THE LOGICAL AND ANALYTICAL STRUCTURE OF THE COMPUTER-AIDED DESIGN PROCESS AS APPLIED TO A CLASS OF MECHANICAL DESIGN PROBLEMS

By

TSE-SHENG LING

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in the
University of Michigan
1964

Doctoral Committee:

Associate Professor Franklin H. Westervelt, Chairman
Associate Professor Steven A. Coons
Associate Professor Bernard A. Galler
Professor Joseph E. Shigley
Lecturer Dean H. Wilson

ABSTRACT

Before the engineer can make effective use of the digital computer as an aid in solving design problems, many fundamental problems will have to be solved. First of all, efficient and effective means of communicating the engineer's concepts to the computer must be developed. Second, the machine must be provided with a variety of displays by which the processed data may be studied, assimilated and put to use by the engineer. The light pen and cathode ray tube media developed at M.I.T. and known as "Sketchpad" provides an interesting and potentially useful interface between man and the computer.

Through the use of "Sketchpad", it is shown in this thesis that "creative solution" to many design problems can be most efficiently accomplished once the human computer "design team" has established the design features. This Computer-Aided Design Program lets the computer be an active partner with the designer, with the computer accepting and analyzing the designer's sketches and performing

all or a substantial amount of the necessary design calculations.

Three examples on automatic design of mechanical systems are given:

(1) The design of a shaft system and further design of bearings to support this shaft system so that all the constraints are satisfied.

(2) The preliminary design of aircraft wings where key design parameters are determined after several views of an aircraft wing are sketched. These design parameters are ready for use in further calculation of aerodynamic properties.

(3) A numerical control program, through which the process of part programming is by-passed. A set of part programming instructions are produced automatically once the desired part is sketched. These instructions can then be used to produce the numerical control tape ready for use by a numerical control machine.

The generality of this program has been kept as first priority. Many other design problems, either mechanical, electrical, industrial or other scientific problems can readily be designed through this program once special subroutines for that particular problem are written.

ACKNOWLEDGEMENTS

I am most grateful to my thesis committee for the continuous interest, support and counsel throughout this work, in particular to Professors Westervelt and Galler who gave many valuable suggestions and assistance.

I would like to express most sincerely my many thanks to Professor Coons and Mr. Timothy Johnson of Massachusetts Institute of Technology for their permission in joining their work on Computer-Aided Design Project. Special thanks are due to Mr. Johnson who generously gave his time and patience to provide me much information on the M.I.T. Sketchpad Systems and the use of the Lincoln Laboratory TX-2 Computer. Without this special arrangement and informative assistance, the work on this thesis would not have been possible.

I am indebted to Prof. Gordon J. VanWylen who has continuously given me encouragement and assistance in every way since the start of my graduate work. The contributions of Dr. M. Eugene Merchent and Dr. Richard R. Weber of the Cincinnati Milling Machine Company and Prof. Kenneth C. Ludema are also greatly appreciated.

This work was supported by the Cincinnati Milling Machine Company, E. I. du Pont de Nemours and Company and the Institute of Science and Technology of the University of Michigan. To these organizations and to their management, I would like to extend my deep appreciation.

Finally, this work would have been more difficult had it not been for my wife Christine who assisted me wherever possible.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

A. Computer Aided Design Project*

During late 1959 a computer-aided design project was initiated at the Massachusetts Institute of Technology, Cambridge, Massachusetts. This project has been a joint endeavor of the Computer Applications Group of the Electronic Systems Laboratory, Electrical Engineering Department, and the Design Division of the Mechanical Engineering Department both of M.I.T. The work was sponsored by the Manufacturing Technology Laboratory, Aeronautical Systems Division, Wright-Patterson Air Force Base, Ohio. Based upon their backgrounds and different disciplines, these groups are taking complimentary approaches to the problem of how to use the computer to assist humans in the design process. It is hoped that this research activity will lead to the specifi-

---

*Most of sections A, B, C and D of this Chapter are adopted from M.I.T. Computer-Aided Design Project Reports. See references 1, 2, 3, 5, and 16 of the Bibliography.

cation of a man-machine system in which a designer and a computer can work together as a team on design problems requiring creative solutions. The long term goal is automatic manufacture once the human-computer "design team" has established the features of a design.

B. Input-Output Requirements

Before the engineer can make effective use of the modern high speed digital computer as an aid in solving many mechanical, electrical, industrial or scientific design problems, many fundamental problems will have to be solved. First of all, efficient and effective means of communicating the engineer's concepts to the computer must be developed. Second, the machine must be provided with a variety of displays in real time by which the processed data may be studied, assimilated and put to use by the engineer.

The standard input-output system of the computer today forces man to reduce his communications to written statements suitable for typing. Many computer languages have been developed for this purpose. Among those widely accepted languages are FORTRAN, ALGOL, COBOL AND MAD.* However, written languages are too

*MAD, Michigan Algorithmic Decoder, developed at the computing center of the University of Michigan is one of the most powerful computer languages ever developed. It is suitable for use in scientific as well as data processing problems.

slow for communication with the computer. Furthermore, they are too cumbersome for expressing many kinds of information, particularly shape description.

C. Man-Machine System

Many design processes begin with a graphical description of a proposed device or system. Rough sketches of perceived ideas precede a precise statement of the refined details of the concept. Thus, the Computer-Aided Design Project at M.I.T., being faced with this need for a means of rapidly communicating structural objects to the computer, developed Sketchpad which makes use of a cathode ray tube (CRT) and a light pen.

The light pen is a hand held photocell which reports immediately to the computer whenever a spot on the CRT screen falls within its small field of view. In order that the computer may realize the exact location on the screen where the light pen initially aimed just prior to sketching, one of the techniques used is random search. That is, the computer displays many single spots one at a time randomly until one of them is seen by the light pen. Another technique currently adopted is displaying a bright spot at a fixed location on the screen, acting as an "ink", and the sketching by the light pen always starts at this position. Once the computer knows where the light pen is, it can follow the motions of the light pen on the display screen very rapidly while a sketch is made by the designer.

D. M.I.T. Sketchpads

The Sketchpad system makes it possible for a man and a computer to communicate with each other very rapidly through the medium of graphical drawings. It enables the designer to manipulate pictures he drew on the display scope in many interesting and useful ways by means of light pen, push buttons and knobs.

In January, 1963, Dr. Ivan E. Sutherland published a technical report based on his doctoral thesis submitted to the Department of Electrical Engineering at the Massachusetts Institute of Technology. The report was entitled "Sketchpad: A Man-Machine Graphical Communication System". The present thesis makes use of Dr. Sutherland's Sketchpad. A more detailed description of his Sketchpad will be given in the next chapter.

Following Dr. Sutherland, Mr. Timothy Johnson of the Mechanical Engineering Department, Massachusetts Institute of Technology published his M.S. thesis

"Sketchpad III, Three Dimensional Graphical Communication with a Digital Computer" in May, 1963.

Sketchpad III is only capable of manipulating straight line "wire frame" figures in three-dimensional space. A knowledge of computers and program writing is not required to operate the system. The definition, construction, and manipulation of three-dimensional surface are not as yet included; hence edges which are normally hidden by forward surfaces are not obscured as they should be. Since all edges are visible, one views a "wire frame" with no covering. Explicit information about the topology of the part is stored as it is sketched. Parts of an object (lines or end points) can be moved in space without erasing. All attached lines will follow the moving part.

Since the display screen is two-dimensional and the objects are three-dimensional, four views of the object are displayed by the program, one in each quadrant of the CRT screen. A perspective view of the object appears in the upper right quadrant, and three orthogonal views in the remaining quadrants: top view--upper left, front view--lower left, and side view--lower right. Any changes or movement in any one view of the graph will cause corresponding changes simultaneously in the remaining three views. This reinforces depth perception.

E. M.I.T. Forthcoming Sketchpad*

An extremely powerful new Sketchpad will soon be introduced by M.I.T. This new Sketchpad is a result of joint efforts by the Computer Aided Design Project personnel at the Massachusetts Institute of Technology. The definition, construction and manipulation of three-dimensional surface will be included. The smooth fair surfaces can be defined by a minimum of design curves. As a matter of fact, the designer can draw as few as two curves and obtain from them a surface automatically. He can then proceed to modify this original surface by drawing additional curves so as to refine and more accurately specify his requirements. This means that in the future Sketchpad can be used to design an airplane fuselage or automobile body style or any other complex shaped three-dimensional object.

One additional useful feature of this forthcoming new M.I.T. Sketchpad system is the capability of

---

*This section is the result of the author's conversation with Mr. Timothy Johnson.

specifying any property of interest on the picture drawn. The unit of electrical resistance, in ohms, or the stiffness of a mechanical spring, in pounds per inch, can be specified at the same time the picture representing the resistance or spring are drawn.

F. Description and Objectives of this Thesis

In order to treat the entire design problem, it becomes necessary to investigate the means of storing, organizing, searching and recalling large amounts of information pertinent to the problem. The recognition of the geometry by the computer is a prerequisite. The computer must be capable of feeding key dimensions and various units as well as other design parameters into subroutines for selected problems. The system should in the same time feed back the properties of the system sketched--either mechanical, electrical or other technological properties--in real-time at the display scope so that the designer can, upon receiving the current information about the system he drew, make intelligent and efficient modifications until such time that the system he is designing meets all of his requirements. These requirements can be weight, cost, volume, temperature, deflection, speed and many others.

This thesis describes a new general data structure which serves two functions: It makes possible the recognition of the geometry of a "simply connected" drawing sketched on the Sketchpad. It also conveniently stores data as well as the design parameters pertinent to the design problem. Thus, this new data structure contains all the information required for the design problem. This is demonstrated by the following three design examples:

(1) The design of a loaded shaft supported by bearings. The designer can modify the shaft system by either changing the size of the shaft, number of supporting bearings, bearing stiffness or locations of the support until the shaft deflection displayed in the scope falls within the given limit. The program will then proceed to design the bearing that will support this shaft and yet meet all the other constraints that may be imposed on the design of the bearing.

(2) The preliminary design of aircraft wings. This thesis will demonstrate how an object requiring more than one view to describe can still be designed through Dr. Sutherland's Sketchpad in conjunction with the author's system. The pertinent

parameters for the calculation of aerodynamic properties are found very easily through the new data structure.

(3) A numerical control program. A numerically controlled machine tool tape can be produced after sketching of an object at the CRT screen and after keying in other parameters such as cutting tool radius, part tolerance, etc. This tape can then be fed into a Cincinnati N/C (Numerical Control) three axis milling machine for actual manufacturing of the part sketched. This is an example of how the long process of part programming in numerical control field can be by-passed in the future.

The present system developed in this thesis is by no means restricted to the above three applications. Any number of other design problems can be incorporated most readily by simply writing one or two subroutines for that particular problem and allow the system to call for the subroutines.

CHAPTER II

THE SKETCHPAD II SYSTEM AND BASIC REQUIREMENTS FOR DESIGN

A. Some Features of Sketchpad II*

The sketchpad system developed by Dr. Sutherland allows one to draw lines and circles, to move existing parts of the drawing around and to point at particular parts of the picture in order to position them or to erase them. In addition, it provides the following three most important capabilities.

(1) A subpicture capability--to allow the symbolic naming of geometric entities and constructions. On command, the symbolic names may be used to generate these subpictures again and again. Moreover, the defining constructions are stored in such a way that the constraints are retained in an abstract sense. By extension, groups of more elementary constructs may be defined and recalled, replicated and modified in a succession of

---

*Sketchpad II in this thesis refers to the Sketchpad System developed by Dr. Sutherland. Readers who are interested in further details of Sketchpad II are directed to Dr. Ivan E. Sutherland's report "Sketchpad: A Man-Machine Graphical Communication System", technical report, no. 296, 30 January 1963, Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, Massachusetts.

hierarchial levels.

(2) A constraint capability--to relate the parts of a drawing in any computable way. One can make lines vertical, horizontal, parallel or perpendicular; he can also make points lie on lines or circles, make symbols appear upright, or of equal size, etc.

(3) A definition copying capability--to build complex relationships from combinations of several simple atomic constraints. For example, a complex constraint may consist of two atomic constraints such as to make two lines parallel and equal in length.

B. Data Storage Structure of Sketchpad II

In the Sketchpad II system, the information on the topology of the drawing is stored in the form of so-called generic blocks. A generic block is a collection of several consecutive registers in storage that contain all the information which distinguish a type of thing from all other types of things. All the references made to a particular generic block are collected together by a string of pointers. These pointers form the Ring Structure. The Ring Structure requires two registers in each generic block. The first one directly points to the first location of another higher level generic block which in turn indicates the type of element. The second register is used to string all the similar references together. The left half of this register points to another register whose right half always points back. In other words, the left half points to "where it came from" and the right half points to "where it goes to". The basic ring consists of two kinds of these register pairs, namely "HEN" and "CHICKEN". The "HEN" pair is contained within a generic block which will be referred to, such as a point block, and the "CHICKEN" pair is contained within a block making reference to another, such as line block. Thus, all the line blocks refer to a point block which is either the starting point or the terminating point of each line.

A drawing sketched through Sketchpad II will consist of many generic blocks; they are master picture block, line block, point block, circle block, instance block and many constraint blocks, etc.

As indicated previously, one of the many powerful features of the Sketchpad II System is its capability to generate subpictures called INSTANCE, again and again on command. A picture may be drawn to represent an electrical resistance, loading vector or anything else. The instance (subpictures) of these symbols can be made to appear on the display scope as a figure geometrically similar to the original picture of which it is an instance.

There may be many different constraint blocks in the topology storage of a drawing. Some of them may be IPCON (instance point constraint) to constrain an attacher (a point in an instance to be attached to any point on other pictures) of an instance to coincide with a particular point specified by the light pen, or ONLINE (online constraint) to restrict an end point of a line to coincide with another line.

These various generic blocks are then interlocked in very complicated ways by various rings. Figure 2-1 shows an example of a possible data structure of a drawing. It is clear that the interconnections among generic blocks in Sketchpad II data structure are rather complicated. Figure 2-1 shows only a few of the many rings that string each block in various ways. The PPART Ring in a Master picture block strings all the instances, lines and circles in the drawing. The PLS Ring in a point block strings all the lines or circles whose line (or circle) starting point or line (or circle) ending point is this point itself. On the other hand, IPCON, ONLINE and many other types of constraint blocks are stringed together by VCON ring

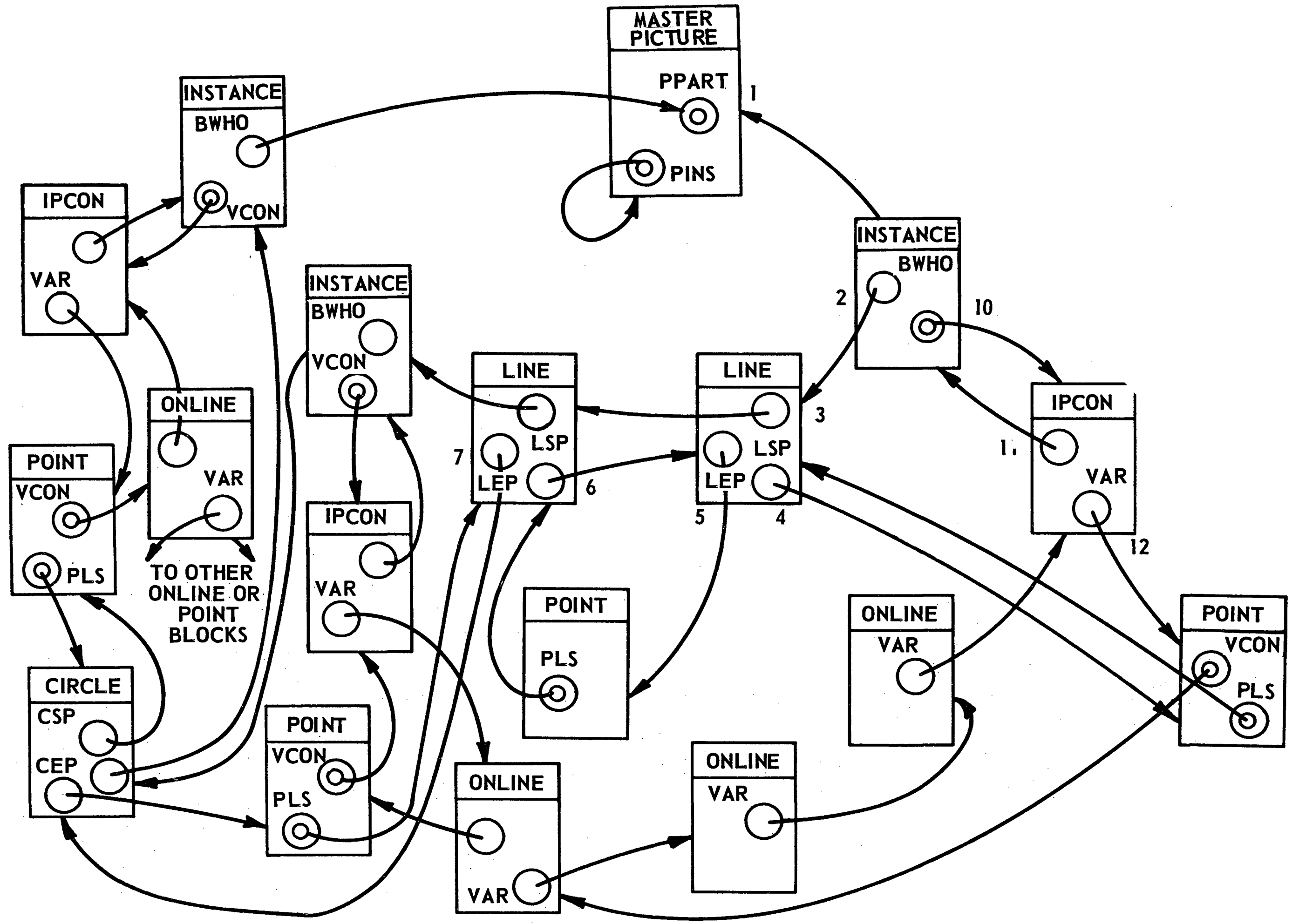


FIGURE 2-1
SIMPLIFIED DATA STRUCTURE OF SKETCHPAD II

of the point block. The manner in which those blocks are tied together by various rings depends on the order in which the designer sketches his lines, points and circles of the picture and, to some degree, the order in which various constraints are applied to the drawing.

C. Basic Requirements for Design with the Sketchpad

After the system to be designed is sketched at Sketchpad, design cannot proceed until the Sketchpad system is modified. It must be able to recognize the geometry of the system sketched, and it must have a *general* data structure through which any particular subroutine can be written for mathematical analysis of the system being designed.

These basic requirements for design with Sketchpad are fulfilled by the program to be described in this thesis.

CHAPTER III

DESIGN PROCESS

Referring to the flow diagram shown in Figure 3-1, a designer will start with a design idea. Ordinarily, he would then like to sketch his idea on paper with a pencil. Using Sketchpad he can sketch his idea in the CRT screen by a light pen. After the system to be designed is sketched on the screen, he should indicate to the computer what type of problem he is to study on this system. He may be interested in the deflection of the shaft he sketched or its stress distribution or its vibrational characteristics. Or, he may be interested in the total weight of the airplane wing or its fuel carrying capacity (wing volume) or its aerodynamic characteristics. Only after receiving an indication of the type of problem to be studied can the computer proceed to the proper subroutine for its calculations.

After the type of problem is defined and corresponding subroutines are executed, the designer would like to see in real-time how the system behaves. He could display, for example, the shaft deflection curve or display the numerical values of other

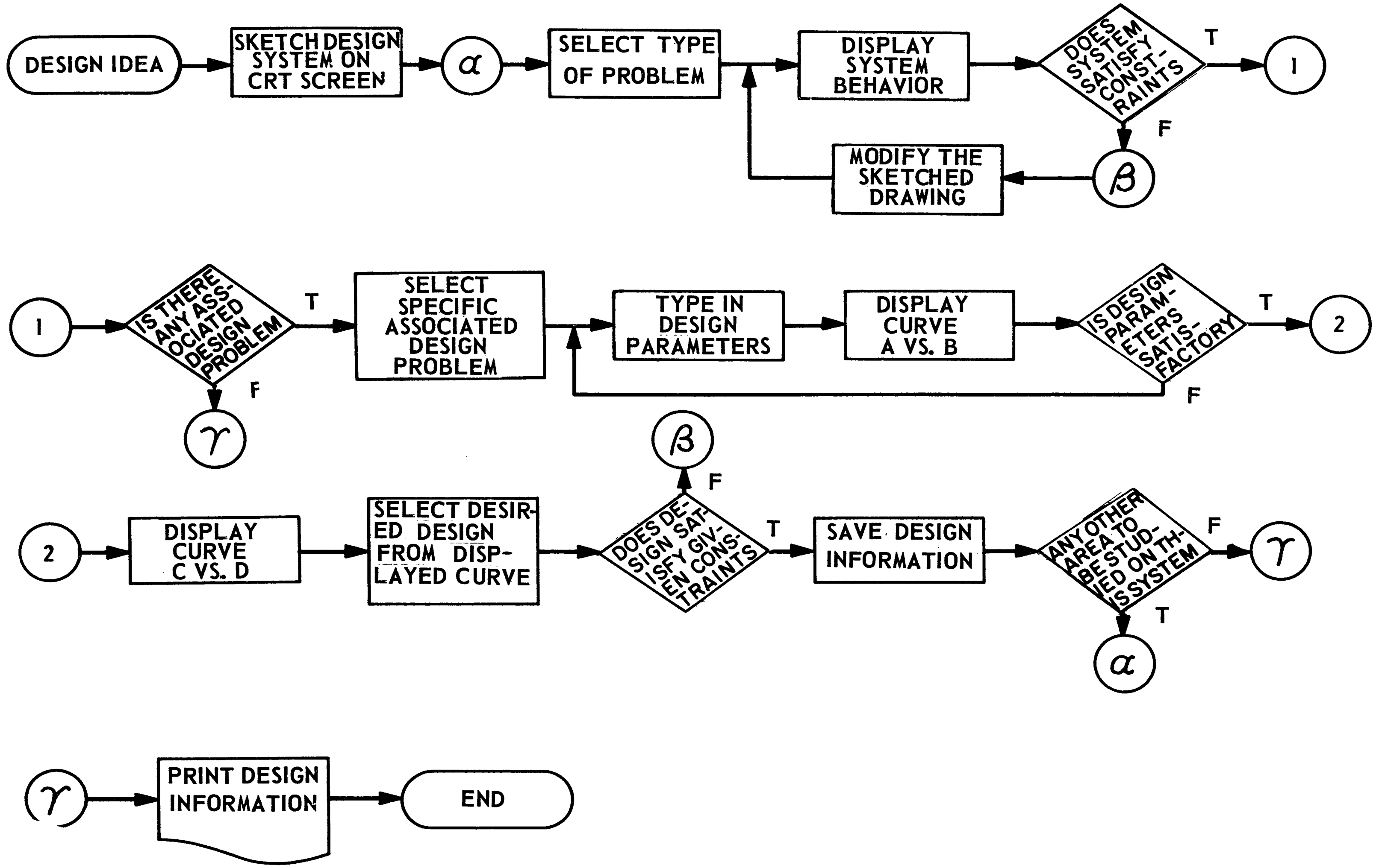


FIGURE 3-1
DESIGN PROCESS FLOW DIAGRAM

properties of the system at a corner of the display screen.

