

GE Time-Sharing Service



World Leader in
Time-Sharing Service

NUMERICAL CONTROL PARTS PROGRAMMING WITH REMAPT

. . . a new dimension in fast,
accurate tape preparation.

GENERAL  ELECTRIC

KILL SLIDE

My presentation today will show you how General Electric Time Sharing can simplify tape preparation for numerically controlled machine tools. We have developed a new program... I can almost say a new technique... to make tape preparation easier and less complicated, particularly working with NC lathes. Today I'd like to introduce you to it, and to give you a detailed illustration of how it works.

SLIDE 1

For those of you who find Time Sharing a buzz word, let me take just a brief moment to tell you what it is and what it does. Those of you who are familiar with time sharing know that this is the modern concept that enables a number of people to use the computer system simultaneously... through the medium of remote terminals equipped with a telephone dial for access to the system and a teletypewriter keyboard for conversation with the system. Perhaps even

SLIDE 2

more important than the easy access and ready availability is the cost-sharing feature of Time Sharing: you pay only for the time the computer is working for you.

SLIDE 3 Let me say at the outset that General Electric has not
SLIDE 4 only emerged as the world leader in time sharing services and
development, but it has been recognized over the years as the
Number One manufacturer of numerical control equipment
as well as one of the world's largest users of NC machine
tools.

SLIDE 5 Being the leader in both Time Sharing and Numerical
Control it was only natural that GE would be the pioneer in
blending the two concepts. Actually, we've run into a large
number of problems ourselves as a user of more than 500 NC
tools, we've been able to analyze a wide variety of problems
and measure the effectiveness of the solutions over the years.

SLIDE 6 In February 1968 we brought out a series of programs on
point-to-point equipment. The first applications were well
received, and we realized we had something that really
pleased our customers.

SLIDE 7 Since the successful introduction of making NC tapes on
time sharing, we have expanded and improved these programs
several times. We steadily improved our service as we
discovered new customer problems.

SLIDE 8 The most recurrent problem is the slow turnaround with
batch computers, but there are others that are significant, or
should I say bothersome:

SLIDE 9 There's the question of who does what. In most
installations, manufacturing personnel have responsibility for
the tapes but no authority to schedule the computer. As a
result, the user frequently has to seek out the function
responsible, probably the Finance operation, and wait his
turn to use the system.

SLIDE 10 When you use an outside service bureau, there are the
problems of communication and delivery, and, just to
mention one other, there's the situation in which many jobs
are done manually because the computer is inaccessible (the
Finance or Inventory boys have it all tied up).

SLIDE 11 As part of our continuing expansion and improvement we now are offering a complete new programming language called REMAPT.

REMAPT is both a part programming language and a computer program that processes the language. It is a subset of the widely-used APT language.

SLIDE 12 Now let's look at what REMAPT does. It takes the ordinary-English-type statements that describe the part to be made, and converts them into the control tape to be used for directing the NC tool.


This procedure operates in the Time Sharing environment. The complete operation is controlled by the part programmer from the terminal conveniently located in your shop or office.

SLIDE 13 In operation, the part programmer describes the geometry of the part and the machine steps required to make this part. This is known as the part program. Six major types of programming statements are available in the REMAPT language. Let's look at them one by one:

SLIDE 14 There are geometric statements, which describe the elements that comprise the finished part. A point is defined by giving it a symbolic name. For example, P1 and its X, Y and Z coordinates.

Symbolic names can be chosen to describe what they represent. The center point of a circle might be defined as you see in Statement No. 120. Having done this, a circle... C1,,,can be identified by its center point, CENTR, and its radius 5. A line from point 1 which is tangent to the right side of the circle 1 is defined in Statement 140. There are 61 different ways of defining part geometry in the REMAPT system.

SLIDE 15 Then there are tool movement statements, which direct the cutter along the previously defined part elements. The cutting tool may be directed to a point by the "GO TO"



statement. In Statement 160 we position the tool on the right side of the workpiece and tell it to cut along Line 5 until it reaches Circle One. From that location we are instructing the cutter to go forward around the circle until it reaches the check surface which is defined within the parenthesis. The check surface in this case is a line perpendicular to L1 and passing through the center of the circle.

With this command we have caused the tool to cut 180 degrees around the circle. Twenty eight tool movement commands are recognized by REMAPT.

SLIDE 16

↓ Computation statements allow the part programmer to define the workpiece from whatever dimensions are available on the engineering drawing. REMAPT performs all the needed calculations, usually by standard trig functions...sine, cosine and arc tangent. When programming for a lathe, you can include a statement such as Number 175 in a loop to cause a series of cuts five-one hundredths deeper than the previous cut.

SLIDE 17

The ↓ machine function statements, or as they are sometimes called, post-processor statements, control the auxiliary functions of a machine tool. Auxiliary functions are those machine actions that must occur before or after actual cutting.

Here are three of the 120 machine function statements that are recognized by REMAPT. The characteristics of your NC tool and the post-processor you are using will determine which of these statements are valid for a particular program. In this example we are selecting the tool in Position 3, turning the coolant on, and turning the spindle on at 1800 RPM in a counter-clockwise direction.

SLIDE 18

Then there ↓ are system control statements. These are used by the part programmer to direct the processing of his program by the computer.

