Steve Schmidt

PART PROGRAMMING MANUAL

PUB 929A

ADVANCED PROGRAMMING LANGUAGE (KT-APL)

KT-GEMINI-D

MACHINE CONTROL SYSTEM

®

Kearney & Trecker Corporation

A Cross & Trecker Company

SAFETY
EVERYBODY'S BUSINESS

PUB-578

First Edition — May, 1976

Kearney & Trecker Corporation
A Cross & Trecker Company

PREFACE

SAFETY . . . Everybody's Business

A metal cutting machine tool can be a very safe machine. Or you as a user can make it be a very dangerous one. The difference depends on you . . . and the other people who are required to perform functions relating to the machine tool operation. Yes, this also depends on management and others who may be observers of the metal cutting machine tool. Everybody must become safety conscious. If everybody thinks safety, accidents can be prevented. If an effort is made by everyone to understand and follow the basic safety procedures that follow and also the safety rules prescribed in your own company . . . you can prevent serious injury to personnel and costly damage to equipment.

We recognize that these suggested safety precautions are a beginning and not an end. They cannot include all of the eventualities that might arise in all companies . . . and in all types of operations on all types of metal cutting machine tools.

So when you are following these suggestions, add a large amount of common sense attitudes and good judgment along with them.

Always think safety! Remember, a metal cutting machine tool can be very safe or very dangerous. It depends on you. It is the intent of the Kearney & Trecker Corporation to update this safety information. We would welcome comments or suggestions from our product users. Your input will be given our serious consideration for inclusion in the next update of this information.

Some parts of the information contained in this booklet are also included in the front of each manual. THE CUSTOMER RECEIVES THESE MANUALS WHEN HE PURCHASES A KEARNEY & TRECKER PRODUCT.

As you read this safety literature, you will notice that it is addressed to all occupation skill groups that are concerned with machine operation, supervision and maintenance. Since the Kearney & Trecker Corporation believes that safety is everybody's business, we are not attempting to separate these safety guide lines where separate manuals would be required for the machine operators, supervisors, maintenance technicians, etc. We believe that you will agree that this is a good approach. Please take time to read the ENTIRE manual.

When an accident occurs, investigation may reveal the cause and pinpoint what should have been done to have prevented the accident. This can only provide HINDSIGHT. The purpose of a good safety program is to prevent the accident from occurring in the first place. This can be accomplished by the use of FORESIGHT and knowing the results of our actions when working around machine tools.

THINK SAFETY — WORK SAFELY

KEARNEY & TRECKER CORP.
MILWAUKEE, WISCONSIN

GENERAL SAFETY REMINDERS

1. The employer is responsible for selecting competent and qualified employees. Kearney & Trecker conducts training classes. We strongly suggest you enroll your employees.

2. MANUALS ARE PROVIDED WITH EACH MACHINE TOOL. THE USER SHOULD HAVE THESE MANUALS AVAILABLE FOR THE PERSONNEL WORKING WITH THE MACHINE TOOL.

3. Guards and shields are to be in place at all times.

4. Operators must be forbidden from changing workpieces or performing other work close to a revolving cutter . . . Stopping the spindle is a good work habit.

5. When contaminants are released to the atmosphere as a result of machining, an exhaust system is recommended.

6. All employees should be aware of first aid facilities and be encouraged to use same, regardless of the severity of the injury.

7. Employees should be encouraged to report to their supervisors any hazardous condition on the machine tool or in the immediate area.

8. Air hoses — it is recommended that air hoses not be available for use around the machine tool. Flying chips may cause personal injury or damage equipment.

9. Levels of illumination in the immediate area of the machine tool should comply with National Standards.

10. Fire prevention must be practiced and fire protection available to prevent loss of life, personal injury and to protect property.

11. Protective footware — safety shoes afford good protection against rolling objects, dropping heavy objects, accidentally kicking sharp edged materials. Comfortable protective footware is a good investment.

12. Eye protection — eye protection devices must be considered as optical instruments and they should be carefully selected, fitted and used. Compulsory wearing of spectacles with impact resistant lenses is a good safety policy.

13. The employer must insist that his employees study this safety information and practice safety.

14. Users must have available adequate lifting facilities capable of lifting within the safe load limits; also appropriate slings and hitches.

15. Every floor opening resulting from the machine tool installed in pits or below floor level should be covered with a material of proper strength with non-slip features.

16. When machines are installed above floor level and a platform is built around the machine tool, the platform must be constructed of materials of proper strength with non-slip features of the walking surfaces. Railings and toe boards must be installed where appicable.

17. Management personnel have the responsibility to alert operating personnel of any unsafe practice they may observe.

MACHINE TOOL SAFETY

Metalcutting machine tools play an essential role in a modern society; HIGH HORSEPOWER, SHARP CUTTING TOOLS, RAPIDLY MOVING PARTS, HIGH VOLTAGE, and AUTOMATIC OPERATION are necessary ingredients of a satisfactory machine tool, and are all POTENTIAL HAZARDS!

The METAL REMOVAL process often results in rapidly propelled metal chips and sharp edges and burrs on the finished workpiece. These are POTENTIAL HAZARDS! Due to these potential dangers inherent in a machine tool, protective guards, safety design features and warning signs are utilized. For MAXIMUM PERSONAL SAFETY it is imperative that all operators, maintenance personnel, observers, and all others that could be exposed to inherent machine hazards are made FULLY AWARE of potential dangers, and are THOROUGHLY INSTRUCTED in the SAFETY PRECAUTIONS they MUST FOLLOW to avoid those dangers. It is essential that persons required to become involved with the machine are properly trained and have the required knowledge and skill to perform their respective functions.

If assistance is required, contact your local Kearney & Trecker representative. Adherence to the following list of recommended SAFETY PRECAUTIONS, plus the application of common sense attitudes, will reduce to a minimum the possibility of personal injury resulting from the use of a Kearney & Trecker machine tool.

FOR MAXIMUM SAFETY AROUND MACHINE TOOLS:

1. BEFORE you START a machine, be sure you know what is going to happen. KNOW your MACHINE.

2. Be sure you know how to STOP the machine before you start it. KNOW your MACHINE.

3. MAINTAIN the machine in GOOD OPERATING ORDER. Report unusual conditions or machine malfunctions immediately.

4. Always STOP your CUTTER before REMOVING a finished WORKPIECE from in front of the spindle. Do not place your hands near a rotating tool.

5. Never attempt to perform any cleaning, chip removal or work piece clamping while units are in motion. This is like running a race and some day you may lose the race.

6. If you are assigned an assistant for any reason, both the assistant and the operator have the responsibility of deciding who will be in command of the machine and its controls. Only one person should control the machine. ANYONE ELSE SHOULD STAND CLEAR AND BE VISIBLE TO THE PERSON WHO IS ASSIGNED TO OPERATE THE MACHINE CONTROLS. Also, be alert for any bystanders or unauthorized person who may be in the area of the machine travel limits and chip disposal areas. An area that may not be a hazard to the machine operator at his control station may be hazardous to an assistant or bystander.

7. Remember that your work area may change during the day as material is delivered to and removed from your machine area. Be alert for pinch point and work hazard areas created by workpiece storage.

8. Keep the immediate area clean. Avoid slippery floors, remove debris, remove obstacles, remove chips, etc.

   Chips that are allowed to accumulate in the area in which you must walk are a hazard that can cause you to fall or slip against the machine or its controls.

9. Remember that many accidents occur that are not related to the actual machine operation and actual metal cutting process. Accidents during material handling represent a large percentage of reported personal injuries.

MACHINE TOOL SAFETY (Cont.)

10. Report any HYDRAULIC OIL LEAKS OR COOLANT LEAKS. These should be CORRECTED IMMEDIATELY and not be allowed to accumulate causing an area to become unsafe for walking.

11. ONE COMMON CAUSE OF INJURY IS ATTEMPTING TO PERFORM MAINTENANCE OR CLEANING WHILE A MACHINE IS IN MOTION OR AUTOMATIC MODE. ANOTHER IS ACCIDENTAL START UP OF THE MACHINE BY A SECOND PERSON WHILE ONE MAN IS PERFORMING MAINTENANCE OR LUBRICATION.

    Never attempt to perform maintenance or lubrication while the machine is in operation. The operator and the person(s) performing maintenance must be mutually aware of each other's presence in the machine area.

    During maintenance or lubrication, the machine should be taken out of service. THE CONTROLS SHOULD BE PROPERLY MARKED WITH A WARNING SIGN AND LOCKED.

    Re-read Items 5 and 6 above.

    Remember that no one will normally be injured by a machine that is taken out of service. Use all precautions to avoid accidental start-up of a machine while it is being serviced.

12. DULL cutting tools are DANGEROUS. DON'T use them. Make certain that all

    CUTTING TOOLS are properly SHARPENED and correctly assembled to appropriate tool holders, arbors, etc.

13. NEVER use cutters, wrenches, or other tools that do not fit the machine properly.

14. Be sure the workpiece is held SECURELY in the vise or fixture.

15. Be sure that all protective GUARDS are in place BEFORE the machine is STARTED.

16. NEVER apply a wrench to MOVING work or parts.

17. DO NOT attempt to ADJUST a tool while the machine is RUNNING. Shut off machine at main disconnect.

18. Be sure of CLEARANCE between CUTTERS and WORKPIECE and FIXTURES before STARTING the machine.

19. DO NOT ADJUST COOLANT FLOW OR DELIVERY NOZZLES with the spindle running. Hands or coolant nozzles may be pulled into the revolving cutter resulting in personal injury or damage to the machine.

20. Be sure that the machine's moving parts are kept CLEAR of all LOOSE OBJECTS, TOOLS, CHUCK WRENCHES, ETC. Do not use the machine as a work bench.

21. DO NOT attempt to BRAKE or SLOW down moving machine parts with your HANDS or MAKESHIFT devices.

22. DO NOT CLEAN a machine with an AIR HOSE. Flying chips can cause personal injury or damage to machine.

23. NEVER CLIMB on or around machine on makeshift devices such as skids, tote pans, boxes, etc. Use only SAFETY APPROVED LADDERS.

24. Be AWARE of conditions that may be a FIRE HAZARD, such as volatile liquids and machining materials with a low flash point.

25. DON'T attempt to use the machine BEYOND its designated CAPABILITIES.

26. NEVER attempt to MEASURE MOVING WORKPIECES in the machine, or NEAR the revolving CUTTER, always stop cutter and machine motion when measuring.

27. The machine should NOT be left RUNNING while UNATTENDED.

28. When making repairs, be sure to take all NECESSARY PRECAUTIONS to prevent PERSONAL INJURY from LOOSENED or UNSUPPORTED machine parts or units.

29. DO NOT wear RINGS, WATCHES, JEWELERY around moving machinery.

30. DO NOT wear LOOSE fitting CLOTHES. Clothing should be comfortable, but LONG SLEEVES, NECKTIES, etc., should NOT be WORN.

31. SHOP APRONS are deliberately made with a light tie-string for a purpose; do NOT REPLACE the light string with a heavy cord or wire.

32. Wearing of SAFETY SHOES is recommended.

33. WEARING of GLOVES or other hand covering articles is not recommended around moving machinery. GLOVES MUST BE CONSIDERED THE SAME AS LOOSE FITTING CLOTHES.

34. ALWAYS WEAR approved SAFETY GLASSES when operating metal cutting machine tools. Clear vision helps you do good work as well as avoid accidents.

35. NEVER LEAN on the MACHINE. You may inadvertently set it in motion, or fall into a moving cutter.

36. LONG HAIR should be covered with a protective cover such as a HAIR NET.

37. NEVER use a RAG or other material to remove chips from a revolving cutter. Use a BRUSH provided for such application.

38. NEVER handle SHARP CUTTERS BAREHANDED. Wrap the cutter in a rag. See also Items 47 and 48.

39. ALWAYS be sure the WRENCH fits the nut. Use a straight arm body pull — do not jerk. Be prepared for the bolt to shear or the wrench to slip. KEEP BALANCED.

40. In using any WRENCH, it is better to PULL than to PUSH. If it is necessary to PUSH, use the open palm. When using an adjustable wrench, exert handle pressure toward the moveable jaw. Stand to one side when you are pulling down on wrenches above your head. DO NOT hammer on a wrench with a hard hammer; if necessary, use a babbitt pounder.

41. Screw drivers should not be used as a WEDGE or CHISEL or PUNCH.

42. Pliers SHOULD NOT be used as a wrench.

43. DO NOT use a file as a pry bar.

44. When filing DO NOT use a file that does not have a HANDLE firmly attached and covering the pointed tang.

45. Hand tools expecially those with POINTS and SHARP EDGES should NEVER be carried in pockets or under trouser belts. Use a tool box or a tool belt.

46. NEVER misuse compressed air as a practical joke. KEEP air at PRESSURE LEVEL recommended by industry standards. Improper use of an air hose may cause permanent disability or even death.

47. When handling cutters which are already mounted on tool holders, grasp the tool holder in the area of the flange, not the area of the cutter. On tooling for MACHINING CENTERS, the code ring area provides a safe place for handling the cutter/tool holder assembly.

    On larger size cutters (exceeding approximately 5-inch diameter) or where the cutter/holder assembly must be changed manually, or where the tool must be inserted horizontally, it will be found practical to use a rag to protect the hands.

48. On STANDARD KNEE TYPE MACHINES with a POWER DRAW ROD unit, heavy cutter/arbor assemblies should be supported on the machine table in suitable cradles and/or wood blocks. The use of the knee, table and saddle axis movements allows you to position the arbor for easy alignment to the spindle and drive keys. This nearly eliminates the possibility of cross threading the start of the arbor and draw-bolt threads. Let the machine do the work for you. This is easier and safer instead of trying to insert a heavy arbor/cutter assembly with your own strength. Use the same wood cradles or blocks when removing a heavy cutter/arbor assembly. This will avoid the possibility of personal injury and also prevent possible expensive damage to the cutters in the event you should drop the heavy cutter/arbor assembly onto the machine table.

ELECTRICAL SAFETY

1. All electrical/electronic troubleshooting and repair should be undertaken only by personnel who are properly trained and have adequate knowledge and skill.

2. It must be assumed at all times that POWER is "ON" so all conditions must be treated as live. This practice develops a caution that may prevent an accident.

3. REMOVE LOAD from circuit or equipment.

4. OPEN disconnect or breaker and (a) lock in open position or (b) remove fuses or (c) install "Hold-Off" tag or (d) have someone stand by. A switch in the open position with a lock that you only have a key for is positive protection from someone accidentally turning power on while you are working.

5. MAKE CERTAIN you have opened the circuit by using the proper test equipment — NOTE: Test equipment must be checked at regular intervals.

6. Capacitors MUST be given time to discharge, otherwise it should be done manually with CARE.

7. When trouble-shooting "LIVE" equipment, the necessary precautions must be taken as follows:

   a. MAKE CERTAIN your tools and body are clear of ground.
   b. Use extra PRECAUTION in DAMP areas.
   c. BE ALERT and work without any outside distraction.

8. BEFORE applying POWER to any equipment, it must be established without a DOUBT that all persons are CLEAR.

9. The CONTROL PANEL DOORS shall be open ONLY when it is necessary to check out the electrical equipment or wiring. After CLOSING the door, make certain that the disconnecting means is operating properly with the DISCONNECT HANDLE MECHANISM.

10. ALL COVERS on junction boxes must be CLOSED before leaving any job.

11. BEFORE STARTING, read and understand all WARNING markings and notices.

ELECTRICAL SAFETY (Cont.)

12. READ all marking such as nameplates and identification plates.
13. DO NOT alter circuits unless authorized to do so by the manufacturer.
14. DO NOT alter or by-pass protective interlocks.
15. DO NOT place jumper wires across fuses.
16. DO NOT alter overcurrent protective devices.
17. WHEN conductors are REPLACED, they shall conform to the manufacturer's SPECIFICATIONS, including PROPER COLOR CODING.
18. GROUND connections cause fault currents to flow directly into the ground instead of following through the body into the ground. ALL ELECTRICAL APPARATUS MUST BE PROPERLY GROUNDED.
19. Use CAUTION when connecting test equipment probes to test points. SHOCK HAZARDS could exist at the test points or in the test point area and/or TRANSIENTS induced by the probes could cause a MACHINE ACTION.
20. AVOID wearing glasses that have a METAL FRAME.
21. AVOID wearing a necklace or chain that is made of a METALLIC substance.

LIFTING EQUIPMENT

There is a variety of accessories that might be needed for a lifting or turning job.

1. Hooks — They are made in many shapes and sizes. The hook must not be loaded over its capacity and must be used properly.
2. Chains — Not recommended for heavy lifts because flaws are difficult to detect and they are too cumbersome to handle.
3. Manila Rope — Provides a good grip on the job and is light and flexible to handle. Must be protected against the weather and other destructive elements such as solvents, oil, and heat. Manila rope should be inspected frequently.
4. Nylon Rope — Gradually replacing manila rope. Basically the same strength and not affected by destructive elements.
5. Wire Rope — Most slings are made of wire rope. It is preferred because it is stronger than fabric ropes; it is flexible and can be shaped to most lifting requirements.

Remember, your first responsibility is to practice safety. Before you begin any lift, we suggest you go through a mental check list. Ask yourself questions like the following:

1. What is the weight of this load?
2. What type of accessory, hitch or connection is required?
3. Will the lift be a straight lift or whether an angle rig is required. This determination will affect the lifting capability of the accessory.
4. Are the slings free of kinks, knots, or broken strands.
5. Is proper clearance available to make the lift safely.

For further information on hoisting your Kearney & Trecker products, we refer you to the Product Installation Instructions provided with your machine.

PERSONAL PROTECTIVE EQUIPMENT

When involved in the lifting of heavy loads, we suggest you protect yourself by wearing the following protective gear.

1. Safety Shoes — Foot injuries can be eliminated or their seriousness can be reduced through the wearing of approved safety shoes.

2. Safety Glasses — Eye protective devices must be considered as optical instrument and they should be carefully selected, fitted and used.

3. Safety Helmets — Because of the possibility of falling objects, the head is exposed to hazards, therefore, head protection devices should be worn.

4. Gloves — Although it is recommended that gloves not be worn unless absolutely necessary, gloves can protect the hands from cuts or abrasions caused by slings or sharp edges.

LIFTING TIPS

Because the type of hoisting equipment in each customer's facility is unknown to the Kearney & Trecker Corporation, these instructions are written with the intent of being general rather than specific. However, whether you have in your facility a bridge, jib, gantry, semi-gantry, cantilever wall crane, or some other lifting crane, these safety rules will apply.

In some cases where adequate hoisting equipment is not available, it may be necessary to contact an outside contractor. Whatever your conditions are, you must practice good safety procedures. The operation of the hoisting equipment may not be your full time job, but if you are involved when hoisting Kearney & Trecker products, you must follow all of these good safety practices. Remember, a child could manipulate the controls. Your job is safety.

1. NEVER lift more than the rated capacity of the hoisting equipment.

2. If in doubt, have the hoisting equipment INSPECTED for safe operating conditions. INSPECT all slings. DON'T TAKE CHANCES, if in doubt, check with proper authority.

3. BALANCE LOAD in sling before LIFTING more than a few inches. Distribute load evenly.

4. Use a sling LARGE enough for the LOAD.

5. To PREVENT damage from the slings to the finished parts of the machine, be sure machine parts are properly PROTECTED by PADDING.

6. ALWAYS be sure your slings are free of knots and kinks. Also, SLACK must be taken out of the sling before lifting.

7. TO PREVENT a serious accident that may be caused by sharp corners and edges cutting the sling, use proper BLOCKING and PADDING.

8. CLARIFY HAND SIGNALS with co-workers. If signals are not understood, make no move until they are clarified.

9. DO NOT lift loads with extra slings or cables suspended from the hook.

LIFTING TIPS (Cont.)

10. ALL loads must be lifted gradually. Sudden jerks may DAMAGE cables or slings.

11. BEFORE lifting a machine or a component of a machine VERIFY that all interrelated connections are disconnected.

12. BEFORE you signal to have a load lifted, be sure that all tools, parts, chips, coolant, oil, etc. are REMOVED from the LOAD.

13. DO NOT carry loads over aisles unless absolutely necessary.

14. As the lift starts OBSERVE the slings or lifting devices to assure that they are FUNCTIONING PROPERLY.

15. BEFORE making the lift, ask all employees to STAND CLEAR. NEVER allow a person to work under the load while the load is suspended from the crane hook.

16. When setting loads down, prepare the landing area beforehand and DON'T create a condition of LEANING and UNSTABLE units.

17. When lowering the load on blocks, the blocks MUST be even in height for a LEVEL and STABLE load.

18. DON'T set your load down in aisles. It is your duty to keep AISLES CLEAR for other employees.

19. ALWAYS be suspicious of a heavy load which must be lifted by eye bolts and NEVER lift such loads at an angle unless the eye bolts are short shouldered.

20. Between lifts, DON'T leave slings or hooks or other equipment lying around on the floor.

21. ALWAYS use good horse sense and good judgment in handling all loads.

PART PROGRAMMING MANUAL

PUB 929A

ADVANCED PROGRAMMING LANGUAGE (KT-APL)

KT-GEMINI-D

MACHINE CONTROL SYSTEM

FIRST EDITION................... September, 1986
SECOND EDITION....................October, 1987

Kearney & Trecker Corporation

A Cross & Trecker Company

11000 WEST THEODORE TRECKER WAY MILWAUKEE, WI USA 53214-1127
TELEPHONE 414-476-8300 TELEX 026-663

Table of Contents

CHAPTER 3 MISCELLANEOUS FUNCTIONS

CHAPTER 4 TYPE II DATA STATEMENTS

CHAPTER 5 PROGRAMMING MACHINE FUNCTIONS

CHAPTER 6 OFFSETS and CUTTER COMPENSATION

CHAPTER 7 PROGRAMMING ADDITIONAL OR OPTIONAL FEATURES

CHAPTER 8 PARAMETER AND SUBROUTINE PROGRAMMING

CHAPTER 9 PROGRAMMING EXAMPLES

CHAPTER 1

GENERAL INFORMATION

INTRODUCTION

KT-GEMINI-D CONTROL SYSTEM

The KT-GEMINI-D Control, with the KT Advanced Programming Language (KT-APL) Executive System, provides a host of standard and available features geared toward optimizing part processing efficiency.

For the part programmer, decimal point programming eliminates the need for leading or trailing zeros. Type II Data statements enable programmable operator messages and facility in tape loading tool number and offset value information. The file editing system allows rapid and comprehensive program editing during part proveouts.

For the machine operator, the control system has two [SELECT] menu pages of displays for quick referencing of machine functions, axis positions, offsets, and other functions which are essential in the total control of the part processing cycle. The Manual Data Input (MDI) feature enables rapid input or alteration of part program data. The self-prompting file handler system makes for ease and speed in the control of file data.

The KT-GEMINI-D Control system is designed to provide flexible, comprehensive and easily managed control over part processing operations. In addition to those mentioned above, the control also makes available the following features:

PROGRAMMING FEATURES

- Decimal point or trailing zero format
- Parameter Programming
- Switchable inch / metric
- Subroutines
- Absolute or incremental programming
- Comment Statements
- Floating zero programming
- Go To Statements
- 3, 4, or 5 axis positioning and contouring
- Boolean Expressions
- One-block 360° circular interpolation
- User Defined Cycles
- Circular interpolation with radius programming
- Arithmetic Functions
- Block delete
- Axis inversion (mirror image)
- Fixed cycles
- Multiprogram storage (hard disk)
- Tool length compensation
- Tool diameter compensation
- Fixture offsets
- Adaptive control
- Spindle probe orbital transformer

OPERATING FEATURES

- Axis position display
- Axis offset display
- Spindle speed and feedrate override
- Block search
- Axis offset
- Manual jog - continuous or incremental
- Auto or block-by-block mode
- Portable control panel (optional)
- Error recovery display

MACHINE PROGRAMMING LANGUAGE

The part program is a series of commands which move the machine tool in a predetermined sequence at indicated feeds and speeds. The end result of a correct part program is a properly machined workpiece.

PRELIMINARY FORMS & DRAWINGS

The process of writing the part program begins with a part drawing which the programmer must code, and a setup sketch which shows the location of the part on the fixture. The programmer must select the proper tools from the tool library and complete a Sequence of Operations and Tool Description Sheet. When the sequence of machining operations has been derived, the coordinate points within the machining envelope are calculated and are documented on a Coordinate Sheet. The next step is to write the actual part program. This can be written on a form such as the Process Planning Sheet (Manuscript).

PALLET MAPPING

With the advent of KT-APL, a new feature called pallet mapping is now included as a standard feature of this controller. This system of Pallet Mapping differs from that which was previously available on System 10 Machine Controllers in that the pallet map is now included in the Program Management Table.

A Pallet Map consists of from 1 to 75 standard part programs as opposed to the previous 1 Pallet - 1 Program method of programming. This allows the programmers to write dedicated programs to specific operations which may be intermingled among similar workpiece programs (part families).

A secondary benefit is the ability to fixture multiple workpieces on the same pallet/fixture with each part having its own program. The operator merely defines which programs are to be included in the pallet map during program executions for each specific pallet.

Program operations are keyed to the currently active pallet number, which may have up to a maximum 75 program entries.

NOTE

The 75 program entries refers to the maximum capacity of the Program Management Display. If a pallet map contains 75 entries, no other pallet maps may be created.

Both Non-Shuttle machining centers, and Standard Shuttle machining centers may use pallet numbers from 1 to 255 to indicate pallet maps. This is because there is no physical pallet coding required on the pallet itself. Machines which use coded pallets such as Unattended Manufacturing Centers, Unattended Machining Centers, and KT/FMS Machine Modules are restricted to 127 Pallet Maps, as this is the maximum allowable pallet code number.

Part programs are written in the same manner as always. However, only the last M30 (End of Program /Pallet Shuttle) is actually recognized as the end of program command. All preceding M30 commands for any previous part programs are considered an "End of Operation" command.

PART PROGRAM PREPARATION

When complete, the information contained on the manuscript is translated into machine-readable code by means of a tape preparation system. The resulting punched tape contains commands in the letter address system, which the control can recognize and process. The control directs the machine tool movements and functions throughout the entire operation of machining the workpiece. This operation is called the "part processing cycle".

BLOCK UNIT INPUT

The basic unit of the part program is the "block" or "data block". Each block may consist of commands for axis motion, establishment of preparatory functions, and for implementing miscellaneous functions. The block may contain a single command or several commands; however, a properly constructed data block will contain no conflicting commands. The block itself may consist of a maximum of 80 characters including the sequence number, any spaces which may be present between commands, and the end of block character which terminates each block. There is also no restriction on the number of non-conflicting G or M codes written within a single data block, except for the 80 character limit as mentioned above.

LETTER ADDRESS SYSTEM

The letter address system is used by the KT/CNC control to identify and separate commands contained within a block of information. Each command consists of a letter followed by a number in the proper format. The letter designates the type of command, the number designates the specific information. The following letter address words are recognized by the control:

N - Sequence number
G - Preparatory function
X - X-Axis command
Y - Y-Axis command
Z - Z-Axis command
A - A-Axis command
B - B-Axis command
C - C-Axis command
R - Fixed cycle clearance plane, Radius for circular interpolation, Surface separation for spindle probe
E - Fixed cycle incremental feed distance, Tolerance zone for spindle probe cycles
F - Feedrate or dwell
S - Spindle speed or programmable keylock position
M - Miscellaneous function
T - Tool select
I - Arc center offset
J - Arc center offset
K - Arc center offset
H - Fixture offset

LETTER ADDRESSES & FUNCTIONS

Axis Related Commands

Axis Departure (Linear Axes)

Letter Address: X, Y, Z

Maximum Number of Digits: inch-8; metric-8
Format: inch-(4.4); metric-(5.3)
Range: inch; 0 to ±1690.9320 (0.0001" resolution)
metric; 0 to ±42949.672 (0.001mm resolution)

Decimal point programming negates the need for trailing zeros. However, if decimal points aren't used, the control assumes the decimal four digits or three digits from the right, according to the inch or metric format rules. Leading zeros are not required.

Definition

Linear axis commands move an axis a selected increment (G91) or to a selected coordinate (G90).

Description

Axis departure commands for the linear axes are written using the correct letter address for the linear axis followed by a number of the proper format to specify distance. The control assumes all departure commands as positive unless a negative sign (-) is programmed following the letter address and preceding the distance number. Maximum linear axis departure commands are limited, on a per block basis, to the machine tool geometries.

Recommended Application

Linear axes may be programmed individually or in multiple axis combinations to generate a vector path for three dimensional contouring or positioning.

Axis Departure (Rotary Axes)

Letter Address: A, B, C

Maximum Number of Digits: 8
Format: (5.3)
Range: 0.0 to ±99999.999° (0.001° resolution)

Leading zeros are not required. Decimal point programming negates the need for trailing zeros, however, if decimal points are not used, the control assumes the decimal three digits from the right.

Definition

Rotary axis commands rotate an axis a selected increment (G91) or to a selected coordinate (G90).

Description

Axis departure commands for rotary axes are written using the correct letter address followed by a number of the proper format to specify rotation. The control assumes all commands are positive unless a negative sign (-) is programmed. The programming of a negative sign rotates the rotary axis counterclockwise (G90 or G91 mode). Some machines which are equipped with an A axis operate in a sightly different manner because of a limited traversing range. In this case the machine operating manual should be consulted for the travel limits of the machine.

Recommended Application

Rotary axes may be programmed individually or in multiple axis combinations to generate helical cuts.

Programming Considerations

1. When programming commands combining linear and rotary axes, and the feedrate along the resultant cutter path is to be precise, the use of inverse time feedrate coding (G93) is required.

2. Combined linear and rotary axis departure commands may be programmed in G94 mode if the feedrate of the rotary axis is not critical. The programmed feedrate applies to the linear axis, while the rotary axis is assigned a feedrate to reach the end point in unison with the linear axes.

3. If combined linear and rotary axis commands are programmed at rapid traverse, the linear axes receive a feedrate of maximum IPM or MMPM; the rotary axis receives a feedrate of maximum DPM.

4. If two rotary axes are commanded in the same block, the programmed feedrate is applied to the B axis. The other axis is automatically assigned a feedrate to reach the end point concurrently.

5. Rotary axis departure commands are not allowed in a block of circular interpolation.

6. In the G90 mode, commands greater than or equal to B360 are 'factored out'; thus B360. becomes B0.(360-360); B445.=B85.(445-360); B765.= B45.(765-360-360), etc.

7. In the G91 mode, the table will be rotated the incremental value of the B word (i.e., B720. will cause the B axis to rotate 720 degrees from its present position.).

Axis Departure (Indexing B Axis)

Letter Address: B

Maximum Number of Digits: 8
Format: (5.3)
Range: 0 to ±99999.000 (1° resolution)

Decimal point programming negates the need for trailing zeros. However, if decimal points are not used, the control assumes the decimal three digits from the right. Leading zeros are not required.

Definition

Indexing B axis commands rotate the B axis a selected increment (G91) or to a selected coordinate (G90).

Description

Axis departure commands for the indexing B axis are written using the letter B followed by a number of the proper format to specify rotation. The control assumes all commands are positive (CW) unless a minus sign (-) is programmed. Programming a minus sign rotates the indexing axis counterclockwise (CCW).

Recommended Application

The indexing axis may be programmed individually or in combination with linear axis departure commands. However, rapid traverse is the only allowable feedrate for B axis rotation and any linear axis commands in the same block must be programmed at rapid traverse.

Programming Considerations

1. If B is programmed in a data block containing a Z axis command, the B axis completes motion prior to any Z axis motion

2. In G90 mode, commands greater than or equal to B360 are 'factored out'. Thus B360. becomes B0. (360-360); B445.=B85. (445-360); B765.= B45. (765-360-360), etc.

Arc Center

Letter Address: I, J, K

Maximum Number of Digits: inch; 8 metric; 8
Format: inch; (4.4); metric; (5.3)
Range: inch; 0 to ±1690.9320 (0.0001" resolution)
metric; 0 to ±42949.672 (0.001mm resolution)

Decimal point programming negates the need for trailing zeros. However, if decimal points are not used, the control assumes the decimal four digits or three digits from the right according to the inch or metric format rules. Leading zeros are not required.

Definition

The arc center commands define the centerpoint of the radius in circular interpolation machining. The I, J, and K entries specify the location of the center point of the arc parallel to the X axis, the Yaxis and the Z axis respectively. In G91 mode the incremental distance is referenced from the start point. When linear motion accompanies circular motion, the arc center for the plane of the linear motion represents the ratio of linear motion distance per arc length in radians (Rise per Radian). This ratio is always positive.

NOTE

The Rise per Radian command is no longer required for Helical Programming, but may be included if desired.

Description

The I,J, and K words are written using the correct letter address followed by a number of the proper format to specify the location of the arc center point. The control assumes arc center offset commands as positive unless a negative (-) sign is programmed following the letter address and preceding the number. The I, J, and K entries are not limited to the dimensions of the coordinate zone.

Fixed Cycle Clearance Plane (R-Word) (Modal)

Letter Address: R

Maximum Number of Digits: 8
Format: inch- (4.4); metric- (5.3)
Range: inch; 0 to ±1690.9320 (0.0001" resolution)
metric; 0 to ±42949.672 (0.001mm resolution)

Decimal point programming negates the need for trailing zeros. However, if decimal points are not used, the control assumes the decimal four digits or three digits from the right according to the inch or metric format rules. Leading zeros are not required.

Definition

For Fixed Cycles, the R-word establishes the Z axis clearance plane to which the tool initially positions at rapid traverse, and to which the tool retracts after satisfying the programmed Z depth during fixed cycle routines. The R-word must be larger than the Z depth.

For radius programming, the R-word establishes the radius of the desired arc in circular interpolation, in the G2 or G3 mode of operation.

R is also used to define the surface separation in the spindle probe applications using the G24 thru G27 commands. (For further information on the use of the R-word in spindle probe programming, see Chapter 7.)

Description

The fixed cycle clearance plane is programmed by writing the letter address R followed by a number of the proper format to specify the coordinate position of the clearance plane. The Z axis clearance plane thus established, is perpendicular to the Z axis.

For circular interpolation, the R-word is programmed by writing the letter address R followed by a number of the proper format to specify the distance between the start point of the arc and the control-calculated center point of the arc.

Fixed Cycle Incremental Feed Distance (E-Word) (Modal)

Letter Address: E

Maximum Number of Digits: 8
Format: inch- (4.4); metric- (5.3)
Range: inch; 0 to ±1690.9320 (0.0001" resolution)
metric; 0 to ±42949.672 (0.001mm resolution)

Decimal point programming negates the need for trailing zeros. However, if decimal points are not used, the control assumes the decimal four digits or three digits from the right according to the inch or metric format rules. Leading zeros are not required.

Definition

The E-word establishes the incremental feed distance during the G83 (chipbreaking) and G87 (deep hole drilling) cycles.

The E-word is used to define tolerance zones or bands for the G24 command in spindle probe programming.

Description

The incremental feed distance is programmed by writing the letter E followed by a number in the proper format to specify the incremental distance command. E-words are always expressed as positive incremental distances and not as coordinate position commands.

OFFSET RELATED COMMANDS

Fixture Offset Compensation

Letter Address: H

Maximum Number of Digits: 8
Format: 8.0
Range: 00000000 to 00000032

Offset Value Format: inch- (4.4); metric- (5.3)
Offset Value Range:
inch; 0 to ±1690.9320 (0.0001" resolution)
metric; 0 to ±42949.672 (0.001mm resolution)

Description

Fixture Offset Compensation is programmed in either Implicit or Explicit mode. Implicit mode (G49) applies compensation automatically, based on Index or Rotary table position; Explicit mode (G48) requires the programming of an H, followed by the number from the Offset Table which represents the desired offset values. Leading zeros are not required.

Function

Fixture Offset Compensation is used to compensate for deviations between the part program location and the actual piece part location as it is fixtured on the table.

PROGRAM ORGANIZATION COMMANDS

Preparatory Function

Letter Address: G

Maximum Number of Digits: 2
Format: 2.0
Range: 0 to 99

Description

Preparatory functions are programmed by writing the letter address word G followed by a one or two digit number. Leading zeros are not required.

Function

Preparatory functions are used to prepare and initiate specific machine functions and to establish desired operating modes.

Miscellaneous Function

Letter Address: M

Maximum Number of Digits: 2
Format: 2.0
Range: 0 to 99

Description

Miscellaneous Functions are programmed by writing the letter address word M followed by a one or two digit number. Leading zeros are not required.

Function

Miscellaneous function commands govern the ON/OFF status of specific machine functions.

Sequence Number (N-Word)

Letter Address: N

Maximum Number of Digits: 8* (see note below)
Format: 8.0
Range: 0 to 99999999

Description

A sequence number is the letter N followed by a number within the specified range. The sequence number is used to label and identify the data block in which it is programmed. Although sequence numbers are not needed for part processing, they are required for execution of the sequence number (block search) feature of the control, and for the block search function of the File Editor. Or the string search function of the control may be used on the Main Display screen. The string search function is only operational on the currently active part program or subroutine.

NOTE

The control accepts sequence numbers containing decimal points in a 5.4 format ranging from 0 to 99999.9999; however, decimal sequence numbers cannot be searched using Sequence Number Search procedure, and the FILE HANDLER Resequence command does not resequence any digits to the right of a decimal point.

NOTE

The NAM Type II Data Statement is not assigned a sequence number. (For further information see Chapter 4.)

Recommended Application

Sequence numbers should be used in every program that is to be used on an input device (tape reader, mag tape or disk storage). Data block entries generated while in the MDI mode need not contain N-words.

Programming Instructions

1. Sequence numbers must be the first entry in the block in which they appear, unless a block delete slash code is also programmed for that block. In this case the slash code (/) precedes the N number.

2. Sequence numbers must occur in ascending order for proper functioning of sequence number search functions. Sequence numbers which are not in numerical order have no adverse effect upon the part processing cycle itself.

3. It is good programming practice to increment sequence numbers in intervals of five or ten (e.g. N10,N20,N30 etc.). This practice may eliminate the need to resequence a part program should additional data blocks have to be added later.

Feedrate Number (F-Word) (Modal)

Letter Address: F

INCH	METRIC	ROTARY
5.1	6.0	3.0
1 - 99999.9	3 - 999999	1 - 999
0.1 Inch	1 mm	1 Degree

TABLE 1-A G94 MODE FORMAT / RANGE

	INCH	METRIC
FORMAT	3.1	5.1
RANGE	1 - 999.9	1 - 99999.9
RESOLUTION	0.1 Unit	0.1 Unit

TABLE 1-B G93 MODE FORMAT / RANGE

Description

The feedrate number specifies the speed at which an axis or axes is to move. The actual speed of an axis departure command depends upon the position of the [FEEDRATE OVERRIDE] switch on the front panel of the control and upon the preparatory function (G93 or G94) in effect. In all cases the programmed, or modified, feedrate is applied to the vector path formed by spindle movement toward the programmed coordinate (i.e. tool path).

Rapid traverse is invoked by programming F0 (zero). The maximum feedrate is applied to the axis with the longest distance to travel. Maximum rotation is applied to rotary axes.

The F-word is also used in conjunction with the G04 preparatory command to represent the length of a programmable dwell in seconds.

	DWELL
FORMAT	5.1
RANGE	.1 - 99999.9
RESOLUTION	0.1 Second

TABLE 1-C G04 MODE FORMAT/ RANGE

Associated Commands

1. G04 Dwell.
2. G93 Inverse Time Feedrate Coding.
3. G94 Direct Feedrate Coding.

The P0 Command

The P0 command may be used to bypass length compensation values in KT-APL software for the block in which it is programmed. The preferred method is to use the M25 command (See Chapter 3).

Spindle Programming (S-Word) (Modal)

Letter Address: S

Maximum Number of Digits: 5
Format: 5.0
Range: 00000 to 99999 (varies per machine)

Description

The S-word specifies the revolutions per minute at which the spindle is to rotate. The actual spindle speed is dependent upon the position of the [SPINDLE SPEED OVERRIDE] switch on the control front panel. The S-word is also used in conjunction with the M19 miscellaneous function to execute a programmable keylock orientation on certain machine tools.

Programming Instructions

1. Program the letter S followed by the desired RPM to specify the speed of spindle rotation.

2. Program the Miscellaneous function M03 or M04 in the data block requiring spindle rotation.

Associated Commands

M03 Spindle ON Clockwise.
M04 Spindle ON Counterclockwise.
M05 Spindle OFF.
M19 Keylock.

Tool Select (T-Word)

Letter Address: T

Format / Range

DECIMAL	OCTAL	BINARY
7.0	5.0	4.0
0000001 to 9999999	0001 to 77777	0101 to 3131

TABLE 1-D

The digits 8 and 9 cannot be used in an octal tool number.

Description

The T-word specifies the number of the tool to be selected from the tool storage magazine for utilization in the part processing cycle.

Programming Instructions

1. It is good programming practice to enter the T-word for the next tool in the first data block following the transfer of the previous tool.

2. In addition to programming T-words within the program, the part programmer is responsible for supplying the machine operator with the list of tools to be used in the part processing cycle.

3. To transfer the selected tool into the spindle, program M06.

4. On machines with no tool reader, the tool table must be loaded by tape or by the operator.

TYPE II DATA STATEMENTS

(For Further Information See Chapter 4)

The KT-GEMINI Controller permits the application of certain RS-447 Type II Data Statements. These statements supplement the Part Program Letter Address System, Preparatory Functions (G Codes) and Miscellaneous Functions (M Codes), which are collectively referred to as Type I Data Statements.

The Type II Data Statement consists of the following items in alphabetical order.

(ACP) - Specify and Modify Adaptive Control

(CER) - Continue Part Program Execution Upon Device Errors

(CLF) - Close File

(CLS) - Call Subroutine

(CPY) - Copy a File

(DAT) - Update Date and Time

(DFC) - Define a Cycle

(DFS) - Define a Subroutine

(DLF) - Delete a File, Cycle or Subroutine

(DLP) - Delete a Parameter

(END) - End a Statement

(EVT) - Record Part Program Event

(FWR) - Write to File

(FXC) - Load to Fixture Offset Table

(GTO) - Go to Label

(IFT) - Conditional "If True"

(LBL) - Label Definition

(MIR) - Mirror Image, Axis Inversion

(MSG) - Programmable Operator Message

(NAM) - Name a Part Program

(OIF) - Name a Spindle Probe Inspection File

(OPF) - Open a File

(PAR) - Parameter Assignment Statement

(PRT) - Print File

(RCP) - Copy a File to or from a Remote Station

(SER) - Stop a Part Program Execution Upon Device Errors

(SET) - Set an Output

(TLD) - Load a Tool Table with Tool Parameters

(U88) - Programmable Boring Orientation Cycle

(UAT) - Select Alternate Tool Data

(UCY) - Maximum Part Program Execution Time-FMS

(UFX) - Calculate and Load Fixture Offset to the Fixture Offset Table

(ULF) - Loads Saved Fixture Offset Table from Disk Storage

(USF) - Fixture Offset Table to Be Saved on Disk Storage

(UTL) - Identifies All Tools Used in a Part Program-FMS

(WAI) - Wait on Input

(ZTB) - Zero Either Tool Table or Fixture Offset Table or Both

CHAPTER 2

PREPARATORY FUNCTIONS

GENERAL INFORMATION

The preparatory functions, or G codes, are used by the control's Executive System to prepare and initiate machine functions. The G codes are programmed by writing the letter G followed by a 1 or 2 digit number. There are no restrictions limiting the number of G codes that can be programmed within a single data block, with the exception of probe cycles, as long as these codes are mutually non-conflicting.

(User defined cycles are explained in Chapter 8 and utilize G801 thru G899).

Care should be taken to avoid programming conflicting G codes in a data block. If conflicting G codes are present in a block, the control system recognizes and executes only the last entered conflicting command(s) in the block.

A G code is designated as being either MODAL or NON-MODAL. Modal (M) preparatory functions are those which, once programmed, remain in effect through the succeeding data blocks until changed by the programming of another command from its group (a conflicting code) or canceled by some control function (e.g. [CLEAR ALL LOGIC]).

Non-Modal (NM) preparatory functions are those that function only in the data block which contains them.

Certain Modal preparatory commands are in effect immediately upon Machine Startup (MS). They remain in effect throughout the part program unless specifically replaced by a conflicting command.

This chapter defines the G codes and documents their functions and recommended use.

G00 POSITIONING MODE (MS)

Functional Description:

Establishes positioning mode for point to point motions, for operations involving the use of plunging type tools, or may be used for milling operations where a dwell mark is okay. When in G0 mode, the control

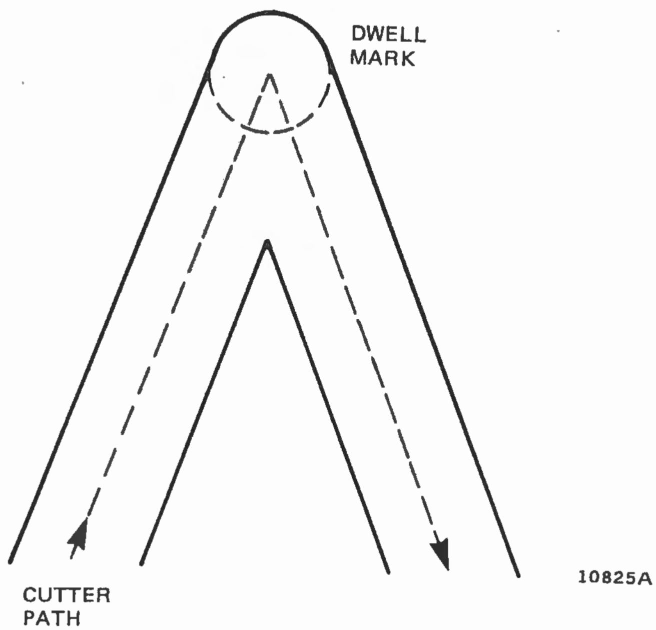


Fig. 2.1 DWELL MARK IN G00 MODE

forces deceleration of machine axes at the end of the currently active block. The ready block does not become active until the axes have positioned to within 0.001" (0.025mm) of the end point. Axes deceleration permits any accumulated servo following error to be closed out prior to block transfer, and may create a dwell mark in the part surface. Axis slides not addressed do not move. Multiaxis moves generate a vector path.

Programming Application:

1. Program G00 for rapid positioning between coordinates, for operations involving plunging type tools (drilling, boring, reaming, tapping) or for milling a non-tangent cutter path.
2. G00 may be programmed at any applicable feedrate (G00 mode does not force rapid traverse).
3. Although not recommended for contouring, the axis motion in a G00 block is identical to that of a G01 block, with the exception of the axes deceleration. G00 may be used, for example, in a milling operation where the cutter is not in

contact with the part surface when the end point is reached. When programming a non-tangent cutter path, G00 should be used where a slightly rounded corner cannot be tolerated.

Cancellation:

G00 cancels G01.
G00 is canceled by G01.

G01 CONTINUOUS MODE (M)

Functional Description:

Establishes continuous mode for linear or circular contouring operations. There is no deceleration of machine axes at the end of the currently active block as in the G00 mode; therefore, no cutter dwell mark is generated on the part surface during block transfer from ready to active. Block transfer occurs when the control has counted down the command.

Programming Application:

1. Program G01 for continuous linear or circular contouring operations with side cutting tools, such as end mills, when no dwell mark must appear on the part surface.
2. G01 is a modal function and need not be reprogrammed unless it has been canceled.

Cancellation:

G01 cancels G00.
G01 is canceled by:
  G00 - Positioning Mode
  M02 - End of Program
  M30 - End of Program - Tape Rewind - Shuttle
  M60 - Pallet Shuttle Within Part Program
  Manual operation of [CLEAR ALL LOGIC]
  Manual operation of Sequence Number Search

G02 CIRCULAR INTERPOLATION CLOCKWISE (NM)

(Programming Instructions for Circular Interpolation are in Chapter 5)

Functional Description:

Indicates that axes motions will generate a clockwise arc when viewing the plane of motion from the negative direction of the perpendicular axis. The plane of motion is established by programming G17 (XY plane), G18 (XZ plane) or G19 (YZ plane). Each data block containing a G02 command may define an arc up to 360 degrees. It is also possible to program a linear axis move in a data block containing a G02 command to perform helical interpolation.

Programming Application:

1. Program G02 in a data block requiring the contouring of a circle or an arc of a circle.
2. G02 may be utilized for circular interpolation in any of three planes defined by the linear axes.
3. G02 may be combined with a linear axis move for helical interpolation.

Cancellation:

G02 is a non-modal function, so is applicable only in the one data block in which it is programmed.

