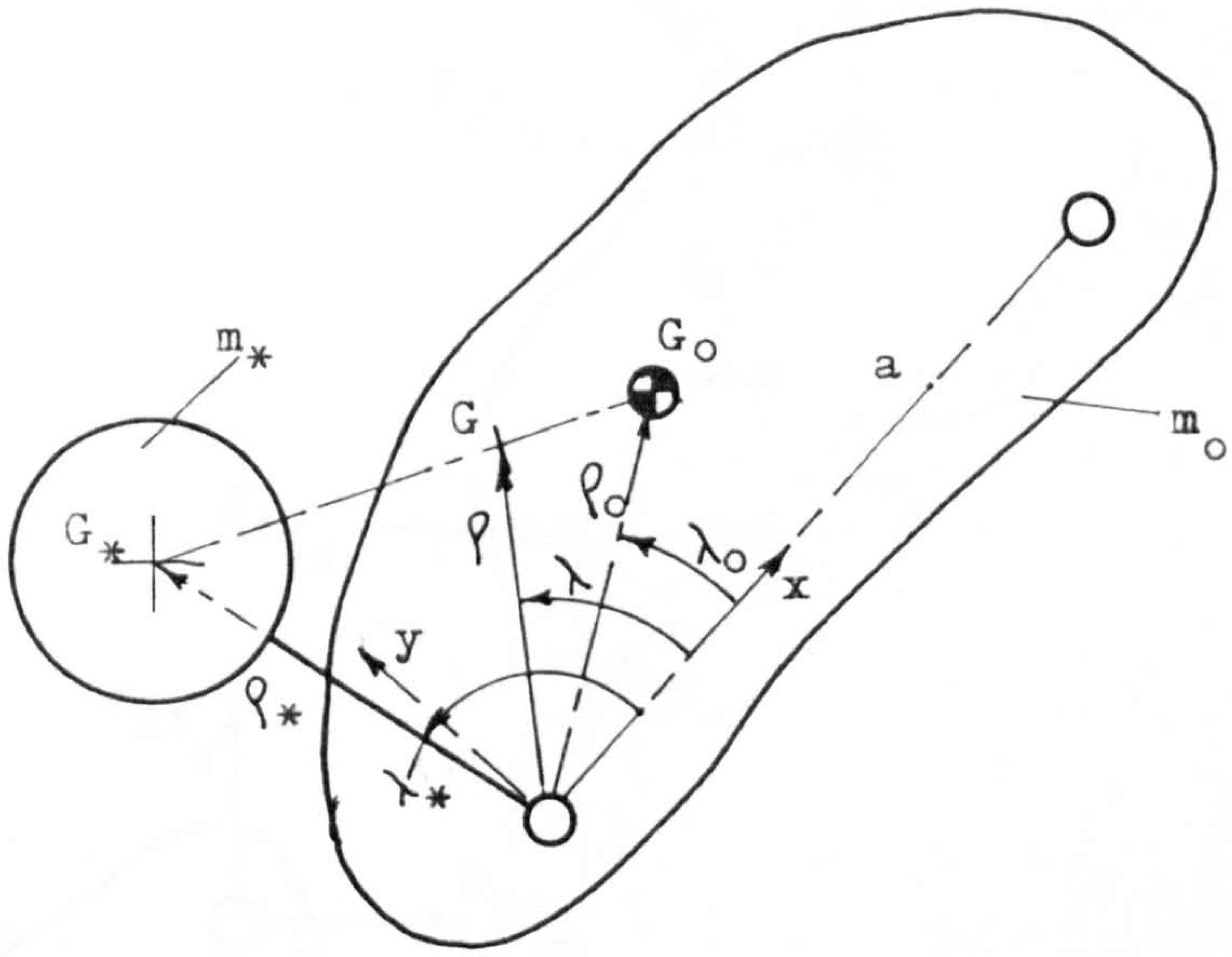


Fig. 8.3 MASS DISTRIBUTION PARAMETERS

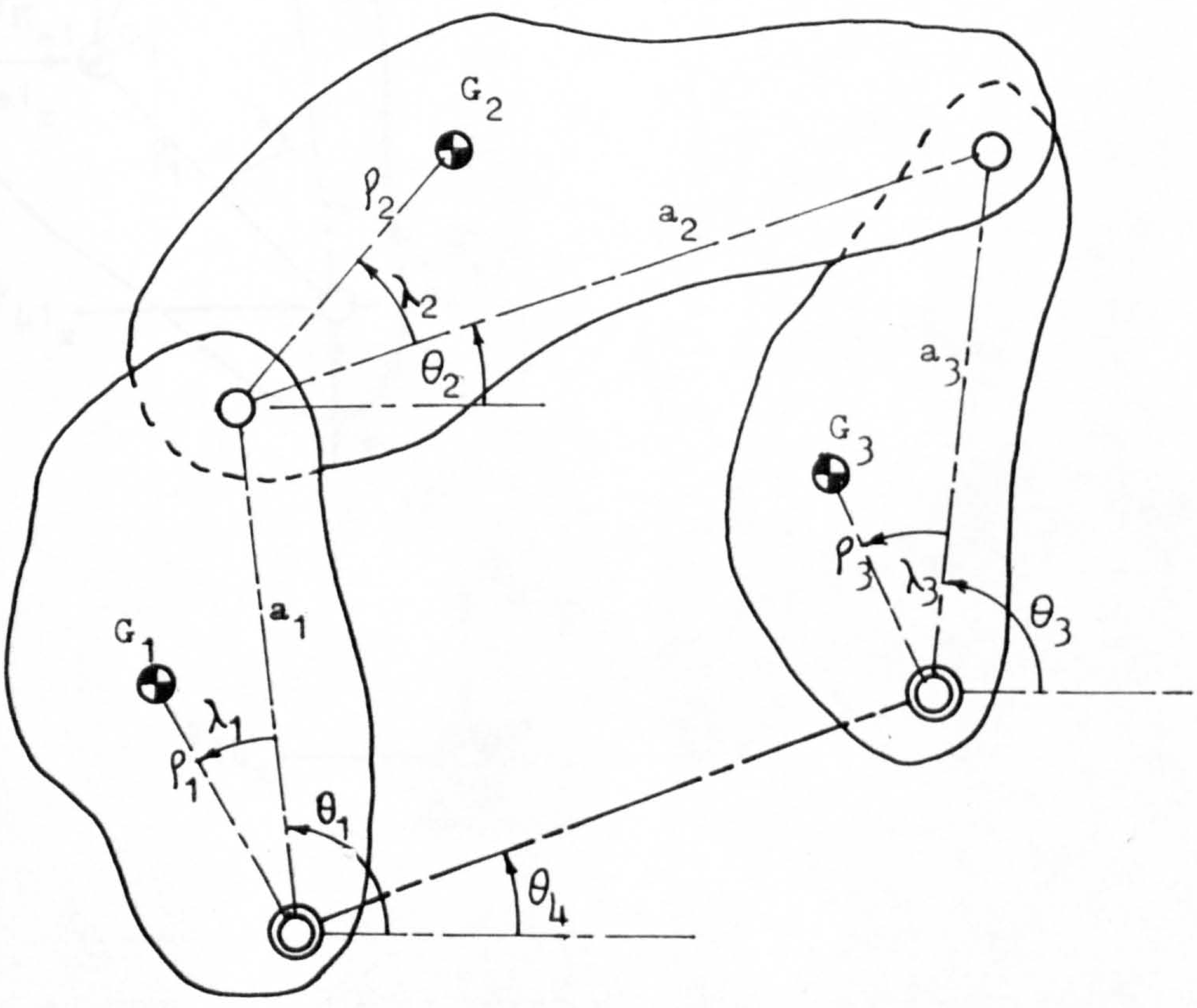


Fig. 8.4 FOUR-BAR LINKAGE SHOWING ARCS AND MASS CENTRES

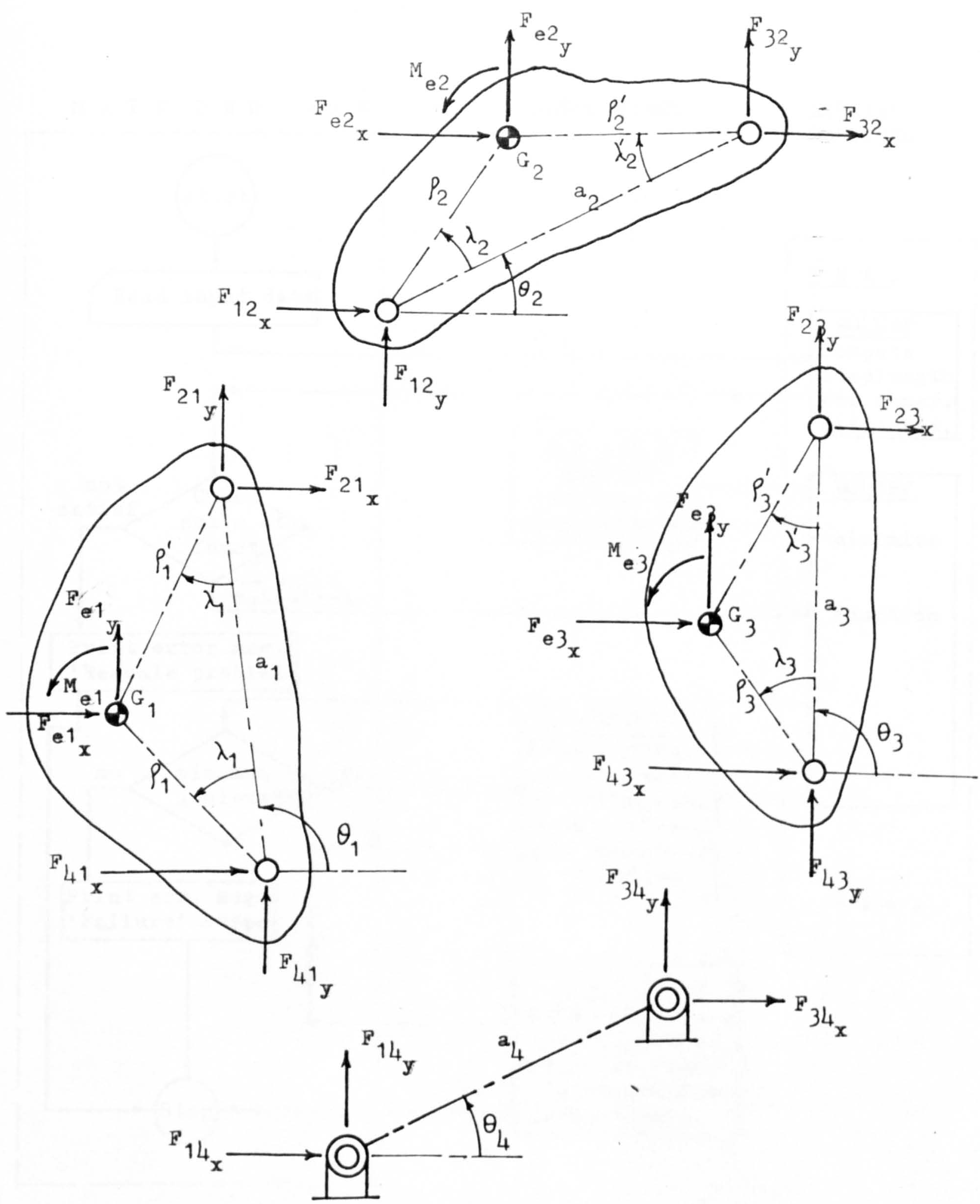


Fig. 8.5 FREE-BODY DIAGRAMS OF FOUR-BAR LINKAGE MEMBERS

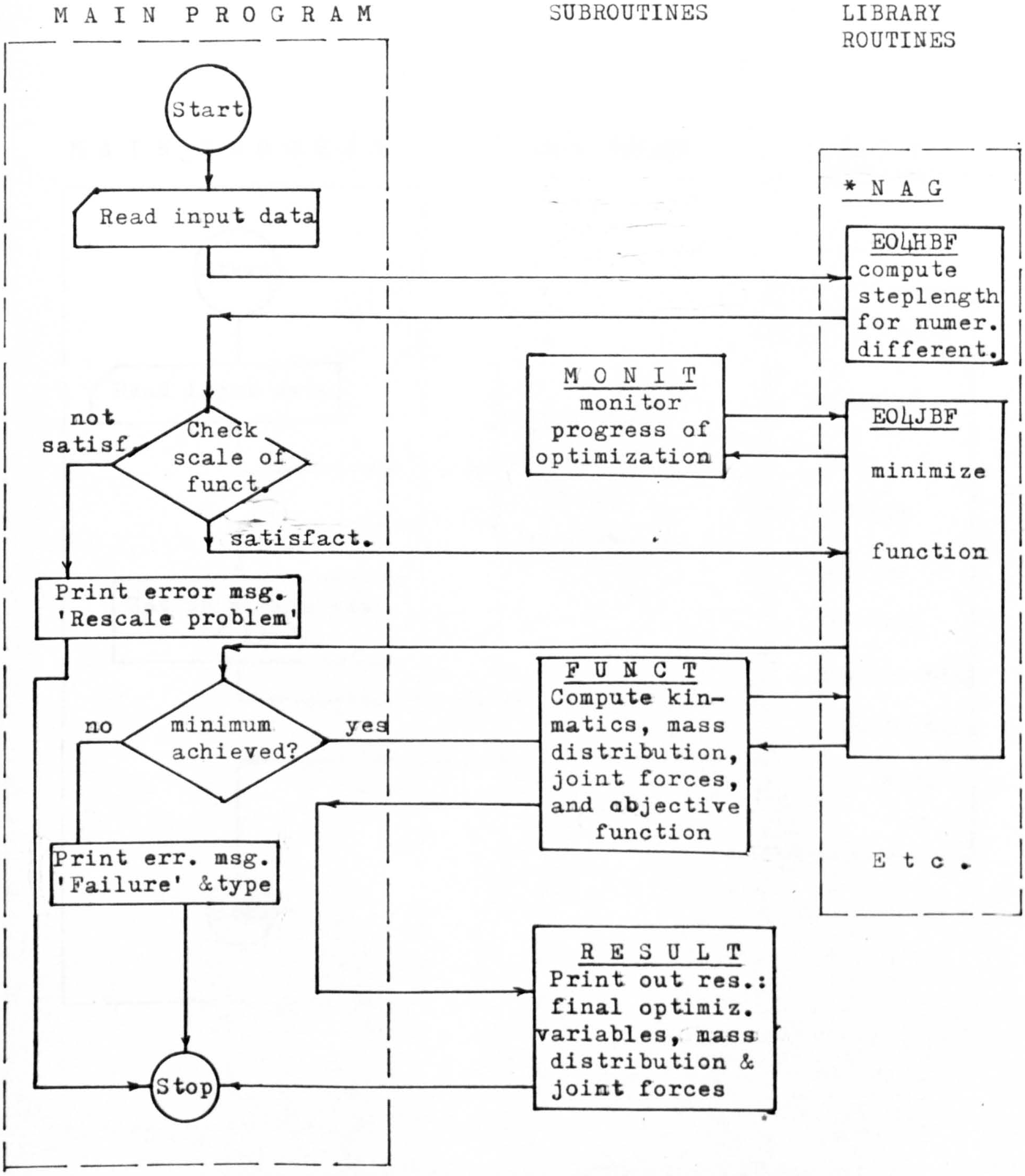


Fig. 9.1 FLOW CHART OF PROGRAM <u>CONTACT</u>

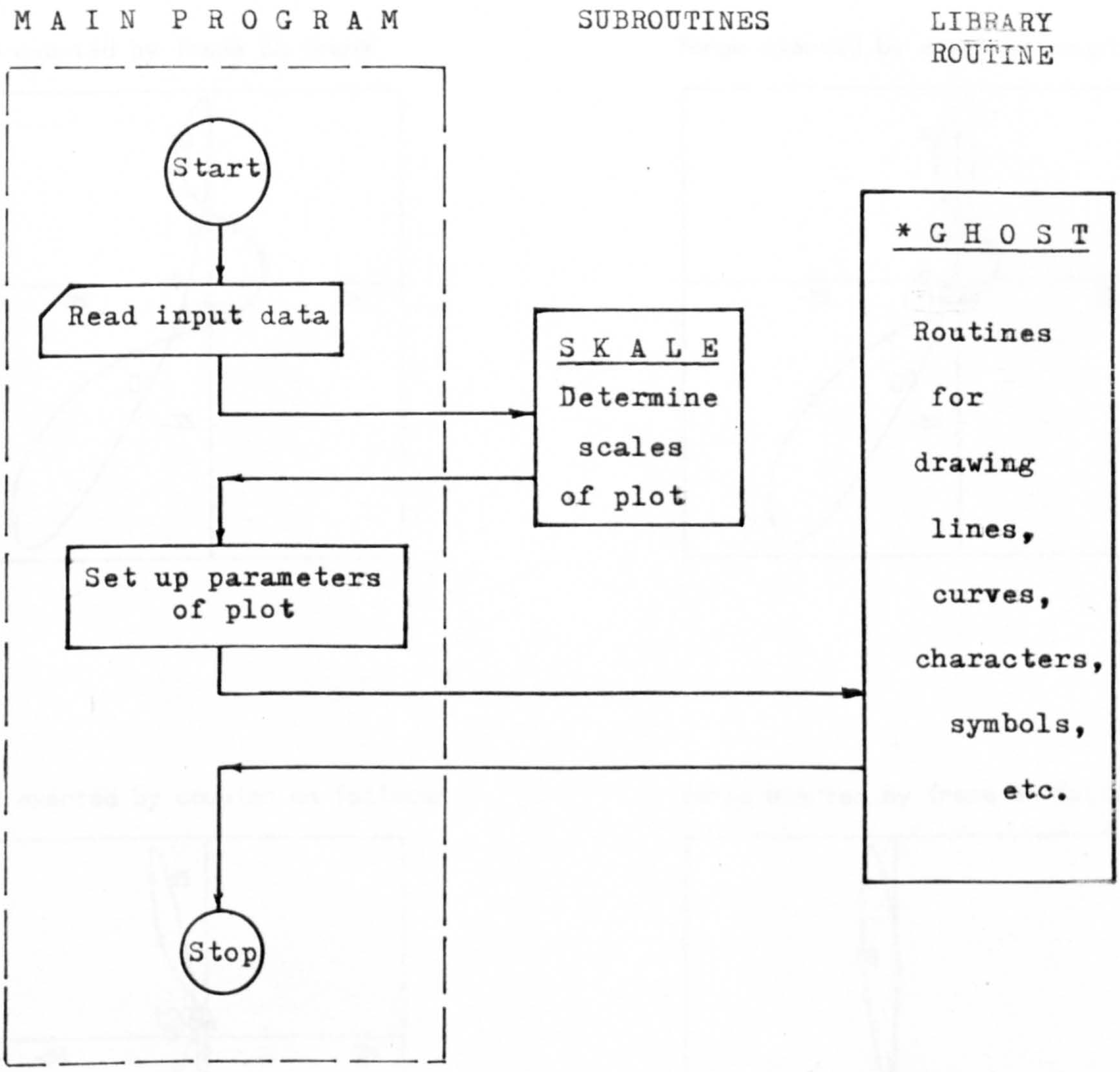


Fig. 9.2 FLOW CHART OF PROGRAM PINFORCE

CRANK-ROCKER WITH ARC LENGTHS 50,100,100,100

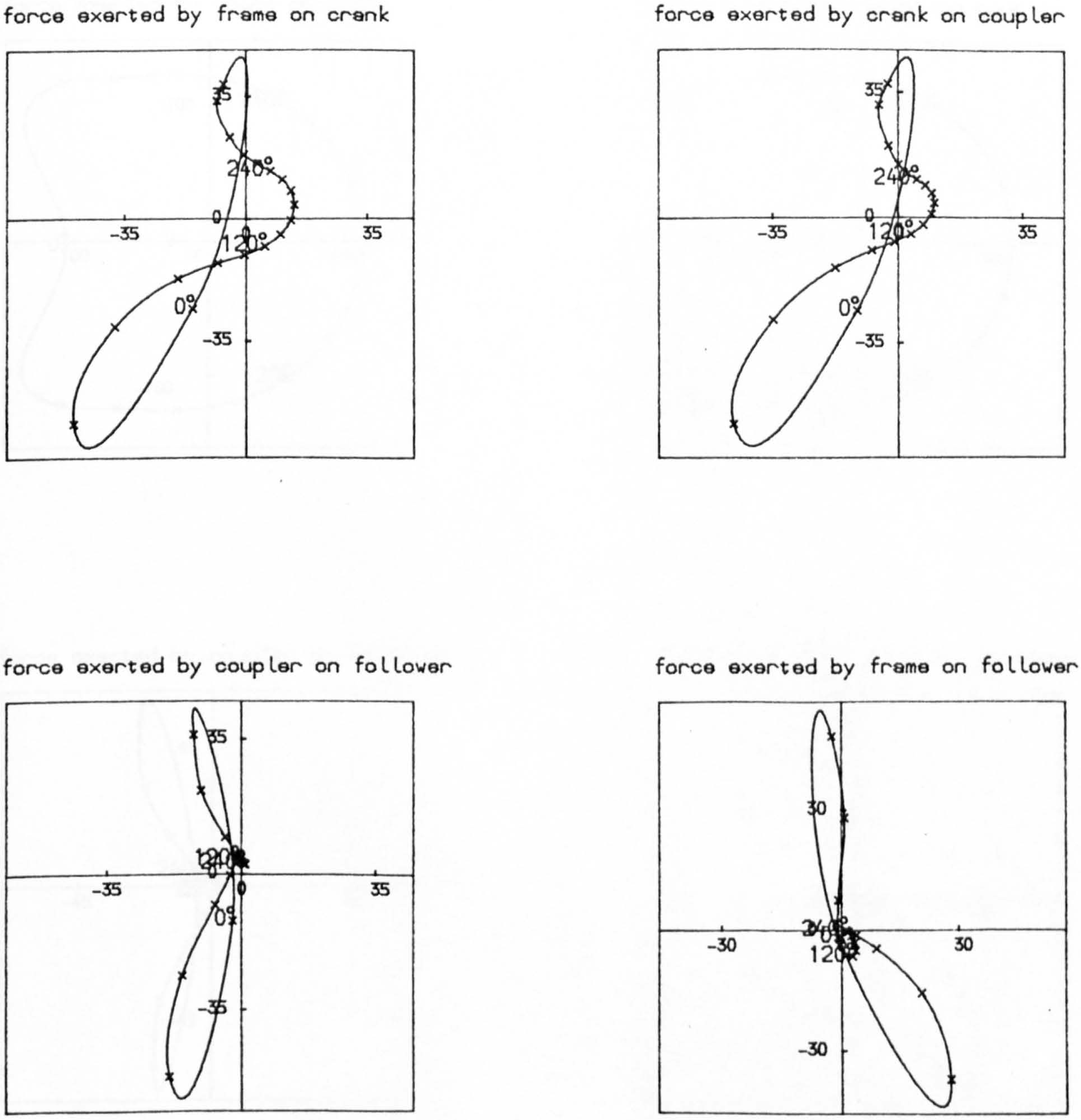


Fig. 10.1 JOINT FORCE PLOTS PRE-OPTIMIZATION FOR CRANK-ROCKER WITH POOR TRANSMISSION ANGLE

CRANK-ROCKER WITH ARC LENGTHS 50,100,100,100

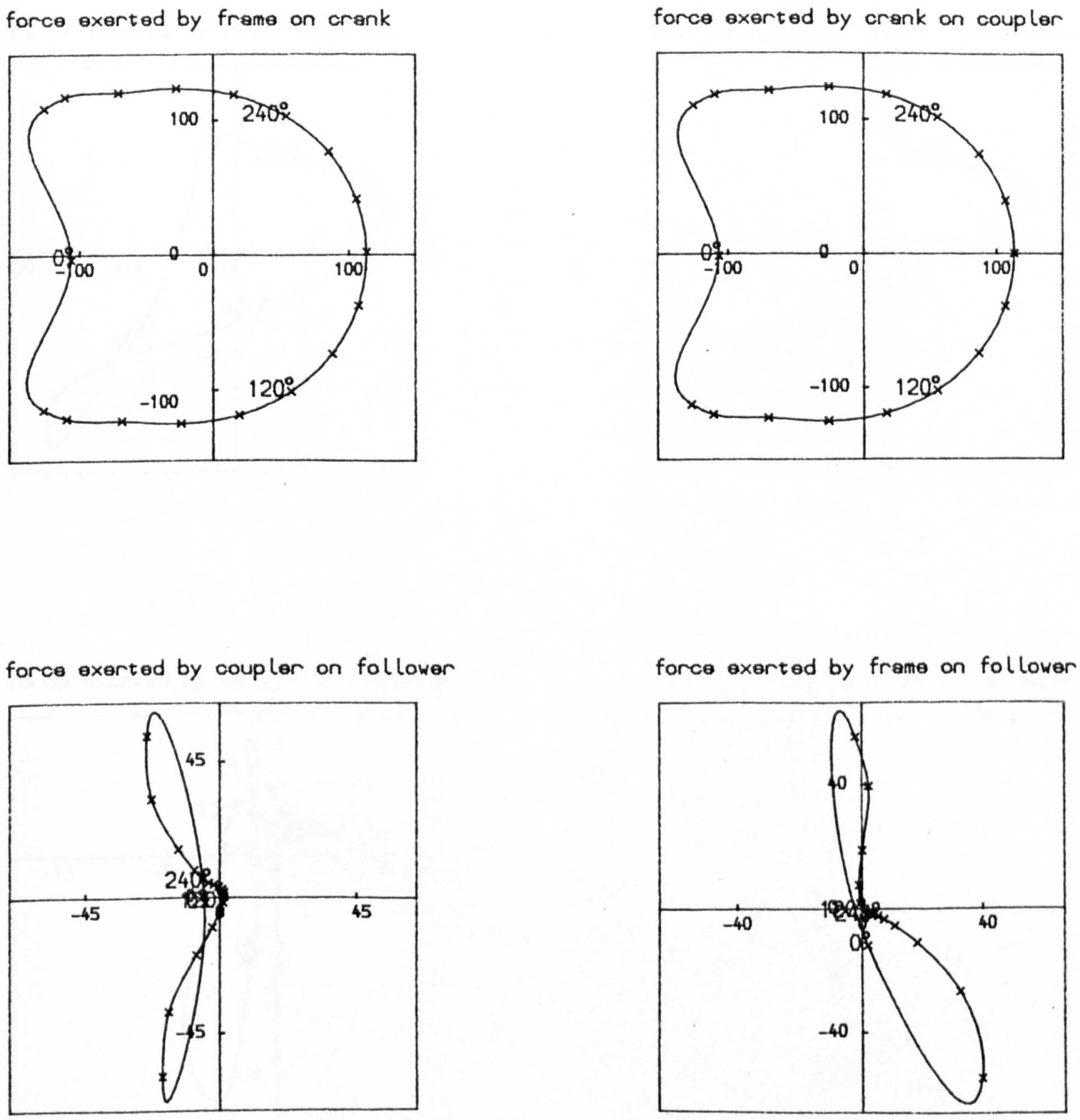


Fig. 10.2 JOINT FORCE PLOTS POST-OPTIMIZATION FOR CRANK-ROCKER WITH POOR TRANSMISSION ANGLE

CRANK-ROCKER WITH ARC LENGTHS 50,100,100,132.5

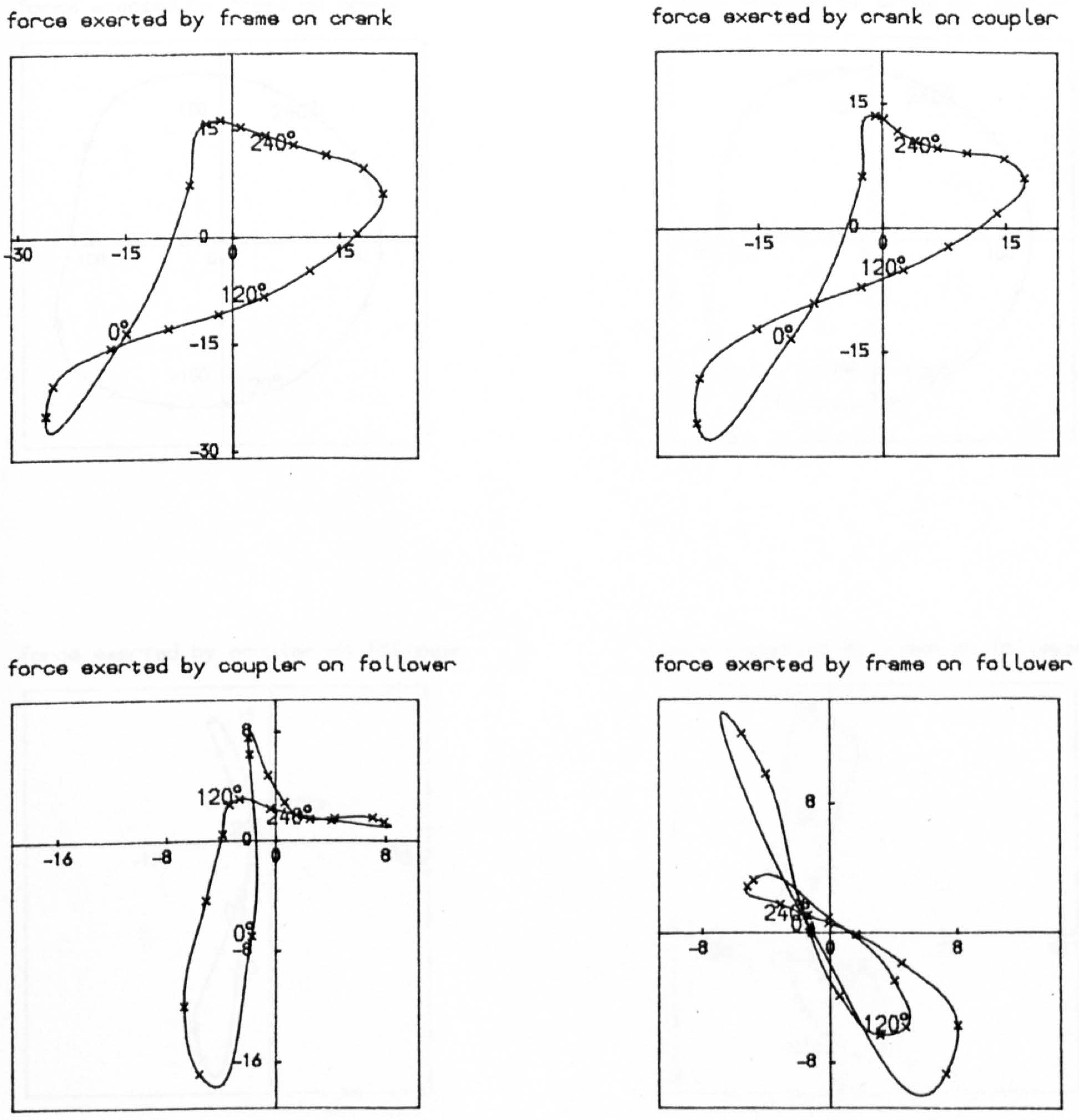


Fig. 10.3 JOINT FORCE PLOTS PRE-OPTIMIZATION FOR CRANK-ROCKER WITH IMPROVED TRANSMISSION ANGLE

CRANK-ROCKER WITH ARC LENGTHS 50,100,100,132.5

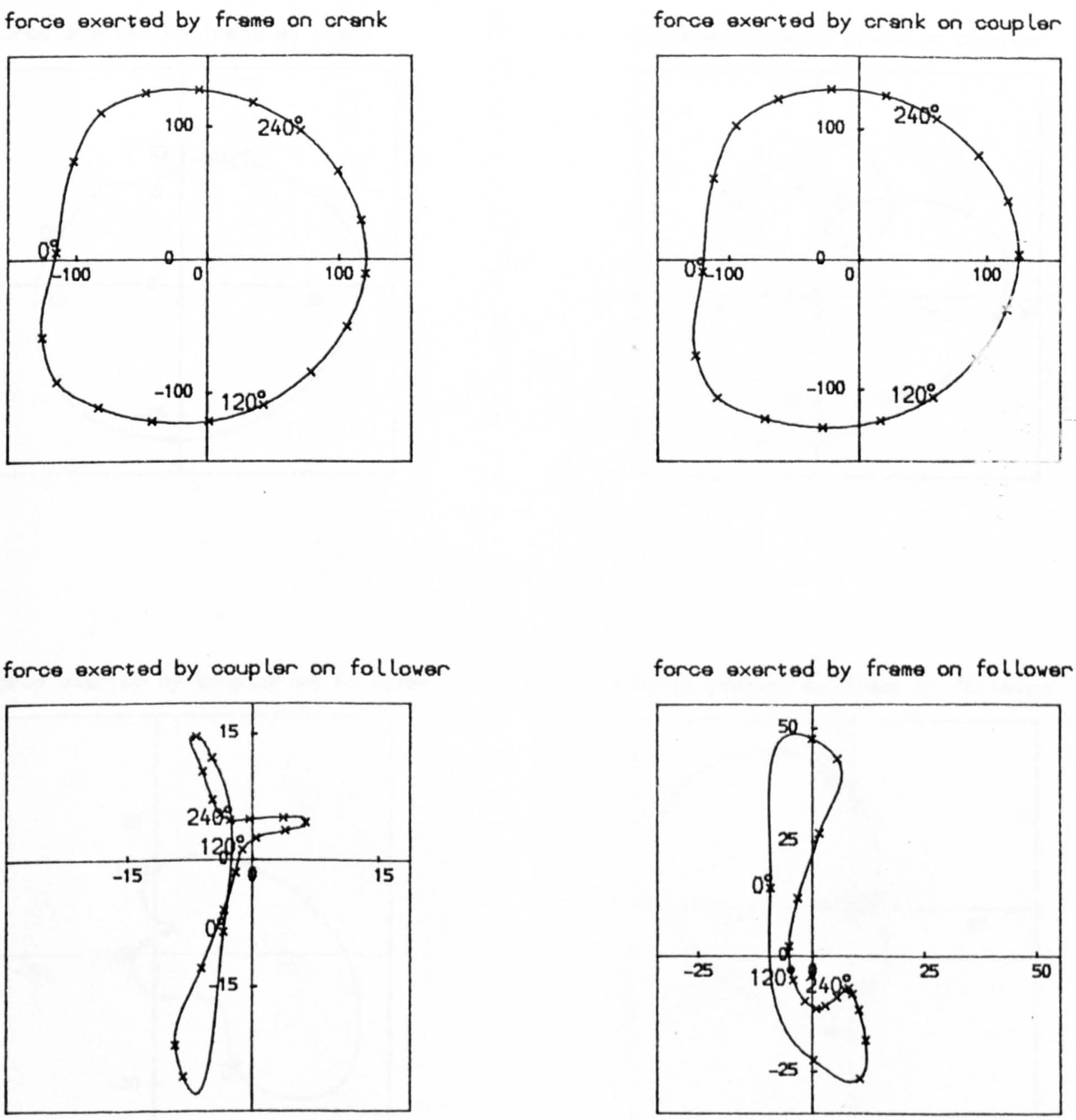


Fig. 10.4 JOINT FORCE PLOTS POST-OPTIMIZATION FOR CRANK-ROCKER WITH IMPROVED TRANSMISSION ANGLE

DRAG-LINK WITH ARC LENGTHS 132.5, 100, 100, 50

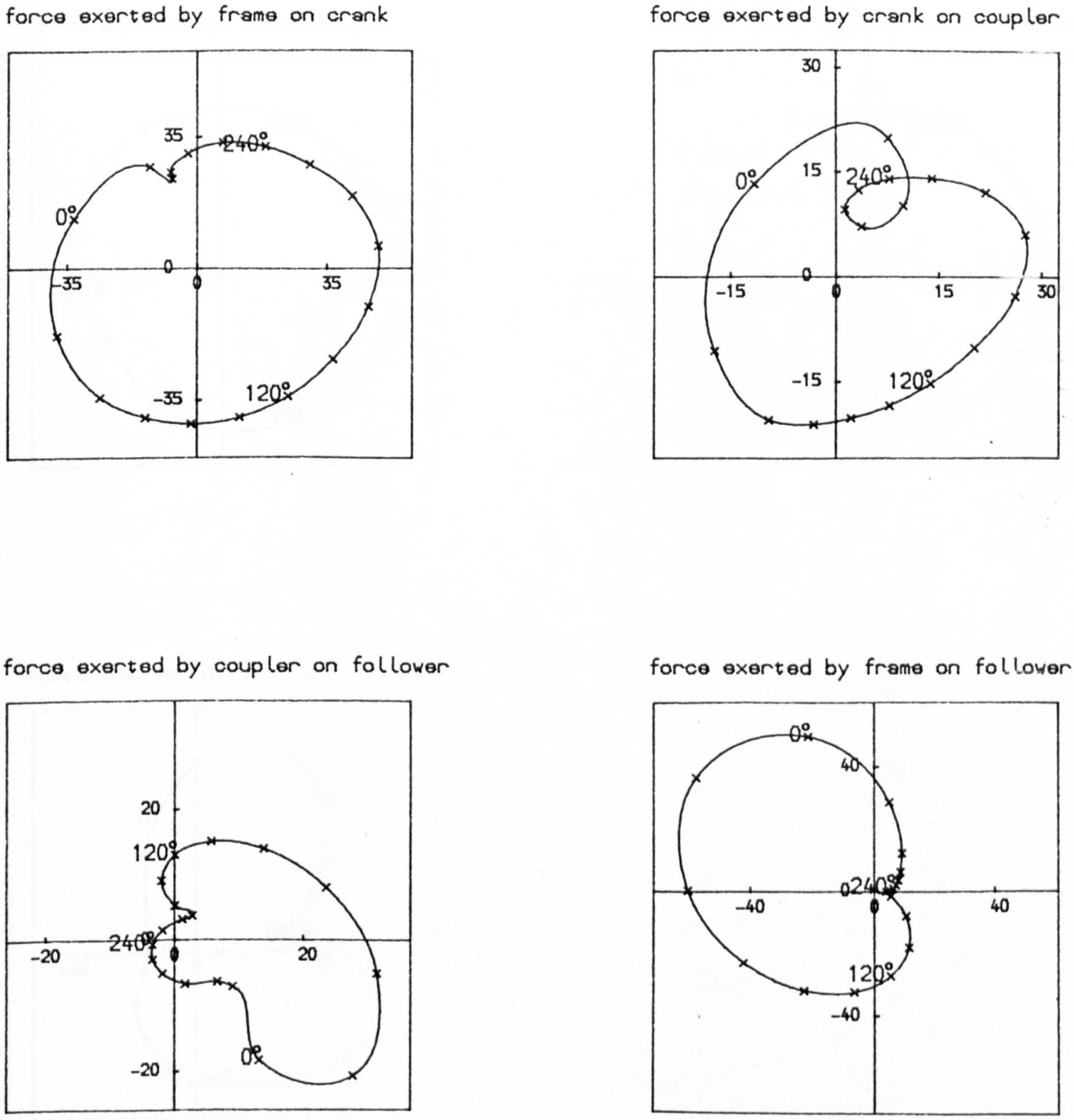


Fig. 10.5 JOINT FORCE PLOTS PRE-OPTIMIZATION FOR DRAG-LINK

DRAG-LINK WITH ARC LENGTHS 132.5, 100, 100, 50

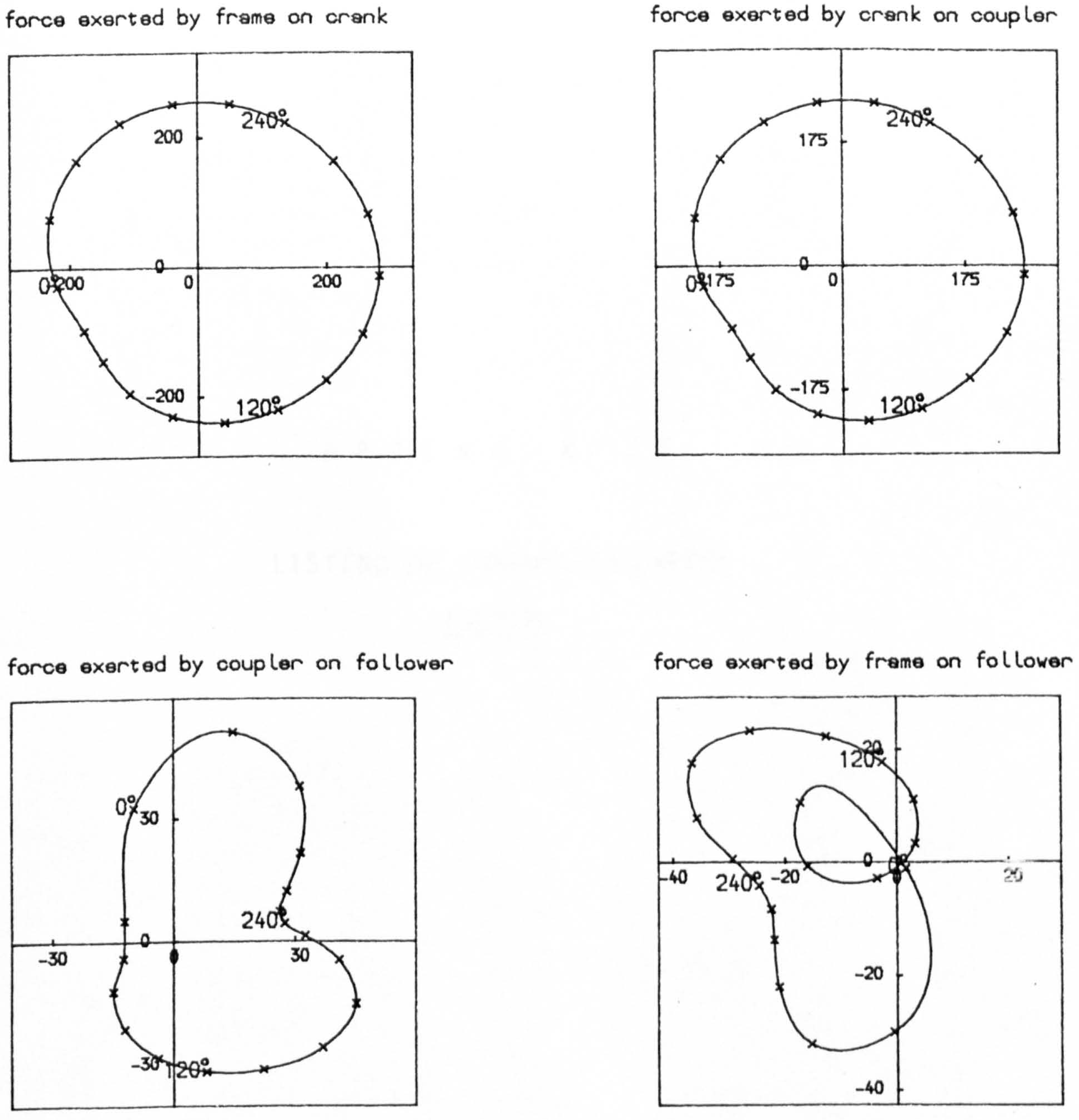


Fig. 10.6 JOINT FORCE PLOTS POST-OPTIMIZATION FOR DRAG-LINK

APPENDIX IA

LISTING OF COMPUTER PROGRAM

TOCALM

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   *  OPTIMAL ALLOCATION OF TOLERANCES AND CLEARANCES  *
C   *            IN PLANE LINKAGE MECHANISMS             *
C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      IMPLICIT REAL*8(A - H,O - Z)
C
      DIMENSION ILHD(10), IGHD(10), DESX(37), DESY(37), DESANG(37),
     1          ANMULT(37), XMULT(37), YMULT(37), OPLOW(30), OPHIGH(30),
     2          IVA(100), IVG(100), MRNUR(100), MGNUR(100), XO(37),
     3          YO(37), RO(37), TO(37), AO(37), SO(37), XM(37), YM(37),
     4          RM(37), TM(37), AM(37), SM(37), LORDER(105), PDAP(100),
     5          PGAP(100), COSTA(105), COSTG(105), TOLER(105),
     6          TOLEG(105), TOLED(105), DIFA(105), DIFG(105), IREP(100),
     7          ITERN(100)
      COMMON ARCL(100), AARCL(100), GAMA(100), GGAMA(100), DEGINC,
     1       IKON(361), ILINK(100), ILOOP(100), ITITLE(18), IPLOT, KKL,
     2       KKM, KKN, KKO, MAXARC, NLOOP, NFORM, NPOINT, MOTOR, IANG,
     3       MOTION, NDES, KON, IILINK(100), IILOOP(100), KLINK(100)
      COMMON /CORR/ OUTOLX, OUTOLY, OUTOLA, OUTOLR, OUTOLT, OUTOLS,
     1       PDX(37,100), PDY(37,100), PDGX(37,100), PDGY(37,100),
     2       PDA(37,100), PDGA(37,100), PDR(37,100), PDT(37,100),
     3       PDGR(37,100), PDGT(37,100), PDS(37,100), PDGS(37,100),
     4       PDOLX(37), PDOLY(37), PDOGX(37), PDOGY(37), PDORLX(37),
     5       PDORLY(37), PDORGX(37), PDORGY(37), PDCRX(37), PDCRY(37),
     6       PDOLR(37), PDOLT(37), PDOGR(37), PDOGT(37), PDORLR(37),
     7       PDORLT(37), PDORGR(37), PDORGT(37), PDCRR(37), PDCRT(37),
     8       PDOGA(37), PDCRA(37), PDCRS(37), TLA(105), TLG(105),
     9       DEVX(37), DEVY(37), DEVA(37), DEVR(37), DEVT(37), DEVS(37),
     *       FACTOR, MOTN
      COMMON /XYZNUR/ XGR(37), YGR(37), SGR(37), AGR(37), RGR(37),
     1       TGR(37)
      COMMON /REST/ CONFIG(10), DELT(361), SLIDE(10), CRNKI, HPI, PI,
     1       DEGRAD, RINC, TWOSTA, IJOINT(10), LARCD(100), LARCF(100),
     2       LARCX(20), LOOPL(10), KOL, KOLL(5), KTHETA, LBA, LKFIN,
     3       LKOUT, LKR, LKRET, MOVER, NFROUT, NKOL, NLIN, NOUTLK,
     4       NOUTLP, IDEL, IDELV, IDELA, IVEL, IACC, LCOL(5), TANGLE
      EQUIVALENCE (DESX(1),DESY(1),DESANG(1),ANMULT(1),XMULT(1),YMULT(1)
     1           ,OPLOW(1),OPHIGH(1)), (IKON(1),IVA(1),IVG(1)),
     2            (XO(1),AO(1),RO(1),SO(1)), (YO(1),TO(1))
C
C     INPUT AND OUTPUT DEVICES
C     5 IS FOR READING THE NAMELIST DATA SETS:&LINKS,&METHOD,&LIMITS
C     6 IS FOR WRITING RESULTS
C     7 IS FOR WRITING RESULTS FOR GRAPHICAL PLOTTING
C
      NAMELIST /LINKS/ ARCL, GAMA, ILINK, ILOOP, MAXARC, NLOOP, OFFSET,
     1         OUTLK, OUTANG, NOUTLP, NOUTLK, IUNITS /METHOD/ DELT,
     2         DELTD, DELT2D, IDEL, IDELV, IDELA, DESX, DESY, DESANG,
     3         CRNKI, CRNKD, CRNK2D, NPOINT, XMULT, YMULT, ANMULT, IVA,
     4         IVG, SMALL, FMIN, VAR, EXPAND, REDUCE, SHRINK, MOTION,
     5         NGUESS, ILKOUT, INGOUT, ICRK, IORIG, IKIN, XADD, YADD,
     6         BOUNCE, ORIGLK, ORIGNG, ISTART, NDES, SNAKE, NITER,
     7         NCHECK, IOFF /LIMITS/ ORIGNG, ORIGLK, TANGLE, IANG, IVEL,
     8         IACC, IPLOT, LAST, KTHETA, OPLOW, OPHIGH, TOGGLE, STEPIN,
     9         LIMVAL, NPRINT, NPRT, NVIOL, COSTA, COSTG, OUTOLX,
     *         OUTOLY, OUTOLA, OUTOLR, OUTOLT, OUTOLS
C
      CALL FTNCMD('$EMPTY FU7;')
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
      CALL FTNCMD('ASSIGN 7=FU7(*L+1);')
   10 CALL TIME(0)
      PI = 4.0 * DATAN(1.0D0)
      HPI = PI / 2.0
      DEGRAD = PI / 180.0
      TANGLE = 50.0
      CRNKI = 0
      CRNKD = 0
      CRNK2D = 0
      OFFSET = 0
      ORIGNG = 0
      ORIGLK = 0
      OUTANG = 0
      OUTLK = 0
      IPLOT = 0
      IANG = 9999
      IDEL = 0
      KTHETA = 0
      LAST = 1
      NDES = 0
      NPOINT = 0
      MOTION = 9999
      IANG = 9999
      FACTOR = 1.0D+0
      IUNITS = 1
      DO 20 IC = 1, 105
        COSTA(IC) = 1.0D+0
        COSTG(IC) = 1.0D+0
   20 CONTINUE
C
      READ (5,1050) ITITLE
      READ (5,LINKS)
      READ (5,METHOD)
      READ (5,LIMITS)
C
      IF (NDES .EQ. 0) NDES = NPOINT
      IF (MOTION .GT. 3) MOTION = IANG
C
      KKL = NLOOP * MAXARC
      KKM = KKL + 1
      KKN = KKM + 1
      KKO = KKN + 1
      DO 30 I = 1, KKL
        IILOOP(I) = ILOOP(I)
        IILINK(I) = ILINK(I)
        KLINK(I) = ILINK(I)
   30 CONTINUE
      ALPHA = 2.5D-1 * DEGRAD
      IF (IUNITS .EQ. 1) CLIMIT = 1.0D-1
      IF (IUNITS .EQ. 2) CLIMIT = 4.0D-3
      DO 40 J = 1, KKO
        TLA(J) = 1.0D+3
        TLG(J) = 1.0D+3
        TOLER(J) = CLIMIT
        TOLEG(J) = 2.5D-1
        TOLED(J) = ALPHA
        DIFA(J) = 0.0D+0
        DIFG(J) = 0.0D+0
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
   40 CONTINUE
      KTHETA = KTHETA + 1
      ARCL(KKM) = OUTLK
      ARCL(KKN) = ORIGLK
      ARCL(KKO + 1) = OFFSET
      GAMA(KKM) = OUTANG
      GAMA(KKN) = ORIGNG
      GAMA(KKO) = CRNKI
      DO 50 I = 1, KKN
        AARCL(I) = ARCL(I)
        GGAMA(I) = GAMA(I)
   50 CONTINUE
      WRITE (6,1060) ITITLE
      IF (IUNITS .EQ. 1) WRITE (6,1070)
      IF (IUNITS .EQ. 2) WRITE (6,1080)
C
      MOTN = MOTION + 1
C
      WRITE (7) ITITLE
      WRITE (7) MOTN, NDES, IUNITS, MAXARC, NLOOP, KKL
      WRITE (7) (DELT(I),I=1,NDES)
C
      NCK = 0
      CALL KALM
      GO TO (60, 80, 100, 120), MOTN
C
   60 DO 70 IP = 1, NDES
        XM(IP) = XGR(IP)
        YM(IP) = YGR(IP)
   70 CONTINUE
      XYRAT = OUTOLX / OUTOLY
      WRITE (6,1090) OUTOLX, OUTOLY
      GO TO 140
C
   80 DO 90 IP = 1, NDES
        AM(IP) = AGR(IP)
   90 CONTINUE
      WRITE (6,1100) OUTOLA
      GO TO 140
C
  100 DO 110 IP = 1, NDES
        RM(IP) = RGR(IP)
        TM(IP) = TGR(IP)
  110 CONTINUE
      RTRAT = OUTOLR / (OUTOLT*DEGRAD)
      WRITE (6,1110) OUTOLR, OUTOLT
      GO TO 140
C
  120 DO 130 IP = 1, NDES
        SM(IP) = SGR(IP)
  130 CONTINUE
      WRITE (6,1120) OUTOLS
C
  140 CALL RESET(1)
      NCK = NCK + 1
      NNUR = 0
      IRNUR = 0
      IGNUR = 0
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
C
  150 NNUR = NNUR + 1
      IREP(NNUR) = 0
      ITERN(NNUR) = 0
      IF (NNUR .GT. KKL) GO TO 550
      LPNUR = (NNUR - 1) / MAXARC + 1
      IF (ILINK(NNUR)) 160, 150, 170
  160 IF (ILOOP(NNUR)) 150, 210, 180
  170 IF (ILOOP(NNUR)) 150, 210, 190
  180 JL = (ILOOP(NNUR) - 1) * MAXARC - ILINK(NNUR)
      IF (ILOOP(JL) .LT. 0) GO TO 210
      IF (GAMA(NNUR) .EQ. 1.8D02 .AND. ARCL(NNUR) .EQ. ARCL(JL))
     1   GO TO 200
      IF (ITERN(JL) .EQ. 0) ITERN(JL) = JL
      ITERN(NNUR) = ITERN(JL)
      GO TO 340
  190 IF (ILOOP(NNUR) .EQ. LPNUR) GO TO 210
      ILINK(NNUR) = -ILINK(NNUR)
      IILINK(NNUR) = ILINK(NNUR)
C
      JL = (ILOOP(NNUR) - 1) * MAXARC - ILINK(NNUR)
  200 IF (IREP(JL) .EQ. 0) IREP(JL) = JL
      IREP(NNUR) = IREP(JL)
      IF (ILOOP(IREP(NNUR)) .LT. 0) GO TO 150
C
C
  210 IRNUR = IRNUR + 1
      MRNUR(IRNUR) = NNUR
C
      IF (ARCL(NNUR) .GE. TOLER(NNUR)) ARCL(NNUR) = ARCL(NNUR) - TOLER(
     1NNUR)
      TDF = 1.0D+0
      IF (ARCL(NNUR) .LT. TOLER(NNUR)) TDF = 5.0D-1
      CALL KALM
C
      GO TO (220, 250, 280, 310), MOTN
C
  220 DO 230 IP = 1, NDES
        XO(IP) = XGR(IP)
        YO(IP) = YGR(IP)
  230 CONTINUE
      CALL RESET(2)
      ARCL(NNUR) = ARCL(NNUR) + TOLER(NNUR)
      CALL KALM
      PDAP(NNUR) = 0.0D+0
      DO 240 IP = 1, NDES
        PDX(IP,NNUR) = (DABS(XGR(IP) - XM(IP)) + DABS(XM(IP) - XO(IP)))
     1  / (2.0*TOLER(NNUR)*TDF)
        IF (PDX(IP,NNUR) .GT. PDAP(NNUR)) PDAP(NNUR) = PDX(IP,NNUR)
        PDY(IP,NNUR) = (DABS(YGR(IP) - YM(IP)) + DABS(YM(IP) - YO(IP)))
     1  / (2.0*TOLER(NNUR)*TDF)
        IF (XYRAT*PDY(IP,NNUR) .GT. PDAP(NNUR)) PDAP(NNUR) = XYRAT *
     1  PDY(IP,NNUR)
  240 CONTINUE
      CALL RESET(2)
      GO TO 150
C
  250 DO 260 IP = 1, NDES
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
          AO(IP) = AGR(IP)
  260 CONTINUE
      CALL RESET(2)
      ARCL(NNUR) = ARCL(NNUR) + TOLER(NNUR)
      CALL KALM
      PDAP(NNUR) = 0.0D+0
      DO 270 IP = 1, NDES
        PDA(IP,NNUR) = (DABS(AGR(IP) - AM(IP)) + DABS(AM(IP) - AO(IP)))
     1  / (2.0*TOLER(NNUR)*TDF)
        IF (PDA(IP,NNUR) .GT. PDAP(NNUR)) PDAP(NNUR) = PDA(IP,NNUR)
  270 CONTINUE
      CALL RESET(2)
      GO TO 150
C
  280 DO 290 IP = 1, NDES
        RO(IP) = RGR(IP)
        TO(IP) = TGR(IP)
  290 CONTINUE
      CALL RESET(2)
      ARCL(NNUR) = ARCL(NNUR) + TOLER(NNUR)
      CALL KALM
      PDAP(NNUR) = 0.0D+0
      DO 300 IP = 1, NDES
        PDR(IP,NNUR) = (DABS(RGR(IP) - RM(IP)) + DABS(RM(IP) - RO(IP)))
     1  / (2.0*TOLER(NNUR)*TDF)
        IF (PDR(IP,NNUR) .GT. PDAP(NNUR)) PDAP(NNUR) = PDR(IP,NNUR)
        PDT(IP,NNUR) = (DABS(TGR(IP) - TM(IP)) + DABS(TM(IP) - TO(IP)))
     1  / (2.0*TOLER(NNUR)*TDF)
        IF (RTRAT*PDT(IP,NNUR) .GT. PDAP(NNUR)) PDAP(NNUR) = RTRAT *
     1  PDT(IP,NNUR)
  300 CONTINUE
      CALL RESET(2)
      GO TO 150
C
  310 DO 320 IP = 1, NDES
        SO(IP) = SGR(IP)
  320 CONTINUE
      CALL RESET(2)
      ARCL(NNUR) = ARCL(NNUR) + TOLER(NNUR)
      CALL KALM
      PDAP(NNUR) = 0.0D+0
      DO 330 IP = 1, NDES
        PDS(IP,NNUR) = (DABS(SGR(IP) - SM(IP)) + DABS(SM(IP) - SO(IP)))
     1  / (2.0*TOLER(NNUR)*TDF)
        IF (PDS(IP,NNUR) .GT. PDAP(NNUR)) PDAP(NNUR) = PDS(IP,NNUR)
  330 CONTINUE
      CALL RESET(2)
      GO TO 150
C
C
  340 IRNUR = IRNUR + 1
      IGNUR = IGNUR + 1
      MRNUR(IRNUR) = NNUR
      MGNUR(IGNUR) = NNUR
C
      IF (ARCL(NNUR) .GE. TOLER(NNUR)) ARCL(NNUR) = ARCL(NNUR) - TOLER(
     1NNUR)
      TDF = 1.0D+0
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
      IF (ARCL(NNUR) .LT. TOLER(NNUR)) TDF = 5.0D-1
      CALL KALM
C
      GO TO (350, 400, 450, 500), MOTN
C
  350 DO 360 IP = 1, NDES
        XO(IP) = XGR(IP)
        YO(IP) = YGR(IP)
  360 CONTINUE
      CALL RESET(2)
      ARCL(NNUR) = ARCL(NNUR) + TOLER(NNUR)
      CALL KALM
      PDAP(NNUR) = 0.0D+0
      DO 370 IP = 1, NDES
        PDX(IP,NNUR) = (DABS(XGR(IP) - XM(IP)) + DABS(XM(IP) - XO(IP)))
     1  / (2.0*TOLER(NNUR)*TDF)
        IF (PDX(IP,NNUR) .GT. PDAP(NNUR)) PDAP(NNUR) = PDX(IP,NNUR)
        PDY(IP,NNUR) = (DABS(YGR(IP) - YM(IP)) + DABS(YM(IP) - YO(IP)))
     1  / (2.0*TOLER(NNUR)*TDF)
        IF (XYRAT*PDY(IP,NNUR) .GT. PDAP(NNUR)) PDAP(NNUR) = XYRAT *
     1  PDY(IP,NNUR)
  370 CONTINUE
      CALL RESET(2)
C
      GAMA(NNUR) = GAMA(NNUR) - TOLEG(NNUR)
      CALL KALM
      DO 380 IP = 1, NDES
        XO(IP) = XGR(IP)
        YO(IP) = YGR(IP)
  380 CONTINUE
      CALL RESET(2)
      GAMA(NNUR) = GAMA(NNUR) + TOLEG(NNUR)
      CALL KALM
      PGAP(NNUR) = 0.0D+0
      DO 390 IP = 1, NDES
        PDGX(IP,NNUR) = (DABS(XGR(IP) - XM(IP)) + DABS(XM(IP) - XO(IP)))
     1   / (2.0D+0*TOLED(NNUR))
        IF (PDGX(IP,NNUR) .GT. PGAP(NNUR)) PGAP(NNUR) = PDGX(IP,NNUR)
        PDGY(IP,NNUR) = (DABS(YGR(IP) - YM(IP)) + DABS(YM(IP) - YO(IP)))
     1   / (2.0D+0*TOLED(NNUR))
        IF (XYRAT*PDGY(IP,NNUR) .GT. PGAP(NNUR)) PGAP(NNUR) = XYRAT *
     1  PDGY(IP,NNUR)
  390 CONTINUE
      CALL RESET(2)
      GO TO 150
C
  400 DO 410 IP = 1, NDES
        AO(IP) = AGR(IP)
  410 CONTINUE
      CALL RESET(2)
      ARCL(NNUR) = ARCL(NNUR) + TOLER(NNUR)
      CALL KALM
      PDAP(NNUR) = 0.0D+0
      DO 420 IP = 1, NDES
        PDA(IP,NNUR) = (DABS(AGR(IP) - AM(IP)) + DABS(AM(IP) - AO(IP)))
     1  / (2.0*TOLER(NNUR)*TDF)
        IF (PDA(IP,NNUR) .GT. PDAP(NNUR)) PDAP(NNUR) = PDA(IP,NNUR)
  420 CONTINUE
```

```
            CALL RESET(2)
      C
            GAMA(NNUR) = GAMA(NNUR) - TOLEG(NNUR)
            CALL KALM
            DO 430 IP = 1, NDES
              AO(IP) = AGR(IP)
        430 CONTINUE
            CALL RESET(2)
            GAMA(NNUR) = GAMA(NNUR) + TOLEG(NNUR)
            CALL KALM
            PGAP(NNUR) = 0.0D+0
            DO 440 IP = 1, NDES
              PDGA(IP,NNUR) = (DABS(AGR(IP) - AM(IP)) + DABS(AM(IP) - AO(IP)))
           1   / (2.0D+0*TOLED(NNUR))
              IF (PDGA(IP,NNUR) .GT. PGAP(NNUR)) PGAP(NNUR) = PDGA(IP,NNUR)