Statement 210 instructs the system to use the GELATH post-processor for Lathe No. 5, and I will explain GELATH in a few moments . Next we request that the cutter location file be saved for possible later examination. In Statement 230 the part programmer decides to make his program statements shorter by creating synonyms for some commonly used instructions. After this statement he can write "GF" whenever he wants the tool to go forward, or "C" when he wants to define a circle.

SLIDE 19

The wide variety...and number...and flexibility of these system control commands can save you time and money while de-bugging your program. ↓ For instance, commands such as FIND, PRINT, AND BACKUP may be used to selectively examine the cutter location listing, the post-processor listing or the punched tape listing.

SLIDE 20

Now we'll get to the sixth and last of the major types of statements available to the part programmer.

↓ The program information statement is an explanatory statement not required by REMAPT, but one that is extremely helpful to the part programmer as a debugging and/or documentation aid.

SLIDE 21
(KILL SLIDE)

He may insert a remark statement wherever it adds to the clarity of the documentation. For short notes he may put a double dollar sign after the normal REMAPT statement and follow this with operator instructions or an explanation of the program. ↓

Using these six types of statements, the part programmer defines the job to be done and REMAPT calculates a generalized solution.

You recognize, of course, that a post-processor is then needed to convert the general solution, or point coordinates, created by REMAPT to fit the specific NC tool you have.

This must consider the format requirements of your controller, and the options or special features you've chosen for your NC tool.

SLIDE 22

The first of a wide variety of post-processors GE is developing to work with REMAPT is a generalized post-processor for NC lathes called GELATH.

SLIDE 23

GELATH performs all the post-processor functions that are normally required for a lathe. In order to tailor GELATH to the specific requirements of your lathe, you use a machine parameter file.

This, I realize, is a term you have not heard before, so I'll take a moment to explain that the machine parameter file is actually a table of specifications about your NC system. It would include information on such things as

- spindle speeds
- register formats
- preparatory functions
- miscellaneous functions
- limiting feeds and speeds, and so forth.

This concept of a machine parameter file is a major simplification of the long-standing post-processor problem. By isolating the machine specifications into a simple data table and including all the coding logic in the generalized post processor we have minimized the time and effort required to adapt GELATH to a specific lathe.

Machine parameter files are already available for a good number of lathes. In many cases you can get them from your lathe supplier, but in the event they are not available, General Electric can work with you to develop an MP file for your lathe.

Now I'd like to take you step-by-step through a sample program to illustrate how REMAPT/GELATH can work for you.

SLIDE 24

The part programmer turns to his Time Sharing terminal and dials the number of the computer. By typing in the

SLIDE 25

information you see underlined on this actual printout, he identifies himself to the computer, indicates he wants to use the REMAPT system, and declares that he is going to enter a new part program named NCDEMO. The computer digests this information and returns to say READY.

SLIDE 26

The part programmer then begins entering his part program. Also, in Line 130, he provides the identification that will be punched at the beginning of the control tape.

Then he reserves some special storage areas, and establishes "XL" as a synonym for "X-Large."

Statement 160 instructs the system to use the GELATH post processor for Lathe No. 1. He then requests that the cutter location file be preserved for possible later examination at his own terminal.

SLIDE 27

The part programmer specifies cutting tolerances and tape header information. He also specifies spindle speed and feed rates, and turns the coolant on.

SLIDE 28

Statements 260 through 340 define the geometric elements of the part. You'll recognize the normal definitions for a setpoint, a plane, several points and some lines. However, note statements 320 and 330, where the symbolic point-name is a subscripted variable. You'll see the significance of this later on.

SLIDE 29

The part programmer defines the machine movements to affect the desired cut. These commands are in the form of what we call a "macro" instruction, which is a technique for storing a group of instructions for execution later, generally for repetitive purposes.

SLIDE 30

Now we get into the actual tool movements. Line 465 establishes the starting point. The subsequent statements cause our "macro" to go into action until the appropriate number of cuts have been made.

SLIDE 31

This series of commands causes the macro to be re-executed for a new series of cuts three inches to the right of the first cuts. This is accomplished by Statement 560, which changes the values to be used by the macro. Statement 570 assures that the correct number of cuts will be made. The remaining statements stop the machine action and rewind the tape. ↓

SLIDE 32

The part programmer now tells REMAPT to run his program. But the computer detects an error in Statement 430, notes the fact that there are errors, and asks what kind of output is desired.

Since there was an error, the parts programmer types "DONE" indicating no output is desired at this time. The computer prints out the time used.

The part programmer then corrects his error by changing statement 430 which referred to an undefined symbol P3. The correct symbol P1, is entered and he tells the system to re-run the program.

Next, REMAPT indicates that there are no further errors and asks what output is desired. The part programmer then answers by indicating that he wants a listing of the cutter location file at his terminal. ↓

SLIDE 33

Here is a portion of the cutter location file, which would be printed, of course, in its entirety at his terminal.

SLIDE 34

After examining the cutter location listing, the parts programmer instructs the computer to print the post-processor listing at his terminal. ↓

SLIDE 35

This post-processor listing is actually a detailed listing of every operation to be performed on the machine. It is printed in a format that can easily serve as a guide to the machine operator.

SLIDE 36

When the computer indicates that the list is complete, the part programmer may request a listing of the actual control tape.

