892

UNITED ARAB REPUBLIC

THE INSTITUTE OF NATIONAL PLANNING

التخطيط القومي

Memo. No. 892

NUMERICAL CONTROL SERIES

POST PROCESSORS

BY

ALWALID ELSHAFEI

OPERATIONS RESEARCH GROUP

MAY 1969

CAIRO

3, MOHAMED MAZHAR - ZAMALEK

INTRODUCTION:

Before we can employ a computer, to generate tapes for numerically controlled machine tools we need to be assured that we have a complete programming system available for the machine tools, control systems and computer, we intend to use. In almost every case, one of the elements of the programming system will be a postprocessor.

The postprocessor is needed whether we are intending to use 2CL, EXPAT,, ADAPT, APT or any of the other general purpose computer programmes now available for NC. More than likely, we need different ones for different machine tool and control system combinations. Why this is so, and exactly what functions the post-processor performs in a complete computer programming system is explained in this article.

In this article, we are going to discuss the postprocessor in relation ONLY to general purpose computer programmes like 2CL, APT and ADAPT, but the principles are valid for most other general purpose programmes as well.

COORDINATION OF WORK WITHIN NC SYSTEMS:

General purpose programming systems for numerically controlled machine tools, such as APT, ADAPT and 2CL consist of two parts: a programming language and a number of computer programmes.

The programming language is used by the part programmer to describe the geometry of the part to beproduced, the tools to be used and the tool motion to cut the desired part. The computer programmes in turn are divided in two sets:

The processor is the first set of programmes. This is used within the computer to interpret the statements written in the programming language by the part programme, evaluate them and calculate a general solution.

The general solution, called cutter location data, is the result of geometric calculations which were executed to solve the problem of calculating the path of one geometric body, the tool, over another geometric body, the part. The general solution is machine tool independent. It will be stored on an intermediate storage device such as magnetic tape or disc.

For the APT system the data are generally stored on a tape, called the CLTAPE (cutter location tape).

The post processor is the name given to the second set of computer programmes.

The postprocessor will adapt the general solution generated by the processor to a specific machine tool with a specific control unit (3).

This procedure of processing of a part programme in two steps, by a processor and then by a postprocessor may seem lengthy. But it has many advantages. The main one is that of increased flexibility and standardization. The separation makes it possible to use only one programming language to programme parts for numerically controlled machine tools having point-to-point as well as multi-axis controls. It requires a large and powerful processor which is not easy to develop. But this in turn also makes it very easy to develop postprocessors to adapt the general solution generated by the processor to a specific machine tool/controller combination.

As an example, it took more than 100 man-years to develop the APT programming system. To develop postprocessors for this system may take an effort of only two to three man-months for a point-to-point postprocessor or of two to three man-years multi-axis postprocessors.

Under these conditions it is now possible to include in the programming language, requirements from users with a wide range of types of machine-tools. This helps to standardize the programming language. Since the format of the interface between processor and postprocessor may easily be standardized, for any new machine tool or any new controller only a new postprocessor has to be written. This then relieves the user or the manufacturer of the new equipment of an enormous programming burden.

PROCESSING AND POSTPROCESSING:

The use of the postprocessor is not very difficult to understand. The postprocessor is the second of two major steps in taping.

(1) Main processing. In the first step, the path that the end of the cutter must take to produce the part is calculated in detail and described numerically. The cutter path is calculated 'in space', as more or less abstract lines.

This first major step is essentially one of solving solid geometry probelems. The programmer, working from the drawing, writes out step by step a detailed mathematical description of the required cutter movements. He writes these procedures in English-like words. If the geometry problems are complex, such as precisely locating the intersection of straight or curved lines and planes, the computer is used. The computer also automatically calculates and compensates for cutter offset.

In America there are at least two large-scale general-purpose computer programmes widely used for this main processing work-APT (Automatic Programmed Tools) and ADAPT (a subset of APT). These two programmes will shortly be supplemented by the British 2CL programme and the German EXAPT I and II programmes. One outstanding feature of all these programmes is that only a few words are required to effect the most complex computer calculations.

The programmer's detailed procedures, are punched on cards, one card to a line. These are then fed into the computer along with the APT or similar programme, which is on magnetic tape. The output of this processing is a second tape, the CLTAPE. The 'CL' means cutter location, the establishment of which is the principal purpose of the first major processing step in producing an NC punched tape.

When the programmer writes up his procedures, he also includes the feeds and speeds for the various cuts to be made. He provides for such instructions as coolant on and off, intensity of spray, direction of the spindle rotation and so on.

Knowing how much stock is to be removed, the programmer must also determine the exact cutter to be used and the number of passes that will be required. All these machining instructions are fed into the computer along with the part description and its geometry problems. However, the computer ignores such machining details at this point. These are simply passed along and included in the CLTAPE.