Upon receiving this information, the designer can modify his sketch by a light pen if the given specifications on the system sketched are not met. For example, in the case of shaft deflection, he could adjust the size of the diameter of the shaft, magnitude of the supporting vector, total number of supports or the location of the supports. Any of these picture modifications by light pen would show in real-time the change in the shaft deflection. If the designer is experienced, he can make a rather intelligent modification on the system based on the current information and trends he sees in real-time at the scope. If he is inexperienced, he may after several attempts learn the trend of the changes of system characteristics as he modifies the picture. This is one of the very powerful advantages of using Sketchpad in design.

After these modifications, the system now satisfies all the given constraints. Next, there may be some design problem which is associated with the original system. For example, a designer may wish to design a bearing to support the recently sketched shaft in order to meet not only deflection specifications but also speed, temperature and other requirements.

During this later design stage the display system can again play a very important part. All displays used to this point are stored. A new problem requiring conventional design technique and related to the previous displays can then be studied. The scope can be used to display this new information in the form of curves. We could assign the ordinate to be the load carrying capacity of the bearing and the abscissa to be its eccentricity ratios, or let the ordinate represent the cost of manufacturing the system designed and let the abscissa represent its certain performance like stiffness of the system.

There are two important uses for this curve display. First, the designer upon receiving the current information about the system can visualize the trend of the changes of the system characteristics resulting from few system modifications. He may then select appropriate design parameters much more cleverly and efficiently than the computer. Secondly, the designer can make his final design decision according to the need of a customer's application. For example, he may like to relax the stiffness requirements of the system in order to reduce the manufacturing cost. This choice can be made right on the display screen by pointing at the point of interest on the curve with the light pen and let the computer print out all the design information corresponding to a particular set of parameters.

Finally, before accepting all the design data, it is necessary to check if the design still satisfies the given constraints. Also it may proceed to study the other problem areas of interest in the same system. It is only after all requirements and problem areas are studied that the design is completed.

CHAPTER IV

GENERAL STRUCTURE OF THE LING* SYSTEM

A. Geometry Recognition

In order that the computer may be directed to perform all of the mathematical analysis and computations for the system sketched, it must know exactly what has been sketched and what to do with the drawing. That is, the computer must, first of all, recognize the exact geometry of the picture drawn. It is not sufficient to have topological information such as the number of lines and circular arcs in the picture, the coordinates of points, the starting or end point of a line, and the type of constraints imposed, etc. Some manipulation of these Sketchpad topological data is needed. The new set of data should then be stored in a very convenient way so that all the information on the geometry of the sketched picture can be found at will whenever needed. For example, it is important to know the diameter and the length of a shaft in addition to the location of the loads and supports before the calculation of the shaft deflection can be performed. The basic problem remaining then is the recognition of the geometry of various drawings.

---

*The Ling system is that described in this thesis.

The recognition of a general drawing can be made based on the following three principles:

(1) All points which form the outline of a shape (or drawing) must be collected in the exact <u>sequence</u> of their first appearance. These points must be connected by two pointers, one points to the previous point and the other to the following point. This gives the "skeleton" of a drawing.

(2) There must be information concerning the type of curve that connects the current point to its adjacent points and all the pertinent information about this curve. For examples, if a straight line is linking two points, we may wish to know its length and slope in addition to the coordinates of two end points. If the line is a circle, it is necessary to know the radius, the coordinate of the center of the circle, the angle of the arc, and the direction in which the circular arc is sketched. If the curve is a "free handed" curve, it may be necessary to find the equation that describes this curve. Thus, the numerical indication of the type of curve

between two adjacent points in addition to the set of points in sequence will provide the computer with the "outline" of a picture.

(3) A reference point must be selected from which the "orientation" of the outline of a drawing can be determined. The key dimensions or parameters of the drawing can be calculated most easily by "counting the points". As an example, a point with the minimum x-coordinate is chosen as the reference point. The first diameter of a shaft is then the difference in y-coordinates between this reference point and the last point of the point series. As a matter of fact, since these two points are tied together by pointers and they must be connected by a line, the line length will be exactly equal to the diamter of the shaft.

Consequently, the recognition of geometry is the fundamental base for design by means of the Sketchpad system. The geometry of a picture is recognized first by its "skeleton" and then by its "outline". Finally, through the choice of a reference point, the orientation of the drawing is determined.

B. General Data Structure

The data structure used in the Ling system, when combined with Sketchpad II for design automation, should as a basic requirement be general. That is, the structure itself should have unlimited depth. Its organization and logic should be independent of the computer so that the operations can be carried out to whatever number or depth necessary on any digital computer. Any system that is useful only for certain design problems cannot be considered as a very powerful design tool. The structure developed in this thesis possesses unlimited capability to insert other design programs and it fulfills the basic requirement of generality.

The data structure of this system should contain the following features:

(1) It should have the capability of accepting any sketched system which requires one, or more than one, views to describe. Furthermore, it should provide clear information as to the type of problem on which studies will be made in connection with the sketched picture, the total number of pictures that are required to describe the system, and the particular view of the system the computer is currently looking at. It should

also distinguish the instances from pictures.

(2) It should have the ability to discover the geometry so that key dimensions can be determined easily for inputs to subroutines that may be coupled to analyze the system characteristics mathematically.

(3) It should have the access to sets of design parameters that may be needed for characteristic studies or numerical display of some of those characteristics.

(4) It should have the flexibility to accept modifications to the data structure without requiring any fundamental change in the system.

In general, these features of the data structure have been accomplished in the following way:

(1) In order that the computer may know the type of problem, the total number of views associated, the particular view of the sketched system and to distinguish instances from pictures, the simplest solution is to indicate them in the data structure. Consequently, a picture name (PCNAME) should be included in the generic block which may make use of it. This PCNAME contains the above mentioned basic information about the

picture. For example, a half register of 18 bits can be divided into four fields each of them carrying certain functions.

PCNAME = 00 0 0 00 (octal)

A B C D

Field A may be used for problem type (GPICTP). As an example, 1 can be used to indicate that it is the design of a shaft system, 2 for airplane wing preliminary design and so forth. Field B may be used to indicate the total number of pictures (VEWCNT) associated with this problem. Field C is the indication of the current view (VIEWNO) and finally, field D may be used to identify the instance. (INSTNO)

(2) As indicated previously, in order to recognize the geometry of a "singly connected" drawing, its "skeleton", "outline" and "orientation" will have to be determined. Thus a general data structure should have all the point blocks connected by pointers in the exact sequence of their appearance in a drawing plus the indication of the type of curve that connects two adjacent

points. Furthermore, there should be an access to a block of registers where all of the information concerning this curve can be found. This information, for example, may be the radius of a circle, the coordinates of the center point and the cycle of the circular arc. With this information, a reference point can then be selected to get the "orientation" of the drawing. This point can be one with a minimum x-coordinate or one that is pointed at by the light pen.

For recognition of a drawing geometry other than a "singly connected" drawing, it it possible to consider that the drawing is made up by superposition of many "singly connected" drawings. Since the data structure will recognize each "singly connected" drawing separately therefore, by making use of certain relationships between each "singly connected" drawing, the entire geometry can be recognized.

(3) Since the calculation of a system characteristic may require many parameters in addition to dimensional information, the data structure should provide access to them. A ring can be included in a main generic block of the drawing. This ring leads to several generic blocks, each containing a set of design parameters.

The number of sets of parameters that can be stored should not be limited. One may wish to study the lubrication problem or power transmission problem on a shaft system. Each of these studies may require a separate set of parameters.

(4) The size of each generic block and the relative arrangement of each register as well as symbolic name should not be fixed. It should provide the flexibility to allow the addition of registers in each block for other uses. Also, should there be any need for another type of generic block, the data structure should be ready to accept this expansion.

Specifically, the general data structure shown in Figure 4-1 possesses the above mentioned features.

The highest level generic block is MASTER which ties all the next level generic blocks, PROBLM by a PICTUR ring. In each PROBLM block is a TIES ring that strings all the next level, SHAPES and INSTNS for a particular system sketched. In each view of this sketched system, there is a corresponding SHAPES just as there will be an INSTNS associated with each type of an instance.

Similarly, for each point of a "singly connected" drawing, there will be a SHAPE block. It might be

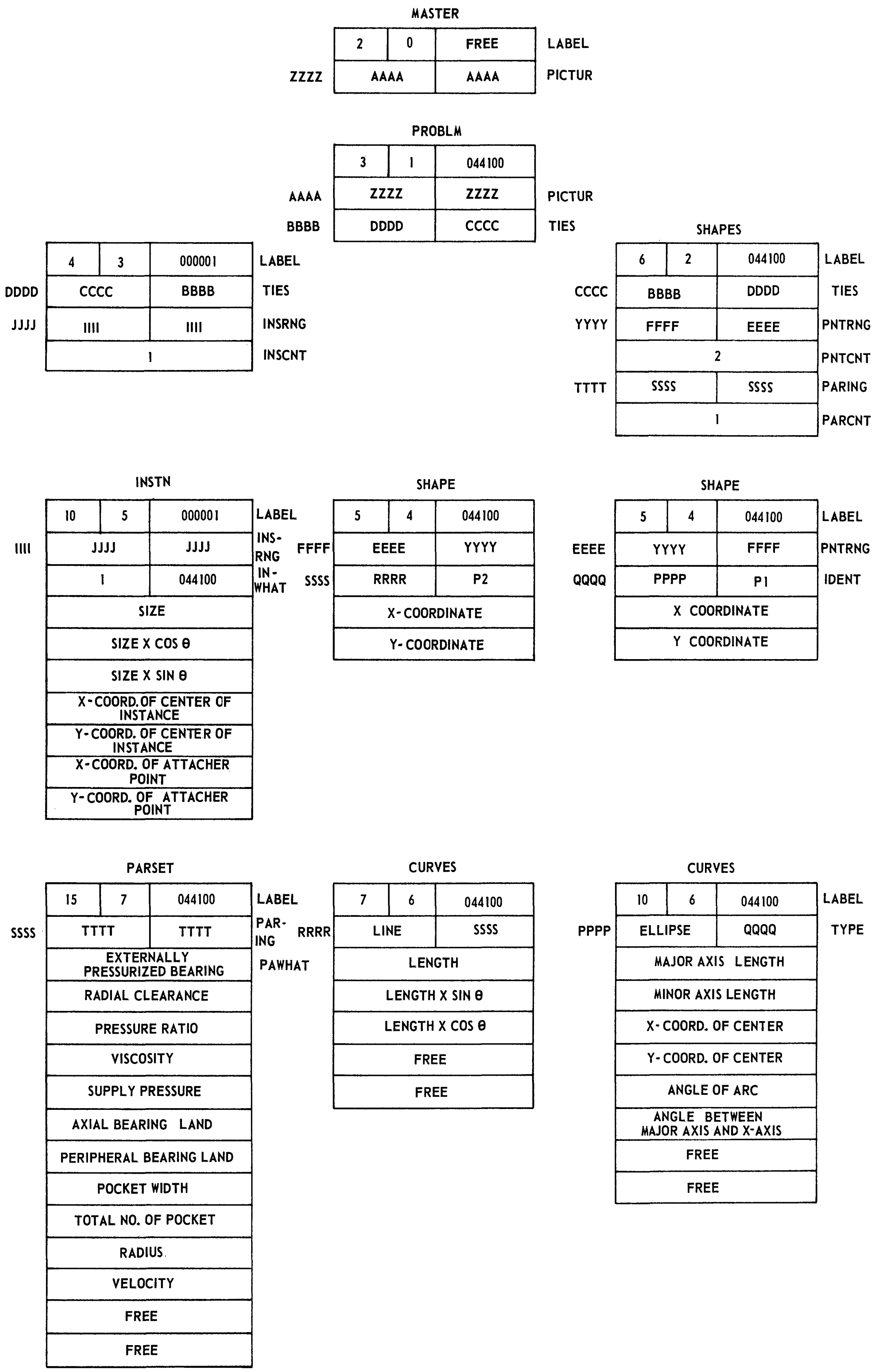


FIGURE 4-1
GENERAL DATA STRUCTURE OF LING'S SYSTEM

more appropriate to call this SHAPE block a POINT block instead. However, since the block belongs to SHAPES in a lower level and the information contained in these blocks constitutes a shape of a drawing, the name SHAPE is implied.

Other generic blocks present are: INSTN which contains all information concerning a particular type of instance, CURVES which contain most of the information on a curve connecting the current point to the next point, PARSET which stores all the parameters for inputs to subroutines performing mathematical analysis in the sketched system, as well as some free registers for storing particular characteristics of the system which the designer may wish to display numerically.

The function of each register in the general data structure is summarized as follows:

LABEL: the first register of each generic block that contains REGCNT (register count), BLKTYP (block type) and PCNAME (picture name) or INSTNO (instance number).

PICTUR: a ring* which ties all the PROBLM of various pictures.

---

*A ring consists of HEAD and TAIL. HEAD points to "where it goes" and TAIL points to "where it came from". Their symbolic names are always given by replacing the last character of the ring name by H or T. For example, the symbolic name of HEAD of PICTUR is PICTUH and that of TAIL is PICTUT. The field sizes of HEAD and TAIL are not fixed.

TIES: a ring which ties all the SHAPES and INSTNS associated with a type of PROBLM.

PNTRNG: a ring which ties all SHAPE blocks.

PNTCNT: a register which contains the total number of points in this picture.

PARING: a ring which ties all PARSET blocks.

PARCNT: a register which contains the total number of sets of parameters.

INSRNG: a ring which ties all INSTN blocks.

INSCNT: a register which contains the count of this type instances present in the <u>entire</u> group of sketchings of a system.

INWHAT: a register which contains VIEWNO (the indication of any particular view) and PCNAME.

IDENT: a register which contains point identification (PNTID) as well as a pointer (PNTER) pointing to TYPE of CURVES.

TYPE: a register which contains a symbol indicating the type of curve (CTYPE) and the pointer (PNTER) points back to IDENT.

PAWHAT: a register which contains a symbol indicating for what kind of problem this set of parameters is.

VNOPCT: a register which contains VIEWNO and PNTCNT.

This data structure appears to be quite complicated. It may prove very convenient to have the condensed data structure such as the one shown in Figure 4-2. PTABLE contains mainly the coordinates of all points in sequence starting with one that has a minimum x-coordinate. Similarly, ITABLE contains mainly the magnitude and the coordinates of the point of application of a type of instance. The first two design examples to be described in Chapter 6 makes very effective use of these condensed data structures.

Figure 4-3 is a dictionary of the general data structure. Any modification of this data structure can easily be made with the aid of this dictionary. The following rules should be followed in reading the dictionary:

(1) Second column, if it is not 777777777776k, then

First column = symbolic name

Second column = generic block which contains the symbolic name (Note, if it is 777777777777k, it indicates "all other generic blocks")

Third column = register number of the generic block (note, if it is 0, it indicates "none existing")

P TABLE

8	8	044100	LABEL
1		2	VNOPCT
LINE		P1	
X-COORDINATE			
Y-COORDINATE			
CIRCLE		P2	
X-COORDINATE			
Y-COORDINATE			

I TABLE

9	8	000001	LABEL
1		044100	INWHAT
		2	INSCNT
MAGNITUDE			
X-COORD. OF ATTACHER			
Y-COORD. OF ATTACHER			
MAGNITUDE			
X-COORD. OF ATTACHER			
Y-COORD. OF ATTACHER			

FIGURE 4-2
CONDENSED DATA STRUCTURE

SYMBOLIC NAME	GENERIC BLOCK	REGISTER	FIELD	
			FIRST	LAST
PCNAME	ITABLE	2	19	36
	INSTN	3	19	36
	MASTER	0	0	0
	INSTNS	0	0	0
	777777777777K	1	19	36
BLKTYP	PTABLE	0	0	0
	ITABLE	0	0	0
	777777777777K	1	10	18
REGCNT	777777777777K	1	1	9
INSTNO	INSTNS	1	19	36
	INSTN	1	19	36
	ITABLE	1	19	36
	777777777777K	0	0	0
VIEWNO	INSTN	3	1	18
	ITABLE	2	1	18
	PTABLE	2	1	18
	777777777777K	0	0	0
PNTCNT	SHAPES	4	1	36
	PTABLE	2	19	36
	777777777777K	0	0	0
INSCNT	INSTNS	4	1	36
	ITABLE	1	10	18
	777777777777K	0	0	0
PARCNT	SHAPES	6	1	36
	777777777777K	0	0	0
PNTER	SHAPE	3	1	18
	CURVES	2	19	36
	777777777777K	0	0	0
PNTID	SHAPE	3	19	36
	777777777777K	0	0	0
CTYPE	CURVES	2	1	18
	777777777777K	0	0	0
PAWHAT	PARSET	3	1	36
	777777777777K	0	0	0
PICTUH	MASTER	2	19	36
	PROBLEM	2	19	36
PICTUT	MASTER	2	1	18
	PROBLEM	2	1	18
PICTUH PICTUT	777777777777K	0	0	0
TIEH	PROBLM	3	19	36
	SHAPES	2	19	36
	INSTNS	2	19	36
TIET	PROBLM	3	1	18
	SHAPES	2	1	18
	INSTNS	2	1	18
TIEH TIET	777777777777K	0	0	0
PNTRNH	SHAPES	3	19	36
	SHAPE	2	19	36
PNTRNT	SHAPES	3	1	18
	SHAPE	2	1	18
PNTRNH PNTRNT	777777777777K	0	0	0
INSRNH	INSTNS	3	19	36
	INSTN	2	19	36
INSRNT	INSTNS	3	1	18
	INSTN	2	1	18
INSRNH INSRNT	777777777777K	0	0	0
PARINH	SHAPES	5	19	36
	PARSET	2	19	36
PARINT	SHAPES	5	1	18
	PARSET	2	1	18
PARINH PARINT	777777777777K	0	0	0
PICRCD	777777777776K	10000	777777777777K	777777777777K
MASTER	777777777776K	331	0	0
PROBLM	777777777776K	336	1	0
SHAPES	777777777776K	341	2	1
INSTNS	777777777776K	341	3	1
SHAPE	777777777776K	346	4	2
INSTN	777777777776K	351	5	3
CURVES	777777777776K	356	6	4
PARSET	777777777776K	346	7	2
PTABLE	777777777776K	20000	8	777777777777K
ITABLE	777777777776K	25000	8	777777777777K

FIGURE 4-3
DICTIONARY OF THE GENERAL DATA STRUCTURE

Fourth column = beginning of the field

Fifth column = end of the field

(2) Second column, if it is 777777777776k, then the

First column = generic block

Second column = 777777777776k

Third column = a pointer points to one level higher generic block in the same dictionary or the location in the dictionary

Fourth column = block type (BLKTYP)

Fifth column = block type of one level higher generic block (note, if it is 777777777777k, it indicates the highest level generic block)

C. <u>Data Manipulation</u>

The writing of any subroutine for mathematical analysis can now be done through data manipulation of the general data structure. The following steps can be followed:

(1) Go around PICTUR ring to find the correct PROBLM. That is, the one with correct GPICTP.

(2) Go around TIES ring to find the desired SHAPES or INSTNS. Check the VIEWNO which

appears in the field C of PCNAME. This is to assure that the correct view of the system is selected.

(3) Go around the PNTRNG to arrive at various SHAPE in sequence. The SHAPE provides the coordinates of the point. Information on a curve connecting this point to the next points is found from CURVES which are located at a location pointed by PNTER of the register IDENT.

Any dimensions are determined by "counting the point".

(4) Similarly, information on instances are found by going around the INSRNG.

(5) Design parameters are found by going around the PARING in SHAPES which leads to PARSET.

