Consiglio Nazionale delle Ricerche
ISTITUTO DI ELABORAZIONE DELLA INFORMAZIONE

AN ANALYSYS OF HOW MOTION COMMANDS ARE PROCESSED IN THE APT SYSTEM

S. Trumpy

Nota Interna B70-16

(Luglio '70)

CONTENTS

----

1. INTRODUCTION

This report contains a description of how motion commands are processed in the APT System.

The present work is part of an analysis of machine tool languages, which is being carried out in order to obtain a clear understanding of the theoretical and practical problems which underly their definition and usage.

Making reference to the geometrical execution statements as they appear in the part program, various processors of the APT System have been examined in order to analyse the preliminary elaborations performed on the source statements, up to the point at wich the ARELEM section is called for the final computation of the three dimensional tool path.

The latter has been described in the Scientific Report "An algorithmic description of cut vector computation in APT System".

The existing documentation on the APT language, i.e. "Part Programming Reference Manual" is not very clear, in particular for wath concerns the description of motion commands. It is furthemore incomplete so that it leaves much freedom to the implementation as well as to the intuition of part programmers. The reason of this is that N.C. languages are based on elaborate geometrical concepts and empirical routines for cutter motion and technological computations, wich may largely deviate from intuition giving in many case rise to unespected results.

This report does not contain original ideas but is intended to enucleate the preceding problems both as a step for a clearer theoretical understanding of the problems involved as well as a preliminary step for standardization work.

2. SECTION I

Particular attention has been devoted to the iter followed by the motion command in the core of various sections of the system; certainly this type of statement gives the programmer a greater possibility of choice and involves a larger number of data processing than other types do.

Section I of the APT System has the following tasks:

1) It translates all APT statements from the Part Program Language into the computer language.

2) It generates the canonical forms of surfaces defined in the Part Program.

3) It executes all computations.

4) It prints and punches the generated data, as requested in the Part Program.

5) It prepares the input for the subsequent sections of the System.

The input of Section I being the APT statements, the outputs are:

IOUTAP = a list of all statements with possible diagnostic errors; there in are also included the PRINT statements specified in the Part Program. This output of Section I normally appears in the heading part of printed-out of computer.

SRFTAP = it collects the large surface data for later use in Section II.

PROTAP = together with SRFTAP, PROTAP is the input of Section II; it contains the sequence of all statements in proper order for the subsequent computations. It is written in the standardized code of CLDATA.

The most important phases of Section I are two. In the first phase the punched cards of the part program are converted into the Section I intermediate language; each statement is labelled according to the type of processing it requires.

The following tables are thus generated:

Synonim table: it contains all the synonims specified in the part program with a SYN/... statement.

Variable table: it contains all variable symbols that appear in the part program. A search into the table is made any time a new variable symbol is defined to see whether it was already present or not. In this case an error is detected.

Number table: it contains all numbers converted into floating point of that part of the program which is translated into intermediate language.

Canon table: it contains all geometrical data of the surfaces contained in that part of the program which is translated into intermediate language; it contains at the same time MACRO bodies partially processed till the end of the program.

PTPP table: it contains codes and flags representing the part program in the intermediate language.

If a statement reveals a nested structure the special "loop table" is generated, to point out the limits of the nested structure and its type.

In the second phase of SECTION I the contents of the mentioned tables are used to generate the PROTAP file.

Since the logic of the APT system is interpretative, each statement in SECTION I is translated and processed before the subsequent is read (NORMAL MODE). After that a statement has had its complete representation in the PROTAP file, the contents of the tables NUMBER,CANON,PTPP, can be erased. There are txo cases in which the logic of the system is varied:

LOOPING MODE - It starts after a LOOPST, till a LOOPND is read. The statements are translated and processed subsequently except after the couple IF .... JUMPTO, which may vary the order of operations. Because of the computed trasfer, the whole looping body must be stored in PTPP table until LOOPND is encountered.
After LOOPND is encountered the NORMAL MODE is automatically restored.