(2) Postprocessing. In the second major step of the procedure, the CLTAPE and the postprocessor (usually in the form of another magnetic tape) are fed into a computer together.

Here the postprocessor takes the data defining the previously calculated path of the cutter and adapts it:

(A) to the specific machine tool that is to produce the part and

(B) to the specific make and model of numerical control system used to run that machine.

FUNCTIONS OF A POST PROCESSOR:

The statements in a part programme may roughly be divided into two main groups: the first being those which describe the geometry of the part and the tool motion, the second being those statements which describe the functions of the machine tool.

The first group of statements will be processed in the computer by the processor of the programming language. Their evaluation results in geometric calculations which are independent of any machine tool.

The second group comprises what we call the postprocessor commands. They can only be handled in the computer by the postprocessor after the geometric calculations have been performed because their evaluation depends on the machine tool and control system to be programmed.

Postprocessor commands are:

- turning coolant on or off,
- programme end,

- rewinding the control tape,
- position of the indexing table,
- feed,
- spindle speed,
- tool select command,
- tool load command,
- tool length,
- position of the turret,
- machining tolerance, and
- delay time and so on.

The statements for the processor and the postprocessor commands are fed into the computer on the same set of punched cards. In the processor stage of computing the postprocessor commands are however not acted upon. They might be checked by the processor for correct spelling, but otherwise they are simply stored on the intermediate storage tape or disc without any processing.

Basically it may be said, that a postprocessor must cover all programming possibilities that the combination of controller and machine tool is capable of. In that sense it must be a true simulator of this combination.

Within the postprocessor all calculations and all functions executed by the postprocessor are completely machine tool dependent. That means that as many machine tools may be programmed with one programming language as there are postprocessors available for that programming system. (Unless different machine tool/controller combinations have been designed with interchangeable postprocessors in mind.

In other words, the postprocessor does automatically what a human would do if he were an expert in the particular combination of machine tool and control system involved. 'Knowing' the capabilities, operating procedures and limitations of the machine tool, and the format and coding that the control requires, the postprocessor takes the data from the CLTAPE and tailors it to the particular metalworking equipment to be used.

It checks the limits of travel (to make sure the programmer hasn't programmed the work table off the slides of the machine), checks the speeds and feeds and dynamics of acceleration (to make sure the machine will be operated efficiently, yet not exceed its limitations), and at the same time interprets the auxiliary commands such as coolant control, spindle control, tool change, rewind, and the like.

Further, it recognizes any conflicts between the part programme and the machine tool and makes compromises if possible. Finally, the postprocessor encodes all this information into the special language of the particular make and model of control that is to be used. Typically, a postprocessor will consist of some 8,000 to 10,000 computer commands.

In the following presentation, we are going to discuss elaborately the functions of the postprocessor.

GENERATION OF CONTROL BLOCKS

Cutter path information (co-ordinate values, codes for auxiliary functions, and feed and speed values have to be combined in control blocks on the control tape. The postprocessor forms the control blocks in the core memory of the computer in the format that is required for the control unit.

If for instance a control block format with word addresses is used, the postprocessor places the word address before the values to be inserted into a control block ('X' before the X co-ordinate, 'Y' before the Y coordinate, 'N' before the sequence number, and so on). If the values of a certain word are equal in consecutive blocks, the postprocessor omits this word in the second and following blocks (in the case of word addresses).

When a control block has been assembled it will be punched out or written out if it is not required any more.

If the postprocessor runs on a small or medium sized computer, the control block will be punched 'online' into the control tape, using a tape punch connected to the computer. If the postprocessor runs on a large computer, an 'image' of the generated block will first be written on magnetic tape. After the postprocessor has written all control blocks on tape the control tape will be punched 'off-line', using a small and therefore cheaper satellite computer, from the information on the magnetic tape. When writing the magnetic tape the postprocessor already considers the number of channels the tape to be punched will have, and the code the characters are to be coded in, for example EIA code or ASCII code. (One exception to this procedure is the Ferranti Multiax control system which is designed to operate direct from the computer's magnetic tape output.)

CHECKING LIMITS

An important task of the postprocessor is to test that the calculated tool path co-ordinates do not exceed the limits of machine table travel or the limits of the tool head travel. Since the processor of a programming system only performs machine tool independent calculations it is possible that because of a programming error in the part programme tool path co-ordinates are calculated which exceed these limits. This error may only be discovered by the postprocessor which is aware of these limits.

In case such an error occurs, the postprocessor should issue a warning to the part programmer on the verification listing and it should terminate the generation of the control tape because the tape will be false and might lead to machine tool damage.

