AIM

Association for Information and Image Management

1100 Wayne Avenue, Suite 1100
Silver Spring, Maryland 20910

301/587-8202

Centimeter

Inches

MANUFACTURED TO AIM STANDARDS
BY APPLIED IMAGE, INC.

1 of 2

DOCUMENT NO.

HW-78712

SERIES AND COPY NO.

DATE August 21, 1963

GENERAL ELECTRIC DECLASSIFIED

HANFORD ATOMIC PRODUCTS OPERATION - RICHLAND, WASHINGTON

THIS DOCUMENT CONTAINS RESTRICTED DATA
DEFINING THE ATOMIC ENERGY ACT
IT IS A VIOLATION OF THE ATOMIC ENERGY ACT
TO REPRODUCE OR TRANSMIT IN ANY MANNER
THE CONTENTS IN ANY MANNER TO ANY UNAUTHORIZED
PERSON IS PROHIBITED.

TITLE

A PROGRAM FOR THE NUMERICAL CONTROL OF A PULSE INCREMENT SYSTEM

☐ OTHER OFFICIAL CLASSIFIED INFORMATION
THIS MATERIAL CONTAINS INFORMATION AFFECTING THE NATIONAL DEFENSE OF THE UNITED STATES WITHIN THE MEANING OF THE ESPIONAGE LAWS, TITLE 18, U. S. C., SECS. 793 AND 794, THE TRANSMISSION OR REVELATION OF WHICH IN ANY MANNER TO AN UNAUTHORIZED PERSON IS PROHIBITED BY LAW.

AUTHOR

D. C. GRAY

ISSUING FILE

RECEIVED AREA

THIS DOCUMENT IS TO BE KEPT UNATTENDED & WHERE AN UNAUTHORIZED PERSON MAY HAVE ACCESS
TO IT. IT IS NOT TO BE USED OR MUST BE STORED IN AN APPROVED LOCKED DEPOSIT WITHIN AN APPROVED
DEPOSIT RECEPT. IF A LOSS OCCURS, REPORT IT IMMEDIATELY TO THE ISSUING AGENCY. RECEIVED FROM
ALL RELEASES, THIS YEAR, IS TO BE KEPT IN ITS ORIGINAL FORM AND ITS CONTENTS ARE IN THE LIMITS OF
THIS PROJECT AND FOR ANY AUTHORITY TO BE IN ITS TRANSFER TO ANY OTHER AGENCY OR FOR ANY
OTHER USE. IF IT IS TO BE USED IN ANY OTHER MANNER, AN ADDITIONAL COPY IS REQUIRED
TO BE OBTAINED FROM THE ISSUING AGENCY. ADVISORY: IF THIS DOCUMENT IS USED IN
TO SIGN IN THE SPACE PROVIDED BELOW.

ROUTE TO:

PAYROLL NO.

LOCATION

FILE NO. 100-368690-100

SIGNATURE AND DATE

MAY 22 1968

Access limited to personnel specifically
authorized to receive
ARMED WEAPON DATA

~~DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED~~

54-3000-340 (3-57) AEC-GE RICHLAND, WASH.

(CLASSIFICATION)

DECLASSIFIED

DECLASSIFIED

HW-1812

ACCESS LIMITED TO PERSONNEL

This document classified by

R. Y. Dean

This document consists of
148 pages.

A PROGRAM FOR THE NUMERICAL CONTROL
OF A PULSE INCREMENT SYSTEM

By

D. C. Gray

Applied Mathematics Operation
Hanford Laboratories

Classification Cancelled and Changed To

DECLASSIFIED

By Authority of DS Lewis

CG-PR-2, 3-28-94

By DK Hanson, 4-16-94

Verified By Garre Maley
4-18-94

August 21, 1963

HANFORD ATOMIC PRODUCTS OPERATION
RICHLAND, WASHINGTON

DISTRIBUTION

1 - C A Bennett	8 - J C Stouffer
2 - R Y Dean	9 - I N Taylor
3 - R C Hollingshead	10 - R E Van der Cook
4 - L L McGregor	11 - H E York
5 - T C Nelson	12 - Records Center
6 - J H Rector	13 - 300 Files
7 - A E Smith	14 - 20 Extra

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DECLASSIFIED

DECLASSIFIED

-2-

HW-78712

A PROGRAM FOR THE NUMERICAL CONTROL
OF A PULSE INCREMENT SYSTEM

I. INTRODUCTION

Purpose

This report will describe the important features of the development of magnetic tapes for the numerical control of a pulse-increment system consisting of a modified Gorton lathe and its associated control unit developed by L. E. Foley of Equipment Development Service, Engineering Services, General Electric Co., Schenectady, N. Y. Included is a description of CUPID (Control and Utilization of Pulse Increment Devices), a FORTRAN program for the design of these tapes on the IBM 7090 computer, and instructions for its operation.

Acknowledgements

The autor wishes to thank E. E. Sundell for his development of the FAP routine for the removal of record gaps upon which the success of this program has been so dependent and for his suggestions concerning its presentation in this report. R. Y. Dean has contributed a large portion of the mathematical analysis.

Prerequisites

Two previous documents by R. Y. Dean, HW-73537, entitled "A Theory of Control For δ -w Rotary Lathes", and HW-74903, entitled "Positioning By Cam On δ -w Rotary Lathes", supply a background for what follows. Familiarity with the content of these reports, in particular with the positioning and timing equations of HW-73537, their corresponding nomenclature, and the problems of cam control discussed in HW-74903 will be assumed in Section II. Further assumptions include the reader's acquaintance with the coordinate geometry of the Gorton lathe, its modes of contour cutting, and the modifications involved in its conversion to numerical control. Elementary

DECLASSIFIED 8712

knowledge of Fortran programming, memory word configuration, and input-output methods for IBM computers, and storage to tape transfer procedures for the IBM 7090 is also prerequisite to the ensuing material. Brief descriptions of some aspects of these features are given, but many details may be omitted: further information is available in IBM's Reference Manual, 709/7090 FORTRAN Programming System, and Reference Manual, 7090 Data Processing System.

Motivation

Originally intended to cut spherical or almost spherical contours under cam control, the Gorton lathe eventually was required to follow contours containing cylindrical and near cylindrical sections. Finding that the system could not reliably interrogate a cam corresponding to the latter type of contour,* it became evident that modifications would be necessary in the existing Gorton models if they were to have the flexibility demanded by the more recent geometrical designs.

To date, two different modified systems have been developed and tested. The first, which is now being used for part fabrication, is a converted x-y coordinate machine controlled by a template. The second, which is still in the prototype stage, is a numerically controlled device which retains δ - ω coordinates.

This latter system is the one with which this report is concerned. Should it prove feasible, several advantages over cam or template type control are obvious.

- (1) Increased flexibility in the variety of geometrical shapes that can be handled reliably.
- (2) Substantial savings in time and expenditures.

* See HW-74903, section entitled "Some Comments, page 7-8.

(3) Elimination of a problem in component synchronization.

The possibility for further cost reduction has motivated the development of a new concept in tape control rather than the mere acceptance and adaptation of a standard unit. The proposed system has such an advantage over present numerically-controlled devices in that the need for special punched tape preparation equipment as well as an electronic control unit known as the director is eliminated. The essential features of this new concept and its inherent advantages and disadvantages will be discussed more fully in later sections.

Numerical Control Principles

A numerically controlled system is one which performs its operations in discrete steps in response to a series of instructions. Such a device differs from a cam or template-controlled system in that a given machine motion under numerical control consists of a series of small rapidly performed movements rather than a single continuous motion. The numerically-controlled prototype system described here operates, with respect to its ram and table movements, upon this discrete step principle. These movements are each controlled by a pulsing motor which moves its respective coordinate a fixed unit distance each time it receives the proper signal.* By feeding these signals to the pulsing motors at a rapid rate, continuous motion of the type produced by a cam operated system can be simulated virtually as closely as desired.**

This means, however, that it is necessary to replace the cam or template in the old system by some other means of guidance in the new. This new

* At present, the table rotates 1 milli-degree per pulse while the ram moves 20 micro-inches per pulse.

** Practically, the coordinate motions probably are continuous due to the inertia of the system.

DECLASSIFIED W-78712

guidance method generally consists of a series of instructions stored on tape. The tape is run through an electronic reading and interpreting unit which transfers these instructions in the order read to the desired operating component, thereby causing the machine to perform in a given manner. The type of operation performed in response to each instruction depends upon its particular tape coded form and the way in which the machine is wired to respond. The sequence in which the instructions are ordered on the tape is identical to the sequence in which the desired operations are to be performed. In the prototype system, the rate at which they are performed depends upon the instruction spacing on the tape and the rate at which it passes through the electronic tape reader. For coordinate movements, no more than one pulse of a particular kind is called for by any single coded instruction.*

It now becomes evident how, under numerical control, the tape instructs the proposed system to machine a surface. Knowing the initial position, coordinate geometry, and required path of the cutter head, the pulse motor instructions are ordered so as to cause the cutter head to very accurately approximate that path and spaced so as to properly control the timing parameters when the tape is read at a predetermined fixed speed. Other instructions are spaced and ordered so as to cause certain desired auxiliary operations to be performed at the proper time. Such a tape coded sequence of instructions defines a particular machining operation.

The remainder of the material to be presented will deal with how an instruction tape for the prototype Pulse Increment Numerical System is developed and processed, how the FORTRAN program CUPID is coded so as to cause the computer to produce such a tape, and how the input data to CUPID must be assembled so as to completely define a given cutting operation.

* In other systems this is not true. One coded instruction may call for numerous pulses of a particular kind to be performed at a given rate.

Nomenclature

One of the problems in describing a process involving new concepts, procedures, or equipment is to establish a nomenclature sufficient to distinguish what is involved and brief enough so as not to weary or confuse the reader. Below is a list of phrases and abbreviations together with their meanings which will be repeated frequently in the remainder of this report. It is hoped that their choice and use will facilitate the discussion that follows:

Pulse Increment Numerical System (PINS) - The system which consists of the Gorton lathe modified for numerical control together with its electronic reading and interpreting unit and other auxiliary equipment.

Control and Utilization of Pulse Increment Devices (CUPID) - The FORTRAN program that produces the master tape and an instruction tape for PINS.

"master tape" - That tape which contains the coded instructions for a given operation properly sequenced but with blank spaces known as record gaps between blocks of information. It is a tape from which instruction tapes can be immediately processed.

"instruction tape" or "control tape" - A tape which contains the coded instructions (properly spaced and sequenced) that define a machining operation. It is ready for use immediately by PINS.

"scrunching" - The process that produces an instruction tape from the master tape. It consists of writing the coded information from the master onto the instruction tape while removing the record gaps.

"data interval" - The contour information contained between two adjacent data points.

"instruction array" or "instruction vector" - A sequenced list of tape or memory stored instructions defining a contour (or portion thereof) and the manner in which it is to be machined.

"record gap" - A variable length spacing between information blocks on a magnetic tape which occurs when the IBM 7090 writes information under its normal mode of operation.

II. MAGNETIC TAPE DEVELOPMENT AND PROCESSING

Tape Storage Principles

In order to understand fully the explanation of tape development and processing that follows and exactly how PINS is able to receive and interpret its instructions, it is first necessary to acquire some elementary knowledge of how information is accumulated in the computer and stored on magnetic tape.

The IBM 7090 computer stores its information on magnetic cores which are subdivided into units called words. Each word has 36 subdivisions called bits which are each assigned either a "0" or a "1". A word normally represents a binary number or a computer instruction but may also be used to represent a series of coded PINS instructions.

The magnetic tape has seven channels running length wise along the tape, six of which may contain bits whose assigned states are determined by the corresponding memory bit states.* When stored information is copied or "written" on the tape, the 36 data bits of each word are reproduced verbatim across the tape in the six usable channels on six adjacent lines which may be spaced 800/in., 556/in. or 200/in. - two high density and one low density mode. A data bit assigned a "1" in memory causes a small magnetized spot to be "written" in the corresponding tape location a "0" in a memory bit causes its respective tape location to be left blank. Each tape bit, therefore, is the analog of a two-state device, where a magnetized spot in the bit location produces a state "1" and the absence of such a spot the state "0".

* The seventh channel is a check-bit channel whose bit configuration is automatically determined by the bit configuration in the other six channels.

DECLASSIFIED

-8-

HW-78712

A line on the tape, containing six usable bits (corresponding to the six usable channels) may be used to store a coded instruction for PINS.* An instruction consists merely of one of 6^4 possible configurations of zero to six magnetized spots. An instruction tape is one which contains a series of these coded configurations, each corresponding to a specific PINS operation on an otherwise blank tape.

The tape itself may be conceptualized schematically as a grid six usable units wide and virtually unlimited**in length as shown in Fig. I. Each little square is a bit location that may be in one of two states. It is easy to see that the number of possible tape configurations and therefore the number of different operations

FIG. I - TAPE SCHEMATIC

definable by this method is for most practical purposes unlimited***. The PINS tape reader simply scans the tape, recognizes the state of each bit and transfers the bit configurations, line by line, to an interpretive system which renders the proper signals to the various operating components. More detailed information concerning computer storage, magnetic tape, and memory to tape transfer is available in IBM's Reference Manual, 7090 Data Processing System.

- * The auxiliary operation instructions actually occupy not only one but a number of tape lines.
- ** A normal tape reel contains 2400 ft. of tape. In low density this is the equivalent of 5,760,000 tape lines.
- *** If the instructions are too widely spaced it is possible to overrun the tape but not likely except for multiple cutting operations defined as one operation.

DECLASSIFIED

Theory of Control - A New Concept

Perhaps the most immediately noticable difference between PINS and standard numerical systems is that it uses magnetic tape rather than punched tape. In punched tape systems, the holes are so distributed along the tape that they form a series of coded instructions similar to those formed by the magnetized spots on magnetic tape. In this sense, the two tapes are analogous.

However, in another sense they are quite different. On standard systems, no attempt is made to control the timing variables by varying the distance between instructions. Instead they are spaced evenly along the tape but with the added feature that the desired rate of performance for each command is also included on the tape in coded form. One instruction may call for a number of executions of the same or different types. In order to handle them, it is necessary that they pass through a unit called a director, which in actuality is a small computer. This device interprets each instruction and adjusts the various performance rates to meet machine limitations by means of a feed-back mechanism which checks to see if the last command has been executed before a new one is released. This insures that commands will be executed either at the rate specified by the instruction or at the maximum allowable rate, whichever is slower. A director will have a memory which can store instructions until they are ready to be performed.

The added restriction of machine limitations adds a heretofore unmentioned variable to the problem. As previously stated, in the PINS prototype system the timing variables are controlled on the magnetic tape by the way the instructions are spaced. The coded forms are interpreted and signals sent to the pulsing motors at the same rate at which they are read by the tape reader. Since there is no feed-back and delay device for protection as in standard systems, a danger exists that a new signal will be received by the machine

DECLASSIFIED

-10-

HW-78712

before the prior one has been completely executed. If this happens, the new signal will not be identified and its execution will be deleted. Therefore, in order to guarantee the quality and reliability of system performance, the instructions must be spaced not only in a manner that properly controls the timing variables, but also in such a way that the various system components are able to respond. The unique feature of the PINS tape control process is just this new concept of continuous tape feed with complete control of the signal execution rate by the tape itself. It requires knowing the PINS response characteristics and including them as input to the FORTRAN program CUPID which produces the system tapes. CUPID then performs a function similar to that of the post-processor on standard systems by taking account of the system characteristics during the tape development process rather than afterwards. The need for a director is thus eliminated.

When response characteristics and timing variable control come into conflict, it is, of course, the latter that must be sacrificed. The limitations on the PINS pulsing motors can be expressed in terms of maximum pulsing speed and the rate of change of pulsing speed or acceleration. As long as these limits are not exceeded, the timing variable control is perfectly compatible with the system and may proceed unhindered. If a limiting value is exceeded, the timing variable curve, must be modified so that the system components continue to operate within specifications. As part of its operation, CUPID checks to see if any limits are exceeded and makes the necessary adjustments. Under the present set-up, it is assumed that if the system components cannot function as desired, the next best thing is to operate them at their limit points, since this seems to keep the timing variable curve closest to the optimum.

Because the coordinate movements are synchronized to control the cutter path, an adjustment in the pulsing rate of one stepping motor automatically means an adjustment in the rate of the other, even though its limits may not be exceeded. The mathematical basis for such an adjustment and the way in which it is incorporated into CUPID will be discussed in later sections.

Contour Description

If the Gorton lathe is to machine a given contour under numerical control, the required cutter head path must be translated onto the instruction tape in a discrete step, machine coordinate language (i.e., ram and table position or δ -w coordinates as defined in HW-73537). As a preliminary step to this operation, the cutter head path must itself be described in a way that makes such a translation possible. The discontinuous property of machine coordinate motion indicates that the cutter head contour should be defined in like manner by a series of closely spaced, discrete points. The aim to insure a smooth surface finish via control of the timing variables indicates the need for point by point data specifying these functions or their equivalents. Such equivalent data is provided in the form of the coordinate velocities desired at the points. As timing information it is then in its most immediately useful form since it allows direct control of instruction spacing. The procedure by which such a description is arrived at is as follows:

- (1) Given the mathematical description of a contour, a large number of closely spaced points on it are chosen to define its locus.
- (2) From these points and the other contour data associated with them, a similar discrete point description of the cutter head path is determined by the methods of HW-73537.
- (3) Knowing the cutting parameters, the coordinate velocities at these points are computed using the formulas given in Section III of this report.

- (4) The difference between the δ and ω coordinate values for each pair of neighboring points are computed and multiplied by an appropriate factor to give the number of pulses in each coordinate necessary to span the interval between each point pair.

Once this has been accomplished, the problem becomes one of how to instruct the cutter head, starting from its initial position, to move the required number of pulses in each interval so as to pass through (or sufficiently close to) each given point and thus approximate the given curve. It should be noted that the contour machined in this manner will not in general coincide exactly with the given curve for two reasons. The first is because the way in which the cutter head will move between given points will be determined more by the geometry of the δ - ω coordinate system than by the curve desired. It is therefore necessary to select small enough point intervals that the deviation from the given curve will be small with respect to contour tolerances. With reasonable care, this should not be a problem. The second reason is because the pulse differences required to move precisely from point to point are not in general integral numbers. However, if the magnitude of the ram and table movements are small compared with the given tolerances, as is now the case, no major difficulties should arise.

Master Tape Development

Given the cutter path data in the form described above, we now recall the specifications and restrictions given thus far on the tape instruction array.

- (1) The instructions are to be sequenced on the tape in the same order as they are to be performed.
- (2) They are to be spaced so as to properly control the timing variables without exceeding the system response limitations.

DECLASSIFIED

- (3) One tape instruction is to correspond to no more than one component operation of any one type. (It may refer to more than one operation provided they are different in kind. In the case of auxiliary functions a single operation may be specified by a series of instructions.)

With these in mind, a skeleton description of the development of a master tape on the IBM 7090 computer, in terms of the major steps involved, is given below. It will be elaborated upon in Section IV on programming.

- (1) The coordinate velocities are adjusted at each point to meet system response limitations (i.e., maximum pulse motor speeds and accelerations).
- (2) Using these corrected velocities, the length of tape (in number of lines) required for each data interval is computed using the methods of Section III.
- (3) The instructions required in each interval are distributed in an adjacent series of memory array locations equivalent in "length" to the computed length of the data interval. Instruction spacing within the locations is either uniform* so as to produce a constant velocity that is the average of the interval end-point velocities or varied so as to uniformly accelerate the pulsing motors from the initial interval velocity to its final velocity.
- (4) Starting with the first interval of a given curve, this process is repeated interval by interval, the memory locations corresponding to any data interval chosen within the given array so as to immediately follow those of the previous interval.
- (5) When the total number of memory locations in which instruction data is distributed exceeds a certain fixed size n , the contents of the first n locations of the array are written out on the tape. That portion of the array is then erased and the remaining instruction data is shifted so as to begin in the initial array location.
- (6) If the contour consists of several different curves joined together, the process proceeds curve by curve, interval by interval within each curve, until the whole contour description is completed and written out on the tape.

* Even a "uniform" spacing may vary somewhat from pulse to pulse, because the possible storage locations are finite and discrete, but will maintain a similar overall pattern.

DECLASSIFIED

-14-

HW-78712

A master tape is thus completed from which any number of "instruction tapes" or "control tapes" can be produced.

Control Tape Processing

In its normal mode of operation, the IBM 7090 writes information onto tape in blocks containing 255 words apiece separated by a blank space of variable length known as a "record gap". This and the fact that the machine has a memory capacity insufficient to hold all the instruction data we need to describe a normal machining operation causes a twofold problem to arise. Since the information written on the tape is gapped, we cannot, with a tape written under a normal mode, space the instruction data so as to properly control the timing variables. In fact, such a tape when read by PINS relays its information in such a jerky, discontinuous manner that the pulsing motors usually jam, making the system unable to respond. The tape writing mode can be changed to write a single continuous record, but only so long as it does not run out of data. Once this happens, the tape is bound to be gapped between any two pieces of information written in separate operations since a small amount of time (and therefore tape) is needed to decelerate and accelerate the tape to the proper speeds between distinct writing operations. But such a multiple writing process is demanded by the computer storage limitations unless a way is found to supply new data as the old is being written, thus retaining in memory a continuous backlog of information until the contour description is complete.

This problem is solved by making the production of a control tape a two stage process. The first stage is the production of a master tape where the instruction data is written out in the standard 255 word per record form. The second stage of record gap removal, which has been termed "scrunching", is one in which reading and writing take place simultaneously. The master tape

data is read into memory while it is being written on the tape. The writing process is delayed until a certain fixed sized record is read into memory; then the writing process begins. The process is synchronized in such a manner that by the time one record has been written on tape, another has already been read into memory, so that writing need never cease until the instruction data on the master tape is exhausted. The process can perhaps be best described by an analogy.

Consider a bucket with a hole in the bottom and a much larger barrel filled with water above it. Suppose we want to supply water to bucket from the barrel automatically by a discrete process that dumps a fixed quantity into the bucket each time the barrel is bumped. Furthermore, once the process is begun, we want water to flow continuously out of the bottom of the bucket without it ever overflowing or going dry until after the barrel is empty. Assuming the previous information and a constant flow out the bottom of the bucket, it is not difficult to establish an optimum fixed rate of barrel bumping.

This is just what is done in the scrunching process where the master tape plays the role of the barrel, a computer memory array the bucket, and the instruction data the water. The time intervals at which the fixed size information blocks are read off the master tape are constant, analogous to the barrel bumping rate, while the tape writing rate is continuous throughout.

The final result of the scrunching process is a control tape free of record gaps which is ready for use by PINS. The more technical details of this process will be discussed in Section IV.

System Comparisons

The main advantage of PINS over cam or template control and over standard numerical control systems were mentioned briefly above. Along side of some

DECLASSIFIED

-16-

HW-78712

of the less obvious benefits, they are restated and enlarged upon below. The disadvantages of the system are also discussed. In making comparisons between PINS and standard systems, the aim has been to generalize the important features common to the latter for the sake of an overall view. In comparison with any particular system, some of the following statements may not be precise or applicable. Also, between PINS and a given standard system many other specific comparisons could be drawn. Much of the information concerning the workings of standard systems was gained during two information exchange meetings with representatives of the Bendix Corp. and the General Electric Company.

In actuality, much of the material presented is somewhat hypothetical in nature as PINS has not yet been thoroughly tested nor compared, except in theory, with other systems. Should a comparative study ever be undertaken, it is hoped that the following outline will be of assistance in determining what prior questions need to be asked and in anticipating what to expect in the way of results.

I. Comparison With Cam or Template Control

A. Advantages

1. Increased flexibility in the number of geometrical shapes that can be handled reliably.

Under numerical control, limitations due to restrictions on cam or template contours and errors due to tracing a curve with a finite sized ball are eliminated. The lathe is able to very accurately describe all two-dimensional paths, with the possible exception of those which demand coordinate sign reversals that introduce significant backlash. The effect of these errors, as far as machining a surface is concerned, can often be controlled by a propitious choice of cutting parameters.

DECLASSIFIED

-17-

HW-78712

2. Substantial savings in time and expenditures

It is a much faster and cheaper process to produce a new tape than it is to fabricate a new cam or template. Once the preliminary contour analysis explained in HW-73537 (which is required for any control system) is completed, a tape can be produced on Hanford's IBM 7090 at a cost of approximately \$150 for the master tape and \$30 for each control tape.* At present, a cam or template must be ordered off-site at a considerably higher unit cost.**

Further savings can be realized with tapes in adapting to any part modifications that may be required. For instance, slight contour changes that might be necessary to compensate for machine idiosyncrasies can be handled with extreme ease on PINS by simply making the necessary changes in the input data to CUPID. Under cam or template control a completely new one must be fabricated, thus increasing both tooling costs and waiting time.