MACRO MODE - It is in effect after a MACRO, until TERMAC is read.

The statements of the MACRO body partially processed are stored in the CANON table till the end of the program.

When a CALL statement is read in the part program, the MACRO is executed. The execution of a MACRO body is equal to the execution of a LOOP body except that, before statements are processed, all variables present in the MACRO are replaced by the values assigned in the CALL statement. When TERMAC is encountered the contents of PTPP and NUMBER tables can be erased but not that of CANON, because it is possible to meet another CALL statement before the end of the program.

In this second phase all nested structures from right to left, from inner to outer, are performed. At this point the whole statement is processed: all computations are executed, all canonical forms are generated. The result is then written in PROTAP file and forms the input of the subsequent sections.

THE ROUTINES OF SECTION I ARE:

2.1 PHASE I

This routine executes the tasks described in the first phase (cfr. 2).

The input of PHASE I are the punched cards of the Part Program and the output is the Part Program translated into the SECTION I reference language and various tables. This reference language is a table of codes and flags representing the content of punched cards. This language will be used in the subsequent routine, that is PHASE 2.

In detail, PHASE I performs the following sequence of operations.

1) Initialization: it prints the sequence number of the statement and performs a first core storage assignment.

2) Statement brake-up: it identifies the elements of the language from the input line. An element of the language may be a number floating point of twelve of less characters, or a name of six or less characters. If the word REMARK is read, what follows is immediately printed without any computation. When a punctuation sign is found, the element of the language is completely identified.

3) Generation of records in reference language: the representation of the unit of information previously identified is now performed.

4) Termination: the end of statement is checked.

5) Logic control: as described in the previous paragraph, the logic of the system can have three different modes, i.e.:

MODE 1 = NORMAL

MODE 2 = LOOP

MODE 3 = MACRO

At this point a control is performed and eventual errors are pointed out.

2.2 PHASE 2

This routine executes the tasks described in the second phase (cfr. 2).

The input of PHASE 2 is the Part Program translated into the SECTION I reference language, and the output are PROTAP records.

This routine controls the processing of all APT statements.

In this study, our attention having been devoted to motion statements only, the input information will be of the following types:

1) Tool to surface relation as TLLFT, TLON, TLRGT, TLONPS, etc.

2) Startup motion commands as: GO/modifier, S1, modifier, S2, modifier, S3 etc.

3) Motion commands along surfaces as:GOLFT, GORGT, GOUP, etc.

4) Directions, points, and incremental motions as: INDIRP, GOTO, GODLTA, etc.

5) Parameters of Section II as: TLAXIS, MULTAX, THICK, etc.

The output of the routine will consist of the corresponding records PROTAP.

PHASE 2 performs the following sequence of operations:

1) Counters and flags are initialized.

2) A branch is set up to transfer to different storage areas the following types of language elements:

A) Vocabulary words

B) Geometric surfaces

C) Numbers

D) Statement identifiers

E) Undefined symbols.

If the language element is of the type A, different operations are performed for different vocabulary words.

If the vocabulary word is one of the following:

TO, PAST, PSTAN, ON, TANTO : they are assumed to be a modifier relative to the check surface for a continuous motion command and they are stored with the data of that surface. The routine RITAPE will print them in the same record of the check surface. In a startup motion one of the above vocabulary words may be the modifier affecting the drive surface or the part surface or the check surface and it is stored with the data of the corresponding surface.

TLLFT, TLRGT, TLON: these words are assumed to be a modifier relative to the drive surface for a continuous motion command and are stored with the data relative to that surface. The system will use this datum

In the case of multiple C.S. four or more numbers must be given in the part program to be applied to all controlling surfaces. A particular PROTAP record is generated.