CODING MACHINE TOOL FUNCTION

It depends on the control unit how the data for programming machine tool functions are put into the control tape. The postprocessor therefore evaluates the postprocessor commands for the auxiliary functions such as turning a coolant on or off, for machine stop, for programme end, and so on, after they have been read in from the intermediate storage device, according to the requirements of the respective control unit. The postprocessor determines the correct code and inserts the data into the appropriate control block.

Spindle speeds programmed in the part programme often have to be changed by the postrpocessor to values actually attainable by the programmed machine tool. The changed values will be put into the control block either directly or coded by a spindle speed number.

CONSIDERATION OF TOOL LENGTH

The intermediate results, computed by the processor of the programming system and stored on an intermediate storage device, represent the path of the tool tip or that of the tool centre. The postprocessor for certain machine tools has to reference the z co-ordinate of the tool path co-ordinates to the control point of the tool head. This is done by using the length of the tool which was passed on from the part programme.

This recalculation of the z coordinate is especially important in case a machine tool with automatic tool change has been programmed and if tools of variable length are utilized. After each tool change the postprocessor has to use the length of the new tool in order to correctly perform the recalculation of the z co-ordinates. The referencing of the z co-ordinate of the control point of the tool head will involve intricate calculations if a multi-axis machine tool is involved.

TOOL CHANGE

The postprocessor for machine tools with automatic tool change capability often has to perform extensive work.

If the postprocessor is designed for a machine tool with tool magazine it generates from the tool select command in the part programme a command necessary for the search in the magazine. For machine tools with a magazine or a turret the command in the part programme to a new tool causes the postprocessor to generate commands for the withdrawal of the last used tool to a position where the tools may be exchanged without endangering either the work piece, the last used tool or other tools. This tool change position may be indicated in different ways:

(1) There may be only one position on the machine tool where a tool change may be performed; in which case the co-ordinates of this point are best programmed into the postprocessor.

(2) The tool change position may change from one part programme to the next. The postprocessor then automatically retrieves the co-ordinates of this point at each tool change command.

(3) The tool change position may change frequently within a part programme. It may in this case either be programmed by a

postprocessor command before each tool change command, so that the postprocessor can generate all necessary control information for the withdrawal of the tool, or the part programmer himself may programme each move in the part programme.

(4) The postprocessor itself may calculate the tool change position. To perform this calculation it takes into account the length of the last used tool, the length of the new tool, the length of tools that have to pass the workpiece before the new tool is in the correct position, and a safety distance or safety plane.

MACHINE TOOL DYNAMICS

A part programmer will always try to keep the production time for the part as short as possible. This is formally achieved by programming the highest possible feedrates. To obtain these feedrates on the machine tool all movable parts of the machine tool have to be accelerated or decelerated as much as possible. Because of the reading time of the tape reader of the control unit, because of the processing time of the data in the control unit, and because of the construction of the machine tool (power of the drives, movable masses and so on) only certain maximum values for acceleration or deceleration for the change of feedrate between two successive control tape records, are possible considering the given limits or, failing that, that the highest possible feedrate will be maintained for as long as possible.

EVALUATION OF CYCLE COMMANDS

A cycle is a fixed sequence of instructions for a machine tool issued by a control unit. In general, cycles are used to programme point-to-point or straight-cut problems like drilling, tapping, face milling and so on. Such a cycle is programmed by giving in the part programme a cycle command, such as a drilling operation, which contains all necessary data and the points at which this cycle is to be executed. The processor does not evaluate the command since its evaluation depends on the machine tool and on the control unit, but it stores the cycle command with its data and the points on the intermediate storage device.

It is the task of the postprocessor to evaluate the cycle command. The evaluation may be done in two ways. The control unit may be equipped with 'canned', that is, built-in, cycles which enable it to execute a complete sequence of instruction for a machining operation in response to a function code read in from the control tape. The postprocessor for such a control unit generates from the general cycle command a control block which contains such a function code and together with all necessary data, followed by a series of control tape records which contain the co-ordinates of the points to be machined under this cycle.

The control unit when reading the first record, stores all information and applies it to the following co-ordinates of points until it reads the command 'Cycle Off'. This function code may either be programmed in the part programme as a postprocessor command or it will be generated by the processor or the postprocessor on encountering the command to begin a new cycle.