3. Elimination of a problem in component synchronization

Control of the lathe timing variables, as well as those of positioning, could be transferred to the tape, thus eliminating the problem of multiple cam coordination. Under the present system, any variation in spindle speed is still cam controlled, but a transfer of this function to the tape would involve, as far as CUPID is concerned, only a minor revision. ***

* Based on the present \$450 per hour cost of 7090 computer time.

** As of September, 1962, Harold York estimated the current unit cost for a model 1251 template at \$500.

*** The additional equipment costs will likely be substantial, however.

DECLASSIFIED

-18-

HW-78712

B. Disadvantages

1. Necessity to allow for the limitations of the pulsing motors.

Some types of contours demand, in certain regions, rapid changes in the relative movements of either or both coordinates. This may mean that, under discrete step control, the desired feed rates cannot be maintained in these regions because the acceleration or speeds demanded of the pulsing motors exceed limitations. Under cam or template control these restrictions do not exist. However, in many cases studied thus far, the regions where the pulse motors have the most difficulty keeping up are just those where a cam or template is difficult to follow. It is a question, therefore, whether continuous control devices have any inherent advantage at this point or if one difficulty is merely being exchanged for another more serious one. In any case, for the types of contours anticipated, the limitation should not be a serious one.

2. Lack of visual "readability"

With a knowledge of a particular machine and an experienced eye, it is possible to deduce from the shape of a cam the approximate shape it is intended to cut. A template, of course, is an exact replica, however, a tape is not so easily "readable". With a knowledge of the tape code, one could perhaps deduce the general shape of the curve a punched tape will cut, but the process would be long and painstaking. With a magnetic tape, such as the one used on PINS, it is impossible to obtain any information by visual inspection since the magnetized spots that make up the instructions are invisible. A magnetic tape must, therefore, be

DECLASSIFIED

-19-

HW-78712

given a closely watched dry run test on PINS before it is actually used to machine a piece in order to make sure that the contour desired is at least approximately described. This is best checked by a plotter connected to the lathe, although no δ - w coordinate equipment exists at this time.*

Lack of readability is essentially a psychological disadvantage, not a technical one. Once the operators become accustomed to using such enigmas, any difficulties involved should disappear.

II. Comparison With Punched Tape Control

A. Punched Tape Control Principles

While both the standard punched tape systems and our magnetic tape systems use similar media of information storage, there are, beyond the method of recording data, some fundamental operational differences between the two. The following is a brief explanation of the basic operating principles behind standard systems in terms of a comparison with PINS as developed here at Hanford.

1. The standard systems read and perform their commands in "blocks" rather than as single line quantities. This means that information contained on a number of tape lines will be read as one instruction. This instruction may call for the execution of numerous pulses of a particular kind rather than being limited to one as in PINS. A separate code denotes the beginning and end of each information block.
2. The tape reading procedure for standard systems is a series of stops and starts that may give the appearance of jerky discontinuous motion as opposed to the constant speed reading of PINS. This is

* An x-y plotter will do if a mapping of the contour from δ - w geometry to cartesian geometry is expedient.

DECLASSIFIED

-20-

HW-78712

because the instructions for standard systems must require a longer execution time than reading time. If such is not the case, the lathe may dwell on the work piece. In order to make sure new information is always available a new instruction is read and stored in a memory location while its predecessor is being executed. A second limitation which arises under this method is concerned with the relative size of two adjacent instruction blocks since it is possible under extreme conditions for the machine to execute a block of information faster than its successor can be read.

3. The rates of performance and acceleration are controlled by a "director"* in accordance with given or measured machine response characteristics rather than by tape spacing. Each instruction block contains a feed - rate number which the director uses to determine the rates at which its information will be executed. It is computed from a desired representative feedrate for the interval, assuming that the travel between the end points is along a chord joining the points.
4. The path description in x-y coordinates amounts to a series of straight lines drawn between closely spaced points, the travel rate along each segment being constant. The director contains an internal clock which is capable, in a given interval, only of putting out signals at constant rates. At most, different types of signals may be assigned different rates; the rate of any one train may not be varied in the interval. In δ - ω geometry the same restrictions would hold except that a combination of straight line and rotary motion will result.

* An electronic black-box, also known as an interpolater, which interprets the tape commands and puts out the desired signals to the various machine components.

B. Advantages of Continuous Feed Control

1. Cost reduction

Included in the cost of standard systems are the director and special punched-tape preparation equipment. Both of these are eliminated with PINS since the tape itself has all the required computations built in and is directly usable as it comes off the IBM 7090.

2. Increased accuracy in contour definition.

PINS may follow a second degree curve between two given points in either or both coordinates, a path which is inherently more accurate than chord approximation. To obtain a further advantage in accuracy, the PINS interval sizes may be made virtually as small as desired while the length of punched tape required in the standard block instruction systems, if the interval size becomes too small, may be unmanageable. As far as present accuracy demands or measuring capabilities are concerned, there would probably be essentially no difference between parts cut on the same machine using either system of control. However, if a technological break-through should occur which allows a significant decrease in machining tolerances, PINS appears to have a decided edge in being capable of adjusting to the new demands that would arise.

3. Tape reliability

Because paper tape tends to wear out quickly under repeated use, most users of punched tape systems have switched to a "milar" tape for actual production line use. But many have had difficulty in obtaining milar tape that is free from flaws and splices that

could hinder the accurate transfer of information. Magnetic tape has been developed for use on high speed digital computing systems and as such has a high degree of reliability that should be adequate for virtually any numerical control operation. Any flaw that could cause the tape to be misread is almost certain to be detected in the 7090 bit check-sum process.*

4. More accurate control of timing variables

In punched tape systems, each block of information entering the director contains a feed-rate number which, together with other pre-set control apparatus, determines a relationship between the timing variables that will remain constant for the interval. In PINS, instruction spacing on the tape is such that the feed-rate may vary continuously as demanded by the shape of the contour and the cutting parameters, thus providing more accurate speed control. Where small interval sizes are chosen, the differences between PINS and a standard system are probably not noticeable at present, but here again, should super-sensitive control of surface smoothness or surface cutting temperature be demanded, such control could become important.

C. Disadvantages

1. Lack of suitable feed-back control on PINS i.e. a check to see if the system has actually performed the number of incremental movements requested. Standard feedback systems interrogate a distance measuring device on the lathe that attempts to compensate for any variation that may occur in increment size during the cutting

* This is an internal checking process which keeps track of whether the sum of the magnetic spots (binary "one's") in each channel and also across each line is odd or even.

DECLASSIFIED

-23-

HW-78712

process. If the error in total movement ever becomes too large, the process is terminated. Because of the difficulty a mechanical system has in maintaining a constant incremental movement over comparatively large travel distances plus the added safety feature of automatic shutdown, this latter type system is considered most desirable and in fact essential by lathe manufacturers. While PINS does not at present have a feed back system capable of interrogating a distance measuring device, it has been estimated that one could be added to a nominal cost.

2. Practical limitations on contour size

With each instruction carrying a minimal amount of information and being spatially timed, it is possible to envisage under PINS a series of lathe maneuvers that would demand more tape than is available on a standard 2400 ft. reel. Block information systems have the advantage of using considerably less tape for most applications, but this does not become important except where tape length becomes a problem. This can only be determined when the maximum contour size anticipated is known. A method for computing the approximate amount of tape required for a particular operation is given later in this report and should be applied to an anticipated maximum travel operation if the use of PINS is being considered. Present applications use only about half a reel or less where the tape reading speed is 20 inches per second. Because one instruction can carry so much more information and no spacing is necessary, block information tapes have no practical contour size limitations.

DECLASSIFIED

-24-

HW-78712

3. Lack of visual readability

Punched tape can also be "read" and the contour perhaps perceived by an experienced eye. For the applications now envisioned, however, this would be impractical except in a very localized area. The discussion on the merits of readability for cams versus magnetic tape also applies here.

System Limitations

As in every physical system, PINS has certain limitations which prevent it from being all that one would like it to be. These are essentially of two types:

(1) Those which could be removed by added sophistication in the system or in the associated computer program CUPID. However, the modifications required do not at this time appear justified in terms of the present development of the system and the additional effort and costs involved.

(2) Those which are inherent in the system.

In the first category are the lack of a suitable feedback control system which has already been discussed and several features of CUPID which will be discussed in the section on programming. The second category contains several items that are worthy of mention here.

First of all there is the limitation on tape length which has already been somewhat discussed. If it ever becomes a problem, two alternatives are available. More than one tape may be used for a given cut, which is feasible (through rather awkward) if the shape of the contour or the accuracy of the lathe is such that the cutting edge can leave and return to the workpiece without scoring or if a smooth surface finish is not essential. Alternately or in conjunction with this, the information on the path may be packed more

DECLASSIFIED

-25-

HW-78712

closely* and the tape reader run at a slower speed ** to compensate for it.

But this can give rise to another problem -- that of representing rapid speeds and rates of change on tapes where instruction packing is dense. We know that instructions can occupy only discrete locations on the tape spaced 200 per inch. Suppose then that we want one of the pulsing motors to run at 900 pulses per second while we read the instruction tape at a rate of 5 inches per second. The number of tape "lines" passing the reading head per second will be

$$200 \text{ lines/in.} \times 5 \text{ in./sec.} = 1000 \text{ lines/sec.}$$

The required spacing between tape instructions then becomes

$$1000 \text{ lines/sec} \div 900 \text{ inst./sec.} = 10/9 \text{ lines/instruction.}$$

Since we can only place the instructions in discrete locations, we must, in order to achieve the desired average spacing place 9 instructions on adjacent lines, leave one line blank, and repeat the process. But this means that in a very localized area (i.e., at the end of every ninth instruction) the rate at which instructions are read almost instantaneously drops in half and then jumps back up again. Changes so rapid as this are intolerable over any appreciable length of tape; over very localized areas, such as in the example above, the effects are uncertain. Indications gathered from test tapes where pulse spacing has been irregular are that such patterns do not yield the most desirable operating characteristics.

The limitations on pulse motor speed and acceleration can be minimized by obtaining the best hardware available but will always be present to some extent. The significance of such limitations is dependent upon the shape of the contour to be cut and the demands on the surface finish.

* Care must be taken not to pack the instructions so densely that the maximum desired pulse motor speeds are unobtainable.

** At speeds that are too slow, the reliability of the reader is questionable.

DECLASSIFIED

-26-

HW-78712

Finally, we come to the question that faces all numerical systems, namely what is the actual shape of a contour cut by a discrete step process? And what accuracy can we expect from such a method?

In δ - ω geometry, if each instruction were performed instantaneously upon being read, we would expect the contour, viewed through a microscope, to appear as a jagged series of ridges made up of circular arcs and straight lines. But because of the inertia of the lathe components and the rapidity with which signals are usually received, the actual coordinate motions are probably fairly continuous in time since a new command is usually received before the old one has been completely executed.* This means that the contour itself will be much smoother than one would otherwise anticipate. Its smoothness on a microscopic level is probably not important so long as step sizes in each coordinate are small enough to produce a good macroscopic finish.

In determining optimum step sizes it is most important that one take into consideration the minimum part tolerances that will be required. This becomes evident when we realize that we can guarantee no closer an approximation to a given coordinate of a point in space than one-half the step size in that coordinate. The optimum step size for any process will depend upon the desired accuracies and cutting times and will have a lower bound determined by the maximum pulse motor speed.

* This is particularly true if the pulsing motors drive a hydraulic system.

III. MATHEMATICAL ANALYSIS

Purpose and Motivation

The aim of this section is to develop a mathematical basis for what has been previously stated and for what is to follow. The presentation of this material has been saved until now because it was judged most helpful to the average reader to have a more general, qualitative description of the system before attempting to understand it from an analytical viewpoint. For those who are content with merely a brief, overall picture, the following sections can be omitted without any serious loss of knowledge at this level. In fact some confusion may even be avoided. For those whose aim it is to have a detailed understanding that would eventually enable them to use CUPID to turn out control tapes for PINS, the ensuing information is essential.

Contour Completion

Instead of the usual mathematical description, a contour or portion of a contour may be defined by a series of points in the plane along with derivatives at the end points. In order to estimate the normals to a surface thusly describe, which are needed for defining the cutter head path and to have more finely divided point intervals, it is necessary to fit the original points and their slopes. The details of this process are given in HW-73467, "A Contour Completion Algorithm", by R. Y. Dean and consist essentially of fitting a unique cubic between each pair of points in such a manner that the first and second derivatives are continuous at the points. The advantages of this method over other procedures include ease in computation and accuracy of fit for point sets taken off known curves.* Its major justification lies in the fact that it is a close approximation to the curve that would be described by a thin spline bent so as to pass through the given points and

* This is no guarantee that it will be more accurate for other curves one may want to fit. There is no absolute concept of a "best" fit except relative to a particular curve whose general form is already known.

have the given end slopes, provided its flexing is not too great.

Assuming that the contour is given in (r, θ) coordinates in terms of the $n+1$ points (r_i, θ_i) $i=1, 2, 3, \dots, n+1$,

$$(1) \quad p_{i+1} r''_i + 2(p_i + p_{i+1}) r'_{i+1} + p_i r'_{i+2} = \frac{3p_i q_{i+1}}{p_{i+1}} + \frac{3p_{i+1} q_i}{p_i}$$

$$i = 1, 2, \dots, n-1 \text{ where } p_i = \theta_{i+1} - \theta_i \text{ and } q_i = r_{i+1} - r_i$$

gives $n-1$ equations * which together with the two given end derivatives r''_1 and r'_{n+1} gives an $n+1$ order simultaneous linear system which is easily solved for r''_i , $i = 1, 2, \dots, n+1$. A cubic is then uniquely defined in each interval in terms of four things - its end point values and their derivatives.

Contour Definition

With the above information given, we may now derive an analytic curve corresponding to each interval of a point defined contour. Taken together, these curves completely define the contour. Referring again to HW-73467 and the above notation, the i^{th} interval representation of the contour becomes

$$(2) \quad r = r_i + r'_i (\theta - \theta_i) + A_i (\theta - \theta_i)^2 + B_i (\theta - \theta_i)^3 \quad \theta_i \leq \theta \leq \theta_{i+1}$$

where

$$(2a) \quad A_i = 1/p_i^4 \{ 3p_i^2 (q_i - r'_i p_i) - p_i^3 (r'_{i+1} - r'_i) \}$$

$$(2b) \quad B_i = 1/p_i^4 \{ p_i^2 (r'_{i+1} - r'_i) - 2p_i (q_i - r'_i p_i) \}$$

Any intermediate point on the curve may now be found by properly choosing a θ and computing an r using (2). Given analytic curves may be completely defined by a known mathematical expression and hence need no contour completion. CUPID follows this procedure for straight line segments and circular arcs which are defined in the following manner:

* For a derivation of (1), see HW-73467.

Straight line segment - in terms of the length (ρ) and angle of rotation (ϕ) of its normal drawn from the origin and the angles of rotation (ϕ_1 and ϕ_2) of the lines drawn from the origin to its end points, four parameters in all as shown in Fig. 2A. The expressions for r , r' , and ψ become

$$(3a) \quad r = \rho \sec (\phi - \theta)$$

$$(3b) \quad r' = -\rho \sec (\phi - \theta) \tan (\phi - \theta) = -r \tan (\phi - \theta)$$

$$(3c) \quad \psi = \pi/2 + \phi - \theta$$

Circular Arc - In terms of the length (ρ) and angle of rotation (ϕ) of the line drawn from the origin to its center, its radius (c), and the angles of rotation (n_1 , n_2) of the lines drawn from its center to the end points of the arc, five parameters in all as shown in Fig. 2B. The pertinent relationships here are

$$(4a) \quad r = \sqrt{\rho^2 + c^2 + 2\rho c \cos (\phi - n)}$$

$$(4b) \quad \theta = \tan^{-1} \left(\frac{\rho \sin \phi + c \sin n}{\rho \cos \phi + c \cos n} \right)$$

$$(4c) \quad r' = r \tan (\theta - n)$$

$$(4d) \quad \psi = \pi/2 - (\theta - n)$$

where $n_1 \leq n \leq n_2$

FIGURE 2A

Straight Line Segment

FIGURE 2B

Circular Arc

Cutter Path Definition

Having analytic expressions of the types mentioned above to describe an entire contour still leaves the problem of finding the cutter head path necessary to cut such a shape. More specifically, given an analytical curve in (r, θ) coordinates which represents a contour to be cut, we want a closely spaced set of (δ, ω) coordinate points that define the position of the cutter center which just bring the head into tangential contact with the surface. Underlying the derivation of these relationships and their use in formulating operational instructions are two basic assumptions:

1. The cutter head profile in the plane of the contour is a circle in those areas where it comes into contact with the cutting surface. Its center is defined to be the center of this circle.
2. Each incremental movement of the ram or table changes the δ - ω position of the cutter head by the prescribed amount in the coordinate specified, leaving the other coordinate unchanged. The cutter head path is assumed to be the path of its center.

Knowing this together with all else that has previously been discussed and the cutter head radius 'a', we restate the tool center path equations as derived by R. Y. Dean in HW-73537,

$$(5a) \quad b - \sin \omega = r \sin \theta \mp a \cos (\theta + \psi)$$

$$(5b) \quad \delta = r \cos \theta - b \cos \omega \pm a \sin (\theta + \psi) *$$

$$(5c) \quad \psi = \tan^{-1} r/r' .$$

The optional signs denote an inside or outside surface is being cut. They may be reduced to the upper sign alone if we choose to call the tool radius negative when an inside contour is desired.

* This equation is opposite in sign from that given by Dean for the inside contour case. Compensation is made for this in programming as we shall show later.

For the special cases previously investigated, substituting (3a), (3b) and (4a), (4b) into (5a), (5b) respectively gives

$$(6a) \quad b \sin \omega = \rho \sec (\phi - \theta) \sin \theta + a \sin \phi$$

$$(6b) \quad \delta = \rho \sec (\phi - \theta) \cos \theta - b \cos \omega + a \cos \phi \text{ for the line segment and}$$

$$(7a) \quad b \sin \omega = \rho \sin \phi + (c+a) \sin n$$

$$(7b) \quad \delta = \rho \cos \phi + (c+a) \cos n - b \cos \omega \text{ for the circular arc.}$$

The use of these formulas for computing useful numerical values and the conventions included therein will be discussed in the section on programming.

Coordinate Velocity Control

Since we are interested not only in the geometry of a contour, but also the surface finish of the piece, it is important that we control certain "machining parameters" that are related to the cutting process. These are spindle speed, tool surface speed, and what is called feed rate or chip size. This control is achieved in part by regulating the rates $\dot{\delta}$ and $\dot{\omega}$ at which the ram and table move and therefore ultimately by regulating the speeds of the pulsing motors. The relationship between these coordinate velocities and the machining parameters will now be derived, using the relationships that hold between the latter under two different lathe operating modes as given by Dean.*

Assuming that at any time t during the machining operation

$$\begin{aligned} \delta &= g(t) \\ \omega &= h(t) \end{aligned}$$

where g and h are everywhere continuous, we differentiate (5a) and (5b) maintaining only the upper of the dual signs

$$(8a) \quad b \cos \omega \dot{\omega} = \{r' \sin \theta + r \cos \theta + a \sin (\theta + \psi) (1 + \psi')\} \dot{\theta}$$

$$(8b) \quad \dot{\delta} = \{r' \cos \theta - r \sin \theta + b \sin \omega \dot{\omega} + a \cos (\theta + \psi) (1 + \psi')\} \dot{\theta}$$

From (5c) we note that

* See HW-73537, p. 9-12

$$\sin \psi = \frac{r}{\sqrt{r^2 + r'^2}}$$

$$\cos \psi = \frac{r'}{\sqrt{r^2 + r'^2}}$$

Using these we obtain from (8a) and (8b)

$$\begin{aligned} b \cos \omega &= \sqrt{r^2 + r'^2} \dot{\theta} \left\{ \cos \psi \sin \theta + \sin \psi \cos \theta + a \frac{1 + \psi'}{\sqrt{r^2 + r'^2}} \sin (\theta + \psi) \right\} \\ &= \sqrt{r^2 + r'^2} \dot{\theta} \sin (\theta + \psi) \left[1 + a \frac{(1 + \psi')}{\sqrt{r^2 + r'^2}} \right] \\ &= \sqrt{r^2 + r'^2} \dot{\theta} \left\{ \cos \psi \cos \theta - \sin \psi \sin \theta + a \frac{1 + \psi'}{\sqrt{r^2 + r'^2}} \cos (\theta + \psi) \right\} \\ &\quad + b \sin \omega \cdot \dot{\omega} \\ &= \sqrt{r^2 + r'^2} \dot{\theta} \cos (\theta + \psi) \left[1 + a \frac{(1 + \psi')}{\sqrt{r^2 + r'^2}} \right] + b \sin \omega \cdot \dot{\omega}. \end{aligned}$$

Now, substituting for $\dot{\omega}$ in (8b) and recognizing the $\frac{(1 + \psi')}{\sqrt{r^2 + r'^2}} = K$, where K

is the curvature of the contour at the point in question our coordinate timing equations become:

$$(9a) \quad \dot{\omega} = \frac{\sqrt{r^2 + r'^2} \dot{\theta} (1 + aK) \sin (\theta + \psi)}{b \cos \omega}$$

$$\begin{aligned} (9b) \quad \dot{\theta} &= \sqrt{r^2 + r'^2} \dot{\theta} (1 + aK) \cos (\theta + \psi) + b \sin \omega \frac{\sqrt{r^2 + r'^2} \dot{\theta} (1 + aK) \sin (\theta + \psi)}{b \cos \omega} \\ &= \frac{\sqrt{r^2 + r'^2} \dot{\theta} (1 + aK)}{\cos \omega} \{ \cos (\theta + \psi) \cos \omega + \sin (\theta + \psi) \sin \omega \} \\ &= \frac{\sqrt{r^2 + r'^2} \dot{\theta} (1 + aK) \cos (\theta + \psi - \omega)}{\cos \omega} \end{aligned}$$

These are applicable at all points except those where $\omega = \pi/2$, where $\dot{\omega}$ and $\dot{\theta}$ are now undefined. In order to express the coordinate velocities at this point we must first assume that $\theta + \psi \rightarrow \pi$ as $\omega \rightarrow \pi/2$. Then we find that

$$(10) \quad \lim_{\omega \rightarrow \pi/2} \left[\frac{\sin (\theta + \psi)}{\cos \omega} \right]^2 = \lim_{\omega \rightarrow \pi/2} \left[\frac{\sin (\theta + \psi) \cdot (1 + \psi')}{\cos \omega} \frac{d\theta}{d\omega} \right]$$

DECLASSIFIED

-33-

HW-78712

by L'hospital's rule. Using the identity $\frac{d\theta}{d\omega} = \frac{\dot{\theta}}{\dot{\omega}}$ and substituting (7a) into (10) we finally obtain

$$\begin{aligned} \lim_{\omega \rightarrow \pi/2} \left[\frac{\sin(\theta+\psi)}{\cos \omega} \right]^2 &= \lim_{\omega \rightarrow \pi/2} \left[\frac{\sin(\theta+\psi)(1+\psi') \dot{\theta}}{\cos \omega \frac{\sqrt{r^2+r'^2} \dot{\theta} (1+aK) \sin(\theta+\psi)}{b \cos \omega}} \right] \\ &= \frac{b(1+\psi')}{\sqrt{r^2+r'^2} (1+aK)} = \frac{bK}{1+aK} \end{aligned}$$

from which

$$\lim_{\omega \rightarrow \pi/2} \left[\frac{\sin(\theta+\psi)}{\cos \omega} \right] = \sqrt{\frac{bK}{1+aK}}$$

and

$$\begin{aligned} \lim_{\omega \rightarrow \pi/2} \frac{\cos(\theta+\psi-\omega)}{\cos \omega} &= \lim_{\omega \rightarrow \pi/2} \left[\cos(\theta+\psi) + \sin \omega \left\{ \frac{\sin(\theta+\psi)}{\cos \omega} \right\} \right] \\ &= -1 + \sqrt{\frac{bK}{1+aK}} \end{aligned}$$

These together with the relationship

$$(10a) \quad \dot{\theta} = \pm \frac{k_c}{\sqrt{r^2+r'^2}} \dot{\alpha} \quad \text{for constant chip size* where } k_c \text{ is the chip size}$$

allow us to express $\dot{\omega}$ and $\dot{\delta}$ in the forms

$$(11) \quad \dot{\omega} = \begin{cases} \pm k_c \dot{\alpha} \frac{\sqrt{K(1+aK)}}{b}, & \omega = \pi/2 \\ \pm \frac{(1+aK)k_c \dot{\alpha}}{b} \frac{\sin(\theta+\psi)}{\cos \omega}, & \text{otherwise} \end{cases}$$

$$(12) \quad \dot{\delta} = \begin{cases} \pm k_c \dot{\alpha} \{ \sqrt{bK(1+aK)} - (1+aK) \}, & \omega = \pi/2 \\ \pm (1+aK)k_c \dot{\alpha} \frac{\cos(\theta+\psi-\omega)}{\cos \omega}, & \text{otherwise} \end{cases}$$

* This is developed by Dean in HW-73537. The sign of the expression on the right hand side is determined by the direction of (ccw = +, cw = -).

where $\ddot{a}$ = $\begin{cases} \sigma & = \text{a constant for the constant spindle speed condition} \\ \frac{v}{\sqrt{k_c^2 + r^2 \sin^2 \theta}} & \text{for the constant tool surface speed condition} \end{cases}$ where v is the surface speed.