        440 CONTINUE
            CALL RESET(2)
            GO TO 150
      C
        450 DO 460 IP = 1, NDES
              RO(IP) = RGR(IP)
              TO(IP) = TGR(IP)
        460 CONTINUE
            CALL RESET(2)
            ARCL(NNUR) = ARCL(NNUR) + TOLER(NNUR)
            CALL KALM
            PDAP(NNUR) = 0.0D+0
            DO 470 IP = 1, NDES
              PDR(IP,NNUR) = (DABS(RGR(IP) - RM(IP)) + DABS(RM(IP) - RO(IP)))
           1  / (2.0*TOLER(NNUR)*TDF)
              IF (PDR(IP,NNUR) .GT. PDAP(NNUR)) PDAP(NNUR) = PDR(IP,NNUR)
              PDT(IP,NNUR) = (DABS(TGR(IP) - TM(IP)) + DABS(TM(IP) - TO(IP)))
           1  / (2.0*TOLER(NNUR)*TDF)
              IF (RTRAT*PDT(IP,NNUR) .GT. PDAP(NNUR)) PDAP(NNUR) = RTRAT *
           1  PDT(IP,NNUR)
        470 CONTINUE
            CALL RESET(2)
      C
            GAMA(NNUR) = GAMA(NNUR) - TOLEG(NNUR)
            CALL KALM
            DO 480 IP = 1, NDES
              RO(IP) = RGR(IP)
              TO(IP) = TGR(IP)
        480 CONTINUE
            CALL RESET(2)
            GAMA(NNUR) = GAMA(NNUR) + TOLEG(NNUR)
            CALL KALM
            PGAP(NNUR) = 0.0D+0
            DO 490 IP = 1, NDES
              PDGR(IP,NNUR) = (DABS(RGR(IP) - RM(IP)) + DABS(RM(IP) - RO(IP)))
           1   / (2.0D+0*TOLED(NNUR))
              IF (PDGR(IP,NNUR) .GT. PGAP(NNUR)) PGAP(NNUR) = PDGR(IP,NNUR)
              PDGT(IP,NNUR) = (DABS(TGR(IP) - TM(IP)) + DABS(TM(IP) - TO(IP)))
           1   / (2.0D+0*TOLED(NNUR))
              IF (RTRAT*PDGT(IP,NNUR) .GT. PGAP(NNUR)) PGAP(NNUR) = RTRAT *
           1  PDGT(IP,NNUR)
        490 CONTINUE
            CALL RESET(2)
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
          GO TO 150
C
      500 DO 510 IP = 1, NDES
            SO(IP) = SGR(IP)
      510 CONTINUE
          CALL RESET(2)
          ARCL(NNUR) = ARCL(NNUR) + TOLER(NNUR)
          CALL KALM
          PDAP(NNUR) = 0.0D+0
          DO 520 IP = 1, NDES
            PDS(IP,NNUR) = (DABS(SGR(IP) - SM(IP)) + DABS(SM(IP) - SO(IP)))
         1  / (2.0*TOLER(NNUR)*TDF)
            IF (PDS(IP,NNUR) .GT. PDAP(NNUR)) PDAP(NNUR) = PDS(IP,NNUR)
      520 CONTINUE
          CALL RESET(2)
C
          GAMA(NNUR) = GAMA(NNUR) - TOLEG(NNUR)
          CALL KALM
          DO 530 IP = 1, NDES
            SO(IP) = SGR(IP)
      530 CONTINUE
          CALL RESET(2)
          GAMA(NNUR) = GAMA(NNUR) + TOLEG(NNUR)
          CALL KALM
          PGAP(NNUR) = 0.0D+0
          DO 540 IP = 1, NDES
            PDGS(IP,NNUR) = (DABS(SGR(IP) - SM(IP)) + DABS(SM(IP) - SO(IP)))
         1   / (2.0D+0*TOLED(NNUR))
            IF (PDGS(IP,NNUR) .GT. PGAP(NNUR)) PGAP(NNUR) = PDGS(IP,NNUR)
      540 CONTINUE
          CALL RESET(2)
          GO TO 150
C
C
      550 GO TO (560, 670, 720, 830), MOTN
C
      560 IF (ARCL(KKM) .LT. 0.1D-3) GO TO 605
          ARCL(KKM) = ARCL(KKM) - TOLER(KKM)
          CALL KALM
          DO 570 IP = 1, NDES
            XO(IP) = XGR(IP)
            YO(IP) = YGR(IP)
      570 CONTINUE
          CALL RESET(2)
          ARCL(KKM) = ARCL(KKM) + TOLER(KKM)
          CALL KALM
          PDOLAP = 0.0D+0
          DO 580 IP = 1, NDES
            PDOLX(IP) = (DABS(XGR(IP) - XM(IP)) + DABS(XM(IP) - XO(IP))) / (
         1  2.0D+0*TOLER(KKM))
            IF (PDOLX(IP) .GT. PDOLAP) PDOLAP = PDOLX(IP)
            PDOLY(IP) = (DABS(YGR(IP) - YM(IP)) + DABS(YM(IP) - YO(IP))) / (
         1  2.0D+0*TOLER(KKM))
            IF (XYRAT*PDOLY(IP) .GT. PDOLAP) PDOLAP = XYRAT * PDOLY(IP)
      580 CONTINUE
          CALL RESET(2)
C
          IF (ARCL(KKM) .LT. 0.1D-3) GO TO 605
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
      GAMA(KKM) = GAMA(KKM) - TOLEG(KKM)
      CALL KALM
      DO 590 IP = 1, NDES
        XO(IP) = XGR(IP)
        YO(IP) = YGR(IP)
  590 CONTINUE
      CALL RESET(2)
      GAMA(KKM) = GAMA(KKM) + TOLEG(KKM)
      CALL KALM
      PDOGAP = 0.0D+0
      DO 600 IP = 1, NDES
        PDOGX(IP) = (DABS(XGR(IP) - XM(IP)) + DABS(XM(IP) - XO(IP))) / (
     1  2.0D+0*TOLED(KKM))
        IF (PDOGX(IP) .GT. PDOGAP) PDOGAP = PDOGX(IP)
        PDOGY(IP) = (DABS(YGR(IP) - YM(IP)) + DABS(YM(IP) - YO(IP))) / (
     1  2.0D+0*TOLED(KKM))
        IF (XYRAT*PDOGY(IP) .GT. PDOGAP) PDOGAP = XYRAT * PDOGY(IP)
  600 CONTINUE
      CALL RESET(2)
C
  605 IF (ARCL(KKN) .LT. 0.1D-3) GO TO 645
      ARCL(KKN) = ARCL(KKN) - TOLER(KKN)
      CALL KALM
      DO 610 IP = 1, NDES
        XO(IP) = XGR(IP)
        YO(IP) = YGR(IP)
  610 CONTINUE
      CALL RESET(2)
      ARCL(KKN) = ARCL(KKN) + TOLER(KKN)
      CALL KALM
      PORLAP = 0.0D+0
      DO 620 IP = 1, NDES
        PDORLX(IP) = (DABS(XGR(IP) - XM(IP)) + DABS(XM(IP) - XO(IP))) /
     1  (2.0D+0*TOLER(KKN))
        IF (PDORLX(IP) .GT. PORLAP) PORLAP = PDORLX(IP)
        PDORLY(IP) = (DABS(YGR(IP) - YM(IP)) + DABS(YM(IP) - YO(IP))) /
     1  (2.0D+0*TOLER(KKN))
        IF (XYRAT*PDORLY(IP) .GT. PORLAP) PORLAP = XYRAT * PDORLY(IP)
  620 CONTINUE
      CALL RESET(2)
      ARCL(KKN) = ARCL(KKN) + 1.0D-3
C
      GAMA(KKN) = GAMA(KKN) - TOLEG(KKN)
      CALL KALM
      DO 630 IP = 1, NDES
        XO(IP) = XGR(IP)
        YO(IP) = YGR(IP)
  630 CONTINUE
      CALL RESET(2)
      GAMA(KKN) = GAMA(KKN) + TOLEG(KKN)
      CALL KALM
      PORGAP = 0.0D+0
      DO 640 IP = 1, NDES
        PDORGX(IP) = (DABS(XGR(IP) - XM(IP)) + DABS(XM(IP) - XO(IP))) /
     1  (2.0D+0*TOLED(KKN))
        IF (PDORGX(IP) .GT. PORGAP) PORGAP = PDORGX(IP)
        PDORGY(IP) = (DABS(YGR(IP) - YM(IP)) + DABS(YM(IP) - YO(IP))) /
     1  (2.0D+0*TOLED(KKN))
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
            IF (XYRAT*PDORGY(IP) .GT. PORGAP) PORGAP = XYRAT * PDORGY(IP)
  640 CONTINUE
      CALL RESET(2)
C
  645 CRNKI = CRNKI - TOLEG(KKO)
      CALL KALM
      DO 650 IP = 1, NDES
        XO(IP) = XGR(IP)
        YO(IP) = YGR(IP)
  650 CONTINUE
      CALL RESET(2)
      CRNKI = CRNKI + TOLEG(KKO)
      CALL KALM
      PDCRAP = 0.0D+0
      DO 660 IP = 1, NDES
        PDCRX(IP) = (DABS(XGR(IP) - XM(IP)) + DABS(XM(IP) - XO(IP))) / (
     1  2.0D+0*TOLED(KKO))
        IF (PDCRX(IP) .GT. PDCRAP) PDCRAP = PDCRX(IP)
        PDCRY(IP) = (DABS(YGR(IP) - YM(IP)) + DABS(YM(IP) - YO(IP))) / (
     1  2.0D+0*TOLED(KKO))
        IF (XYRAT*PDCRY(IP) .GT. PDCRAP) PDCRAP = XYRAT * PDCRY(IP)
  660 CONTINUE
      GO TO 860
C
  670 GAMA(KKM) = GAMA(KKM) - TOLEG(KKM)
      CALL KALM
      DO 680 IP = 1, NDES
        AO(IP) = AGR(IP)
  680 CONTINUE
      CALL RESET(2)
      GAMA(KKM) = GAMA(KKM) + TOLEG(KKM)
      CALL KALM
      PDOGAP = 0.0D+0
      DO 690 IP = 1, NDES
        PDOGA(IP) = (DABS(AGR(IP) - AM(IP)) + DABS(AM(IP) - AO(IP))) / (
     1  2.0D+0*TOLED(KKM))
        IF (PDOGA(IP) .GT. PDOGAP) PDOGAP = PDOGA(IP)
  690 CONTINUE
      CALL RESET(2)
C
      CRNKI = CRNKI - TOLEG(KKO)
      CALL KALM
      DO 700 IP = 1, NDES
        AO(IP) = AGR(IP)
  700 CONTINUE
      CALL RESET(2)
      CRNKI = CRNKI + TOLEG(KKO)
      CALL KALM
      PDCRAP = 0.0D+0
      DO 710 IP = 1, NDES
        PDCRA(IP) = (DABS(AGR(IP) - AM(IP)) + DABS(AM(IP) - AO(IP))) / (
     1  2.0D+0*TOLED(KKO))
        IF (PDCRA(IP) .GT. PDCRAP) PDCRAP = PDCRA(IP)
  710 CONTINUE
      GO TO 860
C
  720 IF (ARCL(KKM) .LT. 0.1D-3) GO TO 765
      ARCL(KKM) = ARCL(KKM) - TOLER(KKM)
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
         CALL KALM
         DO 730 IP = 1, NDES
           RO(IP) = RGR(IP)
           TO(IP) = TGR(IP)
     730 CONTINUE
         CALL RESET(2)
         ARCL(KKM) = ARCL(KKM) + TOLER(KKM)
         CALL KALM
         PDOLAP = 0.0D+0
         DO 740 IP = 1, NDES
           PDOLR(IP) = (DABS(RGR(IP) - RM(IP)) + DABS(RM(IP) - RO(IP))) / (
        1  2.0D+0*TOLER(KKM))
           IF (PDOLR(IP) .GT. PDOLAP) PDOLAP = PDOLR(IP)
           PDOLT(IP) = (DABS(TGR(IP) - TM(IP)) + DABS(TM(IP) - TO(IP))) / (
        1  2.0D+0*TOLER(KKM))
           IF (RTRAT*PDOLT(IP) .GT. PDOLAP) PDOLAP = RTRAT * PDOLT(IP)
     740 CONTINUE
         CALL RESET(2)
   C
         GAMA(KKM) = GAMA(KKM) - TOLEG(KKM)
         CALL KALM
         DO 750 IP = 1, NDES
           RO(IP) = RGR(IP)
           TO(IP) = TGR(IP)
     750 CONTINUE
         CALL RESET(2)
         GAMA(KKM) = GAMA(KKM) + TOLEG(KKM)
         CALL KALM
         PDOGAP = 0.0D+0
         DO 760 IP = 1, NDES
           PDOGR(IP) = (DABS(RGR(IP) - RM(IP)) + DABS(RM(IP) - RO(IP))) / (
        1  2.0D+0*TOLED(KKM))
           IF (PDOGR(IP) .GT. PDOGAP) PDOGAP = PDOGR(IP)
           PDOGT(IP) = (DABS(TGR(IP) - TM(IP)) + DABS(TM(IP) - TO(IP))) / (
        1  2.0D+0*TOLED(KKM))
           IF (RTRAT*PDOGT(IP) .GT. PDOGAP) PDOGAP = RTRAT * PDOGT(IP)
     760 CONTINUE
         CALL RESET(2)
   C
     765 IF (ARCL(KKN) .LT. 0.1D-3) GO TO 805
         ARCL(KKN) = ARCL(KKN) - TOLER(KKN)
         CALL KALM
         DO 770 IP = 1, NDES
           RO(IP) = RGR(IP)
           TO(IP) = TGR(IP)
     770 CONTINUE
         CALL RESET(2)
         ARCL(KKN) = ARCL(KKN) + TOLER(KKN)
         CALL KALM
         PORLAP = 0.0D+0
         DO 780 IP = 1, NDES
           PDORLR(IP) = (DABS(RGR(IP) - RM(IP)) + DABS(RM(IP) - RO(IP))) /
        1  (2.0D+0*TOLER(KKN))
           IF (PDORLR(IP) .GT. PORLAP) PORLAP = PDORLR(IP)
           PDORLT(IP) = (DABS(TGR(IP) - TM(IP)) + DABS(TM(IP) - TO(IP))) /
        1  (2.0D+0*TOLER(KKN))
           IF (RTRAT*PDORLT(IP) .GT. PORLAP) PORLAP = RTRAT * PDORLT(IP)
     780 CONTINUE
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
      CALL RESET(2)
      ARCL(KKN) = ARCL(KKN) + 1.0D-3
C
      GAMA(KKN) = GAMA(KKN) - TOLEG(KKN)
      CALL KALM
      DO 790 IP = 1, NDES
        RO(IP) = RGR(IP)
        TO(IP) = TGR(IP)
  790 CONTINUE
      CALL RESET(2)
      GAMA(KKN) = GAMA(KKN) + TOLEG(KKN)
      CALL KALM
      PORGAP = 0.0D+0
      DO 800 IP = 1, NDES
        PDORGR(IP) = (DABS(RGR(IP) - RM(IP)) + DABS(RM(IP) - RO(IP))) /
     1  (2.0D+0*TOLED(KKN))
        IF (PDORGR(IP) .GT. PORGAP) PORGAP = PDORGR(IP)
        PDORGT(IP) = (DABS(TGR(IP) - TM(IP)) + DABS(TM(IP) - TO(IP))) /
     1  (2.0D+0*TOLED(KKN))
        IF (RTRAT*PDORGT(IP) .GT. PORGAP) PORGAP = RTRAT * PDORGT(IP)
  800 CONTINUE
      CALL RESET(2)
C
  805 CRNKI = CRNKI - TOLEG(KKO)
      CALL KALM
      DO 810 IP = 1, NDES
        RO(IP) = RGR(IP)
        TO(IP) = TGR(IP)
  810 CONTINUE
      CALL RESET(2)
      CRNKI = CRNKI + TOLEG(KKO)
      CALL KALM
      PDCRAP = 0.0D+0
      DO 820 IP = 1, NDES
        PDCRR(IP) = (DABS(RGR(IP) - RM(IP)) + DABS(RM(IP) - RO(IP))) / (
     1  2.0D+0*TOLED(KKO))
        IF (PDCRR(IP) .GT. PDCRAP) PDCRAP = PDCRR(IP)
        PDCRT(IP) = (DABS(TGR(IP) - TM(IP)) + DABS(TM(IP) - TO(IP))) / (
     1  2.0D+0*TOLED(KKO))
        IF (RTRAT*PDCRT(IP) .GT. PDCRAP) PDCRAP = RTRAT * PDCRT(IP)
  820 CONTINUE
      GO TO 860
C
  830 CRNKI = CRNKI - TOLEG(KKO)
      CALL KALM
      DO 840 IP = 1, NDES
        SO(IP) = SGR(IP)
  840 CONTINUE
      CALL RESET(2)
      CRNKI = CRNKI + TOLEG(KKO)
      CALL KALM
      PDCRAP = 0.0D+0
      DO 850 IP = 1, NDES
        PDCRS(IP) = (DABS(SGR(IP) - SM(IP)) + DABS(SM(IP) - SO(IP))) / (
     1  2.0D+0*TOLED(KKO))
        IF (PDCRS(IP) .GT. PDCRAP) PDCRAP = PDCRS(IP)
  850 CONTINUE
C
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
C
  860 GO TO (870, 950, 870, 950), MOTN
C
  870 SUM = 0.0D+0
      DO 880 NIMRA = 1, IRNUR
        SUM = SUM + PDAP(MRNUR(NIMRA)) * DSQRT(COSTA(MRNUR(NIMRA)))
  880 CONTINUE
      IF (IGNUR .EQ. 0) GO TO 900
      DO 890 NIMRA = 1, IGNUR
        SUM = SUM + PGAP(MGNUR(NIMRA)) * DSQRT(COSTG(MGNUR(NIMRA)))
  890 CONTINUE
  900 IF (ARCL(KKM) .LT. 0.1D-3) GO TO 903
      SUM = SUM + PDOLAP * DSQRT(COSTA(KKM))
     1      + PDOGAP * DSQRT(COSTG(KKM))
  903 IF (ARCL(KKN) .LT. 0.1D-3) GO TO 907
      SUM = SUM + PORLAP * DSQRT(COSTA(KKN))
     1      + PORGAP * DSQRT(COSTG(KKN))
  907 SUM = SUM + PDCRAP * DSQRT(COSTG(KKO))
      IF (MOTN .EQ. 3) OUTOLX = OUTOLR
      TLA(1) = OUTOLX / DSQRT(PDAP(1)*SUM/DSQRT(COSTA(1)))
      DIFA(1) = DABS(TLA(1)*FACTOR - TOLER(1))
      CONST = TLA(1) * DSQRT(PDAP(1)/DSQRT(COSTA(1)))
      DO 910 NIMRA = 2, IRNUR
        IZ = MRNUR(NIMRA)
        TLA(IZ) = CONST * DSQRT(DSQRT(COSTA(IZ))/PDAP(IZ))
        DIFA(IZ) = DABS(TLA(IZ)*FACTOR - TOLER(IZ))
  910 CONTINUE
      IF (IGNUR .EQ. 0) GO TO 930
      DO 920 NIMRA = 1, IGNUR
        IZ = MGNUR(NIMRA)
        TLG(IZ) = CONST * DSQRT(DSQRT(COSTG(IZ))/PGAP(IZ))
        DIFG(IZ) = DABS(TLG(IZ)*FACTOR - TOLED(IZ))
  920 CONTINUE
  930 IF (ARCL(KKM) .LT. 0.1D-3) GO TO 933
      TLA(KKM) = CONST * DSQRT(DSQRT(COSTA(KKM))/PDOLAP)
      DIFA(KKM) = DABS(TLA(KKM)*FACTOR - TOLER(KKM))
      TLG(KKM) = CONST * DSQRT(DSQRT(COSTG(KKM))/PDOGAP)
      DIFG(KKM) = DABS(TLG(KKM)*FACTOR - TOLED(KKM))
  933 IF (ARCL(KKN) .LT. 0.1D-3) GO TO 937
      TLA(KKN) = CONST * DSQRT(DSQRT(COSTA(KKN))/PORLAP)
      DIFA(KKN) = DABS(TLA(KKN)*FACTOR - TOLER(KKN))
      TLG(KKN) = CONST * DSQRT(DSQRT(COSTG(KKN))/PORGAP)
      DIFG(KKN) = DABS(TLG(KKN)*FACTOR - TOLED(KKN))
  937 TLG(KKO) = CONST * DSQRT(DSQRT(COSTG(KKO))/PDCRAP)
      DIFG(KKO) = DABS(TLG(KKO)*FACTOR - TOLED(KKO))
C
      CALL COMARC(IREP, TLA, KKL)
      CALL CORFAC(NDES, KKL, KKM, KKN, KKO)
C
      IF (NCK .EQ. 2) GO TO 940
      CALL CHECK(TLA, TLG, TOLER, TOLEG, TOLED, DIFA, DIFG, DEGRAD, KKO,
     1     ICHECK, MRNUR, IRNUR, MGNUR, IGNUR, MOTN)
      IF (ICHECK .EQ. 1) GO TO 140
C
  940 CALL ORDER(TLA, LORDER, KKN)
      CALL RESET(1)
      CALL RESULT(IREP, ITERN, KKL, KKM, NLOOP, MAXARC, LORDER, TLA,
     1     KKN, IGNUR, TLG, KKO, NOUTLK, NOUTLP, KLINK, ARCL, GAMA,
```