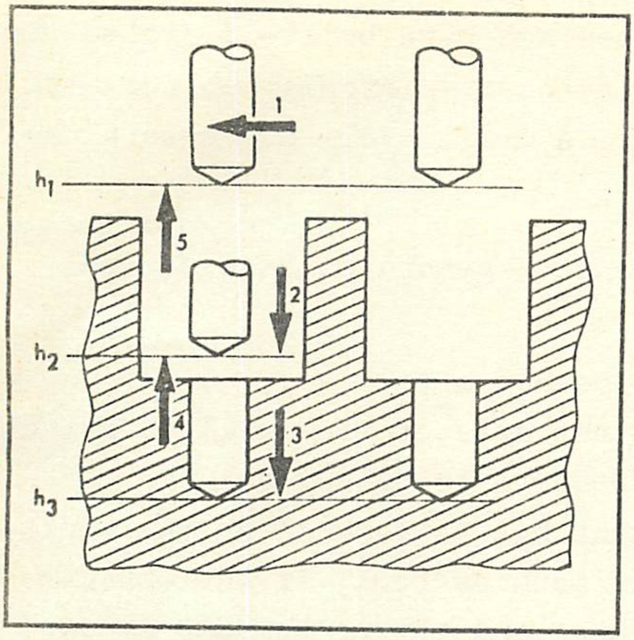
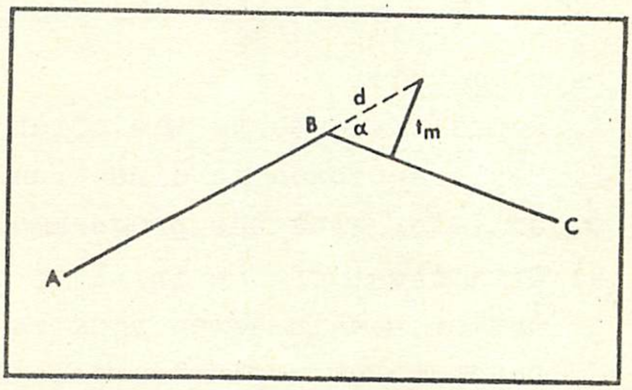


Fig. 1. When evaluating the drill cycle for one hole the postprocessor may have to generate five control tape records.

Fig. 2. With t_m as machining tolerance the largest permissible overshoot at corner B is $d = \frac{t_m}{\sin \alpha}$



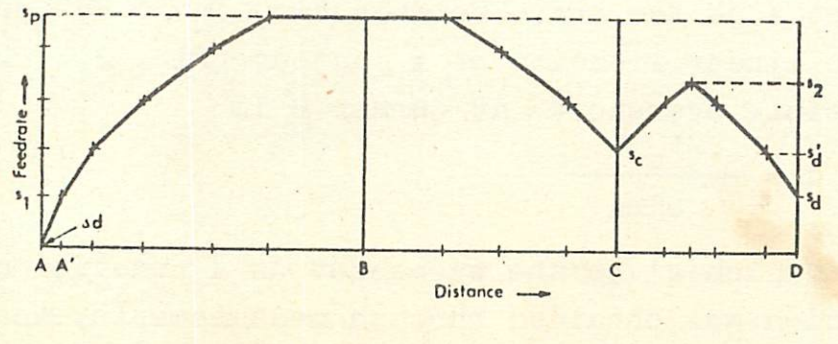


Fig. 3. **Postprocessor** must assure that the programmed feedrate is reached as fast as possible or that the highest possible feedrate will be maintained for as long as possible.

In case the control unit does not have built-in cycles the postprocessor has to generate all necessary information for each point following the cycle command. A drill cycle may be evaluated as follows (Fig.1).

For each point the postprocessor generates five control tape records:

1. Rapid traverse to the point at height h_1;
2. Rapid approach in direction of the axis of the drill to height h_2;
3. Drilling with the programmed feedrate to height h_3;
4. Withdrawal of the drill to height h_2 (it depends on the application and in some cases on the machine tool, if the working feedrate or rapid traverse is used and if the spindle is reversed);
5. Withdrawal of the drill at rapid traverse to height h_1.

CORNERING POSITION

When a tool has to move around a corner, in general, the feedrate used to move it along a preceding linear path may not be employed. The feedrate has to be reduced in order not to exceed a certain 'overshoot'.

In the part programme the programmer may specify a machining tolerance t_m from which the maximum permissible overshoot may be derived. Let a be the angle between two linear cut vectors or between two linear elements of a contour (Fig. 2). Then the largest permissible overshoot at corner B is

$$d = \frac{t_m}{\sin a}$$

From a table which gives the overshoot as a function of the feedrate and which was obtained through measurements, the postprocessor may determine the feedrate with which the tool has to leave cut vector AB and with which it will begin cut vector BC.

FORMAT CONVERSION

All co-ordinates calculated by the computer through the processor of the programming language are stored as normalized decimal numbers. These have in general eight significant digits. Inch dimensions are used. Control units often require, however, numbers of dimension $\frac{1}{1000}$ in. and depending on the size of the machine, the numbers have to have five or six digits.

The postprocessor multiplies the co-ordinate values read in from intermediate storage by an appropriate factor and rounds the results. If the control unit demands numbers of constant length, leading or trailing zeros are added. The postprocessor for certain control units, however, may take into account that either leading or trailing zeros may be suppressed.