For the line segment, $K = Q$, and we may write

$$(13) \quad \ddot{\omega} = \begin{cases} 0, & \omega = \pi/2 \text{ or } \theta = \pi/2 \\ + \frac{k_c \ddot{a}}{b} \frac{\cos \phi}{\cos \omega}, & \text{otherwise} \end{cases}$$

$$(14) \quad \ddot{\delta} = \begin{cases} \mp k_c \ddot{a}, & \omega = \pi/2 \text{ or } \phi = \pi/2 \\ + \frac{k_c \ddot{a}}{\cos \omega} \frac{\sin (\phi - \omega)}{\cos \omega}, & \text{otherwise} \end{cases}$$

For the circular arc, $K = 1/c$ which gives

$$(15) \quad \ddot{\omega} = \begin{cases} + \frac{k_c \ddot{a}}{c} \sqrt{\frac{c+a}{b}}, & \omega = \pi/2 \\ + \frac{(c+a) k_c \ddot{a}}{bc} \frac{\cos n}{\cos \omega}, & \text{otherwise} \end{cases}$$

$$(15a) \quad \ddot{\delta} = \begin{cases} + \frac{k_c \ddot{a}}{c} \{ \sqrt{b(c+a)} - (c+a) \}, & \omega = \pi/2 \\ + \frac{k_c \ddot{a}}{c} \frac{\sin (n - \omega)}{\cos \omega}, & \text{otherwise} \end{cases}$$

Pulse Motor Limitation Adjustments

Previous discussion has revealed that, under certain conditions, equations (11) - (15) cannot be obeyed due to physical limitations on the speed and acceleration at which the pulsing motors will operate reliably. The best that can be done in such a situation is to run the pulsing motors at their operating limits or in such a manner as to optimize the time required to return to desired conditions. To illustrate what is meant, consider the curve of Fig. III to be the desired velocity of a pulse motor as a function of time.

** See HW-23537

DECLASSIFIED

-35-

HW-78712

FIGURE III

If $v_{\max}$ is the greatest allowable speed of the motor, we see immediately that between points A and B we cannot follow the desired curve. In such a case, we assume the next best* condition is to let the pulsing speed equal $v_{\max}$ in this region as represented by the horizontal dashed line. Another example of where the limiting conditions demand an adjustment of the velocity curve occur in intervals such as that between B and C. Here, though the desired velocity is always within limits, the average acceleration demanded is too high and so the pulse motor must be content with accelerating at what is initially a slower absolute rate in hopes of having its speed match that called for by the curve again at some later point. A third possibility is that the desired velocity build up to a local maximum or minimum rather rapidly and return as illustrated in the region DE. When the acceleration limit is exceeded in such a region it has been found desirable to replace part of the peak with a more level section rather than following the general contour outline with a rapid acceleration followed by a rapid deceleration.

* No precise definition of "best" or criterion for determining it are implied here or elsewhere in this discussion. In this particular case it simply means that the pulse motor operates as closely to the desired speed as possible.

DECLASSIFIED

-36-

HW-78712

In all the examples above, the dashed lines represent the new velocity curve the motor would follow when the desired curve makes impossible or impractical demands. Otherwise the motor follows a curve consisting of a series of short, connected straight lines that coincide with the desired curve at * those points where the velocity data has been actually computed. When no limitations are exceeded, this operating curve may be considered to coincide with the desired curve for all practical purposes. Otherwise it will follow a path similar to that described by one of the dashed lines depending upon which restriction has been violated.

The above considerations become more complicated when we consider both pulse motors operating simultaneously, since now we must take into account not only the limitations on each separately, but also the effect which changes in one pulse train have upon the other due to the relationship between coordinate motions demanded by the shape of the workpiece. Of primary importance is establishing a criterion for deciding which motor to adjust to operating limits when demands on both exceed limitations and how to handle the occasional condition where in a particular interval the restrictions on both motors cannot simultaneously be satisfied unless the previous operating pattern is somehow modified.

In order to see how these situations can occur we develop a set of relations that the two pulse motors must satisfy at any point from the elementary equations of motion.

$$(16) \quad v_1 - v_0 = at$$

$$(17) \quad v_1^2 - v_0^2 = 2ap$$

where a is the constant acceleration
 p is the distance traveled
 v_0 is the initial velocity
 v_1 is the final velocity
 t is elapsed time.

Applying (16) to a pulse interval where p is the number of pulses in the interval and t is the time in number of tape lines,** we write

* This is because of the linear acceleration characteristics of the pulse spacing on the tape.

** Since each pulse produces a fixed motion and tape reader speeds are constant these quantities are easily related to the usual units of time and distance.

$$(18) \quad \frac{v_1 - v_0}{\bar{v}_1 - \bar{v}_0} = \frac{a}{\bar{a}}$$

where the barred quantities always refer to one of the coordinates and the unbarred symbols to the other. This relation follows immediately if we remember that the time required to execute the pulses in one coordinate of a given interval must be the same as that required for the pulses of the other coordinate in that interval. In other words $t = \bar{t}$.

From (17)

$$(19) \quad \begin{cases} v_1^2 - v_0^2 = 2 ap \\ \bar{v}_1^2 - \bar{v}_0^2 = 2 \bar{a}\bar{p} \end{cases}$$

and by division

$$\frac{(v_1 - v_0)(v_1 + v_0)}{(\bar{v}_1 - \bar{v}_0)(\bar{v}_1 + \bar{v}_0)} = \frac{ap}{\bar{a}\bar{p}}$$

which we see from (18) reduces to

$$(20) \quad \frac{v_1 + v_0}{\bar{v}_1 + \bar{v}_0} = p/\bar{p}.$$

Now if in a given interval one of the end point velocities (say v_1) exceeds in absolute value the maximum allowable, it is reduced to that maximum while retaining its proper sign. The corresponding velocity in the other train (in this case $\bar{v}_1$) is then adjusted according to (20) and itself checked to be certain it does not exceed the maximum. If so, the adjustment routine is repeated with v_1 and $\bar{v}_1$ reversing their roles. Two such applications of this process are sufficient to guarantee that all restrictions will be obeyed if the pulse train checked first for maximum velocity is that one with the largest pulse count in the interval. More precisely, assuming $|p| > |\bar{p}|$

$$(20a) \quad \begin{cases} v_1 = \text{sign}(v_1) v_{\max} \\ \bar{v}_1 = \bar{p}/p (v_0 + v_1) - \bar{v}_0 \text{ from (20)} \end{cases}$$

DECLASSIFIED

-38-

HW-78712

if $|v_1| > v_{\max}$, leaving v_1 and $\bar{v}_1$ unchanged otherwise. Then we repeat the procedure with the roles of v_1 and $\bar{v}_1$ interchanged.

The checking process for maximum acceleration involves a more complicated procedure. The governing relationships are derived from substituting (20) into (18) giving

$$\begin{aligned} (21) \quad \bar{a}/a &= \frac{\bar{p}/p (v_0 + v_1) - 2 \bar{v}_0}{v_1 - v_0} = \bar{p}/p + 2 \frac{(\bar{p}/p v_0 - \bar{v}_0)}{v_1 - v_0} \\ &= \bar{p}/p + \frac{(\bar{p}/p v_0 - \bar{v}_0)(v_0 + v_1)}{ap} \end{aligned}$$

From which

$$(22) \quad \bar{a} = \bar{p}/p \ a + \frac{(\bar{p}/p v_0 - \bar{v}_0)(v_0 + v_1)}{p}$$

Letting 'a' be that machine coordinate acceleration whose absolute value is highest, if $|a| > a_{\max}$ we then set

$$a = \text{sign}(a) a_{\max}$$

and compute a new $\bar{a}$ from (22).

If $|a| < a_{\max}$, a and $\bar{a}$ are left unchanged.

If the former action was required we now repeat the procedure (as with v_1 and $\bar{v}_1$) with the roles of $\bar{a}$ and a interchanged. The pertinent equation then becomes

$$(22a) \quad a = p/\bar{p} \ \bar{a} + \frac{(p/\bar{p} \ \bar{v}_0 - v_0)(\bar{v}_0 + \bar{v}_1)}{\bar{p}}$$

Unlike the velocity adjustment, however, it is possible that after two applications of the adjustment procedure "a" will still exceed its limit. In order to see how this is possible consider (16) rewritten in the form

$$|at| = |v_1 - v_0|$$

and

$$|\bar{a}\bar{t}| = |\bar{a}t| = |\bar{v}_1 - \bar{v}_0| = |\bar{p}/p \ v_1 + (\bar{p}/p \ v_0 - 2\bar{v}_0)|$$

from (17) and the fact that $\bar{t} = t$.

DECLASSIFIED

FIGURE IV

Comparing the products $|at|$ and $|\bar{a}t|$ as functions of v_1 in Figure IV above we see how, since they are linear with different slopes and zero points, for different ranges of v_1 first one and then the other will be dominant.

In our particular example notice that

$$|a| > |\bar{a}|, \quad v_1 < V_1$$

$$|a| < |\bar{a}| \quad v < v_1 < V_2$$

and again that

$$|a| > |\bar{a}| \text{ for } v_1 > V_2$$

Now suppose that we impose a maximum value on $|a|$ and $|\bar{a}|$ as given by a_1 as shown. For any a or $\bar{a}$ outside this limit we find that $|a| > |\bar{a}|$ and so we need merely adjust v_1 so that $|a| = a_1$ and $\bar{a}$ will automatically fall within limits. For the lower maximum a_2 we notice that the above is true when $v_1 > V_2$, and also for $V_1 < v_1 < V_2$ where now $\bar{a}$ has the largest absolute value. However, because the intersection point p_1 is above $a_2 t$, when $v_1 \leq V_1$ the absolute value of the smaller acceleration a rather than the larger ' a ' must set equal to a_2 . Finally, if our maximum acceleration is a_3 we find that no matter how we adjust v_1 either a or $\bar{a}$ will exceed limitations.

This condition will occur whenever both points p_1 and p_2 are above the maximum acceleration. It is also obvious from Figure IV that when it does,

lowering the acceleration further will not suffice, contrary to what one might immediately think. A more drastic revision of the data is necessary.

In summary, a general procedure for checking and adjusting the raw velocity data is suggested below:

1. Each data point is first checked against velocity limits according to the procedure described, v_1 and $\bar{v}_1$ being modified, if necessary, according to (20).
2. The resulting accelerations required in each interval are computed and the one with largest absolute value compared with the maximum allowable. If excessive, it is set equal to the maximum and its partner is modified according to (22) or (22a), whichever applies. Otherwise the data of that interval may be left unchanged and the next two steps omitted.

Referring again to Fig IV, it is easily shown that for the ordinate of P_2 or $|a t|$

$$(23) \quad |at| = \frac{2\bar{v}_0 - (1 + \bar{p}/p)v_0}{(1 + p/\bar{p})}$$

and therefore

$$(24) \quad |a| = \left| \frac{2\bar{v}_0 - (1 + \bar{p}/p)v_0}{(1 + p/\bar{p})} \cdot \frac{v_0 + v_1}{2p} \right|$$

Now if $\left| \frac{a_3}{a} \right| = R$, we see that by multiplying $\bar{v}_0, v_0, v_1$ in (24) each by $\sqrt{R}$ and a by R gives

$$\left| \frac{2 v_0 \sqrt{R} - (1 + \bar{p}/p) v_0 \sqrt{R}}{(1 + p/\bar{p})} \cdot \frac{v_0 \sqrt{R} + \sqrt{v_0^2 R + 2ap R}}{2p} \right| = |a| R = a_3$$

from which we conclude that if for any point, $a_3 < |a|$ by a factor R we can achieve a condition at the point where both coordinate accelerations are within limits by reducing v_0 and $\bar{v}_0$ by a factor $\sqrt{R}$. In doing this, we essentially reduce the value of $|at|$ at p_2 to $|a_3 t|$.

DECLASSIFIED

-41-

HW-78712

3. If a and $\bar{a}$ were adjusted, the above procedure is repeated with the roles of the components interchanged. If the resultant accelerations both satisfy the given criterion, the process skips to the next interval using as its beginning data the new velocities computed from the adjusted accelerations of the previous one.
4. If at the end of two adjustments, the limit on either of the acceleration components is still exceeded, the rare situation occurring when maximum acceleration is a_3 exists and the data points and their neighbors must be adjusted until the transition is acceptable. The details of this process will be discussed in Section III.

Pulse Distribution

Once the velocity data has been satisfactorily adjusted, the final step remaining is to determine how each pulse instruction is to be physically spaced on the tape in order to produce the desired properties of motion.

In review these are

1. That at the end of every interval each coordinant shall have moved the amount specified by the given data. This requires a certain degree of coordination between the two components of motion.
2. That the coordinante velocities at the interval end points be those given by the adjusted velocity data.
3. That the acceleration be constant within a given interval.

The latter two conditions immediately allow the application of equations (16) and (17) to the adjusted position and velocity data. From (17) we find

$$v_1 - v_0 = \frac{2 ap}{v_1 + v_0}$$

which upon substitution into (16) gives

$$(25) \quad t = \frac{2 ap}{v_0 + v_1}$$

where t is the total time required to traverse a given pulse interval.

DECLASSIFIED

Within this interval we want to know the times t_N at which an integer number of pulses N_p have occurred for

$$N_p = 1, 2, 3 \text{ where } N_p \leq \text{Trun}(p+\epsilon)^*$$

where ϵ is some small number whose sign is that of p . To find these we again use (16) and (17), this time making the application to a pulse interval N_p long.

$$(v_0 + at_N)^2 - v_0^2 = 2 v_0 at_N + a^2 t_N^2 = 2aN_p$$

from which

$$at_N^2 + 2v_0 t_N - 2N_p = 0,$$

Solving for t_N

$$t_N = \frac{-v_0 \pm \sqrt{v_0^2 + 2aN_p}}{a}$$

and since $a = \frac{v_1 - v_0}{t}$

$$\begin{aligned} (26) \quad t_N &= \frac{t}{v_1 - v_0} \left[-v_0 \pm \sqrt{v_0^2 + 2(v_1 - v_0) N_p / t} \right] \\ &= \frac{t}{v_1 - v_0} \left[-v_0 \pm \sqrt{v_0^2 + (v_1^2 - v_0^2) N_p / p} \right]. \end{aligned}$$

Similarly, of course

$$\begin{aligned} (26a) \quad \bar{t}_N &= \frac{t}{\bar{v}_1 - \bar{v}_0} \left[-\bar{v}_0 \pm \sqrt{\bar{v}_0^2 + (\bar{v}_1 - \bar{v}_0) \bar{N}_p / t} \right] \\ &= \frac{t}{\bar{v}_1 - \bar{v}_0} \left[-\bar{v}_0 \pm \sqrt{\bar{v}_0^2 + (\bar{v}_1^2 - \bar{v}_0^2) \bar{N}_p / \bar{p}} \right] \end{aligned}$$

since $\bar{t} = t$ as a result of condition (1) above.

The sign of the radical in both (26) and (26a) depends upon conditions which will be investigated below.

* "Trun" here means "truncated value of" and is used in place of brackets to avoid confusion.

DECLASSIFIED

-43-

HW-78712

FIGURE V
PULSE DISTANCE vs TIME

In general the coordinate motion as a function of time, where the slope of the curve is v_0 at the origin or v_1 at the end of the interval, denoted by t_1 or t_2 , will be similar to the parabolic curve of Fig. V. If both v_0 and v_1 have the same sign (as in the case where $t = t_1$), the sign of the radical is the same as that of v_1 . If v_1 is opposite in sign from v_0 , there will be a portion of the curve (i.e., that for which $p > p_{t_2}$) for which two values of t_N correspond to each N_p and therefore both signs on the radical hold. The physical reason for this is that the coordinate motion reverses its direction within the interval.

Referring again to the requirements above we find that if (26) and (26a) are used to space the pulses for their respective coordinates in a given interval, each pulse train will cause its respective pulsing motor, provided it has the correct starting velocity, to have the desired end point velocity for the interval and to reach that velocity by a constant acceleration. In fact, if as previously suggested, we let t_N take on the unit of "tape spaces" (which it readily does because of the constant speed at which the instruction tape is to be read), we immediately know how to space each instruction, provided, of course, that v_0 , v_1 , etc. are also expressed in the proper units.

Furthermore, within the limits of round-off error, the correct number of pulses for each coordinate will be found in each interval.

The object then is to determine for a given pulse interval those t_N (and $\bar{t}_N$) at which an integral number of pulses N_p (or $\bar{N}_p$) occur. This can be accomplished by the use of (26) and (26a), by which we are able to properly lay out out pulse instructions spatially along upon the tape. The example that follows should illustrate this procedure and serve as a summary of the total process.

An Example

To illustrate some of the tape processing principles that have been described, let us consider the typical example of a first quadrant circular arc (which is the profile one would cut in order to obtain a hemisphere) centered at the origin as shown in Figure VI.

We assume the following dimensions:

Tool arm radius	-	b	=	3 in.
Cutter head radius	-	a	=	0.015 in.
Spindle Speed	-	α	=	200 rev/min
Chip size	-	k_c	=	0.010 in/rev.
Tape reader speed	-	v_T	=	15 in./sec.
Profile radius	-	r_c	=	2.7 in.
Table pulse increment	-	ω_p	=	$2 \cdot 10^{-5}$ in.
Ram pulse increment	-	δ_p	=	10^{-3} degrees
Tape space density	-			200 spaces/in.

In this special case equations (5a) and (5b) reduce to

$$b \sin \omega = (r_c + a) \sin \theta$$

$$\delta = (r_c + a) \cos \theta = b \cos \omega$$

and (11) and (12) under constant spindle speed to

DECLASSIFIED

-45-

HW-78712

$$\dot{\omega} = \pm 57.2957795 \frac{Ka(1+a/r_c)}{b\omega_p} \frac{\cos \theta}{\cos \omega}$$

$$\dot{\delta} = \pm \frac{Ka(1+a/r_c)}{\delta_p} \frac{\sin(\omega-\theta)}{\cos \omega}$$

where the sign in front is determined by whether θ is increasing or decreasing.

Using the interval between $\theta = 75^\circ$ and $\theta = 74.9^\circ$ for purposes of demonstration, Table I contains the information, computed from the above equations, which completes all that is necessary for laying out the coded instructions along the tape - that is, if we neglect any "overlap" from the previous one for the purpose of this example. The number of pulses in the interval and its length in tape spaces are now simply

$$P\omega = \frac{60.897703 - 60.946058}{10^{-3}} = -48.355 \text{ pulses,}$$

$$P\delta = \frac{-0.75184149 - (-0.75420477)}{2 \cdot 10^{-5}} = 118.164 \text{ pulses,}$$

$$\text{No. of tape spaces} = \frac{2(118.164) \text{ pulses}}{(838.028 + 833.744) \frac{\text{pulses}}{\text{sec.}}} \quad 15 \text{ in/sec. 200 spaces/in.}$$

= 424.091 spaces which is assumed to hold for both $P\delta$ and $P\omega$. Fig. VII is a schematic representation of the first 3/4 in. (150 spaces) of tape for the given interval where the X's represent magnetic spots and the instructions shown are

Forward ram pulse	-	X's in cols. 2 & 3
CC table pulse	-	X's in cols. 5 & 6

Cursory inspection of the data and diagram reveals several heretofore unmentioned or glossed-over points.

1. In general, the number of pulses in each interval for either coordinate is non-integral. Therefore we are faced with the problem of being able to represent each interval only by an integral number of pulses on the

DECLASSIFIED

-46-

HW-78712

tape and yet having to insure after many such intervals that the total number of these pulses is within a fraction of the total number summed over to separate intervals. This necessitates some kind of round-off procedure within each interval and a method for taking account of the fractional remainders in the next one.

2. A similar process is necessary to preserve the total number of tape spaces. Computed from (25), this quantity is not exactly equal for both coordinates since the linear acceleration assumption is only an approximation. Yet one estimate must be applied to both which will mean a slight error in the computed acceleration for one coordinate. It is assumed that the points are spaced closely enough and that the variations in the coordinate velocities are sufficiently linear that the error is relatively small.
3. The pulse spacing demanded is in general not an integral number in either coordinate. Because pulses can only be positioned at certain fixed points, this leads even in constant velocity situations, to unevenly spaced, repeating patterns. Where the pulse spacing is small, the local variation may be quite large. What effect this has on pulse motor acceleration has not been satisfactorily determined.
4. In the particular interval demonstrated, there is little difference between a "fixed" spacing according to average velocity and constant acceleration spacing since the velocity change over the interval is small. Over a normal contour, this will be true for most intervals; only where the pulse motors are starting or stopping is there usually any appreciable difference.

FIGURE VI
CIRCULAR PROFILE

TABLE I - CUTTING HEAD POSITION AND VELOCITY

θ deg.	ω (deg)	δ (in)	$\dot{\omega}$ (pps)	$\dot{\delta}$ (pps)
75	60.946058	-.75420477	-341.173	838.028
74.9	60.897703	-.75184149	-342.874	833.744

FIGURE XII
TAPE SECTION

Interval Hooking

As seen from the previous example, a given interval almost always contains a non-integer number of pulses. Care must be taken that this is accounted for in moving from one interval to the next in order to assure a smooth transition. With the proper handling of round off and remainders, this can be accomplished.

Supposing in an interval (1) there are p_1 pulses to be distributed in a time t_1 and that p_1 rounds off to

$$(27) \quad p_1^* = \text{Intf}(p_1 + \epsilon) \text{ where } \epsilon \leq 0.5,$$

the last one of which is to be executed at time t_1^* . Then in going to the next interval (2), the first pulse is to be found at a pulse distance from the beginning of the interval of

$$(28a) \quad p_2^{(1)} = 1 - \Delta p \quad \text{where } \Delta p = p_1 - p_1^*$$

and each pulse thereafter at

$$(28b) \quad p_2^{(k)} = N - \Delta p \quad \text{where } N = 2, 3, \dots, \text{Intf}(p_2 + \Delta p + \epsilon).$$

The execution time for each of these, measured from t_1^* , is then just

$$(29) \quad t = t_2^{(N)} + \Delta t$$

where $t_2^{(N)}$ is the execution time of $p_2^{(N)}$ measured from the beginning of interval (2) and $t = t_1 - t_1^*$.

Usually the quantity ϵ is much less than 0.5 because the time computations are more accurate when the velocities used are those for the interval in which the pulse remainder is contained (though significant variation appears only when velocity is small and acceleration large). The only reason for having only $\epsilon > 0$ at all is to assure that in hooking a number of curves together to form a closed loop, round off error does not cause a pulse to be dropped at the end. A value for ϵ of 0.1 or less should be sufficient under most conditions.

Tape Length Estimation

Often it is desirable to have an estimate of the length of tape required to do a particular job, especially if there is any doubt that it will fit on one reel. A fairly accurate estimate can usually be arrived at by computing the time required to traverse each curve section and adding to find the total. This multiplied by the tape reader speed gives the total length of tape required.

When the equation of the curve is known, the time required to traverse it is given, from a manipulation of (10a), as

$$(30) \quad t = \frac{1}{k_c} \int_{\theta_1}^{\theta_2} \frac{\sqrt{r^2 + r'^2}}{\bar{a}} d\theta$$

$$(30a) \quad = \frac{1}{k_c \bar{a}} \int_{\theta_1}^{\theta_2} \sqrt{r^2 + r'^2} d\theta \text{ for constant spindle speed given by } \bar{a}$$

$$(30b) \quad = \frac{1}{k_c v} \int_{\theta_1}^{\theta_2} \sqrt{(r^2 + r'^2)(k_c^2 + r^2 \sin^2 \theta)} d\theta \text{ for constant surface travel.}$$

where θ_1 and θ_2 are the angles of the end points as measured from the origin.

