COMPUTER-AIDED ANALYSIS OF ENGINEERING TOLERANCES

PETER CHARLES INGHAM

Thesis submitted for the degree of Ph.D.

University of Aston in Birmingham

1977

COMPUTER-AIDED ANALYSIS OF ENGINEERING TOLERANCES

PETER CHARLES INGHAM

Thesis submitted for the degree of Ph.D.

1977

SUMMARY

In the design of mass-produced components, it is essential that manufacturing tolerances should be analysed to make sure that assemblies fit together satisfactorily and that parts are not produced to unnecessarily tight specifications. The analysis may be divided into three stages:

(a) calculation of the sensitivity of a feature's position to the magnitude of the tolerances upon which it depends,

(b) ensuring that the permitted tolerances which together influence a critical measurement are allocated in the most economical way, and

(c) analysing the statistical distribution of tolerances on critical measurements.

This thesis describes a method of performing stage (a). Stages (b) and (c) have been dealt with elsewhere.

It is demonstrated that the analysis of tolerances in all but the most straightforward cases is not a trivial operation and a model is developed to assist with the calculation. This is a location element derived from the classical six-point system for locating a body in three dimensions. Elements may be combined to describe multi-datum machining operations, assemblies and drawing dimension systems by a tree-like structure. The model is analysed mathematically, a compendium of commonly-occurring cases is appended and algorithms for obtaining results of interest to the engineer designer are described. A computer program for the sensitivity analysis is also described and the integration of the method into a full tolerance-analysis system is discussed.

KEYWORDS:

TOLERANCES, DESIGN, ENGINEERING, COMPUTERS

ACKNOWLEDGEMENTS

I wish to record my appreciation of the guidance and encouragement that I have received from my supervisor, Mr. I. H. Gould. I am also grateful to Dr. L. J. Hazlewood for the advice that he gave me on the eigenvalue problem and to Mrs. J. McIndoe who typed the manuscript.

An elementary two-dimensional version of the system was designed while I was working in the Design Office of Girling Ltd., Tyseley, and I should like to thank Mr. W. Harrison who supervised my earlier efforts.

INDEX

1. GENERAL DISCUSSION

1. General Discussion

1.1 The Principle of Infallible Interchangeability

Many of the aims in the design of mass-produced components are unattainable. Examples are zero cost, zero weight, infinite strength and ultimate aesthetic appeal. However, a major aim which can often be achieved is infallible interchangeability. This term, probably first used in ref. T.7 means that a component selected at random from a batch of like components should fit satisfactorily to any one of a batch of mating components. In some situations, selective assembly may be a suitable process but usually it is precluded since it is costly in resources of labour and time. Normally, batches of components must be assembled unselectively.

The principle of infallible interchangeability leads to some difficulties in practice. All manufacturing processes are subject to size variation in a degree depending on the particular process; two components, even when produced on the same machine, will be unlikely to be of the same size. Exact fit is, therefore, another unattainable aim of mass production design and design clearances must make allowance for process error.

1.2 Types of Working Drawings

Unfortunately, modern design is a specialised function and modern designers are not expected to be experts in jig and tool design, in metrology nor in any other of the branches of production engineering. The task of the designer is to specify the functional requirements of the finished part, and, although he will usually have some knowledge of the manufacturing and inspection processes irvolved, he will not normally lay down a rigorous specification for them. This principle is clearly stated in BS 308: Part 2: 1972:

"Production processes or inspection methods should not be specified unless they are essential to ensure satisfactory functioning or interchangeability."

It is also discussed at length in ref. T.7.

There are two types of working drawing. These are -

(a) product drawings completely defining the finished product as required by the designer, and

(b) process drawings defining products in a partly finished state suitably dimensioned for the manufacturing process to be adopted.

A view sometimes expressed is that the product drawing is not the definition of a machined part, but the definition of a gauging method for a machined part. Although this is contrary to the spirit of BS 308 since gauging may be considered to be a manufacturing process, it is partially true.

A part may be therefore dimensioned in three ways:

(a) for its function, so that it may work satisfactorily,

(b) for a process, so that it may be made, and

(c) for inspection, so that sizes may be checked.

Each of these may involve different dimension systems for the same part and it is essential that tolerances arising from (b) and (c) be not greater than those specified in (a). In many cases, this may be checked using simple arithmetic (and some simplifying assumptions, usually) but often it is no trivial process.

1.3 Further Problems in Dimensioning

A machining process will affect some functional clearance, possibly in an indirect way; and often a functional clearance is affected by more than one process. Usually, the production engineer has several possible machining processes available, each having its own accuracy. As a rule, the more accurate a machining process, the higher the unit cost and it is desirable that the more accurate processes are used in features which have the greatest effect on the functional clearance.

Another difficulty is that many common dimensioning systems, even some described in BS 308, are ambiguous and their interpretation depends on some convention. In the majority of cases, alternative interpretations lead to differences in functional clearances which are small. However, this may not always be so and an unequivocal method of describing dimension systems is desirable.

2. BASIC CONCEPTS

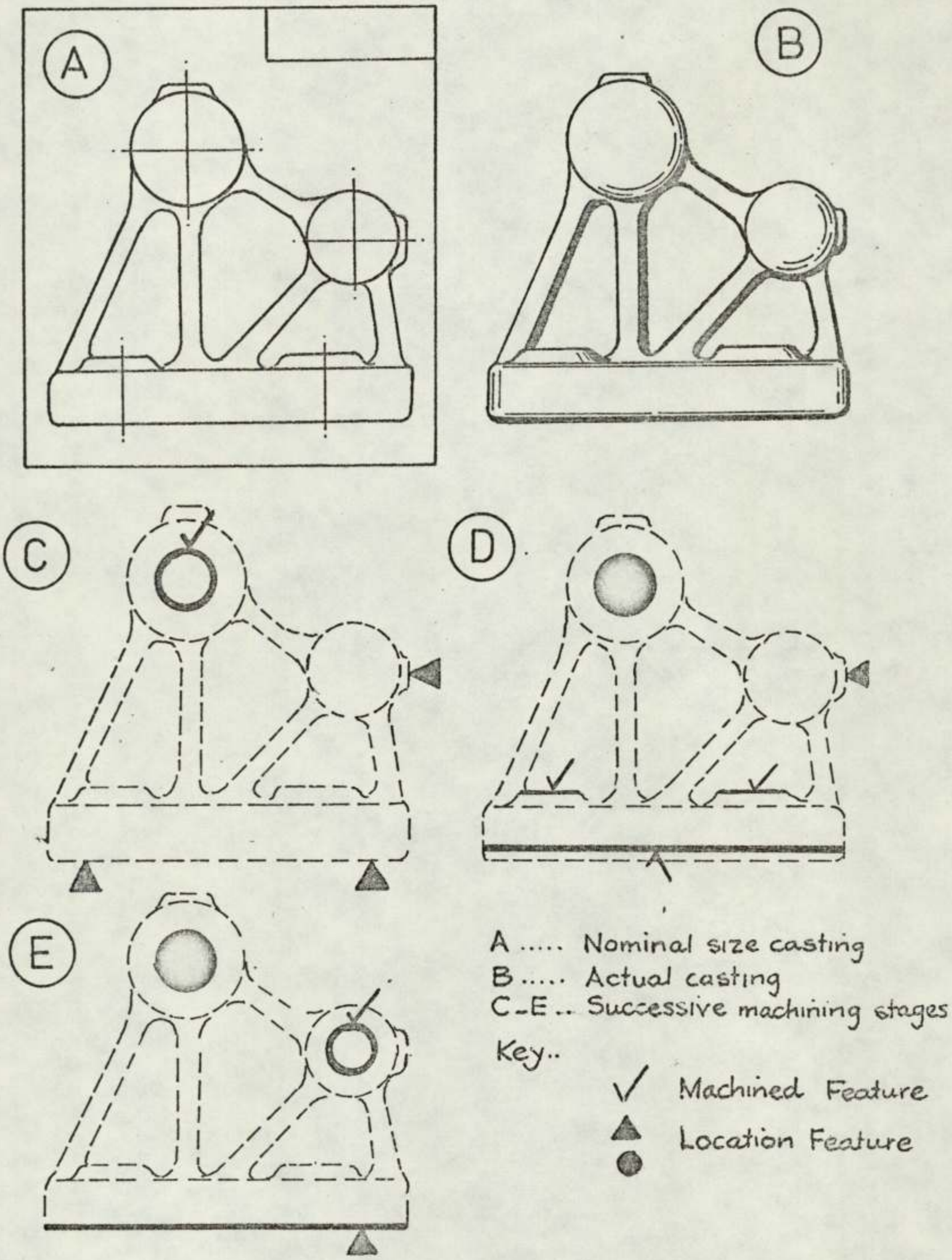


Fig 1 Component Chains

2. Basic Concepts

2.1 Component chains - A model for complex manufacturing processes and assemblies

A feature on a component may be displaced from its nominal position for two reasons. Firstly, the component may be machined in several stages, each stage having its own, possibly distinct, datum systems. Secondly, the component may be assembled on other components each having its own variations in size. Both of these cases can be treated in the same way since they are conceptually identical.

Each stage in the machining of a component may be regarded as a separate physical component, the stages being assembled together to make the finished part. An example is shown in Fig. 1. The reference body is the nominal size drawing of the casting, the actual casting being located on it by the dimensions on the casting drawing. Each subsequent machining stage is located either on the actual cast form, or on previous machining stages, or on both. A machined part may then be treated in the same way as an assembly.

The classical method of location consists of clamping a body to a plane, to a line and to a point, so that six degrees of freedom are removed. This applies both to the physical assembly of components and to a machined feature on a component since six point location is considered to be good jig and tool practice (for example, see ref. G.12, p. 77). Some common dimensioning systems cannot be described in six point location form and these will be discussed at a later stage.

The example shown in Fig. 1 is described diagrammatically in Fig. 2. Each box represents a machining stage and each arrow represents the relationship 'depends on the size of'. Thus variations in size will be passed on, through a chain, to the finished part.

Finished components may also be fitted together to form an assembly. Assemblies may also be described by diagrams similar to Fig. 2. Each box

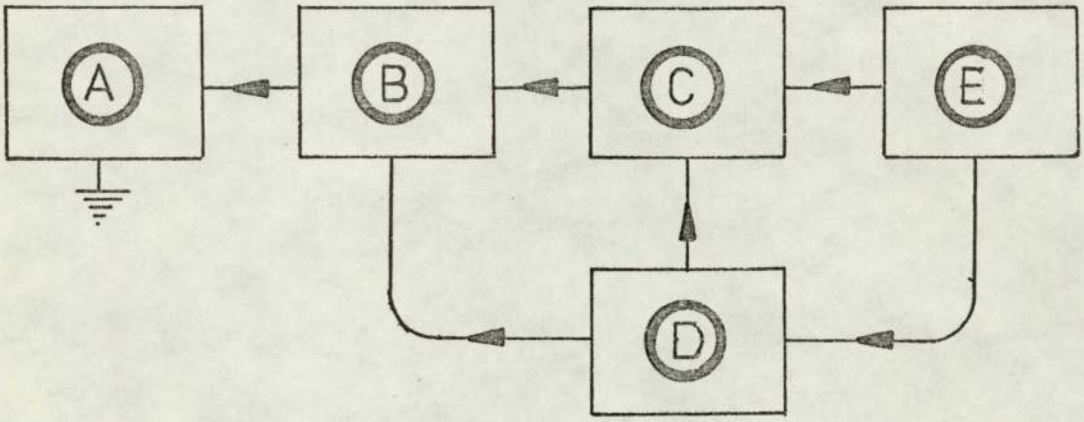


Fig 2 Component Chains.

now represents a finished part, the meaning of the arrows being the same as for machining stages. Multi-stage machining processes and assemblies may, then, be considered in the same way, and, to avoid confusion, they will subsequently be called 'assemblages'.

2.2 Problems to be solved

Assemblages of both kinds may be linked together in the same way. At some stage, a useful limit will be reached, and questions that one might wish to ask are -

(a) in Fig. 2, what is the effect at a feature in stage C of given variations in size at stages A and B,

(b) in Fig. 2, if the maximum permitted variation in position or size in stage C be known, then how should the tolerances be apportioned between stages A and B so as to minimise the process cost, and

(c) what is the clearance between a feature on component D and one on component E in Fig. 2?

The concept of regarding a finished part as an assembly with some of the components possibly occupying the same space as others is fundamental to the system to be described. Its use enables assemblages to be defined in a unified way and questions (a), (b) and (c) may be answered in much the same fashion.

The terms 'tolerance' and 'location' have been used so far in a fairly loose, commonsense way since they are of common currency in engineering. However, as they will be used subsequently in a more specialised sense, some discussion of them follows.

2.3 Tolerances

2.3.1 Definitions

Tolerance is the variation from nominal position of a feature of interest on a component.

Tolerances may be specified bilaterally, the locating dimension consisting of a mean size with a tolerance equally disposed about it;

or unilaterally, the locating dimension consisting of a size at one extreme with a tolerance quoted in the opposite direction. In all the examples which follow tolerances will be specified bilaterally.

A tolerance zone is the zone within which the feature of interest is required to be contained. BS 308 specifies that a tolerance zone is one of the following:

(1) a circle or cylinder

(2) the area between two parallel lines or two parallel straight lines

(3) the space between two parallel surfaces or two parallel planes

(4) the space in a parallelepiped.

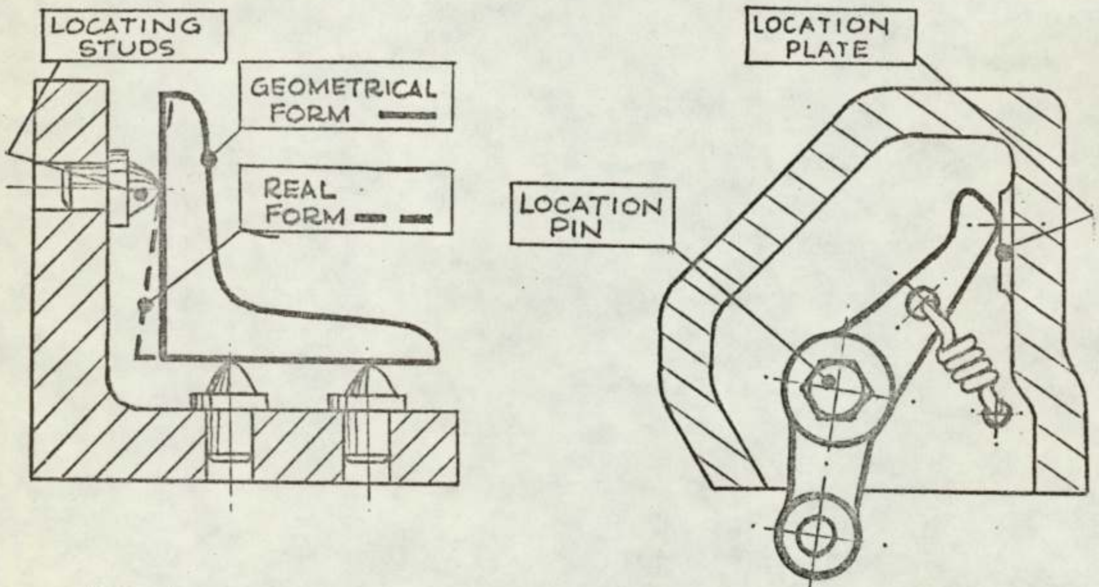
A tolerance zone which is occasionally useful, but which is absent from the list, is a sphere.

2.3.2 Intrinsic and extrinsic tolerances

A feature may be displaced from its nominal position for two reasons. Firstly, there will be a tolerance on its position due to error in the manufacturing stage which has produced it. This will be called intrinsic tolerance, and will be the tolerance quoted on the process drawing of the feature. In following examples, standard BS 308 tolerance frames will be used to show intrinsic tolerance. Secondly, there will be a tolerance on the position of the feature resulting from tolerances on previous manufacturing stages on which its location depends, or tolerances on the finished parts on which it is assembled. This will be called extrinsic tolerance and must be calculated from the intrinsic tolerances on the locating parts. The sum of intrinsic and extrinsic tolerance will be termed 'total tolerance'. In all cases, tolerance is relative to some frame of reference. Intrinsic tolerance is relative to the nominal positions of the locating features, while extrinsic and total tolerances are relative to any feature of interest.

2.4 Locations

There are two types of location system which are described below using two-dimensional examples for illustrative purposes.



JIG LOCATION

LOCATION OF LEVER ASSEMBLY

FIG 3A REAL LOCATIONS

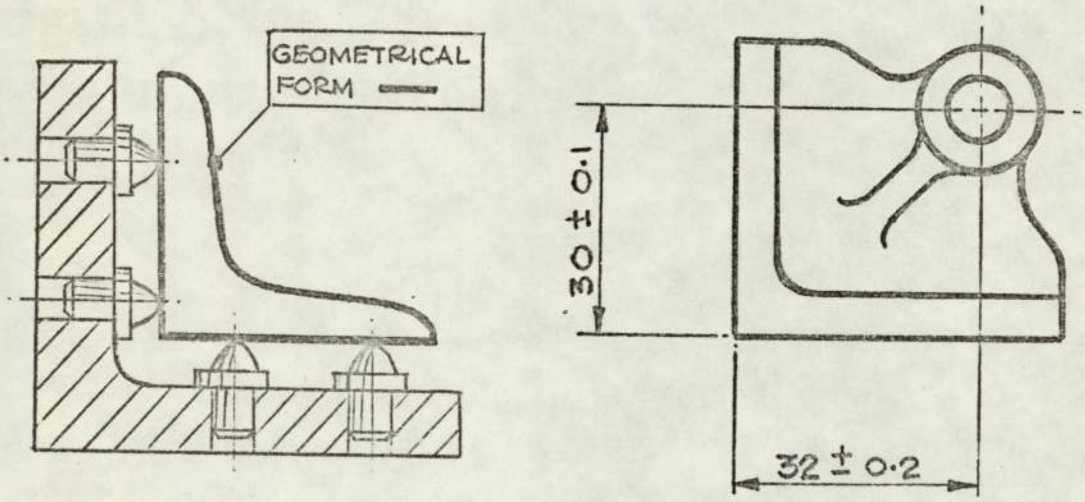


FIG 3B GEOMETRICAL LOCATIONS

Fig 3 Location Systems

2.4.1 Real Locations

These are location systems which may be realised physically and locate real bodies on real locating features, both necessarily having irregularities in form and displacements from nominal positions. Real locations occur in the jigging of manufacturing processes and in the assembly of components. Examples of each are shown in Fig. 3A.

2.4.2 Geometrical Locations

Functional drawings often describe location systems which may not be realised physically since they refer to idealised geometrically exact figures. A common example is shown in Fig. 3B. As may be seen, the drilled hole cannot be located physically on two datum faces. The most plausible interpretation which can be made of this system is that any convenient jigging system (which necessarily involves a real location) is to be used but that on the finished part, the hole is to lie within the parallelepipedal tolerance zone defined on the drawing and centred at the intersection point of two lines parallel with the datum face at a distance from them specified by the drawing dimensions.

A detailed discussion of this dimension system will be found in ref. T.7 pp. 266-270.

2.5 Cumulative Tolerance

The concepts described in the previous sections are illustrated by the example shown in Fig. 4. This is unrealistic but not wildly so.

Faces E and F are assumed to be geometrically exact, while face D is subject to a profile tolerance and lies within a band as shown in (1) on the diagram. It is further assumed that all locating points are exact, but the central axis of the machine tool is subject to a circular tolerance zone relative to the corresponding locations. Form irregularities normal to the plane of the diagram are discounted.

The three holes are machined using separate locating systems as

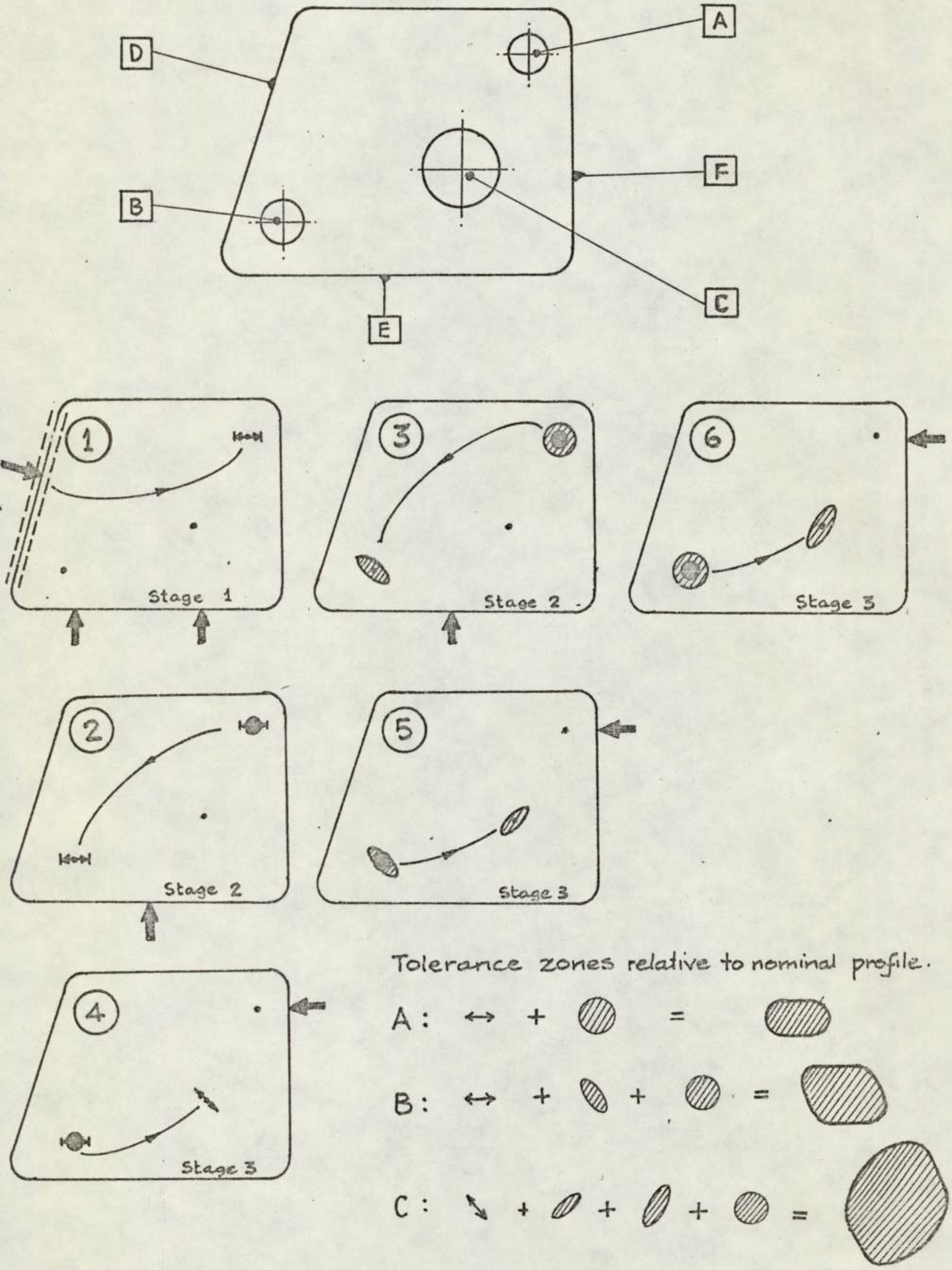


Fig 4 Cumulative Tolerances.

shown below:

Stage 1: Hole A ... datum Face E, and a point on Face D.

Stage 2: Hole B ... datum Hole A, and a point on Face E.

Stage 3: Hole C ... datum Hole B, and a point on Face F.

Stage 1. The linear tolerance at face D will result in the position of hole A relative to all other faces of the plate being subject to a linear tolerance - as in (1). There will also be a circular tolerance zone at A relative to the locating features. This will also be relative to the actual profile of face D and all other nominal faces.

The total tolerance at A relative to all nominal faces will be these two tolerance zones superimposed.

Stage 2. The linear tolerance at A results in a linear tolerance at B (shown in (2)); the circular positional tolerance zone at A (plus clearance between hole A and the locating peg) causes a tolerance zone at B which is approximately elliptical (shown in (3)). There will also be a circular positional tolerance at B, and the total tolerance zone at this hole relative to the nominal profile will be the superimposition of the linear, elliptical and circular tolerance zones.

Stage 3. There are four tolerance zones at C: a linear zone due to linear tolerance at B (shown in (4)), an elliptical zone due to the elliptical zone at B (shown in (5)), an elliptical zone due to the circular zone and clearance at B, and a circular positional tolerance. The total tolerance zone is the superimposition of the four.

All tolerance zones shown in the diagram are grossly exaggerated, and have been obtained by tracing the tolerance loci; but the effect of cumulative tolerance is clearly shown. In a real system, extra complication would be added because of factors which have been conveniently ignored in this model. For instance, the locating points would not be exactly positioned; and face F would be subject to a form tolerance.

Difficulties involved in analysing a multistage system are:

(a) The extreme position of a tolerance zone at a feature is not necessarily the position corresponding to an extreme position of zones at the locating features.

(b) Some tolerance zones are dependent - an example being the two zones shown in (3). The nett displacement between holes B and A does not depend on these zones. Deciding on which displacements at one feature are relative to another can involve much book-keeping and possible error.

(c) Real parts exist in three dimensions and though many tolerance situations may be considered as being two-dimensional, this is not always so. In a three-dimensional system, tolerance zones may be parallelepipedal or ellipsoidal and their effects are difficult to visualise, let alone calculate.

3. DESCRIPTION OF THE LOCATION MODEL

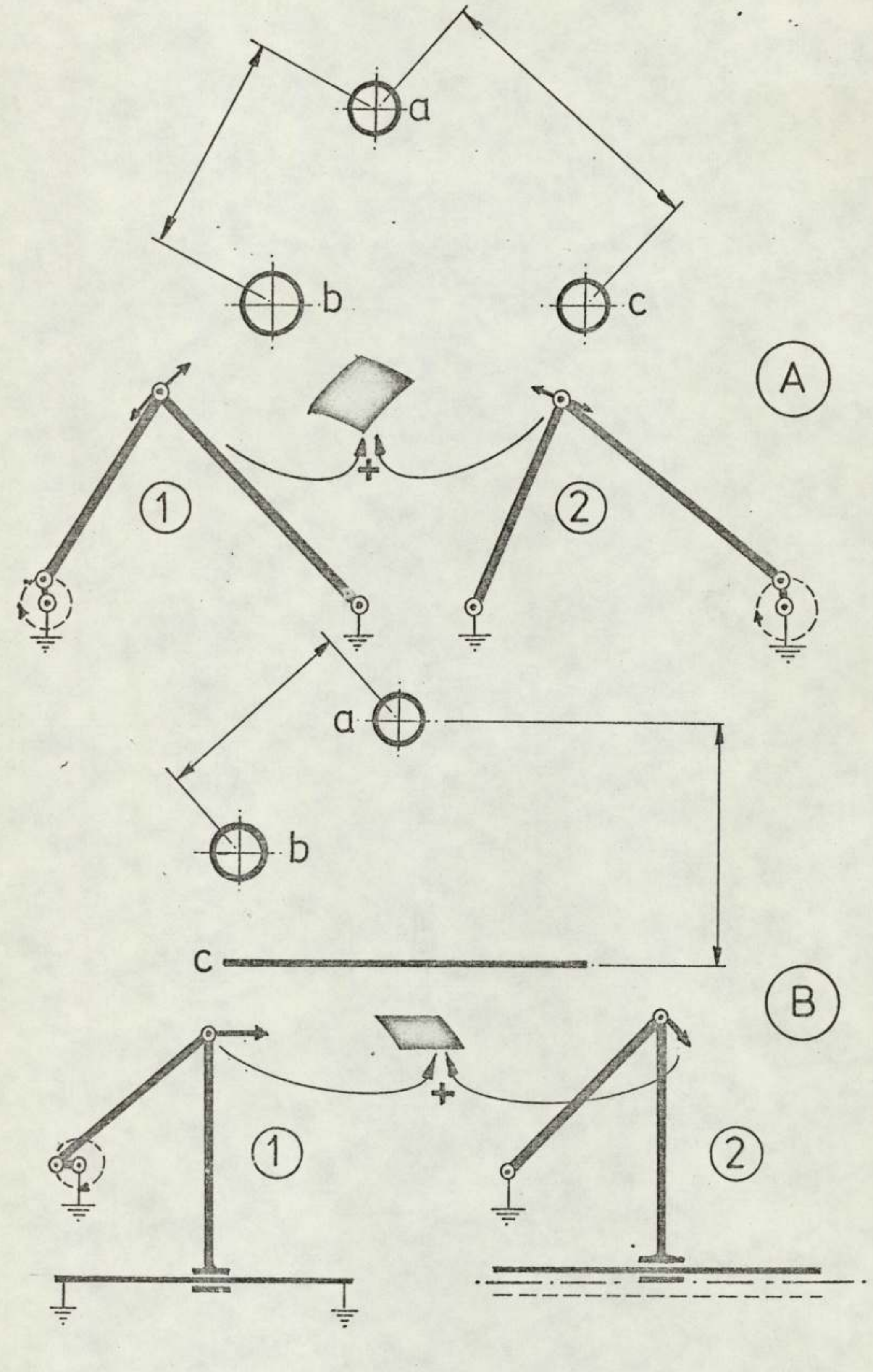


Fig 5 Equivalent Mechanisms

3. Description of the Location Model

3.1 Tolerance Mechanisms

Fig. 5A shows a common dimensioning system. Hole a is located by the centre points of holes b and c. The dimensioning system is analogous to a structure - there are no redundancies in the dimensions and if any dimension is deleted, then the remainder are insufficient to locate the hole. In this case, the three members of the structure are the centre distances of the holes.

Extrinsic tolerances will be passed to the located feature from each of the locations. If a circular tolerance be imposed at hole b, hole c being held at its nominal position, then the dimension system may be regarded as a mechanism, each position of hole b corresponding with an unique position of hole a. The mechanism in the case illustrated is a four bar chain and the locus of hole a is a short circular arc. If now hole b be held at nominal and a circular tolerance zone applied to hole c, then a similar mechanism is obtained - Figures 5 A-1 and A-2. Since the radius of each tolerance zone is small in comparison with the locating dimensions, the two systems may be superimposed to give a total tolerance zone as shown in the figure. This approximates to a parallelogram as the lengths of arc are small.

Another dimensioning system is illustrated in Fig. 5B. In this case, hole a is located by its distance from hole b and by its perpendicular distance from line c. Again the system is exactly determined and may be considered as a structure. If a circular tolerance zone be applied at hole b then an equivalent mechanism may be derived. In this case, since the perpendicular distance from line c is specified, a sliding member is necessary on the line, while the crank centred at the nominal position of hole b generates the circumference of the tolerance zone. At hole a, the zone generated is a short straight line parallel with c.

Similarly, if a tolerance band is allowed at line c, hole b being

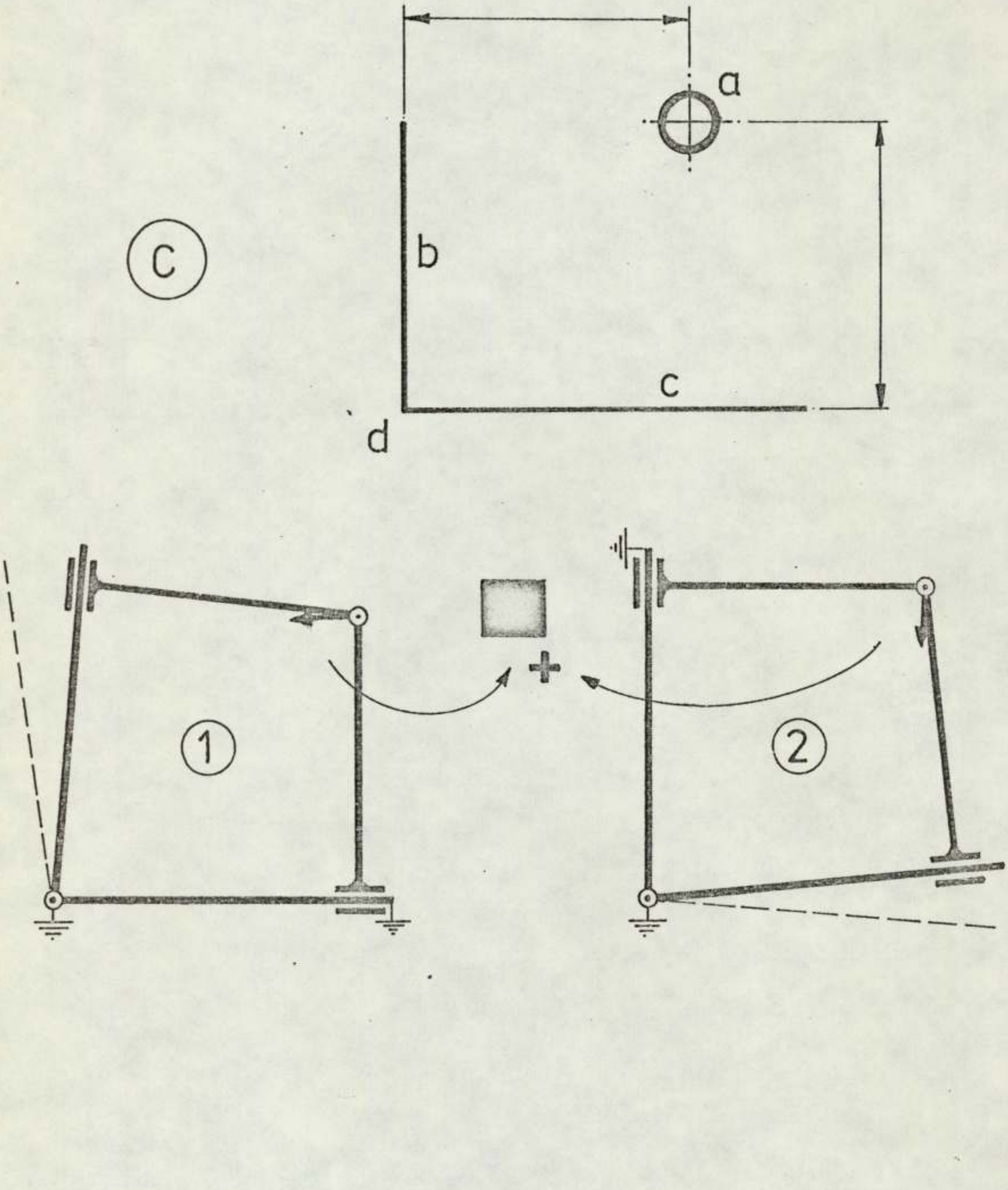


Fig 6 Equivalent Mechanisms.

held at its nominal position, the resulting zone at hole a will be a small circular arc centred at b.

The two tolerance zones may be superimposed and the resulting zone is approximately a parallelogram.

Fig. 6 shows a mechanism with two sliding pairs which is equivalent to a point dimensioned from two straight line datums.

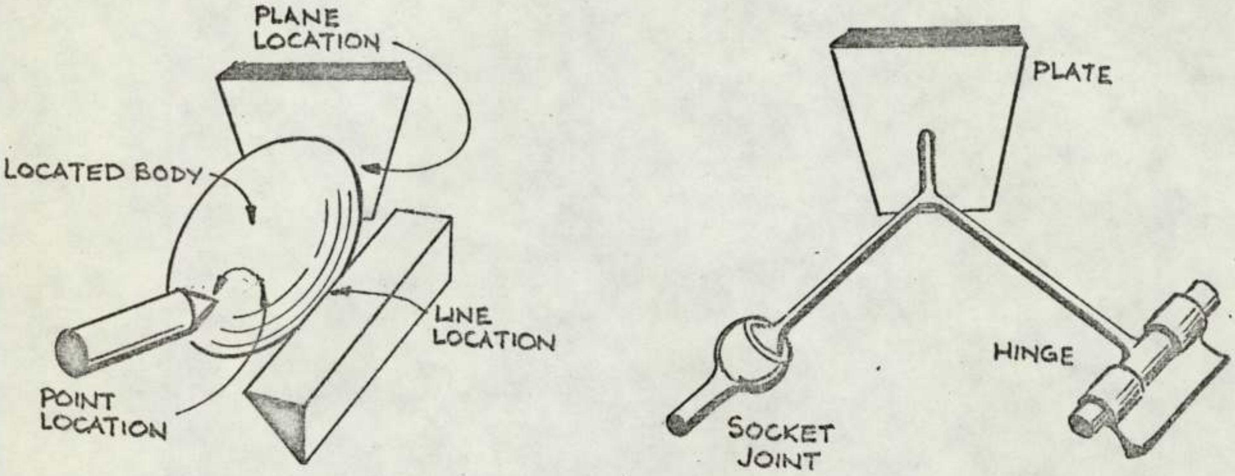
There is a general solution for three dimensional mechanisms comprised of pure turning and sliding pairs (refs. G.6 and G.14), and it seems feasible that a method for the analysis of tolerances based directly on the use of such elements could be derived. This direct approach suffers from some disadvantages, however. As has been demonstrated in section 2.5, explicit tolerance zones are of irregular shapes, and if, for example, hole b in Fig. 5A were located in the same way as hole B in Fig. 4, then the tolerance zone would certainly not be circular. Even if the zone were decomposed into its separate elements, it would be necessary to use a crank arm with a radius varying dynamically with turning angle, one of the elements being elliptical. Also, the method is rather inflexible, as each of the many possible dimension systems would require a separate equivalent mechanism with a separate method of calculation. Although the vector equations for these mechanisms may be written down using the methods of ref. G.6 they do not appear tractable for solution in some cases. These problems are exacerbated in three dimensions.

Equivalent mechanisms are useful in visualising the effects of explicit tolerances, but the preferred method will be based on a standard unit of location. Indeed, the method might be used, with a little modification, in the analysis of the general kinematic mechanism; but this is outside the scope of this thesis.

3.2 Elemental Location

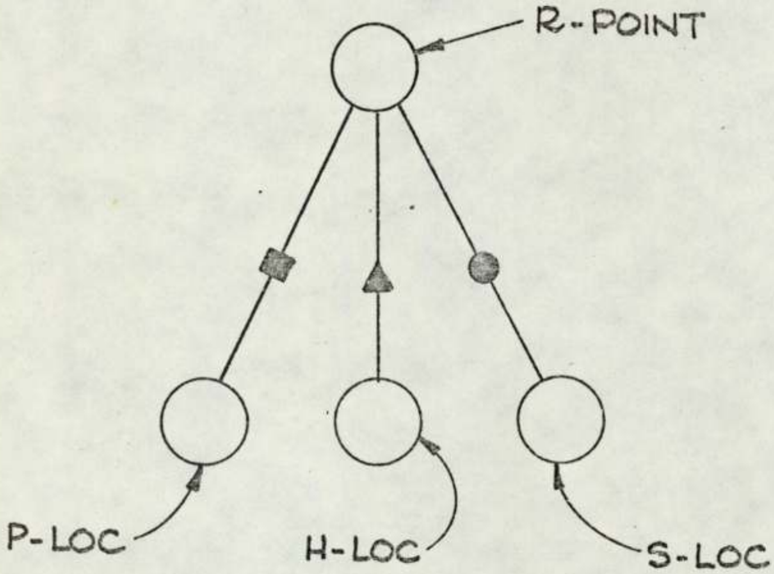
A body is located elementally if -

(a) a point on it is held against a locating plane,



ELEMENTAL LOCATION

MECHANICAL ANALOGY



A LOCATION TRIAD

Fig 7 Elemental Location

(b) a point on it is held against a locating line, and

(c) a point on it is held against a locating point.

The located body is sensitive to small displacements -

(a) along the normal to the locating plane,

(b) orthogonal to the locating line and

(c) in any direction at the locating point.

Elemental location is illustrated in Fig. 7.

In order to avoid confusion, a locating plane will be called a plate, a locating line will be called a hinge and a locating point a socket. The terms have been picked because of their obvious mechanical analogies and also because they have distinct initial letters. In subsequent reference, the following concise terms will often be used:

(a) a plate location will be called a P-loc,

(b) a hinge location will be called an H-loc, and

(c) a socket location will be called an S-loc.

These will be referred to collectively as a location triad.

A displacement at a locating feature will result in a displacement at other points on the located body. A point at which the displacement is required will be called a result-point (or more concisely an R-point).

An R-point located on a triad will be shown graphically as exemplified in Fig. 7. The root node represents the R-point and the three links are distinguished by the convention:

(a) a P-loc is shown by a square,

(b) an H-loc is shown by a triangle, and

(c) an S-loc is shown by a circle.

Each symbol is placed on the appropriate link, and a link indicates the relationship 'is located on', in a top-down sense.

3.3 Displacement Matrices

The general location element, described previously, is analysed by using energy methods since these are commonly used in engineering science.

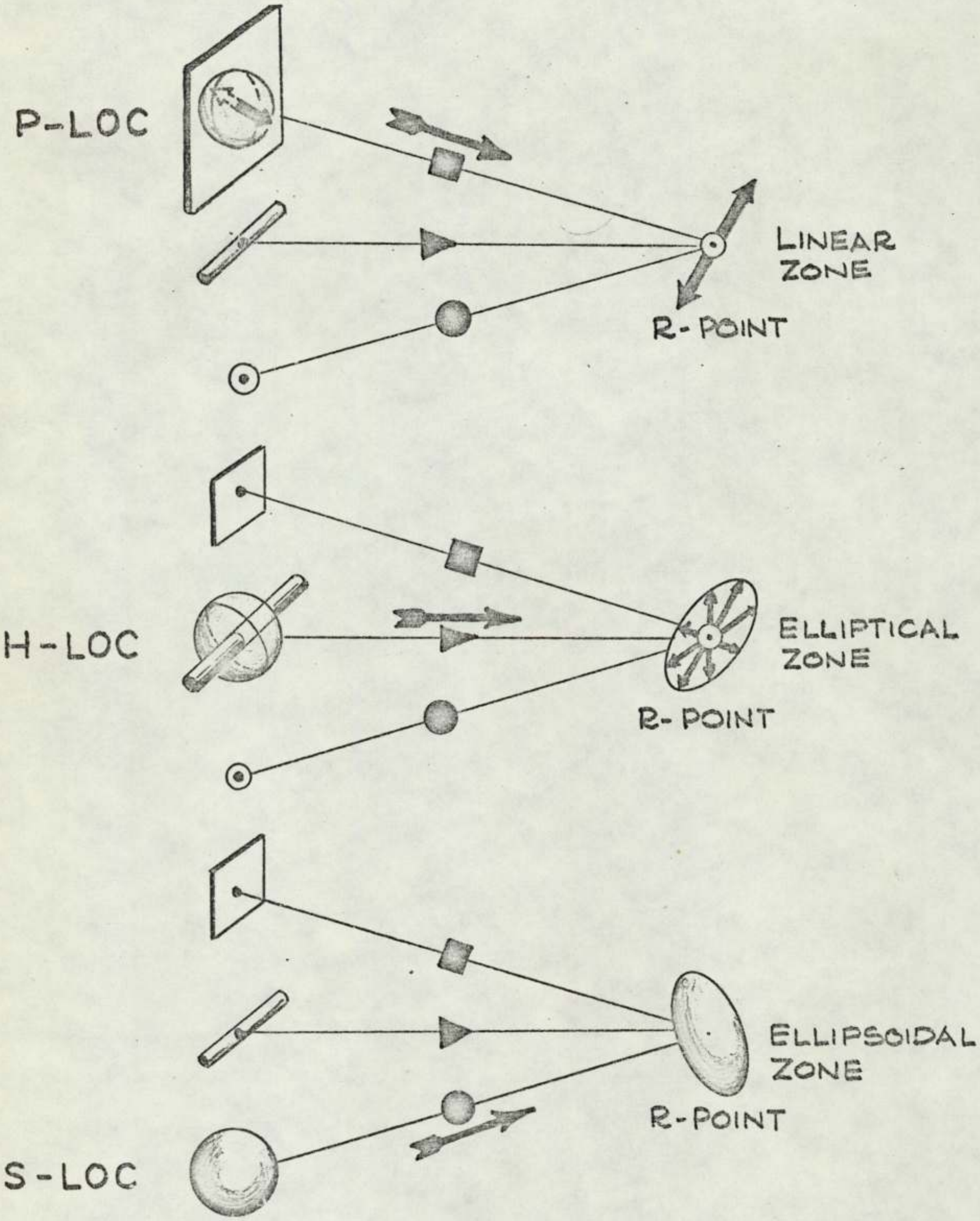


Fig 8 Transformations

Details of the analysis will be found in Appendix A but an outline is given here for reference.

The parameters listed below define the location system, coordinates being relative to some convenient set of orthogonal axes. The sense of the directions of the lines is immaterial.

(a) Coordinates of the R-point.

(b) Coordinates of the points of action of the P-, H- and S-locs.

(c) Direction cosines of the normal to the locating plane and of the direction of the locating line.

If the displacement at a locating feature be $\bar{D}_{in}$, and the displacement at the R-point be $\bar{D}_{out}$, both of these being column vectors, then

$$\bar{D}_{out} = \bar{M}\,\bar{D}_{in}.$$

$\bar{M}$ is a 3 x 3 matrix with coefficients depending on the coordinates of the locating triad. Each location feature will have a different matrix; the notation for these is shown below.

(a) P-loc; $\bar{M} = \bar{P}$

(b) H-loc; $\bar{M} = \bar{H}$

(c) S-loc; $\bar{M} = \bar{S}$

There are, as is discussed in Appendix A, restrictions on the positions and directions of the locating features. These define a proper location and are easily visualised; for instance, a socket may not exactly correspond with the centre of action of a P-loc. If the location features are not restricted in this way, then matrix coefficients may become infinite.

It is proved in Appendix A that a $\bar{P}$-matrix is of rank at most 1, an $\bar{H}$-matrix of rank at most 2 and an $\bar{S}$-matrix of rank at most 3. The matrices may be thought of as three-dimensional transformation operators, and if they act on a unit sphere, then the $\bar{P}$-matrix transforms it to a straight line, the $\bar{H}$-matrix transforms it into an ellipse; and the $\bar{S}$-matrix transforms it into an ellipsoid. These transformations are illustrated in Fig. 8.

4. LOCATION NETWORKS

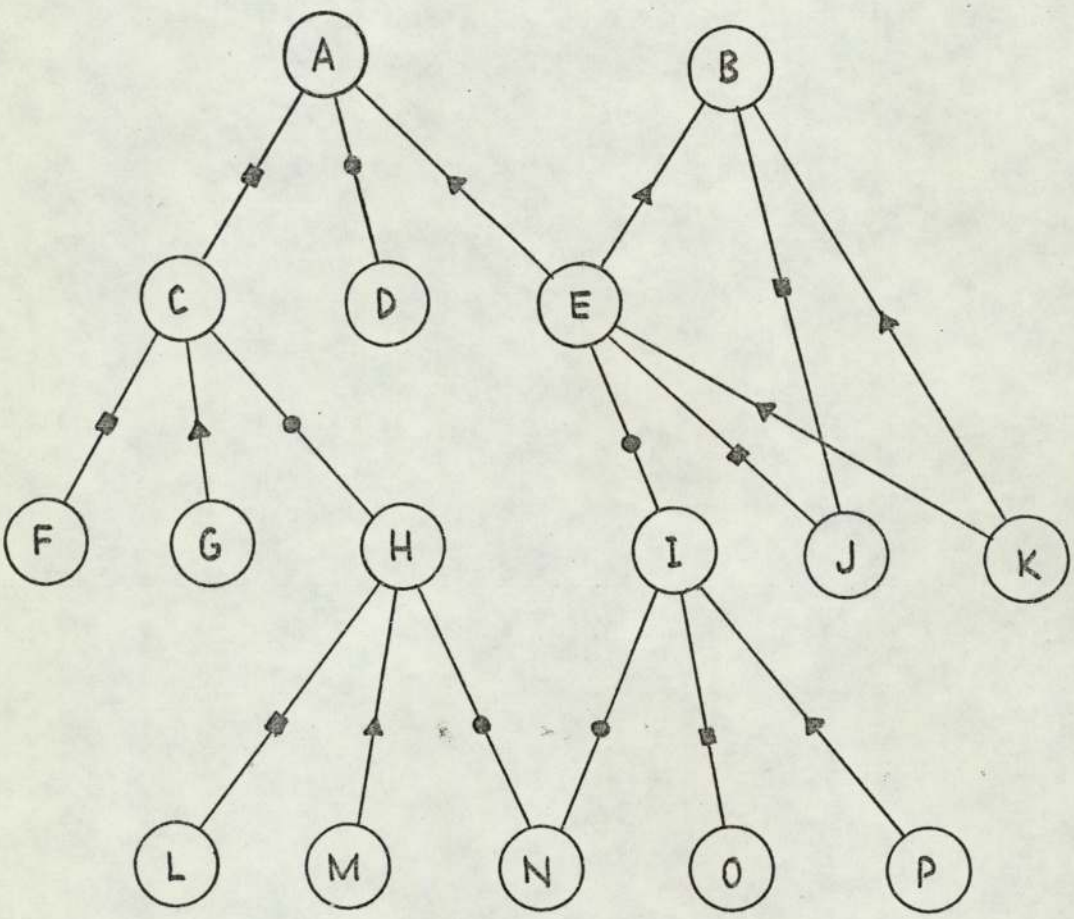


Fig 9 Location Networks

4. Location Networks

4.1 Assemblage network and paths

An assemblage may be represented by a directed graph consisting of location triads linked together as shown in Fig. 9. The effect of a displacement at feature O will be transmitted through the network to feature B. The corresponding displacement at B will be found by multiplying all the matrices corresponding to the edges of the graph lying on the path between the two features. If the product matrix be $\overline{M}$, then the output displacement at B may be found from the general equation:

$$\overline{D}_{out} = \overline{M}\,\overline{D}_{in}$$

where $\overline{D}_{out}$ is the output displacement column vector, and

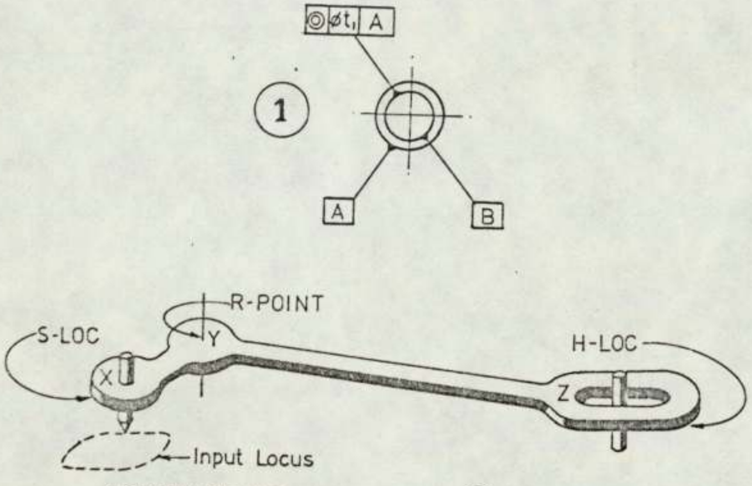
$\overline{D}_{in}$ is the input displacement column vector.

Often multiple paths exist between the location at which the displacement is applied and the R-point. In this case, the calculation of the matrix $\overline{M}$ is not so straightforward and a discussion is to be found in Appendix A. An example of multiple paths is the pair of paths joining nodes N and A in Fig. 9. (Paths N H C A and N I E A).

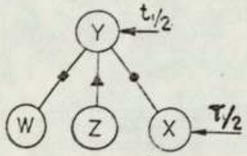
If a spherical displacement locus is applied at O, then the resulting output displacement locus at B will be a transformation of the sphere. In the general case, the output zone might be linear, elliptical or ellipsoidal depending on the rank of the transformation matrix $\overline{M}$ (see Fig. 8). For a pair of nodes which are connected by multiple paths, it is not possible to predict the rank of $\overline{M}$ without first calculating the path products. For a simple path, the rank of $\overline{M}$ will be the lowest rank of matrix associated with any edge along it (Appendix A).

4.2 Examples

The construction of assemblage networks requires some skill in visualising the tolerance mechanisms involved, although some assistance is provided in Appendix D which contains details of all the common locating



EQUIVALENT MECHANISM FOR ①



NETWORK FOR ①

Coordinates:

W: $x_1, y_1, 0$; 0°, 0°;

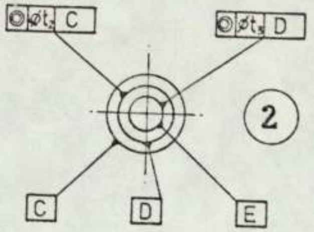
X: $x_1, y_1, 0$; (Hole A).

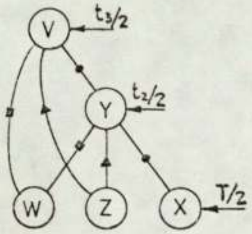
Y: $x_1, y_1, 0$; (Hole B).

Z: 10^6, 0, 0; 90°, 0°.

T = breadth of input tolerance.

Fig 10 Subnetworks & Mechanisms





NETWORK FOR ②

Coordinates:

W: $x_2, y_2, 0$; 0°, 0°.

X: $x_2, y_2, 0$; (Hole C).

Y: $x_2, y_2, 0$; (Hole D).

Z: 10^6, 0, 0; 90°, 0°.

V: $x_2, y_2, 0$; (Hole E).

T = breadth of input tolerance.

Fig 11 Sub-networks & Mechanisms.

systems. Fig. 10 shows a simple example. Hole A is located on the centre of hole B, the nominal centres of both holes being coincident and hole A having a concentricity tolerance relative to hole B. Extrinsic tolerance due to displacement of hole B must be separated from the intrinsic tolerance due to the concentricity tolerance. The situation is shown in Fig. 10, the mechanism XYZ being used to assist in visualising the location triad. Some, possibly irregular, tolerance zone exists at hole B, and this must be passed unchanged to hole A which has its own tolerance relative to hole B. If member XY be made very short and member YZ very long in comparison with other dimensions in the neighbourhood of holes A and B, then the path traced out by point Y will be very nearly the same as that traced out by point X. The displacement at X will be passed unchanged to Y in the limiting case. If point X is taken to represent the locating hole B and point Y is taken to represent the located hole A, then the required locating system has been obtained. The use of links of zero length such as XY, and links of infinite length, such as YZ, is common and there are several standard cases in which these are useful. The final location triad is shown in Fig. 10.

The whole assemblage network is made up from standard components similar to the one just described - another two-dimensional example is illustrated in Fig. 11. In order to generate a two-dimensional system, the P-loc is taken to be in the plane of the paper and the two-dimensional H-loc and S-loc representations are a slot and pivot as shown in Fig. 11.

Further discussion of the network will be found in Appendix D.

4.3 Generation of Tolerance Zones

The input tolerance zone at a feature is always taken to be a sphere. The output tolerance zone at the located feature may be linear, or elliptical or ellipsoidal depending on the matrix of the path joining the two features. The use of a spherical input tolerance zone is not as restrictive as it might appear, since all other common tolerance zones may be generated

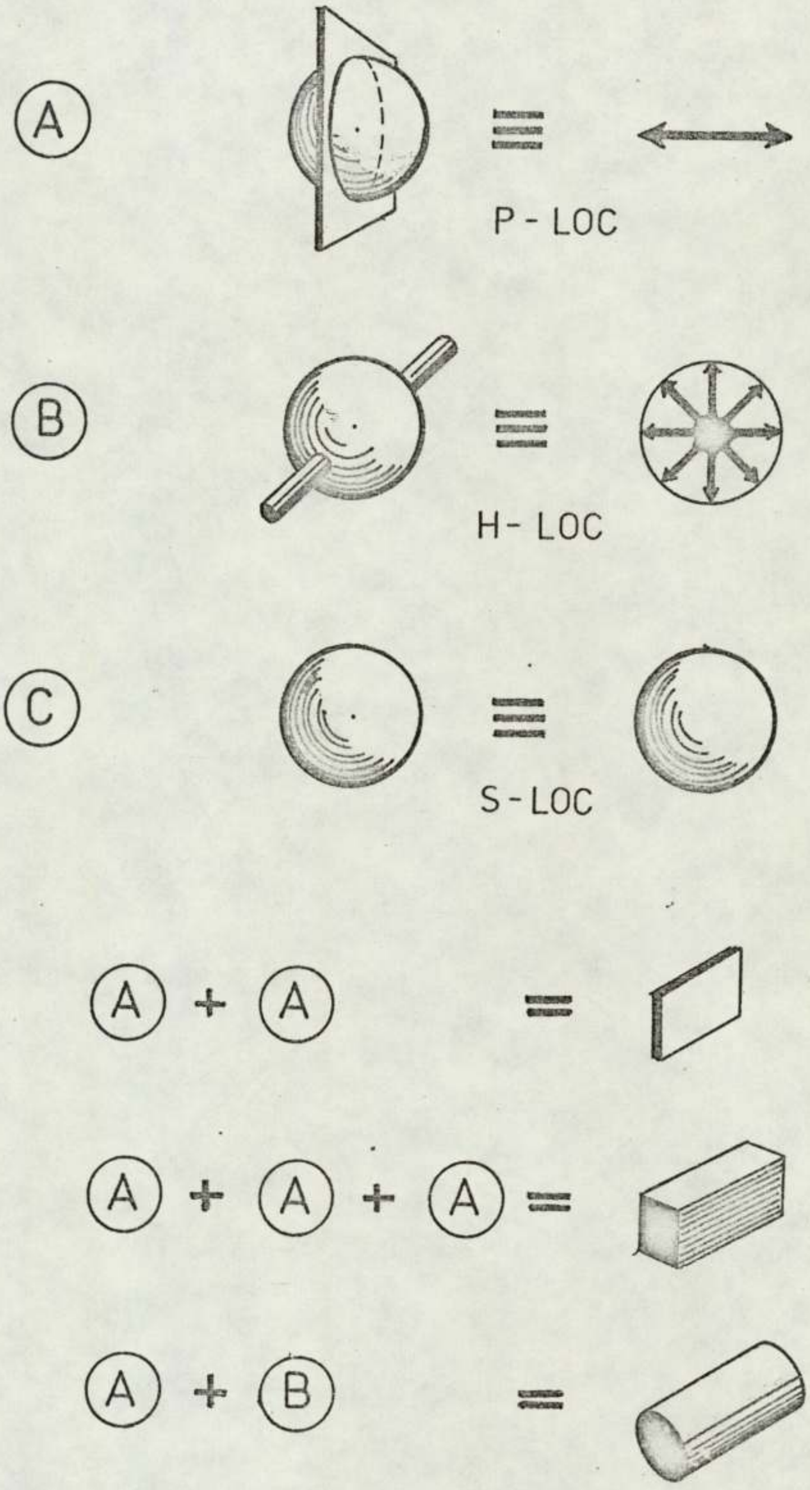


Fig 12 Generation of Tolerance Zones.

from it - see Fig. 12.

If a spherical tolerance is applied to a P-loc, then since only displacements normal to the locating plane are sensed, this is equivalent to a linear tolerance zone. At an H-loc, only displacements at right-angles to the locating line are sensed, so the application of a spherical zone is equivalent to a circular zone. At an S-loc, the whole spherical zone is sensed.

By superimposition of P-locs, a parallelepipedal tolerance zone may be generated. Similarly, superimposition of H- and P-locs generates a cylindrical zone.

A spherical tolerance zone may therefore be used to generate all the standard tolerance zones listed in BS 308 and quoted in section 2.3.1. A more detailed treatment is given in Appendix D.

4.4 Use of a location network

When a location network has been established for a particular assemblage, it may be used to provide qualitative answers to various questions of interest.

(a) The vector displacement at a locating feature is known. What is the effect at a located feature?

The output displacement may be found directly from the relation

$$\bar{D}_{out} = \bar{M}\,\bar{D}_{in}$$

$\bar{M}$ is the path matrix between the two features. If this is to be found automatically, it is essential that if there be no path between the two features, then $\bar{M} = 0$.

(b) If a spherical tolerance zone is applied at a locating feature, what will be the maximum displacement and its direction at a particular R-point on the body?

Again the path matrix $\bar{M}$ is calculated. The maximum displacement and its direction are evaluated by finding the dominant eigenvalue and associated eigenvector of the product of $\bar{M}$ and its transpose. Details of the method used are to be found in Appendix B, but it is unnecessary for the user to know anything about the method used.

(c) A feature of interest depends on several locating features. What will be the effect of unit tolerances at each feature?

$\bar{M}$ matrices are calculated for the paths between each locating feature and the R-point in question. The maximum tolerances for each are evaluated as outlined in (b) and displayed in a convenient way. The results may be used as an aid to the selection of manufacturing processes and locating systems.

(d) What is the relative displacement between two features of interest in the assemblage?

The calculation is performed using a device described in Appendix A which is again transparent to the user. The answer is useful in several ways. It may, for instance, be employed to calculate clearances between points on the assemblage. Another use might be to check whether a particular system of location used in the manufacturing process gives tolerances which are inside the bounds specified in the functional drawing of the part.

5. THE WORKING SYSTEM

5. The Working System

5.1 The Target Computer configuration

It has been assumed, in the design of the pilot system, that

(a) the computer available for engineering use is a bare 16K mini,

(b) the program is to be contained within 8K, the remainder of the store being reserved for array space,

(c) a compiler is available for a reasonably high level language such as ALGOL 60 or FORTRAN.

The selection of the computer is clearly of great importance in the system design; and it was decided at an early stage that the design basis should be the minimal configuration above. A useful network should contain around two hundred nodes and 8K would be sufficient to provide array space for this size of assemblage.

Since the data structure chosen is fairly complex, ideal languages would be ALGOL 68 or PL/1 since both provide reference variables so that data structures may be built up dynamically. It is, however, unlikely that either of these languages would be available on the minimum configuration selected. It was reluctantly decided that the linked structure would be held in array form, links being integer pointers to array elements. This is a common, although artificial, way of holding a linked structure, but it does have the clear advantage that a data structure may be output in a comprehensible form.

The prototype program was written in ALGOL 60 and it did, after some paring, fit into 8K of store. The computer used was a Marconi-Elliott 905 which has an excellent ALGOL 60 compiler with good error diagnostics. ALGOL 60 was used in preference to FORTRAN mainly because it is more suitable for the communication of algorithms. The fact that FORTRAN has no facilities for dynamic arrays is irrelevant, because in this application a fixed area of core is set aside for array space.

5.2 The assemblage network . . . computer representation

The first stage in the analysis of the location systems on an assemblage is to set up the network manually. It might be possible to automate this to some extent, since sub-networks for all the common cases of dimensions systems have been established (Appendix D). These might be stored as a library of standard cases in the computer, probably on backing store; and the relevant sub-system selected by means of an index. The appropriate feature coordinates would also be supplied together with linkage data. A section of network would then be linked into the main data structure together with node information. There are two main difficulties. Firstly, it would be necessary to maintain a dynamic data-structure and this is not conveniently achieved in the languages most commonly available for engineering applications. Secondly, in some sub-systems (for example, those defining symmetric tolerance), some of the coordinates of nodes internal to the sub-system are calculated from externally supplied coordinates, and so a library entry would consist not only of a piece of structure, but would also contain a section of code which would be handled rather like a macro definition. This is an interesting problem but it would complicate the system drastically. For this reason, the library of standard cases is held in a manual in the prototype system and the network completed by hand.

It would be unreasonable if it were required that a whole network were to be compiled by hand for an assemblage as complex as, say, a motor car. Fortunately, networks can be built up piecemeal from more tractable sub-networks which can be separately tested. The physical unit corresponding to such a sub-network might be as small as a single process drawing.

The present prototype program is for general purposes, but in a more elaborate configuration, a separate specialised program for validating sub-networks would be very useful. This might display selected output tolerance zones graphically for given input tolerances so that the correctness of a given sub-network could be checked before incorporation into the main system. The test program would also be a valuable training aid, particularly if output tolerance zones were displayed visually.

5.3 Data Input Format

The details of the network are supplied to the program by providing the data for each node. The format of the input data is described below.

(a) External node index

Each node represents a feature on the assemblage and must have a distinct index. These are provided in random order and the format may be designed to any fixed convention. In the prototype program, simple positive integers were used.

(b) Node type index

For a normal node, the type index is 0. In the case of a node with unitary links (see Appendix D), the type index is 1, such nodes being treated in a special way. Artificial nodes of this kind will have non-zero indices and although the unitary node is the only one included in the prototype program a good case might be made for using others, notably those connected with symmetric tolerances.

(c) Link indices

Each normal node will have three links, each pointing to another node in the network. Leaf nodes will have links pointing to a notional null node, indicated by zero. Artificial nodes are treated in a different way and the unitary node, for example, may have a mixture of zero and non-zero links.

A node may not have a link to itself - this will be rejected at the data validation stage of the program. The convention assumed for the order of the links is (i) P-loc link, (ii) H-loc link, (iii) S-loc link. Weak links (see Appendix D) are distinguished by a negative node number.

(d) Feature coordinates

All nodes have the following five coordinates: X, Y and Z coordinates relative to the general reference axes of the system, and two angles which define the directions of the locating plane for a P-loc and the locating line for an H-loc. These are specified in the prototype program as degrees and in cylindrical coordinates.

Some storage space is wasted by quoting angular coordinates for an S-loc. If these were not included the array structure would become more complicated. Another reason for including them is that it enables an S-loc to be used in a dual role as a P- or H-loc which might be useful for larger networks since this saves nodes.

(e) Tolerance size

The bilateral tolerance size (or radius of the generating spherical tolerance zone) may be included, if it is known, and if qualitative values of displacements are required. This was not done in the prototype, all tolerance zones were considered as being of unit size and the output interpreted as displacement per unit input tolerance or sensitivity co-efficient. Other information which might be useful here is the standard deviation of the process tolerance in the case of a well-established process. This would enable statistical confidence limits to be calculated for output tolerances as is done in the system described in ref. S.5.

5.4 Internal Node Data

(a) Internal node indices

It is desirable, although not essential, that node indices should be provided in random order on input. Networks for large assemblages are built up from smaller sub-networks and the onus of organising the feature references into a form suitable for computer processing is better put on the computer than on the user. The program assigns an internal index to each node which is held as part of the node record. This internal index is an integer with absolute value in the range 1 - N where N is the total number of nodes. Again, 0 is used as the null node and negative integers denote weak links. Internal node indices are assigned to each node in topological order. This is discussed at length in Appendix B but informally may be defined in the following way:

'If nodes are in topological order, then no node can have a link to a node with a lower index, except to node zero, which is a special case.'

Internal node indices may be used in two ways. Firstly, the node data may be sorted so that all the nodes are physically in topological order. Secondly, a node index vector might be held in store and all operations on nodes might be performed indirectly. Each method has its own merits; the former being faster for actual processing of an established network, but resulting in re-ordering of the prime data; the latter requiring that node access has a further degree of indirection which is particularly time-consuming unless the compiler uses Iliffe vector array access. In the prototype program, node records were topologically sorted.

(b) Scratch pad matrix

Associated with each node is a 3 x 3 matrix which has coefficients depending on the coordinates of the location triad of which the node is the R-point. This represents a large overhead of store, but it is difficult to see how processing of networks might be achieved efficiently without it.

(c) Direction cosine vector

For speed of execution, the two cylindrical coordinates which define the normal to the plane or the line of the hinge are converted to direction cosines which are held in the node record as a 3-element vector.