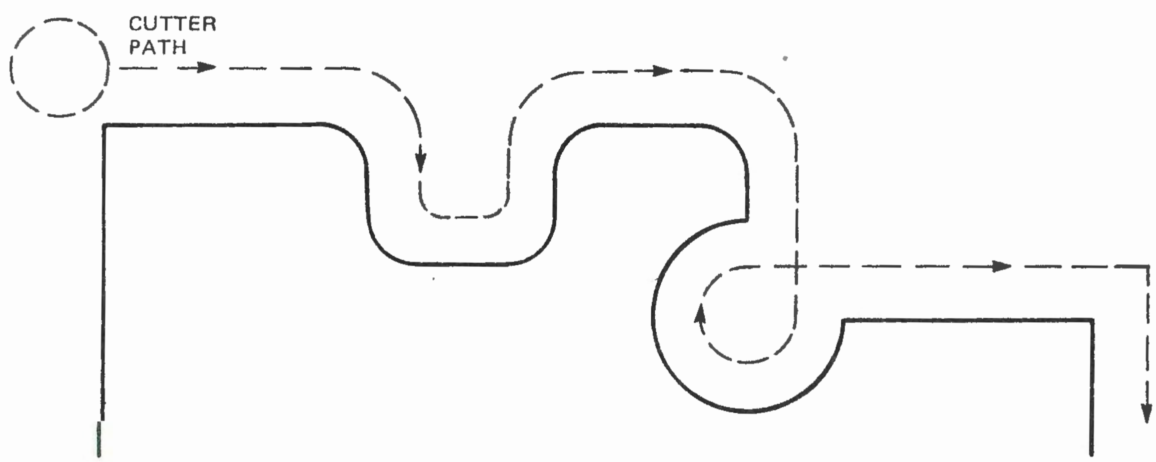


Fig. 2.2 G01 CONTOURING

G03 CIRCULAR INTERPOLATION COUNTERCLOCKWISE (NM)

(Programming Instructions for Circular Interpolation are in Chapter 5)

Functional Description:

Indicates that axes motions are to generate an arc in the counterclockwise direction viewed from the plane of motion, in the negative direction of the perpendicular axis. The plane of motion is established by programming G17 (XY plane), G18 (XZ plane) or G19 (YZ plane). Each block containing a G03 command may define an arc up to 360°. It is also possible to program a linear axis move in a data block containing a G03 command to perform helical interpolation.

Programming Application:

1. Program G03 in a data block requiring the contouring of a circle or an arc of a circle.
2. G03 may be utilized for circular interpolation in any of three planes defined by the linear axes.
3. G03 may be combined with a linear axis move for helical interpolation.

Cancellation:

G03 is a non-modal function, so is applicable only in the one data block in which it is programmed.

G04 PROGRAMMABLE DWELL (NM)

Functional Description:

Initiates a pause in the part processing cycle the duration of which is established by the programming of a feedrate number in the data block containing the G04 command.

Programming Application:

Ensure smooth finish at final depth of plunging type cut.

Programming Instruction:

1. Program G04 and dwell-time-related feedrate number (F-word) in a single data block containing no axis departure commands.
2. The F-word represents the dwell time in tenths of a second permitting dwell times ranging from 0.1 second (F.1) to 99999.9 seconds (F99999.9). This is equivalent to 27.7 hours.
3. G04 is programmable in either G93 or G94 mode and in either G70 or G71 mode.

Programming Conditions:

1. G04 is non-modal and cancels the previous feedrate for that data block only. Previous feedrate is reinstated during execution of the next data block.
2. No axis departure commands can be programmed in a data block containing a G04 command.

Example:

N100 G0 X10.Y10.Z8.1 F0
(position axes to clearance at hole location)

N110 Z7.2 F4.
(plunge to Z depth at feedrate of 4 IPM)

N120 G04 F2.5
(dwell for 2-1/2 seconds)

N130 Z6.8
(plunge further in Z axis, feedrate of 4. reinstated)

G17 XY PLANE SELECTION FOR CIRCULAR INTERPOLATION (MS)

Functional Description:

Establishes the XY plane for circular interpolation. In the G17 mode, the end point of the arc is defined by the X word and Y word. The center point of the arc parallel to the X axis is defined by the I word and the center point of the arc parallel to the Y axis is defined by the J word.

Helical Interpolation in the G17 mode involves an arc in the XY plane with a simultaneous linear motion in the Z axis. A Rise per Radian command, represented by the K word, must be calculated and programmed for each block of helical interpolation in the G17 plane.

Programming Application:

Program G17 along with the necessary associated commands for contouring a circle, or an arc of a circle, in the XY plane.

Cancellation:

G17 is in effect upon machine startup.
G17 cancels G18 or G19.
G17 is canceled by G18 or G19.

G18 XZ PLANE SELECTION FOR CIRCULAR INTERPOLATION (M)

Functional Description:

Establishes the XZ plane for circular interpolation. In the G18 mode, the end point of the arc is defined by the X word and Z word. The center point of the arc parallel to the X axis is defined by the I word and the center point of the arc parallel to the Z axis is defined by the K word. For more details, see Chapter 5.

Helical interpolation in the G18 mode involves an arc in the XZ plane with simultaneous motion in Y axis.

Programming Application:

Program G18 along with the axis commands for contouring a circle, or a circular arc , in the XZ plane.

Cancellation:

G18 cancels G17 or G19.
G18 is canceled by G17 or G19.
G18 is canceled by either end of programs M02 or M30.
G18 is canceled by Manual Operations, Sequence Search or [CLEAR ALL LOGIC].

G19 YZ PLANE SELECTION FOR CIRCULAR INTERPOLATION (M)

Functional Description:

Establishes the YZ plane for circular interpolation. In the G19 mode, the end point of the arc is defined by the Y word and Z word. The center point of the arc parallel to the Y axis is defined by the J word and the center point of the arc parallel to the Z axis is defined by the K word. For more details, see Chapter 5.

Helical interpolation in the G19 mode involves an arc in the YZ plane with simultaneous motion in X axis.

Programming Application:

Program G19 along with the axis commands for contouring a circle, or a circular arc , in the YZ plane.

Cancellation:

G19 cancels G17 or G18.
G18 is canceled by G17 or G18.
G18 is canceled by either end of programs M02 or M30.
G18 is canceled by Manual Operations, Sequence Search or [CLEAR ALL LOGIC].

G20 and G21 AXIS SUBSTITUTION

These commands are used in the description of machine movements for machines equipped to operate in up to six axes. (More information concerning these commands is contained in Chapter 7, in the discussion of the U-Head attachment, and Appendix H)

Plane definition is as follows: the U axis is parallel to the X axis, V is parallel to Y axis, and W is parallel to the Z axis. These commands are in conjunction with one of the circular planes G17, G18 and G19. The programmer may wish to construct a circle in one of the optional axis (U,V,W). G20 and G21 are used to indicate which axes have been replaced by their corresponding parallel axes. G20 implies replacement of the first axis, G21 replacement of the second, G20 and G21 implies that both axes are replaced.

For example, a G02 circle in the ZU plane is described as follows: the second axis (X) has been replaced by it corresponding parallel coaxis (U), the modal state will be G21 as well as G18. See the figure below:

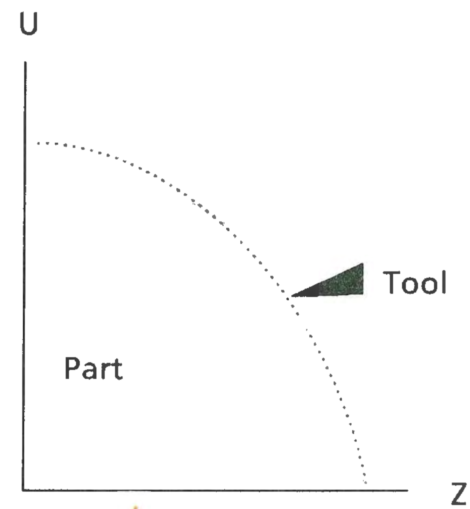


Fig. 2.3 CUTTER PATH GENERATED BY U-HEAD

The G20 / G21 commands do not define cutting planes. They only selectively replace the circular axes with their parallel coaxes and must be used in combination with one of the circular planes G17, G18, or G19.

G22-G37, G60-G62, G92 SPINDLE PROBE CYCLES

(Program, Operating and Maintenance Instructions for the spindle probe are located in Pubs 778, 878, 968 and also Chapter 7 of this publication.)

Functional Description:

The Spindle Probe is a tool containing a sensitive, three axis, contact sensing stylus. Various spindle probe cycles, initiated by the programming of specific G codes, govern the type of operation to be performed by the probe tool. A listing of the probe cycles, and a brief summary of their function, follows.

G22 Record Axis Position "Contact" (NM)

Records all axis positions upon probe surface "contact", halts axis motion and initiates processing of the next part program block. No axis motion occurs if the probe is in a "contact" state when G22 is processed.

G23 Record Axis Position "Non-Contacts" (NM)

Records all axis positions upon probe surface "non-contacts", halts axis motion and initiates processing of the next part program block. No axis motion occurs if probe is in "non-contact" state when G23 is processed.

G24 Tolerance Verification (NM)

Used to detect deviation between the programmed position and the absolute position within a specified tolerance band. Only the axis (X,Y,Z,A,B,C)to be checked need to be programmed. The E-word is used to set the tolerance zone. The R-word is used to define the distance between two surfaces that must be checked.

G25 Automatic Regrid (NM)

Provides automatic resynchronization of linear axes reference (grid) positions. Machine tool axis slides position to a target bushing located on the grid post or to a fixture mounted target bushing. Periodic regridding of axis slides can help ensure continued high accuracy during part processing cycles.

G25R Ball Radius and Grid Preset Calculation (NM)

Establishes and records bushing center (X-Y) and bushing face (Z). Calculates stylus ball radius and probe runout.

G26 Surface Detection (NM)

Used to detect any surface with respect to the X, Y, Z, A, B and C axes. This cycle is particularly useful for inspecting machining results and for determining offsets (fixture or workpiece). It may be used to detect an angular surface with the use of the RA format in which R is the distance from a start point and A is the angular displacement from the 3 o'clock position as viewed from the spindle looking outward toward the part surface. Or the probe may be commanded to detect an angular surface with the use of the XY format in which it is commanded from a start point along some vectored path to an end point in X and Y. Upon part surface contact the control notes the absolute coordinates of the contacted surface. When the G26 cycle is completed the centerline of the spindle will be aligned with the edge of the part surface which has been detected.

G26TZ Calculate Effective Probe Length For Secondary Probe

The G26TZ is used to determine the effective probe length for a second probe that is being utilized in the tool magazine or if the probe stylus has been changed from the original which was used in the spindle probe.

G27 Automatic Position Between Surfaces (NM)

Used to determine the center point between two surfaces in the B, C, X or Y axis. This cycle is particularly useful for obtaining absolute X and Y axis relationships between a previously machined bore and all subsequent machining operations, and may also be used to indicate the center point of a bore just machined. It may be used to detect an angular surface thru the use of the RA format in which R is the distance from a start point and A is the angular displacement from the 3 o'clock position as viewed from the spindle looking outward toward the part surface. Or the probe may be commanded to detect an angular surface thru the use of the XY format in which it is commanded from a start point along some vectored path to an end point in X and Y. When the cycle has been completed the probe is at the centerpoint of the probed part surfaces in X and Y.

G27R Ball Radius Calculation (NM)

The G27 cycle when programmed with an R-word is used to calculate the stylus ball radius. The R-word represents the distance between the two surfaces to be detected, and must be known accurately prior to commanding the G27R routine. G27R is used in lieu of the G25 command if a ball radius calculation is desired without an automatic regrid, or if the machine lacks a grid post or fixture mounted target bushing.

G27E Hub / Obstructed Bore Centerpoint Detection Cycle (NM)

The function of the G27E Cycle is to determine the centerpoint of an obstructed outside diameter such as a hub or boss, or alternately to determine the centerpoint of an obstructed inside diameter such as a circular milled groove. It may be used to detect an angular surface with the use of the RA format in which R is the distance from a start point and A is the angular displacement from the 3 o'clock position as viewed from the spindle looking outward toward the part surface. Or the probe may be commanded to detect an angular surface with the use of the XY format in which it is commanded from a start point along some vectored path to an end point in X and Y. When the cycle has been completed the probe is at the centerpoint of the probed part surfaces in X and Y.

G28 Obstruction Detection (M)

Detects collisions due to obstructions in the path of the stylus, or if broken taps or drills are present in a hole.

G29 Cancel Obstruction Detection (M)

Cancels G28 Obstruction Detection cycle.

G36 Store In-Process Inspection File (M)

Used in conjunction with the G26 and G27 probe cycles to do an in-process inspection of a partially or completely machined workpiece which is still fixtured on the pallet, and to record the results of the inspection in a file. The information recorded is stored on the hard disk unit in the ASCII portion of the Master File Directory (MFD;SYS) and is named using a Type II data statement which is incorporated into the G36 programming format. The G36 command opens a file under the programmed file name and data is recorded upon the first occurrence of a G26 or G27 command.

The inspection data file records the following:

Control Serial Number
Pallet Code Number/Part Number
Number of Part Cycles Completed
Time Clock (YR;MO;DAY) (HR;MIN;SEC)
Machine Coordinate Data resulting from each probe cycle commanded.
Also includes sequence number (N number) in which G26 or G27 was commanded.
Type II Data Statements are recorded and transferred to the inspection file.

G37 End In-Process Inspection File (M)

Terminates the G36 inspection data command and closes the inspection file. When closed, no further data is recorded. An inspection file must be closed before it can be displayed or copied using File Handler commands. The G37 command is programmed in a block by itself. In KT-APL, the preferred method is to use the CLF command to close an inspection file, since this command does not require the reintroduction of the G36 command to open an inspection file.

G60 Cancel G61 or G62

The G60 command cancels either a G61 or G62 command. It is used in a user-defined cycle in conjunction with the G61 or G62 commands.

G61 Record Probe "Contacts" (M)

The G61 is used in user-defined probe cycles to record the number of probe contacts. The total number of probe contacts is recorded in parameter L1022. This count will be reset to 0 at the beginning of a block containing a G61 command. If the probe is in the contact state when the G61 is processed that contact will not be added to the number of contacts that have been recorded in parameter L1022.

G62 Record Probe "Non-Contacts" (M)

The G62 is used in user-defined probe cycles to record the number of probe non-contacts. The total number of probe non-contacts is recorded in parameter L1022. This count will be reset to 0 at the beginning of a block containing a G62 command. If the probe is in the non-contact state when the G62 is processed, that non-contact will not be added to the number of non-contacts that have been recorded in parameter L1022.

G92 Axis Regrid to Commanded Position (NM)

The G92 command is a non-modal G code that is used to regrid the specified axes to the commanded position. For example, G92X11.Y12.Z13. will cause the absolute positions of X, Y and Z to be loaded with 11, 12 and 13, respectively.

G38 BROKEN TOOL DETECTION (NM)

Functional Description:

Used by the Control, in conjunction with a "broken tool detection" checking station which is located in the grid post, to check the setting distance or cutter diameter of the tool in the spindle or to detect a broken tooth on a single point or multiflute cutter. The checking station consists of a post containing two proximity switches. The proximity switches are precision electronic devices that can sense, without physical contact, the presence of metallic objects. If a tool length and/or diameter compensation has been assigned to the tool which is to be checked, these offsets must be in effect during the G38 procedure.

Programming Application: (Chapter 7)

G38 may be programmed during the part processing cycle to detect the presence of a broken tool. If a broken tool is detected, the machine tool is placed in a [CYCLE STOP] condition and the message "TOOL BROKEN" appears on the CRT [MESSAGE] display. If an alternate tool has been loaded into the magazine and tool table, the alternate tool is automatically selected and the broken tool is automatically set to a status of broken in the tool table.

G39 SPECIFIC FIXTURE OFFSETS (M)

Specific fixture offsets are the offset values that programmers and operators have worked with in the past. Fixture offsets are established for each rotary or index axis position. This done using any of the following methods; the operator entering the fixture offset values thru the MDI keys into the Fixture Offset [DISPLAY LIST] [U], or thru Type II data statements, either FXC or UFX. The G39 is modal upon control startup and whenever G47 is not present.

G40 CUTTER DIAMETER COMPENSATION CANCEL (M)

Functional Description:

Used in canceling the cutter diameter compensation feature. If programmed in a block containing no axis motion, the compensation is removed with no resulting slide motion. If programmed in a block which contains axis motion, then compensation is removed gradually in a ramping motion until at the programmed end point compensation is totally removed.

Programming Application: (Chapter 6)

Terminates the offset created by the programming of the cutter diameter compensation feature.

NOTE

The programmer should be aware that cutter diameter compensation may be used in the G17 XY plane, G18 XZ plane and G19 YZ plane.

G41 CUTTER DIAMETER COMPENSATION LEFT (M)

Functional Description:

Activates look ahead cutter diameter compensation left of part surface when programmed in a data block that contains X and / or Y motion. Cutter diameter compensation left is the condition whereby the cutter centerline is left of the part surface when viewed from the part surface in the direction of the cut (see Figure 2.4). Cutter diameter compensation left is enabled by programming G41 in a data block containing an X and/or Y axis departure command of at least 0.0001" (0.001mm) in distance.

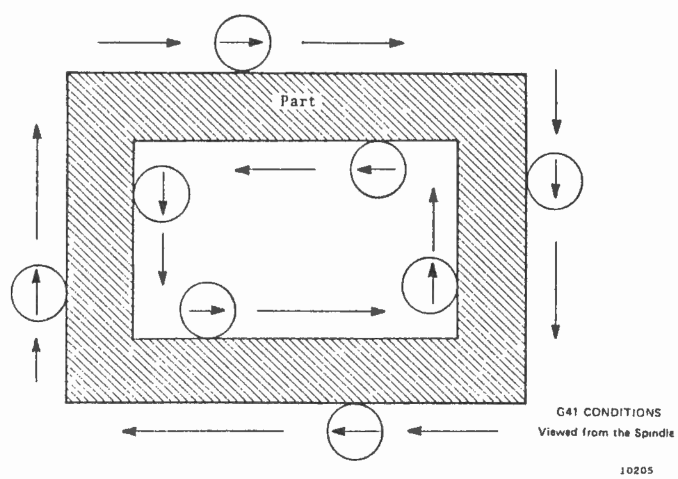


Fig 2.4 G41 PERSPECTIVE

Programming Application: (Chapter 6)

1. For milling operations involving linear or circular interpolation with cutter centerline left of part surface, G41 can compensate for deviations between the "programmed" diameter of the milling cutter and the actual diameter of the milling cutter used in the part processing cycle, or if programming spindle centerline to part contour, can compensate for the entire radius of the tool. The part surface in the direction of the cut block activates the compensation mode by means of a

ramping effect and can be referred to as a "ramp-on" block. The ramp-on block is generated as an extension of the programmed cutter path (linear) or tangent to the cutter path of a programmed arc.

2. For milling operations involving linear or circular interpolation with cutter centerline right of part surface, G41 can compensate for deviations between the "programmed" diameter of the milling cutter and the actual diameter of the milling cutter used in the part processing cycle, or if programming spindle centerline to part contour, can compensate for the entire radius of the tool.

G42 CUTTER DIAMETER COMPENSATION RIGHT (M)

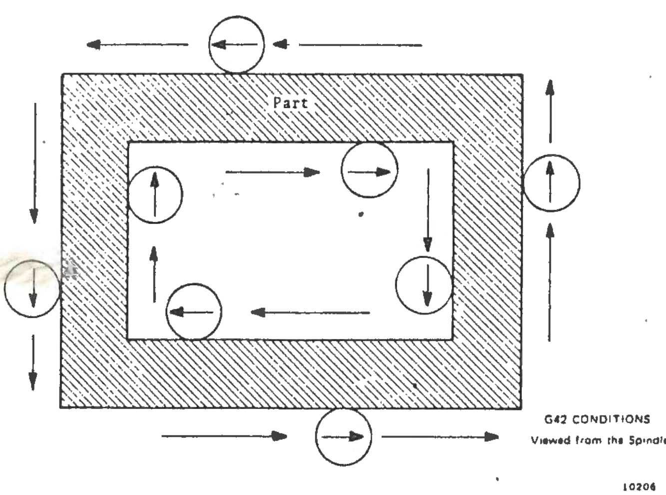


Fig 2.5 G42 PERSPECTIVE

Functional Description:

Activates look ahead cutter diameter compensation to right of part surface, when programmed in a block containing X and/or Y motion. Cutter diameter comp right is the condition whereby the cutter centerline is right of the part surface when viewed from the part facing in cutter travel direction (see Figure 2.5). Cutter diameter comp right is enabled by programming G42 in a block containing an X and/or Y axis move of at least 0.0001" (0.001mm) in distance. This block activates the compensation mode by means of a ramping effect and can be referred to as a "ramp-on" block. The ramp-on block is generated as an extension of the programmed cutter path (linear) or tangent to the cutter path of the programmed arc (circular).

Programming Application: (Chapter 6)

In milling operations involving linear or circular interpolation with the cutter centerline right of part surface, G42 can compensate for deviations between the "programmed" cutter diameter and the actual diameter of the milling cutter used in the processing cycle; or, if programming spindle centerline to part contour, can compensate for the entire tool radius.

G43 MOTION PERPENDICULAR TO THE CUTTER PATH (NM)

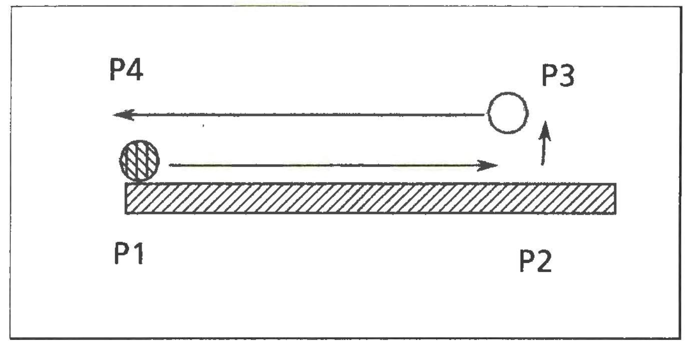


Fig.2.6 G43 MOTION

Functional Description:

This command allows the programmer to position off the part surface a slight distance and return parallel to the original cutter path without generating a cutter compensation error message. Example: Fig. 2.6

```
N120X0Y5.000F50.
N130X2.000G41    !Point 1 cutter comp is in effect
N140X5.000       !Move to Point 2
N150Y5.100G43    !Move to Point 3
N160X2.000G42    !Move to Point 4
```

G44 CHANNEL CUTTING PROGRAMMED FEEDRATE at TOOL CENTERLINE (NM) (OPTION)

This command allows the programmer to assure that a constant velocity will be applied to a compensated cutter path. The control will move the tool at the commanded feedrate. For example the situation may arise in circular interpolation in which the tool is buried in the work. In this case the velocity of either the left or the right side of the tool may exceed the desired feedrate if the G44 command is not used. The G44 command should be used to insure that the desired feedrate is applied to the part surface. The control will generate the proper feedrate dependent upon the cutter size.

G46 BYPASS FILL-IN BLOCKS (M)

Functional Description:

In certain applications it is desirable to bypass fill-in blocks generated by the Look-Ahead Cutter Diameter Compensation Feature.

This is done by programming a G46 Bypass Fill-In Blocks command in the same block with the G41 or G42 used to invoke cutter diameter compensation. The G46 will remain modal until the next executed G41 or G42 command, at which time it will be canceled unless reprogrammed in the same block.

The programmed cutter comp will then be executed ignoring internally generated fill-in blocks.

Programming Application: (Chapter 6)

1. G46 is to be programmed when it is desired to Bypass G41 or G42 generated fill-in blocks.
2. The G46 command MUST be entered in the same block as the G41 or G42 command.

Cancellation:

1. G46 is canceled upon execution of the next commanded G41 or G42.
2. G46 is canceled by M02, M30 or the manual operation [CLEAR ALL LOGIC].

G47 CALCULATED FIXTURE OFFSETS (M)(Option)

G47 puts the control in a mode that will calculate the fixture offset for any index or rotary axis position. To use this feature the programmer or the operator must first establish a fixture offset within the fixture offset table, along with initial rotary axis positions. This may be done using the following methods shown . The control will recognize and appropriately adjust the fixture offsets as the rotary axis is repositioned, as long as G47 remains in effect.

G48 FIXTURE OFFSET EXPLICIT MODE (M)

Functional Description:

Enables the programming of an H-word which establishes a set of offsets located in the Fixture Offset table. Fixture offsets provide a means to compensate for fixturing inaccuracies by automatically applying a position offset equal to the difference between a programmed coordinate and the actual coordinate, as measured on the fixture. Offset values are normally entered into the Fixture Offset table by the machine operator, but may be program entered thru FXC or UFX, Type II data statements.

Programming Application: (Chapter 6)

In the G48 mode, the part programmer enters an H-word for each position of the index (or rotary) table on which a machining operation is to take place. If offset values have been entered into the table adjacent to the H number programmed, those values are applied to the program. If multiple parts are fixtured on a single pallet position the programmer can assign a separate H number for each piece part and call for that H number at the appropriate place in the part program.

Cancellation:

1. G48 cancels G49.
2. G48 is canceled by G49 and upon [CONTROL ON].
3. The offset values currently in effect are canceled by programming H0 or replaced by programming a different H number.

G49 FIXTURE OFFSET IMPLICIT MODE (MS)

Functional Description:

Enables the application of a set of offset values, located in the Fixture Offset table, by the programming of the index (or rotary) table to a specific location. Fixture offsets provide a means of compensating for fixturing inaccuracies by automatically applying a position offset equal to the incremental distance between a programmed coordinate and the actual coordinate as measured on the fixture. The offset values are normally entered into the Fixture Offset table by the machine operator, but may be program entered thru FXC or UFX, Type II data statements.

Programming Application: (Chapter 6)

In G49 mode, the programmer calls for a specific position of the index (or rotary) table. This command MUST be executed with G00 (positioning mode) in effect. If offset values have been entered into the table adjacent to the position specified by the programmed command, those values are applied to the program.

Cancellation:

G49 cancels G48.
49 is canceled by G48.

The offset values currently in effect are canceled by programming H0, or replaced by programming a different index (or rotary) command which establishes a new set of offset values. G49 is modal upon startup.

G50-G53 ADAPTIVE CONTROL COMMANDS

(For additional information, see the Adaptive Control Manual, Pub 850A.)

G50 Adaptive Control OFF (M)

Functional Description:

Cancels the G51 (Adaptive On) mode of operation.

Programming Application:

Program at the end of an Adaptive Control cut to cancel the Adaptive Control feature.

G51 Adaptive Control ON (M)

Functional Description:

During milling operations, Adaptive Control can be programmed to maintain a target horsepower by modifying feedrates and spindle speeds. The Adaptive Control display contains thirteen individually programmable parameters.

Target horsepower and parameter modification (if needed) are programmed with a Type II data statement.

Programming Application:

Adaptive Control is intended for use in milling operations. When properly programmed, Adaptive Control can maximize metal removal rates, maximize cutter life, and maintain a consistent surface finish. In addition, it can reduce cycle time, protect the machine and tool from damage due to heavy cuts, and detect a dull tool by measuring the amount of cutting torque. When the control senses that the tool is dull, it exchanges it for an alternate tool in the magazine, if such a tool has been loaded.

G52 Spindle HP Calibration (NM)

G52 command automatically measures, at four spindle speeds per gear range, the wattage consumed between the spindle motor shaft and the tool with no load on the tool. The wattage is then converted into HP loss (tare) and is measured, recorded and displayed by the spindle calibration feature [DISPLAY LIST] [W].

Programming Application:

The G52 command updates the averaged spindle load, in units of 0.01, on the spindle calibration display. The spindle calibration display is divided into two sections: UPDATED data and CONFIGURED data.

The Updated data section is the G52 calibrated spindle load values. This data is keyboard clearable.

The Configured data is the factory generated spindle load values (contained on the configuration tape). This data is not keyboard clearable. The Updated data value overrides the Configured data values.

If no spindle calibration values are displayed, computations are based on zero (0) HP tare.

Programming Instructions:

1. The auto cycle consists of four calibration speeds per gear range. The cycle will automatically calibrate the speeds for the programmed gear range. Once the active gear range is calibrated the date is entered into the UPDATED table of the Spindle Calibration Display.
2. An M3 or an M4 is required before on in the same block as a G52 command. The G52 should be in a block by itself.
3. Perform a "system save" to store spindle calibration values on the disk. This is required if the machine does not have CONFIGURED values in the Spindle Calibration Display and/or the spindle was removed or repaired.
4. When a M4 tool (spindle counterclockwise) is used, recalibration for that gear range is recommended.

G53 Enable Adaptive Control without Airgap Mode (Option)

Functional Description:

This command cancels G50 and G51. When it is modal the normal Adaptive Control feature is disabled. The control will not recognize the air gap mode of operation, although the air gap parameter will be displayed on [DISPLAY LIST] [W]. The control will command the cutter to operate at the programmed feedrate and ignore the air gap parameter.

Programming Application:

The Adaptive Control feature is primarily used for rough milling to maintain a more uniform chip load.

Programming Instructions:

1. Program the preparatory command G53 to activate the Adaptive Control feature without the air gap mode of operation.
2. The ACP Type II data statement should be in the block preceding the G53 command. All Adaptive Control programmable parameters are entered via the Type II data statement.

G70 INCH INPUT (M)

Functional Description:

Establishes the inch mode of operation. All linear axis departure commands are interpreted in units of inch and all applicable CRT displays reflect inch dimensions. In addition, all offset values are automatically converted to units of inch. Metric programs are not converted to inch unit programs.

Programming Application:

Utilized when part print is dimensioned in inch units.

Programming Instructions:

1. Inch input is programmed in the start-up block of a part program to establish the inch mode of operation. G70 may also be restated in succeeding restart blocks if desired.
2. Rotary axis and indexing axis departure commands are expressed in degrees per minute (DPM) and are independent of G70 mode.

NOTE

Correct placement of the G70 command in a data block containing parameter commands is critical to the proper operation of the block. The G70 must appear either in a block by itself before a block with parameter commands or must be at the beginning of the block.

Example:
N200G70
: : :
N235(PAR,L1500=15.,L1501=16.23)
N240G71XL1500YL1501

Programming Conditions:

1. Inch input commands and metric input commands cannot be executed within the same data block of information.
2. Program G70 in a data block by itself if changing dimensional input from metric to inch within a part program.

Cancellation:

G70 cancels and is canceled by G71

G71 METRIC INPUT (M)

Functional Description:

Establishes the metric mode of operation. All linear axis departure commands are interpreted in units of millimeters and all applicable CRT displays reflect metric dimensions. In addition all offset values are automatically converted to units of millimeters. Inch unit programs are not converted into metric programs.

Programming Application:

Used when part print is dimensioned in metric units.

Programming Instructions:

Metric input is programmed in the start-up block of a part program to establish the metric mode of operation. G71 may also be restated in succeeding restart blocks if desired. Rotary axis and indexing axis departure commands are expressed in DPM and are independent of G71 mode.

Programming Conditions:

1. Metric input commands and inch input commands cannot be executed within the same data block of information.

2. Program G71 in a data block by itself if changing dimensional input from inch to metric within a part program.

NOTE

Correct placement of the G71 command in a data block containing parameter commands is critical to the proper operation of the block. The G71 must appear either in a block by itself preceding a block with parameter commands or must be at the beginning of the block.

Example:
N200G71
: : :
N235(PAR,L1500=15.,L1501=16.23)
N240G70XL1500YL1501

Cancellation:

G71 cancels and is canceled by G70.

G77 CYCLE INHIBIT (NM)

(For more information on this command see Chapter 8)

G77 is a non-modal command that prevents the execution of any defined cycle in blocks that contain a G77 command.

Programming Conditions:

- Program a G77 in a block to prevent any defined cycle in that block from being executed by the control.

G78 IGNORE SUBSTITUTIONS (NM)

A block containing a G78 command causes the control to ignore all undefined parameters that are within that block. This code is mainly used for subroutines and fixed cycles that need to execute conditionally "passed in" information without having to test to see if the parameter was actually passed in.

An example would be for a fixed cycle to move the other commanded axes after retracting the Z axis. No tests would have to be made for axes for A, B, C, X or Y if the motion block contained a G78 as follows:

N40G78XL824YL825ZL826F20.

(Motion would occur only if L824, L825 and L826 were defined in the parameter table.)

NOTE

Correct placement of the G78 command in a data block containing parameter commands is critical to the proper operation of the block. The G78 must appear at the beginning of the block.

G79 WAIT FOR PRECOMPUTE OF MAXIMUM NUMBER OF READY BLOCKS (NM)

Functional Description:

To allow for improved performance during high speed contouring a non-modal G79 code will inhibit any further ready blocks to be transferred to the active until the control has precomputed ahead as far as possible. Further information concerning the G79 command is contained in Chapter 8, Parameter and Subroutine Programming.

Programming Instructions:

G79 should only be used if needed to allow the control to precompute additional blocks of information. The first occurrence of G79 causes the control to devote additional processing capacity which remains in effect until a M02, M30 or a [CLEAR ALL LOGIC] is performed.

G80-G89 FIXED CYCLE COMMANDS

(Programming Instructions for Fixed Cycles are in Chapter 5)

The fixed cycles are preprogrammed routines designed to simplify the programming of repetitive plunging type operations and to reduce program length. Fixed cycle functions are discussed individually below.

G80 Fixed Cycle Cancel (MS)

Functional Description:

Used to cancel any fixed cycle routine.

Programming Application:

Terminate fixed cycle routines.

Programming Instruction:

1. G80 is programmed to cancel the operation of any fixed cycle routine.
2. G80 should be programmed in a block by itself.
3. Fixed cycles may also be canceled by the programming of a different fixed cycle command.

NOTE

In KT-APL the programmer has the capability of modifying the supplied cycles to create user-defined cycles for specific applications.

G81 Drilling Cycle (M)

Functional Description:

When programmed with X, Y, Z, R and F-word information recognizes a clearance plane and provides for feed in-rapid out sequence.

Programming Application:

Suitable for drilling a series of holes having the same diameter.

Cancellation:

1. G81 is canceled by G80 in a block by itself.
2. Another fixed cycle cancels G81.

G81 EXAMPLE:

N123 G0G90G94G81X12.Y10.Z8.R10.1S400F4.M3
N124 G80

(Blocks automatically performed in fixed cycle)

N123 X12.Y10.S400F0M3
N123 Z10.1F0
N123 Z8.F4.
N123 Z10.1F0

G82 Spotfacing Cycle (M)

Functional Description:

When programmed with X, Y, Z, R and F-word data, recognizes a clearance plane and feeds Z axis to within 0.010" (0.254mm) of programmed Z depth. At this point the spindle speed is changed automatically to 100 RPM and the feedrate reduced automatically to 0.5 IPM (12 MMPM) until the Z depth is satisfied. At the programmed Z depth a delay is forced (the length of the delay is set to two seconds but may be modified by the programmer to any desired value) then the tool is repositioned at rapid traverse to the clearance plane and modal spindle speed is restored.

Programming Application:

Suitable for spotfacing a series of holes.

Cancellation:

1. G82 is canceled by G80 in a block by itself.
2. Another fixed cycle cancels G82.

G82 EXAMPLE:

N123 G0G90G94G82X12.Y10.Z8.R10.1S400F4.M3
N124 G80

(Blocks automatically performed in fixed cycle)

N123 X12. Y10. S400 F0 M3
N123 Z10.1 F0
N123 Z8.010 F4.
N123 Z8. S100 F.5
N123 G04 F1.5
N123 Z10.1 F0
N123 S400

G83 Chipbreaking Cycle (M)

Functional Description:

When programmed with X, Y, Z, R, E and F-word information recognizes a clearance plane and feeds in by successive E-word increments until the Z depth is satisfied. A preprogrammed delay, or dwell, suitable for breaking chips is forced after each E-word increment. After the Z depth is satisfied the tool is repositioned at rapid traverse to the clearance plane.

NOTE

The dwell time is inversely proportional to the spindle speed and is calculated to exist for three revolutions of the programmed RPM. This dwell time may be altered by the programmer by modifying the associated parameters.

Programming Application:

Hole drilling where long chips cannot be tolerated.

Cancellation:

1. G83 is canceled by G80 in a block by itself.
2. Another fixed cycle cancels G83.

G83 EXAMPLE:

N123 G0G90G94G83X1.Y1.Z8.R10.1E.5S400F4.M3
N124 G80

(Blocks automatically performed in fixed cycle)

N123 X12.Y10.S400F0M3
N123 Z10.1F0
N123 Z9.6F4.
N123 G04F.4
N123 Z9.1F4.
N123 G04F.4
N123 Z8.6F4.
N123 G04F.4
N123 Z8.1F4.
N123 G04F.4
N123 Z8.F4.
N123 Z10.1F0

G84 Tapping Cycle (M)

Functional Description:

When programmed with X, Y, Z, R and F-word information recognizes a clearance plane and feeds to Z depth at programmed feedrate. When Z depth is satisfied, the spindle is reversed and a delay is forced which is sufficient to allow the spindle to achieve 80% of the programmed RPM. The Z axis then feeds back to the clearance plane. When the clearance plane is reached, the spindle reverses direction in anticipation of the next coordinate. This cycle can be initiated in either the clockwise direction (M03) or the counterclockwise direction (M04). While G84 is in effect, the feedrate and spindle speed are forced to 100% of their programmed value during actual cutting regardless of the setting of the override switches. Feedrate and spindle speed override are re-established during motion to new hole locations.

Programming Application:

Suitable for tapping a series of drilled holes.

Programming Instructions:

1. Determine the proper spindle speed to feedrate ratio for the tap (number of threads per inch, or millimeter). This ratio can be determined by using one of the formulas listed below.
2. The maximum feedrate for tapping cannot exceed 812.8 MMPM (32 IPM).
3. It is not required to Program M40 (Select Tapping Range) in the same data block containing the desired spindle speed.
4. (OPTIONAL) Program M49 (100% Feedrate and Spindle Speed) if the override capability is to be disabled between coordinates. If M48 is modal, the M49 command is automatically instated during actual cutting but not during positioning between holes. Normally the program calls for M49 to be in effect for the entire hole pattern, thereby sacrificing override capability between holes.
5. Consult the proper machine supplement manual for the proper tapping speed range.

Formulas for determining lead of tap:

ENGLISH SYSTEM TAPS

1. $\text{Threads per inch} = \dfrac{\text{spindle speed}}{\text{feedrate}}$

2. $\text{Feedrate} = \dfrac{\text{spindle speed}}{\text{threads per inch}}$

3. Spindle speed = threads per inch x feedrate

METRIC SYSTEM TAPS

1. $\text{Pitch} = \dfrac{\text{feedrate}}{\text{spindle speed}}$

2. $\text{Spindle speed} = \dfrac{\text{feedrate}}{\text{pitch}}$

3. Feedrate = spindle speed x pitch

Cancellation:

1. G84 is canceled by G80 in a block by itself.
2. Another fixed cycle cancels G84.

G84 EXAMPLE:

```
N123 G0G90G94G84X2.Y1.Z8.R10.1M40S210 F4.M3
N124 G80
```

(Blocks automatically performed in fixed cycle)

```
N123 X12.Y10.M40S210F0M3
N123 Z10.1F0
N123 Z8.F4.M5M49
N123 M4
N123 Z10.1F4.
N123 M3M48
```

G85 Reaming Cycle (M)

Functional Description:

When programmed with X, Y, Z, R and F-word information recognizes a clearance plane and provides a feed-in/feed out sequence suitable for reaming operations.

Programming Application:

Used for reaming a series of drilled holes.

Cancellation:

1. G85 is canceled by G80 in a block by itself.
2. Another fixed cycle cancels G85.

G85 EXAMPLE:

```
N123 G0G90G94G85X12.Y10.Z8.R10.1S400F4.M3
N124 G80
```

(Blocks automatically performed in fixed cycle)

```
N123 X12.Y10.S400F0M3
N123 Z10.1F0
N123 Z8.F4.
N123 Z10.1F4.
```

G86 Boring Cycle (M)

Functional Description:

When programmed with X, Y, Z, R and F-word information, recognizes a clearance plane and provides for a feed-in/spindle stop/rapid out sequence suitable for boring operations.

Programming Applications:

Used for boring a series of drilled holes, where a drag mark on the inside diameter can be tolerated.

Cancellation:

1. G86 is canceled by G80 in a block by itself.
2. Another fixed cycle cancels G86.

G86 EXAMPLE:

N123 G0G90G94G86X12.Y10.Z8.R10.1S400F4.M3
N124 G80

(Blocks automatically performed in fixed cycle)

N123 X12.Y10.S400F0M3
N123 Z10.1F0
N123 Z8.F4.
N123 M5
N123 Z10.1F0
N123 M3

G87 Deep Hole Drilling Cycle (M)

Functional Description:

When programmed with X, Y, Z, R, E and F-word information, recognizes a clearance plane; feeds in the E-word increment; retracts at rapid traverse to the clearance plane; advances to within 0.100" (2.54mm) of the last achieved E increment at 100.0 IPM (2540 MMPM); feeds in another E increment plus 0.100" (2.54mm) and retracts at rapid traverse to the clearance plane. This routine continues until the Z depth is satisfied. When the Z depth is satisfied, the Z axis retracts at rapid traverse to the clearance plane.

Programming Application:

Used for drilling deep holes where successive retraction of tool removes chip buildup.

Cancellation:

1. G87 is canceled by G80 in a block by itself.
2. Another fixed cycle cancels G87.

G87 EXAMPLE:

N123 G0G90G94G87X2.Y1.Z8.R10.1E.5S400F4.M3
N124 G80

(Blocks automatically performed in fixed cycle)

N123 X12.Y10.S400F0M3
N123 Z10.1F0
N123 Z9.6F4.
N123 Z10.1F0
N123 Z9.7F100.
N123 Z9.1F4.
N123 Z10.1F0
N123 Z9.2F100.
N123 Z8.6F4.
N123 Z10.1F0
N123 Z8.7F100.
N123 Z8.1F4.
N123 Z10.1F0
N123 Z8.2F100.
N123 Z8.F4.
N123 Z10.1F0

G88 Boring Cycle (ORIENTATION) (M)

The G88 cycle may be used with a default (+X axis move) or a programmable axis move by using the U88 Type II data statement.

Functional Description:

When programmed with X, Y, Z, R and F-word data, recognizes a clearance plane; feeds to programmed Z depth; and keylocks the spindle. By default the X axis moves in the positive direction 0.010" (.254 mm); then Z retracts to the clearance plane at rapid traverse; X is repositioned and spindle is restarted. The default +X move assumes that the point of the boring bar faces the control when the spindle keylocks.

To eliminate the need to preset all boring tools to accommodate the +X move, the programmer or operator may write a U88 Type II data statement which can program either or both of the X and Y axes to move a specified distance following the keylock.

NOTE

If only one axis is to be positioned, that axis and the axis that is to remain at zero must both be programmed. For example if only Y movement is required, the correct expression is (U88,X0Y-.045). Either axis not addressed in the Type II statement remains at its current (default or programmed) value.

G88 Boring cycles following this statement employ a motion of the X axis of -.05 and a motion of the Y axis of +.025 prior to retraction of the boring bar from the bore.

The offset is programmable only in the X and Y axes. The direction of the axis movement is established by the sign of the value. The control assumes the value is positive unless a negative sign is programmed.

The U88 statement is MODAL and remains in effect until canceled by one of the following commands: M30, M60, M02, [CLEAR ALL LOGIC], Sequence Number Search, or a different U88 statement.

PROGRAMMING A U88 STATEMENT

The U88 statement takes the following form:

(U88,Xm,Yn)

Where:

(= control out character
U88, = Mnemonic (comma is required)
m = is a ± offset to be associated with the X axis
n = is a ± offset to be associated with the Y axis
) = control in character

Example:

N10(U88,X-.05,Y.025)
N20G88
N30G80

Programming Application:

Suitable for hole boring operations where no cutter drag mark can be tolerated.

Cancellation:

1. G88 is canceled by G80 in a block by itself.
2. Programming another fixed cycle (G81-G87, G89) will cancel G88.

G88 Example:

N123 G0G90G94G88X12.Y10.Z8.R10.1S400F4.M3
N124 G80

(Blocks automatically performed in fixed cycle)

N123 X12.Y10.S400F0M3
N123 Z10.1
N123 Z8.F4.
N123 M19
N123 X12.01F0
N123 Z10.1
N123 X12.
N123 S400M3

G89 Feed In / Delay / Feed Out Cycle (M)

Functional Description:

When programmed with X, Y, Z, R and F-word data, recognizes a clearance plane and provides a feed-in / delay / feed-out sequence.

NOTE

The delay time is inversely proportional to the spindle speed and is calculated to exist for three revolutions of the programmed RPM. Again the programmer has the option of modifying the parameters to change the delay period.

Programming Application:

Suitable for boring operations requiring a smooth finish at the bottom of the hole.

Cancellation:

1. G89 is canceled by G80 in a block by itself.
2. Programming another fixed cycle (G81-G88) will cancel G89.

G89 Example:

N123 G0G90G94G89X2.Y1.Z8.R10.1E.5S400F4.M3
N124 G80

(Blocks automatically performed in fixed cycle)

N123 X12.Y10.S400F0M3
N123 Z10.1F0
N123 Z8.F4.
N123 G04F.4
N123 Z10.1F4.

G90 ABSOLUTE MODE (M)

Functional Description:

Establishes an operating mode in which the control interprets all axis commands (linear and rotary) as commands to an absolute position as referenced from the machine zero coordinates. Once an axis has been commanded to a position, it remains at that position until commanded to a different location. For example, if the X axis is currently at 10.0000, a command of X10. causes NO axis motion to occur.

Programming Application:

G90 is modal upon machine startup and is generally used when all dimensions on the part print are referenced from the same point. There is no advantage, in machine performance or accuracy, when using absolute mode rather than incremental mode.

Programming Instruction:

1. G90 axis commands consist of the axis letter address word followed by a number in the proper format which represents the commanded position.
2. The control assumes all commands as positive unless a negative sign is entered between letter address and number (e.g. B-270.).
3. G90 may be programmed in either G70 or G71 mode, and with either G00 or G01 in effect.
4. G90 has no effect on feedrates.
5. With a G98 offset (floating zero) in effect, the axis motion may be either positive or negative with respect to the zero point, but remains as an absolute position from that zero point.

Cancellation:

G90 cancels G91.
G90 is canceled by G91.

G91 INCREMENTAL MODE (M)

Functional Description:

Establishes an operating mode in which the control interprets all axis departure commands (linear and rotary) as commands to move the axes a specified distance in either direction from the current position. The current axis position becomes the zero reference point for the next axis departure command. Once an axis has been commanded to move a specified increment, it remains at that position until commanded to move an additional increment. For example, if X has been commanded to move 4 inches in the negative direction (X-4.) another command of X-4. moves the axis an additional 4 inches in the negative direction.

Programming Application:

G91 Incremental Mode is generally used when the dimensions on the part print are referenced from one point to the next point, rather than all dimensions being referenced from the same point. There is no advantage, in machine performance or accuracy, when using incremental mode rather than absolute mode.

Programming Instructions:

1. G91 axis departure commands consist of the axis letter address word followed by a number in the proper format which represents the incremental distance to be moved in that axis.
2. The control assumes all commands as positive unless a negative sign is entered between letter address and number (e.g. Y-6.). It is especially important in the G91 mode that axis departure commands be correctly signed.
3. G91 may be programmed in either G70 or G71 mode, and with either G00 or G01 in effect.
4. G91 has no affect on feedrates.

Cancellation:

1. G91 cancels G90.
2. G91 is canceled by:
   a. G90 Absolute Mode
   b. M02 End of program
   c. M30 End of prog., tape rewind, pallet shuttle
   d. Manual operation of [CLEAR ALL LOGIC]
   e. Manual operation of Sequence Number Search
3. G91 is bypassed on a one block only basis in response to the following commands which automatically instate the G90 mode:
   a. M06 Automatic Tool Transfer
   b. M25 Bypass Offsets

In both cases, the G91 mode of operation is reinstated in the following data block.

G93 INVERSE TIME FEEDRATE CODING (M)

(For additional information turn to Chapter 5)

Functional Description:

Establishes an operating mode which allows feedrate coding by the programming of a feedrate related F-word. The control converts the F-word to the time that is required to perform that block of information.

Programming Application:

G93 feedrate coding is applicable to any axis departure command; however, as G93 involves additional calculation, its use is generally restricted to the following programming conditions:

1. The generation of helical type cutter paths where the distance traveled due to linear motion exceeds

10% of the circumference of the circular motion (arc length).
2. The programming of a rotary (4th or 5th axis) command combined with linear interpolation.
3. The programming of two simultaneous rotary axes motions.

When the feedrate is calculated using inverse time, the control applies the feedrate directly along the cutter path (departure distance). As in the G94 feedrate coding system, axes motions begin together and finish simultaneously.

Programming Instructions:

The G93 F-word must be programmed in each block of axis motion. Although G93 is modal, the F-word is not modal, and must appear in each block even if the F-word does not change from one block to the next.

FORMULAS:

For circular interpolation with one axis linear interpolation (Helical Interpolation), either one of the following two formula may be utilized:

FORMULA A

$$\text{G93 F-Word} = \frac{\text{Desired feedrate (in IPM or MMPM)} \times \text{Angle in Radians}}{\text{Departure Distance}}$$

Where:

"Desired Feedrate" is that calculated for the particular cutter type and part material and the desired chip load (feed per tooth). The calculated feedrate is then expressed in IPM or MMPM.

"Angle in Radians" is calculated by multiplying the angle described by start point to centerpoint and end point to centerpoint (from centerline of cutter) by 0.01745 which is the number of radians in 1°.

"Departure Distance" is the resultant distance traveled by the tool along the helical cutter path. This consists of two parts:

1. The Arc Length. This is the length of the arc generated by the circular motion and can be calculated as follows:

   Radius of circular arc x Angle in radians.

2. The departure distance. Once the arc length has been determined, the departure distance is calculated as follows:

$$\text{departure distance} = \sqrt{(\text{arc length})^2 + (\text{linear motion})^2}$$

   where linear motion is the total distance traveled by the linear axes.