GO/.... (startup motion command)	a particular PROTAP record is generated.
GOLFT,GORGT,GOUP, etc.	a flag is setup to point out the type of mouvement and a particular PROTAP record is generated.
TLAXIS, MULTAX, THICK, etc.	a particular PROTAP record is generated and the numbers following are stored.

If the language element is of the type B, the surfaces counter is increased by one. For a startup motion, the first surface generates a Drive Surface record, the second a Part Surface record, the third a Check Surface record. All surfaces may not have modifiers; if the surface modifier is implied , it is assumed to be TO. All surface records PROTAP are of the same type. Before the Drive Surface record is written, a search is made to see whether the check surface is explicit or not. If not, the Drive Surface of the subsequent motion command will be used as Check Surface of the present motion. The logic of the system has the capability to process two statements at the same time, so in this case a PROTAP record is not generated until the subsequent statement has been read. The modifier of the implied check surface is determined using the stored tool-to-surface relation, relative to the drive surface of the present motion and the type of the subsequent motion command, as it appears in the following table.

TLLFT	GOLFT	TO
TLLFT	GORGT	PAST
TLRGT	GORGT	TO
TLON	GOLFT or GORGT	ON
anyone	GOFWD or GOBACK	TANTO
anyone	GOUP or GODOWN	error

If the controlling surface is a surface involving a large number of data as ruled surfaces, policonics, and tabcyls, it is stored in the SRFTAP file instead of in the PROTAP file.

If the language element is <u>of the type C</u> apart from cases as

GODLTA/scalar, scalar, scalar

GOTO/ scalar, scalar, scalar

THICK/ scalar, scalar, scalar

TOLER/ scalar, scalar, scalar

INTOL/ scalar, scalar, scalar

OUTTOL/scalar, scalar, scalar

in wich the numbers are stored in the record of the word by wich they are preceded, it can have three different meanings:

1) feedrate
2) number of intersections with the check surface
3) numeric statement identifier

If the number is not the last word of the statement, the subsequent word is tested; if it is INTOF, the number has the meaning two and a particular PROTAP record is created; if it is not, the number has the meaning three. No record, at this point, is generated, but the system is prepared to change the type of logic (form NORMAL MODE to MACRO or LOOPING MODE).

If the number is the last word of the statement and the data of three controlling surfaces have been exhausted, the number is a feedrate unless a statement has been previoulsly defined with the same numeric identifier: in this case the number is a numeric statement identifier. If the number is a feedrate, a particular record of postprocessor is generated.

If the language element is <u>of the type D</u>, MACRO MODE or LOOPING MODE are put in effect.

If the language element is a word <u>of the type E</u>, it is assumed that an undefined variable symbol is a statement identifier, which will be defined later.

This is the case of an unconditional transfer wich arises in multiple check surface usage.

2.3 ROUTINE RITAPE

This routine writes on the PROTAP file the previoulsy prepared blocks. PROTAP consists of all informations which are transferred from SECTION I to SECTION II an III and to the post processor.

The input lines representing a motion statement are read by PHASE I from left to right, but they are printed on PROTAP in the following order:

a) data referring to the drive surface.

b) data referring to the check surface.

c) motion command.

When the check surface is not specified in the statement, the motion command and all other PROTAP data are stored, until the check surface can be derived from the subsequent motion statement. After this, the PROTAP block corresponding to the statement is written. PROTAP blocks are always preceded by a statement identifier record.

3. SECTION II

The Section II of the APT System is that wich solves the problem of determining the sequence of the tool positions wich approximates the desired shape of the workpiece. It performs the following main tasks:

1) - It makes a selection of different types of computation for different types of geometrical execution statements. In order to solve the problem of the initial continuous motion command, the routine STRTUP is called; to solve the problem of two dimensional tool paths, the routine ARLM2 is called and to solve the problem of three dimensional tool paths, the routine ARLM3 is called.

2) - It controls whether the initial tool position is compatible with the given motion commands.

3) - It computes the sequence of cut vectors.