```
     2     MOTN, CLIMIT)
C
      IF (MOTN .EQ. 1) WRITE (7) OUTOLX, OUTOLY, (DEVX(KH),KH=1,NDES),
     1     (DEVY(JH),JH=1,NDES), ((PDX(JH,KH),JH=1,NDES),KH=1,KKL),
     2     ((PDY(JH,KH),JH=1,NDES),KH=1,KKL), ((PDGX(JH,KH),JH=1,NDES),
     3     KH=1,KKL), ((PDGY(JH,KH),JH=1,NDES),KH=1,KKL), (PDOLX(JH),
     4     JH=1,NDES), (PDOLY(JH),JH=1,NDES), (PDOGX(JH),JH=1,NDES),
     5     (PDOGY(JH),JH=1,NDES), (PDORLX(JH),JH=1,NDES),
     6     (PDORLY(JH),JH=1,NDES), (PDORGX(JH),JH=1,NDES),
     7     (PDORGY(JH),JH=1,NDES), (PDCRX(JH),JH=1,NDES),
     8     (PDCRY(JH),JH=1,NDES)
      IF (MOTN .EQ. 3) WRITE (7) OUTOLR, OUTOLT, (DEVR(KH),KH=1,NDES),
     1     (DEVT(JH),JH=1,NDES), ((PDR(JH,KH),JH=1,NDES),KH=1,KKL),
     2     ((PDT(JH,KH),JH=1,NDES),KH=1,KKL), ((PDGR(JH,KH),JH=1,NDES),
     3     KH=1,KKL), ((PDGT(JH,KH),JH=1,NDES),KH=1,KKL), (PDOLR(JH),
     4     JH=1,NDES), (PDOLT(JH),JH=1,NDES), (PDOGR(JH),JH=1,NDES),
     5     (PDOGT(JH),JH=1,NDES), (PDORLR(JH),JH=1,NDES),
     6     (PDORLT(JH),JH=1,NDES), (PDORGR(JH),JH=1,NDES),
     7     (PDORGT(JH),JH=1,NDES), (PDCRR(JH),JH=1,NDES),
     8     (PDCRT(JH),JH=1,NDES)
C
      WRITE (6,1130)
      CALL TIME(1, -1, TIM)
      GO TO 1040
C
C
  950 SUM = 0.0D+0
      DO 960 NIMRA = 1, IRNUR
        SUM = SUM + PDAP(MRNUR(NIMRA)) * DSQRT(COSTA(MRNUR(NIMRA)))
  960 CONTINUE
      IF (IGNUR .EQ. 0) GO TO 980
      DO 970 NIMRA = 1, IGNUR
        SUM = SUM + PGAP(MGNUR(NIMRA)) * DSQRT(COSTG(MGNUR(NIMRA)))
  970 CONTINUE
  980 SUM = SUM + PDCRAP * DSQRT(COSTG(KKO))
      IF (MOTN .EQ. 2) SUM = SUM + PDOGAP * DSQRT(COSTG(KKM))
      IF (MOTN .EQ. 2) TOLAS = OUTOLA * DEGRAD
      IF (MOTN .EQ. 4) TOLAS = OUTOLS
      TLA(1) = TOLAS / DSQRT(PDAP(1)*SUM/DSQRT(COSTA(1)))
      DIFA(1) = DABS(TLA(1)*FACTOR - TOLER(1))
      CONST = TLA(1) * DSQRT(PDAP(1)/DSQRT(COSTA(1)))
      DO 990 NIMRA = 2, IRNUR
        IZ = MRNUR(NIMRA)
        TLA(IZ) = CONST * DSQRT(DSQRT(COSTA(IZ))/PDAP(IZ))
        DIFA(IZ) = DABS(TLA(IZ)*FACTOR - TOLER(IZ))
  990 CONTINUE
      IF (IGNUR .EQ. 0) GO TO 1010
      DO 1000 NIMRA = 1, IGNUR
        IZ = MGNUR(NIMRA)
        TLG(IZ) = CONST * DSQRT(DSQRT(COSTG(IZ))/PGAP(IZ))
        DIFG(IZ) = DABS(TLG(IZ)*FACTOR - TOLED(IZ))
 1000 CONTINUE
 1010 IF (MOTN .EQ. 4) GO TO 1020
      TLG(KKM) = CONST * DSQRT(DSQRT(COSTG(KKM))/PDOGAP)
      DIFG(KKM) = DABS(TLG(KKM)*FACTOR - TOLED(KKM))
 1020 TLG(KKO) = CONST * DSQRT(DSQRT(COSTG(KKO))/PDCRAP)
      DIFG(KKO) = DABS(TLG(KKO)*FACTOR - TOLED(KKO))
C
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
            CALL COMARC(IREP, TLA, KKL)
            CALL CORFAC(NDES, KKL, KKM, KKN, KKO)
      C
            IF (NCK .EQ. 2) GO TO 1030
            CALL CHECK(TLA, TLG, TOLER, TOLEG, TOLED, DIFA, DIFG, DEGRAD, KKO,
           1    ICHECK, MRNUR, IRNUR, MGNUR, IGNUR, MOTN)
            IF (ICHECK .EQ. 1) GO TO 140
      C
       1030 CALL ORDER(TLA, LORDER, KKL)
            CALL RESET(1)
            CALL RESULT(IREP, ITERN, KKL, KKM, NLOOP, MAXARC, LORDER, TLA,
           1    KKN, IGNUR, TLG, KKO, NOUTLK, NOUTLP, KLINK, ARCL, GAMA,
           2    MOTN, CLIMIT)
      C
            IF (MOTN .EQ. 2) WRITE (7) OUTOLA, (DEVA(KH),KH=1,NDES),
           1      ((PDA(JH,KH),JH=1,NDES),KH=1,KKL), ((PDGA(JH,KH),JH=1,NDES),
           2      KH=1,KKL), (PDOGA(JH),JH=1,NDES), (PDCRA(JH),JH=1,NDES)
            IF (MOTN .EQ. 4) WRITE (7) OUTOLS, (DEVS(KH),KH=1,NDES),
           1      ((PDS(JH,KH),JH=1,NDES),KH=1,KKL), ((PDGS(JH,KH),JH=1,NDES),
           2      KH=1,KKL), (PDCRS(JH),JH=1,NDES)
      C
            WRITE (6,1130)
            CALL TIME(1, -1, TIM)
      C
       1040 WRITE (6,1140)
            IF (LAST .EQ. 0) GO TO 10
            STOP
      C
       1050 FORMAT (18A4)
       1060 FORMAT (1H1/' ********************************'/' ***/
           1            /*'/' **/ TOCALM ! OSMAN ! FEB 82 /**'/' */
           2              /***'/' ********************************', 1X, 18A4)
       1070 FORMAT (/5X, 'LINEAR DIMENSIONS ARE IN MILLIMETERS')
       1080 FORMAT (/5X, 'LINEAR DIMENSIONS ARE IN INCHES')
       1090 FORMAT (/5X, 'OUTPUT TOLERANCE (CARTESIAN CO-ORDINATES)'/10X, 'X-D
           1IRECTION=', E9.3, 5X, 'Y-DIRECTION=', E9.3)
       1100 FORMAT (/5X, 'OUTPUT TOLERANCE =', F6.3, ' DEGREES')
       1110 FORMAT (/5X, 'OUTPUT TOLERANCE (POLAR CO-ORDINATES)'/10X, 'RADIAL=
           1', E9.3, 5X, 'ANGULAR=', F5.3, ' DEGREES')
       1120 FORMAT (/5X, 'OUTPUT TOLERANCE (SLIDER DISPLACEMENT)=', E9.3)
       1130 FORMAT (///5X, 'EXECUTION TIME')
       1140 FORMAT (1H1)
            END
      C
      C * * * * * * * * * * * * * * * * * * * * * * * *
      C
            SUBROUTINE RESET(NM)
      C     ------------------
      C
            IMPLICIT REAL*8(A - H,O - Z)
      C
            COMMON ARCL(100), AARCL(100), GAMA(100), GGAMA(100), DEGINC,
           1       IKON(361), ILINK(100), ILOOP(100), ITITLE(18), IPLOT, KKL,
           2       KKM, KKN, KKO, MAXARC, NLOOP, NFORM, NPOINT, MOTOR, IANG,
           3       MOTION, NDES, KON, IILINK(100), IILOOP(100), KLINK(100)
      C
            DO 10 I = 1, KKL
              ILOOP(I) = IILOOP(I)
```

```
   10 CONTINUE
      IF (NM .EQ. 1) GO TO 30
      DO 20 I = 1, KKL
        ILINK(I) = IILINK(I)
   20 CONTINUE
      GO TO 50
   30 DO 40 I = 1, KKL
        ILINK(I) = KLINK(I)
        IILINK(I) = KLINK(I)
   40 CONTINUE
   50 DO 60 I = 1, KKN
        ARCL(I) = AARCL(I)
        GAMA(I) = GGAMA(I)
   60 CONTINUE
      RETURN
      END
C
C  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
C
      SUBROUTINE KALM
C     ---------------
C
      IMPLICIT REAL*8(A - H,O - Z)
      DIMENSION LAF(3), ILHD(10), IGHD(10), LAN(7), DESX(37), DESY(37),
     1          DESANG(37), ANMULT(37), XMULT(37), YMULT(37), OPLOW(30),
     2          OPHIGH(30), IVA(100), IVG(100)
      COMMON ARCL(100), AARCL(100), GAMA(100), GGAMA(100), DEGINC,
     1       IKON(361), ILINK(100), ILOOP(100), ITITLE(18), IPLOT, KKL,
     2       KKM, KKN, KKO, MAXARC, NLOOP, NFORM, NPOINT, MOTOR, IANG,
     3       MOTION, NDES, KON, IILINK(100), IILOOP(100), KLINK(100)
      COMMON /XYZNUR/ XGR(37), YGR(37), SGR(37), AGR(37), RGR(37),
     1       TGR(37)
      COMMON /REST/ CONFIG(10), DELT(361), SLIDE(10), CRNKI, HPI, PI,
     1       DEGRAD, RINC, TWOSTA, IJOINT(10), LARCD(100), LARCF(100),
     2       LARCX(20), LOOPL(10), KOL, KOLL(5), KTHETA, LBA, LKFIN,
     3       LKOUT, LKR, LKRET, MOVER, NFROUT, NKOL, NLIN, NOUTLK,
     4       NOUTLP, IDEL, IDELV, IDELA, IVEL, IACC, LCOL(5), TANGLE
      EQUIVALENCE (DESX(1),DESY(1),DESANG(1),ANMULT(1),XMULT(1),YMULT(1)
     1            ,OPLOW(1),OPHIGH(1)), (IKON(1),IVA(1),IVG(1))
C
      CALL SETUP
C
      DO 10 I = 1, KKL
        IF (ILOOP(I) .LT. 0) ARCL(I) = -0.00001
   10 CONTINUE
      IF (MOTOR) 30, 30, 20
   20 ARCL(KKO) = CRNKI
      GO TO 50
   30 GAMA(KKO) = CRNKI
C
      IF (IANG .LE. 3) GO TO 50
      IF (MOTION .LE. 3) GO TO 40
      GO TO 240
   40 IANG = MOTION
   50 NLPOUT = (NOUTLP - 1) * MAXARC
      IF (NOUTLK .LT. 0) GO TO 60
      LKOUT = NLPOUT + NOUTLK
      LKRET = LKOUT + 1
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
          MOUTLK = NOUTLK
          GO TO 70
       60 LKOUT = NLPOUT - NOUTLK
          LKRET = LKOUT
          MOUTLK = -NOUTLK
       70 LKR = LKRET - 1
C
C         SORT OUT LAYOUT FOR RESULTS
C
          I = IANG + 1
          GO TO (110, 90, 110, 80), I
       80 LBA = 3
          GO TO 100
       90 LBA = 1
      100 KOL = 2
          NFORM = 1
          GO TO 120
      110 LBA = 2
          KOL = 3
          NFORM = 7
      120 NCOL = 11 / KOL
C
C         INPUT LINK ANGLES FOR ANALYSIS
C
          IF (NDES .GT. 0) GO TO 130
          IF (NPOINT .GT. 0) GO TO 140
          GO TO 240
      130 IF (NPOINT .EQ. 0) IDEL = 1
          NPOINT = NDES
      140 IF (IDEL .NE. 0) GO TO 160
          IF (MOTOR .LE. 0) GO TO 150
          GO TO 240
      150 POINT = NPOINT
          DEGINC = 360.0 / POINT
          RINC = DSIGN(DEGINC,CRNKD)
          GAMA(KKO) = CRNKI - RINC
          NPOINT = NPOINT + 1
      160 MOVER = 1
          NLIN = (NPOINT + NCOL - 1) / NCOL
          NKOL = NCOL * KOL
          KOLA = KOL - NKOL * NLIN
          NP = NPOINT * KOL
          NLCOL = NPOINT - NCOL * (NLIN - 1)
          DO 170 I = 1, NLCOL
            KOLL(I) = (I - 1) * KOLA
      170 LCOL(I) = I * NLIN
          IF (NLCOL .EQ. NCOL) GO TO 190
          NMCOL = 1 + NLCOL
          NKOLA = NKOL + KOLA
          NKOLB = -NLCOL * NKOL
          DO 180 I = NMCOL, NCOL
            KOLL(I) = (I - 1) * NKOLA + NKOLB
      180 LCOL(I) = I * (NLIN - 1) + NLCOL
      190 NFROUT = NLPOUT + LOOPL(NOUTLP)
          LKFIN = NFROUT - 1
C
C         CHANGE ANGLE OF LAST FRAME ARC IN EACH LOOP
C         FROM RELATIVE TO FRAME ARC IN FIRST LOOP TO RELATIVE TO X-AXIS
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
C
      IFR = LOOPL(1)
      GAMA(IFR) = GAMA(IFR) - 180.0
      IF (NLOOP .EQ. 1) GO TO 220
      DO 210 L = 2, NLOOP
        I = (L - 1) * MAXARC + LOOPL(L)
        JJ = I
  200   J = LARCD(JJ)
        GAMA(I) = GAMA(I) + GAMA(J)
        IF (MOD(J,MAXARC) .EQ. IFR) GO TO 210
        IF (JJ .EQ. J) GO TO 240
        JJ = J
        GO TO 200
  210 CONTINUE
  220 CONTINUE
      DO 230 I = 1, KKN
  230 GAMA(I) = GAMA(I) * DEGRAD
      TANGLR = HPI - TANGLE * DEGRAD
      STANGL = DSIN(TANGLR)
      TWOSTA = 2.0 * STANGL
C
      CALL SOLVE
C
  240 RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE SETUP
C     ----------------
C     CONVERT TOPOLOGICAL INFORMATION INTO APPROPRIATE FORM FOR SOLVE
C
      IMPLICIT REAL*8(A - H,O - Z)
      DIMENSION LAI(3), LAN(7), IDENTA(100), JOINT(10)
      COMMON ARCL(100), AARCL(100), GAMA(100), GGAMA(100), DEGINC,
     1       IKON(361), ILINK(100), ILOOP(100), ITITLE(18), IPLOT, KKL,
     2       KKM, KKN, KKO, MAXARC, NLOOP, NFORM, NPOINT, MOTOR, IANG,
     3       MOTION, NDES, KON, IILINK(100), IILOOP(100), KLINK(100)
      COMMON /REST/ CONFIG(10), DELT(361), SLIDE(10), CRNKI, HPI, PI,
     1       DEGRAD, RINC, TWOSTA, IJOINT(10), LARCD(100), LARCF(100),
     2       LARCX(20), LOOPL(10), KOL, KOLL(5), KTHETA, LBA, LKFIN,
     3       LKOUT, LKR, LKRET, MOVER, NFROUT, NKOL, NLIN, NOUTLK,
     4       NOUTLP, IDEL, IDELV, IDELA, IVEL, IACC, LCOL(5), TANGLE
C
C     CONVERT ILINK,ILOOP TO LOOPL,LARCX,LARCD,LARCF,IDENTA,CONFIG
C
      IWARN = 0
      IERR = 0
      NLPTWO = NLOOP + NLOOP
      NJOINT = 1
      NLINK = 0
      IUNKNO = 0
      DO 500 I = 1, NLOOP
        ISWAP = 0
        LOOPL(I) = 0
        IND = (I - 1) * MAXARC
        DO 460 J = 1, MAXARC
          INDEX = J + IND
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
              INDLK = ILINK(INDEX)
              INDLP = ILOOP(INDEX)
              IDENTA(INDEX) = 0
              LARCD(INDEX) = 0
              LARCF(INDEX) = 0
              IF (INDLP) 10, 50, 360
      10      IF (INDEX - 2) 340, 460, 20
      20      IPRLP = ILOOP(INDEX - 1)
              IPRLK = ILINK(INDEX - 1)
              IF (IPRLP) 30, 40, 460
      30      IJUMP = 1
              IF (INDLP .EQ. (1 - J)) GO TO 110
              IF (IPRLP .EQ. - J) GO TO 100
              GO TO 90
      40      IJUMP = 2
              ISWAP = 1
              IF (-IABS(IPRLK) .EQ. INDLP) GO TO 150
              GO TO 130
      50      IF (INDLK .EQ. 0) GO TO 160
              IF (INDEX - 2) 350, 460, 60
      60      IPRLP = ILOOP(INDEX - 1)
              IPRLK = ILINK(INDEX - 1)
              IF (IPRLP) 70, 80, 460
      70      IJUMP = 2
              IF (-IABS(INDLK) .EQ. IPRLP) GO TO 140
              GO TO 120
C-----THREE REVOLUTE JOINTS
      80      IJOINT(I) = 1
              JOINT(I) = 1
              IJUMP = 1
              GO TO 170
C-----TWO SLIDERS ON DETERMINED LINKS
      90      IJOINT(I) = 2
              JOINT(I) = 8
              GO TO 250
C-----TWO SLIDERS ON FOLLOWING UNDETERMINED AND DETERMINED LINKS
     100      IJOINT(I) = 3
              JOINT(I) = 6
              GO TO 250
C-----TWO SLIDERS ON PRECEDING DETERMINED AND UNDETERMINED LINKS
     110      IJOINT(I) = 4
              JOINT(I) = 7
              GO TO 250
C-----SLIDER ON PRECEDING DETERMINED LINK
     120      IJOINT(I) = 5
              JOINT(I) = 5
              GO TO 250
C-----SLIDER ON FOLLOWING DETERMINED LINK
     130      IJOINT(I) = 5
              JOINT(I) = 4
              GO TO 170
C-----SLIDER ON FOLLOWING UNDETERMINED LINK
     140      IJOINT(I) = 6
              JOINT(I) = 2
              GO TO 250
C-----SLIDER ON PRECEDING UNDETERMINED LINK
     150      IJOINT(I) = 6
              JOINT(I) = 3
```

```
              GO TO 170
C-----NON-EXISTENT ARCS
  160         IF (LOOPL(I) .EQ. 0) LOOPL(I) = J - 1
              GO TO 460
C-----FIRST REVOLUTE JOINT
  170         IF (IPRLK) 180, 460, 190
  180         SLIDE(I) = 1.0
              GO TO 200
  190         SLIDE(I) = -1.0
  200         IUNKNO = IUNKNO + 1
              LARCX(IUNKNO) = J - 1
              NJOINT = NJOINT + 1
              NLINK = NLINK + 4
              GO TO (210, 290), IJUMP
C-----SECOND REVOLUTE JOINT
  210         IF (INDLK) 220, 460, 230
  220         CONFIG(I) = 1.0
              GO TO 240
  230         CONFIG(I) = -1.0
  240         IUNKNO = IUNKNO + 1
              LARCX(IUNKNO) = J
              IF (ISWAP) 460, 460, 330
C-----FIRST SLIDER
  250         IF (IPRLK) 260, 460, 270
  260         SLIDE(I) = HPI
              GO TO 280
  270         SLIDE(I) = -HPI
  280         IUNKNO = IUNKNO + 1
              LARCX(IUNKNO) = J - 1
              NJOINT = NJOINT + 1
              NLINK = NLINK + 4
              INDX = INDEX - 1
              LARCD(INDX) = IND - IPRLP
              LARCF(INDX) = -1
              GO TO (290, 210), IJUMP
C-----SECOND SLIDER
  290         IF (INDLK) 300, 460, 310
  300         CONFIG(I) = HPI
              GO TO 320
  310         CONFIG(I) = -HPI
  320         IUNKNO = IUNKNO + 1
              LARCX(IUNKNO) = J
              LARCD(INDEX) = IND - INDLP
              LARCF(INDEX) = -1
              IF (ISWAP) 460, 460, 330
  330         SLST = SLIDE(I)
              SLIDE(I) = -CONFIG(I)
              CONFIG(I) = -SLST
              LARCX(IUNKNO - 1) = J
              LARCX(IUNKNO) = J - 1
              GO TO 460
C-----SLIDING INPUT LINK
  340         PRISM = -INDLK
              GAMA(KKO) = DSIGN(90.D0,PRISM)
              MOTOR = 1
              INDLK = 1
              GO TO 450
C-----SPATIAL ROTARY INPUT
```

```
  350     MOTOR = -1
          GO TO 450
C-----DETERMINED ARC
  360     IF (INDLP .LT. I) GO TO 370
          IF (I .EQ. 1) GO TO 370
          IERR = 1
  370     IF (INDLK) 420, 460, 380
C-----DOUBLE JOINT
  380     IF (I .EQ. 1) GO TO 440
          INDSI = (INDLP - 1) * MAXARC + INDLK
          IF (IDENTA(INDSI) .EQ. 0) IDENTA(INDSI) = INDSI
          IDLNK = IDENTA(INDSI)
          IDENTA(INDEX) = IDLNK
  390     IF (GAMA(INDEX) .EQ. 0 .AND. ARCL(INDEX) .EQ. ARCL(INDSI))
     1       GO TO 400
          IWARN = 1
          GAMA(INDEX) = 0
  400     IF (LARCD(INDSI) .EQ. 0) GO TO 410
C-----TERNARY LINK WITH DOUBLE JOINT
          LARCD(INDEX) = IDLNK
          INDPR = (IDLNK - 1) / MAXARC
          LARCF(INDEX) = INDPR * MAXARC + LOOPL(INDPR + 1)
          GO TO 460
C-----BINARY LINK WITH DOUBLE JOINT
  410     INDPR = (INDLP - 1) * MAXARC
          LARCD(INDEX) = INDPR + INDLK
          LARCF(INDEX) = INDPR + LOOPL(INDLP)
          GO TO 460
C-----TERNARY LINK
  420     INDPR = (INDLP - 1) * MAXARC
          NLINK = NLINK + 1
          NJOINT = NJOINT + 1
          INDSI = INDPR - INDLK
          LARCD(INDEX) = INDSI
          LARCF(INDEX) = INDPR + LOOPL(INDLP)
          IF (-INDLK .GT. LOOPL(INDLP)) GO TO 430
          IF (ARCL(INDEX) .NE. ARCL(INDSI)) GO TO 460
          IF (DABS(GAMA(INDEX) - 180.0D0) .GT. 1.D-12) GO TO 460
          NINDLK = -INDLK
          IWARN = 1
          IF (IDENTA(INDSI) .EQ. 0) IDENTA(INDSI) = INDSI
          IDENTA(INDEX) = IDENTA(INDSI)
          NJOINT = NJOINT - 1
          NLINK = NLINK - 1
          GO TO 460
  430     INDARC = -INDLK
          IERR = 1
          GO TO 460
C-----PLANAR ROTARY INPUT
  440     IF (INDEX .EQ. 1) MOTOR = 0
  450     LARCF(INDEX) = -1
          LARCD(INDEX) = INDLK
          NJOINT = NJOINT + 1
          NLINK = NLINK + 2
  460   CONTINUE
C-----CHECK UNDETERMINED LINKS
        L1U = LARCX(IUNKNO - 1)
        L2U = LARCX(IUNKNO)
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
              IF (L2U - 1 .EQ. L1U) GO TO 470
              IF (L1U - 1 .EQ. L2U) GO TO 470
              IERR = 1
      470     IND2 = I + I
              IERR = 1
      480     CONTINUE
              IF (LOOPL(I) .EQ. 0) LOOPL(I) = MAXARC
              JST = IND + 1
              JFN = IND + LOOPL(I)
              DO 490 J = JST, JFN
                IF (LARCF(J) .EQ. - 1) LARCF(J) = JFN
      490     CONTINUE
      500 CONTINUE
C
C     IDENTIFY ALL SLIDING,PRISMATIC LINKS BY NEGATIVE ENTRY IN ILOOP
C     TO PICK UP THOSE LINKS IN ERROR,PLACE 'SOURCE' ARCS IN ILINK
C
      510 DO 520 I = 1, KKL
              IF (ILOOP(I) .GT. 0) ILOOP(I) = 0
      520 CONTINUE
          DO 540 I = 1, KKL
              IDLNK = IDENTA(I)
              IF (IDLNK .GT. 0) GO TO 530
              ILINK(I) = I
              GO TO 540
      530     ILINK(I) = IDLNK
              ILOOP(I) = ILOOP(IDLNK)
      540 CONTINUE
          RETURN
C
          END
C
C * * * * * * * * * * * * * * * * * * * * * * *
C
          SUBROUTINE SOLVE
C         ----------------
C         CALCULATE KINEMATICS
C
          IMPLICIT REAL*8(A - H,O - Z)
          DIMENSION THETA(100), THD(100), TH2D(100), ARCLD(100),
         1          ARCL2D(100), LAF(3), LAN(7), ILHD(10), ITHD(10),
         2          MVIOL(5), LAG(3), LAH(3)
          COMMON ARCL(100), AARCL(100), GAMA(100), GGAMA(100), DEGINC,
         1       IKON(361), ILINK(100), ILOOP(100), ITITLE(18), IPLOT, KKL,
         2       KKM, KKN, KKO, MAXARC, NLOOP, NFORM, NPOINT, MOTOR, IANG,
         3       MOTION, NDES, KON, IILINK(100), IILOOP(100), KLINK(100)
          COMMON /APART/ IJNT, J, M4, M5, NPT
          COMMON /XYZNUR/ XGR(37), YGR(37), SGR(37), AGR(37), RGR(37),
         1       TGR(37)
          COMMON /REST/ CONFIG(10), DELT(361), SLIDE(10), CRNKI, HPI, PI,
         1       DEGRAD, RINC, TWOSTA, IJOINT(10), LARCD(100), LARCF(100),
         2       LARCX(20), LOOPL(10), KOL, KOLL(5), KTHETA, LBA, LKFIN,
         3       LKOUT, LKR, LKRET, MOVER, NFROUT, NKOL, NLIN, NOUTLK,
         4       NOUTLP, IDEL, IDELV, IDELA, IVEL, IACC, LCOL(5), TANGLE
          EQUIVALENCE (MVIOL(1),IJNT)
          ITANG = 0
          IK = 1
          KOLIN = KOLL(1)
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
          LCOLN = LCOL(1)
          DO 10 I = 1, KKL
       10 THETA(I) = 0
          OUTLK = ARCL(KKM)
          ORIGLK = ARCL(KKN)
          OUTANG = GAMA(KKM)
          ORIGNG = GAMA(KKN)
          AFR = ARCL(NFROUT)
          GFR = GAMA(NFROUT)
          GFRPI = GFR - PI
          XFR = AFR * DCOS(GFR) + ORIGLK * DCOS(ORIGNG)
          YFR = AFR * DSIN(GFR) + ORIGLK * DSIN(ORIGNG)
          OUTRAD = OUTANG + GFR
          OUTDEG = OUTRAD / DEGRAD
C
C         THE KINEMATICS
C
          IFR = LOOPL(1)
          GREF = GAMA(IFR) / DEGRAD
          IF (MOTOR) 30, 60, 20
C-----SLIDING LINK INPUT
       20 THET1 = GAMA(KKO) * DEGRAD
          ARCL1 = ARCL(KKO)
          ASSIGN 200 TO LB150
          DEGINC = 1.0
          GO TO 90
C-----SPATIAL ROTARY INPUT
       30 THET1 = HPI - GAMA(IFR)
          ARCL1 = ARCL(1)
          ARCL2 = ARCL(2)
          OFFSET = ARCL(KKO + 1)
          CRANK1 = GAMA(KKO)
          IF (IDEL .NE. 0) GO TO 40
          ASSIGN 120 TO LB150
          ASSIGN 180 TO LB160
          GO TO 50
       40 DEGINC = 1.0
          ASSIGN 130 TO LB150
       50 ASSIGN 180 TO LB160
          GO TO 90
C-----PLANAR ROTARY INPUT
       60 CRANK1 = GAMA(KKO) - GREF
          IF (IDEL .NE. 0) GO TO 70
          ASSIGN 110 TO LB150
          GO TO 80
       70 DEGINC = 1.0
          ASSIGN 140 TO LB150
       80 ASSIGN 170 TO LB160
       90 CONTINUE
          ASSIGN 250 TO LB260
C
C         INDEX INPUT LINK TO SPECIFIED ANGLES OR LENGTHS
C
          DO 590 NPT = 1, NPOINT
            IF (NPT .LE. LCOLN) GO TO 100
            IK = IK + 1
            KOLIN = KOLL(IK)
            LCOLN = LCOL(IK)
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
      100   KON = (NPT - 1) * NKOL + 1 + KOLIN
            IKON(NPT) = KON
            GO TO LB150, (110, 120, 130, 140, 200)
C
C     ROTATING LINK INPUT
C
      110   POINT = NPT
            CRANK = CRANK1 + POINT * RINC
            GO TO 150
      120   POINT = NPT
            CRANKF = CRANK1 + POINT * RINC
            CRANK = CRANKF * DEGRAD
            GO TO 160
      130   CRANKF = CRANK1 + DELT(NPT)
            CRANK = CRANKF * DEGRAD
            GO TO 160
      140   CRANK = CRANK1 + DELT(NPT)
      150   CRANKF = CRANK + GREF
      160   IF (CRANKF .LT. 0) CRANKF = CRANKF + 360.0
            IF (CRANKF .GT. 360.0) CRANKF = CRANKF - 360.0
            GO TO LB160, (170, 180)
      170   THETA(MOVER) = CRANK * DEGRAD
            GO TO 210
C
C     SPATIAL ROTARY INPUT
C
      180   THETA(MOVER) = THET1
            SCRA = DSIN(CRANK)
            CCRA = DCOS(CRANK)
            ASCR1 = ARCL1 * SCRA
            ARCL(1) = ASCR1
            ACCR1 = ARCL1 * CCRA
            OFFS = OFFSET - ACCR1
            IF (ARCL2 .GT. DABS(OFFS)) GO TO 190
            IJNT = 1
            J = 1
            M4 = 2
            M5 = 3
            GO TO 550
      190   ARCLT = DSQRT(ARCL2*ARCL2 - OFFS*OFFS)
            ARCL(2) = ARCLT
            GO TO 210
C
C     SLIDING LINK INPUT
C
      200   ARC = ARCL1 + DELT(NPT)
            ARCL(MOVER) = ARC
      210   ASSIGN 220 TO LB220
C
C     ANALYZE EACH LOOP IN TURN
C
            DO 480 J = 1, NLOOP
              M1 = (J - 1) * MAXARC
              M11 = M1 + LOOPL(J)
              MU = J + J
              M2 = LARCX(MU - 1)
              M3 = LARCX(MU)
              M4 = M1 + M2
```