FORMULA B

1. This formula may be used in place of the one above.

$$\text{G93 F-Word} = \frac{\text{Desired Feedrate}}{\sqrt{\text{Radius}^2 + \frac{(\text{linear motion})^2}{(\text{angle in radians})^2}}}$$

2. For Rotary Axis motion combined with one axis linear motion.

$$\text{G93 F-Word} = \frac{\text{Desired Feedrate (in IPM or MMPM)}}{\text{Departure Distance}}$$

   where:

$$\text{Departure Distance} = \sqrt{(\text{rotary arc length})^2 + (\text{linear axis motion})^2}$$

   where: rotary arc length = angle in radians x distance from tool tip to pallet centerline

3. For rotary axis motion combined with two or three axes of linear

$$\text{G93 F-Word} = \frac{\text{Desired Feedrate (in IPM or MMPM)}}{\text{Departure Distance}}$$

   where:

$$\text{Departure Distance} = \sqrt{(\text{rotary arc length})^2 + (\text{linear motion})^2 + (\text{linear motion})^2}$$

   where rotary arc length = angle in radians x distance of tool tip to pallet CL

Cancellation:

1. G93 cancels G94.
2. G93 is canceled by:

   a. G94 - IPM/MMPM Feedrate coding
   b. M02 - End of program
   c. M30 - End of prog., tape rewind, pallet shuttle
   d. Manual operation of [CLEAR ALL LOGIC]
   e. Manual operation of Sequence Number Search

G94 IPM/MMPM FEEDRATE CODING (M) (For additional information turn to Chapter 5)

Functional Description:

Establishes an operating mode in which the control interprets feedrate commands directly in IPM or MMPM according to the following formula:

$$\text{G94 Feedrate} = \frac{\text{Axis movement in linear units or degrees - D}}{\text{Unit of time in minutes - T}}$$

Thus, the G94 feedrate is applied as follows, depending on the axis departure command that is programmed:

1. Feedrate for Linear Interpolation

   a. Single axis - programmed feedrate applied directly to path of single axis move.
   b. Multiaxes - programmed feedrate applied directly to vectorial path of multilinear move. Axis motion begins and ends, simultaneously.

2. Feedrate for Circular Interpolation

   Programmed feedrate applied directly along the programmed arc. The feedrate for each individual axis is calculated by the control to provide the programmed feedrate along the cutter path.

3. Feedrate for Helical Interpolation (2 axes circular, 1 axis linear)

   Programmed feedrate applied directly to arc path (circumference of arc) as is the case for circular interpolation above. The feedrate of the programmed linear axis departure is calculated by the control so that the circular motion and linear motion begin, and complete, simultaneously. If a more accurate feedrate along the cutter path (departure distance) is required, it is necessary to calculate a G93 Inverse Time feedrate. Normally, if the departure distance of the linear move does not exceed 10% of the arc length (circumference of arc) the G94 coding is adequate. If the linear departure distance exceeds 10% of the arc length, use G93 for accurate feedrate along cutter path.

4. Feedrate for Rotary Axis Motion

   Programmed feedrate applied directly to the rotary axis in units of DPM, up to a max of 999 if Inch mode (G70) or 2537 if in Metric mode (G71).

5. Feedrate for combined Rotary and Linear Motion

   Programmed feedrate applied directly to the vectorial path of multilinear move in IPM or MMPM. The feedrate applied to the rotary axis is calculated by the control so that the linear movement and the rotary movement begin, and complete, simultaneously.

   If the control calculates that the rotary axis must rotate at a feedrate which exceeds the maximum number of degrees per minute (i.e. short linear axis departure at high feedrate combined with a long rotary movement), the rotary axis moves at maximum DPM and the control generates an attention message.

   If an accurate feedrate along the cutter path is required, calculate a G93 F-word.

6. Feedrate for two Simultaneous Rotary Motions use G93 feedrate coding.

Programming Application:

G94 is used for most programming applications because it does not require the additional calculation that the G93 format requires.

Cancellation:

G94 is modal upon machine start up.
G94 cancels G93.
G94 is canceled by G93.

G98 INSERT ABSOLUTE OFFSET (M)

(G98 Offset Programming Instructions in Chapter 6)

Functional Description:

Allows the part programmer or machine operator to establish a new zero reference position for any, or all machine axes. For this reason, the G98 command is also referred to as "zero offset". G98 offset requires the positioning of the axes to a known coordinate followed by the programmed G98 code and the "required" coordinate position for each involved axis. NO axis motion occurs as a result of a G98 command. The axis display registers are updated to show the offset, or assumed, coordinates of the offset axes.

Programming Application:

1. Permits the use of the same part program on machining centers having different lengths of axis travel, provided travel limits are not exceeded.
2. Permits compensation for differences between assumed and actual fixture location.
3. When used with a dial indicator or Spindle Probe to align the centerline of the spindle with the centerline of a finished inside or outside diameter, enables the axes to be referenced accurately (offset) with respect to the diameter centerline.
4. G98 may be used in addition to any tool length comp., cutter diameter comp., or fixture offset.

Cancellation:

1. G98 is canceled only by a G99 command.
2. G98 offsets are bypassed on a one-time-only basis in response to the following commands which move axes to their absolute coordinate locations as referenced from machine zero.

    a. M06 Automatic Tool Transfer. Moves axes involved in tool transfer sequence to absolute tool change position.

    b. M25 Bypass Offsets. Moves axes to absolute position programmed in data block with M25 commands.

In both cases the G98 offset mode is reinstated in the following data block.

G99 CANCEL ABSOLUTE OFFSET (M)

Functional Description:

Cancels any offset in a linear indexing or rotary axis which was established by the programming of a G98 command. This is the only command that cancels a G98 offset.

Programming Application:

When programmed, the G99 cancels the G98 offsets and resets the machine axes, unless another offset mode (Tool Length Compensation, Cutter Diameter Compensation, or Fixture Offset [H]) is concurrently in effect also. In this case, the display registers show the offset, or assumed, positions that result as an effect of those particular offsets.

Programming Instructions:

G99 is to be programmed in a block by itself.

CHAPTER 3

MISCELLANEOUS FUNCTIONS

INTRODUCTION

Miscellaneous functions, or M codes, are commands which control an on/off function of a machine system or of a control system. The M codes are divided into groups of mutually exclusive commands. Certain codes in some of these mutually exclusive groups may be selected by the purchaser to be in effect upon machine startup. As these "modal upon startup" commands may differ from one machine to the next, the following pages of M code explanation do not indicate any one command of a mutually exclusive group as being modal upon the startup of the machine tool.

M codes are defined as being either Modal (M) or Non-Modal (NM). Modal refers to commands which, when programmed, remain in effect throughout subsequent data blocks until changed by the programming of a conflicting code or supplanted by the particular code from its group which is modal upon machine startup. Non-modal refers to commands which are in effect only during the data block in which they are programmed. The M codes are also defined as being either Immediate (I) or After (A) type functions. "Immediate" refers to commands which are acted upon immediately, before any axis motion in the block is imparted. "After" refers to commands which take effect after all axis motion in the block is completed. Only one after type miscellaneous function may be programmed in a block.

M00 PROGRAM STOP (NM - A)

Functional Description:

Causes a stop in the part processing cycle. Upon completion of the data block in which this command is programmed the machine tool is placed in a cycle suspend state, halting all axis motion, spindle rotation and coolant flow.

Programming Application:

1. May be utilized whenever the part programmer requires the part processing cycle to stop.
2. Allows operator to perform some manual function such as part inspection, fixture adjustment, etc.

Programming Instructions:

1. The part processing cycle can be resumed after a Program Stop by depressing [CYCLE START].
2. Miscellaneous functions M03, M04, M07 and M08, which are canceled by the M00 command, must be reprogrammed if they are to be used again.
3. Adequate operator instructions should accompany the program stop command if the operator is to perform some action. The programmer writes this message by means of a Type II data statement included in the part program.

Cancellation:

1. M00 cancels M03, M04, M07, M08, M57, M58 and M59.
2. M00 is canceled by pressing [CYCLE START].

M01 OPTIONAL PROGRAM STOP (NM - A)

Functional Description:

Provides the machine operator with the option of stopping the part processing cycle or continuing the program without interruption. The decision to halt or continue is determined by the operator's selection of the [ENABLE OPTION STOP] touchswitch/indicator located on the control front panel. When enabled the cycle suspend state functions in the same manner as an M00 command.

Programming Application:

May be used selectively at portions of the part program which may require a halting of the part processing cycle for some manual operation.

Programming Instructions:

1. Program an M01 at any point in the program where the operator may be required to perform some manual operation, such as inspecting a bore.

2. Provide adequate operator instruction in the form of a message, using a Type II data statement.
3. To resume the part processing cycle after an optional stop, depress [CYCLE START].
4. Miscellaneous functions M03, M04, M07 and M08, which are canceled by the M01 command, must be reprogrammed if they are to be used again.

M02 END OF PROGRAM (M - A)

Functional Description:

Indicates an end of program condition. Clears all active storage areas in the CPU, cancels parity check until the first End-Of-Block (EOB) is read in the forward direction of the tape or mass storage file and sets up the control to begin a new program when [CYCLE START] is depressed.

Programming Application:

Used for ending loop-type tapes.

Programming Instructions:

1. Return all axes to home positions.
2. Program M02 in a block by itself.

M03 SPINDLE ON CLOCKWISE (M - I)

Functional Description:

When used in combination with an S-word, establishes a clockwise (as viewed from the spindle) rotation of the spindle.

Programming Instructions:

1. Once a spindle speed has been established by the programming of an S-word, the M03 command initiates spindle rotation in the clockwise direction.
2. If programmed in a data block containing axis departure commands, spindle rotation at modal spindle speed begins before axis motion is initiated.

Cancellation:

1. M03 cancels M04, M05 or M19
2. M03 is canceled by M00, M01 (when enabled), M02, M30, M04, M05, M06, M19 or the operation of [CLEAR ALL LOGIC] or Sequence Number Search.

M04 SPINDLE ON COUNTERCLOCKWISE (M - I)

Functional Description:

When used in combination with an S-word, establishes a counterclockwise (as viewed from the spindle to the workpiece) rotation of the spindle.

Programming Instructions:

1. Once a spindle speed has been established with an S-word, the M04 command initiates spindle rotation in the counterclockwise direction.
2. If programmed in a block with axis commands, spindle rotation at modal spindle speed begins before axis motion is initiated.

Cancellation:

1. M04 cancels M03, M05 or M19.
2. M04 is canceled by M00, M01 (when enabled), M02, M03, M05, M06, M19, M30, or operation of [CLEAR ALL LOGIC] or Sequence Number Search.

M05 SPINDLE STOP (M - A)

Functional Description:

Commands the spindle to stop rotating and halts any programmed coolant commands.

Programming Instructions:

1. If M05 is programmed in a data block containing axis departure commands, the spindle rotation does not stop until all axis motion is complete.
2. Once an M05 command is executed, the spindle rotation can be resumed by reprogramming a M03 or M04 command

Cancellation:

1. M05 cancels M03, M04.
2. M05 halts coolant flow until spindle rotation is resumed.
3. M05 is canceled by M03, M04.

M06 TOOL TRANSFER (NM - A)

Functional Description:

Causes spindle rotation and coolant flow to stop and positions the axes slides to the tool change position. Automatically transfers a tool from the tool storage magazine to the spindle; or, if the tool is flagged as an "M" (Manual) tool will cause a [CYCLE STOP] to occur both at point of insertion and removal.

Programming Application:

1. Used to automatically transfer any non-oversize tool from the tool storage magazine to the spindle.
2. Used to initiate Manual Tool Change.

Programming Considerations:

1. When an M06 command is executed, the following M and G codes are automatically instated: G00, G90, G94, M05, M25 and M48. Upon tool change complete, modal M and G codes are reinstated.

   NOTE

   If the machine tool is equipped with tapered shank tooling, the M06 command automatically instates an M19 in addition to the above commands.

2. T-numbers may be programmed in the same block as an M06; however, it is recommended that a T-number be entered in the block following an M06.
3. An M06 block always operates at rapid traverse, regardless of the feedrate in effect. The modal feedrate is restored when the block is completed. A feedrate programmed in an M06 block is ignored.
4. If a Manual Tool Change is to be executed, one of two things will occur:
   a. If machine has the STATIONARY Tool Select Feature, the previous tool will automatically be removed from the machine spindle and replaced back in the tool magazine prior to [CYCLE STOP].

   Following completion of operations with the manual tool, the machine will again [CYCLE STOP] with the execution of the next M06 Command.

   The operator removes the Manual tool from the spindle, resets the[TOOL AUTO / UNCLAMP] to AUTO and presses [CYCLE START].

   The machine at this point will place the next Automatic Tool selected into the machine spindle and resume normal operations.

   b. If machine has the PROGRESSIVE Tool Select Feature, the machine will move to the tool change position and [CYCLE STOP] when M06 is executed.

   The operator must remove the previous tool from the machine spindle and set it aside. Then the operator places the Manual Tool in the machine spindle and presses [CYCLE START]. The machine resumes normal operations until the next M06, at which time the machine again cycle stops.

   At this time the operator removes the Manual Tool in the machine spindle, and replaces it with the last automatic tool previously removed.

   Enabling [SINGLE BLOCK] mode, the operator presses [CYCLE START] and the tool exchange is executed. The machine then cycle stops.

   The operator must then check the socket location of the tool returned to the magazine and then manually update the Tool Management Table with the correct socket location of the tool returned to the magazine.

   CAUTION

   The Tool Socket location is not automatically updated.

The operator then enables [AUTO] mode, and presses [CYCLE START].

NOTE

Some machines may be equipped with the optional PROGRESSIVE TOOL SELECTION with STATIONARY-OVERSIZE TOOL FEATURE. This option is discussed in Chapter 5.

Programming Instructions:

1. Move all axes, except those automatically positioned, to a part clearance position before the tool transfer occurs.
2. If M19 is programmed in the same block of information as an M06, the keylock is completed before the tool transfer begins. After an M19, whether programmed explicitly on straight shank tool equipped machines, or applied automatically as on tapered shank tool equipped machines, it is necessary to reprogram a spindle speed.
3. An M06 block may contain linear and/or rotary axis commands; however, if an axis is commanded to a location which is not the tool change position, that command is ignored. All axis motion is completed before the tool transfer begins.

NOTE

It is recommended that M06 be programmed in a block by itself; except for the MH-12.

4. Following a tool transfer, an M03 or M04 must be programmed to resume spindle rotation. If coolant ON is modal, coolant flow resumes when the spindle starts.
5. If G91 is in effect immediately after a tool transfer, the incremental departure distance commanded moves the indicated axis the programmed distance from the tool change position; therefore, it is recommended that the first block containing axis commands following a tool transfer be in G90 mode to position the axes to a specific location before continuing in the incremental mode.

M07 MIST COOLANT ON (M - I)

Functional Description:

Activates an air solenoid to the mist tank, providing mist coolant whenever the spindle is rotating.

Programming Considerations:

1. The [ENABLE COOLANT] touchswitch/indicator on the control front panel must be ON before programmed coolant functions can be actuated.
2. Coolant flows only when the spindle is rotating. When the spindle is stopped, coolant flow is interrupted. When the spindle is restarted, coolant flow is restarted.

Programming Instructions:

1. Only one type of coolant may be ON at a given time, unless machine has the optional coolant-through-the-spindle or coolant-through-the-tool.
2. If programmed in a data block containing axis departure commands, the coolant flow begins before axis motion is initiated.

Cancellation:

1. M07 cancels M08.
2. M07 is canceled by M08.
3. M07 is also canceled by the following commands: M09, M00, M01, M02, M30. If canceled by one of these codes, coolant must be reprogrammed.

M08 FLOOD COOLANT ON (M - I)

Functional Description:

Activates flood coolant pump, providing flood coolant whenever the spindle is rotating.

Programming Considerations:

1. The [ENABLE COOLANT] touchswitch / indicator on the control front panel must be ON before programmed coolant functions can be actuated.
2. Coolant flows only when the spindle is rotating. When the spindle is stopped, coolant flow is interrupted. When the spindle restarts, coolant flow is restarted.

Programming Instructions:

1. Only one type of coolant may be ON at a given time unless machine is equipped with optional coolant-through-the-spindle or coolant-through-the-tool.
2. If programmed in a data block containing axis departure commands, the coolant flow begins before axis motion is initiated.

Cancellation:

1. M08 cancels M07.
2. M08 is canceled by M07.
3. M08 is also canceled by the following commands: M09, M00, M01, M02, M30. If canceled by one of these codes, coolant command must be reprogrammed to initiate coolant flow.

M09 COOLANT OFF (M - A)

Functional Description:

Cancels any programmed coolant command.

Programming Considerations:

If programmed in a data block containing axis departure commands, the M09 command takes effect after axis motion is complete.

Programming Instructions:

Once M09 is executed, coolant flow can be resumed only by reprogramming a COOLANT ON command.

M19 KEYLOCK (NM - I)

Functional Description:

Commands the spindle to stop and locates the spindle in the keylock position. Keylock orientation varies depending upon machine tool.

Programming Instructions:

1. Program M19 to stop the spindle in the keylock position prior to transferring a keyed tool holder from or to the spindle.
2. Program a keylock before transferring the first tool of a part program into the spindle in case a keyed tool remains in the spindle from a previous machining operation.
3. After an M19 is commanded, reprogram the desired spindle speed when starting the spindle to ensure correct RPM. An M03 or M04 is necessary.

NOTE

Tapered shank machine tools keylock as part of the M06 automatic tool change command, straight shank DO NOT.

Cancellation:

1. M19 cancels M03 or M04.
2. M19 is canceled by M03 or M04.

M20 SELECT PRIMARY SPINDLE PROBE (M - I)

This is a modal command which is the default state upon machine start up or [CLEAR ALL LOGIC].

M21 SELECT SECONDARY SPINDLE PROBE (M - I)

The M21 command is a modal command which is cleared upon machine start up, or by a M20, M02, M30 or [CLEAR ALL LOGIC].

M23 Z PRIME ADVANCE (NM - A)

M24 Z PRIME RETRACT (NM - A)

MODU-LINES ONLY

Functional Description:

Applicable only to Traveling Column Modu-Line machining centers equipped with optional Z prime axis.

The Z slide positions the table to one of two positions depending upon the miscellaneous function (M23 or M24) in command. M23 advances the table toward the spindle, M24 retracts the table away from the spindle. Any machining operations can be accomplished in either position, thus affording greater capability in machining large workpieces.

Programming Instructions:

1. M23 or M24 must be programmed alone in a block.
2. The table can be indexed (or rotated) only in the retracted (M24) position.
3. An automatic tool transfer can be programmed in either position of the Z axis.
4. A pallet shuttle, manual or automatic, can occur only in the retracted (M24) position.

M25 BYPASS AXIS OFFSETS (NM - I)

Functional Description:

When programmed in a data block containing axis departure commands, causes all offset information associated with any axis to be bypassed on a per-block basis. This includes any axis offset created through a G98 command, a tool length compensation value, cutter diameter compensation value, fixture offset value, or axis inversion command. The axes move to the absolute position programmed in the data block.

Programming Applications:

1. M25 may be programmed when an axis is commanded to zero or the positive limit of travel to eliminate the possibility of axis overtravel due to axis offset.
2. Should be used if Z Axis is commanded to retract prior to automatic tool change.
3. Should be used if Z Axis is commanded to retract after an automatic tool change in the restart block.

Programming Considerations:

1. The M25 command applies only to the data block of axis motion in which it is programmed. Axis offsets remain in effect in the data blocks following the M25 block.
2. Axis departure commands in an M25 block are interpreted as absolute position commands (G90) regardless of whether incremental mode is in effect. If operating in incremental mode before the M25 block, the incremental mode is restored after the M25 block.
3. An M25 block operates at rapid traverse (F0) regardless of the previous modal feedrate. The modal feedrate is restored when the block is complete. A feedrate entered in an M25 block is operational.
4. The following entries are ignored and not recalled if programmed in a data block containing an M25 command: R, E, I, J, K, P, D, F, and all non-modal preparatory functions.

All modal preparatory functions entered in an M25 block become operative when the M25 block is complete, but have no effect on the operation of the M25 block.

Programming Instructions:

1. Departure commands for linear, indexing and rotary axes may be programmed in an M25 block.
2 If operating in incremental mode, program the first data block containing axis departure commands which follows the M25 block in the absolute mode (G90). This positions the axes to a specific absolute location before continuing in incremental mode.

M26 AUTOMATIC B AXIS CLAMP (M - I) (Optional Feature)

Applies to machines with Rotary B Axis Clamp only.

Functional Description:

Holds the B axis firmly in position when it is not commanded. M26 is the normal mode of operation for the control and is in effect upon machine startup. The clamp is automatically released at the beginning of a data block containing a B axis command and is automatically re-energized upon block completion.

Programming Application:

During part processing a modal M26 command prevents any unwanted movement of the B axis.

Programming Instructions:

It is unnecessary to program M26, at the beginning of a new program or in restart blocks, since it becomes effective upon machine startup and remains modal.

Cancellation:

M26 is canceled by M27.

M27 DISABLE B AXIS CLAMP (M - I) (Optional Feature)

Applies to machines with Rotary B Axis Clamp Only.

Functional Description:

Releases the B axis clamp when a B command is executed. The clamp remains released until an M26 is commanded.

Programming Application:

Can be used when all machining is occurring at or near the center of table rotation and numerous table indexes are commanded per part.

Programming Considerations:

The only advantage to be gained by programming M27 in place of the modal M26 is the saving of a small amount of time by eliminating the unclamping and reclamping of the B table during machining of the type noted above.

Cancellation:

M27 is canceled by M26, M02, M30, or the manual operation of [CLEAR ALL LOGIC] or Sequence Number Search.

M28 AUTOMATIC C AXIS CLAMP (M - I) (Optional Feature)

Applies to machines with Rotary C Axis Clamp Only.

Functional Description:

Holds the C axis firmly in position when it is not commanded. M28 is the normal mode of operation for the control and is in effect upon machine startup. The clamp is automatically released at the beginning of a data block containing a C axis command and is automatically re-energized upon block completion.

Programming Application:

During part processing a modal M28 command prevents any unwanted movement of the C axis.

Programming Instructions:

It is unnecessary to program M28 at the beginning of a new program or in restart blocks, since it becomes effective upon machine startup and remains modal.

Cancellation:

M28 is canceled by M29.

M29 DISABLE C AXIS CLAMP (M - I) (Optional Feature)

Applicable to machine tools with Rotary C Axis Clamp only.

Functional Description:

Releases the C axis clamp upon a C axis command. Clamp remains released until an M28 is commanded.

Programming Application:

Can be used when all machining is occurring at or near the center of table rotation and numerous table indexes are commanded per part.

Programming Considerations:

The only advantage to be gained by programming M29 in place of a modal M28 is the saving of a little time by eliminating the unclamping and reclamping of the C table during machining of the type noted above.

Cancellation:

M29 is canceled by M28, M02, M30, or the operation of [CLEAR ALL LOGIC] or Sequence Number Search.

M30 END OF PROGRAM - TAPE REWIND - PALLET SHUTTLE (M - A)

Functional Description:

Indicates the end of the program. Clears all active storage areas in the CPU, cancels parity check until the first EOB is read in the forward direction of the tape or mass storage file and sets up the control logic for a new program. Also, the M30 commands a tape rewind until the first EOB code is found and initiates an automatic pallet shuttle if the correct conditions are in effect.

Programming Application:

M30 is used to end any part program that is to be run from the tape reader or mass storage device.

Programming Instructions:

Program M30 in a data block by itself.

M40 SELECT TAPPING RANGE (M - I)

(Not applicable to MH-12 Machining Center)

Functional Description:

Makes available a separate group of spindle speeds to be used for tapping operations. Spindle motor speed remains low to provide proper torque for tapping and to allow fast spindle reversal.

Programming Application:

Use when programming a tapping operation to maintain low spindle motor speed for higher torque and for faster starts, stops, and reversal.

Programming Instructions:

1. Program after data block containing M06.
2. Program in the same data block containing desired tapping spindle speed.

Cancellation:

M40 is canceled by a new spindle speed command.

M48 ENABLE SPINDLE SPEED AND FEEDRATE OVERRIDE (M - I)

Functional Description:

Enables the [FEEDRATE OVERRIDE] and [SPINDLE SPEED OVERRIDE] selectors on the remote or control front panels.

Programming Application:

Permits the machine operator to manually override the programmed feedrate or spindle speed using the override selectors on the remote or control front panel.

Programming Instructions:

M48 is the normal mode of operation and is generally canceled only during tapping operations to ensure correct feedrate and spindle speed.

Cancellation:

M48 cancels, and is canceled by, M49.

M49 100% FEEDRATE AND SPINDLE SPEED (M - I)

Functional Description:

Disables the feedrate and spindle speed override selectors. The feedrate and spindle speed are forced to 100% of their programmed value regardless of the setting of the override selectors. The function setup display also may be used to lock out the spindle speed or the feedrate selector control switches (see Pub 883). Also, to allow the spindle to operate over 100%, the spindle override function must be enabled and the feature keyswitch must be in the disable position.

An exception to this is when the [FEEDRATE OVERRIDE] selector switch is set to 0%. In this condition, axis motion is inhibited, but the spindle rotates at 100% of the programmed spindle speed.

Programming Application:

Used whenever the programmed spindle speed and feedrate are critical to the part processing cycle.

Programming Considerations:

The programming of a G84 (fixed cycle-tapping) forces M49 during the metal cutting portion of the cycle.

Programming Instructions:

Program M49 at any point in the part program where it is necessary to maintain programmed feedrate and spindle speed values.

Cancellation:

1. M49 cancels and is canceled by M48.
2. M49 is canceled by M06.

M51 WASHHOUSE FLOOD COOLANT (M-I)

Applies to machines that have Washhouse feature.

Functional Description:

When M51 is commanded, flood coolant is forced out of multiple jets located on a special tower to the right of spindle centerline (viewed looking out from the machine spindle). This stream of flood coolant is used to wash a workpiece / fixture clear of any debris by simultaneously programming pallet rotation.

Programming Considerations:

1. This command is to be programmed immediately prior to pallet shuttle.
2. Coolant must be enabled.
3. Ensure washhouse doors are securely closed.

Programming Instructions:

1. Program M51 to enable Washhouse Flood Coolant.
2. Command B axis rotation to evenly distribute coolant over workpiece/fixture.
3. Use M09 to cancel Washhouse Flood Coolant.

Cancellation:

M51 is canceled by M09, M02, M30, M52 and the operations of [CLEAR ALL LOGIC] and Sequence Search.

M52 WASHHOUSE AIR BLAST (M-I)

Applies to machines that have Washhouse feature.

Functional Description:

When M52 is commanded an air blast is forced out of multiple jets located on a special tower to the right of spindle centerline (viewed from the machine spindle). This stream of air is used to dry a workpiece/fixture following a Washhouse Flood Coolant command by simultaneously programming pallet rotation.

Programming Considerations:

1. This is done immediately prior to pallet shuttle.
2. Coolant must be enabled.
3. Ensure washhouse doors are securely closed.

Programming Instructions:

1. Program M52 to enable Washhouse Air Blast.
2. Command B axis to rotate the workpiece/fixture.
3. Command M09 to cancel Washhouse Air Blast.

Cancellation:

M52 is canceled by M09, M02, M30, M52 and the operations of [CLEAR ALL LOGIC] and Sequence Search.

M57 AUXILIARY MIST COOLANT ON (M - I)

Applicable to machine tools with coolant-through-the-spindle or coolant-through-the-tool options.

Functional Description:

Activates auxiliary mist coolant-through-the-tool (straight shank only) or coolant-through-the-spindle (taper shank only).

Programming Considerations:

1. The [ENABLE COOLANT] touchswitch/indicator on the control front panel must be ON (lit) before programmed coolant functions can be actuated.
2. Coolant flows only when the spindle is rotating. When the spindle stops, coolant flow stops. When the spindle is restarted, coolant flow is restarted.

Programming Instructions:

1. Only one type of optional coolant may be ON at a given time.
2. If programmed in a data block containing axis departure commands, the coolant flow begins before axis motion is initiated.

Cancellation:

1. M57 cancels M07, M08, M58 and M59.
2. M57 is canceled by M07, M08, M58 or M59.
3. M57 is also canceled by the following commands: M09, M00, M01, M02, M30. If canceled by one of the above, coolant must be reprogrammed to initiate coolant flow.

M58 AUXILIARY FLOOD COOLANT ON (M - I)

Applicable to machine tools with optional coolant-through-the-spindle or coolant-through-the-tool.

Functional Description:

Activates auxiliary flood coolant-through-the-tool (straight shank only) or coolant-through-the-spindle (taper shank only).

Programming Considerations:

1. The [ENABLE COOLANT] touchswitch / indicator on the control front panel must be ON (lit) before programmed coolant functions can be actuated.
2. Coolant flows only when the spindle is rotating. When the spindle is stopped, coolant flow is interrupted. When the spindle is restarted, coolant flow is restarted.

Programming Instructions:

1. Only one type of optional coolant may be ON at a given time.
2. If programmed in a data block containing axis departure commands, the coolant flow begins before axis motion is initiated.

Cancellation:

1. M58 cancels M07, M08, M57 and M59.
2. M58 is canceled by M07, M08, M57 or M59.
3. M58 is also canceled by the following commands: M09, M00, M01, M02, M30. If canceled by one of the above, coolant must be reprogrammed to initiate coolant flow.

M59 AUXILIARY HIGH PRESSURE COOLANT ON (M - I)

Applicable to machine tools with optional coolant-through-the-spindle or coolant-through-the-tool.

Functional Description:

Activates auxiliary high pressure pulsating coolant-through-the-tool (straight shank only) or coolant-through-the-spindle (taper shank only).

Programming Considerations:

1. The [ENABLE COOLANT] touchswitch / indicator on the control front panel must be ON (lit) before programmed coolant functions can be actuated.
2. Coolant flows only when the spindle is rotating. When the spindle stops, coolant flow stops. When the spindle is restarted, coolant flow is restarted.

Programming Instructions:

1. Only one type of optional coolant may be ON at a given time.
2. If programmed in a data block containing axis departure commands, the coolant flow begins before axis motion is initiated.

Cancellation:

1. M59 cancels M07, M08, M57 and M58.
2. M59 is canceled by M07, M08, M57 or M58.
3. M59 is also canceled by the following commands: M09, M00, M01, M02, M30. If canceled by one of the above, coolant must be reprogrammed to initiate coolant flow.

M60 PALLET SHUTTLE within PROGRAM (M - A) (Optional)

Functional Description:

NOTE

This command is an optional command which is available on MM 180 Series machines with the D control.

Initiates a pallet shuttle sequence without creating an end of program condition or tape rewind. The requisite conditions involved for automatic pallet shuttle (axes at proper location, etc.) must be satisfied prior to executing the M60 command.

Programming Application:

May be used to process a single part program where the piece parts are to be machined in two separate operations. For example, if a piece part contains both rough boring and finish boring operations, the part programmer may write a single part program to perform both operations; but, if an inspection were required prior to the finish boring operation, the programmer could write an M60 command between the rough boring and finish boring segments of the program. The pallet containing the rough bored piece would shuttle away from the spindle allowing easier access for inspection while the spare pallet would shuttle in front of the spindle. The spare pallet could contain an already inspected rough bored piece and the remainder of the part program would finish bore the workpiece.

Programming Instructions:

1. Program M60 in a data block by itself.
2. Continue sequencing the data blocks in ascending numerical increments following the data block containing the M60 command.

M99 SOUND AUDIBLE ALARM (NM - I)

Functional Description:

Activates the control's audible alarm and generates the message "M99 alarm-See MAIN display" on the [MESSAGE] display. Once activated the alarm can be silenced by pressing the [ALARM RESET] button.

Programming Application:

Can be used to alert the operator to a message or deliberate stoppage of the part processing cycle.

Programming Instructions:

1. Program M99 in the data block preceding a programmable operator message and program stop command. The M99 command does not, by itself, halt the program. Example:

   N100 Z15.M25
   N120 M00M99 !Gauge bore size

2. Instruct operator to [CLEAR] the [MESSAGE] display after silencing alarm. If the M99 alarm message is allowed to remain on the [MESSAGE] display, the next occurrence of M99 will not sound the alarm.

CHAPTER 4

TYPE II DATA STATEMENTS

TYPE II DATA

Standard part programming language consists of the following elements:

1. The Letter Address System
2. Preparatory Functions (G Codes)
3. Miscellaneous Functions (M Codes)

As previously stated these codes form the vast majority of programmed commands, and are collectively defined as Type I data statements.

In order to provide the controller with greater flexibility, Kearney & Trecker has adapted Type II data statements. These commands are based upon formats taken from EIA Specification RS-447. The purpose of Type II data statements is to supplement the standard part programming language.

A Type II data statement consists of the following:

1. A "Control Out" character.
   ASCII RS-358 Format = (
   EIA RS-244 Format = %
2. A three-letter mnemonic (identifier) defining the function of the statement.
3. A comma (,) terminating the mnemonic string.
4. Body of statement - 73 characters maximum
5. A "Control In" character.
   ASCII RS-358 Format =)
   EIA RS-244 Format = %

CURRENT IMPLEMENTATION

The current Type II data statements in alphabetical order are:

(ACP) -Specify and Modify Adaptive Control
(CER) -Continue part program Execution Upon Device Errors
(CLF) -Close File
(CLS) -Call Subroutine
(CPY) -Copy a File
(DAT) -Update Date and Time
(DFC) -Define a Cycle
(DFS) -Define a Subroutine
(DLF) -Delete a File
(DLP) -Delete a Parameter
(END) -End a Statement
(EVT) -Record part program Event
(FWR) -Write to File
(FXC) -Load to Fixture Offset Table
(GTO) -Go to Label
(IFT) -Conditional "If True"
(LBL) -Label Definition
(MIR) -Mirror Image, Axis Inversion
(MSG) -Programmable Operator Message
(NAM) -Name a part program
(OIF) -Name a Spindle Probe Inspection File
(OPF) -Open a File
(PAR) -Parameter Assignment Statement
(PRT) -Print File
(RCP) -Copy a File to or from a Remote Station
(SER) -Stop Program Execution Upon Device Errors
(SET) -Set an Output
(TLD) -Load a Tool Table with Tool Parameters
(U88) -Programmable Boring Orientation Cycle
(UAT) -Select Alternate Tool Data
(UCY) -Maximum part program Execution Time-FMS
(UFX) -Control Calculated Fixture Offset
(ULF) -Recalls Saved Fixture Offset Table
(USF) -Fixture Offset Table Saved on Mass Storage Media
(UTL) -Identifies All Tools Used in a part program-FMS
(WAI) -Wait on Input
(ZTB) -Zero Either Tool Table or Fixture Offset Table or Both

ACP SPECIFY AND MODIFY ADAPTIVE CONTROL STATEMENT

Functional Description:

The ACP Type II data statement is used to when the Adaptive Control feature is specified. The primary function of the ACP statement is to set the Target Horsepower for Adaptive Control cutting operations. A secondary function of ACP is to alter percentage values associated with the Adaptive Control display.

The following demonstrates the format of the ACP Type II data statement:

ASCII = (ACP,XX.X,X=XXX)
EIA = %ACP,XX.X,X=XXX%

Where: XX.X = Target Horsepower
X=XXX = Parameter Revision Value

NOTE

EIA punched tape format has no provision for the equal sign (=) among the standard character set; however, the control recognizes punches in channels 2, 3, 4, 5, and 7 as an equivalent to the equal sign in Type II data statements using the EIA format.

The following are the default percentages for Adaptive Control. These values may be modified within certain limitations. For more information, see Kearney & Trecker Pub 850 - Adaptive Control part programming & Operation.

DEFAULT ADAPTIVE CONTROL PARAMETERS

Param	Percent	Function
1	160	Airgap Feedrate
2	155	Maximum Feedrate
3	130	Decrease Target HP Feedrate
4	125	Increase Spindle RPM Feedrate
5	80	Decrease Spindle RPM Feedrate
6	65	Increase Target HP Feedrate
7	60	Minimum Feedrate
8	125	Airgap RPM
9	110	Maximum RPM
10	90	Minimum RPM
11	110	Maximum Horsepower
12	85	Minimum Horsepower
13	.4997 ms	Minimum Feedstall Time

TABLE 4-A

Programming Application:

1. Used prior to initiating an Adaptive Control cut in order to set the Target HP desired for the cut.
2. Used to modify Parameter Values associated with Adaptive Control.

Programming Instructions:

1. Enter ACP statement in the block immediately preceding a programmed G51 command turning on the Adaptive Control Feature.

NOTE

If Adaptive Control is turned off for any reason, the ACP statement remains valid provided the cutter is not removed and that the Metal Removal Rate (MRR) (Width X Depth X Feedrate) is not altered.

Any time a different cutter is inserted and Adaptive Control is to be used, a new ACP Target Horsepower should be commanded defining Target Horsepower for the new tool.

2. In the block containing the G51 command, both feedrate (F-Word) and spindle speed (S-Word) must be commanded in order for the controller to calculate actual values for defined percentages.

CER CONTINUE MACHINE OPERATION STATEMENT

(For more information on this command see Chapter 7)

Functional Description:

The CER command is used to command the control to continue machine operation, even if certain specified file or device input or output (I/O) errors are encountered during program execution. Upon detection of such an error, a warning message is displayed (no alarm is sounded), but the part program continues to run. This feature was incorporated in the control to allow for cases where the operator might not be present for the entire period that a part program is running, and where it is required that the part production is not halted.

Programming Application:

Enables the machine program to continue running in spite of errors encountered during operation.

Programming Instructions:

1. To specify that machine operation be continued if only a mass storage error is noted, enter (CER,MAS).
2. If machine operation is not be halted upon detection of a sequential I/O error, enter (CER,SEQ).
3. To prevent cycle interruption upon any type of mass storage or I/O error program (CER,MAS,SEQ,REM).
4. Additionally, any combination of two device type mnemonics is also valid.

CLF CLOSE FILE STATEMENT

(For more information on this command see Chapter 7)

Functional Description:

Used to close any file opened by (OIF, filename), (OPF, filename/type) or (OPF, filename,A/type) commands.

Programming Instructions:

1. The CLF command must be used to close the (OPF,filename,A/type).
2. The CLF command does not cancel an active G36.

File access status parameter L1050 may be used with the CLF command to provide a means of allowing the part program to wait until after a file is closed before processing anymore program blocks. The part program will set parameter L1050 to zero before the open file command. The part program will then wait after the CLF block until the file is closed by using a (LBL,10) (IFT,L1050=0,(GTO,10)) block. The system will set L1050 to non-zero after the file is closed. The use of this parameter is optional. If the parameter L1050 is not created by the part program then the CLF will close the file and skip the setting of L1050 to non-zero.

CLS CALL SUBROUTINE STATEMENT

(For more information on this command see Chapter 8)

Functional Description:

The CLS statement commands the control to use the designated subroutine.

Programming Instructions:

The CLS statement may take the following forms:

1. (CLS,name,count)
2. (CLS, name, count, maximum number of 11 parameters assigned, not including the count as a parameter)
3. (CLS,name,count) followed by program blocks

The run count is displayed in the parameter table as L0 and is counted down when the subroutine reaches the (END) statement. When the (END) is recognized and L0 equals zero, the subroutine is terminated and the control returns to the program that called for the subroutine to the block following that which called the subroutine. When no count is specified the control assumes a value of 1 for the count value. Example:

```
N100 (CLS,Circle8,2,L5=7.) !A run count of 2 passes
```

The run count cannot be passed into the control as a mathematical expression, but it may be expressed like any other parameter. Example:

```
N100(CLS,Circle8,L5=7.,L0=L3+2.)!The run count
                                 is expressed as a parameter (L0)
```

Parameter labels may be used in the subroutine up to a maximum of 11 parameter assignments.

CPY COPY A FILE STATEMENT

(For more information on this command see Chapter 7)

Functional Description:

The CPY command allows an existing file to be copied to another file from within a part program.

Programming Instructions:

1. Both the source and the destination file must be of the same type.
2. The copy direction is specified by the use of "TO" or "FROM". If "TO" is used, it is assumed that the first named file is the source file and the second named file is the destination. The opposite is true if the word "FROM" is used. Example:

```
N120 (CPY,PROGRAM1, TO, PART3)
```

3. If a file called PART3 doesn't exist, a file called PART3 is created and the contents of PROGRAM1 are copied to it.
4. Upon successful search and copy of a file, the previous contents of the destination file are erased from the system, so that a file containing the information from PROGRAM1 (the sending file) exists in the control under the name of PART3 (the new destination file).

DAT UPDATE DATE AND TIME PARAMETER STATEMENT

(For more information on this command see Chapter 8)

Functional Description:

The DAT command permits the updating of year, month, day, hour, minute and second parameters L3061-L3066 and the insertion of this information into an inspection report or file.

Programming Instructions:

```
L3061=   Year
L3062=   Month
L3063=   Day
L3064=   Hour
L3065=   Minute
L3066=   Second
```

Example:

```
N210 (OPF, PUMP234)
N220 (DAT)
N230 (FWR,"YEAR=",L3061)
N240 (FWR,"MONTH=",L3062)
N250 (FWR,"DAY=",L3063)
N260 (FWR,"HOUR=",L3064)
N270 (FWR,"MINUTE=",L3065)
N280 (FWR,"SECOND=",L3066)
  :      :
(CLF)
```

Output Results: (sample only)

```
YEAR=1987.0000
MONTH=5.0000000
DAY=16.000000
HOUR=11.000000
MINUTE=9.0000000
SECOND=11.000000
```

The result of this command would be that an file would be opened with the name PUMP234 and would include the current commanded time parameters as generated by the control's internal clock. It is the programmer's responsibility to include the desired parameters in the DAT command.

DFC DEFINE A CYCLE STATEMENT

(For more information on this command see Chapter 8)

Functional Description:

The DFC statement permits the programmer to define and name cycles which have been constructed by the programmer and save these cycles to the disk under some chosen name.

Programming Instructions:

1. A cycle may be defined as "local" in that it is used only with in a specific program or it may be a "general" cycle that has been "saved" to the disk storage and can be accessed by other programs.

Example:

```
N60 (DFC,G881)        !An example of a local cycle
                       definition.
Program Blocks
   "
   "
N65 (END)

N70 (DFC,G882,SAVE)   !An example of a saved
                       cycle.
Program Blocks
   "
   "
N75 (END)
```

2. If the command "SAVE" is included in the statement the cycle is saved on the disk under the cycle name and may be used by other programs by calling the cycle from the disk by using its name (G801 thru G899).

DFS DEFINE A SUBROUTINE STATEMENT

(For more information on this command see Chapter 8)

Functional Description:

The DFS command is used to identify to the control the information contained in the body of a subroutine.

Programming Instructions:

The DFS command must precede the program blocks in a subroutine and contain a name that identifies that subroutine. It is optional to utilize the SAVE command. If the SAVE command is used the subroutine is saved to the disk storage under the assigned name and may be used by other programs.

Example:

```
N65(DFS,CIRCLE8,SAVE)
Program blocks
   "
   "
N95(END)
```

Part program blocks containing DFS commands are not allowed to be cleared by either the [CLEAR READY] or thru the use of the MDI feature. Modification of a block containing a DFS statement is also not allowed while the program is active in the control.

DLF DELETE A FILE

(For more information on this command see Chapter 7)

Functional Description:

This Type II data statement permits the programmer to delete any file that is stored on the disk. This includes part programs, subroutines and cycles.

Programming Instructions:

1. The file to be deleted must be closed.
2. Either ASCII or NC files may be deleted. The default file status is ASCII.
3. Care should be used in deleting file names containing variable parameter values as the value of the parameter may have changed from the time at which the file name was created.

Example:

N20 (DLF,PUMP345)	Deletes a part program
N30 (DLF,G882)	Deletes a user-defined cycle
N40 (DLF,DRILL25)	Deletes a subroutine

DLP DELETE A PARAMETER STATEMENT

(For more information on this command see Chapter 8)

Functional Description:

Allows the programmer to delete parameters from the parameter table when the parameter table capacity of 100 parameters has been reached or those parameters are no longer needed.

Programming Instructions:

The statement should contain the parameters that are to be deleted.

Example: (DLP,L24,L57,L30,etc.)

The result of this example would be to delete parameters 24, 57 and 30 from the parameter table.

END TERMINATE A SUBROUTINE OR DEFINED CYCLE STATEMENT

(For more information on this command see Chapter 8)

Functional Description:

The END statement is used by the programmer to terminate a subroutine or a user-defined cycle.

Programming Instructions:

1. The END statement is required to terminate a subroutine or user-defined cycle.
2. The END statement is required to enable the run count function to operate.

EVT RECORD PART PROGRAM EVENT STATEMENT

Functional Description:

This statement is used to record an event in UMC-FMS applications. This data is then transferred a central supervising computer.

FWR WRITE TO FILE STATEMENT

(For more information on this command see Chapter 7)

Functional Description:

It provides increased flexibility to the programmer in writing messages into an opened inspection file which may contain upper and lower case characters, commas, quotation marks, appended statements, parameters, axis positions and free format operations. Also, the use of this command permits the preprocessing of program information for faster program operating times.

Programming Instructions:

1. Any data line that is in excess of 80 characters will be shortened by the control to 80 characters.
2. Commas may be used inside the text statement after their use following the FWR command.
3. The output file will not contain the FWR command. The quotation mark (") is the symbol used for free format text insertion in the FWR command.
4. Multiple lines of FWR information may be written into a single line by use of the semicolon (;). It must appear as the final character before the closing parenthesis.

Examples:

Input: (FWR,"Using a comma, for this demonstration")
Output: Using a comma, for this demonstration

Input: (FWR, "Parameter L20 equals",L20)
Output: Parameter L20 equals 9.8000000

Input: (FWR, "Parameter L10, truncated, equals", L10I)
Output: Parameter L10, truncated, equals 123.

Input: (FWR, "This example, using a semicolon, ";)
(FWR, "will be added to this line.")
Output: This example, using a semicolon, will be added to this line.

FXC LOAD CALCULATED OFFSETS TO FIXTURE OFFSET TABLE

Functional Description

The FXC Type II data statement is used to load predetermined (discrete) values into the Fixture Offset Table ([SELECT] or [DISPLAY LIST] [U]).

These values are useful where multiple fixtures are to be used on the same Manufacturing Center and there will be little, if any, fixture tear-down and rebuild.

Using the FXC Type II data statement, the user may program load up to 32 fixture offset values per table at a time. If more offsets are desired, multiple fixture offset tables may be created (see the entries on the USF and ULF commands in this chapter) and additional FXC Statements may be entered at some point later in the part program.

The format of this command is as follows:

ASCII = (FXC,HnInXnYnZnAnBnCn)
EIA = %FXC,HnInXnYnZnAnBnCn%

Where: H = Fixture Offset Identifier
I = Implicit Axis Position
X = X Axis Offset
Y = Y Axis Offset
Z = Z Axis Offset
A = A Axis Offset (If so equipped)
B = B Axis Offset (If so equipped)
C = C Axis Offset (If so equipped)
(For use with the calculated fixture offset option)
Ai = A Axis Initial Offset
Bi = B Axis Initial Offset
Ci = C Axis Initial Offset

Programming Application:

Used when it is desired to Program Load the relationship of the fixture or a part feature as stated in the part program to the true position as determined by measurement.

Programming Instructions:

1. FXC Commands are generally programmed at the beginning of a part program or, in a separate program on the Program Management Table.
2. Command FXC and the known values to be established for the particular fixture offset.

Example:

N122 (FXC, H11 I0 X-.0012 Y.0023 Z-1.125)
N123 (FXC, H12 I90. X1.125 Y.0023 Z-.0012)
N124 (FXC, H13 I180. X.0012 Y.0023 Z1.125)
N125 (FXC, H14 I270. X-1.125 Y.0023 Z.0012)

FXC FIXTURE OFFSET SUBSTITUTION

(For more information on this command see Chapter 8)

KT Advanced Programming Language has the capability of utilizing parameter values instead of numerical values.

Programming Instructions:

1. Parameters used to replace the H number must be whole numbers; if not, the control will round to the nearest whole number. Also, the parameter may only range from 1 to 32. The parameter value must be a valid axis offset value.
2. Parameter values from 0 to 2999 are valid.

Example: (FXC,HL22XL2YL43)

GTO GO TO STATEMENT

(For more information on this command see Chapter 8)

Functional Description:

The GTO statement causes the control to search for a labeled reference in the currently active level of the program or the label table.

Programming Instructions:

1. The control will search the currently active main program or subroutine for the designated label.

2. The control can access four levels of the part program. Level 0 is the main part program. Levels 1 to 3 are the subroutine levels. Level 1 could be called from level 0. Level 2 could be accessed from level 1. Finally, level 3 will be called by level 2. The GTO target must be on the currently active level of the program.
3. The control will first search the label reference table which contains the last 10 labels referenced. If the label is not found it will first search in the forward direction in the part program. If the label is not found there, the part program will be searched from the beginning of the current level to the current block.
4. The GTO statement must be the only statement in a program block.
5. Destination labels are numbers from 1 to 9999.
6. Within levels 1 to 3 the END terminator must be in the last block of that level.