The input of Section II being the PROTAP file, the output is CLDATA file, which contains the sequence of tool end coordinates and any other information to be passed to the further Sections of the APT system.

The first three levels of section II routines are shown in the fig. 1.

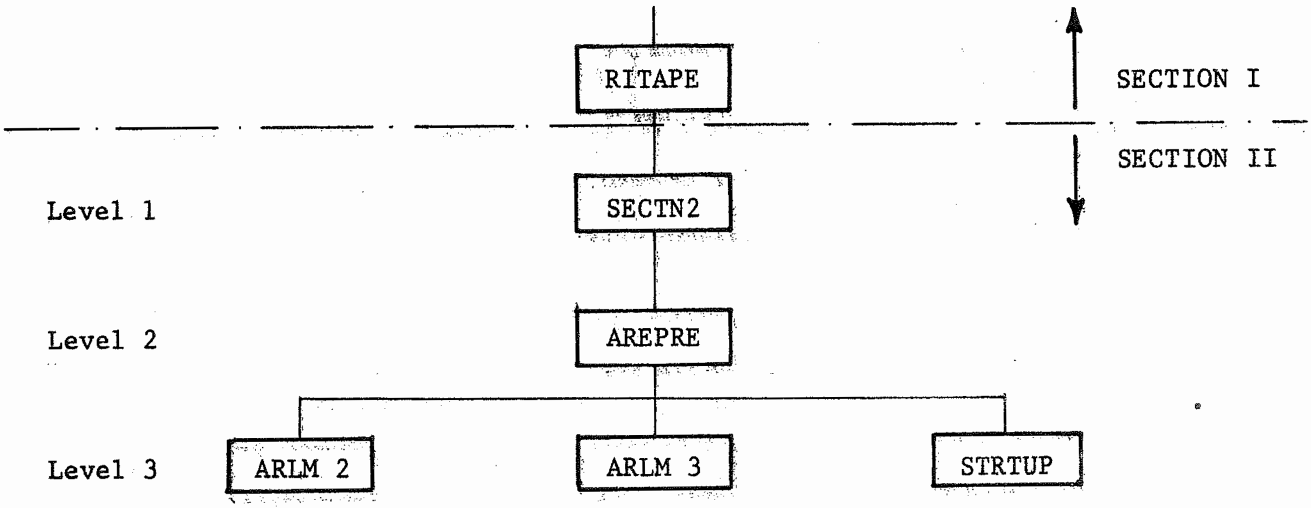


fig. 1

The flow of the computation in the most important routine of the third level (ARLM3) is described in the Scientific Report n° 3 "An algoritmic description of cut vector computation in APT System".

This paragraph describes the other routines except ARLM2. ARLM2 solves the same problem as ARLM3 in simple twodimensional cases.

3.1 <u>SECTN2</u>

This routine analyses the PROTAP records and adresses them to various computations. Each PROTAP record is a series of numeric "words", the second of which characterizes the type of record; the routine SECTN2 reads this word and adresses the record to particular storage areas or directly to special programs.

Three types of logic are met:

a) Normal Mode - the records are read one by one and processed;

b) Trasfer Mode - is activated when a statement of the type TRANTO/IDENTIFIER or with multiple check surfaces is found. With this type of Mode any processing of PROTAP records is not performed till the statement is met whith the identifier to which the transfer is made.

c) Restart Mode - is activated when an error of SECTION II is detected. The records are processed in the Normal Mode except the motion commands which are ignored till a well defined direction can be found to start again with the computations. This condition arises when a startup command like GOTO/POINT, FROM/POINT is met, and the Normal Mode is restored.

A description of how SECTN2 processes the various types of PROTAP records is the following:

a) Statement sequence number record - passes unvaried to the CLTAPE

b) Post-processor word record - passes unvaried to the CLTAPE

c) Surface data record - the third word permits the identification of the type of controlling surface. The modifier of the surface, i.e. the fourth word of the record, is stored. The words to start from the fifth (the data of the surface) are stored in a block of common area. These data will serve later to ARLM3 to find directed distances and normal vectors.