Figure 4-4 is an example of how a subroutine can be written in order to go around a ring in Ling's general data structure assuming that the programmer has no knowledge of how each generic block is formed. Figure 4-5 is an example of how an additional generic block is added to the general data structure.

```
           RSUBROUTINE TO GO AROUND A RING
            EXTERNAL FUNCTION
            ENTRY TO RING.
            PROGRAM COMMON DICTRY
            NORMAL MODE IS INTEGER
            DIMENSION DICTRY(1000)
           RFIND THE STARTING LOCATION OF STRING OF REGISTERS PICRCD IN D
           1ICTRY,WHICH CONTAINS DATA ON SKETCHED DRAWING
            LOCATE.($PICRCD$,I,4)
            PICRCD=DICTRY(I+2)
           RFIND LOCATION OF RING IN GENERIC BLOCK,MASTER,AND THE MASK
            LOCATE.($PICTUH$,I,4)
            LOCATE.($MASTER$,I,I+1)
            WHENEVER DICTRY(I+1).E.0,TRANSFER TO ALPHA
            MPICTH=DICTRY(I+1)
            END1H=DICTRY(I+3)
            MASK.(MSKPIH,DICTRY(I+2),END1H)
            LOCATE.($PICTUT$,I,4)
            LOCATE.($MASTER$,I,I+1)
            WHENEVER DICTRY(I+1).E.0,TRANSFER TO ALPHA
            MPICTT=DICTRY(I+1)
            END1T=DICTRY(I+3)
            MASK.(MSKPIT,DICTRY(I+2),END1T)
           RMAKE A COMPLETE TURN FORWARD AROUND THE RING
            HEN=PICRCD+MPICTT-1
            CHICKN=PICRCD+MPICTH-1
LOOP1       THROUGH LOOP1,FOR CHICKN=(DICTRY(CHICKN).A.MSKPIH).RS.(36-END
           11H),(DICTRY(CHICKN).A.MSKPIH).RS.(36-END1H),CHICKN.E.HEN
            TRANSFER TO BETA
ALPHA       PRINT COMMENT $0ERROR,RING NOT IN GENERIC BLOCK$
            ERROR RETURN
BETA        FUNCTION RETURN
            END OF FUNCTION
$COMPILE MAD,PRINT OBJECT
            INTERNAL FUNCTION (NAMEXX,QI,QSTART)
            ENTRY TO LOCATE.
LP1         THROUGH LP1,FOR QI=QSTART,5,DICTRY(QI).E.NAMEXX.OR.DICTRY(QI)
           1.E.777777777777K.OR.QI.G.DICTRY(2)
            WHENEVER QI.G.DICTRY(2)
            PRINT COMMENT $0ERROR TRYING TO FIND DICTIONARY ENTRY$
            PRINT RESULTS NAMEXX,QI,QSTART
            ERROR RETURN
            END OF CONDITIONAL
            FUNCTION RETURN
            END OF FUNCTION
$COMPILE MAD,PRINT OBJECT
            INTERNAL FUNCTION (MSKXXX,BEGIN,END)
            ENTRY TO MASK.
            MSKXXX=0
            THROUGH LP,FOR QQ=BEGIN,1,QQ.G.END
LP          MSKXXX=MSKXXX.V.(1K.LS.(36-QQ))
            FUNCTION RETURN
            END OF FUNCTION
```

Note:

1. PICRCD is a string of registers inside another string of registers DICTRY. All information on sketched drawing contains in PICRCD.
2. DICTRY(0) contains next available location in the entire storage.
3. DICTRY(1) contains length of dictionary (DICTRY) storage.
4. DICTRY(2) contains next available location in DICTRY.
5. DICTRY(3) contains lowest level block type (ØLKTYP).

FIGURE 4-4
SUBROUTINE FOR GOING AROUND A RING

```
        RINSERT A GENERIC BLOCK XXXXXX,ONE LEVEL BELOW YYYYYY,IN DICTIONARY
         EXTERNAL FUNCTION (XXXXXX,YYYYYY)
         ENTRY TO INSERT.
         NORMAL MODE IS INTEGER
         DIMENSION DICTRY(1000)
         J=DICTRY(2)
         DICTRY(2)=DICTRY(2)+5
         DICTRY(J)=XXXXXX
         DICTRY(J+1)=777777777776K
LOOP     THROUGH LOOP,FOR I=4,5,DICTRY(I).E.YYYYYY.OR.I.G.DICTRY(2)-5
         WHENEVER I.G.DICTRY(2)-5,TRANSFER TO ERROR
         DICTRY(J+2)=I
         DICTRY(J+3)=DICTRY(3)
         DICTRY(3)=DICTRY(3)+1
         DICTRY(J+4)=DICTRY(I+3)
         TRANSFER TO ALPHA
ERROR    PRINT COMMENT $0THERE IS NO GENERIC BLOCK IN THE DICTIONARY C
        1ORRESPONDING TO ARGUMENT 2$
         ERROR RETURN
ALPHA    FUNCTION RETURN
         PROGRAM COMMON DICTRY
         END OF FUNCTION
```

Note: Same note in Figure 4-4 applies.

FIGURE 4-5
SUBROUTINE FOR INSERTING NEW GENERIC BLOCK

CHAPTER V

IMPLEMENTATION OF THE LING SYSTEM WITHIN SKETCHPAD II

A. Picture Records

As stated previously in Chapter 4, the first step that should be taken for geometry recognition is to obtain the set of points exactly in the sequence as they form the outline of a drawing. This series of points in sequence can be obtained from Sketchpad II data structure based on the following principle.

Each line has a starting point and an ending point. The starting point (or ending point) of one line must be in the same time either the ending point or the starting point of the other line provided that the drawing is closed and that there are exactly two lines connected to a point. This is true for circular arcs as well as the mixture of lines and circular arcs.

Consequently, in reference to Figure 2-1, we can start with SPECB Ring of PICTURES block, go around the ring until the picture with the correct name (PCNAME) is found. Then, proceed to PPART ring of this PICTURE block since PPART ring ties all the lines, circles and instances shown on the drawing. We shall then go around this PPART ring to find the first LINE or CIRCLE block. The next step is to get to the line starting point ring LSP, (or the circle starting point ring CSP). Note that the definition of LSP (or CSP) and LEP (or CEP) depends on how the designer sketches his lines or circles by the light pen. As is clear from the context, the point from which the light pen moves in sketching is the starting point and the end point will be the point at which the light pen terminates. Consequently, a point can be LSP or LEP to both lines as well as LSP to one line and LEP to the other. The location of the first register of the POINT block of this starting point is found in the address part of the first register of this ring* (note that each ring is composed of two registers as described previously). The x, y, coordinate of this point is found in the last two registers of the POINT block. Since LSP (or CSP) or LEP (or CEP) ring ties all the lines or the circles to the PLS ring of the POINT block, from here we may exchange LSP (or CSP) ring to LEP (or CEP) ring or vice versa every time a ring is traversed one member except that an extra member of the ring should be traversed if POINT block is traversed instead of the LINE or the CIRCLE block. This is done to find the other end point of the line or the circle arc. If we were at the LSP (or CSP), we like to get LEP (or CEP). On the other hand, if we were at the LEP (or CEP), we

---

*A ring in Sketchpad II data structure is different from that in Ling's data structure. See footnote on page 30.

naturally would like to get LSP (or CSP) in order to go around the shape continuously. The type of curve that connects two adjacent points can be recorded at the same time that the LSP (or CSP) or LEP (or CEP) ring is traversed. The above process is illustrated by following numbers 1, 2, 3, 4, 5, 6, 7 in Figure 2-1. The exchange of two rings and the traverse around the ring are continued until we returned to the very first point. We shall thus complete the entire outline of a "singly connected" drawing. The simplified flow diagram of this process is shown in Figure 5-1.

Since the Sketchpad II system is capable of handling only lines and circles, it is expedient to simplify the general data structure. Recalling that the register, IDENT, in SHAPE contains a pointer, PNTER. This pointer points to the location of CURVES where information on the curve connecting the current point and the next point are stored. This pointer in IDENT is now used for a pointer only when a circular arc is connecting two adjacent points; otherwise it will be 777777K to simply indicate that the type of curve is in fact a line. This facilitates programmings.

B. Instances

As mentioned previously, INSTANCE is a subpicture which can be generated again and again on the screen and

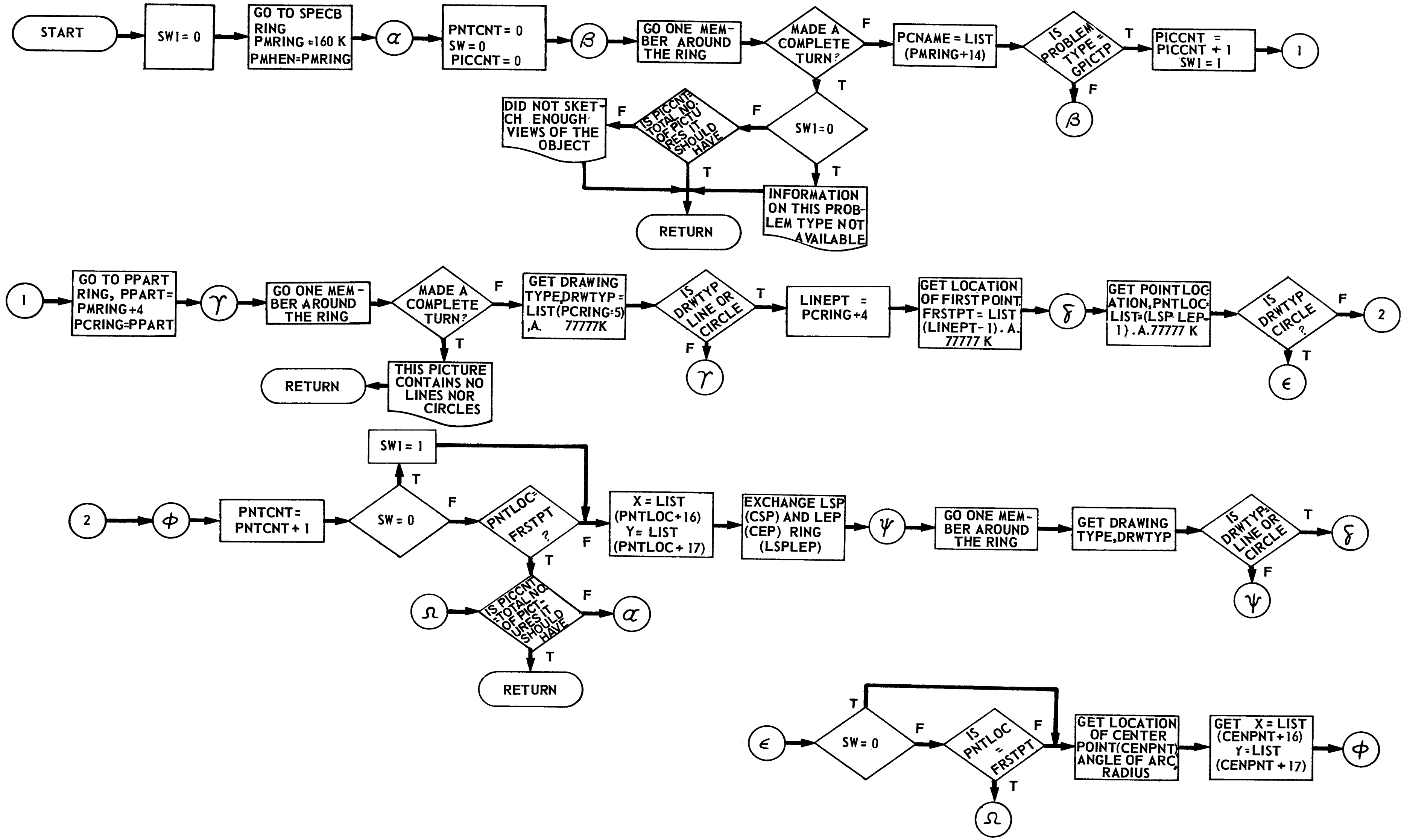


FIGURE 5-1
FLOW DIAGRAM FOR SHAPES

attached to any point of the drawing on command. It is this powerful capability of Sketchpad II that makes it useful in many design problems.

We shall first find the PICTURE block whose PCNAME contains the desired INSTNO again by going around the SPECB ring of the PICTURES block. We shall then get to PINS ring of this PICTURE block. PINS ring ties all the instances of this picture that appeared in all the drawings. The size of the instance, size x Cosθ and size x Sinθ as well as the x, y coordinate of the center of the instance are stored in the last four registers of INSTANCE block.

Since one type of instance may appear in more than one drawing, for example a mechanical system whose front view and side views both contain the same type of instances, it would be necessary to record the view number (VIEWNO) of the drawing so that the coordinates of instance attaching points to be found will be meaningful for each picture.

The coordinates of instance attaching points are found first by going one member around VCON ring of INSTANCE which leads to an IPCON block. Since the VCON ring of the POINT block of the attaching point eventually strings to VAR ring of IPCON, the point location can thus be found at the address portion of the first register of the VAR ring. This process is illustrated by following numbers 1, 2, 10, 11, 12 in

Figure 2-1. This process is summarized in the flow diagram shown in Figure 5-2.

C. Scaling

The TX-2 computer uses 1's complement system* so that all of the numbers are represented within the range of $-1.0 \leq X \leq 1.0$. Consequently, the result of drawing at the 7 in. by 7 in. CRT screen Sketchpad is that all of the coordinate numbers are very small. To obtain more realistic dimensions as well as units, a scaling up of the drawing is necessary.

The scaling is accomplished again by generating a scaling instance in the master picture. In order to obtain the proper relationship between the size of all instances (including instances used for scaling as well as load, support and others) and the current drawing, the following formula is used.

$$P_c = \frac{S_I \times S_{W_D}}{S_M} \qquad (5-1)$$

---

*For conversion of 1's complement fixed point system to floating point system, see Appendix D. The TX-2 computer was developed by the Lincoln Laboratory of M.I.T. All M.I.T. Sketchpads use this computer.

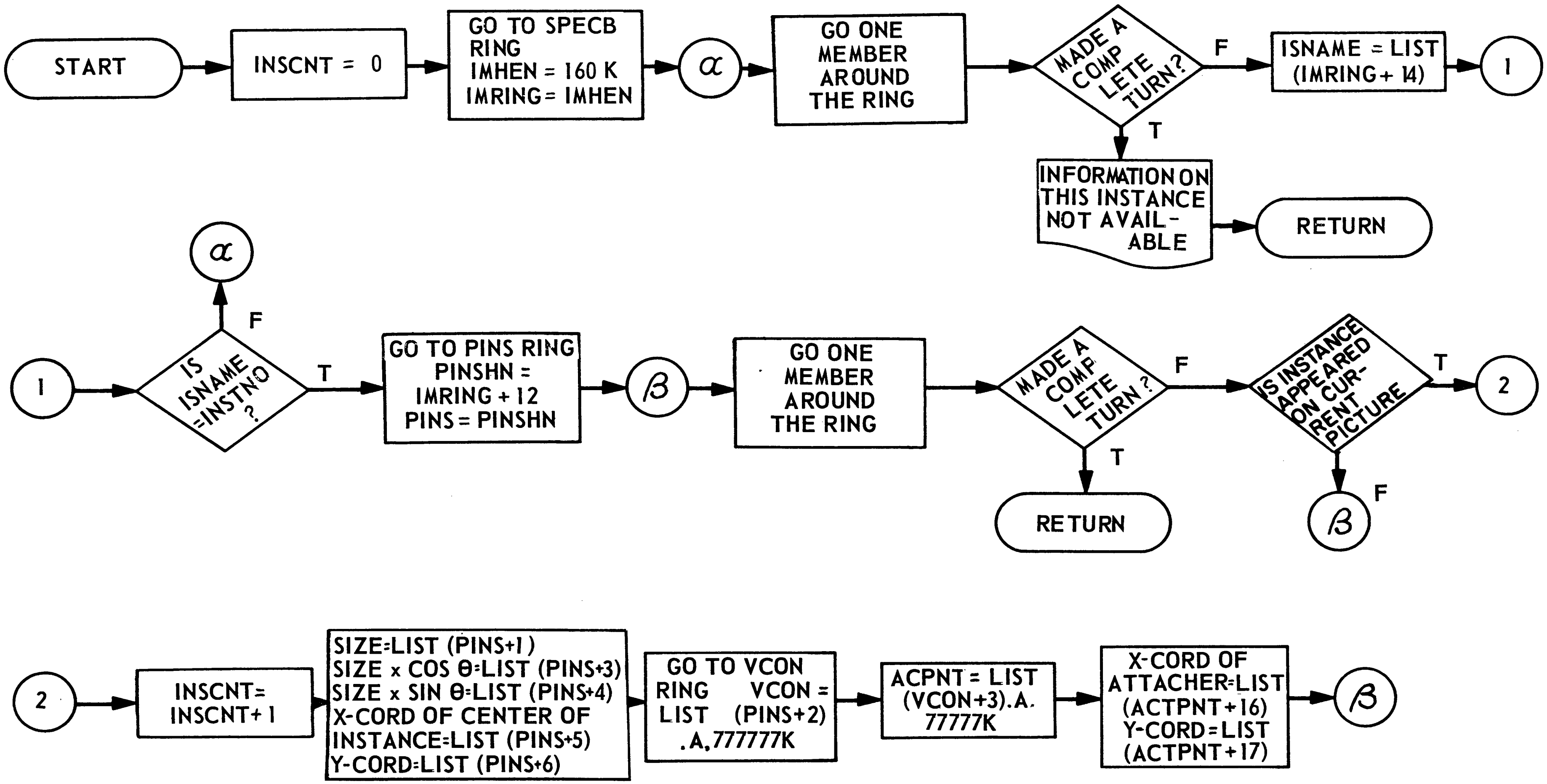


FIGURE 5-2
FLOW DIAGRAM FOR INSTNS

Where

S_M = size of instance master drawing

S_{w_p} = size of current working picture

S_I = size of instance

P_c = page coordinate

Formula (1) maps the instance master drawing into current working drawing.

Thus, the dimension of the distance L in the working drawing is:

$$\frac{L \times S_M}{S_I \times S_{w_p}} \times (\text{scale factor})$$

The scale factor not only gives appropriate dimension but also convenient units.

The current way of scaling a picture is to draw a picture first, and then generate the scaling instance. The designer, by visual judgement selects an approximate scale factor. The scale factor is introduced into one of the registers in the INSTANCE block by a toggle switch. This of course makes the drawing of exact size impossible.

Another difficulty encountered during drawing with the light pen is that the designer cannot draw two views of an object with the correct corresponding dimensions.

This problem can be solved by generating a scaling instance in only the first pictures of a system. The dimension of all other pictures (or views) without scaling instance is matched to the first picture by multiplying another scale factor, α , to all of them. This new scale factor α is found by

$$\alpha = \frac{\Delta_1}{\Delta_2} \qquad (5\text{-}2)$$

Where

Δ_1 = dimension of a length in the picture where scaling instance appears.

Δ_2 = dimension of a length in all other pictures without scaling instance and according to engineering drawing this length should be equal to Δ_1.

Naturally, this way of scaling drawings will cause serious problems if no length Δ_2 can be found in some picture which is supposed to match the length Δ_1 of the first picture. A typical example is an airplane wing which has many different cross sections.

CHAPTER VI

DESIGN EXAMPLES

Three examples of automatic design through Sketchpad II are given in this thesis. Each of these examples requires a different approach of data manipulations. Some applications require only one subroutine for handling of data in PICRCD before proceeding to design calculation while most of the foreseeable design application requires writing only a few subroutines.

A. Shaft System

1. Problem Description

The design of a shaft system is very common in industry yet it may become quite complicated in some cases. A designer may be interested in the stress distribution of the shaft under various loading conditions or he may like to study the dynamic characteristics of the shaft rotating at different speeds. In addition the designer may wish to know the deflection of the shaft receiving several loads and supported by bearings. The deflection problem is selected for consideration in this section.

A designer sketches with a light pen on a CRT screen a shaft with any number of flexible supports. He then calls for instances previously sketched to represent load and support. The designer of this shaft and bearing system may request that any of several constraints be applied to his design. The constraint applied in this example is that the deflection of the shaft at any point in any direction should not exceed some specified value. Next, the program should let the constraints be observed on the CRT screen and appropriate modifications of the design can be done with the light pen. For example, he can use the light pen to move the location of or to change the number of supports. He may also change the size of the shaft or supporting vectors. In other words, the entire design is done initially by a rough sketch at Sketchpad just as artists designing an object first by sketching it on a scratch paper.

Now that the shaft system sketched satisfied the deflection constraint, the designing of the bearings is necessary. The bearings must satisfy the deflection constraint of the shaft and probably also other constraints on the bearing itself such as allowable speed, temperature rise and lubricant flow requirements, etc. The design of liquid-lubricated externally pressurized bearings is here considered because of the large number of parameters available for selection. For example, this bearing type has characteristics dependent upon the number of pockets, pocket-to-supply pressure ratio, bearing land size, presence or absence of axial grooves and clearance. It is only when all the constraints are satisfied that the design of the entire shaft system is completed.

2. Approaches to the Solution of the Problem

Before any calculation on the specific characteristics of the shaft system can be done, it is necessary to know the dimensions of the shaft, points of application of loads and supports. Although SHAPES give all the points in sequence, some means of knowing the arrangement of those points as they form the shape is necessary in order to determine diameters and section lengths of the shaft. This can be accomplished most easily by use of condensed data PTABLE where only point name, curve type and its coordinates are stored in the same sequence starting with the point with the minimum x-coordinate.

Figure 6-1(A) shows a shaft with key points labelled A, B, C, D and E. The reason for the existence of point A is for the convenience of sketching symmetrical shaft by the light pen. Of course, it is possible to draw the shaft without dividing this edge line into two lines such as shown in Figure 6-1(B).

Figure 6-1(C) shows all the possible arrangements of the points in PTABLE for the shaft of Figure 6-1(A). Similarly Figure 6-1(D) is for the shaft of Figure 6-1(B).