IFT CONDITIONAL "IF TRUE"

(For more information on this command see Chapter 8)

Functional Description:

The IFT statement operates with three types of expressions. First, it may use a Boolean expression followed by a Type I data statement such as:

```
N25(IFT,G98,G70)        !If G98 is active, G70 is active
```

Second, it may employ a Type II statement like:

```
N30(IFT,G98,(GTO,125))  !If G98 is active, go to label
                         125
```

Third, a parameter assignment can be used in the expression:

```
N35(IFT,G98,(Par,L23=Z))    !If G98 is active,
                             parameter 23 is given
                             the Z axis value shown
                             on the main display.
```

The following format symbols are used with the IFT statement:

1. Boolean expression = a currently active code
2. NOT (a boolean expression)
3. DEF (a parameter defined in the parameter table)
4. (a boolean expression)AND(a boolean expression)
5. (a boolean expression)OR(a boolean expression)
6. (value expression) <relation operator> (value expression)
7. An active code, either a G or M code, makes the expression true.
8. The relational operators: <>, >, <, =<, >=,. (<> is used to indicate a not equal sign)
9. A value expression is: =, +, -, /(division sign), *(multiplication sign)
10. Constant: ±99,999,999.
11. Letter address of: A, B, C, E, F, I, J, K, P, Q, R, S, T, U, V, W, X, Y, Z. Values used are commanded program positions as displayed on the main display including offsets.
12. Function: SQ, SQRT, ABS, SIN, COS, TAN, ASN, ACS, ATN, FRA, INT, LOG, ALOG, LN, EXP
13. Parameter address: L1...Ln
14. Type II data statements: (IFT,Type II data statement). The DFS, DFC, END, LBL and IFT are excluded from use in the IFT statement.

Operator precedence when parentheses are not used:

1. NOT and DEF
2. AND
3. OR
4. Relational expressions =, >, <, etc.

LBL LABEL STATEMENT

(For more information on this command see Chapter 8)

Functional Description:

The LBL command is the objective of the GTO command. The control searches the currently active program or subroutine in either a forward or backward direction or accesses the label reference table. The label reference table contains the last ten active labels of the active program or subroutine.

Programming Instructions:

1. The LBL statement may be combined on a line with either Type I or II statements.
2. The LBL statement inside a Type II statement or in a comment will not be referenced by a GTO statement.
3. The LBL must contain a whole number from 1 to 9999.

Example: N340(LBL,1492)

MIR MIRROR IMAGE, AXIS INVERSION STATEMENT

(For more information on this command see Chapter 5)

Functional Description:

The Mirror Image feature provides program control of Axis Inversion. With this feature the programmer may enter commands which cause X, Y or XY Axis commands to be inverted producing matching workpieces.

The state of Mirror Image is controlled by an MIR Type II data statement. The following shows the format and specifications for the Mirror Image command.

ASCII = (MIR,XnYn)
EIA = %MIR,XnYn%

Where:
(MIR,X1Y0) = Invert X Axis, but not Y Axis.
(MIR,X0Y1) = Invert Y Axis, but not X Axis.
(MIR,X1Y1) = Invert both X and Y Axes.
(MIR,X0Y0) = Invert neither X Axis nor Y Axis

Default = X0Y0

Programming Application:

Used when is desired to produce matching workpieces or patterns.

Programming Instructions:

1. Position the X and / or Y Axis to a centerpoint defined as the Axis of Symmetry.
2. Either command G98 X0 and / or Y0; or alternately command G91 to invoke the Incremental Mode.
3. Enter the desired Mirror Image Type II data statement.
4. Program all movements with respect to the base work pattern desired.
5. Repeat program commands for normal orientation.
6. When completed, command normal Mirror Image state [(MIR,X0Y0)].
7. Cancel G98 and/or return to Absolute Mode G90.

MSG MESSAGE STATEMENT
! COMMENT STATEMENT

(For more information on these commands, see Chapters 7 and 8)

Functional Description:

NOTE

In KT Advanced Programming Language the use of ! comment sign has replaced the use of the MSG statement for operator messages, but the control will still recognize the MSG statement as a valid command. The programmer should still use the M00 or M01 command to stop the program and allow time for the operator to read the message and carry out the instructions.

The MSG command allows the programmer to include a message for the machine operator. The message appears on the CRT in normal screen magnification as a standard data block.

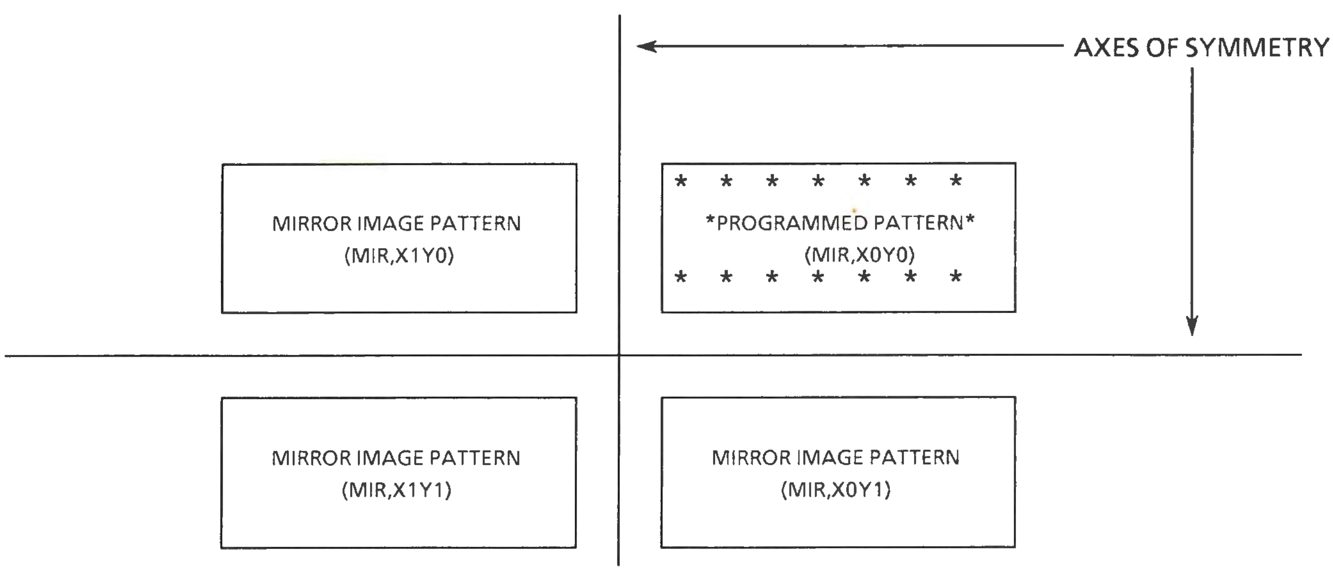


Fig 4.1 MIRROR IMAGE

Programming Application:

Enables the programmer to relay specific information, or instructions, to the machine operator.

Programming Instructions:

1. The programmable operator message is written with a Type II data statement with the mnemonic (three letter code) MSG,.
2. To stop the part program with the message being displayed, a Program Stop command (M00) must be programmed in the data block immediately preceding the data block containing the message.
3. To display a message without halting the part program, a programmable dwell may be used in the block preceding the message. The length of dwell would draw attention to the machine, but would not halt part processing.
4. Messages may be any length but a single line of an MSG message cannot contain more than 73 characters including spaces. If the message exceeds this maximum, create a new line with the remainder of the message. Each additional line however, must begin with the Control Out character, "MSG," Type II data format and be terminated with the Control In character, ")".
5. A maximum eight lines of message data may be on a single display. The Program Stop Command M00 or G04 Programmable Dwell would be in the Active Position, with seven additional lines of data being on the display at this time.
6. Message blocks may begin with a sequence number.

Example:

N210 M0
N220 (MSG, CHECK BORE SIZE) Message within a single line.
N100 M0 Message too long for one line; continuation of message programmed on second line.
N110 (MSG,ALIGN SPINDLE CENTERLINE)
N120 (MSG,WITH PART FEATURE A1)
N130 G0G90G94Z20.M25

Example:

N210 M0
N220 !THIS IS AN EXAMPLE OF A COMMENT STATEMENT
N230 !JUST LIKE THE MESSAGE STATEMENT, THE LINE
N240 !MAY ONLY HAVE A LIMITED NUMBER OF
N250 !CHARACTERS ON IT

Due to size limitations on this page a smaller type size is used in this example to simulate how the message will appear on the screen.

NAM NAME A PART PROGRAM

Functional Description:

The NAM statement lets you name a part program on punched tape with a Type II data statement.

Programming Application:

Use of the NAM Statement saves operator input time and eliminates the possibility of keyboard input error when copying a part program from punched tape onto a mass storage device using the File Handler.

Programming Instructions:

1. The NAM statement is written using a Type II data statement with the mnemonic NAM,.
2. The NAM statement must precede the first sequence numbered data block of the part program.
3. The file name in the NAM statement may consist of alpha and / or numeric characters not to exceed 32 characters.
4. When using the File Handler to copy from punched tape to mass storage, neither the "Original File" nor the "Copy File" prompts require entry in the FILE name field of the file specifier. (See COPY command instructions in File Handler writeup of the operator's manual.)

Example:

(NAM, GEAR HOUSING)

MSG, NAM STATEMENT
(For more information see Chapter 7)

Functional Description

NOTE

Earlier versions of Kearney & Trecker controllers required Inspection Programs be named using the the Type II data statement MSG,NAM. To enhance KT Spindle Probe user upward capability, the MSG,NAM may still be used to name inspection files. Function is identical to that of the OIF statement.

Programming Application:

The MSG,NAM Type II data statement is to appear in the block immediately following the G36 Open Inspection File Command.

Programming Instructions:

1. When it is desired to inspect a workpiece using the Spindle Probe, command G36 to open the inspection file.
2. The block following the G36 command MUST have the MSG,NAM statement naming the inspection file which is opened for writing, on the disk.
3. The inspection file is closed and saved with a G37 Close Inspection File command or the file may be closed with a CLF command which will not require the reintroduction of the G36 command.

NOTE

Subsequent executions of the MSG,NAM command will cause the existing MSG,NAM named file to be written over, if the file has the same name.

Example:

```
N10484 M6
N10485 G0G90G94X5.Y12.Z18.M25
N10486 G25
N10487 G28X6.1Y7.884Z8.471F0
N10488 G29
N10489 G36 ← Open Inspection File
N10490 (MSG,NAM,PART TEST - CT2759)
 ...
 ... (Spindle Probe Inspection Routine)
 ...
N10544 G37 ← Close Inspection File
```

OIF NAME A SPINDLE PROBE INSPECTION FILE STATEMENT

(For more information on this command see Chapter 7)

Functional Description:

The function of the OIF Type II data statement is to create a name for an inspection file for those machines which are equipped with the Spindle Probe feature. The OIF statement MUST be in a block immediately after the G36 command opening the inspection file.

Programming Application:

The OIF Type II data statement is to appear in the block immediately following the G36.

Programming Instructions:

1. When it is desired to inspect a workpiece using the Spindle Probe command G36 to open the inspection file.
2. The block following the G36 command MUST have the OIF statement naming the inspection file which is opened for writing, on the disk.
3. The inspection file is closed and saved by executing a G37 Close Inspection File command or a CLF command. The use of the G37 command will cancel the G36 command. It is recommended that the CLF command be used instead of the G37 command to eliminate the need to reintroduce the G36 command to open another file.

NOTE

Subsequent executions of the OIF command will cause the existing OIF named file to be written over and that information is lost, if that file has the same name.

Example:

```
N10484 M6
N10485 G0 G90 G94 X5. Y12. Z18. M25
N10486 G25
N10487 G28 X6.1 Y7.884 Z8.471 F0
N10488 G29
N10489 G36 ← Open Inspection File
N10490 (OIF,PART TEST - CT2759)
 ...
 ... (Spindle Probe Inspection Routine)
 ...
N10544 G37 ← Close Inspection File
    (or a (CLF) command may be used to close the file)
```

OPF OPEN A FILE STATEMENT

(For more information on this command see Chapter 7)

Functional Description:

The OPF command statement allows the programmer greater flexibility in the creation of an inspection file.

Programming Instructions:

The OPF command differs from the OIF statement in the following areas:

1. Either ASCII or NC files may be created.
2. A G36 command does not have to precede the OPF.
3. None of the machine formatted information output by (OIF,filename) Type II processor (time, date, etc.) is written to the file.

4. Variable file names may be created by the use of parameters directly into the file name.
5. The OPF command is ignored, if a file with the same name has been created and is still in use. If a previous file is still open, it is closed before the OPF command is recognized. If a file exists with the same name, the previous file is deleted and the information is lost. If a similarly named file exists, but is a different file type the systems sounds an alarm and the message "FILE ALREADY EXISTS" is displayed.
6. Files opened with the OPF command must be closed with the CLF command.
7. If the programmer wishes either ASCII or NC files may be created by the placement of either the letters "AS" or "NC" in the "TYPE"portion of the (OPF,FILENAME,TYPE) command. If no file type is indicated the control will assume that an ASCII file is desired. Use of other type commands will generate "Error-MUST BE EITHER NC OR ASCII" message.
8. Variable files names are programmed with the use of the (<) left and (>) right arrow characters which are used to contain one or more parameters. These parameters must have been previously defined in the part program or are system parameters. Truncated parameters are created by using the character (I) to indicate that the integer value of the parameter is to be used in the OPF statement. The value of the parameter is not changed, only that used in the file name is modified.
9. An OPF file may be accessed a number of times, by using without loss of data, the format of (OPF,FILENAME,A/TYPE) where the A/ indicates to the control that it should append new information to the existing file. If no file exists one will be created with the indicated name.

Examples:

N200 (PAR,L25=123.456)	
N300(OPF,TEST<L25>)	!Example containing a parameter
N400 (CLF)	
N500(OPF,TEST<L25I>)	!Example containing a truncation
N600 (CLF)	
N700 (PAR,L27=1.567)	
N800 (OPF,TEST<L25I><L27I>)	!An example of multiple parameters
N900 (CLF)	

Output: TEST123.45600
TEST123
TEST1231

PAR PARAMETER DEFINITION

(For more information on this command see Chapter 8)

Functional Description:

The PAR command loads the designated parameter identification (L number) and associated values into the parameter table.

Programming Instructions:

Parameter statements are made in the following fashion: (PAR, Lnumber = Expression "Optional Description"). Example:

(PAR,L73=3.1415 "THIS IS PI")

An Expression may contain terms and operators:

- An operator is: +,-, / (division sign), * (multiplication sign)
- A term is: (,) constant, letter address value, function
  - A constant is: ±99999999
  - A letter address value: A, B, C, E, F, I, J, K, L, P, R, S, T, U, V, W, X, Y, Z. Values used are commanded program positions as displayed on the Main Display, including offsets. (Note that I,J,K are not returned when used as rise per radian.)
  - A function is: SQ, SQRT, ABS, SIN, COS, TAN, ASN, ACS, ATN, INT, FRA, LOG, ALOG, LN, EXP.

A description is: 1 to 20 alphanumeric characters that are enclosed in quotation marks.

Example: "Here's a description"

The use of "" clears the description area of the display.

Operator precedence when parentheses are not used:

1. Positive or Negative (see Appendix F)
2. Multiplication and Division
3. Addition and Subtraction

Examples are shown in Fig. 4.2

Parameter Restrictions:

1. A maximum of 12 parameters may be assigned in one PAR command.

```
N100(PAR,L1 = 2.54"Centimeters per in.")        !Simple example

N200(PAR,L10 = SQRT(L1*L2) + (X-L2) + 4.,L2 = L3/L4*L5*(X + L2))        !Complex example.

N300(PAR,L2 = L2 + 1.,L3 = L2)        !L3 gets L2 + 1. Processed from left to right.

N400(PAR,L13 = X)        !L13 gets the current value of X as shown on the Main display including current offsets
```

Fig 4.2 PARAMETER EXAMPLES

2. In a parameter statement containing mathematical expressions, only eight *OPEN* parenthetical levels may be used.
3. The maximum number that may be used in an expression is ±99,999,999.
4. Valid parameter identification numbers range from 0 to 2999.

PRT PRINT A FILE STATEMENT

(For more information on this command see Chapter 7)

Functional Description:

The PRT command sends a closed NC or ASCII file from the mass storage device to an external printer.

Programming Instructions:

1. The required file must have been closed thru either a CLF or a G37 command.
2. The PRT command will not cause any information to be printed with respect to file name, date, time etc. unless this information has been input thru the use MSG or FWR statements into the file.
3. File access status parameter L1052 may be used to command the system to wait until after a file has been printed out to proceed with further part program processing. The part program would set L1052=0 before the print command begins the (LBL,10)(IFT,L1052=0,(GTO,10)) block. The system will set L1052 to non-zero after the file has been completely printed. This block is optional. If the block is not created in the program, the control will simply print the file.

Example:

```
N100(PAR,L1052=0)
N200(PRT,PUMP234)
N210(LBL,10)(IFT,L1052=0,(GTO,10))
```

RCP COPY A FILE TO OR FROM A REMOTE STATION

(For more information on this command see Chapter 7)

Functional Description:

The RCP command is used to pass programs or inspection files to and from the FMS cell to and from a host computer that directs the system.

Programming Instructions:

1. Files are of four types ASCII (/AS), NC ASCII (NC), binary (/BI) and object (/OB). The proper identifying abbreviations are contained in the parentheses.
2. Before a file or program may be transmitted, it must be closed out.
3. A file name may contain a maximum of 32 alphanumeric characters with no commas. Leading or trailing spaces permitted for a file that is resident in the machine control unit. Files that reside in the host computer have a similar restriction, except that commas are permitted.
4. It is required to specify the remote station machine number (e.g. KT01) and the host (e.g. HOST).

Example:
N230 (RCP,PUMP341,TO,HOST,HOUSING783/AS)

As a result of this command a file called PUMP341 will be transferred to the host computer and stored under a file name of HOUSING783. The contents of the PUMP341 file are not affected by this transfer.

5. File access status parameter L1054 may be used to allow the system to wait for the completion of the data transfer before continuing part processing.

Example:
N10 (PAR,L1054=0)

: : :

N250 (RCP,PUMP341,TO,HOST,HOUSING783/AS)

: : :

N260 (LBL,10)(IFT,L1054=0,(GTO,10))

SER STOP A PART PROGRAM EXECUTION UPON DEVICE ERROR

(For more information on this command see Chapter 7)

Functional Description:

If the control encounters any of the conditions described in the command it will stop program processing and generate an alarm message.

Programming Instructions:

The control recognizes the following error conditions: mass storage error (SER, MAS), sequential I/O error (SER, SEQ) and the command (SER, MAS, SEQ, REM) would indicate program stop on any type of error.

Example: N125(SER,SEQ,MAS,REM)

SET PROGRAMMED OUTPUT

(For more information on this command see Chapter 8)

Functional Description:

This command is used to allow the part program to interface with some external device such as a robot system or other machines in a FMS cell thru the use of an output voltage from the machine control unit.

Programming Instructions:

1. A maximum of 8 bits can be set in one statement. The outputs are set when the block goes active. Sixteen outputs are standard with the option of sixteen additional outputs available, which are arranged with 2 words of 16 bits each. The part program sets the output to one of two statuses, either 1 (the on state) or 0 (the off state).

2. To set the current output states as the power up state, use the command SET without any parameters, as shown below. This command must be done at least once after a deadload to change the system deadload default status of 0 (off state).

Examples:
N100 (SET) !Set current output states as default

Format Example:

```
Word address bit   Value to set bit
             ↓        ↓
N200(SET,O83D,0=1,O83E,3=0)
               ↑
           Bit number
```

TLD TOOL LIFE DATA STATEMENT

Functional Description:

The TLD Tool Life Data statement is used when it is desired to load the [SELECT] [N] or [DISPLAY LIST] [N] Tool Table via a part program. If it is desired to use the TLD statement, then the user is required to establish in-house rules for tool preset to specified lengths and diameters.

The format of this command is as follows;
ASCII =
(TLD,SnnTnnnnnnnCMWnnnPnnnnnDnnnnnnnnLnnnnHnn.n)
EIA =
%TLD,SnnTnnnnnnnCMWnnnPnnnnnDnnnnnnnnLnnnnHnn.n%

Where:
S = Socket Number
T = Tool Number
C = Check Tool Flag
M = Manual Tool Flag
W = Pallet Number to which Offset is to be Applied.
P = Tool Length Compensation
D = Cutter Diameter Compensation
L = Estimated Tool Life
H = Maximum Allowable Horsepower for Tool
B = Broken
E = Worn

NOTE

Use of Process Compensations is keyed by the W Flag which names a specific pallet for which the offset is to be applied.

Process Compensation length and diameter offsets are restricted to +/-9.999 Millimeters (+/-0.3936 Inches), and are added to or subtracted from the Primary Data Set of tool data which is listed on the on the same line as the tool number.

Programming Application:

1. Used when it is desired or necessary to load Tool Data via a part program.

2. Used when it is desired or necessary to load Suffix Tool (multiple surface) comps via part program.
3. Used when it is desired or necessary to load Process comps (per pallet) via part program.

Programming Instructions - Primary Data Set

1. TLD statements may be either at the beginning of a part program or in a separate program at the beginning of a pallet map.
2. Set the desired mode of measurement, Inch (G70) or Metric (G71)
3. It is recommended that each tool number appear in a separate block in order for easy identification.
4. There are three types of offsets which may be addressed by the TLD statement. The first is the "Primary Data Set" and is formatted in the following manner:
   a. Enter the control out character "(", the letters TLD and a comma.
   b. If tool is to be in a magazine, enter the letter "S" and the socket number in which the tool is to be located.
   c. Enter "T" and the number of the tool.
   d. If the first time the tool is to be used it is desired to check the tool, enter the letter "C" for check.
   e. If the tool is to be manually inserted and not located in the magazine (No "S" number) enter the letter "M" for Manual.
   f. If the Primary Data Set requires a Tool Length compensation, enter the letter "P" and the value + or - of the offset. Plus (+) is assumed unless a minus (-) is entered.

      The range of legal offsets is as follows:

      Metric Mode (G71) = ±.001 to ±999.999mm
      Inch Mode (G70) = ±.0001 to ±39.37"

   g. If the Primary Data Set requires a Cutter Diameter compensation, enter the Letter D and the value + or - of the offset. Plus (+) is assumed unless a minus (-) is entered.

      The range of legal offsets is as follows:

      Metric Mode (G71) = ±.001 to ±999.999mm
      Inch Mode (G70) = ±.0001 to ±39.37"

   h. If a maximum tool usage time limit is desired, enter an "L" and the time limit. Entry is to be a whole number up to 9959 (99 Hours and 59 Minutes). Time is read from left to right with the first two digits assumed to be hours and the 3rd and 4th digits assumed to be minutes.

      L2241 = 22 Hours and 41 Minutes

      The value of the first two digits read (left to right) cannot exceed 99 hours. The value of the third and fourth digits from the right cannot exceed 59 minutes.

   i. If a maximum horsepower limit is desired, enter the letter "H" and the value of the maximum allowable horsepower for the tool.

      Maximum allowable entry is 99.9 (decimal point may be included); however, the value entered should not exceed the rated HP of the machine tool.

Example:

N12 (TLD,S12T33412CP.02D-.25L345H5.9)

Programming - Suffix Tool Data

Suffix Tool TLD statements are written in the same manner as Primary Data Set. However, Suffix Tools are noted by a decimal number following the tool number in a range from 1 to 7, with the Primary Data Set assumed to be zero (0). The decimal number is the number called in the UAT Type II data statement. Valid numbers range from 1 to 7. Example:

N13(TLD,S12T33412.5CP.02D-.25L345H5.9)

Programming Instructions - Process Compensations

NOTE

Before a process compensation is entered either a primary or a suffix compensation must be entered.

1. Process compensations are used to indicate a particular pallet to which unique length and diameter compensations are to be applied.

2. Pallet numbers are entered in the PALLET column. The range of pallet numbers varies per the type of machine. Machines which have machine tables or uncoded pallets may use pallet numbers from 1 to 255. Machines which have coded pallets (FMS, Manufacturing Centers and

UMCs) are limited to pallet numbers in a range from 1 to 127.

3. Process Compensation Offsets are subservient to both Primary Data and Suffix Tool comps.
4. The only values which may be entered in a TLD Process compensation statement are;

   A. T = Tool Number (Suffix Tool allowed)
   B. W = Pallet Number
   C. P = Tool Length Compensations
   D. D = Cutter Diameter Compensations

The range of Process compensation values is limited to the following:

Metric Mode (G71) = ±.001 to ±9.999mm
Inch Mode (G70) = ±.0001 to ±.3937"

These values are added to or subtracted from the either the Primary Data Set or Suffix values whenever the listed pallet is on the machine table.

Example:
N14(TLD,T33412.5W12P.0003D-.0004)

UAT SELECT ALTERNATE TOOL DATA STATEMENT

Functional Description:

The UAT Statement is used to enable Suffix Tool Data, which is an alternate set of Tool Life Data used to compensate multiple surface tools. Each primary data set may have up to seven suffix tool sets created. The Suffix Tool Data Set is noted by a decimal point number directly beneath the tool number.

The UAT Type II data statement merely contains the number of the Suffix Data Set desired (Example: (UAT,1)). Other Suffix Tool Data Sets may be called replacing the current set of data.

Cancellation of Suffix Tool Data is accomplished by commanding the statement (UAT,0).

Programming Application:

Used when multiple tool surfaces are to be applied to the same workpiece.

NOTE

In many respects this functions the same as the old method of programming P and D commands. This use of Suffix Tool Data however, is discouraged.

Programming Instructions:

1. Make certain that the proper tool has been loaded into the machine spindle and that either the Primary Data Set or a Suffix Set of tool data has been enabled.
2. Command the Type II data statement UAT and the number of the desired Suffix Tool Data Set (0 to 7).

Examples:

ASCII - N123(UAT,1)
EIA - N123%UAT,1%

UCY MAXIMUM PART PROGRAM EXECUTION TIME (FMS ONLY)

Functional Description:

The UCY Type II data statement is placed at the beginning of each KT/FMS part program and is used to specify an estimated cycle time for the part program to follow. The time element of this Type II data is to be entered in seconds.

When a pallet is assigned to a machine station as a result of a Route File OPERATE command, the UCY is extracted from each part program to be executed, summed and added to the backlog element of the station. Backlog is used to determine the availability of a station for further operations.

If two or more stations are eligible to process a subsequent pallet, the station with the shortest backlog is allocated to the pallet.

The following shows the range and format of the UCY statement

ASCII = (UCY,nnnnnnn)
EIA = %UCY,nnnnnnn%
Where: nnnnnn = Estimated Average Cycle Time in seconds.

Programming Application:

To be used in ALL KT/FMS part programs.

Programming Instructions:

1. Enter the UCY statement following the Rewind Stop Code, the NAM statement if used, and the UTL statement.

Proper Sequence of KT/FMS Program Heading

a. % (REWIND STOP CODE) ------ REQUIRED
b. (NAM,Program Name) -------- OPTIONAL
c. (UTL,Tool List) ----------- OPTIONAL
d. (UCY,In Seconds) ---------- REQUIRED

2. The cycle time entered is to be an accurate estimate of average cycle time. Attempting to enter a time less than the average, or greater than the average will not be to the users advantage.

UFX CONTROL CALCULATED FIXTURE OFFSET STATEMENT

Functional Description:

The UFX statement allows the programmer to state a point from which the control calculates an incremental distance and direction from the programmed point to the actual absolute location. These calculated offsets are loaded into the Fixture Offset Table for later recall using either the Implicit Mode Fixture Offset feature (G49), or by the Explicit Fixture Offset feature (G48) with programmed H words.

The UFX feature is to be used either with the Spindle Probe or using some other device such as a dial indicator. Once the point is located the UFX statement is commanded thus calculating and loading the fixture offset values.

The format of the command is as follows:

ASCII = (UFX,HnInXnYnZnAnBnCnn)
EIA = %UFX,HnInXnYnZnAnBnCnn%

Where: H = Fixture Offset Identifier
I = Implicit Axis Position
X = Required X Axis Position
Y = Required Y Axis Position
Z = Required Z Axis Position
A = Required A Axis Position
B = Required B Axis Position
C = Required C Axis Position

Programming Application:

Used when it is desired to establish fixture offsets within a part program.

Programming Instructions:
(Manual Location)

1. Command axes to the desired program location.
2. Verify that H0 is active, ensuring that no fixture offset is in effect.
3. Execute a Program Stop command (M00).
4. Provide instructions to the operator to jog each axis until the required location is precisely found using some type of measurement device.
5. Press [CLEAR ACTIVE] disabling the Auto Jog Back feature.
6. Execute a UFX Type II data statement listing the programmed axis position.

The control calculates the incremental distance and direction from the programmed location to the actual absolute location, and enters these values onto the Fixture Offset Table in the commanded H line.

7. Continue on with the part program. When required, invoke the desired fixture offset, either Implicitly or Explicitly.

(Spindle Probe Location)

1. Command the axes to the desired program location.
2. Execute Spindle Probe routines to locate the required axis position.
3. Command H0 canceling any active fixture offset.
4. Execute a UFX Type II data statement listing the programmed axis position.

The control calculates the incremental distance and direction from the programmed location to the actual absolute location, and enters these values onto the Fixture Offset Table in the commanded H line.

6. Continue on with the part program. When required, invoke the desired fixture offset either Implicitly or Explicitly.

ULF RECALL SAVED FIXTURE OFFSET TABLE STATEMENT

Functional Description:

The ULF Type II data statement is used to load a specified or default file from the controllers hard disk unit to the Fixture Offset Table. This is done within the part program saving the operator time when loading offsets. These fixture offsets must have been previously saved using the USF Type II data statement.

The following demonstrates the command format for programming this statement within a part program.

ASCII = (ULF<,a/n>)
EIA = %ULF<,a/n>%

Where: a/n = Optional Saved Fixture Offset Table File Name - 28 characters maximum.

Note that the file to be loaded has the extension ";FXT". Whenever a Fixture Offset Table is copied from memory to the hard disk, the file to which it is copied is labeled with the extension ";FXT". This extension is not required in the ULF command requesting the Fixture Offset File be uploaded.

If no name is specified, the controller will load a saved file according to the currently active pallet map number. For example, if Pallet 38 were the active pallet and a ULF command were executed with the command being in the following format;

Example: N400(ULF)

The uploaded file would have the following name:

PALLET38;FXT

Programming Application:

This command is used when it is desired to recall a stored Fixture Offset File.

Programming Instructions:

1. If table is to cleared prior to commanding the ULF command, execute a "(ZTB,H)" Type II data statement.

NOTE

The ULF command does not clear the Fixture Offset Table, but merely writes over the contents of the current table.

2. Command the statement (ULF,File-Name). The comma and file name are optional. If no comma and file name are entered, the controller will load the corresponding "PALLETnn;FXT" file.
3. Continue the part program.

USF FIXTURE OFFSET TABLE TO BE SAVED ON MASS STORAGE

Functional Description:

The USF Type II data statement will cause the current contents of the Fixture Offset Table to be saved to the controllers hard disk unit. When written, the file may be recalled and written to the Fixture Offset Table by programming a ULF Type II data statement.

The format of this command is as follows:

ASCII = (USF,n/a)
EIA = %USF,n/a%

Where: n/a = File Name - 28 characters maximum

The file which is written to the Disk has the extension ";FXT". This name may be viewed by calling up the directory of file names using the File Handler.

If no name is specified, the controller will name the saved file according to the currently active pallet map number. For example, if Pallet 38 were the active pallet and a USF command were executed with the command being in the following format;

N200(USF)

The saved file would have the following name:

PALLET38;FXT,

Programming Application:

1. Used when it is desired to write the contents of the Fixture Offset Table to a file on the hard disk under a programmer specified name.
2. This may be used in conjunction with the UFX Type II data statements where the part is first probed to find absolute location offsets thus loaded into the fixture offset table could then be written to the hard disk for later recall.

Programming Instructions:

Command USF and, if desired, the file name when it is desired to copy the contents of the Fixture Offset Table to the hard disk unit.

This does not erase the fixture offset table, but will merely save that which is written to the hard disk.

UTL IDENTIFIES ALL TOOLS USED IN A PART PROGRAM-FMS STATEMENT

Functional Description:

The UTL statement is only to be used in part programs written for KT/FMS machine modules. This statement interacts with the Host computer controlling the FMS system during the SMI "DEFINE ROUTE" command.

When a Define Route command is executed, the Host computer will in succession open each named part program (USING) in the part program file and extract the tool numbers listed in the UTL Statement.

Whenever a Route File "OPERATE" command is executed, the Host computer will compare the Tool Table of the selected machine(s) with the tools as listed in the UTL statements. All tools which are listed in the UTL statement must be at the named machine in order for the machine to be allocated to the Pallet Route File Execution.

Example:
ASCII = (UTL,nnnnnnn,nnnnnnn,nnnnnnn,nnnnnnn,nnnnnnn)
EIA = %UTL,nnnnnnn,nnnnnnn,nnnnnnn,nnnnnnn,nnnnnnn%

Where: nnnnnnn = Tool number required for the operation.

NOTE

Tooling listed in this statement should be limited to essential tooling only.

Programming Application:

1. Used in part programs to be executed by KT/FMS machine modules only.
2. Use of this statement is optional and not a requirement for FMS operations.

Programming Instructions:

1. Insert a UTL statement in the block immediately following the Rewind Stop Code and (if used) the NAM statement.
2. If a UCY statement is to be used, the UTL statement must be in the block immediately preceding the UCY statement.

U88 PROGRAMMABLE BORING ORIENTATION CYCLE STATEMENT

Functional Description:

The U88 statement resets the offset command contained within the Fixed Cycle G88 Boring Cycle (Boring Cycle with Orientation). By default, the control assumes the offset to be +0.0100 in the X axis.

Because different types of machines with the same type of tooling keylock in different locations, the G88 default offset will not produce compatible results using the same tool, and may possibly damage the workpiece and tool.

Using a U88 Type II data statement in the block preceding a G88 Finish Bore Cycle command, the programmer can reset the distance and direction of axis offset prior to executing the G88 cycle. Any required distance and direction may be set up from .001 mm (.0001") to 42676.762 mm (1690.9320").

The following demonstrates the format and range of the U88 command.

ASCII = (U88,XnnnnnnnnYnnnnnnnn)
EIA = %U88,XnnnnnnnnYnnnnnnnn%

Where: Xnnnnnnnnn = Value of X Axis Incremental Offset prior to Z Retract in G88 Cycle.
Ynnnnnnnnn = Value of Y Axis Incremental Offset prior to Z Retract in G88 Cycle.

Default = X0.254 Y0.000 mm (X0.0100 Y0.0000")

Programming Application:

Used when incompatible boring bars are to be used on two separate machining centers.

Programming Instructions

1. Enter the Type II data statement U88 in the block preceding the G88 command where an incompatible boring bar is to be used.
2. If a compatible boring bar is to be used in the same part program following use of an incompatible tool, reset the offset using a second U88 command to the standard offset.
3. The G88 offset defaults to the standard X+0.0100 whenever [CLEAR ALL LOGIC] is pressed or whenever a tape search is performed. It is also reset by the M02 and M30 End of Program commands.

WAI WAIT FOR PROGRAMMED INPUT

Functional Description:

This command causes the control to wait for programmed inputs as specified in the statement.

Programming Instructions:

1. Interface statuses are discussed under the SET command section.
2. A maximum of 8 bits may appear in a statement.
3. If the input condition cannot be satisfied, the part program remains in a suspended state until a

[CLEAR ACTIVE] or [CLEAR ALL LOGIC] is performed upon which time program processing may be resumed.

```
                 Word address  Value to wait for
                           ↓        ↓
Example: N40(WAI,I04D,1=1, I03D,3=0)
                                ↑
                           Bit number
```

ZTB STATEMENT

Functional Description:

Zeroes the contents of tables listed in this Type II data Zero Tables Statement; may include the Tool Table (T), and the Fixture Offset Table (H).

Programming Application:

When it is desired to clear the contents of one or more of the currently active tables either before a new part processing cycle is initiated or after the part processing cycle is finished.

Programming Instructions:

1. The Zero Tables statement is written using a Type II data statement with the mnemonic (three-letter code) ZTB,. The ZTB statement may be written as a portion of the part program or may be keyed in by the operator as a Manual Data Input (MDI) command.
2. If written into the part program, the ZTB statement is programmed in a block by itself immediately preceding the end of program (M30) command.
3. The ZTB statement may call for one, or both tables to be zeroed.

NOTE

The operator can also clear the contents of the tables manually by calling them up from the [SELECT DISPLAY] menu. The T and H tables may be cleared with a single command which the operator may enter using the alphanumeric keyboard.

Example:

1. (ZTB, T, H) clears both T table and H table.
2. (ZTB, T) clears only the T table.
3. (ZTB, H) clears only the H table.

A comma is always required after the mnemonic. If both the Tool Table and the Fixture Offset Table are to be zeroed, then a comma is required both following the mnemonic and after the first table.

CHAPTER 5

PROGRAMMING MACHINE FUNCTIONS

DECIMAL POINT PROGRAMMING

The control is capable of accepting numeric input information for the following letter address words in either decimal point or trailing zero format:

A,B,C,E,I,J,K,L,R,S,T,U,V,W,X,Y,Z,F* and the numeric values for tool length cutter diameter compensations, and fixture offsets when entered into the appropriate tables.

*Metric feedrate is always a whole number so a decimal point is never required.

Both INCH and METRIC input require adherence to a specific format;

For linear axes commands: INCH = 4.4
(X,Y,Z,R,E,I,J,K) METRIC = 5.3

For indexing or rotary commands: INCH = 5.3
(A,B,C,U,V,W) METRIC = 5.3

Examples:

1. To command X to the absolute 10 inch position, the command may be written:

   G70G90X10. (using the decimal point)
   or
   G70G90X100000 (not using decimal point but trailing zero format)

The "implied" decimal is 4 digits to the left of the least significant digit in Inch mode (G70).

2. To command X to the absolute 100 millimeter position, the command may be written:

   G71G90X100. (using the decimal point)
   or
   G71G90X100000 (not using decimal point but trailing zero format)

The "implied" decimal is 3 digits to the left of the least significant digit in Metric mode (G71).

It is also permissible to "mix" decimal point and trailing zero format within the same data block, but it is highly recommended to be consistent in using one format or the other to avoid the possibility of error.

e.g. N10G90G70G94X10.Y150000Z20.F2000

Here, the X and Z commands use the decimal, the Y and F commands use the trailing zero format.

If the control reads no decimal point in an axis command, it defaults to the trailing zero format. A command Z20 where no decimal is used is interpreted by the control as Z0.0020" or 0.020 mm.

INCH / METRIC FORMAT

At the time of purchase, the user specifies the machine tool and control as being either "Basic State Inch" or "Basic State Metric". This determines the basic format of the control upon machine startup, i.e., whether inch input (G70) or metric input (G71) is in effect when the control is first turned on.

Part programs may be written using either inch or metric format, regardless of the basic state. It is important, however, to establish the correct format in the initial startup block of every part program. The control does NOT convert metric departure commands into the inch equivalent, nor does it convert inch departure commands into the metric equivalent. It simply follows the "format" of the current operating mode; inch = 4.4, metric = 5.3. For example:

1. X100000 Y50000 Z5000 (not using decimal point)

- If in G70 mode, this command is read as X 10 inch, Y 5 inch, Z.5inch
- If in G71 mode this command is read as X100.mm Y50.mm Z5.mm

2. X12.75 Y13.5 Z20. (using decimal point)

- If in G70 mode this command is read as X12.75 inches Y13.5 inches Z20. inches.
- If in G71 mode this command is read as X12.75 mm Y13.5 mm Z20.mm

PROGRAMMING THE LINEAR AXES

The linear axes are programmed by writing the axis letter address word followed by a number in the proper format. A data block may contain a programming command for a single axis or for multiaxis departures. There are other commands that affect the linear axis commands, depending upon the operation performed.

LINEAR AXIS POSITIONING MOVES

Positioning moves are generally programmed in the G0 mode and with a rapid traverse (F0) feedrate. Axes may be commanded to a specific coordinate within the programming envelope (G90) or they may be commanded to move a specified distance in an indicated direction from their current location (G91).

N50G90G0X5.Y8.Z9.5F0 (absolute)
N50G91G0X-5.Y2.2F0 (incremental)

PLUNGING TYPE MOVES (DRILLING, BORING, REAMING, TAPPING)

Plunging type operations generally involve at least one positioning move to a required coordinate followed by a linear move of the cutting axis at a calculated feedrate. These moves may be done in absolute (G90) or in incremental (G91) and generally in the G0 mode. In programming plunging type operations, the programmer may wish to use fixed cycle routines.

(ABSOLUTE)	(INCREMENTAL)
N50 G90G0X10.Y8.F0	N50 G91G0X-5.Y2.F0
N60 Z6.5	N60 Z-6.
N70 Z5.F17.	N70 Z-1.5F17.
N80 Z6.5F0	N80 Z1.5F0

CONTOURING MODES (MILLING)

Contouring operations may be programmed in either the G0 or G1 mode, depending upon whether a dwell mark in the part surface can be tolerated. Milling cuts may also be programmed in absolute (G90) or in incremental (G91) and may involve departure paths in one, two, or all three linear axes.

PROGRAMMING THE INDEX TABLE

The index table is a 360 position device intended for accurate positioning of the workpiece. Index table positioning is programmable in 1° increments. NO METAL CUTTING IS ALLOWED DURING B AXIS INDEX.

PROGRAMMING FORMAT - B INDEX

B axis index is commanded by programming letter address B followed by a number in the proper format. Trailing zeroes must be included if a decimal point is not used (e.g. B990. or B90000). A negative sign (-) between letter address and number indicates counter-clockwise rotation (e.g. B-90.).

INDEX TABLE FEEDRATE

Rapid traverse (F0) is the ONLY allowable feedrate in a data block containing a B axis index command. If the B command is programmed in a block which contains no linear axis departure commands, the control automatically applies F0 for that block, after which the previous modal feedrate is reinstated. If the B command is programmed in a block which does include linear axis departure commands the F0 command must be programmed in that block, or F0 must be modal from a previous data block. Example:

1. N290 B270. (Control applies F0 automatically regardless of modal feedrate.)
2. N75 X10.Y10.B90.F0 (index and linear combined; F0 must be programmed.)
3. N160 X10.Y10.B90. (index and linear combined; F0 modal from previous block.)

AXIS PRIORITY DURING B INDEX
(not applicable to MM1015 - See **Note 2**)

If a Z axis departure command is combined with a B axis index command in the same data block, the B axis always moves first. When B motion is complete the Z departure motion begins. In a block containing multi-linear departure commands combined with a B axis index, the X and/or Y axes move simultaneously with B. When their motion is complete, the Z departure motion begins. Example:

1. N320 B180.Z6.5F0 (B moves first, then Z)
2. N175 X10.Y10.Z8.B90. (X, Y and B move simultaneously, then Z)

NOTE

1. **It is recommended to program a B index in a block by itself to avoid the possibility of a collision.**
2. **MM1015 machining centers permit Z and B axes to move simultaneously. The feedrate must still be rapid traverse.**

DIRECTION OF B-AXIS INDEX

The B axis may be indexed in either a clockwise (positive) direction or in a counterclockwise (negative) direction upon the discretion of the programmer. The control assumes all B axis index commands as positive (i.e. clockwise) unless a negative sign is programmed between letter address word and number.

Example:

1. N260 B-180. (table indexes counterclockwise)
2. N120 B90. (table indexes clockwise)

B INDEX IN ABSOLUTE MODE (G90)

In the G90 mode each B index command is interpreted as a move to a specific position on the index table. Example:

1. N50 G90B270.
   (Table moves in CW direction to absolute 270° position. If B is at 270°, no motion occurs.)
2. N100 G90B-270.
   (Table moves in CCW direction to absolute 270° position. If B is at 270°, no motion occurs.)
3. N75 G90B0
   (Table moves in CW direction to absolute 0° position. If B is currently at 0°, no motion occurs.)

NOTE

When programming index table to 0°, no decimal point or trailing zeroes are necessary.

4. N45 G0B-0
   (Table moves CCW to absolute 0° position. If B is currently at 0°, no motion occurs.)

B INDEX IN INCREMENTAL MODE (G91)

In the G91 mode each B index command is interpreted as a move a specified number of degrees from the current position. Example:

1. N50 G91B90.
   (Table moves 90° in positive (clockwise) direction from its current location.)
2. N100 G91B-90.
   (Table moves 90° in negative (counterclockwise) direction from its current location.)
3. N75 G91B0
   (no motion occurs regardless of current B location.)
4. N45 G91B-0
   (no motion occurs regardless of current B location.)

PROGRAMMING FEEDRATES

Axes feedrates are programmed by writing the letter address F, followed by a number in the proper format. Feedrate commands are modal. Once a feedrate is programmed, it remains in effect until changed by programming a different feedrate. The programmer is responsible for calculating the best feedrate for metal removal based on part material, cutter type, spindle speed, and chip load. Feedrates may be programmed in G94 or G93 mode. The G93 (inverse time) mode is generally used only for helical or rotary machining.

Certain commands, M06 and M25, override the programmed feedrate for the one block in which they are programmed and automatically apply a rapid traverse feedrate. In the block following the M06 or M25, the previous modal feedrate is reinstated.

In a G04 (Dwell) block, the F word is interpreted as a unit of time. G04 is non-modal; the next data block restores modal feedrate.

RAPID TRAVERSE (F0)

The rapid traverse command (F0) is used for rapid positioning of the machine axes. The F0 command generates a vectored path from the current position of the axes to the end point. In a multilinear move, the rapid traverse feedrate is applied to the axis which has the longest distance to travel. The control calculates the feedrates to be applied to the other linear axes so that axes motions begin and end simultaneously. The F0 command is modal and may be programmed in any of the following modes:

G0, G01, G70, G71, G90, G91, G93, or G94.

F0 must never be programmed in the same block with a Circular Interpolation (G02 or G03) command.

PROGRAMMING 100% FEEDRATE

Programming a code M49 disables the [FEEDRATE OVERRIDE] switch on the front panel of the Control. This ensures that the programmed feedrate cannot be altered by the operator regardless of the setting on the [FEEDRATE OVERRIDE] switch - with one important exception: placing the [FEEDRATE OVERRIDE] switch at the 0 (zero) % setting causes a feedhold condition even in the M49 mode.

M49 is of particular value in tapping operations. The M49 code is applied automatically during the metal cutting portion of the G84 (tapping) fixed cycle.

PROGRAMMING SPINDLE SPEEDS AND KEYLOCK

SPINDLE SPEED PROGRAMMING

Spindle speeds are programmed in direct RPMs by writing the letter address word S followed by a number which corresponds to the desired RPM. The upper and lower limits of the S word command vary per machine type and options. The control automatically selects the proper gear range and/or pole for the commanded spindle speed. S words are modal functions; therefore, the programmed spindle speeds remain in effect until a new speed is programmed.

Spindle speed and / or direction can be changed while the spindle is stopped, or while the spindle is rotating.

NOTE

If spindle speed change requires gear range and/or pole change the spindle decelerates to creep speed before accelerating to new spindle speed.

In addition to the S word, the appropriate miscellaneous function (M03 or M04) must be programmed to initiate spindle rotation. These commands need not be programmed in the same data block. As an example, the S word may be programmed in the restart block following a tool change, and the Spindle ON command programmed in a succeeding data block. (See Fig 5.1)

Spindle rotation is halted by programming M05, M06, M19, M00, M01 (when enabled), M30, or M60. None of these commands cancel the modal spindle speed.

SPINDLE TAPPING SPEEDS

(Instructions for programming tapping operations are in Chapter 5.)

A separate set of two spindle speed ranges is available for programming tapping operations, except on MH-12 machining centers. The tapping range is selected by programming an M40 in the same block with the tapping spindle speed. The control automatically selects the proper range for the programmed S word.

PROGRAMMING 100% SPINDLE SPEED

Programming a code M49 disables the [SPINDLE SPEED OVERRIDE] switch on the front panel of the control. This ensures that the programmed spindle speed cannot be altered by the operator regardless of the setting on the [SPINDLE SPEED OVERRIDE] switch. This function is of particular value in tapping operations. The M49 code is applied automatically during the metal cutting portion of the G84 (tapping) fixed cycle.

KEYLOCK

A keylock of the spindle is programmed by writing the miscellaneous function M19. This function cancels spindle rotation (M03 or M04), interrupts the flow of coolant, and positions the spindle drive key in a specific radial orientation. Keylock positions vary depending upon machine type and tool holder type (straight or tapered shank). Keylock position is always considered 0°.

The M19 command may be programmed in a block by itself or with axis departure commands. M19 is an immediate function; if programmed in a block with axis departure commands, the keylock orientation begins simultaneously with axis motion, but keylock must be complete before block transfer can occur. Following M19, a Spindle ON command (M03 or M04) must be programmed before spindle rotation can resume. The keylock command is done automatically in the following circumstances:

1. During automatic tool transfer (M06) on machine tools equipped with tapered shank tool holders.

```
N160 M06                              ! (tool change)
N165 G0G70G90G94X10.Y10.Z20.F0T123S490 ! (restart, position, establish S word)
N170 Z8.1                             ! (position)
N175 Z7.25M03F17                      ! (spindle on, plunge)
N180 Z8.1M05                          ! (feed out, then spindle stop)
```

Fig 5.1

2. At programmed Z depth of a G88 (Boring) fixed cycle routine. No reprogramming of SPINDLE ON command is necessary after keylock during G88 cycle.