Normally in the record of the Drive Surface of a continuous motion command, the modifier is zero, since there is a special record to indicate the tool-to-surface relation (TLLFT or TLRGT etc.).

In the records of the controlling surfaces of a startup command, all the modifiers are always present.

d) Tool to surface relation record - It has been noted that the tool to surface relation has the most important function to determine the modifier of an implied check surface. From the semantical point of view, it is a modifier in relation to the Drive Surface or Part Surface. In particular, TLLFT, TLRGT, TLOFPS are stored in the same way as TO, the first two refer to the Drive Surface, the third to the Part Surface. TLON, TLONPS are stored as ON, the first refers to the Drive Surface, the second to the Part.

If the check surface is explixit, the tool to surface relation is a pleonastic information, but is stored too.

e) Direction and point record - The following table shows how the system treats this type of information.

CONTENT OF RECORD	OPERATIONS AND STORAGE	MEANING FOR SECTION II
INDIRP/P	$\overline{FWD}$ = UNIT ($\overline{P}$ - TE) $\overline{TEK}$ = $\overline{TE}$ - $\overline{FWD}$	The new direction of motion is that of the present tool position (TE) toward point (P)
INDIRV/V	$\overline{FWD}$ = UNIT ($\overline{V}$) $\overline{TEK}$ = $\overline{TE}$ - $\overline{FWD}$	The new direction of motion is toward (V).
FROM/P	$\overline{TE}$ = $\overline{P}$ $\overline{TEK}$ = (0,0,0)	The present tool position is (P).
GODLTA/V GODLTA/n	$\overline{TEK}$ = $\overline{TE}$ $\overline{TE}$ = $\overline{TE}$ + $\overline{V}$ TE = TE + n($\overline{TA}$)	The tool position is increased of (V) or of n inches along the tool axis.
GOTO/P	$\overline{TEK}$ = $\overline{TE}$ TE = P	The new tool position is P.
SRFVCT/$\overline{V}$	SRFVCT = ($\overline{V}$)	An orientation relative to the surface given in the part program is established.

The ($\overline{FWD}$) vector represents the direction of the final cut vector of the previous motion.

($\overline{TEK}$) point represents the starting position of the tool end in the present motion.

($\overline{TE}$) point represents the final position of the tool end in the present motion.

f) Parameter and flag record.

The following table shows how the system treats this type of information.

CONTENT OF RECORD	OPERATIONS AND STORAGE	MEANING FOR SECTION II
CUT or DNTCUT	a switch is set up a switch is set up	prints the tool path on CLTAPE. Does not allow the tool path to go on CLTAPE.
3DCALC 2DCALC	a flag is set up a flag is set up	performs three dimensional com putations performs two dimensional compu tations.
n, INTOF	stores the number n	choises the n^{th} intersection of the Drive or Part Surface with the appropriate check surface.
INTOL/n	stores the number n and establishes the pa rameters depending on it	determines the maximum allowed offset from a controlling sur- face.
OUTTOL/n	" " "	" " "
CUTTER	stores the data of the cutter	defines the profile of the tool which will be used later.
TRANTO/ID.	stores the statement identifier	checks the PROTAP until the sto- red statement identifier is found.
Multiple check surface identifier	stores the statements identifiers	" " "

g) Startup motion instruction record.

The motion command is stored and the routine AREPRE is called. AREPRE will pass the control to the routine STRTUP.

h) Continuous motion instruction record.

The motion command is stored and the routine AREPRE is called. AREPRE will pass the control to the routine ARLM2 or ARLM3.

i) Arelem parameter record.

The following table shows how the system treats this type of information.