As is clear from Figure 6-1(C), there are three different cases of calculating shaft diameters and section lengths and two cases in Figure 6-1(D). For example, the diameter and length of each section for Case I can be found easily with the following five MAD instructions:

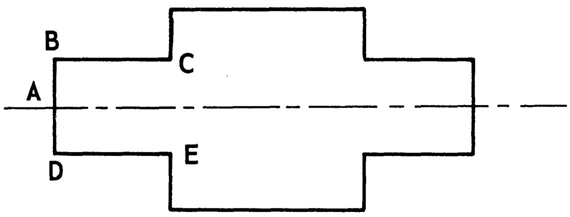
```
     J = (N-2)/2
     THROUGH LOOP1,FOR I=1,1,I.G.J
LOOP1     D(I)=.ABS.(PTABLE((I+1)*3+2)-PTABLE((N-I+1)*3+2))
          THROUGH LOOP2,FOR I=1,1,I.G.(J-1)
LOOP2     LL(I)=.ABS.(PTABLE((I+2)*3+1)-PTABLE((I+1)*3+1))
```

Where N= total number of points

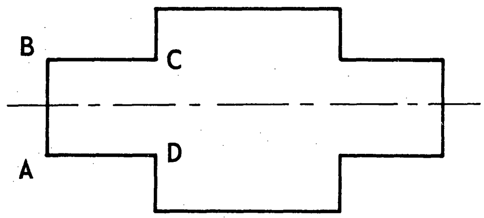
J= total sections

and PTABLE (I*3+2) contains y-coordinate of a point

PTABLE (I*3+1) contains x-coordinate of a point



(A)



(B)

POINT \ CASE	I		II		III	
A	1	1	N	N	2	2
B	2	N	N-1	1	3	1
C	3	N-1	N-2	2	4	N
D	N	2	1	N-1	1	3
E	N-1	3	2	N-2	N	4

(C)

POINT \ CASE	IV		V	
A	1	2	1	N
B	2	1	N	1
C	3	N	N - 1	2
D	N	3	2	N - 1

(D)

N=TOTAL NUMBER OF POINTS OR THE LAST POINT IN TABLE

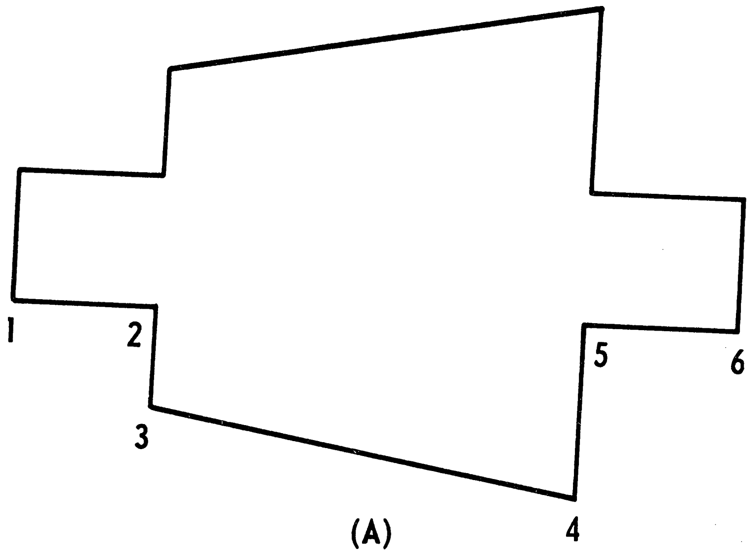
FIGURE 6-1
DIAMETER AND SECTION LENGTH DETERMINATION OF A SHAFT

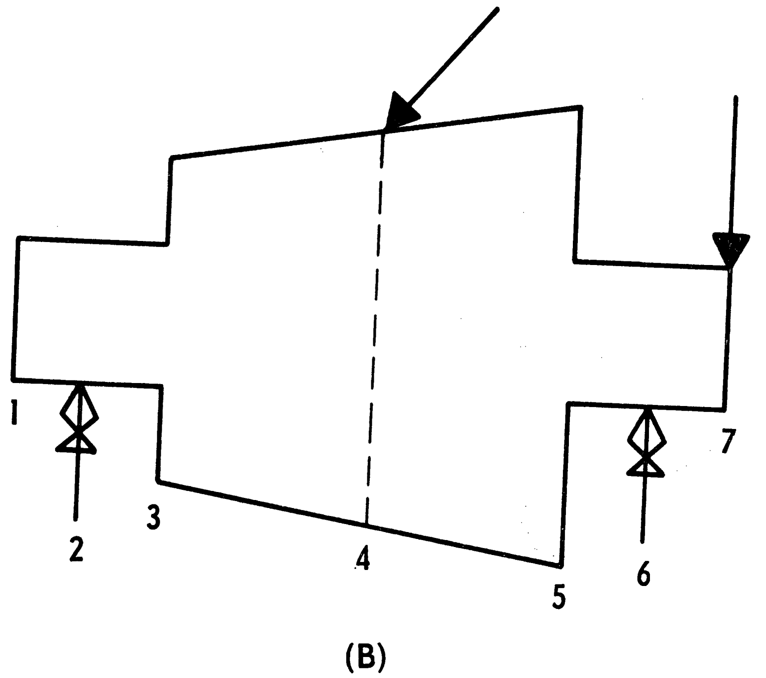
This process for all cases is carried out by subroutine SHAFT.*

The calculation of the shaft deflection requires the input data of diameter, length, load and support at each section. Referring to Figure 6-2(A), the section number according to the previous calculation is shown. Here the lengths of some sections are zero. For example, LL(2) and LL(4) are zero due to steps. These zero lengths will have to be removed and sections renumbered. At the same time, new sections may be introduced due to the application of loads and supports such as section nos. 2, 4 and 6 in Figure 6-2(B). This causes recalculation of some section lengths affected. For a tapered shaft, calculation of a new diameter is necessary, such as D(4). In order to complete the input data required for shaft deflection calculation, some zero magnitude load, support or both will have to be inserted. For instance, section no. 1 should have both zero load and support, section no. 2 should have zero load and section no. 4 should have zero support. This is accomplished by subroutine COMBIN.

---

*For descriptions of all subroutines, see Appendix A.





1,2,3, ... = SECTION NUMBER

FIGURE 6-2
PREPARATION OF INPUT DATA TO SHAFT DEFLECTION CALCULATION ROUTINE

Each section up to this point may be of considerable length. In order that a fairly continuous deflection curve may be displayed at the CRT screen, the displayed spots should be fairly close to each other.* Two approaches can be used here. First, a curve fitting technique to produce many intermediate points can be used. Second, many new sections with zero load and support can be inserted. The new section should have section length equal or less than the specified value small enough to show the continuous curve. The author used the latter approach. This is done by subroutine DISPLY.

Figure 6-3(A), through (E) are the photographs taken from the TX-2 display scope. They are arranged in the sequence of design. Picture (A) is the initial sketch. Notice that shaft deflection exceeded the given specification, the location of support no. 2 is adjusted by light pen.

---

*TX-2 display scope uses point display system. A considerable number of points is required to display a smooth curve. The vector display scope hardware is now being developed at M.I.T. and by some computer manufacturers.

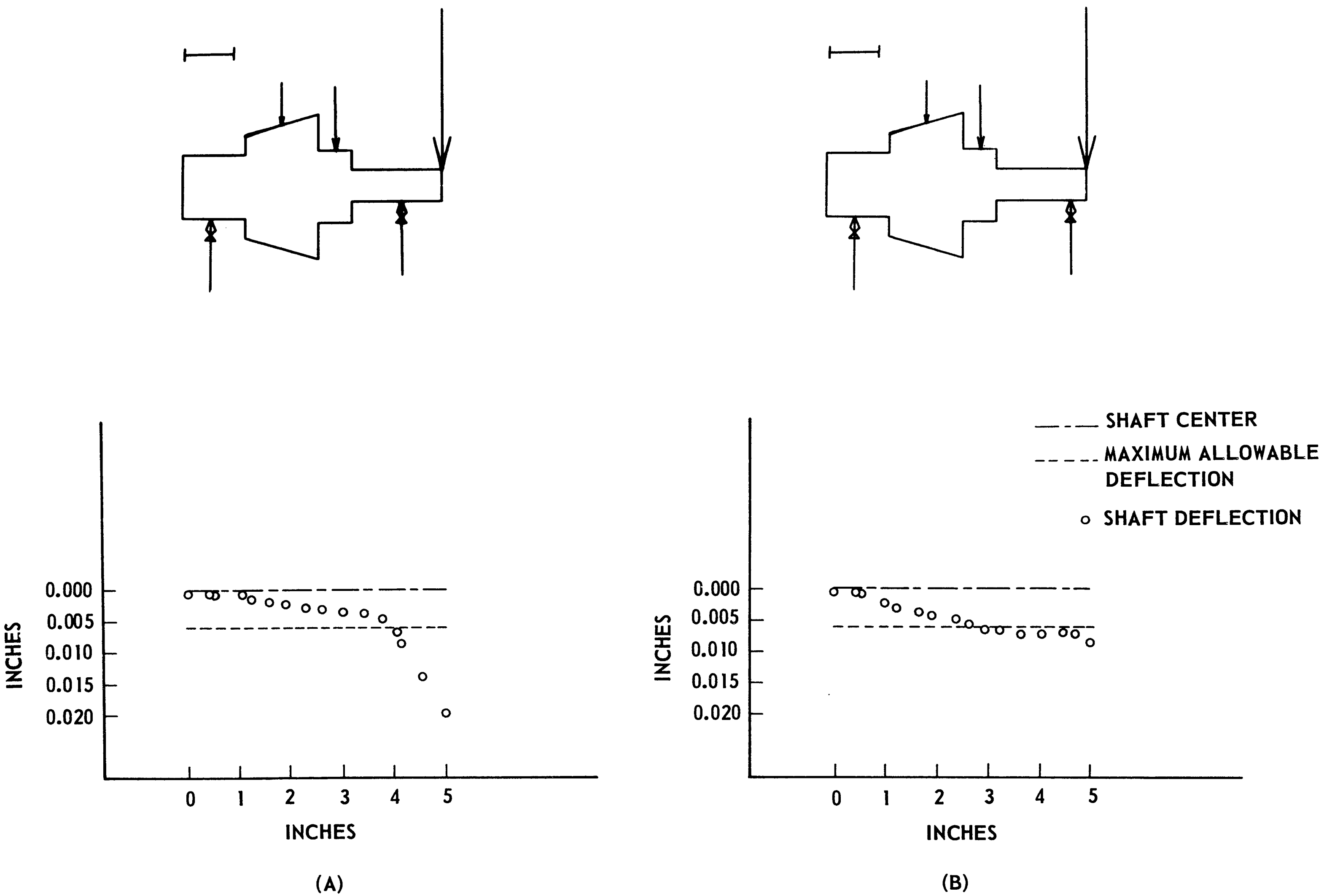
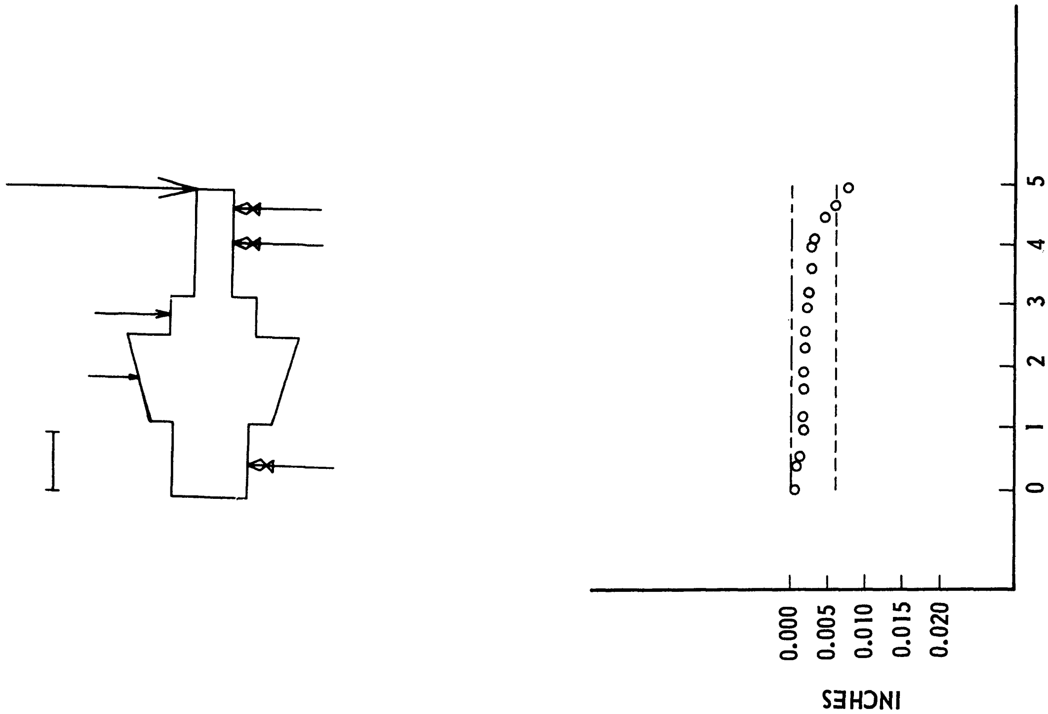
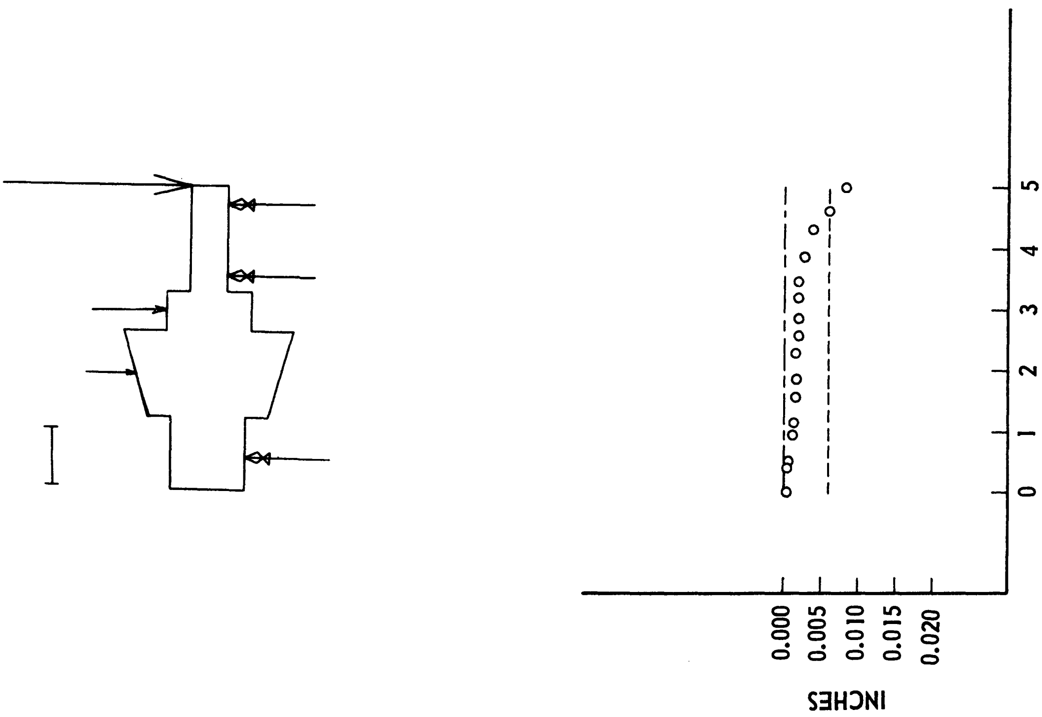


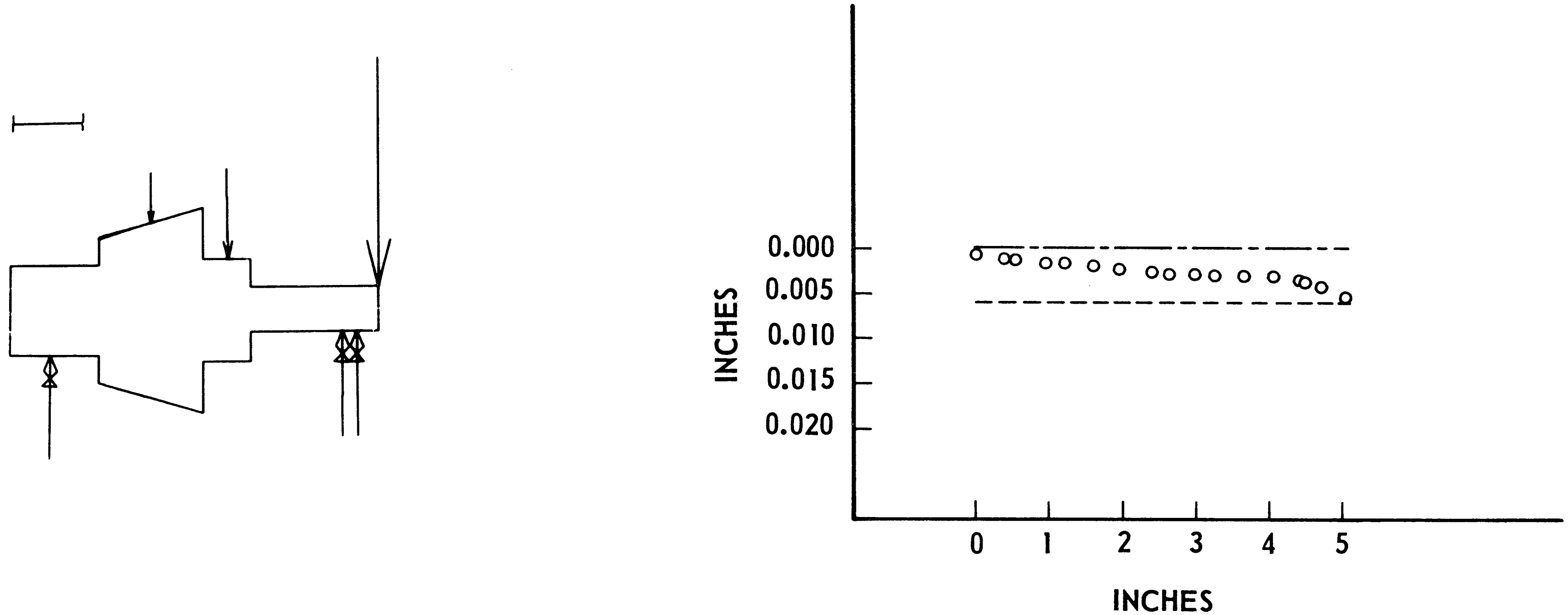
FIGURE 6-3
PHOTOGRAPHS OF SKETCHED SHAFT SYSTEM AND THE PLOTTED SHAFT DEFLECTION CURVES


0.000
0.005
0.010
0.015
0.020
INCHES
0
1
2
3
4
5
INCHES

(D)


0.000
0.005
0.010
0.015
0.020
INCHES
0
1
2
3
4
5
INCHES

(C)



(E)

Several possible adjustments of the location of support, failed to satisfy the deflection constraint. A third support was added. As is clear, the shaft deflection curve now falls within the limits. All the shaft deflection curves were plotted by hand here since the TX-2 feedback was not available to the author.

The next operation is the design of the bearings. The bearings should not only provide enough stiffness to maintain the shaft deflection within limit, but also it should satisfy other constraints that may be imposed. These constraints may include the capability of rotating the shaft at high speeds, lower temperature rise and lesser lubricant flow. The description of the design of externally pressurized journal bearings is given in Appendix C. The computer output from subroutine BEARING is shown in Figure 6-4. The bearing stiffness is 1,100,000 pounds per inch while the minimum stiffness required was 930,000 pounds per inch.

B. Aircraft Wings

1. Description of Aircraft Wing Preliminary Design*

*This section is based largely on a letter dated July 1, 1964 from Mr. Richard Q. Boyles, specialist, preliminary design, Lockheed-Georgia Company, Marietta, Georgia.

```
IZ= 2 TOTAL NUMBER OF POCKETS= 4 RADIUS=    .6610 POCKET LENGTH=    .5000 CLEARANCE=   .000500 AXIAL BEARING LAND=    .1875


 PERIPHERAL BEARING LAND=    .1875 SPEED=        .00 VISCOSITY=   .00000530 SUPPLY PRESSURE=  1000.00 PRESSURE RATIO=   .60
ATTITUDE ANGLE INCREMENTED BY     .2000
AP=      .000000  APP=     .200000  ECCENTRICITY RATIO=   .00  HORIZONTAL FORCE=      .00000  PRESSURE( 1)=     .600000
AP=      .000000  APP=     .200000  ECCENTRICITY RATIO=   .00  HORIZONTAL FORCE=   327.22479  PRESSURE( 2)=     .600000
AP=      .000000  APP=     .200000  ECCENTRICITY RATIO=   .00  HORIZONTAL FORCE=   327.22240  PRESSURE( 3)=     .600000
AP=      .000000  APP=     .200000  ECCENTRICITY RATIO=   .00  HORIZONTAL FORCE=     -.00240  PRESSURE( 4)=     .600000
NT= 4  ECC. RATIO=   .00  STIFFNESS=  -1006199.84  LOAD=       .00238  FLOW=     .052241  CAPPILARY CONST.=     .000033
```

FIGURE 6-4
COMPUTER OUTPUTS OF BEARING DESIGN DATA

The final design of an aircraft wing is composed of a very large number of highly-detailed drawings by which each of the multitude of components is machined, fabricated, and assembled into the final complete structural assembly. To demonstrate the use of Sketchpad in aircraft wing design, our interest is only in basic parameters which influence the general configuration of an aircraft wing. The detail design problem is left for future development.