It is possible that some, or all, of the internal node data might be omitted and the associated quantities calculated as required during processing. As usual, the compromise must be made between minimising storage space and reducing running time. In the prototype program, it was decided that 200 nodes should be sufficient for most practical problems and

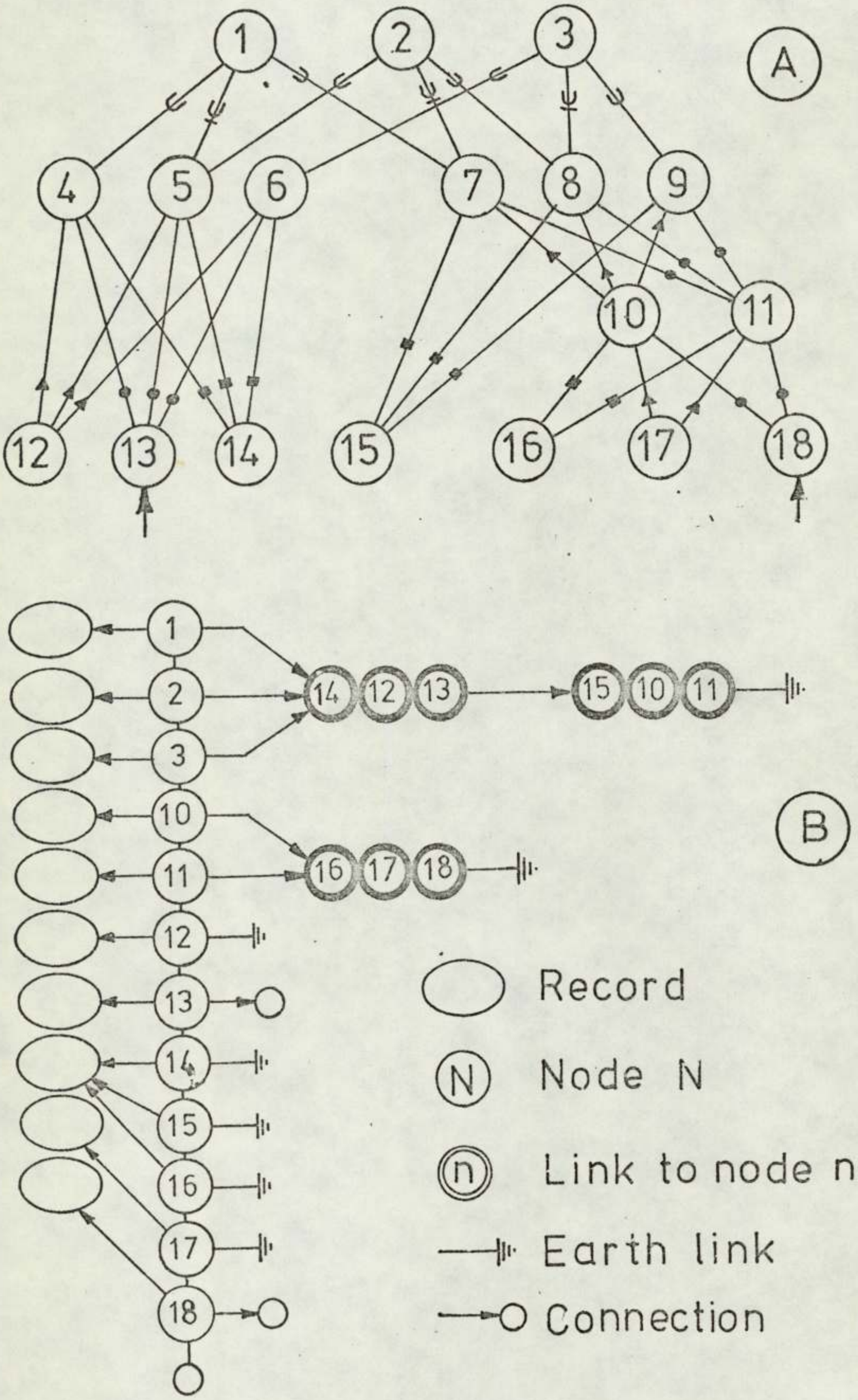


Fig 13 Alternative Data Structures.

so all internal data was included since sufficient array space was available for them.

5.5 The structure - alternatives

So far, the data structure has been referred to rather tentatively as 'a network'. There are several possibilities for the actual form of the data structure: two of these are particularly useful and a discussion of their respective merits follows:

(a) This is the most natural structure and an example will be found in Fig. 13A. The set of nodes is connected unidirectionally. Nodes have outdegree three except for the leaf nodes which have outdegree zero. One or more root nodes have indegree zero, while the remaining nodes are not restricted as to indegree. There are no closed loops. Although it is tempting to refer to the structure as a ternary tree, it would be incorrect to do so since all sub-structures are not disjoint (ref. G.1). A similar structure without the restriction on outdegree is termed 'a generalised arborescence' and 'a hierarchical structure' in ref. G.15. To avoid inventing another name equally as clumsy as these, the structure will inaccurately but concisely be called a 'tree' from now on.

(b) Another structure which is more flexible than the previous one is shown in Fig. 13B. The main advantage of this structure is that duplicated input data is avoided since each node in the structure does not carry directly its associated record, but merely a pointer to it. The structure is divorced from the prime data and so all calculations are indirect. Another clear advantage is that horizontal links represent superimposition and so unitary matrices are not required.

The second structure is appropriate where a language is used which has reference (ALGOL 68) or pointer (PL/1) variables. For more commonly used

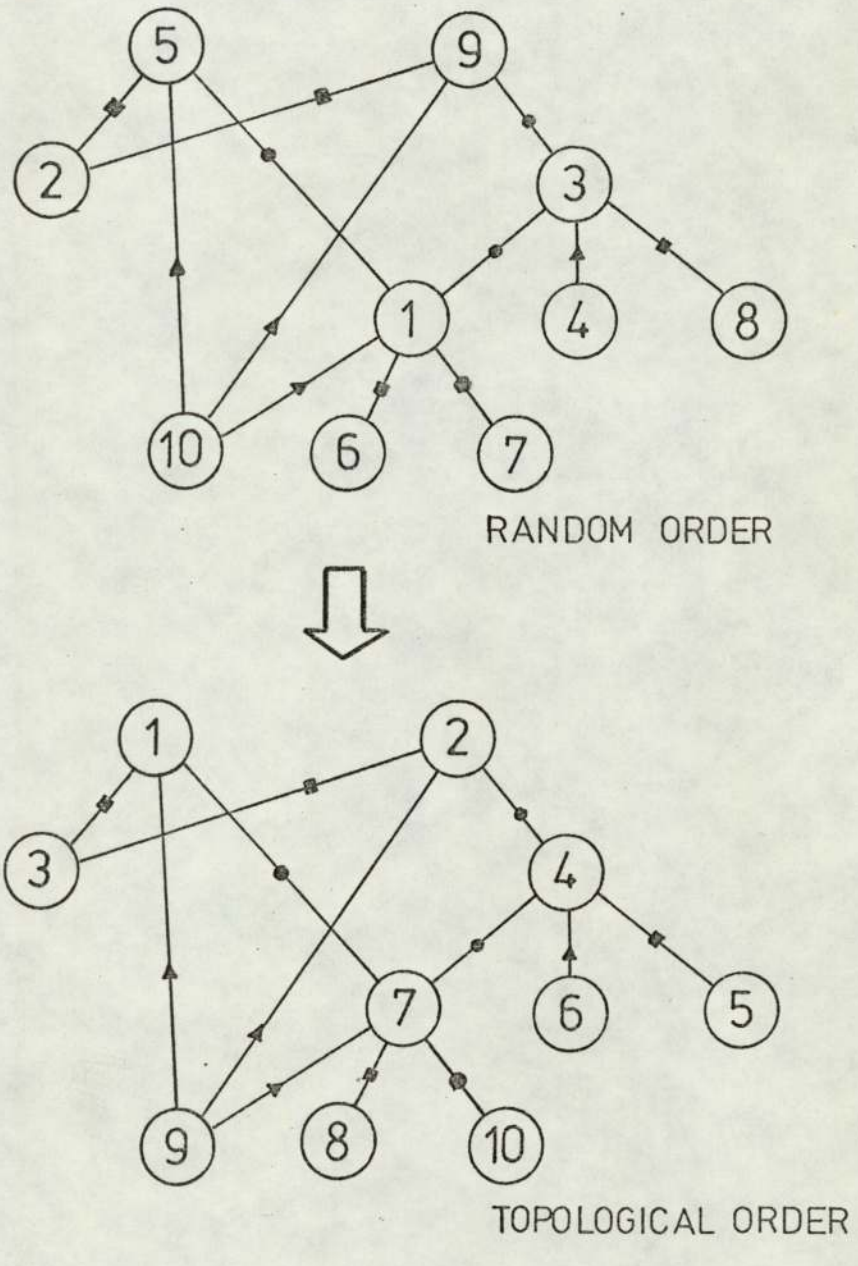


Fig 14 Topological Sorting

languages, the indirection involved is probably an intolerable overhead of processing time, and for this reason in the prototype program, the less compact but more natural 'tree' was used.

5.6 Data validation phase

The only data which can be directly checked for validity are the external node index and the node type index. The former must conform with a fixed format and the latter is restricted to a fixed number of integers (0 and 1 in the prototype program). In the prototype program, if N nodes were input, then each external node index would be a separate integer in the range 1 - N but in random order. This is not excessively restrictive on the user, but might be inconvenient where a large network was to be built up from smaller sub-'trees'.

At this stage, the 'tree' must be checked to ensure that it contains no closed loops which would be physically impossible and would result in the program looping. This may be done by topologically sorting the nodes, which also facilitates processing of the 'tree' at a later stage. Topological sorting is described in refs. G.1, G.13 and G.15; the algorithm used being a modification of the one described in ref. G.13.

Each node is re-numbered, node indices again being consecutive integers in the range 1 - N when N is the number of nodes. After sorting, a node (K) does not point, even indirectly to nodes (1) to (K - 1) - see Fig. 14.

The algorithm is described in Appendix B; it constructs a sort-index which is a vector of N elements showing the sorted position of each node. The sort-index is used to sort the data physically. Although this is not absolutely necessary, subsequent processing being possible by referring to the sort-index, it is convenient because -

(a) it is useful to separate the routines for setting up the structure from those used for processing it,

(b) the structure should be permanent after being validated and so the physical sort is only required once,

(c) much indirect referencing is obviated in subsequent processing, and

(d) subsequent programming is easier. The algorithm used for sorting on the index was also obtained from ref. G.13 (see Appendix B).

At the end of this phase, the data structure is ready for processing. The internal node index is the index of the node in topological order. It is convenient at this stage to convert the angular coordinates of P- and H-locs to direction cosines which are written into the direction cosine vector.

5.7 Processing the structure - prototype program

In the interests of conserving storage space, nodes were re-numbered during the sort phase. All nodes must be subsequently referred to by their new numbers in the prototype program. For this reason a sort-index is output at the beginning of the processing stage. The user must re-number the nodes on his 'tree' diagram with the help of this index. In a larger system, the original node numbers would still be available and an inverted list used to access the sorted node numbers which would only be used internally. The existing system is mildly inconvenient.

Two options only are available in the prototype program and are described below. For test purposes, the path matrix elements and the number of iterations required in the eigenvalue calculation may be output. These may be suppressed if, as is likely, they are not required.

(a) Maximum displacement

Required: a maximum sensitivity coefficient - i.e. the maximum displacement at a result feature caused by the application of a unit tolerance at an input feature.

Input (i) the result feature node number,

(ii) the input feature node number.

The 'tree' is traversed from result feature node to input feature node, path matrices being calculated cumulatively at each node encountered en route as described in Appendix A, section A4.

The maximum displacement is calculated as described in Appendix A, section A6, and output together with its associated direction cosines.

(b) Relative displacement

Required: the relative displacement between two result features due to a unit tolerance at an input feature which affects either, or both.

Input (i) the code 0,
(ii) the two result feature node numbers,
(iii) the input feature node number.

A dummy node is attached to the two result points and the 'tree', of which it is the root node, is traversed. This device is described in Appendix A, section A4. The relative displacement is calculated and output with its associated direction cosines which are used merely for checking purposes.

5.8 Further extensions

The options described in section 5.7 are sufficient for normal use, but several more may be added for convenience. Two of these are:

(a) Calculation of all the sensitivity coefficients at a result feature.

Since all the nodes at which tolerances occur are known to the user, these may be flagged on input to the system. A list of sensitivity coefficients may be obtained by traversing the 'tree' and calculating maximum displacements at each input node. In order to do this efficiently and to obviate superfluous output, a more elegant method of traversal is desirable. This would require an appreciable increase in the size of the program and so was omitted from the prototype program.

(b) Calculation of the maximum displacement in a particular direction.

This is done easily from the paths matrix and since it is a straightforward calculation by hand, it was omitted from the prototype program.

5.9 An Integrated Tolerance Control System

The system described in sections 5.1 - 5.7 is designed as a stand-alone program for a mini-computer. Its input is a network description of an assemblage together with a list of features of interest and the points having tolerances which affect them. The output is a list of sensitivity coefficients for each feature of interest. This is a useful tool for the analysis of tolerances in its present form. However, if a more ambitious configuration were available, several other sub-programs involving established techniques could be amalgamated to form an integrated tolerance control system. In view of the interest displayed in the system described in ref. S.5 which also assists the designer in part of the analysis of dimensional tolerancing, an integrated system would be an invaluable aid in this field.

A possible configuration might consist of the following modules:

(a) Network Proving Sub-program

As each component of an assemblage were considered, its individual location networks might be separately proved by using a specially tailored version of the prototype program. Ideally, interactive graphics would be used to display the envelope of the output tolerance zone for one input tolerance zone or several acting simultaneously. This would enable the user to prove the sub-network to his own satisfaction. A set of standard sub-networks, such as those given in Appendix D could be stored and displayed on demand individually, and tested interactively ensuring that the case selected was appropriate to the location situation. The catalogue of standard cases would be augmented by cases which had been thoroughly proven. It would also be convenient to display information regarding the purpose and usage of each standard sub-network on demand.

(b) Network Building Sub-program

A difficulty with the current prototype program is that if the sensitivities obtained for a particular application are not satisfactory, then the complete network must be modified and re-input. This is due in some measure to the restricted core available on the target configuration, but also because Algol 60 is not a suitable language for handling data structures of any complexity. If Algol 68 or PL/1 were available then the structures could be dynamically modified and it would be possible to delete sections of network, to insert modified sub-networks and to append proven sub-networks to an existing network. A complex assemblage network could be built up section by section interactively, which is a more natural method of developing the data structure.

(c) Sensitivity Coefficient Sub-program

The next stage in the design process would be to process the established network and obtain sensitivity coefficients for all tolerances affecting points of interest. This would be a refined version of the prototype program; an obvious improvement being to trade off some storage space for a quicker and more elegant method of traversing paths in the network. The output from this sub-program would be lists of sensitivity coefficients for each critical feature in the assemblage. Possibly, some of these might be sufficiently low to be ignored and the network might then be re-defined omitting them in the interests of running efficiency.

(d) Allocation Sub-program

Eventually, a stage would be reached when the designer was content with the assemblage description. The tolerance allocation could then be optimised on a least cost basis. Two options would be available: statistical and surefit bases (see Appendix C). This would be a logically straightforward section but judging from the variety of the methods available for non-linear optimisation, it would probably require study by a specialist in the field. The output from this sub-program would be the actual tolerances at each input point. It is possible that some of these

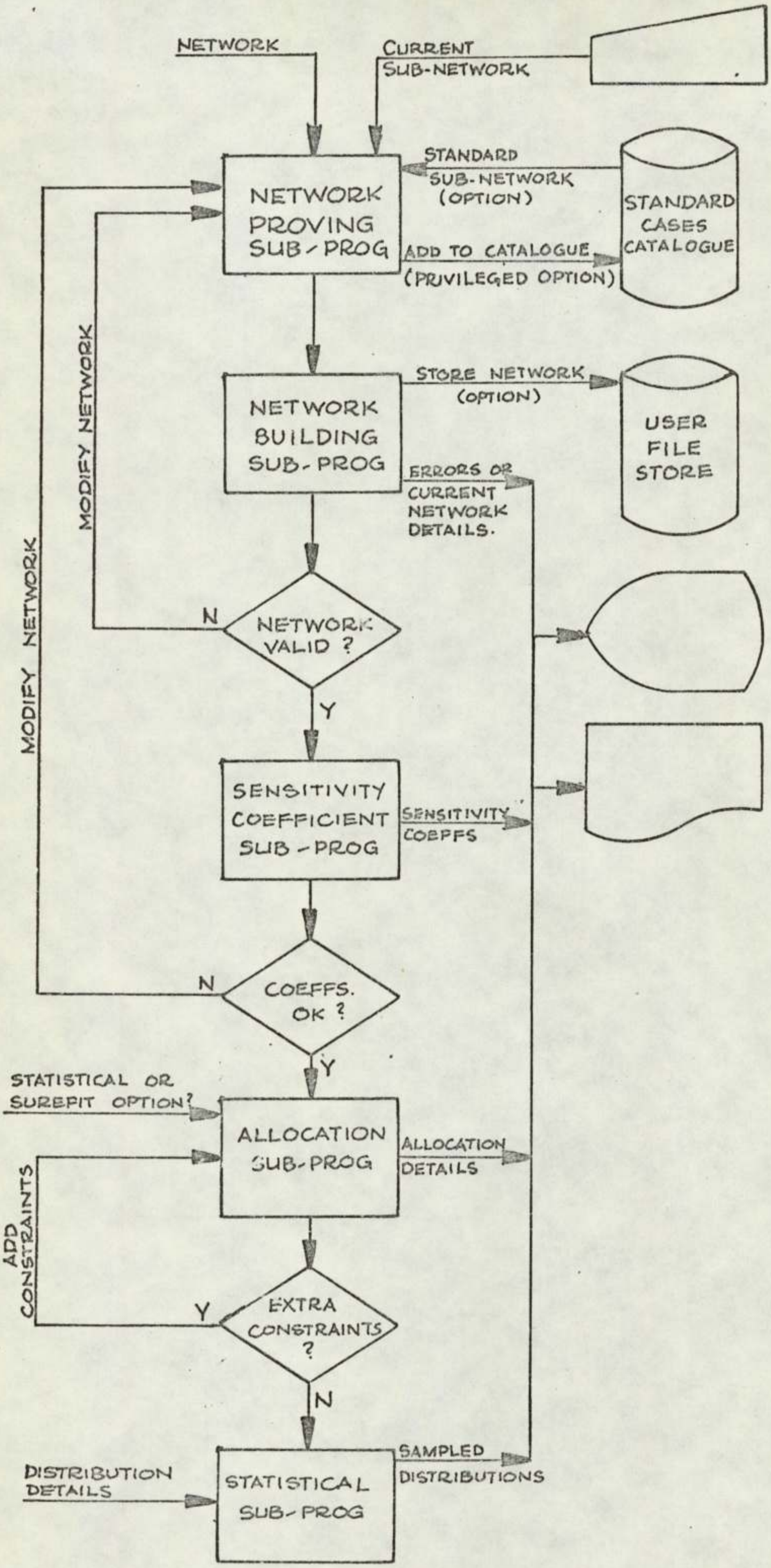


Fig 15. An Integrated Tolerance Control System.

might be too small for practical considerations.necessitating further constraints to be applied (see Appendix C) and subsequent reprocessing and recalculation.

(e) Statistical Analysis Sub-program

The final sub-program in the system would be a system similar to the one described in ref. S.5 but operating on the output from section (d) rather than on sub-programs and data supplied by the user. The results which could be in histogram form would give the distribution of the variations in size on features of interest.

A suggested system is illustrated in Fig. 15.

5.10 Comments on the System

Much of the preliminary work has been done with dimensioning equally as eccentric as that on the examples discussed in Appendix E.1. A problem in applying the method to practical examples is that most designed components are under-defined dimensionally, occasionally even in critical measurements and assumptions must be made particularly with regard to tolerances such as squareness, flatness and parallelism which are normally not specified explicitly. Usually it is assumed, even by experienced detail designers, that some features of a component are geometrically exact. A location network is certainly a more precise method of specifying a part than most dimensioned drawings.

For most of the applications which have been checked analytically, the sensitivity coefficients obtained have been accurate to two decimal places even when dimensions have been scaled from a drawing. Occasionally it is difficult to check a particular network and interactive graphics would be a great help.

The method is reasonably easy to use after a little practice. To date, a sub-network has been found for every dimensioning system encountered and the method should be particularly suitable for the use of engineering designers, who are normally good at visualising mechanisms. Since assemblies are represented by real, rather than the more complex geometrical, locations, applying the method to assemblies is very easy.

The restrictions of the target configuration have resulted in the system being a little inconvenient to use. Networks are best developed bit by bit in a similar way to that described in the examples but due to limited core it was necessary to keep the program as short as possible and it was not feasible to generate major networks dynamically. As each sub-network is proved, it is necessary to modify the network manually and this is then re-presented to the program.

It would seem from work done so far that this is a powerful method for analysing small displacements and it may have applications other than tolerancing. Some preliminary investigation has been performed on the analysis of kinematic mechanisms, and this seems promising.

Three-dimensional kinematic mechanisms are easier to model using the system than are tolerance mechanisms and the program may be used in its present form to determine instantaneous velocities of links in mechanisms. This has been done successfully in a variety of cases and further work is being carried out on the analysis of accelerations. Some recent papers have described efforts to analyse tolerance at joints in mechanisms; it seems that the system is useful for this purpose.

APPENDIX A

ANALYSIS OF THE MODEL

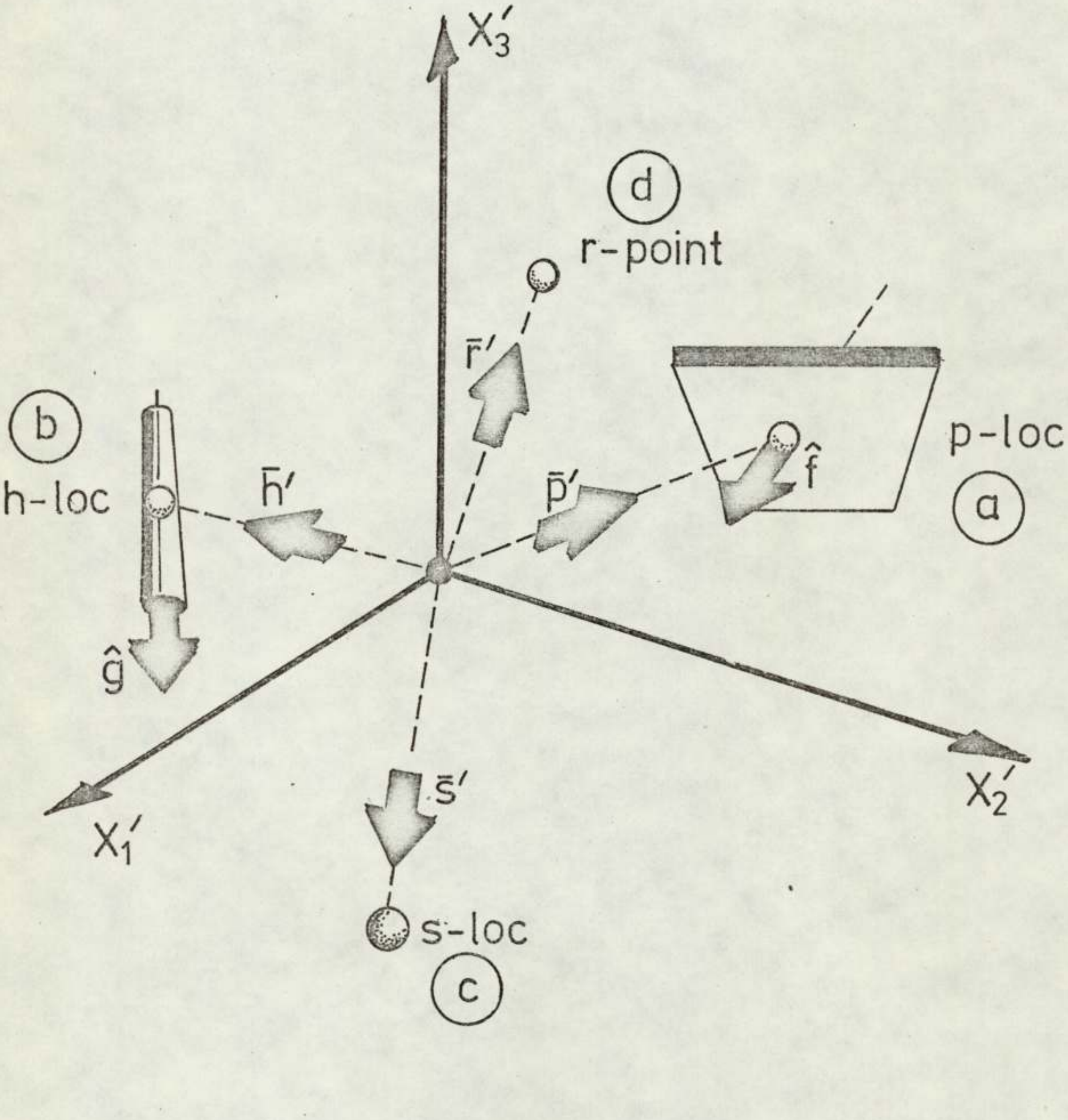


Fig A1. A Location Triad.

A.1 Analysis of the Location Triad

The body is located in a set of mutually orthogonal right handed axes $\left\{x_1', x_2', x_3'\right\}$ and is constrained as follows:

(i) At a point on a plane (P-loc). Fig. A1a.

$\bar{p}'$ is the position vector of the point of application on the plane.

$\hat{f}$ is the unit normal to the plane.

(ii) At a point on a line (H-loc). Fig. A1b.

$\bar{h}'$ is the position vector of the point of application on the line.

$\hat{g}$ is the unit vector along the line.

(iii) At a point (S-loc). Fig. A1c.

$\bar{s}'$ is the position vector of the point.

A general point on the located body (R-point) has position vector $\bar{r}'$. Fig. A1d.

Energy methods will be used to obtain displacements, and forces at P-, H- and S-locs, and the R-point are $\bar{P}$, $\bar{H}$, $\bar{S}$ and $\bar{R}$ respectively.

The system may be described by vector equations (i) - (iv):

(i) $\bar{P} + \bar{H} + \bar{S} + \bar{R} = \bar{0}$

(ii) $\bar{P} \times \bar{p}' + \bar{H} \times \bar{h}' + \bar{S} \times \bar{s}' + \bar{R} \times \bar{r}' = \bar{0}$

(iii) $\hat{g} \cdot \bar{H} = 0$

(iv) $\bar{P} = |\bar{P}|\,\hat{f}.$

Equations (i) and (ii) are the general equilibrium equations, vector sums of forces and moments being zero. Equation (iii) represents the condition that force $\bar{H}$ is at right-angles to the H-loc. Equation (iv) represents the condition that force $\bar{P}$ acts normal to the P-loc.

Known values are:

P-loc: $\bar{p}'$ and $\hat{f}$.

H-loc: $\bar{h}'$ and $\hat{g}$.

S-loc: $\bar{s}'$

R-point: $\bar{R}$ and $\bar{r}'$

Required:

P-loc: $|\bar{P}|$

H-loc: $\bar{H}$

S-loc: $\bar{S}$

Solution:

(a) The equations (i) - (iv) are transformed by changing the coordinate axes to $\{x_1, x_2, x_3\}$ a parallel system with the S-loc as the origin. Position vectors will be modified as follows:

$$\bar{p} = \bar{p}' - \bar{s}'$$

$$\bar{h} = \bar{h}' - \bar{s}'$$

$$\bar{s} = \bar{0}$$

$$\bar{r} = \bar{r}' - \bar{s}'$$

Equation (ii) now becomes

$$\text{(iia)} \quad \bar{P} \times \bar{p} + \bar{H} \times \bar{h} + \bar{R} \times \bar{r} = \bar{0}$$

(b) Taking the scalar product of $\bar{h}$ with (iia)

$$\bar{h} \cdot (\bar{P} \times \bar{p} + \bar{H} \times \bar{h} + \bar{R} \times \bar{r}) = 0$$

$$= \bar{h} \cdot (\bar{P} \times \bar{p}) + \bar{h} \cdot (\bar{R} \times \bar{r}) \quad \text{since generally } \bar{h} \cdot (\bar{H} \times \bar{h}) = 0$$

$$= |\bar{P}|\ \bar{h} \cdot (\hat{f} \times \bar{p}) + \bar{h} \cdot (\bar{R} \times \bar{r}) \text{ since } \bar{P} = |\bar{P}|\ \hat{f} \text{ (equation iv).}$$

$$\text{Finally, } |\bar{P}| = - \frac{\bar{h} \cdot (\bar{R} \times \bar{r})}{\bar{h} \cdot (\hat{f} \times \bar{p})}\ ; \quad \bar{h} \cdot (\hat{f} \times \bar{p}) \neq 0.$$

(c) Taking the vector product of $\hat{g}$ with (iia).

$$\hat{g} \times (\bar{P} \times \bar{p} + \bar{H} \times \bar{h} + \bar{R} \times \bar{r}) = \bar{0}$$

$$= \hat{g} \times (\bar{P} \times \bar{p}) + (\hat{g} \cdot \bar{h})\bar{H} - (\hat{g} \cdot \bar{H})\bar{h} + \hat{g} \times (\bar{R} \times \bar{r})$$

since generally $\bar{a} \times (\bar{b} \times \bar{c}) = (\bar{a} \cdot \bar{c})\bar{b} - (\bar{a} \cdot \bar{b})\bar{c}$

$$= \hat{g} \times (\bar{P} \times \bar{p}) + (\hat{g} \cdot \bar{h})\bar{H} + \hat{g} \times (\bar{R} \times \bar{r})$$

since $\hat{g} \cdot \bar{H} = 0$ from equation (iii).

$$\text{Finally, } \bar{H} = - \frac{\hat{g} \times (\bar{P} \times \bar{p} + \bar{R} \times \bar{r})}{\hat{g} \cdot \bar{h}}\ ; \quad \hat{g} \cdot \bar{h} \neq 0.$$

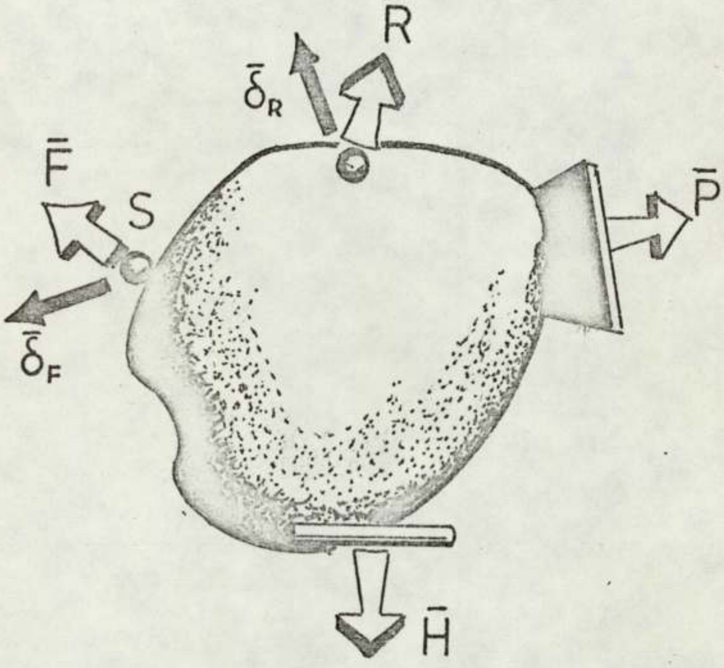


Fig A2. A Located Body

(d) From equation (i)

$$\bar{S} = -(\bar{P} + \bar{H} + \bar{R})$$

This completes the solution of equations (i) - (iv).

Summarising:

Solutions:

$$\text{(v)} \quad |\bar{P}| = -\frac{\bar{h} \cdot (\bar{R} \times \bar{r})}{\bar{h} \cdot (\hat{f} \times \bar{p})}$$

$$\text{(vi)} \quad \bar{H} = -\frac{\hat{g} \times (\bar{P} \times \bar{p} + \bar{R} \times \bar{r})}{\hat{g} \cdot \bar{h}}$$

$$\text{(vii)} \quad \bar{S} = -(\bar{P} + \bar{H} + \bar{R})$$

Conditions

$$\text{(viii)} \quad \bar{h} \cdot (\hat{f} \times \bar{p}) \neq 0$$

$$\text{(ix)} \quad \hat{g} \cdot \bar{h} \neq 0$$

Conditions (viii) and (ix) describe a proper location system.

A.2 Displacement at a Result Point

Fig. A2 shows a body located on the triad PHS. If a displacement $\bar{\delta}_F$ be applied to the locating feature F (which might be any one of P, H or S) then there will be a resulting displacement $\bar{\delta}_R$ at the R-point. If an arbitrary force $\bar{R}$ be applied at the R-point, then forces $\bar{P}$, $\bar{H}$ and $\bar{S}$ will result at the P-, H- and S-locs. These forces may be found from equations (v) - (vii) and in particular the force at F will be $\bar{F}$.

From energy considerations:

$$\bar{F} \cdot \bar{\delta}_F + \bar{R} \cdot \bar{\delta}_R = 0$$

The general locating feature F has a vector displacement $\{\xi_1, \xi_2, \xi_3\}$ where the subscripts here and in subsequent expressions denote components in the x_1, x_2 and x_3 directions respectively. The corresponding displacement at the R-point is $\{\eta_1, \eta_2, \eta_3\}$.

To find displacement component η_1, say, a unit force in the x_1 direction is applied at R. The resulting force at F may be written as

$\overline{F}_1 = \left\{F_{11}, F_{12}, F_{13}\right\}$, the first subscript denoting the direction of the unit force at R, the second denoting the component direction at F.

Similarly unit forces in the x_2 and x_3 directions yield the equations

$$\overline{F}_2 = \left\{F_{21}, F_{22}, F_{23}\right\}$$

$$\overline{F}_3 = \left\{F_{31}, F_{32}, F_{33}\right\}.$$

From energy considerations:

$$\left\{\eta_1, \eta_2, \eta_3\right\} = - \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} \left\{\xi_1, \xi_2, \xi_3\right\}$$

— or using the tensor suffix convention:

$$\eta_i = - F_{ij} \; \xi_j ; \quad i, j = 1, 2, 3$$

The F_{ij} may be found from equations (v) – (vii) using R_i in place of the general R.

$$\overline{P}_i = - \frac{\overline{h} \cdot (\overline{R}_i \times \overline{r})}{\overline{\overline{h}} \cdot (\hat{f} \times \overline{\overline{p}})} \hat{f}$$

$$\overline{H}_i = - \frac{\hat{g} \times (\overline{P}_i \times \overline{p} + \overline{R}_i \times \overline{r})}{\hat{g} \cdot \overline{h}}$$

$$\overline{S}_i = - (\overline{P}_i + \overline{H}_i + \overline{R}_i)$$

where subscript i denotes the ith row vector of the matrices subscripted and $\overline{R}$ is the unitary matrix.

It is useful to abandon vector notation at this stage since it is no longer convenient. Using the suffix summation convention, and the operators

Kronecker delta: $\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$

Permutation operator $\epsilon_{ijk} = \begin{cases} 1 & \text{for } ijk = 123, 231, 312. \\ -1 & \text{for } ijk = 132, 213, 321. \\ 0 & \text{otherwise,} \end{cases}$

the equations may be written in the compact form:

$$\text{(x)} \quad P_{ij} = -\frac{\epsilon_{rit}\, h_r\, r_t\, f_j}{K}$$

$$\text{(xi)} \quad H_{ij} = \frac{\epsilon_{rit}\, h_r\, r_t\, (g_d p_d f_j - g_e f_e p_j) - K(g_f r_f\, \delta_{ij} - g_i r_j)}{K g_g h_g}$$

$$\text{(xii)} \quad S_{ij} = -(P_{ij} + H_{ij} + \delta_{ij})$$

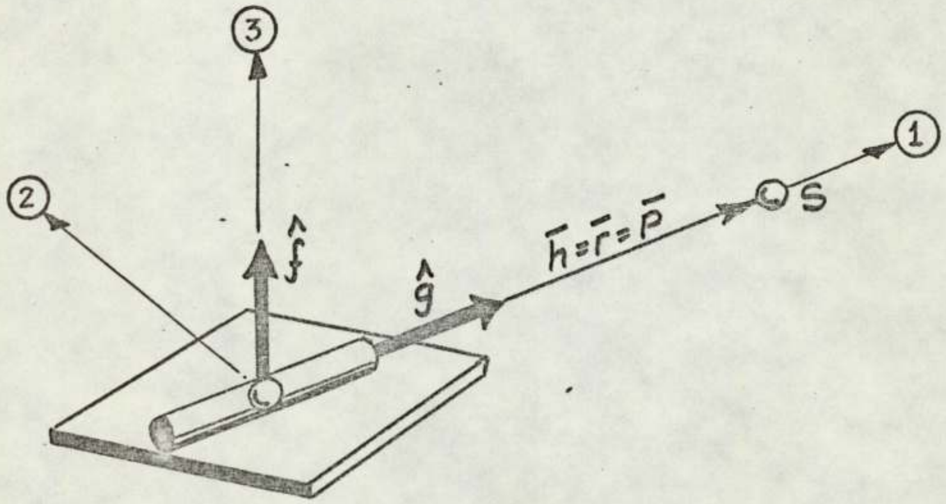
where $K = \epsilon_{abc}\ h_a f_b p_c$.

The subscripts in (xi) may be simplified a little and (xii) may be expanded, but the forms given are the most useful. When these expressions are used in the analysis of location 'trees', they are modified by being multiplied by −1, so that the general displacement equation may be written as

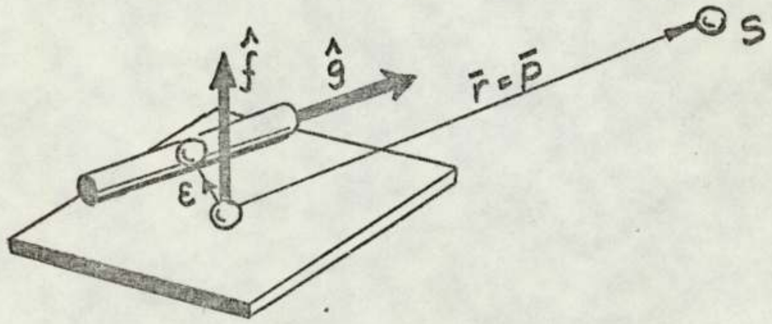
$$\eta_i = F_{ij}\ \xi_j \qquad i,\ j = 1,\ 2,\ 3.$$

This is purely a matter of convenience.

Although subscript notation is very convenient for algebraic manipulation, it did not prove very efficient for calculation of the matrix coefficients on the computer. The main reason for this was that in calculating the values of the permutation operators, most of the time was spent in evaluating zero coefficients. Special purpose algebraic manipulation languages have been implemented which, it is claimed, can evaluate expressions of this kind efficiently and conveniently (e.g. MATHLAB) but as most designers have access to the more mundane ALGOL 60 and FORTRAN, the vector approach was used in the prototype program. As usual, it was apparent that the commoner computer languages are not very suitable for mathematical purposes.



R, H & P coincident



H perturbed by vector $\bar{\varepsilon}$.
R & P coincident.

Fig A3. Coincident Features.

A.3 Conditions for a proper Location Triad

The conditions for proper locations are summarised by the inequalities:

(i) $\hat{g} \cdot \bar{h} \neq 0$,

(ii) $\bar{h} \cdot (\hat{f} \times \bar{p}) \neq 0$.

(i) and (ii) imply that

(a) the S-loc must not coincide with the H- or P-locs,

(b) the line joining the points of application of the S- and H-locs must not be perpendicular to the line of action of the H-loc, and

(c) the normal to the plane of the P-loc and the lines joining the S-loc and the H- and P-locs must not be coplanar.

There is no restriction on the position of the R-point. Occasionally, it is useful to employ limiting cases of (i) and (ii), an example being illustrated in Fig. A3. The P- and H-locs, and the R-point are coincident, the normal to the P-loc is perpendicular to the line of the H-loc, the line of the H-loc lies along the lines joining them with the S-loc. Since the H- and P-locs are coincident, condition (ii) is not observed. All the location matrices will have infinite coefficients. Use can be made of this system, however, if the H-loc is slightly displaced, so that $\bar{h}$ becomes $\bar{h} + \bar{e}$. In the case illustrated:

$\bar{r} = \bar{p} = \lambda \hat{g}$ where λ is a scalar

and $f_1 = f_2 = g_2 = g_3 = 0; \quad f_3 = g_1 = 1.$

$$P_{ij} = - \frac{\epsilon_{rit}(h_r + e_r)\,\lambda g_t\, f_j}{\epsilon_{abc}(h_a + e_a) f_b\, \lambda\, g_c}$$

$$= - \frac{\epsilon_{rit}\, e_r\, g_t\, f_j}{\epsilon_{abc}\, e_a\, f_b\, g_c}$$

$$P_{ij} = - \frac{\epsilon_{ri1}\, e_r\, f_j}{e_2}$$

$$\text{i.e. } P_{ij} = \begin{cases} - \dfrac{\epsilon_{ri1}\, e_r}{e_2} & \text{for } j = 3 \\ 0 & \text{for } j \neq 3 \end{cases}$$

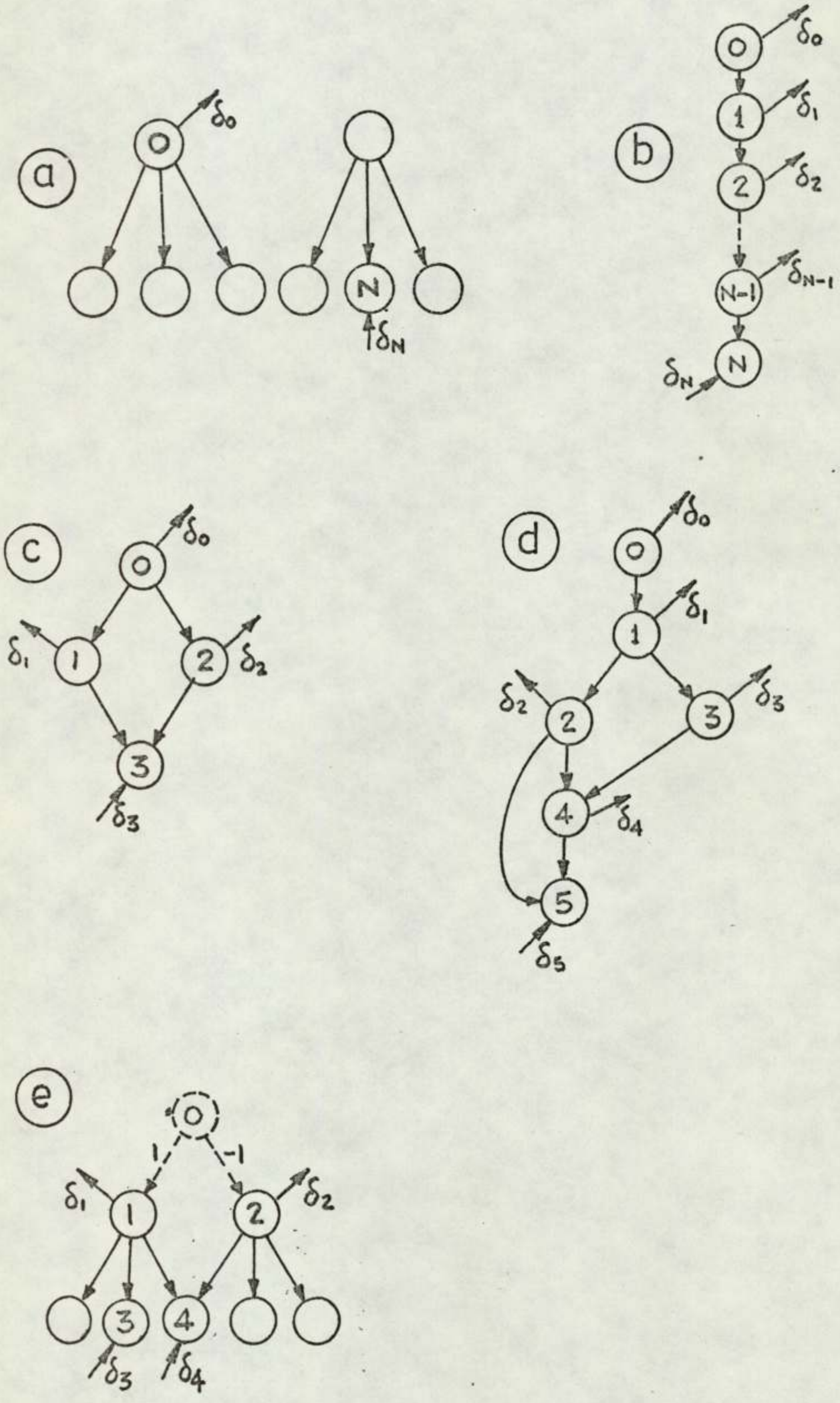


Fig A4. Network Paths.

If the H-loc is perturbed in the plane of the P-, S-locs and R-point, then

$$P_{ij} = \begin{cases} -1 & \text{for } i = j = 3 \\ 0 & \text{for } i,j \neq 3 \end{cases}$$

The equation for H_{ij} may be written in the form:

$$H_{ij} = - \frac{g_d p_d P_{ij} - g_e P_{ie} p_j + g_f r_f \delta_{ij} - g_i r_j}{g_h h_h}$$

$$= - \frac{g_1 p_1 P_{ij} - g_1 P_{i1} p_j + g_1 r_1 \delta_{ij} - g_i r_j}{g_1 h_1}$$

or $$H_{ij} = - \frac{g_1 p_1 P_{ij} + g_1 r_1 \delta_{ij} - g_i r_j}{g_1 h_1}$$

which reduces to

$$H_{ij} = \begin{cases} -1 & \text{for } i = j = 2 \\ 0 & \text{for } i,j \neq 2 \end{cases}$$

Similarly,

$$S_{ij} = \begin{cases} -1 & \text{for } i = j = 1 \\ 0 & \text{for } i,j \neq 1 \end{cases}$$

This limiting case depends not only on the coordinates of the locations, but also on the direction of the perturbation. Similar analyses may be performed for all the cases in which this technique is used.

This system is useful in passing three orthogonal displacements to a result point, and may be used in the generation of a general parallelepipedal tolerance zone.

A.4 Path matrix products

(a) Disjoint case Fig. A4a.

A displacement δ_N is applied at node N. It is required to find the displacement δ_0 at node 0. Since node 0 is not located, even indirectly, on node N, the equation

$\delta_0 = F_{0N} \delta_N$ requires that in this case where no path exists between nodes 0 and N, $F_{0N} = \bar{0}$.

(b) Simple path Fig. A4b.

A displacement δ_N is applied to node N, resulting in a displacement δ_{N-1} at node N-1. δ_{N-1} will cause a displacement at node N-2 and so on along the path joining nodes N and 0.

$$\delta_0 = F_{01} \delta_1$$
$$\delta_1 = F_{12} \delta_2$$
$$\vdots$$
$$\delta_{N-1} = F_{N-1\ N} \delta_N$$
$$\delta_0 = F_{01} F_{12} \cdots F_{N-1,N} \delta_N$$

In the evaluation of a simple path matrix, the matrices corresponding to the links on the path are multiplied.

(c) Multiple paths Fig. A4c.

A displacement δ_3 is applied at node 3 and the displacement δ_0 is required.

$$\delta_1 = F_{13} \delta_3$$
$$\delta_2 = F_{23} \delta_3$$
$$\delta_0 = F_{01} \delta_1 + F_{02} \delta_2$$
$$\delta_0 = (F_{01} F_{13} + F_{02} F_{23}) \delta_3 = F_{03} \delta_3$$

The path matrix products are evaluated separately and added to give the nett path matrix product F_{03}.

(d) Multiple paths Fig. A4d.

A displacement δ_5 is applied at node 5 and again the displacement δ_0 is required.

$$\delta_0 = F_{01} \delta_1$$
$$\delta_1 = F_{12} \delta_2 + F_{13} \delta_3 \text{ by superposition.}$$

$$\delta_2 = F_{24}\,\delta_4 + F_{25}\,\delta_5 \quad \text{by superposition.}$$

$$\delta_3 = F_{34}\,\delta_4$$

$$\delta_4 = F_{45}\,\delta_5$$

Back-substitution gives

$$\begin{aligned}\delta_0 &= (F_{01}\,F_{12}\,F_{24}\,F_{45} + F_{01}\,F_{12}\,F_{25} + F_{01}\,F_{13}\,F_{34}\,F_{45})\,\delta_5 \\ &= F_{05}\,\delta_5.\end{aligned}$$

Again, path matrix products are evaluated separately and added to give the nett path matrix product F_{05}. In this case, it is most convenient to calculate partial matrix products at each node moving down from the R-point.

At node 1, product = F_{01}

At node 2, product = $F_{01}\,F_{12}$

At node 3, product = $F_{01}\,F_{13}$

At node 4, product = $F_{01}\,F_{12}\,F_{24} + F_{01}\,F_{13}\,F_{34}$

At node 5, product = $F_{01}\,F_{12}\,F_{25} + F_{01}\,F_{12}\,F_{24}\,F_{45} + F_{01}\,F_{13}\,F_{34}\,F_{45}$

(e) Relative displacements Fig. A4e.

A displacement δ_4 is applied at node 4. The displacement of node 1 relative to that of node 2 is required.

A method is needed to evaluate

$$\delta_1 - \delta_2 = (F_{14} - F_{24})\,\delta_4 \quad \text{conveniently.}$$

The method which will be used is to attach a dummy node 0 to nodes 1 and 2 as shown in the figure. If F_{01} is set to the unitary matrix δ_{ij} and F_{02} is set to $-\delta_{ij}$, then the result will be obtained by the methods earlier described in (a) – (d).

$$\begin{aligned}\delta_0 &= ((\delta_{ij}\,F_{14}) + (-\delta_{ij}\,F_{24}))\,\delta_4 \\ &= (F_{14} - F_{24})\,\delta_4.\end{aligned}$$

The method also works for the case where there is no path between the input node and one, or two of the output nodes. This case is illustrated in Fig. A4e, the input being applied at node 3.

Some useful results will now be derived.

A.5 Matrix Rank

A.5.1 The rank of a P-matrix

Since $$P_{ij} = \frac{\epsilon_{rit} h_t r_r f_j}{\epsilon_{abc} h_a f_b p_c}$$

$$\bar{P}_1 = \frac{\epsilon_{r1t} h_t r_r}{\epsilon_{abc} h_a f_b p_c} \bar{f}$$

$$\bar{P}_2 = \frac{\epsilon_{r2t} h_t r_r}{\epsilon_{abc} h_a f_b p_c} \bar{f}$$

$$\bar{P}_3 = \frac{\epsilon_{r3t} h_t r_r}{\epsilon_{abc} h_a f_b p_c} \bar{f} .$$

P is of rank 1 at most, since each row vector is a multiple of vector $\bar{f}$.

A.5.2 The rank of an H-matrix

It has been proved (equation xi) that:

$$H_{ij} = \frac{\epsilon_{rit} h_r r_t (g_d p_d f_j - g_e f_e p_j) - K(g_f r_f \delta_{ij} - g_i r_j)}{K g_g h_g}$$

Multiplying each element in(xi) by r_i and considering each term in the numerator in turn:

(a) $\epsilon_{rit} h_r r_t r_i$ may be summed over i.

$$\sum_i \epsilon_{rit} h_r r_t r_i = -\sum_i \epsilon_{rit} h_r r_i r_t,$$

if i and t be interchanged.

$$\therefore \sum_i \epsilon_{rit} h_r r_t r_i \equiv 0$$

(b) $(g_f r_f r_i \delta_{ij} - g_i r_i r_j)$ may be summed over i.

$$\sum_i (g_f r_f r_i \delta_{ij} - g_i r_i r_j) = \sum_i (g_f r_f r_j - g_i r_i r_j) = 0.$$

Hence $$r_1 \bar{H}_1 + r_2 \bar{H}_2 + r_3 \bar{H}_3 \equiv 0$$

and an H-matrix is of rank at most 2.

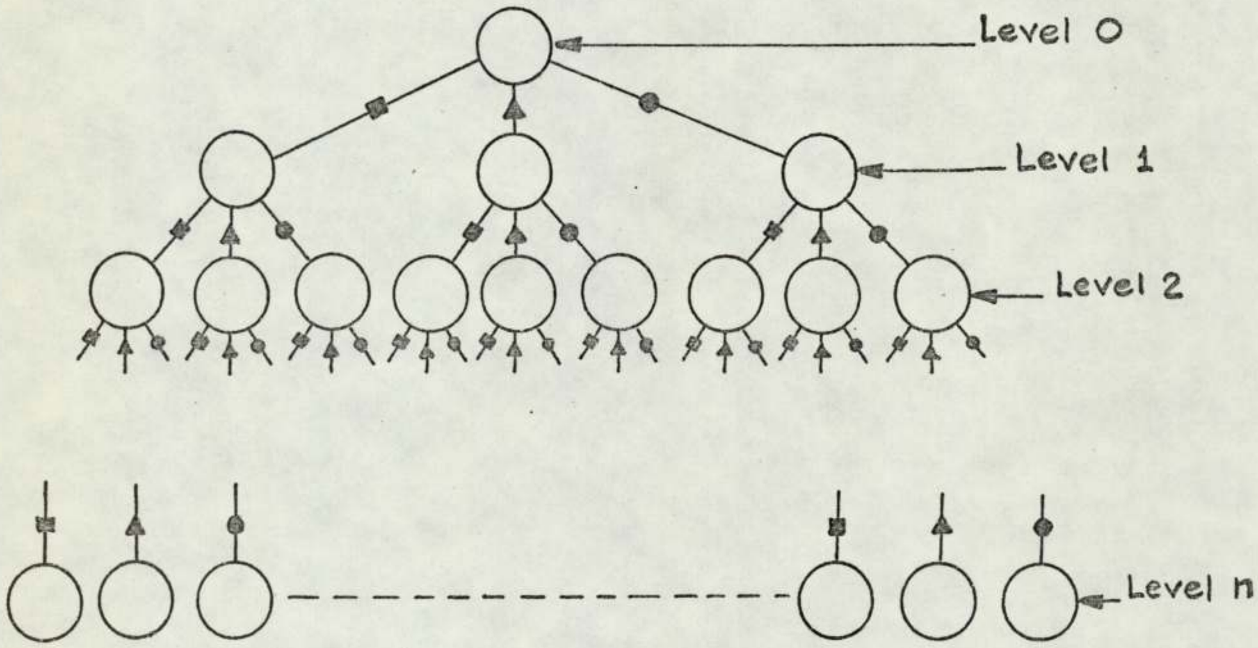


Fig A5 A Balanced Tree.

A.5.3 The rank of an S-matrix

The rank of an H-matrix is at most 2 and of a P-matrix at most 1. In the special case $\overline{r} = \overline{0}$ (i.e. S-loc and R-point coincident), $\overline{p}$, $\overline{h} \neq \overline{0}$ then both $\overline{P}$ and $\overline{H}$ are of rank 0 since $\overline{P} = \overline{H} = \overline{0}$.

From equation (xii)

$$S_{ij} = -(P_{ij} + H_{ij} + \delta_{ij}) = -\delta_{ij} \text{ if } \overline{P} \text{ and } \overline{H} \text{ are of null rank.}$$

S_{ij} is of rank 3 since all row vectors of δ_{ij} are linearly independent $(\det(\delta_{ij}) = 1)$.

An S-matrix is of rank at most 3.

A.5.4 The rank of a matrix product

A result proved in ref. 14 is:

'the product AB has a rank not greater than the rank of either factor.' The rule may be applied to path matrix products to give the following conclusions:

(a) if a path contains the node of a P-loc, then the path matrix product is of rank at most 1,

(b) if a path contains the node of an H-loc, then the path matrix product is of rank at most 2, and

(c) if a path contains the node of an S-loc, then the path matrix product is of rank at most 3.

The rank of the path matrix product is determined by the most stringent of conditions (a), (b) and (c) which can be applied to the path.

A.5.5 Relative Numbers of P-, H- and S-matrices

It will assist in assessing algorithms used in processing 'trees' if the relative proportions of matrix ranks are known. For the fully balanced ternary tree shown in Fig. A5 the following results apply:

(a) At level n, there are 3^n nodes.

(b) If, at level n-1 there are N_{n-1} paths with matrix products of rank 1 at most, then there will be $3^{n-1} + 2N_{n-1}$ paths with similar rank at level n.

$N_n = 3^{n-1} + 2N_{n-1}$.

Solving this difference equation with $N_1 = 1$ gives

$N_n = 3^n - 2^n$ paths with product matrices of at least 1 at level n. Summing over n levels gives:

$T_n = \frac{(3^{n+1} - 3)}{2} - (2^{n+1} - 2)$ – the total number of paths with product matrix rank at least 1 down to level n.

(c) At each level there will be one path with path matrix product of rank at least 3. Over n levels there will be a total of n such paths.

(d) The total number of paths down to level n is $\frac{(3^{n+1} - 3)}{2}$

(e) The total number of paths with product matrices of rank at least 2 down to level n may be found by subtracting the number of paths with product matrices of ranks 1 and 3 at most from the total number of paths down to level n. This is $2^{n+1} - 2 - n$.

(f) For n levels, there will be $\frac{3^{n+1} - 1}{2}$ nodes.

Tabulating values up to 6 levels:

	Path Product Matrix Maximum Rank			No. of Nodes
Level	1	2	3	
1	1	1	1	4
2	6	4	2	13
3	25	11	3	40
4	90	26	4	121
5	301	57	5	364
6	966	120	6	1093

For large n, rank 1 matrices will predominate and the ratio of rank 1 matrices to rank 2 matrices $= O(\frac{1}{2}\,(\frac{3}{2})^{n+1})$. For the system designed, about 200 nodes is considered as a maximum. The number of levels of the corresponding balanced ternary tree will lie between 4 and 5 giving a ratio of approximately 4.5 rank 1-at-most matrices to 1 rank 2-at-most matrices. The number of rank 3-at-most matrices will be negligible.

Although a practical location 'tree' is unlikely to be a balanced tree, it seems reasonable to assume that these ratios will be approximately correct and will serve as a useful measure in the assessment of algorithms used for processing the 'tree'.

A.6 The maximum output displacement (sensitivity coefficient)

In the discussion which follows, the matrix results listed below will be used. They are proved in refs. G2 and G8.

(i) If A* denotes the transpose of matrix A, then the product A*A is positive definite, ref. G8, p.46.

(ii) If A is positive definite, then all its eigenvalues are positive, ref. G8, p.46.

(iii) If the eigenvalues of A*A are λ_i, then the eigenvalues of $(A^{-1})^*A^{-1}$ are $(^1/\lambda_i)$, the associated eigenvectors being identical, ref. G8, p.43.

Generally, the output tolerance $\bar{\delta}$ resulting from an input displacement $\bar{\epsilon}$ applied to a path with matrix F is given by:

$$\bar{\delta} = F\,\bar{\epsilon}\,.$$

A general spherical input tolerance zone of radius r may be written

$$\bar{\epsilon}^*\,\bar{\epsilon} = r^2.$$

In particular, if F be of rank 3, then it will have an inverse, say G, and

$$G\bar{\delta} = \bar{\epsilon}$$

or $$\bar{\epsilon}^* = \bar{\delta}^*G^*$$

and $$\bar{\epsilon}^*\,\bar{\epsilon} = \bar{\delta}^*G^*G\,\bar{\delta}$$

or $$\bar{\delta}^*G^*G\bar{\delta} = r^2$$

The product G*G is symmetric and positive definite by result (i), and so may be diagonalised giving the relation:

$$\bar{\delta}^*\,L\,\bar{\delta} = r^2,$$

L being the diagonal matrix $L_{ij} = \begin{cases} \lambda_i, i = j \\ 0, i \neq j \end{cases}$.

The λ_i are the eigenvalues of L.

Since G*G is positive definite, $\lambda_i > 0$ and the relation describes an ellipsoid:

$$\lambda_1 \delta_1^2 + \lambda_2 \delta_2^2 + \lambda_3 \delta_3^2 = r^2.$$

The maximum axis of the ellipsoid will be given by:

$$\delta_{\max} = \frac{r}{\sqrt{\lambda}}$$

where $\lambda = \min(\lambda_i)$ in a direction given by the associated eigenvector of λ. This will be the maximum output displacement.

The matrix result (iii) simplifies the calculation considerably. If the eigenvalues of G*G are λ_i, then the eigenvalues of F*F are $\frac{1}{\lambda_i}$. Also the maximum eigenvalue of F*F is the minimum eigenvalue of G*G. The corresponding eigenvectors are identical.

The procedure for finding the maximum output displacement, and its direction, may be summarised as follows.

If a spherical input tolerance zone radius r be applied at the base of a location chain with path matrix F (F non-singular), then the maximum output displacement is given by

$$\delta_{\max} = \sqrt{\lambda}\, r,$$

where λ is the dominant eigenvalue of the product F*F. The direction of $\delta_{\max}$ will be given by the corresponding eigenvector. For a more detailed development, see ref. G16.

The result has been derived for a rank 3 matrix but it can be shown to be generally true for ranks 2 and 1. The argument is broadly the same, but since ranks 2 and 1 3 x 3 matrices are singular, they have no inverse and it is necessary to consider the natural or general inverse. Ref. G2 contains a concise description of the use of general inverses, while ref. G3 is completely devoted to them.

APPENDIX B

NOTES ON ALGORITHMS

B.1 Evaluation of Eigenvalues

B.1.1 Choice of algorithm

The methods available are:

(i) to obtain a closed solution by expanding the characteristic polynomial which will, in the most general case, be a cubic;

(ii) to use an iterative method, such as the power method, or

(iii) to use a transformation method, e.g. Householder's method.

The factors governing the choice of the method are:

(i) all matrices are of order 3,

(ii) the bulk of the matrices involved will be of rank 1,

(iii) the dominant eigenvector is also required,

(iv) the number of matrices is likely to be large,

(v) accuracy of solution is not extremely critical - accuracy of 1 in 10^5 should suffice, and

(vi) matrices are positive definite and symmetric.

The power method was chosen since

(i) in this case, it may be used generally for all ranks,

(ii) the dominant eigenvalue is obtained naturally and

(iii) the corresponding eigenvector is obtained at the same time.

B.1.2 Description of the algorithm

The power method is detailed in ref. G8 and an error analysis provided in ref. G9, but the method will be briefly described.

If the dominant eigenvalue of matrix A is required, then the computing scheme is

$$y^{(P)} = Ay^{(P-1)} \; ; \; y^{(0)} \text{ arbitrary.}$$

The $y^{(N)}$ are successive iterates and are vectors. It is customary to select as initial vector $y^{(0)} = \{1,1,1\}$. The ratios of corresponding components of successive vectors $y^{(K)}$ and $y^{(K-1)}$ will converge to λ, the dominant eigenvalue of A, if the method is successful. Further, each iterate $y^{(K)}$ is an estimate of the corresponding unnormalised eigenvector. The rate of convergence depends on the ratio between the dominant and sub-dominant eigenvalues. Methods are available for accelerating convergence.

Since the majority of the matrices will be of rank 1, it would seem advantageous to select as initial vector $y^{(0)} = \{A_{11}, A_{21}, A_{31}\}$ or any other column vector of A. The dominant eigenvalue would then be obtained in one iteration only, since the corresponding eigenvector (unnormalised) is a column vector of A. Unfortunately, it is common for one or more of the column vectors in location matrices to consist of all zero elements. This is a particular case of a general problem in selecting the initial vector for use with the power method. If the initial vector is exactly the eigenvector corresponding to an infra-dominant eigenvalue, then the method will yield that eigenvalue in one iteration. Clearly, an answer obtained in one iteration should be viewed with suspicion and the calculation repeated with the original initial vector slightly perturbed. This problem is not mentioned in most of the standard texts (ref. G8 is an exception) nor is it considered in any of the programs described in the less theoretical books on numerical methods. Unfortunately, the case of a location matrix having an infra-dominant eigenvector $\{1,1,1\}$ is not uncommon in practice. In view of the fact that the bulk of the matrices processed will be of rank 1 and single iteration answers will be common, it is considered that repeating the calculation with a perturbed initial vector would be an intolerable overhead of time. The initial vector $\{\pi/4, e, \log_e 10\}$ is used in the program. Even though there is a remote possibility that these values will give an incorrect answer, it is considered worthwhile to use them because of the saving in time.

Since the successive $y^{(K)}$ are unnormalised, their components tend to increase rapidly and it is necessary to normalise at each stage. This is done by dividing each element of $y^{(K)}$ by $\|y^{(K)}\|_\infty$, where $\|y^{(K)}\|_\infty$ is the maximum value of $|y_i^{(K)}|$ over all i.

A test program was written to check the efficiency of various programs for calculating eigenvalues. A random number generator was used in conjunction with a method of generating matrices of prescribed eigenvalues and eigenvectors which was found in ref. G10.

B.1.3 Generating test data

Batches of 40 matrices were generated in the following manner:

(i) The constitution of each batch was

1 rank 3 matrix,
7 rank 2 matrices,
32 rank 1 matrices.

These proportions were approximately those calculated in Appendix A, section A.5.5 for 200 nodes.

(ii) A matrix S was constructed with column vectors mutually orthogonal, but otherwise random. If p_i denote the i-th random number generated, then

$$S_{i1} = \left\{p_1 , p_2 , p_3 \right\}$$

$$S_{i2} = \left\{p_3p_4, p_3p_5, -(p_1p_4 + p_2p_5)\right\}$$

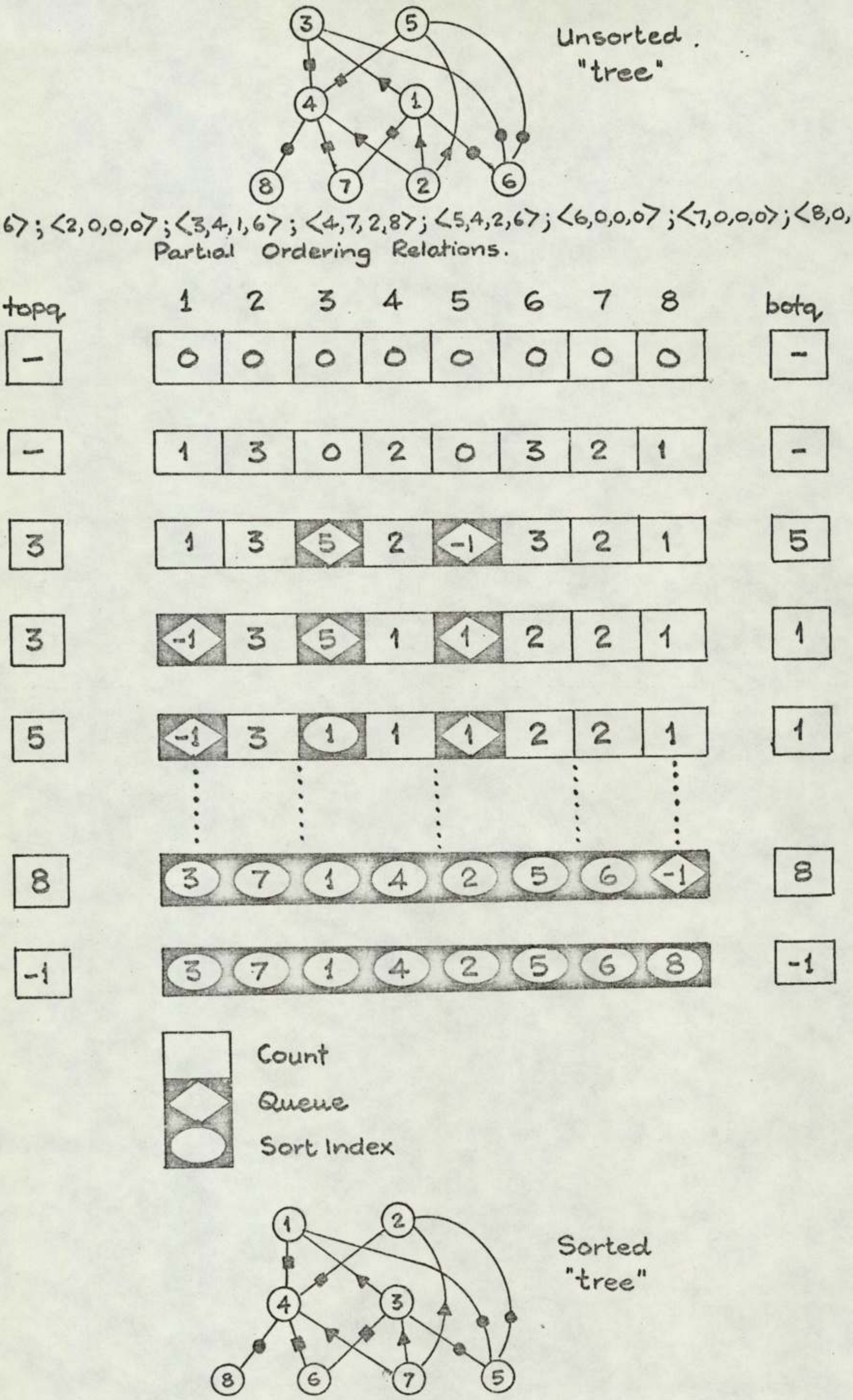
$$S_{i3} = \left\{(p_1p_2p_4 + p_2^2p_5 + p_3^2p_5), - (p_3^2p_4 + p_1^2p_4 + p_1p_2p_5), - (p_1p_3p_5 - p_2p_3p_5)\right\}$$

(iii) Since column vectors of 3 are linearly independent, S^{-1} exists and is calculated directly.

(iv) The diagonal matrix A is constructed where the diagonal elements are the eigenvalues.

$$A_{ij} = \begin{cases} p_5 + i & \text{for } i = 1 \text{ to the rank of A, } i = j \\ 0 & \text{otherwise} \end{cases}$$

(v) The matrix product $S^{-1}A\ S$ will have the required properties (ref. G2, p.99).



<1,7,2,6> ; <2,0,0,0> ; <3,4,1,6> ; <4,7,2,8> ; <5,4,2,6> ; <6,0,0,0> ; <7,0,0,0> ; <8,0,0,0>

Partial Ordering Relations.

topq	1	2	3	4	5	6	7	8	botq
–	0	0	0	0	0	0	0	0	–
–	1	3	0	2	0	3	2	1	–
3	1	3	◇5	2	◇-1	3	2	1	5
3	◇-1	3	◇5	1	◇1	2	2	1	1
5	◇-1	3	○1	1	◇1	2	2	1	1
8	○3	○7	○1	○4	○2	○5	○6	◇-1	8
-1	○3	○7	○1	○4	○2	○5	○6	○8	-1

Fig B1 Topological Sort Algorithm.

This rather elaborate procedure was used so that the batches of matrices should be as representative as possible of those occurring in practice.

A separate program was used for the generation of batches of matrices, so that the time taken in their generation could be separated from the time taken for calculating their dominant eigenvalues and eigenvectors.