PROGRAMMABLE KEYLOCK

On machine tools equipped with the programmable keylock feature, the spindle drive key may be keylocked in 1° increments from the normal keylock position (0°). The programmable keylock feature is commanded by writing the miscellaneous function M19 followed by letter address word S followed by a positive number which represents degrees orientation in the clockwise direction from 0° (viewed looking out of spindle).

The range of this number is from 1 to 359 in increments of 1°. No decimal point or trailing zeroes are required after this number (e.g. M19S1, M19S2, M19S250 etc.).

If an M19S command is programmed while the spindle is rotating, that keylock position is obtained in the direction of the spindle rotation. If an M19S command is programmed while the spindle is stationary, the spindle rotates in the direction which affords the shortest rotary distance to the commanded keylock position. Example:

1. N260 M19S96 (keylocks 96° in CW direction from 0°.)
2. N150 Z8. M19S275 (keylocks 275° in CW direction from 0°, positions Z.)

BLOCK DELETE (/) PROGRAMMING

Functional Description:

Used to bypass certain portions of a part program. The Block Delete Slash code is used in conjunction with the [ENABLE BLOCK DELETE] switch on the front panel of the control. When [BLOCK DELETE] is enabled, the control recognizes the slash code at the beginning of the block and bypasses that block of information. When [BLOCK DELETE] is disabled, the control ignores the slash code and executes that block of information.

Programming Application:

1. Useful whenever a trial cut procedure is required (see example below).
2. May be utilized as a part of an axis inversion (mirror image) procedure.
3. Can be used in the manufacture of two different parts which are similar but not identical.

Programming Instructions:

The Block Delete Slash code must be the first entry in the data block, preceding the sequence number.

The last axis positions programmed in the block delete sequence must be identical to the axis positions prior to entering the block delete sequence. Example:

```
N100 M06
N110 G00G90G94Z(home)M25T1000
N120 XnYnB180.F0
N130 ZnS1500M03      ! Z Clearance
/N140 ZnF30.         ! Trial Cut
/N150 Zn             ! Z Clearance
/N160 Z(home)M25
/N170 B225.F0M00
/N180 (MSG,CHECK BORE SIZE)
/N190 G00G90G94Z(home)M25
/N200 XnYnB-180.F0
/N210 ZnS1500M03     ! Z Clearance
/N220 ZnF30.         ! Final Depth
/N230 etc.
```

PROGRAMMING COOLANT FUNCTIONS

Coolant commands are programmed by writing the appropriate Miscellaneous function. The Coolant On commands are M07-Mist Coolant On and M08-Flood Coolant On. Only one Coolant On command can be in effect at a time. M07 and M08 are immediate functions. Coolant will only flow when the spindle is rotating and the coolant control switch must be on.

Coolant On commands are interrupted, but not canceled, by the following commands which halt spindle rotation:

M05 - SPINDLE OFF
M06 - TOOL TRANSFER
M19 - KEYLOCK

Following these commands, coolant flow is resumed upon spindle start-up.

Coolant On commands are canceled by the following:

M00 - PROGRAM STOP
M01 - OPTION STOP (when enabled)
M02 - END OF PROGRAM
M09 - COOLANT OFF

M30 - END OF PROGRAM, TAPE REWIND, PALLET SHUTTLE
M60 - PALLET SHUTTLE WITHIN PROGRAM

After one of these commands, coolant flow is resumed by programming both Spindle On and Coolant On commands. Example:

N50M06 ! (Tool change)
N55G0G90G94X10.Y8.Z20.F0M08S550 ! (Restart, establish spindle speed and flood coolant)
N60Z6.1 ! (Position Z)
N65G01Z5.8M03F10. ! (Spindle ON, coolant flow begins then Z to depth at feed)
N70X12.Y12.Z5.3M09 ! (Contour, coolant flow canceled at end point)

PROGRAMMING AUXILIARY COOLANT (JET PULSER UNIT)

The JET PULSER unit provides for coolant-through-the-tool (straight shank tool holders) or coolant-through-the-spindle (tapered shank tool holders). The Auxiliary Coolant commands are programmed by writing the appropriate Miscellaneous function:

M57 - Auxiliary Mist Coolant On
M58 - Auxiliary Flood Coolant On
M59 - Auxiliary High Pressure Pulsating Coolant On

Only one of these Coolant On commands can be in effect at a time; however, any one of the auxiliary coolants may be programmed in combination with either M07 or M08. Example:

N150 Z7.5F11.M07M58 (Mist Coolant and Auxiliary Flood)
N275 Z4.9F9.M08M59 (Flood Coolant and Auxiliary High Pressure Pulsating Coolant)

The auxiliary coolant commands are governed by the same rules applying to the M07 and M08 commands.

Wash House Programming

The entire work area of the machine may be enclosed by an optional Coolant Enclosure. Doors to the enclosure are automatically opened and closed during pallet shuttle, allowing pallets to be moved between the machine table and the pallet shuttle unit.

Adjacent to the machine table is a coolant manifold, which has a series of nozzles directed towards the workpiece/fixture. At the end of each machining cycle, the programmer may command the machine table to rotate several times, while at the same time flooding the part with high pressure, high volume coolant. This cleans chips from the workpiece and fixture.

A second manifold is also provided which allows the programmer to again rotate the workpiece/fixture, while blasting it with high pressure air. This will blow away excess coolant from the workpiece and fixture.

This "Wash and Dry" operation returns a clean, dry workpiece / fixture to a Load / Unload station.

The following commands control the functions of the manifolds within the Wash House Enclosure.

1. M51 - Wash House Flood Coolant (Modal - Immediate)

When commanded, M51 will cause the Pallet/Fixture to be washed with continuous streams of coolant. When commanded on, coolant is forced out of the manifold to one side of the machine table. Once on, the programmer should command one or more B Axis rotations to ensure that coolant is spread evenly over the Pallet/Fixture.

M09 (Coolant Off) will cancel the M51 Command.

2. M52 - Wash House Air Blast (Modal - Immediate)

When commanded, M52 will cause the Pallet/Fixture to be blown by continuous jets of air. When commanded on, high pressure air is forced out of the manifold to one side of the machine table. Once on, the programmer should command one or more B Axis rotations to ensure that the air is spread evenly over the Pallet/Fixture to dry the part as much as possible.

M09 (Coolant Off) will cancel the M52 Command.

Wash House Sample Program

N123 X15.Y11.5Z18.F0M5M25P0H0
N124 B0
N125 M51B180.
N126 B0
N127 B180.
N128 B0M9
N129 M52B180.
N130 B0
N131 B180.
N132 B0M9
N133 X30.Y11.5Z18.F0
N134 M30

PROGRAMMING TOOL SELECT

Tool select commands are programmed by writing the letter address T, followed by a number which conforms to the tool select system on the machine. This select system may be octal, binary, or decimal. All KT machine tools are equipped with Random Tool Select.

The Tool Select command initiates rotation of the tool storage magazine to search for the commanded tool number. Completion of the tool select command is not a requirement for completion of the data block in which the T word is commanded. Tool storage magazine rotation continues through as many succeeding blocks as required. For this reason, it is recommended to write the tool select command in the restart block following transfer of the preceding tool.

The tool select command must be entered only once for each tool. If a tool transfer command (M06) is read before the tool select sequence is complete, the transfer cycle waits until the selected tool is located before initiating the tool transfer operation. Example:

```
N250 M06                          ! (tool transfer)
N260 G0G90G94X10.Z20.F0S420M08T2225
          !(select tool in advance for next tool transfer)
```

NOTE

MH-12 and Milwaukee Series tool select commands may be programmed in the block containing the M06 command used to insert the selected tool.

OCTAL TOOL CODING SYSTEM (RANDOM SELECT)

This system consists of a 5-digit number addressed by the letter T, ranging from T00001 to T77777 and provides a total of 32,767 tool codes. To identify a tool, various combinations of code rings and spacers are assembled on the coding collar of each tool holder. These code rings and spacers are the same thickness, but the outside diameter of the code ring is larger than the outside diameter of the spacer. This difference is used to identify the tool at the tool reading head. Each tool holder must accommodate a total of 15 code rings and spacers to form a binary coded octal number. The octal number can be determined by using five groups of three binary digits (4, 2, 1). This octal number must be read starting from the shank end of the tool.

NOTE

The digits 8 and 9 cannot be configured and must not appear in the tool select command.

BINARY TOOL CODING SYSTEM (RANDOM SELECT)

The Binary Tool Code System consists of a four-digit number addressed by the letter T. The range of the tool code is T0101 to a maximum entry of T-3131 for a total of 961 (31 x 31) tool codes.

The first two digits of a tool code form the group number; the last two digits represent the tool number. The tool code number consists of ten rings and spacers in various combinations that are assembled on the coding collar of each tool holder. The thickness of the rings and spacers is identical, but the outside diameter of the rings is greater than the outside diameter of the spacers. (See Fig. 5.2)

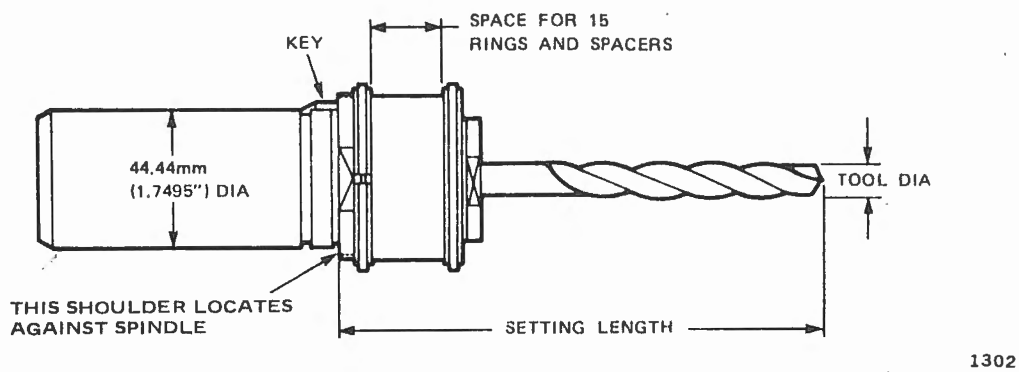
Starting from the shank end of the tool, the first five places for rings and spacers identify the group number and the last 5 places for rings and spacers identify the tool number. A numerical value of 1 is assigned to the first space, 2 to the second space, 4 to the third space, 8 to the fourth space, 16 to the fifth space. The 2 digit group number, then, is the sum of the numerical values of the first five spaces where the rings are placed. A spacer indicates the absence of the numerical value from the sum. If all five spaces have a ring, the numerical sum is 31; if only the first space has a ring, the numerical sum is 1. Thus, any sum between 1 and 31 can be represented by combinations of rings and spacers. The 2 digit tool number is determined in the same method. Therefore, the Binary Tool Code number is the combination of the group number and the tool number (see Figure 5.2).

DECIMAL TOOL CODING SYSTEM (RANDOM SELECT)

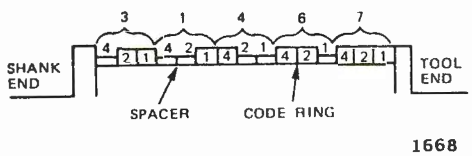
The Decimal Tool Coding system consists of a 1 to 7 digit number addressed by the letter T. The range of the tool code is 0000001 to 9999999 (00000 01 to 99999 99 with tool serialization, used with laser coding). The Decimal system is used with tapered shank tool holders and with straight shank tool holders if no tool code reader is present. The programmer may wish to write a Table Load command utilizing a Type II data statement. Doing so reduces operator set-up time and lessens the possibility of error inherent in manually keying in numbers into the Tool Table.

LASER BAR CODING (OPTIONAL)

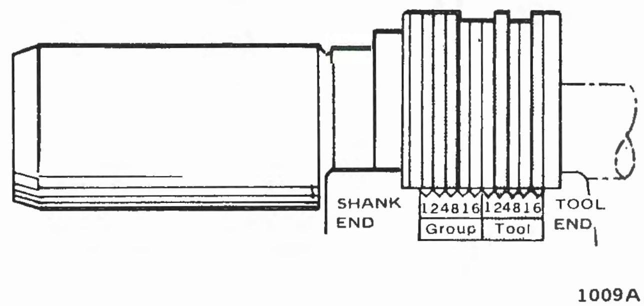
(Progressive Selection System)



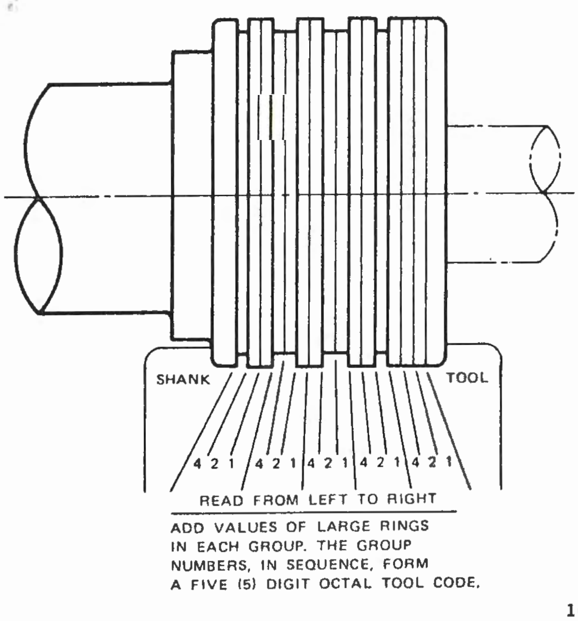
Typical MILWAUKEE-MATICE Tool Holder



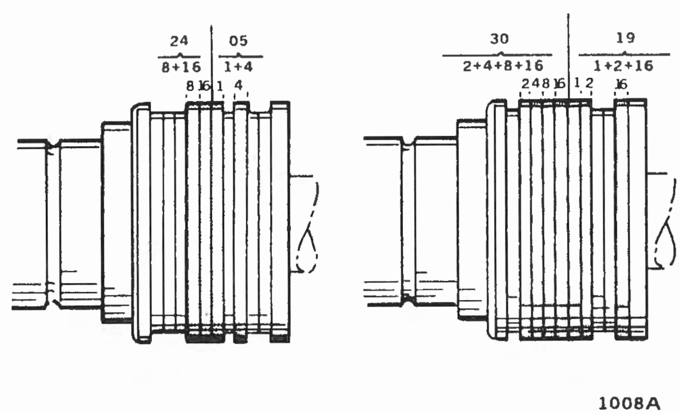
Octal Tool Coding Example



Example of Tool Coding (Binary)



Reading the Tool Code (Octal)



Reading Tool Codes (Binary)

FIG. 5.2 - TOOL CODING

The Progressive selection system puts tools removed from the spindle into the socket from which the tool currently in the spindle was removed. The net effect is to advance selected tools each time the tool is called by the control. This system will ensure that a socket is always available for tooling removed from the spindle. The operator may place additional tooling in any open socket, except the socket at the tool change position.

CAUTION

It is important that the positions of oversize tooling in the magazine be carefully considered before such tools are loaded into the system.

While there is always an open socket for tools to be returned to the magazine, oversize tooling requires that adjacent sockets also be empty. Because tools advance one socket when returned to the magazine the position of this socket is not always predictable.

When oversize tooling is used, it is the responsibility of the programmer to specify the positioning of all tools in the magazine, or to use the PROGRESSIVE TOOL SELECTION WITH STATIONARY-OVERSIZE TOOL FEATURE in which the socket from which the oversize tool was selected is reselected as the socket to which the tool is to be returned. If this function is used, both the programmer and the operator must insure that the non-oversized tooling must be capable of changing sockets without colliding with either oversize or non-oversize tooling.

The bar coding system is used to identify tapered shank toolholders. When an initial tool select command is executed, the controller will cause the tool magazine to rotate counterclockwise until the socket containing the commanded tool is identified. The tool number is then entered into the Tool Table ([SELECT] [N]) automatically. Once the position and number of all tools in the magazine is noted, rotation of the tool magazine is bi-directional. The commanded tool number is moved to the magazine tool change position by rotating the magazine in the shortest clockwise or counterclockwise direction. Decimal numbers (0 to 9) in a range from 1 to 9999999 or 99999 99 (tool serialization) are used with laser coding. The type of software executive system that a machine tool uses will determine if a machine uses either decimal coding or tool serialization.

Bar numbers range from 0 through 9 and are indicated by the position of each code bar. Refer to KT Publication 939, Lasereader Operating and Maintenance Manual for further information.

PROGRESSIVE TOOL SELECTION WITH OVERSIZE TOOL CAPABILITY

This feature applies to machine systems with either decimal tool coding or lasereader tool systems. It requires either the programmer or the operator to load the status column of the tool table with the proper configuring code to indicate to the control the location of the oversize tool and adjacent empty sockets.

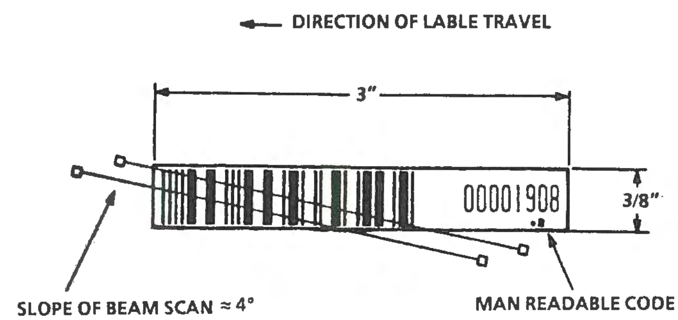


Fig. 5.3 EXAMPLE OF BAR CODE

Both the programmer and the operator must be aware that the progressive tool selection system on the Orion 2300 operates differently from the progressive tool selection system described in Publications 929 and 883B. The system used on the Orion 2300 uses the status column to indicate the number and the location of empty sockets needed to handle oversize tooling. This feature makes it unnecessary to perform dummy tool changes. The programmer or operator may make use of this feature thru the TLD Type II data statement by entering the proper code in the status portion of the statement. The operator may also use the [SELECT] [N] (Tool Table) display to enter this data.

By entering the following configuration codes into the status column the control will maintain the proper number of empty sockets adjacent to the oversize tool. The letter X is used to indicate the presence of an oversized tool. 0 indicates an empty socket.

Configuration Codes:

O1	0X0
O2	0X
O3	X0
O4	00X00
O5	00X
O6	X00
O7	LONG TOOL

Example: Code O1

Socket Numbers	24	25	26
Socket status	empty	tool	empty

This could be entered into the status column by the expression O1 by the operator on the line in the tool table which is associated with socket 25. Care should be used to enter the letter O and not the number 0.

The following events will occur on an Orion 2300 assuming that the MCP system has just been loaded.

1. All the required tools have been loaded into the tool magazine. (All the sockets may be filled at this point.)
2. The operator updates the tool table and enters the oversize status for any tools in the system.
3. A part program is assigned and begins to run.
4. After the first tool change is called the operator will see the following message:

   1023. 1ST TOOL CHANGE: REMOVE SPINDLE TOOL.

   The operator removes the tool in the spindle manually. This tool **IS NOT** to be replaced in the tool magazine, if the tool system has been filled to its capacity. If empty sockets exist, it may be placed at some other location in the tool storage magazine and the socket location in the tool table updated by the operator. This event described above will occur only once after the MCP system has been loaded and it will occur only for the first program that is commanded to run after the MCP is loaded.

5. The first time that a non-oversized tool is loaded into the spindle, its socket is reserved as empty. The Machine Status [SELECT] [V] will display the following message:

   SOCKET 4 RESERVED AS "EMPTY SOCKET"

6. When a over-sized tool is loaded into the spindle, and a non-oversized tool is returned to the tool magazine, the non-oversized tool is placed into the "EMPTY SOCKET" and the Machine Status [SELECT] [V] will read:

   SOCKET 0 RESERVED AS "EMPTY SOCKET"

   With this 0 display, the control is indicating that no socket has been reserved as an empty socket for this tool and it will be returned to the empty socket from which it came.
7. When a non-oversized tool is loaded into the spindle and a oversized tool is returned to the tool magazine, it will be returned to the socket from which it was removed. The socket from which the non-oversized tool was removed will be made the new "EMPTY SOCKET".

   SOCKET 14 RESERVED AS "EMPTY SOCKET"

The net result of these operations would be for the positions of non-oversized tools to advance as a result of the tool change, while that of the oversized tools and the associated sockets would remain fixed.

COMPUTERIZED DECIMAL CODING
(Stationary Selection System)

With this mode of operation, a selected tool is removed from a socket is returned to the same socket from which it was removed. If oversize tooling is used in the system, it is always returned to the same socket from which it came. Once again, adjacent sockets must be empty to assure clearance between the oversize tools and neighboring tools in the storage system.

Since there is no physical identification of tapered toolholders, these toolholders use a decimal coding scheme. Decimal numbers (0 to 9) in a range from 1 to 9999999 or 99999 99 are assigned by the programmer or operator to each tool to be used in the part program. Tools can then be loaded into the tool magazine in any desired order, with the operator recording the number of each tool in each tool socket.

Once all of the tools are loaded into the tool storage magazine, it is necessary to associate the "Socket Numbers" with the "Tool Numbers". These entries appear on the Tool Management Table [SELECT] [N] and may be entered in either of two methods.

The most common method in which tool numbers are entered on the Tool Management Table is to have the operator manually enter the tool numbers behind the appropriate socket numbers on the Tool Management Table. Refer to KT Publication 883B CONTROL OPERATING MANUAL for KT GEMINI-D Machine control System, for further details regarding loading the tool table.

AUTO TOOL GENERATE (KT/LASEREADER)

Tool generate is used to locate tools previously loaded into the machine's tool storage magazine. This function is initiated using the [SELECT] [L] miscellaneous tables.

When commanded, the TOOL GENERATE WITH DELETE command will cause the tool storage magazine to rotate to the first socket position. The tool storage magazine then makes one complete revolution. As this occurs, all contents of the current tool table are deleted. It is replaced with a list of all tool numbers and the respective sockets in which these tools are located. In the case where alternate tools are used, the operator may manually update these socket locations. Only the first tool number located is assigned a socket position due to the fact that the system is capable of making a determination as to which is the primary tool and which tools are alternates thru the use of tool serialization numbers.

The following is the entry sequence for this command:

Select [MANUAL] mode.
Press [SELECT] [L].
Press [8], [1], [,], [1].
Press [CYCLE START].

[SELECT DISPLAY] [N] TOOL TABLE

Pressing [SELECT] [N] will display the Tool Management System Table, a comprehensive display showing all tool related data on a single display (see Figure 5.4). Length comps associated with tooling are applied as soon as a tool is loaded into the spindle. This table provides the following special functions:

- Alternate tool numbers
- Flags for broken, check, data transfer, manual, U-Head and worn tools
- Process (per pallet) compensations
- Length compensation data
- Diameter compensation data
- Measured cutting time
- Maximum time limit for a tool
- Maximum horsepower limit for a tool

ENTER: _

TOOL NO.	STATUS (BCDMUW)	SOCKET NO.	PALLET NO.	LENGTH COMP.	DIAMETER COMP.	CUTTING TIME	TIME LIMIT	H.P. LIMIT
16	MW					3:23	3:20	
42	M			1.5000	.0025	3:02	3:30	
2431		3				1:15	2:00	
			16	(.4471)	(.0500)			
2951	C 01	7					3:20	
			81	(.0100)	(.0002)			
4421		16		2.5000	.0005	1:16	2:25	1.6
4421		17		.0305				
. 1					.0005	0:15	4:20	6.0
11113		14				1:01	3:20	2.8
11113	W	20				3:38	3:20	
24113	01	28		.0001			3.50	
			126	(.0010)				
			**	(.0005)				
55543						2:18	3:20	

ADD RECORD	LOCATE	SPINDLE	TABLE	ENTRY	EARLY	ADVANCE
E, TOOL	LT, TOOL	S, TOOL	C,0 TABLE	# DATA	WARNING	<
#/* PALLET	LS, SOCKET	S,0 CLEAR	SL, LIST	0 CLEAR	W, 100%	NONE

Fig. 5.4. TOOL MANAGEMENT SYSTEM DISPLAY

PROGRAMMING AN AUTOMATIC TOOL TRANSFER

An automatic tool transfer is programmed by writing an M06. The control will automatically activate specific M and G codes and move the involved axes to the tool change position at rapid traverse.

CAUTION

It is the programmer's responsibility when writing part programs, and the operator's when commanding the machine during MDI operations, to ensure that there is no possibility of collision during automatic tool change. All axis not involved in the tool change cycle should be safely positioned prior to initiating an Automatic Tool Change.

Items to be taken into consideration include:

- The tool(s) being changed
- Tool changer housing and arms
- The pallet
- A Spindle Probe / Broken Tool Detector Post or ...Multipoint Tool Sensor
- The fixture
- The workpiece

A tool select command (T word) must be programmed in order to execute a tool change command. It is recommended to program M06 in a data block by itself.

STRAIGHT SHANK TOOL TRANSFER

Upon execution of the M06 command the spindle is forced to stop. The spindle drive key assumes any random orientation as a result of the spindle stop. If a tool holder is equipped with a key, an M19 (keylock) must be programmed for transferring the tool both into and out of the spindle. The keylock command may be programmed in the same block with the M06 code or in the previous block. A keylock should always be programmed in the initial setup block of a new part program in case a keyed tool holder remains in the spindle from a previous part program. (See Fig.5.5)

Tool holders not employing a key do not require an M19 for transfer into or out of the spindle. The programmer, however, may wish to "play it safe" by including an M19 with each tool transfer command. This procedure, while ensuring safety, increases the time required for each tool transfer cycle. Following an M19 command, it is necessary to program a spindle speed when the spindle is started.

TAPERED SHANK TOOL TRANSFER

Upon execution of the M06 command, the spindle is stopped in the keylock position to ensure correct alignment of tool holder in the spindle. (See Fig 5.6)

PROGRAMMING AN AUTOMATIC PALLET SHUTTLE

The pallet shuttle command, on machine tools so equipped, is programmed by writing an M30, which also ends the part program and rewinds the tape, or by writing the Miscellaneous function M60, which causes a shuttle within the part program. The shuttle command exchanges the pallet on the table with the pallet on the shuttle station. The part programmer must enter commands which position each axis to either the required shuttle position or to a clearance position prior to the shuttle command. The machine operator must also enable the proper switches in order to execute a shuttle cycle.

```
N10G0G70G90G94X10.Y10.X20.F0M19T12       (initial set-up block including keylock command)
 : : :
 : : :
N75 G0Z20.M25 .............................. (retract Z for tool change clearance)
N80 M19M06 ...................................... (keylock spindle, tool transfer)
N85G0G70G90G94X14.Y9.Z20.F0S1900T22226 ............ (restart block, select next tool)
 : : :
 : : :
N110 X10.Z20.F0M19 ........................ (axes to clearance, keylock spindle)
N120 M06 ......................................... (tool transfer in block by itself)
N130G0G70G90G94G48X14.Y9.Z20.F0S2400tT225 ........... (restart block, select next tool)
```

Fig.5.5 STRAIGHT SHANK TOOL TRANSFER SEQUENCE

ALL MACHINES EXCEPT MH-12 AND MILWAUKEE SERIES

N35 G0G48G90G94X14.Y9.Z20.S1900M3M35T54321 .... *(Restart Block)*
: : :
(Program Operations)
: : :
N75 G0Z20.M25 .... *(Retract for Tool Change)*
N80 M06 .... *(Insert Tool 54321)*
N85 G0G48G90G94X14.Y9.Z20.S1900M3M25T12345 .... *(Restart Block)*

MH-12 AND MILWAUKEE SERIES ONLY

N35 G0G48G90G94X14.Y9.Z20.S1900M3M35 .... *(Restart Block)*
: : :
N75 G0Z20.M25 .... *(Retract for Tool Change)*
N80 M06T54321 .... *(Insert Tool 54321)*
N85 G0G48G90G94X14.Y9.Z20.S1900M3M25 .... *(Restart Block)*

Fig. 5.6 TAPERED SHANK TOOL TRANSFER SEQUENCE

PALLET SHUTTLE USING M30

The M30 command ends the part program and rewinds the tape (or resets the disk) in addition to performing the shuttle cycle. The M30 command must be programmed in a block by itself. (See Fig. 5.7)

NOTE

Refer to Machine Supplement Manual for shuttle positions.

PALLET SHUTTLE USING M60

The M60 command performs the shuttle cycle within a part program. No end of program or tape rewind (or reset of the disk) occurs as a result of the M60 command. The M60 command must be programmed in a block by itself. The data block following the M60 command should be a restart block containing the next highest sequence number and axes commands which match those already in effect, so that no axis motion is generated. (See Fig. 5.8)

NOTE

Refer to Machine Supplement Manual for shuttle positions.

PROGRAM TAPPING OPERATIONS

Tapping operations require the calculation of the proper spindle speed to feedrate ratio for the lead (number of threads per inch or threads per millimeter) of the tap. This ratio can be determined by using one of the following formulas:

INCH MODE FORMULAS

$$\text{Threads per Inch} = \frac{\text{Spindle Speed}}{\text{Feedrate}}$$

$$\text{Feedrate} = \frac{\text{Spindle Speed}}{\text{Threads per Inch}}$$

$$\text{Spindle Speed} = \text{Threads per Inch} \times \text{Feedrate}$$

$$\text{Pitch} = \frac{\text{Feedrate}}{\text{Spindle Speed}}$$

$$\text{Spindle Speed} = \frac{\text{Feedrate}}{\text{Pitch}}$$

$$\text{Feedrate} = \text{spindle speed} \times \text{pitch}$$

Spindle speed for tapping operations is selected from the tapping spindle speed ranges. This is programmed by writing the desired spindle speed and the Miscellaneous function M40 (M40 is not applicable to MH-12) in the same data block. The programmed values for both spindle speed and feedrate are maintained at 100% by programming the Miscellaneous function M49. This code disables the

```
N850 X30.Y11.5Z20.F0P0H0 ........ (linear axes to clearance or shuttle position)
N860 B0 ........................ (index or rotary axis to shuttle position)
N870 M30 ....................... (shuttle command in block by itself)
```

Fig. 5.7

```
N975 X30.Y11.5Z20.F0P0 ............ (Linear axes to shuttle position.)
N980 B0 ........................... (Index or rotary axis to shuttle position.)
N985 M60 .......................... (Shuttle command in block by itself.)
N990 G0G70G90G94X30.Y11.5Z20. ..... (N word continues in sequence, axes positions
                                    identical to those programmed in N975 and N980.)
N995 B0 ........................... (Continuation of part program.)
```

Fig. 5.8

[SPINDLE SPEED OVERRIDE] and [FEEDRATE OVERRIDE] selectors on the control panel. Example:

1. Tap with 3/4-10 at 200 RPM. To calculate Feedrate:

$$\text{Feedrate} = \frac{\text{Spindle Speed}}{\text{Threads per Inch}} = \frac{200}{10} = 20 \text{ IPM}$$

Entered as F200 (no decimal) or F20. (with decimal).

Spindle speed selected from tapping range as in N130, below in Figure 5.9A.

2. Tap with M20-2.5 at 200 RPM. To calculate Feedrate:

Feedrate = RPM x pitch
= 200 x 2.5
= 500 MMPM

Spindle speed selected from tapping range as in N130 below, in Figure 5.9B.

No external devices are required for performing tapping operations; however, special tapping tool holders with built-in axial float allowance are required. The axial float in these holders is generally ±0.250", which compensates for minor spindle speed and Z axis feedrate deviations for the desired tap lead.

```
N100 M06 ........................ (tool change - tap put into spindle)
N110 G0G70G90G94Z20.M25T123 ..... (restart block)
N120 X10.Y12.B-270.F0 ........... (position X, Y and B)
N130 Z8.1S200M40M03M08 .......... (Z to clearance, select tapping range, spindle on CW)
N140 Z7.25F20.M05M49 ............ (Z to depth at 100% F.R. and S.S., spindle stop)
N150 Z8.1M04 .................... (spindle reverse, Z to clearance at 100% F.R. and S.S.)
N160 X15.Y9.F0M03M48 ............ (spindle reverse - M40 in effect, position X and Y, enable override)
```

Fig. 5.9A TAPPING IN G70

```
N100 M06 ........................................ (tool change - tap put into spindle)
N110 G0G71G90G94Z508.M25T123 ............................... (restart block)
N120 X154.Y305.B-270.F0 ..................................... (position X, Y, and B)
N130 Z206.S200M40M03M08 ........... (Z to clearance, select tapping range, spindle CW)
N140 Z184.F500M05M49 ............... (Z to depth at 100% F.R. and S.S., spindle stop)
N150 Z206.M04 ................. (spindle reverse, Z to clearance at 100% F.R. and S.S.)
N160 X381.Y229.F0M48 ..... (spindle reverse-M40 in effect, position X and Y, enable override)
```

Fig. 5.9B TAPPING IN G71

Tapping operations may also be programmed using the FIXED CYCLE TAPPING sequence (G84).

PROGRAMMING FIXED CYCLES

Fixed Cycles (or "Canned Cycles") are programmed by writing the appropriate Preparatory function (G81-G89) along with the necessary axis departure and clearance information. Fixed cycles are canceled by writing Preparatory function G80 in a data block by itself. The fixed cycles are preprogrammed routines which simplify the programming of several plunging type operations as well as reducing tape length.

The fixed cycles are programmed by writing a setup block followed by positioning blocks which repeat the programmed routine at the required coordinates with the following restrictions:

1. Fixed cycles are programmable only in the XY plane (G17).
2. No "after" type Miscellaneous functions may be programmed in any fixed cycle block (e.g. M05, M09, etc.).

FIXED CYCLE SETUP BLOCK

In the fixed cycle set up block the programmer establishes the following modes of operation:

- G0 - position mode
- G90 - absolute positioning
- G94 - direct insertion of feedrate

(These codes are modal upon start-up, and may already be in effect from previous operations. It is still recommended, though not required, that they be repeated in the fixed cycle set up block).

- G81 - drilling
- G82 - spot facing
- G83 - chipbreaking
- G84 - tapping
- G85 - reaming
- G86 - boring
- G87 - deep hole drilling
- G88 - boring (keylock)
- G89 - feed in-dwell-feed out

- X-axis coordinate position for first operation
- Y-axis coordinate position for first operation
- Z-depth at coordinate location
- R-Z axis clearance coordinate at hold location
- E-incremental feed distance for G83 ad G87 cycles

The Z, R, and E words remain modal throughout the fixed cycle routine, unless changed by programming a new value. Example:

```
N120G0G90G94G81X10.Y11.Z6.25R8.1S350M03F4.8
    :   :   :   :
N175G0G90G94G87X14.Y8.6Z6.225R8.4E.515S290M03F5.
```

```
N120 G0G90G94G81X10.Y11.Z7.25R8.1S350M03F4.8 ............ (Drill hole at X10.Y11.)
N130 X12. ............................................... (Drill hole at X12.Y11.)
N140 Y13. ............................................... (Drill hole at X12.Y13.)
N150 X15.Y16. ........................................... (Drill hole at X15.Y16.)
N160 G80 ..................................... (Cancel fixed cycle in a block by itself)
```

Fig. 5.10

FIXED CYCLE POSITIONING BLOCKS

After having established the set up block the programmer needs only to program new coordinates in the succeeding data blocks. The fixed cycle routine is repeated automatically at each new position programmed. See Fig. 5.10.

SYSTEM OPERATION DURING FIXED CYCLE ROUTINE

During a fixed cycle block, the control directs the axis slides to move at rapid traverse between programmed coordinate locations, and to the programmed clearance plane (R word). Axis priority is established automatically by the control using the following guidelines:

1. In the initial set up block the control reads the programmed R word and compares it to the current location of the spindle in the Z axis.

   a. If the R word coordinate is negative with respect to current Z location (i.e. Z axis must move closer to pallet centerline to satisfy R), then the X and / or Y axis is positioned first (at F0) followed by the Z move (also at F0) to the programmed R coordinate. The Z axis then feeds to depth as per Z entry in set up block and returns to R coordinate. Example:

N100 G0G90G94Z20.F0
(Z at retract position)

N110 G81X10.Y12.Z6.R8.F10.S300M03
(X and Y axes rapid first, then Z rapids to programmed R location, feeds to Z depth, returns to R location).

   b. If the R word is positive with respect to the current Z location (i.e. Z axis must move further away from pallet centerline to satisfy R) then the Z axis is positioned first (at F0) to the programmed R coordinate followed by the X and/or Y axis move (also at F0). The Z axis then feeds to depth as per Z entry in set up block and returns to R coordinate.

N100 G0G90G94Z3.5F0
(Z near pallet centerline)

N110 G81X10.Y10.Z7.25R9.F10.S300M03
(Z rapids to programmed R location first, then X and Y axes rapid to location. Z feeds to depth, returns to R coordinate).

2. In the fixed cycle positioning blocks the X and/or Y axis positions at rapid between coordinates at the Programmed R location. Each fixed cycle block begins and ends at the Z position determined by the R word. Example:

N90 G0G90G94X15.Y11.5Z20.M25
(Restart Block)

N100 G81X15.Y11.Z6.5R7.5F15.S475M03
(set up block. The Z axis position at end of this block is 7.5 - the R word value)

N110 X14.Y9.
(X and Y at rapid with Z axis at 7.5 - the R word value)

N120 Y6.
(Y at rapid with Z axis at 7.5 - the R word value)

N130 G80
(cancel fixed cycle in block by itself. Z currently at 7.5 - the R word value)

FIXED CYCLE WITH VARYING Z DEPTHS AND R PLANES

The programmer may often find that a particular group of holes requires the same diameter tool and the same machining methods but that hole depths vary or the surface planes on which the holes are located may change. A fixture strap, or clamp, may also require a change of R value if the projection precludes the use of the original R value for positioning between holes.

NOTE

With KT-APL it is recommended that the G77 Cycle Inhibit command be used to maneuver the spindle around some obstruction such as a clamp. For more information on this command see Chapter 8.

The KT/CNC control is capable of making the following adjustments and decisions should the above circumstances occur:

1. When a new Z depth plane is programmed, the plane change is recognized for the hole coordinates which appear in the same block as the new Z plane command. The new Z plane remains in effect for all hole coordinates for that fixed cycle until Z is changed or fixed cycle is canceled.

2. When a new R clearance plane is programmed, a decision is made by the control as follows:

   a. If the new R plane programmed is positive with respect to the last R plane, Z axis retracts at rapid traverse to the new R plane and then X and / or Y axis positions at rapid traverse.

   b. If the new R plane programmed is negative with respect to the last R plane, X and/or Y positions at rapid traverse and then Z axis approaches the new R plane at rapid traverse.

3. New Z and R planes can be programmed in the same block and both operate according to the description given above. Example:

N90 Z20.F0 (Retract)
Z retracted

N100 G0G90G94G81X10.Y14.Z9.R10.1F18.S475M3 (Point 1)
Set up block. X and Y position at rapid, to Point 1, then Z advances to R plane on surface A at rapid. Fixed cycle performed. Z retracts to R plane.

N110 X16.Y12. (Point 2)
X and Y position at rapid to Point 2 with Z remaining at R plane. Fixed cycle performed.

N120 X9.Y10.Z13.R14.1 (Point 3)
New R plane (14.1 inches) is positive with respect to previous R plane (10.1 inches), new Z depth coordinate. Z positions at rapid to R coordinate, then X and Y position at rapid to Point 3 on Surface B. Fixed cycle performed.

N130 X14.Y9. (Point 4)
X and Y position at rapid to Point 4 with Z remaining at R plane. Fixed cycle performed.

N140 X9.Y6.Z6.R7.1 (Point 5)
New R plane (7.1 inches) is negative with respect to previous R plane (14.1 inches), new Z depth coordinate. X and Y position at rapid to Point 5 with Z at (14.1 inches); then Z advances at rapid to new R coordinate. Fixed cycle performed.

N150 X16.Y3. (Point 6)
X and Y position at rapid to Point 6 with Z remaining at R plane. Fixed cycle performed.

N160 G80 (cancel)
Fixed cycle canceled in a block by itself. Z axis remains at R coordinate (7.1 inches) until reprogrammed.

This example demonstrates use of G77 cycle where clearance plane must be changed to avoid collision with fixture strap or clamp.

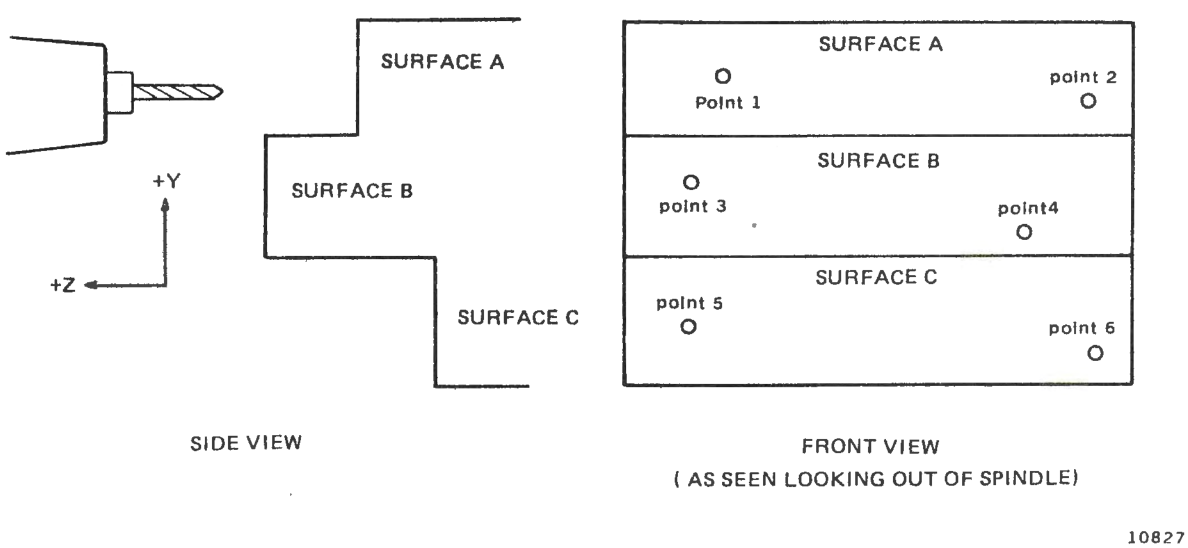


Fig 5.11 FIXED CYCLE with VARYING Z DEPTHS and R PLANES

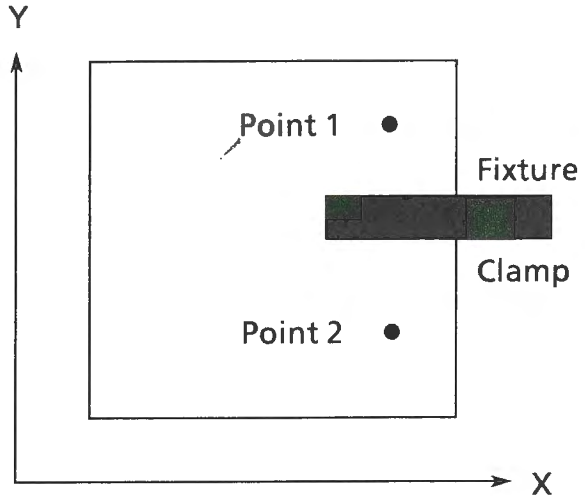


Fig. 5.12 G77 OPERATION

N100 G81X10.Y10.Z6.R7.1F13.S525M03 ... (Point 1)
Set up block. Drill Point 1. The fixture clamp projects from the part surface in line with Point 2. This precludes a rapid move along the clearance plane established in this block. A new clearance plane must be programmed to prevent a collision.

N110 G77Z10.1 (fixed cycle inhibit (non-modal) to clearance above fixture clamp).

G77 establishes new clearance position for the next rapid move in X and Y to clear the fixture clamp. Z retracts at rapid to clearance R plane (10.1").

N120 X11.Y5.Z6. .... (Point 2)
X and Y position at rapid to Point 2 with Z at (10.1 inches). Then Z advances at rapid to R value (7.1 inches). Fixed cycle performed.

N130 G80.. (cancel)
Fixed cycle canceled in block by itself.

FIXED CYCLES WITH B-AXIS INDEX

A fixed cycle routine may need to be performed on two or more B axis positions. For example, a hole pattern at B0 and an identical hole pattern at B90. Under these circumstances it is recommended that the fixed cycle be canceled after the operations performed at each B position. Program Z axis to clearance, index or rotate B axis to new location then re-establish the fixed cycle at that location.

AXIS INVERSION (MIRROR IMAGE)

Functional Description:

The Axis Inversion feature inverts the program-generated motions for each axis selected. When an axis is inverted, the absolute machine zero position for that axis becomes the positive end of maximum travel and the positive end of maximum travel becomes zero.

Programming Application:

1. Can be used to reduce program length and programming time on symmetrical parts by enabling the programming of one-half to one-fourth of the complete part.
2. Matching or facing parts can be generated from a single part program.

Programming Format:

Axis Inversion is programmed using a Type II data statement. This consists of a control Out character, the mnemonic MIR followed by a comma, a space, the letter address word X followed by either a 1 or a 0, the letter address word Y followed by either a 1 or a 0, and finally the control In character. The numbers 1 or 0 which follow the axis letter address word indicate the state of the axis inversion mode; 1 = Turn On mode, 0 = Turn Off mode. Examples:

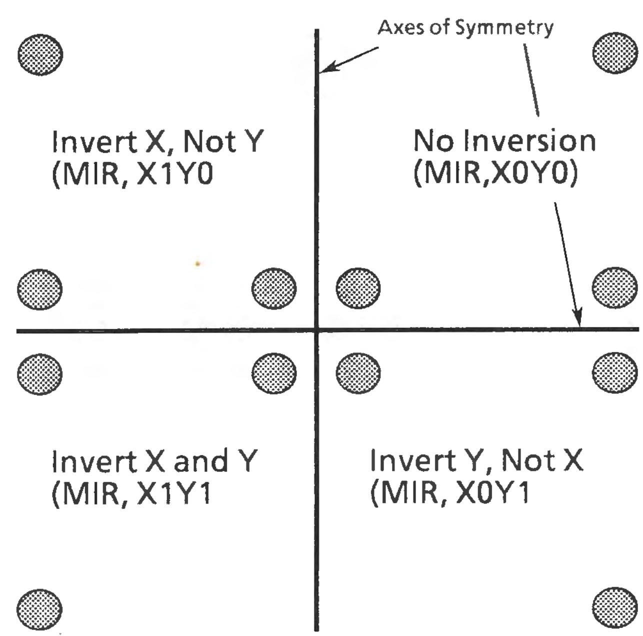


Fig. 5.13 AXES of SYMMETRY - MIRROR IMAGE

(MIR, X1Y0) ... Invert X but not Y.
(MIR, X0Y1) ... Invert Y but not X.
(MIR, X1Y1) ... Invert both X and Y.
(MIR, X0Y0) ... Invert neither X or Y. Turn off mode.

In EIA (odd parity) format, the Control Out and Control In characters are represented by a percent sign (%).

%MIR, X1Y1%

Programming Considerations:

1. Axis Inversion is operational only in the XY plane (G17).
2. The axis inversion block may be preceded by a sequence number and a block delete slash code (/).
3. Axis inversion has no effect on axis grid.
4. Axis inversion has no effect on axis jog; however, do not jog any axis if in G98 mode.
5. G02 or G03 is reversed if only one of the two axes is inverted.
6. G41 and G42 is reversed if only one of the two axes is inverted.
7. When G98 is modal, the offset entries are inverted.
8. When M25 or M06 is commanded, Axis Inversion is bypassed for that block only.
9. Circular Interpolation and Tool Length Compensation work correctly with Axis Inversion.

Special Programming Consideration for G91 Mode:

If Axis Inversion is programmed in G91 mode, special consideration must be given to the return of the X and/or Y axis slide to its original position after the programming of an M06 or M25 command.

Programming Instructions:

1. The programming of Axis Inversion may require the combined efforts of the part programmer and the machine operator. If this is the case the programmer must provide adequate instructions, either in written form, or in programmable operator messages within the text of the part program.
2. Position the axis slides to a central point (the axis of symmetry) around which the Axis Inversion is to occur.
3. With the axis slides at the axis of symmetry program the Type II data statement containing the Axis Inversion commands.
4. The Axis Inversion command must be programmed in a data block by itself.

Axis Inversion Operation:

There are three methods of using Axis Inversion, both involve minimal operator intervention.

Method 1.

1. Command the axis slides to the axis of symmetry.
2. Program an M00 (Program Stop).
3. In the next data block, insert a programmable operator message telling the machine operator to enter the appropriate Axis Inversion commands via the MDI mode. The machine operator may then return to the AUTO mode and continue the part processing cycle.
4. Prior to the end of the part program, cancel the Axis Inversion mode with (MIR, X0Y0) in a data block by itself.

Method 2.

1. In writing the part program insert consecutive sequence numbered data blocks containing the Axis Inversion commands. Each command must be in a data block by itself.
2. Precede each of the numbered data blocks in step 1 with a block delete slash code (/).
3. Prior to the end of the program, cancel the Axis Inversion mode with (MIR, X0Y0) in a block by itself.

In this manner, the operator may proceed through the first run of the program with Axis Inversion disabled by Block Delete. Following an M30 command, the operator can, by [SINGLE BLOCK] method, select the appropriate Axis Inversion command disabling Block Delete for that block only. Adequate operator instructions should accompany this procedure!!

Method 3.

In the following example, we will program all four quadrants of the axis of symmetry. To Mirror Image only one additional quadrant, the following example may be tailored to your specific needs.