```
                M5 = M1 + M3
                THETA(M11) = PI
                GM11 = GAMA(M11)
                C1 = -ARCL(M11)
                C2 = 0
                L = LOOPL(J) - 1
C
C      SUMMATE THE DETERMINED ARCS FOR THE LOOP
C
                DO 250 I = 1, L
                  IF (I .EQ. M2 .OR. I .EQ. M3) GO TO 250
                  M1I = M1 + I
                  GO TO LB220, (220, 230)
  220             ASSIGN 230 TO LB220
                  I1 = I
                  I3 = I
                  GO TO 240
  230             I2 = LARCF(M1I)
                  I3 = LARCD(M1I)
                  THETA(M1I) = THETA(I3) + GAMA(M1I) - (GM11 - GAMA(I2))
                  I1 = ILINK(M1I)
                  ARCL(M1I) = ARCL(I1)
  240             ALM1I = ARCL(M1I)
                  THM1I = THETA(M1I)
                  CTHM = DCOS(THM1I)
                  STHM = DSIN(THM1I)
                  C1I = ALM1I * CTHM
                  C2I = ALM1I * STHM
                  C1 = C1 + C1I
                  C2 = C2 + C2I
  250           CONTINUE
C
C      CALCULATE THE UNDETERMINED DYAD
C
                IJNT = IJOINT(J)
                GO TO (420, 260, 280, 290, 310, 350), IJNT
C
C      TWO PRISMATIC JOINTS,ONE REVOLUTE JOINT
C
C-----TWO SLIDERS ON KNOWN LINKS
  260           I1 = LARCD(M4)
                THM4 = THETA(I1) - SLIDE(J)
                I2 = LARCD(M5)
                THM5 = THETA(I2) + CONFIG(J)
                SINDTH = DSIN(THM4 - THM5)
                IF (DABS(SINDTH) .GT. 1.0D-12) GO TO 270
                NCOUNT = IJNT
                GO TO 550
  270           THETA(M4) = THM4
                THETA(M5) = THM5
                STHM5 = DSIN(THM5)
                CTHM5 = DCOS(THM5)
                STHM4 = DSIN(THM4)
                CTHM4 = DCOS(THM4)
                CSPM4 = CTHM4 + STHM4
                CSPM5 = CTHM5 + STHM5
                ST21 = -SINDTH
                CCCS = C2 * CTHM5 - C1 * STHM5
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
            ALM4 = CCCS / ST21
            ALM5 = -(C1 + C2 + ALM4*CSPM4) / CSPM5
            ARCL(M4) = ALM4
            ARCL(M5) = ALM5
            GO TO 480
C-----TWO SLIDERS ON FOLLOWING UNDETERMINED AND DETERMINED ARCS
  280       I2 = LARCD(M5)
            THM5 = THETA(I2) + CONFIG(J)
            THM4 = THM5 + SLIDE(J)
            SINDTH = DSIGN(1.0D0,SLIDE(J))
            GO TO 300
C-----TWO SLIDERS ON PRECEDING DETERMINED AND UNDETERMINED ARCS
  290       I2 = LARCD(M4)
            THM4 = THETA(I2) - SLIDE(J)
            THM5 = THM4 - CONFIG(J)
            SINDTH = DSIGN(1.0D0,CONFIG(J))
  300       THETA(M4) = THM4
            THETA(M5) = THM5
            STHM5 = DSIN(THM5)
            CTHM5 = DCOS(THM5)
            ARCL(M4) = -SINDTH * (C2*CTHM5 - C1*STHM5)
            ARCL(M5) = -(C1*CTHM5 + C2*STHM5)
            GO TO 480
C
C     ONE PRISMATIC JOINT,TWO REVOLUTE JOINTS
C
C-----SLIDER ON DETERMINED ARC
  310       ARCL5 = ARCL(M5)
            I1 = LARCD(M4)
            THM4 = THETA(I1) - SLIDE(J)
            THETA(M4) = THM4
            STHM4 = DSIN(THM4)
            CTHM4 = DCOS(THM4)
            C3 = C1 * STHM4 - C2 * CTHM4
            C4 = ARCL5 * STHM4
            C5 = -ARCL5 * CTHM4
            IF (DABS(STHM4) .GT. 0.001) GO TO 320
            SINDTH = C2 / ARCL5
            SINDTH = DABS(SINDTH)
            IF (SINDTH .GT. 1.0) GO TO 320
            THM5 = THM4 - DSIGN(HPI,C2) - CONFIG(J) * (DARCOS(SINDTH))
            THM5 = (THM5 - PI - PI) * CTHM4
            STHM5 = DSIN(THM5)
            CTHM5 = DCOS(THM5)
            THETA(M5) = THM5
            GO TO 410
  320       ARG = (ARCL5 + C3) * (ARCL5 - C3)
            CFS = ARCL5 - DABS(C3)
            C6 = ARCL5 * ARCL5
            IF (CFS) 340, 330, 380
C-----FIXED ARC ON LIMIT
  330       ALN2 = C1 * CTHM4 + C2 * STHM4
            ARCL(M4) = ALN2
            CSPM4 = CTHM4 + STHM4
            SINM5 = C3 * CTHM4 / ARCL5
            COSM5 = -(C3 + C5*SINM5) / C4
            CSPM5 = CTHM5 + STHM5
            THETA(M5) = DATAN2(SINM5,COSM5)
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
              GO TO 480
C-----FIXED LENGTH ARC TOO SHORT
  340         SHORT = -CFS
              GO TO 550
C-----SLIDER ON UNDETERMINED ARC
  350         ARCL5 = ARCL(M5)
              CFS = C1 * C1 + C2 * C2
              PSEUDO = DSQRT(CFS)
              IF (DABS(C1) .GT. 0.001) GO TO 360
              COSDTH = ARCL5 / PSEUDO
              IF (DABS(COSDTH) .GT. 1.0) GO TO 360
              THM5 = DATAN2(-C2,-C1) - CONFIG(J) * DARCOS(COSDTH)
              STHM5 = DSIN(THM5)
              CTHM5 = DCOS(THM5)
              THETA(M5) = THM5
              GO TO 410
  360         C4 = C1
              C5 = C2
              C6 = CFS
              C3 = ARCL5
              ARG = CFS - C3 * C3
              ARGS = PSEUDO - C3
              IF (ARGS) 550, 370, 380
C-----GUIDE ON LIMIT
  370         ARCL(M4) = 0
              C1 = -C1
              THM5 = -DATAN2(C2,C1)
              STHM5 = DSIN(THM5)
              CTHM5 = DCOS(THM5)
              GO TO 390
C-----GUIDE TOO LONG
  380         STHM5 = (-C3*C5 + CONFIG(J)*C4*DSQRT(ARG)) / C6
              CTHM5 = -(C3 + C5*STHM5) / C4
              THM5 = DATAN2(STHM5,CTHM5)
  390         THETA(M5) = THM5
  400         IF (IJNT .EQ. 5) GO TO 410
              THM4 = THETA(M5) + SLIDE(J)
              STHM4 = DSIN(THM4)
              CTHM4 = DCOS(THM4)
              THETA(M4) = THM4
  410         CONTINUE
              CSPM5 = CTHM5 + STHM5
              CSPM4 = CTHM4 + STHM4
              ALN2 = -(C1 + C2 + ARCL5*CSPM5) / CSPM4
              ARCL(M4) = ALN2
              GO TO 480
C
C     NO PRISMATIC JOINTS,THREE REVOLUTE JOINTS
C
  420         ARCL4 = ARCL(M4)
              ARCL5 = ARCL(M5)
              CFS = C1 * C1 + C2 * C2
              C3 = CFS + (ARCL5 + ARCL4) * (ARCL5 - ARCL4)
              TARCL5 = 2.0 * ARCL5
              C4 = TARCL5 * C1
              C5 = TARCL5 * C2
              IF (DABS(C1) .GT. 0.00001) GO TO 430
              COSDTH = C3 / (TARCL5*DSQRT(CFS))
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
                IF (DABS(COSDTH) .LE. 1.0) GO TO 450
  430           C52S = C5 * C5
                C6 = C4 * C4 + C52S
                ARG = (C4 + C3) * (C4 - C3) + C52S
                TRYA = TWOSTA * ARCL4 * ARCL5
                TRYAS = TRYA * TRYA
                IF (ARG .GE. TRYAS) GO TO 460
                IF (ARG .GE. 0.00001) GO TO 440
                FLK = DSQRT(CFS)
                GO TO 550
  440           TRYA = DSQRT(ARG)
                STANG = TRYA / ARCL4 / TARCL5
                ATANG = (HPI - DARSIN(STANG)) / DEGRAD
                IF (ITANG .EQ. 1) GO TO 460
                ITANG = 1
C-----FREE ENDS OF DETERMINED ARCS LIE IN A VERTICAL LINE
  450           THM5 = DATAN2(-C2,-C1) - CONFIG(J) * DARCOS(COSDTH)
                SINM5 = DSIN(THM5)
                COSM5 = DCOS(THM5)
                THETA(M5) = THM5
                GO TO 470
  460           SINM5 = (-C3*C5 + CONFIG(J)*C4*DSQRT(ARG)) / C6
                COSM5 = -(C3 + C5*SINM5) / C4
                THETA(M5) = DATAN2(SINM5,COSM5)
  470           SINM4 = -(C2 + ARCL5*SINM5) / ARCL4
                COSM4 = -(C1 + ARCL5*COSM5) / ARCL4
                THETA(M4) = DATAN2(SINM4,COSM4)
  480         CONTINUE
C
C     CALCULATE RESULTS
C
              GO TO (490, 510, 500), LBA
C
C     ANGULAR GENERATION
C
  490         ANGDEG = THETA(LKOUT) / DEGRAD + OUTDEG
              AGR(NPT) = ANGDEG * DEGRAD
              GO TO 570
C
C     SLIDING LINK OUTPUT
C
  500         SGR(NPT) = ARCL(LKOUT)
              GO TO 570
C
C     FUNCTION PATH GENERATION
C
  510         THOUT = THETA(LKOUT) + OUTRAD
              X = XFR
              Y = YFR
              XD = 0
              YD = 0
              X2D = 0
              Y2D = 0
              IF (LKFIN .EQ. LKR) GO TO 530
C
C     RETRACE OUTPUT LOOP BACK TO OUTPUT LINK
C
              DO 520 I = LKRET, LKFIN
```

```
               ALINT = ARCL(I)
               THINT = THETA(I) + GFRPI
               CTHI = DCOS(THINT)
               STHI = DSIN(THINT)
               XI = ALINT * CTHI
               YI = ALINT * STHI
               X = X + XI
               Y = Y + YI
  520        CONTINUE
C
C          ADD ON CONTRIBUTION OF OUTPUT LINK
C
  530        CTHO = DCOS(THOUT)
             STHO = DSIN(THOUT)
             XI = OUTLK * CTHO
             YI = OUTLK * STHO
             X = X + XI
             Y = Y + YI
             IF (IANG .EQ. 2) GO TO 540
             XGR(NPT) = X
             YGR(NPT) = Y
             GO TO 570
  540        R = DSQRT(X*X + Y*Y)
             T = 0
             IF (R .GT. 1.0D-12) T = (DSIGN((DARSIN(Y/R) - HPI),X) + HPI)
             RGR(NPT) = R
             TGR(NPT) = T
             GO TO 570
C
C          CONSTRAINTS OF LOOP CLOSURE VIOLATED
C
  550        KOLE = KOL + IKON(NPT) - KON
             DO 560 I = 1, KOLE
               J = KON + I - 1
  560        CONTINUE
             GO TO 590
C
C          CHECK ON LENGTH AND ANGULAR POSITION OF EACH ARC
C
  570        DO 580 I = 1, NLOOP
               LKEND = (I - 1) * MAXARC
               M = LOOPL(I)
               LKM = LKEND + M
               FRANG = GAMA(LKM)
               DO 580 J = 1, M
                 L = LKEND + J
                 THETA(L) = (THETA(L) + FRANG) / DEGRAD
  580        CONTINUE
  590    CONTINUE
           IF (MOTOR .GE. 0) RETURN
           ARCL(1) = ARCL1
           ARCL(2) = ARCL2
           RETURN
C
           END
C
C    *    *    *    *    *    *    *    *    *    *    *    *    *    *    *    *    *
C
```

```
C --ROUTINE TO CHECK NUMERICAL DIFFERENCIATION STEP AGAINST TOLERANCE
C
      SUBROUTINE CHECK(TLA, TLG, TOLER, TOLEG, TOLED, DIFA, DIFG,
     1           DEGRAD, KKO, ICHECK, MRNUR, IRNUR, MGNUR, IGNUR, MOTN)
C
      IMPLICIT REAL*8(A - H,O - Z)
      DIMENSION TLA(KKO), TLG(KKO), TOLER(KKO), TOLEG(KKO), TOLED(KKO),
     1          DIFA(KKO), DIFG(KKO), MRNUR(IRNUR), MGNUR(103)
C
      KKN = KKO - 1
      KKM = KKN - 1
C
      DO 10 J = 1, IRNUR
        KL = MRNUR(J)
        IF (DIFA(KL) .GT. TLA(KL)*5.0D-2) GO TO 60
   10 CONTINUE
      IF (IGNUR .EQ. 0) GO TO 30
      DO 20 J = 1, IGNUR
        KL = MGNUR(J)
        IF (DIFG(KL) .GT. TLG(KL)*5.0D-2) GO TO 60
   20 CONTINUE
   30 IF (MOTN .EQ. 2 .OR. MOTN .EQ. 4) GO TO 40
      IF (DIFA(KKM) .GT. TLA(KKM)*5.0D-2) GO TO 60
      IF (DIFA(KKN) .GT. TLA(KKN)*5.0D-2) GO TO 60
      IF (DIFG(KKN) .GT. TLG(KKN)*5.0D-2) GO TO 60
   40 IF (MOTN .EQ. 4) GO TO 50
      IF (DIFG(KKM) .GT. TLG(KKM)*5.0D-2) GO TO 60
   50 IF (DIFG(KKO) .GT. TLG(KKO)*5.0D-2) GO TO 60
      ICHECK = 0
      GO TO 120
C
   60 DO 70 J = 1, IRNUR
        KL = MRNUR(J)
        TOLER(KL) = TLA(KL)
   70 CONTINUE
      IF (IGNUR .EQ. 0) GO TO 90
      DO 80 J = 1, IGNUR
        KL = MGNUR(J)
        TOLED(KL) = TLG(KL)
        TOLEG(KL) = TLG(KL) / DEGRAD
   80 CONTINUE
   90 IF (MOTN .EQ. 2 .OR. MOTN .EQ. 4) GO TO 100
      TOLER(KKM) = TLA(KKM)
      TOLER(KKN) = TLA(KKN)
      TOLED(KKN) = TLG(KKN)
      TOLEG(KKN) = TLG(KKN) / DEGRAD
  100 IF (MOTN .EQ. 4) GO TO 110
      TOLED(KKM) = TLG(KKM)
      TOLEG(KKM) = TLG(KKM) / DEGRAD
  110 TOLED(KKO) = TLG(KKO)
      TOLEG(KKO) = TLG(KKO) / DEGRAD
      ICHECK = 1
C
  120 RETURN
      END
C
C   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
C
```

```
C  --ROUTINE TO IDENTIFY IDENTICAL COMMON ARCS & THEIR ALLOWABLE
C DEVIATIONS
C
      SUBROUTINE COMARC(IREP, TLA, KKL)
C
      IMPLICIT REAL*8(A - H,O - Z)
C
      DIMENSION IREP(KKL), MK(100), ML(10), MJ(10,10), NC(10), TLA(KKL)
C
      NR = 0
      JL = 0
      DO 20 L = 1, KKL
        IF (IREP(L) .EQ. 0) GO TO 20
        IF (IREP(L) .NE. L) GO TO 10
        NR = NR + 1
        ML(NR) = L
   10   JL = JL + 1
        MK(JL) = L
   20 CONTINUE
C
      IF (NR .EQ. 0) GO TO 80
C
      DO 40 KL = 1, NR
        NC(KL) = 0
        DO 30 NL = 1, JL
          IF (IREP(MK(NL)) .NE. IREP(ML(KL))) GO TO 30
          NC(KL) = NC(KL) + 1
          MJ(KL,NC(KL)) = MK(NL)
   30   CONTINUE
   40 CONTINUE
C
      DO 70 KL = 1, NR
        SMALL = 1.0D+03
        MC = NC(KL)
        DO 50 NL = 1, MC
          IF (TLA(MJ(KL,NL)) .LT. SMALL) SMALL = TLA(MJ(KL,NL))
   50   CONTINUE
        DO 60 NL = 1, MC
          TLA(MJ(KL,NL)) = SMALL
   60   CONTINUE
   70 CONTINUE
C
   80 RETURN
      END
C
C  *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
C
C --ROUTINE FOR REARRANGING ARC DEVIATIONS IN ASCENDING ORDER
C
      SUBROUTINE ORDER(TLA, MN, M)
      IMPLICIT REAL*8(A - H,O - Z)
C
C
      DIMENSION TLA(M), MN(M)
C
      DO 10 I = 1, M
        MN(I) = I
   10 CONTINUE
```

```
C
      DO 30 K = 2, M
        DO 20 L = 2, K
          I = K - L + 1
          J = I + 1
          IF (TLA(MN(J)) .GE. TLA(MN(I))) GO TO 30
          MT = MN(J)
          MN(J) = MN(I)
          MN(I) = MT
   20   CONTINUE
   30 CONTINUE
C
      RETURN
      END
C
C  *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
C
C -- ROUTINE FOR MODIFYING DEVIATIONS BY MULTIPLYING BY RATIO OF
C    ALLOWABLE/MAXIMUM OF OUTPUT DEVIATION
C
      SUBROUTINE CORFAC(NDES, KKL, KKM, KKN, KKO)
C
      IMPLICIT REAL*8(A - H,O - Z)
C
      COMMON /CORR/ OUTOLX, OUTOLY, OUTOLA, OUTOLR, OUTOLT, OUTOLS,
     1       PDX(37,100), PDY(37,100), PDGX(37,100), PDGY(37,100),
     2       PDA(37,100), PDGA(37,100), PDR(37,100), PDT(37,100),
     3       PDGR(37,100), PDGT(37,100), PDS(37,100), PDGS(37,100),
     4       PDOLX(37), PDOLY(37), PDOGX(37), PDOGY(37), PDORLX(37),
     5       PDORLY(37), PDORGX(37), PDORGY(37), PDCRX(37), PDCRY(37),
     6       PDOLR(37), PDOLT(37), PDOGR(37), PDOGT(37), PDORLR(37),
     7       PDORLT(37), PDORGR(37), PDORGT(37), PDCRR(37), PDCRT(37),
     8       PDOGA(37), PDCRA(37), PDCRS(37), TLA(105), TLG(105),
     9       DEVX(37), DEVY(37), DEVA(37), DEVR(37), DEVT(37), DEVS(37),
     *       FACTOR, MOTN
C
      DEGRAD = DATAN(1.0D+0) / 4.5D+1
C
      DO 210 J = 1, NDES
        GO TO (10, 60, 110, 160), MOTN
C
   10   DEVX(J) = 0.0D+0
        DEVY(J) = 0.0D+0
        DO 50 I = 1, KKL
          IF (TLA(I) .GT. 9.9D+2) GO TO 20
          DEVX(J) = DEVX(J) + (PDX(J,I)*TLA(I)) ** 2
          DEVY(J) = DEVY(J) + (PDY(J,I)*TLA(I)) ** 2
          GO TO 30
   20     PDX(J,I) = -1.0D+0
          PDY(J,I) = -1.0D+0
   30     IF (TLG(I) .GT. 9.9D+2) GO TO 40
          DEVX(J) = DEVX(J) + (PDGX(J,I)*TLG(I)) ** 2
          DEVY(J) = DEVY(J) + (PDGY(J,I)*TLG(I)) ** 2
          GO TO 50
   40     PDGX(J,I) = -1.0D+0
          PDGY(J,I) = -1.0D+0
   50   CONTINUE
        IF (TLA(KKM) .GT. 9.9D+2) GO TO 52
```

```
      DEVX(J) = DEVX(J) + (PDOLX(J)*TLA(KKM)) ** 2 + (PDOGX(J)*TLG(
     1 KKM)) ** 2
      DEVY(J) = DEVY(J) + (PDOLY(J)*TLA(KKM)) ** 2 + (PDOGY(J)*TLG(
     1 KKM)) ** 2
      GO TO 54
   52 PDOLX(J) = -1.0D+0
      PDOLY(J) = -1.0D+0
      PDOGX(J) = -1.0D+0
      PDOGY(J) = -1.0D+0
   54 IF (TLA(KKN) .GT. 9.9D+2) GO TO 56
      DEVX(J) = DEVX(J) + (PDORLX(J)*TLA(KKN)) ** 2 + (PDORGX(J)*
     1 TLG(KKN)) ** 2
      DEVY(J) = DEVY(J) + (PDORLY(J)*TLA(KKN)) ** 2 + (PDORGY(J)*
     1 TLG(KKN)) ** 2
      GO TO 58
   56 PDORLX(J) = -1.0D+0
      PDORLY(J) = -1.0D+0
      PDORGX(J) = -1.0D+0
      PDORGY(J) = -1.0D+0
   58 DEVX(J) = DEVX(J) + (PDCRX(J)*TLG(KKO)) ** 2
      DEVY(J) = DEVY(J) + (PDCRY(J)*TLG(KKO)) ** 2
C
      DEVX(J) = DSQRT(DEVX(J))
      DEVY(J) = DSQRT(DEVY(J))
C
      GO TO 210
C
   60 DEVA(J) = 0.0D+0
      DO 100 I = 1, KKL
        IF (TLA(I) .GT. 9.9D+2) GO TO 70
        DEVA(J) = DEVA(J) + (PDA(J,I)*TLA(I)) ** 2
        GO TO 80
   70   PDA(J,I) = -1.0D+0
   80   IF (TLG(I) .GT. 9.9D+2) GO TO 90
        DEVA(J) = DEVA(J) + (PDGA(J,I)*TLG(I)) ** 2
        GO TO 100
   90   PDGA(J,I) = -1.0D+0
  100 CONTINUE
      DEVA(J) = DEVA(J) + (PDOGA(J)*TLG(KKM)) ** 2 + (PDCRA(J)*TLG(
     1 KKO)) ** 2
      DEVA(J) = DSQRT(DEVA(J))
C
      GO TO 210
C
  110 DEVR(J) = 0.0D+0
      DEVT(J) = 0.0D+0
      DO 150 I = 1, KKL
        IF (TLA(I) .GT. 9.9D+2) GO TO 120
        DEVR(J) = DEVR(J) + (PDR(J,I)*TLA(I)) ** 2
        DEVT(J) = DEVT(J) + (PDT(J,I)*TLA(I)) ** 2
        GO TO 130
  120   PDR(J,I) = -1.0D+0
        PDT(J,I) = -1.0D+0
  130   IF (TLG(I) .GT. 9.9D+2) GO TO 140
        DEVR(J) = DEVR(J) + (PDGR(J,I)*TLG(I)) ** 2
        DEVT(J) = DEVT(J) + (PDGT(J,I)*TLG(I)) ** 2
        GO TO 150
  140   PDGR(J,I) = -1.0D+0
```

```
               PDGT(J,I) = -1.0D+0
  150       CONTINUE
            IF (TLA(KKM) .GT. 9.9D+2) GO TO 152
            DEVR(J) = DEVR(J) + (PDOLR(J)*TLA(KKM)) ** 2 + (PDOGR(J)*TLG(
     1      KKM)) ** 2
            DEVT(J) = DEVT(J) + (PDOLT(J)*TLA(KKM)) ** 2 + (PDOGT(J)*TLG(
     1      KKM)) ** 2
            GO TO 154
  152       PDOLR(J) = -1.0D+0
            PDOLT(J) = -1.0D+0
            PDOGR(J) = -1.0D+0
            PDOGT(J) = -1.0D+0
  154       IF (TLA(KKN) .GT. 9.9D+2) GO TO 156
            DEVR(J) = DEVR(J) + (PDORLR(J)*TLA(KKN)) ** 2 + (PDORGR(J)*
     1      TLG(KKN)) ** 2
            DEVT(J) = DEVT(J) + (PDORLT(J)*TLA(KKN)) ** 2 + (PDORGT(J)*
     1      TLG(KKN)) ** 2
            GO TO 158
  156       PDORLR(J) = -1.0D+0
            PDORLT(J) = -1.0D+0
            PDORGR(J) = -1.0D+0
            PDORGT(J) = -1.0D+0
  158       DEVR(J) = DEVR(J) + (PDCRR(J)*TLG(KKO)) ** 2
            DEVT(J) = DEVT(J) + (PDCRT(J)*TLG(KKO)) ** 2
C
            DEVR(J) = DSQRT(DEVR(J))
            DEVT(J) = DSQRT(DEVT(J))
C
            GO TO 210
C
  160       DEVS(J) = 0.0D+0
            DO 200 I = 1, KKL
               IF (TLA(I) .GT. 9.9D+2) GO TO 170
               DEVS(J) = DEVS(J) + (PDS(J,I)*TLA(I)) ** 2
               GO TO 180
  170          PDS(J,I) = -1.0D+0
  180          IF (TLG(I) .GT. 9.9D+2) GO TO 190
               DEVS(J) = DEVS(J) + (PDGS(J,I)*TLG(I)) ** 2
               GO TO 200
  190          PDGS(J,I) = -1.0D+0
  200       CONTINUE
            DEVS(J) = DEVS(J) + (PDCRS(J)*TLG(KKO)) ** 2
            DEVS(J) = DSQRT(DEVS(J))
C
  210 CONTINUE
C
      GO TO (220, 240, 260, 280), MOTN
C
  220 CALL DEVMAX(DEVX, NDES, DMAX)
      FACTRX = OUTOLX / DMAX
      CALL DEVMAX(DEVY, NDES, DMAX)
      FACTRY = OUTOLY / DMAX
      FACTOR = FACTRX
      IF (FACTRY .LT. FACTRX) FACTOR = FACTRY
      DO 230 J = 1, NDES
         DEVX(J) = DEVX(J) * FACTOR
         DEVY(J) = DEVY(J) * FACTOR
  230 CONTINUE
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
C
      GO TO 300
C
  240 CALL DEVMAX(DEVA, NDES, DMAX)
      FACTOR = OUTOLA * DEGRAD / DMAX
      DO 250 J = 1, NDES
        DEVA(J) = DEVA(J) * FACTOR / DEGRAD
  250 CONTINUE
C
      GO TO 300
C
  260 CALL DEVMAX(DEVR, NDES, DMAX)
      FACTRR = OUTOLR / DMAX
      CALL DEVMAX(DEVT, NDES, DMAX)
      FACTRT = OUTOLT * DEGRAD / DMAX
      FACTOR = FACTRR
      IF (FACTRT .LT. FACTRR) FACTOR = FACTRT
      DO 270 J = 1, NDES
        DEVR(J) = DEVR(J) * FACTOR
        DEVT(J) = DEVT(J) * FACTOR / DEGRAD
  270 CONTINUE
C
      GO TO 300
C
  280 CALL DEVMAX(DEVS, NDES, DMAX)
      FACTOR = OUTOLS / DMAX
      DO 290 J = 1, NDES
        DEVS(J) = DEVS(J) * FACTOR
  290 CONTINUE
C
  300 DO 310 I = 1, KKO
        IF (TLA(I) .GT. 9.9D+2) GO TO 310
        TLA(I) = TLA(I) * FACTOR
        IF (TLG(I) .GT. 9.9D+2) GO TO 310
        TLG(I) = TLG(I) * FACTOR
  310 CONTINUE
C
      RETURN
      END
C
C   *    *    *    *    *    *    *    *    *    *    *    *    *    *    *    *    *
C
C -- ROUTINE FOR IDENTIFYING MAXIMUM DEVIATION OF OUTPUT
C
      SUBROUTINE DEVMAX(ARRAY, N, DMAX)
C
      IMPLICIT REAL*8(A - H,O - Z)
      DIMENSION ARRAY(N)
C
      DMAX = ARRAY(1)
      DO 10 I = 2, N
        IF (ARRAY(I) .GT. DMAX) DMAX = ARRAY(I)
   10 CONTINUE
C
      RETURN
      END
C
C   *    *    *    *    *    *    *    *    *    *    *    *    *    *    *    *    *
```

```
      C
      C--ROUTINE TO ALLOCATE ARC LENGTH & ANGLE TOLERANCES & JOINT CLEARANCES
      C
            SUBROUTINE RESULT(IREP, ITERN, KKL, KKM, NLOOP, MAXARC, LORDER,
           1           TLA, KKN, IGNUR, TLG, KKO, NOUTLK, NOUTLP, KLINK, ARCL,
           2           GAMA, MOTN, CLIMIT)
      C
            IMPLICIT REAL*8(A - H,O - Z)
      C
            DIMENSION IREP(KKL), ITERN(KKL), LORDER(KKN), TLA(KKN), TLG(KKO),
           1          KLINK(KKL), MAXR(10), TLNCE(10,10), TLANG(10,10),
           2          CLNCE(10,10,10), ARCL(KKN), GAMA(KKO), LTERN(105),
           3          ICL(10,10,10)
      C
            RADEG = 4.5D+1 / DATAN(1.0D+0)
      C
            DO 30 JL = 1, NLOOP
              IF (MAXARC .EQ. 4) GO TO 20
              DO 10 JA = 5, MAXARC
                LN = (JL - 1) * MAXARC + JA
                IF (KLINK(LN) .NE. 0) GO TO 10
                MAXR(JL) = JA - 1
                GO TO 30
         10   CONTINUE
         20   MAXR(JL) = MAXARC
         30 CONTINUE
      C
            DO 40 JL = 1, NLOOP
              NIM = MAXR(JL)
              DO 40 JA = 1, NIM
                JM = JA - 1
                IF (JM .EQ. 0) JM = MAXR(JL)
                CLNCE(JM,JA,JL) = -1.0D+0
                ICL(JM,JA,JL) = 1
                TLNCE(JA,JL) = -1.0D+0
         40 CONTINUE
      C
            NNN = KKN
            IF (MOTN .EQ. 2 .OR. MOTN .EQ. 4) NNN = KKL
      C
            DO 720 IT = 1, NNN
              IORD = LORDER(IT)
              IF (TLA(IORD) .GT. 9.9D+02) GO TO 730
              IF (IORD .EQ. KKM) GO TO 640
              IF (IORD .EQ. KKN) GO TO 690
      C
              CALL JSBSCR(IORD, MAXARC, LPNUM, IRNUM, LP, LF, MAXR, NLOOP)
      C
              IF (IREP(IORD) .EQ. 0 .OR. IREP(IORD) .EQ. IORD) GO TO 50
              JORD = IREP(IORD)
              CALL JSBSCR(JORD, MAXARC, NOLP, NOAR, LPI, LFI, MAXR, NLOOP)
              JPF = (LPNUM - 1) * MAXARC + LP
              JFF = (LPNUM - 1) * MAXARC + LF
              JPS = (NOLP - 1) * MAXARC + LPI
              JFS = (NOLP - 1) * MAXARC + LFI
              TLNCE(IRNUM,LPNUM) = TLNCE(NOAR,NOLP)
              CLNCE(LP,IRNUM,LPNUM) = CLNCE(LPI,NOAR,NOLP)
              IF ((IREP(JPF) .EQ. IREP(JPS) .OR. IREP(JPF) .EQ. IREP(JFS)) .
```