AN ILLUSTRATIVE EXAMPLE:

The following example may illustrate the function of the postprocessor. Let the tool path AD (Fig. 3), for which the feedrate S_p was programmed, be approximated by the cut vectors AB, BC, and CD through the processor of the programming language after having processed a particular part programme. Let the programmed feedrate be so high that it may not be reached within one step, that is through the execution of one control tape record.

Beginning at point A, the postprocessor calculates feedrate considering the given limits. Based on this feedrate an intermediate point A', d away from point A, may be computed. A tape record for the path AA' will be generated. If the step from S_1 to S_p is still too big, a new intermediate value for the feedrate and a new point will be calculated from which a new intermediate tape record will be generated. This process, the segmentation of a cut vector, continues until either the programmed feedrate is reached or until the calculated cut vector is used up. It always begins at that end of the cut vector which requires the smaller feedrate.

At the end of cut vector BC the feedrate is required to have the value S_c. Since this value is smaller than the value at point B, segmentation starts at C. This ensures that the correct feedrate value at C is reached and that on as large a part of cut vector BC as possible the high feedrate value is applied.

The task of thepostprocessor becomes somewhat more difficult when, as happens frequently in contouring programmes, a calculated cut vector is very short, a high feedrate was programmed for the total path, but because of the cornering condition small feedrates are required at the corners (cut vector CD). Here segmentation begins at point D since feedrate S_d is smaller than S_c. At first the cut vector will be segmented until an intermediate value of the feedrate (S'_d) is approximately equal to the larger feedrate S_c. As soon as this condition is fulfilled the remainder of the cut vector will be segmented from both ends. Now the programmed feedrate S_p will not be reached but only the value S_2, but the cornering requirements, which are more important in this case, will be satisfied.

If the cut vector is too short and if the difference between the feedrates at its ends is too big so that deceleration to the required final feedrate cannot be achieved within its length, the deceleration must commence in blocks which will already have been generated by the time this situation is discovered. To accommodate in these previously generated blocks the new deceleration requirements, these blocks have to be altered, or as it is sometimes called, they have to be reworked. Reworking these blocks is only possible, however if they are still in the core storage of the computer. Postprocessors which have to consider machine tool dynamics are therefore coded in that always a certain number of control blocks (ten to twenty blocks, depending on the extent of the required rework) are kept in core. When the postprocessor at any time has generated for instance twenty records, the first ten will be written out and the next ten will move on the place of the first ten, so that room for ten new records is available.

Accounting for machine tool dynamics may somewhat be regarded as a simulation of the combination of machine tool and controller. The results obtained herewith completely satisfy all practical needs. A true simulation of the response of the system is not possible or is only possible with great effort. It would require that for each moment of the machining process of a part its decrease of weight, for instance, would have to be known and that exact data of the machine tool and their possible dependence on time, temperature and so on would have to be available.

POSTPROCESSORS AND GEOMETRY DESCRIPTIONS:

Since a postprocessor is designed to accommodate a machine-and-tool combination. Because there are more types of machine tools than controls, the processor is usually written for a particular make of control. Sufficient information is then included to accommodate quite a few, different machines. The complete Bunker-Ramo postprocessor, for example, includes procedures for more than 40 machines (see list).