Basic wing geometric parameters can be listed as follows:

Area ---------------- s

Span----------------- b

Aspect Ratio---------- $A_R = b^2/s$

Root Chord----------- C_R

Tip Chord------------ C_T

Taper Ratio----------- λ

Sweep Angle----------- Λ

Thickness Ratio------- t/c

Airfoil Section

Incidence Angle

Dihedral Angle

Referring to Figure 6-5, the definitions of these parameters are:

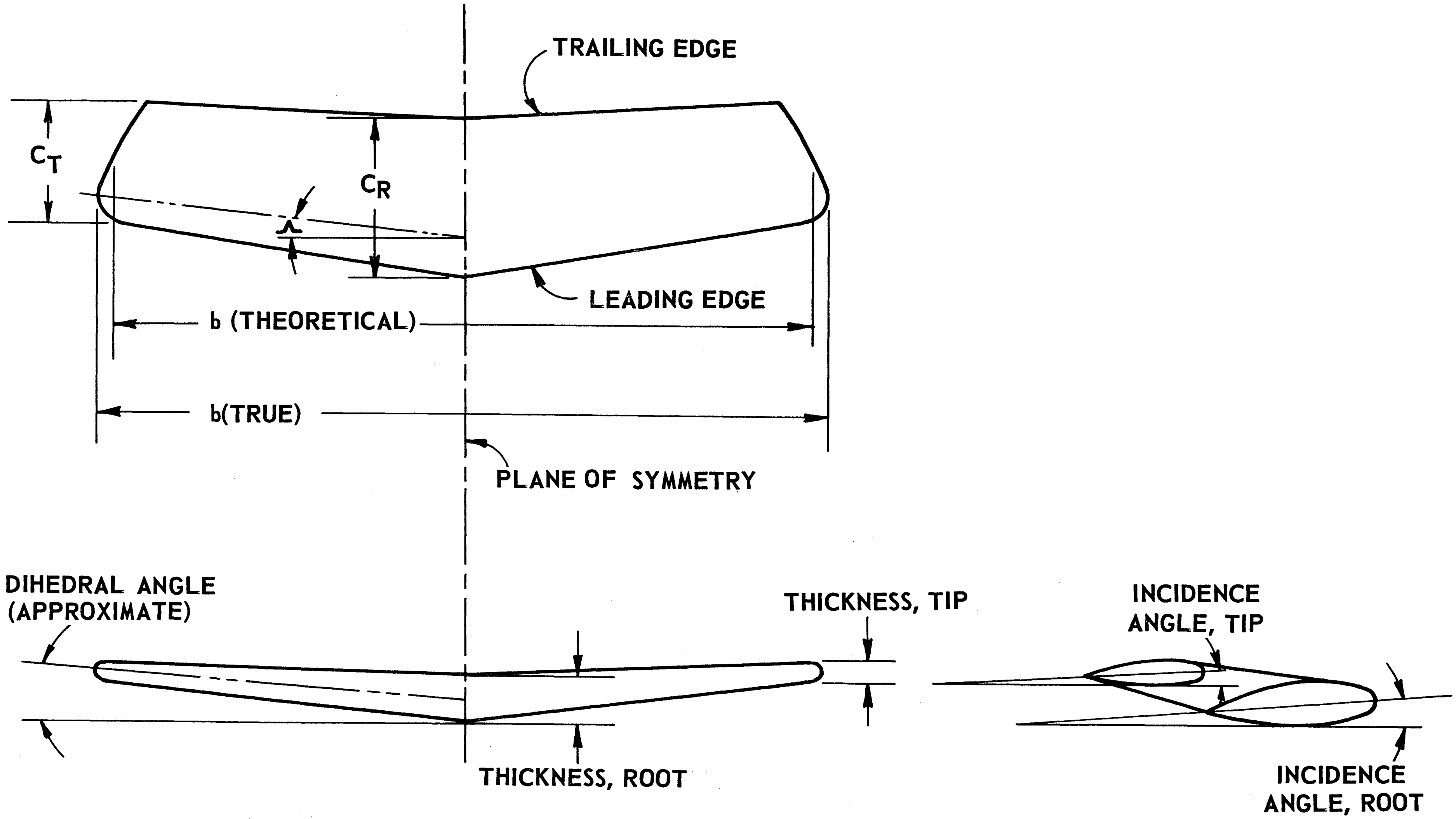


FIGURE 6-5
DEFINITION OF AIRCRAFT WING PRELIMINARY DESIGN PARAMETERS

Wing Area, S, is, as the name implies, the projected area of the planform of the wing, usually quoted in square feet.

Wing Span, b, is the linear distance from wing tip to wing tip (can be from theoretical to theoretical or actual to actual tip locations, depending upon conditions specified).

Wing Aspect Ratio represents an effective wing slenderness ratio and has a major influence upon aerodynamic, structural weight, and aeroelastic characteristics. Aspect Ratio is established by the expression, $A_R = \frac{b^2}{s}$.

Wing Chord is the linear dimension from the leading edge to trailing edge of the wing normally measured parallel to the wing plane of symmetry. In the plane of symmetry the chord is referred to as the root chord. At the wing tip the chord is referred to as the tip chord. The ratio of the tip chord length to the root chord length is termed the wing "Taper Ratio". Wing Taper Ratio has a strong effect upon the spanwise lift distribution across the wing, as well as the wing structural weight.

Wing Sweep Angle refers to the angle between a normal to the plane of symmetry and a reference line on the wing. Sweep is usually referenced to the line passing through the 1/4 chord point of each wing station, but in some instances refers to the wing leading edge.

The Thickness Ratio of the wing defines the ratio of the maximum linear distance between the wing top surface and bottom surface at any given section of the chord length of that section. This parameter has a major influence upon wing drag, especially as sonic velocity is approached, and upon wing weight.

The Airfoil Section is the shape of the cross section of the wing which is derived to create a pressure field which will integrate to produce a resultant vertical force on the wing. A multitude of airfoil sections has been derived to satisfy many varied requirements, and the selection of an airfoil section will be dependent upon the performance requirements of the aircraft.

Incidence Angle is the angle between the airfoil reference line (or reference plane for a complete wing) and some basic aircraft reference line, normally the fuselage reference line. This fixed geometric angle is built into the aircraft in order to position the aircraft parasitic items in a minimum drag configuration.

Dihedral Angle is the geometric angle between the reference planes of the wing panels and a normal to the aircraft plane of symmetry, as viewed along the line of intersection of the wing reference planes and the plane of symmetry.

It is apparent that even a simple layout of an aircraft wing is preceded by the establishment of a significant number of independently variable parameters. The initial values of these parameters are selected by the designer as experience dictates for a known set of aircraft performance requirements. Subsequent detailed analyses normally require several iterations before even a preliminary wing configuration can be selected. It will be shown that these iterations can be most conveniently done through the Sketchpad system.

2. Design of Aircraft Wing by the Sketchpad System

Since Sketchpad II is capable of drawing lines and circles, some simplification is needed to represent the empirical curves (or high order mathematical curves) used in airfoil section and wing tips. For the time being, it will be assumed that these curves are parabolas. In doing so, a very nice representation of a parabola by two straight lines can be made. This representation is shown in Figure 6-6. Suppose points P_1 and P_2 are the initial and the end points of a parabola. The two tangent lines at these points to the parabola intersect at Point P_4. If P_5 is the middle point of line connecting points P_1 and P_2 then the middle point, P_3 of the line connecting point P_4 and P_5 lies exactly on the parabola. Consequently, it is possible to represent this parabola by lines $\overline{P_1P_3}$ and $\overline{P_3P_2}$. That is, given $\overline{P_1P_3}$ and $\overline{P_3P_2}$, the equation of the parabola can be determined.

A minimum of four pictures are required to represent a simple aircraft wing, top view, front view, wing cross section at the root and that at the wing tip. (It is apparent that many wing cross sections are needed for more precise description of the wing. Only two sections are used here for simplicity). In order to find all the parameters listed in previous sections, it is most convenient to use data from PTABLE. For example, Span b is found, according to Figure 6-7 by

$$b = X_{P_5} - X_{P_1}$$

and Root Chord is

$$C_R = Y_{P_7} - Y_{P_3} .$$

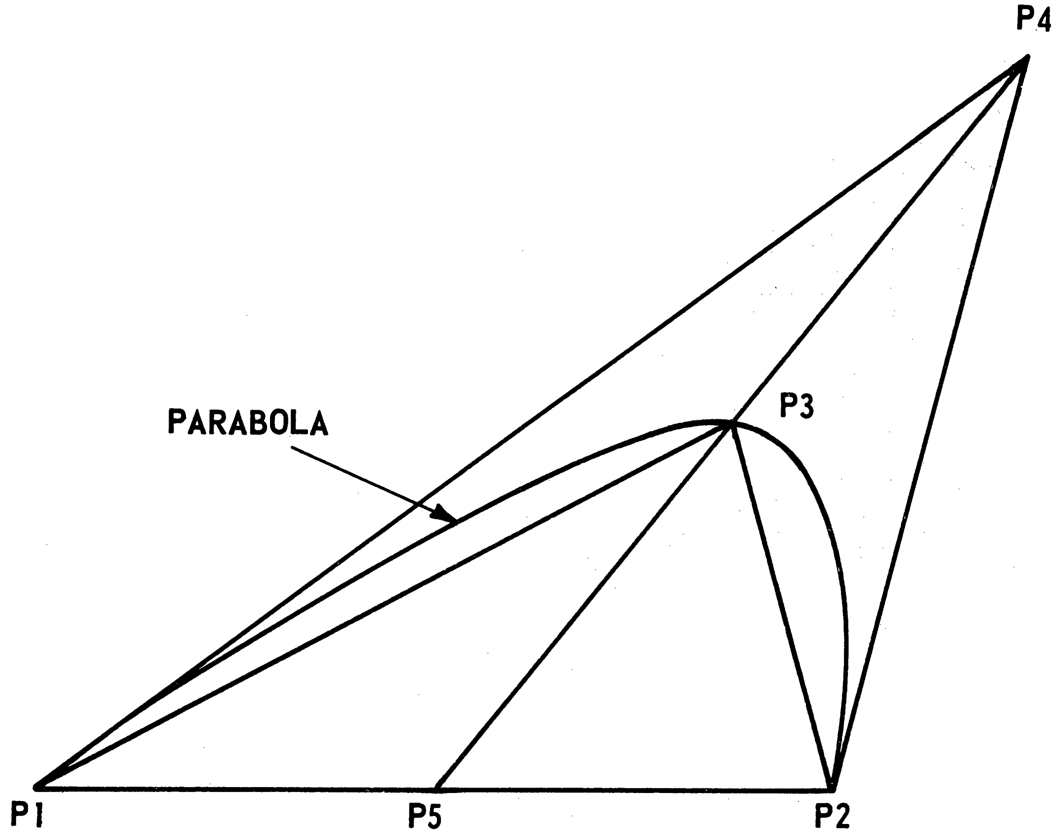


FIGURE 6-6
REPRESENTATION OF A PARABOLA BY TWO LINES

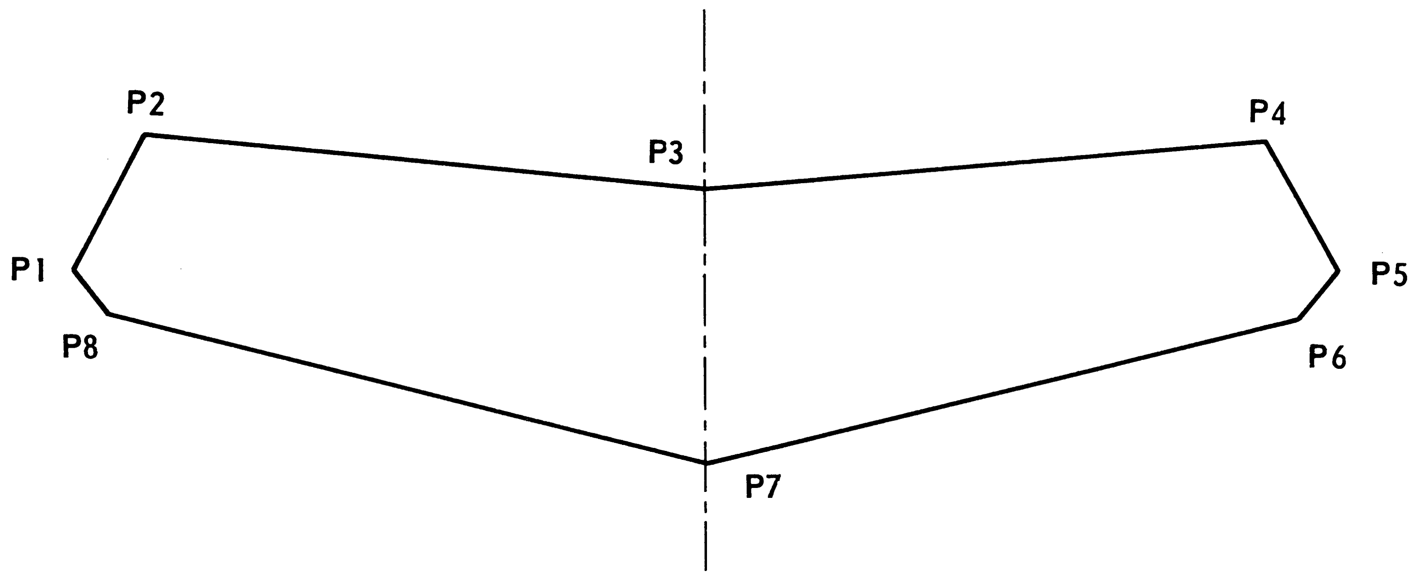


FIGURE 5-7
POINTS ARRANGEMENT FOR TOP VIEW OF AN
AIRCRAFT WING

Although all the points are stored in PTABLE starting with point P_1 which has the minimum X-coordinate, it is possible that these points may be arranged in a clockwise as well as in a counter-clockwise direction.

To avoid confusion, data in PTABLE are rearranged, if necessary, so that the Y-coordinate of P_3 is always greater than that of P_7. The subroutine WINGS will calculate all the required parameters for aircraft performance calculation. The adjustment of these parameters is done again by using the light pen. The calculation of aircraft performance is not included here since this does not add too much significance to this demonstration of use of Sketchpad in aircraft wing preliminary design. Figure 6-8 gives four views of the aircraft wing sketched in TX-2 CRT screen. The design parameters based on these drawings are shown in Figure 6-9 as the results of subroutine WINGS.

C. Numerical Control

1. Brief Description of Numerical Control

Numerical Control is an automatic machine-tool control technique which is applied to many manufacturing processes previously limited to manual

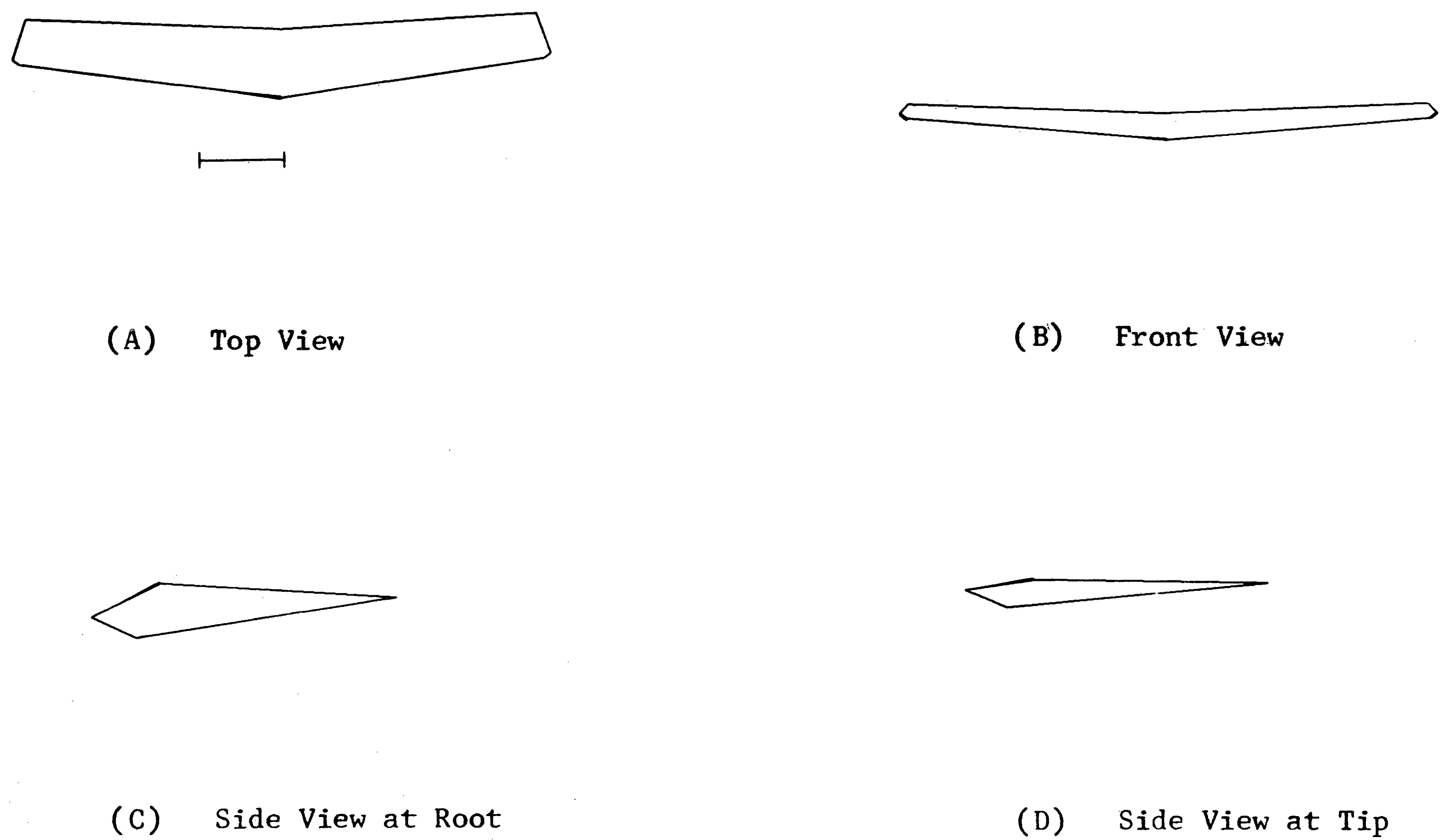


FIGURE 6-8
PHOTOGRAPH OF SKETCHED FOUR VIEWS OF
AN AIRCRAFT WING

AREA= 367.196781	SPAN= 57.908680	ASPECT RATIO= 9.132474
ROOT CHORD= 7.763092	TIP CHORD= 5.300224	TAPER RATIO= .682747
SWEEP ANGLE= .098267	THICKNESS= 2.993762	THICKNESS RATIO= .385640
DIHEDRAL ANGLE= .051652	ROOT INCIDENCE ANGLE= .078687	TIP INCIDENCE ANGLE= .023615

FIGURE 6-9
AIRCRAFT WINGS PRELIMINARY DESIGN
PARAMETERS RESULTED FROM FIGURE 6-8

production methods. In the past, various types of production tools have been designed which can automatically execute complete machining operations under the control of built-in devices. These devices, such as cams, masters, and templates, etc. are time consuming and costly to fabricate. The manufacturing cost of parts requiring such devices is extremely high, particularly if it is a short-run production.

The application of numerical control in machine tools means that the tools are guided over the work piece in response to a series of instructions previously recorded in a numerical code on such media as cards, paper tape, magnetic tape, etc. These series of instruction include coordinates of a locus which the tool should follow in order to machine the desired shape and many other miscellaneous instructions such as feed rate, speed, coolant on-off, etc.

These numerical codes or series of instructions are automatically produced by a digital computer through special computer languages designed for this purpose. There are many numerical control languages in use by many industries. The most widely used language is that of APT, (Automatically Programmed Tool System) originally developed at

Massachusetts Institute of Technology. Besides APT, there are languages such as Autoprompt, CINAP, UMAN, etc.

A so-called part programmer uses one of these languages to describe not only the shape of the part he wishes to machine but also how the part is to be machined. The programmer can specify the accuracy of the part and the machining sequence in which the part is to be made. The computer could interpret these instructions written by a part programmer and translate them into a series of numerical codes suitable for use in machine tool control system.

2. CINAP I System

CINAP I System* (Cincinnati Numerical Automatic Program) developed by the Cincinnati Milling Machine Company, Cincinnati, Ohio is a computer program for the users of the Cincinnati Automatic Thousand Series Numerical Control Systems.

---

*The CINAP I System is selected here mainly because of the connection the author has had with the Cincinnati Milling Machine Company, Cincinnati 9, Ohio. Other languages may just as well be used without fundamental change in this program.

The program consists of several integrated computer routines capable of handling two dimensional parts, described by lines and circles, adaptable to numerical control.* CINAP I provides a mnemonic language whereby the programmer can translate the blueprint information into instructions for controlling the cutter movements and auxiliary functions.

A three field input is used by CINAP in order to generate all the information necessary to control the machine tool. Each line, circle, and point is identified in the card type field. They are then defined by the terms in definition fields 1 and 2.

For example, if a line were defined by two points, the line would be identified in the card type field and the points would be placed in the definition fields. The information in a particular field can be numeric as well as alphabetic. The sequence number of the input is placed in the sequence field to insure that the contour is calculated in the correct order.

The geometric definition of points, lines and circles as well as auxiliary information on feed rate, cutter offset, spindle, coolant and cutter compensation are summarized in CINAP cards which are shown in Figure 6-10. These cards are for the guide to CINAP part Programmers.

---

*A new version of CINAP System is now in use. It is capable of handling many two dimensional parts including those described by CONICS. Since Sketchpad II does not handle CONICS, the old version of the CINAP System is used.