For constant spindle speed, note that

$$(31) \quad \sqrt{r^2 + r'^2} d\theta = dl \text{ where } l \text{ is the curve length}$$

and therefore

$$(32) \quad t = \frac{l}{k_c \bar{a}}$$

No constant surface travel expressions will be given here. In fact, no exact solutions to (30b) exist, nor any approximation that is applicable to all types of curves such as (32). However, since the constant spindle speed case usually takes more time, an upper bound on the time required can usually be found

DECLASSIFIED

HW-78712

by using it in place of the more difficult (30b). Alternately, (30b) may be closely approximated for many particular cases by neglecting k_c under the radical.

An important case where (30) does not apply is when the cutter head is not actually machining the workpiece, but moving in air to position itself for another pass. Such movements are usually governed by the specified motions routine where the coordinate undergoing the greatest movement is accelerated to a given maximum speed. The time required for this is

$$(33) \quad t = p/v_{\max} + \frac{v_{\max}}{a_{\max}}$$

where p is the number of pulses required of the above coordinate and $v_{\max}$ and $a_{\max}$ are the maximum allowable speed and acceleration respectively.

To find the total time required for a given cutting process then means adding all the times required for the various curve sections as given by (32) or (33) and also the idling times acquired by blank space specification. If the total time multiplied by the desired tape reader speed demands more than 2400 ft. of tape, either the time t required for the process must be decreased or the tape reader speed reduced.

IV. PROGRAMMING AND OPERATION

Introduction

The Fortran program CUPID incorporates the concepts, equations, and methods discussed in Sections I - III to produce a magnetic tape on the IBM 7090 that can be transferred directly to PINS to control a given cutting operation.

This section has basically two purposes:

- (1) To describe the important features of CUPID from a programming standpoint in order to convey a fuller appreciation of its abilities and limitations and serve as a guideline for future improvements and modifications.
- (2) To describe the input format for future users.

Included here for these purposes are a Fortran listing of the program and a brief description of important Fortran symbols.

The program is the result of approximately 16 man months of labor during which many of the original concepts have been drastically modified or discarded for better or more comprehensive ones as experience and a clearer understanding of the problem inspired new ideas. Modifications in the numerical control unit have also antiquated some of the original specifications, thereby making portions of the present program superfluous. Some of these sections still remain where their inclusion makes the program more general and works no hardship on its operation.

Programming

CUPID consists of a main program and 12 subroutines. Its organization has been partially a result of convenience for debugging purposes and partially a result of how the program itself has developed. At present almost every subroutine incorporates some specific procedure or series of computations that seem to make it a natural subdivision of the total program. All are Fortran coded except one, the subprogram LNREC, which performs the actual

DECLASSIFIED

-52-

HW-78712

scrunching process and is in FAP. An auxiliary program, SQUASH, which is used when any redundancy occurs in the scrunching process, is identical with the subroutine SQUASH except for a few statements and the fact that it is a main program. Its use will be explained later in this section.

Because both CUPID and the control unit are still in the experimental stage, many of the checks written into the usual finished program have been omitted. A fairly detailed knowledge of the program, therefore, is required for its operation. An offsetting advantage, however, is that this results in greater logical simplicity.

Since the program involves concepts of a different type than those usually encountered, it has been deemed advantageous to use a somewhat different method to explain the logical flow. Each program contains appropriately spaced comment cards which give it an outline form. This same outline form is used to explain each program in the material that follows, the explanation under each point referring to the Fortran statements under the corresponding comment in the listing. Flow charts are not included because their complexity has not been judged helpful in clarifying the situation. No attempt has been made to rigidly separate the information concerning the logical flow of CUPID and that concerning the restrictions on the input data since many times the latter is most easily and clearly present along side of the former. Such information may not be repeated in the "Input Specifications" subsection, and therefore the remainder of the material presented should be read as a unit. As has been previously stated, the purpose of CUPID is to produce a continuous feed, magnetic control tape for a two-dimensional contour lathe with δ - ω coordinate geometry. An outline of the main program is given below.

A. Initialize Parameters

The first step is to set up the boolean masking array that will later be used to select the proper bit position within a memory word for locating

any given instruction. Then the parameters used to store data during various phases of the pulse layout routine are initialized, along with pulse counting vector, the tape instruction array, and counting and indexing parameters.

B. Read Heading and Case Data

A heading card of alpha-numeric characters, cols. 1-80, is read, and also a list of variables that will remain constant for any lathe cutting operation controlled by a single tape. If the cutting arm radius $b \leq 0$, the program assumes its computations are complete and exits. Otherwise it goes on to read the tape instruction code and write it out following the contents of the heading card.

C. Mount Output Tapes

The program prints a message to the operator to mount the output tapes as specified on the operating procedures card and pauses. This section is skipped if the previously read variable, "MOUNT" = 1.

D. Read Curve Section Input

A curve section card is read that specifies the next type of curve the cutter head is to follow from among the possible types available and other input pertinent to that part of the machining process. If the curve specification number ≤ 0 , the program goes into the scrunching procedure, except where MOUNT = 1. The program then returns to the beginning to look for a new case involving new tapes, if such are desired.

E. Set Up Parameters and Write Output

Parameters for determining coordinate velocities and maximum acceleration are converted into the units used in pulse distribution and more input is written out for checking purposes.

F. Read and Set Up Auxiliary Functions

The program calls AUXIL, which reads and sets up any auxiliary functions that have been requested.

G. Compute Curve Data

The program chooses one of four types of curves to compute or to leave blank space and calls the proper subroutine or transfers to the appropriate location to set up the necessary data, depending on the value of ICURV.

H. Adjust Curve Data

After the position and velocity data for a curve has been computed, the latter is adjusted according to specified limitations by calling ADJUST. Then the δ - ω and r - θ curve coordinate positions are written out in a table.

I. Distribute Instructions In Memory

The program distributes instructions in a memory array according to their desired tape layout using the δ - ω coordinate positions and the adjusted coordinate velocities at these points. For each coordinate, interval by interval, the tape distance and number of pulses required to traverse the interval are computed and added to any fractional remainders left over from the previous interval. The proper instruction type is chosen and if the number of pulses required ≤ 1 with round-off, the data for that interval is saved and added to the next interval. Otherwise, the number of rounded off pulses for the interval are distributed according to a constant speed rule or according to linear acceleration by (26). This is done interval by interval, alternately for each coordinate within each interval. It is accomplished by a boolean logical multiplication and addition using the instruction type vector DOT and the masking vector RMASK, the correct element of each chosen according to the type of instruction and the relative location within the instruction memory word desired. This location is found from the remainder after division by 6 (the number

~~DECLASSIFIED~~

HW-78712

of tape lines per word) of the tape line distance to locate the correct memory position. The remaining fractional data is computed and saved for use in the next interval.

J. Compute Memory Array Data

After each data interval is completed, the amount of memory array space filled is computed. If the instruction array overlaps its allowable space, a message, "Tape Vector Limits Have Been Exceeded", is written and the program exists.

K. Distribute Auxiliary Functions

If any auxiliary functions are required, these are checked to see if the completed tape array has reached the point in time where they are called for and if so the subroutine PLACE is called, which enters them in the memory array

L. Write Memory Array on Tape

If the memory space filled exceeds 2550 words, the first 2550 of these are written out on the master tape under the normal writing mode, while the remainder of the instruction data is transferred to the beginning of the memory array to provide room for more data.

M. Read New Curve Data

When the instructions corresponding to every data interval for the given curve section have been distributed in memory, the program returns to read the data for a new curve section and the process is repeated.

AUXIL

This subroutine reads and prepares auxiliary function data for later use.

A. Save Remaining Auxiliaries

Any auxiliary data read in with previous curve sections but not distributed in the instruction array is shifted to the first positions of the auxiliary arrays.

B. Read New Auxiliaries

New data is read and added to the end of the data arrays.

C. Sort Auxiliaries

The arrays are sorted so that the functions are ordered in the sequence in which they are to appear on the tape.

D. Correct For Overlap

Any auxiliaries whose instructions could possibly overlap each other on the tape (some auxiliaries require more than one tape line as will be explained later) are shifted to avoid conflict.

LINE

This subroutine computes data points on a specified straight line and the corresponding cutter head path between two given angles, $THET_1$ and $THET_2$, as measured from the origin.

A. Read and Write Input Data and Initialize

The data defining the line and the restrictions on its machining are read and written. Other parameters are initialized

B. Special Cases.

If the angles $THET_1$ and $THET_2$ are equal, the line is degenerate and control is returned to the main program except where $THET_1 = 0$, in which case it is assumed that a pure ram motion of a distance equal to RHO in length is desired.

C. General Cases

In the general case the data points are computed from the line in $r-\theta$ coordinates and for the cutter head path in $\delta-\omega$ coordinates according to (3a), (3b), (3c) and (6a), (6b). The coordinate velocity data is computed according to (13) and (14).

CIRCLE

This subroutine computes data points on a specified circle and on the corresponding cutter head path between two given angles THET_1 and THET_2 as measured from the center of the circle.

A. Read and Write Input Data and Initialize

The data defining the line and the restrictions on its machining are read and written. Other parameters are initialized.

B. Special Case - Arc Centered at Origin

The case is treated separately from the more general one.

C. General Case

The data points are computed for the circle in $r-\theta$ coordinates and for the cutter head path in $\delta-\omega$ coordinates according to (4a), (4b), (4c) and (7a), (7b). The coordinate velocity data is computed according to (15) and (15a).

PMET

This subroutine computes data points for specified cutter head motions in both coordinates and linearly relates them. This is accomplished by making the ratio of the coordinate velocities equal to the ratio of the number of pulses required. This routine is generally used to specify the motion of the cutter head when it is off the workpiece and being positioned for a new pass.

A. Read and Write Input Data and Initialize

The data defining the motion of each coordinate and the velocity of that one with the greatest pulse count is read along with certain restrictions on the motion. From these, other parameters are computed.

B. Compute Data Points

The data points for the cutter head path and the desired velocity at those points are computed from parameters defined under A above.

C. Adjust for No Coordinate Motion

If the desired movement in a given coordinate is smaller than a certain input parameter, EPS, index parameters are set so that the movement is considered to be zero.

CONTR

This subroutine computes contour and cutter path data for a point defined contour, where each adjacent point pair is fit with a unique cubic. A larger, more closely spaced set of point positions and their coordinate velocities are then computed.

A. Read and Write Input Data and Initialize

The input data points, the end point derivatives, and other curve defining parameters are read and written.

B. Compute Cubic Coefficients and r-θ Point Data

Subroutine CURVE is called to fill in the desired point position data.

C. Compute Reflected r-θ Point Data

If specified by the input, the r-θ data for a specified number of input points adjacent to the beginning point of the curve defined above are reflected about that point, thereby extending the beginning of the curve.

The reflection is valid only if at the beginning point of the unextended curve $\theta = 90^\circ$, and if its derivative there = 0. Then the equation of reflection become

$$\begin{aligned}
 (34) \quad \begin{aligned} r_f &= r \\ \theta_f &= 180^\circ - \theta \\ r_f' &= -r' \\ r_f'' &= r'' \\ r_f''' &= -r''' \end{aligned}
 \end{aligned}$$

where (r_f, θ_f) is the reflected point.

D. Read and Compute r-θ Extended Point Data

Again as specified the curve is extended by reading in new data points and fitting with cubics, using in each new interval the first and second derivatives from the beginning of the last one as determined by the previous curve fitting operation.

E. Compute δ-ω Point Data

The cutter head path and coordinate velocities are computed using the previously derived equations (5a), (5b), (5c), and (11), (12).

CURVE

This subroutine fits each adjacent point pair of a point defined curve with a unique cubic, given the points and the end point derivatives. This is accomplished by the symbolic inversion of a matrix to solve a set of simultaneous linear equations formed by assuming continuity of the curve and its first two derivatives at the interior points.

A. Initialize Parameters

Parameters to be used extensively in later calculations are computed here.

B. Special case, only two points

In the case of only two points (there must be at least two) with given end derivatives the cubic coefficients a, b, c of

$$Y = Y_0 + c \bar{X} + b \bar{X}^2 + a \bar{X}^3$$

$$\text{where } \bar{X} = X - X_0$$

are easily found to be

$$\begin{aligned} c &= Y_0' \\ b &= 3 \frac{Y_1 - Y_0}{(X_1 - X_0)^2} - \frac{Y_1' + 2Y_0'}{X_1 - X_0} \\ a &= \frac{Y_0' + Y_1'}{(X_1 - X_0)^2} - 2 \frac{Y_1 - Y_0}{(X_1 - X_0)^3} \end{aligned}$$

where (X_0, Y_0) , (X_1, Y_1) are the given points.

DECLASSIFIED

HW-78712

C. Compute Cubic Coefficients

The cubic coefficients a_1 , b_1 , c_1 are computed for each interval from relationships given by Dean [3].

D. Compute Intermediate Data Points and Their First Two Derivatives

Using the cubic parameters, intermediate data points and their first two derivatives are computed. This is done by breaking each interval into a specified number of sub-intervals.

ADJUST

This subroutine adjusts coordinate velocity data points according to the limitations of the numerical system as expressed in terms of a maximum velocity and a maximum acceleration. When limitations are exceeded, the adjustments made attempt to bring the exceeded variables to their maximum values.

A. Initialize Parameters

Assigned GOTO's are set according to whether there is motion in two coordinates or only one and the maximum velocity is converted to the proper units for comparison with the data points.

B. Adjustment For Maximum velocity

The beginning velocity is set equal to the end point velocity of the last curve to insure continuity and the coordinate with the largest total motion (if there is more than one) is used to check first.

If a data point exceeds the maximum velocity it is adjusted to that value and similarly, its counterpart in the other coordinate according to (20a). Then the process is repeated with the coordinate roles exchanged.

C. Initialize Parameters and Adjust End Point Velocity

Parameters of data point selection are set so that the following DO loop will traverse the array from beginning to end and the end point velocities are modified as given by the input data.

DECLASSIFIED

HW-78712

D. Adjust For Maximum Acceleration

The interval location is selected and then the routine begins to compute the acceleration in that interval for each coordinate motion and check the largest value against maximum acceleration. If the computed value is within limits, the velocities are left unchanged, the time to traverse the interval is computed, and the routine goes on to the next interval. Otherwise, the largest acceleration is adjusted to maximum and the other coordinate checked to see if its required modification is within limits. If so, the routine goes on to the next interval; if not, the procedure is repeated with the roles of the coordinates reversed. If the readjusted acceleration values still exceed limitations, the routine backs up an interval and repeats the whole checking process in hopes of changing the beginning velocities of the original interval enough to overcome the previous barrier. This back-up procedure continues as long as the acceleration of one coordinate in any one of the included intervals exceeds the maximum up to a specified number of points, at which time it prints out a failure message and goes on to the interval beyond the one at which back-up was first initiated.

During back-up new accelerations are computed for each interval where the end point velocity of the interval is considered to be the end point velocity of the one in which the back-up was initiated. In other words, if back-up was initiated in the interval with end point velocities u_K and u_{K+1} for a particular coordinate, the new acceleration at the i^{th} back-up is considered to be

$$(36) \quad a = \frac{u_{K+1}^2 - u_{K-1}^2}{2 P_{K-1}}$$

rather than the original $\frac{u_{K+1}^2 - u_K^2}{2 P_{K-1}}$. The reason for this is two fold. First, when backing off an interval, we must change the end point

DECLASSIFIED

DECLASSIFIED

velocities of the previous one, if we are ever to satisfy the restrictions that have created the impasse. But this cannot be done unless we change the accelerations during the interval. Secondly, by using u_{K+1} in particular, we have cast the adjustment process into a form that will readily smooth out the velocity peaking effects illustrated in the region DE of Figure III in Section III.

When the acceleration limits are exceeded in an interval where the end point velocities have opposite signs, it is not immediately obvious just how to determine the sign of the adjusted velocity. This is because the motion changes direction in the interval, as shown in Figure VIII, a fact that is not

FIGURE VII
COORDINATE MOTION REVERSAL

accounted for in computing pulses for the interval from the distance between the end point positions as represented by P_0 and P_1 .

What is done in this case is to choose the sign of the adjusted end point velocity to be the same as the sign of the pulse count. In this way it will correctly identify the direction of motion in the interval. The traversal of this region will be somewhat discontinuous because the information in the peak of the curve above P_1 will be neglected, but this should work no hardship on the pulsing motors since they are able to follow rather large instantaneous rate changes at low speeds.

The total adjustment process is applied in succession to every data interval of the curve section.

E. Set Parameters For Another Pass

If, after the acceleration adjustment routine has checked every curve interval, there are one or more points at which an adjustment failure occurs, the program modifies the data point selection parameters so that the whole velocity data array will be traversed in the opposite direction beginning with the last point and the process is repeated. If intervals remain in which limitations are exceeded after the second pass, no further adjustments are made and the velocity data remains as last computed.

PLACE

This subroutine distributes in memory auxiliary functions of two different types. The type of auxiliary and its relative time position with respect to the beginning of a curve section on the tape is specified by the input data.

A. Initialize Parameters

The relative memory array location is computed and certain switching and indexing parameters are set.

B. Choose Auxiliary Type

The program transfers to the section that handles the type of auxiliary specified.

C. Multi-Instruction Type

In this type of auxiliary the number of instructions of a specific type determine the function to be performed. Each instruction of the series is placed so as to be 0.1 sec. down the tape from the previous one. When this is completed two "execution" instructions are added which trigger the execution of the function. The numerical control unit, when reading the tape counts these instructions and stores the number in a counter until

it encounters an execution instruction pair, and then causes the auxiliary corresponding to the number in the counter to be performed.

D. Single Instruction Type

This type of auxiliary uses a single tape line as the pulse instructions. In fact an extra pulse can be added in by use of this routine if the input is properly specified. When a pulse instruction is called for the routine goes into an elaborate checking procedure to see if the same instruction or one for motion of the same coordinant in the opposite direction will be within a specified distance in either direction on the tape. If this is so for an identical instruction, the relative location of the instruction is adjusted to the specified distance from the other or if caught between two identical instructions whose distance apart is less than twice the specified minimum, it is centered between them. If it is so far an instruction governing the same coordinate motion but of opposite direction, that instruction is erased from memory. An auxiliary of the single instruction type that is not a pulse instruction is merely placed in its specified position.

SQUASH

This subroutine calls the "scruncher" subroutine LNREC, and determines the results of the scrunching process.

A. Rewind Tapes, Call Scruncher, and Rewind Again.

The master and control tapes are rewound, LNREC is called, and the tapes are again rewound after the scrunching process is completed.

B. Determine Scrunching Results

After the scrunching process is completed, a check is made on the values of M and N to see if the writing channel disconnected at any time during the process or if any check sum errors have occurred. If M and N are both zero the process is assumed to have been completed successfully and a message

DECLASSIFIED

-65-

HW-78712

is written out to this effect. If M is non-zero, a tape redundancy is assumed to have occurred and if N is non-zero the channel is assumed to have disconnected. Appropriate messages are written in either case.

LNREC

This subroutine writes information from the master tape onto the control tape, removing the record gaps in the process. The method, which has been termed "scrunching", was developed by E. E. Sundell of EDPO. Two capabilities are required of a computer performing this operation:

1. It must have at least two data channels with the ability to operate independent of the main frame.
2. It must have the ability to discontinue the delay of the main frame with channel instructions.

A sketch of the FAP subroutine, developed by Sundell for the purpose of "scrunching", that uses the above two properties of the IBM 7090 is as follows:

1. A fixed number of records are originally read from the gapped tape into memory through data channel A while the main frame is delayed. After the specified record is completely read, the computer begins writing it out onto the second tape through channel B.
2. During the writing process the main frame is again delayed until a certain fixed number of the total records originally read in have been written, then the reading process is repeated through channel A into the same memory array and over the old data as shown in Figure IX A.
3. When channel B is through writing the original data, without disconnecting it immediately returns again to the beginning location of the array it has just written, there to find the old data overlayed by new from channel A, and repeats the writing process as shown in Figure IX B. When channel A is through filling the memory block it disconnects and the main frame again is

delayed until channel B has reached the position where the delay is discontinued and the cycle is repeated.

4. This cyclic process of continuous writing through channel B and intermittent reading through channel A continues until the reader sees an end of file mark on the gapped tape, at which time the writing of the array then in memory is completed, channel B is disconnected, and the process is terminated.
5. At termination the program checks for redundancy in the reading and writing processes and also to see if channel B has disconnected at any time prior to termination.

FIGURE IX
"SCRUNCHER" MEMORY SCHEMATIC

Three prerequisites to the process are evident from the above description. First, we need an end of file mark at the end of the data on the gapped tape, which is easily supplied after the last record of information is written and before the "scrunching" begins. Secondly, the total number of records on the gapped tape must be an integral multiple of the number brought into memory during each reading operation through channel A. This is required because the reader

looks for an end of file mark only after each reading. The problem is trivial, however, since blank records can be added at the end of the data to complete the last array.

Thirdly, there exists the most important problem of timing the reading-writing cycle described above. Since channel A reads only intermittently and channel B writes continuously, we see immediately that we must be able to read through the former at a faster rate than we write through the latter. This can be accomplished by reading in a high density mode of 556 bits/in. or 800/bits in. and writing in a low density mode of 200 bits/in. Also the periods of reading must be spaced in time so as to continually keep memory supplied with new data but never overlap data that has not yet been written.

The average IBM 7090 reading-writing time in milliseconds for an array of information is given by *

$$t_{200} = .267 \times \text{Number of words} + 7.3 \times \text{Number of record gaps} \quad (37)$$

for low density of 200 bits/in.

and

$$t_{556} = .096 \times \text{Number of words} + 7.3 \times \text{Number of record gaps for the} \quad (38)$$

high density mode of 556 bits/in.

Letting N be the number of records in the fixed size array, the writing time from (38) becomes

$$t_w = .267 (255N)** = 68 \text{ IN} \quad (39)$$

where the record gap term is dropped because the gaps have been eliminated.

The reading time in high density becomes

$$t_r = .096 (256 N) + 7.3 N - 31.9 N \quad (40)$$

By comparison of (39) and (40) we see that for arrays of just a few or more records the reading time is sufficiently less than the writing time to allow for the type of procedure we have described above.

* These can be deduced from the 729 IV tape speed and other tape parameters See [1], p. 68-9.

** While records are normally 256 words long, the first word of each record contains merely housekeeping information and therefore is read but not written.

For the other timing question, namely at what time during the writing process to begin reading a new array into memory, we refer again to Figure IX A. Letting n be the number of records written when the reading begins we require that it be possible for the remainder of the array to be written before new information overlaps it.

That is from (39) and (40)

$$31.9 N > 68.1 (N-n)$$

from which

$$n > .532 N$$

(41)

Therefore we see that if $N=10$, for instance, the reading process may begin anywhere from the end of the sixth record to the end of the ninth. Thus it is seen that the proposed method is feasible from a timing standpoint.

The only other major concern is whether it is possible to obtain magnetic tapes sufficiently free from scratches and blemishes to accept a long continuous record of information without redundancy. Experience has not only shown this possible, but with a minimum of selectivity highly probable. With moderately used tapes and writing a 1000 - 1200 ft. record, only on about one tape in four would any redundancy occur - a significantly smaller ratio than one might expect.

ARCTAN

This function subprogram accurately computes the arctangent of an angle in radians.

ARCINV

This function subprogram computes an angle, as specified by its sine and cosine in angular degrees. The angle itself is assumed to lie between 0 and 180 degrees.

INPUT SPECIFICATIONS

While many details have been omitted, it is hoped that a reader of this report who is familiar with Fortran could, from the information thus far given, by carefully observing the Fortran listing deduce the input necessary for performing a desired series of operations under given specifications. This section, however, is designed to facilitate an understanding of the input data format and the many types of operations and variations of them that it is possible to specify. Suggestions for improvements in the present input format will be discussed later.

The input itself can be divided into two major sections for each control tape to be produced:

1. Case Input, which specifies parameters that will remain constant through the tape producing operation.
2. Curve Section Input, which specifies the type and dimensions of each curve section and the data required to carry out the necessary computations and properly lay out the tape instructions.

The case input is read first, and then the curve section input, each new section being added after the previous one has been successfully translated into tape instruction language. After the final section has been completed, a blank card signifies the end of the operational sequence desired and the tape is "scrunched". The machine then looks for data for a new tape. If it reads four more blank cards (all zero case input data), the program exits.

The Case Input is as follows:

<u>Card</u>	<u>Variables</u>	<u>Format</u>
1	(HEAD(J), J=1,14)	13A6, A2
2	BAR, ARAD, DELD(1), DELD(2) SPEED, TRAVEL, RDI (1), RDI (2)	8 E 10.6
3	DLIM, KNUM, MOUNT	E10.6, 2I5
4	(DOT (J), J=1, KNUM)	6012
4 (Continued)	DOT	
:	:	
:	:	

DECLASSIFIED

HW-78712

HEAD is a vector storing alpha-numeric information to identify the case contained on col 1-80 of the first card.