```
     1  AND. IREP(JPF) .NE. 0) ICL(LP,IRNUM,LPNUM) = 0
        CLNCE(IRNUM,LF,LPNUM) = CLNCE(NOAR,LFI,NOLP)
        IF ((IREP(JFF) .EQ. IREP(JFS) .OR. IREP(JFF) .EQ. IREP(JPS)) .
     1  AND. IREP(JFF) .NE. 0) ICL(IRNUM,LF,LPNUM) = 0
        GO TO 90
C
   50   IF (CLNCE(LP,IRNUM,LPNUM) .LT. 0.0D+0 .AND. CLNCE(IRNUM,LF,
     1     LPNUM) .LT. 0.0D+0) GO TO 80
        IF (CLNCE(IRNUM,LF,LPNUM) .LT. 0.0D+0) GO TO 70
        IF (CLNCE(LP,IRNUM,LPNUM) .LT. 0.0D+0) GO TO 60
C
C .. CLEARANCES AT BOTH ENDS OF ARC ALREADY DETERMINED
        TLNCE(IRNUM,LPNUM) = TLA(IORD)
        NLT = 1
        GO TO 90
C
C .. CLEARANCE AT FOLLOWING JOINT ALREADY DETERMINED
   60   TLNCE(IRNUM,LPNUM) = TLA(IORD) / DSQRT(1.25D+0)
        CLNCE(LP,IRNUM,LPNUM) = 0.5D+0 * TLNCE(IRNUM,LPNUM)
        IF (ARCL(IORD) .LT. TLA(IORD)) CLNCE(LP,IRNUM,LPNUM) = TLA(IORD)
        NLT = 2
        IF (CLNCE(LP,IRNUM,LPNUM) .LE. CLIMIT) GO TO 90
        CLNCE(LP,IRNUM,LPNUM) = CLIMIT
        TLNCE(IRNUM,LPNUM) = DSQRT(TLA(IORD)**2 - CLIMIT**2)
        GO TO 90
C
C .. CLEARANCE AT PRECEEDING JOINT ALREADY DETERMINED
   70   TLNCE(IRNUM,LPNUM) = TLA(IORD) / DSQRT(1.25D+0)
        CLNCE(IRNUM,LF,LPNUM) = 0.5D+0 * TLNCE(IRNUM,LPNUM)
        IF (ARCL(IORD) .LT. TLA(IORD)) CLNCE(IRNUM,LF,LPNUM) = TLA(IORD)
        NLT = 3
        IF (CLNCE(IRNUM,LF,LPNUM) .LE. CLIMIT) GO TO 90
        CLNCE(IRNUM,LF,LPNUM) = CLIMIT
        TLNCE(IRNUM,LPNUM) = DSQRT(TLA(IORD)**2 - CLIMIT**2)
        GO TO 90
C
C .. CLEARANCES AT BOTH ENDS YET TO BE DETERMINED
   80   TLNCE(IRNUM,LPNUM) = TLA(IORD) / DSQRT(1.5D+0)
        CLNCE(LP,IRNUM,LPNUM) = 0.5D+0 * TLNCE(IRNUM,LPNUM)
        IF (ARCL(IORD) .LT. TLA(IORD)) CLNCE(LP,IRNUM,LPNUM) = TLA(IORD)
     1   / DSQRT(2.0D+0)
        CLNCE(IRNUM,LF,LPNUM) = CLNCE(LP,IRNUM,LPNUM)
        NLT = 4
        IF (CLNCE(LP,IRNUM,LPNUM) .LE. CLIMIT) GO TO 90
        CLNCE(LP,IRNUM,LPNUM) = CLIMIT
        CLNCE(IRNUM,LF,LPNUM) = CLIMIT
        TLNCE(IRNUM,LPNUM) = DSQRT(TLA(IORD)**2 - 2.0D+0*CLIMIT**2)
C
C .. IF LINK IS TERNARY REALLOCATE TOLERANCES AND CLEARANCES
   90   IF (ITERN(IORD) .EQ. 0) GO TO 720
        CALL TERNRY(ARCL, ITERN, IORD, LTERN, KKL, NLOOP, KXX)
C
        DO 100 KTI = 1, KXX
          KORD = LTERN(KTI)
          CALL JSBSCR(KORD, MAXARC, NOLP, NOAR, LPT, LFT, MAXR, NLOOP)
          IF (TLNCE(NOAR,NOLP) .GT. 0.0D+0) GO TO 720
  100   CONTINUE
C
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
            DO 630 KNN = 1, KXX
              KORD = LTERN(KNN)
              CALL JSBSCR(KORD, MAXARC, NOLP, NOAR, LPT, LFT, MAXR, NLOOP)
              NJP = (NOLP - 1) * MAXARC + LPT
              MJP = (LPNUM - 1) * MAXARC + LP
              NJF = (NOLP - 1) * MAXARC + LFT
              MJF = (LPNUM - 1) * MAXARC + LF
C
              IF ((IREP(MJP) .EQ. IREP(NJP) .AND. IREP(MJP) .NE. 0) .OR. (
     1           ITERN(MJP) .EQ. ITERN(NJP) .AND. ITERN(MJP) .NE. 0))
     2           GO TO 110
              IF ((IREP(MJF) .EQ. IREP(NJF) .AND. IREP(MJF) .NE. 0) .OR. (
     1           ITERN(MJF) .EQ. ITERN(NJF) .AND. ITERN(MJF) .NE. 0))
     2           GO TO 240
              IF ((IREP(MJP) .EQ. IREP(NJF) .AND. IREP(MJP) .NE. 0) .OR. (
     1           ITERN(MJP) .EQ. ITERN(NJF) .AND. ITERN(MJP) .NE. 0))
     2           GO TO 370
              GO TO 500
C
C .. PRECEEDING JOINTS OF BOTH ARCS COMMON
  110         IF (ITERN(NJP) .EQ. ITERN(KORD)) GO TO 140
              IF (IREP(NJP) .EQ. 0) GO TO 120
              IF (ITERN(IREP(NJP)) .EQ. ITERN(KORD)) GO TO 140
  120         IF (IREP(KORD) .EQ. 0) GO TO 130
              IF (ITERN(NJP) .EQ. ITERN(IREP(KORD))) GO TO 140
  130         CLNCE(LPT,NOAR,NOLP) = CLNCE(LP,IRNUM,LPNUM)
  140         IF (ITERN(KORD) .NE. KORD) ICL(LPT,NOAR,NOLP) = 0
              IF (ITERN(IORD) .NE. IORD) ICL(LP,IRNUM,LPNUM) = 0
              CLENG = 0.5D+0 * TLA(NJF) / DSQRT(1.25D+0)
              IF (KORD .LT. IORD) GO TO 150
              TLANG(NOAR,NOLP) = TLG(KORD) / DSQRT(1.5D+0)
              CLANG = 0.5D+0 * ARCL(IORD) * TLANG(NOAR,NOLP)
              GO TO 160
  150         TLANG(IRNUM,LPNUM) = TLG(IORD) / DSQRT(1.5D+0)
              CLANG = 0.5D+0 * ARCL(IORD) * TLANG(IRNUM,LPNUM)
  160         IF (CLANG .GT. CLNCE(IRNUM,LF,LPNUM)) GO TO 210
              IF (CLNCE(NOAR,LFT,NOLP) .GT. 0.0D+0) GO TO 170
              CLUNG = 0.5D+0 * TLA(KORD) / DSQRT(1.25D+0)
              CLING = CLANG * ARCL(KORD) / ARCL(IORD)
              IF (CLING .GT. CLUNG) GO TO 170
              CLNCE(IRNUM,LF,LPNUM) = CLANG
              IF (CLING .LT. CLENG .AND. CLING .LT. CLIMIT) CLNCE(NOAR,LFT,
     1        NOLP) = CLING
              GO TO 180
  170         CLANG = CLANG * DSQRT(1.2D+0)
              IF (KORD .GT. IORD) TLANG(NOAR,NOLP) = 2.0D+0 * CLANG / ARCL(
     1        IORD)
              IF (IORD .GT. KORD) TLANG(IRNUM,LPNUM) = 2.0D+0 * CLANG /
     1        ARCL(IORD)
              CLNCE(IRNUM,LF,LPNUM) = CLANG
  180         GO TO (630, 630, 190, 200), NLT
  190         TLNCE(IRNUM,LPNUM) = TLA(IORD)
              GO TO 630
  200         TLNCE(IRNUM,LPNUM) = TLA(IORD) / DSQRT(1.25D+0)
              CLNCE(LP,IRNUM,LPNUM) = 0.5D+0 * TLNCE(IRNUM,LPNUM)
              CLNCE(LPT,NOAR,NOLP) = CLNCE(LP,IRNUM,LPNUM)
              IF (CLNCE(LP,IRNUM,LPNUM) .LE. CLIMIT) GO TO 630
              CLNCE(LP,IRNUM,LPNUM) = CLIMIT
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
                CLNCE(LPT,NOAR,NOLP) = CLIMIT
                TLNCE(IRNUM,LPNUM) = DSQRT(TLA(IORD)**2 - CLIMIT**2)
                GO TO 630
   210          IF (CLNCE(NOAR,LFT,NOLP) .GT. 0.0D+0) GO TO 220
                CLING = CLANG * DSQRT(1.2D+0) * ARCL(KORD) / ARCL(IORD)
                CLUNG = 0.5D+0 * TLA(KORD) / DSQRT(1.25D+0)
                IF (CLUNG .GT. CLING) GO TO 230
   220          IF (KORD .GT. IORD) TLANG(NOAR,NOLP) = TLG(KORD)
                IF (IORD .GT. KORD) TLANG(IRNUM,LPNUM) = TLG(IORD)
                GO TO 630
   230          IF (CLING .LT. CLENG .AND. CLING .LT. CLIMIT) CLNCE(NOAR,LFT,
     1          NOLP) = CLING
                IF (KORD .GT. IORD) TLANG(NOAR,NOLP) = 2.0D+0 * CLING / ARCL(
     1          KORD)
                IF (IORD .GT. KORD) TLANG(IRNUM,LPNUM) = 2.0D+0 * CLING /
     1          ARCL(KORD)
                GO TO 630
C
C .. FOLLOWING JOINTS OF BOTH ARCS COMMON
   240          IF (ITERN(KORD) .EQ. ITERN(NJF)) GO TO 270
                IF (IREP(KORD) .EQ. 0) GO TO 250
                IF (ITERN(IREP(KORD)) .EQ. ITERN(NJF)) GO TO 270
   250          IF (IREP(NJF) .EQ. 0) GO TO 260
                IF (ITERN(KORD) .EQ. ITERN(IREP(NJF))) GO TO 270
   260          CLNCE(NOAR,LFT,NOLP) = CLNCE(IRNUM,LF,LPNUM)
   270          IF (ITERN(KORD) .NE. KORD) ICL(NOAR,LFT,NOLP) = 0
                IF (ITERN(IORD) .NE. IORD) ICL(IRNUM,LF,LPNUM) = 0
                CLENG = 0.5D+0 * TLA(NJP) / DSQRT(1.25D+0)
                IF (KORD .LT. IORD) GO TO 280
                TLANG(NOAR,NOLP) = TLG(KORD) / DSQRT(1.5D+0)
                CLANG = 0.5D+0 * ARCL(IORD) * TLANG(NOAR,NOLP)
                GO TO 290
   280          TLANG(IRNUM,LPNUM) = TLG(IORD) / DSQRT(1.5D+0)
                CLANG = 0.5D+0 * ARCL(IORD) * TLANG(IRNUM,LPNUM)
   290          IF (CLANG .GT. CLNCE(LP,IRNUM,LPNUM)) GO TO 340
                IF (CLNCE(LPT,NOAR,NOLP) .GT. 0.0D+0) GO TO 300
                CLUNG = 0.5D+0 * TLA(KORD) / DSQRT(1.25D+0)
                CLING = CLANG * ARCL(KORD) / ARCL(IORD)
                IF (CLING .GT. CLUNG) GO TO 300
                CLNCE(LP,IRNUM,LPNUM) = CLANG
                IF (CLING .LT. CLENG .AND. CLING .LT. CLIMIT) CLNCE(LPT,NOAR,
     1          NOLP) = CLING
                GO TO 310
   300          CLANG = CLANG * DSQRT(1.2D+0)
                IF (KORD .GT. IORD) TLANG(NOAR,NOLP) = 2.0D+0 * CLANG / ARCL(
     1          IORD)
                IF (IORD .GT. KORD) TLANG(IRNUM,LPNUM) = 2.0D+0 * CLANG /
     1          ARCL(IORD)
                CLNCE(LP,IRNUM,LPNUM) = CLANG
   310          GO TO (630, 630, 320, 330), NLT
   320          TLNCE(IRNUM,LPNUM) = TLA(IORD)
                GO TO 630
   330          TLNCE(IRNUM,LPNUM) = TLA(IORD) / DSQRT(1.25D+0)
                CLNCE(IRNUM,LF,LPNUM) = 0.5D+0 * TLNCE(IRNUM,LPNUM)
                CLNCE(NOAR,LFT,NOLP) = CLNCE(IRNUM,LF,LPNUM)
                IF (CLNCE(IRNUM,LF,LPNUM) .LE. CLIMIT) GO TO 630
                CLNCE(IRNUM,LF,LPNUM) = CLIMIT
                CLNCE(NOAR,LFT,NOLP) = CLIMIT
```

```
            TLNCE(IRNUM,LPNUM) = DSQRT(TLA(IORD)**2 - CLIMIT**2)
            GO TO 630
  340       IF (CLNCE(LPT,NOAR,NOLP) .GT. 0.0D+0) GO TO 350
            CLING = CLANG * DSQRT(1.2D+0) * ARCL(KORD) / ARCL(IORD)
            CLUNG = 0.5D+0 * TLA(KORD) / DSQRT(1.25D+0)
            IF (CLUNG .GT. CLING) GO TO 360
  350       IF (KORD .GT. IORD) TLANG(NOAR,NOLP) = TLG(KORD)
            IF (IORD .GT. KORD) TLANG(IRNUM,LPNUM) = TLG(IORD)
            GO TO 630
  360       IF (CLING .LT. CLENG .AND. CLING .LT. CLIMIT) CLNCE(LPT,NOAR,
     1      NOLP) = CLING
            IF (KORD .GT. IORD) TLANG(NOAR,NOLP) = 2.0D+0 * CLING / ARCL(
     1      KORD)
            IF (IORD .GT. KORD) TLANG(IRNUM,LPNUM) = 2.0D+0 * CLING /
     1      ARCL(KORD)          /
            GO TO 630
C
C .. PRECEEDING JOINT OF CURRENT ARC & FOLLOWING OF OTHER COMMON
  370       IF (ITERN(KORD) .EQ. ITERN(NJF)) GO TO 400
            IF (IREP(KORD) .EQ. 0) GO TO 380
            IF (ITERN(IREP(KORD)) .EQ. ITERN(NJF)) GO TO 400
  380       IF (IREP(NJF) .EQ. 0) GO TO 390
            IF (ITERN(KORD) .EQ. ITERN(IREP(NJF))) GO TO 400
  390       CLNCE(NOAR,LFT,NOLP) = CLNCE(LP,IRNUM,LPNUM)
  400       IF (ITERN(KORD) .NE. KORD) ICL(NOAR,LFT,NOLP) = 0
            IF (ITERN(IORD) .NE. IORD) ICL(LP,IRNUM,LPNUM) = 0
            CLENG = 0.5D+0 * TLA(NJP) / DSQRT(1.25D+0)
            IF (KORD .LT. IORD) GO TO 410
            TLANG(NOAR,NOLP) = TLG(KORD) / DSQRT(1.5D+0)
            CLANG = 0.5D+0 * ARCL(IORD) * TLANG(NOAR,NOLP)
            GO TO 420
  410       TLANG(IRNUM,LPNUM) = TLG(IORD) / DSQRT(1.5D+0)
            CLANG = 0.5D+0 * ARCL(IORD) * TLANG(IRNUM,LPNUM)
  420       IF (CLANG .GT. CLNCE(IRNUM,LF,LPNUM)) GO TO 470
            IF (CLNCE(LPT,NOAR,NOLP) .GT. 0.0D+0) GO TO 430
            CLUNG = 0.5D+0 * TLA(KORD) / DSQRT(1.25D+0)
            CLING = CLANG * ARCL(KORD) / ARCL(IORD)
            IF (CLING .GT. CLUNG) GO TO 430
            CLNCE(IRNUM,LF,LPNUM) = CLANG
            IF (CLING .LT. CLENG .AND. CLING .LT. CLIMIT) CLNCE(LPT,NOAR,
     1      NOLP) = CLING
            GO TO 440
  430       CLANG = CLANG * DSQRT(1.2D+0)
            IF (KORD .GT. IORD) TLANG(NOAR,NOLP) = 2.0D+0 * CLANG / ARCL(
     1      IORD)
            IF (IORD .GT. KORD) TLANG(IRNUM,LPNUM) = 2.0D+0 * CLANG /
     1      ARCL(IORD)
            CLNCE(IRNUM,LF,LPNUM) = CLANG
  440       GO TO (630, 630, 450, 460), NLT
  450       TLNCE(IRNUM,LPNUM) = TLA(IORD)
            GO TO 630
  460       TLNCE(IRNUM,LPNUM) = TLA(IORD) / DSQRT(1.25D+0)
            CLNCE(LP,IRNUM,LPNUM) = 0.5D+0 * TLNCE(IRNUM,LPNUM)
            CLNCE(NOAR,LFT,NOLP) = CLNCE(LP,IRNUM,LPNUM)
            IF (CLNCE(LP,IRNUM,LPNUM) .LE. CLIMIT) GO TO 630
            CLNCE(LP,IRNUM,LPNUM) = CLIMIT
            CLNCE(NOAR,LFT,NOLP) = CLIMIT
            TLNCE(IRNUM,LPNUM) = DSQRT(TLA(IORD)**2 - CLIMIT**2)
```

```
             GO TO 630
   470       IF (CLNCE(LPT,NOAR,NOLP) .GT. 0.0D+0) GO TO 480
             CLING = CLANG * DSQRT(1.2D+0) * ARCL(KORD) / ARCL(IORD)
             CLUNG = 0.5D+0 * TLA(KORD) / DSQRT(1.25D+0)
             IF (CLUNG .GT. CLING) GO TO 490
   480       IF (KORD .GT. IORD) TLANG(NOAR,NOLP) = TLG(KORD)
             IF (IORD .GT. KORD) TLANG(IRNUM,LPNUM) = TLG(IORD)
             GO TO 630
   490       IF (CLING .LT. CLENG .AND. CLING .LT. CLIMIT) CLNCE(LPT,NOAR,
     1       NOLP) = CLING
             IF (KORD .GT. IORD) TLANG(NOAR,NOLP) = 2.0D+0 * CLING / ARCL(
     1       KORD)
             IF (IORD .GT. KORD) TLANG(IRNUM,LPNUM) = 2.0D+0 * CLING /
     1       ARCL(KORD)
             GO TO 630
C
C .. FOLLOWING JOINT OF CURRENT & PRECEEDING OF OTHER COMMON
   500       IF (ITERN(NJP) .NE. ITERN(KORD)) GO TO 530
             IF (IREP(NJP) .EQ. 0) GO TO 510
             IF (ITERN(IREP(NJP)) .EQ. ITERN(KORD)) GO TO 530
   510       IF (IREP(KORD) .EQ. 0) GO TO 520
             IF (ITERN(NJP) .EQ. ITERN(IREP(KORD))) GO TO 530
   520       CLNCE(LPT,NOAR,NOLP) = CLNCE(IRNUM,LF,LPNUM)
   530       IF (ITERN(KORD) .NE. KORD) ICL(LPT,NOAR,NOLP) = 0
             IF (ITERN(IORD) .NE. IORD) ICL(IRNUM,LF,LPNUM) = 0
             CLENG = 0.5D+0 * TLA(NJF) / DSQRT(1.25D+0)
             IF (KORD .LT. IORD) GO TO 540
             TLANG(NOAR,NOLP) = TLG(KORD) / DSQRT(1.5D+0)
             CLANG = 0.5D+0 * ARCL(IORD) * TLANG(NOAR,NOLP)
             GO TO 550
   540       TLANG(IRNUM,LPNUM) = TLG(IORD) / DSQRT(1.5D+0)
             CLANG = 0.5D+0 * ARCL(IORD) * TLANG(IRNUM,LPNUM)
   550       IF (CLANG .GT. CLNCE(LP,IRNUM,LPNUM)) GO TO 600
             IF (CLNCE(NOAR,LFT,NOLP) .GT. 0.0D+0) GO TO 560
             CLUNG = 0.5D+0 * TLA(KORD) / DSQRT(1.25D+0)
             CLING = CLANG * ARCL(KORD) / ARCL(IORD)
             IF (CLING .GT. CLUNG) GO TO 560
             CLNCE(LP,IRNUM,LPNUM) = CLANG
             IF (CLING .LT. CLENG .AND. CLING .LT. CLIMIT) CLNCE(NOAR,LFT,
     1       NOLP) = CLING
             GO TO 570
   560       CLANG = CLANG * DSQRT(1.2D+0)
             IF (KORD .GT. IORD) TLANG(NOAR,NOLP) = 2.0D+0 * CLANG / ARCL(
     1       IORD)
             IF (IORD .GT. KORD) TLANG(IRNUM,LPNUM) = 2.0D+0 * CLANG /
     1       ARCL(IORD)
             CLNCE(LP,IRNUM,LPNUM) = CLANG
   570       GO TO (630, 630, 580, 590), NLT
   580       TLNCE(IRNUM,LPNUM) = TLA(IORD)
             GO TO 630
   590       TLNCE(IRNUM,LPNUM) = TLA(IORD) / DSQRT(1.25D+0)
             CLNCE(IRNUM,LF,LPNUM) = 0.5D+0 * TLNCE(IRNUM,LPNUM)
             CLNCE(LPT,NOAR,NOLP) = CLNCE(IRNUM,LF,LPNUM)
             IF (CLNCE(IRNUM,LF,LPNUM) .LE. CLIMIT) GO TO 630
             CLNCE(IRNUM,LF,LPNUM) = CLIMIT
             CLNCE(LPT,NOAR,NOLP) = CLIMIT
             TLNCE(IRNUM,LPNUM) = DSQRT(TLA(IORD)**2 - CLIMIT**2)
             GO TO 630
```

```
 600     IF (CLNCE(NOAR,LFT,NOLP) .GT. 0.0D+0) GO TO 610
         CLING = CLANG * DSQRT(1.2D+0) * ARCL(KORD) / ARCL(IORD)
         CLUNG = 0.5D+0 * TLA(KORD) / DSQRT(1.25D+0)
         IF (CLUNG .GT. CLING) GO TO 620
 610     IF (KORD .GT. IORD) TLANG(NOAR,NOLP) = TLG(KORD)
         IF (IORD .GT. KORD) TLANG(IRNUM,LPNUM) = TLG(IORD)
         GO TO 630
 620     IF (CLING .LT. CLENG .AND. CLING .LT. CLIMIT) CLNCE(NOAR,LFT,
     1   NOLP) = CLING
         IF (KORD .GT. IORD) TLANG(NOAR,NOLP) = 2.0D+0 * CLING / ARCL(
     1   KORD)
         IF (IORD .GT. KORD) TLANG(IRNUM,LPNUM) = 2.0D+0 * CLING /
     1   ARCL(KORD)
C
 630  CONTINUE
C
      GO TO 720
C
C .. OUTPUT ARC
 640  TLNCEO = TLA(IORD) / DSQRT(1.25D+0)
      CLNCEO = 0.5D+0 * TLNCEO
      TLANGO = TLG(IORD) / DSQRT(1.25D+0)
      NAUT = (NOUTLP - 1) * MAXARC + IABS(NOUTLK)
      CLANG = 0.5D+0 * TLANGO * ARCL(NAUT)
      IF (NOUTLK .LT. 0) GO TO 670
      NNL = NOUTLK - 1
      IF (CLANG .GT. CLNCEO) GO TO 650
      TLNCEO = TLA(IORD)
      IF (CLANG .GT. CLNCE(NNL,NOUTLK,NOUTLP) .AND. CLNCE(NNL,NOUTLK,
     1   NOUTLP) .GT. 0.0D+0) GO TO 660
      CLNCE(NNL,NOUTLK,NOUTLP) = CLANG
      IF (CLANG .LE. CLIMIT) GO TO 720
      CLNCE(NNL,NOUTLK,NOUTLP) = CLIMIT
      TLANGO = DSQRT(TLG(IORD)**2 - (CLIMIT/ARCL(NAUT))**2)
      GO TO 720
 650  IF (CLNCEO .GT. CLNCE(NNL,NOUTLK,NOUTLP) .AND. CLNCE(NNL,NOUTLK,
     1   NOUTLP) .GT. 0.0D+0) GO TO 660
      TLANGO = TLG(IORD)
      CLNCE(NNL,NOUTLK,NOUTLP) = CLNCEO
      IF (CLNCEO .LE. CLIMIT) GO TO 720
      CLNCE(NNL,NOUTLK,NOUTLP) = CLIMIT
      TLNCEO = DSQRT(TLA(IORD)**2 - CLIMIT**2)
      GO TO 720
 660  TLNCEO = TLA(IORD)
      TLANGO = TLG(IORD)
      GO TO 720
C
 670  NOUTLK = -NOUTLK
      NNL = NOUTLK + 1
      IF (CLANG .GT. CLNCEO) GO TO 680
      TLNCEO = TLA(IORD)
      IF (CLANG .GT. CLNCE(NOUTLK,NNL,NOUTLP) .AND. CLNCE(NOUTLK,NNL,
     1   NOUTLP) .GT. 0.0D+0) GO TO 660
      CLNCE(NOUTLK,NNL,NOUTLP) = CLANG
      IF (CLANG .LE. CLIMIT) GO TO 720
      CLNCE(NOUTLK,NNL,NOUTLP) = CLIMIT
      TLANGO = DSQRT(TLG(IORD)**2 - (CLIMIT/ARCL(NAUT))**2)
      GO TO 720
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
      680   IF (CLNCEO .GT. CLNCE(NOUTLK,NNL,NOUTLP) .AND. CLNCE(NOUTLK,NNL,
         1     NOUTLP) .GT. 0.0D+0) GO TO 660
            TLANGO = TLG(IORD)
            CLNCE(NOUTLK,NNL,NOUTLP) = CLNCEO
            IF (CLNCEO .LE. CLIMIT) GO TO 720
            CLNCE(NOUTLK,NNL,NOUTLP) = CLIMIT
            TLNCEO = DSQRT(TLA(IORD)**2 - CLIMIT**2)
            GO TO 720
C
      690   IF (ARCL(IORD) .LT. 0.1D-3) GO TO 720
C
C .. ORIGIN ARC
            TLNCEG = TLA(IORD) / DSQRT(1.25D+0)
            NJI = MAXR(1)
            CLNCEG = 0.5D+0 * TLNCEG
            TLANGG = TLG(IORD) / DSQRT(1.25D+0)
            CLANG = 0.5D+0 * TLANGG * ARCL(IORD)
            IF (CLANG .GT. CLNCEG) GO TO 700
            TLNCEG = TLA(IORD)
            IF (CLANG .GT. CLNCE(NJI,1,1) .AND. CLNCE(NJI,1,1) .GT. 0.0D+0)
         1     GO TO 710
            CLNCE(NJI,1,1) = CLANG
            IF (CLANG .LE. CLIMIT) GO TO 720
            CLNCE(NJI,1,1) = CLIMIT
            TLANGG = DSQRT(TLG(IORD)**2 - (CLIMIT/ARCL(IORD))**2)
            GO TO 720
      700   IF (CLNCEG .GT. CLNCE(NJI,1,1) .AND. CLNCE(NJI,1,1) .GT. 0.0D+0)
         1     GO TO 710
            CLNCE(NJI,1,1) = CLNCEG
            TLANGG = TLG(IORD)
            IF (CLNCEG .LE. CLIMIT) GO TO 720
            CLNCE(NJI,1,1) = CLIMIT
            TLNCEG = DSQRT(TLA(IORD)**2 - CLIMIT**2)
            GO TO 720
      710   TLNCEG = TLA(IORD)
            TLANGG = TLG(IORD)
C
C
      720 CONTINUE
C
C .. OUTPUT ARC ANGLE
      730 IF (MOTN .NE. 2) GO TO 760
          TLANGO = TLG(KKM) / DSQRT(1.25D+0)
          NAUT = (NOUTLP - 1) * MAXARC + IABS(NOUTLK)
          CLANG = 0.5D+0 * TLANGO * ARCL(NAUT)
          IF (NOUTLK .LT. 0) GO TO 750
          NNL = NOUTLK - 1
          IF (CLANG .GT. CLNCE(NNL,NOUTLK,NOUTLP)) GO TO 740
          CLNCE(NNL,NOUTLK,NOUTLP) = CLANG
          GO TO 760
      740 TLANGO = TLG(KKM)
          GO TO 760
      750 NOUTLK = -NOUTLK
          NNL = NOUTLK + 1
          IF (CLANG .GT. CLNCE(NOUTLK,NNL,NOUTLP)) GO TO 740
          CLNCE(NOUTLK,NNL,NOUTLP) = CLANG
C
C .. REF POSITION OF INPUT
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
      760 TLANGC = TLG(KKO) / DSQRT(1.25D+0)
          MNJ = MAXR(1)
          MNI = MNJ - 1
          CLANG = 0.5D+0 * TLANGC * ARCL(MNJ)
          IF (CLNCE(MNI,MNJ,1) .LT. CLANG .OR. ARCL(MNJ) .LT. TLA(MNJ))
         1   GO TO 770
          CLNCE(MNI,MNJ,1) = CLANG
          GO TO 780
      770 TLANGC = TLG(KKO)
    C
    C .. PRINT OUT RESULTS
      780 WRITE (6,860)
          DO 790 ILP = 1, NLOOP
            NIM = MAXR(ILP)
            DO 790 IAR = 1, NIM
              IJK = (ILP - 1) * MAXARC + IAR
              IF (TLA(IJK) .GT. 8.0D+2 .OR. ARCL(IJK) .LT. TLA(IJK) .OR. (
         1       IREP(IJK) .NE. 0 .AND. IREP(IJK) .NE. IJK)) GO TO 790
              WRITE (6,870) IAR, ILP, ARCL(IJK), TLNCE(IAR,ILP)
      790 CONTINUE
          IF (IGNUR .EQ. 0) GO TO 810
          WRITE (6,880)
          DO 800 ILP = 1, NLOOP
            NIM = MAXR(ILP)
            DO 800 IAR = 1, NIM
              IJK = (ILP - 1) * MAXARC + IAR
              IF (ITERN(IJK) .EQ. 0 .OR. ITERN(IJK) .EQ. IJK) GO TO 800
              IF (TLG(IJK) .GT. 8.0D+2) GO TO 800
              TLANG(IAR,ILP) = TLANG(IAR,ILP) * RADEG
              WRITE (6,870) IAR, ILP, GAMA(IJK), TLANG(IAR,ILP)
      800 CONTINUE
      810 WRITE (6,890)
          DO 820 ILP = 1, NLOOP
            NIM = MAXR(ILP)
            DO 820 IAR = 1, NIM
              JAR = IAR + 1
              IF (IAR .EQ. MAXR(ILP)) JAR = 1
              IF (CLNCE(IAR,JAR,ILP) .LT. 0.0D+0) GO TO 820
              IF (ICL(IAR,JAR,ILP) .EQ. 0) GO TO 820
              WRITE (6,900) IAR, JAR, ILP, CLNCE(IAR,JAR,ILP)
      820 CONTINUE
          IF (MOTN .EQ. 4) GO TO 850
          IF ((MOTN .EQ. 1 .OR. MOTN .EQ. 3) .AND. ARCL(KKM) .LT. 0.1D-3)
         1 GO TO 840
          WRITE (6,910)
          TLANGO = TLANGO * RADEG
          IF (MOTN .EQ. 2) GO TO 830
          WRITE (6,920) ARCL(KKM), TLNCEO, GAMA(KKM), TLANGO
          GO TO 840
      830 WRITE (6,930) GAMA(KKM), TLANGO
      840 IF (MOTN .EQ. 2 .OR. ARCL(KKN) .LT. 0.1D-3) GO TO 850
          TLANGG = TLANGG * RADEG
          WRITE (6,940) ARCL(KKN), TLNCEG, GAMA(KKN), TLANGG
      850 TLANGC = TLANGC * RADEG
          WRITE (6,950) GAMA(KKO), TLANGC
    C
          RETURN
    C
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
      860 FORMAT (////5X, 'TOLERANCES ON ARC LENGTHS :-'//10X, 'ARC NUMBER',
     1         5X, 'LOOP NUMBER', 5X, 'ARC LENGTH', 5X, 'TOLERANCE')
      870 FORMAT (/14X, I2, 13X, I2, 10X, F8.3, 6X, E10.4)
      880 FORMAT (////, 5X, 'TOLERANCES ON ARC ANGLES :-'//10X, 'ARC NUMBER'
     1  , 5X, 'LOOP NUMBER', 5X, 'ANGLE MAG.', 5X, 'TOLERANCE'/41X, '(DE
     2GREES)', 6X, '(DEGREES)')
      890 FORMAT (////, 5X, 'CLEARANCES IN JOINTS :-'//10X, 'PRECEEDING',
     1         5X, 'FOLLOWING', 6X, 'LOOP NUMBER', 5X, 'CLEARANCE'/10X, 'A
     2RC NUMBER', 5X, 'ARC NUMBER')
      900 FORMAT (/14X, I2, 13X, I2, 13X, I2, 10X, E10.4)
      910 FORMAT (////5X, 'OUTPUT ARC :-'/)
      920 FORMAT (5X, 'ARC LENGTH =', F8.3, ' ,', 5X, 'TOLERANCE ON ARC LENG
     1TH =', E11.4//5X, 'ARC ANGLE  =', F8.3, ' ,', 5X, 'TOLERANCE ON AR
     2C ANGLE  =', E11.4/5X, '(DEGREES)', 25X, '(DEGREES)')
      930 FORMAT (5X, 'ARC ANGLE  =', F8.3, ' ,', 5X, 'TOLERANCE ON ARC ANGL
     1E  =', E11.4/5X, '(DEGREES)', 25X, '(DEGREES)')
      940 FORMAT (////5X, 'ORIGIN ARC :-'//5X, 'ARC LENGTH =', F8.3, ' ,',
     1        5X, 'TOLERANCE ON ARC LENGTH =', E11.4//5X, 'ARC ANGLE  =',
     2        F8.3, ' ,', 5X, 'TOLERANCE ON ', 'ARC ANGLE  =', E11.4/5X,
     3        '(DEGREES)', 25X, '(DEGREES)')
      950 FORMAT (////5X, 'INPUT REFERENCE POSITION :-'//5X, 'REFERENCE ANGL
     1E =', F8.3, ' ,', 5X, 'TOLERANCE ', 'ON ANGLE =', E11.4/8X, '(DEGR
     2EES)', 25X, '(DEGREES)')
C
          END
C
C   *    *    *    *    *    *    *    *    *    *    *    *    *    *    *    *    *
C
          SUBROUTINE JSBSCR(IRD, IX, NL, NA, LP, LF, JAX, NLP)
C
          IMPLICIT REAL*8(A - H,O - Z)
C
          DIMENSION JAX(NLP)
C
          NL = (IRD - 1) / IX + 1
          NA = IRD - IX * (NL - 1)
C
          IF (NA .NE. 1) GO TO 10
          LP = JAX(NL)
          LF = 2
          GO TO 30
C
       10 IF (NA .NE. JAX(NL)) GO TO 20
          LP = JAX(NL) - 1
          LF = 1
          GO TO 30
C
       20 LP = NA - 1
          LF = NA + 1
C
       30 RETURN
          END
C
C   *    *    *    *    *    *    *    *    *    *    *    *    *    *    *    *
C
          SUBROUTINE TERNRY(ARCL, ITERN, MT, LT, KKL, NLP, KX)
C
          IMPLICIT REAL*8(A - H,O - Z)
```

Listing of TOCALM at 16:46:02 on JUL 23, 1982 for CCid=MCB8

```
C
      DIMENSION ARCL(KKL), LT(NLP), ITERN(KKL)
C
      DO 10 I = 1, NLP
        LT(I) = 0
   10 CONTINUE
C
      KX = 0
      DO 20 J = 1, KKL
        IF (J .EQ. MT) GO TO 20
        IF (ITERN(J) .NE. ITERN(MT)) GO TO 20
        IF (ARCL(J) .LT. 1.0D-3) GO TO 20
        KX = KX + 1
        LT(KX) = J
   20 CONTINUE
C
      RETURN
      END
```

A P P E N D I X   I B

INPUT DATA FOR PROGRAM TOCALM

INPUT FORMAT

The input data is read from computer device 5. The first line in the data set contains the title which is printed at the top of the output. The length allowed is 72 characters.

The remaining data is read in using NAMELIST, which is a specification statement used in conjunction with the READ (n, x) and WRITE (m, x) statements. It provides for reading and writing data without including the list specification in the READ and WRITE statements. The NAMELIST statement declares a name x to refer to a particular list of variables or array names. The first character in each record to be read must be blank. The second character in the first record of a list x must be an ampersand '&', immediately followed by the NAMELIST name x. The NAMELIST name must be followed by a blank. This name is followed by data items separated by commas. The end of a data list is signalled by '&END'.

For the present program the data is divided into three groups (lists x) called LINKS, METHOD and LIMITS. The list of variables and arrays in each is as follows:

Data Group LINKS

NLOOP	the number of loops in the linkage.
MAXARC	the maximum number of arcs in any of the loops.
ILINK, ILOOP	the arrays containing the topological description as defined in reference [31].
ARCL	array containing the lengths of the arcs.
GAMA	array containing the arc angles in degrees.

NOUTLK	the number in its own loop of the arc common with the output arc. It is negative if the output arc is measured from the start of NOUTLK
NOUTLP	the number of the loop containing NOUTLK.
OUTLK	the length of the output arc.
OUTANG	output arc angle
IUNITS	1 if the arc lengths are in millimeters 2 if the arc lengths are in inches.

Data Group METHOD

CRNKI	input reference angle.
CRNKD	input angular velocity.
NDES or NPOINT	the number of input link positions.
DELT	array of input positions measured from the reference position.
MOTION	0 if output is given by x-y cartesian coordinates of a point on the output arc 1 if the output is the angular position of the output arc. 2 if the output is given by the polar coordinates of a point on the output arc. 3. if the output is the linear position of a sliding output link.

IDEL	0	if output is to be computed at equally spaced input positions
	1	if output is to be computed at input positions specified in DELT

Data Group LIMITS

ORIGLK,ORIGNG	polar coordinates of input link pivot.
OUTOLX,OUTOLY	output tolerance in x- and y-directions if MOTION = 0.
OUTOLA	output tolerance in degrees, if MOTION = 1.
OUTOLR,OUTOLT	output tolerance in polar coordinate directions, MOTION = 2.
OUTOLS	output tolerance, if MOTION = 3.
IANG	same as MOTION
COSTA	array containing cost-tolerance constants B_i for arc lengths parameters.
COSTG	array containing cost-tolerance constants for arc angle parameters.
LAST	0 if this is not the last set of input data, i.e. another follows 1 if this is the last set of data.

SAMPLE INPUT DATA