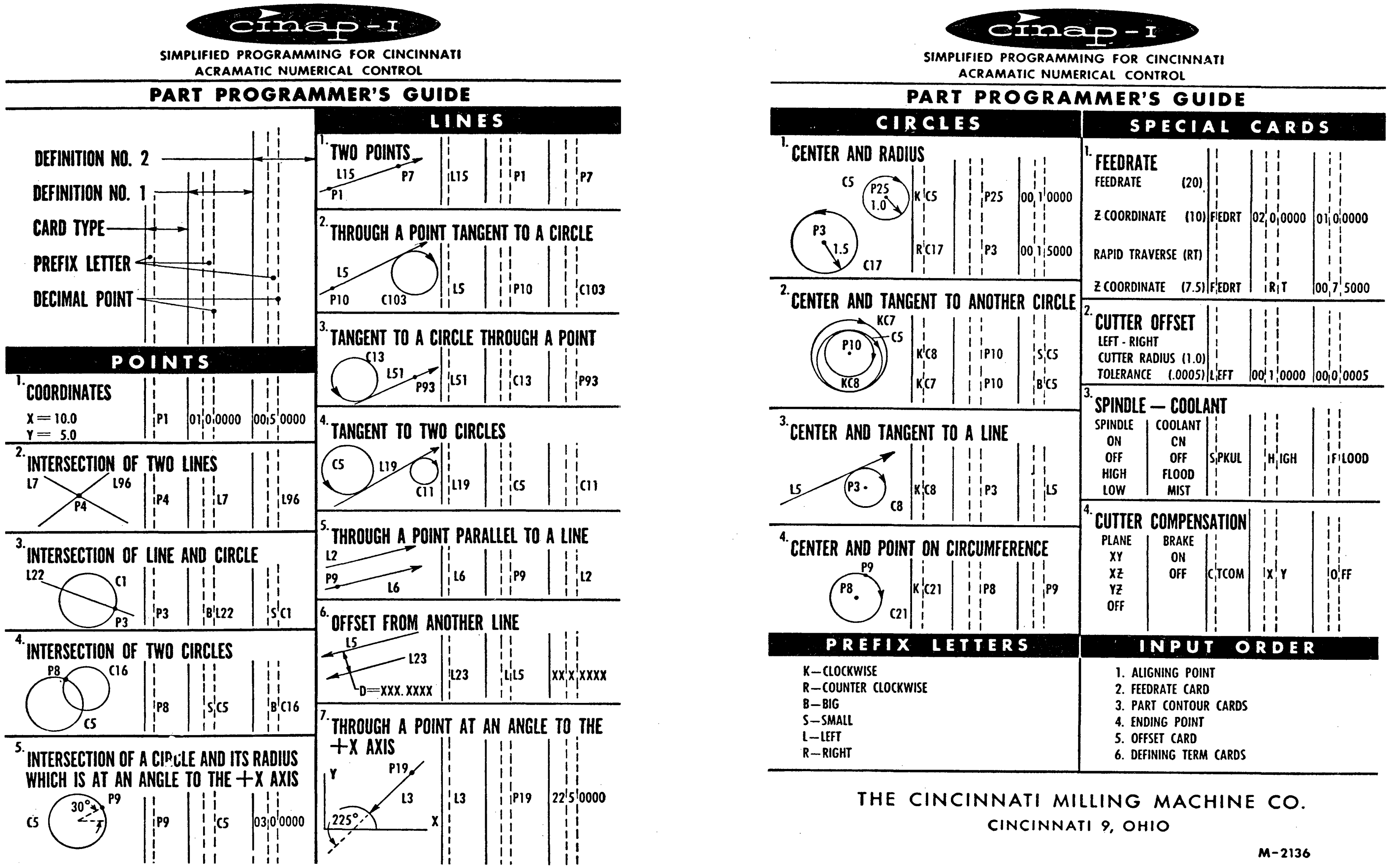


FIGURE 6-10
CINAP CARD

3. New Concept in Numerical Control

Perhaps one of the most exciting applications of Sketchpad in automatic design in the future is the capability of producing numerical control tapes ready for manufacturing immediately after the desired shape of an object is sketched at the CRT screen. It is predicted that the design of an entire automobile style can be accomplished at the screen in the future so that small models may be made from the numerical control tapes obtained according to the style drawn at the Sketchpad. Since the entire data of the automobile body style will be stored, a real scale numerical control tape can be reproduced on command if the proposed style is accepted by the management (after the management has seen the small models of course).

This most challenging goal appeared to have passed the dreaming stage and its realization seems to be only the question of time. In fact, the two dimensional numerical control tape can now be obtained through this program immediately after the object is sketched at the CRT screen. The need for a part programmer could eventually be eliminated. At present the function of a part programmer is not simply to programming but to program from the point of view of better machinability. However, inclusion of the consideration of machinability characteristics into this program is not impossible in the future.

The data structure of SHAPE provides a very valuable help to the development of this program. As mentioned, all the points that form the outline of an object are stored in SHAPE blocks, each of them is tied together in sequence by PNTRNG, ring. Moreover, the type of curve (line or circle) that connects two adjacent points is clearly stored. If a circle connects two points, the information on this circle such as center point coordinates, radius, angle of arc can readily be found by following the pointer (PNTER) to CURVES. If it is a line, 777777k replaces the pointer. This data structure thus gives all the information necessary to describe the outline of an object. Just as a blueprint is provided for a part programmer, this data structure provides all the important information about the geometry before the subroutines which will proceed to produce desired numerical control tape.

There are two approaches that can be used to produce numerical control tapes directly from sketching of pictures at the CRT screen. First, an arithmetic element similar to what is used in the APT, Autoprompt, CINAP and many other systems can be built in to give the coordinates of the locus of the center of the cutter directly. Second, by certain data manipulation, instructions in numerical control programming languages are produced just as they would have been written by a part programmer. The proper arithmetic routines can then be called to translate those instructions.

The author favored the second approach for the following reasons:

1. If an arithmetic element is to be built in, considerable duplication of work that has been carried out for many years by the APT personnel and other mathematicians and engineers in numerical control fields cannot be avoided. Even if these works had been provided to this author, considerable changes may be necessary for use by this program.
2. Many users of numerical control machines may prefer the use of a certain numerical control language over others due to some reasons, the use of the second approach will give them this choice. For each language, it requires the writing of only one sub-routine for that particular language.

It will be demonstrated here that a set of CINAP instructions is produced in sequence according to the numerical control manufacturing requirements after the desired part is sketched on the TX-2 CRT screen.

In order that the computer will understand how the sketched object will be machined, more information is needed. First of all, is it to be machined in a counterclockwise or clockwise direction? Where is the starting point? Is the inside or the outside of the shape to be machined? All of these questions in addition to tolerance, tool radius and all auxiliary functions are required to be answered before the computer can proceed to produce manufacturing sequence instructions.

The data structure in SHAPE does not identify in which direction those points are stored in sequence. Moreover, it is impossible to recognize if a point lies inside the shape or outside. Naturally, it is dangerous to let the computer pick up arbitrarily the starting point of cut.

These problems are solved most conveniently by use of the light pen. This can best be explained by referring to Figure 6-11 which is a sketch of a part to be machined. A point on the outline of this part can be pointed at by the light pen. This will be recognized as the first point (FSTPNT) by the computer. Next, the second point (SNDPNT) is pointed at, the second point being in a direction either clockwise or counterclockwise from the first point in accordance with direction in which the part is to be

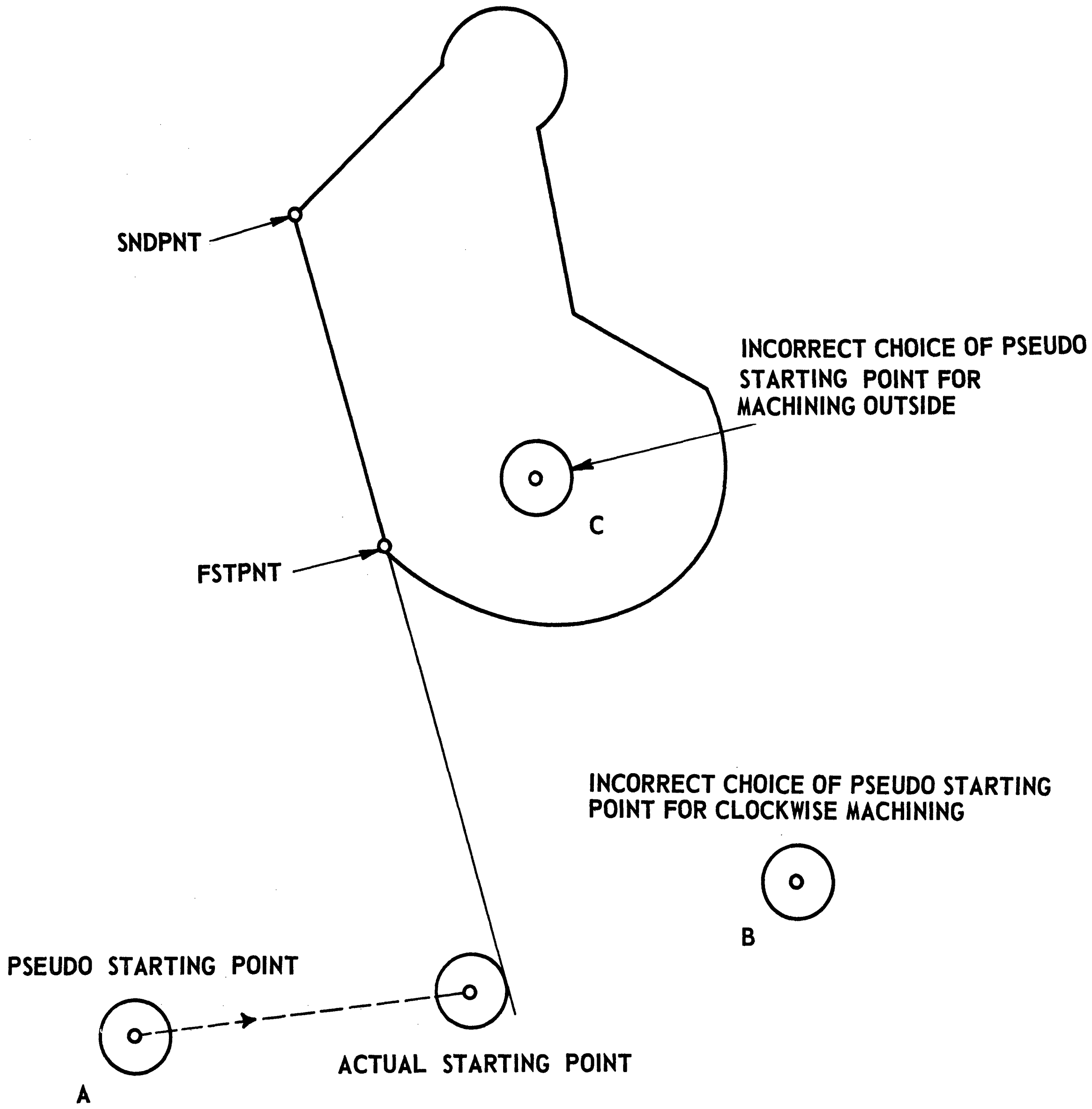


FIGURE 6-11
CHOICES OF PSUEDO STARTING POINT, FSTPNT AND SNDPNT BY LIGHT PEN IN NUMERICAL CONTROL APPLICATION

machined. Finally, the designer should select a point as the pseudo starting point. This point should be chosen outside the outline of the sample part if the designer is to machine the outside of the part; otherwise, it should be somewhere inside the part. The computer will proceed to find the actual starting point at which point the cutter will be just tangent to the first line or circle to be cut. The pseudo starting point should be on the correct side of the line or circle connecting FSTPNT and SNDPNT. For example, the starting point selected at position B of Figure 6-11 will be incorrect since it would be impossible to follow the path of from FSTPNT to SNDPNT and to machine the outside of the part.

Figure 7-12 is the photograph of a sample part sketched on the CRT screen. Figure 6-13 is the output of the subroutine CINAP which has all the necessary instruction needed for machining this part by a numerical control machine.

D. Other Potential Areas of Application

While it is not possible to mention all the possible applications of Sketchpad in design problems, few of them will be pointed out here.

1. Plant Layout Problem

By taking advantage of visual display and the

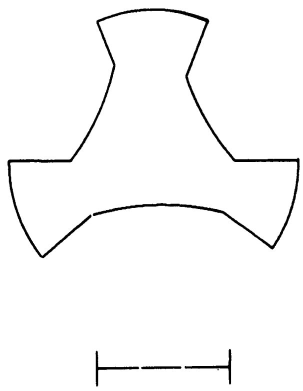


FIGURE 6-12
PHOTOGRAPH OF A SKETCHED PART TO BE MACHINED
BY A NUMERICAL CONTROL MACHINE

0	FECRT	RT	7.5000
1	SPKUL	ON	ON
2	CTCOM	XZ	OFF
3	LEFT	.5000	.0010
4	PC	3.3578	12.3578
5	FECRT	10.0000	7.5000
6	L 1	P 1	P 2
7	RC 1	P 3	9.7117
8	L 2	P 4	P 5
9	KC 2	P 6	6.0069
10	L 3	P 7	P 8
11	RC 3	P 9	10.8723
12	L 4	P 10	P 11
13	KC 4	P 12	6.0062
14	L 5	P 13	P 14
15	RC 5	P 15	9.9145
16	L 6	P 16	P 17
17	KC 6	P 18	6.0056
18	L1	P1	P2
19	SPKUL	OFF	OFF
20	CTCOM	OFF	
21	FECRT	RT	7.5000
22	PC	3.3578	12.3578
23	P 1	2.1461	8.1212
24	P 2	1.4384	6.4104
25	P 3	10.8788	8.6902
26	P 4	3.3023	2.6144
27	P 5	5.8569	2.6457
28	P 6	-.1495	2.5721
29	P 7	4.5158	-1.2107
30	P 8	2.4867	.4344
31	P 9	-.4407	-10.0363
32	P 10	-3.0059	.5291
33	P 11	-5.0346	-.9222
34	P 12	-.1495	2.5721
35	P 13	-6.1560	2.5788
36	P 14	-3.5026	2.5758
37	P 15	-10.8788	9.2010
38	P 16	-1.5256	5.9121
39	P 17	-2.4371	8.1250
40	P 18	-.1495	2.5721
41	P 19	2.1461	8.1212

FIGURE 6-13
NUMERICAL CONTROL PART PROGRAMMING INSTRUCTIONS
RESULTED FROM FIGURE 6-12

capability of the Sketchpad System to move the instances (both rotation and transformation), the plant layout problem can be studied by inventing many picture symbols to represent various machines. For example, a triangle may be drawn as an instance to mean a lathe, a circle with an inside cross is taken to mean a Grinding Machine, etc. These instances are then connected by lines to give the relative location of each instance. The light pen can be used to point at the entry point of raw materials as well as the exit point of finished products. The manufacturing rate or cost of this single flow process can then be studied. The light pen can again be used to relocate various instances until an optimized arrangement is found.

2. Heat Transfer Problem

A part whose temperature distribution is to be studied can be sketched on the CRT screen. Perhaps, two picture symbols are required. (As a matter of fact, the load vector used in the shaft display problem (INSTNO = 0), and supporting vector (INSTNO = 1) can be used here as well.) As shown in Figure 6-14, the size of instance 0 may represent the magnitude of the temperature acting on that boundary where the point of action of instance 0 lies between the two end

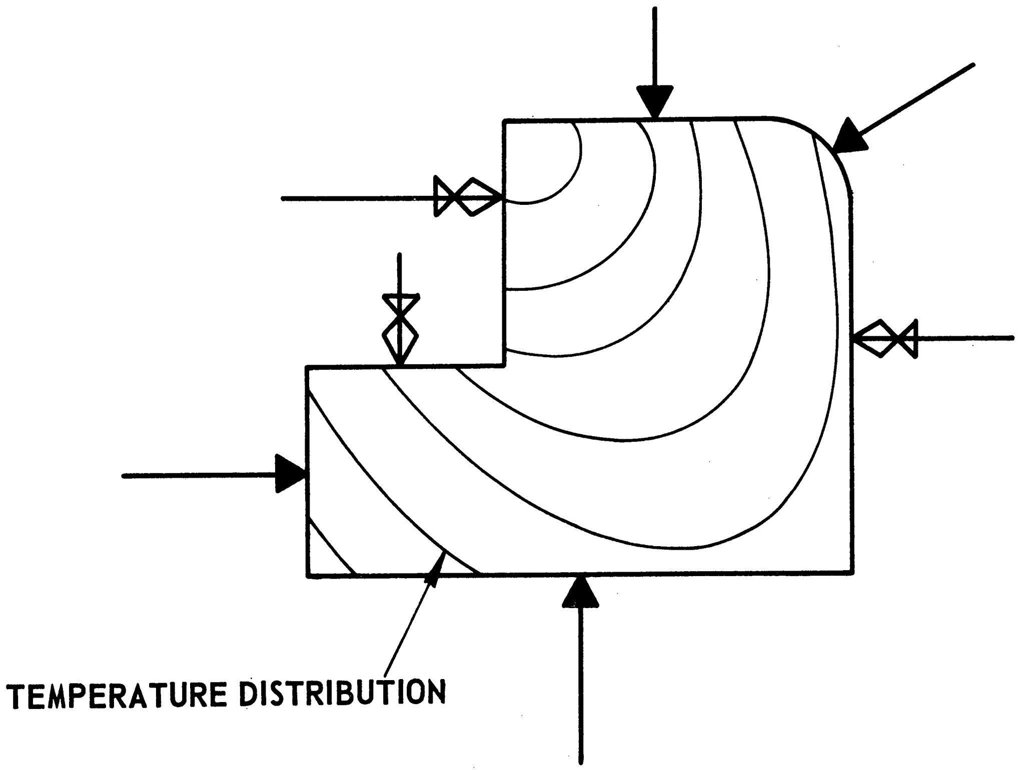


FIGURE 6-14
HEAT TRANSFER APPLICATION

points of that boundary.

Similarly, the second symbol may be taken to represent an isolated boundary regardless of its instance size. The isothermal lines of the part under those boundary conditions can then be displayed. The adjustment of either boundary conditions or the shape of the part can be made until acceptable temperature distribution is obtained.

3. Stress Analysis Problem

A part can first be drawn. The load vector with various size may be applied at any position of the part. The stress distribution or isobaric lines can then be displayed. Again, the light pen can be used to adjust either the geometry of the part or instances size and the location until stress distribution satisfied the given constraints.

4. Fluid Mechanics Problem

A pipe may be sketched as shown in Figure 6-15. The size of vectors may very well represent the pressure at inlet and outlet of the pipe. A program can be written to display isobaric lines, stream lines or velocity profiles. The designer can adjust the shape of the pipe until the desired fluid mechanics properties are observed.

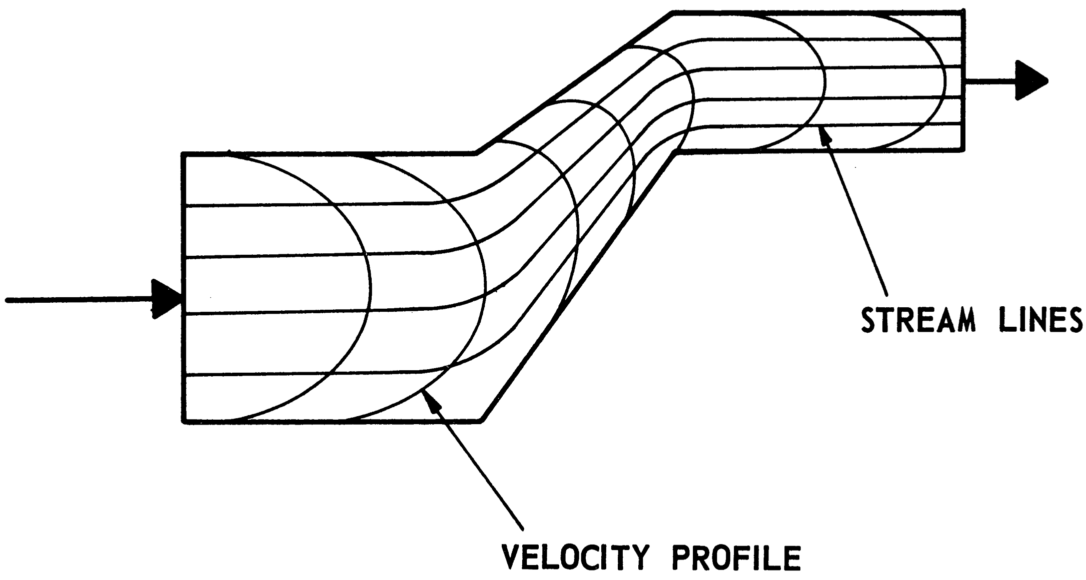


FIGURE 6-15
FLUID MECHANICS APPLICATION

CHAPTER VII

CURRENT LIMITATIONS

A. Recognition of "Multiply Connected" Pictures

1. Difficulties in Geometry Recognition

The process of recognizing drawing geometry described in Chapter 4 is restricted to "simply connected" drawings. This process will not be adequate if the drawing becomes complicated. In other words if there are more than two lines, circles, or combinations attached to a point, or if the drawing is "multiply-connected", this process fails to describe the geometry. Examples of such drawings are a hollow shaft or a shaft with key ways, screw lines or a bridge.

2. Proposed Alternate Processes

a. By Instances

The recognition of instances is independent of the lines and circles in the master picture. This suggests the use of instances for drawing of internal lines and circles in the "multiply connected" pictures.

If a single line with two attachers, A and B, as the one shown in Figure 7-1 is considered as an instance, it is possible to find the coordinates of these attachers as they appeared in the master picture.

FIGURE 7-1
INSTANCE FOR LINE DRAWING

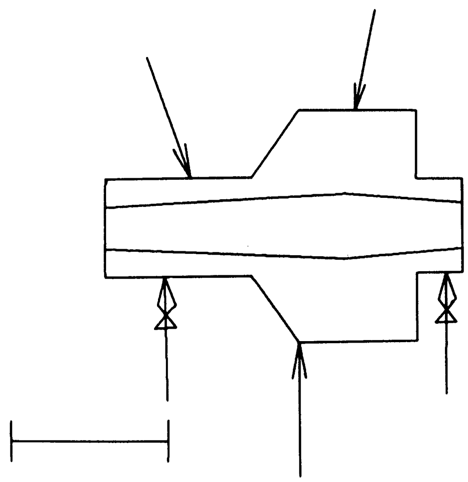


FIGURE 7-2
PHOTOGRAPH OF A SKETCHED HOLLOW SHAFT

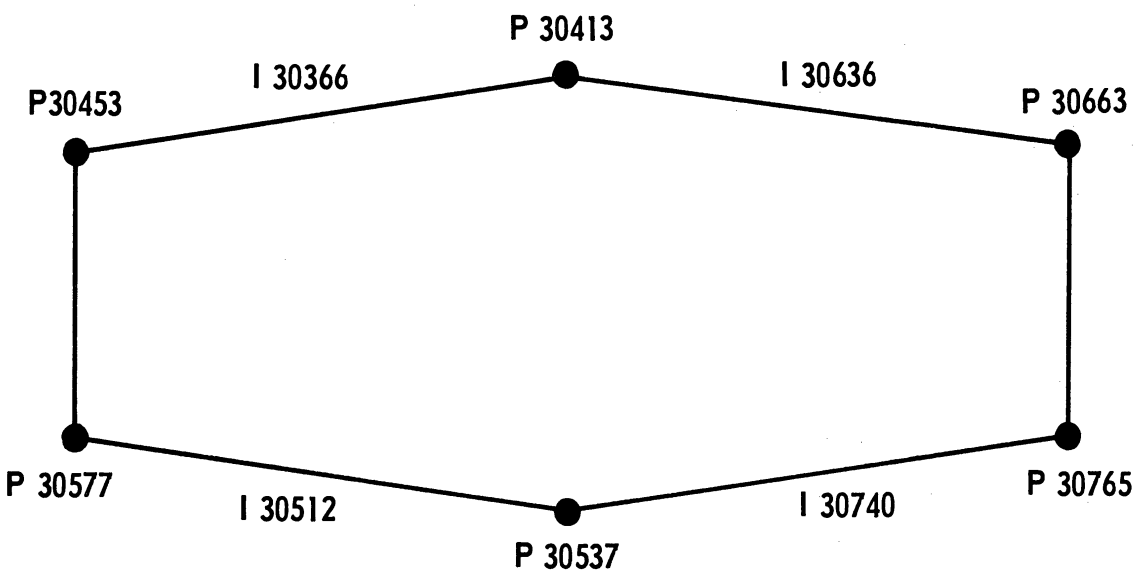
The next step is to compare those coordinates and to form the geometry.