BAR is the length of the cutter arm in inches.

ARAD is the radius of the cutter head in inches positive for an outside contour, negative for an inside one.

DELD is a vector storing the step sizes, first for the ram and then for the table in inches & degrees respectively.

SPEED is the speed at which the control tape is to be read on the numerical control in inches per second.

TRAVEL is the spindle speed in rev. per min. for the constant spindle speed case or could be the surface speed of the cutter head for the constant surface speed case when this option is available.

RDI is the vector specifying the beginning pulse motor speeds for the ram and table respectively in the units pulses per tape space. This vector usually has both elements equal to zero.

DLIM is the round-off parameter in pulses. When we are within DLIM pulses of an extra pulse when truncating a fractional number of pulses, the total number of integral pulses is increased by one.

KNUM specifies the number of tape instructions to be read under the vector DOT.

MOUNT is a control parameter which determines whether or not the program writes on a master tape and scrunches. If MOUNT = 1, no master or control tape writing is allowed; otherwise the writing proceeds as previously described.

DOT is the tape instruction vector whose length is determined by KNUM. Each word contains a two bit octal instruction code repeated 6 times. (Example - 40 40 40 40 40 40). The first four words of DOT must contain the codes for the counter storage and counter trigger for the multi-instruction auxiliaries respectively.

Input for each new curve section begins with a similar card:

Card	Variables	Format
1	ICURV, NDUM, EPS, CHIP, ACEL, JCUN	2I5, 3E10.4, I5

ICURV specifies the type of curve section. The present code is

- 1 - Line
- 2 - Circle
- 3 - Specified coordinate motions, linearly related (PMET)
- 4 - Point defined curve
- 5 - Blank tape (no coordinate motion)

NDUM specifies the number of new auxiliary functions to be read on the next card,

EPS is a some small positive number used to compensate for round-off error in checking for equality between two numbers in a variety of places.

CHIP is the chip size in inches per revolution

ACEL is the maximum allowable pulse motor acceleration in pulses per second.

JCON specifies whether the memory array written on the master tape is also to be written on tape A_3 for inspection. This is done only if JCON = 1.

if NDUM > 0, another card (or cards) follows this one which contains the auxiliary function information:

Card	Variables	Format
2 : : : (Cont'd)	(DSPEC(J), ISPEC(J), KLIM(J), J=1, NDUM)	8(F4.0, 2I3)

DSPEC is the time down the tape from the beginning of the new curve section data that the auxiliary is to be located.

ISPEC denotes the type of auxiliary (See explanation of AUXIL for various types)

KLIM is input only where the auxiliary is a pulse instruction, in which case it specifies a desirable minimum spacing in tape lines from the nearest pulse of the same type. In other cases KLIM is either unused or is computed internally and has a slightly different meaning.

The format for the final card (or group of cards) of a curve section depends upon the type of curve specified.

DECLASSIFIED

For the line

Card	Variables	Format
3	RH0, PHI, THET ₁ , THET ₂ , ITOT, IREV, KDET, KREV, LDET	(4E10.6, 5I5)
RH0	is the perpendicular distance from the origin to the line segment.	
PHI	is the angle of the perpendicular with respect to the horizontal or positive X-axis.	
THET ₁	is the angle made by the line drawn from the origin to the beginning point of the segment with the X axis.	
THET ₂	is the angle made by the line drawn from the origin to the end point of the segment with the X axis.	
ITOT	is the number of data intervals into which the segment is to be divided. The division is equi-angular where the angles are measured similarly to those specified above.	
IREV	is the maximum pulse motor speed in the data interval.	
IDET	specifies whether the end point velocity is to be modified. This is necessary if the transition point between curve sections does not have a continuous derivative, if coordinate motion is to cease at the end of the section, or if the chip size changes from section to section lest the acceleration demands at the transition point become too great. A "1" or "2" for KDET specifies that the ram or table motion, respectively, is to be set equal to a given value at the end point, while the other coordinate is also modified according to (20).	
KREV	specifies the modified end point velocity in pulses per second.	
LDET	specifies the maximum number of points in the data array that the program may back up in attempting to achieve a satisfactory velocity adjustment. (See the explanation of ADJUST)	

For the circle

Card	Variables	Format
3	RH0, PHI, DRAD, THET ₁ , THET ₂ , CSIGN, ITOT, IREV, KDET, KREV, LDET	(5E10.6, F5.1, 5I5)

DECLASSIFIED

-73-

HW-78712

RH0 is the distance from the origin to the center of the circle.

PHI is the angle of RH0 with respect to the x axis.

THET₁ is the angle made by the line drawn from the center of the circle to the beginning of the segment with the x axis.

THET₂ is the angle made by the line drawn from the center of the circle to the end of the segment with the x axis.

ITOT is the number of data intervals into which the segment is to be divided. The division is equi-angular where the angles are measured similarly to those specified above.

CSIGN specifies whether the cutter head is located on the outside or inside contour of the circle. Its value must be either "1" or "-1" for the outside or inside contour respectively when an outside surface is being machined. For inside machining, the meaning of the signs is reversed.

IREV, KDET, KREV, and LDET have functions identical with the same variables for a line.

For the linearly related coordinate motions,

Card	Variables	Format
3	RH0, PHI, ITOT, IREV, KDET, KREV, LDET	(2E10.6, 5I5,
RH0	is the distance to be moved in δ .	
PHI	is the distance to be moved in ω .	
ITOT	is the number of intervals into which each coordinate segment is to be divided. The intervals will mark of equal distances in each coordinate.	

IREV, KDET, KREV, and LDET are the same as before.

For the point defined curve,

Card	Variables	Format
3	DY ₁ , DY ₂ , NMAX, NINT, IBEG, MC, IREV, KDET, KREV, LDET	2E10.6, 8I5
4	(X(J), Y(J), J=1, NMAX)	8 F 10.6
(Cont'd)		
:		
:		
:		

DECLASSIFIED

Card	Variables	Format
5	(XS(J), YS(J), J-1, MC)	8 F 10.6
(Cont'd)		
.		
.		
.		
DY ₁	is the derivative at the beginning in inches per deg. point of the portion of the contour to be defined with cubics by means of the symbolic matrix inversion.	
DY ₂	is the derivative at the end point of the contour.	
NMAX	is the number of points involved in the linear system of equations to be solved by the matrix inversion.	
NINT	is the number of segments into which each data interval is to be divided. The division is equi-angular where the angles are measured from the origin.	
MC	is the number of extended points to be added to the beginning of the given point array.	
IBEG	is the number of reflected points to be added between the extended points and the originally given points.	
IREV, KDET, KREV, and LDET are the same as before.		
X	is the vector of θ coordinates of the points involved in the matrix inversion.	
Y	is the vector of r coordinates corresponding to the vector of θ coordinates.	
XS	is the vector of θ coordinates of the extended points.	
YS	is the vector of r coordinates of the extended points.	

For the blank type

Card	Variables	Format
3	ITOT	I5
ITOT	in this case is the number of seconds of blank space desired when the tape is run at the given speed.	

DECLASSIFIED

-75-

HW-78712

Input Restrictions

Some restrictions on the input data exist which the program itself does not at present check and which, therefore, should be checked externally before a run is attempted. They occur either because of the nature of the variables involved, the dimensions of the Fortran variables, or the dimensions of the program arrays.

The restrictions that are known but unchecked are as follows:

- (1) BAR, DELD(1), DELD(2), SPEED, TRAVEL, and DLIM should always be positive numbers.
- (2) $1 \leq \text{KNUM} \leq 64$
- (3) $1 \leq \text{ICUPV} \leq 5$
- (4) NDUM, EPS, and CHIP should be positive.
- (5) DSPEC(1) - - - DSPEC (NDUM), KLIM(1) - - - KLIM(NDUM) are always positive
- (6) $\text{ISPEC}(J) < 382.5/\text{SPEED} + \text{KNUM} - 3$
- (7) RH@ is always positive for the line and circle.
- (8) $|\text{THET}_2 - \text{THET}_1| < 180^\circ$, $|\text{THET}_1 - \text{PHI}| < 90^\circ$, and $|\text{THET}_2 - \text{PHI}| < 90^\circ$ for the line.
- (9) IREV, KREV, and LDET are always positive.
- (10) CSIGN = ± 1
- (11) $1 \leq \text{ITOT} \leq 1000$ except for the blank space format.
- (12) $2 \leq \text{NMAX} + \text{IBEG} + \text{MC} \leq 60$, $\text{NMAX} \geq 2$
- (13) $0 \leq \text{MC} \leq 10$
- (14) IBEG and NINT are positive, $\text{NINT} \geq 1$
- (15) $(\text{NMAX} + \text{IBEG} + \text{MC}) \cdot \text{NINT} \leq 1000$
- (16) Total number of tape instruction records ≤ 376 (Determined by assuming a normal 2400 ft. reel)

DECLASSIFIED

-76-

HW-78712

Operating Procedures

The procedures card submitted with the program should contain the numbers and locations of the reels to be used as master tape and control tape. This master tape is to be mounted on tape unit A5 and the control tape on unit B6. If it is desired that the operator select either or both tapes, the reel number should be omitted but a debug location should still be assigned to both. If more than one case is to be run involving more than one set of tapes, this should be carefully noted.

If a redundancy error should occur on the control tape, only the tape scrunching process need be rerun. This requires that the subroutine SQUASH be converted to a main program in the following manner:

- (1) Remove the initial card that labels the program as a subroutine
(SUBROUTINE SQUASH)
- (2) Replace all RETURN statements with CALL EXIT statements.
- (3) Add the following statements as the first executable ones in the deck:

PRINT 1100

PAUSE 77777

and also format statement 1100 from the main program.

With LNREC, this program can be used repeatedly until the "scrunching" process is successful. Its running time should be less than 5 minutes.

The running time of CUPID varies considerably with the number, length, and types of curve sections involved and therefore is best discovered for one's own particular problems by actual practice. A control tape that is to be run at 20 in. per sec. and containing enough data to fill approximately half of the reel has taken the program about 15 minutes to completely process.

Sometimes where various curve sections are linked, and it is vital to know the velocities at the point where they join in order to effect a smooth transition, it is easier to do a trial run of certain curve segments than resort

to hand calculations. For runs of this sort, the tape writing may be omitted by setting the input variable MOUNT = 1. This also allows for a check to see if there are any points where the velocity data cannot be adjusted to machine limitations so that the proper input modifications can be made prior to actually processing a tape.

In selecting the variable EPS, care must be taken to choose a value small enough to avoid certain undesirable numerical results. One of these can occur if a curve section has a portion where the table rotation passes through 90 deg. If the point mesh is fine, there is a danger of points near but not coincidental with this 90 deg. point also being assigned a 90 deg. value of ω , unless the value of EPS is very small. Values of 10^{-6} or less would be sufficient in most cases.

Modifications

Some modifications in CUPID as it now stands seem either necessary or desirable. For example it will be necessary to make a few slight changes in order to allow for computing velocity data for the constant surface speed condition. These would probably consist of an added input parameter to denote the machining condition and a division by $\sqrt{K^2 + r^2 \sin^2 \theta}$ if the constant tool surface speed is requested.

Testing procedures for the previously mentioned unchecked restrictions could also be added profitably along with some modifications in the input data format to make possible errors more difficult. In this latter area a method probably could be designed for determining the optimum number of data points into which a curve section should be divided. This would eliminate the possibility of making the data intervals too large, thereby exceeding the tape instruction array limit.

If additional flexibility in the number of curve types available is desired, it can be obtained readily by programming the necessary data computations in a subroutine, adding an extra numbered call statement to the list following

statement 51 in the main program and an extra alternative to the GOTO statement following statement 90.

Other modifications may be required as the numerical control unit undergoes further development and the program experiences more extended use.

Fortran Variables

A_1 $u_1^2 - u_0^2$ for ram or table data

A_2 $\frac{2P}{u_1^2 - u_0^2}$

A_3 $u_0^2 + \frac{K_p(u_1^2 - u_0^2)}{P}$

ACEL* - maximum allowable acceleration

ADIST - number of pulses in interval for δ and ω .

AMEL** - computed interval acceleration.

ARAD - cutter head radius (a)

A - cubic coefficient array - X^3 term

BAR - cutter arm length (b)

B - cubic coefficient array - X^2 term

CDETR - pulse count array

CHIP - chip size

COMEG - $\cos \omega$

COSUM - $\cos (\omega + \theta)$

C - cubic coefficient array - X term

CSIGN - ± 1 to designate inside or outside contour for circle.

CTAU - $\cos \tau$

DELD - pulse increment vector

DEL - δ - ω data vector

DIFT - tape distance remainder vector

* In ADJUST this becomes AMAX

** In ADJUST this becomes ACEL

DECLASSIFIED

-79-

HW-78712

DIST - tape interval distance

DLIM - round-off parameter

DOT - tape code vector

DRAD - radius of circle

DRATE* - coordinate velocity array

DSPEC - auxiliary time vector

DUMMY - coordinate velocity conversion parameter

DY₁ - point defined contour beginning derivative

DY₂ - point defined contour end derivative

EPS - comparison constant

HEAD - heading array

IBEG - number of reflected points for point defined contour.

IBUG - word distance for instruction from array start

ICURV - curve type designation

IMOP - dummy fill-in variable for pt. defined contour

IPOS - tape space instruction distance from interval beginning in number of tape lines.

IREV - maximum pulsing speed

ISPEC - auxiliary function type specification

ISTART - reflected curve end point

ITEL - auxiliary specification number.

ITEP - number of remaining auxiliaries from last interval

ITOT - total number of contour points in a tape interval

JCON - print-out determination parameter

JCRIT - number of memory words corresponding to 1/2 sec.

JSTART - beginning of reflected curve data

KDET - curve section end point data determination

* In ADJUST this becomes PRATE

DECLASSIFIED

-80-

HW-78712

KLIM - number of allowable shift spaces for auxiliaries

KNUM - tape code count

KPOS - rounded off number of pulses in an interval (with remainder from previous interval).

KREC - number of completed tape arrays

KREV - desired curve section adjusted end point velocity

LDET - maximum number of back-off points for data adjusting.

LPOS - rounded-off pulse distance in tape lines

M₁ - specifies if curve section contains δ motion.

M₂ - specifies if curve section contains ω motion.

MC - number of extended points

MDETR - number of instruction - filled words in tape array.

MDUM - total number of auxiliaries to be distributed in the curve section.

MX - selects coordinate whose last pulse is in the lowest memory array position.

MOUNT - determines if master and control tapes are to be written.

MPOS - vector specifying the position of the last pulse of the interval within the memory word for each coordinate.

NDUM - number of new auxiliaries specified for the interval

NINT - number of divisions between each given point for a point defined contour.

NMAX - total number of data points for the point defined curve

NPOS - specifies position within the word for each instruction.

NTYP - specifies instruction type.

PADD - auxiliary pulse spacing for multi-instruction type.

PCEL - computed interval acceleration vector.

PHI - angular coordinate specification for linear circle or angular rotation for linearly related motion.

Q - array of matrix inversion for point defined curve

RAT - Y_1/Y_i

RDELL - $YDELL(I)/XDELL(I)$

RDI - curve section end point velocity vector.

REFER - pulse remainder with pulse sign

RHO - distance specification for a line or circle or ram motion for linearly related motions.

RINT - NINT in floating point form

RIST - number of pulses in interval with remainder

RMASK - tape masking array for instruction distribution

SDIST - pulse remainder

SISUM - $\sin(\omega + \theta)$

SKURV - curvature at a point

SOMEG - $\sin \omega$

SPEED - tape speed

SPOL - $1 + a K$

S - array of matrix inversion for point defined curve.

STAU - $\sin \tau$

TAL - $u_1 + u_0$

TAPE - memory instruction array

THET₁ - beginning point angle for line or circular arc

THET₂ - end point angle for line or circular arc

TIME - interval time estimates

TIST - pulse distance

TRAVEL - spindle speed

T - array of matrix inversion for point defined curve.

VELR - corrected velocity vector when acceleration exceeds maximum

VIST - interval pulse count storage vector

WIST - pulse number minus pulse remainder of the last interval.

DECLASSIFIED

-82-

HW-78712

XDELL - $\theta_1 - \theta_2$ array for original given points
XFILL - θ_1 fill-in array
X - originally given θ array for point defined contour.
XS - θ coordinate array for extended points
YDELL - $r_1 - r_0$ array for original given points
YFILL - r_1 fill-in array
YPR₂ - r''_1 fill-in array
YPR - r''_1 fill-in array
Y - original given r array for point defined contour.
YS - r coordinate array for extended points

EXAMPLE

As a final example to demonstrate the flexibility of CUPID and provide a check-out procedure for the program, consider Figure X where the cutter head is machining an inside contour. The dimensions of the curve are given, the sections being a circle, a line and a circle defined by 5 deg. points between 0 and 30 degrees. The linearly related motion curve section is used to return the cutter arm to its original position.

FIGURE X

SAMPLE CONTOUR

Besides the curve dimensions, other input parameters include

Cutter head radius (a)	-	.01 in.
Ram pulse increment	-	.001 in.
Table pulse increment	-	.01 deg.
Tape reader speed	-	15 in/sec
Spindle speed	-	300 rev/min.
Chip size	-	.06 in/rev.
Max. acceleration	-	200 pulses/sec. ²
Max. velocity	-	1000 pulses/sec.

Table II belows shows the input data in the manner in which it is punched on the cards, the numbers above the arrows at the top indicating the position of every 10th column. The arrows at the sides show the beginning of each new curve section and the location of the instruction codes.

<u>TABLE II</u>		<u>INPUT DATA FOR SAMPLE</u>							
		10	20	30	40	50	60	70	80
		↓	↓	↓	↓	↓	↓	↓	↓
		EXAMPLE FOR CUPID REPORT - INCLUDES ALL 5 CURVE SECTION TYPES-INSIDE CONTOUR							
		4990000+0-1000000-2 1000000-3 1000000-2 1500000+1 3000000+2							
		2500000-1 7 1							
Instruction codes	→	40404040404020202020201010101010100404040404041414141414141020202020202010101010101							
Blank Tape	→	5	10000-6	60000-2	20000+2				
		5							
Circle	→	2	10000-6	60000-2	20000+2				
		0000000+0 0000000+0 5000000+0 9000000+1 6000000+1 10 50 1000 20							
Line	→	1	10000-6	60000-2	20000+2				
		5000000+0 6000000+1 6000000+1 3000000+1 50 1000 20							
Point Defined	→	4	10000-6	60000-2	20000+2				
		-5817764-2 7 5 1000 2 20							
		30000000 5773503 25000000 6042052 20000000 6264617 15000000 6430505							
		10000000 6565385 5000000 6641298 0000000 6666667							
Linear Motion	→	3	10000-6	60000-2	20000+2				
Return to Start		-1666667+0 9000000+1 100 1000 2 30							
Blank Tape	→	5	1	10000-6	60000-2	20000+2			
With Auxiliary		1 5							
		5							

-84-

RECEIVED

HM-78712

DECLASSIFIED

Finally, the output data for the above input is shown below where the heading and tape code is printed out first and then the data for each curve section as it is encountered. Each curve section data layout in which actual coordinate motion occurs contains at least two tables labeled "VELOCITY VECTOR", where the three columns are $\dot{\delta}$ in pulses per tape space, $\dot{\omega}$ in pulses per tape space, and the time in tape spaces required to traverse the data interval, respectively. This last quantity is correct only in the last of these tables, which gives the final modified values of all quantities. The first table gives uncorrected values of $\dot{\delta}$ and $\dot{\omega}$. At points where back-up occurs, this is indicated with a statement for each back-up operation. *The table with heading DELTA - @MEGA - X - Y gives δ , ω , θ , r respectively for the interval data points. When a tape is completed, the pulse count vector is written out where the first four elements are the forward ram pulses, ccw table pulses, backward ram pulses, and cw table pulses, respectively. The remaining elements refer to the auxiliary function data.

* This is not true where linear related motion is concerned. Here the relative position from a zero starting point is given. Also, all θ values in the table are given in radians.

DECLASSIFIED

HW-78712

1. 1971 FID CUPID REPORT - 10/10/71 5 CURVE SQUARED TYPE 5-LABOR CC4000

2. 1971 FID CUPID REPORT - 10/10/71 5 CURVE SQUARED TYPE 5-LABOR CC4000

3. 1971 FID CUPID REPORT - 10/10/71 5 CURVE SQUARED TYPE 5-LABOR CC4000

4. 1971 FID CUPID REPORT - 10/10/71 5 CURVE SQUARED TYPE 5-LABOR CC4000

5. 1971 FID CUPID REPORT - 10/10/71 5 CURVE SQUARED TYPE 5-LABOR CC4000

6. 1971 FID CUPID REPORT - 10/10/71 5 CURVE SQUARED TYPE 5-LABOR CC4000

DECLASSIFIED

HW-78712

0:0:0:0:

00100100100
00100100100
00100100100
00100100100
00100100100

0:0:0:0:

[illegible][illegible][illegible]

257991-1

1990

0.11459155E-01	0.11459155E-01	0.11459155E-01
0.11459155E-01	0.11459155E-01	0.11459155E-01
0.11459155E-01	0.11459155E-01	0.11459155E-01

	OMEGA	X	Y
0.	0.89999999E 02	0.15707963E 01	0.49999999E 01
0.	0.89399999E 02	0.15603243E 01	0.49999999E 01
0.	0.88799999E 02	0.15498523E 01	0.49999999E 01
0.	0.88199998E 02	0.15393803E 01	0.49999999E 01
0.	0.87599998E 02	0.15289084E 01	0.49999999E 01
0.	0.86999999E 02	0.15184364E 01	0.49999999E 01
0.	0.86399999E 02	0.15079644E 01	0.49999999E 01
0.	0.85799999E 02	0.14974925E 01	0.49999999E 01
0.	0.85199998E 02	0.14870205E 01	0.49999999E 01
0.	0.84599998E 02	0.14765485E 01	0.49999999E 01
0.	0.83999999E 02	0.14660765E 01	0.49999999E 01
0.	0.83399998E 02	0.14556046E 01	0.49999999E 01
0.	0.82799999E 02	0.14451326E 01	0.49999999E 01
0.	0.82199997E 02	0.14346606E 01	0.49999999E 01
0.	0.81599996E 02	0.14241886E 01	0.49999999E 01
0.	0.80999999E 02	0.14137167E 01	0.49999999E 01
0.	0.80399998E 02	0.14032447E 01	0.49999999E 01
0.	0.79799999E 02	0.13927727E 01	0.49999999E 01
0.	0.79199997E 02	0.13823007E 01	0.49999999E 01
0.	0.78599998E 02	0.13718288E 01	0.49999999E 01
0.	0.77999999E 02	0.13613568E 01	0.49999999E 01
0.	0.77399996E 02	0.13508848E 01	0.49999999E 01
0.	0.76799999E 02	0.13404128E 01	0.49999999E 01
0.	0.76199998E 02	0.13299409E 01	0.49999999E 01
0.	0.75599998E 02	0.13194688E 01	0.49999999E 01
0.	0.74999999E 02	0.13089969E 01	0.49999999E 01
0.	0.74399998E 02	0.12985249E 01	0.49999999E 01
0.	0.73799999E 02	0.12880529E 01	0.49999999E 01
0.	0.73199999E 02	0.12775810E 01	0.49999999E 01
0.	0.72599998E 02	0.12671090E 01	0.49999999E 01
0.	0.71999998E 02	0.12566371E 01	0.49999999E 01
0.	0.71399997E 02	0.12461650E 01	0.49999999E 01
0.	0.70799998E 02	0.12356931E 01	0.49999999E 01
0.	0.70199998E 02	0.12252211E 01	0.49999999E 01
0.	0.69599998E 02	0.12147491E 01	0.49999999E 01
0.	0.68999999E 02	0.12042771E 01	0.49999999E 01
0.	0.68399999E 02	0.11938052E 01	0.49999999E 01
0.	0.67799998E 02	0.11833332E 01	0.49999999E 01
0.	0.67199999E 02	0.11728612E 01	0.49999999E 01
0.	0.66599998E 02	0.11623892E 01	0.49999999E 01
0.	0.65999999E 02	0.11519173E 01	0.49999999E 01
0.	0.65399999E 02	0.11414453E 01	0.49999999E 01

0.
0.
0.
0.
0.
0.
0.
0.
0.
0.

0.61709999 02
0.61109999 02
0.63599999 02
0.62999999 02
0.62399999 02
0.61799999 02
0.61199999 02
0.60599999 02
0.59999999 02

0.11509733 01
0.11205011 01
0.11100230 01
0.10995570 01
0.10892050 01
0.10786130 01
0.10681410 01
0.10576690 01
0.10471970 01

0.40099970 01
0.40099970 01
0.40099970 01
0.40099990 01
0.40099990 01
0.40099990 01
0.40099990 01
0.40099990 01
0.40099990 01

-93-
DECLASSIFIED

HW-78712

AIM

Association for Information and Image Management

1100 Wayne Avenue, Suite 1100
Silver Spring, Maryland 20910
301/587-8202

Centimeter

Inches

MANUFACTURED TO AIM STANDARDS
BY APPLIED IMAGE, INC.