B.2 Topological Sort Algorithm

The algorithm used was a modification of that described in detail in ref. G13. Its action is shown diagrammatically in Fig. B1.

Data used is selected from the 'tree' description supplied as input. If the 'tree' contains N nodes, each allotted a distinct integer in the range 1-N, and if M of these are non-leaf nodes, then there will be M partial ordering relations of the form $\langle R,P,H,S\rangle$, which are needed by the algorithm. Each integer R represents an R-point; and P, H and S are the node numbers of the corresponding P-, H- and S-locs. A further N-M relations of the form $\langle R,0,0,0\rangle$ are also available - the R, in this case, representing a leaf-node and 0 being a notional earth-node.

A vector, length N is used in three guises. It is used initially to count the direct predecessors of each node; it is used as a queue for unprocessed nodes and, finally, it is used as an index to show topological sort order. Two pointers are used to point to the head and tail of the queue of unprocessed nodes. The sort process is as follows:

(i) Zeroise count [i] for i = 1 to n.

(ii) Count the predecessors of each node.

(iii) All nodes with zero count are root nodes. Set up a queue for these; if there are no root nodes, then the structure is incorrect.

(iv) Select the first item on the queue. If its three successor nodes are not earth nodes, then their counts are reduced by one. If any count becomes zero, then the corresponding node is ready for processing and its index number is queued.

(v) The node at the head of the queue is deleted and it is next in topological order. It is allocated the next index number.

(vi) If the queue is empty, and all nodes have been processed, then the sort has been successfully accomplished. If the queue is empty and all nodes have not been processed, then the structure is incorrect. If the queue is not empty, then the algorithm is continued from stage (iv).

It is claimed in ref. G13 that the algorithm is near-optimal. Processing time is of the order of $C_1 M + C_2 N$ where C_1 and C_2 are constants, and storage is used economically.

It is possible to re-order the records during the algorithm but this was not done for various reasons. Firstly, it is considered good practice to divorce the data validation stage of a program from the data processing stage both as a policy and because, for some configurations, it might be necessary to perform these operations by separate programs. Also, subsequent programming is neater and more easily tested if a structured approach is used.

B.3 Inverting the Topological Sort Index

It is possible to refer to each record indirectly using the sort index, but it saves much processing time if the records are re-sorted. Re-sorting may be performed in several ways; the familiar dilemma of time taken versus extra storage required applies in this case, as in all sorting problems. The following is a sample of the methods possible:

Given a sequence of records $R_1, R_2, \ldots R_N$ and a sort index $T_1, T_2, \ldots T_N$ where T_K shows the required sort position of record R_K,

(a) Perform an exchange sort, repeatedly passing through the list R, and exchanging $R(T_K)$ until no exchanges are necessary,

(b) Invert the sort-index T by $I(T_K) = K$ to form another sort-index $I_1, I_2, \ldots I_N$. I_K shows the number of the record which is to be placed into position K. Records can now be exchange-sorted in one pass by:

temp 1: = $R(I_K)$; $R(I_K)$: = R_K ;
temp 2: = $R(T(I_K))$; $R(T(I_K))$: = temp 1
$R(I_K)$: = temp 2.

(c) Invertthe sort-index in situ, saving setting up an extra sort-index vector. The algorithm used may be found in refs. G1 and G13. In particular, ref. G13 quotes two algorithms for this purpose, but one, though more elegant, may be discounted since it is less efficient. Records may then be exchange-sorted in one pass as in (b).

It was decided in the interests of storage economy, to select method (c). The method is analysed in ref. G13 and the processing time is of the order of C N where C is a constant. Exchange-sorting is performed in one pass and so it appears that method (c) is better on a processing time basis than (a) since normal exchange-sorting time is proportional to N^2, and better from storage economy considerations than method (b) since no inverted sort-index is required.

B.4 Processing the 'tree'

From tests conducted on the prototype program, this section of the processing is easily the most lengthy. Not only must all the paths from input location to R-point be traversed but also matrices for each link on the paths must be calculated. Although it is a straightforward matter to minimise the matrix calculation time by writing efficient code, path traversal is a difficult problem. For N nodes Warshall's algorithm (ref. G1) requires an N x N matrix and is out of the question because of the storage limitations. Alternatively, linked lists may be held for the immediate predecessors of each node - this also needs an unacceptable overhead of store. This problem is largely overcome by an elegant algorithm quoted in ref. G15 but this is not general and depends on the relative number of leaf nodes and nodes with multiple antecedents. A possible method is to use a marking algorithm, tagging in some way all edges on the paths between input location and R-point using a stack - this again needs extra store.

The method used in the prototype program was crude but straightforward, priority being given to economy of store. All matrices were evaluated and multiplied, and were added at junctions for all nodes whose indices lay between those of the input node and the R-point.

The problem of traversing paths of structures of this kind occurs in many diverse applications and it seems that the algorithms available involve considerable storage overhead. For the configuration considered, it is unavoidable that this brute force method should be used in preference to one more sophisticated but requiring more store.

APPENDIX C

ALLOCATION OF TOLERANCES

C.1 The Allocation of Tolerances - sure-fit

If there are m critical clearances D_i $(i = 1, 2, \ldots m)$ in an assemblage and each is affected by one or more of the set of tolerances x_j $(j = 1, 2, \ldots n)$, then m linear inequalities may be written:

$$S_{ij} x_j \leqslant D_i \qquad i = 1, 2, \ldots m; \quad j = 1, 2, \ldots n.$$

The constants S_{ij} are sensitivity coefficients, S_{1k}, for instance being the effect at the critical clearance D_1 of a unit tolerance at the point where x_k acts. In the case where a critical clearance D_1 does not depend on tolerance x_k, then $S_{1k} = 0$.

Subject to these constraints, it is required that the total cost of maintaining the critical clearance should be minimised. Clearly, the more precise a part is made, the higher will be the unit cost but there seems to be some disagreement in the references as to the exact form of the cost-tolerance relationship. Models suggested are -

(i) $\text{cost} = kx^{-2}$ where k is a constant (ref. A2)

(ii) $\text{cost} = kx^{a}$ where k and a are constant and $a < 0$ (ref. A1)

(iii) $\text{cost} = k + le^{mx}$ where k, l and m are constant (ref. A8)

Although (iii) is the most widely used model and is used in several American papers, where it is called Speckhart's Exponential Model, (ii) appears to have been based on rather more solid experimental foundations. Studies of data on the cost-tolerance relationship for various manufacturing processes were analysed and best curves fitted by the method of least squares. No experimental basis is described for model (iii), the author baldly stating that the expression fits cost-tolerance data 'very well'. Model (i) is comprehensively (if rather unfairly) discredited in ref. A1. The evidence would suggest that the most suitable model is (ii) and this is used in the following development.

The cost of maintaining tolerance x is given by the expression:

$$C = kx^{a}$$

$- 0.8 < a < - 0.4$ and depends on the process. k depends on the shape and size of the component. Values of k are not critical, only the relative values being of significance.

The cost of maintaining the n tolerances x_i is given by:

$$C = \sum_{i=i}^{n} k_i x_i^{a_i}$$

The Allocation of Tolerances problem may now be stated in its full form:

$$\text{Minimise } C = k_1 x_1^{a_1} + k_2 x_2^{a_2} + \ldots + k_n x_n^{a_n}$$

Subject to constraints:

$$\begin{array}{ccccccc} S_{11}x_1 & + & S_{12}x_2 & + \ldots + & S_{in}x_n & \leqslant & D_1 \\ \vdots & & \vdots & & \vdots & & \vdots \\ S_{m1}x_1 & + & S_{m2}x_2 & + \ldots + & S_{mn}x_n & \leqslant & D_m \end{array}$$

There are also implicit constraints

$$x_1, x_2, \ldots x_n > 0$$

and there may also be constraints due to practical considerations:

$$x_i \geqslant e_i$$

where e_i is the lowest practicable bound on x_i.

This is an optimisation problem with a non-linear objective function and linear constraints.

C.2 The Allocation of Tolerances - statistical-fit

The development of equations for allocating tolerances on a statistical-fit basis follows broadly the lines of that for sure-fit basis. In this case, however, it is usual to assume that the tolerance distributions follow a Gaussian distribution and often the tolerance range is taken as the nominal position plus or minus three standard deviations, 99.7% of the parts produced then having the dimension within the allowed tolerance range. The cost is that of maintaining the dimension within plus or minus three standard deviations around the nominal dimension. Using the properties of the Gaussian distribution, it may be established that the Allocation of Tolerances problem may be stated:

Minimise $C = K_1 x_1^{a_1} + k_2 x_2^{a_2} + \dots k_n x_n^{a_n}$

Subject to constraints:

$$\begin{array}{lllll} S_{11}^2 x_1^2 + & S_{12}^2 x_2^2 & + \dots + & S_{1n}^2 x_n^2 \leqslant & D_1^2 \\ \vdots & \vdots & & \vdots & \vdots \\ S_{m1}^2 x_1^2 + & S_{m2}^2 x_2^2 & + \dots + & S_{mn}^2 x_n^2 \leqslant & D_m^2 \end{array}$$

With implicit constraints:

$x_1, x_2, \dots x_n > 0$

and possibly practical minima on tolerances:

$x_i \geqslant e_i$

All constants and variables are as defined in C1.

This is an optimisation problem with a non-linear objective function and non-linear constraints.

C.3 Solution of the Allocation Problem

Refs. A1 and A8 both use the classical technique of Lagrange multipliers in order to solve the Allocation Problem, obtaining what both call lambda equations, which are solved by an iterative technique such as Newton's method. This is a straightforward technique for one critical dimension, but in the more general case, where more than one critical dimension is concerned, the method discussed in ref. A1 requires considerable manual work before submission to the computer program and is only applicable to the sure-fit case. Ref. A8 uses an iterative procedure but it is stated that there is no guarantee that the procedure described will converge.

Several general methods of non-linear optimisation are described in ref. G4 and it would seem that the methods described in refs. A1 and A8 have been superseded by later techniques. It is probable that these are more suitable methods and this particular aspect of tolerance analysis merits investigation. Possibly different methods would be required for the sure-fit and statistical-fit cases, since it is stated in ref. G4 that a universal optimizer does not exist and a method suitable for linear constraints (sure-fit) may not be adequate for quadratic constraints (statistical-fit). This is, however, outside the scope of this thesis.

APPENDIX D

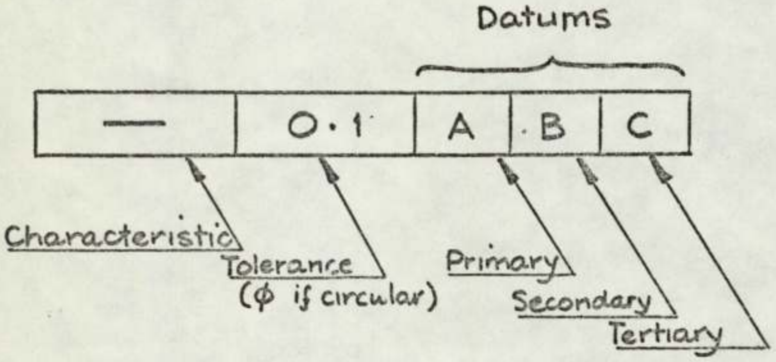
STANDARD CASES

BS 308. PT 3

Symbols used in tolerance frames

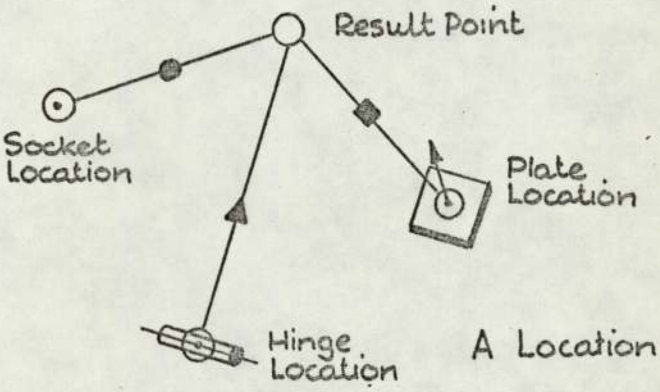
CHARACTERISTICS

—	Straightness
//	Parallelism
⊥	Squareness
∠	Angularity
⌖	Position
◎	Concentricity
⌯	Symmetry
▱	Flatness
○	Roundness
⌭	Cylindricity
⌒	Profile of a line
⌓	Profile of a surface
↗	Run-out.

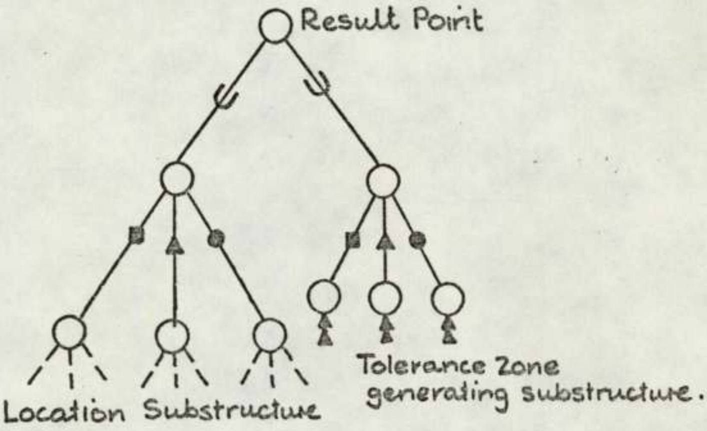


TOLERANCE FRAME

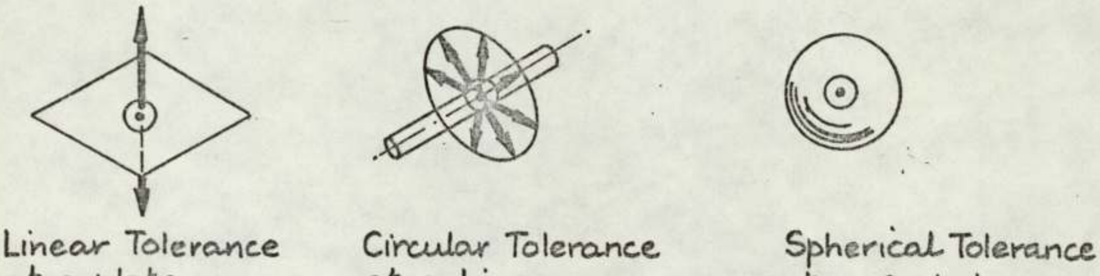
A ─◀	Datum.
Ⓜ	Maximum Metal Condition.
30	True Position.



A Location Triad



General Displacement at a Point



Linear Tolerance at a plate

Circular Tolerance at a hinge

Spherical Tolerance at a socket.

Fig D1 Displacement at a Result Point.

D.1 The Displacement at a Point - Fig. D.1.

A point must be located on a location triad of plate, hinge and socket.

The displacement at a point is made up of two components:

(a) the extrinsic displacement which is due to displacements of the features on which it is located, and

(b) the intrinsic displacement which is due to the permitted tolerance at the point.

Extrinsic tolerance is passed to the point by the location sub-system upon which it depends; intrinsic tolerance must be applied directly at the point or indirectly through a tolerance generating sub-structure.

(i) If the result point is on a plane, then a linear tolerance normal to the plane may be applied directly.

(ii) If the result point lies on a line, then a circular tolerance in a plane perpendicular to the line may be applied directly.

(iii) If the result point is a general point, then a spherical tolerance may be applied directly.

The majority of tolerance situations will be covered by (i), (ii) and (iii) since these are the positional tolerances recommended in BS 308, but occasionally a rectangular or parallelepipedal tolerance zone is quoted; and it is sometimes necessary to generate this by using a tolerance generating sub-structure. The method will be described later.

It is necessary to use bi-lateral tolerances when this method is applied. For most systems, the tolerance is small in comparison with nominal dimensions and so the nominal dimension does not need to be altered. For instance, a tolerance of $1.000 \begin{matrix} + .010 \\ - .000 \end{matrix}$ may be considered as $1.000 \begin{matrix} + .005 \\ - .005 \end{matrix}$. The reason for this is that only clearances are

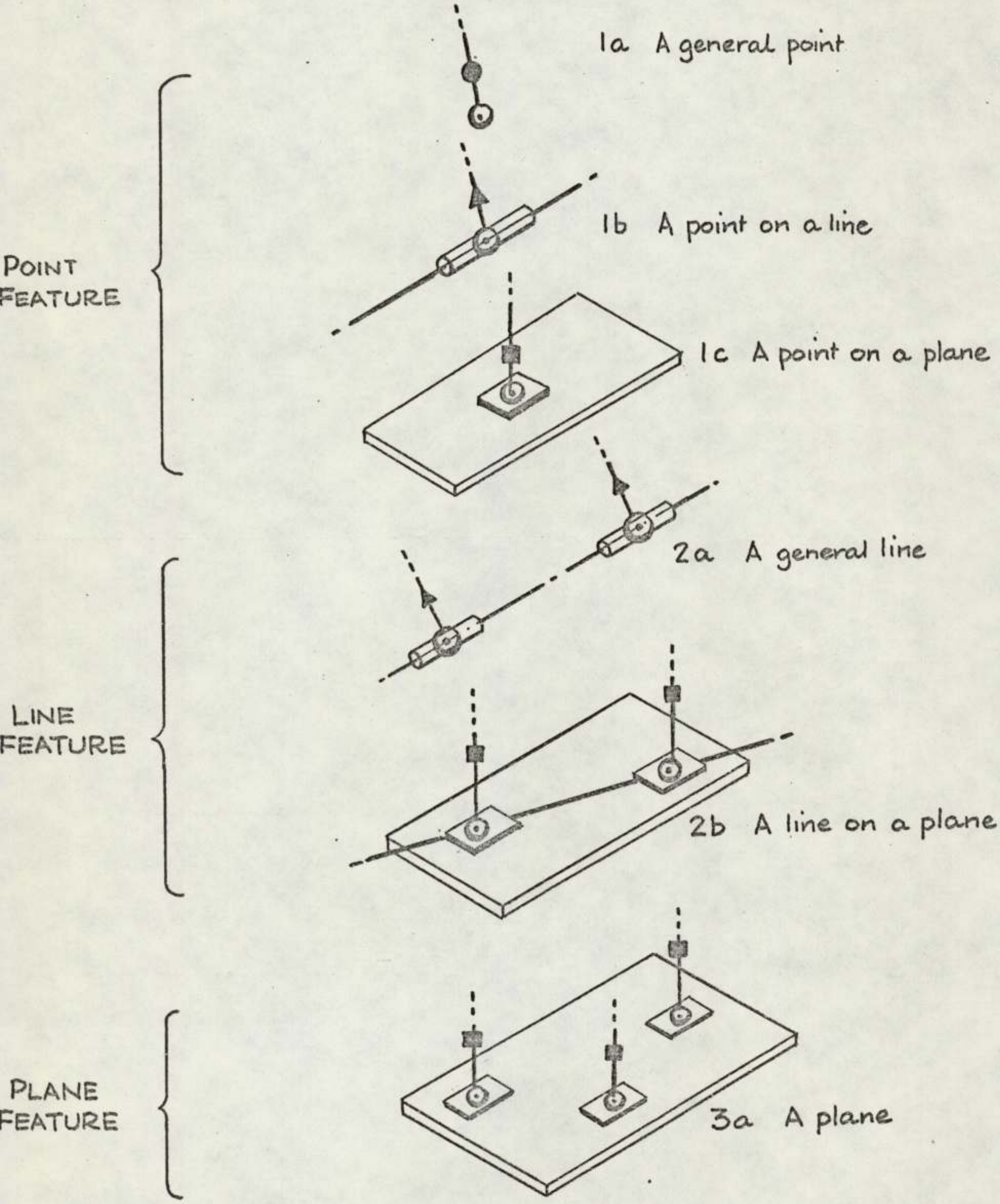


Fig D2 Definition of Features.

analysed in this method. Nominal dimensions are only used in intermediate calculations and so do not necessarily need to be extremely accurate. Of course, if it is preferred the dimension can be quoted as 1.005 ± .005, but this should make very little difference to the results obtained.

D.2 Definition of Features - Fig. D.2.

1. A point feature.
   (a) A general point is described by a socket.
   (b) A point on a line is described by a hinge.
   (c) A point on a plane is described by a plate.
2. A line feature.
   (a) A general line is described by two hinges.
   (b) A line on a plane is described by two plates.
3. A plane feature.
   (a) A plane is described by three plates.

In order to define a feature, the points at which the locations are centred may be chosen arbitrarily subject only to the following restrictions:

(a) in order to define a line, the points must not be coincident, and
(b) in order to define a plane, the points must not be collinear.

Plates defining a plane must have normals parallel with the normal to the plane, and hinges defining a line must have directions along the line.

D.3 General Points on Lines and Planes

A general point on a line or a plane may be considered as being located on the line or plane. Since a point is located on a plate, hinge and socket in the basic location triad, the general point cannot be located directly on the line or plane, which are defined by two hinges or plates, or three plates respectively. This problem is resolved

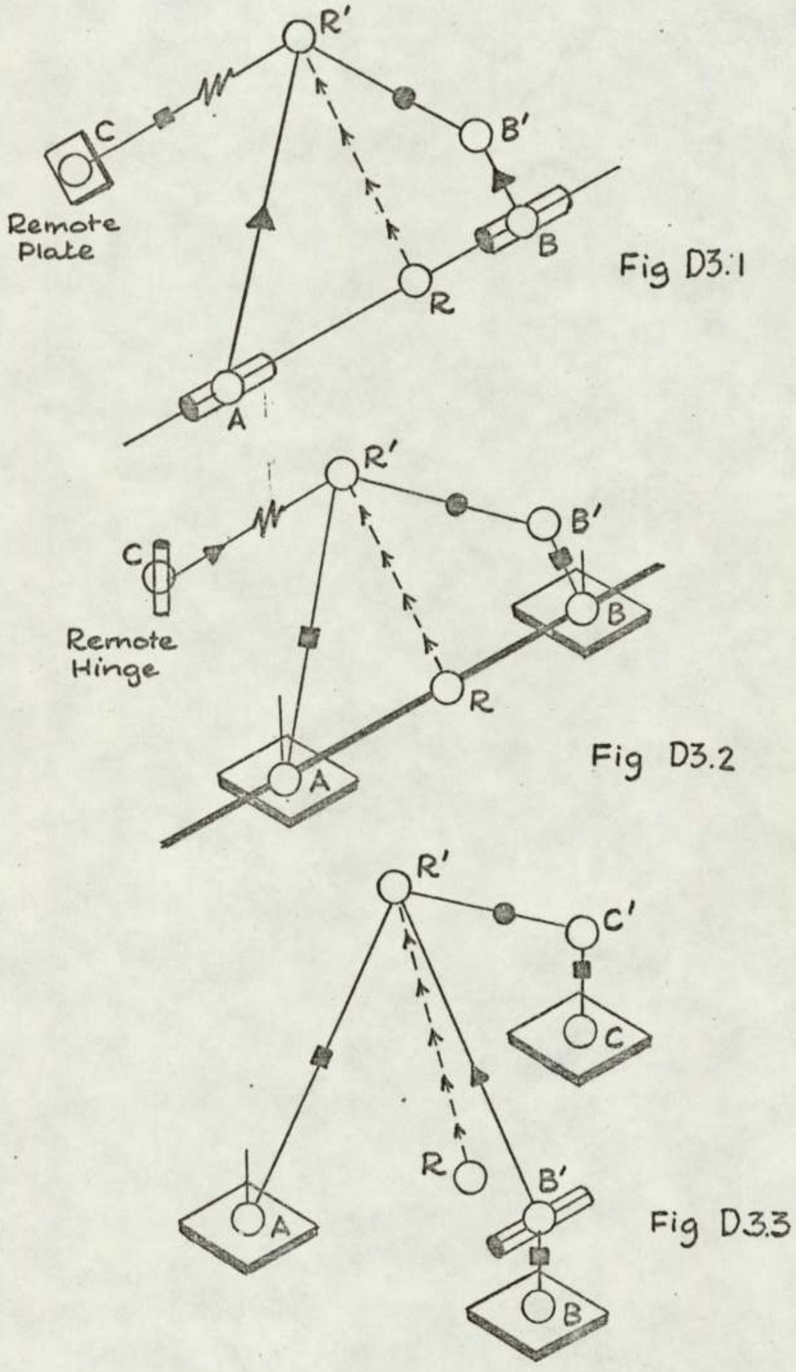


Fig D3.1

Fig D3.2

Fig D3.3

Fig D3 General Points on Lines & Planes.

as follows:

(a) A general point on a line.

(i) A general line - see Fig. D.3.1.

The general point R is on the line A-B. The displacement at B must be passed unchanged to the coincident point B' so that a valid socket location can exist at B'. R is also located on plate C which is stationed at a remote point with normal parallel with A-B.

(ii) A line on a plane - see Fig. D.3.2.

The general point R is on the line A-B. Again the displacement at B must be passed unchanged to the coincident point B' so that a valid socket can exist at B'. R is also located at hinge C, stationed at a remote point with direction parallel with the normals to A and B.

(b) A general point on a plane - see Fig. D.3.3.

The general point R is on the plane A-B-C. Displacements at B and C are passed unchanged to B' and C' for valid locations.

D.4 Remote Locations

Many useful location systems can be devised using remote locations. In cases studied so far, the following dimensions give adequate accuracy:

(i) A small displacement - of the order of 10^{-2}.

(ii) A neighbouring feature - one within a radius of about 10^2 centred at the point being considered.

(iii) A remote feature - one further than 10^6 from the point being considered.

(a) To pass a displacement unchanged from a feature to a neighbouring point.

(i) From a plate - Fig. D.4.1.

The displacement at plate A is passed unchanged to point A' (which may be coincident with A). A' can be any neighbouring point along the normal to plate A. The angle BAC is a right angle; and the direction

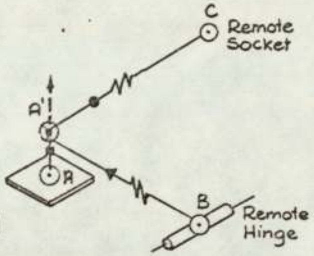


Fig D4.1

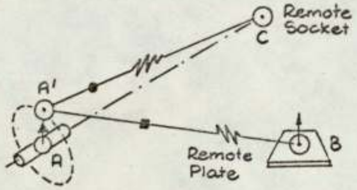


Fig D4.2

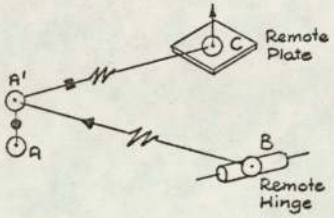
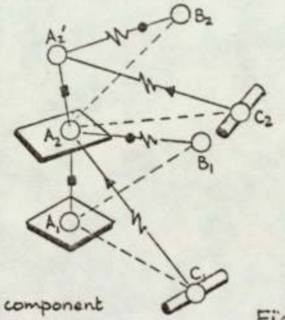


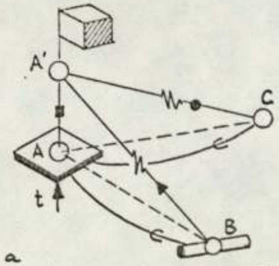
Fig D4.3

Fig D4. Use of Remote Features for Transfer



Selecting a component
of a tolerance.

Fig D5.1



Generating a
cubic tolerance zone

Fig D5.2

Fig D5 Use of Remote Features - Components.

of hinge B is parallel with AC. Both B and C are remote from A.

(ii) From a hinge - Fig. D.4.2.

The displacement at hinge A is passed unchanged to point A' (which may be coincident with A). A' can be any neighbouring point whose position vector $\overline{AA'}$ is at right angles to the hinge direction. The remote socket C is approximately in line with the hinge direction and remote plate B is arbitrarily positioned and oriented.

(iii) From a socket - Fig. D.4.3.

The displacement at point A is passed unchanged to A' (which must not be exactly coincident with A, and is displaced a small amount). There is no restriction on the position of B nor on the position of C.

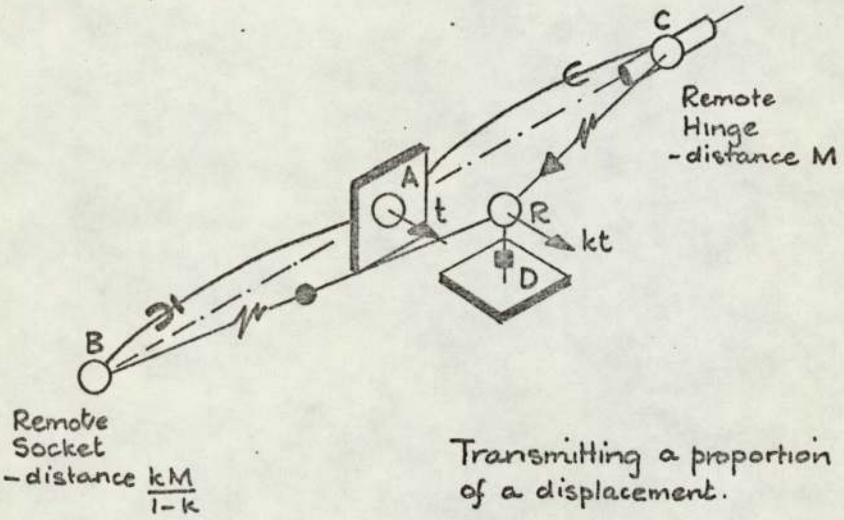
(b) To transfer a selected component of a displacement to a point - Fig. D.5.1.

The normal to plate A_2 is in the direction of the selected component. B_2 and C_2 are in the plane of A_2, the direction of C_2 is parallel with A_2B_2, and A_2' is coincident with A_2. The selected component of the displacement at A_1 is transmitted to A_2.

(c) To rotate a linear displacement through 90° - Fig. D.5.2.

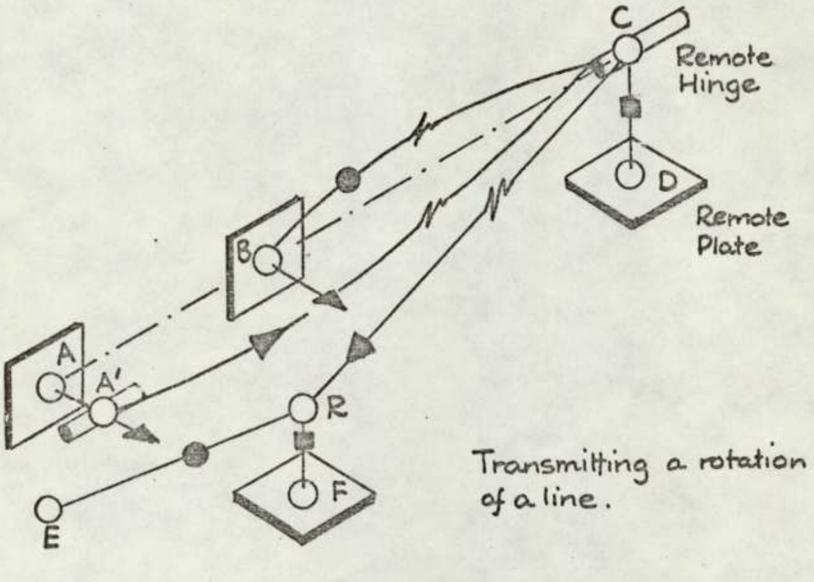
(Useful for generating a square or cubic tolerance zone).

Tolerance t is passed, using a unitary matrix, unchanged to remote hinge and socket B and C. The methods of (a) could be used to transmit 6, but use of a unitary matrix saves nodes, and avoids the need for a higher order of 'remoteness'. If B and C are orthogonal features, then a cubic tolerance, side t will be induced at A', which may be coincident with A. Adjustment of the direction of B and the positions of B and C will result in a parallelepipedal tolerance zone. If the locating features are stationed at the same large distances from the R-point with the direction of the H-loc and the normal to the P-loc lying along the lines joining them to the R-point, a unit parallelepipedal tolerance



Transmitting a proportion of a displacement.

Fig D6.1



Transmitting a rotation of a line.

Fig D6.2

Fig D6 Use of Remote Features _ Transmission zone will be generated. This will have sides normal to the directions of the lines joining P-, H- and S-locs to the R-point. This device is particularly useful and will be used for other purposes.

(d) To transmit a proportion of a displacement to a neighbouring point - Fig. D.6.1.

(Useful in generating symmetric tolerances).

It is required to pass a proportion k of the tolerance at plate A to the neighbouring point R. Displacement at A is passed unchanged to remote hinge C stationed at distance M in the plane of the plate. The neighbouring point R is located on C, and on B, a remote socket placed at a distance $\frac{kM}{1-k}$ on the line AC, but on the opposite side of A to the position of C. R is also located on plate D which has normal orthogonal to the plane RAC and passing through R. It may for convenience be coincident with R. In the case of symmetric tolerances, k must be $\frac{1}{2}$, and so B and C are equidistant from A. It is important that A, B and C should not be exactly collinear but only approximately so.

(e) To transmit the rotation of a line to a point which rotates about a neighbouring point - Fig. D.6.2.

Due to displacements at A and B, the line joining them will rotate in space. It is required that neighbouring result point R should rotate the same angle, in the same plane about location point E. Displacement at A is passed unchanged to A' a hinge with direction parallel with AB. To be consistent, the displacement at B should be passed to a coincident socket, but a node may be saved by locating directly at B. Displacements at A and B are passed to the remote hinge C by way of hinge and socket locations. C is stationed approximately in line with AB and located on D, a coincident remote plate whose normal is orthogonal to normals at A and B. The result point R is located on the socket E, the hinge C and coincident plate F. A, B, E and R are co-planar, and the normal of plate F is orthogonal to this plane.

Any displacement at E is passed unchanged to R.

(f) Use of unitary links to pass a displacement unchanged from a feature to a point.

The unitary link is an artificial device used for the generation of geometric tolerance networks which cannot be done by using real locations. It may be used to obtain the results shown in (a) with a saving of nodes. In most of the networks which follow, the preferred method is to use the devices shown in (a), since this leads to a more natural system. However, occasionally, unitary links have been included so as to give examples of their use. A separate section is devoted to this application. Unitary links are used for the superposition of separate tolerance systems.

(g) Use of weak links.

A result point is located on plate, hinge and socket, and displacement at the locations may be considered to be transmitted along the link, usually. Occasionally, however, it is convenient, in the interests of node conservation, to use a device called the weak link. A weak link points to a location node which is only used for the calculation of displacements transmitted from other nodes. Displacement occurring at the weakly linked node is not transmitted to the result node. This device, like the unitary link, is not essential to the system but avoids node duplication. An example of a weak link is shown in Fig. D.6.1, the unitary link from remote socket B being distinguished as weak by the bar drawn across the link. Examples of the use of weak links will be found later in this Appendix.

D.5 Use of Unitary Links

The use of imaginary locations in transferring displacements unchanged from features to neighbouring points has been described in detail. This

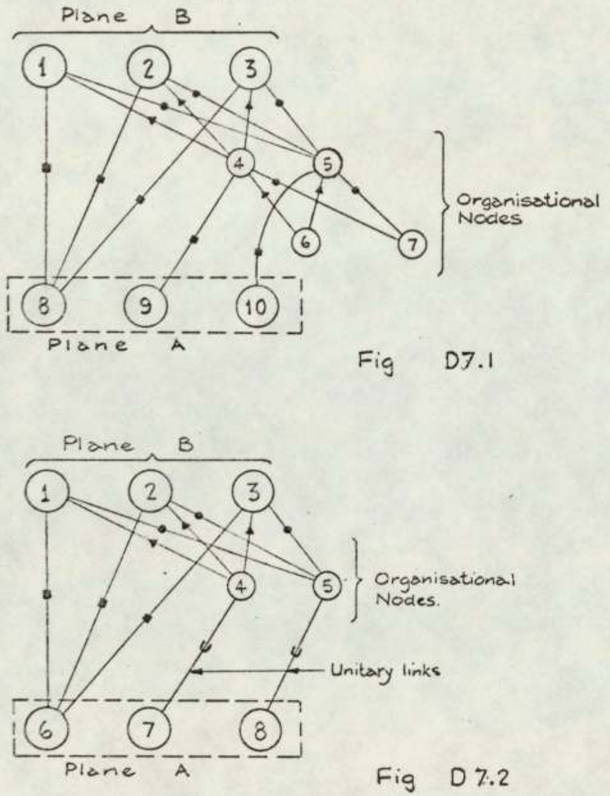


Fig D7.1

Fig D7.2

Fig D7 Use of Unitary Links

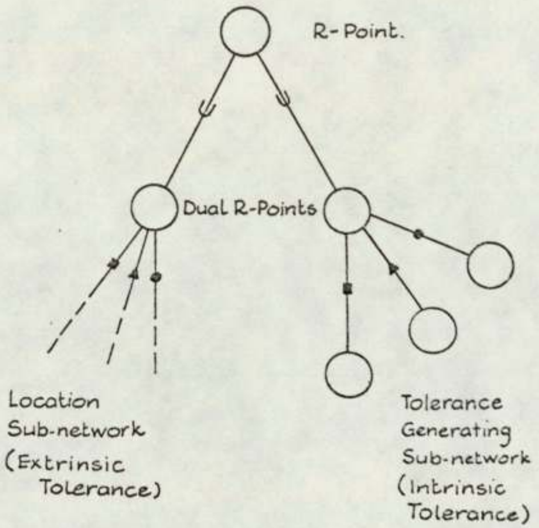


Fig D8. Use of Unitary Links.

device is necessary in many cases - for instance a point located on a plane requires that the point should be located in three plane features. This is clearly impossible to do directly, since the model demands that each feature should be located on a triad of plate, hinge and socket. The point can be located on one of the plates, while displacements at the other two plates must be passed unchanged to coincident features, one a hinge, the other a socket. All calculation of matrices will then be consistent. However, this necessitates the introduction of nodes which do not represent actual points, or features on the assembly, these being termed 'organisational nodes'. They occur in most practical location systems, represent a considerable overhead in storage and also tend to make sub-networks appear more complex than they actually are. A typical example is shown in Fig. D.7.1.

The unitary link is an artificial device which obviates the need for most of these organisational nodes. Instead of the effect of unchanged transfer being obtained by using features at infinity, the matrix is evaluated directly. An advantage of this method is that one or more of the links may be null without affecting the validity of the model. An example is shown in Fig. D.7.2 which is the equivalent of Fig. D.7.1.

The network illustrated represents a plane located on a datum plane and occurs in tolerances of parallelism and angularity. In the examples which follow,unitary links are not usually used but it is probable that in a large practical network, it would be necessary to conserve storage by using them.

Another use of unitary links is to superpose the extrinsic and intrinsic tolerances at an R-point. This may often be done by other means but using the unitary link method avoids complication. An example is shown in Fig. D.8.

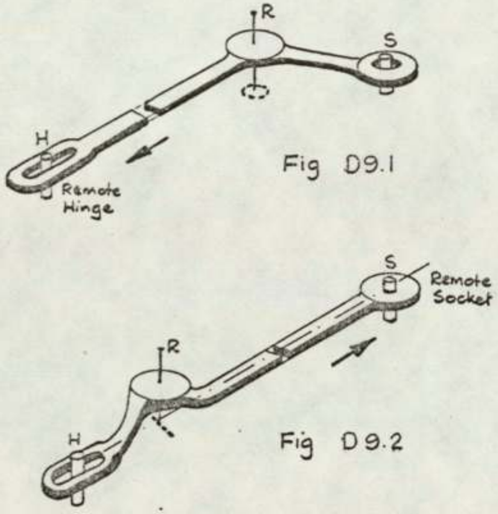


Fig D9.1

Fig D9.2

Fig D9 Equivalent Mechanisms.

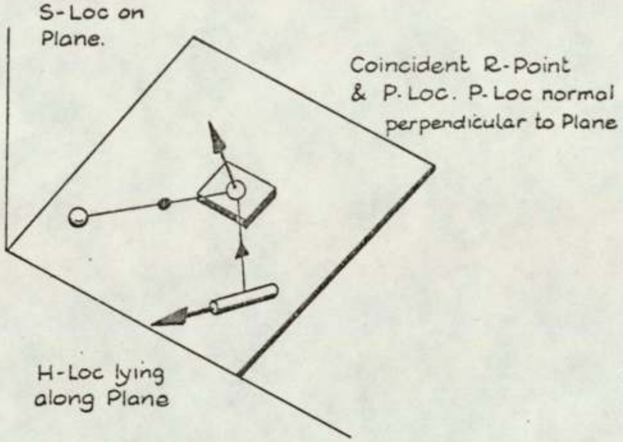


Fig D10. Location in a Plane.

D.6 Equivalent Mechanisms

The devices shown on previous pages are a few of the many which can be created to attain special effects. It is useful, in considering systems of this kind, to think of the location triad as a mechanism - this is particularly useful when the system is two dimensional.

Two examples are shown.

(a) Fig. D.9.1 - Passing a tolerance unchanged from a socket to a neighbouring point. H is a hinge which can in the planar case be imagined as a closely fitting slot around a fixed pin. Clearance at S will be passed to R unchanged when the mechanism is moved around.

(b) Fig. D.9.2 - Passing a tolerance unchanged from a hinge to a neighbouring point. In this case S is a remote socket freely pivoting around a close fitting pin. If S is in line with H, then when the mechanism is moved, clearance at H will be passed unchanged to R.

D.7 Two Dimensional Cases

The most common dimensioning system occurs when features are located on a plane. These are the most easily visualised using equivalent mechanisms. The main principle involved is the stationing of a P-loc coincident with the R-point with normal perpendicular to the plane of interest. This ensures that all displacement at the R-point caused by features on the plane of interest is in the plane. Tolerancing systems on the plane may now be described by using H-locs lying in the plane and S-locs lying on it. An example is shown in Fig. D.10.

Some common cases of dimensioning in two dimensions will be discussed, but it is first necessary to consider another use of remote features.

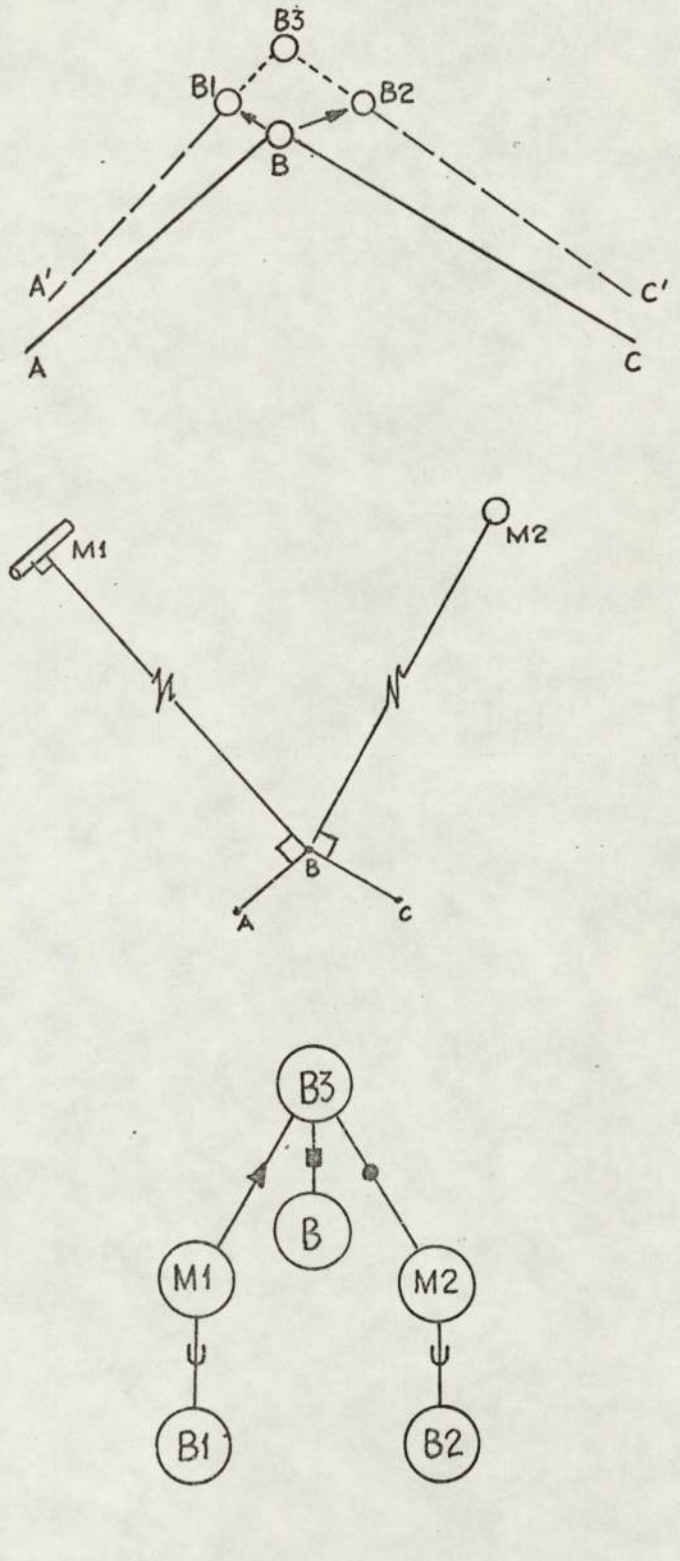


Fig D11. The Intersection of Two Lines

D.7.1 Intersection Points of Lines

Often a point is located geometrically. This usually implies the superposition of two or more real location systems. A common example is the point described by the intersection of two straight lines. In Fig. D.11 lines AB and CB intersect at the point feature B. If lines AB and CB are separately located, there will be an ambiguity at B. Point B1 on line AB will be displaced due to intrinsic displacement of AB and point B2 will be displaced due to intrinsic displacement of CB and these displacements will not be identical. Consequently, the true intersection of AB and CB will be neither B1 nor B2 but some point B3. If B1 and B2 are at right angles, the point B3 will be defined by the vector sum of the displacements of B1 and B2 but this is not generally so. It is convenient to consider the displacement of B1 as being constrained by the line CB which does not contain it. A sub-network for the generation of the displacement of point B3 from its nominal position at B is given by the following:

(a) two remote features M1 and M2 are set at equal distances from the origin, for example at 10^6 units,

(b) M1 is stationed at right angles to AB, M2 at right angles to BC,

(c) either is chosen as an H-loc, the other as a P-loc, the direction of the hinge vector being at right angles to the line joining it to the origin,

(d) displacement at B1 is passed unchanged via a unitary link to M1, displacement at B2 is passed similarly to M2,

(e) the R-point is taken at the nominal position of B.

This sub-network is useful in various situations and its validity may easily be proved by considering instantaneous centres.

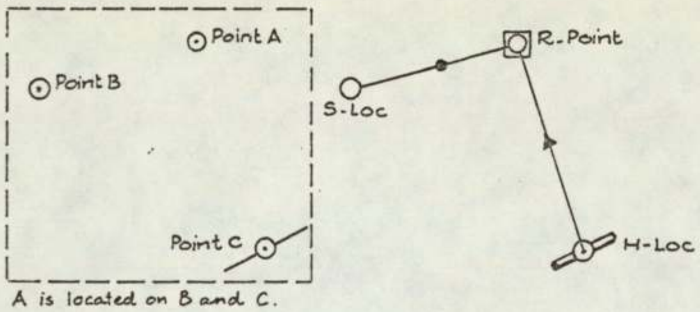


Fig D12. A Point Located on a Point and a Point on a Line.

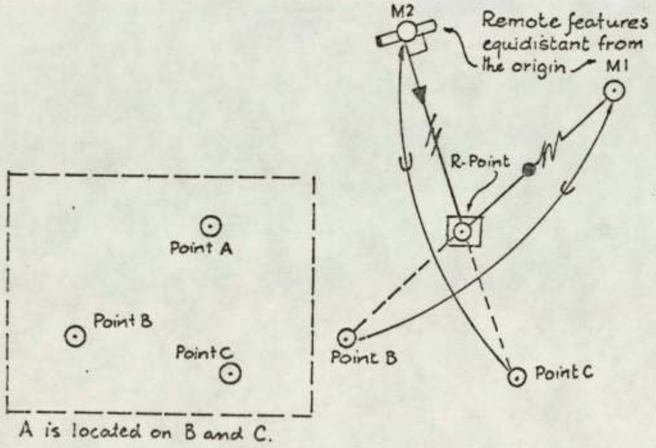


Fig D13. A Point Located on Two Points

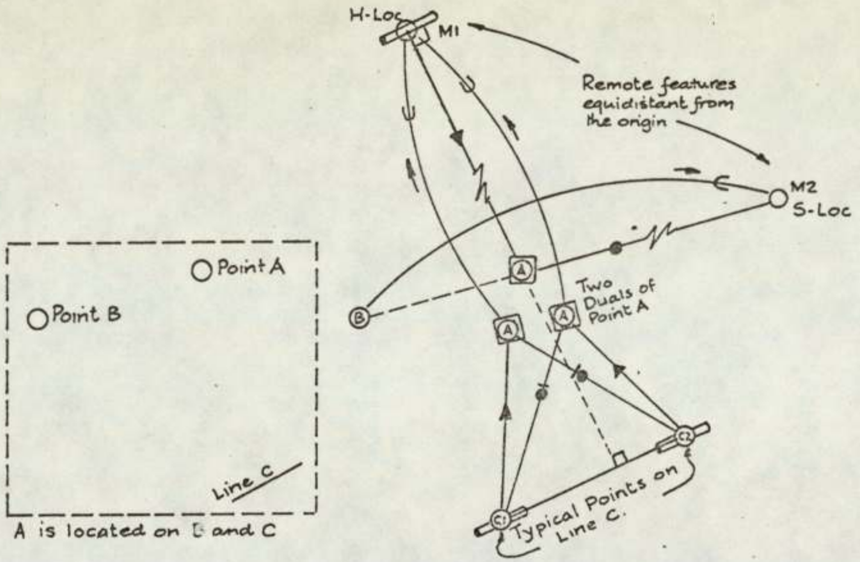


Fig D14 A Point Located on a Point and a Line

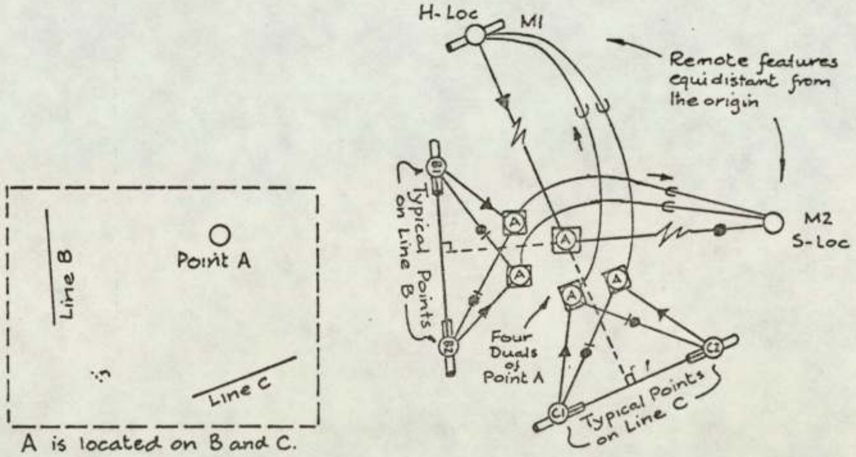


Fig D15 A Point Located on Two Lines

D.7.2 Location of a Point in a Plane

(a) Location by a point and a point on a line.

This is the simplest case, since it is a real location. An example is shown in Fig. D.12.

(b) Location by the distances from two points.

This case may be described by considering it as a case of constrained displacement. M1 and M2 are taken along the lines joining the points and the R-point as shown in Fig. D.13.

(c) Location by the distance from a line and distance from a point.

The remote features M1 and M2 are set at right angles to the line and along the line joining the point and the R-point as shown in Fig. D.14.

(d) Location by the perpendicular distances from two lines.

M1 and M2 are set at right angles to each line. The point is located separately on each line by using two networks as described in (a) and superimposing them. Note that only the H-loc displacement is passed from each point, weak links being used, and that each point on the line is used in a dual capacity as H-loc and S-loc. An example is shown in Fig. D.15. This case is very common, occurring in coordinated dimensions.

In each of these cases, the sub-network describes the extrinsic tolerance. Intrinsic tolerance is handled differently for the two standard methods.

(a) Positional tolerance is applied directly at the R-point.

(b) Tolerances on dimensions are applied directly at the remote features usually.

They may all be generalised to three dimensions.

The most common cases of tolerancing as shown in BS 308 will now be described.

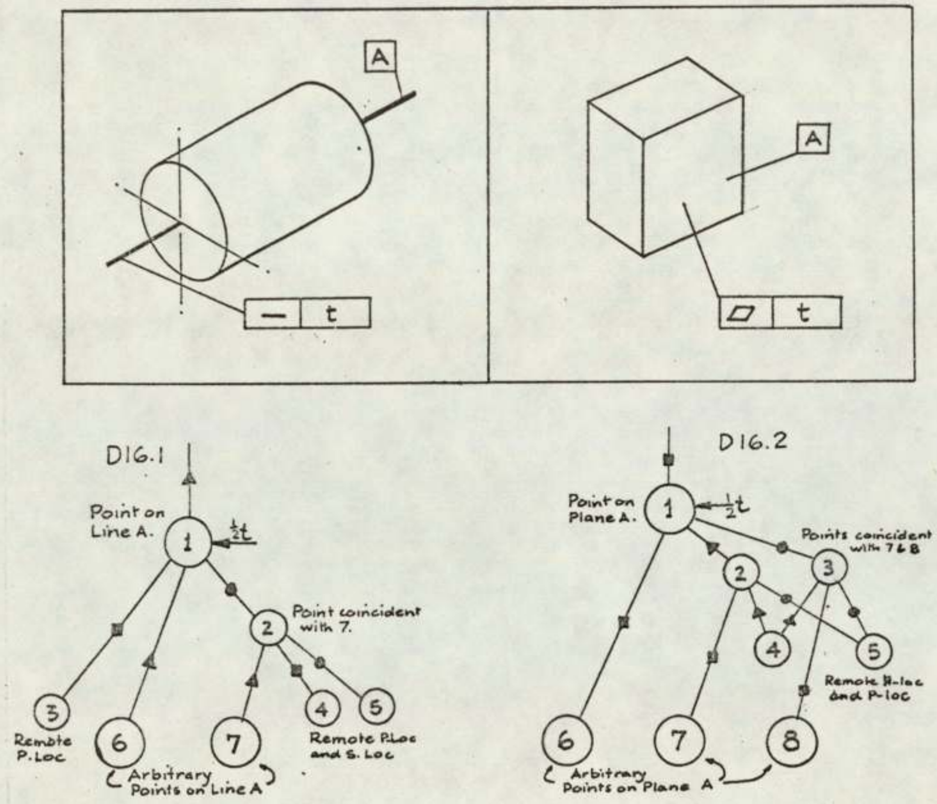


Fig D16 Straightness & Flatness.

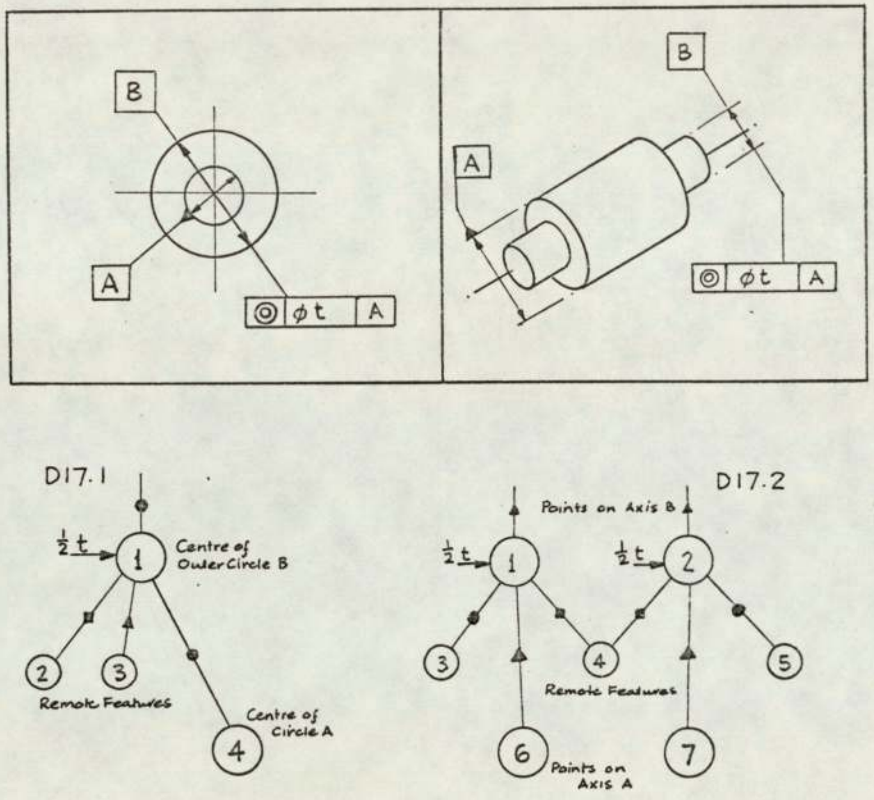


Fig D17 Concentricity.

D.8 Three Dimensional Systems

D.8.1 Tolerances of Straightness and Flatness

(a) Tolerance of straightness of a line - Fig. D.16.1.

The centre line A of the cylinder is subject to a straightness tolerance $t/_2$. This is applied directly to node 1. Nodes 6 and 7 describe the centre line; and the network transmits displacements at 6 and 7 to the node 1.

(b) Tolerance of flatness of a plane - Fig. D.16.2.

Plane A is subject to a flatness tolerance $t/_2$. This is applied directly to node 1, which describes a point on the plane. Nodes 6, 7 and 8 describe the plane; and the network transmits displacements at 6, 7 and 8 to the node 1.

D.8.2 Tolerances of Concentricity

(a) Concentricity of a point - Fig. D.17.1.

The centre of circle B is required to lie within a circle diameter t, concentric with the centre of the datum circle A. Displacement at node 4 - centre of circle A - is passed unchanged to node 1 which describes the centre of circle B. The tolerance $t/_2$ is applied directly to node 1.

(b) Concentricity of a line - Fig. D.17.2.

The axis of the cylinder B is required to be contained within a cylinder diameter t co-axial with cylinder A. Displacements at nodes 6 and 7 which describe the axis of A are passed unchanged to nodes 1 and 2 describing axis B. The tolerance $t/_2$ is applied directly to nodes 1 and 2. In this case, nodes 1 and 2 are chosen to be coincident with nodes 6 and 7 which results in a simple network. In the more general case where 1 and 2 are not coincident with 6 and 7, then displacements at both 6 and 7 will result in displacements at both of 1 and 2; and the network is consequently more complex.

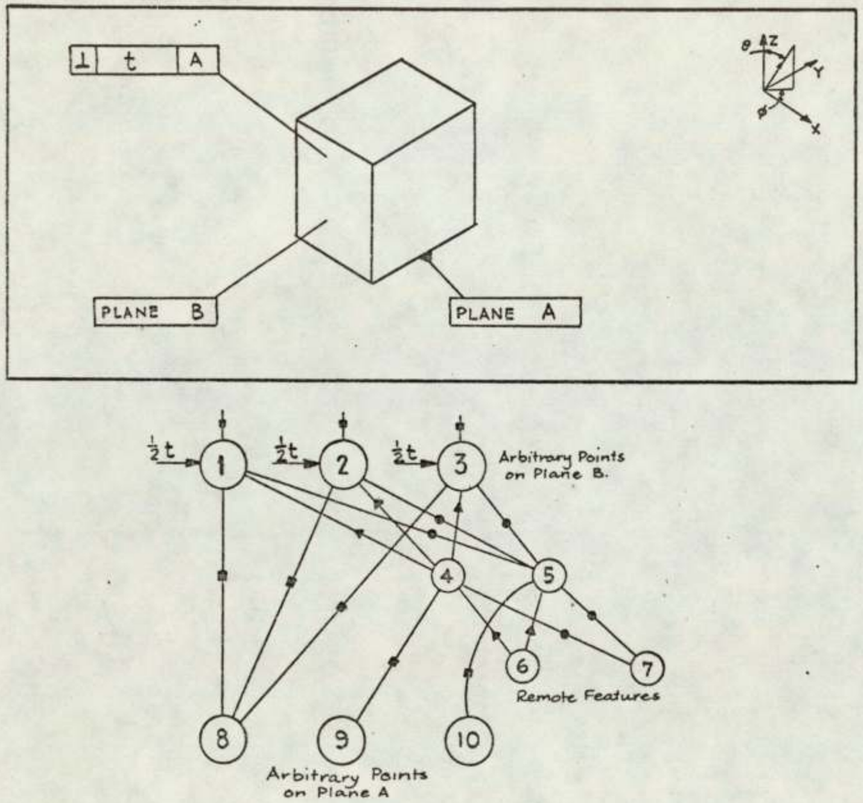


Fig D18 Squareness

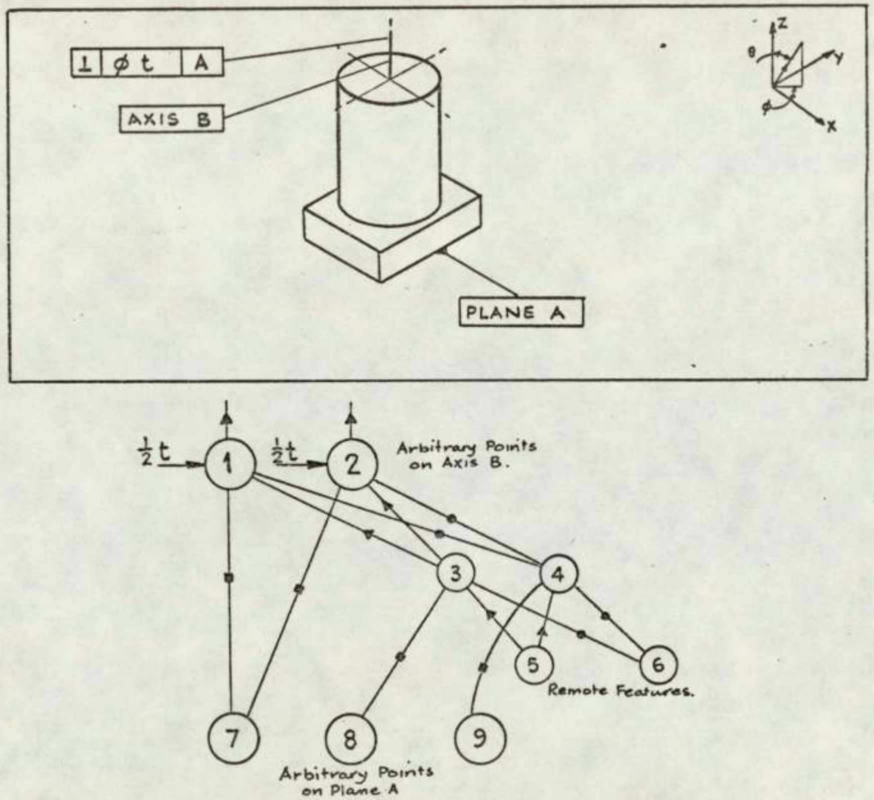


Fig D19 Squareness

D.8.3 Squareness Tolerances

(a) Squareness of a plane relative to a datum plane - Fig. D.18.

Datum plane A is described by plates stationed arbitrarily on A - nodes 8, 9 and 10. Displacements at 9 and 10 are passed unchanged to coincident features 4 and 5 using remote features 6 and 7. Features 4 and 5 act as hinge and socket locations for features 1, 2 and 3 which describe the dependent plane B. Features 1, 2 and 3 are also located directly on feature B. The squareness tolerance $t/2$ is applied directly at 1, 2 and 3, indirect displacements at 8, 9 and 10 being passed through the network.

(b) Squareness of a line relative to a datum plane - Fig. D.19.

The case shown is that of an axis of symmetry which is square to a plane within a cylindrical tolerance zone. The network organisation is similar to that in (a) except that there are two hinge features 1 and 2 describing the dependent line.

An alternative case of this kind is that of a line on a plane square to the datum plane, within a rectangular tolerance band. This system is identical to the one shown in Fig. D.19 except that 1 and 2 will be plate features with normal in the plane of the tolerance band.

D.8.4 Tolerances of Angularity

(a) Angularity of a face relative to a datum plane - Fig. D.20
- Tolerance band.

Fig. D.20 shows the method of specifying angularity tolerances recommended in BS 308. The network system is identical with that for a squareness tolerance which is a particular case of angularity tolerance, if this tolerancing method is used.

(b) Angularity of a face relative to a datum plane - Tolerance on angle.

For this non-standard case, the angle is shown as, for example

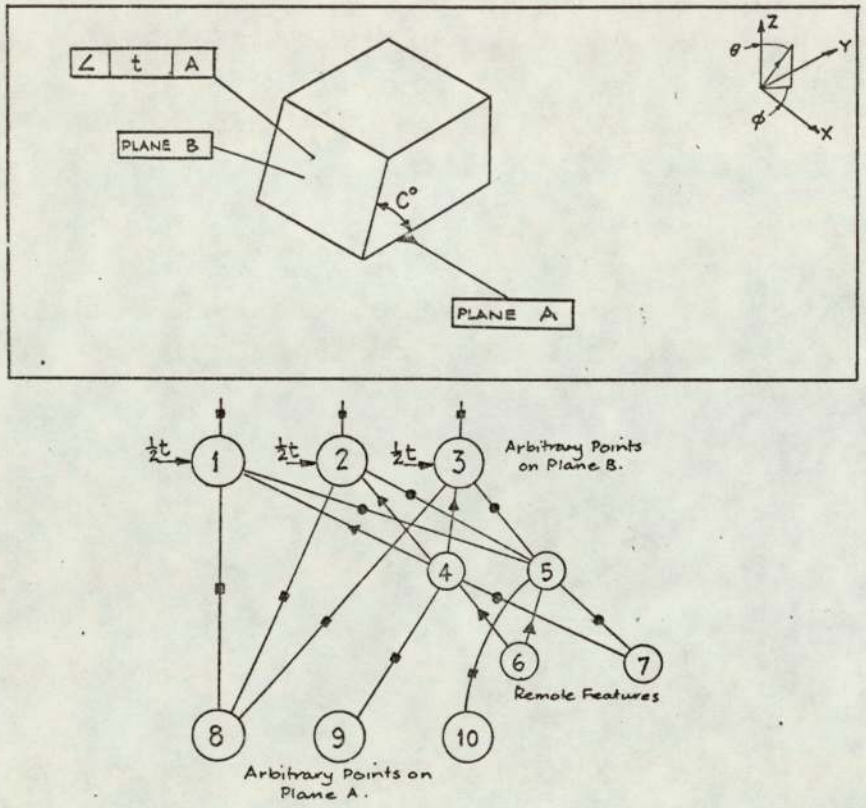


Fig D20 Angularity

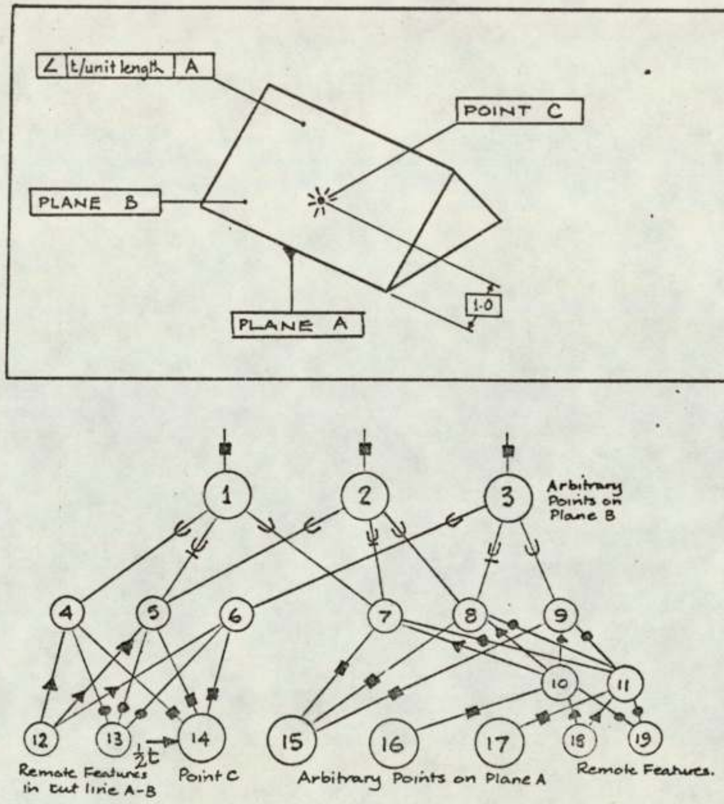


Fig D21 Angularity

$60^{\circ} \pm 1^{\circ}$. The easiest way to deal with this case is calculate the actual tolerances at features 1, 2 and 3; and to apply these at features 1, 2 and 3 as shown in Fig. D.20. Each tolerance value will be different in this case, but the network will be the same as (a).

(c) Angularity of a face relative to a datum plane - Unit tolerance, i.e. tolerance/unit distance.

Again, in this non-standard case, it is most convenient to calculate the actual tolerance at features 1, 2 and 3, so as to retain the same network. However, since this case occurs fairly frequently, a separate treatment follows.