```
of DATA at 17:22:50 on JUL 27, 1982 for CCid=MCB8

    FOUR-BAR SINE FUNCTION GENERATOR
    &LINKS NLOOP=1,MAXARC=4,ILINK=1,-2,-3,4,ILOOP=1,0,0,1,
    ARCL=2.06980,2.42146,0.74613,1.0,
    NOUTLK=3,NOUTLP=1,OUTANG=287.148,IUNITS=2 &END
    &METHOD MOTION=1,CRNKI=296.211,NPOINT=19,IDEL=1,CRNKD=1.0,
    DELT=0.0,5.0,10.0,15.0,20.0,25.0,30.0,35.0,40.0,45.0,
    50.0,55.0,60.0,65.0,70.0,75.0,80.0,85.0,90.0 &END
    &LIMITS IANG=1,TANGLE=65.0,OUTOLA=1.0,COSTA=4.2,6.1,0.5,1.0,
    101*0.0,COSTG=4*0.0,2.25,2.25,4.0,98*0.0,
    LAST=0 &END
    SLIDER CRANK MECHANISM
    &LINKS NLOOP=1,MAXARC=4,ILINK=1,-2,-3,4,ILOOP=1,0,-4,1,
    ARCL=50.0,200.0,2*0.0,GAMA=4*0.0,
    NOUTLK=-3,NOUTLP=1,IUNITS=1 &END
    &METHOD MOTION=3,CRNKI=270.0,NPOINT=37,IDEL=1,CRNKD=1.0,
    DELT=0.0,10.0,20.0,30.0,40.0,50.0,60.0,70.0,80.0,90.0,
    100.0,110.0,120.0,130.0,140.0,150.0,160.0,170.0,180.0,
    190.0,200.0,210.0,220.0,230.0,240.0,250.0,260.0,270.0,
    280.0,290.0,300.0,310.0,320.0,330.0,340.0,350.0,360.0 &END
    &LIMITS OUTOLS=0.25,COSTA=100.0,1600.0,1.0,100.0,
    LAST=0 &END
    SIX-BAR SINE FUNCTION GENERATOR
    &LINKS NLOOP=2,MAXARC=5,ILINK=1,-2,-3,4,0,
    1,-2,-3,-4,-4,ILOOP=1,0,0,1,0,1,1,0,0,1,
    ARCL=7.5,30.139,21.116,36.716,0.0,7.5,28.833,
    21.421,61.832,39.998,GAMA=6*0.0,179.99,2*0.0,
    64.537,IUNITS=1,NOUTLK=4,NOUTLP=2,OUTANG=31.0902 &END
    &METHOD MOTION=1,CRNKI=212.333,NPOINT=37,IDEL=1,CRNKD=1.0,
    DELT=0.0,5.0,10.0,15.0,20.0,25.0,30.0,35.0,40.0,45.0,50.0,
    55.0,60.0,65.0,70.0,75.0,80.0,85.0,90.0,95.0,100.0,105.0,
    110.0,115.0,120.0,125.0,130.0,135.0,140.0,145.0,150.0,
    155.0,160.0,165.0,170.0,175.0,180.0 &END
    &LIMITS OUTOLA=0.5,COSTA=1.0,15.0,10.0,25.0,1.0,1.0,15.0,
    10.0,60.0,25.0,COSTG=6*1.0,10.0,2*1.0,25.0,25.0,1.0,60.0,
    LAST=0 &END
    TEN-BAR NEEDLE MECHANISM
    &LINKS NLOOP=4,MAXARC=6,ILINK=1,2,3,4,0,0,1,2,3,
    -4,0,0,1,2,-3,-4,-3,4,1,2,3,-4,-5,-4,76*0,
    ILOOP=1,0,0,1,0,0,1,0,0,1,0,0,1,2,0,0,1,1,1,2,3,0,0,1,76*0,
    ARCL=19.0,95.0,104.5,112.4681,0.0,0.0,19.0,105.0,31.0,
    110.0639,0.0,0.0,19.0,105.0,61.0,12.5,24.0,112.4681,19.0,
    105.0,61.0,245.5,85.0,352.7155,76*0.0,GAMA=9*0.0,
    33.5934,6*0.0,47.3839,6*0.0,20.4425,76*0.0,
    NOUTLK=5,NOUTLP=4,OUTLK=121.6,OUTANG=10.0484,IUNITS=1 &END
    &METHOD MOTION=1,CRNKI=121.6409,
    DELT=0.0,15.0,30.0,45.0,60.0,75.0,90.0,105.0,120.0,135.0,
    150.0,165.0,180.0,195.0,210.0,225.0,240.0,255.0,270.0,
    285.0,300.0,315.0,330.0,345.0,360.0,336*0.0,NDES=25 &END
    &LIMITS KTHETA=2,OUTOLA=1.0,COSTA=1.0,25.0,30.0,36.0,2*0.0,
    1.0,30.0,2.3,36.0,2*0.0,1.0,30.0,9.5,0.5,2.0,36.0,1.0,30.0,
    9.5,170.0,20.0,350.0,81*0.0,COSTG=9*0.0,0.5,6*0.0,0.75,
    6*0.0,0.2,4.5,0.0,4.0,78*0.0,
    LAST=1 &END
```

A P P E N D I X   I C

LISTING OF COMPUTER PROGRAM

PSODPLOTS

Listing of PSODPLOTS at 17:26:38 on JUL 23, 1982 for CCid=MCB8

```
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C *      PROGRAM FOR PLOTTING OUTPUT SENSITIVITIES TO PARAMETER      *
C *         AND OUTPUT DEVIATION AGAINST CRANK INPUT POSITION        *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT REAL*8(D,P,T,O), LOGICAL*1(L)
      DIMENSION DELT(37), DEVX(37), DEVY(37), DEVA(37), DEVR(37),
     1          DEVT(37), DEVS(37), STEP(37), ERPX(37), ERNX(37),
     2          ERPY(37), ERNY(37), ERPA(37), ERNA(37), ERPR(37),
     3          ERNR(37), ERPT(37), ERNT(37), ERPS(37), ERNS(37),
     4          PDX(37,100), SNX(37,100), PDY(37,100), SNY(37,100),
     5          PDGX(37,100), SNGX(37,100), PDGY(37,100), SNGY(37,100),
     6          PDOLX(37), SNOLX(37), PDOLY(37), SNOLY(37), PDOGX(37),
     7          SNOGX(37), PDOGY(37), SNOGY(37), PDORLX(37), SNORLX(37),
     8          PDORLY(37), SNORLY(37), PDORGX(37), SNORGX(37),
     9          PDORGY(37), SNORGY(37), PDCRX(37), SNCRX(37), PDCRY(37),
     *          SNCRY(37), VAR(37), LTITLE(72)
C
C---------READ IN DATA
C
      READ (7) LTITLE
      READ (7) MOTN, NDES, IUNITS, MAXARC, NLOOP, KKL
      READ (7) (DELT(I),I=1,NDES)
C
      DO 10 K = 1, NDES
        STEP(K) = DELT(K)
   10 CONTINUE
C
      GO TO (20, 50, 80, 110), MOTN
C
   20 READ (7) OUTOLX, OUTOLY, (DEVX(I),I=1,NDES), (DEVY(J),J=1,NDES),
     1     ((PDX(I,J),I=1,NDES),J=1,KKL), ((PDY(I,J),I=1,NDES),J=1,KKL),
     2     ((PDGX(I,J),I=1,NDES),J=1,KKL), ((PDGY(I,J),I=1,NDES),J=1,
     3     KKL), (PDOLX(I),I=1,NDES), (PDOLY(I),I=1,NDES),
     4     (PDOGX(I),I=1,NDES), (PDOGY(I),I=1,NDES), (PDORLX(I),I=1,
     5     NDES), (PDORLY(I),I=1,NDES), (PDORGX(I),I=1,NDES),
     6     (PDORGY(I),I=1,NDES), (PDCRX(I),I=1,NDES), (PDCRY(I),I=1,
     7     NDES)
C
      ADEVX = OUTOLX
      ADEVY = OUTOLY
      VSCALX = 1.2 * ADEVX
      VSCALY = 1.2 * ADEVY
C
      DO 30 I = 1, NDES
        ERPX(I) = DEVX(I)
        ERNX(I) = -ERPX(I)
        ERPY(I) = DEVY(I)
        ERNY(I) = -ERPY(I)
        SNOLX(I) = PDOLX(I)
        SNOLY(I) = PDOLY(I)
        SNOGX(I) = PDOGX(I)
        SNOGY(I) = PDOGY(I)
        SNORLX(I) = PDORLX(I)
        SNORLY(I) = PDORLY(I)
        SNORGX(I) = PDORGX(I)
        SNORGY(I) = PDORGY(I)
        SNCRX(I) = PDCRX(I)
```

Listing of PSODPLOTS at 17:26:38 on JUL 23, 1982 for CCid=MCB8

```
            SNCRY(I) = PDCRY(I)
   30 CONTINUE
C
      DO 40 J = 1, KKL
        DO 40 I = 1, NDES
          SNX(I,J) = PDX(I,J)
          SNY(I,J) = PDY(I,J)
          SNGX(I,J) = PDGX(I,J)
          SNGY(I,J) = PDGY(I,J)
   40 CONTINUE
C
      GO TO 140
C
   50 READ (7) OUTOLA, (DEVA(I),I=1,NDES), ((PDX(I,J),I=1,NDES),J=1,KKL)
     1     , ((PDGX(I,J),I=1,NDES),J=1,KKL), (PDOGX(I),I=1,NDES),
     2     (PDCRX(I),I=1,NDES)
C
      ADEVA = OUTOLA
      VSCALA = 1.2 * ADEVA
C
      DO 60 I = 1, NDES
        ERPA(I) = DEVA(I)
        ERNA(I) = -ERPA(I)
        SNOGX(I) = PDOGX(I)
        SNCRX(I) = PDCRX(I)
   60 CONTINUE
C
      DO 70 J = 1, KKL
        DO 70 I = 1, NDES
          SNX(I,J) = PDX(I,J)
          SNGX(I,J) = PDGX(I,J)
   70 CONTINUE
C
      GO TO 140
C
   80 READ (7) OUTOLR, OUTOLT, (DEVR(I),I=1,NDES), (DEVT(J),J=1,NDES),
     1     ((PDX(I,J),I=1,NDES),J=1,KKL), ((PDY(I,J),I=1,NDES),J=1,KKL),
     2     ((PDGX(I,J),I=1,NDES),J=1,KKL), ((PDGY(I,J),I=1,NDES),J=1,
     3     KKL), (PDOLX(I),I=1,NDES), (PDOLY(I),I=1,NDES),
     4     (PDOGX(I),I=1,NDES), (PDOGY(I),I=1,NDES), (PDORLX(I),I=1,
     5     NDES), (PDORLY(I),I=1,NDES), (PDORGX(I),I=1,NDES),
     6     (PDORGY(I),I=1,NDES), (PDCRX(I),I=1,NDES), (PDCRY(I),I=1,
     7     NDES)
C
      ADEVR = OUTOLR
      ADEVT = OUTOLT
      VSCALR = 1.2 * ADEVR
      VSCALT = 1.2 * ADEVT
C
      DO 90 I = 1, NDES
        ERPR(I) = DEVR(I)
        ERNR(I) = -ERPR(I)
        ERPT(I) = DEVT(I)
        ERNT(I) = -ERPT(I)
        SNOLX(I) = PDOLX(I)
        SNOLY(I) = PDOLY(I)
        SNOGX(I) = PDOGX(I)
        SNOGY(I) = PDOGY(I)
```

Listing of PSODPLOTS at 17:26:38 on JUL 23, 1982 for CCid=MCB8

```
               SNORLX(I) = PDORLX(I)
               SNORLY(I) = PDORLY(I)
               SNORGX(I) = PDORGX(I)
               SNORGY(I) = PDORGY(I)
               SNCRX(I) = PDCRX(I)
               SNCRY(I) = PDCRY(I)
          90 CONTINUE
      C
             DO 100 J = 1, KKL
               DO 100 I = 1, NDES
                 SNX(I,J) = PDX(I,J)
                 SNY(I,J) = PDY(I,J)
                 SNGX(I,J) = PDGX(I,J)
                 SNGY(I,J) = PDGY(I,J)
         100 CONTINUE
      C
             GO TO 140
      C
         110 READ (7) OUTOLS, (DEVS(I),I=1,NDES), ((PDX(I,J),I=1,NDES),J=1,KKL)
            1     , ((PDGX(I,J),I=1,NDES),J=1,KKL), (PDCRX(I),I=1,NDES)
      C
             ADEVS = OUTOLS
             VSCALS = 1.2 * ADEVS
      C
             DO 120 I = 1, NDES
               ERPS(I) = DEVS(I)
               ERNS(I) = -ERPS(I)
               SNCRX(I) = PDCRX(I)
         120 CONTINUE
      C
             DO 130 J = 1, KKL
               DO 130 I = 1, NDES
                 SNX(I,J) = PDX(I,J)
                 SNGX(I,J) = PDGX(I,J)
         130 CONTINUE
      C
      C----------PLOT GRAPHS
      C
         140 XMTR = 0.1 * (STEP(NDES) - STEP(1))
             NORT = 1
      C
             CALL PAPER(1)
      C
             GO TO (150, 400, 480, 730), MOTN
      C
         150 CALL SETPLS(STEP, XMTR, NDES, VSCALX, IUNITS, MOTN, NORT, LTITLE)
      C
             CALL CTRSET(2)
             CALL PLOTCS(STEP(1), 1.1*VSCALX, 'OUTPUT DEVIATION IN ', 20)
             CALL TYPECS('X-DIRECTION', 11)
             CALL BROKEN(5, 10, 5, 10)
             CALL POSITN(STEP(1), ADEVX)
             CALL JOIN(STEP(NDES), ADEVX)
             CALL POSITN(STEP(1), -ADEVX)
             CALL JOIN(STEP(NDES), -ADEVX)
             CALL FULL
             CALL CTRSET(4)
             CALL THICK(1)
```

Listing of PSODPLOTS at 17:26:38 on JUL 23, 1982 for CCid=MCB8

```
            CALL PTPLOT(STEP, ERPX, 1, NDES, 54)
            CALL THICK(3)
            CALL NSCURV(STEP, ERPX, 1, NDES)
            CALL THICK(1)
            CALL PTPLOT(STEP, ERNX, 1, NDES, 54)
            CALL THICK(3)
            CALL NSCURV(STEP, ERNX, 1, NDES)
      C
            CALL FRAME
            CALL SETPLS(STEP, XMTR, NDES, VSCALY, IUNITS, MOTN, NORT, LTITLE)
            CALL CTRSET(2)
            CALL PLOTCS(STEP(1), 1.1*VSCALY, 'OUTPUT DEVIATION IN ', 20)
            CALL TYPECS('Y-DIRECTION', 11)
            CALL BROKEN(5, 10, 5, 10)
            CALL POSITN(STEP(1), ADEVY)
            CALL JOIN(STEP(NDES), ADEVY)
            CALL POSITN(STEP(1), -ADEVY)
            CALL JOIN(STEP(NDES), -ADEVY)
            CALL FULL
            CALL CTRSET(4)
            CALL THICK(1)
            CALL PTPLOT(STEP, ERPY, 1, NDES, 54)
            CALL THICK(3)
            CALL NSCURV(STEP, ERPY, 1, NDES)
            CALL THICK(1)
            CALL PTPLOT(STEP, ERNY, 1, NDES, 54)
            CALL THICK(3)
            CALL NSCURV(STEP, ERNY, 1, NDES)
      C
            NLP = 0
            DO 180 J = 1, KKL
              IF (SNX(1,J) .LT. - 5.0D-1) GO TO 180
              IF (J .LE. MAXARC*NLP) GO TO 160
              CALL FRAME
              MAS = 49
              CALL YSCALE(SNX, VORD, NDES, KKL, MAXARC, J)
              CALL PAXES(STEP, XMTR, VORD, NDES, LTITLE)
              CALL PLOTCS(STEP(2), 1.05*VORD, 'SENSITIVITY IN X-DIRECTION',
           1        26)
              CALL CTRORI(1.0)
              CALL POSITN(-1.75*XMTR, 0.325*VORD)
              IF (IUNITS .EQ. 1) CALL TYPECS('( MM / MM )', 11)
              IF (IUNITS .EQ. 2) CALL TYPECS('( IN / IN )', 11)
              CALL CTRORI(0.0)
        160   NLP = (J - 1) / MAXARC + 1
              NAR = J - (NLP - 1) * MAXARC
              MAS = MAS + 1
              DO 170 I = 1, NDES
                VAR(I) = SNX(I,J)
        170   CONTINUE
              CALL CTRSET(4)
              CALL THICK(1)
              CALL PTPLOT(STEP, VAR, 1, NDES, MAS)
              CALL THICK(2)
              CALL NSCURV(STEP, VAR, 1, NDES)
              CALL SYMKEY(STEP, VORD, XMTR, NDES, NAR, NLP, MAS, 11)
        180 CONTINUE
      C
```

Listing of PSODPLOTS at 17:26:38 on JUL 23, 1982 for CCid=MCB8

```
      YORD = 0.0
      DO 190 I = 1, NDES
        IF (SNOLX(I) .GT. YORD) YORD = SNOLX(I)
        IF (SNORLX(I) .GT. YORD) YORD = SNORLX(I)
  190 CONTINUE
      IF (YORD .LT. 1.0D-9) GO TO 210
      VORD = 1.1 * YORD
      CALL FRAME
      CALL PAXES(STEP, XMTR, VORD, NDES, LTITLE)
      CALL PLOTCS(STEP(2), 1.05*VORD, 'SENSITIVITY IN X-DIRECTION', 26)
      CALL CTRORI(1.0)
      CALL POSITN(-1.75*XMTR, 0.325*VORD)
      IF (IUNITS .EQ. 1) CALL TYPECS('( MM / MM )', 11)
      IF (IUNITS .EQ. 2) CALL TYPECS('( IN / IN )', 11)
      CALL CTRORI(0.0)
      IF (SNOLX(1) .LT. - 5.0D-1) GO TO 200
      CALL CTRSET(4)
      CALL THICK(1)
      CALL PTPLOT(STEP, SNOLX, 1, NDES, 51)
      CALL THICK(2)
      CALL NSCURV(STEP, SNOLX, 1, NDES)
      CALL SYMBKY(STEP, VORD, XMTR, NDES, 25, 11, 51)
  200 IF (SNORLX(1) .LT. - 5.0D-1) GO TO 210
      CALL THICK(1)
      CALL CTRSET(4)
      CALL PTPLOT(STEP, SNORLX, 1, NDES, 52)
      CALL THICK(2)
      CALL NSCURV(STEP, SNORLX, 1, NDES)
      CALL SYMBKY(STEP, VORD, XMTR, NDES, 17, 11, 52)
C
  210 NLP = 0
      DO 240 J = 1, KKL
        IF (J .EQ. MAXARC*NLP + 1) NGG = 0
        IF (SNGX(1,J) .LT. - 5.0D-1) GO TO 240
        NGG = NGG + 1
        IF (J .LE. MAXARC*NLP .AND. NGG .NE. 1) GO TO 220
        CALL FRAME
        NAS = 49
        CALL YSCALE(SNGX, VORD, NDES, KKL, MAXARC, J)
        CALL PAXES(STEP, XMTR, VORD, NDES, LTITLE)
        CALL PLOTCS(STEP(2), 1.05*VORD, 'SENSITIVITY IN X-DIRECTION',
     1      26)
        CALL CTRORI(1.0)
        CALL POSITN(-1.75*XMTR, 0.325*VORD)
        IF (IUNITS .EQ. 1) CALL TYPECS('( MM / RAD )', 12)
        IF (IUNITS .EQ. 2) CALL TYPECS('( IN / RAD )', 12)
        CALL CTRORI(0.0)
  220   NLP = (J - 1) / MAXARC + 1
        NAR = J - (NLP - 1) * MAXARC
        NAS = NAS + 1
        DO 230 I = 1, NDES
          VAR(I) = SNGX(I,J)
  230   CONTINUE
        CALL THICK(1)
        CALL CTRSET(4)
        CALL PTPLOT(STEP, VAR, 1, NDES, NAS)
        CALL THICK(2)
        CALL NSCURV(STEP, VAR, 1, NDES)
```

Listing of PSODPLOTS at 17:26:38 on JUL 23, 1982 for CCid=MCB8

```
            CALL SYMKEY(STEP, VORD, XMTR, NDES, NAR, NLP, NAS, 17)
  240 CONTINUE
C
      YORD = 0.0
      DO 250 I = 1, NDES
        IF (SNOGX(I) .GT. YORD) YORD = SNOGX(I)
        IF (SNORGX(I) .GT. YORD) YORD = SNORGX(I)
        IF (SNCRX(I) .GT. YORD) YORD = SNCRX(I)
  250 CONTINUE
      VORD = 1.1 * YORD
      CALL FRAME
      CALL PAXES(STEP, XMTR, VORD, NDES, LTITLE)
      CALL PLOTCS(STEP(2), 1.05*VORD, 'SENSITIVITY IN X-DIRECTION', 26)
      CALL CTRORI(1.0)
      CALL POSITN(-1.75*XMTR, 0.325*VORD)
      IF (IUNITS .EQ. 1) CALL TYPECS('( MM / RAD )', 12)
      IF (IUNITS .EQ. 2) CALL TYPECS('( IN / RAD )', 12)
      CALL CTRORI(0.0)
      IF (SNOGX(1) .LT. - 5.0D-1) GO TO 260
      CALL CTRSET(4)
      CALL THICK(1)
      CALL PTPLOT(STEP, SNOGX, 1, NDES, 51)
      CALL THICK(2)
      CALL NSCURV(STEP, SNOGX, 1, NDES)
      CALL SYMBKY(STEP, VORD, XMTR, NDES, 25, 17, 51)
  260 IF (SNORGX(1) .LT. - 5.0D-1) GO TO 270
      CALL THICK(1)
      CALL CTRSET(4)
      CALL PTPLOT(STEP, SNORGX, 1, NDES, 52)
      CALL THICK(2)
      CALL NSCURV(STEP, SNORGX, 1, NDES)
      CALL SYMBKY(STEP, VORD, XMTR, NDES, 17, 17, 52)
  270 CALL THICK(1)
      CALL CTRSET(4)
      CALL PTPLOT(STEP, SNCRX, 1, NDES, 53)
      CALL THICK(2)
      CALL NSCURV(STEP, SNCRX, 1, NDES)
      CALL SYMBKY(STEP, VORD, XMTR, NDES, 13, 17, 53)
C
      NLP = 0
      DO 300 J = 1, KKL
        IF (SNY(1,J) .LT. - 5.0D-1) GO TO 300
        IF (J .LE. MAXARC*NLP) GO TO 280
        CALL FRAME
        MAS = 49
        CALL YSCALE(SNY, VORD, NDES, KKL, MAXARC, J)
        CALL PAXES(STEP, XMTR, VORD, NDES, LTITLE)
        CALL PLOTCS(STEP(2), 1.05*VORD, 'SENSITIVITY IN Y-DIRECTION',
     1      26)
        CALL CTRORI(1.0)
        CALL POSITN(-1.75*XMTR, 0.325*VORD)
        IF (IUNITS .EQ. 1) CALL TYPECS('( MM / MM )', 11)
        IF (IUNITS .EQ. 2) CALL TYPECS('( IN / IN )', 11)
        CALL CTRORI(0.0)
  280   NLP = (J - 1) / MAXARC + 1
        NAR = J - (NLP - 1) * MAXARC
        MAS = MAS + 1
        DO 290 I = 1, NDES
```

Listing of PSODPLOTS at 17:26:38 on JUL 23, 1982 for CCid=MCB8

```
              VAR(I) = SNY(I,J)
   290      CONTINUE
            CALL THICK(1)
            CALL CTRSET(4)
            CALL PTPLOT(STEP, VAR, 1, NDES, MAS)
            CALL THICK(2)
            CALL NSCURV(STEP, VAR, 1, NDES)
            CALL SYMKEY(STEP, VORD, XMTR, NDES, NAR, NLP, MAS, 11)
   300 CONTINUE
C
       YORD = 0.0
       DO 310 I = 1, NDES
         IF (SNOLY(I) .GT. YORD) YORD = SNOLY(I)
         IF (SNORLY(I) .GT. YORD) YORD = SNORLY(I)
   310 CONTINUE
       IF (YORD .LT. 1.0D-9) GO TO 330
       VORD = 1.1 * YORD
       CALL FRAME
       CALL PAXES(STEP, XMTR, VORD, NDES, LTITLE)
       CALL PLOTCS(STEP(2), 1.05*VORD, 'SENSITIVITY IN Y-DIRECTION', 26)
       CALL CTRORI(1.0)
       CALL POSITN(-1.75*XMTR, 0.325*VORD)
       IF (IUNITS .EQ. 1) CALL TYPECS('( MM / MM )', 11)
       IF (IUNITS .EQ. 2) CALL TYPECS('( IN / IN )', 11)
       CALL CTRORI(0.0)
       IF (SNOLY(1) .LT. - 5.0D-1) GO TO 320
       CALL CTRSET(4)
       CALL THICK(1)
       CALL PTPLOT(STEP, SNOLY, 1, NDES, 51)
       CALL THICK(2)
       CALL NSCURV(STEP, SNOLY, 1, NDES)
       CALL SYMBKY(STEP, VORD, XMTR, NDES, 25, 11, 51)
   320 IF (SNORLY(1) .LT. - 5.0D-1) GO TO 330
       CALL THICK(1)
       CALL CTRSET(4)
       CALL PTPLOT(STEP, SNORLY, 1, NDES, 52)
       CALL THICK(2)
       CALL NSCURV(STEP, SNORLY, 1, NDES)
       CALL SYMBKY(STEP, VORD, XMTR, NDES, 17, 11, 52)
C
   330 NLP = 0
       DO 360 J = 1, KKL
         IF (J .EQ. MAXARC*NLP + 1) NGG = 0
         IF (SNGY(1,J) .LT. - 5.0D-1) GO TO 360
         NGG = NGG + 1
         IF (J .LE. MAXARC*NLP .AND. NGG .NE. 1) GO TO 340
         CALL FRAME
         NAS = 49
         CALL YSCALE(SNGY, VORD, NDES, KKL, MAXARC, J)
         CALL PAXES(STEP, XMTR, VORD, NDES, LTITLE)
         CALL PLOTCS(STEP(2), 1.05*VORD, 'SENSITIVITY IN Y-DIRECTION',
     1        26)
         CALL CTRORI(1.0)
         CALL POSITN(-1.75*XMTR, 0.325*VORD)
         IF (IUNITS .EQ. 1) CALL TYPECS('( MM / RAD )', 12)
         IF (IUNITS .EQ. 2) CALL TYPECS('( IN / RAD )', 12)
         CALL CTRORI(0.0)
   340   NLP = (J - 1) / MAXARC + 1
```

```
                NAR = J - (NLP - 1) * MAXARC
                NAS = NAS + 1
                DO 350 I = 1, NDES
                  VAR(I) = SNGY(I,J)
          350   CONTINUE
                CALL THICK(1)
                CALL CTRSET(4)
                CALL PTPLOT(STEP, VAR, 1, NDES, NAS)
                CALL THICK(2)
                CALL NSCURV(STEP, VAR, 1, NDES)
                CALL SYMKEY(STEP, VORD, XMTR, NDES, NAR, NLP, NAS, 17)
          360 CONTINUE
      C
              YORD = 0.0
              DO 370 I = 1, NDES
                IF (SNOGY(I) .GT. YORD) YORD = SNOGY(I)
                IF (SNORGY(I) .GT. YORD) YORD = SNORGY(I)
                IF (SNCRY(I) .GT. YORD) YORD = SNCRY(I)
          370 CONTINUE
              VORD = 1.1 * YORD
              CALL FRAME
              CALL PAXES(STEP, XMTR, VORD, NDES, LTITLE)
              CALL PLOTCS(STEP(2), 1.05*VORD, 'SENSITIVITY IN Y-DIRECTION', 26)
              CALL CTRORI(1.0)
              CALL POSITN(-1.75*XMTR, 0.325*VORD)
              IF (IUNITS .EQ. 1) CALL TYPECS('( MM / RAD )', 12)
              IF (IUNITS .EQ. 2) CALL TYPECS('( IN / RAD )', 12)
              CALL CTRORI(0.0)
              IF (SNOGY(1) .LT. - 5.0D-1) GO TO 380
              CALL THICK(1)
              CALL CTRSET(4)
              CALL PTPLOT(STEP, SNOGY, 1, NDES, 51)
              CALL THICK(2)
              CALL NSCURV(STEP, SNOGY, 1, NDES)
              CALL SYMBKY(STEP, VORD, XMTR, NDES, 25, 17, 51)
          380 IF (SNORGY(1) .LT. - 5.0D-1) GO TO 390
              CALL THICK(1)
              CALL CTRSET(4)
              CALL PTPLOT(STEP, SNORGY, 1, NDES, 52)
              CALL THICK(2)
              CALL NSCURV(STEP, SNORGY, 1, NDES)
              CALL SYMBKY(STEP, VORD, XMTR, NDES, 17, 17, 52)
          390 CALL THICK(1)
              CALL CTRSET(4)
              CALL PTPLOT(STEP, SNCRY, 1, NDES, 53)
              CALL THICK(2)
              CALL NSCURV(STEP, SNCRY, 1, NDES)
              CALL SYMBKY(STEP, VORD, XMTR, NDES, 13, 17, 53)
      C
              GO TO 810
      C
          400 CALL SETPLS(STEP, XMTR, NDES, VSCALA, IUNITS, MOTN, NORT, LTITLE)
      C
              CALL CTRSET(2)
              CALL BROKEN(5, 10, 5, 10)
              CALL POSITN(STEP(1), ADEVA)
              CALL JOIN(STEP(NDES), ADEVA)
              CALL POSITN(STEP(1), -ADEVA)
```

```
      CALL JOIN(STEP(NDES), -ADEVA)
      CALL FULL
      CALL CTRSET(4)
      CALL THICK(1)
      CALL PTPLOT(STEP, ERPA, 1, NDES, 54)
      CALL THICK(3)
      CALL NSCURV(STEP, ERPA, 1, NDES)
      CALL THICK(1)
      CALL PTPLOT(STEP, ERNA, 1, NDES, 54)
      CALL THICK(3)
      CALL NSCURV(STEP, ERNA, 1, NDES)
C
      NLP = 0
      DO 430 J = 1, KKL
        IF (SNX(1,J) .LT. - 5.0D-1) GO TO 430
        IF (J .LE. MAXARC*NLP) GO TO 410
        CALL FRAME
        MAS = 49
        CALL YSCALE(SNX, VORD, NDES, KKL, MAXARC, J)
        CALL PAXES(STEP, XMTR, VORD, NDES, LTITLE)
        CALL CTRORI(1.0)
        CALL POSITN(-1.75*XMTR, 0.325*VORD)
        IF (IUNITS .EQ. 1) CALL TYPECS('( RAD / MM )', 12)
        IF (IUNITS .EQ. 2) CALL TYPECS('( RAD / IN )', 12)
        CALL CTRORI(0.0)
  410   NLP = (J - 1) / MAXARC + 1
        NAR = J - (NLP - 1) * MAXARC
        MAS = MAS + 1
        DO 420 I = 1, NDES
          VAR(I) = SNX(I,J)
  420   CONTINUE
        CALL THICK(1)
        CALL CTRSET(4)
        CALL PTPLOT(STEP, VAR, 1, NDES, MAS)
        CALL THICK(2)
        CALL NSCURV(STEP, VAR, 1, NDES)
        CALL SYMKEY(STEP, VORD, XMTR, NDES, NAR, NLP, MAS, 11)
  430 CONTINUE
C
      NLP = 0
      DO 460 J = 1, KKL
        IF (J .EQ. MAXARC*NLP + 1) NGG = 0
        IF (SNGX(1,J) .LT. - 5.0D-1) GO TO 460
        NGG = NGG + 1
        IF (J .LE. MAXARC*NLP .AND. NGG .NE. 1) GO TO 440
        CALL FRAME
        MAS = 49
        CALL YSCALE(SNGX, VORD, NDES, KKL, MAXARC, J)
        CALL PAXES(STEP, XMTR, VORD, NDES, LTITLE)
        CALL CTRORI(1.0)
        CALL POSITN(-1.75*XMTR, 0.325*VORD)
        CALL TYPECS('( RAD / RAD )', 13)
        CALL CTRORI(0.0)
  440   NLP = (J - 1) / MAXARC + 1
        NAR = J - (NLP - 1) * MAXARC
        MAS = MAS + 1
        DO 450 I = 1, NDES
          VAR(I) = SNGX(I,J)
```

```
      450   CONTINUE
            CALL THICK(1)
            CALL CTRSET(4)
            CALL PTPLOT(STEP, VAR, 1, NDES, NAS)
            CALL THICK(2)
            CALL NSCURV(STEP, VAR, 1, NDES)
            CALL SYMKEY(STEP, VORD, XMTR, NDES, NAR, NLP, NAS, 17)
      460 CONTINUE
    C
          YORD = 0.0
          DO 470 I = 1, NDES
            IF (SNOGX(I) .GT. YORD) YORD = SNOGX(I)
            IF (SNCRX(I) .GT. YORD) YORD = SNCRX(I)
      470 CONTINUE
          VORD = 1.1 * YORD
          CALL FRAME
          CALL PAXES(STEP, XMTR, VORD, NDES, LTITLE)
          CALL CTRORI(1.0)
          CALL POSITN(-1.75*XMTR, 0.325*VORD)
          CALL TYPECS('( RAD / RAD )', 13)
          CALL CTRORI(0.0)
          CALL THICK(1)
          CALL CTRSET(4)
          CALL PTPLOT(STEP, SNOGX, 1, NDES, 51)
          CALL THICK(2)
          CALL NSCURV(STEP, SNOGX, 1, NDES)
          CALL SYMBKY(STEP, VORD, XMTR, NDES, 25, 17, 51)
          CALL THICK(1)
          CALL CTRSET(4)
          CALL PTPLOT(STEP, SNCRX, 1, NDES, 52)
          CALL THICK(2)
          CALL NSCURV(STEP, SNCRX, 1, NDES)
          CALL SYMBKY(STEP, VORD, XMTR, NDES, 13, 17, 52)
    C
          GO TO 810
    C
      480 CALL SETPLS(STEP, XMTR, NDES, VSCALR, IUNITS, MOTN, NORT, LTITLE)
    C
          CALL CTRSET(2)
          CALL PLOTCS(STEP(1), 1.1*VSCALR, ' OUTPUT DEVIATION IN ', 21)
          CALL TYPECS('RADIAL DIRECTION', 16)
          CALL BROKEN(5, 10, 5, 10)
          CALL POSITN(STEP(1), ADEVR)
          CALL JOIN(STEP(NDES), ADEVR)
          CALL POSITN(STEP(1), -ADEVR)
          CALL JOIN(STEP(NDES), -ADEVR)
          CALL FULL
          CALL CTRSET(4)
          CALL THICK(1)
          CALL PTPLOT(STEP, ERPR, 1, NDES, 54)
          CALL THICK(3)
          CALL NSCURV(STEP, ERPR, 1, NDES)
          CALL THICK(1)
          CALL PTPLOT(STEP, ERNR, 1, NDES, 54)
          CALL THICK(3)
          CALL NSCURV(STEP, ERNR, 1, NDES)
    C
          CALL FRAME
```

Listing of PSODPLOTS at 17:26:38 on JUL 23, 1982 for CCid=MCB8

```
      NORT = 2
      CALL SETPLS(STEP, XMTR, NDES, VSCALT, IUNITS, MOTN, NORT, LTITLE)
      CALL CTRSET(2)
      CALL PLOTCS(STEP(1), 1.1*VSCALT, ' OUTPUT DEVIATION IN ', 21)
      CALL TYPECS('POLAR ANGLE', 11)
      CALL BROKEN(5, 10, 5, 10)
      CALL POSITN(STEP(1), ADEVT)
      CALL JOIN(STEP(NDES), ADEVT)
      CALL POSITN(STEP(1), -ADEVT)
      CALL JOIN(STEP(NDES), -ADEVT)
      CALL FULL
      CALL CTRSET(4)
      CALL THICK(1)
      CALL PTPLOT(STEP, ERPT, 1, NDES, 54)
      CALL THICK(3)
      CALL NSCURV(STEP, ERPT, 1, NDES)
      CALL THICK(1)
      CALL PTPLOT(STEP, ERNT, 1, NDES, 54)
      CALL THICK(3)
      CALL NSCURV(STEP, ERNT, 1, NDES)
C
      NLP = 0
      DO 510 J = 1, KKL
        IF (SNX(1,J) .LT. - 5.0D-1) GO TO 510
        IF (J .LE. MAXARC*NLP) GO TO 490
        CALL FRAME
        MAS = 49
        CALL YSCALE(SNX, VORD, NDES, KKL, MAXARC, J)
        CALL PAXES(STEP, XMTR, VORD, NDES, LTITLE)
        CALL PLOTCS(STEP(2), 1.05*VORD, 'SENSITIVITY IN POLAR RADIUS',
     1       27)
        CALL CTRORI(1.0)
        CALL POSITN(-1.75*XMTR, 0.325*VORD)
        IF (IUNITS .EQ. 1) CALL TYPECS('( MM / MM )', 11)
        IF (IUNITS .EQ. 2) CALL TYPECS('( IN / IN )', 11)
        CALL CTRORI(0.0)
  490   NLP = (J - 1) / MAXARC + 1
        NAR = J - (NLP - 1) * MAXARC
        MAS = MAS + 1
        DO 500 I = 1, NDES
          VAR(I) = SNX(I,J)
  500   CONTINUE
        CALL THICK(1)
        CALL CTRSET(4)
        CALL PTPLOT(STEP, VAR, 1, NDES, MAS)
        CALL THICK(2)
        CALL NSCURV(STEP, VAR, 1, NDES)
        CALL SYMKEY(STEP, VORD, XMTR, NDES, NAR, NLP, MAS, 11)
  510 CONTINUE
C
      YORD = 0.0
      DO 520 I = 1, NDES
        IF (SNOLX(I) .GT. YORD) YORD = SNOLX(I)
        IF (SNORLX(I) .GT. YORD) YORD = SNORLX(I)
  520 CONTINUE
      IF (YORD .LT. 1.0D-9) GO TO 540
      VORD = 1.1 * YORD
      CALL FRAME
```