CONTENT OF RECORD	OPERATIONS AND STORAGE	MEANING FOR SECTION II
TLAXIS/V	$\overline{TA} = \overline{V}$	Keeps this TA for all computations.
TLAXIS/NORMPS	$\overline{TA}$ P.S.	for each position a TA is associated.
MULTAX	a switch is set up	" " "
MAXDP/D	stores the number	the lenght of the single cut vector must be smaller than D.
NUMPST/N	stores the number	the specifies maximum number of points for a cut sequence.
THICK/N_1,N_2,N_3,N_4	stores the number	the thickness to be added to various controlling surfaces, is established in the order P.S, D.S, C.S.1, C.S.2
NOPS	a switch is set up	the startup motion is performed along the direction of the minimum distance without regard to a Part Surface previously defined. After the startup motion the previously defined Part Surface, is restored.

l) Special routine record - At present the only special routine used is POKET routine to which SECTN2 transfers the control when this type of record is read.

m) <u>FINI record</u> - It is passed unvaried to CLTAPE.

3.2 AREPRE

AREPRE is called by SECTN2 after that a record of motion instruction is read. If the motion is a startup motion, AREPRE passes the control directly to the STRTUP routine. If the motion is continuous, the routine performs a series of preliminary operations, as follows:

Operation 1 - By the routine AMIND the minimum distance from the tool to the Part surface and Drive surface is computed. If both distances are smaller than tolerances, then the processing can continue, otherwise an error is pointed out.

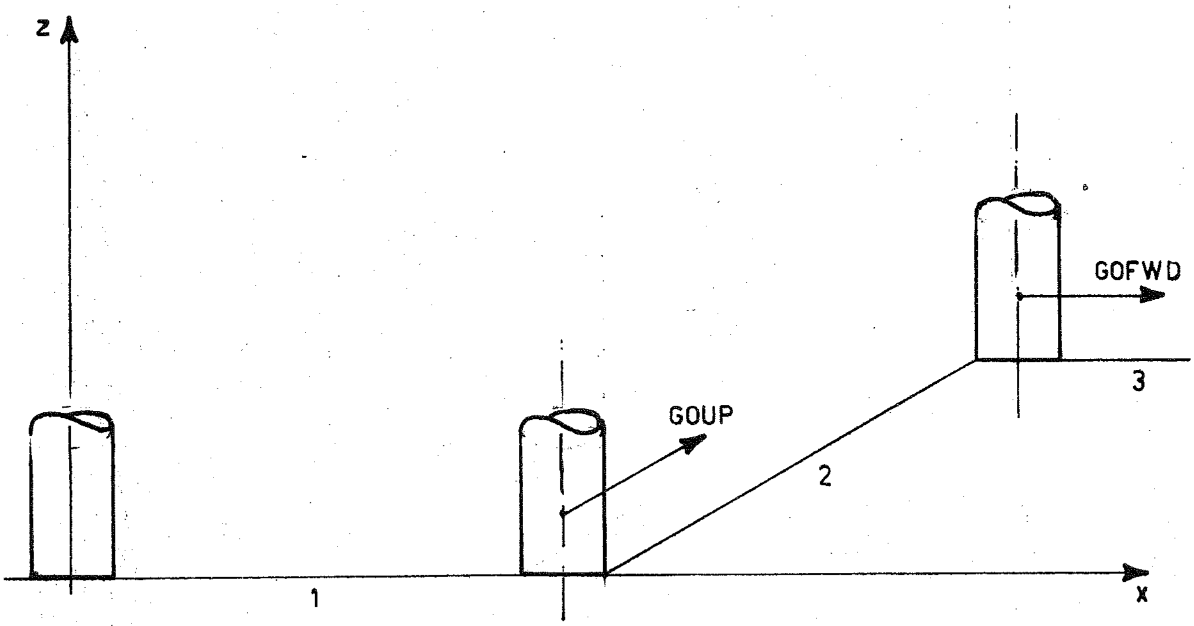
Operation 2 - The final direction of the previous motion $\overline{FWD}$ = $(\overline{TEK} - \overline{TE})$ is saved from the previous computations. If $\overline{FWD}$ is approximately parallel to the tool axis $(\overline{TA})$, the last motion not along the tool axis, is checked to find $\overline{FWD}$. If it does not exist, an error is pointed out.