In the case shown in Fig. D.21, tolerance $t/_2$ is applied at unit distance from the intersection line of the two planes A and B. The point of application is labelled as 'Point C'.

This construction uses unitary links. The displacements due to displacement of datum plane A (indirect tolerance), and the unit angular tolerance t are superposed by means of unitary links. The weak links at nodes 1, 2 and 3 are redundant, and may be omitted from the diagram, but are included for the sake of consistency - each node has outdegree 3.

The network shown can be simplified to some extent, and some of the nodes omitted; but in all the networks shown in the examples, the most direct method has been used even if this has necessitated using extra nodes.

D.8.5 Tolerances of Symmetry

(a) Symmetric tolerance - datum planes parallel - Fig. D.22.

Again unitary links are used; and weak links to ensure that unwanted tolerances are not passed along a path. In general, the presence of a unitary link implies that the location system is not physical but geometric. In view of the number of nodes used in this system, it might be considered useful to provide an artificial 'half-unitary' node, which

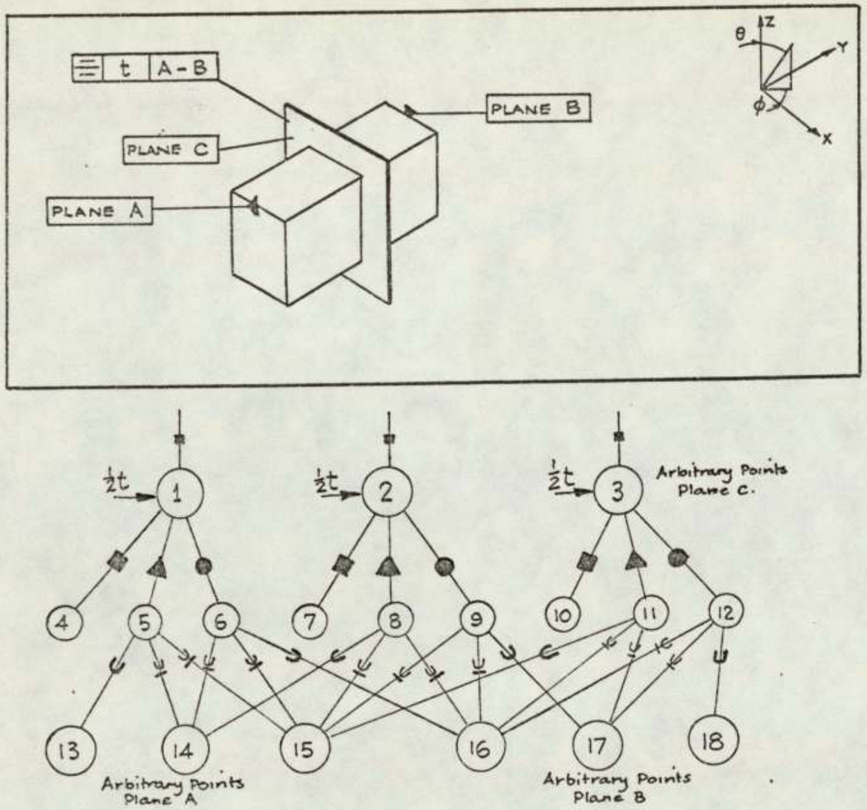


Fig D22 Symmetry.

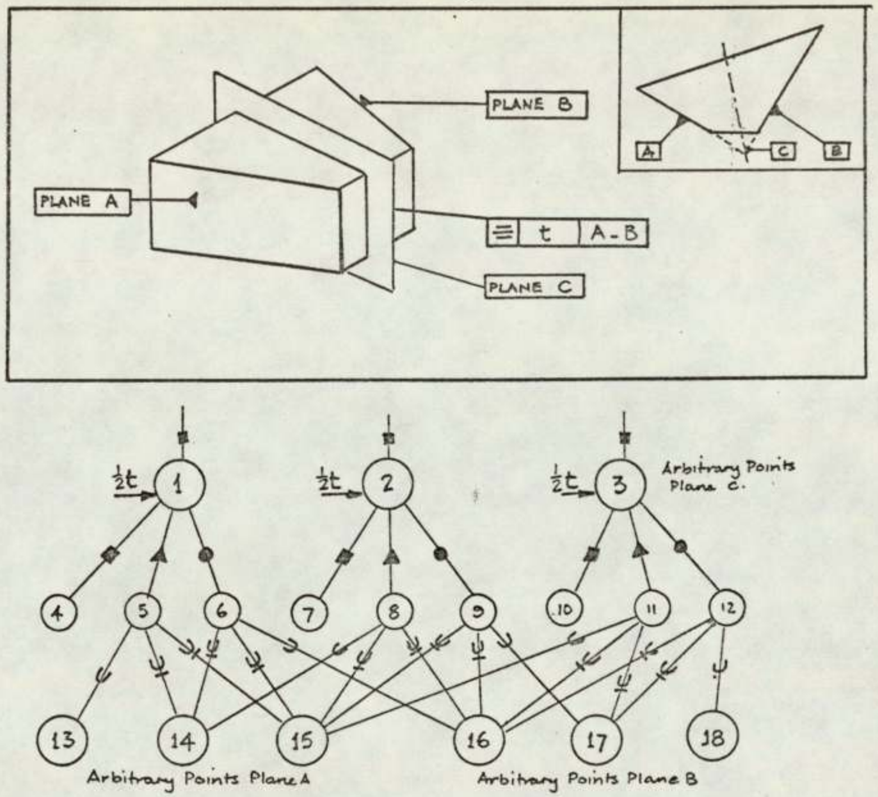


Fig D23 Symmetry

would considerably simplify the network. Once again, the view is taken that it is better to be a little prodigal with nodes rather than to complicate the system.

(b) Symmetric tolerance - datum planes not parallel - Fig. D.23.

This case is more general than (a), but is basically the same network.

D.8.6 Tolerances of Parallelism

(a) Parallelism of a plane relative to a plane - Fig. D.24.

Nodes 8, 9 and 10 describe the datum plane; nodes 1, 2 and 3 describe the located plane.

D.8.7 Coordinate Distances from Three Planes

(Cartesian coordinate system) - Fig. D.25.

Point P is located on three flat faces A, B and C. Each of the three planes is defined by a sub-network shown dotted in Fig. D.25 and previously described in section D.3 . Remote features X, Y and Z are used to separate out components of displacements in the directions indicated by their names. Z is a plate stationed at infinity along the Z axis, with normal along the Z axis, X is a hinge stationed at infinity along the X axis with direction at right angles to the X axis; Y is a socket stationed at infinity along the Y axis. These will select the components in the directions of the axes along which they are stationed and these components are passed to P using unitary links.

The network for this system appears rather complicated and requires a disproportionate number of nodes. However, each extra point located on this system only needs four extra nodes similar to P; and the spare unitary links at X, Y and Z may also be used for two extra points, extra X, Y and Z nodes being necessary for each three result points. The number of nodes can also be reduced by using the remote features in more complicated ways. For instance, instead of the six remote nodes used in the

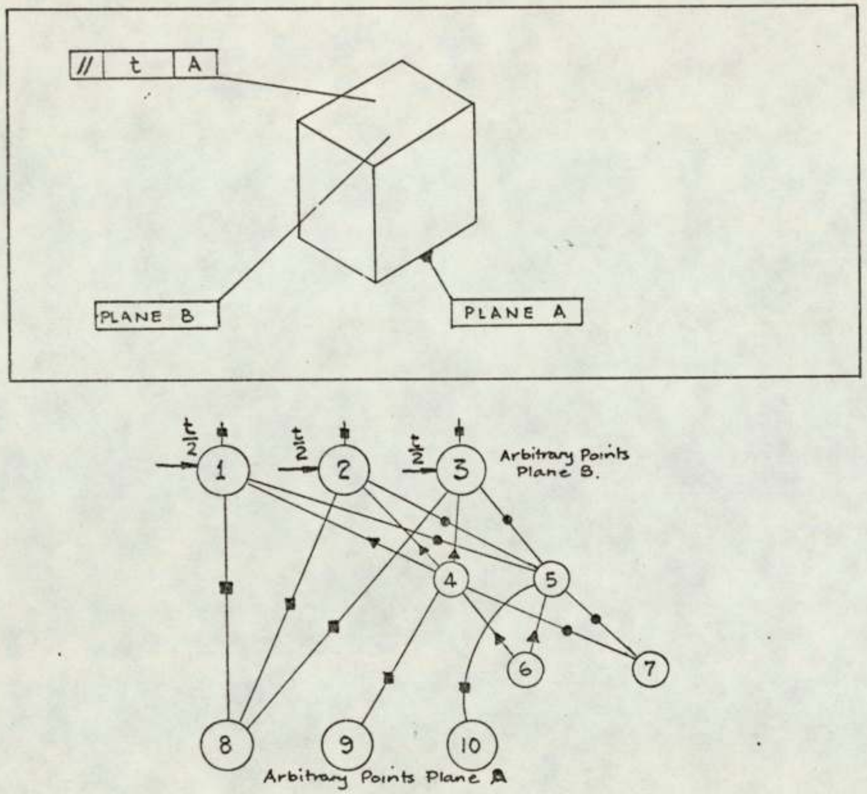


Fig D24 Parallelism.

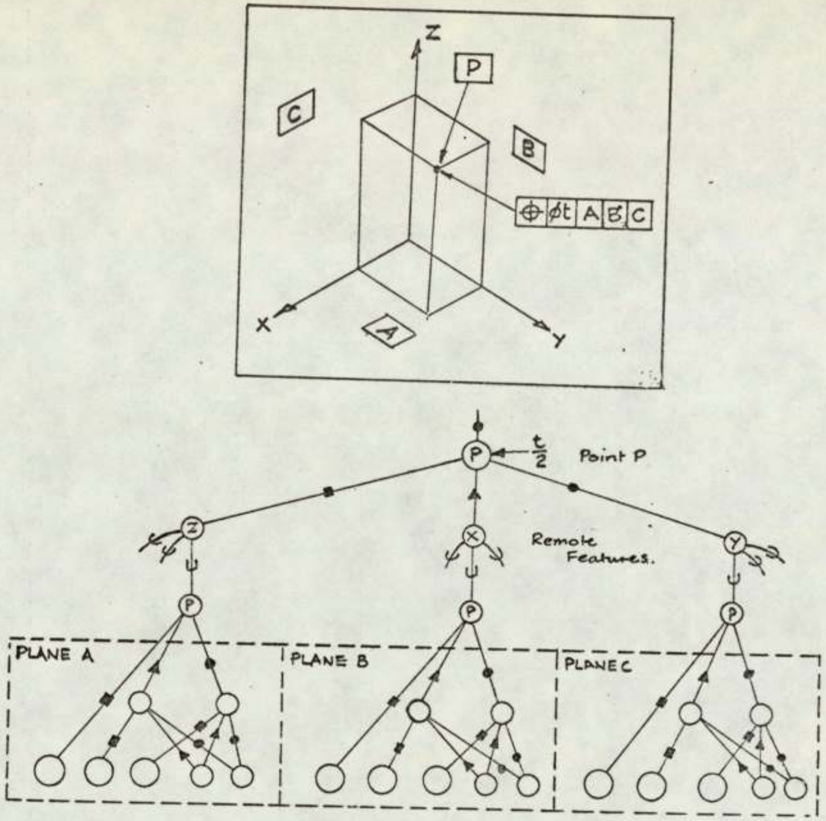


Fig D25. Co-ordinates

sub-network for planes A, B and C, three features may be used jointly, each feature being used once as a hinge and once as a socket. However, once again the most direct network system has been chosen. Fig. D.25 shows a spherical tolerance $t/_2$ applied directly at the result point P. If the tolerance is parallelepipedal, then the three components may be applied directly at X, Y and Z.

The sub-networks described in this Appendix are by no means an exhaustive set but they should be sufficient to handle most common dimensioning systems.

APPENDIX E

PRACTICAL EXAMPLES

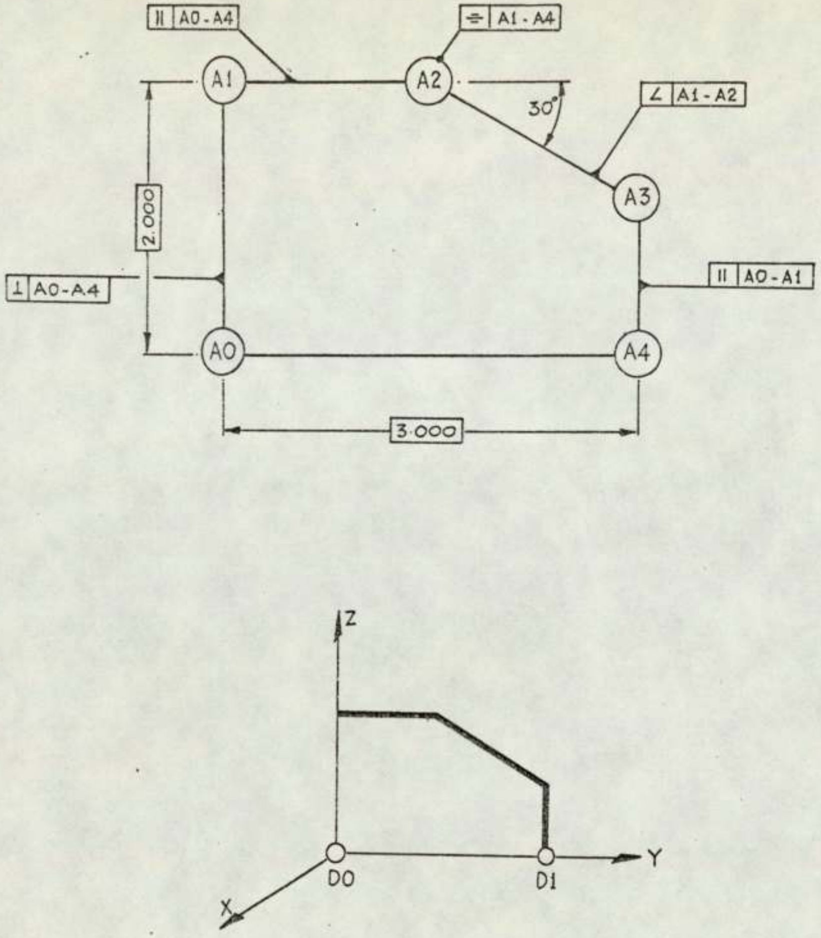
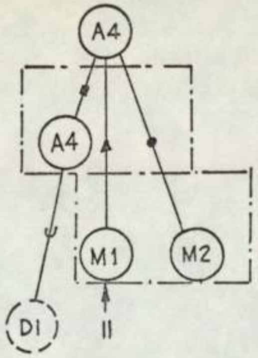


Fig E1. Definition of Plate.



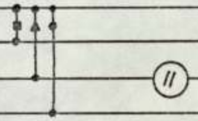
Node	Network	Description	Co-ords			Angs		Links		
			X	Y	Z	θ	ϕ	■	▲	●
1		Point A4	0	3	0	0	0	2	3	4
2		P-loc A4	0	3	0	90	0	0	0	0
3		Remote H-loc	0	-106	0	0	90	0	0	0
4		Remote S-loc	0	0	-106	0	0	0	0	0

Fig E2. Description of Point A4.

E.1 Examples

The cases considered will be didactic rather than practical and the dimensioning sufficiently eccentric as to include most of the common datum systems met in practice. It is quite difficult to check some of the results, and this has been done mainly by calculation but occasionally by drawing.

(a) Two dimensional system. Fig. E.1

The plate is defined by the five points A0, A1, A2, A3 and A4. Each of these points may be considered as two subsidiary points coincident in the plane. For example, point A1 lying at the intersection of lines A0 - A1 and A1 - A2 may be considered as points A1a lying on A0 - A1 and A1b lying on A1 - A2. The lamina is located in orthogonal coordinates as shown in Fig. E.1 and since all displacement is in the plane of the lamina, a plate feature is set at each point with its normal perpendicular to the plane of the lamina. The input tolerance at this P-loc will be zero, thus ensuring that displacements at all the features on the lamina will be in its plane. The datum is chosen as the point D0 (coincident with A0) and a line through D0 and D1 (coincident with A4).

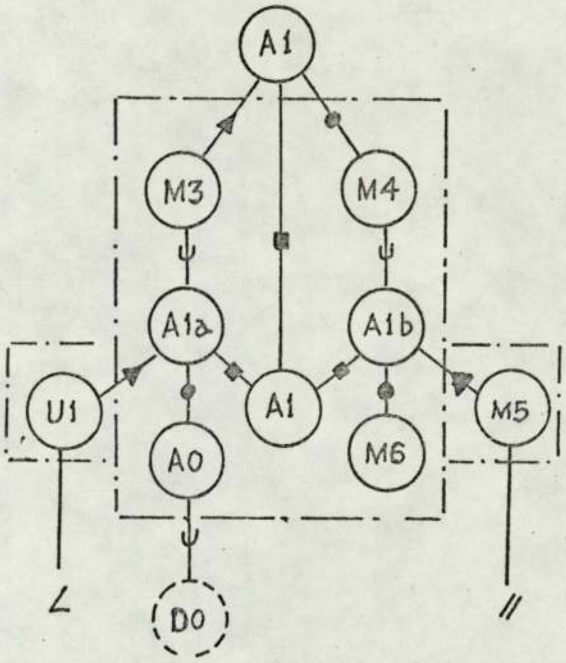
The five points will be considered separately.

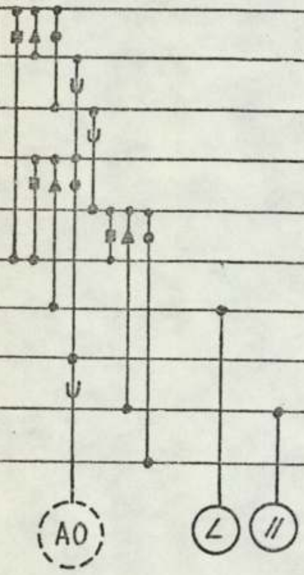
(i) Point A4. Fig. E.2.

A parallelism tolerance is applied directly to A4. Since only one tolerance acts at this point, it is not necessary to use the dual point.

(ii) Point A1. Fig. E.3.

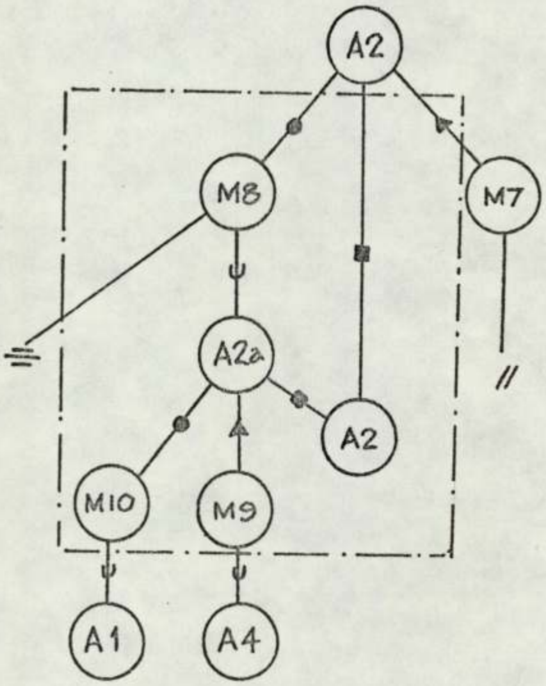
Two tolerances are applied at A1, an angularity tolerance at A1a and a parallelism tolerance at A1b. These will be compounded to give the actual position of point A1. Since the angularity tolerance is quoted as 'tolerance per unit distance', a subsidiary point U1 is taken at unit distance from A0 along A0 - A1 and an H-loc set at U1. This is passed via a tolerance generating network to A1a.



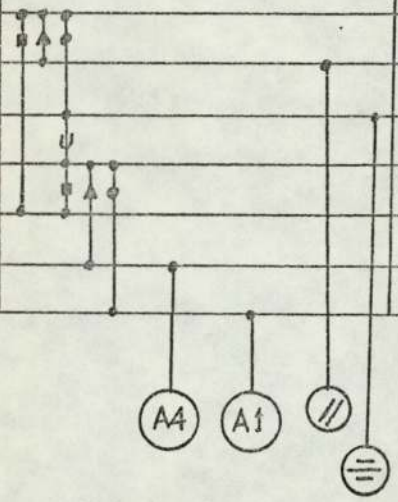
Node	Network	Description	Co-ords			Angs		Links		
			X	Y	Z	θ	φ	■	▲	●
1		Point A1	0	0	2	0	0	6	2	3
2		Remote H-loc M3	0	-106	0	0	90	4	0	0
3		Remote S-loc M4	0	0	-106	0	0	5	0	0
4		Dual Point A1a	0	0	2	0	0	6	7	8
5		Dual Point A1b	0	0	2	0	0	6	9	10
6		P-loc A1	0	0	2	90	0	0	0	0
7		U1	0	0*	1	0	90	0	0	0
8		A0 - location	0	0	0	0	0	●	0	0
9		Remote H-loc M5	0	0	-106	90	90	●	0	0
10		Remote S-loc M6	0	-106	0	0	0	0	0	0

* Perturbed Dimension
● External Link

Fig E3. Description of Point A1

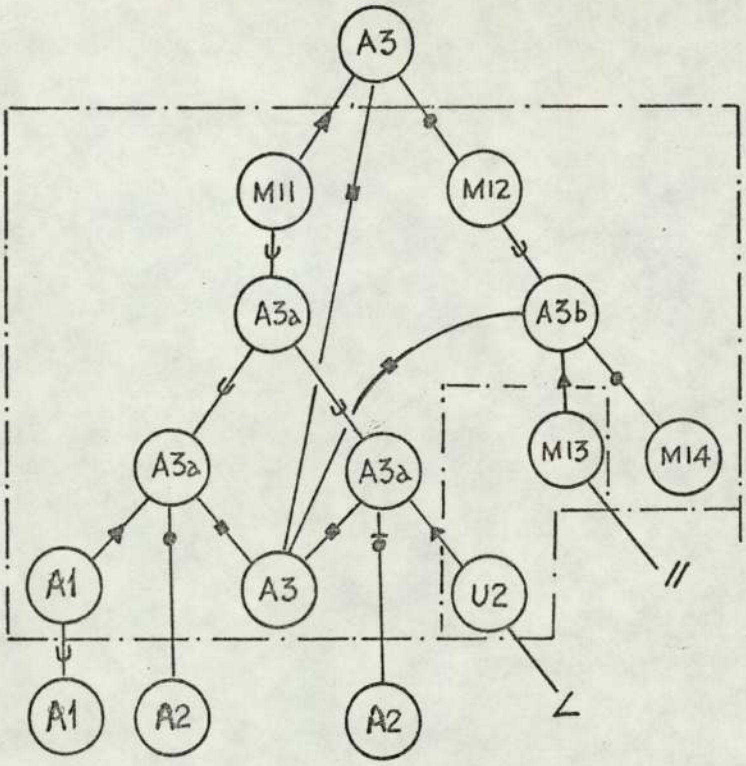


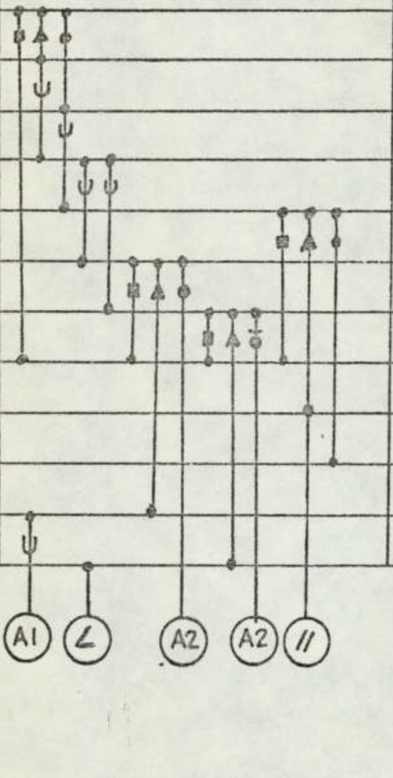
Node	Network	Description	Co-ords			Angs		Links		
			X	Y	Z	θ	φ	■	▲	●
1		Point A2	0	1.5	2	0	0	5	2	3
2		Remote H-loc M7	0	0	-106	90	90	●	0	0
3		Remote S-loc M8	0	-106	0	0	0	4	0	0
4		Dual point A2a	0	1.5	2	0	0	5	6	7
5		P-loc A2	0	1.5	2	90	0	0	0	0
6		Remote M9	0	0	106	0	90	●	0	0
7		Remote M10	0	0	-106	0	0	●	0	0



● External Link.

Fig E4. Description of Point A2



Node	Network	Description	Co-ords			Angs		Links		
			X	Y	Z	θ	φ	■	▲	●
1		Point A3	0	3	1.134	0	0	8	2	3
2		Remote H-loc M11	0	$\cdot5_{10}6$	$\cdot866_{10}6$	-60	90	4	0	0
3		Remote S-loc M12	0	$_{10}6$	0	0	0	5	0	0
4		Dual Point A3a	0	3	1.134	0	0	6	7	0
5		Dual Point A3b	0	3	1.134	0	0	8	9	10
6		Dual Point A3a'	0	3	1.134	0	0	8	11	●
7		Dual Point A3a"	0	3	1.134	0	0	8	12	-●
8		P-loc A3	0	3	1.134	90	0	0	0	0
9		Remote H-loc M13	0	$-_{10}6$	0	0	90	●	0	0
10		Remote S-loc M14	0	0	$-_{10}6$	0	0	0	0	0
11		Point A1 - H-loc	0	0	2	90	90	●	0	0
12		Point U2.	0	2.366	1.5	-60	90	0	0	0

● External Link

-N Weak Link.

Fig E5 Description of Point A3

Since A1 is located on a datum which is earthed, there is no extrinsic tolerance network necessary and the parallelism tolerance may be applied directly to point A1b. Displacements at A1a and Alb are compounded using remote points M3 and M4 at right angles to lines A0 - A1 and A1 - A2 respectively. This completes the definition of point A1.

This location system may be imagined as a 'black box' with one or more input terminals at which are applied extrinsic tolerances due to location, one or more input terminals at which are applied intrinsic tolerance due to permitted displacement of the point itself and one output terminal which may be connected with a unitary link to another 'black box'.

(iii) Point A2. Fig. E.4.

There are two tolerances acting at this point. As with A1, the parallelism tolerance acts directly and may be applied through a tolerance defining sub-network to point A2a. The symmetric tolerance depends on the two defining points A1 and A4, and is passed through a sub-network to A2b, using the remote features M3 and M4 stationed as shown in the figure. This completes the definition of point A2.

(iv) Point A3. Fig. E.5.

Dual point A3a is fixed in relation to A1 (a point on A1 - A2) and point A2. Displacements at these features are transmitted to A3a through a sub-network as shown. The angularity tolerance is passed through a tolerance generating sub-network to A3a, the weak link from A2 ensuring that tolerance is not passed twice from the same point. The parallelism tolerance is applied directly to point A3b and the displacements compounded using remote features normal to lines A2 - A3 and A4 - A3.

The plate is now defined. Each of the 'black boxes' describing the points may be tested separately before being linked into a full network

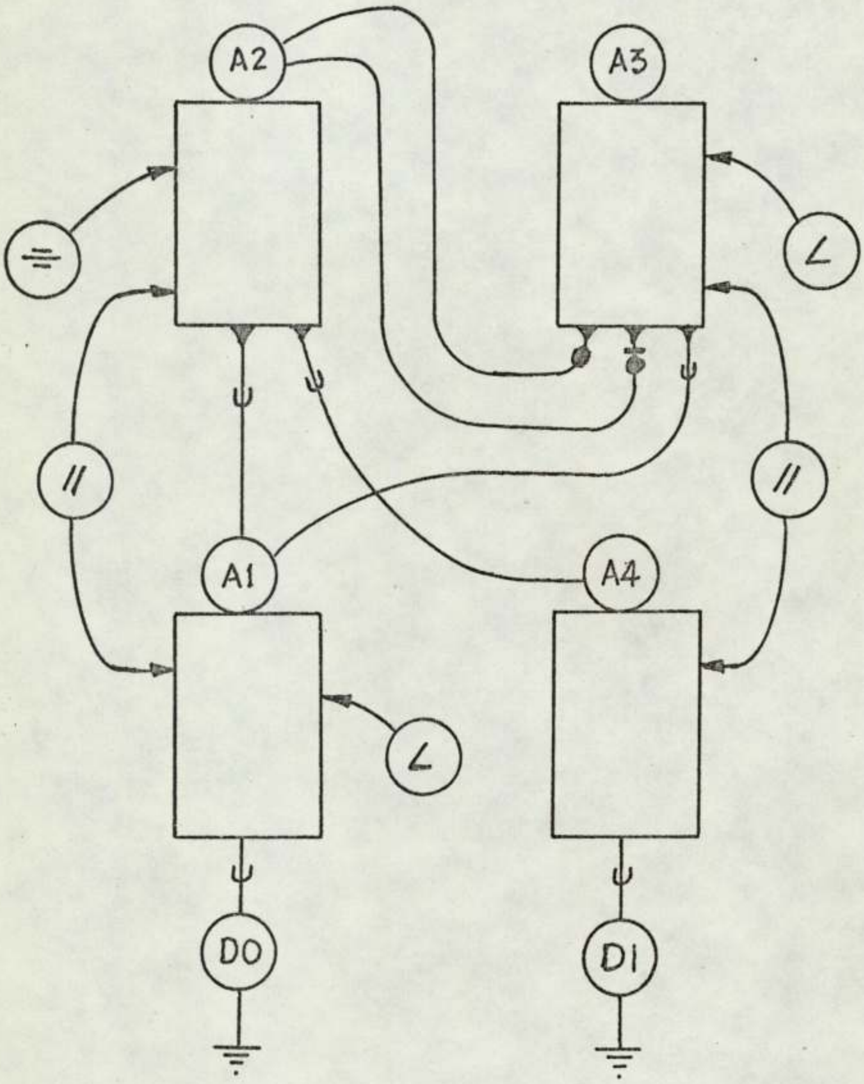


Fig E6 Description of Plate.

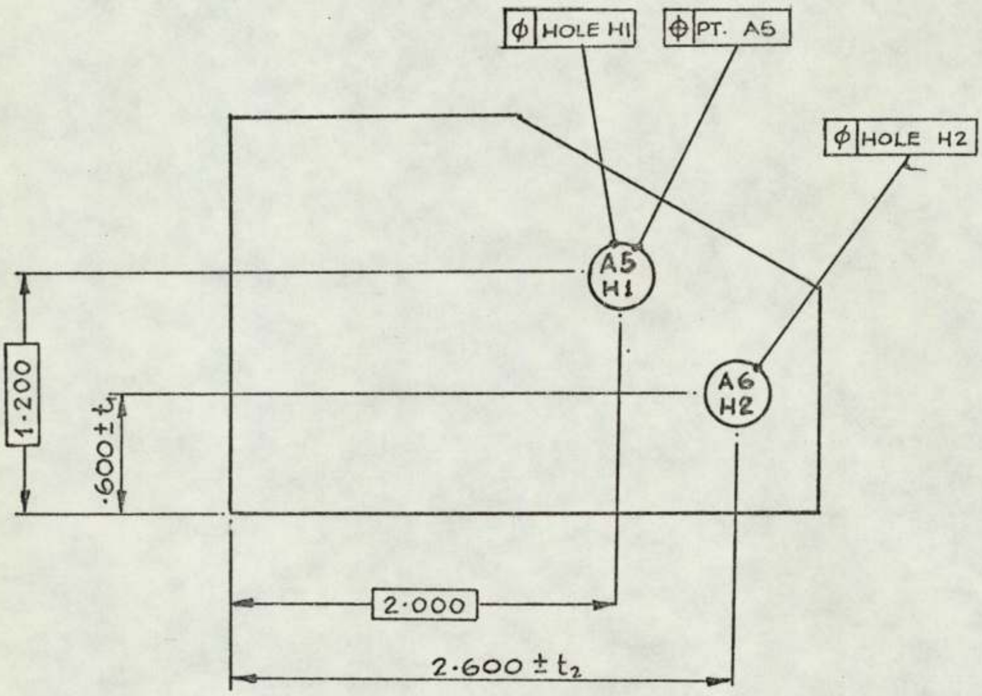
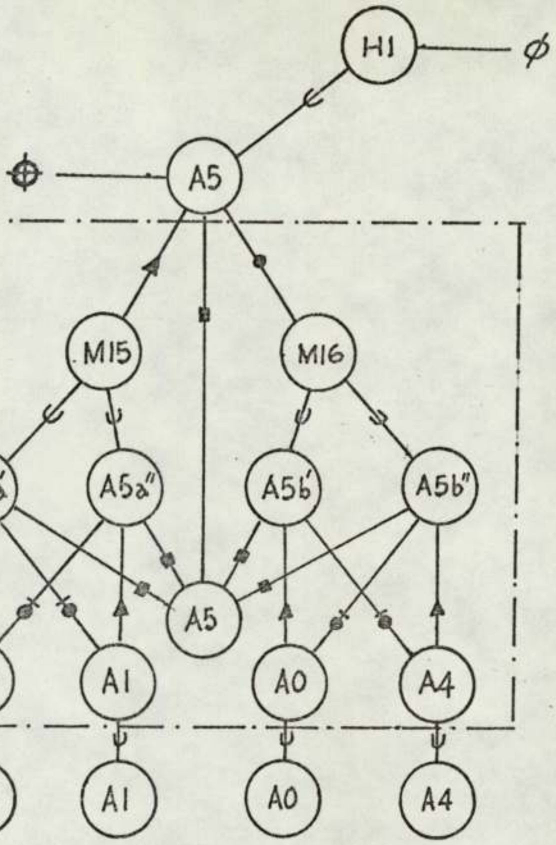
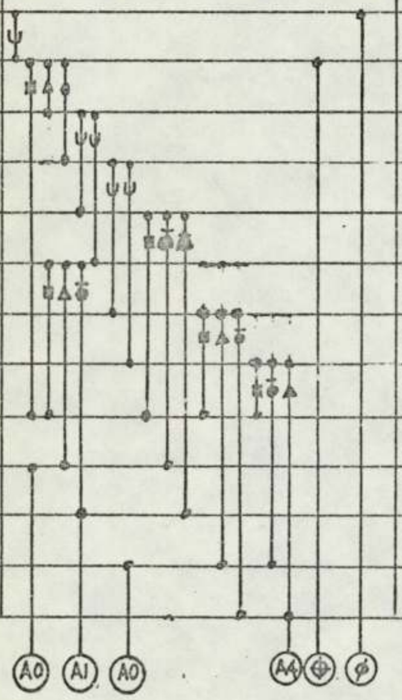


Fig E7. Definition of Drilled Plate





Node	Network	Description	Co-ords			Angs		Links		
			X	Y	Z	θ	φ	■	▲	●
1		Hole H1	0	2 *	1·2	0	0	2	0	0
2		Point A5	0	2	1·2	0	0	9	3	4
3		Remote H-loc M15	0	-10^6	0	0	90	5	6	0
4		Remote S-loc M15	0	0	-10^6	0	0	7	8	0
5		Dual point A5a'	0	2	1·2	0	0	9	10	-11
6		Dual point A5a"	0	2	1·2	0	0	9	11	-10
7		Dual point A5b'	0	2	1·2	0	0	9	12	-13
8		Dual point A5b"	0	2	1·2	0	0	9	13	-12
9		P-loc A5 - loc. in Z-Y.	0	2	1·2	90	0	0	0	0
10		Point A0 - location	0	0	0	0	90	●	0	0
11		Point A1 - "	0	0	2	0	90	●	0	0
12		Point A0 "	0	0	0	0	90	●	0	0
13		Point A4 "	0	3	0	0	90	●	0	0

● External links

-N Weak links

* Perturbed dimension.

Fig E8 Descriptions of Point A5 and Hole H1

(Fig. E.6). Using separate definitions of points is rather wasteful of nodes, many more unitary links being used than are necessary but it is considered much easier to develop the network point by point using standard cases than to regard it as an entity.

The system may now be used to establish sensitivities and in this case those on the height A4 - A3 were found. The non-zero coefficients are:

Tolerance		Sensitivity at A4 - A3
line A0 - A1	Angularity with D0 - D1	0.577
line A1 - A2	Parallelism with D0 - D1	3.667
line A3 - A4	Parallelism with A0 - A1	0.866
line A2 - A3	Angularity with A1 - A2	2.000
point A2	Symmetry with A1 and A4	0.577

These coefficients may be used to find the tolerance on the height A4 - A3 for existing values of tolerance. For instance if angularity tolerances are 1 in 100 (about $\pm \frac{1}{2}^{\circ}$), parallelism is ± .010 and the tolerance of symmetry ± .020, the tolerance on height A4 - A3 will be ± .083.

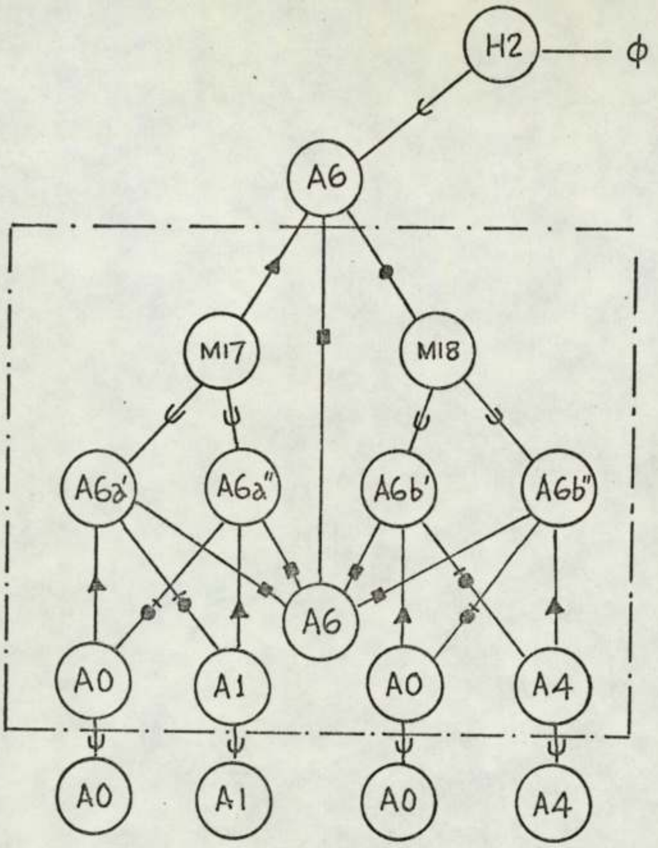
Alternatively, they may be used to allocate manufacturing tolerance either in an informal way or, if cost details are known, by using an optimising program.

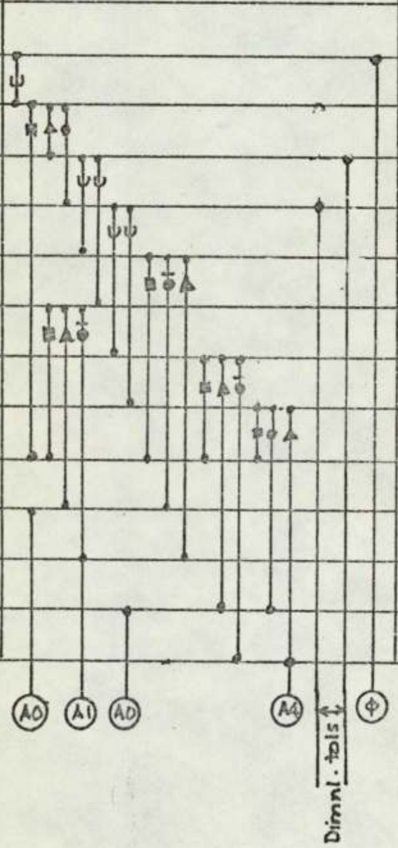
(b) Two dimensional system.

Two holes will now be added to the plate analysed in (a). Both are located by dimensions from sides A0 - A1 and A0 - A4, but centre A5 is positionally toleranced, while centre A6 is dimensionally toleranced (Fig. E.7).

(i) Centre A5. Fig. E.8.

The dimensional system is described by the network shown in the figure. The positional tolerance may be applied directly to A5 - this being a general principle in positional tolerancing. Hole H1 may be



Node	Network	Description	Co-ords			Angs		Links		
			X	Y	Z	θ	φ	■	▲	●
1		Hole H2	0	2.6*	·6	0	0	2	0	0
2		Point A6	0	2.6	·6	0	0	9	3	4
3		Remote H-loc M17	0	-106	0	0	90	5	6	0
4		Remote S-loc M18	0	0	-106	0	0	7	8	0
5		Dual point A6a'	0	2.6	·6	0	0	9	10	-11
6		Dual point A6a''	0	2.6	·6	0	0	9	11	-10
7		Dual point A6b'	0	2.6	·6	0	0	9	12	-13
8		Dual point A6b''	0	2.6	·6	0	0	9	13	-12
9		P-loc A6 -loc in Z-Y	0	2.6	·6	90	0	0	0	0
10		Point A0-location	0	0	0	0	90	●	0	0
11		Point A1 "	0	0	2	0	90	●	0	0
12		Point A0 "	0	0	0	0	0	●	0	0
13		Point A4 "	0	0	3	0	0	●	0	0

● External links
-N Weak links
* Perturbed dimension

Fig E9 Descriptions of Point A6 and Hole H2.

located on its centre using a unitary link, the diametral tolerance being applied directly.

(ii) Centre A6. Fig. E.9.

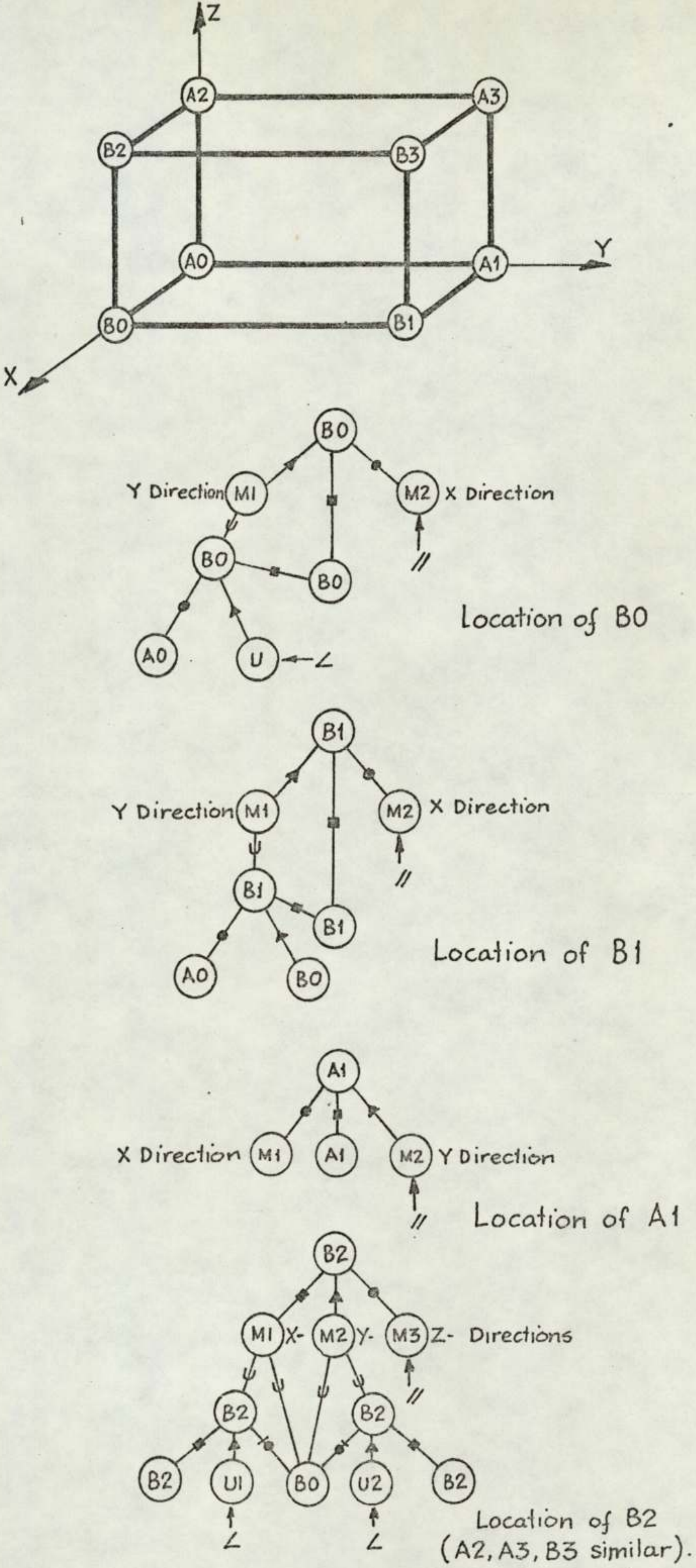
This is located in a similar fashion to A5. The tolerance may be applied by means of a tolerance generating network or if storage is tight, it may be applied to the location network. H2 is located on A6 by means of a unitary link.

This completes the definition of the plate and holes and sensitivity coefficients may be computed. Two sets are shown below for the distances between points on holes H1 and H2, and between a point on H1 and line A2 - A3.

	Tolerance	Sensitivity at H1 - H2
Dimensions	Parallelism with A0 - A1	0.424
Dimensions	Position tolerance on A5	1.000
Dimensions	Dimensional tolerance on A6 (i)	0.707
	Dimensional tolerance on A6 (ii)	0.707
Hole H1	Radial tolerance	1.000
Hole H2	Radial tolerance	1.000

	Tolerance	Sensitivity at H1 - line A2 - A3
line A1 - A2	Parallelism with D0 - D1	1.974
line A3 - A4	Parallelism with A0 - A1	0.250
line A0 - A1	Angularity with D0 - D1	0.097
line A2 - A3	Angularity with A1 - A2	0.832
point A2	Symmetry with A1 and A4	0.500
point A5	Positional with dimensions	1.000
hole H1	Radial tolerance	1.000

If the three sets of sensitivity coefficients are for the influence of tolerances on three critical function dimensions, then they may be



Location of B0

Location of B1

Location of A1

Location of B2
(A2, A3, B3 similar).

Fig E10. Description of Solid Form.

used as input for an optimisation program. The second set was computed for the distance between H1 and a point R1 which was located on line A2 - A3 by a sub-network. The coordinates of R1 were calculated for one computation and measured from a drawing for another run and there was little difference between the coefficients obtained.

Displaying the network in the 'tree' form used previously is not very convenient nor easily read. The form illustrated in Figs. E.2 - E.9 is an improvement but it is important that for clarity, the nodes should be in topological order.

(c) Three dimensional systems.

This example is deliberately very detailed. In practice much of the resulting network (where features of no interest are concerned) may be omitted. The solid form shown in Fig. E.10 will be described.

A plane (the X-Y plane), a line on it (the line OY) and a point on this line (the origin), will be taken as the datum system. Points A0 and A1 are positioned relative to this system. Point A0 is fixed at the origin, but A1 may be displaced along the line OY.

The line A0 - B0 is located on line A0 - A1 with an applied tolerance of angularity relative to A0 - A1 acting at point B0a.

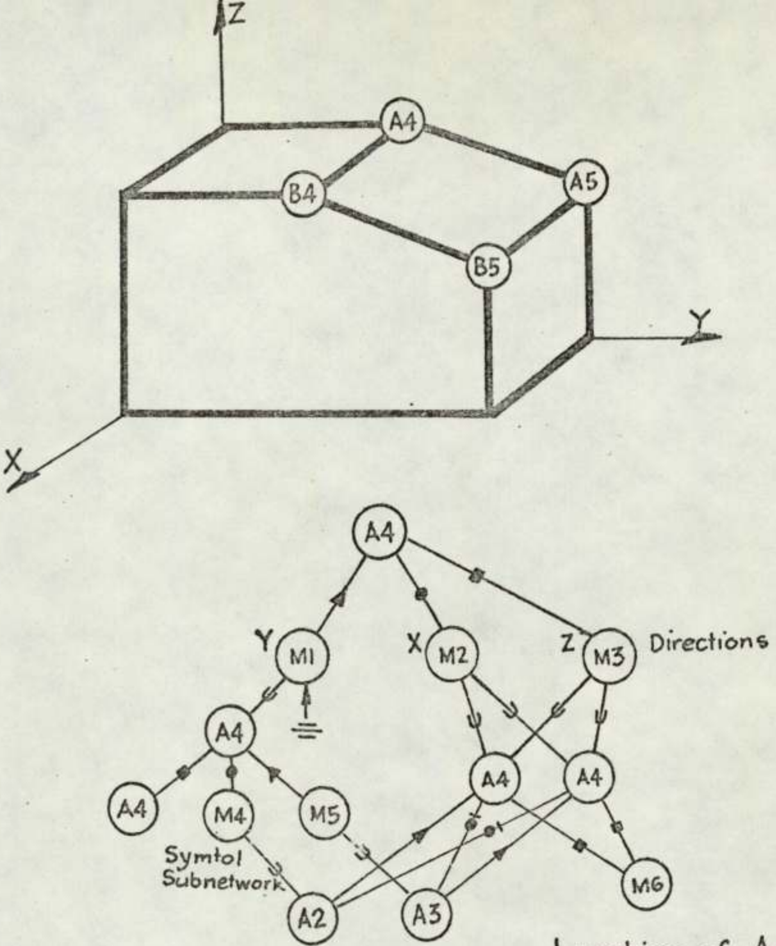
The line B0 - B1 is located on line A0 - A1 with applied tolerances of parallelism acting on B0b and B1a. B0a and B0b displacements are compounded at B0.

The line B1 - A1 is located on line A0 - B0 with applied tolerances of parallelism acting at B1b and A1. The two tolerances at B1a and B1b are compounded to form the intersection point B1.

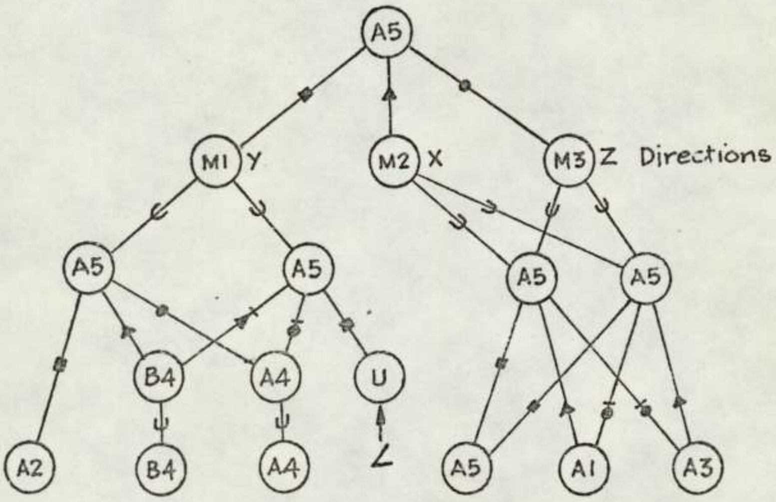
In addition, flatness tolerances may also be imposed at B0 and B1. These are not shown in the figure.

This completes the description of the plane A0 - B1.

Lines A0 - A2, A1 - A3, B1 - B3 and B0 - B2 are defined by locating



Location of A4
(B4 Similar)



Location of A5
(B5 similar).

Fig E 11. Description of Plane A4-B5.

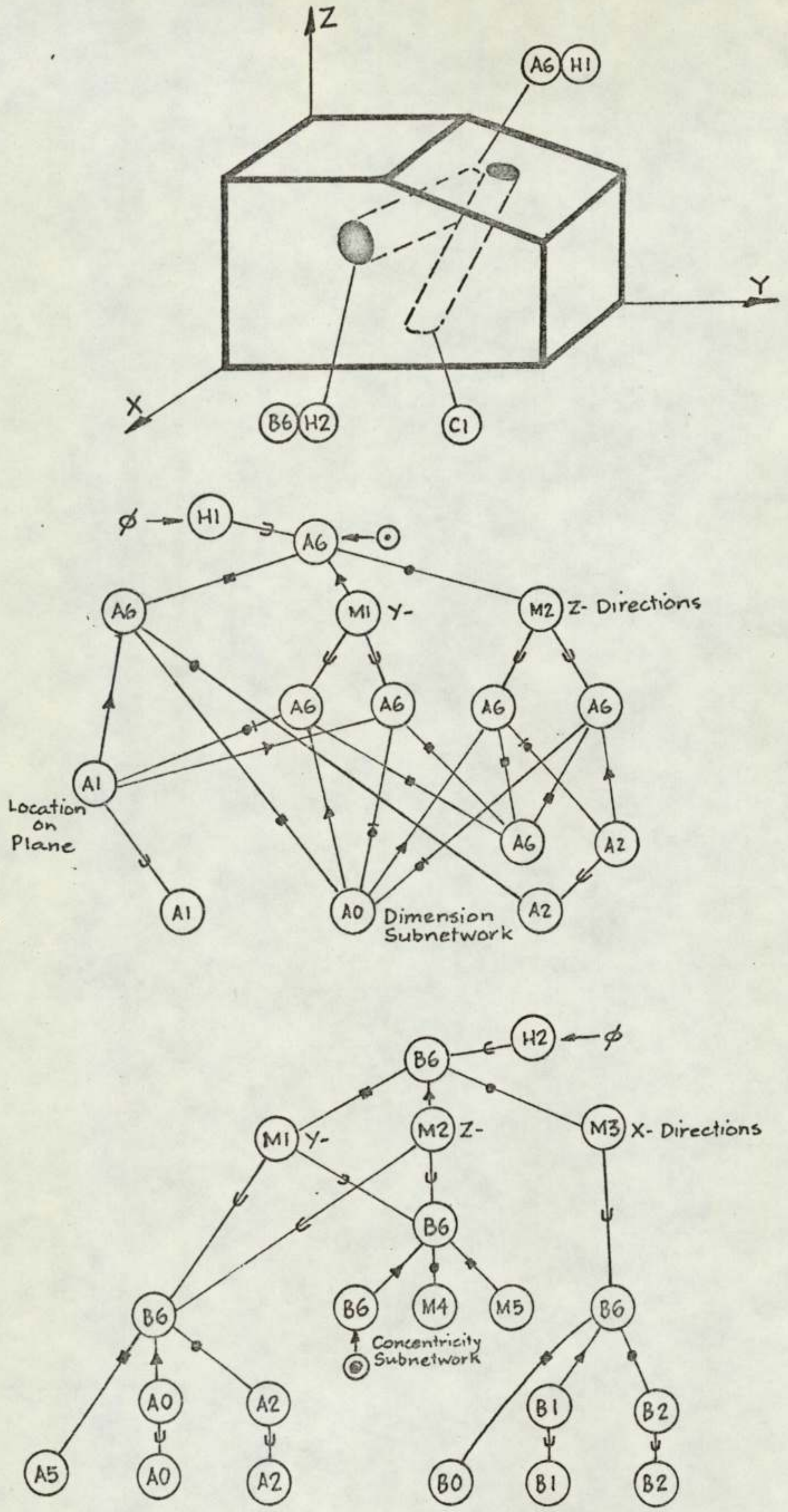
the points A2, A3, B3 and B2 on the plane A0 - B1. Each of these points may have three displacements; one parallel to the XY plane due to the locating point, a squareness tolerance relative to the XY plane and a parallelism tolerance relative to the XY plane. These three displacements, the first extrinsic, the others intrinsic must be compounded in the three orthogonal directions to give the intersection points. Since the displacements are also orthogonal, they may be simply superposed.

This completes the description of the solid form.

Points A4 - B4 are now to be added. A4 is symmetrically toleranced relative to A2 and A3, B4 relative to B2 and B3. The displacements due to this tolerance will be in the OY direction. There will also be extrinsic displacements on these points due to their situation on the lines A2 - A3 and B2 - B3, in the OX and OZ directions. These may also be superposed to give the true displacements at A4 and B4.

The cutting plane A4 - B5 is located on line A4 - B4 with an angularity tolerance relative to plane A2 - B4. It is sufficient to consider this plane as defined by the points A2, A4, B4 for the purpose of applying the angularity tolerance to points A5a and B5a. In addition, since A5 and B5 are necessarily constrained to lie on lines A1 - A3 and B1 - B3, dual points A5b and B5b are located on these lines and the consequent displacements compounded with those at A5a and B5a to give the actual positions of A5 and B5. Fig. E.11 illustrates the procedure.

A hole is to be drilled at an angle to face A1 - A2. The centre at point A6 is located on plane A1 - A2 and is dimensioned from datum lines A0 - A2 and A0 - A1 in the same way as in the two-dimensional case, but the displacement due to its position on the plane is applied to the P-loc. The point on face B1 - B2 where the axis of the hole runs out (point B6) will be subject to an angular tolerance relative to plane A1 - A2 and will also be constrained to lie on plane B1 - B2. These



The hole at C1 is similarly described.

Fig E12. Description of Holes.

displacements will be compounded to give the true run-outpoint B6. The hole H1 is described by setting H-locs along A6-B6, tolerances at A6 and B6 being passed directly by means of unitary links.

A further hole H2 is drilled normal to face A4 - B5. This is described in the same way as H1, except that there need be no complication at the run-out point on face A0 - B1, this being a datum. The network is shown in Fig. E.12.

It is now possible to obtain sensitivity coefficients for the distance between holes H1 and H2. Although this has been a very detailed analysis of the form, much of it being unnecessary for obtaining these results, all the working has made use of a few standard networks and may be done reasonably quickly with a little practice. If it were known in advance that only the sensitivity coefficients between H1 and H2 were required, face B1 - B2 could be ignored and this has been done using seventy nodes only.

APPENDIX F

SUMMARY OF REFERENCES

F.1 Summary of References

Comparatively little work has been published since 1960 on the subject of engineering tolerances. The Secretary of the Institution of Engineering Designers has suggested to the author, in private communication, that this might be because this topic is very much tied up with a company's profitability. About fifty papers, articles and books have been published during the period 1960-1976. Historically, papers published prior to 1960 are concerned with good drawing practice; from 1960-1970, they are concerned with statistical implications and from 1970 onwards, they are mainly about the allocation problem.

(i) Practical treatments dealing with 'good practice'.

A few detailed manuals of dimensioning practice have been written. Possibly the best is ref. T7. This contains details ofall the common dimensioning systems and was written as a companion volume to BS 308 (ref. G5) which was the first wholehearted attempt to systematise design practice in this country. A further, extended version of BS 308 was published in 1972 and this is widely regarded as the standard for drawing practice. Ref. T7 contains much that is relevant to the latest version of BS 308, but has not been re-printed by the publisher.

There are several excellent papers and articles which may be found in the list of references. Refs. T1, T2, T13 may be singled out as being particularly useful.

(ii) Analysis of Statistical Tolerancing

The principle of infallible interchangeability (sometimes called sure-fit) assumes that all the critical dimensions of a manufactured part are at an extreme limit of the allowed tolerance range. It has long been recognised that this is normally an unlikely eventuality and that this pessimistic approach often results in parts which are specified with unnecessary precision. This causes high unit manufacturing costs and excessive rejection rates. A common assumption is that the bulk of the tolerances involved in an assembly will be distributed in Gaussian fashion and in cases where this is not so, the overall 'stacked' tolerance will be approximately Gaussian, as proved in the Central Limit Theorem. Thus, the majority of assemblies will have critical clearances which are reasonably close to the nominal. An excellent introduction to this treatment of statistical tolerancing may be found in ref. S3, and a more advanced description in ref. S8.

In practice, tolerances are not quite so well behaved and often, for a variety of reasons, the distribution is not Gaussian. For instance, it is good practice to allow for tool wear by starting to remove metal at one end of the tolerance range and to drift towards the other limit as the tool wears. A batch of parts machined in this way may, then, all be close to extreme tolerance limit. Another example is in the drilling of a hole through a locating bush. The hole centre will very likely be at extreme tolerance.

A detailed discussion of these situations may be found in ref. T1.

It seems safer to assume a more general distribution than the Gaussian for the component tolerances of highly critical clearances. A description of a computer package for the statistical analysis of tolerances with general distributions may be found in ref. S5. The designer uses the system interactively with a graphic console. He provides the system with an expected statistical distribution for each tolerance, together with a sub-program (written in PL/1) describing the geometrical relationships between each dimension. The computer then generates representative critical dimensions for each component, sampling from the appropriate distributions. The number of simulations is typically of the order of 1,000 and results are displayed graphically in various forms. Tolerances may be adjusted interactively by the user.

The system has been widely used in the Body Division of General Motors and is regarded as a useful design tool.

The main drawback of the system is that the geometric form of the dimensioning system must be specified by a sub-program either written by the user or submitted by him to a specialist programmer. This seems inconvenient at best and certainly unsatisfactory in a firm smaller than General Motors. Another problem is that it does not seem possible, as far as can be judged, to specify complex multi-stage machining processes in which dimensions at a stage depends on those obtained at a previous stage. The system, however, is the first to use a computer for this purpose and will, no doubt, be progressively refined as further experience is gained in its use. It would, clearly, be an improvement if the dimensioning and datum system were submitted to the program not as a sub-program but as data.

(iii) Allocation of Tolerances

Although the problem of allocating tolerances to the component dimensions of a critical clearance has been recognised for many years, the first paper to be published in this field was ref. A4. The problem is clearly defined and solved in ref. A2. This is an elegant account but uses an inverse square law for the tolerance cost which has been superseded in later papers by more realistic models. Possibly the most important paper is ref. A8. This uses a negative exponential model (now commonly called 'the Speckhart model') for the tolerance cost function. The method of Lagrange multipliers is used to minimise the total cost and various practical examples are analysed. The author has developed a program to calculate optimum tolerance allocations on both sure-fit and statistical-fit bases.

Another approach to the minimisation of tolerance cost is to use dynamic programming. An account may be found in ref. A5.

The most detailed and comprehensive description to date is a two-part paper ref. A1. Earlier papers on the subject are reviewed, cost-tolerance data obtained from various sources analysed and practical models derived for different manufacturing processes. A mathematical analysis is performed for sure-fit cases and the method is applied to several practical examples. A further paper dealing with statistical-fit cases is promised.

An account of the problem and method of solution will be found in Appendix C. This is largely eclectic drawing mainly on refs. A1 and A8 and is included for reference.

It is interesting to note that ref. A1 pre-dates ref. A8, is a fuller treatment and the cost equations are based on a thorough investigation of practical results and yet the latter paper seems to be considered definitive in the literature.

(iv) Geometric Calculation of Tolerances

This topic in tolerancing theory has not been dealt with systematically. Various papers have been written on specific problems (refs. C1 and C2) but the examples quoted are either trivial or too specialised to be of much general interest. It is hoped that this paper might fulfil a need in this respect.

REFERENCES

For convenience, references are separated into 5 classifications, each being given a distinct prefix.

G . . .	General references which are mainly textbooks used in the theoretical development.
T . . .	These refer to general principles of tolerancing and dimensions.
S . . .	Deal with statistical considerations.
A . . .	Concerned with the problem of allocation of tolerances.
C . . .	Deal with the calculation of tolerations geometrically.

GENERAL REFERENCES

G.1 Berztiss, A T Data Structures Pub. A P

G.2 Bickley, W G and Thompson, R S H Matrices Pub. E U P

G.3 Boullion and Odell Generalised Inverse Matrices Pub. Wiley

G.4 Box, M J, Davies, D and Swann, W H Non-linear Optimisation Techniques ICI Monograph No. 5 Pub. Oliver & Boyd

G.5 British Standards Institution BS 308 Engineering Drawing Practice

Part I General Principles

Part II Dimensioning and Tolerancing of Size

Part III Geometrical Tolerancing

G.6 Chace, M A Analysis of 3-D Mechanisms PhD Thesis Univ. of Michigan 1964

G.7 Ferrar, W L Finite Matrices Pub. Clarendon Press

G.8 Fox, L An Introduction to Numerical Linear Algebra Pub. O U P

G.9 Fox, L and Mayers, D F Computing Methods for Scientists and Engineers Pub. O U P

G.10 Gerrish, F Construction of Defective 3 x 3 Matrices having prescribed Eigenvalues and Eigenvectors Note 279, Mathematical Gazette, December 1972

G.11 Hamburger, and Grimshaw, Linear Transformations Pub. C U P

G.12 Kempster, M H A Principles of Jig and Tool Design Pub. E U P

G.13 Knuth, D E The Art of Programming Vol I Pub. Addison-Wesley

G.14 Shigley, J E Kinematic Mechanisms Pub. McGraw-Hill

G.15 Smyth, W F and Radeceanu, E A Storage Scheme for Hierarchic Structures Computer Journal Vol 17 No. 2

G.16 Yefimov, N V Quadratic Forms and Matrices Pub. Academic Paperbacks

TOLERANCING PRACTICE

T.1 Abbott, W The Dimensioning of Engineering Drawings Pub. Blackie

T.2 Burr, I W Simplify Dimensioning and Tolerancing to Cut Cost Design Engineer V 19 No. 2 1973 pp. 34-41

T.3 Conway, H G Engineering Tolerances Pub. Pitmans

T.4 Hatton, W J Design Tolerance: Relationship to Planning Work Study V 25 1976 pp. 24-26

T.5 Holland, T Ten Ways to Design Expensive Tolerances Engineering V 215 1975 pp. 642-644

T.6 Oddie, D A Geometric Tolerancing of Engineering Drawings Quality Engineer V 34 1970 pp. 12-13

T.7 Parker, S Drawings and Dimensions Pub. Pitmans

T.8 Spencer, J and Cheney, R L Comprehension of Numerical Information in Engineering Drawings Engineering Designer Aug 1971 pp. 5-9

T.9 Stewart, G Tolerances for Production Australasian Engineer Mar 1968 pp. 42-53

T.10 Tallack, W J Some Effects of Tolerances in Measurement Quality Engineer V 30 1966 pp. 172-173

T.11 Tarasevich, Y and Yavoich, E Fits, Tolerances and Engineering Measurement Pub. Foreign Languages Publishing House, Moscow

T.12 Wakefield, L P Dimensioning Hole Centres Engineering Designer
Part I Nov 1962 pp. 4-6
Part II Jan 1963 p. 7

T.13 Wakefield, L P Dimensions, Tolerances: Implications
Engineering Designer Part I July 1970 pp. 5-11
Part II Aug 1970 pp. 9-14

STATISTICAL ASPECTS

S.1 Burr, I W and James, R E Specifying the Desired Distribution in Lieu of Tolerance Limits A S M E paper n 74-DE-9 1974

S.2 Burrows, G L Statistical Tolerance Limits - What are they? Applied Statistics V 12 June 1964 pp. 133-144

S.3 Donaldson, C S Fit and Tolerance Engineering Materials and Design V 11 1968 pp. 923-929

S.4 Gilson, J A New Approach to Engineering Tolerances Pub. The Machinery Publishing Co. Ltd.

S.5 Gugel, H W Monte Carlo Simulation with Interactive Graphics General Motors Corporation, Research Publication GMR-1531-1974

S.6 Mansoor, E M Application of Tolerances used in Engineering Designs Inst. of Mech. Eng. Proc. 178 Pt. 1 No. 1 1963-64 pp. 29-51

S.7 Mansoor, E M Selective Assembly - its Analysis and Applications Int. J. Prod. Res. V 1 No. 1 1961 pp. 13-24

S.8 Spotts, M F An Application of Statistics to the Dimensioning of Machine Parts A S M E Jour. of Engr. Ind. 1959 pp. 317-322

S.9 Spotts, M F Dimensioning of Clearance Fits with Overlapping Tolerances using Probability Theory A S M E Paper n 74-DE-8 1974

ALLOCATION OF TOLERANCES

A.1 Bennett, G and Gupta, L C Least Cost Tolerances Int. J. Prod. Res. Part I Vol 8 No. 1 1970 pp. 65-74
Part II Vol 8 No. 2 1970 pp. 169-181

A.2 Hillier, M J Systematic Approach to the Cost Optimisation of Tolerances in Complex Assemblies Bull. of Mech. Eng. Education Vol 5 Apr 1966 pp. 157-161

A.3 Hindmarsh, G W Manufacturing Tolerance Allocation Prod. Engr. Vol 20 Jul 1973 pp. 257-263

A.4 Latta, L W Assignment of Least Cost Tolerances Engr. Digest Vol 25 Jan 1964 pp. 86-87

A.5 Moy, W A Assignment of Tolerances by Dynamic Programming Machine Design Vol 36 No. 12 1964 pp. 215-218

A.6 Peters, J Tolerancing the Components of an Assembly for Minimum Cost A S M E paper 70-Prod-9 1970

A.7 Smathers, E W and Ostwald, P E Optimisation of Component Functional Dimensions and Tolerances A S M E paper 72-DE-18 1972

A.8 Speckhart, F H Calculation of Tolerance Based on a Minimum Cost Approach A S M E paper VIBR-114 1971

A.9 Spotts, M F Allocation of Tolerances to Minimise Cost of Assembly A S M E paper 72-WA/DE-6 1972

A.10 Wilde, D Simplifying Discrete Tolerance Assignment A S M E paper 75-DET-106 1975

A.11 Wilde, D and Prentice, E Minimum Exponential Cost Allocation of Sure-fit Tolerances J. Eng. Ind. Trans. A S M E Vol 97 Ser B No. 4 1975 pp. 1395-1398

GEOMETRIC CALCULATION

C.1 Dacosse, J Note sur la théorie des écarts et tolérance de fabrication Méthode de détermination des tolérances complexes Revue de la Mecanique Vol 5 No. 4 1959 pp. 167-179

C.2 Knappe, L F A Technique for Analysis of Mechanical Tolerances Machine Design Vol 35 No. 10 Apr 1963 pp. 155-157

COMPUTER-AIDED ANALYSIS OF ENGINEERING TOLERANCES

PETER CHARLES INGHAM

Thesis submitted for the degree of Ph.D.