2 of 2

0.11075441E-01	0.2359829E-03
-0.11075441E-01	0.2359829E-03
-0.10977752E-02	0.73858750E-03
-0.10738505E-02	0.62265801E-03
-0.10577947E-02	0.55078381E-03
-0.10419340E-02	0.52359912E-03
-0.10272944E-02	0.52359829E-03
-0.10131489E-02	0.52359995E-03
-0.99971639E-01	0.52359829E-03
-0.98656109E-01	0.52359829E-03
-0.97454159E-01	0.52359912E-03
-0.96272060E-01	0.52359829E-03
-0.95136426E-01	0.52359995E-03
-0.94044173E-01	0.52359829E-03
-0.92992508E-01	0.52359829E-03
-0.91978868E-01	0.52359912E-03
-0.91000913E-01	0.52359829E-03
-0.90056494E-01	0.52359995E-03
-0.89143636E-01	0.52359829E-03
-0.88260530E-01	0.52359829E-03
-0.87405493E-01	0.52359912E-03
-0.86576978E-01	0.52359829E-03
-0.85773558E-01	0.52359995E-03
-0.84993900E-01	0.52359829E-03
-0.84236778E-01	0.52359829E-03
-0.83501037E-01	0.52359912E-03
-0.82785616E-01	0.52359912E-03
-0.82389523E-01	0.52359829E-03
-0.81411826E-01	0.52359912E-03
-0.80751663E-01	0.52359829E-03
-0.80108222E-01	0.52359912E-03
-0.79480743E-01	0.52359912E-03
-0.78868516E-01	0.52359829E-03
-0.78270876E-01	0.52359912E-03
-0.77687195E-01	0.52359829E-03
-0.77116803E-01	0.52359912E-03
-0.76559392E-01	0.52359912E-03
-0.76014201E-01	0.52359829E-03
-0.75480817E-01	0.52359912E-03
-0.74958780E-01	0.52359829E-03
-0.74447650E-01	0.52359912E-03
-0.73947036E-01	0.52359912E-03
-0.73456533E-01	0.52359829E-03
-0.72975782E-01	0.52359912E-03
-0.72504441E-01	0.52359829E-03

SECRET

96-
JUL 19 1964

15-129182 ZOI *
15-129183 ZOI *
15-129184 ZOI *
15-129185 ZOI *
15-129186 ZOI *

0.01459153E-00	0.011279553E-00	0.52326853E 03
0.011688441E-00	-0.11688441E-00	0.52516726E 03
0.10937732E-00	-0.10937732E-00	0.52478161E 03
0.10736505E-00	-0.10736505E-00	0.52484918E 03
0.10573247E-00	-0.10573247E-00	0.52515613E 03
0.10419340E-00	-0.10419340E-00	0.52565541E 03
0.10272044E-00	-0.10272044E-00	0.52628402E 03
0.10131489E-00	-0.10131489E-00	0.52705297E 03
0.99971639E-01	-0.99971639E-01	0.52795536E 03
0.98686109E-01	-0.98686109E-01	0.52896631E 03
0.97454159E-01	-0.97454159E-01	0.53010882E 03
0.96272060E-01	-0.96272060E-01	0.53138268E 03
0.95136426E-01	-0.95136426E-01	0.53277975E 03
0.94044173E-01	-0.94044173E-01	0.53429901E 03
0.92992508E-01	-0.92992508E-01	0.53594123E 03
0.91978868E-01	-0.91978868E-01	0.53771412E 03
0.91000913E-01	-0.91000913E-01	0.53961794E 03
0.90056494E-01	-0.90056494E-01	0.54165135E 03
0.89143636E-01	-0.89143636E-01	0.54381012E 03
0.88260530E-01	-0.88260530E-01	0.54611113E 03
0.87405493E-01	-0.87405493E-01	0.54854041E 03
0.86576978E-01	-0.86576978E-01	0.55110931E 03
0.85773558E-01	-0.85773558E-01	0.55382318E 03
0.84993900E-01	-0.84993900E-01	0.55667622E 03
0.84236778E-01	-0.84236778E-01	0.55968241E 03
0.83501037E-01	-0.83501037E-01	0.56282170E 03
0.82785616E-01	-0.82785616E-01	0.56612273E 03
0.82089523E-01	-0.82089523E-01	0.56957424E 03
0.81411826E-01	-0.81411826E-01	0.57318696E 03
0.80751663E-01	-0.80751663E-01	0.57696514E 03
0.80108222E-01	-0.80108222E-01	0.58089837E 03
0.79480743E-01	-0.79480743E-01	0.58500431E 03
0.78868516E-01	-0.78868516E-01	0.58928615E 03
0.78270876E-01	-0.78270876E-01	0.59374822E 03
0.77687195E-01	-0.77687195E-01	0.59838918E 03
0.77116883E-01	-0.77116883E-01	0.60321909E 03
0.76559392E-01	-0.76559392E-01	0.60824069E 03
0.76014201E-01	-0.76014201E-01	0.61346181E 03
0.75480817E-01	-0.75480817E-01	0.61888532E 03
0.74958780E-01	-0.74958780E-01	0.62451959E 03
0.74447656E-01	-0.74447656E-01	0.63036815E 03
0.73947036E-01	-0.73947036E-01	0.63644372E 03
0.73456533E-01	-0.73456533E-01	0.64274564E 03
0.72975782E-01	-0.72975782E-01	0.64928420E 03
0.72504441E-01	-0.72504441E-01	0.65607186E 03

100

2.T191-K11

0.50000000E-01
0.50000000E-01
0.51000000E-01

-0.71500000E-01
-0.71143723E-01
-0.70706957E-01
-0.70278154E-01

0.67041051E-03
0.67797975E-03
0.68583344E-03
0.69397371E-03

BELTA

OMEGA

X

Y

-0.29802322E-07
0.53966045E-03
0.21219840E-02
0.46962202E-02
0.82166791E-02
0.12642413E-01
0.17936796E-01
0.24066538E-01
0.31001866E-01
0.38715059E-01
0.47184438E-01
0.56385636E-01
0.66300194E-01
0.76910615E-01
0.88201314E-01
0.10015839E-00
0.11276972E-00
0.12602466E-00
0.13991392E-00
0.15442935E-00
0.16956446E-00
0.18521349E-00
0.20167208E-00
0.21863699E-00
0.23620579E-00
0.25437737E-00
0.27315033E-00
0.29252705E-00
0.31250730E-00
0.33309400E-00
0.35429346E-00
0.37610030E-00
0.39852852E-00
0.42158098E-00
0.44526401E-00
0.46958512E-00
0.49455746E-00
0.52017504E-00
0.54646730E-00
0.57342646E-00
0.60107733E-00
0.62932800E-00

0.59999999E-02
0.59404123E-02
0.58818430E-02
0.58242212E-02
0.57674842E-02
0.57115727E-02
0.56564327E-02
0.56020136E-02
0.55482689E-02
0.54951546E-02
0.54426304E-02
0.53906567E-02
0.53391984E-02
0.52882210E-02
0.52376918E-02
0.51875806E-02
0.51378579E-02
0.50884959E-02
0.50394680E-02
0.49907487E-02
0.49423137E-02
0.48941390E-02
0.48462020E-02
0.47984811E-02
0.47509541E-02
0.47036014E-02
0.46564018E-02
0.46093361E-02
0.45623852E-02
0.45155299E-02
0.44687522E-02
0.44220337E-02
0.43753561E-02
0.43287022E-02
0.42820544E-02
0.42353951E-02
0.41887072E-02
0.41419735E-02
0.40951768E-02
0.40482999E-02
0.40013260E-02
0.39542333E-02

0.10471975E-01
0.10367255E-01
0.10262536E-01
0.10157815E-01
0.10053096E-01
0.99483767E-00
0.98436569E-00
0.97389372E-00
0.96342175E-00
0.95294977E-00
0.94247779E-00
0.93200582E-00
0.92153385E-00
0.91106187E-00
0.90058989E-00
0.89011791E-00
0.87964594E-00
0.86917397E-00
0.85870199E-00
0.84823001E-00
0.83775804E-00
0.82728606E-00
0.81681409E-00
0.80634211E-00
0.79587014E-00
0.78539816E-00
0.77492619E-00
0.76445422E-00
0.75398224E-00
0.74351026E-00
0.73303828E-00
0.72256631E-00
0.71209434E-00
0.70162236E-00
0.69115038E-00
0.68067841E-00
0.67020644E-00
0.65973446E-00
0.64926248E-00
0.63879051E-00
0.62831853E-00
0.61784656E-00

0.49999999E-01
0.50002741E-01
0.50010968E-01
0.50024684E-01
0.50043897E-01
0.50068616E-01
0.50098858E-01
0.50134636E-01
0.50175974E-01
0.50222890E-01
0.50275414E-01
0.50333572E-01
0.50397398E-01
0.50466927E-01
0.50542200E-01
0.50623256E-01
0.50710142E-01
0.50802909E-01
0.50901607E-01
0.51006294E-01
0.51117029E-01
0.51233877E-01
0.51356906E-01
0.51486185E-01
0.51621793E-01
0.51763809E-01
0.51912315E-01
0.52067404E-01
0.52229167E-01
0.52397701E-01
0.52573110E-01
0.52755503E-01
0.52944992E-01
0.53141696E-01
0.53345738E-01
0.53557249E-01
0.53776365E-01
0.54003227E-01
0.54237984E-01
0.54480793E-01
0.54731813E-01
0.54991216E-01

0.70000000 00
0.70000000 00
0.70000000 00
0.70000000 00
0.70000000 00
0.70000000 00
0.70000000 00
0.70000000 00
0.70000000 00
0.70000000 00

0.35000000 02
0.35000000 02
0.35121100 02
0.37043901 02
0.37164608 02
0.36683220 02
0.36199396 02
0.35712983 02
0.35223799 02

0.50727490 00
0.50690241 00
0.50643360 00
0.57595860 00
0.56548668 00
0.55501471 00
0.54454273 00
0.53407076 00
0.52359878 00

0.52091790 01
0.50535880 01
0.50021520 01
0.50116311 01
0.50420445 01
0.50734167 01
0.50957650 01
0.50391192 01
0.50735027 01

[illegible][illegible]

ACILE 10

POINT OF ORDER

DY13 -0.5 1776 -0.1 DY24 -0.

CLARK# 7 NINT# 5 IRCH# -0 MC# -0 IREV#1000 KPEI# 2 KREV# -0

LDEF4 20

INTERPOLATED DATA TABLE

XFILL Jn	YFILL Jn	DY1ZJn	DY2ZJn
30.00000000	5.77355225	-0.58177640E-01	-0.17598468E-02
29.00000000	5.83079781	-0.56409606E-01	-0.17762190E-02
28.00000000	5.88631666	-0.54625201E-01	-0.17925911E-02
27.00000000	5.94034215	-0.52824423E-01	-0.18089633E-02
26.00000000	5.99196015	-0.51007275E-01	-0.18253355E-02
25.00000000	6.04204117	-0.49173753E-01	-0.18417049E-02
24.00000000	6.09031220	-0.47325269E-01	-0.18552628E-02
23.00000000	6.13661785	-0.45463227E-01	-0.18688208E-02
22.00000000	6.18122657	-0.43587627E-01	-0.18823788E-02
21.00000000	6.22333567	-0.41698469E-01	-0.18959367E-02
20.00000000	6.26441667	-0.39795753E-01	-0.19094919E-02
19.00000000	6.30347610	-0.37880931E-01	-0.19201579E-02
18.00000000	6.34053524	-0.35955448E-01	-0.19308139E-02
17.00000000	6.37553534	-0.34019303E-01	-0.19414750E-02
16.00000000	6.40956113	-0.32072498E-01	-0.19521360E-02
15.00000000	6.43950438	-0.301115031E-01	-0.19627945E-02
14.00000000	6.46363719	-0.28148391E-01	-0.19704851E-02
13.00000000	6.49179211	-0.26174361E-01	-0.19781757E-02
12.00000000	6.52094266	-0.24192040E-01	-0.19858663E-02
11.00000000	6.54411233	-0.22202328E-01	-0.19935569E-02
10.00000000	6.56533493	-0.20204926E-01	-0.20012452E-02
9.00000000	6.58458267	-0.18201376E-01	-0.20058557E-02
8.00000000	6.60173370	-0.16193215E-01	-0.20104662E-02
7.00000000	6.61691335	-0.14180443E-01	-0.20150775E-02
6.00000000	6.63014538	-0.12163061E-01	-0.20196872E-02
5.00000000	6.64129402	-0.10141068E-01	-0.20242972E-02
4.00000000	6.65042663	-0.81159875E-02	-0.20258638E-02
3.00000000	6.65752941	-0.60893405E-02	-0.20274333E-02
2.00000000	6.66260461	-0.40611269E-02	-0.20289969E-02
1.00000000	6.66568113	-0.20313468E-02	-0.20305614E-02
0.	6.66568113	-0.11641532E-02	-0.20321299E-02

-100-

10

NY-78722

WET-100

0.20000000-01	-0.70067270E-01	0.
0.20000000-01	-0.73511540E-01	0.52826853E-03
0.20000000-01	-0.76779117E-01	0.52815723E-03
0.20000000-01	-0.79873111E-01	0.52478141E-03
0.20000000-01	-0.82797439E-01	0.52484918E-03
0.20000000-01	-0.85556617E-01	0.52515610E-03
0.20000000-01	-0.88154697E-01	0.52565541E-03
0.20000000-01	-0.90595894E-01	0.52628402E-03
0.20000000-01	-0.92885913E-01	0.52705297E-03
0.20000000-01	-0.95030537E-01	0.52795336E-03
0.20000000-01	-0.97035527E-01	0.52896631E-03
0.20000000-01	-0.98905836E-01	0.53010882E-03
0.24280136E-01	-0.10064604E-00	0.53138268E-03
0.20533764E-01	-0.10226171E-00	0.53277975E-03
0.20941098E-01	-0.10375813E-00	0.53429901E-03
0.10357044E-01	-0.10514031E-00	0.53594123E-03
0.17828901E-01	-0.10641241E-00	0.53771412E-03
0.10351746E-01	-0.10757836E-00	0.53961794E-03
0.10710000E-01	-0.10864153E-00	0.54165135E-03
0.10520103E-01	-0.10960670E-00	0.54381012E-03
0.10182704E-01	-0.11047710E-00	0.54611113E-03
0.10567104E-01	-0.11125566E-00	0.54854041E-03
0.00332004E-02	-0.11194462E-00	0.55110931E-03
0.03270717E-02	-0.11254668E-00	0.55382318E-03
0.70970100E-02	-0.11306418E-00	0.55667622E-03
0.50800000E-02	-0.11349903E-00	0.55968241E-03
0.40000000E-02	-0.11385285E-00	0.56282170E-03
0.30010000E-02	-0.11412657E-00	0.56612273E-03
0.20200000E-02	-0.11432133E-00	0.56957424E-03
0.10000000E-02	-0.11443785E-00	0.57318695E-03
0.00000000E-00	-0.11447662E-00	0.57696514E-03

BACK-UP AT 10:30

BACK-UP AT 10:30

101-100100100

HW-70712

0.76778154E-01	-0.76778154E-01	0.
0.76778154E-01	-0.76778154E-01	0.11629840E 01
0.76778154E-01	-0.76778154E-01	0.11637770E 01
0.76778154E-01	-0.76778154E-01	0.11637650E 04
0.76778154E-01	-0.76778154E-01	0.11637520E 04
0.76778154E-01	-0.76778154E-01	0.11637421E 04
0.76778154E-01	-0.76778154E-01	0.11637272E 04
0.76778154E-01	-0.76778154E-01	0.11637173E 04
0.76778154E-01	-0.76778154E-01	0.11637095E 04
0.76778154E-01	-0.76778154E-01	0.11636998E 04
0.76778154E-01	-0.76778154E-01	0.11636924E 04
0.76778154E-01	-0.76778154E-01	0.11636823E 04
0.76778154E-01	-0.76778154E-01	0.11632794E 04
0.76778154E-01	-0.76778154E-01	0.11632834E 04
0.76778154E-01	-0.76778154E-01	0.11632807E 04
0.76778154E-01	-0.76778154E-01	0.11632872E 04
0.76778154E-01	-0.76778154E-01	0.11632680E 04
0.76778154E-01	-0.76778154E-01	0.11632738E 04
0.76778154E-01	-0.76778154E-01	0.11632741E 04
0.76778154E-01	-0.76778154E-01	0.11632868E 04
0.76778154E-01	-0.76778154E-01	0.11632824E 04
0.76778154E-01	-0.76778154E-01	0.11632647E 04
0.76778154E-01	-0.76778154E-01	0.11632735E 04
0.76778154E-01	-0.76778154E-01	0.11632817E 04
0.76778154E-01	-0.76778154E-01	0.11632761E 04
0.76778154E-01	-0.76778154E-01	0.11632925E 04
0.76778154E-01	-0.76778154E-01	0.11632743E 04
0.76778154E-01	-0.76778154E-01	0.11632456E 04
0.76778154E-01	-0.76778154E-01	0.13392258E 04
0.76778154E-01	-0.76778154E-01	0.22361176E 04
0.76778154E-01	-0.76778154E-01	0.76819920E 04

DELTA	OMEGA	X	Y
0.91864298E 00	0.35223801E 02	0.52359878E 00	0.57735029E 01
0.97654075E 00	0.34388322E 02	0.50614543E 00	0.58307979E 01
0.10312903E 01	0.33513799E 02	0.48869219E 00	0.58863166E 01
0.10330000E 01	0.32602266E 02	0.47123890E 00	0.59400428E 01
0.11310000E 01	0.31655724E 02	0.45378561E 00	0.59919600E 01
0.11770000E 01	0.30676121E 02	0.43633232E 00	0.60420519E 01
0.12211110E 01	0.29665358E 02	0.41887902E 00	0.60903025E 01
0.12610000E 01	0.28625281E 02	0.40142573E 00	0.61366979E 01
0.13000000E 01	0.27557684E 02	0.38397244E 00	0.61812244E 01
0.13350000E 01	0.26464292E 02	0.36651915E 00	0.62230686E 01
0.13600000E 01	0.25346763E 02	0.34906585E 00	0.62646169E 01

103-
100-789722

V.11

-0.61723451-01	0.3333333E-00	0.
-0.61723451-01	0.3333333E-00	0.11620363E 04
-0.61723451-01	0.3333333E-00	0.11637770E 04
-0.61723451-01	0.3333333E-00	0.11637655E 04
-0.61723451-01	0.3333333E-00	0.11637559E 04
-0.61723451-01	0.3333333E-00	0.11637421E 04
-0.61723451-01	0.3333333E-00	0.11637272E 04
-0.61723451-01	0.3333333E-00	0.11637178E 04
-0.61723451-01	0.3333333E-00	0.11637095E 04
-0.61723451-01	0.3333333E-00	0.11636998E 04
-0.61723451-01	0.3333333E-00	0.11636924E 04
-0.61723451-01	0.3333333E-00	0.11636823E 04
-0.61723451-01	0.3333333E-00	0.11632794E 04
-0.61723451-01	0.3333333E-00	0.11632834E 04
-0.61723451-01	0.3333333E-00	0.11632807E 04
-0.61723451-01	0.3333333E-00	0.11632872E 04
-0.61723451-01	0.3333333E-00	0.11632680E 04
-0.61723451-01	0.3333333E-00	0.11632738E 04
-0.61723451-01	0.3333333E-00	0.11632741E 04
-0.61723451-01	0.3333333E-00	0.11632863E 04
-0.61723451-01	0.3333333E-00	0.11632824E 04
-0.61723451-01	0.3333333E-00	0.11632647E 04
-0.61723451-01	0.3333333E-00	0.11632735E 04
-0.61723451-01	0.3333333E-00	0.11632817E 04
-0.61723451-01	0.3333333E-00	0.11632761E 04
-0.61723451-01	0.3333333E-00	0.11632925E 04
-0.61723451-01	0.3333333E-00	0.11632740E 04
-0.61723451-01	0.3333333E-00	0.11632466E 04
-0.61723451-01	0.3333333E-00	0.13392258E 04
-0.61723451-01	0.3333333E-00	0.22361176E 04
-0.61723451-01	0.3333333E-00	0.76819925E 04
-0.61723451-01	0.3333333E-00	0.58009837E 03
-0.61723451-01	0.3333333E-00	0.58500431E 03
-0.61723451-01	0.3333333E-00	0.58928615E 03
-0.61723451-01	0.3333333E-00	0.59374822E 03
-0.61723451-01	0.3333333E-00	0.59838918E 03
-0.61723451-01	0.3333333E-00	0.60321909E 03
-0.61723451-01	0.3333333E-00	0.60824069E 03
-0.61723451-01	0.3333333E-00	0.61346181E 03
-0.61723451-01	0.3333333E-00	0.61888532E 03
-0.61723451-01	0.3333333E-00	0.62451959E 03
-0.61723451-01	0.3333333E-00	0.63036815E 03
-0.61723451-01	0.3333333E-00	0.63644372E 03
-0.61723451-01	0.3333333E-00	0.64274564E 03
-0.61723451-01	0.3333333E-00	0.64928420E 03
-0.61723451-01	0.3333333E-00	0.65607186E 03

-105-

SECRET

REF ID: A66712

-106-
HV-7872

RECEIVED

BACK-UP AT L1101

BACK-UP AT L1101

BACK-UP AT L1101

BACK-UP AT L1101

BACK-UP AT L1101

BACK-UP AT L1101

BACK-UP AT L1101

BACK-UP AT L1101

BACK-UP AT L1101

BACK-UP AT L1101

BACK-UP AT L1101

BACK-UP AT L1101

DECLASSIFIED

HW-78712

DECLASSIFIED
BACI 10-20-1990

[illegible][illegible][illegible]

100

HW-78712

-11-

[REDACTED]

Condition	Control (%)	MCI (%)	AD (%)
A	~95	~85	~75
B	~90	~80	~70
C	~85	~75	~65
D	~95	~90	~85

111

• • •

9

END OF DATA

PULSE 1000000

9000.

1000.

9000.

1.

0.

COMPLETE LISTING OF FORTRAN PROGRAM CUPID

SECRET

C THIS PROGRAM PRODUCES A C TITOCUS FEED, MAGNETIC NUMERICAL CONTROL
 C TAPE FOR A TWO-DIMENSIONAL CONTOUR LATHE WITH DELTA-OMEGA COORDINATE
 C GEOMETRY WHOSE MOVEMENTS ARE GOVERNED BY PULSE-INCREMENT DEVICES.