Operation 3 - Basing on the stored type of the present motion command, $\overline{FWD}$ vector is modified as shown below.

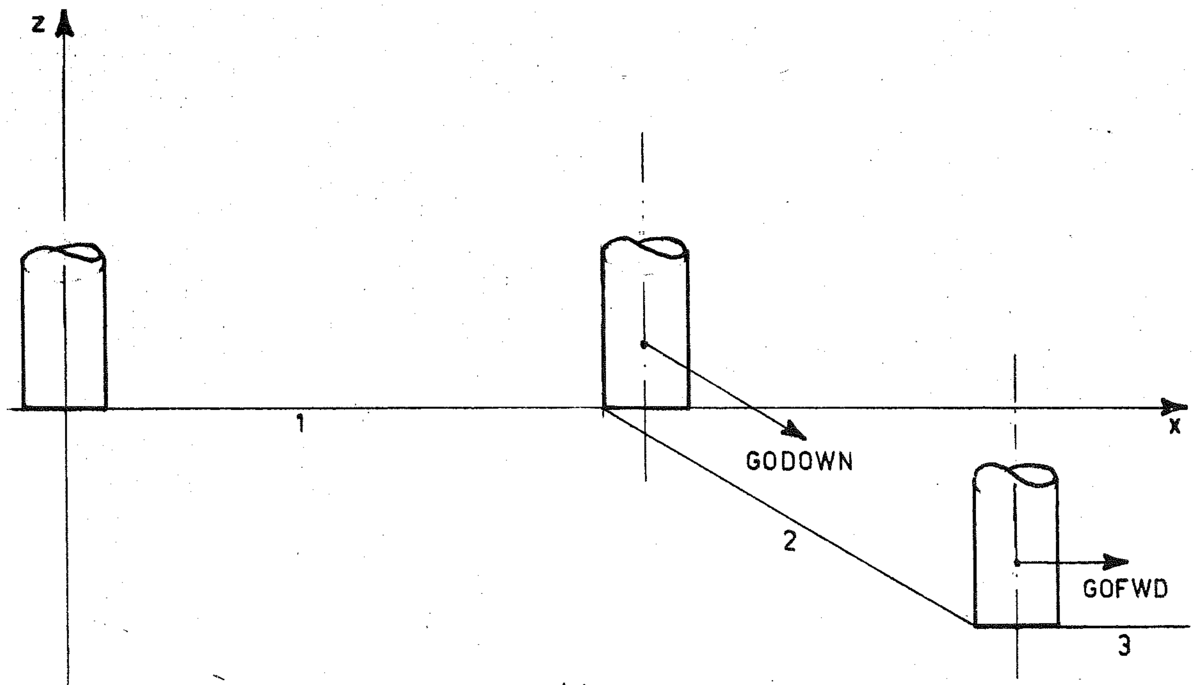
GOUP	$(NEW)\overline{FWD} = \overline{FWD}$
GODOWN	$(NEW)\overline{FWD} = \overline{FWD}$
GOFWD	$(NEW)\overline{FWD} = \overline{FWD}$
GOBACK	$(NEW)\overline{FWD} = -\overline{FWD}$
GORGT	$(NEW)\overline{FWD} = \overline{FWD} \wedge$ UNIT $(\overline{TA})$
GOLFT	$(NEW)\overline{FWD} = -\overline{FWD} \wedge$ UNIT $(\overline{TA})$

It may be noted that the system considers the changements of directions only in the plane ortogonal to TA.