University of Aston in Birmingham

1977

COMPUTER-AIDED ANALYSIS OF ENGINEERING TOLERANCES

PETER CHARLES INGHAM

Thesis submitted for the degree of Ph.D.

1977

SUMMARY

In the design of mass-produced components, it is essential that manufacturing tolerances should be analysed to make sure that assemblies fit together satisfactorily and that parts are not produced to unnecessarily tight specifications. The analysis may be divided into three stages:

(a) calculation of the sensitivity of a feature's position to the magnitude of the tolerances upon which it depends,

(b) ensuring that the permitted tolerances which together influence a critical measurement are allocated in the most economical way, and

(c) analysing the statistical distribution of tolerances on critical measurements.

This thesis describes a method of performing stage (a). Stages (b) and (c) have been dealt with elsewhere.

It is demonstrated that the analysis of tolerances in all but the most straightforward cases is not a trivial operation and a model is developed to assist with the calculation. This is a location element derived from the classical six-point system for locating a body in three dimensions. Elements may be combined to describe multi-datum machining operations, assemblies and drawing dimension systems by a tree-like structure. The model is analysed mathematically, a compendium of commonly-occurring cases is appended and algorithms for obtaining results of interest to the engineer designer are described. A computer program for the sensitivity analysis is also described and the integration of the method into a full tolerance-analysis system is discussed.

KEYWORDS:

TOLERANCES, DESIGN, ENGINEERING, COMPUTERS

ACKNOWLEDGEMENTS

I wish to record my appreciation of the guidance and encouragement that I have received from my supervisor, Mr. I. H. Gould. I am also grateful to Dr. L. J. Hazlewood for the advice that he gave me on the eigenvalue problem and to Mrs. J. McIndoe who typed the manuscript.

An elementary two-dimensional version of the system was designed while I was working in the Design Office of Girling Ltd., Tyseley, and I should like to thank Mr. W. Harrison who supervised my earlier efforts.

INDEX

1. GENERAL DISCUSSION

1. General Discussion

1.1 The Principle of Infallible Interchangeability

Many of the aims in the design of mass-produced components are unattainable. Examples are zero cost, zero weight, infinite strength and ultimate aesthetic appeal. However, a major aim which can often be achieved is infallible interchangeability. This term, probably first used in ref. T.7 means that a component selected at random from a batch of like components should fit satisfactorily to any one of a batch of mating components. In some situations, selective assembly may be a suitable process but usually it is precluded since it is costly in resources of labour and time. Normally, batches of components must be assembled unselectively.

The principle of infallible interchangeability leads to some difficulties in practice. All manufacturing processes are subject to size variation in a degree depending on the particular process; two components, even when produced on the same machine, will be unlikely to be of the same size. Exact fit is, therefore, another unattainable aim of mass production design and design clearances must make allowance for process error.

1.2 Types of Working Drawings

Unfortunately, modern design is a specialised function and modern designers are not expected to be experts in jig and tool design, in metrology nor in any other of the branches of production engineering. The task of the designer is to specify the functional requirements of the finished part, and, although he will usually have some knowledge of the manufacturing and inspection processes involved, he will not normally lay down a rigorous specification for them. This principle is clearly stated in BS 308: Part 2: 1972:

"Production processes or inspection methods should not be specified unless they are essential to ensure satisfactory functioning or interchangeability."

It is also discussed at length in ref. T.7.

There are two types of working drawing. These are -

(a) product drawings completely defining the finished product as required by the designer, and

(b) process drawings defining products in a partly finished state suitably dimensioned for the manufacturing process to be adopted.

A view sometimes expressed is that the product drawing is not the definition of a machined part, but the definition of a gauging method for a machined part. Although this is contrary to the spirit of BS 308 since gauging may be considered to be a manufacturing process, it is partially true.

A part may be therefore dimensioned in three ways:

(a) for its function, so that it may work satisfactorily,

(b) for a process, so that it may be made, and

(c) for inspection, so that sizes may be checked.

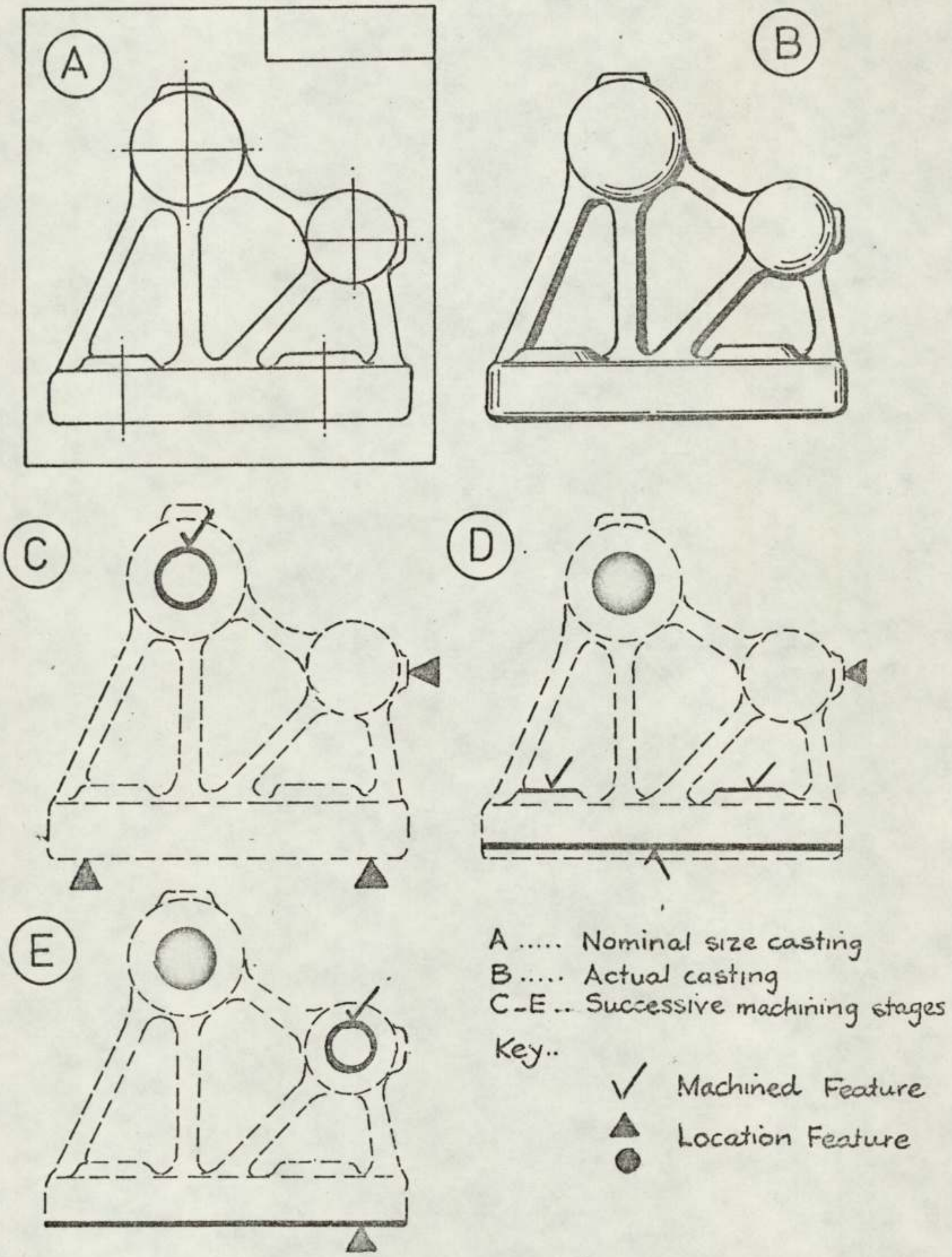
Each of these may involve different dimension systems for the same part and it is essential that tolerances arising from (b) and (c) be not greater than those specified in (a). In many cases, this may be checked using simple arithmetic (and some simplifying assumptions, usually) but often it is no trivial process.

1.3 Further Problems in Dimensioning

A machining process will affect some functional clearance, possibly in an indirect way; and often a functional clearance is affected by more than one process. Usually, the production engineer has several possible machining processes available, each having its own accuracy. As a rule, the more accurate a machining process, the higher the unit cost and it is desirable that the more accurate processes are used in features which have the greatest effect on the functional clearance.

Another difficulty is that many common dimensioning systems, even some described in BS 308, are ambiguous and their interpretation depends on some convention. In the majority of cases, alternative interpretations lead to differences in functional clearances which are small. However, this may not always be so and an unequivocal method of describing dimension systems is desirable.

2. BASIC CONCEPTS



A ..... Nominal size casting
B ..... Actual casting
C-E .. Successive machining stages

Key..

✓ Machined Feature
▲ Location Feature
●

Fig 1 Component Chains

2. Basic Concepts

2.1 Component chains - A model for complex manufacturing processes and assemblies

A feature on a component may be displaced from its nominal position for two reasons. Firstly, the component may be machined in several stages, each stage having its own, possibly distinct, datum systems. Secondly, the component may be assembled on other components each having its own variations in size. Both of these cases can be treated in the same way since they are conceptually identical.

Each stage in the machining of a component may be regarded as a separate physical component, the stages being assembled together to make the finished part. An example is shown in Fig. 1. The reference body is the nominal size drawing of the casting, the actual casting being located on it by the dimensions on the casting drawing. Each subsequent machining stage is located either on the actual cast form, or on previous machining stages, or on both. A machined part may then be treated in the same way as an assembly.

The classical method of location consists of clamping a body to a plane, to a line and to a point, so that six degrees of freedom are removed. This applies both to the physical assembly of components and to a machined feature on a component since six point location is considered to be good jig and tool practice (for example, see ref. G.12, p. 77). Some common dimensioning systems cannot be described in six point location form and these will be discussed at a later stage.

The example shown in Fig. 1 is described diagrammatically in Fig. 2. Each box represents a machining stage and each arrow represents the relationship 'depends on the size of'. Thus variations in size will be passed on, through a chain, to the finished part.

Finished components may also be fitted together to form an assembly. Assemblies may also be described by diagrams similar to Fig. 2. Each box

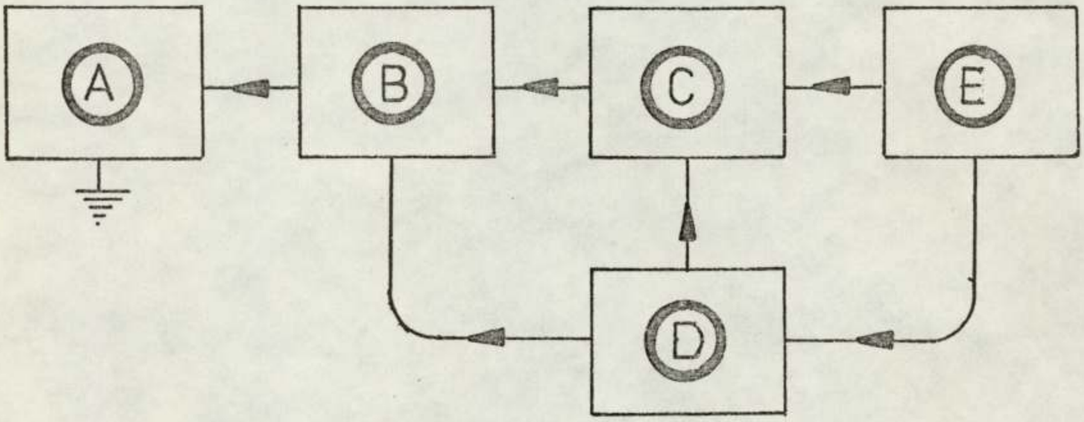


Fig 2 Component Chains.

now represents a finished part, the meaning of the arrows being the same as for machining stages. Multi-stage machining processes and assemblies may, then, be considered in the same way, and, to avoid confusion, they will subsequently be called 'assemblages'.

2.2 Problems to be solved

Assemblages of both kinds may be linked together in the same way. At some stage, a useful limit will be reached, and questions that one might wish to ask are -

(a) in Fig. 2, what is the effect at a feature in stage C of given variations in size at stages A and B,

(b) in Fig. 2, if the maximum permitted variation in position or size in stage C be known, then how should the tolerances be apportioned between stages A and B so as to minimise the process cost, and

(c) what is the clearance between a feature on component D and one on component E in Fig. 2?

The concept of regarding a finished part as an assembly with some of the components possibly occupying the same space as others is fundamental to the system to be described. Its use enables assemblages to be defined in a unified way and questions (a), (b) and (c) may be answered in much the same fashion.

The terms 'tolerance' and 'location' have been used so far in a fairly loose, commonsense way since they are of common currency in engineering. However, as they will be used subsequently in a more specialised sense, some discussion of them follows.

2.3 Tolerances

2.3.1 Definitions

Tolerance is the variation from nominal position of a feature of interest on a component.

Tolerances may be specified bilaterally, the locating dimension consisting of a mean size with a tolerance equally disposed about it;

or unilaterally, the locating dimension consisting of a size at one extreme with a tolerance quoted in the opposite direction. In all the examples which follow tolerances will be specified bilaterally.

A tolerance zone is the zone within which the feature of interest is required to be contained. BS 308 specifies that a tolerance zone is one of the following:

(1) a circle or cylinder

(2) the area between two parallel lines or two parallel straight lines

(3) the space between two parallel surfaces or two parallel planes

(4) the space in a parallelepiped.

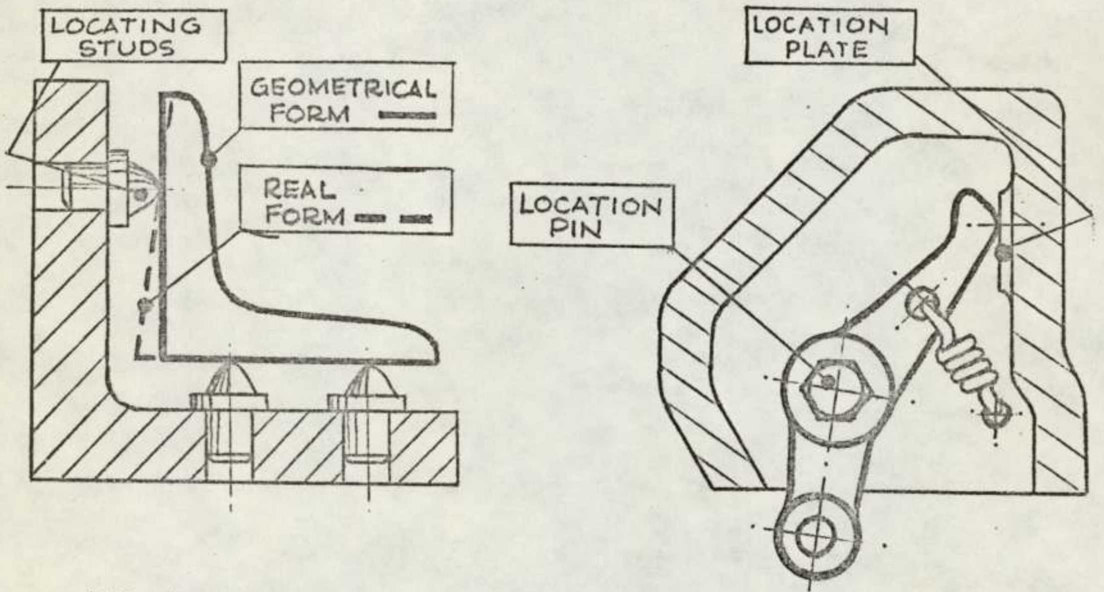
A tolerance zone which is occasionally useful, but which is absent from the list, is a sphere.

2.3.2 Intrinsic and extrinsic tolerances

A feature may be displaced from its nominal position for two reasons. Firstly, there will be a tolerance on its position due to error in the manufacturing stage which has produced it. This will be called intrinsic tolerance, and will be the tolerance quoted on the process drawing of the feature. In following examples, standard BS 308 tolerance frames will be used to show intrinsic tolerance. Secondly, there will be a tolerance on the position of the feature resulting from tolerances on previous manufacturing stages on which its location depends, or tolerances on the finished parts on which it is assembled. This will be called extrinsic tolerance and must be calculated from the intrinsic tolerances on the locating parts. The sum of intrinsic and extrinsic tolerance will be termed 'total tolerance'. In all cases, tolerance is relative to some frame of reference. Intrinsic tolerance is relative to the nominal positions of the locating features, while extrinsic and total tolerances are relative to any feature of interest.

2.4 Locations

There are two types of location system which are described below using two-dimensional examples for illustrative purposes.



JIG LOCATION

LOCATION OF LEVER ASSEMBLY

FIG 3A REAL LOCATIONS

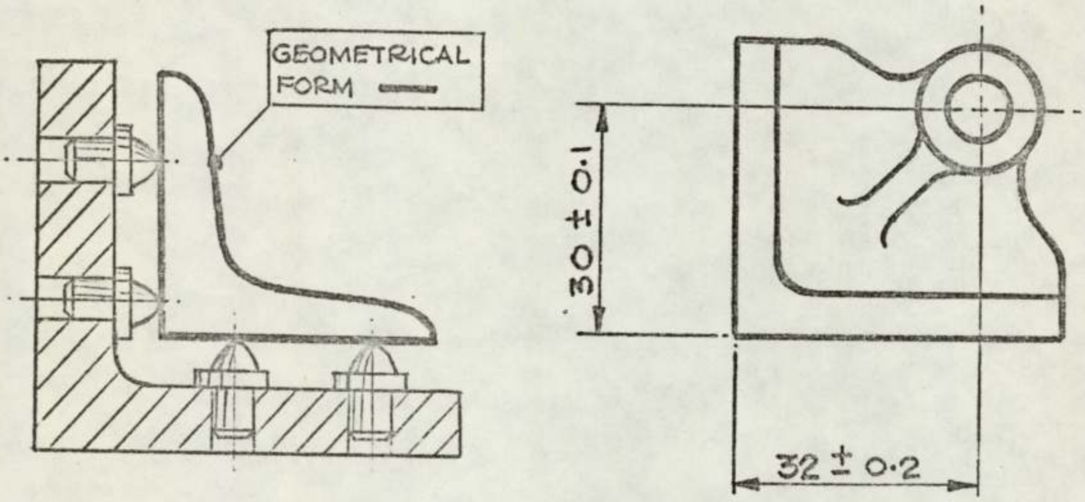


FIG 3B GEOMETRICAL LOCATIONS

Fig 3 Location Systems

2.4.1 Real Locations

These are location systems which may be realised physically and locate real bodies on real locating features, both necessarily having irregularities in form and displacements from nominal positions. Real locations occur in the jigging of manufacturing processes and in the assembly of components. Examples of each are shown in Fig. 3A.

2.4.2 Geometrical Locations

Functional drawings often describe location systems which may not be realised physically since they refer to idealised geometrically exact figures. A common example is shown in Fig. 3B. As may be seen, the drilled hole cannot be located physically on two datum faces. The most plausible interpretation which can be made of this system is that any convenient jigging system (which necessarily involves a real location) is to be used but that on the finished part, the hole is to lie within the parallelepipedal tolerance zone defined on the drawing and centred at the intersection point of two lines parallel with the datum face at a distance from them specified by the drawing dimensions.

A detailed discussion of this dimension system will be found in ref. T.7 pp. 266-270.

2.5 Cumulative Tolerance

The concepts described in the previous sections are illustrated by the example shown in Fig. 4. This is unrealistic but not wildly so.

Faces E and F are assumed to be geometrically exact, while face D is subject to a profile tolerance and lies within a band as shown in (1) on the diagram. It is further assumed that all locating points are exact, but the central axis of the machine tool is subject to a circular tolerance zone relative to the corresponding locations. Form irregularities normal to the plane of the diagram are discounted.

The three holes are machined using separate locating systems as

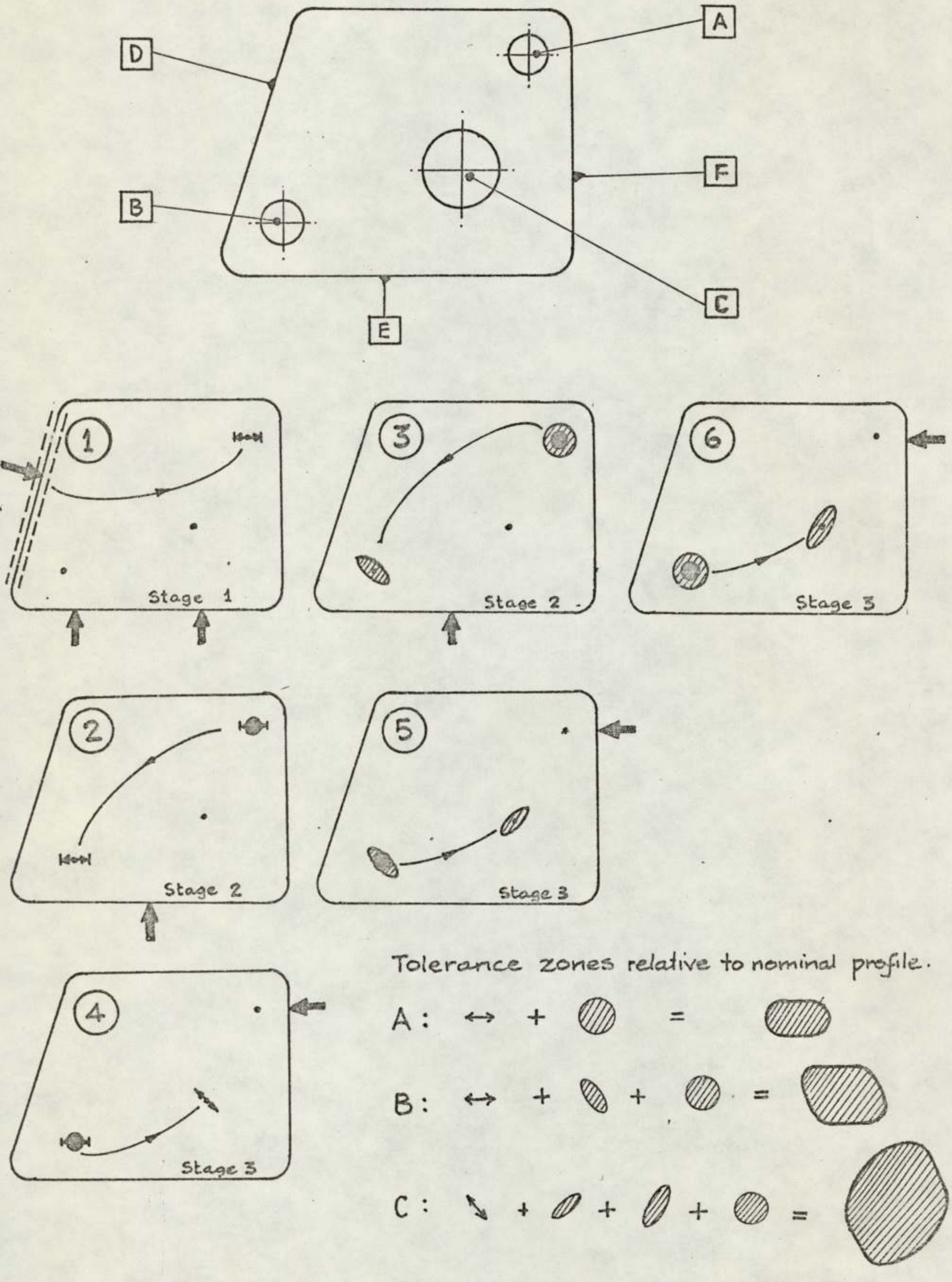


Fig 4 Cumulative Tolerances.

shown below:

Stage 1: Hole A ... datum Face E, and a point on Face D.

Stage 2: Hole B ... datum Hole A, and a point on Face E.

Stage 3: Hole C ... datum Hole B, and a point on Face F.

Stage 1. The linear tolerance at face D will result in the position of hole A relative to all other faces of the plate being subject to a linear tolerance - as in (1). There will also be a circular tolerance zone at A relative to the locating features. This will also be relative to the actual profile of face D and all other nominal faces.

The total tolerance at A relative to all nominal faces will be these two tolerance zones superimposed.

Stage 2. The linear tolerance at A results in a linear tolerance at B (shown in (2)); the circular positional tolerance zone at A (plus clearance between hole A and the locating peg) causes a tolerance zone at B which is approximately elliptical (shown in (3)). There will also be a circular positional tolerance at B, and the total tolerance zone at this hole relative to the nominal profile will be the superimposition of the linear, elliptical and circular tolerance zones.

Stage 3. There are four tolerance zones at C: a linear zone due to linear tolerance at B (shown in (4)), an elliptical zone due to the elliptical zone at B (shown in (5)), an elliptical zone due to the circular zone and clearance at B, and a circular positional tolerance. The total tolerance zone is the superimposition of the four.

All tolerance zones shown in the diagram are grossly exaggerated, and have been obtained by tracing the tolerance loci; but the effect of cumulative tolerance is clearly shown. In a real system, extra complication would be added because of factors which have been conveniently ignored in this model. For instance, the locating points would not be exactly positioned; and face F would be subject to a form tolerance.

Difficulties involved in analysing a multistage system are:

(a) The extreme position of a tolerance zone at a feature is not necessarily the position corresponding to an extreme position of zones at the locating features.

(b) Some tolerance zones are dependent - an example being the two zones shown in (3). The nett displacement between holes B and A does not depend on these zones. Deciding on which displacements at one feature are relative to another can involve much book-keeping and possible error.

(c) Real parts exist in three dimensions and though many tolerance situations may be considered as being two-dimensional, this is not always so. In a three-dimensional system, tolerance zones may be parallelepipedal or ellipsoidal and their effects are difficult to visualise, let alone calculate.

3. DESCRIPTION OF THE LOCATION MODEL

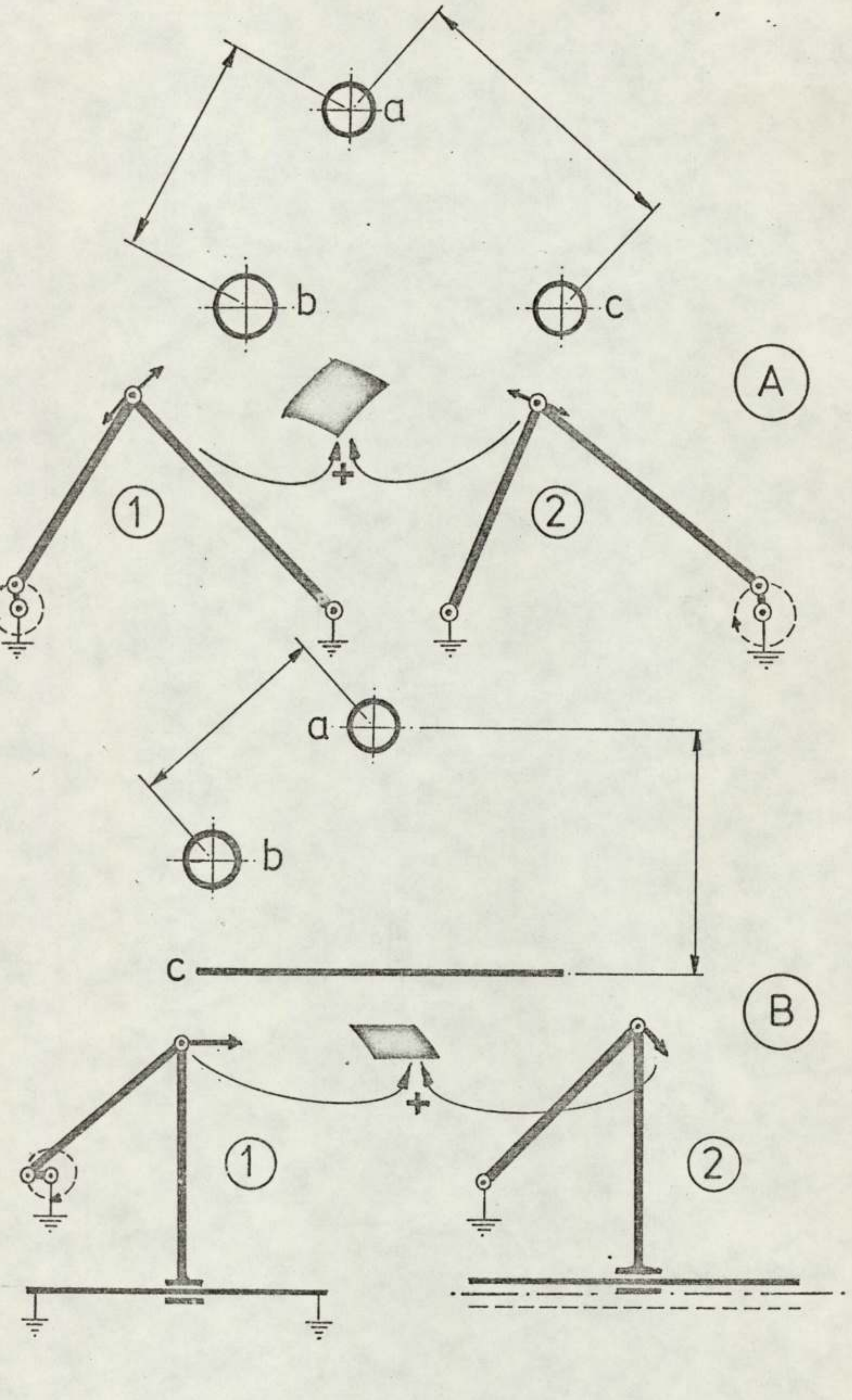


Fig 5 Equivalent Mechanisms

3. Description of the Location Model

3.1 Tolerance Mechanisms

Fig. 5A shows a common dimensioning system. Hole a is located by the centre points of holes b and c. The dimensioning system is analogous to a structure - there are no redundancies in the dimensions and if any dimension is deleted, then the remainder are insufficient to locate the hole. In this case, the three members of the structure are the centre distances of the holes.

Extrinsic tolerances will be passed to the located feature from each of the locations. If a circular tolerance be imposed at hole b, hole c being held at its nominal position, then the dimension system may be regarded as a mechanism, each position of hole b corresponding with an unique position of hole a. The mechanism in the case illustrated is a four bar chain and the locus of hole a is a short circular arc. If now hole b be held at nominal and a circular tolerance zone applied to hole c, then a similar mechanism is obtained - Figures 5 A-1 and A-2. Since the radius of each tolerance zone is small in comparison with the locating dimensions, the two systems may be superimposed to give a total tolerance zone as shown in the figure. This approximates to a parallelogram as the lengths of arc are small.

Another dimensioning system is illustrated in Fig. 5B. In this case, hole a is located by its distance from hole b and by its perpendicular distance from line c. Again the system is exactly determined and may be considered as a structure. If a circular tolerance zone be applied at hole b then an equivalent mechanism may be derived. In this case, since the perpendicular distance from line c is specified, a sliding member is necessary on the line, while the crank centred at the nominal position of hole b generates the circumference of the tolerance zone. At hole a, the zone generated is a short straight line parallel with c.

Similarly, if a tolerance band is allowed at line c, hole b being

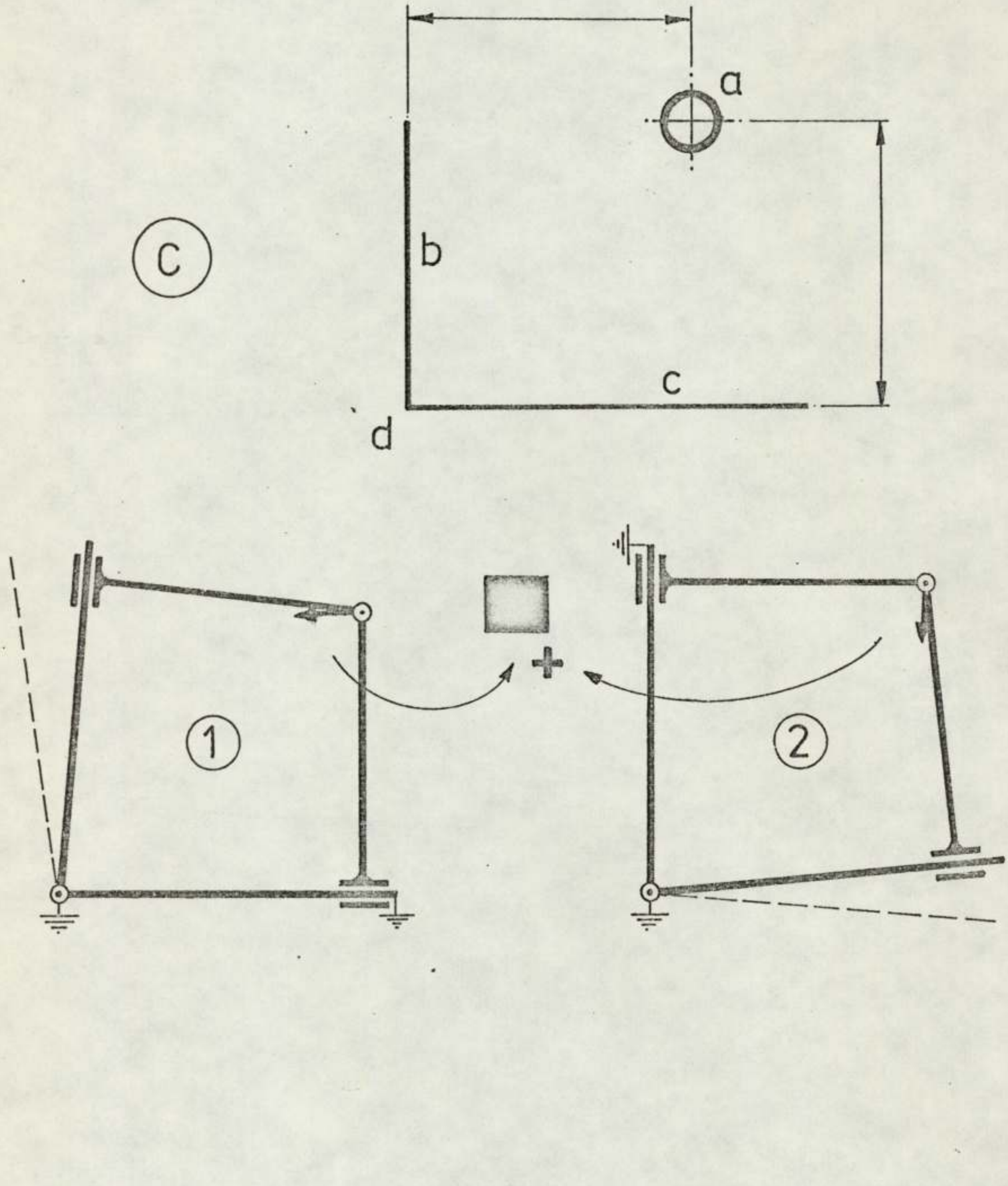


Fig 6 Equivalent Mechanisms.

held at its nominal position, the resulting zone at hole a will be a small circular arc centred at b.

The two tolerance zones may be superimposed and the resulting zone is approximately a parallelogram.

Fig. 6 shows a mechanism with two sliding pairs which is equivalent to a point dimensioned from two straight line datums.

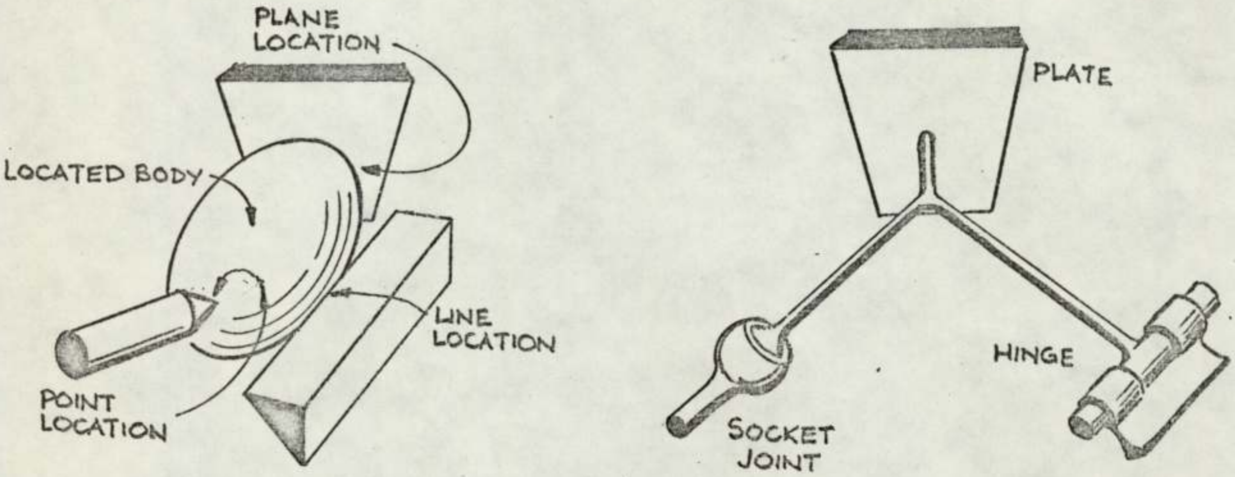
There is a general solution for three dimensional mechanisms comprised of pure turning and sliding pairs (refs. G.6 and G.14), and it seems feasible that a method for the analysis of tolerances based directly on the use of such elements could be derived. This direct approach suffers from some disadvantages, however. As has been demonstrated in section 2.5, explicit tolerance zones are of irregular shapes, and if, for example, hole b in Fig. 5A were located in the same way as hole B in Fig. 4, then the tolerance zone would certainly not be circular. Even if the zone were decomposed into its separate elements, it would be necessary to use a crank arm with a radius varying dynamically with turning angle, one of the elements being elliptical. Also, the method is rather inflexible, as each of the many possible dimension systems would require a separate equivalent mechanism with a separate method of calculation. Although the vector equations for these mechanisms may be written down using the methods of ref. G.6 they do not appear tractable for solution in some cases. These problems are exacerbated in three dimensions.

Equivalent mechanisms are useful in visualising the effects of explicit tolerances, but the preferred method will be based on a standard unit of location. Indeed, the method might be used, with a little modification, in the analysis of the general kinematic mechanism; but this is outside the scope of this thesis.

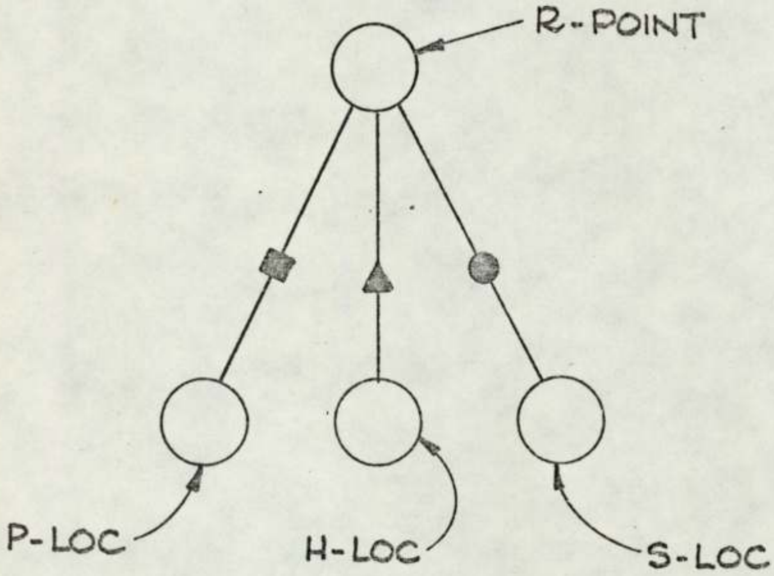
3.2 Elemental Location

A body is located elementally if -

(a) a point on it is held against a locating plane,



ELEMENTAL LOCATION MECHANICAL ANALOGY



A LOCATION TRIAD

Fig 7 Elemental Location

(b) a point on it is held against a locating line, and

(c) a point on it is held against a locating point.

The located body is sensitive to small displacements -

(a) along the normal to the locating plane,

(b) orthogonal to the locating line and

(c) in any direction at the locating point.

Elemental location is illustrated in Fig. 7.

In order to avoid confusion, a locating plane will be called a plate, a locating line will be called a hinge and a locating point a socket. The terms have been picked because of their obvious mechanical analogies and also because they have distinct initial letters. In subsequent reference, the following concise terms will often be used:

(a) a plate location will be called a P-loc,

(b) a hinge location will be called an H-loc, and

(c) a socket location will be called an S-loc.

These will be referred to collectively as a location triad.

A displacement at a locating feature will result in a displacement at other points on the located body. A point at which the displacement is required will be called a result-point (or more concisely an R-point).

An R-point located on a triad will be shown graphically as exemplified in Fig. 7. The root node represents the R-point and the three links are distinguished by the convention:

(a) a P-loc is shown by a square,

(b) an H-loc is shown by a triangle, and

(c) an S-loc is shown by a circle.

Each symbol is placed on the appropriate link, and a link indicates the relationship 'is located on', in a top-down sense.

3.3 Displacement Matrices

The general location element, described previously, is analysed by using energy methods since these are commonly used in engineering science.

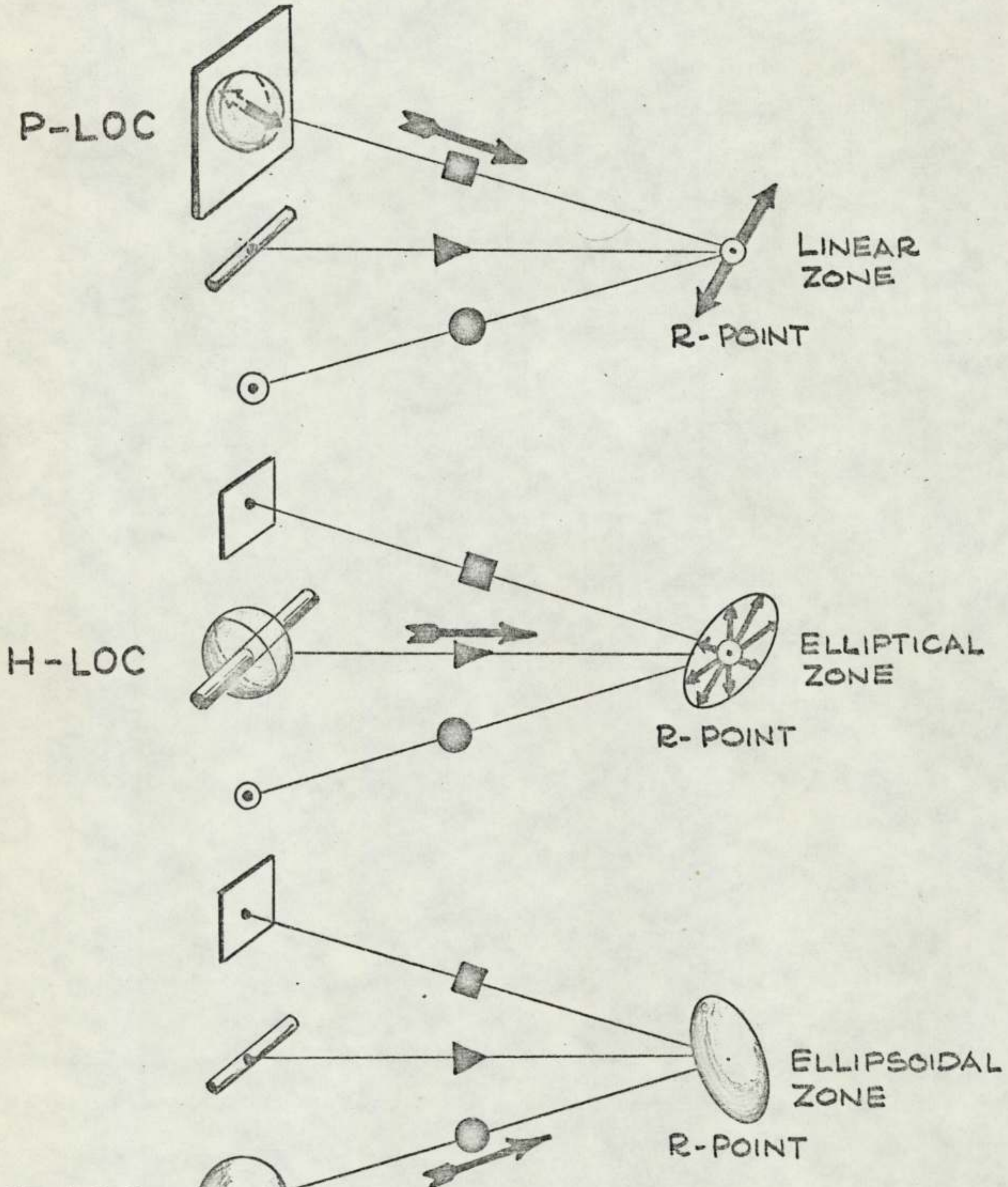


Fig 8 Transformations

Details of the analysis will be found in Appendix A but an outline is given here for reference.

The parameters listed below define the location system, coordinates being relative to some convenient set of orthogonal axes. The sense of the directions of the lines is immaterial.

(a) Coordinates of the R-point.

(b) Coordinates of the points of action of the P-, H- and S-locs.

(c) Direction cosines of the normal to the locating plane and of the direction of the locating line.

If the displacement at a locating feature be $\overline{D}_{in}$, and the displacement at the R-point be $\overline{D}_{out}$, both of these being column vectors, then

$$\overline{D}_{out} = \overline{M}\,\overline{D}_{in}.$$

$\overline{M}$ is a 3 x 3 matrix with coefficients depending on the coordinates of the locating triad. Each location feature will have a different matrix; the notation for these is shown below.

(a) P-loc; $\overline{M} = \overline{P}$

(b) H-loc; $\overline{M} = \overline{H}$

(c) S-loc; $\overline{M} = \overline{S}$

There are, as is discussed in Appendix A, restrictions on the positions and directions of the locating features. These define a proper location and are easily visualised; for instance, a socket may not exactly correspond with the centre of action of a P-loc. If the location features are not restricted in this way, then matrix coefficients may become infinite.

It is proved in Appendix A that a $\overline{P}$-matrix is of rank at most 1, an $\overline{H}$-matrix of rank at most 2 and an $\overline{S}$-matrix of rank at most 3. The matrices may be thought of as three-dimensional transformation operators, and if they act on a unit sphere, then the $\overline{P}$-matrix transforms it to a straight line, the $\overline{H}$-matrix transforms it into an ellipse; and the $\overline{S}$-matrix transforms it into an ellipsoid. These transformations are illustrated in Fig. 8.

4. LOCATION NETWORKS

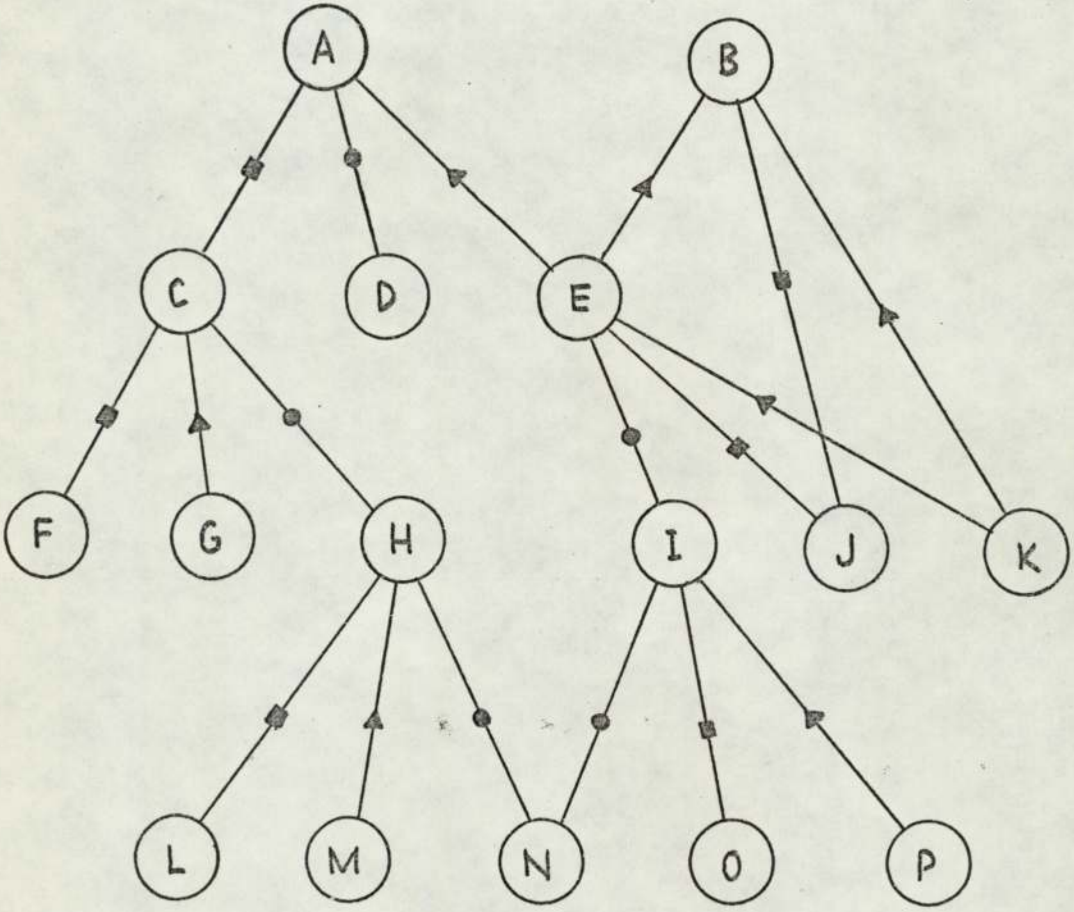


Fig 9 Location Networks

4. Location Networks

4.1 Assemblage network and paths

An assemblage may be represented by a directed graph consisting of location triads linked together as shown in Fig. 9. The effect of a displacement at feature 0 will be transmitted through the network to feature B. The corresponding displacement at B will be found by multiplying all the matrices corresponding to the edges of the graph lying on the path between the two features. If the product matrix be $\overline{M}$, then the output displacement at B may be found from the general equation:

$$\overline{D}_{out} = \overline{M}\,\overline{D}_{in}$$

where $\overline{D}_{out}$ is the output displacement column vector, and

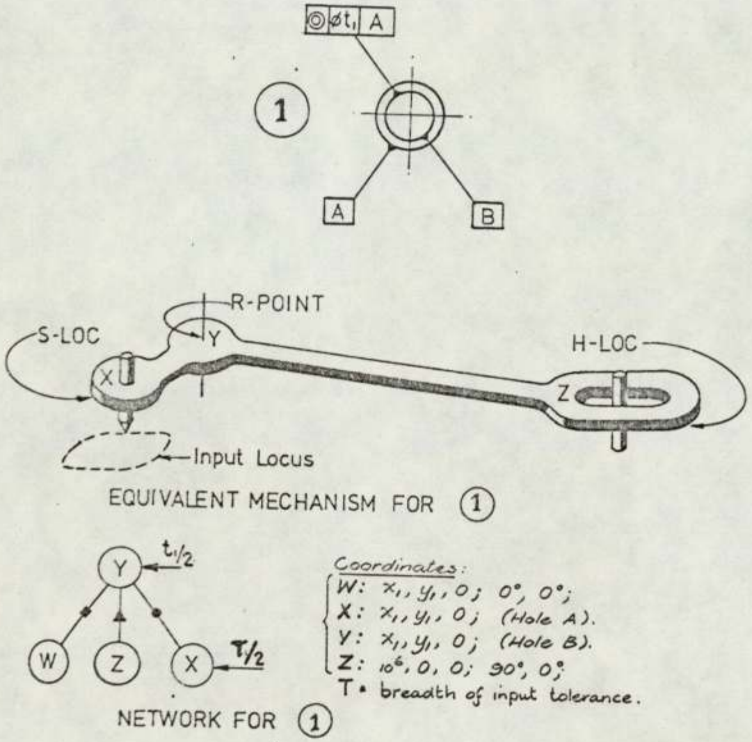
$\overline{D}_{in}$ is the input displacement column vector.

Often multiple paths exist between the location at which the displacement is applied and the R-point. In this case, the calculation of the matrix $\overline{M}$ is not so straightforward and a discussion is to be found in Appendix A. An example of multiple paths is the pair of paths joining nodes N and A in Fig. 9. (Paths N H C A and N I E A).

If a spherical displacement locus is applied at 0, then the resulting output displacement locus at B will be a transformation of the sphere. In the general case, the output zone might be linear, elliptical or ellipsoidal depending on the rank of the transformation matrix $\overline{M}$ (see Fig. 8). For a pair of nodes which are connected by multiple paths, it is not possible to predict the rank of $\overline{M}$ without first calculating the path products. For a simple path, the rank of $\overline{M}$ will be the lowest rank of matrix associated with any edge along it (Appendix A).

4.2 Examples

The construction of assemblage networks requires some skill in visualising the tolerance mechanisms involved, although some assistance is provided in Appendix D which contains details of all the common locating



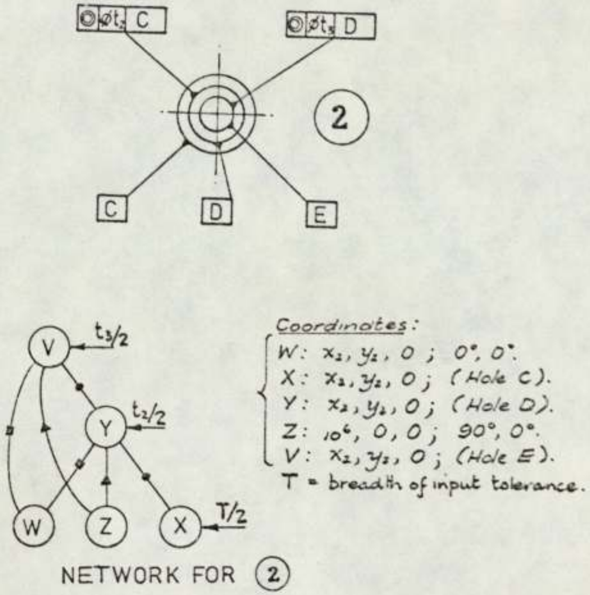
EQUIVALENT MECHANISM FOR ①

NETWORK FOR ①

Coordinates:

W: $x_1, y_1, 0$; 0°, 0°;
X: $x_1, y_1, 0$; (Hole A).
Y: $x_1, y_1, 0$; (Hole B).
Z: $10^6, 0, 0$; 90°, 0°;
T = breadth of input tolerance.

Fig 10 Subnetworks & Mechanisms



Coordinates:

W: $x_2, y_2, 0$; 0°, 0°.
X: $x_2, y_2, 0$; (Hole C).
Y: $x_2, y_2, 0$; (Hole D).
Z: $10^6, 0, 0$; 90°, 0°.
V: $x_2, y_2, 0$; (Hole E).
T = breadth of input tolerance.

NETWORK FOR ②

Fig 11 Sub-networks & Mechanisms.

systems. Fig. 10 shows a simple example. Hole A is located on the centre of hole B, the nominal centres of both holes being coincident and hole A having a concentricity tolerance relative to hole B. Extrinsic tolerance due to displacement of hole B must be separated from the intrinsic tolerance due to the concentricity tolerance. The situation is shown in Fig. 10, the mechanism XYZ being used to assist in visualising the location triad. Some, possibly irregular, tolerance zone exists at hole B, and this must be passed unchanged to hole A which has its own tolerance relative to hole B. If member XY be made very short and member YZ very long in comparison with other dimensions in the neighbourhood of holes A and B, then the path traced out by point Y will be very nearly the same as that traced out by point X. The displacement at X will be passed unchanged to Y in the limiting case. If point X is taken to represent the locating hole B and point Y is taken to represent the located hole A, then the required locating system has been obtained. The use of links of zero length such as XY, and links of infinite length, such as YZ, is common and there are several standard cases in which these are useful. The final location triad is shown in Fig. 10.

The whole assemblage network is made up from standard components similar to the one just described - another two-dimensional example is illustrated in Fig. 11. In order to generate a two-dimensional system, the P-loc is taken to be in the plane of the paper and the two-dimensional H-loc and S-loc representations are a slot and pivot as shown in Fig. 11.

Further discussion of the network will be found in Appendix D.

4.3 Generation of Tolerance Zones

The input tolerance zone at a feature is always taken to be a sphere. The output tolerance zone at the located feature may be linear, or elliptical or ellipsoidal depending on the matrix of the path joining the two features. The use of a spherical input tolerance zone is not as restrictive as it might appear, since all other common tolerance zones may be generated

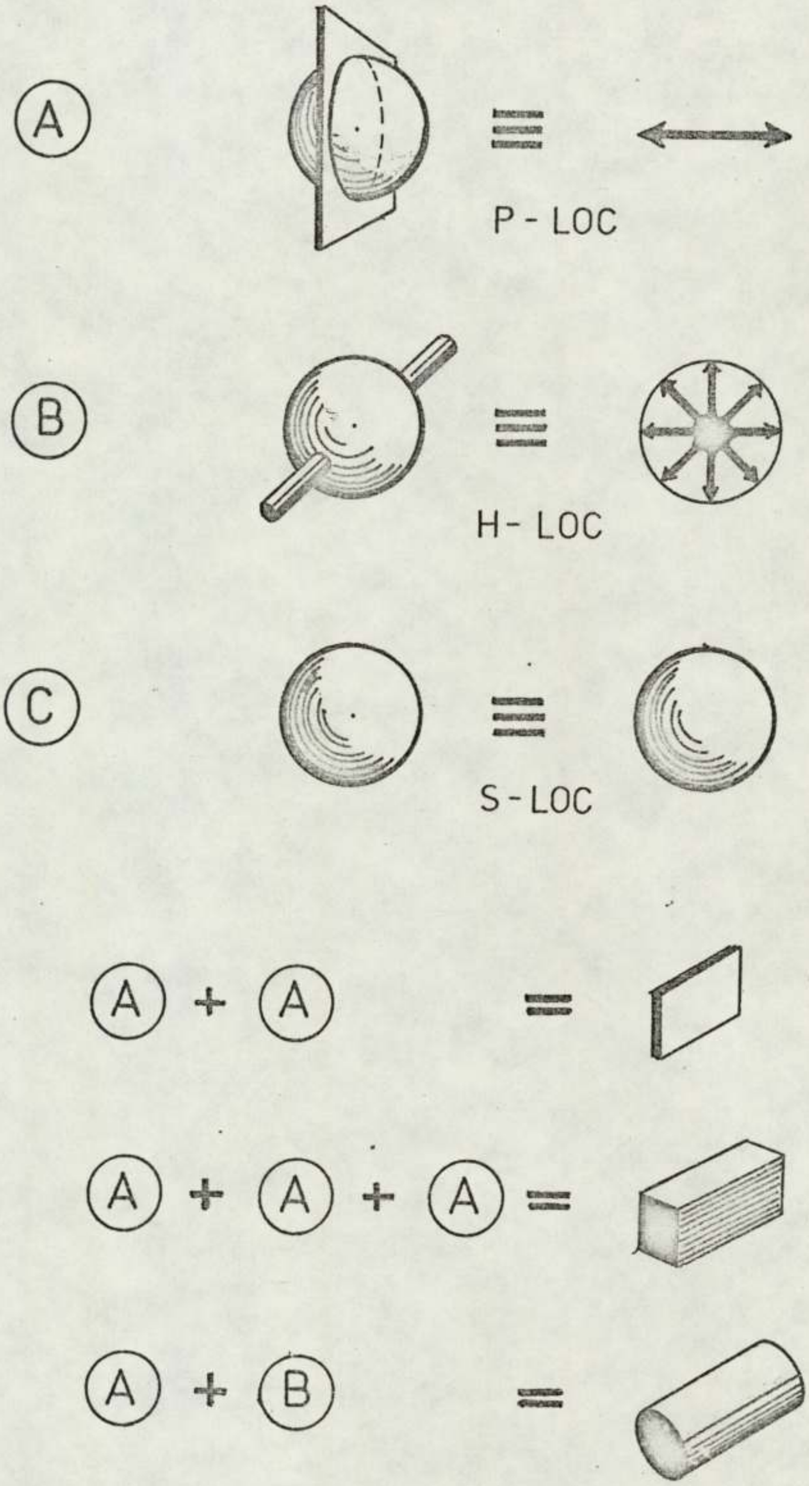


Fig 12 Generation of Tolerance Zones.

from it – see Fig. 12.

If a spherical tolerance is applied to a P-loc, then since only displacements normal to the locating plane are sensed, this is equivalent to a linear tolerance zone. At an H-loc, only displacements at right-angles to the locating line are sensed, so the application of a spherical zone is equivalent to a circular zone. At an S-loc, the whole spherical zone is sensed.

By superimposition of P-locs, a parallelepipedal tolerance zone may be generated. Similarly, superimposition of H- and P-locs generates a cylindrical zone.

A spherical tolerance zone may therefore be used to generate all the standard tolerance zones listed in BS 308 and quoted in section 2.3.1. A more detailed treatment is given in Appendix D.

4.4 Use of a location network

When a location network has been established for a particular assemblage, it may be used to provide qualitative answers to various questions of interest.

(a) The vector displacement at a locating feature is known. What is the effect at a located feature?

The output displacement may be found directly from the relation

$$\overline{D}_{out} = \overline{M}\,\overline{D}_{in}$$

$\overline{M}$ is the path matrix between the two features. If this is to be found automatically, it is essential that if there be no path between the two features, then $\overline{M} = 0$.

(b) If a spherical tolerance zone is applied at a locating feature, what will be the maximum displacement and its direction at a particular R-point on the body?

Again the path matrix $\overline{M}$ is calculated. The maximum displacement and its direction are evaluated by finding the dominant eigenvalue and associated eigenvector of the product of $\bar{M}$ and its transpose. Details of the method used are to be found in Appendix B, but it is unnecessary for the user to know anything about the method used.

(c) A feature of interest depends on several locating features. What will be the effect of unit tolerances at each feature?

$\bar{M}$ matrices are calculated for the paths between each locating feature and the R-point in question. The maximum tolerances for each are evaluated as outlined in (b) and displayed in a convenient way. The results may be used as an aid to the selection of manufacturing processes and locating systems.

(d) What is the relative displacement between two features of interest in the assemblage?

The calculation is performed using a device described in Appendix A which is again transparent to the user. The answer is useful in several ways. It may, for instance, be employed to calculate clearances between points on the assemblage. Another use might be to check whether a particular system of location used in the manufacturing process gives tolerances which are inside the bounds specified in the functional drawing of the part.

5. THE WORKING SYSTEM

5. The Working System

5.1 The Target Computer configuration

It has been assumed, in the design of the pilot system, that

(a) the computer available for engineering use is a bare 16K mini,

(b) the program is to be contained within 8K, the remainder of the store being reserved for array space,

(c) a compiler is available for a reasonably high level language such as ALGOL 60 or FORTRAN.

The selection of the computer is clearly of great importance in the system design; and it was decided at an early stage that the design basis should be the minimal configuration above. A useful network should contain around two hundred nodes and 8K would be sufficient to provide array space for this size of assemblage.

Since the data structure chosen is fairly complex, ideal languages would be ALGOL 68 or PL/1 since both provide reference variables so that data structures may be built up dynamically. It is, however, unlikely that either of these languages would be available on the minimum configuration selected. It was reluctantly decided that the linked structure would be held in array form, links being integer pointers to array elements. This is a common, although artificial, way of holding a linked structure, but it does have the clear advantage that a data structure may be output in a comprehensible form.

The prototype program was written in ALGOL 60 and it did, after some paring, fit into 8K of store. The computer used was a Marconi-Elliott 905 which has an excellent ALGOL 60 compiler with good error diagnostics. ALGOL 60 was used in preference to FORTRAN mainly because it is more suitable for the communication of algorithms. The fact that FORTRAN has no facilities for dynamic arrays is irrelevant, because in this application a fixed area of core is set aside for array space.

5.2 The assemblage network . . . computer representation

The first stage in the analysis of the location systems on an assemblage is to set up the network manually. It might be possible to automate this to some extent, since sub-networks for all the common cases of dimensions systems have been established (Appendix D). These might be stored as a library of standard cases in the computer, probably on backing store; and the relevant sub-system selected by means of an index. The appropriate feature coordinates would also be supplied together with linkage data. A section of network would then be linked into the main data structure together with node information. There are two main difficulties. Firstly, it would be necessary to maintain a dynamic data-structure and this is not conveniently achieved in the languages most commonly available for engineering applications. Secondly, in some sub-systems (for example, those defining symmetric tolerance), some of the coordinates of nodes internal to the sub-system are calculated from externally supplied coordinates, and so a library entry would consist not only of a piece of structure, but would also contain a section of code which would be handled rather like a macro definition. This is an interesting problem but it would complicate the system drastically. For this reason, the library of standard cases is held in a manual in the prototype system and the network completed by hand.

It would be unreasonable if it were required that a whole network were to be compiled by hand for an assemblage as complex as, say, a motor car. Fortunately, networks can be built up piecemeal from more tractable sub-networks which can be separately tested. The physical unit corresponding to such a sub-network might be as small as a single process drawing.

The present prototype program is for general purposes, but in a more elaborate configuration, a separate specialised program for validating sub-networks would be very useful. This might display selected output tolerance zones graphically for given input tolerances so that the correctness of a given sub-network could be checked before incorporation into the main system. The test program would also be a valuable training aid, particularly if output tolerance zones were displayed visually.

5.3 Data Input Format

The details of the network are supplied to the program by providing the data for each node. The format of the input data is described below.

(a) External node index

Each node represents a feature on the assemblage and must have a distinct index. These are provided in random order and the format may be designed to any fixed convention. In the prototype program, simple positive integers were used.

(b) Node type index

For a normal node, the type index is 0. In the case of a node with unitary links (see Appendix D), the type index is 1, such nodes being treated in a special way. Artificial nodes of this kind will have non-zero indices and although the unitary node is the only one included in the prototype program a good case might be made for using others, notably those connected with symmetric tolerances.

(c) Link indices

Each normal node will have three links, each pointing to another node in the network. Leaf nodes will have links pointing to a notional null node, indicated by zero. Artificial nodes are treated in a different way and the unitary node, for example, may have a mixture of zero and non-zero links.

A node may not have a link to itself - this will be rejected at the data validation stage of the program. The convention assumed for the order of the links is (i) P-loc link, (ii) H-loc link, (iii) S-loc link. Weak links (see Appendix D) are distinguished by a negative node number.

(d) Feature coordinates

All nodes have the following five coordinates: X, Y and Z coordinates relative to the general reference axes of the system, and two angles which define the directions of the locating plane for a P-loc and the locating line for an H-loc. These are specified in the prototype program as degrees and in cylindrical coordinates.

Some storage space is wasted by quoting angular coordinates for an S-loc. If these were not included the array structure would become more complicated. Another reason for including them is that it enables an S-loc to be used in a dual role as a P- or H-loc which might be useful for larger networks since this saves nodes.

(e) Tolerance size

The bilateral tolerance size (or radius of the generating spherical tolerance zone) may be included, if it is known, and if qualitative values of displacements are required. This was not done in the prototype, all tolerance zones were considered as being of unit size and the output interpreted as displacement per unit input tolerance or sensitivity coefficient. Other information which might be useful here is the standard deviation of the process tolerance in the case of a well-established process. This would enable statistical confidence limits to be calculated for output tolerances as is done in the system described in ref. S.5.

5.4 Internal Node Data

(a) Internal node indices

It is desirable, although not essential, that node indices should be provided in random order on input. Networks for large assemblages are built up from smaller sub-networks and the onus of organising the feature references into a form suitable for computer processing is better put on the computer than on the user. The program assigns an internal index to each node which is held as part of the node record. This internal index is an integer with absolute value in the range 1 - N where N is the total number of nodes. Again, 0 is used as the null node and negative integers denote weak links. Internal node indices are assigned to each node in topological order. This is discussed at length in Appendix B but informally may be defined in the following way:

'If nodes are in topological order, then no node can have a link to a node with a lower index, except to node zero, which is a special case.'

Internal node indices may be used in two ways. Firstly, the node data may be sorted so that all the nodes are physically in topological order. Secondly, a node index vector might be held in store and all operations on nodes might be performed indirectly. Each method has its own merits; the former being faster for actual processing of an established network, but resulting in re-ordering of the prime data; the latter requiring that node access has a further degree of indirection which is particularly time-consuming unless the compiler uses Iliffe vector array access. In the prototype program, node records were topologically sorted.

(b) Scratch pad matrix

Associated with each node is a 3 x 3 matrix which has coefficients depending on the coordinates of the location triad of which the node is the R-point. This represents a large overhead of store, but it is difficult to see how processing of networks might be achieved efficiently without it.

(c) Direction cosine vector

For speed of execution, the two cylindrical coordinates which define the normal to the plane or the line of the hinge are converted to direction cosines which are held in the node record as a 3-element vector.