1 FORMAT(8E10.6/E10.6,2I5)
 2 FORMAT(2I5,3E10.4,I5)
 3 FORMAT(15H0END OF PROBLEM)
 4 FORMAT(12H0END OF CASE)
 5 FORMAT(13H0PULSE COUNT-/(1X,6F15.0))
 6 FORMAT(7H1ICURV=I2,3X,5HNDUM=I3,3X,4HEPS=E10.3,3X,6HSPEED=E10.3,3X
 X,7HTRAVEL=E10.3,3X,5HCHIP=E10.3,3X,4HBAR=E14.6/6H0ARAD=E10.3,3X,12
 XH0STEP SIZES=2E10.3,3X,9HST. VEL.=2E10.3,5HDLIM=E10.3/6H0ACEL=E10.
 X3)
 7 FORMAT(13A6,A2)
 24 FORMAT(1H0,9X,5HDELTA,15X,5H0MEGA,18X,1HX,19X,1HY/1H0/(1H ,4E20.8)
 X)
 45 FORMAT(60I2)
 46 FORMAT(1H1,13A6,A2/11H0TAPE CODE-/(1H ,5020))
 49 FORMAT(38H0TAPE VECTOR LIMITS HAVE BEEN EXCEEDED)
 66 FORMAT(1H0/(1H ,5020))
 67 FORMAT(14H0VECTOR NUMBER,I6)
 68 FORMAT(22H0TOTAL NO. OF VECTORS=,I6)
 1100 FORMAT(56H1MOUNT GIVEN TAPES ON A5 AND B6 AS PER CARD INSTRUCTIONS
 X/19H0A5 IS HIGH DENSITY/75H0SET DATA CHANNEL 3 SO THAT LOW DENSITY
 X IS 200 BITS PER INCH---PRESS START////////)
 DIMENSION RMASK(6),DIFT(2),SDIST(2),REFER(2),ADIST(2),VIST(2),MPOS
 X(2),CDETR(64),DELD(2),RDI(2),JMAX(2),DEL(2,1000),XFILL(1000),YFILL
 X(1000),TIME(1000),DRATE(2,1000),DSPEC(100),ISPEC(100),KLIM(100),TA
 XPE(5100),DOT(64),X(60),Y(60),XDELL(60),YDELL(60),RDELL(60),T(60),Q
 X(60),S(60),A(60),B(60),C(60),HEAD(14),XS(10),YS(10),AMEL(2),PCEL(2
 X),VLLR(2),YPR(1000),YPR2(1000)
 COMMON RMASK,DIFT,SDIST,REFER,ADIST,VIST,MPOS,CDETR,DELD,RDI,JMAX,
 XDEL,XFILL,YFILL,TIME,DRATE,DSPEC,ISPEC,KLIM,TAPE,DOT,X,Y,XDELL,YDE
 XLL,RDELL,T,Q,S,A,B,C,HEAD,JOK,MDETR,XREC,ITEL,MDUM,BAR,ARAD,DLIM,
 XKNUM,MOUNT,ICURV,NDUM,EPS,SPEED,TRAVEL,CHIP,ACEL,JCON,DUMMY,M1,M2,
 XITOT,JCRIT,NTYP,RIST,IPOS,DIST,KPOS,TAL,A1,A2,WIST,TIST,A3,LPOS,NP
 XOS,IBUG,JRITE,JOT,MK,KREC,JLT,ITEP,PADD,DY1,DY2,NMAX,NINT,IBEG,MC,
 XIREV,KDET,KREV,LDET,ISTART,JSTART,RINT,XS,YS,RAT,STAU,CTAU,SISUM,
 XCUSUM,SOMEG,COMEG,SKURV,SPOL,IMOP,RHO,PHI,DRAD,THET1,THET2,CSIGN,
 XAMEL,PCEL,VELR,YPR,YPR2

C
 C A INITIALIZE PARAMETERS

C
 B RMASK(1) = 770000000000
 B RMASK(2) = 007700000000
 B RMASK(3) = 000077000000


```

      IF (ICURV)611,611,613
C
C   1 COMPLETE TAPE ON BLANK CARD
C
      611 WRITE OUTPUT TAPE 3,5,(CDLTR(I),I=1,KNUM)
      GO TO (1610,1611),COUNT
      1610 WRITE OUTPUT TAPE 3,4
      GO TO 20
      1611 KREC = KREC + 2
      WRITE OUTPUT TAPE 3,68,KREC
      WRITE TAPE 5,(TAPE(K),K=1,5100)
      610 END FILE 5
C
C   2 SCRUNCH TAPE-REMOVE RECORD GAPS
C
      CALL SQUASH
C
C   3 READ NEW CASE DATA
C
      GO TO 20
C
C   E SET UP PARAMETERS AND WRITE INPUT
C
      613 DUMMY = TRAVEL*CHIP/(1.2E+4*SPEED)
      WRITE OUTPUT TAPE 3,6,ICURV,NDUM,EPS,SPEED,TRAVEL,CHIP,BAR,ARAD,
      X(DEL(J),J=1,2),(RDI(I),I=1,2),DLIM,ACEL
      18 ACEL = ACEL/(200.0*SPEED)**2
      403 M1 = 1
      M2 = 2
C
C   F READ AND SET UP AUXILIARY FUNCTION DATA
C
      CALL AUXIL
      ITEL = ITEL
      ASSIGN 780 TO NGO
      IF(MDUM)90,90,91
      91 ASSIGN 901 TO NGO
C
C   G COMPUTE CURVE DATA
C
      90 ASSIGN 13 TO NCRIT
      GO TO (15,16,400,9,11),ICURV
      11 ASSIGN 14 TO NCRIT
      READ INPUT TAPE 2,2,ITOT
      ITOT = 2*ITOT + 1
      JCRIT = XINTF((200.0*SPEED)/12.0+EPS)
      DO 51 J = 1,2

```



```

      GO TO 52
15 CALL LINE
      GO TO 17
16 CALL CIRCLE
      GO TO 17
40 CALL PMET
      GO TO 17
9 CALL CONTR
C
C 11 ADJUST CURVE DATA
C
      17 CALL ADJUST
245 WRITE OUTPUT TAPE 3,24, ((DEL(K,J),K = 1,2),XFILL(J),YFILL(J),J=1,
      XITOT)
52 ASSIGN 97 TO LWRIT
      IF(JCON-1)53,54,53
54 ASSIGN 95 TO LWRIT
C
C 1 DISTRIBUTE INSTRUCTIONS IN MEMORY
C
      53 DO 800 L = 2,ITOT
          GO TO NCRIT,(13,14)
12 DO 890 M = M1,M2
250 DIST = TIME(L) + DIFT(M)
      85 ADIST(M) = (DEL(M,L) - DEL(M,L-1))/DELD(M)
          RIST=ADIST(M)+ REFER(M)
          NTYP = M
          IF(RIST)720,43,43
720 NTYP = M + 2
C
C 1 LESS THAN ONE PULSE-SAVE DATA
C
      43 IF(ABSF(RIST) - 1.0)730,731,731
730 REFER(M) = RIST
      SDIST(M) = ABSF(RIST)
      DIFT(M) = DIST - INTF(DIST)
      IPOS = XINTF(DIST) + MPOS(M)
      JMAX(M) = 1 + (IPOS - 1)/6
      MPOS(M) = IPOS - 6*(JMAX(M) - 1)
      VIST(M) = RIST
      GO TO 890
C
C 2 ONE PULSE OR MORE
C
731 KPOS = XINTF(ABSF(RIST + SIGNF(DLIM,RIST)))
      TAL = DRATE(M,L-1) + DRATE(M,L)
      A1 = (DRATE(M,L)-DRATE(M,L-1))*TAL

```

```

C      A CHOICE CONSTANT IN THE LINEAR ACCELERATION DISTRIBUTION
C
      IF (ABSF(A1) - EPS) 1200, 1200, 1203
1200 ASSIGN 950 TO MARK
      GO TO 41
1203 ASSIGN 951 TO MARK
      DO 722 K = 1, KPOS
      RIST = FLOATF(K) - SDIST(M)
      GO TO MARK, (950, 951)
950 TIST = WIST*TIME(L)/ABSF(ADIST(M))
      GO TO 952
951 A3 = DRATE(M, L-1)**2 + A1*RIST/ABSF(ADIST(M))
C
C      1 COMPUTE TAPE DISTANCE
C
1215 TIST = A2*(-DRATE(M, L-1) + SIGNF(1.0, TAL)*SQRTF(MAX1F(0.0, A3)))
952 TIST = ABSF(TIST) + DIFT(M)
C
C      2 COMPUTE MEMORY WORD LOCATION AND REMAINDER
C
      LPOS = XINTF(ABSF(TIST) + 0.5)
      IPOS = LPOS + MPOS(M)
      JMAX(M) = 1 + (IPOS - 1)/6
      NPOS = IPOS - 6*(JMAX(M) - 1)
      IPUG = JMAX(M) + MDETR
C
C      3 PLACE INSTRUCTION IN MEMORY ARRAY
C
      TAPE(1BUG) = TAPE(1BUG) + RMASK(NPOS)*DOT(NTYP)
722 CDETR(NTYP) = CDETR(NTYP) + 1.0
C
C      4 SAVE REMAINDER DATA
C
642 DIFT(M) = ABSF(DIST) - FLOATF(LPOS)
      SDIST(M) = ABSF(RIST) - FLOATF(KPOS)
      REFER(M) = SDIST(M)*SIGNF(1.0, RIST)
      MPOS(M) = NPOS
      VIST(M) = RIST
800 CONTINUE
C
C      J COMPUTE MEMORY ARRAY DATA
C
14 IF (M2 - M1) 81, 81, 80
81 M1 = M2
      JRITE = JMAX(M1) - 1
      JOT = JMAX(M1) + MDETR

```

```

      JRITE = XMIN + (JMAX(1) - JMAX(2)) * (JOT - 1)
777 JOT = XMAXPF(JMAX(1), JMAX(2)) + MDETR
      IF (JMAX(1) - JMAX(2)) 771, 772, 773
771 MK = 2
      GO TO 792
772 MK = 1
792 MPUS(MK) = MPUS(MK) + 6 * XABSF(JMAX(1) - JMAX(2))
      IF (JOT - 5100) 785, 784, 784
784 WRITE OUTPUT TAPE 3, 49
      GO TO 411
785 JOK = JOT + 1
C
C K DISTRIBUTE AUXILIARY FUNCTIONS
C
      GO TO NGO, (901, 780)
901 DO 900 K = ITEL, MDUM
900 DSPEC(K) = DSPEC(K) - 6.0 * FLGATF(JRITE)
903 IF (DSPEC(ITEM) + FLGATF(KLIM(ITEM))) 904, 904, 780
904 CALL PLACF
      ITEL = ITEL + 1
      IF (ITEM - MDUM) 903, 903, 907
907 ASSIGN 780 TO NGO
C
C L WRITE MEMORY ARRAY ON TAPE
C
780 IF (MDETR - 2550) 781, 781, 782
782 KREC = KREC + 1
      GO TO KWRIT, (94, 96)
96 WRITE TAPE 5, (TAPE(K), K = 1, 2550)
94 GO TO LWRIT, (95, 97)
95 WRITE OUTPUT TAPE 3, 67, KREC
      WRITE OUTPUT TAPE 3, 66, (TAPE(K), K = 1, 2550)
97 JOT = JOT - 2550
      JOK = JOT + 1
      DO 790 K = 1, JOT
      JFT = 2550 + K
790 TAPE(K) = TAPE(JFT)
      DO 1790 K = JOK, 5100
B1790 TAPE(K) = 000000000000
      MDETR = MDETR - 2550
781 MDETR = JRITE + MDETR
900 CONTINUE
C
C M READ NEW CURVE DATA
C
      GO TO 10
      END

```

C / FILE-THIS SUBROUTINE READS AND PREPARES AUXILIARY FUNCTION DATA

C

```

C      FORMAT(8(F4.0,2I3))
C      FORMAT(12H0AUXILIARIES/1H,4X,5HDSPEC,9X,5HISPEC,6X,4HKLIM/1X/(1X,
XFI10,4,2I10))
      DIMENSION RMASK(6),DIFT(2),SDIST(2),REFER(2),ADIST(2),VIST(2),MPUS
X(2),CDETR(64),DELD(2),RDI(2),JMAX(2),DEL(2,1000),XFILL(1000),YFILL
X(1000),TIME(1000),DRATE(2,1000),DSPEC(100),ISPEC(100),KLIM(100),TA
APE(5100),DOT(64),X(60),Y(60),XDELL(60),YDELL(60),RDELL(60),T(60),Q
X(60),S(60),A(60),B(60),C(60),HEAD(14),XS(10),YS(10),AMEL(2),PCEL(2
X),VELR(2),YPR(1000),YPR2(1000)
      COMMON RMASK,DIFT,SDIST,REFER,ADIST,VIST,MPUS,CDETR,DELD,RDI,JMAX,
XDEL,XFILL,YFILL,TIME,HEAD,DSPEC,ISPEC,KLIM,TAPE,DOT,X,Y,XDELL,YDE
XLL,RDELL,T,Q,S,A,B,C,HEAD,JOK,DETR,KREC,ITEL,MDUM,BAR,ARAD,DLIM,
XKNUM,MOUNT,ICURV,MDUM,EPS,SPEED,TRAVEL,CHIP,ACEL,JCON,DUMMY,M1,M2,
XITUT,JCRIT,INTYP,RIST,IPUS,DIST,KPUS,TAL,A1,A2,MIST,TIST,A3,LPOS,NP
XUS,IBUG,JRITE,JUT,INK,KREC,JET,ITEP,PADD,DY1,DY2,NMAX,NINT,IBEG,MC,
XIREV,KDET,KREV,LDET,ISTART,JSTART,RINT,XS,YS,RAT,STAU,CTAU,SISUM,
XCOSUR,SOMEG,COMEG,SKURV,SPOL,IMCP,RHO,PHI,DRAD,THET1,THET2,CSIGN,
XAMEL,PCEL,VELR,YPR,YPR2

```

C

C A SAVE REMAINING AUXILIARIES

C

```

      ITEP = MDUM - ITEL + 1
      IF(ITEP)22,22,23
23 DO 27 J = 1,ITEP
      ITFV = ITEL - 1 + J
      ISPEC( J) = ISPEC( ITEV)
25 KLIM( J) = KLIM( ITEV)
27 DSPEC(J) = DSPEC(ITEV)
22 ITEL = 1
      MDUM = MDUM + ITEP
      IF(MDUM - ITEP)98,98,90

```

C

C B READ NEW AUXILIARIES

C

```

90 ITEP = ITEP + 1
      READ INPUT TAPE 2,5,(DSPEC(J),ISPEC(J),KLIM(J),J=ITEP,MDUM)
      DO 237 I = ITEP,MDUM
      DSPEC(I) = DSPEC(I)*(200.-*SPEED)
      NUMB = ISPEC(I)-KNUM+2
      IF(NUMB)237,237,238
238 KLIM(I)=2*(NUMB+1)*XINTF(PADD)
237 CONTINUE

```

-123-

RM-78712

```

      DO 22 K = 1,NDUM
      STAN = DSPEC(K)
      DO 23 L = 1,NDUL
      IF(STAN-DSPEC(L))93,93,94
04  JTAN = ISPEC(K)
      KTAN = KLIM(K)
      DSPEC(K) = DSPEC(L)
      ISPEC(K) = ISPEC(L)
42  KLIM(K) = KLIM(L)
      DSPEC(L) = STAN
      ISPEC(L) = JTAN
      KLIM(L) = KTAN
      STAN = DSPEC(K)
03  CONTINUE
02  CONTINUE
01  NDUL = MDUM - 1
      IF(NDUL)99,99,100

```

C
C D CORRECT FOR OVERLAP
C

```

100 DO 95 J = 1,NDUL
      DSP = FLOAT(KLIM(J)) + DSPEC(J)
      IF(NUMB)105,105,106
106 DSP = DSP + 4.0*PADD
105 IF(DSP - DSPEC(J+1))95,97,97
      97 DSPEC(J+1) = DSP
      95 CONTINUE
      99 IF(NDUL)101,102,102
102 WRITE OUTPUT TAPE 3,200,(DSPEC(J),ISPEC(J),KLIM(J),J=1,MDUM)
101 RETURN
      END

```

```

C
C 111-THIS SUBROUTINE CALCULATES POINTS FOR A STRAIGHT LINE CURVE
C SECTION
C
C
C   FORMAT(4E10.7,5I7)
C   FORMAT(5HOLINE/5HDELTA=14.6,3X,4HPHI=E14.6,3X,6HTHE1=E14.6,3X,6HT
XDEL2=E14.6/6HDELTA=15.3X,5HKDEL=12,3X,5HKREV=14,3X,5H
VDEL=14)
56 FORMAT(19HOLINE HAS NO LENGTH)
  DIMENSION KMASK(6),DIFT(2),SDIST(2),REFER(2),ADIST(2),VIST(2),MPOS
X(2),CDETR(64),DELD(2),RDI(2),JMAX(2),DEL(2,1000),XFILL(1000),YFILL
X(1000),TIME(1000),DRATE(2,1000),DSPEC(100),ISPEC(100),KLIM(100),TA
XPE(5100),DOT(64),X(60),Y(60),XDELL(60),YDELL(60),RDELL(60),T(60),Q
X(60),S(60),A(60),B(60),C(60),HEAD(14),XS(10),YS(10),AMEL(2),PCEL(2
X),VELR(2),YPR(1000),YPR2(1000)
  COMMON KMASK,DIFT,SDIST,REFER,ADIST,VIST,MPOS,CDETR,DELD,RDI,JMAX,
XDEL,XFILL,YFILL,TIME,DRATE,DSPEC,ISPEC,KLIM,TAPE,DOT,X,Y,XDELL,YDE
ALL,RDELL,T,Q,S,A,B,C,HEAD,JCK,RDETR,KREC,ITEL,MOUM,BAK,ARAD,DLIM,
XKNUM,MOUNT,ICURV,KOUM,EPS,SPEED,TRAVEL,CHIP,ACEL,JCON,DUMMY,M1,M2,
XITOT,JCRIT,NTYP,KIST,IPOS,DIST,KPOS,TAL,A1,A2,WIST,TIST,A3,LPOS,NP
XUS,IBUG,JRITE,JUT,KK,KREC,JLT,ITEP,PAUD,DY1,DY2,NMAX,NINT,IBEG,MC,
XIREV,KDET,KREV,LDEL,ISTART,JUSTART,RINT,XS,YS,RAT,STAU,CTAU,SISUM,
XCOSUM,SOMEG,COMEG,SKURV,SPOL,IMOP,RHO,PHI,DRAD,THE1,THE2,CSIGN,
XAVEL,PCEL,VELP,YPR,YPR2
C
C A READ AND WRITE INPUT DATA AND INITIALIZE
C
  READ INPUT TAPE 2,3,RHO,PHI,THE1,THE2,ITOT,IREV,KDET,KREV,LDET
  WRITE OUTPUT TAPE 3,4,RHO,PHI,THE1,THE2,ITOT,IREV,KDET,KREV,LDET
  ITOT = ITOT + 1
  SIGP = 1.0
  SIGN = 1.0
  IF(ABS(PHI) - 90.0)15,15,16
C
C B SPECIAL CASES
C
  16 SIGN = -1.0
  15 IF(THE2 - THE1)49,51,50
  51 IF(THE1)54,52,54
C
C 1 DEGENERATE LINE
C
  54 WRITE OUTPUT TAPE 3,56
  GO TO 55
C
C 2 PURE DELTA MOTION = RHO

```

```

60 TOT = RHO/FLOATF(11,J-1)
   IF(TOT)70,71,71
70 SIGP = -1.0
71 N2 = 1
   DIP = SIGP*DUMMY/DELD(1)
   DO 53 J=1,ITOT
     XFILL(J) = TOT*FLOATF(J-1)
     DEL(1,J) = DEL(1,ITOT) + TOT*FLOATF(J-1)
     YFILL(J) = PAR*COMEG
     DEL(2,J) = DEL(2,ITOT)
     DRATE(1,J) = DIP
     DRATE(2,J) = 0.0
53 YPR(J) = 0.0
   GO TO 55

```

```

C
C C GENERAL CASES
C

```

```

40 SIGP = -1.0
50 ASSIGN 433 TO MCURV
   DIP = DUMMY*SIGP
   THET1 = THET1/57.2957795
   THET2 = THET2/57.2957795
   IF(PHI-EPS-90.0)440,430,430
440 IF(PHI+EPS-90.0)430,430,431

```

```

C
C 1 PHI = 90 DEG-SET SWITCH
C

```

```

431 N2 = 1
   ASSIGN 432 TO MCURV
430 PHI = PHI/57.2957795
   TOT = (THET2 - THET1)/(FLOATF(ITOT-1))
   DO 425 J= 1,ITOT

```

```

C
C 2 COMPUTE R-THETA DATA POINTS
C

```

```

   PAR = THET1 +TOT * FLOATF(J-1)
   XFILL(J) = PAR
   YFILL(J) = RHO/COSF(PAR-PHI)
   YPR(J) = YFILL(J)*SINF(PAR-PHI)/COSF(PAR-PHI)

```

```

C
C 3 COMPUTE DELTA-OMEGA DATA POINTS
C

```

```

   GO TO MCURV,(432,433)
432 SOMEQ =(RHO + ARAD)/BAR
   IF(ABSF(SOMEQ)+EPS-1.0)434,434,435
435 COMEG=0.0
   GO TO 436

```

```

446 COMEG = SQRT(1.0 - SOME**2)*SIGN
447 DEL(1,J) = RHO*DEL(PAR)/SINF(PAR) - BAR*COMEG
      DRATE(1,J) = -DIP/DEL(1)
      DRATE(2,J) = 0.0
      GO TO 425
433 SOME = (RHO*SINF(PAR)/COSF(PHI-PAR) + ARAD*SINF(PHI))/BAR
      IF(ABSF(SOME)+EPS-1.0)437,437,438
439 COMEG = 0.0
      GO TO 439
437 COMEG = SQRT(1.0 - SOME**2)*SIGN
439 DEL(1,J) = RHO*COSF(PAR)/COSF(PHI-PAR) + ARAD*COSF(PHI) - BAR*COMEG
      SISUB = SOME*COSF(PHI) - COMEG*SINF(PHI)
      DRATE(1,J) = DIP*SISUB/(DEL(1)*COMEG)
      DRATE(2,J) = 57.2957795*DIP*COSF(PHI)/(BAR*COMEG*DEL(2))
425 DEL(2,J) = ARCINV(SOME*COMEG)
55 RETURN
END

```


ר' יצחק

9

0

•

```

YFILL(J) = 0.0
YPR(J) = 0.0
DRATE(1,J) = 0.0
DRATE(2,J) = 57.2957795*UMAYASIGP/(DRAD*DELD(2))
100 DEL(2,J) = XFILL(J)*57.2957795
GO TO 79

```

C
C GENERAL CASE

```

76 SMUL = SIGP*DUMMY*(1.0 + ARAD/DRAD)
DO 78 J=1,ITOT
PAR = THET1 + TOT*FLOATF(J-1)
TPAR = RHO*COSF(PHI) + DRAD*COSF(PAR)
IF(ABSF(TPAR)-EPS)80,80,81
80 XFILL(J) = 1.57079633
GO TO 82
81 SPAR = (RHO*SINF(PHI) + DRAD*SINF(PAR))/TPAR
XFILL(J) = ARCTAN(SPAR)
82 YFILL(J) = SQRTF(RHO**2 + DRAD**2 + 2.0*RHO*DRAD*COSF(PHI-PAR))
YPR(J) = YFILL(J)*SINF(XFILL(J)-PAR)/COSF(XFILL(J)-PAR)
SOMEG = (RHO*SINF(PHI)+(ARAD+DRAD)*SINF(PAR))/BAR
SIGN = 1.0
IF(ABSF(SOMEG)+EPS-1.0)75,75,72

```

C
C 1 OMEGA = 90 DEG

```

72 COMEG = 0.0
ASP = SQRTF(BAR/(ARAD + DRAD))
DRATE(1,J) = SMUL*(ASP - 1.0)/DELD(1)
DRATE(2,J) = 57.2957795*SMUL*ASP/(BAR*DELD(2))
GO TO 73

```

C
C 2 GENERAL OMEGA

```

75 IF(PAR-1.57079633)71,71,15
15 IF(BAR-ARAD-DRAD-RHO*SINF(PHI)-EPS)16,71,71
16 SIGN = +1.0
71 COMEG = SQRTF(1.0 - SOMEG**2)*SIGN
SISUB = SOMEG*COSF(PAR) - COMEG*SINF(PAR)
DRATE(1,J) = SISUB*SMUL/(DELD(1)*COMEG)
DRATE(2,J) = 57.2957795*SMUL*COSF(PAR)/(BAR*DELD(2)*COMEG)
73 DEL(1,J) = RHO*COSF(PHI) + (ARAD+DRAD)*COSF(PAR) - BAR*COMEG
70 DEL(2,J) = ARCINV(SOMEG,COMEG)
IF(ABSF(DEL(1,ITOT) - DEL(1,1)) - EPS)79,78,78
79 M1=2
78 ARAD = CSIGN*ARAD
RETURN