```
      CALL PAXES(STEP, XMTR, VORD, NDES, LTITLE)
      CALL PLOTCS(STEP(2), 1.05*VORD, 'SENSITIVITY IN POLAR RADIUS', 27)
      CALL CTRORI(1.0)
      CALL POSITN(-1.75*XMTR, 0.325*VORD)
      IF (IUNITS .EQ. 1) CALL TYPECS('( MM / MM )', 11)
      IF (IUNITS .EQ. 2) CALL TYPECS('( IN / IN )', 11)
      CALL CTRORI(0.0)
      IF (SNOLX(1) .LT. - 5.0D-1) GO TO 530
      CALL CTRSET(4)
      CALL THICK(1)
      CALL PTPLOT(STEP, SNOLX, 1, NDES, 51)
      CALL THICK(2)
      CALL NSCURV(STEP, SNOLX, 1, NDES)
      CALL SYMBKY(STEP, VORD, XMTR, NDES, 25, 11, 51)
  530 IF (SNORLX(1) .LT. - 5.0D-1) GO TO 540
      CALL THICK(1)
      CALL CTRSET(4)
      CALL PTPLOT(STEP, SNORLX, 1, NDES, 52)
      CALL THICK(2)
      CALL NSCURV(STEP, SNORLX, 1, NDES)
      CALL SYMBKY(STEP, VORD, XMTR, NDES, 17, 11, 52)
C
  540 NLP = 0
      DO 570 J = 1, KKL
        IF (J .EQ. MAXARC*NLP + 1) NGG = 0
        IF (SNGX(1,J) .LT. - 5.0D-1) GO TO 570
        NGG = NGG + 1
        IF (J .LE. MAXARC*NLP .AND. NGG .NE. 1) GO TO 550
        CALL FRAME
        NAS = 49
        CALL YSCALE(SNGX, VORD, NDES, KKL, MAXARC, J)
        CALL PAXES(STEP, XMTR, VORD, NDES, LTITLE)
        CALL PLOTCS(STEP(2), 1.05*VORD, 'SENSITIVITY IN POLAR RADIUS',
     1       27)
        CALL CTRORI(1.0)
        CALL POSITN(-1.75*XMTR, 0.325*VORD)
        IF (IUNITS .EQ. 1) CALL TYPECS('( MM / RAD )', 12)
        IF (IUNITS .EQ. 2) CALL TYPECS('( IN / RAD )', 12)
        CALL CTRORI(0.0)
  550   NLP = (J - 1) / MAXARC + 1
        NAR = J - (NLP - 1) * MAXARC
        NAS = NAS + 1
        DO 560 I = 1, NDES
          VAR(I) = SNGX(I,J)
  560   CONTINUE
        CALL THICK(1)
        CALL CTRSET(4)
        CALL PTPLOT(STEP, VAR, 1, NDES, NAS)
        CALL THICK(2)
        CALL NSCURV(STEP, VAR, 1, NDES)
        CALL SYMKEY(STEP, VORD, XMTR, NDES, NAR, NLP, NAS, 17)
  570 CONTINUE
C
      YORD = 0.0
      DO 580 I = 1, NDES
        IF (SNOGX(I) .GT. YORD) YORD = SNOGX(I)
        IF (SNORGX(I) .GT. YORD) YORD = SNORGX(I)
        IF (SNCRX(I) .GT. YORD) YORD = SNCRX(I)
```

Listing of PSODPLOTS at 17:26:38 on JUL 23, 1982 for CCid=MCB8

```
  580 CONTINUE
      VORD = 1.1 * YORD
      CALL FRAME
      CALL PAXES(STEP, XMTR, VORD, NDES, LTITLE)
      CALL PLOTCS(STEP(2), 1.05*VORD, 'SENSITIVITY IN POLAR RADIUS', 27)
      CALL CTRORI(1.0)
      CALL POSITN(-1.75*XMTR, 0.325*VORD)
      IF (IUNITS .EQ. 1) CALL TYPECS('( MM / RAD )', 12)
      IF (IUNITS .EQ. 2) CALL TYPECS('( IN / RAD )', 12)
      CALL CTRORI(0.0)
      IF (SNOGX(1) .LT. - 5.0D-1) GO TO 590
      CALL THICK(1)
      CALL CTRSET(4)
      CALL PTPLOT(STEP, SNOGX, 1, NDES, 51)
      CALL THICK(2)
      CALL NSCURV(STEP, SNOGX, 1, NDES)
      CALL SYMBKY(STEP, VORD, XMTR, NDES, 25, 17, 51)
  590 IF (SNORGX(1) .LT. - 5.0D-1) GO TO 600
      CALL THICK(1)
      CALL CTRSET(4)
      CALL PTPLOT(STEP, SNORGX, 1, NDES, 52)
      CALL THICK(2)
      CALL NSCURV(STEP, SNORGX, 1, NDES)
      CALL SYMBKY(STEP, VORD, XMTR, NDES, 17, 17, 52)
  600 CALL THICK(1)
      CALL CTRSET(4)
      CALL PTPLOT(STEP, SNCRX, 1, NDES, 53)
      CALL THICK(2)
      CALL NSCURV(STEP, SNCRX, 1, NDES)
      CALL SYMBKY(STEP, VORD, XMTR, NDES, 13, 17, 53)
C
      NLP = 0
      DO 630 J = 1, KKL
        IF (SNY(1,J) .LT. - 5.0D-1) GO TO 630
        IF (J .LE. MAXARC*NLP) GO TO 610
        CALL FRAME
        MAS = 49
        CALL YSCALE(SNY, VORD, NDES, KKL, MAXARC, J)
        CALL PAXES(STEP, XMTR, VORD, NDES, LTITLE)
        CALL PLOTCS(STEP(2), 1.05*VORD, 'SENSITIVITY IN POLAR ANGLE',
     1        26)
        CALL CTRORI(1.0)
        CALL POSITN(-1.75*XMTR, 0.325*VORD)
        IF (IUNITS .EQ. 1) CALL TYPECS('( RAD / MM )', 12)
        IF (IUNITS .EQ. 2) CALL TYPECS('( RAD / IN )', 12)
        CALL CTRORI(0.0)
  610   NLP = (J - 1) / MAXARC + 1
        NAR = J - (NLP - 1) * MAXARC
        MAS = MAS + 1
        DO 620 I = 1, NDES
          VAR(I) = SNY(I,J)
  620   CONTINUE
        CALL THICK(1)
        CALL CTRSET(4)
        CALL PTPLOT(STEP, VAR, 1, NDES, MAS)
        CALL THICK(2)
        CALL NSCURV(STEP, VAR, 1, NDES)
        CALL SYMKEY(STEP, VORD, XMTR, NDES, NAR, NLP, MAS, 11)
```

```
  630 CONTINUE
C
      YORD = 0.0
      DO 640 I = 1, NDES
        IF (SNOLY(I) .GT. YORD) YORD = SNOLY(I)
        IF (SNORLY(I) .GT. YORD) YORD = SNORLY(I)
  640 CONTINUE
      IF (YORD .LT. 1.0D-9) GO TO 660
      VORD = 1.1 * YORD
      CALL PAXES(STEP, XMTR, VORD, NDES, LTITLE)
      CALL PLOTCS(STEP(2), 1.05*VORD, 'SENSITIVITY IN POLAR ANGLE', 26)
      CALL CTRORI(1.0)
      CALL POSITN(-1.75*XMTR, 0.325*VORD)
      IF (IUNITS .EQ. 1) CALL TYPECS('( RAD / MM )', 12)
      IF (IUNITS .EQ. 2) CALL TYPECS('( RAD / IN )', 12)
      CALL CTRORI(0.0)
      IF (SNOLY(1) .LT. - 5.0D-1) GO TO 650
      CALL CTRSET(4)
      CALL THICK(1)
      CALL PTPLOT(STEP, SNOLY, 1, NDES, 51)
      CALL THICK(2)
      CALL NSCURV(STEP, SNOLY, 1, NDES)
      CALL SYMBKY(STEP, VORD, XMTR, NDES, 25, 11, 51)
  650 IF (SNORLY(1) .LT. - 5.0D-1) GO TO 660
      CALL THICK(1)
      CALL CTRSET(4)
      CALL PTPLOT(STEP, SNORLY, 1, NDES, 52)
      CALL THICK(2)
      CALL NSCURV(STEP, SNORLY, 1, NDES)
      CALL SYMBKY(STEP, VORD, XMTR, NDES, 17, 11, 52)
C
  660 NLP = 0
      DO 690 J = 1, KKL
        IF (J .EQ. MAXARC*NLP + 1) NGG = 0
        IF (SNGY(1,J) .LT. - 5.0D-1) GO TO 690
        NGG = NGG + 1
        IF (J .LE. MAXARC*NLP .AND. NGG .NE. 1) GO TO 670
        CALL FRAME
        NAS = 49
        CALL YSCALE(SNGY, VORD, NDES, KKL, MAXARC, J)
        CALL PAXES(STEP, XMTR, VORD, NDES, LTITLE)
        CALL PLOTCS(STEP(2), 1.05*VORD, 'SENSITIVITY IN POLAR ANGLE',
     1       26)
        CALL CTRORI(1.0)
        CALL POSITN(-1.75*XMTR, 0.325*VORD)
        CALL TYPECS('( RAD / RAD )', 13)
        CALL CTRORI(0.0)
  670   NLP = (J - 1) / MAXARC + 1
        NAR = J - (NLP - 1) * MAXARC
        NAS = NAS + 1
        DO 680 I = 1, NDES
          VAR(I) = SNGY(I,J)
  680   CONTINUE
        CALL THICK(1)
        CALL CTRSET(4)
        CALL PTPLOT(STEP, VAR, 1, NDES, NAS)
        CALL THICK(2)
        CALL NSCURV(STEP, VAR, 1, NDES)
```

```
              CALL SYMKEY(STEP, VORD, XMTR, NDES, NAR, NLP, NAS, 17)
        690 CONTINUE
      C
            YORD = 0.0
            DO 700 I = 1, NDES
              IF (SNOGY(I) .GT. YORD) YORD = SNOGY(I)
              IF (SNORGY(I) .GT. YORD) YORD = SNORGY(I)
              IF (SNCRY(I) .GT. YORD) YORD = SNCRY(I)
        700 CONTINUE
            VORD = 1.1 * YORD
            CALL FRAME
            CALL PAXES(STEP, XMTR, VORD, NDES, LTITLE)
            CALL PLOTCS(STEP(2), 1.05*VORD, 'SENSITIVITY IN POLAR ANGLE', 26)
            CALL CTRORI(1.0)
            CALL POSITN(-1.75*XMTR, 0.325*VORD)
            CALL TYPECS('( RAD / RAD )', 13)
            CALL CTRORI(0.0)
            IF (SNOGY(1) .LT. - 5.0D-1) GO TO 710
            CALL THICK(1)
            CALL CTRSET(4)
            CALL PTPLOT(STEP, SNOGY, 1, NDES, 51)
            CALL THICK(2)
            CALL NSCURV(STEP, SNOGY, 1, NDES)
            CALL SYMBKY(STEP, VORD, XMTR, NDES, 25, 17, 51)
        710 IF (SNORGY(1) .LT. - 5.0D-1) GO TO 720
            CALL THICK(1)
            CALL CTRSET(4)
            CALL PTPLOT(STEP, SNORGY, 1, NDES, 52)
            CALL THICK(2)
            CALL NSCURV(STEP, SNORGY, 1, NDES)
            CALL SYMBKY(STEP, VORD, XMTR, NDES, 17, 17, 52)
        720 CALL THICK(1)
            CALL CTRSET(4)
            CALL PTPLOT(STEP, SNCRY, 1, NDES, 53)
            CALL THICK(2)
            CALL NSCURV(STEP, SNCRY, 1, NDES)
            CALL SYMBKY(STEP, VORD, XMTR, NDES, 13, 17, 53)
      C
            GO TO 810
      C
        730 CALL SETPLS(STEP, XMTR, NDES, VSCALS, IUNITS, MOTN, NORT, LTITLE)
      C
            CALL CTRSET(2)
            CALL BROKEN(5, 10, 5, 10)
            CALL POSITN(STEP(1), ADEVS)
            CALL JOIN(STEP(NDES), ADEVS)
            CALL POSITN(STEP(1), -ADEVS)
            CALL JOIN(STEP(NDES), -ADEVS)
            CALL FULL
            CALL CTRSET(4)
            CALL THICK(1)
            CALL PTPLOT(STEP, ERPS, 1, NDES, 54)
            CALL THICK(3)
            CALL NSCURV(STEP, ERPS, 1, NDES)
            CALL THICK(1)
            CALL PTPLOT(STEP, ERNS, 1, NDES, 54)
            CALL THICK(3)
            CALL NSCURV(STEP, ERNS, 1, NDES)
```

Listing of PSODPLOTS at 17:26:38 on JUL 23, 1982 for CCid=MCB8

```
C
      NLP = 0
      DO 760 J = 1, KKL
        IF (SNX(1,J) .LT. - 5.0D-1) GO TO 760
        IF (J .LE. MAXARC*NLP) GO TO 740
        CALL FRAME
        MAS = 49
        CALL YSCALE(SNX, VORD, NDES, KKL, MAXARC, J)
        CALL PAXES(STEP, XMTR, VORD, NDES, LTITLE)
        CALL CTRORI(1.0)
        CALL POSITN(-1.75*XMTR, 0.325*VORD)
        IF (IUNITS .EQ. 1) CALL TYPECS('( MM / MM )', 11)
        IF (IUNITS .EQ. 2) CALL TYPECS('( IN / IN )', 11)
        CALL CTRORI(0.0)
  740   NLP = (J - 1) / MAXARC + 1
        NAR = J - (NLP - 1) * MAXARC
        MAS = MAS + 1
        DO 750 I = 1, NDES
          VAR(I) = SNX(I,J)
  750   CONTINUE
        CALL THICK(1)
        CALL CTRSET(4)
        CALL PTPLOT(STEP, VAR, 1, NDES, MAS)
        CALL THICK(2)
        CALL NSCURV(STEP, VAR, 1, NDES)
        CALL SYMKEY(STEP, VORD, XMTR, NDES, NAR, NLP, MAS, 11)
  760 CONTINUE
C
      NLP = 0
      DO 790 J = 1, KKL
        IF (J .EQ. MAXARC*NLP + 1) NGG = 0
        IF (SNGX(1,J) .LT. - 5.0D-1) GO TO 790
        NGG = NGG + 1
        IF (J .LE. MAXARC*NLP .AND. NGG .NE. 1) GO TO 770
        CALL FRAME
        MAS = 49
        CALL YSCALE(SNGX, VORD, NDES, KKL, MAXARC, J)
        CALL PAXES(STEP, XMTR, VORD, NDES, LTITLE)
        CALL CTRORI(1.0)
        CALL POSITN(-1.75*XMTR, 0.325*VORD)
        IF (IUNITS .EQ. 1) CALL TYPECS('( MM / RAD )', 12)
        IF (IUNITS .EQ. 2) CALL TYPECS('( IN / RAD )', 12)
        CALL CTRORI(0.0)
  770   NLP = (J - 1) / MAXARC + 1
        NAR = J - (NLP - 1) * MAXARC
        MAS = MAS + 1
        DO 780 I = 1, NDES
          VAR(I) = SNGX(I,J)
  780   CONTINUE
        CALL THICK(1)
        CALL CTRSET(4)
        CALL PTPLOT(STEP, VAR, 1, NDES, MAS)
        CALL THICK(2)
        CALL NSCURV(STEP, VAR, 1, NDES)
        CALL SYMKEY(STEP, VORD, XMTR, NDES, NAR, NLP, MAS, 17)
  790 CONTINUE
C
      YORD = 0.0
```

```
          DO 800 I = 1, NDES
            IF (SNCRX(I) .GT. YORD) YORD = SNCRX(I)
      800 CONTINUE
          VORD = 1.1 * YORD
          CALL FRAME
          CALL PAXES(STEP, XMTR, VORD, NDES, LTITLE)
          CALL CTRORI(1.0)
          CALL POSITN(-1.75*XMTR, 0.325*VORD)
          IF (IUNITS .EQ. 1) CALL TYPECS('( MM / RAD )', 12)
          IF (IUNITS .EQ. 2) CALL TYPECS('( IN / RAD )', 12)
          CALL CTRORI(0.0)
          CALL THICK(1)
          CALL CTRSET(4)
          CALL PTPLOT(STEP, SNCRX, 1, NDES, 51)
          CALL THICK(2)
          CALL NSCURV(STEP, SNCRX, 1, NDES)
          CALL SYMBKY(STEP, VORD, XMTR, NDES, 13, 17, 51)
C
C
      810 CALL CTRSET(1)
          CALL THICK(1)
          CALL GREND
          STOP
          END
C
C   *    *    *    *    *    *    *    *    *    *    *    *    *    *    *    *    *
C
          SUBROUTINE SETPLS(STEP, XMTR, NDES, VSCALE, IUNITS, MOTN, NORT,
         1             LTITLE)
C   --------------------------------------------------------------------
C
          IMPLICIT LOGICAL*1(L)
          DIMENSION LTITLE(72)
          REAL STEP(NDES)
C
          CALL CTRSET(1)
          CALL PSPACE(0.2, 0.85, 0.15, 0.75)
          CALL WINDOW(STEP(1), STEP(NDES), -VSCALE, VSCALE)
          CALL MAP(STEP(1), STEP(NDES), -VSCALE, VSCALE)
          CALL BORDER
          CALL THICK(2)
          CALL PLOTCS(STEP(1), 1.2*VSCALE, LTITLE, 72)
          CALL CTRSET(2)
          CALL AXES
          CALL PLOTCS(STEP(1) + 2.0*XMTR, -0.15*VSCALE, 'INPUT POSITION',
         1     14)
          CALL TYPECS(' (DEGREES)', 10)
          CALL CTRORI(1.0)
          CALL PLOTCS(STEP(1) - 1.5*XMTR, -0.8*VSCALE, 'OUTPUT DEVIATION',
         1     16)
          GO TO (10, 20, 30, 10), MOTN
       10 IF (IUNITS .EQ. 1) CALL TYPECS(' (MILLIMETERS)', 14)
          IF (IUNITS .EQ. 2) CALL TYPECS(' ( INCHES )', 11)
          GO TO 40
       20 CALL TYPECS(' (-DEGREES )', 12)
          GO TO 40
       30 IF (NORT .EQ. 1) GO TO 10
          GO TO 20
```

```
   40 CALL CTRORI(0.0)
      CALL WINDOW(STEP(1), STEP(NDES), -1.2*VSCALE, VSCALE)
      CALL POSITN(STEP(1) + 0.5*XMTR, -1.2*VSCALE)
      CALL BROKEN(5, 10, 5, 10)
      CALL JOIN(STEP(1) + 1.5*XMTR, -1.2*VSCALE)
      CALL TYPECS(' MAX. ALLOWABLE OUTPUT DEVIATION', 32)
      CALL FULL
      CALL THICK(2)
C
      RETURN
      END
C
C   *   *   *   *   *   *   *   *   *   *   *   *   *   *
C
      SUBROUTINE PAXES(STEP, XMTR, VORD, NDES, LTITLE)
C
      IMPLICIT LOGICAL*1(L)
      DIMENSION STEP(NDES), LTITLE(72)
C
      CALL CTRSET(1)
      CALL THICK(1)
      CALL PSPACE(0.20, 0.80, 0.20, 0.80)
      CALL WINDOW(STEP(1), STEP(NDES), 0.0, VORD)
      CALL MAP(STEP(1), STEP(NDES), 0.0, VORD)
      CALL BORDER
      CALL THICK(2)
      CALL PLOTCS(STEP(1), 1.1*VORD, LTITLE, 72)
      CALL CTRSET(2)
      CALL UNDLIN(1)
      CALL PLOTCS(STEP(NDES) + XMTR, VORD, 'KEY :-', 6)
      CALL UNDLIN(0)
      CALL AXES
      CALL PLOTCS(1.5*XMTR, -0.1*VORD, 'INPUT POSITION ( DEGREES )', 26)
      CALL CTRORI(1.0)
      CALL PLOTCS(-2.75*XMTR, 0.25*VORD, 'OUTPUT SENSITIVITY', 18)
      CALL PLOTCS(-2.25*XMTR, 0.2*VORD, 'TO PARAMETER DEVIATION', 22)
      CALL CTRORI(0.0)
C
      RETURN
      END
C
C   *   *   *   *   *   *   *   *   *   *   *   *   *   *
C
      SUBROUTINE YSCALE(SNU, VORD, NDS, KKL, MXR, NUM)
C
      DIMENSION SNU(37,100)
C
      LK = NUM / MXR
      NAM = (LK + 1) * MXR
      SMX = 0.0
      DO 10 J = NUM, NAM
        DO 10 I = 1, NDS
          IF (SNU(I,J) .GT. SMX) SMX = SNU(I,J)
   10 CONTINUE
C
      VORD = 1.1 * SMX
      RETURN
      END
```

Listing of PSODPLOTS at 17:26:38 on JUL 23, 1982 for CCid=MCB8

```
C
C    *    *    *    *    *    *    *    *    *    *    *    *    *    *
C
      SUBROUTINE SYMKEY(STEP, VORD, XMTR, NDES, NR, NL, NS, NC)
C
      DIMENSION STEP(NDES)
C
      PN = FLOAT(NS - 49)
      PS = VORD - 0.125 * PN * VORD
      RS = STEP(NDES) + 1.5 * XMTR
      CALL CTRSET(4)
      CALL THICK(1)
      CALL PLOTNC(RS, PS, NS)
      CALL HSPACE(5)
      CALL THICK(2)
      IF (NC .EQ. 11) CALL CTRSET(2)
      CALL TYPENC(NC)
      CALL SUFFIX
      CALL TYPENI(NR)
      CALL TYPENC(38)
      CALL TYPENI(NL)
      CALL CTRSET(2)
      CALL NORMAL
C
      RETURN
      END
C
C    *    *    *    *    *    *    *    *    *    *    *    *    *    *
C
      SUBROUTINE SYMBKY(STEP, VORD, XMTR, NDES, NS, NC, MS)
C
      DIMENSION STEP(NDES)
C
      PN = FLOAT(MS - 50)
      PS = VORD - 0.125 * PN * VORD
      RS = STEP(NDES) + 1.5 * XMTR
      CALL CTRSET(4)
      CALL THICK(1)
      CALL PLOTNC(RS, PS, MS)
      CALL HSPACE(5)
      CALL THICK(2)
      IF (NC .EQ. 11) CALL CTRSET(2)
      CALL TYPENC(NC)
      CALL SUFFIX
      CALL CTRSET(2)
      CALL TYPENC(NS)
      CALL NORMAL
C
      RETURN
      END
```

APPENDIX IIA

KINEMATIC ANALYSIS OF A FOUR-BAR LINKAGE

IIA.1 DISPLACEMENTS

With reference to Fig. 8.4, the vector equation for the polygon with arcs a_1, a_2, a_3 and a_4 may be expressed as

$$a_1\, e^{i\theta_1} + a_2\, e^{i\theta_2} - a_3\, e^{i\theta_3} - a_4\, e^{i\theta_4} = 0 \qquad \text{IIA.1}$$

Expanding and separating the real and imaginary parts gives

$$\begin{aligned} a_1 \cos\theta_1 + a_2 \cos\theta_2 - a_3 \cos\theta_3 - a_4 \cos\theta_4 &= 0 \\ a_1 \sin\theta_1 + a_2 \sin\theta_2 - a_3 \sin\theta_3 - a_4 \sin\theta_4 &= 0 \end{aligned} \qquad \text{IIA.2}$$

Eliminating θ_2 gives

$$R_1 \sin\theta_3 + R_2 \cos\theta_3 - R_3 = 0 \qquad \text{IIA.3}$$

where

$$R_1 = \sin\theta_1 - \frac{a_4}{a_1} \sin\theta_4$$

$$R_2 = \cos\theta_1 - \frac{a_4}{a_1} \cos\theta_4$$

$$R_3 = \frac{a_1^2 - a_2^2 + a_3^2 + a_4^2}{2a_1a_3} - \frac{a_1}{a_3} \cos(\theta_1 - \theta_4)$$

Substituting for $\sin\theta_3$ and $\cos\theta_3$ in terms of $\tan\frac{\theta_3}{2}$ and rearranging gives

$$(R_2 + R_3) \tan^2\frac{\theta_3}{2} - 2 R_1 \tan\frac{\theta_3}{2} - (R_2 - R_3) = 0$$

IIA.4

Hence

$$\tan\frac{\theta_3}{2} = \frac{R_1 \pm \sqrt{R_1^2 + R_2^2 - R_3^2}}{R_2 + R_3} \qquad \text{IIA.5}$$

The configuration shown in Fig. 8.4 corresponds to the negative square root,

$$\theta_3 = 2\tan^{-1}\frac{R_1 - \sqrt{R_1^2 + R_2^2 - R_3^2}}{R_2 + R_3} \qquad \text{IIA.6}$$

Substituting in IIA.2 and rearranging gives

$$\theta_2 = \tan^{-1}\frac{-a_1\sin\theta_1 + a_3\sin\theta_3 + a_4\sin\theta_4}{-a_1\cos\theta_1 + a_3\cos\theta_3 + a_4\cos\theta_4} \qquad \text{IIA.7}$$

IIA.2 VELOCITIES

Differentiating IIA.1 w.r.t. time gives

$$a_1\dot{\theta}_1 e^{i\theta_1} + a_2\dot{\theta}_2 e^{i\theta_2} - a_3\dot{\theta}_3 e^{i\theta_3} = 0 \qquad \text{IIA.8}$$

Rewriting in component form

$$a_1\dot{\theta}_1\cos\theta_1 + a_2\dot{\theta}_2\cos\theta_2 - a_3\dot{\theta}_3\cos\theta_3 = 0$$

$$\qquad \text{IIA.9}$$

$$a_1\dot{\theta}_1\sin\theta_1 + a_2\dot{\theta}_2\sin\theta_2 - a_3\dot{\theta}_3\sin\theta_3 = 0$$

Eliminating $\dot{\theta}_2$ gives after rearranging

$$\dot{\theta}_3 = \frac{a_1 \sin(\theta_2 - \theta_1)}{a_3 \sin(\theta_2 - \theta_3)} \dot{\theta}_1 \qquad \text{IIA.10}$$

Eliminating $\dot{\theta}_3$ from equations IIA.9 and rearranging gives

$$\dot{\theta}_2 = \frac{a_1 \sin(\theta_3 - \theta_1)}{a_2 \sin(\theta_2 - \theta_3)} \dot{\theta}_1 \qquad \text{IIA.11}$$

IIA.3 ACCELERATIONS

Differentiating IIA.10 with respect to time gives

$$\ddot{\theta}_3 = \frac{a_1 \cos(\theta_2 - \theta_1)}{a_3 \sin(\theta_2 - \theta_3)} \left[\dot{\theta}_2 - \dot{\theta}_1\right]\dot{\theta}_1$$

$$- \frac{a_1 \sin(\theta_2 - \theta_1)\cos(\theta_2 - \theta_3)}{a_3 \sin^2(\theta_2 - \theta_3)} \left[\dot{\theta}_2 - \dot{\theta}_3\right]\dot{\theta}_1$$

$$+ \frac{a_1 \sin(\theta_2 - \theta_1)}{a_3 \sin(\theta_2 - \theta_3)} \ddot{\theta}_1$$

Substituting from IIA.10 and IIA.11 then for $\dot{\theta}_1$ = constant

$$\ddot{\theta}_3 = \dot{\theta}_3 \left[\frac{\dot{\theta}_2 - \dot{\theta}_1}{\tan(\theta_2 - \theta_1)} - \frac{\dot{\theta}_2 - \dot{\theta}_3}{\tan(\theta_2 - \theta_3)}\right] \qquad \text{IIA.12}$$

Differentiating IIA.11 and substituting as above gives

$$\ddot{\theta}_2 = \dot{\theta}_2 \left[\frac{\dot{\theta}_3 - \dot{\theta}_1}{\tan(\theta_3 - \theta_1)} - \frac{\dot{\theta}_2 - \dot{\theta}_3}{\tan(\theta_2 - \theta_3)} \right] \qquad \text{IIA.13}$$

IIA.4 MOTION OF MASS CENTRES

IIA.4.1 Displacements

Link 1 : Crank

$$x_1 = \rho_1 \cos(\theta_1 + \lambda_1)$$
$$y_1 = \rho_1 \sin(\theta_1 + \lambda_1)$$

Link 2 : Coupler

$$x_2 = a_1 \cos\theta_1 + \rho_2 \cos(\theta_2 + \lambda_2)$$
$$y_2 = a_1 \sin\theta_1 + \rho_2 \sin(\theta_2 + \lambda_2)$$

Link 3 : Follower

$$x_3 = \rho_3 \cos(\theta_3 + \lambda_3)$$
$$y_3 = \rho_3 \sin(\theta_3 + \lambda_3)$$

IIA.14

IIA.4.2 Velocities

Differentiating equations IIA.14 and substituting for $\dot{\theta}_2$ and $\dot{\theta}_3$ where applicable gives:

$$\dot{x}_1 = -\rho_1\dot{\theta}_1 \sin(\theta_1 + \lambda_1)$$
$$\dot{y}_1 = \rho_1\dot{\theta}_1 \cos(\theta_1 + \lambda_1)$$

IIA.15

$$\dot{x}_2 = -a_1\dot{\theta}_1 \sin\theta_1 - \frac{\rho_2 a_1 \dot{\theta}_1 \sin(\theta_3 - \theta_1)}{a_2 \sin(\theta_2 - \theta_3)} \sin(\theta_2 + \lambda_2)$$

$$\dot{y}_2 = a_1\dot{\theta}_1 \cos\theta_1 + \frac{\rho_2 a_1 \dot{\theta}_1 \sin(\theta_3 - \theta_1)}{a_2 \sin(\theta_2 - \theta_3)} \cos(\theta_2 + \lambda_2)$$

$$\dot{x}_3 = -\rho_3\dot{\theta}_1 \frac{a_1 \sin(\theta_2 - \theta_1)}{a_3 \sin(\theta_2 - \theta_3)} \sin(\theta_3 + \lambda_3)$$

$$\dot{y}_3 = \rho_3\dot{\theta}_1 \frac{a_1 \sin(\theta_2 - \theta_1)}{a_3 \sin(\theta_2 - \theta_3)} \cos(\theta_3 + \lambda_3)$$

IIA.15

IIA.4.3 Accelerations

Differentiating equations IIA.14 twice gives

$$\ddot{x}_1 = -\rho_1\dot{\theta}_1^2 \cos(\theta_1 + \lambda_1)$$

$$\ddot{y}_1 = -\rho_1\dot{\theta}_1^2 \sin(\theta_1 + \lambda_1)$$

$$\ddot{x}_2 = -a_1\dot{\theta}_1^2\cos\theta_1 - \rho_2\dot{\theta}_2^2\cos(\theta_2 + \lambda_2) - \rho_2\ddot{\theta}_2\sin(\theta_2 + \lambda_2)$$

$$\ddot{y}_2 = -a_1\dot{\theta}_1^2\sin\theta_1 - \rho_2\dot{\theta}_2^2\sin(\theta_2 + \lambda_2) + \rho_2\ddot{\theta}_2\cos(\theta_2 + \lambda_2)$$

$$\ddot{x}_3 = -\rho_3\dot{\theta}_3^2\cos(\theta_3 + \lambda_3) - \rho_3\ddot{\theta}_3\sin(\theta_3 + \lambda_3)$$

$$\ddot{y}_3 = -\rho_3\dot{\theta}_3^2\sin(\theta_3 + \lambda_3) + \rho_3\ddot{\theta}_3\cos(\theta_3 + \lambda_3)$$

IIA.16

APPENDIX IIB

DYNAMIC ANALYSIS OF A FOUR-BAR LINKAGE

IIB.1 TERMINOLOGY - Refer to Fig. 8.5

a_i = vector connecting the joints on link i - i.e. arc lengths

ρ_i, λ_i = polar coordinates (radius and angle respectively) defining mass centre position of link i, relative to local origin and arc length vector.

I_i = second moment of mass of link i about local origin.

F_{ij_x}, F_{ij_y} = cartesian components along the x- and y-direction respectively of the joint force exerted by link i on j.

F_{ei}, M_{ei} = Resultant external force and moment respectively acting on link i

θ_i = angular orientation of link i relative to the x-direction.

ρ_i', λ_i' = polar coordinates of mass centre from the other joint on the link as seen on Fig. 8.5

IIB.2 EQUATIONS OF MOTION

With reference to the free body diagrams in Fig. 8.5, the following equations of motion (force equations along the cartesian coordinates and moment equation about the centre of

~~of~~ mass) may be written for the respective moving links.

Link 1

$$F_{e1_x} + F_{21_x} + F_{41_x} = m_1 \ddot{x}_1$$

$$F_{e1_y} + F_{21_y} + F_{41_y} = m_1 \ddot{y}_1 \qquad \text{IIB.1}$$

$$F_{41_x}\rho_1\sin(\theta_1+\lambda_1) - F_{41_y}\rho_1\cos(\theta_1+\lambda_1) - F_{21_x}\rho_1\sin(\theta_1-\lambda_1)$$

$$+ F_{21_y}\rho_1\cos(\theta_1-\lambda_1) + M_{e1} = (I_1 - m_1\rho_1^2)\ddot{\theta}_1$$

Link 2

$$F_{e2_x} + F_{12_x} + F_{32_x} = m_2 \ddot{x}_2$$

$$F_{e2_y} + F_{12_y} + F_{32_y} = m_2 \ddot{y}_2 \qquad \text{IIB.2}$$

$$F_{12_x}\rho_2\sin(\theta_2+\lambda_2) - F_{12_y}\rho_2\cos(\theta_2+\lambda_2) - F_{32_x}\rho_2\sin(\theta_2-\lambda_2)$$

$$+ F_{32_y}\rho_2\cos(\theta_2-\lambda_2) + M_{32} = (I_2 - m_2\rho_2^2)\ddot{\theta}_2$$

Link 3

$$F_{e3_x} + F_{23_x} + F_{43_x} = m_3 \ddot{x}_3$$

$$F_{e3_y} + F_{23_y} + F_{43_y} = m_3 \ddot{y}_3 \qquad \text{IIB.3}$$

$$F_{43_x}\rho_3\sin(\theta_3+\lambda_3) - F_{43_y}\rho_3\cos(\theta_3+\lambda_3) - F_{23_x}\rho_3\sin(\theta_3-\lambda_3)$$

$$+ F_{23_y}\rho_3\cos(\theta_3-\lambda_3) + M_{e3} = (I_3 - m_3\rho_3^2)\ddot{\theta}_3$$

IIB.3 JOINT FORCES

Note the joint forces F_{ij} and F_{ji} are equal and opposite, i.e. $F_{ij} = -F_{ji}$. Hence substituting for F_{12_x} and F_{12_y} from the force equations into the moment equation of set IIB.2 gives;

$$F_{23_x}\left[\rho_2 \sin(\theta_2 + \lambda_2) + \rho_2' \sin(\theta_2 - \lambda_2')\right]$$

$$-F_{23_y}\left[\rho_2 \cos(\theta_2 + \lambda_2) + \rho_2' \cos(\theta_2 - \lambda_2')\right]$$

$$-F_{e2_x}\, \rho_2 \sin(\theta_2 + \lambda_2) + F_{e2_y}\, \rho_2 \cos(\theta_2 + \lambda_2) + M_{ez}$$

$$= (I_2 - m_2\rho_2^2)\ddot{\theta}_2 - m_2\rho_2\left[\ddot{x}_2 \sin(\theta_2 + \lambda_2) - \ddot{y}_2 \cos(\theta_2 + \lambda_2)\right]$$

IIB.4

Note, with reference to Fig. 8.5, that

$$\rho_2 \sin(\theta_2 + \lambda_2) + \rho_2' \sin(\theta_2 - \lambda_2') = a_2 \sin\theta_2$$

and

$$\rho_2 \cos(\theta_2 + \lambda_2) + \rho_2' \cos(\theta_2 - \lambda_2') = a_2 \cos\theta_2$$

and from Appendix IIA.

$$\ddot{x}_2 \sin(\theta_2 + \lambda_2) - \ddot{y}_2 \cos(\theta_2 + \lambda_2)$$

$$= -a_1\, \dot{\theta}_1^2 \sin(\theta_2 + \lambda_2 - \theta_1) - \rho_2\, \ddot{\theta}_2$$

Substituting in IIB.4 gives

$$F_{23_x} a_2 \sin\theta_2 - F_{23_y} a_2 \cos\theta_2 - F_{e2_x}\rho_2 \sin(\theta_2 + \lambda_2)$$

$$+ F_{e2_y} \rho_2 \cos(\theta_2 + \lambda_2) + M_{e2}$$

$$= I_2\ddot{\theta}_2 + m_2\rho_2 a_1\dot{\theta}_1^2 \sin(\theta_2 + \lambda_2 - \theta_1) \qquad \text{IIB.5}$$

Similarly the set of equations IIB.3 gives

$$-F_{23_x} a_3 \sin\theta_3 + F_{23_y} a_3 \cos\theta_3 - F_{e3}\rho_3 \sin(\theta_3 + \lambda_3)$$

$$+ F_{e3_y} \rho_3 \cos(\theta_3 + \lambda_3) + M_{e3}$$

$$= I_3 \ddot{\theta}_3 \qquad \text{IIB.6}$$

Let

$$P = I_2\ddot{\theta}_2 + m_2\rho_2 a_1\dot{\theta}_1^2 \sin(\theta_2 + \lambda_2 - \theta_1)$$

$$Q = I_3\ddot{\theta}_3$$

If the external forces and moments are zero or negligible then equations IIB.5, IIB.6 become

$$F_{23_x} a_2 \sin\theta_2 - F_{23_y} a_2 \cos\theta_2 = P$$

$$-F_{23_x} a_3 \sin\theta_3 + F_{23_y} a_3 \cos\theta_3 = Q$$

Solving these two equations simultaneously gives

$$F_{23_x} = \frac{P \cos\theta_3}{a_2 \sin\mu} + \frac{Q \cos\theta_2}{a_3 \sin\mu}$$

$$F_{23_y} = \frac{P \sin\theta_3}{a_3 \sin\mu} + \frac{Q \sin\theta_2}{a_3 \sin\mu} \qquad \text{IIB.7}$$

where $\mu = \theta_2 - \theta_3$

Hence

$$F_{43_x} = m_3 \ddot{x}_3 - F_{23_x}$$

$$F_{43_y} = m_3 \ddot{y}_3 - F_{23_y} \qquad \text{IIB.8}$$

$$F_{12_x} = m_2 \ddot{x}_2 + F_{23_x}$$

$$F_{12_y} = m_2 \ddot{y}_2 + F_{23_y} \qquad \text{IIB.9}$$

$$F_{41_x} = m_1 \ddot{x}_1 + m_2 \ddot{x}_2 + F_{23_x}$$

$$F_{41_y} = m_1 \ddot{y}_1 + m_2 \ddot{y}_2 + F_{23_y} \qquad \text{IIB.10}$$

Substituting for $\ddot{x}_i$ and $\ddot{y}_i$ from Appendix IIA

$$F_{43_x} = -m_3\rho_3\left[\dot{\theta}_3^2\cos(\theta_3 + \lambda_3) + \ddot{\theta}_3\sin(\theta_3 + \lambda_3)\right] - F_{23_x}$$

$$F_{43_y} = -m_3\rho_3\left[\dot{\theta}_3^2\sin(\theta_3 + \lambda_3) - \ddot{\theta}_3\cos(\theta_3 + \lambda_3)\right] - F_{23_x}$$

IIB.11

$$F_{12_x} = -m_2\left\{a_1\dot{\theta}_1^2\cos\theta_1 + \rho_2\left[\dot{\theta}_2^2\cos(\theta_2+\lambda_2) + \ddot{\theta}_2\sin(\theta_2+\lambda_2)\right]\right\} + F_{23_x}$$

$$F_{12_y} = -m_2\left\{a_1\dot{\theta}_1^2\sin\theta_1 + \rho_2\left[\dot{\theta}_2^2\sin(\theta_2+\lambda_2) - \ddot{\theta}_2\cos(\theta_2+\lambda_2)\right]\right\} + F_{23_y}$$

$$F_{41_x} = -m_1\rho_1\dot{\theta}_1^2\cos(\theta_1 + \lambda_1) + F_{12_x}$$

IIB.13

$$F_{41_y} = -m_1\rho_1\dot{\theta}_1^2\sin(\theta_1 + \lambda_1) + F_{12_y}$$

The resultant joint force is given by

$$F_{ij} = \sqrt{F_{ij_x}^{\ 2} + F_{ij_y}^{\ 2}} \qquad \text{IIB.14}$$

i and j taking the respective values above.

The forces are represented by their vector components in Fig. 8.2.

A P P E N D I X   I I C

LISTING OF COMPUTER PROGRAM

<u>CONTACT</u>

Listing of CONTACT at 16:17:27 on JUL 23, 1982 for CCid=MCB8