It is possible that some, or all, of the internal node data might be omitted and the associated quantities calculated as required during processing. As usual, the compromise must be made between minimising storage space and reducing running time. In the prototype program, it was decided that 200 nodes should be sufficient for most practical problems and

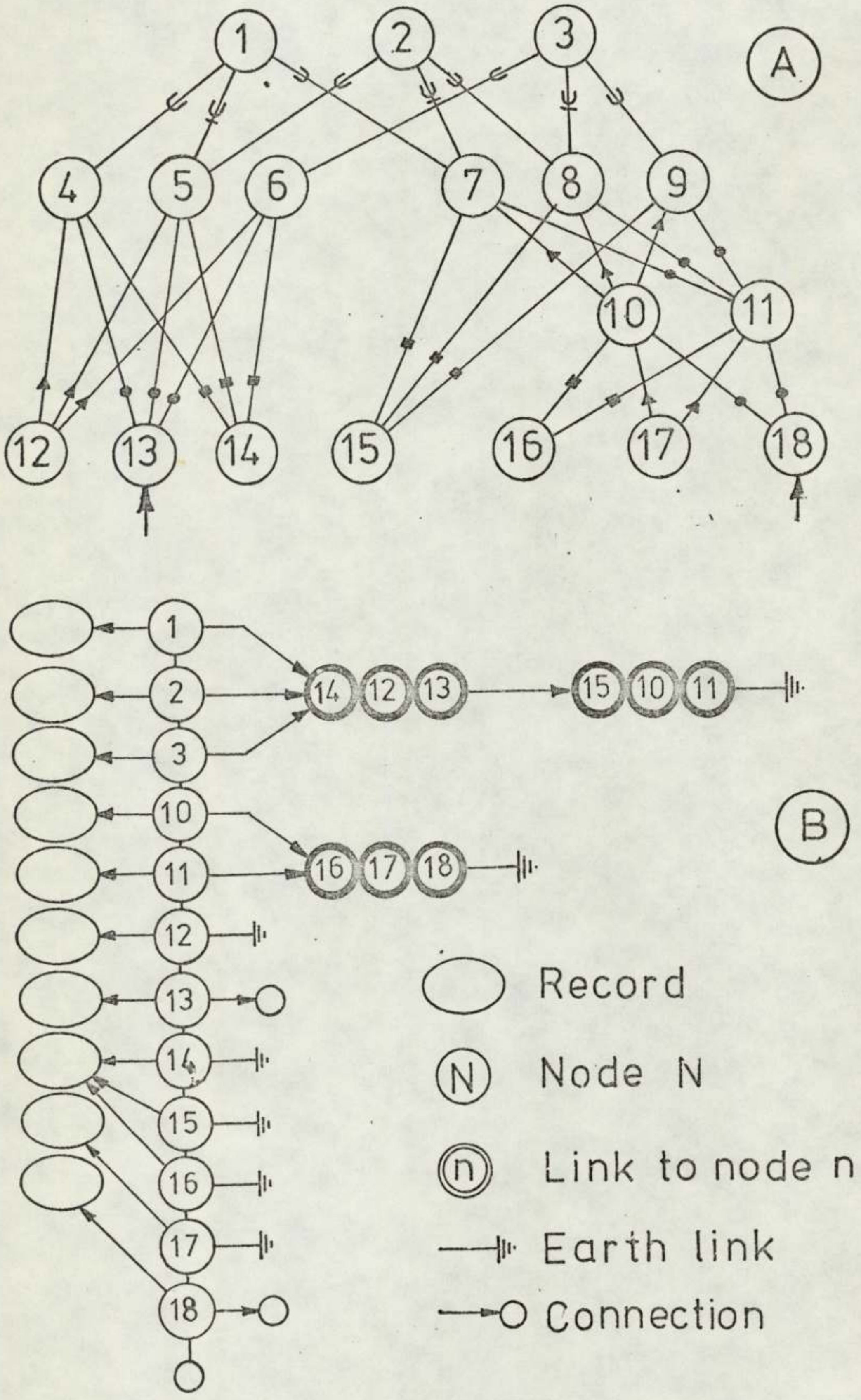


Fig 13 Alternative Data Structures.

so all internal data was included since sufficient array space was available for them.

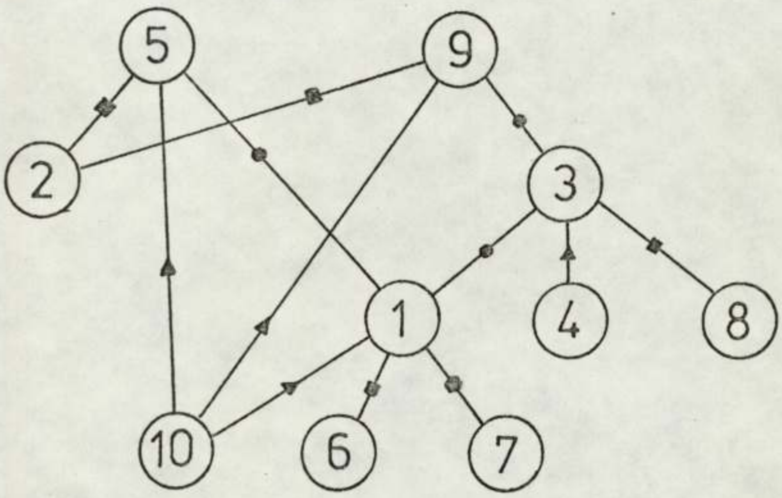
5.5 The structure - alternatives

So far, the data structure has been referred to rather tentatively as 'a network'. There are several possibilities for the actual form of the data structure: two of these are particularly useful and a discussion of their respective merits follows:

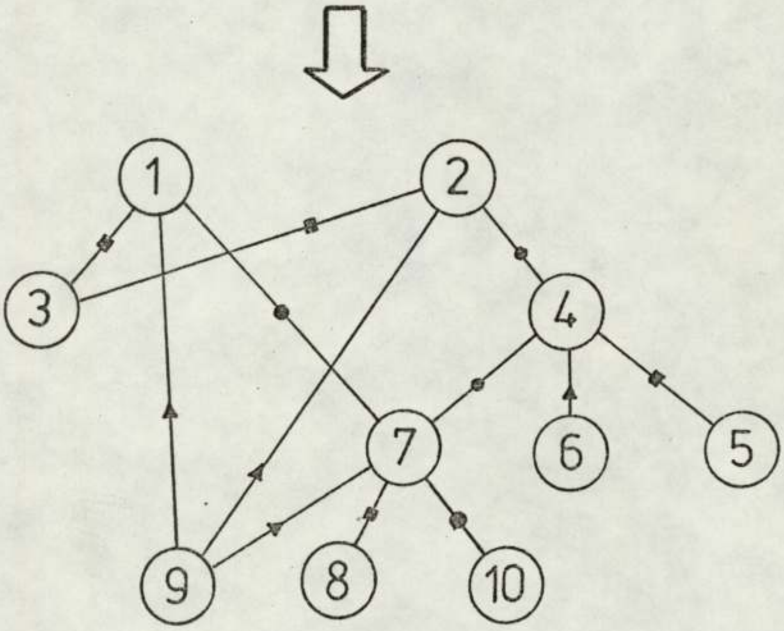
(a) This is the most natural structure and an example will be found in Fig. 13A. The set of nodes is connected unidirectionally. Nodes have outdegree three except for the leaf nodes which have outdegree zero. One or more root nodes have indegree zero, while the remaining nodes are not restricted as to indegree. There are no closed loops. Although it is tempting to refer to the structure as a ternary tree, it would be incorrect to do so since all sub-structures are not disjoint (ref. G.1). A similar structure without the restriction on outdegree is termed 'a generalised arborescence' and 'a hierarchical structure' in ref. G.15. To avoid inventing another name equally as clumsy as these, the structure will inaccurately but concisely be called a 'tree' from now on.

(b) Another structure which is more flexible than the previous one is shown in Fig. 13B. The main advantage of this structure is that duplicated input data is avoided since each node in the structure does not carry directly its associated record, but merely a pointer to it. The structure is divorced from the prime data and so all calculations are indirect. Another clear advantage is that horizontal links represent superimposition and so unitary matrices are not required.

The second structure is appropriate where a language is used which has reference (ALGOL 68) or pointer (PL/1) variables. For more commonly used



RANDOM ORDER



TOPOLOGICAL ORDER

Fig 14 Topological Sorting

languages, the indirection involved is probably an intolerable overhead of processing time, and for this reason in the prototype program, the less compact but more natural 'tree' was used.

5.6 Data validation phase

The only data which can be directly checked for validity are the external node index and the node type index. The former must conform with a fixed format and the latter is restricted to a fixed number of integers (0 and 1 in the prototype program). In the prototype program, if N nodes were input, then each external node index would be a separate integer in the range 1 - N but in random order. This is not excessively restrictive on the user, but might be inconvenient where a large network was to be built up from smaller sub-'trees'.

At this stage, the 'tree' must be checked to ensure that it contains no closed loops which would be physically impossible and would result in the program looping. This may be done by topologically sorting the nodes, which also facilitates processing of the 'tree' at a later stage. Topological sorting is described in refs. G.1, G.13 and G.15; the algorithm used being a modification of the one described in ref. G.13.

Each node is re-numbered, node indices again being consecutive integers in the range 1 - N when N is the number of nodes. After sorting, a node (K) does not point, even indirectly to nodes (1) to (K - 1) - see Fig. 14.

The algorithm is described in Appendix B; it constructs a sort-index which is a vector of N elements showing the sorted position of each node. The sort-index is used to sort the data physically. Although this is not absolutely necessary, subsequent processing being possible by referring to the sort-index, it is convenient because -

(a) it is useful to separate the routines for setting up the structure from those used for processing it,

(b) the structure should be permanent after being validated and so the physical sort is only required once,

(c) much indirect referencing is obviated in subsequent processing, and

(d) subsequent programming is easier. The algorithm used for sorting on the index was also obtained from ref. G.13 (see Appendix B).

At the end of this phase, the data structure is ready for processing. The internal node index is the index of the node in topological order. It is convenient at this stage to convert the angular coordinates of P- and H-locs to direction cosines which are written into the direction cosine vector.

5.7 Processing the structure - prototype program

In the interests of conserving storage space, nodes were re-numbered during the sort phase. All nodes must be subsequently referred to by their new numbers in the prototype program. For this reason a sort-index is output at the beginning of the processing stage. The user must re-number the nodes on his 'tree' diagram with the help of this index. In a larger system, the original node numbers would still be available and an inverted list used to access the sorted node numbers which would only be used internally. The existing system is mildly inconvenient.

Two options only are available in the prototype program and are described below. For test purposes, the path matrix elements and the number of iterations required in the eigenvalue calculation may be output. These may be suppressed if, as is likely, they are not required.

(a) Maximum displacement

Required: a maximum sensitivity coefficient - i.e. the maximum displacement at a result feature caused by the application of a unit tolerance at an input feature.

Input (i) the result feature node number,

(ii) the input feature node number.

The 'tree' is traversed from result feature node to input feature node, path matrices being calculated cumulatively at each node encountered en route as described in Appendix A, section A4.

The maximum displacement is calculated as described in Appendix A, section A6, and output together with its associated direction cosines.

(b) Relative displacement

Required: the relative displacement between two result features due to a unit tolerance at an input feature which affects either, or both.

Input (i) the code 0,
(ii) the two result feature node numbers,
(iii) the input feature node number.

A dummy node is attached to the two result points and the 'tree', of which it is the root node, is traversed. This device is described in Appendix A, section A4. The relative displacement is calculated and output with its associated direction cosines which are used merely for checking purposes.

5.8 Further extensions

The options described in section 5.7 are sufficient for normal use, but several more may be added for convenience. Two of these are:

(a) Calculation of all the sensitivity coefficients at a result feature.

Since all the nodes at which tolerances occur are known to the user, these may be flagged on input to the system. A list of sensitivity coefficients may be obtained by traversing the 'tree' and calculating maximum displacements at each input node. In order to do this efficiently and to obviate superfluous output, a more elegant method of traversal is desirable. This would require an appreciable increase in the size of the program and so was omitted from the prototype program.

(b) Calculation of the maximum displacement in a particular direction.

This is done easily from the paths matrix and since it is a straightforward calculation by hand, it was omitted from the prototype program.

5.9 An Integrated Tolerance Control System

The system described in sections 5.1 – 5.7 is designed as a stand-alone program for a mini-computer. Its input is a network description of an assemblage together with a list of features of interest and the points having tolerances which affect them. The output is a list of sensitivity coefficients for each feature of interest. This is a useful tool for the analysis of tolerances in its present form. However, if a more ambitious configuration were available, several other sub-programs involving established techniques could be amalgamated to form an integrated tolerance control system. In view of the interest displayed in the system described in ref. S.5 which also assists the designer in part of the analysis of dimensional tolerancing, an integrated system would be an invaluable aid in this field.

A possible configuration might consist of the following modules:

(a) Network Proving Sub-program

As each component of an assemblage were considered, its individual location networks might be separately proved by using a specially tailored version of the prototype program. Ideally, interactive graphics would be used to display the envelope of the output tolerance zone for one input tolerance zone or several acting simultaneously. This would enable the user to prove the sub-network to his own satisfaction. A set of standard sub-networks, such as those given in Appendix D could be stored and displayed on demand individually, and tested interactively ensuring that the case selected was appropriate to the location situation. The catalogue of standard cases would be augmented by cases which had been thoroughly proven. It would also be convenient to display information regarding the purpose and usage of each standard sub-network on demand.

(b) Network Building Sub-program

A difficulty with the current prototype program is that if the sensitivities obtained for a particular application are not satisfactory, then the complete network must be modified and re-input. This is due in some measure to the restricted core available on the target configuration, but also because Algol 60 is not a suitable language for handling data structures of any complexity. If Algol 68 or PL/1 were available then the structures could be dynamically modified and it would be possible to delete sections of network, to insert modified sub-networks and to append proven sub-networks to an existing network. A complex assemblage network could be built up section by section interactively, which is a more natural method of developing the data structure.

(c) Sensitivity Coefficient Sub-program

The next stage in the design process would be to process the established network and obtain sensitivity coefficients for all tolerances affecting points of interest. This would be a refined version of the prototype program; an obvious improvement being to trade off some storage space for a quicker and more elegant method of traversing paths in the network. The output from this sub-program would be lists of sensitivity coefficients for each critical feature in the assemblage. Possibly, some of these might be sufficiently low to be ignored and the network might then be re-defined omitting them in the interests of running efficiency.

(d) Allocation Sub-program

Eventually, a stage would be reached when the designer was content with the assemblage description. The tolerance allocation could then be optimised on a least cost basis. Two options would be available: statistical and surefit bases (see Appendix C). This would be a logically straightforward section but judging from the variety of the methods available for non-linear optimisation, it would probably require study by a specialist in the field. The output from this sub-program would be the actual tolerances at each input point. It is possible that some of these

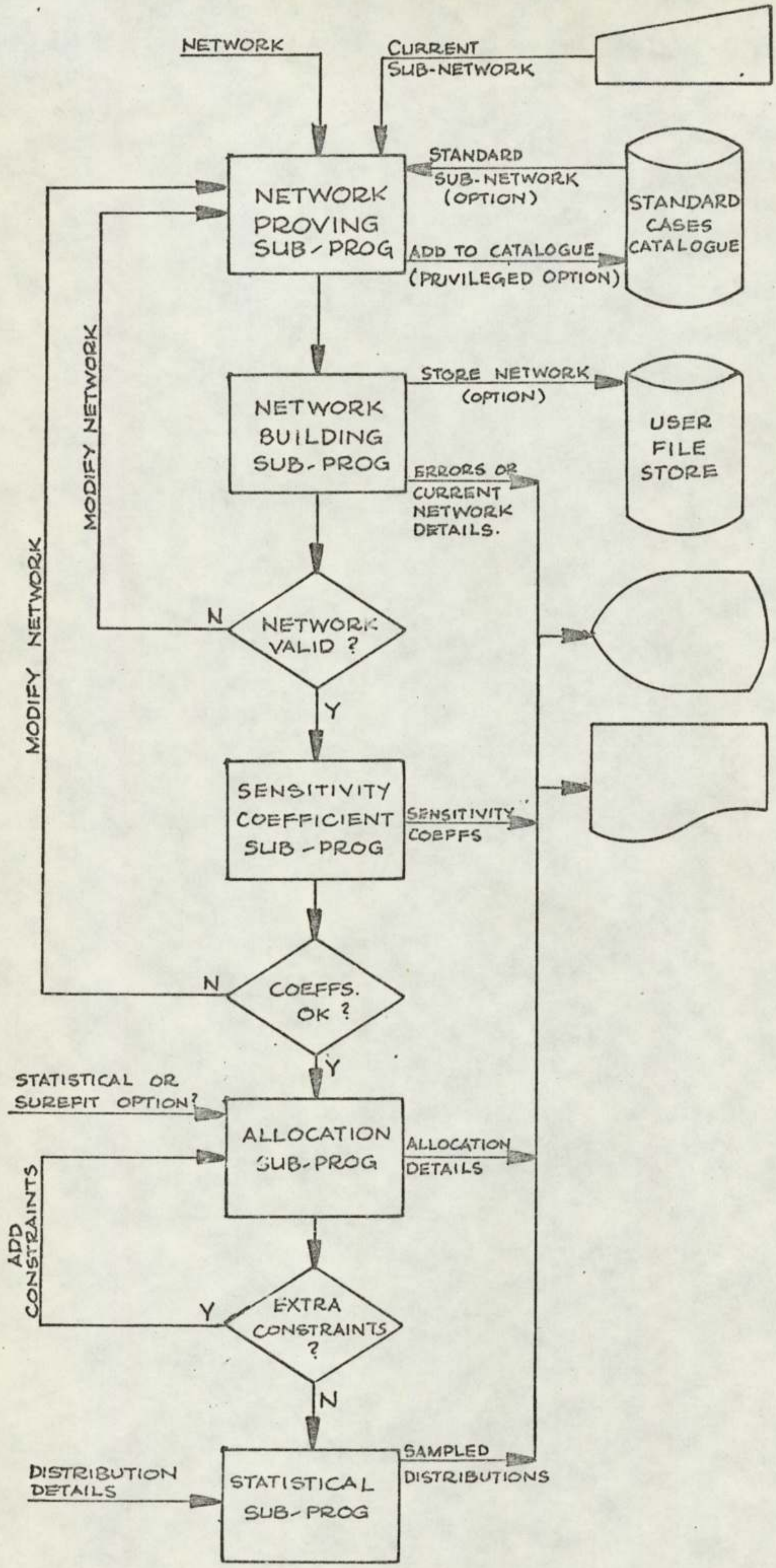


Fig 15. An Integrated Tolerance Control System.

might be too small for practical considerations.necessitating further constraints to be applied (see Appendix C) and subsequent reprocessing and recalculation.

(e) Statistical Analysis Sub-program

The final sub-program in the system would be a system similar to the one described in ref. S.5 but operating on the output from section (d) rather than on sub-programs and data supplied by the user. The results which could be in histogram form would give the distribution of the variations in size on features of interest.

A suggested system is illustrated in Fig. 15.

5.10 Comments on the System

Much of the preliminary work has been done with dimensioning equally as eccentric as that on the examples discussed in Appendix E.1. A problem in applying the method to practical examples is that most designed components are under-defined dimensionally, occasionally even in critical measurements and assumptions must be made particularly with regard to tolerances such as squareness, flatness and parallelism which are normally not specified explicitly. Usually it is assumed, even by experienced detail designers, that some features of a component are geometrically exact. A location network is certainly a more precise method of specifying a part than most dimensioned drawings.

For most of the applications which have been checked analytically, the sensitivity coefficients obtained have been accurate to two decimal places even when dimensions have been scaled from a drawing. Occasionally it is difficult to check a particular network and interactive graphics would be a great help.

The method is reasonably easy to use after a little practice. To date, a sub-network has been found for every dimensioning system encountered and the method should be particularly suitable for the use of engineering designers, who are normally good at visualising mechanisms. Since assemblies are represented by real, rather than the more complex geometrical, locations, applying the method to assemblies is very easy.

The restrictions of the target configuration have resulted in the system being a little inconvenient to use. Networks are best developed bit by bit in a similar way to that described in the examples but due to limited core it was necessary to keep the program as short as possible and it was not feasible to generate major networks dynamically. As each sub-network is proved, it is necessary to modify the network manually and this is then re-presented to the program.

It would seem from work done so far that this is a powerful method for analysing small displacements and it may have applications other than tolerancing. Some preliminary investigation has been performed on the analysis of kinematic mechanisms, and this seems promising.

Three-dimensional kinematic mechanisms are easier to model using the system than are tolerance mechanisms and the program may be used in its present form to determine instantaneous velocities of links in mechanisms. This has been done successfully in a variety of cases and further work is being carried out on the analysis of accelerations. Some recent papers have described efforts to analyse tolerance at joints in mechanisms; it seems that the system is useful for this purpose.

APPENDIX A

ANALYSIS OF THE MODEL

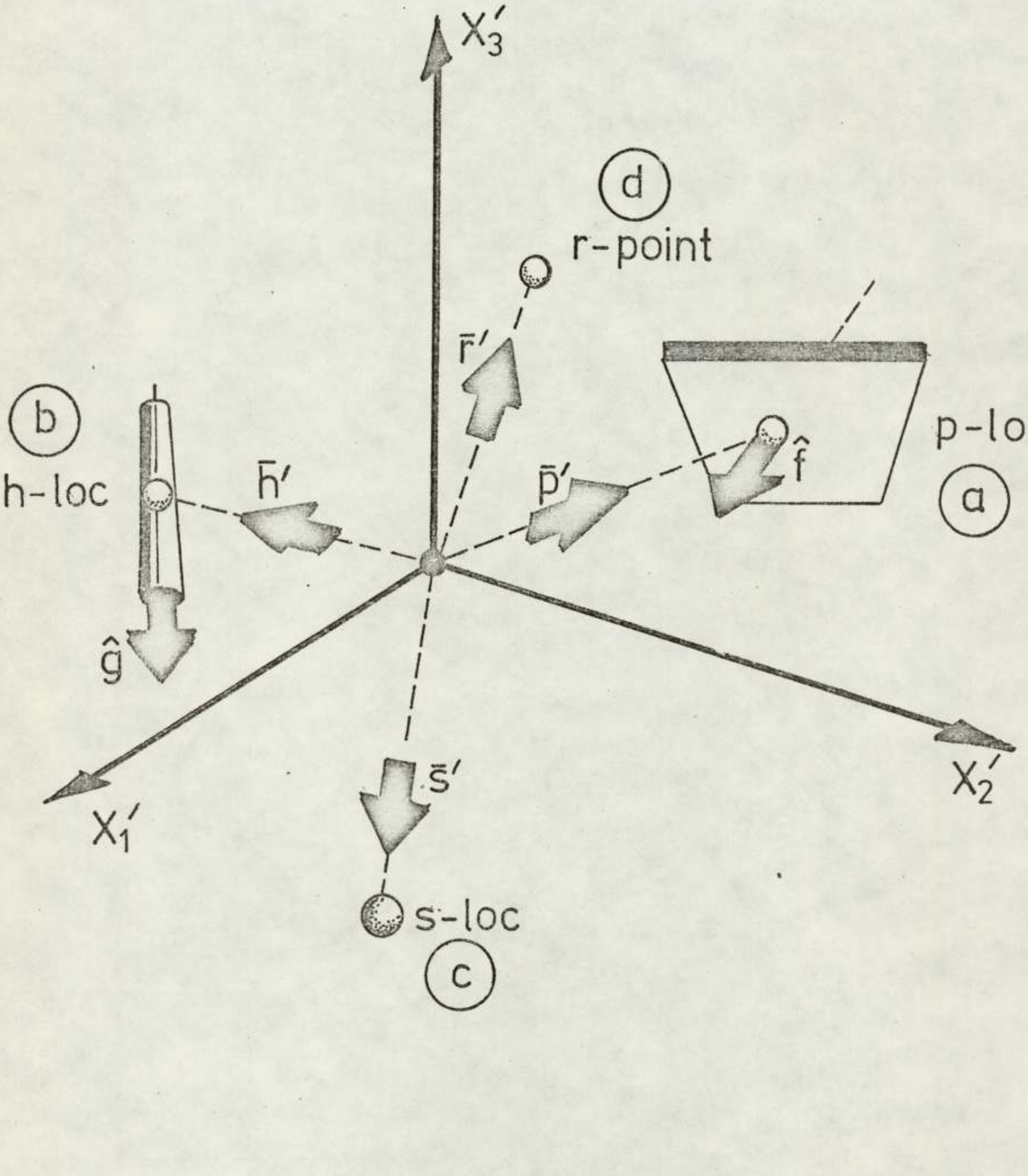


Fig A1. A Location Triad.

A.1 Analysis of the Location Triad

The body is located in a set of mutually orthogonal right handed axes $\left\{x'_1, x'_2, x'_3\right\}$ and is constrained as follows:

(i) At a point on a plane (P-loc). Fig. A1a.

$\overline{p}'$ is the position vector of the point of application on the plane.

$\hat{f}$ is the unit normal to the plane.

(ii) At a point on a line (H-loc). Fig. A1b.

$\overline{h}'$ is the position vector of the point of application on the line.

$\hat{g}$ is the unit vector along the line.

(iii) At a point (S-loc). Fig. A1c.

$\overline{s}'$ is the position vector of the point.

A general point on the located body (R-point) has position vector $\overline{r}'$. Fig. A1d.

Energy methods will be used to obtain displacements, and forces at P-, H- and S-locs, and the R-point are $\overline{P}$, $\overline{H}$, $\overline{S}$ and $\overline{R}$ respectively.

The system may be described by vector equations (i) - (iv):

(i) $\overline{P} + \overline{H} + \overline{S} + \overline{R} = \overline{0}$

(ii) $\overline{P} \times \overline{p}' + \overline{H} \times \overline{h}' + \overline{S} \times \overline{s}' + \overline{R} \times \overline{r}' = \overline{0}$

(iii) $\hat{g} . \overline{H} = 0$

(iv) $\overline{P} = |\overline{P}| \hat{f}$.

Equations (i) and (ii) are the general equilibrium equations, vector sums of forces and moments being zero. Equation (iii) represents the condition that force $\overline{H}$ is at right-angles to the H-loc. Equation (iv) represents the condition that force $\overline{P}$ acts normal to the P-loc.

Known values are:

P-loc: $\overline{p}'$ and $\hat{f}$.

H-loc: $\overline{h}'$ and $\hat{g}$.

S-loc: $\bar{s}'$

R-point: $\bar{R}$ and $\bar{r}'$

Required:

P-loc: $|\bar{P}|$

H-loc: $\bar{H}$

S-loc: $\bar{S}$

Solution:

(a) The equations (i) - (iv) are transformed by changing the coordinate axes to $\{x_1, x_2, x_3\}$ a parallel system with the S-loc as the origin. Position vectors will be modified as follows:

$$\bar{p} = \bar{p}' - \bar{s}'$$

$$\bar{h} = \bar{h}' - \bar{s}'$$

$$\bar{s} = \bar{0}$$

$$\bar{r} = \bar{r}' - \bar{s}'$$

Equation (ii) now becomes

(iia) $\bar{P} \times \bar{p} + \bar{H} \times \bar{h} + \bar{R} \times \bar{r} = \bar{0}$

(b) Taking the scalar product of $\bar{h}$ with (iia)

$\bar{h} \cdot (\bar{P} \times \bar{p} + \bar{H} \times \bar{h} + \bar{R} \times \bar{r}) = 0$

$= \bar{h} \cdot (\bar{P} \times \bar{p}) + \bar{h} \cdot (\bar{R} \times \bar{r})$ since generally $\bar{h} \cdot (\bar{H} \times \bar{h}) = 0$

$= |\bar{P}|\ \bar{h} \cdot (\hat{f} \times \bar{p}) + \bar{h} \cdot (\bar{R} \times \bar{r})$ since $\bar{P} = |\bar{P}|\ \hat{f}$ (equation iv).

Finally, $|\bar{P}| = -\dfrac{\bar{h} \cdot (\bar{R} \times \bar{r})}{\bar{h} \cdot (\hat{f} \times \bar{p})}$; $\bar{h} \cdot (\hat{f} \times \bar{p}) \neq 0$.

(c) Taking the vector product of $\hat{g}$ with (iia).

$\hat{g} \times (\bar{P} \times \bar{p} + \bar{H} \times \bar{h} + \bar{R} \times \bar{r}) = \bar{0}$

$= \hat{g} \times (\bar{P} \times \bar{p}) + (\hat{g} \cdot \bar{h})\bar{H} - (\hat{g} \cdot \bar{H})\bar{h} + \hat{g} \times (\bar{R} \times \bar{r})$

since generally $\bar{a} \times (\bar{b} \times \bar{c}) = (\bar{a} \cdot \bar{c})\bar{b} - (\bar{a} \cdot \bar{b})\bar{c}$

$= \hat{g} \times (\bar{P} \times \bar{p}) + (\hat{g} \cdot \bar{h})\bar{H} + \hat{g} \times (\bar{R} \times \bar{r})$

since $\hat{g} \cdot \bar{H} = 0$ from equation (iii).

Finally, $\bar{H} = -\dfrac{\hat{g} \times (\bar{P} \times \bar{p} + \bar{R} \times \bar{r})}{\hat{g} \cdot \bar{h}}$; $\hat{g} \cdot \bar{h} \neq 0$.

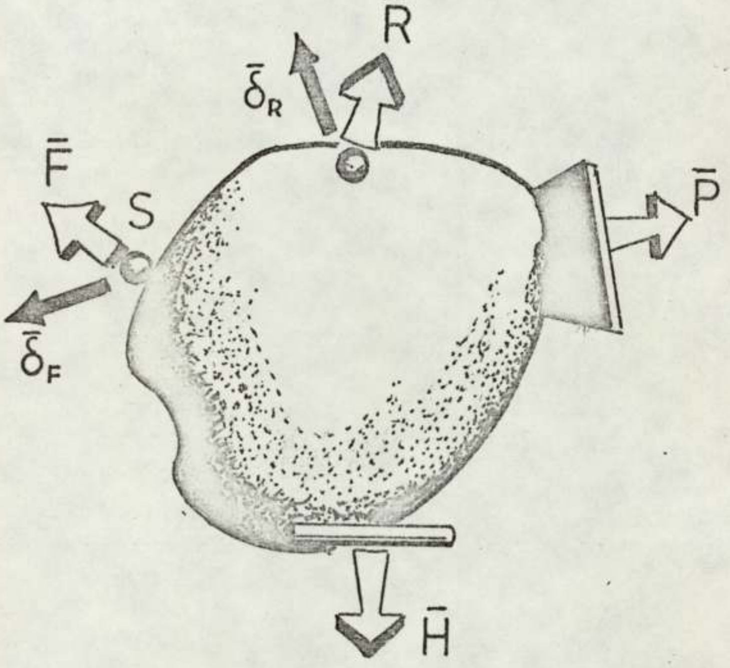


Fig A2. A Located Body

(d) From equation (i)

$$\bar{S} = -(\bar{P} + \bar{H} + \bar{R})$$

This completes the solution of equations (i) - (iv).

Summarising:

Solutions:

$$\text{(v)} \quad |\bar{P}| = -\frac{\bar{h} \cdot (\bar{R} \times \bar{r})}{\bar{h} \cdot (\hat{f} \times \bar{p})}$$

$$\text{(vi)} \quad \bar{H} = -\frac{\hat{g} \times (\bar{P} \times \bar{p} + \bar{R} \times \bar{r})}{\hat{g} \cdot \bar{h}}$$

$$\text{(vii)} \quad \bar{S} = -(\bar{P} + \bar{H} + \bar{R})$$

Conditions

$$\text{(viii)} \quad \bar{h} \cdot (\hat{f} \times \bar{p}) \neq 0$$

$$\text{(ix)} \quad \hat{g} \cdot \bar{h} \neq 0$$

Conditions (viii) and (ix) describe a proper location system.

A.2 Displacement at a Result Point

Fig. A2 shows a body located on the triad PHS. If a displacement $\bar{\delta}_F$ be applied to the locating feature F (which might be any one of P, H or S) then there will be a resulting displacement $\bar{\delta}_R$ at the R-point. If an arbitrary force $\bar{R}$ be applied at the R-point, then forces $\bar{P}$, $\bar{H}$ and $\bar{S}$ will result at the P-, H- and S-locs. These forces may be found from equations (v) - (vii) and in particular the force at F will be $\bar{F}$.

From energy considerations:

$$\bar{F} \cdot \bar{\delta}_F + \bar{R} \cdot \bar{\delta}_R = 0$$

The general locating feature F has a vector displacement $\{\xi_1, \xi_2, \xi_3\}$ where the subscripts here and in subsequent expressions denote components in the x_1, x_2 and x_3 directions respectively. The corresponding displacement at the R-point is $\{\eta_1, \eta_2, \eta_3\}$.

To find displacement component η_1, say, a unit force in the x_1 direction is applied at R. The resulting force at F may be written as

$\overline{F}_1 = \left\{F_{11}, F_{12}, F_{13}\right\}$, the first subscript denoting the direction of the unit force at R, the second denoting the component direction at F.

Similarly unit forces in the x_2 and x_3 directions yield the equations

$$\overline{F}_2 = \left\{F_{21}, F_{22}, F_{23}\right\}$$

$$\overline{F}_3 = \left\{F_{31}, F_{32}, F_{33}\right\}.$$

From energy considerations:

$$\left\{\eta_1, \eta_2, \eta_3\right\} = - \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} \left\{\xi_1, \xi_2, \xi_3\right\}$$

– or using the tensor suffix convention:

$$\eta_i = - F_{ij} \; \xi_j ; \quad i, j = 1, 2, 3$$

The F_{ij} may be found from equations (v) – (vii) using R_i in place of the general R.

$$\overline{P}_i = - \frac{\overline{h} \cdot (\overline{R}_i \times \overline{r})}{\overline{h} \cdot (\hat{f} \times \overline{p})} \hat{f}$$

$$\overline{H}_i = - \frac{\hat{g} \times (\overline{P}_i \times \overline{p} + \overline{R}_i \times \overline{r})}{\hat{g} \cdot \overline{h}}$$

$$\overline{S}_i = - (\overline{P}_i + \overline{H}_i + \overline{R}_i)$$

where subscript i denotes the ith row vector of the matrices subscripted and $\overline{R}$ is the unitary matrix.

It is useful to abandon vector notation at this stage since it is no longer convenient. Using the suffix summation convention, and the operators

Kronecker delta: $\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$

Permutation operator $\epsilon_{ijk} = \begin{cases} 1 & \text{for } ijk = 123, 231, 312. \\ -1 & \text{for } ijk = 132, 213, 321. \\ 0 & \text{otherwise,} \end{cases}$

the equations may be written in the compact form:

$$(x) \quad P_{ij} = - \frac{\epsilon_{rit} h_r r_t f_j}{K}$$

$$(xi) \quad H_{ij} = \frac{\epsilon_{rit} h_r r_t (g_d p_d f_j - g_e f_e p_j) - K(g_f r_f \delta_{ij} - g_i r_j)}{K g_g h_g}$$

$$(xii) \quad S_{ij} = - (P_{ij} + H_{ij} + \delta_{ij})$$

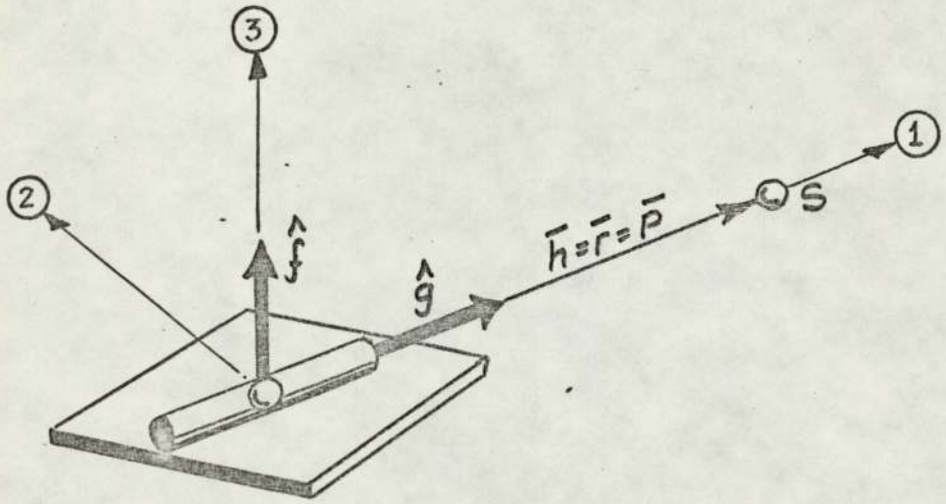
where $K = \epsilon_{abc} h_a f_b p_c$.

The subscripts in (xi) may be simplified a little and (xii) may be expanded, but the forms given are the most useful. When these expressions are used in the analysis of location 'trees', they are modified by being multiplied by −1, so that the general displacement equation may be written as

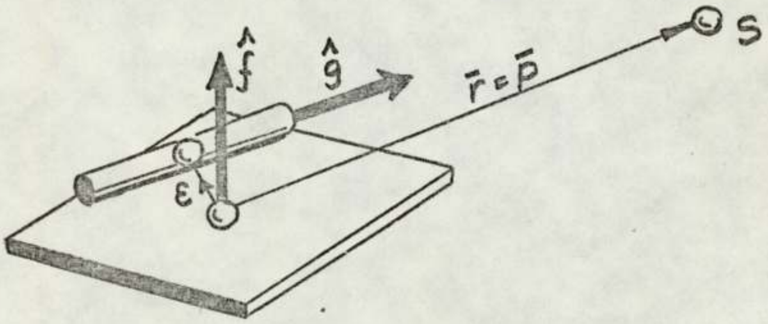
$$\eta_i = F_{ij} \xi_j \qquad i, j = 1, 2, 3.$$

This is purely a matter of convenience.

Although subscript notation is very convenient for algebraic manipulation, it did not prove very efficient for calculation of the matrix coefficients on the computer. The main reason for this was that in calculating the values of the permutation operators, most of the time was spent in evaluating zero coefficients. Special purpose algebraic manipulation languages have been implemented which, it is claimed, can evaluate expressions of this kind efficiently and conveniently (e.g. MATHLAB) but as most designers have access to the more mundane ALGOL 60 and FORTRAN, the vector approach was used in the prototype program. As usual, it was apparent that the commoner computer languages are not very suitable for mathematical purposes.



R, H & P coincident



H perturbed by vector $\bar{\varepsilon}$.
R & P coincident.

Fig A3. Coincident Features.

A.3 Conditions for a proper Location Triad

The conditions for proper locations are summarised by the inequalities:

(i) $\hat{g} \cdot \bar{h} \neq 0$,

(ii) $\bar{h} \cdot (\hat{f} \times \bar{p}) \neq 0$.

(i) and (ii) imply that

(a) the S-loc must not coincide with the H- or P-locs,

(b) the line joining the points of application of the S- and H-locs must not be perpendicular to the line of action of the H-loc, and

(c) the normal to the plane of the P-loc and the lines joining the S-loc and the H- and P-locs must not be coplanar.

There is no restriction on the position of the R-point. Occasionally, it is useful to employ limiting cases of (i) and (ii), an example being illustrated in Fig. A3. The P- and H-locs, and the R-point are coincident, the normal to the P-loc is perpendicular to the line of the H-loc, the line of the H-loc lies along the lines joining them with the S-loc. Since the H- and P-locs are coincident, condition (ii) is not observed. All the location matrices will have infinite coefficients. Use can be made of this system, however, if the H-loc is slightly displaced, so that $\bar{h}$ becomes $\bar{h} + \bar{e}$. In the case illustrated:

$\bar{r} = \bar{p} = \lambda \hat{g}$ where λ is a scalar

and $f_1 = f_2 = g_2 = g_3 = 0; \quad f_3 = g_1 = 1.$

$$P_{ij} = -\frac{\epsilon_{rit}(h_r + e_r)\,\lambda g_t\, f_j}{\epsilon_{abc}(h_a + e_a) f_b\, \lambda\, g_c}$$

$$= -\frac{\epsilon_{rit}\, e_r\, g_t\, f_j}{\epsilon_{abc}\, e_a\, f_b\, g_c}$$

$$P_{ij} = -\frac{\epsilon_{ri1}\, e_r\, f_j}{e_2}$$

$$\text{i.e. } P_{ij} = \begin{cases} -\dfrac{\epsilon_{ri1}\, e_r}{e_2} & \text{for } j = 3 \\ 0 & \text{for } j \neq 3 \end{cases}$$

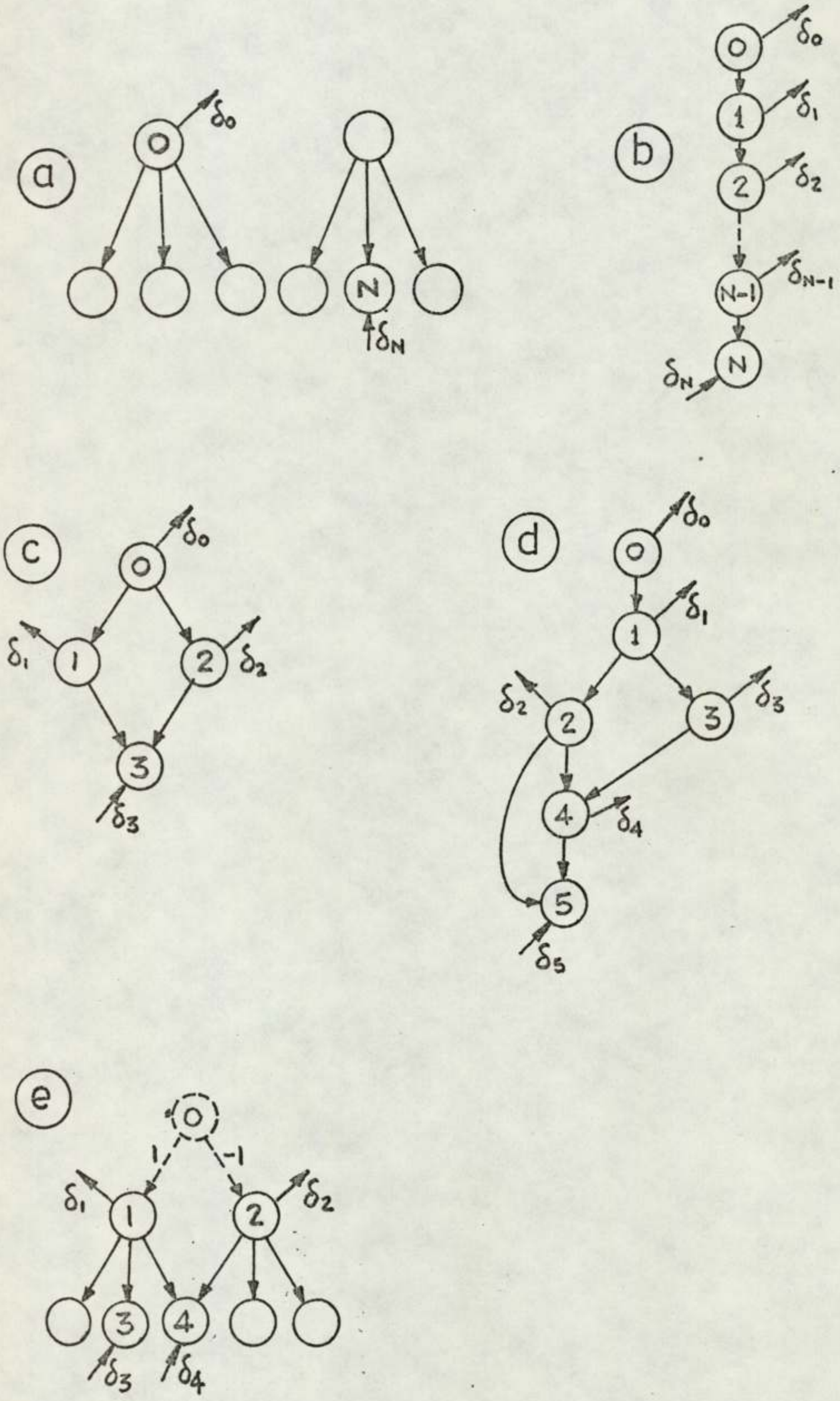


Fig A4. Network Paths.

If the H-loc is perturbed in the plane of the P-, S-locs and R-point, then

$$P_{ij} = \begin{cases} -1 & \text{for } i = j = 3 \\ 0 & \text{for } i,j \neq 3 \end{cases}$$

The equation for H_{ij} may be written in the form:

$$H_{ij} = - \frac{g_d p_d P_{ij} - g_e P_{ie} p_j + g_f r_f \delta_{ij} - g_i r_j}{g_h h_h}$$

$$= - \frac{g_1 p_1 P_{ij} - g_1 P_{i1} p_j + g_1 r_1 \delta_{ij} - g_i r_j}{g_1 h_1}$$

or $$H_{ij} = - \frac{g_1 p_1 P_{ij} + g_1 r_1 \delta_{ij} - g_i r_j}{g_1 h_1}$$

which reduces to

$$H_{ij} = \begin{cases} -1 & \text{for } i = j = 2 \\ 0 & \text{for } i,j \neq 2 \end{cases}$$

Similarly,

$$S_{ij} = \begin{cases} -1 & \text{for } i = j = 1 \\ 0 & \text{for } i,j \neq 1 \end{cases}$$

This limiting case depends not only on the coordinates of the locations, but also on the direction of the perturbation. Similar analyses may be performed for all the cases in which this technique is used.

This system is useful in passing three orthogonal displacements to a result point, and may be used in the generation of a general parallelepipedal tolerance zone.

A.4 Path matrix products

(a) Disjoint case Fig. A4a.

A displacement δ_N is applied at node N. It is required to find the displacement δ_0 at node 0. Since node 0 is not located, even indirectly, on node N, the equation

$\delta_0 = F_{ON}\,\delta_N$ requires that in this case where no path exists between nodes 0 and N, $F_{ON} = \overline{0}$.

(b) Simple path Fig. A4b.

A displacement δ_N is applied to node N, resulting in a displacement δ_{N-1} at node N-1. δ_{N-1} will cause a displacement at node N-2 and so on along the path joining nodes N and 0.

$$\delta_0 = F_{01}\,\delta_1$$
$$\delta_1 = F_{12}\,\delta_2$$
$$\vdots$$
$$\delta_{N-1} = F_{N-1\ N}\,\delta_N$$
$$\delta_0 = F_{01}\,F_{12}\,\cdots\,F_{N-1,N}\,\delta_N$$

In the evaluation of a simple path matrix, the matrices corresponding to the links on the path are multiplied.

(c) Multiple paths Fig. A4c.

A displacement δ_3 is applied at node 3 and the displacement δ_0 is required.

$$\delta_1 = F_{13}\,\delta_3$$
$$\delta_2 = F_{23}\,\delta_3$$
$$\delta_0 = F_{01}\,\delta_1 + F_{02}\,\delta_2$$
$$\delta_0 = (F_{01}\,F_{13} + F_{02}\,F_{23})\,\delta_3 = F_{03}\,\delta_3$$

The path matrix products are evaluated separately and added to give the nett path matrix product F_{03}.

(d) Multiple paths Fig. A4d.

A displacement δ_5 is applied at node 5 and again the displacement δ_0 is required.

$$\delta_0 = F_{01}\,\delta_1$$
$$\delta_1 = F_{12}\,\delta_2 + F_{13}\,\delta_3 \quad \text{by superposition.}$$

$$\delta_2 = F_{24}\,\delta_4 + F_{25}\,\delta_5 \text{ by superposition.}$$

$$\delta_3 = F_{34}\,\delta_4$$

$$\delta_4 = F_{45}\,\delta_5$$

Back-substitution gives

$$\begin{aligned}\delta_0 &= (F_{01}\,F_{12}\,F_{24}\,F_{45} + F_{01}\,F_{12}\,F_{25} + F_{01}\,F_{13}\,F_{34}\,F_{45})\,\delta_5\\ &= F_{05}\,\delta_5.\end{aligned}$$

Again, path matrix products are evaluated separately and added to give the nett path matrix product F_{05}. In this case, it is most convenient to calculate partial matrix products at each node moving down from the R-point.

At node 1, product = F_{01}

At node 2, product = $F_{01}\,F_{12}$

At node 3, product = $F_{01}\,F_{13}$

At node 4, product = $F_{01}\,F_{12}\,F_{24} + F_{01}\,F_{13}\,F_{34}$

At node 5, product = $F_{01}\,F_{12}\,F_{25} + F_{01}\,F_{12}\,F_{24}\,F_{45} + F_{01}\,F_{13}\,F_{34}\,F_{45}$

(e) Relative displacements Fig. A4e.

A displacement δ_4 is applied at node 4. The displacement of node 1 relative to that of node 2 is required.

A method is needed to evaluate

$$\delta_1 - \delta_2 = (F_{14} - F_{24})\,\delta_4 \text{ conveniently.}$$

The method which will be used is to attach a dummy node 0 to nodes 1 and 2 as shown in the figure. If F_{01} is set to the unitary matrix δ_{ij} and F_{02} is set to $-\delta_{ij}$, then the result will be obtained by the methods earlier described in (a) - (d).

$$\begin{aligned}\delta_0 &= ((\delta_{ij}\,F_{14}) + (-\delta_{ij}\,F_{24}))\,\delta_4\\ &= (F_{14} - F_{24})\,\delta_4.\end{aligned}$$

The method also works for the case where there is no path between the input node and one, or two of the output nodes. This case is illustrated in Fig. A4e, the input being applied at node 3.

Some useful results will now be derived.

A.5 Matrix Rank

A.5.1 The rank of a P-matrix

$$\text{Since } P_{ij} = \frac{\epsilon_{rit}\, h_t\, r_r\, f_j}{\epsilon_{abc}\, h_a\, f_b\, p_c}$$

$$\overline{P}_1 = \frac{\epsilon_{r1t}\, h_t\, r_r}{\epsilon_{abc}\, h_a\, f_b\, p_c}\ \overline{f}$$

$$\overline{P}_2 = \frac{\epsilon_{r2t}\, h_t\, r_r}{\epsilon_{abc}\, h_a\, f_b\, p_c}\ \overline{f}$$

$$\overline{P}_3 = \frac{\epsilon_{r3t}\, h_t\, r_r}{\epsilon_{abc}\, h_a\, f_b\, p_c}\ \overline{f}\ .$$

P is of rank 1 at most, since each row vector is a multiple of vector $\overline{f}$.

A.5.2 The rank of an H-matrix

It has been proved (equation xi) that:

$$H_{ij} = \frac{\epsilon_{rit}\, h_r\, r_t\, (g_d p_d f_j - g_e f_e p_j) - K(g_f r_f\, \delta_{ij} - g_i r_j)}{Kg_g\, h_g}$$

Multiplying each element in(xi) by r_i and considering each term in the numerator in turn:

(a) $\epsilon_{rit}\, h_r\, r_t\, r_i$ may be summed over i.

$$\sum_i \epsilon_{rit}\, h_r\, r_t\, r_i = -\sum_i \epsilon_{rit}\, h_r\, r_i\, r_t,$$

if i and t be interchanged.

$$\therefore \sum_i \epsilon_{rit}\, h_r\, r_t\, r_i \equiv 0$$

(b) $(g_f\, r_f\, r_i\, \delta_{ij} - g_i\, r_i\, r_j)$ may be summed over i.

$$\sum_i (g_f\, r_f\, r_i\, \delta_{ij} - g_i\, r_i\, r_j) = \sum_i (g_f\, r_f\, r_j - g_i\, r_i\, r_j) = 0.$$

$$\text{Hence}\quad r_1\, \overline{H}_1 + r_2\, \overline{H}_2 + r_3\, \overline{H}_3 \equiv 0$$

and an H-matrix is of rank at most 2.

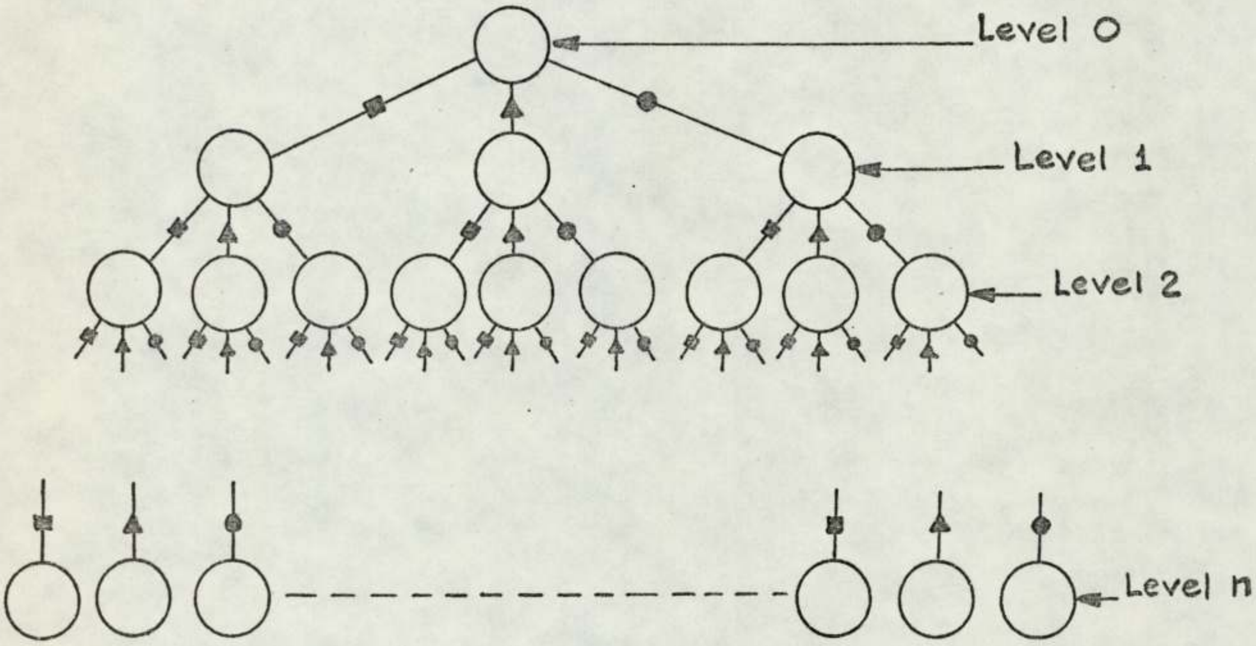


Fig A5 A Balanced Tree.

A.5.3 The rank of an S-matrix

The rank of an H-matrix is at most 2 and of a P-matrix at most 1. In the special case $\bar{r} = \bar{0}$ (i.e. S-loc and R-point coincident), $\bar{p}, \bar{h} \neq \bar{0}$ then both $\bar{P}$ and $\bar{H}$ are of rank 0 since $\bar{P} = \bar{H} = \bar{0}$.

From equation (xii)

$$S_{ij} = -(P_{ij} + H_{ij} + \delta_{ij}) = -\delta_{ij} \text{ if } \bar{P} \text{ and } \bar{H} \text{ are of null rank.}$$

S_{ij} is of rank 3 since all row vectors of δ_{ij} are linearly independent ($\det(\delta_{ij}) = 1$).

An S-matrix is of rank at most 3.

A.5.4 The rank of a matrix product

A result proved in ref. 14 is:

'the product AB has a rank not greater than the rank of either factor.'

The rule may be applied to path matrix products to give the following conclusions:

(a) if a path contains the node of a P-loc, then the path matrix product is of rank at most 1,

(b) if a path contains the node of an H-loc, then the path matrix product is of rank at most 2, and

(c) if a path contains the node of an S-loc, then the path matrix product is of rank at most 3.

The rank of the path matrix product is determined by the most stringent of conditions (a), (b) and (c) which can be applied to the path.

A.5.5 Relative Numbers of P-, H- and S-matrices

It will assist in assessing algorithms used in processing 'trees' if the relative proportions of matrix ranks are known. For the fully balanced ternary tree shown in Fig. A5 the following results apply:

(a) At level n, there are 3^n nodes.

(b) If, at level n-1 there are N_{n-1} paths with matrix products of rank 1 at most, then there will be $3^{n-1} + 2N_{n-1}$ paths with similar rank at level n.

$N_n = 3^{n-1} + 2N_{n-1}$.

Solving this difference equation with $N_1 = 1$ gives

$N_n = 3^n - 2^n$ paths with product matrices of at least 1 at level n. Summing over n levels gives:

$T_n = \frac{(3^{n+1} - 3)}{2} - (2^{n+1} - 2)$ - the total number of paths with product matrix rank at least 1 down to level n.

(c) At each level there will be one path with path matrix product of rank at least 3. Over n levels there will be a total of n such paths.

(d) The total number of paths down to level n is $\frac{(3^{n+1} - 3)}{2}$

(e) The total number of paths with product matrices of rank at least 2 down to level n may be found by subtracting the number of paths with product matrices of ranks 1 and 3 at most from the total number of paths down to level n. This is $2^{n+1} - 2 - n$.

(f) For n levels, there will be $\frac{3^{n+1} - 1}{2}$ nodes.

Tabulating values up to 6 levels:

	Path Product	Matrix	Maximum Rank	No. of
Level	1	2	3	Nodes
1	1	1	1	4
2	6	4	2	13
3	25	11	3	40
4	90	26	4	121
5	301	57	5	364
6	966	120	6	1093

For large n, rank 1 matrices will predominate and the ratio of rank 1 matrices to rank 2 matrices $= O(\frac{1}{2}(\frac{3}{2})^{n+1})$. For the system designed, about 200 nodes is considered as a maximum. The number of levels of the corresponding balanced ternary tree will lie between 4 and 5 giving a ratio of approximately 4.5 rank 1-at-most matrices to 1 rank 2-at-most matrices. The number of rank 3-at-most matrices will be negligible.

Although a practical location 'tree' is unlikely to be a balanced tree, it seems reasonable to assume that these ratios will be approximately correct and will serve as a useful measure in the assessment of algorithms used for processing the 'tree'.

A.6 The maximum output displacement (sensitivity coefficient)

In the discussion which follows, the matrix results listed below will be used. They are proved in refs. G2 and G8.

(i) If A* denotes the transpose of matrix A, then the product A*A is positive definite, ref. G8, p.46.

(ii) If A is positive definite, then all its eigenvalues are positive, ref. G8, p.46.

(iii) If the eigenvalues of A*A are λ_i, then the eigenvalues of $(A^{-1})^*A^{-1}$ are $({}^1/\lambda_i)$, the associated eigenvectors being identical, ref. G8, p.43.

Generally, the output tolerance $\bar{\delta}$ resulting from an input displacement $\bar{\epsilon}$ applied to a path with matrix F is given by:

$$\bar{\delta} = F\,\bar{\epsilon}\,.$$

A general spherical input tolerance zone of radius r may be written

$$\bar{\epsilon}^*\bar{\epsilon} = r^2.$$

In particular, if F be of rank 3, then it will have an inverse, say G, and

$$G\bar{\delta} = \bar{\epsilon}$$

or $\quad \bar{\epsilon}^* = \bar{\delta}^*G^*$

and $\quad \bar{\epsilon}^*\bar{\epsilon} = \bar{\delta}^*G^*G\bar{\delta}$

or $\quad \bar{\delta}^*G^*G\bar{\delta} = r^2$

The product G*G is symmetric and positive definite by result (i), and so may be diagonalised giving the relation:

$$\bar{\delta}^*L\bar{\delta} = r^2,$$

L being the diagonal matrix $L_{ij} = \begin{cases} \lambda_i, i = j \\ 0 , i \neq j \end{cases}$.

The λ_i are the eigenvalues of L.

Since G*G is positive definite, $\lambda_i > 0$ and the relation describes an ellipsoid:

$$\lambda_1 \delta_1^2 + \lambda_2 \delta_2^2 + \lambda_3 \delta_3^2 = r^2.$$

The maximum axis of the ellipsoid will be given by:

$$\delta_{max} = \frac{r}{\sqrt{\lambda}}$$

where $\lambda = \min(\lambda_i)$ in a direction given by the associated eigenvector of λ. This will be the maximum output displacement.

The matrix result (iii) simplifies the calculation considerably. If the eigenvalues of G*G are λ_i, then the eigenvalues of F*F are $\frac{1}{\lambda_i}$. Also the maximum eigenvalue of F*F is the minimum eigenvalue of G*G. The corresponding eigenvectors are identical.

The procedure for finding the maximum output displacement, and its direction, may be summarised as follows.

If a spherical input tolerance zone radius r be applied at the base of a location chain with path matrix F (F non-singular), then the maximum output displacement is given by

$$\delta_{max} = \sqrt{\lambda}\, r,$$

where λ is the dominant eigenvalue of the product F*F. The direction of δ_{max} will be given by the corresponding eigenvector. For a more detailed development, see ref. G16.

The result has been derived for a rank 3 matrix but it can be shown to be generally true for ranks 2 and 1. The argument is broadly the same, but since ranks 2 and 1 3 x 3 matrices are singular, they have no inverse and it is necessary to consider the natural or general inverse. Ref. G2 contains a concise description of the use of general inverses, while ref. G3 is completely devoted to them.

APPENDIX B

NOTES ON ALGORITHMS

B.1 Evaluation of Eigenvalues

B.1.1 Choice of algorithm

The methods available are:

(i) to obtain a closed solution by expanding the characteristic polynomial which will, in the most general case, be a cubic;

(ii) to use an iterative method, such as the power method, or

(iii) to use a transformation method, e.g. Householder's method.

The factors governing the choice of the method are:

(i) all matrices are of order 3,

(ii) the bulk of the matrices involved will be of rank 1,

(iii) the dominant eigenvector is also required,

(iv) the number of matrices is likely to be large,

(v) accuracy of solution is not extremely critical - accuracy of 1 in 10^5 should suffice, and

(vi) matrices are positive definite and symmetric.

The power method was chosen since

(i) in this case, it may be used generally for all ranks,

(ii) the dominant eigenvalue is obtained naturally and

(iii) the corresponding eigenvector is obtained at the same time.

B.1.2 Description of the algorithm

The power method is detailed in ref. G8 and an error analysis provided in ref. G9, but the method will be briefly described.

If the dominant eigenvalue of matrix A is required, then the computing scheme is

$$y^{(P)} = Ay^{(P-1)} \; ; \; y^{(0)} \text{ arbitrary.}$$

The $y^{(N)}$ are successive iterates and are vectors. It is customary to select as initial vector $y^{(0)} = \{1,1,1\}$. The ratios of corresponding components of successive vectors $y^{(K)}$ and $y^{(K-1)}$ will converge to λ, the dominant eigenvalue of A, if the method is successful. Further, each iterate $y^{(K)}$ is an estimate of the corresponding unnormalised eigenvector. The rate of convergence depends on the ratio between the dominant and sub-dominant eigenvalues. Methods are available for accelerating convergence.

Since the majority of the matrices will be of rank 1, it would seem advantageous to select as initial vector $y^{(0)} = \left\{A_{11}, A_{21}, A_{31}\right\}$ or any other column vector of A. The dominant eigenvalue would then be obtained in one iteration only, since the corresponding eigenvector (unnormalised) is a column vector of A. Unfortunately, it is common for one or more of the column vectors in location matrices to consist of all zero elements. This is a particular case of a general problem in selecting the initial vector for use with the power method. If the initial vector is exactly the eigenvector corresponding to an infra-dominant eigenvalue, then the method will yield that eigenvalue in one iteration. Clearly, an answer obtained in one iteration should be viewed with suspicion and the calculation repeated with the original initial vector slightly perturbed. This problem is not mentioned in most of the standard texts (ref. G8 is an exception) nor is it considered in any of the programs described in the less theoretical books on numerical methods. Unfortunately, the case of a location matrix having an infra-dominant eigenvector $\left\{1,1,1\right\}$ is not uncommon in practice. In view of the fact that the bulk of the matrices processed will be of rank 1 and single iteration answers will be common, it is considered that repeating the calculation with a perturbed initial vector would be an intolerable overhead of time. The initial vector $\left\{\pi/4, e, \log_e 10\right\}$ is used in the program. Even though there is a remote possibility that these values will give an incorrect answer, it is considered worthwhile to use them because of the saving in time.

Since the successive $y^{(K)}$ are unnormalised, their components tend to increase rapidly and it is necessary to normalise at each stage. This is done by dividing each element of $y^{(K)}$ by $\|y^{(K)}\|_\infty$, where $\|y^{(K)}\|_\infty$ is the maximum value of $|y_i^{(K)}|$ over all i.

A test program was written to check the efficiency of various programs for calculating eigenvalues. A random number generator was used in conjunction with a method of generating matrices of prescribed eigenvalues and eigenvectors which was found in ref. G10.

B.1.3 Generating test data

Batches of 40 matrices were generated in the following manner:

(i) The constitution of each batch was

1 rank 3 matrix,

7 rank 2 matrices,

32 rank 1 matrices.

These proportions were approximately those calculated in Appendix A, section A.5.5 for 200 nodes.

(ii) A matrix S was constructed with column vectors mutually orthogonal, but otherwise random. If p_i denote the i-th random number generated, then

$$S_{i1} = \left\{ p_1 \;, p_2 \;, p_3 \right\}$$

$$S_{i2} = \left\{ p_3p_4, p_3p_5, -(p_1p_4 + p_2p_5) \right\}$$

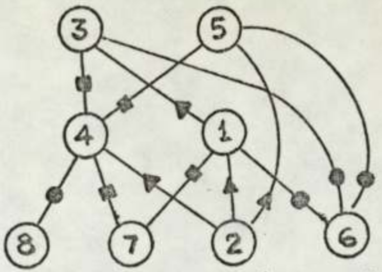
$$S_{i3} = \left\{ (p_1p_2p_4 + p_2^2p_5 + p_3^2p_5), -(p_3^2p_4 + p_1^2p_4 + p_1p_2p_5), - (p_1p_3p_5 - p_2p_3p_5) \right\}$$

(iii) Since column vectors of 3 are linearly independent, S^{-1} exists and is calculated directly.

(iv) The diagonal matrix A is constructed where the diagonal elements are the eigenvalues.

$$A_{ij} = \begin{cases} p_5 + i & \text{for } i = 1 \text{ to the rank of A, } i = j \\ 0 & \text{otherwise} \end{cases}$$

(v) The matrix product $S^{-1}A\,S$ will have the required properties (ref. G2, p.99).



Unsorted "tree"

<1,7,2,6> ; <2,0,0,0> ; <3,4,1,6> ; <4,7,2,8> ; <5,4,2,6> ; <6,0,0,0> ; <7,0,0,0> ; <8,0,0,0>

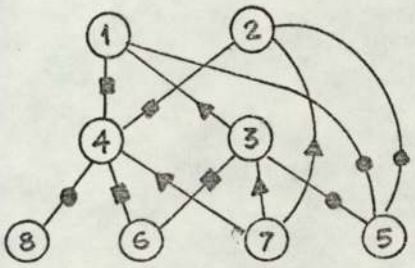
Partial Ordering Relations.

topq	1	2	3	4	5	6	7	8	botq
–	0	0	0	0	0	0	0	0	–
–	1	3	0	2	0	3	2	1	–
3	1	3	◇ 5	2	◇ -1	3	2	1	5
3	◇ -1	3	◇ 5	1	◇ 1	2	2	1	1
5	◇ -1	3	○ 1	1	◇ 1	2	2	1	1
	⋮		⋮		⋮		⋮	⋮	
8	○ 3	○ 7	○ 1	○ 4	○ 2	○ 5	○ 6	◇ -1	8
-1	○ 3	○ 7	○ 1	○ 4	○ 2	○ 5	○ 6	○ 8	-1

☐ Count

◇ Queue

○ Sort Index



Sorted "tree"

Fig B1 Topological Sort Algorithm.

This rather elaborate procedure was used so that the batches of matrices should be as representative as possible of those occurring in practice.

A separate program was used for the generation of batches of matrices, so that the time taken in their generation could be separated from the time taken for calculating their dominant eigenvalues and eigenvectors.

B.2 Topological Sort Algorithm

The algorithm used was a modification of that described in detail in ref. G13. Its action is shown diagrammatically in Fig. B1.

Data used is selected from the 'tree' description supplied as input. If the 'tree' contains N nodes, each allotted a distinct integer in the range 1-N, and if M of these are non-leaf nodes, then there will be M partial ordering relations of the form $\langle R,P,H,S \rangle$, which are needed by the algorithm. Each integer R represents an R-point; and P, H and S are the node numbers of the corresponding P-, H- and S-locs. A further N-M relations of the form $\langle R,0,0,0 \rangle$ are also available - the R, in this case, representing a leaf-node and 0 being a notional earth-node.

A vector, length N is used in three guises. It is used initially to count the direct predecessors of each node; it is used as a queue for unprocessed nodes and, finally, it is used as an index to show topological sort order. Two pointers are used to point to the head and tail of the queue of unprocessed nodes. The sort process is as follows:

(i) Zeroise count [i] for i = 1 to n.

(ii) Count the predecessors of each node.

(iii) All nodes with zero count are root nodes. Set up a queue for these; if there are no root nodes, then the structure is incorrect.

(iv) Select the first item on the queue. If its three successor nodes are not earth nodes, then their counts are reduced by one. If any count becomes zero, then the corresponding node is ready for processing and its index number is queued.