Bunker-Ramo/APT III postprocessor
Machine tool geometry descriptions

American Tool Works	2-axis lathe
Arrow	2-axis profiler
	3-axis profiler
Bardell	3-axis ultrasonic inspection machine
Cincinnati Milling	3-axis vertical Hydrotel (several)
	3-axis bridge mill (several)
	3-axis horizontal Hydrotel (several)
	3-axis skin mill (several)
	4-axis skin mill
Ekstrom-Carlson	3-axis bridge mill
	2-axis router
	3-axis router
Froriep	2-axis vertical turret lathe
	4-axis vertical turret lathe
Giddings & Lewis	3-axis DiMil
	3-axis boring mill (several)
	3-axis skin mill
	5-axis Variax
Gorton	3-axis knee mill
Jones & Lamson	3-axis horizontal turret lathe
Kearney & Trecker	5-axis Milwaukee-Matic III
	3-axis profile mill
Marwin	3-axis bridge mill
Morey	3-axis bridge mill (several)
Motch & Merryweather	2-axis vertical turret lathe
Onsrud	3-axis bridge mill
	3-axis skin mill
	4-axis skin mill
Sundstrand	3-axis No. 3 horizontal mill
	3-axis M-21 Jigmatic mill
	3-axis OM-2 machining center
	4-axis OM-2 machining center
	5-axis OM-3 machining center
	5-axis OM-4 machining center
Turchan	3-axis profile mill (several)
Universal Drafting	2-axis drafting machine
Warner & Swasey	4-axis horizontal turret lathe
Wilson	3-axis bridge mill
	4-axis Versatel mill

```
TRWWSL       POST PROCESSOR OUTPUT FOR WARNER    AND SWASEY, MODEL MODEL 4500 LATHE

PARTNO PSD-50055-752  401017 OPER. 25  W-S LATHE  12/23/65  AL COX

DELETE  SEQ      G      DELTA      DELTA      DELTA       DELTA       DELTA      F     S     T    M    M   CARD   FEED
        NO.               X          Y          Z           U           W                                NO.    RATE

        001
        002                                                                   999   004        03        001  000.00
        003                                                                   999         01             001  000.00
        004                                                                   999         01             001  000.00
        005                                                                   999              40        001  000.00
        006                                                                   999              43        001  000.00
 COORDINATES ARE  X = -009.0460              Z = +014.2430  U = +000.7190  W = +032.2330 MACHINE TIME =  000.248
        007     08    + 6.7951              - 3.5741                          132                        001  101.35
        008     09    + 0.9009              - 0.4739                          999                        002  101.69
 COORDINATES ARE  X = -001.3500              Z = +010.2000  U = +000.7190  W = +032.2330 MACHINE TIME =  000.344
---------------------------------------------------------------------------------------------------------------------
        233           + 0.2409                                     - 0.2455   009                        076  000.00
            FEED RATE XZ HEAD =   0.22                            FEED RATE  UW HEAD =  0.22
        284                                 + 0.0500               - 0.2455   009                        076  000.00
            FEED RATE XZ HEAD =   0.04                            FEED RATE  UW HEAD =  0.22
DUAL HEAD MOTION CAUSES COLLISION AT FOLLOWING TOOL TIP COORDINATES
X=    -1.4220     Z=    10.0960          U=      1.4370      W=      7.4631
        285                                                        - 0.2455   009                        076  000.20
DUAL HEAD MOTION CAUSES COLLISION AT FOLLOWING TOOL TIP COORDINATES
X=    -1.4220     Z=    10.0960          U=      1.4370      W=      7.2176
        286                                                        - 0.2455   009                        076  000.20
DUAL HEAD MOTION CAUSES COLLISION AT FOLLOWING TOOL TIP COORDINATES
X=    -1.4220     Z=    10.0960          U=      1.4370      W=      6.9721
---------------------------------------------------------------------------------------------------------------------
        331                                           + 0.7441                005                        087  000.37
 COORDINATES ARE  X = -009.0000              Z = +009.5000  U = +001.4621  W = +007.8900 MACHINE TIME =  130.480
        332                                           - 0.6391    - 0.6727   004                        087  000.37
 COORDINATES ARE  X = -009.0000              Z = +009.5000  U = +000.8230  W = +007.2173 MACHINE TIME =  132.980
        333                                           - 0.0500    + 0.0500   052                        087  000.37
        334                                                                   999              43        087  000.00
MAIN HEAD COLLIDES WITH SIDE RAM
GEOMETRY CONTINUES,NO MACHINE DYNAMICS PROCESSED
ERROR IN CLTAPE RECORD      420
```

APT printout of Bunker-Ramo's postprocessor for a Warner & Swasey turret lathe illustrates warning diagnostic (in second segment) and terminal diagnostic (in bottom segment).

OUTPUT GENERATED BY THE POSTPROCESSOR:

The output of the postprocessing operation is usually the 'image' of the NC punched tape put on magnetic tape. The punching process itself is relatively so slow that it isn't economical to tie up an ultra-highspeed computer to do the actual tape-punching directly. So the magnetic tape is transferred off-line to another piece of equipment for magnetic-tape-to-punched-tape conversion.

A good postprocessor does more, however, than simply make a magnetic tape image of the NC punched tape. It will print out, or manuscript, a complete guide to the entire programme. This printout, as illustrated, will consist of numbers and letters in a code that is easily read by trained personnel. The printout will include a detailed record of all input commands for quick checking, if any questions should arise.

In fact, the printout of a good processor will include every available bit of information that will help the programmer in trouble-shooting or the operator in running his NC machine. A good printout will include such details as actual feedrate in inches per minute, feed-rate number, absolute coordinates given at frequent intervals in the programme length of the punched tape, and machine tool run time.

The value of this complete prinout detail in NC operations should not be underestimated.

The absolute coordinate listings, for example, can be used by the operator to restart the programme quickly, should operations be interrupted for any reason, such as a tool breakage. The coordinates permit the operator to restart this machine near the point of interruption through manual data input, thus eliminating the need to re-run the entire programme from the beginning of the tape.