1. Write base part program for machining operations in the Top Half, Right Hand Side, or the Upper Right Hand Quadrant of the axis of symmetry. This is the 12:00 to 3:00 position. Either Incremental, or Absolute Mode with a G98 offset, may be used.
2. Load onto the hard disk unit under a permanent file name which we will refer to as "File A".

3. Copy "File A" to a second file on the disk we will call "File B".
4. Using the File Handler, to main file "File A" append (Command 3) "File B".
5. Repeat this procedure two additional times if it is desired to mirror all four quadrants.
6. Delete (2) "File B".
7. Invoking the File/Editor ([SELECT] [J]) edit "File A".
8. Block search to the last N-number in the original program.
9. Going to character mode, delete the line with the M30 command used to end the part program.
10. Move the cursor down to the block following the motion command to the axis of symmetry.
11. Open a new line and enter the statement "(MIR,X1Y0)" statement. This is the 9:00 o'clock to 12:00 o'clock position.
12. Repeat steps 8, 9, 10 and 11, entering the second "(MIR," statement, (MIR,X0Y1) mirror imaging Y axis motions. This is the 3:00 to 6:00 o'clock.
13. After entering the second "(MIR," statement, move the cursor up to the line following the motion to the axis of symmetry at the end of the initial program.
14. Open a new line and enter the MIR statement (MIR,X0Y0).
15. Repeat steps 8, 9, 10 and 11 entering the statement "(MIR,X1Y1)" mirroring both X and Y axes. (6 o'clock to 9:00 o'clock quadrant.)
16. Press [HOME] [C] moving the cursor to the end of the file.
17. Move the cursor back up to the block following the move to the axis of symmetry.
18. Enter the command "(MIR,X0Y0)" canceling Mirror Image.
19. Press [HOME] [A] reverting back to the Command Mode.
20. Press [1] [ENTER] invoking the Output Mode.
21. Enter the name of "File A" and press [ENTER]. In response to the question "Delete Duplicate File?" respond yes ([1] and [ENTER]).
22. Invoke the File Handler with [SELECT] [F].
23. Complete the procedure by resequencing the part program.

PROGRAMMING CIRCULAR INTERPOLATION USING I, J AND K COMMANDS

Circular interpolation is programmed by writing the Preparatory function G02 or G03 in a data block which contains information relating to center point and end point coordinates for the desired arc path. The control generates an arc path by the coordinated motion of two axes, individually proportioned according to a sine-cosine relationship. Interpolation may be in any one of three planes. A typical data block generating a circular arc might look like this:

N147 G17G01G02X13.5Y7.I9.J7.F9.5

NOTE

Circular interpolation may also be programmed using an R-word instead of centerpoint words. See the discussion later in this chapter.

GENERAL DESCRIPTION

Circular interpolation is used in contour milling requiring the generation of a circular cutter path. Any size arc, up to a maximum of 360° may be programmed in a single block. The center point of a programmed arc does not need to lie within the machine axis slide quadrants (i.e., within the machining envelope). Circular interpolation may be programmed in the inch or metric mode using either absolute or incremental positioning, and in either the positioning mode (G00) or the contouring mode (G01). Axes motion during a circular interpolation block may be generated in either the clockwise or counterclockwise direction. Any, or all, of the available offset commands may be in effect during circular interpolation.

FORMAT FOR CIRCULAR INTERPOLATION

1. The Starting Point - This is programmed prior to the circular interpolation block.
2. The Plane Selection - This may be programmed prior to (in the case of G17) or in the same block as (in the case of G18 or G19) the circular interpolation block.
3. The Preparatory Function - These must be programmed in the circular interpolation block.
4. The End Point
5. The Center Point

These, of course, are in addition to other programming considerations such as spindle speed and direction, feedrate, coolant, etc.

1. The Starting Point

The axes must be positioned to a start point prior to generating circular motion. This may involve one, or several, blocks of information depending upon the circumstances. This start coordinate must be a point on the circular arc.

2. The Plane Selection

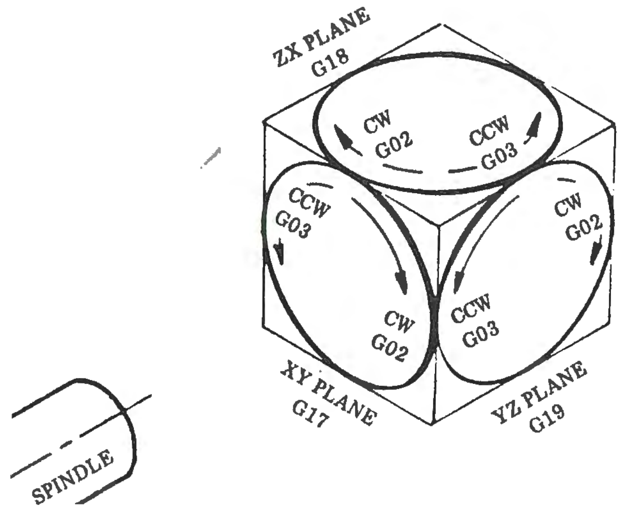


Fig. 5.14 CIRCULAR INTERPOLATION PLANES

The plane selection command determines the axes involved in the generation of the circular arc. The plane selection codes are G17 (XY plane), G18 (XZ plane), and G19 (YZ plane):

G17 Plane - Modal upon machine startup, does not need to be reprogrammed in circular interpolation block. Generates circular arc in the XY plane.

G18 Plane - Generates circular arc in XZ plane. For Circular Interpolation in the XZ plane, the G18 command is generally written in the circular interpolation block, rather than in a preceding block.

G19 Plane - Generates circular arc in YZ plane. For circular interpolation in the YZ plane, the G19 command is generally written in the circular interpolation block, rather than in a preceding block.

3. The Preparatory Function

Circular motion is initiated by programming a circular interpolation preparatory command. The G02 code initiates motion in the clockwise direction, G03 motion in the counterclockwise direction. These commands are non-modal so must be reprogrammed in each data block which is to generate a circular arc.

4. The End Point

The end point coordinates for the linear axes must be programmed in the circular interpolation block. The end point coordinates for circular interpolation in the G17 plane are an X and Y entry; for the G18 plane an X and Z entry; for the G19 plane a Y and Z entry.

5. The Center Point

The center point coordinates around which the circular arc is generated must be programmed in the circular Interpolation block. The center point coordinates are represented by the letter address words I, J, and K.

The I word represents center point information parallel to the X axis.

The J word represents center point information parallel to the Y axis.

The K word represents center point information parallel to the Z axis.

In the G17 mode the arc center point is programmed by the I and J word.

In the G18 mode the arc center point is programmed by the I and K word.

In the G19 mode the arc center point is programmed by the J and K word.

CONSIDERATIONS, RESTRICTIONS AND FORMULAS

In writing commands for circular interpolation there are certain considerations and restrictions which the programmer must keep in mind in addition to the basic format:

Programming Considerations:

1. The distance from the start point of the arc to the center point of the arc MUST be equal to the distance from the end point of the arc to the center point of the arc (see formula below).
2. The radius of the programmed arc is defined as the distance from the start point coordinate to the programmed center point coordinate. The distance from the end point to the center point must be equal to the radius.
3. The arc may be programmed in the G01 (contouring) mode to eliminate the presence of cutter dwell marks.
4. Feedrate is generally programmed in G94 mode as this requires no further calculations. Feedrate is applied directly to arc path.

Programming Restrictions:

1. Circular interpolation cannot be programmed at rapid traverse (F0).
2. Rotary axis commands (A, B, or C) or table index CANNOT be programmed in the same block with circular interpolation.
3. The minimum radius that can be programmed is .001" (.01 mm). The maximum radius that can be programmed is 1690.9320 inches. (42949.672 mm).

Formulas:

1. To determine if distance from start point to center point is equal to distance from end point to center point. This formula is applicable only in G90 and G17 modes.

$$\text{Rs (radius at start)} = \sqrt{(Xs - I)^2 + (Ys - J)^2}$$

$$\text{Re (radius at end)} = \sqrt{(Xe - I)^2 + (Ye - J)^2}$$

Rs must equal Re

Where:

- Xs = X coordinate at the start point of the arc.
- Ys = Y coordinate at the start point of the arc.
- I = I entry in block containing G2 or G3.
- J = J entry in block containing G2 or G3.
- Xe = X coordinate at the end point of the arc.
- Ye = Y coordinate at the end point of the arc.
- Rs = Radius at the start of the arc.
- Re = Radius at the start of the arc.

2. For G91 mode in G17 plane:

Radius = $(I)^2 + (J)^2$

CIRCULAR INTERPOLATION IN ABSOLUTE MODE G90

In the G90 mode all axis related commands in the circular interpolation block are defined as absolute positions from machine zero or from a programmed zero. This includes the X, Y, and Z commands for the end point and the I, J, and K commands for the center point. (See Fig. 5.15)

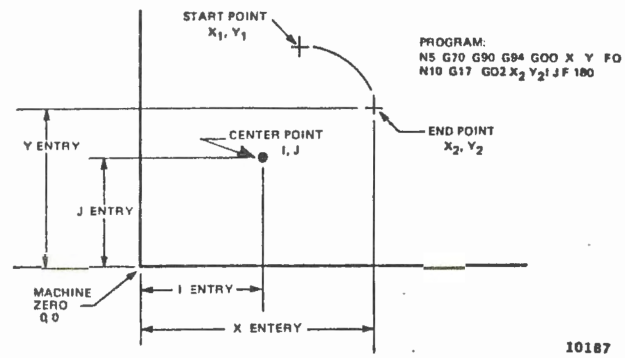


Fig. 5.15 CIRCULAR INTERPOLATION (G90)

CIRCULAR INTERPOLATION IN INCREMENTAL MODE G91

In the G91 mode all axis related commands in the circular interpolation block are defined as incremental distances from the start point coordinate of the arc. This includes the X, Y, and Z commands for the end point and the I, J, and K commands for the center point. If any of these commands represents a negative incremental distance, then that command must be so signed. The control always assumes these commands as positive unless a negative sign is programmed. Example:

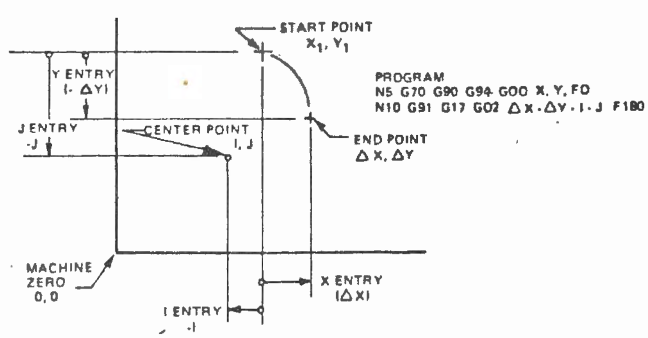


Fig. 5.16 CIRCULAR INTERPOLATION (G91)

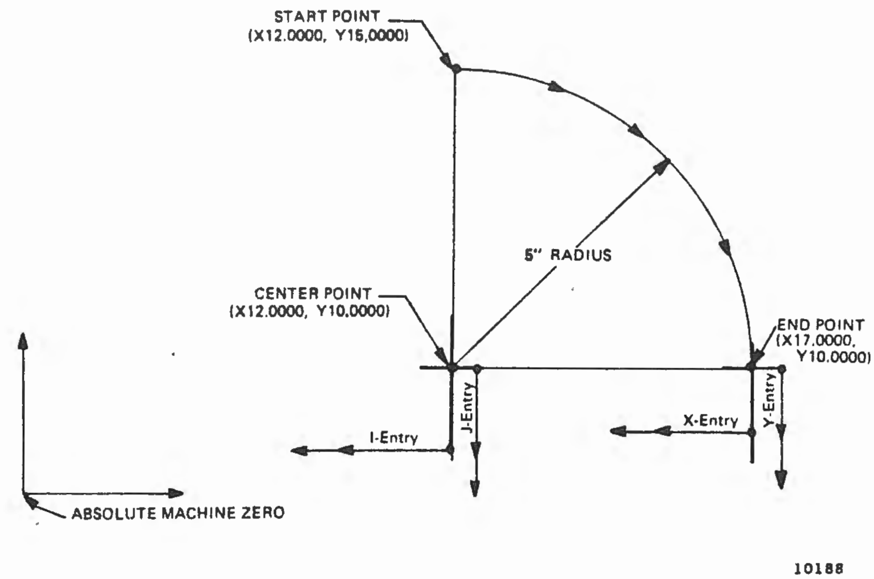


Fig. 5.17 90° ARC with G2 / G90 MODE

90° ARC with G2 / G90 MODE

Start point coordinates are given on the drawing:

X = 12.0000 Y = 15.0000

End point coordinates are given on the drawing:

X Entry = 17.0000 Y Entry = 10.0000

Arc center coordinates are given on the drawing:

I Entry = 12.0000 J Entry = 10.0000

Part Program:

G17 G70 G90 G94 X12. Y15. F0
G02 X17. Y10. I12. J10. F17.

30° ARC with G03 / G90 MODE

Start point calculations:

X = 12.0000 + (radius x cos 30°)
= 12.0000 + (5.0000 x .866026)
= 12.0000 + (4.3301) = 16.3301

Y = 10.0000 + (radius x sin 30,°)
= 10.0000 + (5.0000 x .5000)
= 10.0000 + (2.5000) = 12.5000

End point calculations

X = 12.0000 + (radius x cos 65°)
= 12.0000 + (5.0000 x .422618)
= 12.0000 + (2.1131) = 14.1131

Y = 10.0000 + (radius x sin 65°)
= 10.0000 + (5.0000 x .906308)
= 10.0000 + (4.5315) = 14.5315

Center point coordinates are given on the drawing.

I Entry = 12.0000 J Entry = 10.0000

Part Program:

G17 G70G90G94X16.3301Y12.5F0
G03 X14.1131Y14.5315I12.J10.F13.

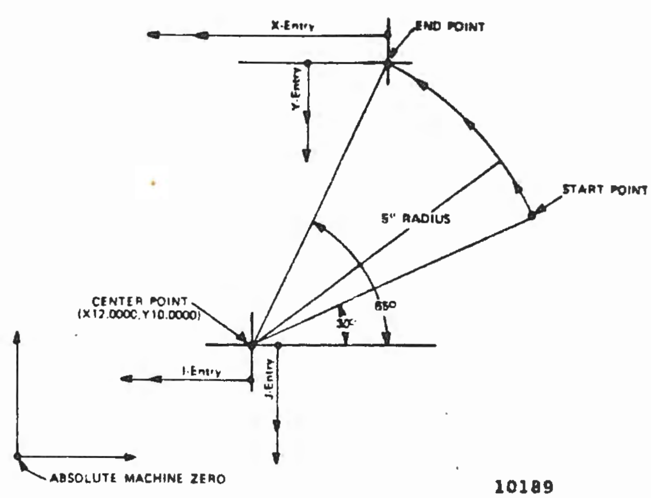


Fig. 5.18 30° ARC with G3 / G90 MODE

Formulas used to determine if the arc is programmed correctly.

$$Rs = \sqrt{(Xs-I)^2 + (Ys-J)^2}$$

$$Rs = \sqrt{(16.3301-12.0000)^2 + (12.5000-10.0000)^2}$$

$$Rs = \sqrt{(4.3301)^2 + (2.5000)^2}$$

$$Rs = \sqrt{25.0000}$$

$$Rs = \sqrt{5.0000}$$

Fig. 5.19 RADIUS AT START FORMULA

$$Re = \sqrt{(Xe-I)^2 + (Ye-J)^2}$$

$$Re = \sqrt{(14.1131-12.0000)^2 + (12.5000-10.0000)^2}$$

$$Re = \sqrt{(2.1131)^2 + (4.5315)^2}$$

$$Re = \sqrt{25.0000}$$

$$Re = \sqrt{5.0000}$$

Fig. 5.20 RADIUS AT END FORMULA

360° ARC WITH G02/G90 MODE

Start point coordinates are given on the drawing.

X = 9.0000 Y = 17.0000

End point coordinates are given on the drawing.

X Entry = 9.0000 Y Entry = 17.0000

Center point coordinates are given on the drawing.

I Entry = 9.0000 J Entry = 13.0000

Part Program:

G17 G70 G90 G94 X9. Y17. F0
G02 X9. Y17. I9. J13. F14.

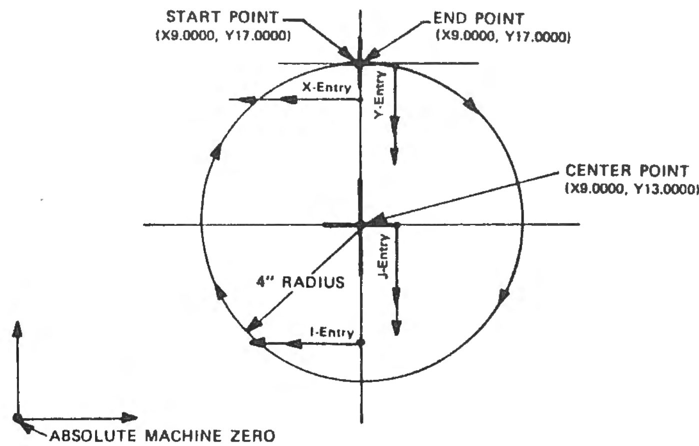


Fig. 5.21 360° ARC with G2 / G90 MODE

INCREMENTAL MODE (G91)

90° ARC with G2 / G91 MODE

Start point coordinates are given on the drawing.

X = 12.0000 Y = 15.0000

End point entries are determined from the drawing.

X Entry = 5.0000 Y Entry = -5.0000

Center point entries are determined from the drawing.

I Entry = 0 J Entry = -5.0000

Part Program:

G17 G70 G90 G94 X12. Y15. F0
G91 G02 X5. Y-5. I0 J-5. F14.5

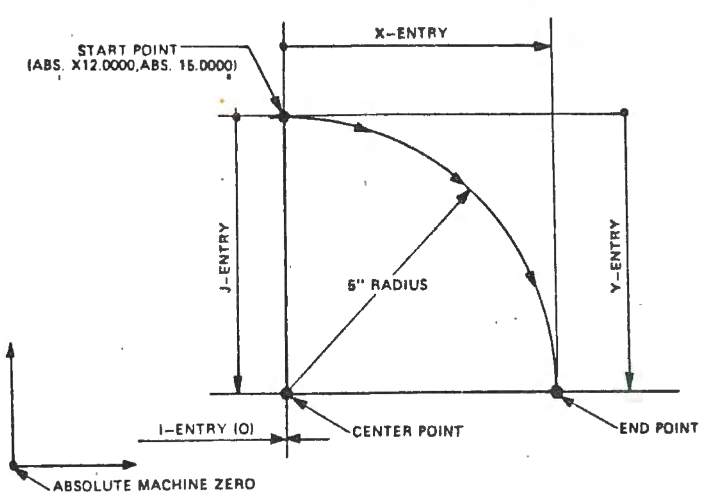


Fig. 5.22 90° ARC with G2 / G91MODE

30° ARC with G03/G91 MODE

Start point Calculations

X = 12.0000 + (radius x cos 30°)
= 12.0000 + (5.0000 x .866026)
= 12.0000 + (4.3301) X = 16.3301

Y = 10.0000 + (radius x sin 30°)
= 10.0000 + (5.0000 x .5000)
= 10.0000 + (2.5000) Y = 12.5000

End point Calculations

X = (radius x cos 65°) + I
= (5.0000 x .422618) + (-4.3301)
= 2.1131 -4.3301 X = -2.2170

Y = (radius x sin 65°) + J
= (5.0000 x .906308) + (-2.5000)
= 4.5315 -2.5000 Y = 2.0315

Center point calculations

I Entry = 0 -(radius x cos 30°)
= 0 -(5.0000 x .866026) I Entry = -4.3301

J Entry = 0 -(radius x cos 30°)
= 0 -(5.0000 x .5000) J Entry = -2.5000

Part Program:

G17 G70 G90 G94 X16.3301 Y12.5 F0
G91 G03 X-2.217 Y2.0315 I-4.3301 J-2.5 F12.8

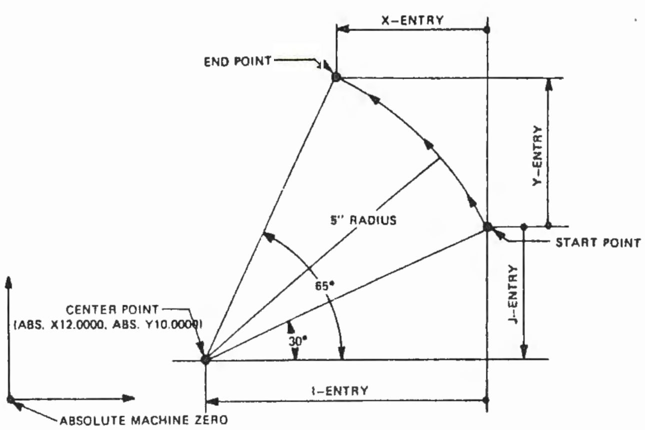


Fig. 5.23 30° ARC with G3 / G91 MODE

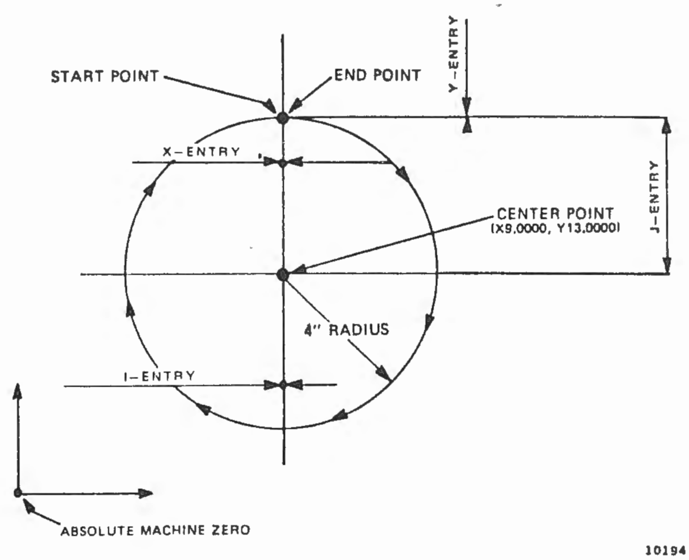


Fig. 5.24 360° ARC with G2 / G91 MODE

360° ARC with G02/G91 MODE

Start point coordinates are given on the drawing.

X = 9.0000 Y = 17.0000

End point entries are determined from the drawing.

X Entry = 0 Y Entry = 0

Center point entries are determined from the drawing.

I Entry = 0 J Entry = -4.0000

Part Program:

G17 G70 G90 G94 X9. Y17. F0
G91 G02 X0 Y0 I0 J-4. F9.3

PROGRAMMING HELICAL INTERPOLATION

Helical interpolation is programmed by writing a block of circular interpolation which includes an end point command for the third linear axis and a command which represents the rise per radian. If no rise per radian command is included, the control calculates it automatically. The control generates a helix-type path by the coordinated motion of two axes forming a circular arc with the simultaneous linear movement of the third axis. Interpolation may be in any one of three planes.

A typical data block generating a helical path might look like this:

N10G17G01G02X14.Y10.Z6.7I10.J10.K1.25G93F2.3

GENERAL DESCRIPTION

Helical interpolation is used in contour milling which requires the generation of a helical cutter path. Any size helical arc up to a maximum of 360° may be programmed in a single data block. The center point of a programmed arc does not need to lie within the machining envelope. Helical interpolation may be programmed in the inch or metric mode using either absolute or incremental positioning, and in either the positioning mode (G00) or the contouring mode (G01). Axes motion during a helical interpolation block may be generated in either the clockwise or counterclockwise direction.

Any, or all, of the available offset commands may be in effect during helical interpolation machining.

HELICAL INTERPOLATION FORMAT

The basic format for programming helical interpolation includes the following components:

1. The Starting Point - this is programmed prior to the helical interpolation block.
2. The Plane Selection - this may be programmed prior to (in the case of G17) or in the same block as (in the case of G18 and G19) the helical interpolation block.
3. The Preparatory Function
4. The End Point
5. The Center Point

The above items must be programmed in the helical interpolation block.

6. The Rise Per Radian (if desired)
7. The G93 F Word (if required)

BLOCK INPUT DESCRIPTION

1. The Starting Point

The axes must be positioned to the start point of the helical arc prior to generating any helical motion. This may involve one, or several, blocks of information depending upon the circumstances. This start coordinate must be a point on the helical arc.

2. The Plane Selection

The Plane selection command determines the two axes involved in the generation of the circular arc. The third linear axis departs in a linear motion. The plane selection codes are G17 (XY plane), G18 (XZ plane) and G19 (YZ plane):

G17 Plane - Modal upon machine startup, so does not need to be reprogrammed in helical interpolation block. Generates circular arc in the XY plane, linear movement of the Z axis during helical cut.

G18 Plane - Generates circular arc in the XZ plane, linear movement of the Y axis during the helical cut. For helical interpolation in the XZ plane the G18 command is generally written in the helical interpolation block, rather than in a preceding block.

G19 Plane - Generates circular arc in the YZ plane, linear movement of the X axis during the helical cut. For helical interpolation in the YZ plane the G19 command is generally written in the helical interpolation block, rather than in a preceding block.

3. The Preparatory Function

Helical motion is initiated by the programming of the circular interpolation preparatory command. The G02 code initiates motion in the clockwise direction, G03 initiates motion in the counterclockwise direction. These commands are non-modal so must be programmed in each data block which is to generate a helical arc.

4. The End Point

The end point coordinates for all three axes must be programmed in the helical interpolation block. The end point coordinates for helical interpolation in the G17 plane are an X and Y entry for the circular arc, and a Z entry for the linear move; in the G18 plane an X and Z entry for the circular arc, and a Y entry for the linear move; in the G19 plane a Y and Z entry for the circular arc, and an X entry for the linear move.

5. The Center Point

The center point coordinates around which the circular arc is generated must be programmed in the helical interpolation block. The center point coordinates are represented by the letter address words I, J, and K.

- The I word represents center point information parallel to the X axis.
- The J word represents center point information parallel to the Y axis.
- The K word represents center point information parallel to the Z axis.
- In the G17 mode the center point is programmed by the I and J word.
- In the G18 mode the center point is programmed by the I and K word.
- In the G19 mode the center point is programmed by the J and K word.

6. The Rise Per Radian (if desired)

Inclusion of rise per radian commands is optional in this system for helical motion. If desired, the user may include a rise per radian command (I, J, or K); however this is not a requirement for helical interpolation.

If no rise per radian entry is included in a helical interpolation block, the control calculates the rise per radian value automatically.

If a rise per radian command is included it defines the relative motion of the third (linear) axis move in a helical cut so that the circular motion and the linear motion begin at the same time and reach the end point at the same time. The rise per radian entry is always a positive number and is calculated by the control if not included in the block. The rise per radian value is also used to calculate G93 feedrates for Formula 2 below.

7. The G93 F Word (if required)

Feedrates for helical contouring may be programmed in G94 or G93 mode. In G94 mode the programmed feedrate is applied to the two-axis circular arc path. The control calculates the feedrate for the third (linear) axis, based on the rise per radian command.

In G93 mode the programmed feedrate is applied directly to the helical path (i.e., tool path). The G93 F-word is determined by calculation based upon the use of one of the two formulas listed below.

The type of feedrate coding to use for the helical cut is based upon the following general rules:

1. If the length of the linear departure is less than 10% of the circular arc length, then use G94 feedrate coding.
2. If the length of the linear departure is greater than 10% of the circular arc length, then use G93 feedrate coding.

FORMULAS (See Chapter 2 - G93 Description)

Below is a list of formulas applicable to programming helical interpolation. Examples of the application of these formulas are detailed later in this section:

A. G93 Formulas: (Use ONE of the following)

FORMULA No. 1

$$\text{G93 F-Word} = \frac{\text{Desired Feedrate x Angle in Radians}}{\text{Departure Distance}}$$

$$\text{G93 F-Word} = \frac{\text{Desired Feedrate x Angle in Degrees}}{\text{Departure Distance x 57.2958}}$$

$$\text{G93 F-Word} = \frac{\text{Desired Feedrate x Angle in Degrees x 0.01745}}{\text{Departure Distance}}$$

Where:

$$\text{Departure Distance} = \sqrt{(\text{Arc Length})^2 + \text{Length of Linear Axis Motion})^2}$$

$$\text{Arc Length} = \frac{\text{Angle In Degrees x Radius}}{57.2958}$$

$$= \text{Angle in Degrees x 0.01745 x Radius}$$

FORMULA No. 2

$$\text{G93 F-Word} = \frac{\text{Desired Feedrate (in IPM or MMPM)}}{\sqrt{\text{Radius}^2 + \dfrac{(\text{Linear Motion})^2}{(\text{Angle in Radians})^2}}}$$

$$\text{G93 F-Word} = \frac{\text{Desired Feedrate (in IPM or MMPM)}}{\sqrt{\text{Radius}^2 + (\text{I,J or K})^2}}$$

(:where I,J or K is the rise per radian value)

B. General Formulas:

2 Pi Radians = 360°   360° = 2 Pi Radians

$1 \text{ Radian} = \frac{360°}{2\text{ Pi}}$ $1° = \frac{2 \text{ Pi Radians}}{360}$

1 Radian = 57.2958°   1° = 0.01745 Radians

C. Rise per radian calculation:

$$K = \frac{\text{Z Departure}}{\text{XY Arc in Radians}}$$

$$J = \frac{\text{Y Departure}}{\text{ZX Arc in Radians}}$$

$$I = \frac{\text{X Departure}}{\text{YZ Arc in Radians}}$$

EXAMPLE

Program the helical arc shown below in the absolute mode, using the G93 feedrate coding. Feedrate along the cutter path (departure distance) to be 14.5 IPM. The arc is generated in a clockwise direction through 90° with a concurrent linear Z axis motion of 2.5".

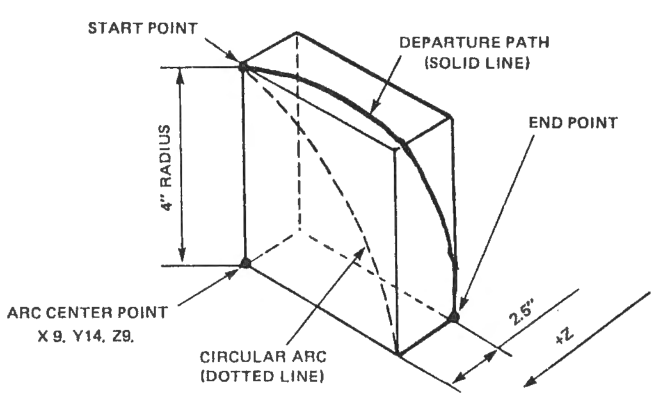


Fig. 5.25 HELICAL INTERPOLATION

1. Start point coordinates are determined from the drawing.

   X = 9.   Y = 18.   Z = 9.

2. End point coordinates are determined from the drawing.

   X = 13. Y = 14.   Z = 9.-2.5 = 6.5

3. Center point coordinates are given on the drawing.

   I = 9.   J = 14.

4. Rise per radian calculation (calculated by control)

   $$K = \frac{\text{Z Departure (linear motion)}}{\text{XY Arc in Radians}}$$

   $$= \frac{2.5}{90 \text{ X } 0.01745}$$

   K = 1.5918

5. Arc length calculation (*is G93 necessary?*)

   Arc Length = Radius x Angle in Radians

   = 4. x 90. x 0.01745

   = 6.282" Length of circular arc (dotted line)

   Length of linear departure (Z) = 2.5"10% of arc length = .6282 (.10 x 6.282)

Therefore G93 feedrate coding is necessary. (2.5 is greater than .6282)

6. G93 Feedrate Calculation

$$\text{G93 F-Word} = \frac{\text{Desired Feedrate x Angle in Radians}}{\text{Departure Distance}}$$

$$\text{G93 F-Word} = \frac{14.5 \times 90. \times 0.01745}{\sqrt{(\text{Arc Length})^2 + (\text{Linear Motion})^2}}$$

$$\text{G93 F-Word} = \frac{14.5 \times 90. \times 0.01745}{\sqrt{(6.282)^2 + (2.5)^2}}$$

$$\text{G93 F-Word} = \frac{22.77225}{6.76117777} = 3.4$$

FORMULA No. 1

$$\text{G93 F-Word} = \frac{\text{Desired Feedrate (in IPM or MMPM)}}{\sqrt{\text{Radius}^2 + \frac{(\text{Linear Motion})^2}{(\text{Angle in Radians})^2}}}$$

$$\text{G93 F-Word} = \frac{14.5}{\sqrt{4^2 + \frac{(2.5)^2}{(1.5705)^2}}}$$

$$\text{G93 F-Word} = \frac{14.5}{4.3051115} = 3.368089$$

$$\text{G93 F-Word} = 3.4 \text{ rounded}$$

FORMULA No. 2

Write the Part Program

N100 G17G70G90G94X9.Y18.Z9.1F0
N110 Z9.F14.5
N120 G02G93X13.Y14.Z6.5I9.J14.F3.4

CIRCULAR AND HELICAL INTERPOLATION USING R WORD

An arc can be generated by programming a G02 or G03 in a block which contains information relating to the desired end point coordinates (X and Y words) and the radius of the arc (R word). A typical data block using this method might look like this:

N200 G17 G03 X8. Y11. R3.5 F12. (arc)
N200 G17 G02 G93 X8. Y11. Z4.75 R3.5 F1.2 (helix)

RESTRICTIONS AND LIMITATIONS

Arcs and helical paths programmed using an R word are restricted to the Absolute Mode (G90) in the XY plane (G17). Also, this method cannot be used to cut a complete circle. The arc path becomes increasingly less accurate as the angle through which the cutter traverses approaches 360°.

DEFINITIONS

1. The Start Point

The axes must be positioned to a start point prior to generating the arc. The starting coordinate must be a point on the arc path.

2. The Preparatory Function

The arc path or helical motion is initiated by programming the circular interpolation preparatory command. The G02 code initiates motion in the clockwise direction, the G03 code initiates motion in the counterclockwise direction. These commands are non-modal so must be reprogrammed in each data block which is to generate an arc or helix. In addition, the preparatory functions G90 Absolute Mode and G17 XY plane selection must be in effect.

3. The End Point

The end point coordinates must be programmed in the data block containing the G02 or G03 command. If the end point coordinates are X and Y positions, then the resulting path will be an arc. If end point coordinates are an X and Y position along with a Z axis departure word, then the resulting path will be a helix.

4. The Center Point

The center point coordinates are calculated by the control and need not be entered into the data block. The center point may be one of two possible points depending upon the sign affixed to the R word, as illustrated in Figures 5.25 and 5.26.

5. The Radius

The radius is defined as the distance from the start point of the arc (or any point along the arc) to the center point of the arc. This distance from the center point is maintained throughout the arc or helical path until the end point coordinates are reached. The radius value is programmed into the data block by the R word. The R word may have either a positive value or a negative value. If the R word has a negative value, the negative sign must be programmed. The negative sign serves as a "flag" indicating to the control which one of the two possible paths is desired to reach the programmed end point.

6. The Rise Per Radian (for helical)

The rise per radian is the amount of Z axis departure per one radian of arc (57.2958°). It is calculated by the control and used to determine the feedrate applied to the Z axis. It is not necessary to enter the rise per radian command (K word) in a helical block, but if the programmer chooses to calculate and enter it into the data block using the formulas in previous helix section of Chapter 5, then the entered value supersedes the control calculated value.

Format

Once the linear axes have been positioned to a point on the desired arc path (the start point coordinates), the block which initiates the arc or helical path is next. This block contains the end point coordinates and the signed radius value, along with the G02 or G03 command. The control is able to calculate the center point coordinates because the start point is known, the end point is known, and the radius is known.

For each set of points (start and end point coordinates) connected by a specific radius and driven in a specific direction (either CW or CCW) there can be two possible center points. (See Figures 5.26 and 5.27)

As illustrated in Figures 5.25 and 5.26 there are four possible paths that can be taken to get from the same start point to the same end point. The path actually taken is determined by 1.) the direction chosen (G02 or G03) and 2.) the sign of the radius value (R word) which is subject to the following rules:

Rule 1

If the radius value is negative, the resulting arc path will always define an angle equal to or greater than 180° but less than 360°.

Rule 2

If the radius value is positive, the resulting arc path will always define an angle equal to or less than 180° but greater than 0°.

In Figures 5.26 and 5.27, the angles formed by programming a positive R word are shown to be about 60°. The complementary angles which are formed by programming a negative R word would then be about 300° (360 minus 60).

In Figure 5.27, the cutter originates at a start point of X10. Y10. and is driven to an end point of X11.69500 Y9.514 in a counterclockwise direction on a 1.5000" radius. The figure shows the two possible arc paths which are determined by the sign of the R word. Also included is the center point coordinates for the two

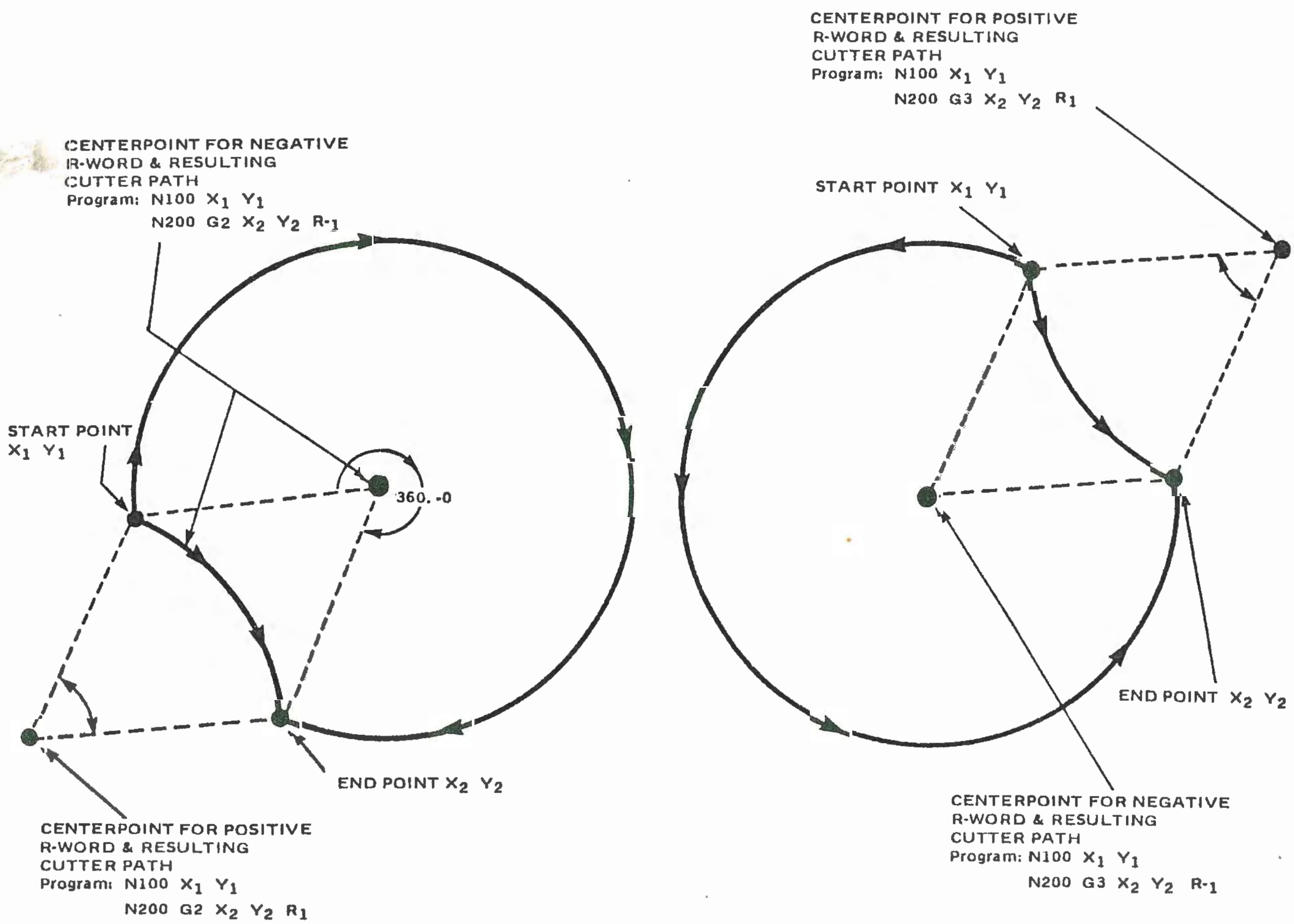


Fig. 5.26 POSITIVE and NEGATIVE R WORDS

possible arcs which the control automatically calculates. Arc path 1 generates an arc with an interior angle of 72°; arc path 2 generates an arc with an interior angle of 288°.

For Arc Path 1:

N100 X10.Y10.F0
(position axes to start point)
N110 G03X11.695Y9.514R1.5F6.5
(generate CCW arc-positive R value)

For Arc Path 2:

N100 X10.Y10.F0
(position axes to start point)
N110 G03X11.695Y9.514R-1.5F6.5
(generate CCW arc-negative R value)

PROGRAMMING ROTARY AXES

Rotary axis commands are programmed by writing the letter address word for the rotary axis followed by a number in the proper format which represents the desired degree position (G90) or incremental departure distance of the rotary axis (G91).

GENERAL DESCRIPTION

If equipped with a fourth or fifth axis, the machine tool is able to perform stand-alone rotary departures or rotary motion combined with linear axis moves. This enables four and five axis contouring capabilities.

ROTARY FEEDRATES AND AXES PRIORITY

1. G94 Mode

Single Rotary Only - When no linear axis departure is programmed in a block with a rotary departure command the F-Word is interpreted as degrees per minute in an F3.0 format (three digits, no decimal place). The Range is from F1 to F999 DPM.

Two Rotary Combined - When departure commands for two rotary axes are programmed in the same block, the feedrate is applied to the B axis only. The F-Word is interpreted as DPM in an F3.0 format (three digits, no decimal place). The range is from F1 to F999DPM. The second rotary axis (A or C) is assigned a feedrate by the control so that interpolation begins, and completes, simultaneously.

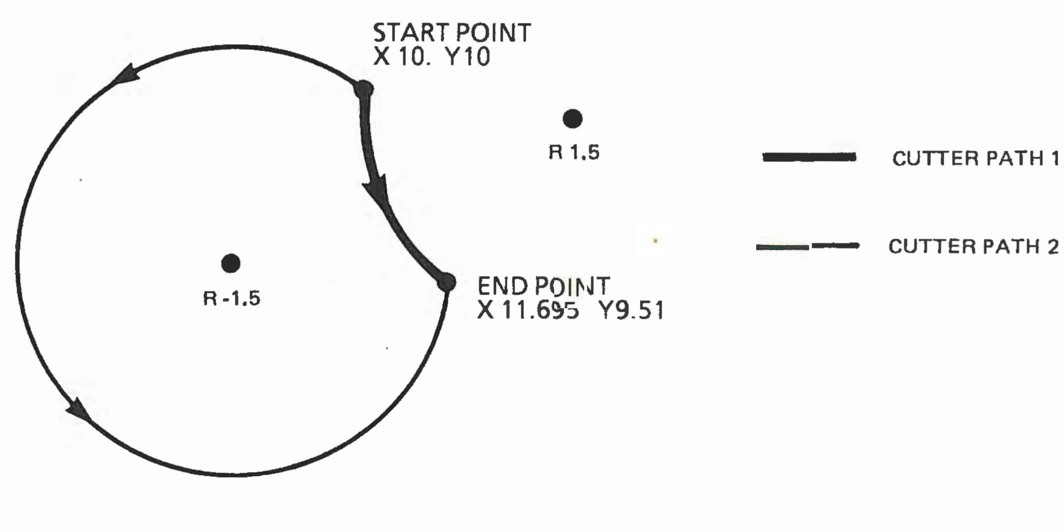


Fig.5 27

Rotary and Linear Combined - When departure commands for linear axes are programmed in the same block with rotary departures (4th and/or 5th axis), the feedrate is applied to the LINEAR departures. The control calculates the rotary motion feedrate to ensure that the rotary and linear interpolation begin, and complete, simultaneously. The resultant rotary feedrate can be calculated using the following:

$$\text{ROTARY FEEDRATE} = \frac{\dfrac{\text{ROTARY DEPARTURE (in degrees)}}{\text{LINEAR DEPARTURE}}}{\text{PROGRAMMED FEEDRATE}}$$

$$\text{ROTARY FEEDRATE} = \frac{\dfrac{\text{DEGREES}}{\text{INCHES}}}{\text{IPM}}$$

ROTARY FEEDRATE = DPM of Rotary Travel

FORMULA 3

2. G93 MODE

The G93 (Inverse Time) feedrate coding is preferable whenever linear and rotary departures are combined in a single data block. In the G93 mode, the feedrate is applied directly to the cutter path. The examples below illustrate the correct methods and formulas for calculating the inverse time feedrate.

ROTARY AXIS RAPID TRAVERSE

Rapid traverse (F0) moves involving rotary axes are subject to the following characteristics:

1. Single Axis Rotary (A or B or C) - The involved rotary axis receives a feedrate of maximum DPM.

2. Two Rotary Combined (B and A, or B and C) - The B axis receives a feedrate of maximum DPM. The second rotary axis receives a feedrate which is determined by the control so that interpolation begins, and ends, simultaneously.

3. Combined Linear and Rotary - The linear axis (axes) receive a feedrate of maximum inches or millimeters per minute along the vector path of their programmed departure. The rotary axis (axes) are individually assigned a feedrate by the control so that the combined linear and rotary departure begins, and completes, simultaneously.

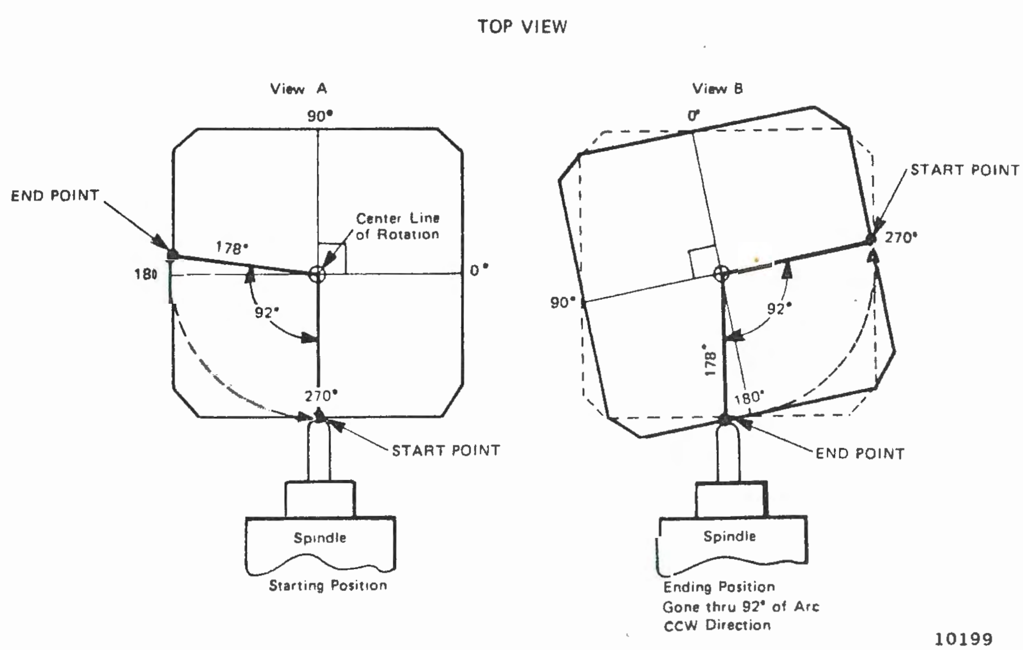


Fig. 5.28 ROTARY B EXAMPLE

ROTARY AXIS PROGRAMMING

RESTRICTIONS

Rotary Axes entries are NOT allowed in a block of circular interpolation. Examples:

1. B Axis Rotary - No Linear Departure

   B axis currently at 270°. Rotate B 92° in CCW direction. Tool tip is located 7.5" from centerline of table rotation. Desired feedrate is 4 IPM. Solve for departure distance due to B rotation.