```

END

401 1 = 2
402 IF (ABS(Phi) - 0.5) 424, 424
403 *2 = 1
404 RETURN
END

-131-

HW-78712

```

C
C C- THIS SUBROUTINE COMPUTES DATA POINTS FOR A POINT DEFINED CURVE,
C WHERE EACH ADJACENT POINT DATA IS FIT WITH A UNIQUE CUBIC
C
C
3 FORMAT(2E10.6,8I5/(8F1.6))
4 FORMAT(14HPOINT DEFINED/4HDY1=E14.6,3X,4HDY2=E14.6,3X,5HNMAY=13,
X3X,5HNINT=13,3X,5HIBEG=13,3X,3HMC=13,3X,5HIREV=14,3X,5HKDET=13,3X,
X5HKREV=13/6HLDDET=13)
5 FORMAT(24HINTERPOLATED DATA TABLE/81H0          XFILL(J)
X   YFILL(J)          DY1(J)          DY2(J) /1H /(1H 2F20
X.8,2E20.8))
6 FORMAT(8F10.6)
   DIMENSION RMASK(6),DIFT(2),SDIST(2),REFER(2),ADIST(2),VIST(2),MPOS
X(2),CDETR(64),DELD(2),RDI(2),JMAX(2),DEL(2,1000),XFILL(1000),YFILL
X(1000),TIME(1000),DRATE(2,1000),DSPEC(100),ISPEC(100),KLIM(100),TA
XPE(5100),DOT(64),X(60),Y(60),XDELL(60),YDELL(60),RDELL(60),T(60),Q
X(60),S(60),A(60),B(60),C(60),HEAD(14),XS(10),YS(10),AMEL(2),PCEL(2
X),VELR(2),YPR(1000),YPR2(1000)
   COMMON RMASK,DIFT,SDIST,REFER,ADIST,VIST,MPOS,CDETR,DELD,RDI,JMAX,
XDEL,XFILL,YFILL,TIME,DRATE,DSPEC,ISPEC,KLIM,TAPE,DOT,X,Y,XDELL,YDE
XLL,RDELL,T,Q,S,A,B,C,HEAD,JOK,HDETR,KREC,ITEL,KOUM,BAR,ARAD,DLIM,
XKNUM,MOUNT,ICURV,NDUM,EPS,SPEED,TRAVEL,CHIP,ACEL,JCON,DUMMY,M1,M2,
XITOT,JCRIT,NTYP,RIST,IPUS,DIST,KPOS,TAL,A1,A2,WIST,TIST,A3,LPOS,NP
XUS,IBUG,JRITE,JOT,MK,KREC,JET,ITEP,PADD,DY1,DY2,NMAX,NINT,IBEG,MC,
XIREV,KDET,KREV,LDET,ISTART,JSTART,RINT,XS,YS,RAT,STAU,CTAU,SISUM,
XCOSUM,SOMEG,COMEG,SKURV,SPOL,IMOP,RHO,PHI,DRAD,THET1,THET2,CSIGN,
XAMEL,PCEL,VELR,YPR,YPR2
C
C A READ AND WRITE INPUT DATA AND INITIALIZE
C
   READ INPUT TAPE 2,3,DY1,DY2,NMAX,NINT,IBEG,MC,IREV,KDET,KREV,LDET,
X(X(J),Y(J),J=1,NMAX)
   WRITE OUTPUT TAPE 3,4,DY1,DY2,NMAX,NINT,IBEG,MC,IREV,KDET,KREV,LDE
XT
   ISTART = (IBEG + MC)*NINT
   JSTART = MC*NINT + 1
   ITOT = ISTART + 1
C
C B COMPUTE CUBIC COEFFICIENTS AND R-THETA POINT DATA
C
   CALL CURVE
   ITOT = ITOT
   IF(ISTART)167,167,166
C
C C COMPUTE R-THETA REFLECTED POINT DATA
C

```

```

      = ISTART + JSTART + 1
      NSUR = ISTART + JSTART + 1
      XFILL(NSUB) = 2. * X(1) - FILL(LADD)
      YFILL(NSUB) = YFILL(LADD)
      YPR(NSUB) = -YPR(LADD)
165 YPR2(NSUB) = YPR2(LADD)
167 RINT = NINT
      IF(RC)600,601,8
C
C   READ AND COMPUTE R-THETA EXTENDED POINT DATA
C
C   READ INPUT TAPE 2,6,(XS(J),YS(J),J = 1,MC)
      DO 164 K = 1,VC
        XST = XFILL(NSUB)
        YST = YFILL(NSUB)
        XD = XS(K) - XST
        YD = YS(K) - YST
        Y3 = 6.0*(YD/XD - YPR(NSUB) - YPR2(NSUB)*XD/2.0)/XD**2
        XD = XD/FLOAT(NINT)
        YD = YD/FLOAT(NINT)
      DO 163 J = 1,NINT
        NSOB = NSUR - J
        XDP = XD*FLOAT(J)
        XFILL(NSOB) = XST + XDP
        YFILL(NSOB) = YST + YPR(NSUB)*XDP + YPR2(NSUB)*XDP**2/2.0 + Y3*XDP
        Y**3/6.0
        YPR(NSOB) = YPR(NSUB) + YPR2(NSUB)*XDP + Y3*XDP**2/2.0
163 YPR2(NSOB) = YPR2(NSUB) + Y3*XDP
164 NSUR = NSOB
C
C   E COMPUTE DELTA-OMEGA POINT DATA
C
C   600 SIGP = 1.0
      WRITE OUTPUT TAPE 3,5, (XFILL(J),YFILL(J), YPR(J),YPR2(J),J = 1,IT
      XOT)
      IF(XFILL(ITOT) - XFILL(1))598,599,599
598 SIGP = -1.0
599 DO 500 J=1,ITOT
      XFILL(J)=XFILL(J)/57.2957795
      YPR(J) = YPR(J)*57.2957795
      YPR2(J) = YPR2(J)*57.2957795**2
      RAT = YPR(J)/YFILL(J)
      DUM1 = SQRT( 1.0 + RAT**2)
      STAU = 1.0/DUM1
      CTAU = RAT/DUM1
      SISUM=SINF(XFILL(J))*CTAU + COSF(XFILL(J))*STAU
      COSUM= COSF(XFILL(J))*CTAU - SINF(XFILL(J))*STAU

```

```

SINT = (YFILL(J)*SIN(DEL(J)) - ARAD*COSUM)/BAR
COMEG = SQRT(1.0 - SOME**2)
IF(ABS(SOME) + EPS - 1.0) 13,513,512
C
C 1 WHEN OMEGA = 90 DEG
C
513 COMEG = 0.0
SOME = 1.0
SKURV = (YFILL(J) - YPR2(J))/(DUP1*YFILL(J)**2)
SPOL = 1.0 + ARAD*SKURV
DRATE(1,J) = SIGP*DUMMY*(SQRT(BAR*SKURV*SPOL) - SPOL)/DELD(1)
DRATE(2,J) = 57.2957795*SIGP*DUMMY*SQRT(SKURV*SPOL/BAR)/DELD(2)
GO TO 168
C
C 2 GENERAL OMEGA
C
513 IF(XFILL(J) - EPS - 1.57079633) 514,515,515
514 COMEG = -COMEG
514 SKURV = (2.0 - (1.0 + YPR2(J)/YFILL(J))*STAU **2)/(YFILL(J)*DUP1
X)
SINT = SIGP*DUMMY*(1.0 + ARAD*SKURV)/COMEG
SISUB = COSUM*COMEG + SISUM*SOME
DRATE(1,J) = SISUB*SINT/DELD(1)
DRATE(2,J) = 57.2957795*SISUM*SINT/(BAR*DELD(2))
168 DEL(1,J) = YFILL(J)*COSF(XFILL(J)) + ARAD*SISUM - BAR*COMEG
DEL(2,J) = ARCINV(SOME,COMEG)
500 CONTINUE
RETURN
END

```



```

      C(NMAX) = DY2
      S(1) = XDELL(1)/G(1)
      T(1) = (T(2)-XDELL(2)*Y1)/G(2)
      DO 330 I=3,NMIN
      S(I)=XDELL(I-1)/G(I)-XFILL(I)*S(I-1)
430 T(I) = (T(I)-XDELL(I)*T(I-1))/G(I)-XDELL(I)*S(I-1)
      C(1) = DY1
      C(NMAX) = DY2
      DO 331 I=2,NMIN
      B = NMAX-I+1
331 C(I) = T(B)-S(B)*C(B+1)
      DO 335 I=1,NMIN
      B(I) = 3.0*(YDELL(I)-C(I)*XDELL(I))/XDELL(I)**2 - (C(I+1)-C(I))/XD
      YFLL(I)
335 A(I) = (C(I+1)-C(I))/XDELL(I)**2 - 2.0*(YDELL(I)-C(I)*XDELL(I))/XD
      YFLL(I)**3
C
C  COMPUTE INTERMEDIATE DATA POINTS AND 1ST AND 2ND DERIVATIVES
C
      RINT = NINT
      JFIN = NINT - 1
      DO 120 N = 1,NMIN
      XFILL(ITOT) = X(N)
      YFILL(ITOT) = Y(N)
      YPR(ITOT) = C(N)
      YPR2(ITOT) = 2.0*B(N)
      DO 130 K = 1,JFIN
      M = K
      XDUM = XDELL(N)*M/RINT
      IMOP = ITOT + K
      XFILL(IMOP) = X(N) + XDUM
      YFILL(IMOP) = XDUM*(C(N) + XDUM*(B(N) + XDUM*A(N))) + Y(N)
      YPR(IMOP) = C(N) + XDUM*(2.0*B(N) + 3.0*XDUM*A(N))
130 YPR2(IMOP) = 2.0*(B(N) + 3.0*XDUM*A(N))
120 ITOT = ITOT + NINT
      XFILL(ITOT) = X(NMAX)
      YFILL(ITOT) = Y(NMAX)
      YPR(ITOT) = C(NMIN) + XDELL(NMIN)*(2.0*B(NMIN) + 3.0*XDELL(NMIN)*A
      X(NMIN))
      YPR2(ITOT) = 2.0*(B(NMIN) + 3.0*XDELL(NMIN)*A(NMIN))
377 RETURN
      END

```

NOTE-THIS SUBROUTINE LIMITS COORDINATE VELOCITY DATA POINTS ACCORDING TO THE LIMITATIONS OF THE NUMERICAL SYSTEM AS EXPRESSED IN TERMS OF A MAXIMUM VELOCITY AND A MAXIMUM ACCELERATION.

```

1  FORMAT(1H0/14H0 JACK-UP AT L=13)
2  FORMAT(1H0/28H0 JACK-UP LIMIT REACHED - N =,I3)
3  FORMAT(16H1 VELOCITY VECTOR/1H0/(1H,3E20.8))
4  DIMENSION RMASK(6),DIFT(2),SDIST(2),REFR(2),ADIST(2),VIST(2),MPOS
5  X(2),CDETR(64),DELD(2),RDI(2),JMAX(2),DEL(2,1000),XFILL(1000),YFILL
6  X(1000),TIME(1000),PRATE(2,1000),DSPEC(100),ISPEC(100),KLIM(100),TA
7  PE(2,100),DOT(64),X(60),Y(60),XDELL(60),YDELL(60),RDELL(60),T(60),Q
8  X(60),S(60),A(60),B(60),C(60),HEAD(14),XS(10),YS(10),ACEL(2),PCEL(2
9  X),VELR(2),YPR(1000),YPR2(1000)
10 COMMON RMASK,DIFT,SDIST,REFR,ADIST,VIST,MPOS,CDETR,DELD,RDI,JMAX,
11 XDELL,XFILL,YFILL,TIME,PRATE,DSPEC,ISPEC,KLIM,TAPE,DOT,X,Y,XDELL,YDE
12 ALL,RDELL,T,Q,S,A,B,C,HEAD,JOK,MDETR,KREC,ITEL,MDSUM,BAR,ARAD,DLIM,
13 XKNUM,MOUNT,ICURV,LDUM,LPS,SPEED,TRAVEL,CHIP,AMAX,JCON,DUMMY,M1,M2,
14 XITOT,JCRIT,NTYP,RIST,IPOS,DIST,KPOS,TAL,A1,A2,WIST,TIST,A3,LPOS,NP
15 XOS,IBUG,JRITE,JUT,KK,KREC,JET,ITEP,PADD,DY1,DY2,NMAX,NINT,IBEG,MC,
16 XIREV,KDET,KREV,LDET,ISTART,JSTART,RINT,XS,YS,RAT,STAU,CTAU,SISUM,
17 XCOSUM,SONEG,COMEG,SKURV,SPOL,ITOP,RHO,PHI,DRAD,THE1,THE2,CSIGN,
18 XACFL,PCEL,VELR,YPR,YPR2

```

C A INITIALIZE PARAMETERS

```

50 NPICK = 0
WRITE OUTPUT TAPE 3,5,((PRATE(J,K),J=1,2),TIME(K),K=1,ITOT)
ASSIGN 265 TO KPIC
ASSIGN 600 TO LL
ASSIGN 17 TO MM
ASSIGN 15 TO NN
IF(M1 - M2)310,311,310
311 LOC = M1
ASSIGN 266 TO KPIC
ASSIGN 601 TO LL
ASSIGN 13 TO MM
ASSIGN 11 TO NN
310 VMAX = FLOAT(IREV)/(200.0*SPEED)
DO 10 M = 1,2
10 PRATE(M,1) = RDI(M)
GO TO MM,(13,17)
17 DO 12 M=1,2
12 ADIST(M) = (DEL(M,ITOT) - DEL(M,1))/DELD(M)
LOC = 1 + XMINOF(XINT(2.0*ABSF(ADIST(2)))/(ABSF(ADIST(1)) + ABSF(A
XDIST(2))))),1)

```

```

      C C C INITIALIZE FOR MAXIMUM ACCELERATION
      C
      C C INITIALIZE PARAMETERS AND ADJUST END POINT VELOCITIES
      C
1311 MSIGN = 1
      MREF = 0
      IF(XINTF(FLOATF(KDET)/2.0 + 0.6) - 1)288,287,288
287 PRATE(KDET,ITOT) = FLOATF(KREV)/(200.0*SPEED)
      IF(M1-M2)289,288,289
289 KDOT = XMODF(KDET,2) + 1
      DO 171 M=1,2
171 ADIST(M) = (DEL(M,ITOT) - DEL(M,ITOT-1))/DELD(M)
      PRATE(KDOT,ITOT) = (ADIST(KDOT)/ADIST(KDET))*(PRATE(KDET,ITOT-1)+
      XPRATE(KDET,ITOT)) - PRATE(KDOT,ITOT-1)
      C
      C C D ADJUST FOR MAXIMUM ACCELERATION
      C
288 DO 280 L = 2,ITOT
      K = MREF + L*MSIGN
      IMOD = 1
278 KMOD = IMOD
      KP = K - (KMOD + 1)*MSIGN
      DO 20 N = 1,KMOD
      KP = KP + MSIGN
      NP = KP + MSIGN
      LP = XMAXRF(KP,NP)
      C
      C C 1 COMPUTE INTERVAL ACCELERATION
      C
      DO 315 M = M1,M2
      ADIST(M) = (DEL(M,NP) - DEL(M,KP))*FLOATF(MSIGN)/DELD(M)
      PCEL(M) = (PRATE(M,K)**2 - PRATE(M,KP)**2)/(2.0*ADIST(M))
315 ACCEL(M) = SIGNF(MINIF(ABSF(PCEL(M)),AMAX),PCEL(M))

```

```

      ACCEL = 1 + XMINOF(VALLOC, ABSF(PCEL(2))/(ABSF(PCEL(1))+ ABSF(PCEL
      Y(2))),1)
C
C 1 CHECK AGAINST MAXIMUM ACCELERATION
C
266 IF(ABSF(PCEL(LOC)) - AMAX)51,51,283
51 DO 52 M=M1,M2
52 PRATE(M,NP) = PRATE(M,K)
GO TO 26
C
C 2 ATTEMPT TO ADJUST ACCELERATION TO MAXIMUM
C
283 IF(L + N - KMOD - ITOT)284,23,23
284 DO 21 MA = M1,M2
VELP = PRATE(LOC,KP)**2 + 2.0*ACEL(LOC)*ADIST(LOC)
317 IF(PRATE(LOC,KP)*PRATE(LOC,K))318,319,319
319 SI = PRATE(LOC,K) + PRATE(LOC,KP)
GO TO 316
318 SI = ADIST(LOC)
316 VELR(LOC) = SIGNF(SQRTF(ABSF(VELP)),SI)
GO TO LL,(600,601)
C
C 4 ADJUST OTHER COORDINATE TO CORRESPOND
C
600 MOC = XMODF(LOC,2) + 1
RHO = ADIST(MOC)/ADIST(LOC)
PHI = (RHO*PRATE(LOC,KP) - PRATE(MOC,KP))*(PRATE(LOC,KP) + VELR(LO
XC))/ADIST(LOC)
ACEL(MOC) = RHO*ACEL(LOC) + PHI
IF(ABSF( ACEL(MOC) ) - AMAX)24,24,211
211 ACEL(MOC) = SIGNF(AMAX,ACEL(MOC))
21 LOC = MOC
C
C 5 BACK UP A POINT AND TRY AGAIN
C
23 IMOD = IMOD + 1
WRITE OUTPUT TAPE 3,1,L
IF(IMOD - XMINOF(LDET+1,L))278,279,279
C
C 6 BACK UP LIMIT REACHED
C
279 WRITE OUTPUT TAPE 3,2,N
TIME(LP) = 2.0*ABSF(ADIST(LOC) /(PRATE(LOC,KP) + PRATE(LOC,NP)))
SENSE LIGHT 1
GO TO 280
C
C 7 ADJUSTMENT SUCCESSFUL- COMPUTE NEW VELOCITY DATA

```

```

C
24 PRATE(MOC,NP) = (A*INT(LOC)/ADIST(LOC))*(PRATE(LOC,KP) + VELR(LOC)
  * ) - PRATE(MOC,KP)
601 PRATE(LOC,NP) = VELR(LOC)
C
C   COMPUTE TAPE LENGTH OF DATA INTERVAL
C
20 TIME(LP) = 2. *ADSF(ADIST(LOC) / (PRATE(LOC,KP) + PRATE(LOC,NP)))
280 CONTINUE
  NPICK = NPICK + 1
  WRITE OUTPUT TAPE 3,5,((PRATE(J,K),J=1,2),TIME(K),K=1,ITOT)
  GO TO (300,302),NPICK
300 IF(SENSE LIGHT 1)587,302
C
C   SET UP PARAMETERS FOR ANOTHER PASS
C
587 MRFG = ITOT + 1
  MSIGN = -1
  GO TO 288
302 DO 295 J=1,2
295 RDI(J) = PRATE(J,ITOT)
  SENSE LIGHT 0
  RETURN
  END

```



```

      NTYP = KTEST
      KCHK = 2*NUMK
      DO 905 K = 1,2
      DO 912 M = 1,KCHK
      JSPEC = 1 + (NSPEC-1)/6
      LSPEC = NSPEC - 6*(JSPEC-1)
D 908 TAPE(JSPEC) = TAPE(JSPEC) + RMASK(LSPEC)*DOT(KTEST)
      912 NSPEC = NSPEC + XINTF(PADD)
      KCHK = 2
      905 KTEST = KNUM
      GO TO 911
C
C D SINGLE INSTRUCTION TYPE
C
      913 IF(NTYP-4)914,914,896
C
C 1 AUXILIARY IS A PULSE INSTRUCTION
C
      914 ISIGN = 1
      MTYP = ISPEC( ITEL)
      LTYP = MTYP
      310 KCHK = KLIM( ITEL)
      311 DO 660 J=1,KCHK
      JDUM1 = NSPEC + ISIGN*J -1
      JSPEC = 1 + (JDUM1-1)/6
      LSPEC = NSPEC - 6*(JSPEC-1)
      DO 365 M = 1,2
      DO 300 L = 1,3
      TEST = TAPE(JSPEC)*(-(DOT(MTYP) + DOT(LTYP))) + (-TAPE(JSPEC))*
B X(DOT(MTYP) + DOT(LTYP))
D IF(TEST*RMASK(LSPEC))360,661,360
      661 WRITE OUTPUT TAPE3,950,ITEL
      GO TO (670,671)*M
      670 GO TO(691,692),JGO
      691 NSPEC = JDUM1 - ISIGN*KCHK
      JGO = 2
      KGO = KGO + 1
      GO TO 366
      692 NSPEC = (NSPEC + JDUM1 - ISIGN*KCHK)/2
      GO TO 896
      671 NTYP = NTYP + 2
      IF(NTYP - 4) 304,304,305
      305 NTYP = NTYP - 4
B 304 TAPE(JSPEC) = TAPE(JSPEC)*(-RMASK(LSPEC)) + (-DOT(NTYP))*TAPE(JSPE
R XC)*RMASK(LSPEC)
      CDTR(NTYP) = CDTR(NTYP) - 1.0
      GO TO 895

```

```

300 LTYP = LTYP + L
    IF (LTYP - 4) 300, 300, 300
303 LTYP = LTYP - 4
300 CONTINUE
365 MTYP = LTYP
660 CONTINUE
    KGO = KGO + 1
    IF (KGO - 2) 366, 896, 896
366 ISIGN = -ISIGN
    LTYP = MTYP
    GO TO 310
C
C / NON-PULSE INSTRUCTION, COMPUTE POSITION AND DISTRIBUTE
C
896 JSPEC = 1 + (NSPEC-1)/6
    LSPEC = NSPEC - 6*(JSPEC-1)
906 TAPE(JSPEC) = TAPE(JSPEC) + RMASK(LSPEC)*DOT(NTYP)
911 CDETR(NTYP) = CDETR(NTYP) + 1.0
995 RETURN
    END

```

-143-

RM-78712


```

      FUNCTION ARCTAN(X)
C
C  ARCTAN-THIS FUNCTION COMPUTES ACCURATELY COMPUTES THE ARCTANGENT OF
C  AN ANGLE
C
C
      N = 1
      IF (ABS(X) - 1.25) 10,11,11
10  R = (X*X)/(1.0 + (X*X))
      ARCTAN = 1.0
      CONST = 1.0
      VAR = 1.0
12  P = FLOATF(2*N)
      CONST = CONST*P/(1.0 + P)
      VAR = VAR*R
      TERM = CONST*VAR
      ARCTAN = ARCTAN + TERM
      N = N+1
      IF (TERM - 1.0E-8) 13,13,12
13  ARCTAN = (X/(1.0 + (X*X)))*ARCTAN
      GO TO 15
11  R = -1.0/(X*X)
      VAR = -1.0/X
      ARCTAN = SIGNF(1.57079633,X)
14  TERM = VAR/FLOATF(2*N-1)
      ARCTAN = ARCTAN + TERM
      VAR = VAR*R
      N = N+1
      IF (ABS(TERM) - 1.0E-8) 15,15,14
15  RETURN
      END

```

FUNCTION ARCINV(SOME, COME)

C
C ARCINV-THE FUNCTION ARCINV COMPUTES AN ANGLE AS SPECIFIED BY ITS
C SINE AND COSINE IN ANGULAR DEGREES
C
C

IF (ABS(COME)-EPS) 530, 531, 531
530 ARCINV = 90.0000000
GO TO 500
531 TONEG=SOME/COME
ARCINV = ARCTAN(TONEG)*57.2957795
IF (TONEG) 520, 500, 510
510 IF (SOME) 521, 500, 500
520 IF (SOME) 500, 500, 521
521 ARCINV = 180.0 + ARCINV
500 RETURN
END

```

SUBROUTINE SQUASH
C DASH-THIS SUBROUTINE CALLS THE SCRUNCHER SUBROUTINE LNREC, AND DE-
C TERMINATES THE RESULTS OF THE SCRUNCHING PROCESS
101 FORMAT(70H1PROGRAM HAS FAILED. REMOVE AND SAVE TAPES A5 AND B6----
X-----END OF JOB/////)
102 FORMAT(67H1TAPE REDUNDANCY. REMOVE AND SAVE TAPES A5 AND B6-----
X-END OF JOB/////)
104 FORMAT(72H1A TAPE HAS BEEN COMPLETED. REMOVE AND SAVE TAPES A5 AND
XB6-----END OF JOB/////)
COMMON M,N
C A REWIND TAPES, CALL SCRUNCHER, AND REWIND AGAIN
2 REWIND 5
REWIND 6
CALL LNREC
REWIND 5
REWIND 6
C B DETERMINE SCRUNCHING RESULTS
IF(N)4,5,6
C 1 WRITING CHANNEL DISCONNECTED
4 PRINT 101
WRITE OUTPUT TAPE 3,101
PAUSE 22222
RETURN
5 IF(N)6,7,6
C 2 TAPE REDUNDANCY
6 PRINT 102
WRITE OUTPUT TAPE 3,102
PAUSE 33333
RETURN
C 3 TAPE SCRUNCED SUCCESSFULLY
7 PRINT 104
WRITE OUTPUT TAPE 3,104
PAUSE 55555
RETURN
END

```

COUNT 60
 ENTRY LNREC
 LNREC CLA CONT
 STO CMNDR
 OCT 0776000-22-5
 RTBA 5
 RCHA CMNDA
 TCOA *
 WTRB 6
 RCHB CMNDR
 WAIT TCOA *
 TFFA OUT
 LCHP CMNDR1
 PTRB 5
 RCHA CMNDA
 TPA WAIT
 OUT STZ CMNDR
 TCOB **3
 STL N
 TRA **2
 STZ N
 TCOB *
 TPCR **3
 STZ M
 TRA **2
 STL M
 TRA 1,4
 A RSS 255
 B RSS 255
 C RSS 255
 D RSS 255
 E RSS 255
 F RSS 255
 G RSS 255
 H RSS 255
 I RSS 255
 J RSS 255
 CMNDR IOCP A,,255
 IOCP R,,255
 IOCP C,,255
 IOCP D,,255
 IOCP E,,255
 IOCP F,,255
 IOCP G,,255
 CMNDR1 IOCP H,,255
 IOCP I,,255

DATE

FILMED

10/17/94

END