As an example, consider a hollow shaft with loading vectors represented by arrows and supporting vectors as shown in Figure 7-2. The internal diameter of this shaft is drawn by four instances of the type shown in Figure 7-1. The coordinates of each attacher and the resulting drawing shape are shown in Figure 7-3. Clearly, it is possible to write a special subroutine to recognize the shape by comparison of the coordinates of those points.

The disadvantages of this process are that an additional subroutine is needed for each individual type of problem. Also, the process may become extremely tedious for more complicated pictures.

b. By Superposition

A better approach for geometry recognition of drawings other than "singly connected" is to consider those drawings being "multiply connected". That is, a "multiply connected" drawing is made up by superposition of several "singly connected" drawings. If this is so, the general data structure of this system can be extended to handle "multiply connected" drawings without the slightest modification in its structure. A hollow shaft with a key way shown in Figure 7-4(A) can be considered as the superposition



P = POINT
I = INSTANCE

P 30413 { X = .000229992, Y = .000182718

P 30453 { X = -.000407070, Y = .000151820

P 30537 { X = .000229992, Y = .000014283

P 30577 { X = -.000407062, Y = .000044733

P 30663 { X = .000547871, Y = .000157855

P 30765 { X = .000547871, Y = .000038952

FIGURE 7-3
GEOMETRY FORMED BY INSTANCES

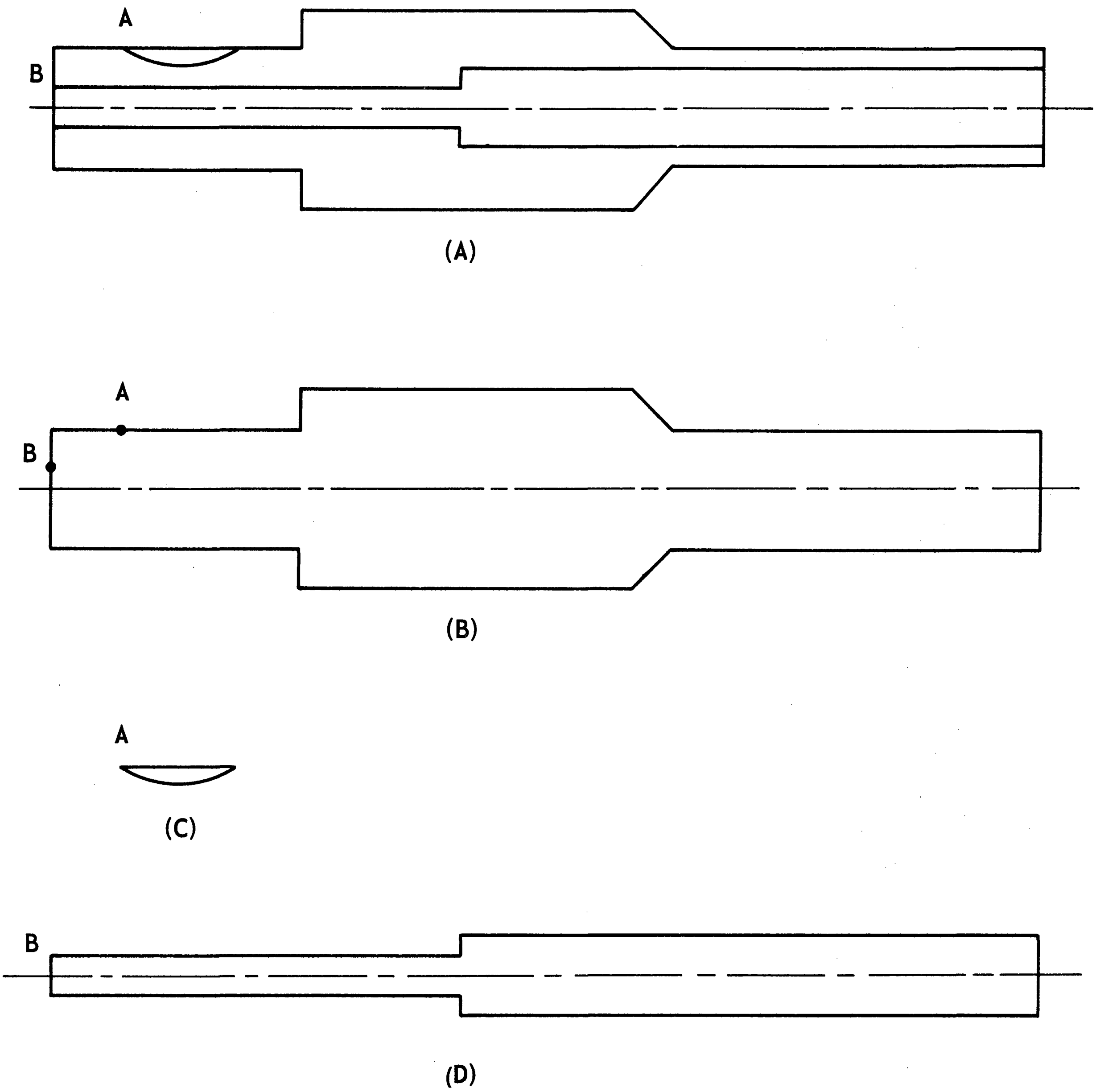


FIGURE 7-4
EXAMPLE OF "MULTIPLY CONNECTED" DRAWING

of three "simply connected" drawings shown in Figure 7-4(B), (C), and (D). In order to correlate those drawings, the designer can simply select reference points such as points A and B in Figure 7-4. These reference points provide the proper relationship between the "singly connected" drawings. The writing of a subroutine for the mathematical analysis can proceed accordingly.

B. Exact Dimensioning

The current method of scaling a drawing makes exact dimensioning impossible. The designer sketches this picture first at the scope and then generates the scaling instance. By visual judgement, the designer selects an appropriate scale factor. He could also move his scaling instance around at the scope, acting similar to a measuring rule, in order to adjust the length of a line to obtain an approximate desired length. A Sketchpad system capable of displaying numeric dimensions of any line after being adjusted by a scale factor will be highly desirable.

C. Display

The power of the system developed in this thesis has been reduced due to the lack of a display scope. The TX-2 computer at the Lincoln Laboratories of Massachusetts Institute of Technology has been under a very tight running schedule. It was not possible to use the TX-2 computer and get the immediate feedback at the scope. Nevertheless, it was possible to sketch many pictures at the TX-2 scope and get the drawing data on the magnetic tape for running by the IBM 7090 computer at the University of Michigan.

Consequently, the shaft deflection curves shown in Figure 6-3 are hand plotted based on the computer outputs. The proposed display of curve A vs. B described in Chapter 3 for certain system characteristics and human intervention of computer iteration process has not been made possible. However, it is felt that these displays involve only writing of display tables and that this limitation does not affect the general data structure previously described in Chapter 4.

CHAPTER VIII

PROPOSED FUTURE SYSTEM AND CONCLUSIONS

A. Two Dimensional Sketchpad System

1. Two Dimensions vs. Three Dimensions

A Sketchpad System that is capable of displaying three dimensional surfaces and allows the designer to manipulate those surfaces in various ways would be an extremely powerful design tool. On the other hand the present two-dimensional Sketchpad system is a sufficiently powerful tool for many applications. The designer is now linked with the computer for the solution of design problems. In fact, a 3-D Sketchpad is not always necessary. A very large portion of any design problem can in reality be described by a set of two-dimensional drawings. Each of these drawings is related geometrically in some ways. One object may require only three views--top, front and side views--while another object may need as much as ten or twenty views in order to describe it in a more precise manner. A good example of the later case is the many cross sections of an aircraft wing. As long as the geometrical relationship between drawings are kept, a two dimensional Sketchpad system is capable of handling the job of communicating pictorial information between humans and the computer. Furthermore, the computation involved in a design problem might in some cases be even less tedious than that involved with the use of three dimensional Sketchpad.

On the other hand, there are also many design problems for which a three dimensional Sketchpad may prove to be the most efficient communication means. The design of an automobile style is an example.

2. The Display of Conic and Higher Order Curves

Certainly, there are many drawings that cannot be made by combinations of lines and circles alone, Particularly, if the designer is to make the full use of the Sketchpad system from the time he conceived a design idea. Perhaps, he would start with the sketch of a rough picture of his idea at the scope and proceed to refine his design with assistance from the computer. This rough initial sketch may consist of conic curves or most likely a free hand sketch of higher order curves or curves initially described by a mesh of points for which a curve fitting technique may be required. A Sketchpad system capable of displaying and storing topological information on those curves in addition to lines and circles will be most desirable. It is worth mentioning here that the very useful nature of a parabola described in Section B of Chapter 6 can play a rather important part in shape design of an object through the Sketchpad. If a point P_3 in Figure 6-6 is pointed at by light pen and moved, all sorts of interesting parabolas corresponding to each position of point P_3 can be formed.

3. Additional Features Desired

In addition to all the useful features of Dr. Sutherland's Sketchpad System, the following additional capabilities may prove to be very desirable.

(1) The present geometry recognition program is capable of recognizing only pictures that are "singly connected". For a more complicated picture such as one with more than two curves attached to a point, the development of very general geometry recognition program appears to be very difficult. If Sketchpad will allow writing of a certain symbol into one of the registers in each of the generic blocks when an outline of the picture is completed, then a complicated picture can be made up by superimposing numbers of "single connected" drawings. The recognition of the entire

geometry will then become possible.

(2) The capability of Sketchpad to generate and manipulate subpictures (instances) by command is a very powerful tool. This capability is further amplified if the image (or mirror) of the instance can be generated in similar manner. This will enable the designer to sketch symmetrical drawings with greater ease.

(3) The numeric display of the length of lines and instance size after being adjusted by a scaling factor will be very helpful to the designer. A program can be written to display the dimension of a line when it is pointed at by the light pen.

(4) For the purpose of implementing a very general numerical control program such as the APT system into the Sketchpad system, the recording of unit direction vector ($\sin\theta$ and $\cos\theta$) every time a curve is drawn will be extremely helpful. Without this, the recognition of the exact path through which a curve is drawn will be very difficult. For example, it will be impossible to distinguish whether a circular arc is drawn in a clockwise or counterclockwise direction from the starting point to the end point.

The numerical control program described in section C of Chapter 6 assumes that all angles of circular arc cannot exceed 90 degrees. Thus, two or more circular arcs are needed to form one that has its angle of circular arc larger than 90 degrees.

B. Conclusions

It has been shown that pictorial means of communication between human and machines does indeed provide very interesting and highly promising results to the field of Computer Aided Design. By use of light pen, CRT screen and the Sketchpad System, the designer can easily communicate with the computer the idea he has conceived. The computer can then be a very active partner to the designer, to accept and analyze the designer's sketches. It will not only perform all or a substantial amount of the necessary design calculations, but most importantly, it displays the system characteristics so that the designer may make the design modifications most intelligently and efficiently until he visualizes that all the constraints applied on the system have been satisfied.

The design examples given here, shaft systems, aircraft wings and numerical control of machine tools are just a few of many possible uses of Sketchpad for design automation. The geometry and instance recognition computation system initiated by the pictorial inputs are very powerful and the data storage structure used is very general so that numerous other design applications can readily be incorporated as part of the library with little difficulty.

It appears that the practical realization of original broad concept of the Computer-Aided Design System initiated by M.I.T. is indeed not too far in the future.

BIBLIOGRAPHY

1. Coons, S. A., Notes on Graphical Input Methods, Memorandum 8436-M-17, Dynamic and Control Laboratory, Department of Mechanical Engineering, Massachusetts Institute of Technology, May 4, 1960.

2. Coons, S. A. and R. W. Mann, Computer-Aided Design Related to the Engineering Design Process, Technical Memorandum 8436-TM-5, Electronic Systems Laboratory, Massachusetts Institute of Technology, October, 1960.

3. Coons, S. A., An Outline of the Requirements for a Computer-Aided Design System, Proceedings of the Spring Joint Computer Conference, Detroit, Michigan, May 21-23, 1963. Also Technical Memorandum ESL-TM-169, Electronic Systems Laboratory, Massachusetts Institute of Technology, March, 1963.

4. Coons, S. A., Surfaces for Computer-Aided Design of Space Figures, Technical Report, Electronic Systems Laboratory, Massachusetts Institute of Technology, January, 1964.

5. Johnson, T. E., Sketchpad III, Three Dimensional Graphical Communication with a Digital Computer, M. S. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, May, 1963.

6. Johnston, L. E., Graphical Communication with a Digital Computer, S. M. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, June, 1961.

7. Licklider, J.C.R., and W.E. Clark, On-Line Man-Computer Communication, Proc. AFIPS Spring Joint Computer Conference 21, 113 (1962).

8. Ling, Marvin T. S., Analytical and Experimental Studies of the Dynamics of Liquid-Lubricated Bearings, Progress Report No. 3, Project No. M-98160-3, The Cincinnati Milling Machine Company, February, 1960.

9. Ling, Marvin, T. S., On the Optimization of the Stiffness of Externally Pressurized Bearings, Transaction of ASME, Series D, No. 2, Vol. 84, March, 1962.

10. Loewe, R. T., et al., Computer Generated Displays, Proc. of the IRE, Vol. 49, No. 1, p. 185-195, January, 1961.

11. Parmelee, R. P., A Study of a Stress Analysis Facility for Computer Aided Design, S. M. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, May, 1961.

12. Purvis, J. D., Jr., An Investigation of a Standard Parts Selection Facility for Computer-Aided Design, S. M. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, August, 1961.

13. Smith, A. F., Method for Computer Visualization, Technical Memorandum 8436-TM-2, Electronic Systems Laboratory, Massachusetts Institute of Technology, September, 1960.

14. Sevcik, J. K., System Vibration and Static Analysis, ASME Paper No. 63-AHGT-57, January 15, 1963.

15. Stotz, R. H., Specialized Computer Equipment for Generation and Display of Three Dimensional Curvilinear Figures, Technical Memorandum ESL-TM-167, Electronic Systems Laboratory, Massachusetts Institute of Technology, March, 1963.

16. Sutherland, I. E., Sketchpad, A Man-Machine Communication System, Technical Report No. 296, Lincoln Laboratory, Massachusetts Institute of Technology, January 30, 1963.

17. Randa, G. C., Design of a Remote Display Console, Report ESL-R-132, M.I.T. Project DSR 8753, Electronic Systems Laboratory, Massachusetts Institute of Technology, February, 1962.

18. Randa, G. C., and J. Grondstra, Design for a Manual Intervention Console, Internal Memorandum 8753-M-55, Electronic Systems Laboratory, Massachusetts Institute of Technology, March, 1962.

19. Roberts, L. G., Machine Perception of Three Dimensional Solids, Ph.D. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, June, 1963.

20. Ross, D. T., Computer-Aided Design: A Statement of Objectives, Technical Memorandum 8436-TM-4, Electronic Systems Laboratory, Massachusetts Institute of Technology, September, 1960.

21. Ross, D. T., and S. A. Coons, Investigations in Computer-Aided Design for Numerically Controlled Production, M.I.T. Report ESL-IR-164, Project

DSR 8753, Interim Technical Progress Report No. 5, Electronic Systems Laboratory, Massachusetts Institute of Technology, February, 1963.

22. Ross, D. T. and J. E. Rodriguez, Theoretical Foundations for the Computer-Aided Design System, Proceedings of the Spring Joint Computer Conference, Detroit, Michigan, May 21-23, 1963. Also, Technical Memorandum ESL-TM-170, Electronic Systems Laboratory, Massachusetts Institute of Technology, March, 1963.

23. Verderber, J. A., Graphical Input-Output Devices for Computer-Aided Design, S. M. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, June, 1961.

24. Walsh, J. F., and A. F. Smith, Computer Utilization, Interim Engineering Report 6873-IR-10 and 11, Electronic Systems Laboratory, Massachusetts Institute of Technology, November 30, 1959.

25. ______________, Investigations in Computer-Aided Design, Interim Engineering Report 8436-IR-1, Electronic Systems Laboratory, Massachusetts Institute of Technology, May 30, 1960.

26. ______________, Investigations in Computer-Aided Design, Interim Engineering Report 8436-IR-2, Electronic Systems Laboratory, Massachusetts Institute of Technology, February 28, 1961.

27. ______________, Programmers Manual for Cincinnati Acramatic Numerical Control System, The Cincinnati Milling Machine Company, Cincinnati, Ohio.

28. ______________, Acramatic Thousand Series Universal Numerical Control Programmer's Manual-DE, The Cincinnati Milling Machine Company, Cincinnati, Ohio

APPENDIX A

EXTERNAL FUNCTIONS

A. Basic Functions

1. SHAPE

Purpose: To find all information on points and store them in sequence in PICRCD and CIRCEN as they form the outline of a geometric shape.

Call: SHAPE. (LIST, GPICTP, PICRCD, FPICRD, CIRCEN, FCIRCN, SWP)

Arguments:

LIST	Sketchpad II data as dumped from TX-2 computer.
GPICTP	Type of problem to be studied on the picture drawn.
PICRCD	A list of locations to store the information on the picture, in fixed point.
FPICRD	Same as PICRCD, except in floating point.
CIRCEN	A list of locations to store the information in circles appeared in the picture. Fixed point.
FCIRCN	Same as CIRCEN, except in floating point.
SWP	Working picture size.

Note:

Equivalence (PICRCD, FPICRD), (CIRCEN, FCIRCN)

2. <u>INSTN</u>

Purpose: To find all information on INSTANCES and store them in PICRCD.

Call: INSTN. (LIST, INSTNO, GPICTP, PICRCD, FPICRD)

Arguments:

LIST	Sketchpad II data as dumped from TX-2 computer.
INSTNO	Instance number.
GPICTP	Type of problem to be studied of the picture drawn.
PICRCD	A list of locations to store the information on circles appeared in the picture. Fixed point.
FPICRD	Same as PICRCD, except in floating point.

<u>Note</u>:

Equivalence (PICRCD, FPICRD)

3. SCALE

Purpose: To find SM, SI and SCFACT to calculate the final scaling factor.

Call: SCALE. (LIST, SM, SI, SCFACT)

Arguments:

LIST	Sketchpad II data as dumped from TX-2 computer.
SM	Size of Instance Master drawing.
SI	Size of Instance.
SCFACT	Scaling factor written in by designer through togle switch.

4. <u>POINTS</u>

Purpose: To form a condensed data set into a string of storage called TABLE.

Call: POINTS. (GPICTP, PICRCD, FPICRD, TABLE, FTABLE)

Arguments:

GPICTP	Type of problem to be studied on the picture drawn.
PICRCD	A list of locations to store the information on the picture, in fixed point.
FPICRD	Same as PICRCD, except in floating point.
TABLE	A list of locations where condensed data on SHAPE will be stored, in fixed point.
FTABLE	Same as TABLE, except in floating point.

<u>Note</u>:

Equivalence (PICRCD, FPICRD), (TABLE, FTABLE)

5. <u>LADSPT</u>

Purpose: To form condensed data about the information on Instances on locations LOAD and SUPPRT.

Call: LADSPT. (GPICTP, PICRCD, FPICRD, LOAD, FLOAD, SUPPRT, FSUPRT)

Arguments:

GPICTP	Type of problem to be studied on the picture drawn.
PICRCD	A list of locations to store the information on the picture, in fixed point.
FPICRD	Same as PICRCD, except in floating point.
LOAD	A list of locations where condensed data in Instance No. 0 will be stored, in fixed point.
FLOAD	Same as LOAD, except in floating point.
SUPPRT	A list of locations where condensed data on Instance No. 1 will be stored, in fixed point.
FSUPPRT	Same as SUPPRT, except in floating point.

Note:

Equivalence (PICRCD, FPICRD), (LOAD, FLOAD), (SUPPRT, FSUPPRT)

6. <u>CONV</u>

Purpose: To convert a data from one's complement fixed point number to a floating number.

Call: CONV. (X)

Argument:

X	One's complement fixed point number

7. <u>MTAPE</u>

Purpose: To read TX-2 dumped data from magnetic tape and further shift storing of data from forward to backward into string of locations LIST.

Call: MTAPE. (LIST)

Argument:

LIST — A string of locations where sketchpad II dumped data will be stored.

B. Special Functions

1. SHAFT

Purpose: To calculate the diameter and length of each section based on the data in TABLE.

Call: SHAFT. (TABLE, FTABLE, D, LL, SECTN)

Arguments:

TABLE	A list of location where condensed data in SHAPE are stored, in fixed point.
FTABLE	Same as TABLE, except in floating point.
D	Diameter of each shaft section.
LL	Length of each shaft section.
SECTN	Total number of shaft sections.

Note:

Equivalence (TABLE, FTABLE)

2. COMBIN

Purpose: To combine load vectors and supporting vectors to shaft sections found from SHAFT and create new set of section diameters and section length.

Call: COMBIN. (LOAD, FLOAD, SUPPRT, FSUPRT, SECTN, TABLE, FTABLE, D, LL, POD, PID, PP, PK, PL, PM, SCNCNT, KREQRD)

Arguments:

LOAD	A list of locations where condensed data on Instance No. 0 are stored, in fixed point.
FLOAD	Same as LOAD, except in floating point.
SUPPRT	A list of locations where condensed data on Instance No. 1 are stored, in fixed point.
FSUPRT	Same as SUPPRT, except in floating point.
SECTN	Total number of shaft sections resulting from SHAFT.
TABLE	A list of locations where condensed data on SHAPE are stored, in fixed point.
FTABLE	Same as TABLE, except in floating point.
D	Shaft section diameter from SHAFT.
LL	Shaft section length from SHAFT.
POD	New shaft section outside diameter.
PID	New shaft section inside diameter.