(v) The node at the head of the queue is deleted and it is next in topological order. It is allocated the next index number.

(vi) If the queue is empty, and all nodes have been processed, then the sort has been successfully accomplished. If the queue is empty and all nodes have not been processed, then the structure is incorrect. If the queue is not empty, then the algorithm is continued from stage (iv).

It is claimed in ref. G13 that the algorithm is near-optimal. Processing time is of the order of C_1 M + C_2 N where C_1 and C_2 are constants, and storage is used economically.

It is possible to re-order the records during the algorithm but this was not done for various reasons. Firstly, it is considered good practice to divorce the data validation stage of a program from the data processing stage both as a policy and because, for some configurations, it might be necessary to perform these operations by separate programs. Also, subsequent programming is neater and more easily tested if a structured approach is used.

B.3 Inverting the Topological Sort Index

It is possible to refer to each record indirectly using the sort index, but it saves much processing time if the records are re-sorted. Re-sorting may be performed in several ways; the familiar dilemma of time taken versus extra storage required applies in this case, as in all sorting problems. The following is a sample of the methods possible:

Given a sequence of records $R_1, R_2, \ldots R_N$ and a sort index $T_1, T_2, \ldots T_N$ where T_K shows the required sort position of record R_K,

(a) Perform an exchange sort, repeatedly passing through the list R, and exchanging $R(T_K)$ until no exchanges are necessary,

(b) Invert the sort-index T by $I(T_K) = K$ to form another sort-index $I_1, I_2, \ldots I_N$. I_K shows the number of the record which is to be placed into position K. Records can now be exchange-sorted in one pass by:

```
temp 1: = $R(I_K)$ ; $R(I_K)$ : = $R_K$ ;
temp 2: = $R(T(I_K))$ ; $R(T(I_K))$ : = temp 1
$R(I_K)$ : = temp 2.
```

(c) Invertthe sort-index in situ, saving setting up an extra sort-index vector. The algorithm used may be found in refs. G1 and G13. In particular, ref. G13 quotes two algorithms for this purpose, but one, though more elegant, may be discounted since it is less efficient. Records may then be exchange-sorted in one pass as in (b).

It was decided in the interests of storage economy, to select method (c). The method is analysed in ref. G13 and the processing time is of the order of C N where C is a constant. Exchange-sorting is performed in one pass and so it appears that method (c) is better on a processing time basis than (a) since normal exchange-sorting time is proportional to N^2, and better from storage economy considerations than method (b) since no inverted sort-index is required.

B.4 Processing the 'tree'

From tests conducted on the prototype program, this section of the processing is easily the most lengthy. Not only must all the paths from input location to R-point be traversed but also matrices for each link on the paths must be calculated. Although it is a straightforward matter to minimise the matrix calculation time by writing efficient code, path traversal is a difficult problem. For N nodes Warshall's algorithm (ref. G1) requires an N x N matrix and is out of the question because of the storage limitations. Alternatively, linked lists may be held for the immediate predecessors of each node - this also needs an unacceptable overhead of store. This problem is largely overcome by an elegant algorithm quoted in ref. G15 but this is not general and depends on the relative number of leaf nodes and nodes with multiple antecedents. A possible method is to use a marking algorithm, tagging in some way all edges on the paths between input location and R-point using a stack - this again needs extra store.

The method used in the prototype program was crude but straightforward, priority being given to economy of store. All matrices were evaluated and multiplied, and were added at junctions for all nodes whose indices lay between those of the input node and the R-point.

The problem of traversing paths of structures of this kind occurs in many diverse applications and it seems that the algorithms available involve considerable storage overhead. For the configuration considered, it is unavoidable that this brute force method should be used in preference to one more sophisticated but requiring more store.

APPENDIX C

ALLOCATION OF TOLERANCES

C.1 The Allocation of Tolerances - sure-fit

If there are m critical clearances D_i $(i = 1, 2, \ldots m)$ in an assemblage and each is affected by one or more of the set of tolerances x_j $(j = 1, 2, \ldots n)$, then m linear inequalities may be written:

$$S_{ij}\, x_j \leqslant D_i \qquad i = 1, 2, \ldots m; \quad j = 1, 2, \ldots n.$$

The constants S_{ij} are sensitivity coefficients, S_{1k}, for instance being the effect at the critical clearance D_1 of a unit tolerance at the point where x_k acts. In the case where a critical clearance D_1 does not depend on tolerance x_k, then $S_{1k} = 0$.

Subject to these constraints, it is required that the total cost of maintaining the critical clearance should be minimised. Clearly, the more precise a part is made, the higher will be the unit cost but there seems to be some disagreement in the references as to the exact form of the cost-tolerance relationship. Models suggested are -

(i) $\text{cost} = kx^{-2}$ where k is a constant (ref. A2)

(ii) $\text{cost} = kx^{a}$ where k and a are constant and $a < 0$ (ref. A1)

(iii) $\text{cost} = k + le^{mx}$ where k, l and m are constant (ref. A8)

Although (iii) is the most widely used model and is used in several American papers, where it is called Speckhart's Exponential Model, (ii) appears to have been based on rather more solid experimental foundations. Studies of data on the cost-tolerance relationship for various manufacturing processes were analysed and best curves fitted by the method of least squares. No experimental basis is described for model (iii), the author baldly stating that the expression fits cost-tolerance data 'very well'. Model (i) is comprehensively (if rather unfairly) discredited in ref. A1. The evidence would suggest that the most suitable model is (ii) and this is used in the following development.

The cost of maintaining tolerance x is given by the expression:

$$C = kx^{a}$$

$-0.8 < a < -0.4$ and depends on the process. k depends on the shape and size of the component. Values of k are not critical, only the relative values being of significance.

The cost of maintaining the n tolerances x_i is given by:

$$C = \sum_{i=i}^{n} k_i x_i^{a_i}$$

The Allocation of Tolerances problem may now be stated in its full form:

$$\text{Minimise } C = k_1x_1^{a_1} + k_2x_2^{a_2} + \ldots + k_nx_n^{a_n}$$

Subject to constraints:

$$\begin{array}{ccccccc} S_{11}x_1 & + & S_{12}x_2 & + \ldots + & S_{in}x_n & \leqslant & D_1 \\ \vdots & & \vdots & & \vdots & & \vdots \\ S_{m1}x_1 & + & S_{m2}x_2 & + \ldots + & S_{mn}x_n & \leqslant & D_m \end{array}$$

There are also implicit constraints

$$x_1, x_2, \ldots x_n > 0$$

and there may also be constraints due to practical considerations:

$$x_i \geqslant e_i$$

where e_i is the lowest practicable bound on x_i.

This is an optimisation problem with a non-linear objective function and linear constraints.

C.2 The Allocation of Tolerances - statistical-fit

The development of equations for allocating tolerances on a statistical-fit basis follows broadly the lines of that for sure-fit basis. In this case, however, it is usual to assume that the tolerance distributions follow a Gaussian distribution and often the tolerance range is taken as the nominal position plus or minus three standard deviations, 99.7% of the parts produced then having the dimension within the allowed tolerance range. The cost is that of maintaining the dimension within plus or minus three standard deviations around the nominal dimension. Using the properties of the Gaussian distribution, it may be established that the Allocation of Tolerances problem may be stated:

Minimise $C = K_1x_1^{a_1} + k_2x_2^{a_2} + \dots k_nx_n^{a_n}$

Subject to constraints:

$$\begin{array}{ccccccc} S_{11}^2x_1^2 & + & S_{12}^2x_2^2 & + \dots + & S_{1n}^2x_n^2 & \leqslant & D_1^2 \\ \vdots & & \vdots & & \vdots & & \vdots \\ S_{m1}^2x_1^2 & + & S_{m2}^2x_2^2 & + \dots + & S_{mn}^2x_n^2 & \leqslant & D_m^2 \end{array}$$

With implicit constraints:

$x_1, x_2, \dots x_n > 0$

and possibly practical minima on tolerances:

$x_i \geqslant e_i$

All constants and variables are as defined in C1.

This is an optimisation problem with a non-linear objective function and non-linear constraints.

C.3 Solution of the Allocation Problem

Refs. A1 and A8 both use the classical technique of Lagrange multipliers in order to solve the Allocation Problem, obtaining what both call lambda equations, which are solved by an iterative technique such as Newton's method. This is a straightforward technique for one critical dimension, but in the more general case, where more than one critical dimension is concerned, the method discussed in ref. A1 requires considerable manual work before submission to the computer program and is only applicable to the sure-fit case. Ref. A8 uses an iterative procedure but it is stated that there is no guarantee that the procedure described will converge.

Several general methods of non-linear optimisation are described in ref. G4 and it would seem that the methods described in refs. A1 and A8 have been superseded by later techniques. It is probable that these are more suitable methods and this particular aspect of tolerance analysis merits investigation. Possibly different methods would be required for the sure-fit and statistical-fit cases, since it is stated in ref. G4 that a universal optimizer does not exist and a method suitable for linear constraints (sure-fit) may not be adequate for quadratic constraints (statistical-fit). This is, however, outside the scope of this thesis.

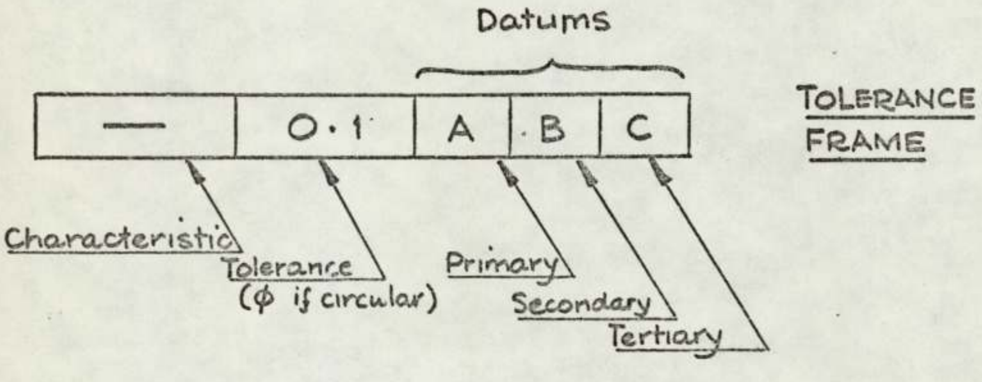
APPENDIX D

STANDARD CASES

BS 308 . PT 3

Symbols used in tolerance frames

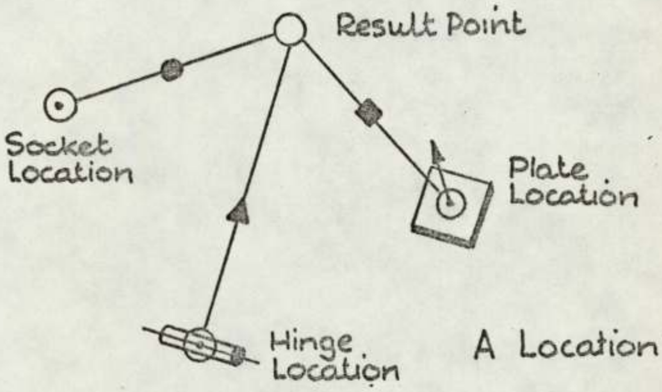
—	. . . . Straightness	
//	. . . . Parallelism	
⊥	. . . . Squareness	
∠	. . . . Angularity	
⌖	. . . . Position	CHARACTERISTICS
◎	. . . . Concentricity	
⌯	. . . . Symmetry	
▱	. . . . Flatness	
○	. . . . Roundness	
⌭	. . . . Cylindricity	
⌒	. . . . Profile of a line	
⌓	. . . . Profile of a surface	
↗	. . . . Run-out.	



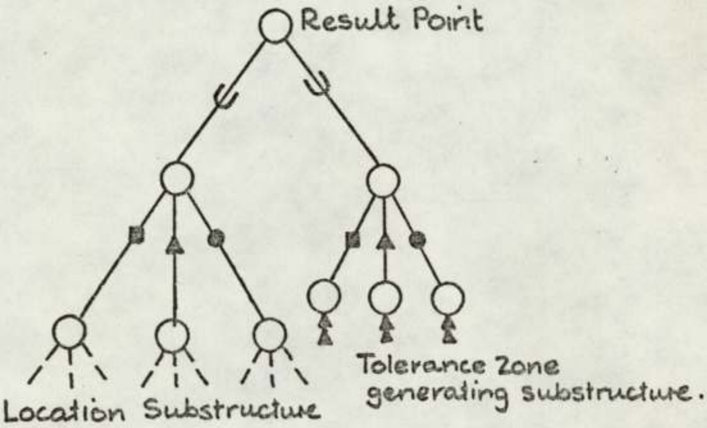
. . . . Datum.

Ⓜ . . . . Maximum Metal Condition.

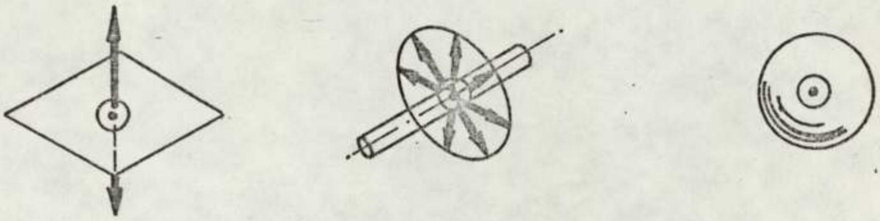
$\boxed{30}$ . . . . True Position.



A Location Triad



General Displacement at a Point



Linear Tolerance at a plate

Circular Tolerance at a hinge

Spherical Tolerance at a socket.

Fig D1 Displacement at a Result Point.

D.1 The Displacement at a Point - Fig. D.1.

A point must be located on a location triad of plate, hinge and socket.

The displacement at a point is made up of two components:

(a) the extrinsic displacement which is due to displacements of the features on which it is located, and

(b) the intrinsic displacement which is due to the permitted tolerance at the point.

Extrinsic tolerance is passed to the point by the location sub-system upon which it depends; intrinsic tolerance must be applied directly at the point or indirectly through a tolerance generating sub-structure.

(i) If the result point is on a plane, then a linear tolerance normal to the plane may be applied directly.

(ii) If the result point lies on a line, then a circular tolerance in a plane perpendicular to the line may be applied directly.

(iii) If the result point is a general point, then a spherical tolerance may be applied directly.

The majority of tolerance situations will be covered by (i), (ii) and (iii) since these are the positional tolerances recommended in BS 308, but occasionally a rectangular or parallelepipedal tolerance zone is quoted; and it is sometimes necessary to generate this by using a tolerance generating sub-structure. The method will be described later.

It is necessary to use bi-lateral tolerances when this method is applied. For most systems, the tolerance is small in comparison with nominal dimensions and so the nominal dimension does not need to be altered. For instance, a tolerance of $1.000 {}^{+.010}_{-.000}$ may be considered as $1.000 {}^{+.005}_{-.005}$. The reason for this is that only clearances are

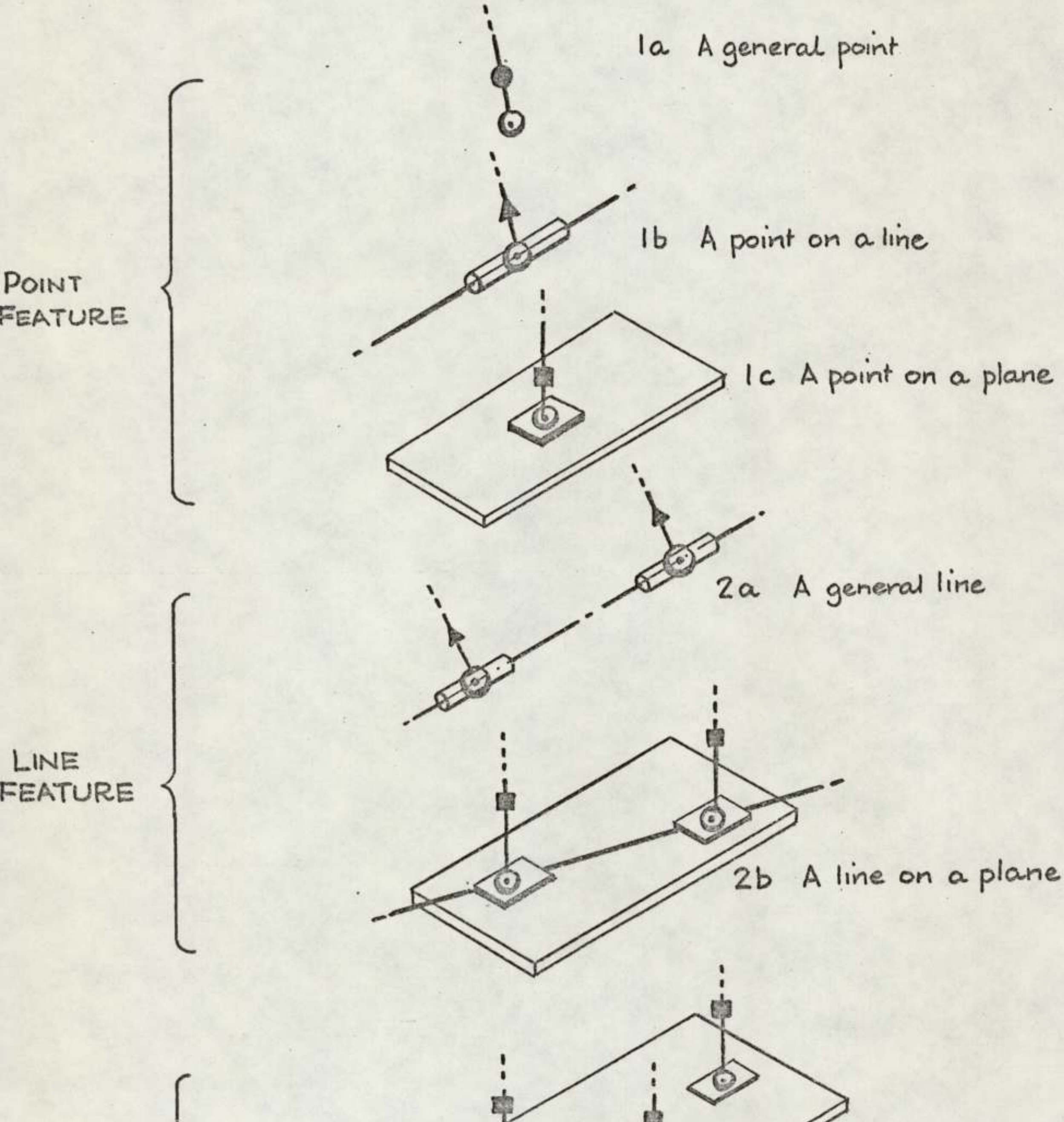


Fig D2 Definition of Features.

analysed in this method. Nominal dimensions are only used in intermediate calculations and so do not necessarily need to be extremely accurate. Of course, if it is preferred the dimension can be quoted as 1.005 $\pm$ .005, but this should make very little difference to the results obtained.

D.2 Definition of Features - Fig. D.2.

1. A point feature.
   - (a) A general point is described by a socket.
   - (b) A point on a line is described by a hinge.
   - (c) A point on a plane is described by a plate.
2. A line feature.
   - (a) A general line is described by two hinges.
   - (b) A line on a plane is described by two plates.
3. A plane feature.
   - (a) A plane is described by three plates.

In order to define a feature, the points at which the locations are centred may be chosen arbitrarily subject only to the following restrictions:

(a) in order to define a line, the points must not be coincident, and

(b) in order to define a plane, the points must not be collinear.

Plates defining a plane must have normals parallel with the normal to the plane, and hinges defining a line must have directions along the line.

D.3 General Points on Lines and Planes

A general point on a line or a plane may be considered as being located on the line or plane. Since a point is located on a plate, hinge and socket in the basic location triad, the general point cannot be located directly on the line or plane, which are defined by two hinges or plates, or three plates respectively. This problem is resolved

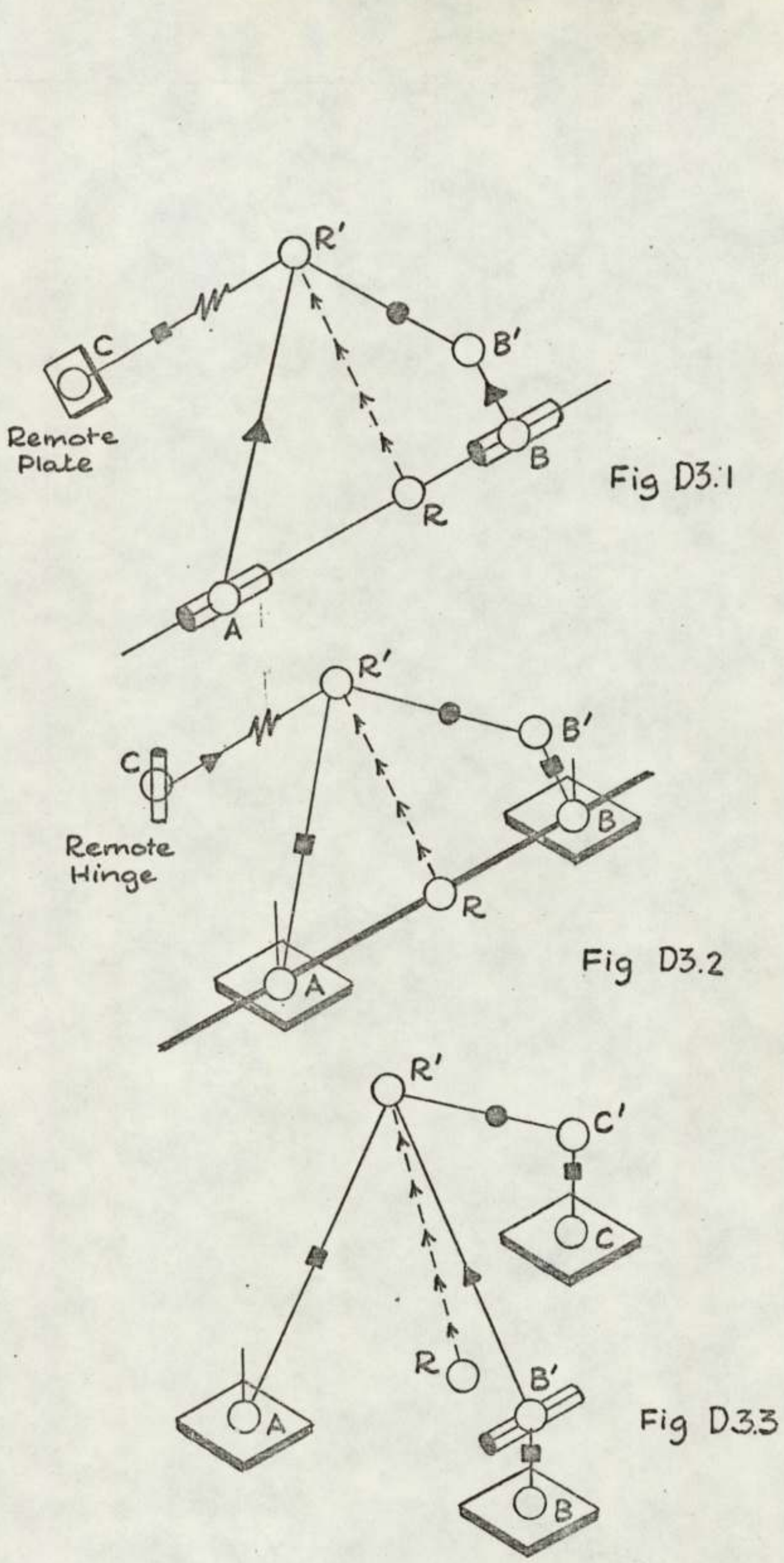


Fig D3.1

Fig D3.2

Fig D3.3

Fig D3 General Points on Lines & Planes.

as follows:

(a) A general point on a line.

(i) A general line - see Fig. D.3.1.

The general point R is on the line A-B. The displacement at B must be passed unchanged to the coincident point B' so that a valid socket location can exist at B'. R is also located on plate C which is stationed at a remote point with normal parallel with A-B.

(ii) A line on a plane - see Fig. D.3.2.

The general point R is on the line A-B. Again the displacement at B must be passed unchanged to the coincident point B' so that a valid socket can exist at B'. R is also located at hinge C, stationed at a remote point with direction parallel with the normals to A and B.

(b) A general point on a plane - see Fig. D.3.3.

The general point R is on the plane A-B-C. Displacements at B and C are passed unchanged to B' and C' for valid locations.

D.4 Remote Locations

Many useful location systems can be devised using remote locations. In cases studied so far, the following dimensions give adequate accuracy:

(i) A small displacement - of the order of 10^{-2}.

(ii) A neighbouring feature - one within a radius of about 10^{2} centred at the point being considered.

(iii) A remote feature - one further than 10^{6} from the point being considered.

(a) To pass a displacement unchanged from a feature to a neighbouring point.

(i) From a plate - Fig. D.4.1.

The displacement at plate A is passed unchanged to point A' (which may be coincident with A). A' can be any neighbouring point along the normal to plate A. The angle BAC is a right angle; and the direction

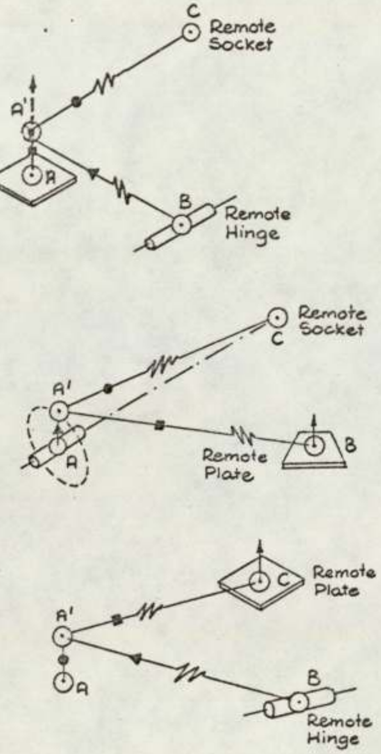
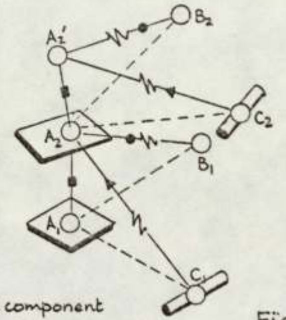


Fig D4.1

Fig D4.2

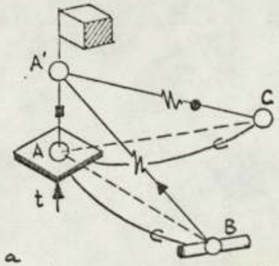
Fig D4.3

Fig D4. Use of Remote Features for Transfer



Selecting a component
of a tolerance.

Fig D5.1



Generating a
cubic tolerance zone

Fig D5.2

Fig D5 Use of Remote Features _ Components.

of hinge B is parallel with AC. Both B and C are remote from A.

(ii) From a hinge - Fig. D.4.2.

The displacement at hinge A is passed unchanged to point A' (which may be coincident with A). A' can be any neighbouring point whose position vector $\overline{AA'}$ is at right angles to the hinge direction. The remote socket C is approximately in line with the hinge direction and remote plate B is arbitrarily positioned and oriented.

(iii) From a socket - Fig. D.4.3.

The displacement at point A is passed unchanged to A' (which must not be exactly coincident with A, and is displaced a small amount). There is no restriction on the position of B nor on the position of C.

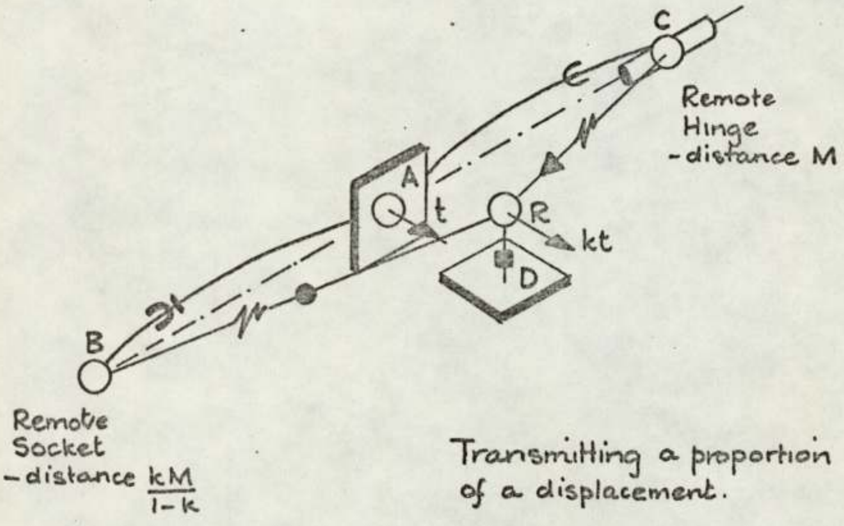
(b) To transfer a selected component of a displacement to a point - Fig. D.5.1.

The normal to plate A_2 is in the direction of the selected component. B_2 and C_2 are in the plane of A_2, the direction of C_2 is parallel with A_2B_2, and A_2' is coincident with A_2. The selected component of the displacement at A_1 is transmitted to A_2.

(c) To rotate a linear displacement through 90° - Fig. D.5.2.

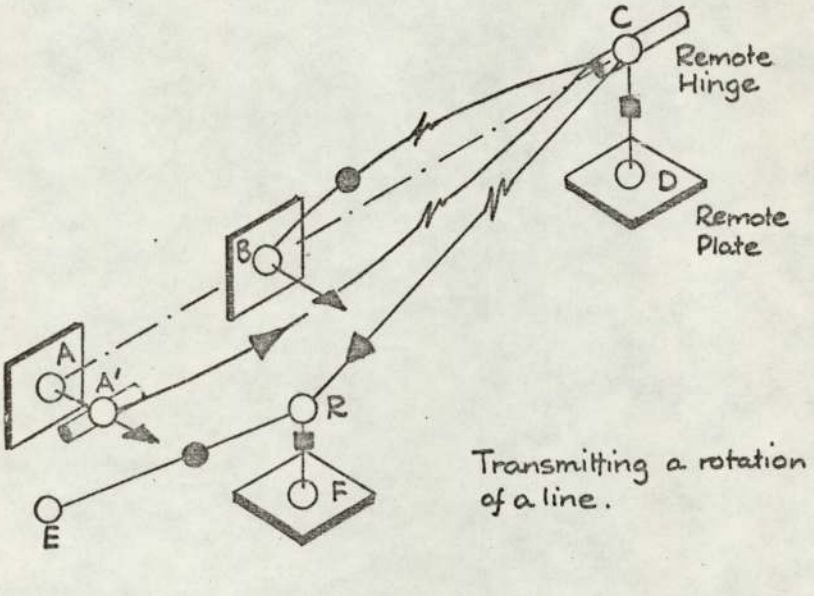
(Useful for generating a square or cubic tolerance zone).

Tolerance t is passed, using a unitary matrix, unchanged to remote hinge and socket B and C. The methods of (a) could be used to transmit 6, but use of a unitary matrix saves nodes, and avoids the need for a higher order of 'remoteness'. If B and C are orthogonal features, then a cubic tolerance, side t will be induced at A', which may be coincident with A. Adjustment of the direction of B and the positions of B and C will result in a parallelepipedal tolerance zone. If the locating features are stationed at the same large distances from the R-point with the direction of the H-loc and the normal to the P-loc lying along the lines joining them to the R-point, a unit parallelepipedal tolerance



Transmitting a proportion of a displacement.

Fig D6.1



Transmitting a rotation of a line.

Fig D6.2

Fig D6 Use of Remote Features _ Transmission zone will be generated. This will have sides normal to the directions of the lines joining P-, H- and S-locs to the R-point. This device is particularly useful and will be used for other purposes.

(d) To transmit a proportion of a displacement to a neighbouring point - Fig. D.6.1.

(Useful in generating symmetric tolerances).

It is required to pass a proportion k of the tolerance at plate A to the neighbouring point R. Displacement at A is passed unchanged to remote hinge C stationed at distance M in the plane of the plate. The neighbouring point R is located on C, and on B, a remote socket placed at a distance $\frac{kM}{1-k}$ on the line AC, but on the opposite side of A to the position of C. R is also located on plate D which has normal orthogonal to the plane RAC and passing through R. It may for convenience be coincident with R. In the case of symmetric tolerances, k must be $\frac{1}{2}$, and so B and C are equidistant from A. It is important that A, B and C should not be exactly collinear but only approximately so.

(e) To transmit the rotation of a line to a point which rotates about a neighbouring point - Fig. D.6.2.

Due to displacements at A and B, the line joining them will rotate in space. It is required that neighbouring result point R should rotate the same angle, in the same plane about location point E. Displacement at A is passed unchanged to A' a hinge with direction parallel with AB. To be consistent, the displacement at B should be passed to a coincident socket, but a node may be saved by locating directly at B. Displacements at A and B are passed to the remote hinge C by way of hinge and socket locations. C is stationed approximately in line with AB and located on D, a coincident remote plate whose normal is orthogonal to normals at A and B. The result point R is located on the socket E, the hinge C and coincident plate F. A, B, E and R are co-planar, and the normal of plate F is orthogonal to this plane.

Any displacement at E is passed unchanged to R.

(f) Use of unitary links to pass a displacement unchanged from a feature to a point.

The unitary link is an artificial device used for the generation of geometric tolerance networks which cannot be done by using real locations. It may be used to obtain the results shown in (a) with a saving of nodes. In most of the networks which follow, the preferred method is to use the devices shown in (a), since this leads to a more natural system. However, occasionally, unitary links have been included so as to give examples of their use. A separate section is devoted to this application. Unitary links are used for the superposition of separate tolerance systems.

(g) Use of weak links.

A result point is located on plate, hinge and socket, and displacement at the locations may be considered to be transmitted along the link, usually. Occasionally, however, it is convenient, in the interests of node conservation, to use a device called the weak link. A weak link points to a location node which is only used for the calculation of displacements transmitted from other nodes. Displacement occurring at the weakly linked node is not transmitted to the result node. This device, like the unitary link, is not essential to the system but avoids node duplication. An example of a weak link is shown in Fig. D.6.1, the unitary link from remote socket B being distinguished as weak by the bar drawn across the link. Examples of the use of weak links will be found later in this Appendix.

D.5 Use of Unitary Links

The use of imaginary locations in transferring displacements unchanged from features to neighbouring points has been described in detail. This

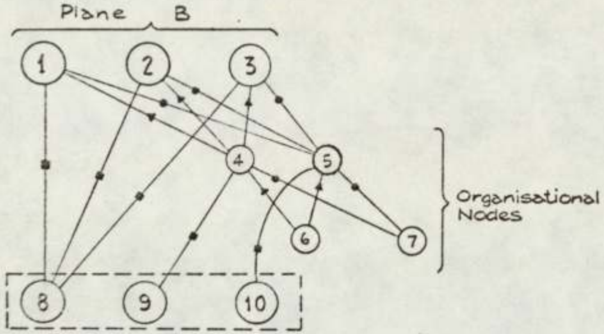


Fig D7.1

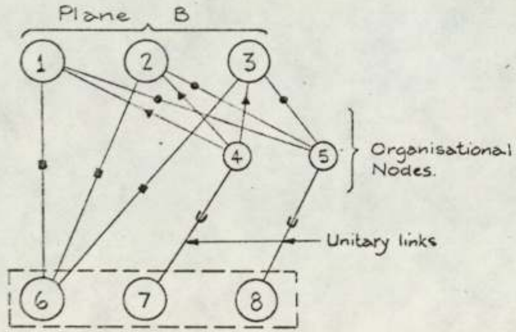


Fig D7.2

Fig D7 Use of Unitary Links

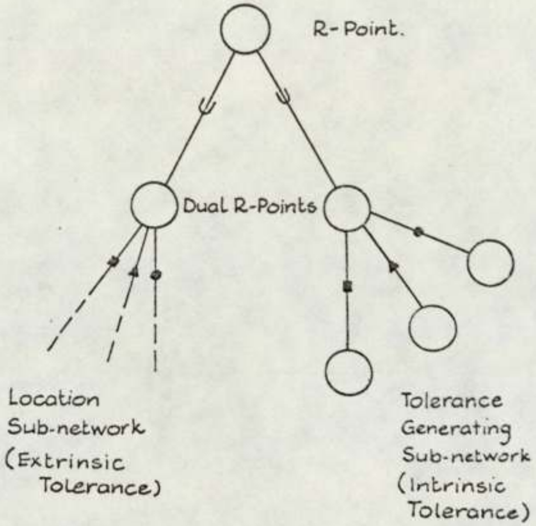


Fig D8. Use of Unitary Links.

device is necessary in many cases - for instance a point located on a plane requires that the point should be located in three plane features. This is clearly impossible to do directly, since the model demands that each feature should be located on a triad of plate, hinge and socket. The point can be located on one of the plates, while displacements at the other two plates must be passed unchanged to coincident features, one a hinge, the other a socket. All calculation of matrices will then be consistent. However, this necessitates the introduction of nodes which do not represent actual points, or features on the assembly, these being termed 'organisational nodes'. They occur in most practical location systems, represent a considerable overhead in storage and also tend to make sub-networks appear more complex than they actually are. A typical example is shown in Fig. D.7.1.

The unitary link is an artificial device which obviates the need for most of these organisational nodes. Instead of the effect of unchanged transfer being obtained by using features at infinity, the matrix is evaluated directly. An advantage of this method is that one or more of the links may be null without affecting the validity of the model. An example is shown in Fig. D.7.2 which is the equivalent of Fig. D.7.1.

The network illustrated represents a plane located on a datum plane and occurs in tolerances of parallelism and angularity. In the examples which follow,unitary links are not usually used but it is probable that in a large practical network, it would be necessary to conserve storage by using them.

Another use of unitary links is to superpose the extrinsic and intrinsic tolerances at an R-point. This may often be done by other means but using the unitary link method avoids complication. An example is shown in Fig. D.8.

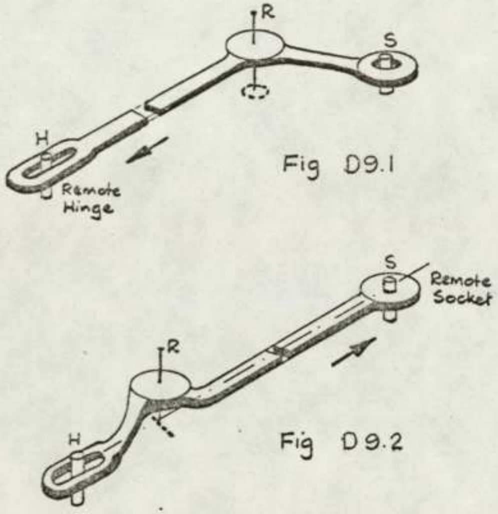


Fig D9.1

Fig D9.2

Fig D9 Equivalent Mechanisms.

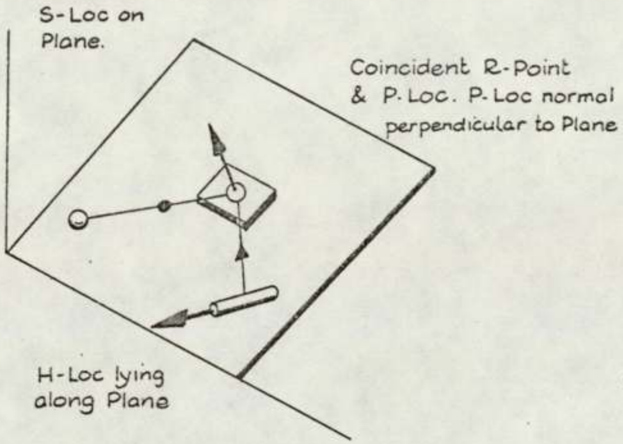


Fig D10. Location in a Plane.

D.6 Equivalent Mechanisms

The devices shown on previous pages are a few of the many which can be created to attain special effects. It is useful, in considering systems of this kind, to think of the location triad as a mechanism – this is particularly useful when the system is two dimensional.

Two examples are shown.

(a) Fig. D.9.1 – Passing a tolerance unchanged from a socket to a neighbouring point. H is a hinge which can in the planar case be imagined as a closely fitting slot around a fixed pin. Clearance at S will be passed to R unchanged when the mechanism is moved around.

(b) Fig. D.9.2 – Passing a tolerance unchanged from a hinge to a neighbouring point. In this case S is a remote socket freely pivoting around a close fitting pin. If S is in line with H, then when the mechanism is moved, clearance at H will be passed unchanged to R.

D.7 Two Dimensional Cases

The most common dimensioning system occurs when features are located on a plane. These are the most easily visualised using equivalent mechanisms. The main principle involved is the stationing of a P-loc coincident with the R-point with normal perpendicular to the plane of interest. This ensures that all displacement at the R-point caused by features on the plane of interest is in the plane. Tolerancing systems on the plane may now be described by using H-locs lying in the plane and S-locs lying on it. An example is shown in Fig. D.10.

Some common cases of dimensioning in two dimensions will be discussed, but it is first necessary to consider another use of remote features.

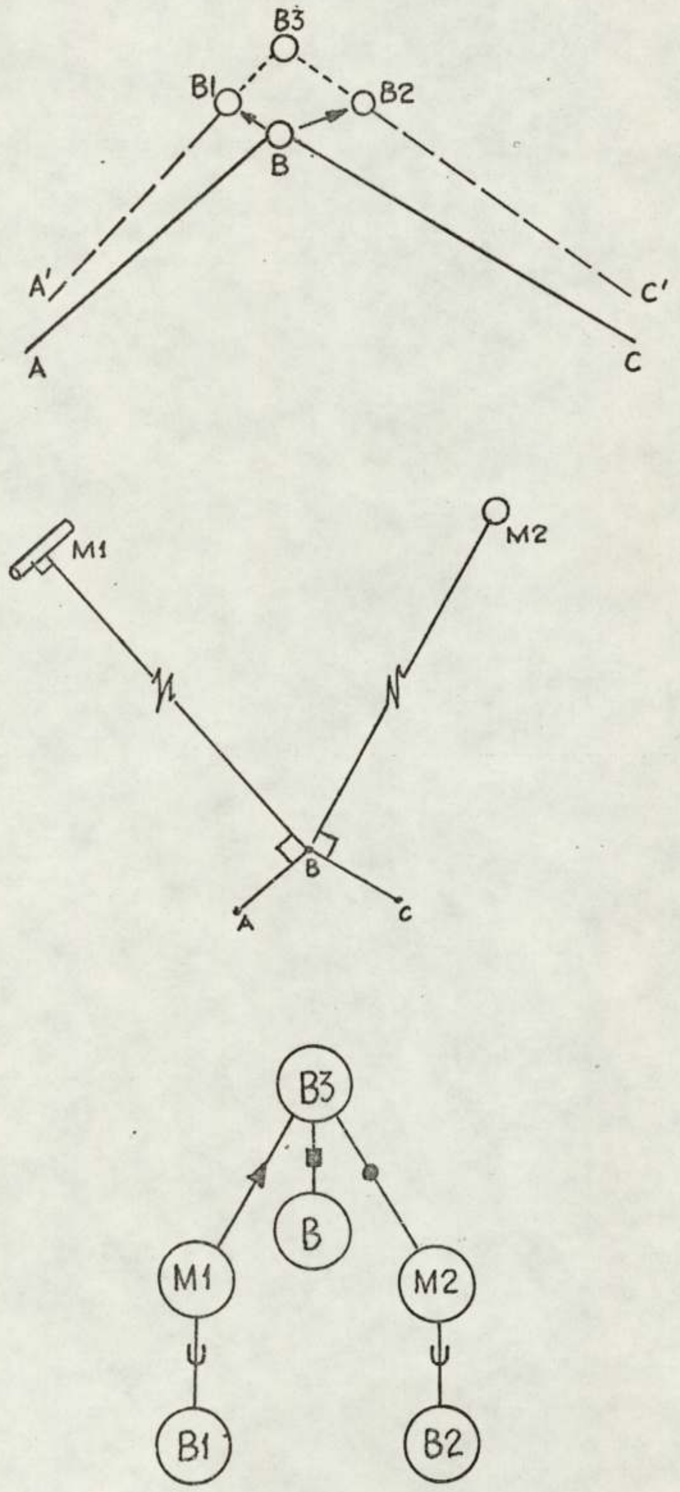


Fig D11. The Intersection of Two Lines

D.7.1 Intersection Points of Lines

Often a point is located geometrically. This usually implies the superposition of two or more real location systems. A common example is the point described by the intersection of two straight lines. In Fig. D.11 lines AB and CB intersect at the point feature B. If lines AB and CB are separately located, there will be an ambiguity at B. Point B1 on line AB will be displaced due to intrinsic displacement of AB and point B2 will be displaced due to intrinsic displacement of CB and these displacements will not be identical. Consequently, the true intersection of AB and CB will be neither B1 nor B2 but some point B3. If B1 and B2 are at right angles, the point B3 will be defined by the vector sum of the displacements of B1 and B2 but this is not generally so. It is convenient to consider the displacement of B1 as being constrained by the line CB which does not contain it. A sub-network for the generation of the displacement of point B3 from its nominal position at B is given by the following:

(a) two remote features M1 and M2 are set at equal distances from the origin, for example at 10^6 units,

(b) M1 is stationed at right angles to AB, M2 at right angles to BC,

(c) either is chosen as an H-loc, the other as a P-loc, the direction of the hinge vector being at right angles to the line joining it to the origin,

(d) displacement at B1 is passed unchanged via a unitary link to M1, displacement at B2 is passed similarly to M2,

(e) the R-point is taken at the nominal position of B.

This sub-network is useful in various situations and its validity may easily be proved by considering instantaneous centres.

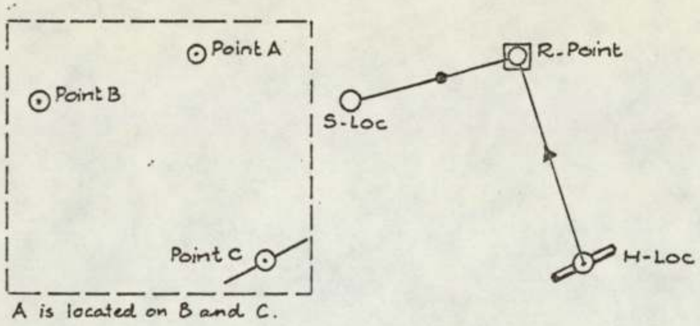


Fig D12. A Point Located on a Point and a Point on a Line.

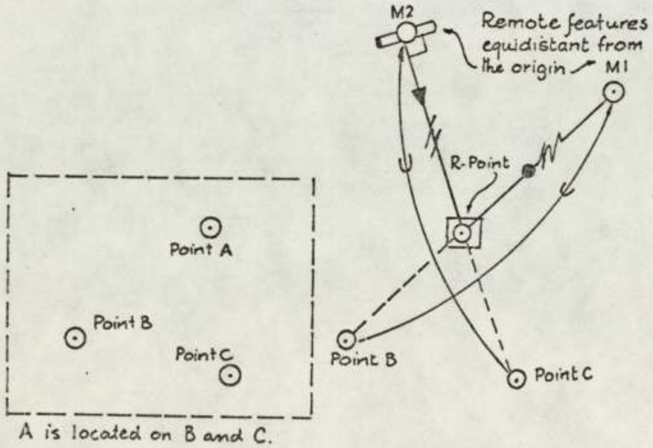


Fig D13. A Point Located on Two Points

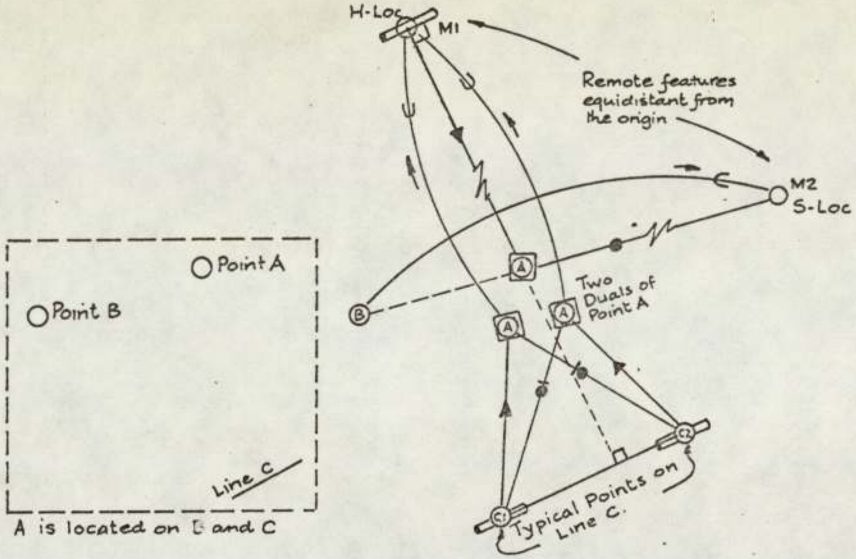


Fig D14 A Point Located on a Point and a Line

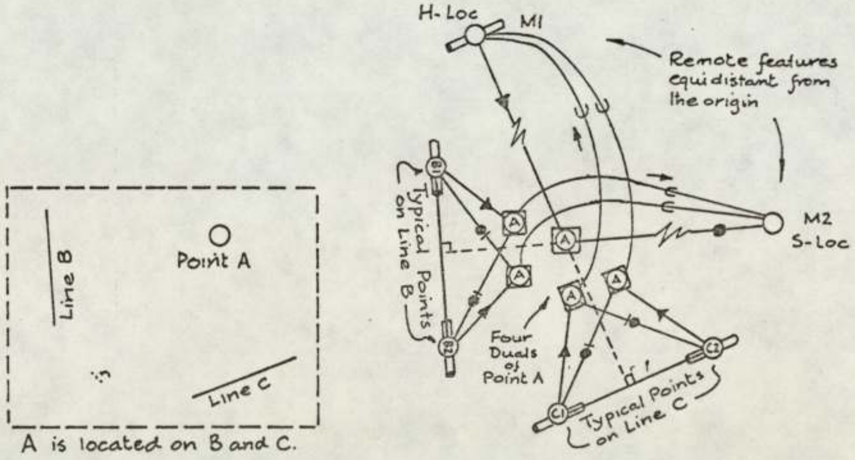


Fig D15 A Point Located on Two Lines

D.7.2 Location of a Point in a Plane

(a) Location by a point and a point on a line.

This is the simplest case, since it is a real location. An example is shown in Fig. D.12.

(b) Location by the distances from two points.

This case may be described by considering it as a case of constrained displacement. M1 and M2 are taken along the lines joining the points and the R-point as shown in Fig. D.13.

(c) Location by the distance from a line and distance from a point.

The remote features M1 and M2 are set at right angles to the line and along the line joining the point and the R-point as shown in Fig. D.14.

(d) Location by the perpendicular distances from two lines.

M1 and M2 are set at right angles to each line. The point is located separately on each line by using two networks as described in (a) and superimposing them. Note that only the H-loc displacement is passed from each point, weak links being used, and that each point on the line is used in a dual capacity as H-loc and S-loc. An example is shown in Fig. D.15. This case is very common, occurring in coordinated dimensions.

In each of these cases, the sub-network describes the extrinsic tolerance. Intrinsic tolerance is handled differently for the two standard methods.

(a) Positional tolerance is applied directly at the R-point.

(b) Tolerances on dimensions are applied directly at the remote features usually.

They may all be generalised to three dimensions.

The most common cases of tolerancing as shown in BS 308 will now be described.

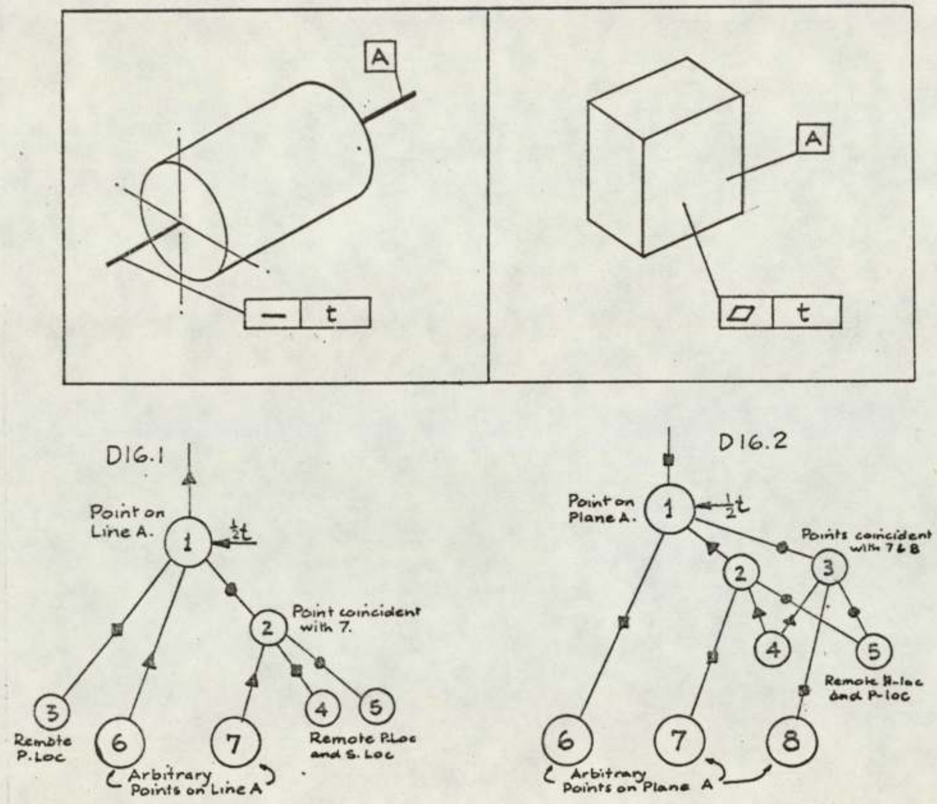


Fig D16 Straightness & Flatness.

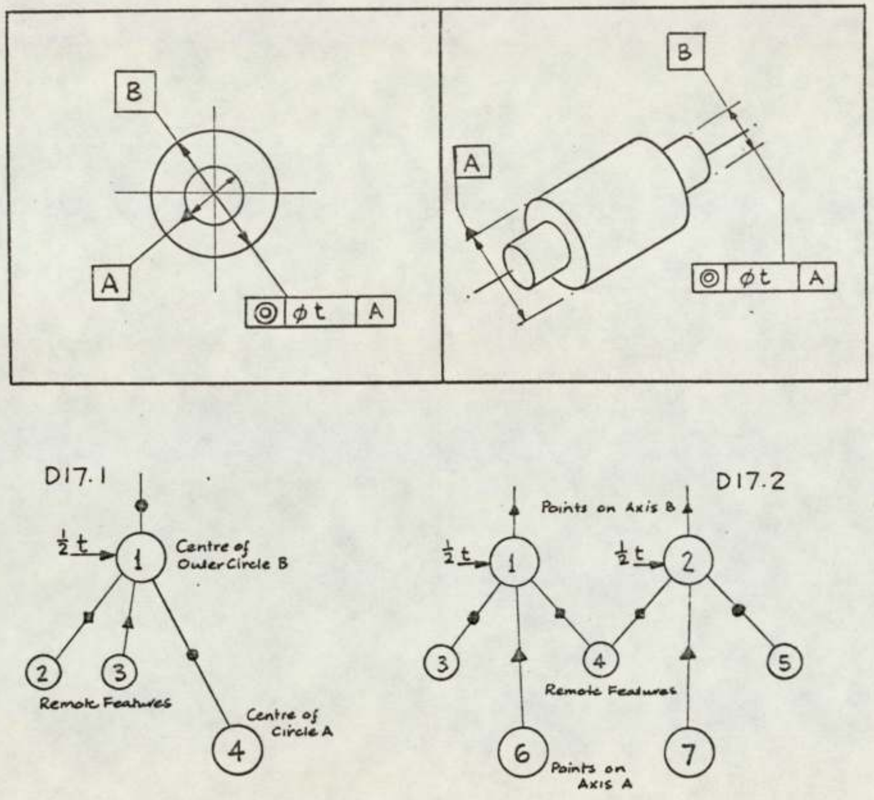


Fig D17 Concentricity.

D.8 Three Dimensional Systems

D.8.1 Tolerances of Straightness and Flatness

(a) Tolerance of straightness of a line - Fig. D.16.1.

The centre line A of the cylinder is subject to a straightness tolerance $t/_2$. This is applied directly to node 1. Nodes 6 and 7 describe the centre line; and the network transmits displacements at 6 and 7 to the node 1.

(b) Tolerance of flatness of a plane - Fig. D.16.2.

Plane A is subject to a flatness tolerance $t/_2$. This is applied directly to node 1, which describes a point on the plane. Nodes 6, 7 and 8 describe the plane; and the network transmits displacements at 6, 7 and 8 to the node 1.

D.8.2 Tolerances of Concentricity

(a) Concentricity of a point - Fig. D.17.1.

The centre of circle B is required to lie within a circle diameter t, concentric with the centre of the datum circle A. Displacement at node 4 - centre of circle A - is passed unchanged to node 1 which describes the centre of circle B. The tolerance $t/_2$ is applied directly to node 1.

(b) Concentricity of a line - Fig. D.17.2.

The axis of the cylinder B is required to be contained within a cylinder diameter t co-axial with cylinder A. Displacements at nodes 6 and 7 which describe the axis of A are passed unchanged to nodes 1 and 2 describing axis B. The tolerance $t/_2$ is applied directly to nodes 1 and 2. In this case, nodes 1 and 2 are chosen to be coincident with nodes 6 and 7 which results in a simple network. In the more general case where 1 and 2 are not coincident with 6 and 7, then displacements at both 6 and 7 will result in displacements at both of 1 and 2; and the network is consequently more complex.

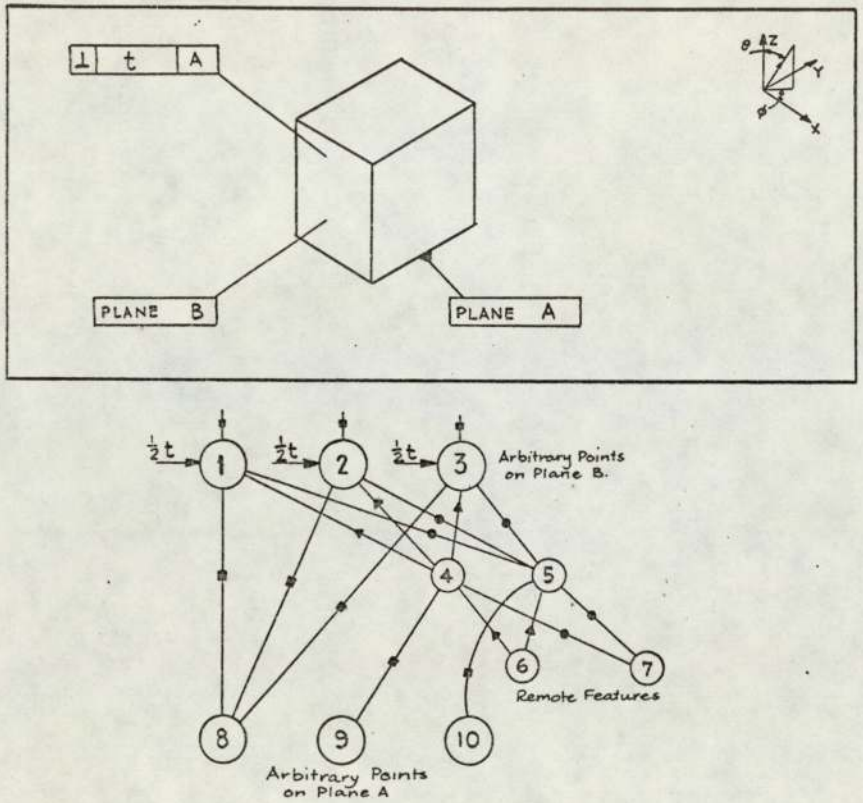


Fig D18 Squareness

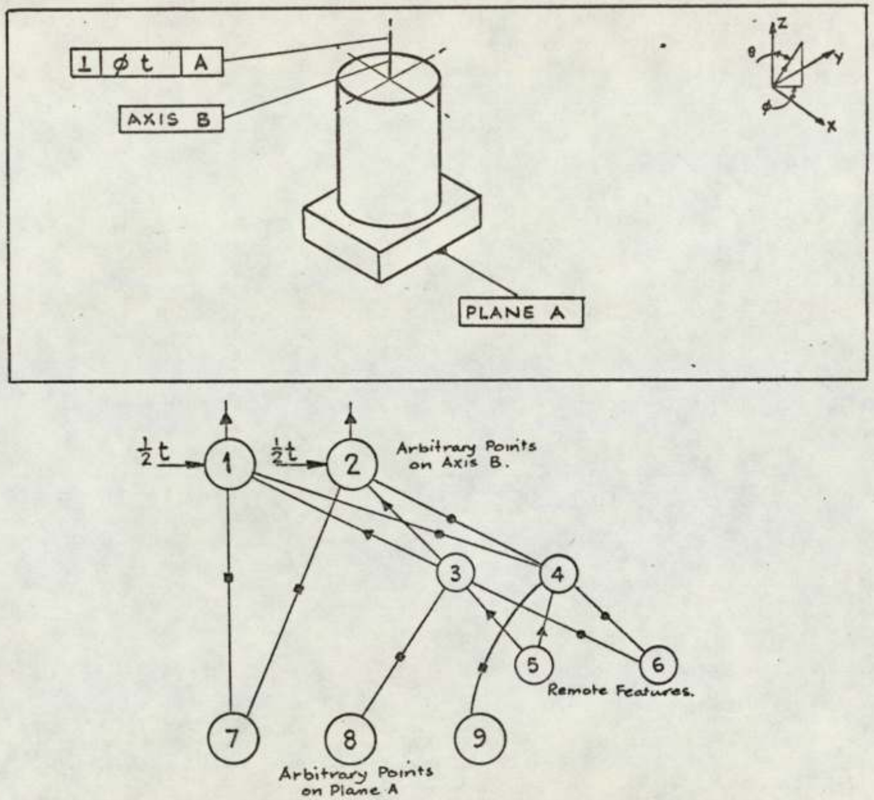


Fig D19 Squareness

D.8.3 Squareness Tolerances

(a) Squareness of a plane relative to a datum plane - Fig. D.18.

Datum plane A is described by plates stationed arbitrarily on A - nodes 8, 9 and 10. Displacements at 9 and 10 are passed unchanged to coincident features 4 and 5 using remote features 6 and 7. Features 4 and 5 act as hinge and socket locations for features 1, 2 and 3 which describe the dependent plane B. Features 1, 2 and 3 are also located directly on feature B. The squareness tolerance $t/_2$ is applied directly at 1, 2 and 3, indirect displacements at 8, 9 and 10 being passed through the network.

(b) Squareness of a line relative to a datum plane - Fig. D.19.

The case shown is that of an axis of symmetry which is square to a plane within a cylindrical tolerance zone. The network organisation is similar to that in (a) except that there are two hinge features 1 and 2 describing the dependent line.

An alternative case of this kind is that of a line on a plane square to the datum plane, within a rectangular tolerance band. This system is identical to the one shown in Fig. D.19 except that 1 and 2 will be plate features with normal in the plane of the tolerance band.

D.8.4 Tolerances of Angularity

(a) Angularity of a face relative to a datum plane - Fig. D.20
- Tolerance band.

Fig. D.20 shows the method of specifying angularity tolerances recommended in BS 308. The network system is identical with that for a squareness tolerance which is a particular case of angularity tolerance, if this tolerancing method is used.

(b) Angularity of a face relative to a datum plane - Tolerance on angle.

For this non-standard case, the angle is shown as, for example

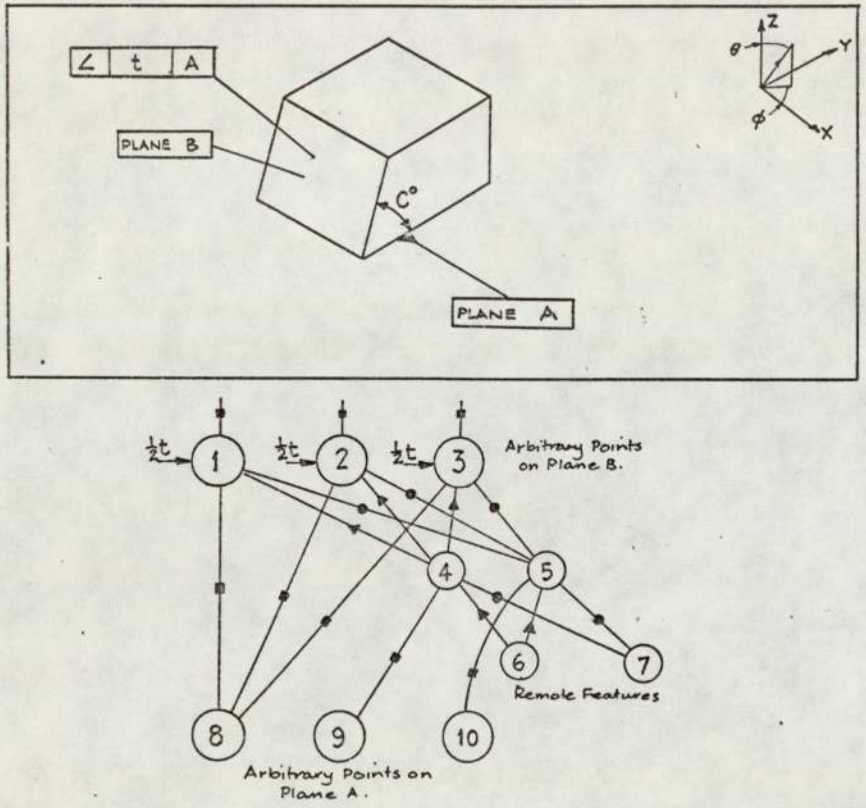


Fig D20 Angularity

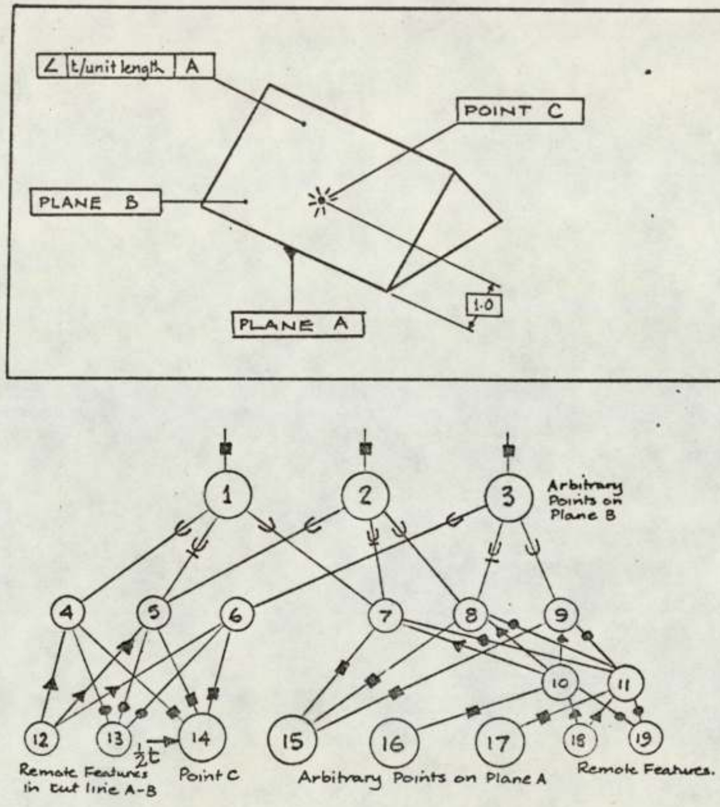


Fig D21 Angularity

$60^{\circ} \pm 1^{\circ}$. The easiest way to deal with this case is calculate the actual tolerances at features 1, 2 and 3; and to apply these at features 1, 2 and 3 as shown in Fig. D.20. Each tolerance value will be different in this case, but the network will be the same as (a).

(c) Angularity of a face relative to a datum plane - Unit tolerance, i.e. tolerance/unit distance.

Again, in this non-standard case, it is most convenient to calculate the actual tolerance at features 1, 2 and 3, so as to retain the same network. However, since this case occurs fairly frequently, a separate treatment follows.

In the case shown in Fig. D.21, tolerance $t/_2$ is applied at unit distance from the intersection line of the two planes A and B. The point of application is labelled as 'Point C'.

This construction uses unitary links. The displacements due to displacement of datum plane A (indirect tolerance), and the unit angular tolerance t are superposed by means of unitary links. The weak links at nodes 1, 2 and 3 are redundant, and may be omitted from the diagram, but are included for the sake of consistency - each node has outdegree 3.

The network shown can be simplified to some extent, and some of the nodes omitted; but in all the networks shown in the examples, the most direct method has been used even if this has necessitated using extra nodes.

D.8.5 Tolerances of Symmetry

(a) Symmetric tolerance - datum planes parallel - Fig. D.22.