In addition to the features already mentioned, one of the most important aids in the printout are the English-word warnings and comments given whenever the computer finds a mistake in the programme. These are called 'diagnostics'. In the Bunker-Ramo postprocessor, the diagnostics are in two forms:
(1) warning and (2) terminal diagnostics.

A WARNING DIAGNOSTIC is one in which the error found by the computer can be handled by the postprocessor. This is a 'recoverable' diagnostic in that the postprocessor analyses the error and assumes a correcting. For example, the programmer may have forgotten to indicate a feedrate, or he may have specified a wrong spindle speed, or failed to call for rapid traverse in a movement that requires a rapid traverse.

In such instances, the postprocessor will assume a feedrate or a different spindle speed or correct for the rapid traverse oversight. At the same time, the processor will print out exactly what the error is, and the selection it has made to correct the error. Having assumed a correction, the computer will then continue to process the programme without interruption.

A TERMINAL DIAGNOSTIC involves an error that is so obviously incorrect (such as a 20ft. move on a 3ft. machine, or a move that would have a tool cutting into its own setup) that the postprocessor cannot handle it. This type of diagnostic will reject or stop the entire computer operation. However, the computer will first print out enough information to aid the programmer in correcting for the gross error.

It is obvious that the more a postprocessor can keep errors in the 'warning' rather than the 'terminal' classification the better. If a postprocessor is unable to cope with minor errors, it will reject the programme on slight provocation. As a result, a number of re-runs through the computer will be required. This can be very expensive, considering the cost of computer time and the delay from computer to programmer and back again.

It is obviously important, therefore, that a good postprocessor must be able to solve a maximum number of programme errors, go as far as possible through the part programme, and print out a clear and easily-understood diagnosis of each error. This produces a good machining tape with a minimum of computer runs.

HOW TO BUILD A POSTPROCESSOR

The modern postprocessor is built around the fact that routines for any one make of control are identical, that many routines for all machine tools are identical, and that many other routines are identical for all machines of a particular group (3-axis profilers, for example). The modern processor is, therefore, written around descriptions for such basic routines. This eliminates a tremendous amount of duplication. Once the basics are included, all that has to be added to accommodate any individual machine is the specific description applying to that one machine tool.

There are many advantages to this 'overlay' technique. Chief advantage is that to remove or add the routines for any particular machine, you simply remove or add the sections of punched cards that generate these segments in the computer. For example, a manufacturer acquires a programme that includes 30 routines. The plant, however, has only three different NC machine tools. In preparing the processor tape for this specific setup, the NC people simply signal the computer to delete the cards for the 27 routines they have no use for. This produces a compact, handy-to-work-with, easily maintainable, and reliable tape that will completely serve the company's current needs.

If at a later date one of the 27 other machines is added to the plant, the cards for that unit are simply put back into the tape. If a complete new machine is added even one not in existence at the time the processor was acquired-the necessary new cards are added to the tape.

In the same way, if there is some special application the company wants to develop, only a minimum amount of data will have to be prepared for it. This can readily be done in the company's own plant or by the machine tool or control builder.

Another advantage of the overlay technique is that it is complete for all machines included in the package - and is ready to operate. No additional descritpion or instructions have to be generated into this type of postprocessor by the user.

ADDITIONAL FACILITIES AND AVAILABLE SUBROUTINES:

Besides the previously described functions the postprocessor has several supporting tasks. The most important of them is the writing of listings: it writes a verification list which documents its own performance and therefore shows the contents of the control tapes, and it writes a list for the operator of the machine tool.

Another important task is the calculation of the production time. The postprocessor continuously has control over the path the tool will follow due to the commands contained in one tape record and it has control over the feedrate used. It is therefore easy to compute the time required to execute one tape record, the time needed for the execution of a part of the programme, and the time required to run the complete programme or the machine tool. On the verification listing the time required by the execution of one tape record will be printed after the data of this record. Frequently the time is summed over each page of this list and will be printed at the bottom of the page. At the end of part of a programme the intermediate sum of the times will be printed. At the end of the list the total time will be printed.

To support the planning the postprocessor may calculate the length of the control tape. It is not a difficult task, since the postprocessor can count the number of characters it inserts into the control blocks. This way it will be possible to know in advance how much tape will be needed for 'off-line' punching and how much time this will require.

Typical of now available on modern postprocessors are routines for the simultaneous operation of the ram and cross slide turrets of a Warner & Swasey NC turret lathe; routines for thread cutting on a centre lathe (all the programmer has to specify is the lead, the diameter, the length and type of thread to be cut, and the spindle speed): and linearization routines for machining centres like the Sundstrand Omnimil. These latter routines relate back into the X, Y and Z coordinates the linear movements that result from the twisting, tilting and rotary motions which are a prime feature of such 4-and 5-axis machining centres.