PP	Load at each section.
PK	Support at each section.
PL	New shaft section length.
PM	Moment at each section.
SCNCNT	New total number of sections.
KREQRD	Minimum size of supporting vector.

<u>Note</u>:

Equivalence (LOAD, FLOAD), (SUPPRT, FSUPRT),
(TABLE, FTABLE).

3. <u>DISPLY</u>

Purpose: To insert zero load vectors and zero support vectors in order to reduce each section length so that a rather continuous shaft deflection curve can be displayed.

Call: DISPLY. (SCNCNT, SPACNG, POD, PID, PP, PIC, PL, PM, OD, ID, P, K, L, M, TOTSEC)

Arguments:

SCNCNT	Total number of sections resulting from COMBIN.
SPACNG	The maximum length allowed for any section length.
POD	Section outside diameter.
PID	Section inside diameter.
PP	Load at each section.
PK	Support at each section.
PL	Section length.
PM	Moment at each section.
OD	New section outside diameter.
ID	New section inside diameter.
P	Load at each section.
K	Support at each section.
L	Section length.
M	Moment at each section.
TOTSEC	New total number of sections.

4. SHFDEF

Purpose: To calculate the shaft deflection at each section.

Call: SHFDEF. (OD, ID, L, P, K, M, EP, TOTSEC, DELTA)

Arguments:

OD	Section outside diameter.
ID	Section inside diameter.
L	Section length.
P	Load at each section.
K	Support at each section.
M	Moment at each section.
EP	Young's modulus.
TOTSEC	Total number of sections.
DELTA	Shaft deflection at each section.

5. BEARING

Purpose: To study characteristics of an externally pressurized liquid lubricated journal bearing.

Call: BEARING. (IZ, NT, R, SB, CL, AL, PL, V, U, PS, W, APPP, S, SUMK, SUMFV, SUMQ, CONST, AP, PRE)

Arguments:

IZ	A switch. If 1, bearing characteristics at all eccentricity ratios are calculated, if 2, only those at eccentricity ratio of zero are calculated.
NT	Total number of pocket.
R	Radius of the shaft; inches.
SB	Pocket width; inches.
CL	Radial clearance; inches.
AL	Axial bearing land; inches.
PL	Peripheral bearing land; inches.
V	Surface velocity; inches per second.
U	Viscosity; Reyn.
PS	Supply pressures; psi.
W	Pocket to supply pressure ratio.
APPP	Angle initially incremented during interactions; radians.
S	Eccentricity ratio.
SUMK	Rigidity; pounds per inch.

SUMFV	Load carrying capacity; pound.
SUMQ	Flow; cubic inches per second.
CONST	Restrictor constant.
AP	Attitude angle; radians.
PRE	Pocket pressure; psi.

6. WINGS

Purpose: To calculate various aircraft wing preliminary design parameters.

Call: WINGS. (TABLE, FTABLE, AREA, SPAN, ASPECT, RTCHRD, TPCHRD, TAPER, SWEEP, THCKNS, THKRTO, DIHEDL, INCIDR, INCIDT)

Arguments:

TABLE	A link of locations where condensed data on SHAPE are stored, in fixed point.
FTABLE	Same as TABLE, except in floating point.
AREA	Aircraft wing area; square feet.
SPAN	Span; feet.
ASPECT	Aspect ratio.
RTCHRD	Root Chord; feet.
TPCHRD	Tip Chord; feet.
TAPER	Taper ratio.
SWEEP	Sweep angle, radians.
THCKNS	Thickness; feet.
THKRTO	Thickness ratio.
DIHEDL	Dihedral angle; radians.
INCIDR	Root incidence angle; radians.
INCIDT	Tip incidence angle; radians.

7. CINAP

Purpose: To produce CINAP instructions for numerically controlled machine tools.

Call: CINAP. (GPICTP, PICRCD, FPICRD, CIRCEN, FCIRCN, FEDRTN, SPKULN, CTCOMN, OFFSET, TLRAD, TOLER, FSTPNT, SNDPNT, XCORD, YCORD, DELTAX, ZCORD1, ZCORD2, FERTXY)

Arguments:

GPICTP	Type of problem to be studied on the picture drawn.
PICRCD	A list of locations to store the information on the picture, in fixed point.
FPICRD	Same as PICRCD, except in floating point.
CIRCEN	A list of locations to store the information on circles appeared in the picture. Fixed point.
FCIRCN	Same as CIRCEN, except in floating point.
FEDRIN	Feedrate number. 1 for rapid traverse; 2 otherwise.
SPKULN	Spindle and coolant. 1 for spindle on coolant on; 2 for (off, off), 3 for (high, flood), 4 for (low, mist).
CTCOMN	Cutter compensation. 1 for (XY, on), 2 for (XZ, off), 3 for YZ plane, 4 for off.
OFFSET	Cutter offset, 1 for right; 2 for left.
TLRAD	Tool radius; inches.
TOLER	Tolerance; inches.

FSTPNT	First point on the drawing.
SNDPNT	Second point on the drawing.
XCORD	X-coordinate of pseudo staring point.
YCORD	Y-coordinate of pseudo starting point.
DELTAX	A number incremented during calculation of distance; inches.
ZCORD1	Feed rate in Z-direction.
ZCORD2	Feed rate in Z-direction.
FERTXY	Feed rate in X or Y direction.

APPENDIX B

STATIC DEFLECTION OF A CIRCULAR, NON-UNIFORM SHAFT ON FLEXIBLE SUPPORTS

The calculation of the deflection of a shaft with an arbitrary number of supports and concentrating loads is a statically indeterminate problem. Consider the shaft to be composed of n-sections; each of them is subjected to various forces and moments, as indicated in the Figure B-1. The moment at arbitrary point x is:

$$EI_i \frac{d^2y}{dx^2} = (R_i - V_i - P_i) X + M_i + S_i \qquad (1)$$

where

R_i = Reaction force

V_i = Shearing force

P_i = Applied load

M_i = Moment

S_i = Applied moment

Integrating equation (1), we obtain expressions for slope and deflection:

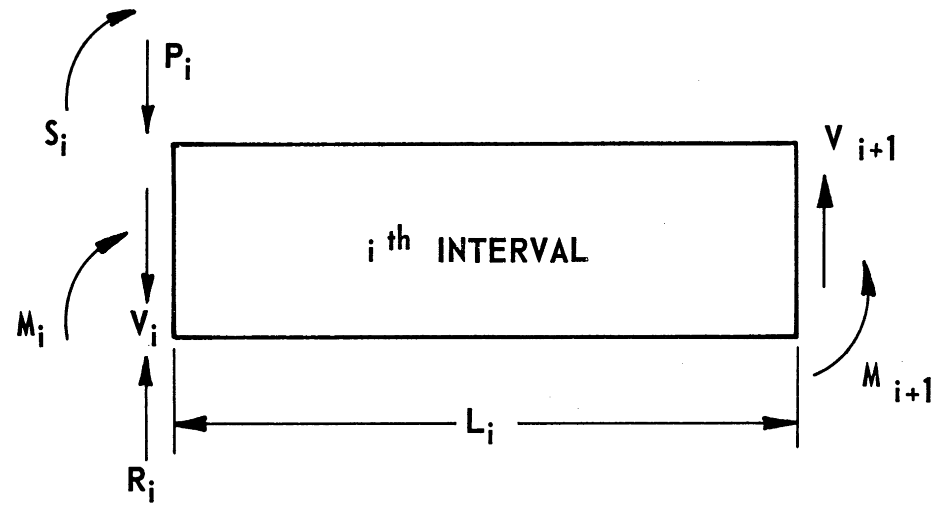


FIGURE B-1
SHAFT SECTION WITH FORCES AND MOMENTS

$$EI_i \frac{dy}{dx} = (R_i - V_i - P_i)\frac{X^2}{2} + (M_i + S_i)X + C_{i_1} \quad (2)$$

$$EI_i y = (R_i - V_i - P_i)\frac{X^3}{6} + (M_i + S_i)\frac{X^2}{2} + C_{i_1} X + C_{i_2} \quad (3)$$

where

$$C_{i_1} = EI_i \frac{dy_i}{dx} \quad \text{and } C_{i_2} = EI_i y_i$$

since $\frac{dy}{dx} = \tan\theta$ for small θ and

$R_i = k_i y_i$ for flexible supports (k_i = stiffness), the expressions for θ_{i+1}, y_{i+1}, V_{i+1} and M_{i+1} are obtained by letting $x = \ell_i$ in the above equations.

$$\theta_{i+1} = (k_i y_i - V_i - P_i)\frac{\ell_i^2}{2EI_i} + (M_i + S_i)\frac{\ell_i}{EI_i} + \theta_i \quad (4)$$

$$y_{i+1} = (k_i y_i - V_i - P_i)\frac{\ell_i^3}{6EI_i} + (M_i + S_i)\frac{\ell_i^2}{2EI_i} + \theta_i \ell_i + y_i \quad (5)$$

$$V_{i+1} = V_i + P_i - k_i \ell_i \quad (6)$$

$$M_{i+1} = M_i + (k_i \ell_i - V_i - P_i)\ell_i + S_i \quad (7)$$

Now, in order to express, V_i, M_i, θ_i and y_i in terms of y_0 and θ_0, we assume that they take the following forms:

$$V_i = A_i + B_i y_0 + C_i \theta_0 \tag{8}$$

$$M_i = D_i + E_i' y_0 + F_i \theta_0 \tag{9}$$

$$\theta_C = P_i' + \theta_i y_0 + R_i' \theta_0 \tag{10}$$

$$y_i = S_i' + T_i y_0 + U_i \theta_0 \tag{11}$$

The coefficients in equation (8) through (11) are found by substituting them into equations (4) through (7). These coefficients at i = 1 are all zero except

$$R_i' = T_1 = 1$$

Since $V_n = A_n + B_n y_0 + C_n \theta_0 = 0$

and $M_n = D_n + E_n' y_0 + F_n \theta_0 = 0$

thus,

$$y_0 = \frac{\begin{vmatrix} -A_n & C_n \\ -D_n & F_n \end{vmatrix}}{\begin{vmatrix} B_n & C_n \\ E_n' & F_n \end{vmatrix}} \qquad \theta_0 = \frac{\begin{vmatrix} B_n & -A_n \\ E_n' & -D_n \end{vmatrix}}{\begin{vmatrix} B_n & C_n \\ E_n' & F_n \end{vmatrix}}$$

Consequently, shearing force, moment, slope and deflection at any section of the shaft can be calculated through equations (8) through (11).

APPENDIX C

STUDIES ON EXTERNALLY PRESSURIZED LIQUID-LUBRICATED JOURNAL BEARINGS*

1. Bearings with Axial Grooves:

Considering each pocket separately as shown in Figure C-1 there are two types of flow, namely, axial and circumferential flow. Both types of flow can be found by applying the fundamental equation for incompressible fluid through a finite slot.

$$Q = \frac{\Delta Pbh^3}{12\mu \ell} \qquad (1)$$

where

- b = slot width
- ℓ = slot length
- h = slot height and
- ΔP = pressure difference between two ends of the slot.

The axial flow is therefore

$$Q_{a_n} = 2 \int_{\theta_{n,1}}^{\theta_{n,2}} \frac{P_n rh^3 d\theta}{12\mu \ell_a}$$

---

*The results of the detail studies of this type bearings is to be published.

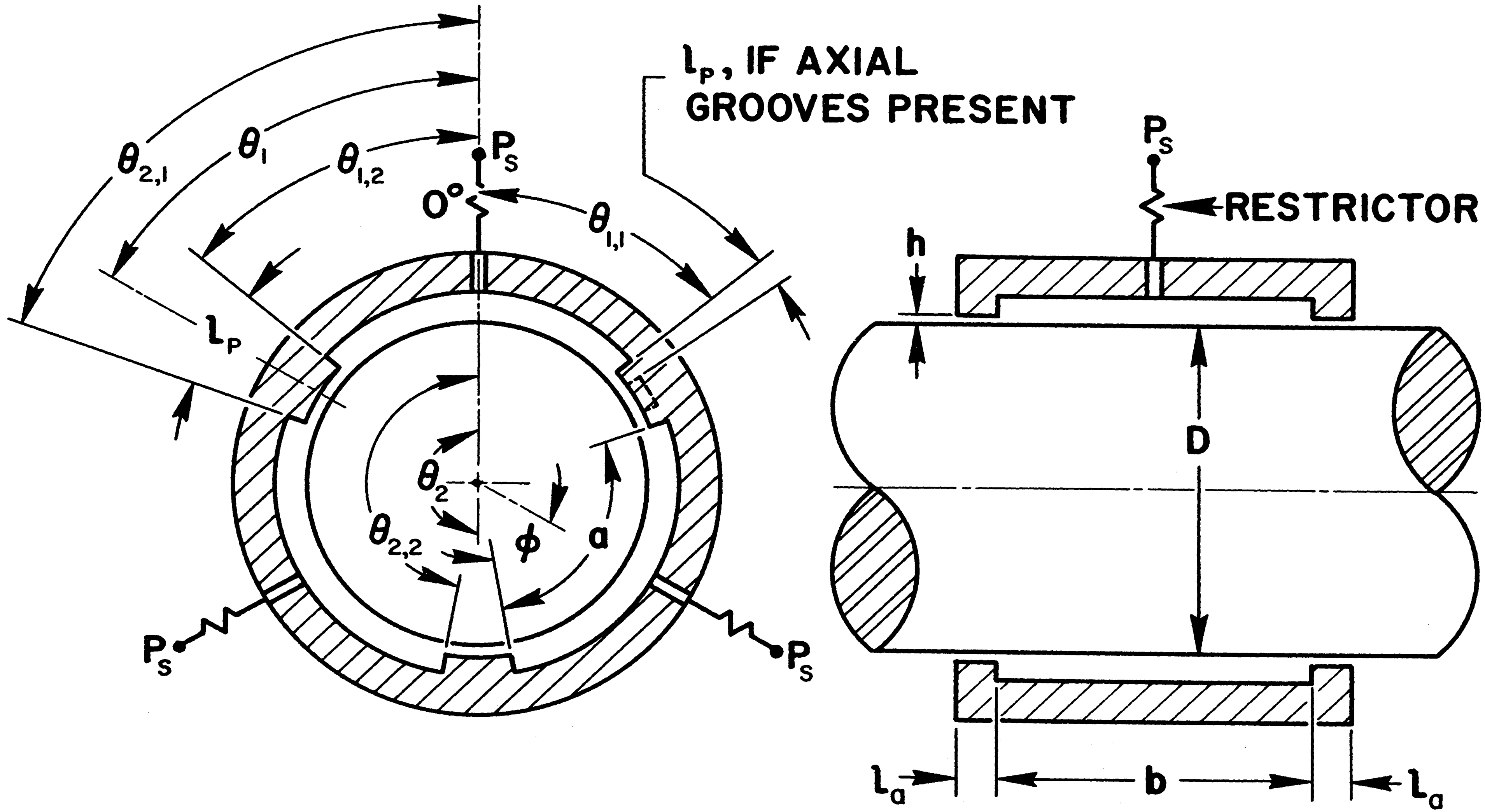


FIGURE C-1
EXTERNALLY PRESSURIZED JOURNAL BEARING

where n=pocket number = 1, 2, 3 ---------N

N=total number of pockets

Replacing h by $h = c(1 + \varepsilon \cos\theta)$,

$$\theta_{a_n} = 2 \int_{\theta_{n,1}}^{\theta_{n,2}} \frac{P_n r c^3 (1 + \varepsilon \cos\theta)^3}{12 \mu \ell_a} d\theta = \frac{r c^3 P_n E_n}{12 \mu \ell_a} \qquad (2)$$

where

$$E_n = 2 \int_{\theta_{n,1}}^{\theta_{n,2}} (1 + \varepsilon \mathrm{Cos}\theta)^3 \, d\theta$$

The circumferential flow is:

$$Q_{P_n} = \frac{b(h_{n,1}^3 + h_{n,2}^3) P_n}{12 \mu \ell_p} \qquad (3)$$

where

$$b = L - 2\ell_a$$

The flow through a capillary restrictor can be expressed by

$$\theta_n = \frac{\Pi d_c^4}{128 \mu \ell_c} P_s (1 - P_n / P_s) \qquad (4)$$

Since the hydrodynamic effect is ignored, because of continuity, the bearing flow should be equal to the capillary flow.

$$Q_n = Q_{a_n} + Q_{P_n} \tag{5}$$

Substituting equations (2), (3) and (4) into (5)

$$\frac{\pi d_c^{\,4}}{128\,\mu \ell_c} P_s(1-P_n/P_s) = \frac{r c^3 P_n E_n}{12 \mu \ell_a} + \frac{b(h_{n,1}^3 + h_{n,2}^3)}{12 \mu \ell_p} P_n$$

or

$$A(1-P_n/P_s) = Bc^3 E_n\, P_n/P_s + C(h_{n,1}^3 + h_{n,2}^3)\, P_n/P_s$$

Solving for P_n/P_s,

$$P_n/P_s = \frac{A}{A+Bc^3E_n+C(h_{n,1}^3 + h_{n,2}^3)} \tag{6}$$

Equation (6) expresses the pressure at each pocket for various bearing configurations and eccentricity ratios.

The load carrying capacity of the bearing is the algebraic sum of the forces exerted by the constant pocket pressure acting on the pocket area and the linearly dropped pressure acting on the bearing lands.

$$F_n = P_n \int_{\theta_{n,1}}^{\theta_{n,2}} (b + 2\frac{\ell_a}{2})\, r\cos\theta \, d\theta + \frac{P_n}{2} \ell_p b(\cos\theta_{n,1} + \cos\theta_{n,2})$$

$$= \frac{P_n}{P_s} [r(b + \ell_a)(\sin\theta_{n,2} - \sin\theta_{n,1}) + \frac{\ell_p b}{2} (\cos\theta_{n,1} + \cos\theta_{n,2})]$$

$$P_s \qquad (7)$$

Therefore, the total bearing load carrying capacity is

$$W = \sum_{n=1}^{N} F_n \qquad (8)$$

The bearing stiffness is the first derivative of load with respect to eccentricity

$$K = \frac{d_W}{d_e} = \frac{d_W}{c d\varepsilon} = \sum_{h=1}^{N} [r(b + \ell_a)(\sin\theta_{n,2} - \sin\theta_{n,1}) + \frac{\ell_p b}{2}$$

$$\cos\theta_{n,1} + \cos\theta_{n,2})] \times \frac{A[Bc^3\dot{E}_n + C(H_{n,1} + H_{n,2})]}{[A + Bc^3E_n + C(h^3_{n,1} + h^3_{n,2})]^2} P_s \qquad (9)$$

where

$$H_{n,1} = \frac{d}{d\varepsilon}(h^3_{n,1}) = 3h^2_{n,1}\, c \cdot \cos\theta_{n,1}$$

$$H_{n,2} = \frac{d}{d\varepsilon}(h^3_{n,2}) = 3h^2_{n,2}\, c \cdot \cos\theta_{n,2}$$

and

$$\dot{E}_n = \frac{dE_n}{d\varepsilon}$$

The total bearing flow is the summation of the flow throughout each pocket.

$$Q_{Total} = \sum_{n=1}^{N} [B_c{}^3 E_n + C(h^3_{n,1} + h^3_{n,2})]P_n \qquad (10)$$

For other types of restrictors such as orifice or constant flow valve, the similar approach in analysis can be used.

(2) Bearings without axial grooves:

The removal of pocket separating axial grooves destroys the independency of each pocket. Due to the pressure difference existing between various pockets at certain eccentricity ratio, there will be pressure-induced circumferential flow from a pocket to its adjacent pockets. Thus, each pocket pressure is affected by the others. The continuity equations, taking into account this pressure-induced circumferential flow and velocity-induced flow, becomes

$$Q_{1,IN} = Q_{1,OUT} + Q_{1,2} + Q_{1,N} - \Delta Q_{V,1}$$

$$Q_{2,IN} = Q_{2,OUT} + Q_{2,3} + Q_{2,1} - \Delta Q_{V,2}$$

.
.
.

$$Q_{N,IN} = Q_{N,OUT} + Q_{N,N+1} + Q_{N,N-1} - \Delta Q_{V,N}$$

where

$$Q_{N,IN} = K_c(P_s - P_n)$$

$$Q_{N,OUT} = \frac{rc^3 E_n}{12 \mu \ell_a} P_n$$

$$Q_{N,N+1} = \frac{bh^3{}_n}{12 \mu \ell_a} (P_n - P_{n+1} \qquad (12)$$

$$Q_{N,N-1} = \frac{bh^3{}_{n-1}}{12 \mu \ell_p} (P_n - P_{n-1})$$

and $\Delta Q_{V,N} = \frac{bV}{2} (h_n - h_{n-1})$

Substituting equations (12) into (11) and rearranging them in terms of pressure ratios P_n / P_s, they become n-linear equations with n-unknowns. Each pressure ratio can thus be solved. The bearing load, stiffness and flow are expressed by:

$$W = \sum_{n=1}^{N} A_{n_{eff.}} \quad P_s \, . \, \frac{P_n}{P_s} \qquad (13)$$

$$K = \sum_{n=1}^{N} A_{n_{eff.}} \frac{P_s}{c} \cdot \frac{\dot{P}_n}{P_s} \qquad (14)$$

$$Q = \sum_{n=1}^{N} \frac{rc^3}{12\mu \ell_a} E_n P_n \qquad (15)$$

where

$$A_{n_{eff.}} = \int_{\theta_{n,1}}^{\theta_{n,2}} r(b+\ell a) \cos\theta d\theta$$

The term $\dot{P}_n/P_s$ which appeared in equation (14) is found by differentiating pressure ratio equations with respect to eccentricity ratio. It is very easy to note that the same procedure used to solve P_n/P_s can be used to solve $\dot{P}_n/P_s$.

APPENDIX D

PROGRAM FOR MODE CONVERSION

The TX-2 computer uses one's complement system. All numbers fall between -1.0 and +1.0. To convert its fixed point number into the floating point number, the following program is used.

```
CLA     X
TPL     *+3
CLA     =K777777777777
SUB     X
ARS     8
ORA     =K200K9
FAD     =K200K9
STθ     X
```