Again unitary links are used; and weak links to ensure that unwanted tolerances are not passed along a path. In general, the presence of a unitary link implies that the location system is not physical but geometric. In view of the number of nodes used in this system, it might be considered useful to provide an artificial 'half-unitary' node, which

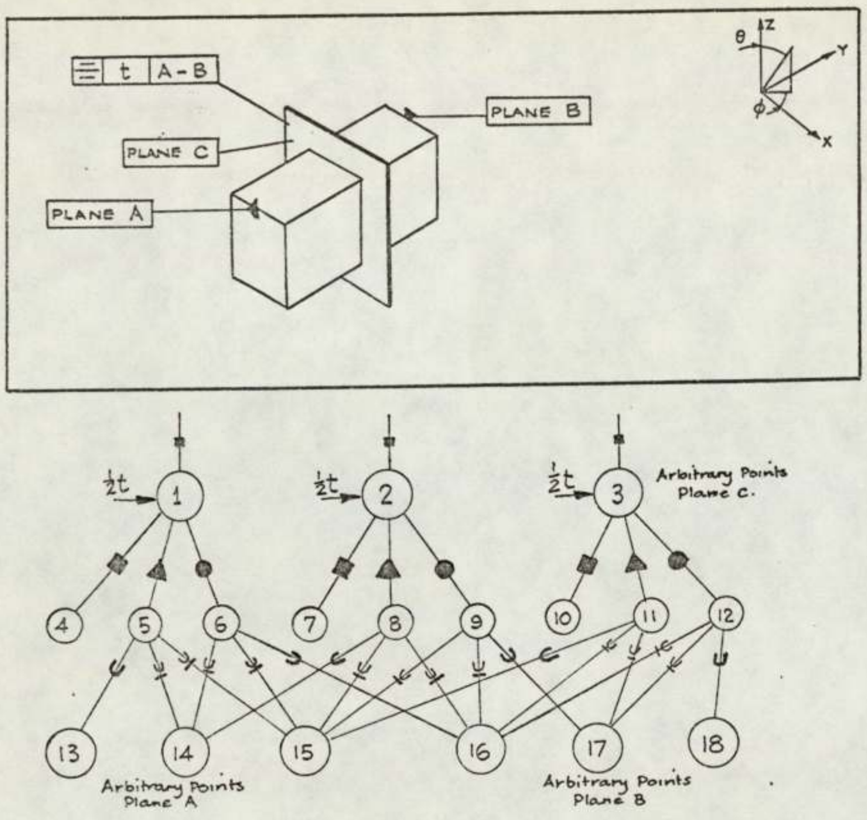


Fig D22 Symmetry.

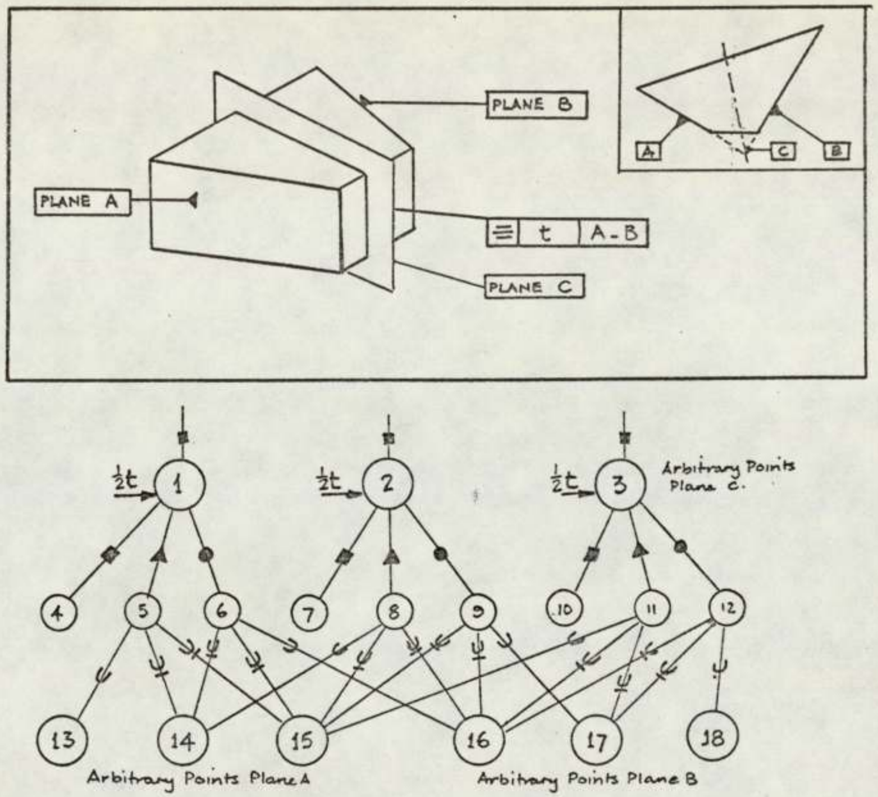


Fig D23 Symmetry

would considerably simplify the network. Once again, the view is taken that it is better to be a little prodigal with nodes rather than to complicate the system.

(b) Symmetric tolerance - datum planes not parallel - Fig. D.23.

This case is more general than (a), but is basically the same network.

D.8.6 Tolerances of Parallelism

(a) Parallelism of a plane relative to a plane - Fig. D.24.

Nodes 8, 9 and 10 describe the datum plane; nodes 1, 2 and 3 describe the located plane.

D.8.7 Coordinate Distances from Three Planes

(Cartesian coordinate system) - Fig. D.25.

Point P is located on three flat faces A, B and C. Each of the three planes is defined by a sub-network shown dotted in Fig. D.25 and previously described in section D.3 . Remote features X, Y and Z are used to separate out components of displacements in the directions indicated by their names. Z is a plate stationed at infinity along the Z axis, with normal along the Z axis, X is a hinge stationed at infinity along the X axis with direction at right angles to the X axis; Y is a socket stationed at infinity along the Y axis. These will select the components in the directions of the axes along which they are stationed and these components are passed to P using unitary links.

The network for this system appears rather complicated and requires a disproportionate number of nodes. However, each extra point located on this system only needs four extra nodes similar to P; and the spare unitary links at X, Y and Z may also be used for two extra points, extra X, Y and Z nodes being necessary for each three result points. The number of nodes can also be reduced by using the remote features in more complicated ways. For instance, instead of the six remote nodes used in the

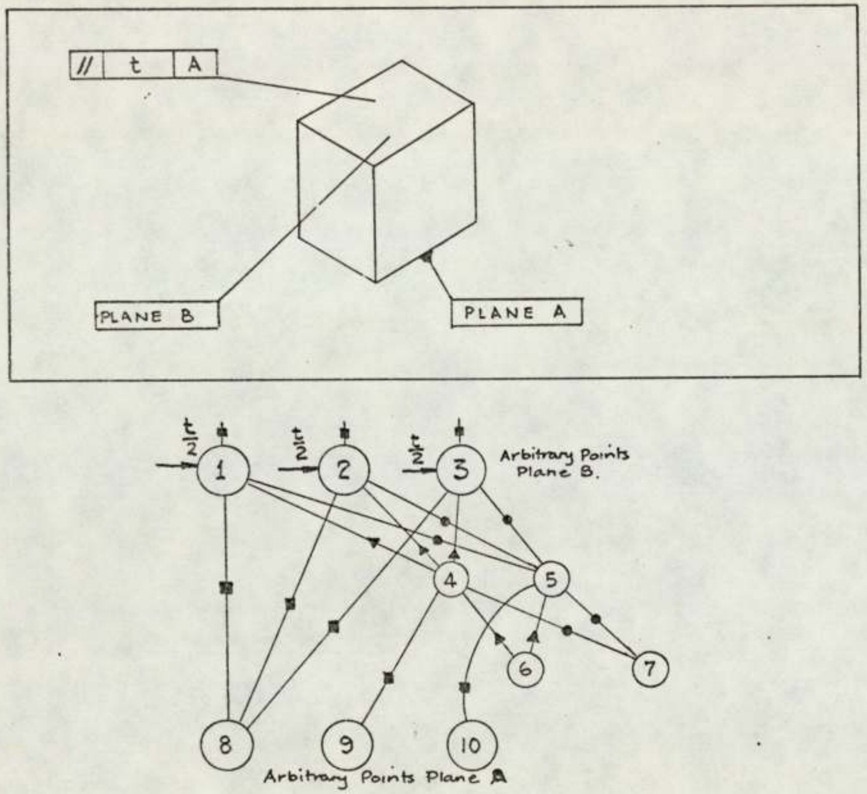


Fig D24 Parallelism.

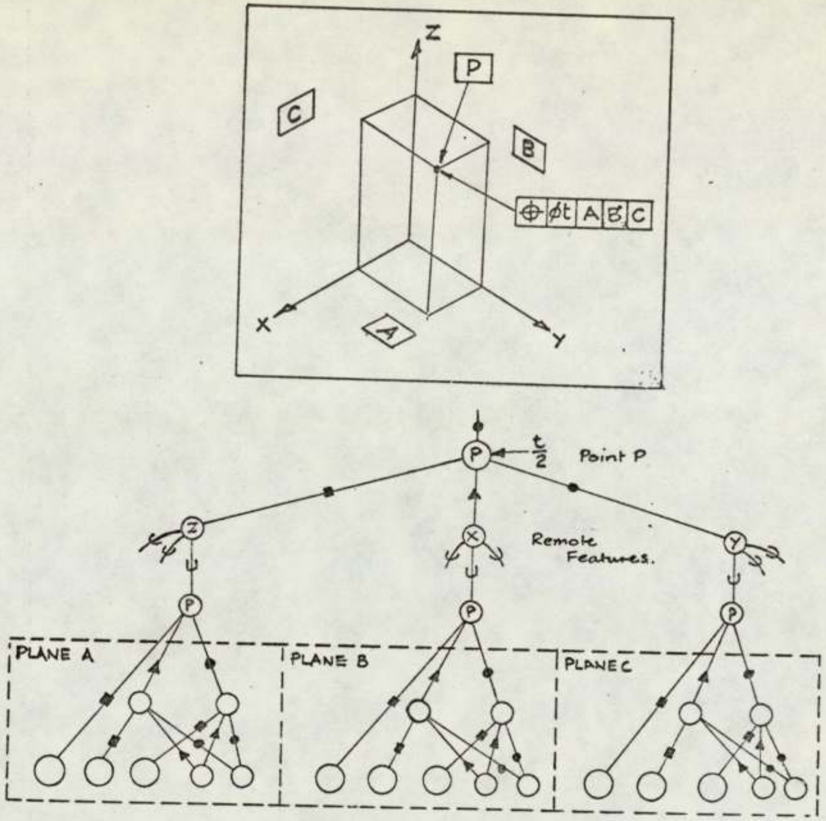


Fig D25. Co-ordinates

sub-network for planes A, B and C, three features may be used jointly, each feature being used once as a hinge and once as a socket. However, once again the most direct network system has been chosen. Fig. D.25 shows a spherical tolerance $t/_2$ applied directly at the result point P. If the tolerance is parallelepipedal, then the three components may be applied directly at X, Y and Z.

The sub-networks described in this Appendix are by no means an exhaustive set but they should be sufficient to handle most common dimensioning systems.

APPENDIX E

PRACTICAL EXAMPLES

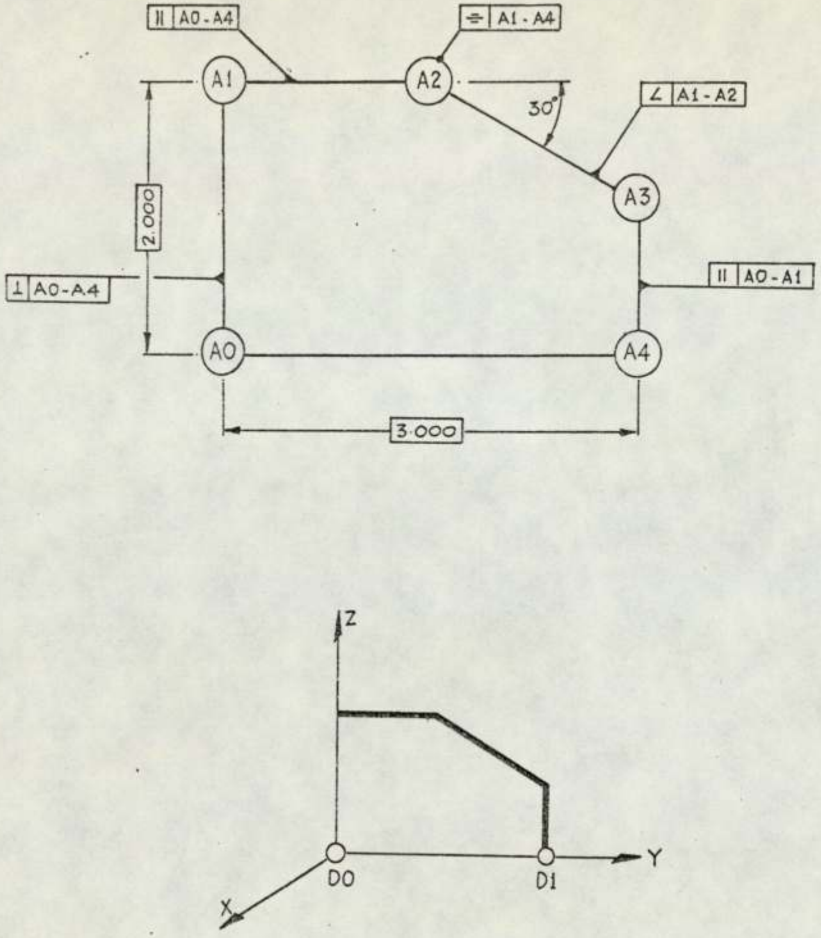
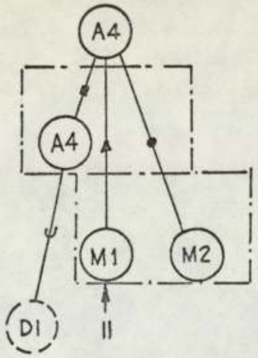


Fig E1. Definition of Plate.



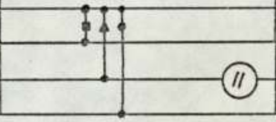
Node	Network	Description	Co-ords			Angs		Links		
			X	Y	Z	θ	ϕ	■	▲	●
1		Point A4	0	3	0	0	0	2	3	4
2		P-loc A4	0	3	0	90	0	0	0	0
3		Remote H-loc	0	-106	0	0	90	0	0	0
4		Remote S-loc	0	0	-106	0	0	0	0	0

Fig E2. Description of Point A4.

E.1 Examples

The cases considered will be didactic rather than practical and the dimensioning sufficiently eccentric as to include most of the common datum systems met in practice. It is quite difficult to check some of the results, and this has been done mainly by calculation but occasionally by drawing.

(a) Two dimensional system. Fig. E.1

The plate is defined by the five points A0, A1, A2, A3 and A4. Each of these points may be considered as two subsidiary points coincident in the plane. For example, point A1 lying at the intersection of lines A0 - A1 and A1 - A2 may be considered as points A1a lying on A0 - A1 and A1b lying on A1 - A2. The lamina is located in orthogonal coordinates as shown in Fig. E.1 and since all displacement is in the plane of the lamina, a plate feature is set at each point with its normal perpendicular to the plane of the lamina. The input tolerance at this P-loc will be zero, thus ensuring that displacements at all the features on the lamina will be in its plane. The datum is chosen as the point D0 (coincident with A0) and a line through D0 and D1 (coincident with A4).

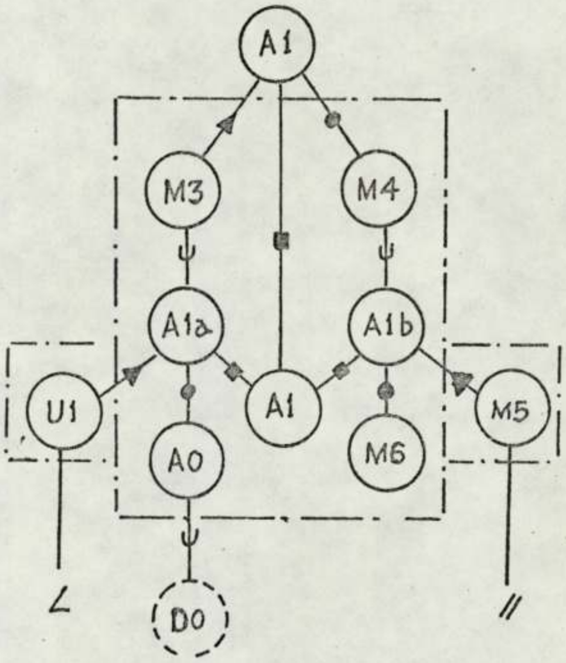
The five points will be considered separately.

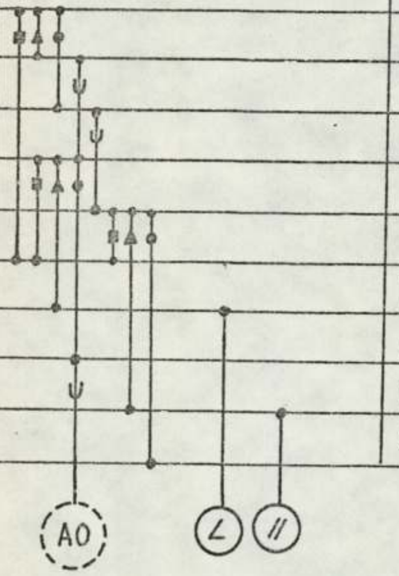
(i) Point A4. Fig. E.2.

A parallelism tolerance is applied directly to A4. Since only one tolerance acts at this point, it is not necessary to use the dual point.

(ii) Point A1. Fig. E.3.

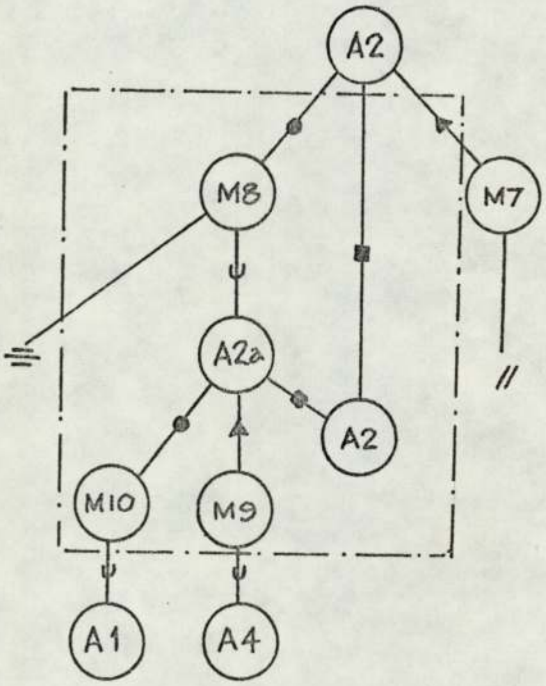
Two tolerances are applied at A1, an angularity tolerance at A1a and a parallelism tolerance at A1b. These will be compounded to give the actual position of point A1. Since the angularity tolerance is quoted as 'tolerance per unit distance', a subsidiary point U1 is taken at unit distance from A0 along A0 - A1 and an H-loc set at U1. This is passed via a tolerance generating network to A1a.



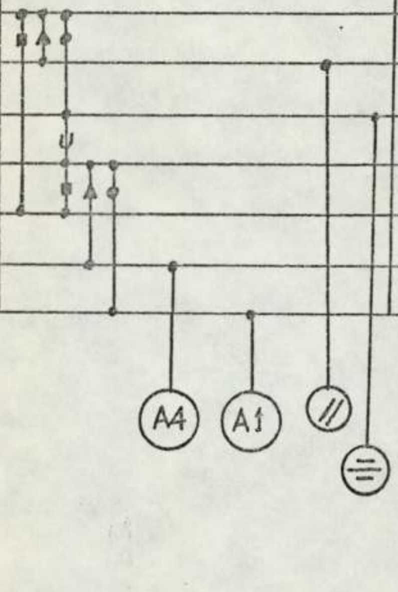
Node	Network	Description	Co-ords			Angs		Links		
			X	Y	Z	θ	φ	■	▲	●
1		Point A1	0	0	2	0	0	6	2	3
2		Remote H-loc M3	0	-106	0	0	90	4	0	0
3		Remote S-loc M4	0	0	-106	0	0	5	0	0
4		Dual Point A1a	0	0	2	0	0	6	7	8
5		Dual Point A1b	0	0	2	0	0	6	9	10
6		P-loc A1	0	0	2	90	0	0	0	0
7		U1	0	0*	1	0	90	0	0	0
8		A0 - location	0	0	0	0	0	●	0	0
9		Remote H-loc M5	0	0	-106	90	90	●	0	0
10		Remote S-loc M6	0	-106	0	0	0	0	0	0

* Perturbed Dimension
● External Link

Fig E3. Description of Point A1

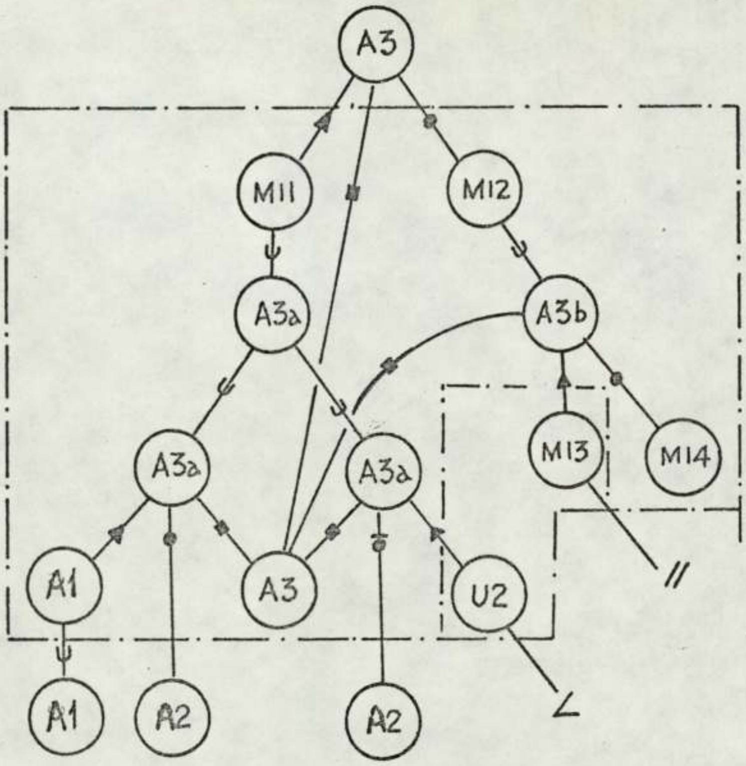


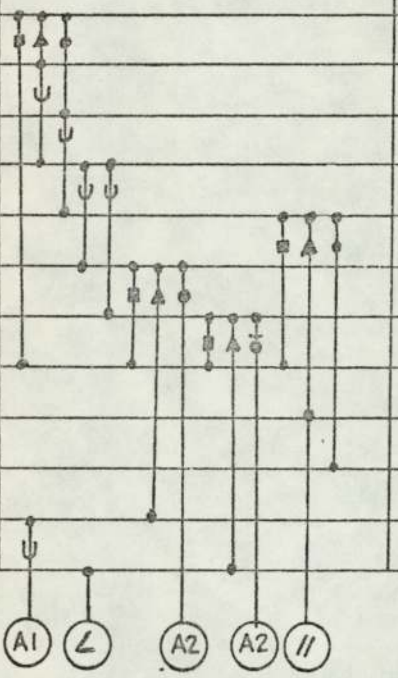
Node	Network	Description	Co-ords			Angs		Links		
			X	Y	Z	θ	φ	■	▲	●
1		Point A2	0	1·5	2	0	0	5	2	3
2		Remote H-loc M7	0	0	-106	90	90	●	0	0
3		Remote S-loc M8	0	-106	0	0	0	4	0	0
4		Dual point A2a	0	1·5	2	0	0	5	6	7
5		P-loc A2	0	1·5	2	90	0	0	0	0
6		Remote M9	0	0	106	0	90	●	0	0
7		Remote M10	0	0	-106	0	0	●	0	0



● External Link.

Fig E4. Description of Point A2





Node	Network	Description	Co - ords			Angs		Links		
			X	Y	Z	θ	φ	■	▲	●
1		Point A3	0	3	1·134	0	0	8	2	3
2		Remote H-loc M11	0	$\cdot5_{10}6$	$\cdot866_{10}6$	-60	90	4	0	0
3		Remote S-loc M12	0	$_{10}6$	0	0	0	5	0	0
4		Dual Point A3a	0	3	1·134	0	0	6	7	0
5		Dual Point A3b	0	3	1·134	0	0	8	9	10
6		Dual Point A3a′	0	3	1·134	0	0	8	11	●
7		Dual Point A3a″	0	3	1·134	0	0	8	12	-●
8		P-loc A3	0	3	1·134	90	0	0	0	0
9		Remote H-loc M13	0	$-_{10}6$	0	0	90	●	0	0
10		Remote S-loc M14	0	0	$-_{10}6$	0	0	0	0	0
11		Point A1 - H-loc	0	0	2	90	90	●	0	0
12		Point U2.	0	2.366	1·5	-60	90	0	0	0

● External Link
-N Weak Link.

Fig E5 Description of Point A3

Since A1 is located on a datum which is earthed, there is no extrinsic tolerance network necessary and the parallelism tolerance may be applied directly to point A1b. Displacements at A1a and Alb are compounded using remote points M3 and M4 at right angles to lines AO - A1 and A1 - A2 respectively. This completes the definition of point A1.

This location system may be imagined as a 'black box' with one or more input terminals at which are applied extrinsic tolerances due to location, one or more input terminals at which are applied intrinsic tolerance due to permitted displacement of the point itself and one output terminal which may be connected with a unitary link to another 'black box'.

(iii) Point A2. Fig. E.4.

There are two tolerances acting at this point. As with A1, the parallelism tolerance acts directly and may be applied through a tolerance defining sub-network to point A2a. The symmetric tolerance depends on the two defining points A1 and A4, and is passed through a sub-network to A2b, using the remote features M3 and M4 stationed as shown in the figure. This completes the definition of point A2.

(iv) Point A3. Fig. E.5.

Dual point A3a is fixed in relation to A1 (a point on A1 - A2) and point A2. Displacements at these features are transmitted to A3a through a sub-network as shown. The angularity tolerance is passed through a tolerance generating sub-network to A3a, the weak link from A2 ensuring that tolerance is not passed twice from the same point. The parallelism tolerance is applied directly to point A3b and the displacements compounded using remote features normal to lines A2 - A3 and A4 - A3.

The plate is now defined. Each of the 'black boxes' describing the points may be tested separately before being linked into a full network

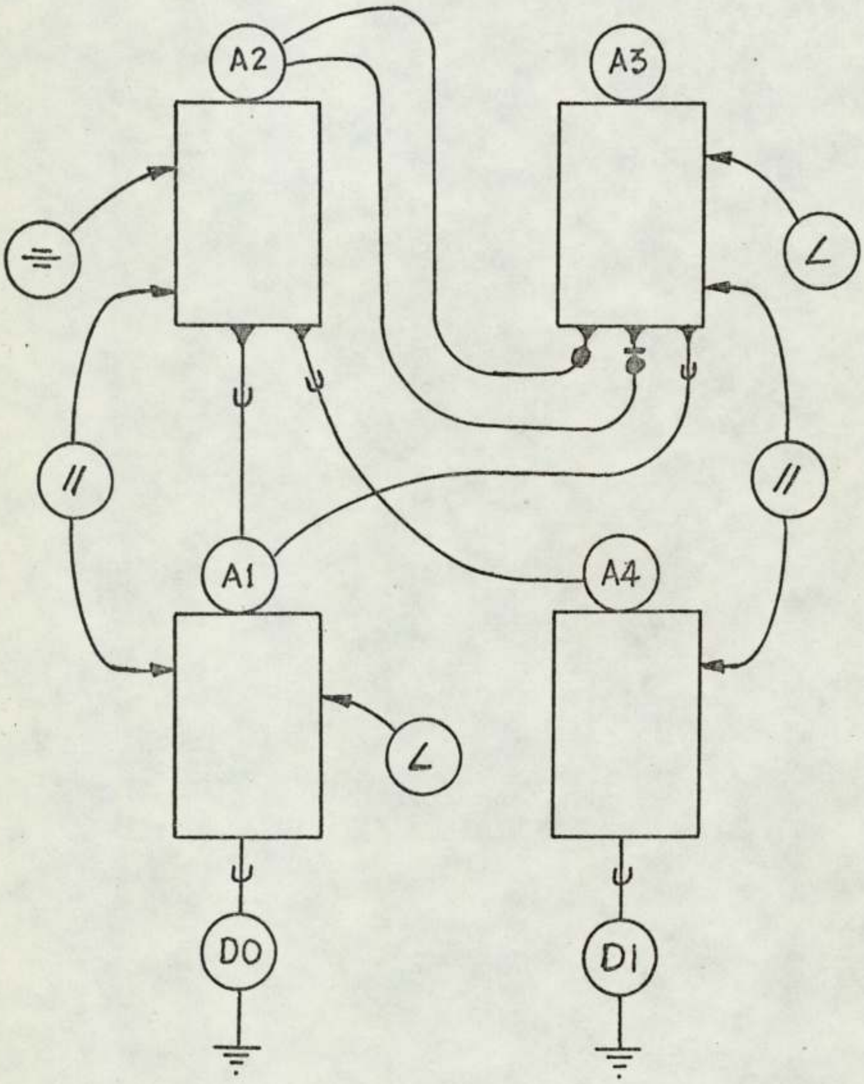


Fig E6 Description of Plate.

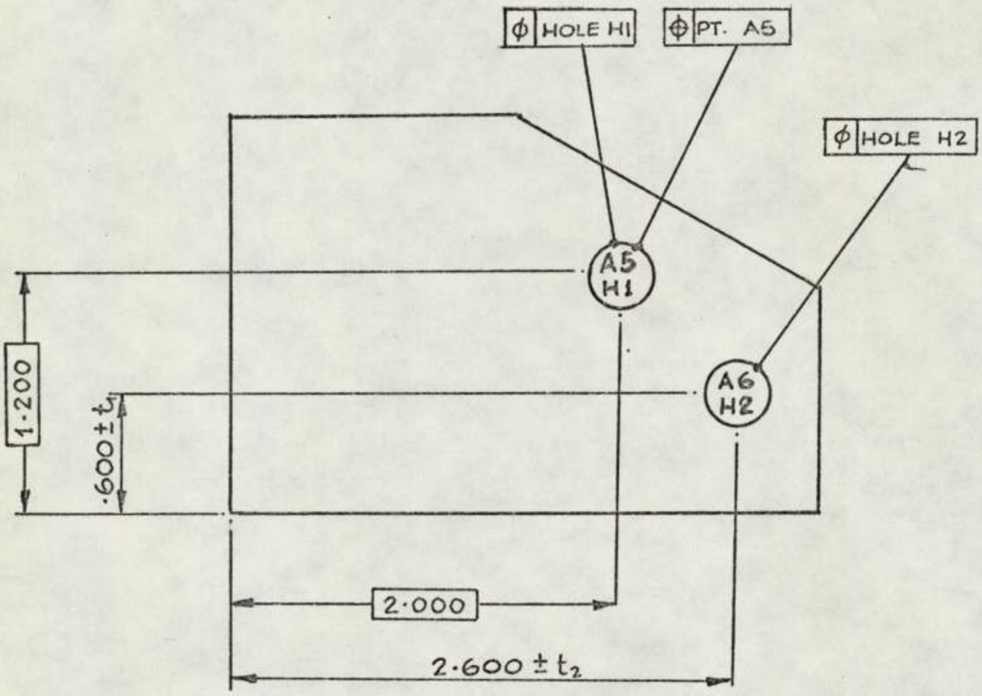
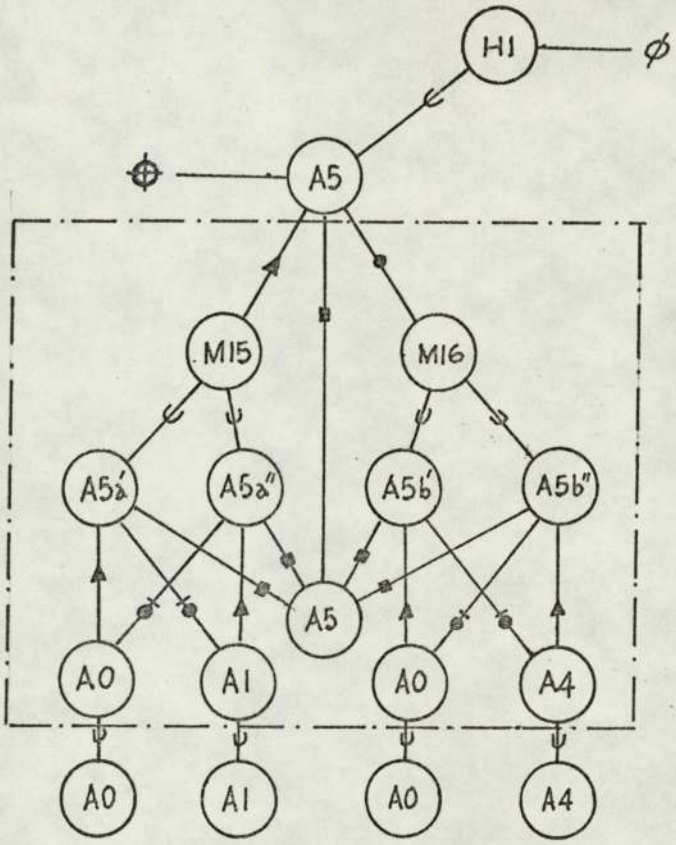
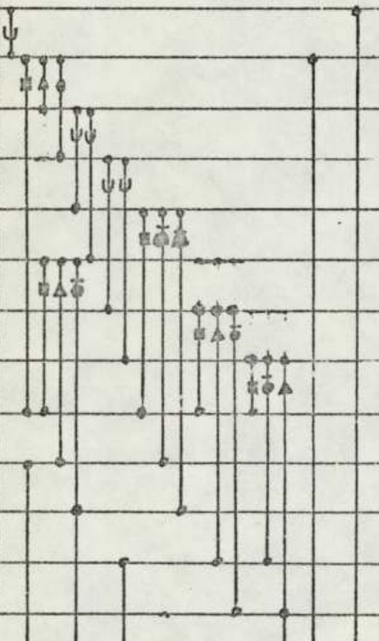


Fig E7. Definition of Drilled Plate



Node	Network	Description	Co-ords			Angs		Links		
			X	Y	Z	θ	φ	■	▲	●
1		Hole H1	0	2 *	1·2	0	0	2	0	0
2		Point A5	0	2	1·2	0	0	9	3	4
3		Remote H-loc M15	0	-10^{6}	0	0	90	5	6	0
4		Remote S-loc M15	0	0	-10^{6}	0	0	7	8	0
5		Dual point A5a'	0	2	1·2	0	0	9	10	-11
6		Dual point A5a''	0	2	1·2	0	0	9	11	-10
7		Dual point A5b'	0	2	1·2	0	0	9	12	-13
8		Dual point A5b''	0	2	1·2	0	0	9	13	-12
9		P-loc A5 - loc. in Z-Y.	0	2	1·2	90	0	0	0	0
10		Point A0 - location	0	0	0	0	90	●	0	0
11		Point A1 - "	0	0	2	0	90	●	0	0
12		Point A0 "	0	0	0	0	90	●	0	0
13		Point A4 "	0	3	0	0	90	●	0	0

● External links
-N Weak links
* Perturbed dimension.

Fig E8 Descriptions of Point A5 and Hole H1

(Fig. E.6). Using separate definitions of points is rather wasteful of nodes, many more unitary links being used than are necessary but it is considered much easier to develop the network point by point using standard cases than to regard it as an entity.

The system may now be used to establish sensitivities and in this case those on the height A4 - A3 were found. The non-zero coefficients are:

Tolerance		Sensitivity at A4 - A3
line A0 - A1	Angularity with D0 - D1	0.577
line A1 - A2	Parallelism with D0 - D1	3.667
line A3 - A4	Parallelism with A0 - A1	0.866
line A2 - A3	Angularity with A1 - A2	2.000
point A2	Symmetry with A1 and A4	0.577

These coefficients may be used to find the tolerance on the height A4 - A3 for existing values of tolerance. For instance if angularity tolerances are 1 in 100 (about $\pm \frac{1}{2}^{\circ}$), parallelism is ± .010 and the tolerance of symmetry ± .020, the tolerance on height A4 - A3 will be ± .083.

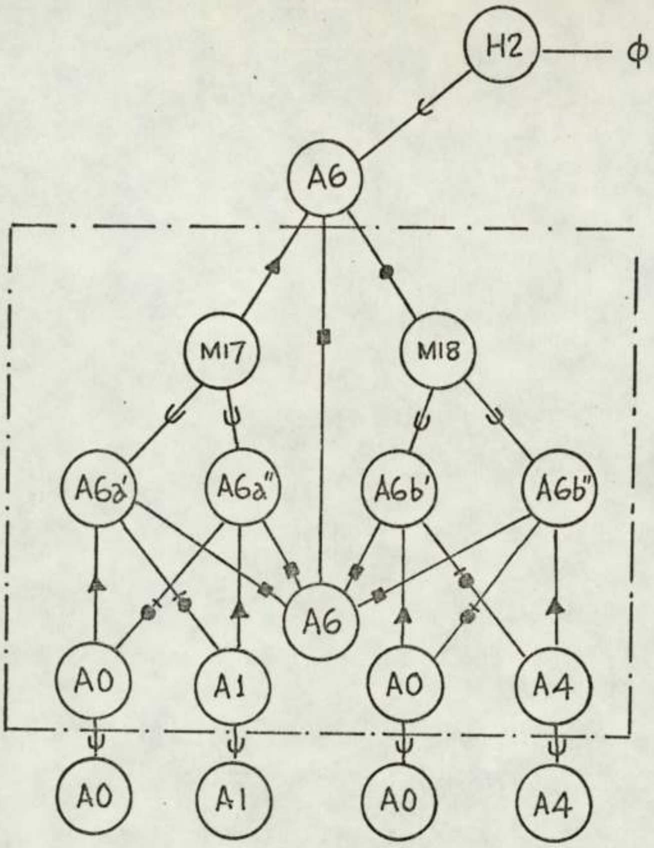
Alternatively, they may be used to allocate manufacturing tolerance either in an informal way or, if cost details are known, by using an optimising program.

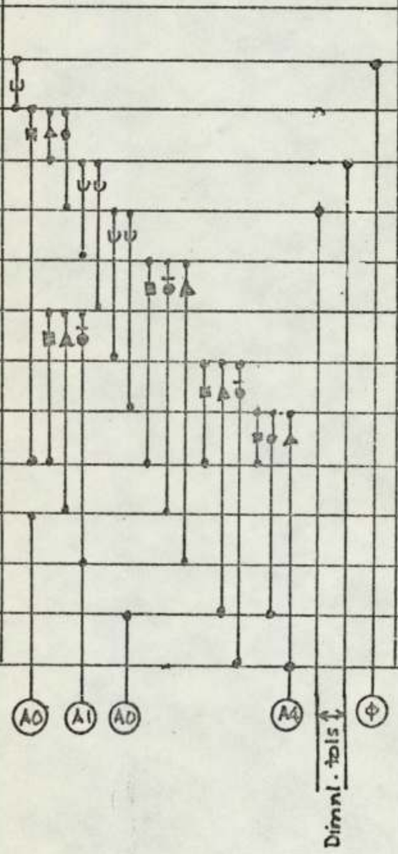
(b) Two dimensional system.

Two holes will now be added to the plate analysed in (a). Both are located by dimensions from sides A0 - A1 and A0 - A4, but centre A5 is positionally toleranced, while centre A6 is dimensionally toleranced (Fig. E.7).

(i) Centre A5. Fig. E.8.

The dimensional system is described by the network shown in the figure. The positional tolerance may be applied directly to A5 - this being a general principle in positional tolerancing. Hole H1 may be





Node	Network	Description	Co-ords			Angs		Links		
			X	Y	Z	θ	ϕ	■	▲	●
1		Hole H2	0	2.6*	·6	0	0	2	0	0
2		Point A6	0	2.6	·6	0	0	9	3	4
3		Remote H-loc M17	0	-106	0	0	90	5	6	0
4		Remote S-loc M18	0	0	-106	0	0	7	8	0
5		Dual point A6a'	0	2.6	·6	0	0	9	10	-11
6		Dual point A6a"	0	2.6	·6	0	0	9	11	-10
7		Dual point A6b'	0	2.6	·6	0	0	9	12	-13
8		Dual point A6b"	0	2.6	·6	0	0	9	13	-12
9		P-loc A6 -loc in Z-Y	0	2.6	·6	90	0	0	0	0
10		Point A0-location	0	0	0	0	90	●	0	0
11		Point A1 "	0	0	2	0	90	●	0	0
12		Point A0 "	0	0	0	0	0	●	0	0
13		Point A4 "	0	0	3	0	0	●	0	0

● External links
-N Weak links
* Perturbed dimension

Fig E9 Descriptions of Point A6 and Hole H2.

located on its centre using a unitary link, the diametral tolerance being applied directly.

(ii) Centre A6. Fig. E.9.

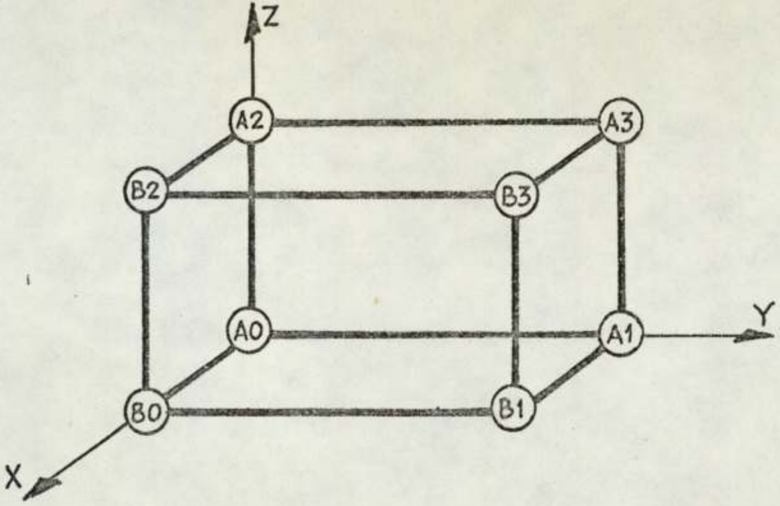
This is located in a similar fashion to A5. The tolerance may be applied by means of a tolerance generating network or if storage is tight, it may be applied to the location network. H2 is located on A6 by means of a unitary link.

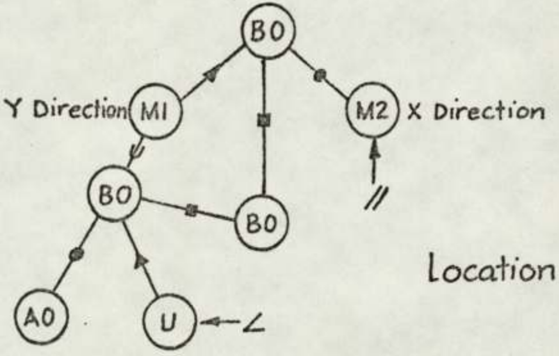
This completes the definition of the plate and holes and sensitivity coefficients may be computed. Two sets are shown below for the distances between points on holes H1 and H2, and between a point on H1 and line A2 - A3.

	Tolerance	Sensitivity at H1 - H2
Dimensions	Parallelism with A0 - A1	0.424
Dimensions	Position tolerance on A5	1.000
Dimensions	Dimensional tolerance on A6 (i)	0.707
	Dimensional tolerance on A6 (ii)	0.707
Hole H1	Radial tolerance	1.000
Hole H2	Radial tolerance	1.000

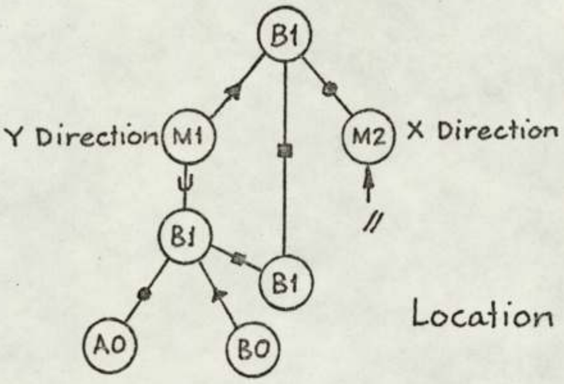
	Tolerance	Sensitivity at H1 - line A2 - A3
line A1 - A2	Parallelism with D0 - D1	1.974
line A3 - A4	Parallelism with A0 - A1	0.250
line A0 - A1	Angularity with D0 - D1	0.097
line A2 - A3	Angularity with A1 - A2	0.832
point A2	Symmetry with A1 and A4	0.500
point A5	Positional with dimensions	1.000
hole H1	Radial tolerance	1.000

If the three sets of sensitivity coefficients are for the influence of tolerances on three critical function dimensions, then they may be

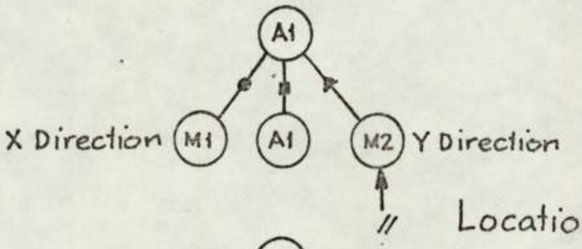




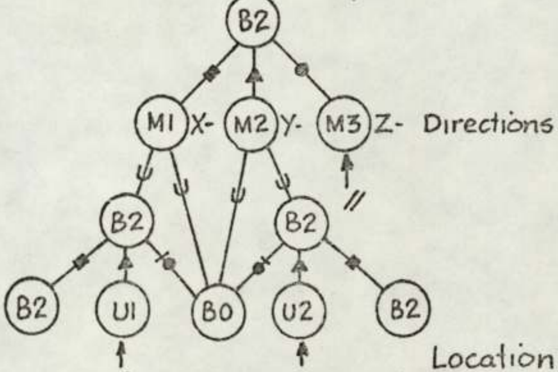
Location of B0



Location of B1



Location of A1



Location of B2
(A2, A3, B3 similar).

Fig E10. Description of Solid Form.

used as input for an optimisation program. The second set was computed for the distance between H1 and a point R1 which was located on line A2 - A3 by a sub-network. The coordinates of R1 were calculated for one computation and measured from a drawing for another run and there was little difference between the coefficients obtained.

Displaying the network in the 'tree' form used previously is not very convenient nor easily read. The form illustrated in Figs. E.2 - E.9 is an improvement but it is important that for clarity, the nodes should be in topological order.

(c) Three dimensional systems.

This example is deliberately very detailed. In practice much of the resulting network (where features of no interest are concerned) may be omitted. The solid form shown in Fig. E.10 will be described.

A plane (the X-Y plane), a line on it (the line OY) and a point on this line (the origin), will be taken as the datum system. Points A0 and A1 are positioned relative to this system. Point A0 is fixed at the origin, but A1 may be displaced along the line OY.

The line A0 - B0 is located on line A0 - A1 with an applied tolerance of angularity relative to A0 - A1 acting at point B0a.

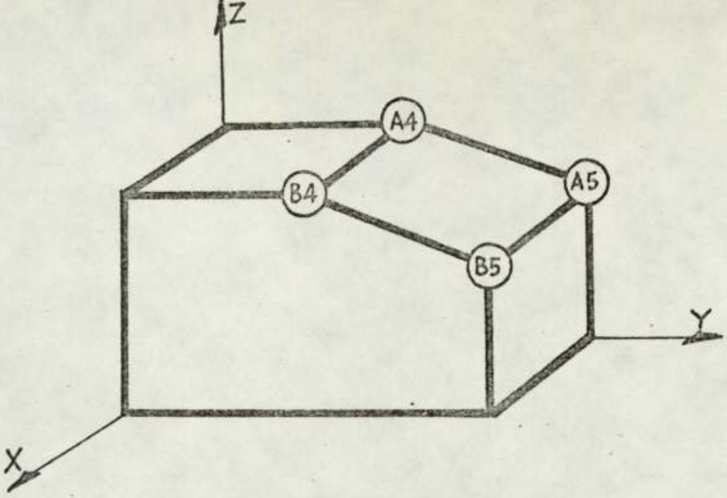
The line B0 - B1 is located on line A0 - A1 with applied tolerances of parallelism acting on B0b and B1a. B0a and B0b displacements are compounded at B0.

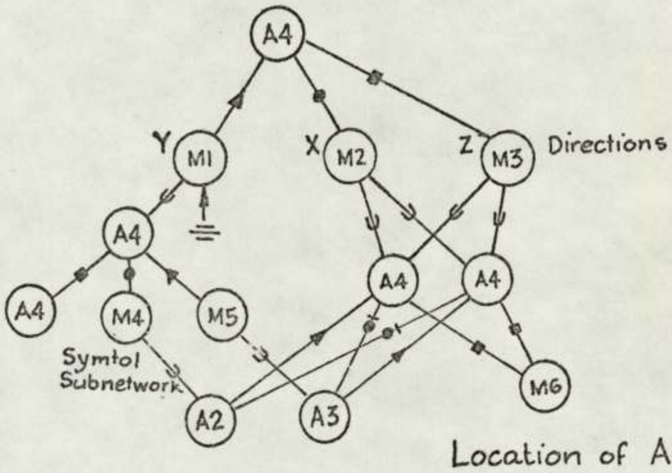
The line B1 - A1 is located on line A0 - B0 with applied tolerances of parallelism acting at B1b and A1. The two tolerances at B1a and B1b are compounded to form the intersection point B1.

In addition, flatness tolerances may also be imposed at B0 and B1. These are not shown in the figure.

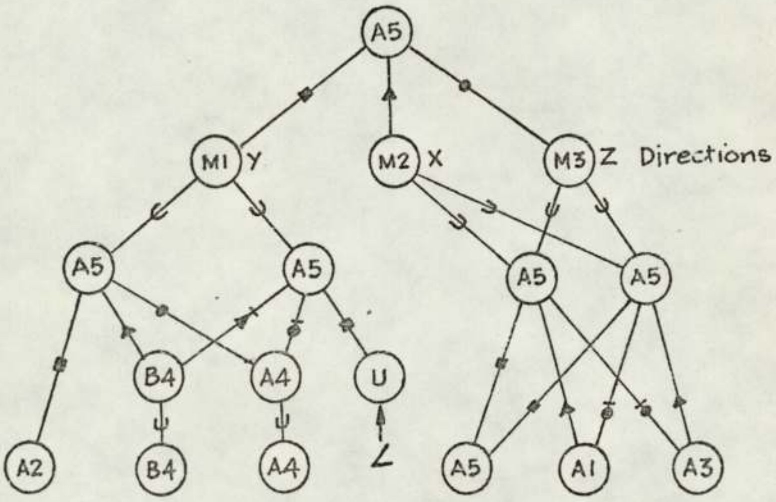
This completes the description of the plane A0 - B1.

Lines A0 - A2, A1 - A3, B1 - B3 and B0 - B2 are defined by locating





Location of A4
(B4 Similar)



Location of A5
(B5 similar).

Fig E11. Description of Plane A4-B5.

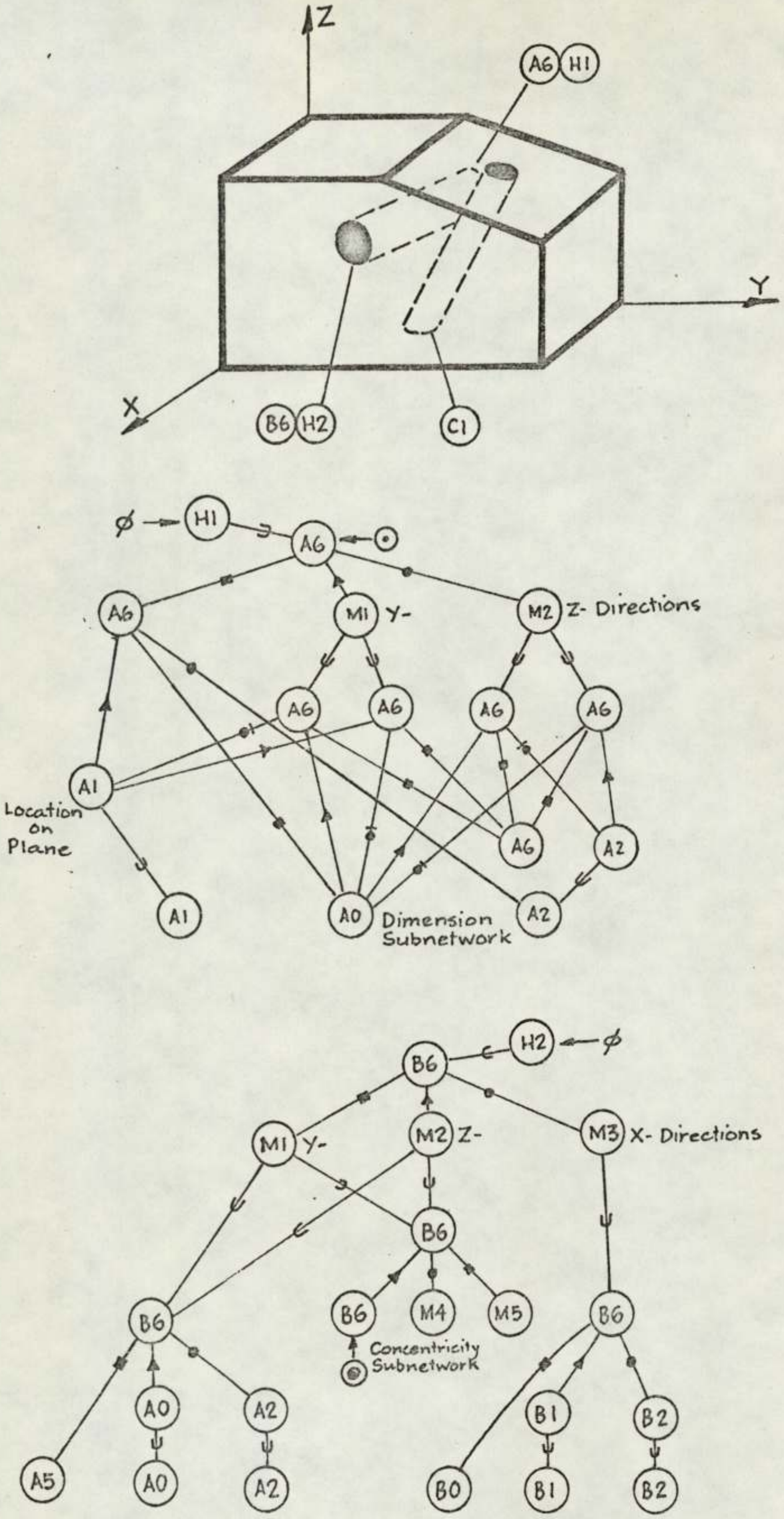
the points A2, A3, B3 and B2 on the plane A0 - B1. Each of these points may have three displacements; one parallel to the XY plane due to the locating point, a squareness tolerance relative to the XY plane and a parallelism tolerance relative to the XY plane. These three displacements, the first extrinsic, the others intrinsic must be compounded in the three orthogonal directions to give the intersection points. Since the displacements are also orthogonal, they may be simply superposed.

This completes the description of the solid form.

Points A4 - B4 are now to be added. A4 is symmetrically toleranced relative to A2 and A3, B4 relative to B2 and B3. The displacements due to this tolerance will be in the OY direction. There will also be extrinsic displacements on these points due to their situation on the lines A2 - A3 and B2 - B3, in the OX and OZ directions. These may also be superposed to give the true displacements at A4 and B4.

The cutting plane A4 - B5 is located on line A4 - B4 with an angularity tolerance relative to plane A2 - B4. It is sufficient to consider this plane as defined by the points A2, A4, B4 for the purpose of applying the angularity tolerance to points A5a and B5a. In addition, since A5 and B5 are necessarily constrained to lie on lines A1 - A3 and B1 - B3, dual points A5b and B5b are located on these lines and the consequent displacements compounded with those at A5a and B5a to give the actual positions of A5 and B5. Fig. E.11 illustrates the procedure.

A hole is to be drilled at an angle to face A1 - A2. The centre at point A6 is located on plane A1 - A2 and is dimensioned from datum lines A0 - A2 and A0 - A1 in the same way as in the two-dimensional case, but the displacement due to its position on the plane is applied to the P-loc. The point on face B1 - B2 where the axis of the hole runs out (point B6) will be subject to an angular tolerance relative to plane A1 - A2 and will also be constrained to lie on plane B1 - B2. These



The hole at C1 is similarly described.

Fig E12. Description of Holes.

displacements will be compounded to give the true run-outpoint B6. The hole H1 is described by setting H-locs along A6-B6, tolerances at A6 and B6 being passed directly by means of unitary links.

A further hole H2 is drilled normal to face A4 - B5. This is described in the same way as H1, except that there need be no complication at the run-out point on face A0 - B1, this being a datum. The network is shown in Fig. E.12.

It is now possible to obtain sensitivity coefficients for the distance between holes H1 and H2. Although this has been a very detailed analysis of the form, much of it being unnecessary for obtaining these results, all the working has made use of a few standard networks and may be done reasonably quickly with a little practice. If it were known in advance that only the sensitivity coefficients between H1 and H2 were required, face B1 - B2 could be ignored and this has been done using seventy nodes only.

APPENDIX F

SUMMARY OF REFERENCES

F.1 Summary of References

Comparatively little work has been published since 1960 on the subject of engineering tolerances. The Secretary of the Institution of Engineering Designers has suggested to the author, in private communication, that this might be because this topic is very much tied up with a company's profitability. About fifty papers, articles and books have been published during the period 1960-1976. Historically, papers published prior to 1960 are concerned with good drawing practice; from 1960-1970, they are concerned with statistical implications and from 1970 onwards, they are mainly about the allocation problem.

(i) Practical treatments dealing with 'good practice'.

A few detailed manuals of dimensioning practice have been written. Possibly the best is ref. T7. This contains details ofall the common dimensioning systems and was written as a companion volume to BS 308 (ref. G5) which was the first wholehearted attempt to systematise design practice in this country. A further, extended version of BS 308 was published in 1972 and this is widely regarded as the standard for drawing practice. Ref. T7 contains much that is relevant to the latest version of BS 308, but has not been re-printed by the publisher.

There are several excellent papers and articles which may be found in the list of references. Refs. T1, T2, T13 may be singled out as being particularly useful.

(ii) Analysis of Statistical Tolerancing

The principle of infallible interchangeability (sometimes called sure-fit) assumes that all the critical dimensions of a manufactured part are at an extreme limit of the allowed tolerance range. It has long been recognised that this is normally an unlikely eventuality and that this pessimistic approach often results in parts which are specified with unnecessary precision. This causes high unit manufacturing costs and excessive rejection rates. A common assumption is that the bulk of the tolerances involved in an assembly will be distributed in Gaussian fashion and in cases where this is not so, the overall 'stacked' tolerance will be approximately Gaussian, as proved in the Central Limit Theorem. Thus, the majority of assemblies will have critical clearances which are reasonably close to the nominal. An excellent introduction to this treatment of statistical tolerancing may be found in ref. S3, and a more advanced description in ref. S8.

In practice, tolerances are not quite so well behaved and often, for a variety of reasons, the distribution is not Gaussian. For instance, it is good practice to allow for tool wear by starting to remove metal at one end of the tolerance range and to drift towards the other limit as the tool wears. A batch of parts machined in this way may, then, all be close to extreme tolerance limit. Another example is in the drilling of a hole through a locating bush. The hole centre will very likely be at extreme tolerance.

A detailed discussion of these situations may be found in ref. T1.

It seems safer to assume a more general distribution than the Gaussian for the component tolerances of highly critical clearances. A description of a computer package for the statistical analysis of tolerances with general distributions may be found in ref. S5. The designer uses the system interactively with a graphic console. He provides the system with an expected statistical distribution for each tolerance, together with a sub-program (written in PL/1) describing the geometrical relationships between each dimension. The computer then generates representative critical dimensions for each component, sampling from the appropriate distributions. The number of simulations is typically of the order of 1,000 and results are displayed graphically in various forms. Tolerances may be adjusted interactively by the user.

The system has been widely used in the Body Division of General Motors and is regarded as a useful design tool.

The main drawback of the system is that the geometric form of the dimensioning system must be specified by a sub-program either written by the user or submitted by him to a specialist programmer. This seems inconvenient at best and certainly unsatisfactory in a firm smaller than General Motors. Another problem is that it does not seem possible, as far as can be judged, to specify complex multi-stage machining processes in which dimensions at a stage depends on those obtained at a previous stage. The system, however, is the first to use a computer for this purpose and will, no doubt, be progressively refined as further experience is gained in its use. It would, clearly, be an improvement if the dimensioning and datum system were submitted to the program not as a sub-program but as data.

(iii) Allocation of Tolerances

Although the problem of allocating tolerances to the component dimensions of a critical clearance has been recognised for many years, the first paper to be published in this field was ref. A4. The problem is clearly defined and solved in ref. A2. This is an elegant account but uses an inverse square law for the tolerance cost which has been superseded in later papers by more realistic models. Possibly the most important paper is ref. A8. This uses a negative exponential model (now commonly called 'the Speckhart model') for the tolerance cost function. The method of Lagrange multipliers is used to minimise the total cost and various practical examples are analysed. The author has developed a program to calculate optimum tolerance allocations on both sure-fit and statistical-fit bases.

Another approach to the minimisation of tolerance cost is to use dynamic programming. An account may be found in ref. A5.

The most detailed and comprehensive description to date is a two-part paper ref. A1. Earlier papers on the subject are reviewed, cost-tolerance data obtained from various sources analysed and practical models derived for different manufacturing processes. A mathematical analysis is performed for sure-fit cases and the method is applied to several practical examples. A further paper dealing with statistical-fit cases is promised.

An account of the problem and method of solution will be found in Appendix C. This is largely eclectic drawing mainly on refs. A1 and A8 and is included for reference.

It is interesting to note that ref. A1 pre-dates ref. A8, is a fuller treatment and the cost equations are based on a thorough investigation of practical results and yet the latter paper seems to be considered definitive in the literature.

(iv) Geometric Calculation of Tolerances

This topic in tolerancing theory has not been dealt with systematically. Various papers have been written on specific problems (refs. C1 and C2) but the examples quoted are either trivial or too specialised to be of much general interest. It is hoped that this paper might fulfil a need in this respect.

REFERENCES

For convenience, references are separated into 5 classifications, each being given a distinct prefix.

G . . . General references which are mainly textbooks used in the theoretical development.

T . . . These refer to general principles of tolerancing and dimensions.

S . . . Deal with statistical considerations.

A . . . Concerned with the problem of allocation of tolerances.

C . . . Deal with the calculation of tolerations geometrically.

GENERAL REFERENCES

G.1 Berztiss, A T Data Structures Pub. A P

G.2 Bickley, W G and Thompson, R S H Matrices Pub. E U P

G.3 Boullion and Odell Generalised Inverse Matrices Pub. Wiley

G.4 Box, M J, Davies, D and Swann, W H Non-linear Optimisation Techniques ICI Monograph No. 5 Pub. Oliver & Boyd

G.5 British Standards Institution BS 308 Engineering Drawing Practice
Part I General Principles
Part II Dimensioning and Tolerancing of Size
Part III Geometrical Tolerancing

G.6 Chace, M A Analysis of 3-D Mechanisms PhD Thesis Univ. of Michigan 1964

G.7 Ferrar, W L Finite Matrices Pub. Clarendon Press

G.8 Fox, L An Introduction to Numerical Linear Algebra Pub. O U P

G.9 Fox, L and Mayers, D F Computing Methods for Scientists and Engineers Pub. O U P

G.10 Gerrish, F Construction of Defective 3 x 3 Matrices having prescribed Eigenvalues and Eigenvectors Note 279, Mathematical Gazette, December 1972

G.11 Hamburger, and Grimshaw, Linear Transformations Pub. C U P

G.12 Kempster, M H A Principles of Jig and Tool Design Pub. E U P

G.13 Knuth, D E The Art of Programming Vol I Pub. Addison-Wesley

G.14 Shigley, J E Kinematic Mechanisms Pub. McGraw-Hill

G.15 Smyth, W F and Radeceanu, E A Storage Scheme for Hierarchic Structures Computer Journal Vol 17 No. 2

G.16 Yefimov, N V Quadratic Forms and Matrices Pub. Academic Paperbacks

TOLERANCING PRACTICE

T.1 Abbott, W The Dimensioning of Engineering Drawings Pub. Blackie

T.2 Burr, I W Simplify Dimensioning and Tolerancing to Cut Cost
Design Engineer V 19 No. 2 1973 pp. 34-41

T.3 Conway, H G Engineering Tolerances Pub. Pitmans

T.4 Hatton, W J Design Tolerance: Relationship to Planning
Work Study V 25 1976 pp. 24-26

T.5 Holland, T Ten Ways to Design Expensive Tolerances
Engineering V 215 1975 pp. 642-644

T.6 Oddie, D A Geometric Tolerancing of Engineering Drawings
Quality Engineer V 34 1970 pp. 12-13

T.7 Parker, S Drawings and Dimensions Pub. Pitmans

T.8 Spencer, J and Cheney, R L Comprehension of Numerical
Information in Engineering Drawings Engineering Designer
Aug 1971 pp. 5-9

T.9 Stewart, G Tolerances for Production Australasian Engineer
Mar 1968 pp. 42-53

T.10 Tallack, W J Some Effects of Tolerances in Measurement
Quality Engineer V 30 1966 pp. 172-173

T.11 Tarasevich, Y and Yavoich, E Fits, Tolerances and Engineering
Measurement Pub. Foreign Languages Publishing House, Moscow

T.12 Wakefield, L P Dimensioning Hole Centres Engineering Designer
Part I Nov 1962 pp. 4-6
Part II Jan 1963 p. 7

T.13 Wakefield, L P Dimensions, Tolerances: Implications
Engineering Designer Part I July 1970 pp. 5-11
Part II Aug 1970 pp. 9-14

STATISTICAL ASPECTS

S.1 Burr, I W and James, R E Specifying the Desired Distribution in Lieu of Tolerance Limits A S M E paper n 74-DE-9 1974

S.2 Burrows, G L Statistical Tolerance Limits - What are they? Applied Statistics V 12 June 1964 pp. 133-144

S.3 Donaldson, C S Fit and Tolerance Engineering Materials and Design V 11 1968 pp. 923-929

S.4 Gilson, J A New Approach to Engineering Tolerances Pub. The Machinery Publishing Co. Ltd.

S.5 Gugel, H W Monte Carlo Simulation with Interactive Graphics General Motors Corporation, Research Publication GMR-1531-1974

S.6 Mansoor, E M Application of Tolerances used in Engineering Designs Inst. of Mech. Eng. Proc. 178 Pt. 1 No. 1 1963-64 pp. 29-51

S.7 Mansoor, E M Selective Assembly - its Analysis and Applications Int. J. Prod. Res. V 1 No. 1 1961 pp. 13-24

S.8 Spotts, M F An Application of Statistics to the Dimensioning of Machine Parts A S M E Jour. of Engr. Ind. 1959 pp. 317-322

S.9 Spotts, M F Dimensioning of Clearance Fits with Overlapping Tolerances using Probability Theory A S M E Paper n 74-DE-8 1974

ALLOCATION OF TOLERANCES

A.1 Bennett, G and Gupta, L C Least Cost Tolerances Int. J. Prod. Res. Part I Vol 8 No. 1 1970 pp. 65-74
Part II Vol 8 No. 2 1970 pp. 169-181

A.2 Hillier, M J Systematic Approach to the Cost Optimisation of Tolerances in Complex Assemblies Bull. of Mech. Eng. Education Vol 5 Apr 1966 pp. 157-161

A.3 Hindmarsh, G W Manufacturing Tolerance Allocation Prod. Engr. Vol 20 Jul 1973 pp. 257-263

A.4 Latta, L W Assignment of Least Cost Tolerances Engr. Digest Vol 25 Jan 1964 pp. 86-87

A.5 Moy, W A Assignment of Tolerances by Dynamic Programming Machine Design Vol 36 No. 12 1964 pp. 215-218

A.6 Peters, J Tolerancing the Components of an Assembly for Minimum Cost A S M E paper 70-Prod-9 1970

A.7 Smathers, E W and Ostwald, P E Optimisation of Component Functional Dimensions and Tolerances A S M E paper 72-DE-18 1972

A.8 Speckhart, F H Calculation of Tolerance Based on a Minimum Cost Approach A S M E paper VIBR-114 1971

A.9 Spotts, M F Allocation of Tolerances to Minimise Cost of Assembly A S M E paper 72-WA/DE-6 1972

A.10 Wilde, D Simplifying Discrete Tolerance Assignment A S M E paper 75-DET-106 1975

A.11 Wilde, D and Prentice, E Minimum Exponential Cost Allocation of Sure-fit Tolerances J. Eng. Ind. Trans. A S M E Vol 97 Ser B No. 4 1975 pp. 1395-1398

GEOMETRIC CALCULATION

C.1 Dacosse, J Note sur la théorie des écarts et tolérance de fabrication Méthode de détermination des tolérances complexes Revue de la Mecanique Vol 5 No. 4 1959 pp. 167-179

C.2 Knappe, L F A Technique for Analysis of Mechanical Tolerances Machine Design Vol 35 No. 10 Apr 1963 pp. 155-157