   Departure Distance = rotary motion in radians x distance from tool tip to Pallet Centerline

   92 x 0.01745 x 7.5 = 12.0405

   $$\text{G93 F-Word} = \frac{\text{desired feedrate}}{\text{departure distance}}$$

   $$\frac{4}{12.0405} = .3322$$

Part Program:

G90G93G0G70B-178.F.3 (absolute)
G91G93G0G70B-92.F.3 (incremental)

2. B Axis Rotary with One Axis Linear Departure

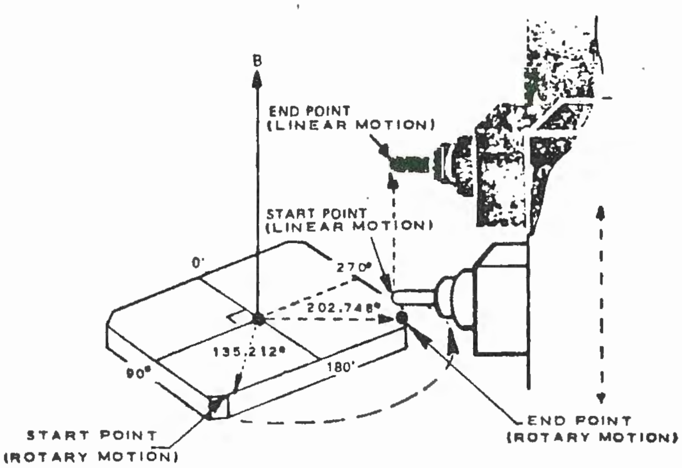


Fig.5. 29 B AXIS with ONE LINEAR AXIS

Y axis currently at 7.0000"; B axis at 135.212°. Position Y axis to 13.5321 while B rotates 67.536° CW. Tool tip is located 11.3567 from center of table of rotation. Desired feedrate is 6.3 IPM.

a. Solve for Arc Length (motion distance)

   A_L = Rotary motion in radians x distance from tool tip to table centerline:

   67.536 x 0.01745 x 11.3567 = 13.3839

b. Solve for Departure Distance

   $$\text{Departure Distance} = \sqrt{(\text{arc length})^2 + (\text{linear motion})^2}$$

   $$14.8928 = \sqrt{(13.3839)^2 + (6.5321)^2}$$

c. Solve for G93 F-Word

   $$\text{G93 F-Word} = \frac{\text{desired feedrate}}{\text{departure distance}}$$

   $$\frac{6.3}{14.8928} = .4230 \text{ (.4 rounded off)}$$

Part Program:

G90G93G0G70Y13.5321B202.748F.4 (absolute)
G91G93G0G70Y6.5321B67.536F.4 (incremental)

3. B Axis Rotary with 2 Axes Linear Departure

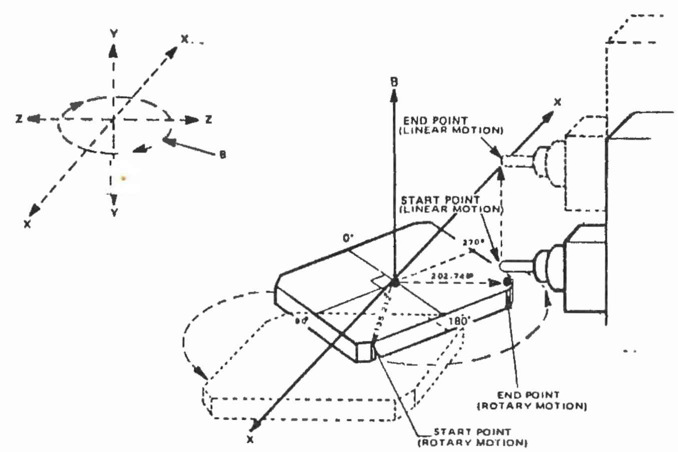


Fig.5. 30 B AXIS with TWO LINEAR AXES

X axis currently at 6.5000"; Y axis currently at 7.2500"; B axis currently at 180.000°. Program axes to end position of X8.3750 Y14.0520 B270.000. Tool tip is located, and remains, at 6.3750" from center of table rotation. Desired feedrate is 3.6 IPM.

a. Solve for Arc Length (motion distance)

A_L = Rotary motion in radians x tool tip to table CL:

90. x 0.01745 x 6.3750 = 10.0119

b. Solve for Departure Distance

$$\text{Departure Distance} = \sqrt{(\text{arc length})^2 + X^2 + Y^2}$$

$$12.2483 = \sqrt{(10.0119)^2 + (1.8750)^2 + (6.8020)2}$$

c. Solve for G93 F-Word

$$\text{G93 F-Word} = \frac{\text{desired feedrate}}{\text{departure distance}}$$

$$\frac{3.6}{12.2483} = .2939 \quad .3 \text{ (rounded off)}$$

Part Program:

G90G93G0G70X8.375Y14.052B270.F.3 (absolute)
G91G93G0G70X1.875Y6.802B90.F.3 (incremental)

FOUR-AXIS PART PROGRAMMING

When programming four-axis contouring motions, computer-assist programming is often recommended. To visualize the programming involved, one need only remember that program motions are relative to the tip of the tool passing the work surface. This motion is the departure distance and is determined using an extension of the previously described formula.

1. Solve for the Arc Length.

$$A_L B = \frac{\text{Rotary Motion in Degrees x Tool Tip to Table Centerline}}{57.29580° \text{ (1 Radian)}}$$

Note that the tool tip distance to table centerline will increase or decrease as a result of Z axis motions; however, this is taken into account by inserting Z axis motion into the formula used to determine the departure distance.

2. Solve for the Departure Distance

$$\text{Departure Distance} = \sqrt{(\text{arc length B})^2 + X^2 + Y^2 + Z^2}$$

3. Solve for the G93 Inverse Time Feedrate

Calculating an Inverse Time Feedrate is now simplified by merely dividing the desired feedrate by the departure distance.

$$\text{G93 F-Word} = \frac{\text{desired feedrate}}{\text{departure distance}}$$

FIVE-AXIS PART PROGRAMMING

As with four axis part programming, computer assist programming is recommended when programming five axis contouring motions. Again, to visualize the programming involved, one need only remember that program motions are relative to the tip of the tool passing the work surface. This motion is described as the departure distance and is determined using an extension of the previously described axis formulas.

1. Solve for Arc Length A (or C)

$$A_L A = \frac{\text{Rotary Motion in Degrees x Tool Tip to Table CL}}{57.29580° \text{ (1 Radian)}}$$

2. Solve for the Arc Length B.

$$A_L B = \frac{\text{Rotary Motion in Degrees x Tool Tip to Table CL}}{57.29580° \text{ (1 Radian)}}$$

Note that the tool tip distance to table center line will increase or decrease as a result of either X, Y and Z axes linear motions; however, this is taken into account by the X, Y and Z axes motions into the formula used to determine the departure distance.

3. Calculate the Departure Distance

$$DD = \sqrt{(\text{Arc Length A})^2 + (\text{Arc Length B})^2 + X^2 + Y^2 + Z^2}$$

Calculating the Five-Axis Inverse Time Feedrate is now a matter of dividing the desired feedrate by the departure distance, which is the total distance the tool tip must travel in passing the part.

4. Solve for the G93 Inverse Time Feedrate

$$\text{G93 F-Word} = \frac{\text{desired feedrate}}{\text{departure distance}}$$

CHAPTER 6

OFFSETS and CUTTER COMPENSATION

INTRODUCTION

The Machine Control System has five distinct types of offsets for use when programming and operating the machine. Since offsets are interactive, they require an understanding of function and application by both the part programmer and the machine operator. There are five types of offsets: G98 Absolute Axis Offsets, Fixture Offsets, and Primary, Suffix and Process Compensations.

G98 ABSOLUTE AXIS OFFSETS

G98 offsets are primarily intended to compensate for variations in the workpiece, although it may also be used to compensate for variations in fixture assembly and location. G98 offsets are usually entered in the part program, although it is possible for the machine operator to enter them using MDI procedures.

FIXTURE OFFSETS

Another type of offset possible is the Fixture Offset feature. The fixture offsets feature allows the user to enter up to 32 sets of offsets. Each of the 32 sets may contain one offset for each configured machine axis. The optional G47 command is discussed in Chapter 2. The G47 command is used to access calculated index or rotary axis offsets from the fixture offset table.

Fixture offset values are the incremental distance and direction from the programmed location to the actual absolute location.

PRIMARY TOOL COMPENSATIONS

Primary Tool Compensations are automatically called from the Tool Table whenever a tool is loaded into the machine spindle via an M06 Automatic Tool Change command. The operator may manually enable these offsets; however, this will occur only under abnormal circumstances. Primary tool compensations which are enabled via the M06 command include:

- Primary Length Compensation Values
- Primary Cutter Diameter Compensation Values

In addition to the offsets entered, Primary Tool Compensations may also be used to establish the following Tool Monitors:

- Tool Usage Time Limit
- Estimated Cutting Time
- Maximum Tool Horsepower Limit

In addition to Primary Tool Compensations, a feature called Process Compensations may also be enabled depending upon the Pallet Map currently active.

SUFFIX COMPENSATIONS

Suffix (Tool Surface) Compensations are indicated in the Tool Number Column of the Tool Table by "Decimal Point Numbers" directly beneath Primary Tool Compensations. Up to seven Suffix Comps may be entered for each tool. Suffix Compensations are enabled by programming a (UAT,n) Type II data statement after a primary tool compensation has been automatically or manually enabled.

NOTE

An optional feature whereby P numbers are used in place of the "UAT," statement is available upon customer request.

- Suffix Length Compensation Values
- Suffix Cutter Diameter Compensation

NOTE

G40 (Cutter Compensation Cancel) must be modal in order to invoke a Suffix Compensation (UAT or P).

In addition to the offsets entered, Suffix compensations, just as Primary Tool compensations, may also be used to establish the following Tool Monitors.

- Tool Usage Time Limit
- Estimated Cutting Time
- Maximum Tool Horsepower Limit

PROCESS COMPENSATIONS

Process comps alter Primary Tool compensation values associated with a particular tool. These values (in parentheses) on the Tool Table [SELECT DISPLAY] [N] are added to, or subtracted from Primary Tool comp values whenever a specific Pallet Map Number is active. A pallet number may be specified for a process comp, or the operator may set an exclusive situation, where every pallet except those named will have a Process comp associated with it. Process comps may be either Length or Cutter Diameter comps.

In addition, Process compensations may also be applied to subsets of Primary Tool compensations called Suffix (tool surface) compensations.

OFFSET DISPLAYS

Primary Tool compensations, Process compensations, and Suffix compensations are taken from the [SELECT] or [DISPLAY LIST] [N] Tool Table. A typical example of the Tool Table would appear as follows: See Fig. 6.1

Pressing [SELECT] or [DISPLAY LIST], [N] will cause the CRT to display the Tool Table. This is a comprehensive table showing all tool related data on a single display. Compensations associated with tooling are applied as soon as a tool is loaded into the machines spindle.

Fixture offsets are found on the [SELECT] or [DISPLAY LIST] [U] Fixture Offset Table. There are two pages for this table. Page 1 contains the first 16 offsets and Page 2 contains offsets 17 thru 32. (See Chapter 2 for an illustration of a fixture offset table using the optional G47 command.)

Any combination of offsets may be used when running N/C or KT/SETUP part programs. All offsets work in unison to create the desired finished workpiece. The values of any active offsets may be viewed by pressing the [SELECT DISPLAY], [O] touchswitches. See Fig. 6.3

The Axis Offset display lists current axis offsets in effect. Also listed are both the Offset Axis Position, and the Absolute Position of all machine axes. Offsets for G98, Fixture Offsets, Tool Length and Cutter Diameter Compensation, and active G88 Offsets (Boring Cycle - Orientation) are listed on this table.

If G70 is modal, the display is in inch units. If G71 is modal, the displayed entries are in millimeter units.

ENTER: __

TOOL NO.	STATUS (BCDMUW)	SOCKET NO.	PALLET NO.	LENGTH COMP.	DIAMETER COMP.	CUTTING TIME	TIME LIMIT	H.P. LIMIT
16	MW					3:23	3:20	
42	M			1.5000	.0025	3:02	3:30	
2431		3				1 :15	2:00	
			16	(.4471)	(.0500)			
2951	C O1	2					3:20	
			81	(.0100)	(.0002)			
4421		16		2.5000	.0005	1:16	2:25	1.6
4421		17		.0305				
. 1					.0005	0:15	4:20	6.0
11113		14				1:01	3:20	2.8
11113	W	20				3:38	3:20	
24113		18		.0001			3.50	
			126	(.0010)				
			**	(.0005)				
55543						2:18	3:20	

ADD RECORD	LOCATE	SPINDLE	TABLE	ENTRY	EARLY	ADVANCE
E, TOOL	LT, TOOL	S, TOOL	C,0 TABLE	# DATA	WARNING	<
#/* PALLET	LS, SOCKET	S,0 CLEAR	SL, LIST	0 CLEAR	W, 100%	NONE

Fig. 6.1 TOOL MANAGEMENT SYSTEM DISPLAY

FIXTURE OFFSETS Page 1

ENTER:

H	I (B)	X	Y	Z	B	Bi
1	<	0.0000	0.0000	0.0000	0.000	0.000
2		0.0000	0.0000	0.0000	0.000	0.000
3		0.0000	0.0000	0.0000	0.000	0.000
4		0.0000	0.0000	0.0000	0.000	0.000
5		0.0000	0.0000	0.0000	0.000	0.000
6		0.0000	0.0000	0.0000	0.000	0.000
7		0.0000	0.0000	0.0000	0.000	0.000
8		0.0000	0.0000	0.0000	0.000	0.000
9		0.0000	0.0000	0.0000	0.000	0.000
10		0.0000	0.0000	0.0000	0.000	0.000
11		0.0000	0.0000	0.0000	0.000	0.000
12		0.0000	0.0000	0.0000	0.000	0.000
13		0.0000	0.0000	0.0000	0.000	0.000
14		0.0000	0.0000	0.0000	0.000	0.000
15		0.0000	0.0000	0.0000	0.000	0.000
16		0.0000	0.0000	0.0000	0.000	0.000

Clear	Clear Column	Clear Entry	Assign	Advance	Table
C,0 table	C,I implicit	-0 implicit	I,<axis>	<	LD, load
0 line	C, <axis>	0 offset	I,- none	none	SV, save

Fig. 6.2 SELECT DISPLAY U - FIXTURE OFFSETS

	X	Y	Z	B
	-----------	-----------	-----------	-----------
POSITION	15.0000	10.5000	30.0000	92.998
G98 OFFSET	0.0000	0.0000	0.0000	0.000
H00 OFFSET	0.0000	0.0000	0.0000	0.000
LENGTH OFFSET			0.0000	
ABSOLUTE POS.	15.0000	10.5000	30.0000	92.998
DIAMETER OFFSET	0.0000	0.0000		
G88 OFFSET	0.0100 (0.0100)	0.0000 (0.0000)		

Fig. 6.3 SELECT DISPLAY O - AXIS OFFSET

CALCULATED and SPECIFIC FIXTURE OFFSETS

Functional Description:

In addition to the standard form of fixture offsets, Kearney & Trecker has now added the ability to calculate fixture offsets for rotary axis positioning. This means that a fixture offset can be established for a particular rotary or index axis position. That offset will then be calculated and applied to any other index or rotary axis position. It is important to note that the calculated fixture offset feature is only applied when the rotary axis is used for positioning or indexing, it is not applied dynamically during machining.

With the advent of calculated fixture offsets, new Preparatory functions (G-codes) have been added to the control and machine features:

G39 Specific Fixture Offsets

Specific fixture offsets are the offset values that programmers and operators have worked with in the past. Fixture offsets are established for each rotary or index axis position. This done thru any of the following methods; the operator entering the fixture offset values thru the MDI keys into the Fixture Offset display, or thru FXC or UFX Type II data statements. The G39 is modal upon control startup and whenever G47 is not present.

G47 Calculated Fixture Offsets (Option)

G47 puts the control in a mode that will calculate the fixture offset for any index or rotary axis position. To use this feature the programmer or the operator must first establish a fixture offset within the fixture offset table, along with initial rotary axis positions. This may be done using the methods shown above. The control will recognize and appropriately adjust the fixture offsets as the rotary axis is repositioned, as long as G47 remains in effect.

Fixture Offset Display

The standard fixture offset table has been changed to reflect the rotary axis position at which the initial offsets were established.

An Ai, Bi and Ci column have been added to the display. These columns reflect the rotary positions for the initial axes offsets. In the example above, when the B axis as at 20° position, the offset is X-.500, Z-.7500, and -3.000 in B. If G47 is in effect and the B axis is commanded to move 110°, the calculated offsets would be X.75 and Z-.500. The calculated fixture offset values could be observed on the Axis Offset display [SELECT] [O] or [DISPLAY LIST] [O].

Establishing Calculated Fixture Offsets

Fixture offsets are constructed either by the operator thru the MDI panel or by the programmer using the proper FXC or UFX Type II Data Statement format.

G48 Explicit Mode or G49 Implicit Mode

If the program is to run the Explicit Mode G48 the program must include the specific H-number to bring the desired offset into effect. Alternatively, if G49 is in effect, the chosen axis must be commanded to the proper position to call the desired offset into effect.

Fixture Offset Priority

Specific fixture offsets are modal upon control startup and are in effect whenever G47 is not in effect. G39 is canceled by G47.

				(Or	*A*	*B*	*Ai*	*Bi)*
H	I(B)	X	Y	Z	B	C	Bi	Ci
1	45.00	-0.5000	0.000	-.7500	-3.000	0.0000	0.0000	20.000

Fig. 6.4 FIXTURE OFFSET DISPLAY with CALCULATED FIXTURE OFFSETS and OPTIONAL AXES

Calculated fixture offsets, G47, is in effect only when the preparatory command G47 has been called out in the part program or by manual data input.

G47 is canceled by:

G39 or [CLEAR ALL LOGIC]
Sequence Number Search
M02 or M30 end of program command.

G98 ABSOLUTE AXIS OFFSETS

The G98 offset command is programmed by writing a G98 in a block containing the letter address and desired position of the axis or axes to be offset. The block with the G98 command is a no motion block. *No other commands are allowed in this block.*

Functional Description:

The G98 command is also referred to as "Zero Set", "Zero Offset" or "Zero Preset". This command allows the programmer to offset the zero position of one or more configured machine axes with respect to the machine tool zero reference position. Upon G98 command execution, the displayed axis position on the Main Display on the control update to show the axis position as offset.

Programming Applications:

1. May be used to indicate the centerline of a finished inside or outside diameter for accurate referencing from that point. This application requires either a physical measurement by the machine operator or if the the machine is equipped with the optional Spindle Probe feature may be automatically located using the G27 Probe Cycle.

2. May be used to execute a part processing cycle utilizing a part tape which was written for a machine tool with a different length of axis travel.

3. May be used to compensate for inaccuracies or inconsistencies in workpiece fixture location. This however is usually accounted for by fixture offsets.

Programming Instructions:

1. Position the axes to a specific absolute position required by the part program. This may done by a manual axis jog, or by a programmed command. Ensure that any previously entered G98 command has been canceled with a G99 Command.

Program the code G98 and in the same data block, the axis letter address word(s) and the desired axis position coordinate required for each axis to be offset.

Programming Considerations:

1. G98 may be used in conjunction with any, or all, of the other available offsets.

2. G98 may be bypassed on a per-block basis by programming an M25 command in a block along with a command for desired absolute axis position.

3. If programming in incremental mode (G91) with a G98 offset in effect and a bypass axis block M25 is programmed, the axes must be first be repositioned in absolute mode (G90) before continuing in incremental mode.

4. An automatic tool change command (M06) forces an M25 command for the tool change block. This positions the axes to the absolute position required for the tool transfer. In the next block the G98 offset is again in effect.

Restrictions:

1. Do not program any of the following commands in a G98 block:

   M Codes .... (G98 is a no motion command)
   G Codes .... (Other than G98)
   R, E .... (Used only in Fixed Cycles)
   F .... (G98 is a no motion command)
   S .... (G98 is a no motion command)
   I, J, K ... (Used only in circular interpolation)

The only commands appearing in a G98 Block should be desired axis positions. This includes the following letter addresses, X, Y, Z, A, B and C, along with desired axis positions.

Cancellation:

The G98 command is canceled by programming a G99 command. It is recommended that the G99 be programmed in a block by itself.

Example:

1. To establish bore centerline as X0 Y0 by moving to center:

N170G90X10.Y9.F0 .... (move to bore centerline)
N180G98X0Y0

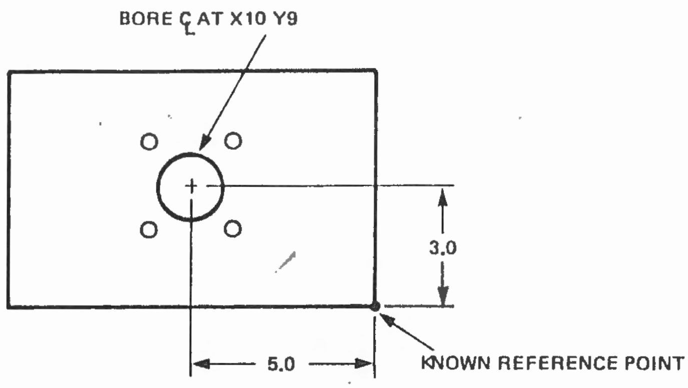


Fig 6.5 G98 OFFSET EXAMPLE

2. To establish bore centerline as X0Y0 by moving to a known reference point:

N200G90X15.Y6.F0 .... (move to reference point)
N210G98X5.Y-3. .... (Establish offset)
N220X0Y0F0 .... (Position to bore centerline)

TOOL MANAGEMENT SYSTEM

The Tool Management System incorporates all tool related data into a single table accessed by pressing either [SELECT] [N] or [DISPLAY LIST] [N]. It is from this table that compensation related data for Primary Tool compensations, Process compensations and Suffix compensations is obtained.

PRIMARY TOOL COMPENSATIONS

Tool Length and Cutter Diameter compensations may be entered for Primary Tool compensations. The range of compensations is from ±0.0001 in (±0.001 mm) to ±39.3700 in (±999.999 mm). If no values are entered into the primary compensation positions, the tool is assumed to be the correct length and diameter.

PROCESS COMPENSATIONS

Modifications to a Primary Tool compensation value for Tool Length and Cutter Diameter compensation values may be entered as Process compensations. The range of process compensations is from ±0.0001 in (±0.001 mm) to ±0.3936 in (±9.999 mm). If no values are entered in the process compensation positions, the tool is assumed to be the correct length and diameter.

Process comps are activated whenever the pallet number at the top of the RUN list on the Program Management Table [ASSIGN PROGRAM] matches a process compensation pallet number for the tool currently in the machine spindle.

Process comps may be entered to provide a specific offset for all pallets *except* those which are listed. This exclusive case is enabled by entering an asterisk (*) in the Pallet Number column for the specific tool.

SUFFIX COMPENSATIONS

Suffix compensations are considered subsets of Primary Tool compensations. When entered they are entered as follows:

E, Tool Number - Decimal Point - Comp. Number

e.g. E,97841.5

Up to seven Suffix compensations may be entered for each tool listed on the Tool Table provided the table does not exceed 200 lines of data.

Suffix compensations are applied first by loading the required tool into the machine spindle using an M06 command. When loaded in this manner the Primary Tool compensation is enabled. To enable the Suffix compensation, the programmer enters the command:

(UAT,Suffix Compensation Number)

e.g. (UAT,5)

This command cancels the primary tool compensations and enables the specified Suffix Compensation.

Suffix compensations allow for compensations in the same manner as Primary Tool compensations. When the programmed UAT command is executed, the Primary compensations are removed, and the Suffix compensations are substituted in its place. The range of Tool Surface compensations is the same as Primary Tool compensations. This is in a range from ±0.0001" (±0.001 mm) to ±39.3700" (±999.999 mm). If no values are entered in the Primary comp positions, the tool is assumed to be the correct length and diameter.

The following two sections describe features associated with the tooling compensations, Length compensation and Diameter compensation.

LENGTH COMPENSATION

Tool Length compensation is accommodated for by entering a value for Length comp in the Tool Table.

Functional Description:

Tool Length comp. is a means of compensating for deviations between the programmed set length of a tool and the actual length of the tool used in the part program. The compensation value is entered into the Tool Table by the operator or by tape. The control governs the Tool Length compensation feature by adjusting the Z axis slide in the positive or negative direction by the value entered into the Tool Table.

Programming Instructions:

1. If part surface programming format is used, the value entered in the Tool Table for Length comp is equal to the set length of the tool being used.

2. If programming for preset tools, no value need be entered in the Tool Table for Length comp unless there is a deviation between the 'programmed' set length and the actual set length of the tool used. The value then entered is equal to the signed incremental difference between the two lengths.

3. If programming Z to the positive limit of travel (for clearance, etc.), insert an M25 command in the block to eliminate the possibility of overtravel if the compensation value is a positive number.

Considerations:

1. Length comp becomes effective in the data block containing the tool transfer command (M06) which inserts the tool into the spindle or indicates to the machine operator to change the manual tool.

2. A positive compensation value indicates that the actual tool is longer than the programmed tool. The Z slide adjusts by moving the spindle face farther from the part surface.

3. A negative compensation value indicates that the actual tool is shorter that the programmed tool. The Z slide adjusts by moving the spindle face closer to the part surface.

4. If the value of the length compensation for a tool is equal to zero, no compensation is applied.

DIAMETER COMPENSATION

Diameter compensation is an offset which is used to compensate for deviations between the programmed diameter of a milling cutter and the diameter of the actual tool which is used in the part processing cycle. This deviation may be the entire tool radius if the part surface programming format is used, or it may be the the incremental difference between the programmed tool and the actual tool if the program is written for preset tooling. Diameter compensation is turned on by programming a G41 CUTTER LEFT or G42 CUTTER RIGHT command, and is turned off by programming a G40 command. 41 and G42 conditions for both inside and outside cuts are shown in Fig. 6.6.

Cutter Diameter compensation is accommodated for by entering a Diameter comp value into the Tool Table for a particular tool. This may be done by the operator or by a programmed command using the Type II data statement (TLD,). If the part programming surface format is used, the entire tool diameter is entered into the Tool Table. If the program is written for preset

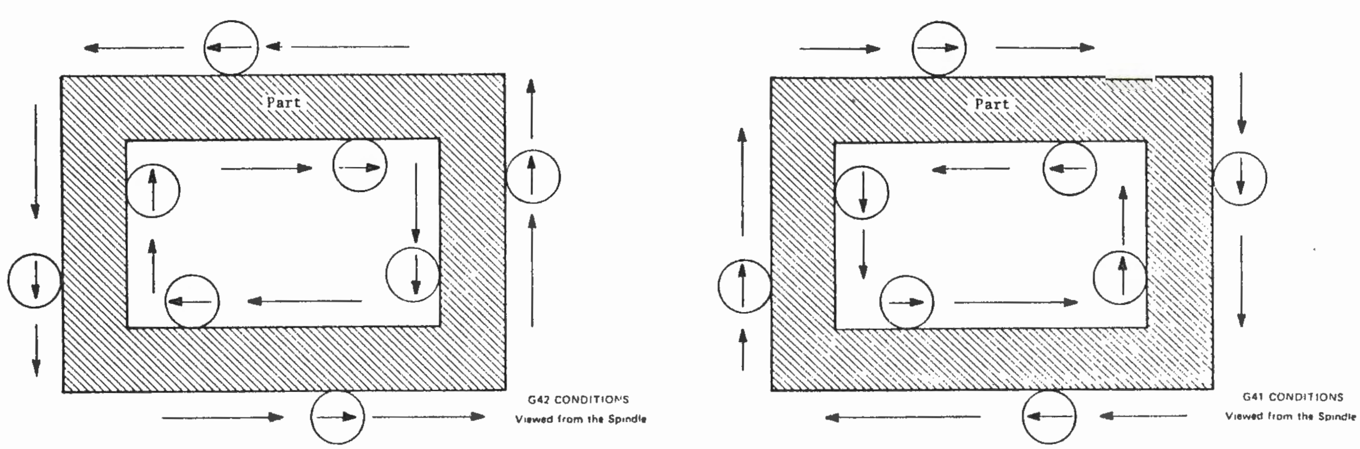


Fig. 6.6 G41 and G42 CONDITIONS

tools, only the signed incremental difference between the tools is entered. A positive entry indicates that the actual tool is larger than the programmed tool, a negative entry indicates that the actual tool is smaller in diameter than the programmed tool.

Functional Description:

When it is active during a part processing cycle, the Cutter Diameter comp feature adjusts the X and/or Y axis slide in the G17 plane command (the Z axis slide is positioned in the G18 or G19 plane command) to place the, edge of the milling cutter in the correct position for metal removal. It also "looks ahead" into the part program to determine upcoming cutter paths, and can detect the presence of comp errors in advance.

When the part program has been assigned and is being run, the control reads ahead into the program and analyzes upcoming data blocks. If the programmed path is to be compensated by the Cutter Diameter compensation feature, as indicated by a G41 (CUTTER LEFT) or G42 (CUTTER RIGHT) command, the cutter is adjusted at 90° to the next programmed path. During successive data block execution the control continues looking one block ahead into the program to guard against possible programming errors.

Cutter Diameter compensation is used for inside cuts on either linear (single or multiaxis) circular or helical contours or for rotary axis machining.

Considerations:

1. For part surface programming, it is important that the approximate diameter of any milling cutter used in the program, is known in advance. This helps to prevent collisions because of insufficient clearance, to protect the integrity of the part contours and to prevent Cutter Comp Errors.

2. Cutter Diameter compensation may be used in conjunction with any, or all, of the other available compensation modes (G98, Tool Length comp, Fixture Offset).

Programming Instructions:

1. Where a contour milling operation is required, determine the approximate diameter of the milling cutter needed to machine the part, keeping in mind any necessary clearance between the cutting tool and fixture clamps or projections of the workpiece contours.

2. Program a "ramp-on" block to turn the Cutter Diameter compensation feature on. The ramp-on block is the block which contains the G41 or G42 command and must be applied regardless of whether the programmed path is an inside or outside cut. At the beginning of the ramp-on block no compensation is in effect, at the end point of the ramp-on block all of the compensation (the tool radius) is in effect.

The ramp-on block must be a motion block which contains a minimum axis departure of 0.0001" in the X and/or Y axis. A Z axis command may also be included in this block if desired.

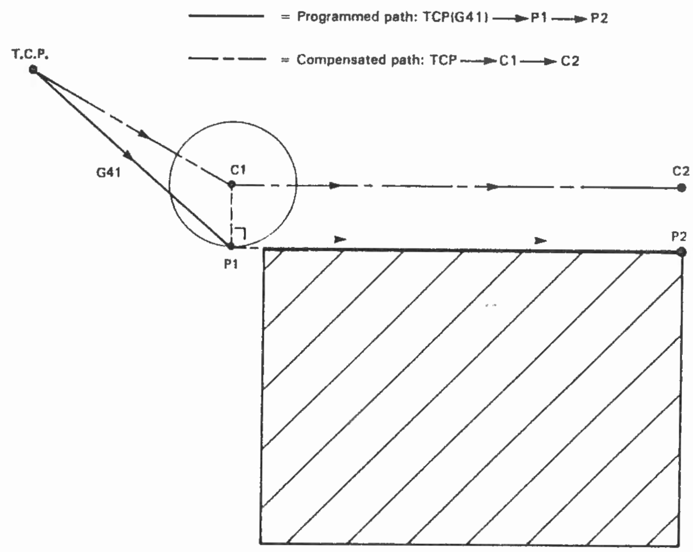


Fig. 6.7 RAMP-ON from TOOL CHANGE POSITION (Outside Cut to Linear Path)

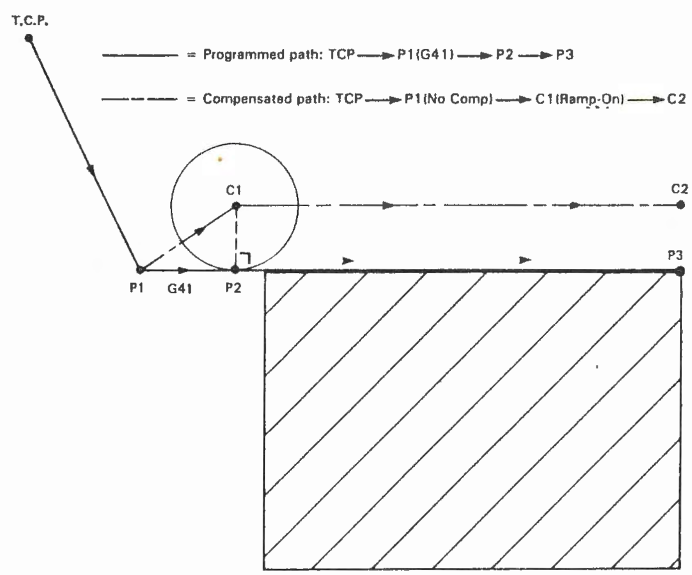


Fig. 6.8 SHORT RAMP-ON Outside Cut to Linear Path

A. For outside cuts to a linear path, the easiest method is to program a G41 or G42 command in the block which moves the milling cutter to the clearance point near the part surface. The ramp-on block *needn't be an extension of the programmed path* for a linear cut (Figures 6.7, 6.9, 6.11).

An alternate method involves positioning the cutter close to the part surface with no compensation in effect, then turning the compensation on is a short movement toward the workpiece. The cutter must not make contact with the workpiece at the end of this movement. (Figures 6.8, 6.10, 6.12).

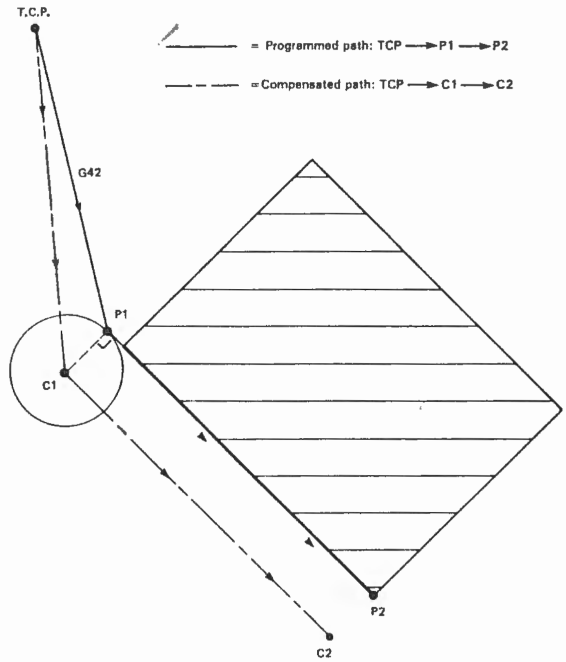


Fig. 6.9 RAMP-ON from TOOL CHANGE POSITION
(Outside Cut to Linear Path)

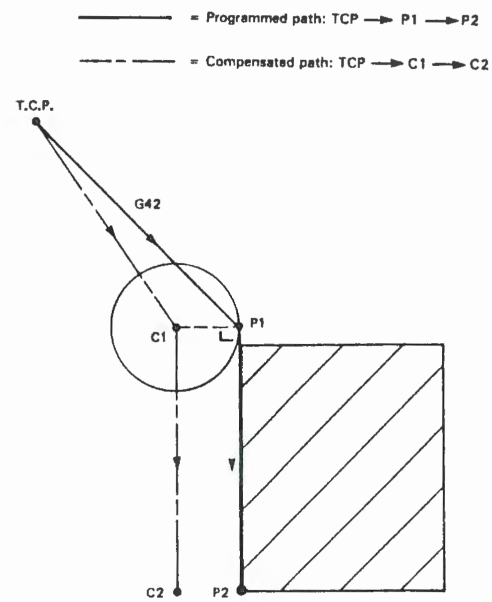


Fig. 6.11 RAMP-ON from TOOL CHANGE POSITION
(Outside Cut to Linear Path)

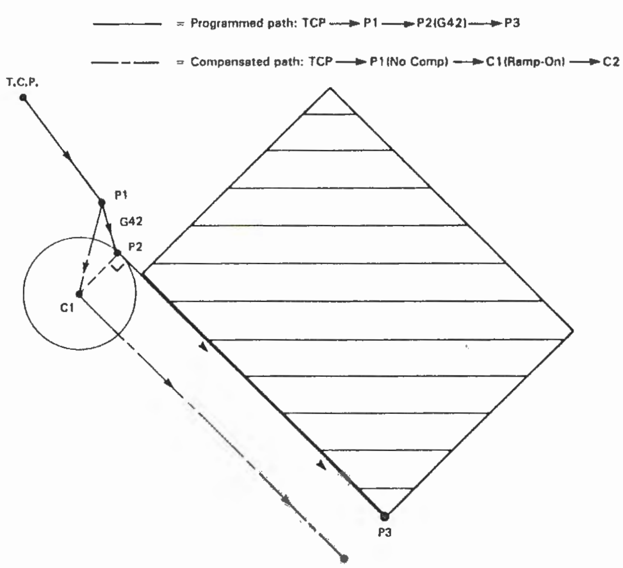
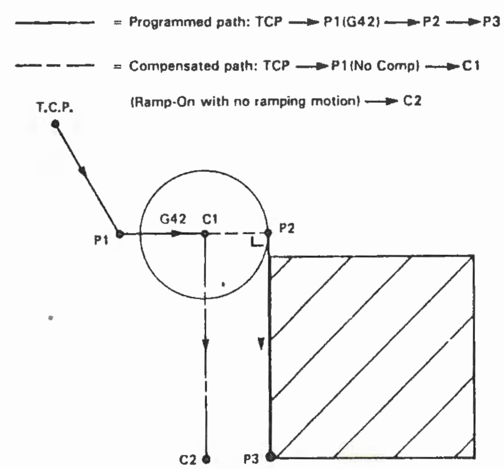
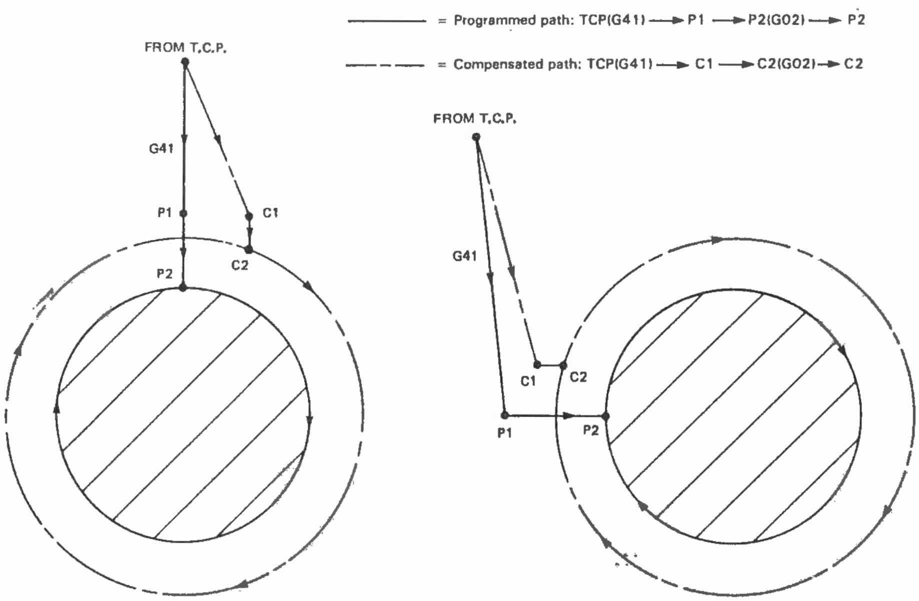


Fig. 6.10 SHORT RAMP-ON
(Outside Cut to Linear Path)



PROGRAMMED PATH IS FROM T.C.P. TO P1, THEN TO P2 THRU to P3. RAMP-ON (G42) IS PROGRAMMED IN MOVE FROM P1 TO P2. NOTE THAT COMPENSATED PATH IS IDENTICAL TO PROGRAMMED PATH UNTIL CENTER OF CUTTER REACHES C1. NO RAMPING MOTION IS NECESSARY BECAUSE AT C1 THE CUTTER IS AT 90 DEGREES TO THE NEXT PROGRAMMED PATH (P2 TO P3).

Fig. 6.12 SHORT RAMP-ON
(Outside Cut to Linear Path)



TWO EXAMPLES OF RAMP-ON FROM TOOL CHANGE POSITION TO NON-TANGENT POINT ON CIRCULAR PATH. POINT P1 MUST BE GREATER THAN TOOL RADIUS FROM PART SURFACE.

Fig. 6.13 RAMP-ON from TOOL CHANGE POSITION to NON-TANGENT POINT on a CIRCULAR PATH

B. For outside cuts to a circular path, one method is to program the ramp-on in the move from the tool change position to a clearance position next to the part surface. The clearance distance will depend upon the end point of the next programmed path:

If the end point of the next programmed path is a point on the circle which is not tangent, then that clearance position must be a distance greater than the radius of the cutter to be used. A Cutter Comp Error C001 or C002 results if the clearance is less than the cutter radius. See Figure 6.13.

If the end point of the next programmed path is a point on the circle and tangent in the direction of the cut, then that clearance position is not restricted by cutter radius. (See Figure 6.14.)

An alternate method is to program the ramp-on in a short move from a clearance position toward another clearance position tangent to the part surface. The cutter does not contact the part surface at the end of this move (See Figure 6.15A). The distance to allow for the clearance from the uncompensated position to the part surface depends upon the cutter radius.

Or you can also program the ramp-on in a short move from a clearance position to the part surface. The cutter makes contact with the part surface at the end of this movement. The distance to allow between the clearance position an the point on the part surface must be a distance greater than the radius of the cutter to be used. A Cutter Comp Error 0001 or 0002 results if the clearance distance is less than the cutter radius. See figure 6.15B.

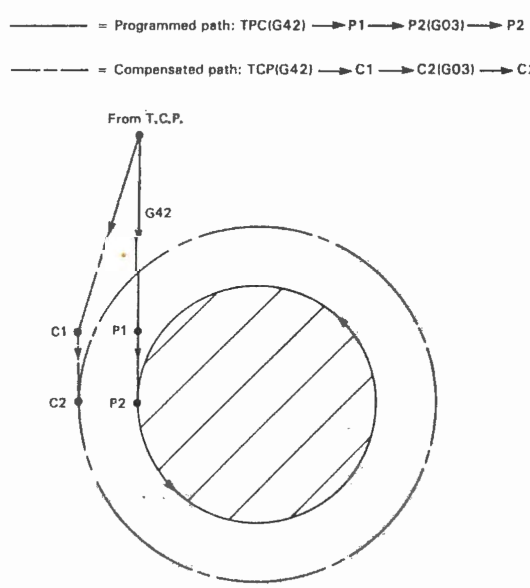


Fig. 6.14 RAMP-ON from TOOL CHANGE POSITION (to Tangent Point on a Circular Path)

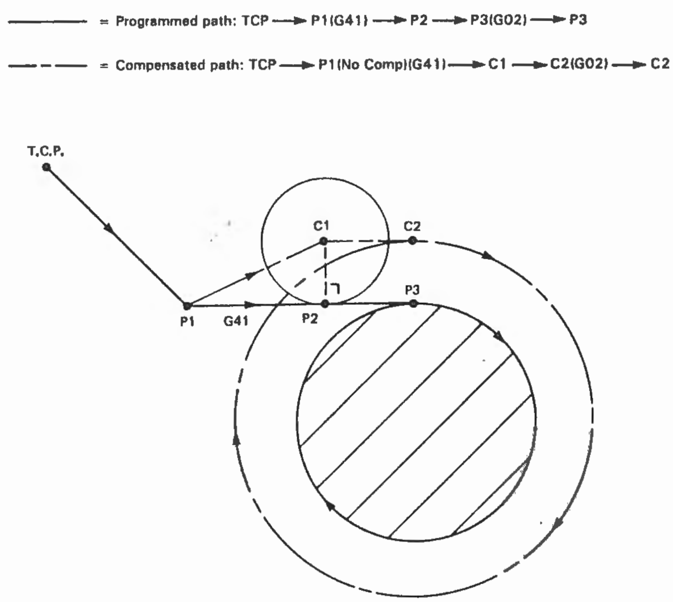
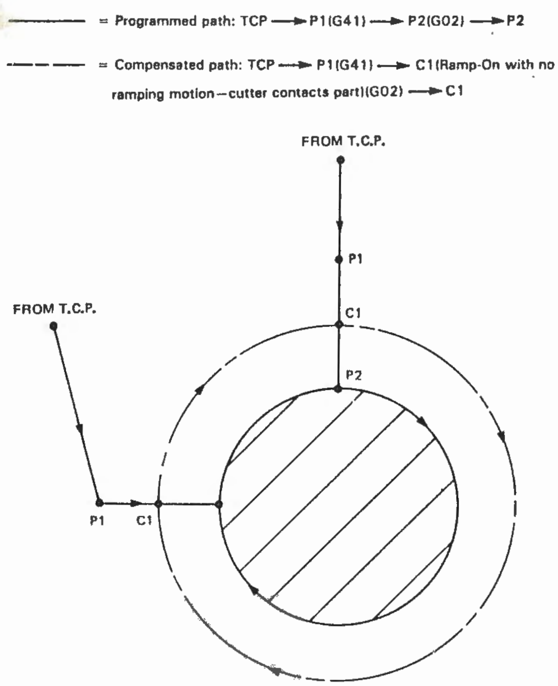


Fig. 6.15A Short Ramp -on to a Circular Path on a Outside Cut



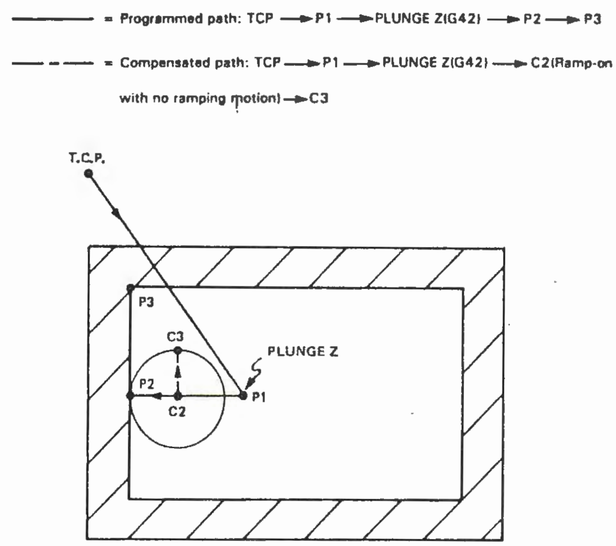
TWO DIFFERENT CONDITIONS SHOWN. RAMP-ON OCCURS WITH NO RAMPING MOTION. CLEARANCE POSITION MUST BE GREATER THAN TOOL RADIUS.

Fig. 6.15B Short Ramp -on from a Clearance Position to a Non-tangent Point on the Part Surface

C. For inside cuts to either a linear or circular path, the recommended procedure is to position the cutter at a clearance position above the part with no comp in effect, then plunge Z to the required depth. Make the ramp-on block a single axis move (either X or Y) to the part surface. In this way the ramp-on requirement is fulfilled with no ramping motion and the cutter is aligned at 90° to the next programmed path. (Figures 6.16, 6.17, 6.18.)

3. Program a G40 command to turn the Cutter Diameter compensation feature off.

A. The G40 command must precede the next tool change command. An automatic tool change cannot be programmed with G41 or G42 in effect.



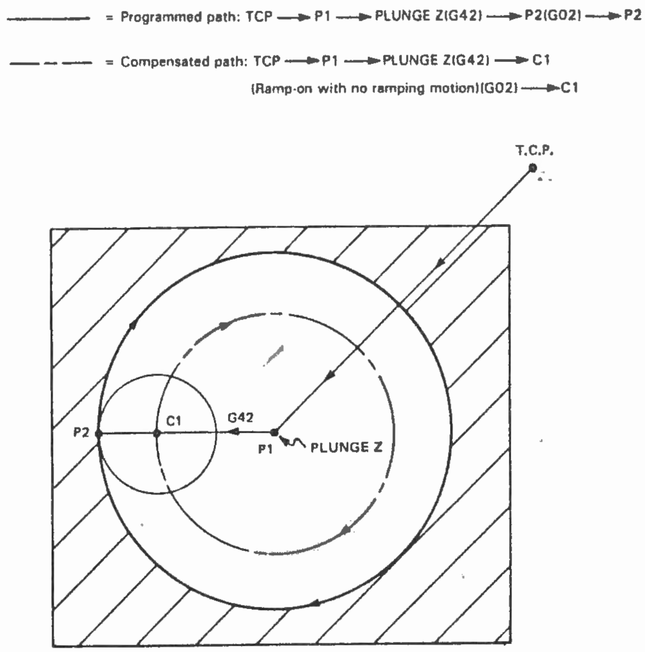
NOTE:

FOR INSIDE CUTS TO A LINEAR PATH, PROGRAM THE RAMP-ON BLOCK AS A SINGLE AXIS MOVE TOWARD THE WORKPIECE.

Fig. 6.16 Short Ramp -on (Inside Cut to Rectangular Path)

B. It is NOT necessary to program and axis departure in order to turn Cutter Diameter compensation off. If no axis departure command is programmed with the G40 command, the cutter remains at its current position (i.e. cutter doesn't automatically move to to the uncompensated position). However, the axis register displays on the CRT are updated to show the uncompensated position.

C. If an axis departure command IS programmed with the G40 command, and if the departure command moves the cutter to point which is tangent to the previous block, the compensation is canceled in a "ramping" motion. Compensation is

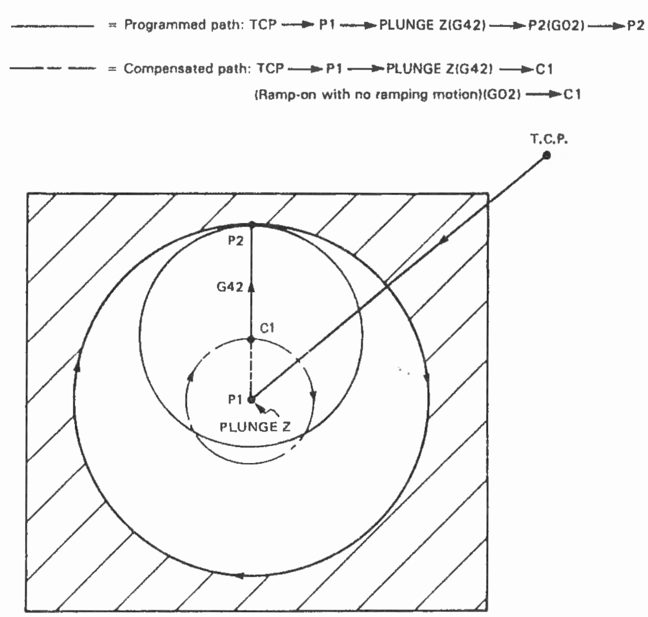


NOTE:

FOR INSIDE CUTS TO A CIRCULAR PATH, PROGRAM THE RAMP-ON BLOCK AS A SINGLE AXIS MOVE TOWARD THE WORKPIECE.