One of the most important and yet most neglected areas in evaluating NC processing equipment is the backup or supporting literature. If instruction and maintenance manuals are not complete, confusion results and the manufacturer loses in time money, and customer good will.

Here are a few of the key questions to ask in evaluating the support literature:

(1) Are part programming details so spelled out that they are easy to follow and practical to use?

(2) Do they emphasize special instructions and machine tool details?

(3) Do they provide sample programmes and tape formats?

(4) Is the computer manual completely flow-charted so that computer personnel can readily work with the programme?.

INTERPOLATION FACILITIES

In general the processor of the programming language, when evaluating part programmes for continuous path problems, approximates a non-linear tool path by a sequence of short cut vectors. The number of calculated cut vectors depends essentially on the quality of the surface desired by the part programmer. The computed points may almost immediately be used to control the machine tool in case only the linear interpolation of the control unit is used.

Besides the capability of linear interpolation, many controllers have also the capability of circular or parabolic interpolation. If one of these interpolation methods is to be applied to the data which will be ready by the controller from the control tape, the postprocessor processes accordingly the points of the non-linear tool path read in from the intermediate storage device. For a circular interpolator, for instance, the postprocessor calculates the sine or the cosine of the ending angle of an arc, or it determines the distance from the starting point of the circular arc to the centre of the circle. Since some circular interpolators can process a circular arc only within one quadrant, the postprocessor will determine if an arc extends beyond one quadrant, in order to segment it if necessary. The function codes that are required will automatically be inserted into the control blocks.

For some control units it is required that the part programmer determines what interpolation method is to be used at a given time (the linear instead of the parabolic interpolator may, for instance, be applied to a non-linear tool path). In this case the applicable interpolation method will be programmed by means of a postprocessor command in the part programme so that the postprocessor may process the data from the intermediate storage accordingly.

FEATURES OF A GOOD POSTPROCESSOR:

The best postprocessor is one that is coded in the most up-to-date language. Currently, that language is Fortran IV. It is the most complete language developed to date and is the one for which all the major computer companies are generating their latest NC processing models. A postprocessor written in some other language may not operate as efficiently on these new computers. In some cases, the postprocessor may not operate at all and will have to be re-coded.

There are several other major criteria to the selection of a good postprocessor. One such criterion is in the answer to the question 'As NC operations in the plant are expanded, will the postprocessor be serviceable for the new equipment or will another postprocessor be needed for every new machine?.

If each time another new machine is added the whole postprocessor operation has to be torn apart or possibly scrapped altogether, and if programmers and computer personnel have to learn a whole new programme, then the postprocessing operations are wrong and can be very expensive. Here is where the advantages of the overlay technique are obvious. Only a relatively few details about each new machine have to be added to this type of processor and it is ready to operate. The basic procedures have already been completely checked out in a programme that plant people are thoroughly familiar with.

Another criterion of a good postprocessor is whether or not it includes the new and the special machines that are currently available. This is important even if programmes for such new and special machines are not needed in the plant for the present. It is a good indication that the writers of the programme are progressive and forward-looking and will undoubtedly keep on top of future developments in the industry as fast as these come.

QUALIFICATIONS NEEDED TO WRITE POSTPROCESSORS:

The persons to be involved in coding a postprocessor should have the following qualifications. The person has to have a very good knowledge of the machine tool and of the control unit. In addition he has to know the programming of computers very well. Also required is some knowledge about 'system programming'. The latter is necessary since the postprocessor rarely is run as a detached programme but frequently is used in the framework of a system of programmes. The knowledge on how the postprocessor has to work in this system and how it is to be connected to it is of importance.

The basis of the postprocessor is a detailed definition of the functions of the machine tool and that of the controller. Plenty of time should be spent to define and describe these functions.

There are many sources from which a postprocessor may be obtained. Frequently the postprocessor will be furnished or coded by the manufacturer of the control unit. This if of advantage since the controller manufacturer has the personnel that not only knows the controller and the machine tools in detail, but often also knows how to programme computers. Besides, this firm naturally is most experienced in writing postprocessors for its controllers.

Sometimes the postprocessor is written by the buyer of the system. This should only be done by large companies who have the appropriate personnel available.

Another possibility would be that the manufacturers of the machine tool and of the controller jointly code the postprocessor. This has the advantage that possible uncertainties about a component of the system may be resolved by the most competent person.

A last but frequently used method is to give an order for writing a postprocessor to a data centre or a programming service. These frequently have qualified personnel.

Whatever source for the postprocessor is used, the buyer of a numerically controlled machine tool should ensure that an operational postprocessor either be delivered with the machine tool, or, even better, that it will be delivered several months before the arrival of the tool, so that it may be tested and possibly modified in order to have a fully operational postprocessor at the moment the machine tool is in operation.