If the tool axis points along z axis of the coordinate system, the changements of direction are considered on_ly in the xy plane. This fact is shown in the fig. 2



a)



b)

The $\overline{FWD}$ vector, after motion number 2 points in both cases a) and b) in the direction of the positive x axis. If the path number 2 becomes ortogonal to the path 1, it is the antiprecedent motion 1 to determine the $\overline{FWD}$ vector of motion number 3.

Operation 4 - The $\overline{TI}$ vector is determined as the intersection of the Part Surface and Drive Surface approximating planes in the starting point of the present motion. If $\overline{TI}$ cannot be determined $\overline{TI} = \overline{FWD}$ is established.

Operation 5 - With this operation the direction along $\overline{TI}$ is determined using the $\overline{FWD}$ vector previously computed. The direction along $\overline{TI}$ is that which points in the same side as the $\overline{FWD}$ vector, i.e. in the side where there is the positive component of $\overline{FWD}$ on $\overline{TI}$. The condition is

$$\overline{FWD} \times \cos \widehat{FWD.TI} > 0$$

When the angle $\widehat{FWD.TI}$ is too close to 90°, a case of numeric undetermination arises. In this case the diagnostic "Initial motion direction or its orientation cannot be determined" is printed. The vector TI is the starting point of the computations of ARLM3.

Operation 6 - At this point a choise between two or three dimensional computations is made. If in the part program was present the statement 2DCALC or 3DCALC, AREPRE passes the control respectively to ARLM2 or ARLM3 routine. Otherwise, if all the following conditions are satisfied, AREPRE calls ARLM2; in all other cases calls ARLM3.

a) The part surface is a plane.

b) Only a check surface is specified and it is a plane or a cylinder parallel to $\overline{TA}$.

c) The drive surface is a plane or a cylinder parallel to $\overline{TA}$.

d) The angle β of the cutter is zero (the tool is cylindric).

3.3. STRTUP

STRTUP has the aim of positioning the cutter in tolerance of one, two or three surfaces from a position relatively far from at least one of the surfaces.

Two fundamental problems have to be solved.

1) - To determine that part of the surface to which the tool is intended to be moved.

If the tool is completely away from the surface to which it must move, no additional motion command must be given to solve eventual ambiguities in choosing the stopping position of the tool. The controlling point of the tool is ($\overline{TM}$). This point is on the tool axis, in the middle of the tool height. If the modifier is TO or ON, the tool must be positioned on the same side of the surface as $\overline{TM}$, otherwise on the opposite side. When the point $\overline{TM}$ is very close to the controlling surface and it is difficult to determine on which side it is, the statement SRFVCT/V_{DS} , V_{PS} , V_{CS} must be given. This solves the ambiguity, because it puts the tool in the same condition, as if it came from away.

The vectors V_{DS}, V_{PS}, V_{CS} point from $\overline{TM}$ toward the controlling surface and the tool is considered to come from a position relatively far in that direction.

2) - To solve the problem of multiple intersections.

This problem not always arises. If it arises, the statement INDIRV/V or INDIRP/P must be given in the part program. This sta tement, together with the startup command, has the effect of carrying the tool to a position closer to the final desired. As a consequence of this statement, the system executes a first time the startup command, computing an interim move of the tool from its position to the controlling surface to which INDIRV or INDIRP is referred, in the specified direction. The routines called are AMIND and CENTR. From that provisional position, calling the same routines, the startup command is again executed in the normal mode without any ambiguity. If, in going in the specified direction, the tool does not meet any surface, it is placed along the INDIRV or INDIRP direction at a distance equal to the maximum of the minimum distances from all controlling surfaces.

.-.-.-.-.-.-.

REFERENCES

(1) - "APT Encyclopedid", IITRI (1965)

(2) - "APT Computer programs", IITRI (1965)

(3) - "Proceedings of the APT thecnical chimical meeting", IITRI (April 10-12, 1967)

(4) - "Proceedings of the APT technical meeting", IITRI (April 9-11, 1968)

(5) - "Long range program", IITRI

oooooooooooo