```
      C  ****    ***    ***    ***    ****    ***    ***    ***    ****
      C  ***                                                        ***
      C  **   PROGRAM FOR OPTIMIZING MASS DISTRIBUTION TO MAINTAIN   **
      C  **   CONTACT IN THE JOINTS OF FOUR-BAR LINKAGE MECHANISMS   **
      C  ***                                                        ***
      C  ****    ***    ***    ***    ****    ***    ***    ***    ****
      C
            IMPLICIT REAL*8(A-H,O-Z)
      C
            DIMENSION X(15),G(15),W(135),BL(15),BU(15),DELTA(15),
           *    HESD(15),HESL(105)
            INTEGER IW(2),ISTATE(15),ITITLE(18)
            LOGICAL LOCSCH
      C
            COMMON ARC1,ARC2,ARC3,ARC4,CTA4,CTA1(361),CTA1D,PI,AMASO1,
           *      AMASO2,AMASO3,RRO1,RRO2,RRO3,ALMDO1,ALMDO2,
           *      ALMDO3,AIO1,AIO2,AIO3,M
      C
            EXTERNAL E04JBQ,FUNCT,MONIT
            CALL FTNCMD('ASSIGN 7=JOINT(*L+1);')
            CALL TIME(0)
      C
            PI=4.0D0*DATAN(1.0D0)
            DEGRAD=PI/180.0D0
      C
            N=15
            LH=105
            LIW=2
            LW=135
      C
            IPRINT=1000
            MAXCAL=40*N*(N+5)
            ETA=5.0D-01
            XTOL=1.0D-03
            STEPMX=5.0D+00
      C
            READ(5,1005) ITITLE
            READ(5,1010) ARC1,ARC2,ARC3,ARC4,CTA4,CTA1D,M
            READ(5,1015) AMASO1,RRO1,ALMDO1,AIO1,AMASO2,RRO2,ALMDO2,AIO2,
           *            AMASO3,RRO3,ALMDO3,AIO3
            READ(5,1015) (X(I),I=1,15)
            READ(5,1015)(BL(I),BU(I),I=1,15)
            WRITE(6,1110)
            WRITE(6,1005) ITITLE
      C
      C-------CHANGE LENGTHS FROM MILLIMETERS TO METERS
      C
            CTOM=1.0D-3
            ARC1=ARC1*CTOM
            ARC2=ARC2*CTOM
            ARC3=ARC3*CTOM
            ARC4=ARC4*CTOM
            RRO1=RRO1*CTOM
            RRO2=RRO2*CTOM
            RRO3=RRO3*CTOM
            X(2)=X(2)*CTOM
            X(5)=X(5)*CTOM
            X(7)=X(7)*CTOM
```

```
      X(9)=X(9)*CTOM
      X(11)=X(11)*CTOM
      BL(2)=BL(2)*CTOM
      BU(2)=BU(2)*CTOM
      BL(5)=BL(5)*CTOM
      BU(5)=BU(5)*CTOM
      BL(7)=BL(7)*CTOM
      BU(7)=BU(7)*CTOM
      BL(9)=BL(9)*CTOM
      BU(9)=BU(9)*CTOM
      BL(11)=BL(11)*CTOM
      BU(11)=BU(11)*CTOM
C
C-------CHANGE ANGLES FROM DEGREES TO RADIANS
C
      CTA4=CTA4*DEGRAD
      X(3)=X(3)*DEGRAD
      X(6)=X(6)*DEGRAD
      X(10)=X(10)*DEGRAD
      X(12)=X(12)*DEGRAD
      X(13)=X(13)*DEGRAD
      X(14)=X(14)*DEGRAD
      X(15)=X(15)*DEGRAD
      BL(3)=BL(3)*DEGRAD
      BU(3)=BU(3)*DEGRAD
      BL(6)=BL(6)*DEGRAD
      BU(6)=BU(6)*DEGRAD
      BL(10)=BL(10)*DEGRAD
      BU(10)=BU(10)*DEGRAD
      BL(12)=BL(12)*DEGRAD
      BU(12)=BU(12)*DEGRAD
      BL(13)=BL(13)*DEGRAD
      BU(13)=BU(13)*DEGRAD
      BL(14)=BL(14)*DEGRAD
      BU(14)=BU(14)*DEGRAD
      BL(15)=BL(15)*DEGRAD
      BU(15)=BU(15)*DEGRAD
C
        DO 20 I=1,M
          MI = I-1
          FMI = DFLOAT(MI)
          FMM = DFLOAT(M)
          CTA1(I)=FMI*2.0D0*PI/FMM
   20   CONTINUE
C
      IFAIL=1
C
      CALL E04HBF(N,FUNCT,X,J,DELTA,HESL,LH,HESD,F,G,IW,
     *            LIW,W,LW,IFAIL)
C
      WRITE(6,1020) J
C
      BIG=HESD(1)
      SMALL=HESD(1)
C
        DO 30 I=2,N
          IF(BIG .LT. HESD(I)) BIG=HESD(I)
          IF(SMALL .GT. HESD(I))SMALL=HESD(I)
```

Listing of CONTACT at 16:17:27 on JUL 23, 1982 for CCid=MCB8

```
   30  CONTINUE
C
      IF(BIG .LT. 1.0D+6*SMALL)GOTO 35
C
      WRITE(6,1030) (HESD(I),I=1,N)
      GOTO 60
C
   35 LOCSCH=.FALSE.
      INTYPE=1
      FEST=0.0D0
      IBOUND=0
C
      CALL E04JBF(N,FUNCT,MONIT,IPRINT,LOCSCH,INTYPE,E04JBQ,
     *            MAXCAL,ETA,XTOL,STEPMX,FEST,DELTA,IBOUND,
     *            BL,BU,X,HESL,LH,HESD,ISTATE,F,G,IW,LIW,
     *            W,LW,IFAIL)
C
C     ..  TEST WHETHER IFAIL IS NON-ZERO  ..
C
      IF(IFAIL .NE. 0) WRITE(6,1040) IFAIL
      IF(IFAIL .EQ. 1) GOTO 60
C
      WRITE(6,1050) F
      WRITE(6,1060) (X(J),J=1,N)
C
      CALL FUNCT(1,N,X,F,G,IW,LIW,W,LW)
      CALL RESULT(M,ITITLE)
C
      IF(IFAIL .NE. 2)GOTO 60
C
      WRITE(6,1070) (ISTATE(J),J=1,N)
      WRITE(6,1080) (HESL(J),J=1,LH)
C
      WRITE(6,1090) (HESD(J),J=1,N)
C
   60 WRITE(6,1100)
      CALL TIME(1,-1,TIM)
      WRITE(6,1110)
      STOP
C
 1005 FORMAT(18A4)
 1010 FORMAT(6F10.5,I5)
 1015 FORMAT(4E15.6)
 1020 FORMAT(///1H ,I3, 37H FUNCTION EVALUATIONS WERE NEEDED BY ,
     *       6HE04HBF)
 1030 FORMAT(///48H TRY RESCALING THE PROBLEM. ELEMENTS OF HESD ARE,
     *        3(/1P5E15.4))
 1040 FORMAT(///16H ERROR EXIT TYPE, I3,23H - SEE ROUTINE DOCUMENT)
 1050 FORMAT(1H1,27H FUNCTION VALUE ON EXIT IS , E15.6)
 1060 FORMAT(13H AT THE POINT, 3(/5E15.6))
 1070 FORMAT(22H WHERE ISTATE CONTAINS, 15I5, 1H,)
 1080 FORMAT(14H HESL CONTAINS, 3(/1P5E20.4))
 1090 FORMAT(18H AND HESD CONTAINS, 3(/1P5E20.4))
 1100 FORMAT(///'EXECUTION TIME')
 1110 FORMAT(1H1)
C
      END
C
```

Listing of CONTACT at 16:17:27 on JUL 23, 1982 for CCid=MCB8

```
C  ***     ***     ***     ***     ***     ***     ***     ***
C
      SUBROUTINE FUNCT(IFLAG,N,XC,FC,GC,IW,LIW,W,LW)
C
      IMPLICIT REAL*8(A-H,P-Z)
C
      DIMENSION XC(N),W(LW),GC(N),Z41SQ(360),Z12SQ(360),
     *          Z23SQ(360),Z43SQ(360),FZ12(360),FZ23(360),
     *          FZ43(360),FZ41(360)
C
      INTEGER IW(LIW)
C
      COMMON ARC1,ARC2,ARC3,ARC4,CTA4,CTA1(361),CTA1D,PI,AMASO1,
     *       AMASO2,AMASO3,RRO1,RRO2,RRO3,ALMDO1,ALMDO2,
     *       ALMDO3,AIO1,AIO2,AIO3,M
C
      COMMON /RESLT/AMASS1,RRS1,ALMDS1,AIS1,AMASS2,RRS2,ALMDS2,AIS2,
     *             AMASS3,RRS3,ALMDS3,AIS3,F12X(360),F12Y(360),
     *             F23X(360),F23Y(360),F43X(360),F43Y(360),
     *             F41X(360),F41Y(360),PID
C
      PID=8.0D+00*DATAN(1.0D+00)
      DNIL=0.0D+00
C
      AMASS1=XC(1)
      RRS1=XC(2)
      ALMDS1=DMOD(XC(3),PID)
      AIS1=AMASS1*RRS1**2
      AMASS2=XC(4)
      RRS2=XC(5)
      ALMDS2=DMOD(XC(6),PID)
      AIS2=(RRS2**2+XC(7)**2)*AMASS2
      AMASS3=XC(8)
      RRS3=XC(9)
      ALMDS3=DMOD(XC(10),PID)
      AIS3=(RRS3**2+XC(11)**2)*AMASS3
      ZT12=DMOD(XC(12),PID)
      ZT23=DMOD(XC(13),PID)
      ZT43=DMOD(XC(14),PID)
      ZT41=DMOD(XC(15),PID)
C
      AMAS1=AMASO1+AMASS1
      AMAS2=AMASO2+AMASS2
      AMAS3=AMASO3+AMASS3
C
      XP1=(AMASO1*RRO1*DCOS(ALMDO1)+AMASS1*RRS1*DCOS(ALMDS1))/AMAS1
      YP1=(AMASO1*RRO1*DSIN(ALMDO1)+AMASS1*RRS1*DSIN(ALMDS1))/AMAS1
      XP2=(AMASO2*RRO2*DCOS(ALMDO2)+AMASS2*RRS2*DCOS(ALMDS2))/AMAS2
      YP2=(AMASO2*RRO2*DSIN(ALMDO2)+AMASS2*RRS2*DSIN(ALMDS2))/AMAS2
      XP3=(AMASO3*RRO3*DCOS(ALMDO3)+AMASS3*RRS3*DCOS(ALMDS3))/AMAS3
      YP3=(AMASO3*RRO3*DSIN(ALMDO3)+AMASS3*RRS3*DSIN(ALMDS3))/AMAS3
C
      RR1=DSQRT(XP1**2+YP1**2)
      RR2=DSQRT(XP2**2+YP2**2)
      RR3=DSQRT(XP3**2+YP3**2)
C
      ALMD1=DATAN2(YP1,XP1)
      IF(ALMD1 .LT. DNIL) ALMD1=ALMD1+PID
```

```
            ALMD2=DATAN2(YP2,XP2)
            IF(ALMD2 .LT. DNIL) ALMD2=ALMD2+PID
            ALMD3=DATAN2(YP3,XP3)
            IF(ALMD3 .LT. DNIL) ALMD3=ALMD3+PID
      C
            AI1=AI01+AIS1
            AI2=AI02+AIS2
            AI3=AI03+AIS3
      C
            FUN = 0.0D+00
      C
              DO 10 I=1,M
      C
      C   ...   COMPUTE KINEMATICS   ...
      C
                DC1=CTA1(I)
                RR=DSIN(DC1)-DSIN(CTA4)*ARC4/ARC1
                ES=DCOS(DC1)-DCOS(CTA4)*ARC4/ARC1
                DC14=DC1-CTA4
                TE=(ARC4**2+ARC1**2-ARC2**2+ARC3**2)/(2.0D0*ARC1
     *               *ARC3)-DCOS(DC14)*ARC4/ARC3
                DPSQ=RR**2+ES**2-TE**2
                TYY=RR-DSQRT(DPSQ)
                TXX=ES+TE
                DC3=2.0D0*DATAN2(TYY,TXX)
      C
                YYY=ARC3*DSIN(DC3)+ARC4*DSIN(CTA4)-ARC1*DSIN(DC1)
                XXX=ARC3*DCOS(DC3)+ARC4*DCOS(CTA4)-ARC1*DCOS(DC1)
                DC2=DATAN2(YYY,XXX)
      C
                DC31=DC3-DC1
                DC21=DC2-DC1
                DC23=DC2-DC3
                C2DU=ARC1*DSIN(DC31)*CTA1D/(ARC2*DSIN(DC23))
                C3DU=ARC1*DSIN(DC21)*CTA1D/(ARC3*DSIN(DC23))
      C
                C2DDU=CTA1D*ARC1*(DCOS(DC31)*(C3DU-CTA1D)-DSIN(DC31)*
     *               DCOS(DC23)*(C2DU-C3DU)/DSIN(DC23))/(ARC2*DSIN(DC23))
                C3DDU=CTA1D*ARC1*(DCOS(DC21)*(C2DU-CTA1D)-DSIN(DC21)*
     *               DCOS(DC23)*(C2DU-C3DU)/DSIN(DC23))/(ARC3*DSIN(DC23))
      C
                DC1L=DC1+ALMD1
                DC2L=DC2+ALMD2
                DC3L=DC3+ALMD3
                X1DDU=-RR1*DCOS(DC1L)*CTA1D**2
                Y1DDU=-RR1*DSIN(DC1L)*CTA1D**2
                X2DDU=-ARC1*DCOS(DC1)*CTA1D**2-RR2*(C2DDU*DSIN(DC2L)
     *               +C2DU**2*DCOS(DC2L))
                Y2DDU=-ARC1*DSIN(DC1)*CTA1D**2+RR2*(C2DDU*DCOS(DC2L)
     *               -C2DU**2*DSIN(DC2L))
                X3DDU=-RR3*(C3DDU*DSIN(DC3L)+C3DU**2*DCOS(DC3L))
                Y3DDU=RR3*(C3DDU*DCOS(DC3L)-C3DU**2*DSIN(DC3L))
      C
                DC2L1 = DC2L - DC1
                A2U = AI2*C2DDU + AMAS2*RR2*ARC1*DSIN(DC2L1)*CTA1D**2
                A3U = AI3*C3DDU
                ZSQM = (1.25D+00*CTA1D)**2
      C
```

```
                F23X(I)=(A2U*DCOS(DC3)/ARC2+A3U*DCOS(DC2)/ARC3)/DSIN(DC23)
                F23Y(I)=(A2U*DSIN(DC3)/ARC2+A3U*DSIN(DC2)/ARC3)/DSIN(DC23)
                Z23SQ(I)=(F23X(I)**2+F23Y(I)**2)/ZSQM
                C1Z23=DC1+ZT23
                FZ23(I)=(F23X(I)*DSIN(C1Z23)-F23Y(I)*DCOS(C1Z23))**2
C
                F43X(I)=AMAS3*X3DDU-F23X(I)
                F43Y(I)=AMAS3*Y3DDU-F23Y(I)
                Z43SQ(I)=(F43X(I)**2+F43Y(I)**2)/ZSQM
                C1Z43=DC1+ZT43
                FZ43(I)=(F43X(I)*DSIN(C1Z43)-F43Y(I)*DCOS(C1Z43))**2
C
                F12X(I)=AMAS2*X2DDU+F23X(I)
                F12Y(I)=AMAS2*Y2DDU+F23Y(I)
                Z12SQ(I)=(F12X(I)**2+F12Y(I)**2)/ZSQM
                C1Z12=DC1+ZT12
                FZ12(I)=(F12X(I)*DSIN(C1Z12)-F12Y(I)*DCOS(C1Z12))**2
C
                F41X(I)=AMAS1*X1DDU+F12X(I)
                F41Y(I)=AMAS1*Y1DDU+F12Y(I)
                Z41SQ(I)=(F41X(I)**2+F41Y(I)**2)/ZSQM
                C1Z41=DC1+ZT41
                FZ41(I)=(F41X(I)*DSIN(C1Z41)-F41Y(I)*DCOS(C1Z41))**2
C
   10           CONTINUE
C
         U1 = 1.0D-09
         DO 20 I = 1, M
           FUN=FUN + DLOG(U1+FZ12(I))**2 + DLOG(U1+Z12SQ(I))**2
     *             + DLOG(U1+FZ23(I))**2 + DLOG(U1+Z23SQ(I))**2
     *             + DLOG(U1+FZ43(I))**2 + DLOG(U1+Z43SQ(I))**2
     *             + DLOG(U1+FZ41(I))**2 + DLOG(U1+Z41SQ(I))**2
C
   20    CONTINUE
C
         FC = 1.0D-04 * FUN
C
         RETURN
         END
C
C         **  **  **  **  **  **  **  **  **  **  **  **  **  **  **
C
         SUBROUTINE MONIT(N,XC,FC,GC,ISTATE,GPJRNM,COND,POSDEF,
     *                    NITER,NF,IW,LIW,W,LW)
C
         IMPLICIT REAL*8(A-H,P-Z)
         DIMENSION GC(N),W(LW),XC(N)
         INTEGER IW(LIW),ISTATE(N)
         LOGICAL POSDEF
C
         WRITE(6,1110) NITER,NF,FC,GPJRNM
         WRITE(6,1120)
C
           DO 100 J=1,N
             ISJ=ISTATE(J)
             IF(ISJ .GT. 0) GOTO 20
             ISJ=-ISJ
             GOTO (40,60,80) , ISJ
```

Listing of CONTACT at 16:17:27 on JUL 23, 1982 for CCid=MCB8

```
C
   20      WRITE(6,1130)J,XC(J),GC(J)
           GOTO 100
C
   40      WRITE(6,1140) J,XC(J),GC(J)
           GOTO 100
C
   60      WRITE(6,1150) J,XC(J),GC(J)
           GOTO 100
C
   80      WRITE(6,1160) J,XC(J),GC(J)
           GOTO 100
C
  100   CONTINUE
C
      IF(COND .EQ. 0.0D0) RETURN
      IF(COND .LE. 1.0D+6) GOTO 120
C
      WRITE(6,1170)
      GOTO 140
C
  120 WRITE(6,1180) COND
C
  140 IF(.NOT. POSDEF) WRITE(6,1190)
C
      RETURN
C
 1110 FORMAT(/5H0ITNS, 5X, 8HFN EVALS, 11X, 8HFN VALUE, 11X,
     *       21HNORM OF PROJ GRADIENT/I4,6X,I5,2(6X,1PE20.4))
 1120 FORMAT(3H0 J, 11X, 4HX(J), 16X,4HG(J), 13X, 6HSTATUS)
 1130 FORMAT(1H , I2, 1X, 1P2E20.4, 5X, 4HFREE)
 1140 FORMAT(1H , I2, 1X, 1P2E20.4, 5X, 11HUPPER BOUND)
 1150 FORMAT(1H , I2, 1X, 1P2E20.4, 5X, 11HLOWER BOUND)
 1160 FORMAT(1H , I2, 1X, 1P2E20.4, 5X, 8HCONSTANT)
 1170 FORMAT(50H0ESTIMATED CONDITION NUMBER OF PROJECTED HESSIAN I,
     *       18HS MORE THAN 1.0E+6)
 1180 FORMAT(50H0ESTIMATED CONDITION NUMBER OF PROJECTED HESSIAN =,
     *       1H , 1PE10.2)
 1190 FORMAT(50H0PROJECTED HESSIAN MATRIX IS NOT POSITIVE DEFINITE)
C
      END
C
C ***      ***      ***      ***      ***      ***      ***      ***
C
      SUBROUTINE RESULT(M,ITITLE)
C
      IMPLICIT REAL*8(A-H,P-Z)
      DIMENSION ITITLE(18)
C
      COMMON /RESLT/AMASS1,RRS1,ALMDS1,AIS1,AMASS2,RRS2,ALMDS2,AIS2,
     *              AMASS3,RRS3,ALMDS3,AIS3,F12X(360),F12Y(360),
     *              F23X(360),F23Y(360),F43X(360),F43Y(360),
     *              F41X(360),F41Y(360),PID
C
      WRITE(7) M, ITITLE
      WRITE(7) (F12X(I),F12Y(I),F23X(I),F23Y(I),F43X(I),F43Y(I),
     *          F41X(I),F41Y(I),I=1,M)
C
```

Listing of CONTACT at 16:17:27 on JUL 23, 1982 for CCid=MCB8

```
      DL = 1.0D+03
      DLC = 1.0D+06
      DG = 3.6D+02/PID
      RRS1 = RRS1 * DL
      RRS2 = RRS2 * DL
      RRS3 = RRS3 * DL
      ALMDS1 = ALMDS1 * DG
      ALMDS2 = ALMDS2 * DG
      ALMDS3 = ALMDS3 * DG
      AIS1 = AIS1 * DLC
      AIS2 = AIS2 * DLC
      AIS3 = AIS3 * DLC
C
      WRITE(6,1200)
      WRITE(6,1210) AMASS1,RRS1,ALMDS1,AIS1,AMASS2,RRS2,ALMDS2,AIS2,
     *              AMASS3,RRS3,ALMDS3,AIS3
C
      RETURN
C
 1200 FORMAT(///'ADDED MASS PARAMETERS :-'//38X,'POSITION OF C. OF M.'
     *       /10X,'LINK',10X,'MASS',10X,'RADIUS',10X,'ANGLE',
     *       9X,'INERTIA')
 1210 FORMAT(/8X,'CRANK',4X,4E15.6/8X,'COUPLER',2X,4E15.6
     *       /8X,'FOLLOWER',1X,4E15.6)
C
      END
```

A P P E N D I X   I I D

LISTING OF COMPUTER PROGRAM

PINFORCE

Listing of PINFORCE at 16:34:02 on JUL 23, 1982 for CCid=MCB8

```
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C *  PROGRAM FOR PRODUCING JOINT FORCE PLOTS FOR FOUR-BAR LINKAGES  *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      IMPLICIT REAL*8(F)
      DIMENSION P12X(360),P12Y(360),P23X(360),P23Y(360),P43X(360),
     *          P43Y(360),P41X(360),P41Y(360),
     *          F12X(360),F12Y(360),F23X(360),F23Y(360),F43X(360),
     *          F43Y(360),F41X(360),F41Y(360),ITITLE(18)
C
      PI=4*ATAN(1.0)
      READ(7) M, ITITLE
      READ(7) (F12X(I),F12Y(I),F23X(I),F23Y(I),F43X(I),F43Y(I),
     *         F41X(I),F41Y(I),I=1,M)
C
      DO 5 I=1,M
        P12X(I) = F12X(I)
        P12Y(I) = F12Y(I)
        P23X(I) = F23X(I)
        P23Y(I) = F23Y(I)
        P43X(I) = F43X(I)
        P43Y(I) = F43Y(I)
        P41X(I) = F41X(I)
        P41Y(I) = F41Y(I)
    5 CONTINUE
C
      M1=1
      N1=0
      M2=M/3+1
      N2=M/3*360/M
      M3=2*M/3+1
      N3=2*M/3*360/M
      CALL PAPER(1)
C
      CALL PSPACE(0.20,0.45,0.55,0.80)
      VMX = -1.0E+04
      VMN = 1.0E+04
      CALL SKALE(P41X,VMX,VMN,M)
      CALL SKALE(P41Y,VMX,VMN,M)
C
      ST = 0.1 * (VMX - VMN)
      EXP=VMX+ST
      EXN=VMN-ST
      CALL ANOTAT(ST,ANO)
C
      CALL MAP(EXN,EXP,EXN,EXP)
      CALL DENSTY(2)
      CALL CTRSET(1)
      CALL CTRMAG(16)
      CALL PLOTCS(EXN,EXP+6.0*ST,ITITLE,72)
      CALL CTRMAG(7)
      CALL AXESSI(ANO,ANO)
      CALL CTRSET(2)
      CALL DENSTY(1)
      CALL PTPLOT(P41X,P41Y,1,M,34)
      CALL DENSTY(2)
      CURVEC(P41X,P41Y,1,M)
      CALL DENSTY(1)
      CALL CTRMAG(9)
```

```
          CALL PLOTNI(P41X(M1),P41Y(M1),N1,1)
          CALL SUPFIX
          CALL TYPECS('0',1)
          CALL NORMAL
          CALL PLOTNI(P41X(M2),P41Y(M2),N2,3)
          CALL SUPFIX
          CALL TYPECS('0',1)
          CALL NORMAL
          CALL PLOTNI(P41X(M3),P41Y(M3),N3,3)
          CALL SUPFIX
          CALL TYPECS('0',1)
          CALL NORMAL
          CALL DENSTY(2)
          CALL PLOTCS(EXN,EXP+ST,'FORCE EXERTED BY FRAME ON CRANK',31)
          CALL DENSTY(1)
          CALL BORDER
C
          CALL PSPACE(0.60,0.85,0.55,0.80)
          VMX = -1.0E+04
          VMN = 1.0E+04
          CALL SKALE(P12X,VMX,VMN,M)
          CALL SKALE(P12Y,VMX,VMN,M)
C
          ST = 0.1 * (VMX - VMN)
          EXP=VMX+ST
          EXN=VMN-ST
          CALL ANOTAT(ST,ANO)
C
          CALL MAP(EXN,EXP,EXN,EXP)
          CALL DENSTY(2)
          CALL CTRSET(1)
          CALL CTRMAG(7)
          CALL AXESSI(ANO,ANO)
          CALL CTRSET(2)
          CALL DENSTY(1)
          CALL PTPLOT(P12X,P12Y,1,M,34)
          CALL DENSTY(2)
          CALL CURVEC(P12X,P12Y,1,M)
          CALL DENSTY(1)
          CALL CTRMAG(9)
          CALL PLOTNI(P12X(M1),P12Y(M1),N1,1)
          CALL SUPFIX
          CALL TYPECS('0',1)
          CALL NORMAL
          CALL PLOTNI(P12X(M2),P12Y(M2),N2,3)
          CALL SUPFIX
          CALL TYPECS('0',1)
          CALL NORMAL
          CALL PLOTNI(P12X(M3),P12Y(M3),N3,3)
          CALL SUPFIX
          CALL TYPECS('0',1)
          CALL NORMAL
          CALL DENSTY(2)
          CALL PLOTCS(EXN,EXP+ST,'FORCE EXERTED BY CRANK ON COUPLER',33)
          CALL DENSTY(1)
          CALL BORDER
C
          CALL PSPACE(0.20,0.45,0.15,0.40)
```

Listing of PINFORCE at 16:34:02 on JUL 23, 1982 for CCid=MCB8

```
      VMX = -1.0E+04
      VMN = 1.0E+04
      CALL SKALE(P23X,VMX,VMN,M)
      CALL SKALE(P23Y,VMX,VMN,M)
C
      ST = 0.1 * (VMX - VMN)
      EXP=VMX+ST
      EXN=VMN-ST
      CALL ANOTAT(ST,ANO)
C
      CALL MAP(EXN,EXP,EXN,EXP)
      CALL DENSTY(2)
      CALL CTRSET(1)
      CALL CTRMAG(7)
      CALL AXESSI(ANO,ANO)
      CALL CTRSET(2)
      CALL DENSTY(1)
      CALL PTPLOT(P23X,P23Y,1,M,34)
      CALL DENSTY(2)
      CALL CURVEC(P23X,P23Y,1,M)
      CALL DENSTY(1)
      CALL CTRMAG(9)
      CALL PLOTNI(P23X(M1),P23Y(M1),N1,1)
      CALL SUPFIX
      CALL TYPECS('O',1)
      CALL NORMAL
      CALL PLOTNI(P23X(M2),P23Y(M2),N2,3)
      CALL SUPFIX
      CALL TYPECS('O',1)
      CALL NORMAL
      CALL PLOTNI(P23X(M3),P23Y(M3),N3,3)
      CALL SUPFIX
      CALL TYPECS('O',1)
      CALL NORMAL
      CALL DENSTY(2)
      CALL PLOTCS(EXN,EXP+ST,'FORCE EXERTED BY COUPLER ON FOLLOWER',36)
      CALL DENSTY(1)
      CALL BORDER
C
      CALL PSPACE(0.60,0.85,0.15,0.40)
      VMX = -1.0E+04
      VMN = 1.0E+04
      CALL SKALE(P43X,VMX,VMN,M)
      CALL SKALE(P43Y,VMX,VMN,M)
C
      ST = 0.1 * (VMX - VMN)
      EXP=VMX+ST
      EXN=VMN-ST
      CALL ANOTAT(ST,ANO)
C
      CALL MAP(EXN,EXP,EXN,EXP)
      CALL DENSTY(2)
      CALL CTRSET(1)
      CALL CTRMAG(7)
      CALL AXESSI(ANO,ANO)
      CALL CTRSET(2)
      CALL DENSTY(1)
      CALL PTPLOT(P43X,P43Y,1,M,34)
```

Listing of PINFORCE at 16:34:02 on JUL 23, 1982 for CCid=MCB8

```
      CALL DENSTY(2)
      CALL CURVEC(P43X,P43Y,1,M)
      CALL DENSTY(1)
      CALL CTRMAG(9)
      CALL PLOTNI(P43X(M1),P43Y(M1),N1,1)
      CALL SUPFIX
      CALL TYPECS('0',1)
      CALL NORMAL
      CALL PLOTNI(P43X(M2),P43Y(M2),N2,3)
      CALL SUPFIX
      CALL TYPECS('0',1)
      CALL NORMAL
      CALL PLOTNI(P43X(M3),P43Y(M3),N3,3)
      CALL SUPFIX
      CALL TYPECS('0',1)
      CALL NORMAL
      CALL DENSTY(2)
      CALL PLOTCS(EXN,EXP+ST,'FORCE EXERTED BY FRAME ON FOLLOWER',34)
      CALL DENSTY(1)
      CALL BORDER
C
      CALL CTRMAG(20)
      CALL GREND
      STOP
      END
C
C * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE SKALE(VAR,VMX,VMN,N)
C
      REAL VAR(N)
C
      DO 10 I=1,N
      IF(VMX .LT. VAR(I)) VMX=VAR(I)
   10 IF(VMN .GT. VAR(I)) VMN=VAR(I)
C
      RETURN
      END
C
C   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
C
      SUBROUTINE ANOTAT(ST,ANO)
C
      DT = 4.0 * ST
C
      IF (DT .GE. 200.0) ANO = FLOAT (IFIX (DT / 200.0)) * 200.0
      IF (DT .GE. 200.0) GO TO 20
      IF (DT .GE. 50.0 .AND. DT .LT. 200.0)
     *    ANO = FLOAT (IFIX (DT / 25.0)) * 25.0
      IF (DT .GE. 50.0) GO TO 20
      IF (DT .GE. 10.0 .AND. DT .LT. 50.0)
     *    ANO = FLOAT (IFIX (DT / 5.0)) * 5.0
      IF (DT .GE. 10.0) GO TO 20
      IF (DT .GE. 2.0 .AND. DT .LT. 10.0)
     *    ANO = FLOAT (IFIX (DT / 2.0)) * 2.0
      IF (DT .GE. 2.0) GO TO 20
      IF (DT .GE. 0.5 .AND. DT .LT. 2.0)
     *    ANO = FLOAT (IFIX (DT / 0.25)) * 0.25
```

Listing of PINFORCE at 16:34:02 on JUL 23, 1982 for CCid=MCB8

```
      IF (DT .GE. 0.5) GO TO 20
      IF (DT .GE. 0.1 .AND. DT .LT. 0.5)
     *   ANO = FLOAT (IFIX (DT / 0.05)) * 0.05
      IF (DT .GE. 0.1) GO TO 20
      IF (DT .GE. 0.02 .AND. DT .LT. 0.1)
     *   ANO = FLOAT (IFIX (DT / 0.02)) * 0.02
      IF (DT .GE. 0.02) GO TO 20
      IF (DT .GE. 0.005 .AND. DT .LT. 0.02)
     *   ANO = FLOAT (IFIX (DT / 0.005)) * 0.005
      IF (DT .GE. 0.005) GO TO 20
      IF (DT .LT. 0.005) ANO = FLOAT (IFIX (DT / 0.001)) * 0.001
C
   20 RETURN
      END
```