Fig. 6.17 Short Ramp -on (Inside Cut to Circular Path)

removed gradually along the departure path until the end point of the axis departure has been reached. When the end point has been reached, all of the compensation will have been removed. See Figure 6.19A



NOTE:

CUTTER DIAMETER IS LARGE IN RELATION TO HOLE DIAMETER.

Fig. 6.18 Short Ramp -on (Inside Cut to Circular Path

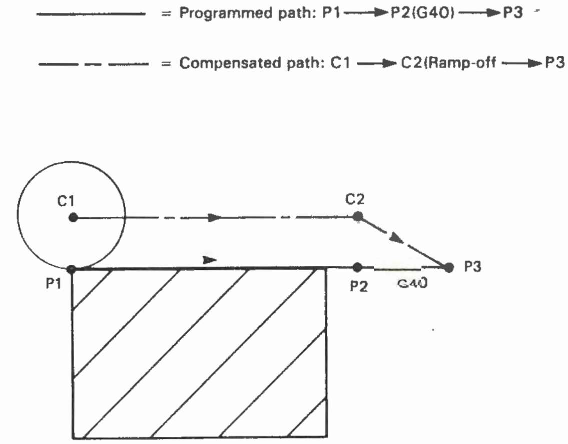


Fig. 6.19A Ramp-off Block Tangent to Previous Path

If an axis departure command IS programmed with the G40 command, and if the departure command moves the cutter directly from the end point of the previous compensated path to a position commanded in the G40 block. See Figure 6.19B.

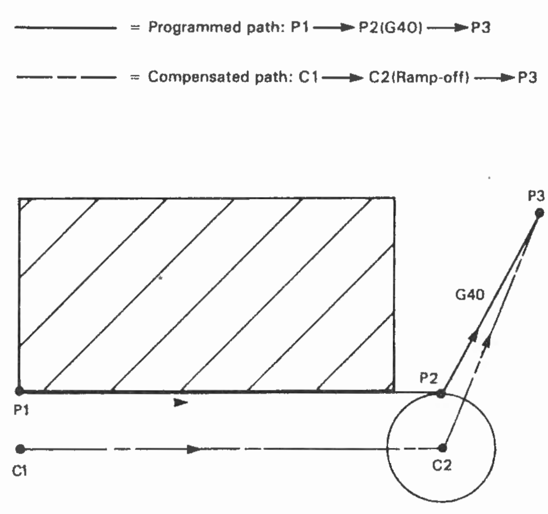


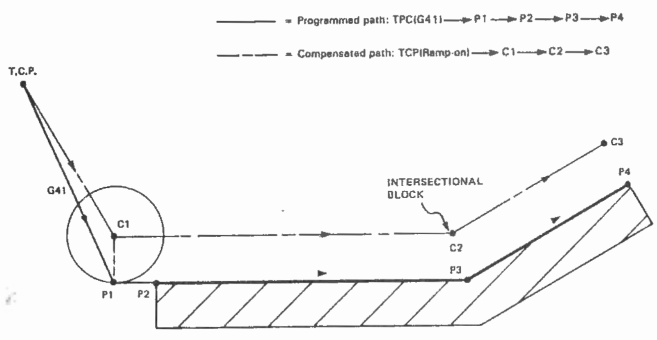
Fig. 6.19B Ramp-off Block Non-Tangent to Previous Path

D. For conditions where the same milling cutter will be used for both a periphery cut, and then for additional milling of a part contour, the comp should be turned OFF after the periphery cut then restated prior to the additional milling operation.

Compensated Cutter Paths

The following drawings and the accompanying text illustrate both the programmed cutter path and the compensated cutter path which results through the Cutter Diameter compensation feature. Several different metal cutting conditions are represented.

1. Compensated cutter paths which change direction, to move toward the compensation, result in "Intersectional Blocks".



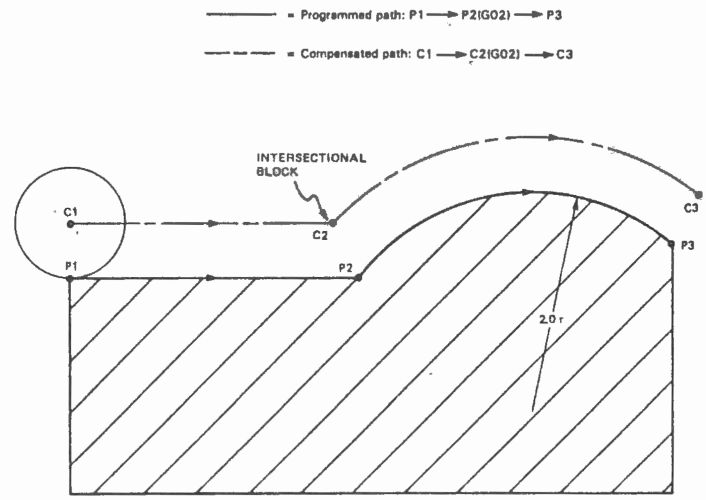
COMPENSATED CUTTER PATH—LINEAR TO LINEAR SHOWING RAMP-ON FROM TOOL CHANGE POSITION. (DIRECTION CHANGE TOWARD COMPENSATION)

Fig. 6.20 Compensated Cutter Path-Linear to Linear (Direction Changes Towards Compensation)

If the compensated path is left of the programmed path (i.e. G42) and the next block requires a left turn, or the compensated path is on the right of the programmed path (i.e. G42) and the next block requires a right turn, then an "intersectional block" results. An intersectional block occurs when the intersection of the two compensated paths is reached by the center of the cutter. See Figures 6.20, 6.21 & 6.22.

2. Compensated cutter paths which change direction, to move away from the compensation result in "Fill-In-Blocks".

If the compensated path is left of the programmed path (i.e.) G41 and the next block requires a right turn, or if the compensated path is on the right of the programmed path (i.e. G42) and the next block requires a left turn, then a "fill-in" block results. A fill-in block consists of a circular arc of the radius

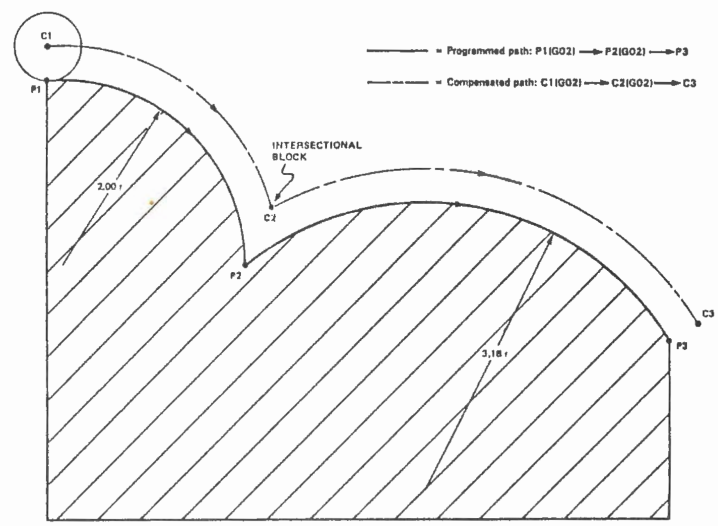


COMPENSATED CUTTER PATH—LINEAR TO CIRCULAR. (DIRECTION CHANGE TOWARD COMPENSATION)

Fig. 6.21 Compensated Cutter Path-Linear to Circular.(Direction Changes Towards Compensation)

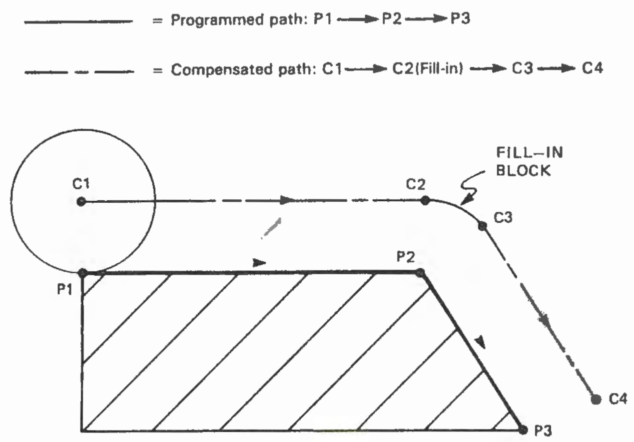
equal to one half of the diameter compensation. It is centered on the intersection of the programmed paths and smoothly connects the two compensated cutter paths. See Figures 6.23, 6.24 & 6.25.

In certain applications it is desirable to bypass fill-in blocks generated by the look-ahead cutter diameter compensation feature. This is accomplished by programming a G46 Bypass Fill-in Blocks command in the same line with the G41 or G42 used to invoke cutter diameter compensation. The G46 will remain in effect until the next executed G41 or G42 command, at which time it will be canceled unless reprogrammed in the same block.



COMPENSATED CUTTER PATH—CIRCULAR TO CIRCULAR. (DIRECTION CHANGE TOWARD COMPENSATION)

Fig. 6.22 Compensated Cutter Path- Circular to Circular.(Direction Changes Towards Compensation)

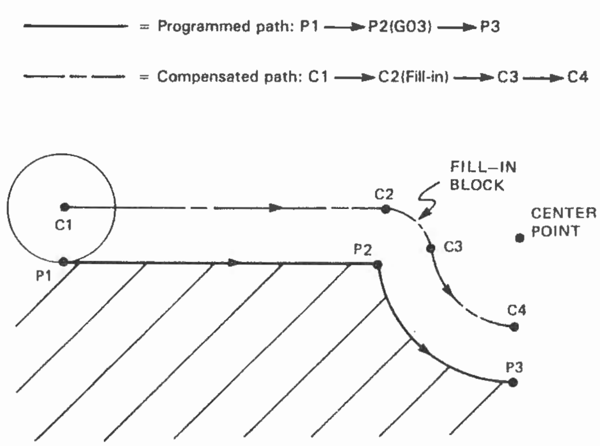


COMPENSATED CUTTER PATH—LINEAR TO LINEAR SHOWING FILL-IN BLOCK. (DIRECTION CHANGE AWAY FROM COMPENSATION)

Fig. 6.23 Compensated Cutter Path-Linear to Linear (Direction Changes Away from Compensation)

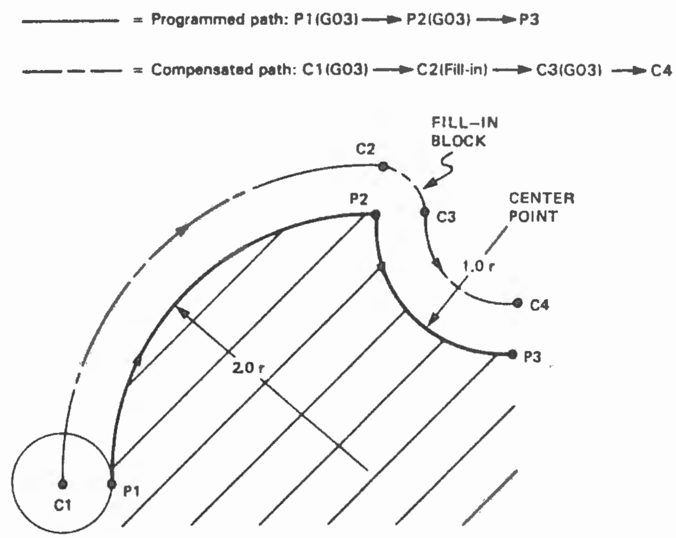
The programmed cutter compensation will be executed ignoring internally generated fill-in blocks.

G46 is canceled upon execution of the next commanded G41 or G42, or M02, M30 or manual operation of [CLEAR ALL LOGIC].



COMPENSATED CUTTER PATH—LINEAR TO CIRCULAR SHOWING FILL-IN BLOCK. (DIRECTION CHANGE AWAY FROM COMPENATION)

Fig. 6.24 Compensated Cutter Path-Linear to Circular.(Direction Changes Away from Comp)



COMPENSATED CUTTER PATH—CIRCULAR TO CIRCULAR SHOWING FILL-IN BLOCK. (DIRECTION CHANGE AWAY FROM COMPENSATION)

Fig. 6.25 Compensated Cutter Path Circular to Circular.(Direction Changes Away from Comp)

Cutter Compensation Errors

The "numbered" Cutter Compensation Errors cause an interruption in the part processing cycle by generating a [CYCLE STOP] condition and issuing a message on the [MESSAGE] display:

Cutter Comp Error 0001
Intersection error in previous block

Cutter Comp Error 0002
Intersection error in current block

Explanation:

Both Compensation Error 0001 and 0002 are the result of two simultaneous conditions: a large compensation value combined with a short programmed path which must turn in a direction which is toward the compensation. As a result of these conditions either the current look ahead block or the previous look ahead block would require the cutter to run in a direction which is opposite to its programmed path in order to reach the logical end point. The compensated path segments in a "intersectional block" do not actually intersect. See Figure 6.26. The control is placed in a [CYCLE STOP] state and a message "Cutter Compensation Error 0002" is generated.

NOTE

1. If the programmed path is in the direction of the arrows in Figure 6.26

P1 -------> P2 -------> P3 -------> P4:

Point C2 represents the compensated position which corresponds to the programmed Point P2. In order for the compensated cutter path to be at 90 degrees to the next programmed path (which is from P2 to P3) the cutter would have to move from C2 to C3 - which is opposite the programmed direction.

2. If the programmed path is opposite the direction of the arrows in Figure 6.26

P1 <------- P2 <------- P3 <------- P4:

Point C4 represents the compensated position which corresponds to the programmed Point P3. In order for the compensated cutter path to be at 90 degrees to the next programmed path (which is from P3 to P2) the cutter would have to move from C4 to C3 - which is opposite the programmed direction.

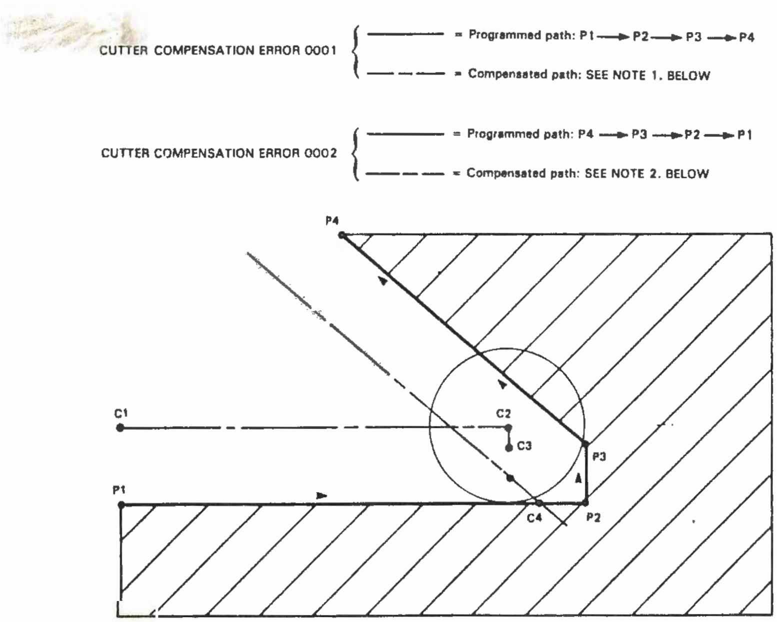


Fig. 6.26 CUTTER COMP ERRORS 0001 and 0002

Correction:

Edit the lines in the program which contain the intersection error, or substitute a smaller diameter milling cutter for the one currently being used.

This error is most likely to arise if the approximate diameter of the cutting tool to be used in the program was not accurately determined in advance of actually running the part program.

Cutter Comp Error 0003
Not enough blocks on ready list

Explanation:

WHILE RUNNING THE PART PROGRAM, there must be a minimum of five "ready blocks" in order for the block containing the G41 CUTTER LEFT or G42 CUTTER RIGHT command to be transferred to "active". The reason for this rule is linked to the "look ahead" feature. An error caused by neglecting this rule is most likely to result if the part processing cycle is interrupted, and an attempt is made to enter additional commands via the MDI feature. The control is placed in [CYCLE STOP] state and a message "Cutter Diameter Compensation Error 0003" is generated.

Correction:

If manually entering data via the MDI feature, make certain that at least five data blocks are in "ready" before depressing the [CYCLE START].

Cutter Comp Error 0004
Comp too large for inside circle diameter

Explanation:

If a circle or an arc is programmed with a diameter less than the amount of diameter comp, an "INPUT SOURCE ERROR" message appears in the Ready Buffer on the [MAIN] display. In addition the control generates a [CYCLE STOP] and issues the message on the [MESSAGE] display. This error is most likely to arise if the approximate diameter of the cutting tool to be used in the program was not accurately determined in advance of actually running the part program.

The control protects the integrity of the workpiece by preventing inadvertent undercutting of the part surface due to the use of a cutter with too large a diameter for the programmed contour.

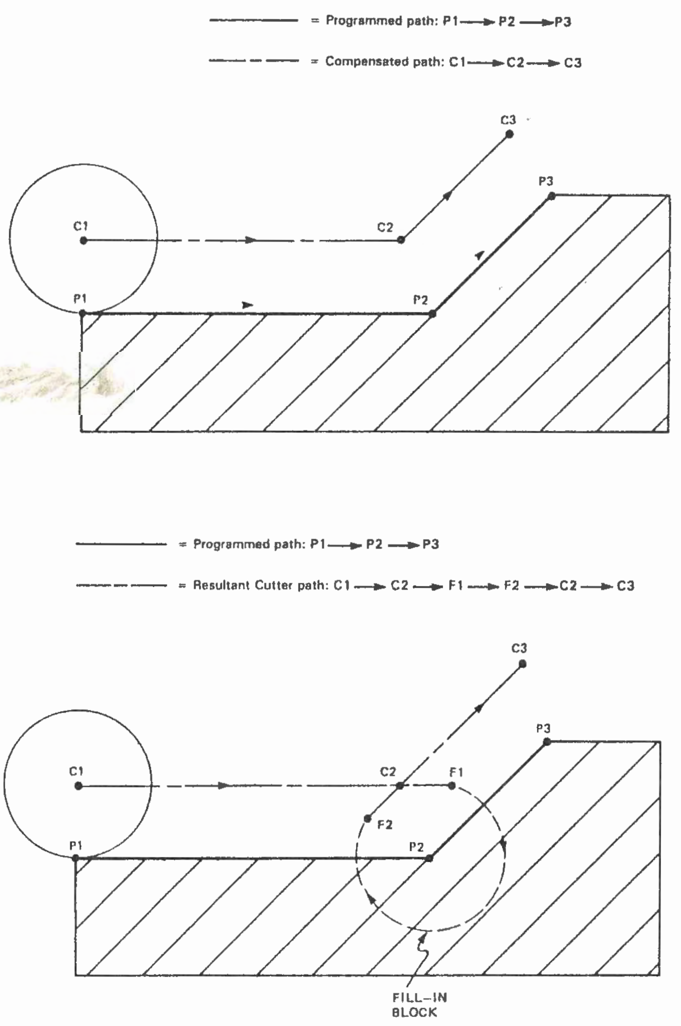
Correction:

The operator should replace the oversize tool with one of the correct size and enter the correct cutter diameter compensation into the tool table. If the tool in the spindle is the correct size as specified by the programmer, the program itself may be in error and the programmer should examine the part program.

Too many NO MOTION blocks during compensated path programming.

Explanation:

This Cutter Comp Error is more accurately classified as a restriction than as an error but does issue issue a [CYCLE STOP] and a "Cutter Compensation 0003" error message. Since the Cutter Diameter compensation feature looks ahead into the part program to determine upcoming cutter paths, you must make certain that the program contains upcoming programmed paths. Failure to comply with this restriction results in a "fill-in arc" which can seriously undercut the workpiece. See Figure 6.27.



UPPER DRAWING SHOWS REQUIRED CUTTER PATH, PROGRAMMED AND COMPENSATED. LOWER DRAWING SHOWS LARGE FILL-IN ARC THAT RESULTS FORM TOO MANY NON-CUTTER COMP BLOCKS. THE WORK-PIECE IS SEVERELY UNDERCUT.

Fig. 6.27 Too Many Non-cutter Comp Blocks

Correction:

During compensated path programming, make certain that motion blocks are not separated by more than four blocks that do not contain an X or Y axis departure command.

Special Cases and Conditions

Retracing a Compensated Cutter Path

To retrace a compensated cutter path, either linear or circular, program the return move using the opposite compensation. For example, if the forward move was programmed with G41 active, program the return move with G42.

This is the only situation in which the cutter comp can be altered without intermediate ramp-on blocks.

FIXTURE OFFSETS

GENERAL INTRODUCTION

The Fixture Offset feature provides a means of establishing a fixture placement related offset which may be used for one of three purposes:

1. To compensate for deviations which exist as a result of incorrect positioning of the fixture on the machine pallet or table.
2. To establish a zero point, on workpiece or fixture, corresponding to a datum point on the part print.
3. The optional G47 calculated fixture offset function is used to provide for the reapplication of fixture offsets after either axis rotation or indexing.

There are two Preparatory functions (G codes) used to invoke one of two modes of fixture offsets; G48 or G49.

G48 EXPLICIT FIXTURE OFFSETS

When in Explicit mode, fixture offsets are established by including "H"word and the desired number of the fixture offset set to be invoked in the part program. The commanded H number corresponds to a set of offset values on the Fixture Offset display.

G49 IMPLICIT FIXTURE OFFSETS

When in fixture offsets Implicit mode, fixture offsets are established whenever the defined implicit axis is commanded to a specific position with absolute mode

(G90) active which corresponds to the "I (a)" entry in the Fixture Offset display. With this system, the operator may select which machine axis is the Implicit axis using a fixture offset display entry procedure.

NOTE

Additionally, there is an optional G47 command that is used to access fixture offset data from the fixture offset table. This command is discussed in Chapter 2. G47 is commanded when the programmer or the operator wishes to use the calculated fixture offset option.

In addition to the commands G48 and G49 there are also four Type II data statements used with fixture offsets. These are defined as follows:

1. UFX, --- Calculate & Load Fixture Offsets

This Type II data statement is to contain the H number for the calculated set of fixture offsets, the I Implicit axis position if required, and the axis coordinates were the spindle is to be located.

When executed, the control calculates the incremental distance and direction from where the axes currently are to the entered position in the Type II data statement.

The calculated offset is then loaded into the specified H word location on the Fixture Offset Table.

2. FXC, --- Load a predetermined Set of Fixture Offsets.

The entry in this statement is to include the H number of the set of fixture offsets to be loaded, followed by the Implicit axis position if any and then the predetermined axis offsets.

This is only to be used when the offsets have been previously determined.

3. USF, --- Save Current Fixture Offset Table

This Type II data statement is used within a part program to save the contents of the fixture offset file to the default mass storage device, usually the hard disk. The file is saved with the name extension ";FXT". The saved Fixture Offset Table may then be later recalled for later use, either manually (LD,) or restored as part of a part program using the ULF Type II data statement.

The programmer may choose one of two methods for naming the saved Fixture Offset Table.

A. If the file is to be given a specific name, then include it after the letters USF.

   e.g. N200(USF,PP12-11)

This will cause the saved table to be loaded in a file named PP12-11;FXT.

B. If the file is to be written for a specific pallet , the programmer need only enter (USF).

   e.g. N200(USF)

This will cause the saved file to be named according to the current pallet number active in the program management table. If Pallet 52 were active, the saved fixture offset file would be named PALLET52;FXT.

This file may be reloaded at a later time using the command (ULF). No name need be entered provided the currently active pallet number is Pallet 52.

4. ULF, --- Load Fixture Offset Table from Saved File

This Type II data statement is used within a part program to restore the contents of a previously saved fixture offset file from the controller's default mass storage device, the hard disk. The file which was saved has the name extension ";FXT".

The programmer may choose one of two methods for restoring a saved Fixture Offset Table.

A. If the file is to be given a specific name, then include it after the letters ULF.

   e.g. N200(ULF,PP12-11)

This will cause the saved table to be loaded from a file named PP12-11;FXT.

B. If the file is to be reloaded for a specific pallet , the programmer need only enter (ULF).

   e.g. N200(ULF)

This will cause the file labeled with the name of the currently active pallet number in the program management table. If, for example, Pallet Number 52 were active, the fixture offset file restored would have been named PALLET52;FXT.

This file may be recopied at a later time using the command (USF).

C. (ZTB,H) - Zero Fixture Offset Table

This Type II data statement deletes the entire contents of the current fixture offset table.

A total of 32 sets of fixture offsets are included in the two-page Fixture Offset display. Each H number may contain an offset for all axes, for one axis or for any combination of axes.

FUNCTIONAL DESCRIPTION

1. Fixture offsets may be used to compensate for deviations in either the workpiece, or the fixture from that which was originally programmed.

Using this method presupposes that the part program coordinates originate at a known reference position of the axes slides, and that the workpiece and fixture are positioned at some absolute distance from the origin.

In situations where deviations are likely to exist as result of incorrect positioning of the fixture on the table, the part programmer can make provisions within the program itself to correct for fixture inaccuracies. It is the machine operators responsibility to ensure that the correct values are entered into the Fixture Offset Table.

2. Fixture offsets may be used to establish a new zero point or floating zero at a known datum point on the part or fixture.

Use of this method presupposes that the programmer has selected some known datum from which the part program coordinates will originate. Once the fixture offset is established, the part program can be written according to print dimensions.

ESTABLISHING AND ENTERING FIXTURE OFFSETS

Fixture offset values are established by the machine operator, who jogs the axes slides to a known datum (target bushing, corner of workpiece, etc) using a dial indicator, edge finder or similar device inserted into the spindle. Once the datum coordinates have been found,the axes display registers on the main display show the absolute positions. The next step is to enter the offset values into the Fixture Offset display [SELECT] [U].

The offset value itself, which is the number entered into the fixture offset table, represents the signed incremental distance as measured *from the programmed coordinate* (presumed to be the correct coordinate) *to the required coordinate* (determined by measuring). This is to say that the fixture offset value is equal to the incremental distance and direction from the programmed location to the actual absolute location.

There are several methods by which the offset values can be entered into this display:

1. Through a programmed Type II data command, (UFX,) included in the part program, which automatically calculates and enters the offset values into the display. Explained below.

2. Through a programmed Type II data command, (FXC,) which can be entered through MDI by the machine operator. Explained below.

3. By the operator, who can call up the Fixture Offset display and manually enter the values into the correct columns using the keyboard. This method is explained in the Operators Manual.

4. By saving a fixture Offset File using a (USF) Type II data statement relating fixture offsets to a specific pallet map number. This saved file may then be reloaded to the Fixture Offset Table by commanding a (ULF) Type II data statement.

Entering Offsets Through UFX

The UFX statement is used to calculate the value of a specific offset, and to enter the result of the control's automatic calculation into the Fixture Offset Table. The UFX statement may be entered through through MDI by the machine operator, or may be included as a data block within the part program.

Once a fixture offset has been established and entered through a UFX statement, the offset values may be applied to the part program through an H command (G48 Mode), or through a programmed command which positions the Implicit axis to the specific coordinate for which the offset is intended.

NOTE

There are several different methods by which fixture offsets can be established. In practice, use whichever method is most consistent with your own shop practices. The method explained below is just one possible approach.

A typical UFX statement for a specific offset might appear as follows:

N40 (UFX,H1I90.X0Y0Z0B90.)

Where:

The open and close parentheses are the 'control out' and 'control in' characters respectively.

The 'UFX,' is the mnemonic. The comma is required.

The 'H1' is the fixture offset number or H-word. This identifies the specific line into which the calculated fixture offset is to be loaded. Any number from 1 thru 32 may be used following the letter address H.

The 'I90.' is the absolute B axis position, B90., at which the specific set of offsets for X, Y, Z and B included in this statement, will be automatically applied if operating in the G49 Mode.

The 'X0Y0Z0' is the coordinate required by the part program. This is the location to which the slides are positioned in response to the programmed command X0Y0Z0.

In this particular statement, the current positions of the X, Y, and Z axes are now recognized by the control as the zero point for the part program. The offset values entered into the table are the arithmetic difference between the zero reference position and the current position of the axes.

Had the statement instead had different values for X and Y; e.g., (UFX,I90.X6.5Y9.75Z0B90.), then the current positions of the X and Y axes would be recognized by the control as X6.5 and Y9.75. The offset values entered into the table automatically are the arithmetic difference between the coordinates in the UFX statement and the current position of the axes.

For example, assume that the operator is to indicate the corner of a workpiece and establish that corner as the zero point for the X and Y coordinates in the part program. If the actual coordinates of the corner point are X10.0552 and Y2.1282 and B91. and the offset command is (UFX,H1I90.X0Y0B90.), then the values entered into the Fixture Offset Display are 10.0552 for X Axis, 2.1282 for the Y Axis and 1.0 for the B axis.

On the other hand if the 'programmed' coordinates for that corner point are to be X10.Y3.B90. but, because of inaccuracies in the fixturing, the corner point is at an actual coordinate of X10.0552 and Y2.1282 and the offset command is (UFX,H1I90.X10.Y3.B90.), then the values entered into the Fixture Offset Display would be .0052 for X and -.8718 for Y. No offset value will be entered for the B Axis, since B axis is at the correct position. These offset values are for the arithmetic difference between the coordinates in the UFX statement and the current position of the axes.

In using the UFX statement, the programmer must take steps to ensure that the machine operator is informed of the necessary procedure. If desired, the programmer can insert instructions into the part program using a programmable operator message (MSG,) or the comment statement (!). For example:

```
N10 X10.Y5.B90.H0F0
N20 Z14.
N30 M0
N40 (MSG, INDICATE CORNER, CANCEL JOG RETURN)
N50 (UFX,H1I90.X0Y0B90.)
```

If fixture offsets are to be used in the G49 mode, then the UFX statement must contain an "I" command which specifies the B axis coordinate at which the specific set of offsets will be automatically applied.

If the fixture offsets are to be used only in the G48 mode, then the "I" command need not be included in the UFX statement, since the values will be applied to the part program through an H command.

If operating in G48 mode, and it is desired to ensure that no implicit position be included, the entry "I-0" may be included in the UFX Statement, as this is a nonexistent position. For example:

N40 (UFX,H1X0Y0B90.)

Entering Offsets Through FXC

The FXC Type II data statement is used to load a set of predetermined fixture offsets into the fixture offset table. No calculations are performed by the controller's computer. Instead, the signed dimensional difference between the required position and the current position of the axes is to be calculated by the operator and entered into the statement. A typical FXC statement might appear as follows:

N40 (FXC,I90.X14.Y11.5Z7.25)

Where:

The open and close parentheses are the 'control out' and 'control in' characters respectively.

The 'FXC,' is the mnemonic. The comma is required.

The 'H1' is the fixture offset number, or H-word. Any number from 1 through 32 may be used following the letter address H.

The 'I90.' is the absolute B axis position. (B90.), at which the specific set of offsets for X, Y, Z and B which are included in this statement, will automatically be applied if operating in the G49 mode.

The 'X14.Y11.5Z7.25' is the dimensional difference between the zero reference positions for the axes and the current position of the spindle.

In this particular statement, the current positions of the X, Y, and Z axes are recognized by the control as the new zero point for the part program. The offset values entered into the table are the arithmetic difference between the zero reference position and the current position of the axes.

Had the statement instead had different values for X, Y, and Z; e.g. (FXC,H1I90.X-.5Y.075), then the current positions of the X, Y and Z axes would be recognized by the control as the corrected coordinates for that particular workpiece.

If fixture offsets are to be used in the G49 mode, then it is required that the FXC statement contains an "I" command which specifies the B axis coordinate at which that specific set of offsets is to be automatically applied. If the fixture offsets are to be used only in the G48 mode, then the "I" command need not be included in the FXC statement, since the values will be applied to the part program through an H command.

Example:

N40 (FXC,H1X-.5Y.075)

SAVING A FIXTURE OFFSET TABLE

Both the programmer and operator have the ability to save the entire contents of a Fixture Offset Table. This table may then be restored at a later time in order to process a fixtured pallet where the fixture setup is not to be altered while other pallets are being processed.

The operators procedure for saving the fixture offset table involves entering the command "SV," and entering the name under which the offsets are to be saved. The procedure for this is outlined in the Operators Manual, PUB 883.

The programmer may save the contents in a similar manner, however; the command entered is a Type II data statement contained within a part program. This command is the USF command, which stands for User Save File. If desired, the programmer may enter a specific name under which the file is to saved. However, a simpler procedure is to merely enter the command "(USF)" which will then save the contents of the Fixture Offset Table to a file named "PALLETnn;FXT", where "nn" represents the number of the currently active pallet on the program management table. In either case, the file saved is processed and written using FXC Type II data statements, as well as certain other statements. The values written into the FXC lines represent the discrete values present on each line of the Fixture Offset Table.

The contents of this file may be reloaded manually by the operator, or may be reloaded within a part program which contains the "ULF" Type II data statement.

RESTORING A SAVED FIXTURE OFFSET TABLE

A saved Fixture Offset Table may be restored either by the machine operator, or by the part programmer by writing a table load command within a part program.

In the event the operator is to restore the Fixture Offset Table, the command LD, on the Fixture Offset display is used to restore a named saved Fixture Offset File.

When it is desired to automatically restore the Fixture Offset Table, the programmer may enter the Type II data statement "(ULF," within the part program. If the command "(ULF," is entered, the controller will attempt to load the fixture offset file with any name entered following the comma. Names may be up to 28 characters in length. All saved fixture offset files have the extension ";FXT", however this needn't be entered.

If only the command (ULF) is entered in the part program the controller will attempt to load a file named with the number of the currently active pallet in the Program Management display. The format of the name is as follows:

PALLETnn;FXT

where "nn" represents the number of the currently active pallet.

ACTIVATING FIXTURE OFFSETS

Fixture offsets are activated by one of two methods. The control defaults to G49 Implicit mode upon power up. With G49 modal, the control applies listed fixture offsets whenever an absolute axis command to a listed implicit axis position is executed. Fixture Offset display contains a column labeled "I(a)", under which can be entered an implicit axis positions where specific sets of fixture offsets are required. The 'I' indicates implicit and the '(a)' indicates which of the configured machine axes is the Implicit axis. Generally, this is the B axis; however, any configured axis indexing, rotary or linear may be activated as the Implicit axis.

With G49 active, Implicit axis position commands can then be used to apply fixture offsets automatically. In addition, the display contains an entry for Implicit axis itself, allowing event the Implicit axis to be offset by an absolute axis position command. Once the offset values for the Implicit axis have been determined, and entered into the Fixture Offset Table, then any time an Implicit axis position is commanded to a position listed on the Fixture Offset Table, the offset values associated with that Implicit axis position are automatically applied to the program.

The offsets are applied automatically when G49 mode is in effect. Additionally, the G0 mode must be in effect during the departure command which moves the B axis to a position listed in the Fixture Offset display.

If the G48 FIXTURE OFFSET EXPLICIT mode is in effect, fixture offsets are invoked only when the an "H" word command is entered. The H word invokes a specific set of fixture offsets which must have previously entered into the Fixture Offset Table.

PROGRAMMING CONSIDERATIONS FOR FIXTURE OFFSETS

1. During the part processing cycle, an H word or an implicitly applied set of offsets updates the display registers to show the offset position, but the actual compensation is not applied until each offset axis is commanded to a position. This means that offsets will be applied even if the block contains what appears to be a no motion command.

2. The current fixture offset number in effect is highlighted in intense video on the [MAIN] display.

3. The Axis Offset display, [DISPLAY LIST] [O], has a register which shows the actual offset in effect, as well as the absolute position of each axis.

4. When the G48 mode is in effect, no fixture offsets are applied automatically by programming to a new B axis coordinate.

When the G49 mode is in effect, the control will not accept a programmed H word, other than H0 which may be used to cancel an active fixture offset.

The programmer is however allowed to alternate between Explicit and Implicit modes during the course of the program by commanding the desired preparatory function, G48 or G49.

BYPASSING AND CANCELING FIXTURE OFFSETS

Fixture offsets may be bypassed, and the offset axes positioned to absolute coordinate at any time by programming an M25 command in a data block which contains the required absolute position.

For example, a command, N100X20.Y27.Z20.M25 positions the axes to those indicated coordinates regardless of any offsets in effect.

In addition, an M06 command bypasses the active offsets to move the slides to the tool change position.

If the G49 mode is in effect, the fixture offsets associated with a particular B axis coordinate can be bypassed by programming a G01 command in the block in which the B axis to the coordinate at which a fixture offset would normally be invoked.

Fixture offsets may be canceled by programming an H0. Additionally, a new fixture offset command cancels the previous one and establishes the related new set of offsets.

USING FIXTURE OFFSETS WITH OTHER OFFSET MODES

The fixture offset feature is designed to function with each of the other offset modes; Length comp, Diameter comp and G98. If using G98 offset mode, it is preferable to establish the fixture offset first, then establish the G98 offset from the new position established by the fixture offset.

The following example demonstrates the methods by which fixture offsets can be utilized to compensate for a deviation in the X and Y axes resulting from a mispositioning of the fixture on the pallet.

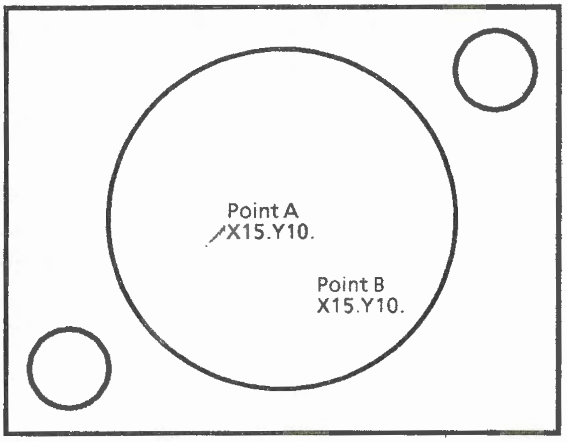


Fig. 6.28 Fixture Offset Example

Assume that the centerpoint of the bored hole (represented by Point A) should be at an absolute position of X15. Y10. at a B axis position of B90., but because of the fixturing inaccuracy, the centerline of the spindle is actually below and to the right of the required hole location when the command X15. Y10. is executed (represented by Point B).

The procedure is to establish an offset so that when the part program calls for X15. Y10., the spindle is positioned to the center of the bored hole (Point A) and not to Point B which is the absolute X15. Y10. position. Subsequently, any other point on the part will also maintain a correct relationship with the centerpoint of the bored hole and no editing of part program coordinates will be necessary.

The first step is to determine the absolute coordinates of the centerpoint (Point A) and second step is to determine the distance and direction measured from the programmed point to the required point (Point B). This second step can be eliminated if a UFX command is used.

Next, the offsets must be entered into the Fixture Offset Table. Following are some methods by which Fixture Offsets can be entered and activated in the part program.

Entering Offsets Using FXC Statement

A part program contains the following data blocks:

N50 X15.Y10.F0
N60 M0
N70 (MSG,INDICATE BORE CL. CANCEL JOG RETURN. ENTER OFFSET)

Explanation:

1. Block N50 contains a command which moves the X and Y slides to the programmed bore centerline.
2. Block N60 contains a program stop command to halt the part program.
3. Block N70 contains a message telling the operator to find the coordinates of the bore centerline and enter the offset values into the Fixture Offset Table. With an indicator in the spindle, the operator jogs the axes slides until the centerline is found. The axis registers on the [MAIN] display now read X14.8775 Y10.1334. The incremental distance from the programmed coordinate to the required coordinate is a negative 0.1225 inches for the X axis and a positive 0.1334 inches for the Y axis. The operator keys in the following data in the MDI mode:

   (FXC,H4I90.X-.1225Y.1334)

Entering Offsets Using UFX Statement

A part program contains the following data blocks:

N50 X15.Y10.F0
N60 M0
N70 (MSG,INDICATE BORE CL. CANCEL JOG RETURN. ENTER OFFSET)
N80 (UFX,H4I90X15.Y10.)

Explanation:

1. Block N50 contains a command which moves the X and Y axes slides to the programmed bore location.
2. Block N60 contains a Program Stop command.
3. Block N70 contains a message telling the operator to find the coordinates of the bore centerline and cancel the automatic jog return feature.
4. With an indicator in the spindle, the operator jogs the axis slides until the bore centerline is found. The axis registers on the [MAIN] display now read X14.8775Y0.1334. After the operator has canceled the jog return, the [CYCLE START] command executes block 80.
5. Block N80 contains the programmed UFX statement. The control algebraically sums the coordinate values from the [MAIN] display and the values entered into the Type II statement and automatically enters the values into the Fixture Offset Table. In this case the entered values are -0.1225 for X and 0.1334 for Y.

INVOKING OFFSETS IN G48 EXPLICIT MODE

In the G48 mode the specific set of offsets must be established "explicitly" by programming an H number which corresponds to the required offset values.

N100 M6
N110 G0G48G70G90G94
N115X15.Y10.Z20.F0T2
N120 B90.
N130 X15.Y10.H4 (X and Y axes move to their offset position)

INVOKING OFFSETS IN G49 IMPLICIT MODE

In the G49 mode, offsets are established "implicitly" by programming an Implicit axis (generally B) command which corresponds to a coordinate in the "I(a)" column of the Fixture Offset Table

N100 M6
N110 G0G48G70G90G94
N115X15.Y10.Z20.F0T2
N130 B90. (Automatically establishes the fixture offset values)

CHAPTER 7

PROGRAMMING ADDITIONAL OR OPTIONAL FEATURES

BROKEN TOOL DETECTION

NOTE

The user's machine may be equipped with the Multipoint Tool Sensor. It is not the tool sensor which is discussed in this section. For information on a Multipoint Tool Sensor, refer to Pub 935.

Broken Tool Detection is programmed by writing the preparatory function G38 in a data block which also contains an appropriate set of commands for the type of tool to be checked. The Broken Tool Detection feature is used in conjunction with a checking station which is located on the grid post mounted on the X axis table unit (See Figure 7.1).

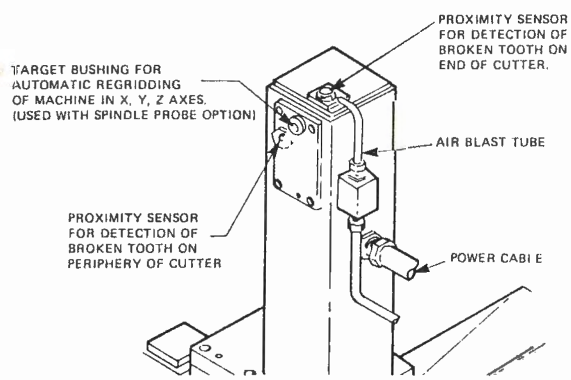


Fig.7.1 Broken Tool Detection Station

Functional Description:

The Preparatory function G38 is non-modal and is used by the control to check the setting distance or cutter diameter of the tool in the spindle. Prior to commanding this function, the cutter must be positioned to within 1.0mm (0.040") of the proximity switch.

The G38 function should be programmed only when the Broken Tool Detection feature is desired.

Programming Application:

A G38 cycle may be programmed at any point in the part processing cycle to determine the status of the tool currently in the spindle. Once positioned to the checking station, a proximity switch determines the presence or absence of cutting teeth on the tool. If the system detects that the tool is broken, the message "TOOL BROKEN" is generated on the message screen and the machine tool is placed in a [CYCLE STOP] condition. A visual inspection of the tool should be made before continuing the part processing cycle.

Programming Instructions:

Both multiflute and single point tools may be checked using the Broken Tool Detection feature. The following instructions document the procedures involved for both tool types:

Multiflute Cutters

When checking for a broken blade contained within a multiflute cutter, position the cutting edge in front of or above the proximity switch desired with the spindle rotating. In the next block of program data, enter a G38 command along with an S number. The S number represents the total number of blades to be checked in one revolution of the cutter. If a blade is not detected, the proximity switch is NOT triggered and system designates the tool as broken. The block-by-block descriptions which follow describe the recommended part programming procedures for checking a multiflute cutter (See Figure 7.2 for examples).

1st Block

A. Sequence number.
B. Position X, Y, Z to within 1.0mm (0.040") of the proximity switch desired.
C. Enter a spindle speed of S30. This spindle speed is recommended when checking a multiflute cutter; however, the spindle speed can be changed.
D. Enter the Miscellaneous function M03 to cause spindle rotation.

NOTE

If a Length and/or Diameter compensation is assigned to the tool being checked, their values must be in effect when positioning the tool to the checking station on the grid post.

2nd Block

A. Sequence number.
B. Enter the Preparatory function G38. This code is non-modal and is active for this one block only.
C. Enter an S number (Example S16). The S number indicates to the control that there are 16 flutes in the cutter to be checked in one revolution of the spindle.

Example:

N70 X Y Z S30 M03 F	1st Block
N80 G38 S16	2nd Block

Single Point Cutter

When checking for a broken blade contained in a single point cutter, position the cutting edge in front of or above the proximity switch desired with the spindle in a keylock position. The keylock position orients the blade of the cutter to a known location. The next block of program data must contain a G38 command to check the presence or absence of the blade. If the blade is not detected, the proximity is NOT triggered and the system designates the tool as broken. The block-by-block descriptions which follow describe the recommended part programming procedures for checking a single point cutter. The illustrations (Figure 7.3) are provided to demonstrate the correct procedure.

1st Block

A. Sequence number.
B. Position X, Y, Z to within 1.0 mm (0.040 inch) of the proximity switch desired.
C. Enter Miscellaneous function M19 to keylock spindle.
D. Enter an S value to orient the spindle keylock position to point the cutter bit downward. (Example S270).

NOTE

If the bit of the cutter contained in the tool holder is pointing down when the spindle is placed in the zero keylock position, the S value is NOT required.

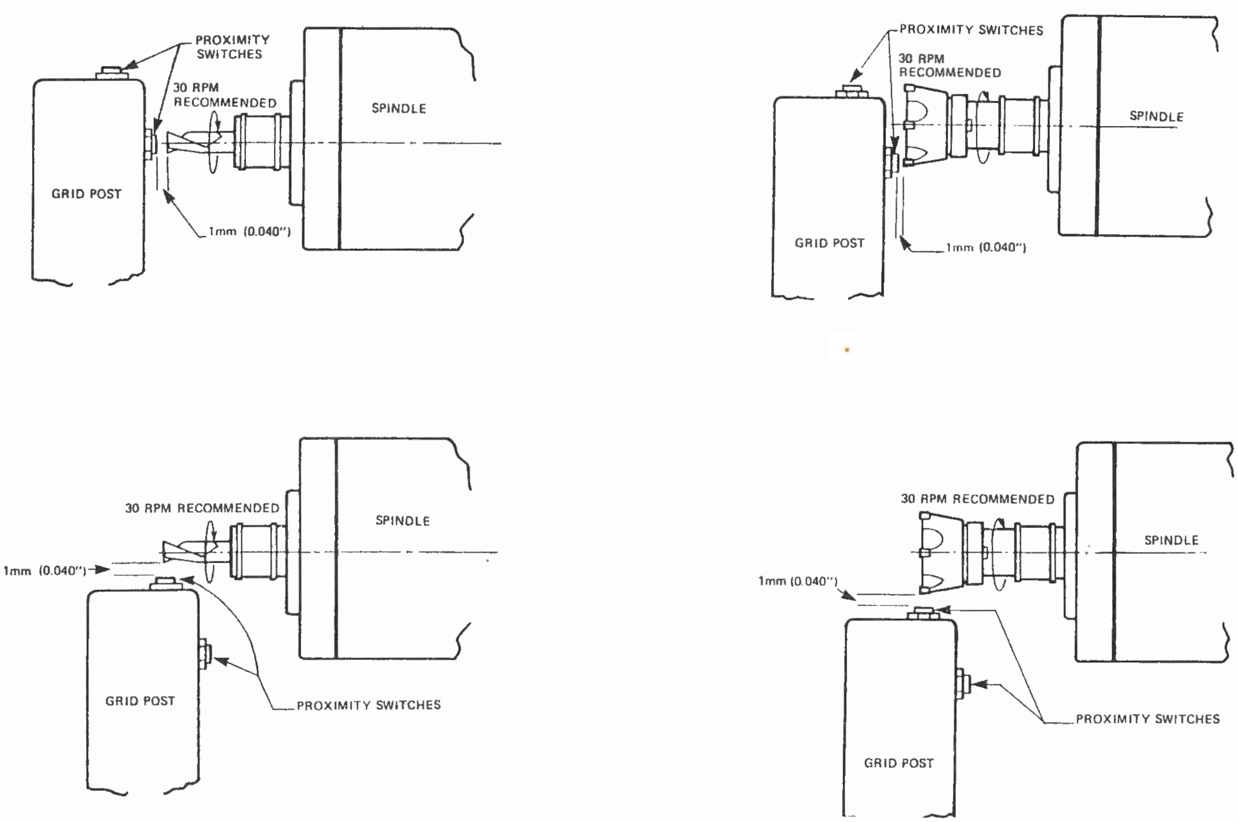


Fig. 7.2 MULTIFLUTE EXAMPLES

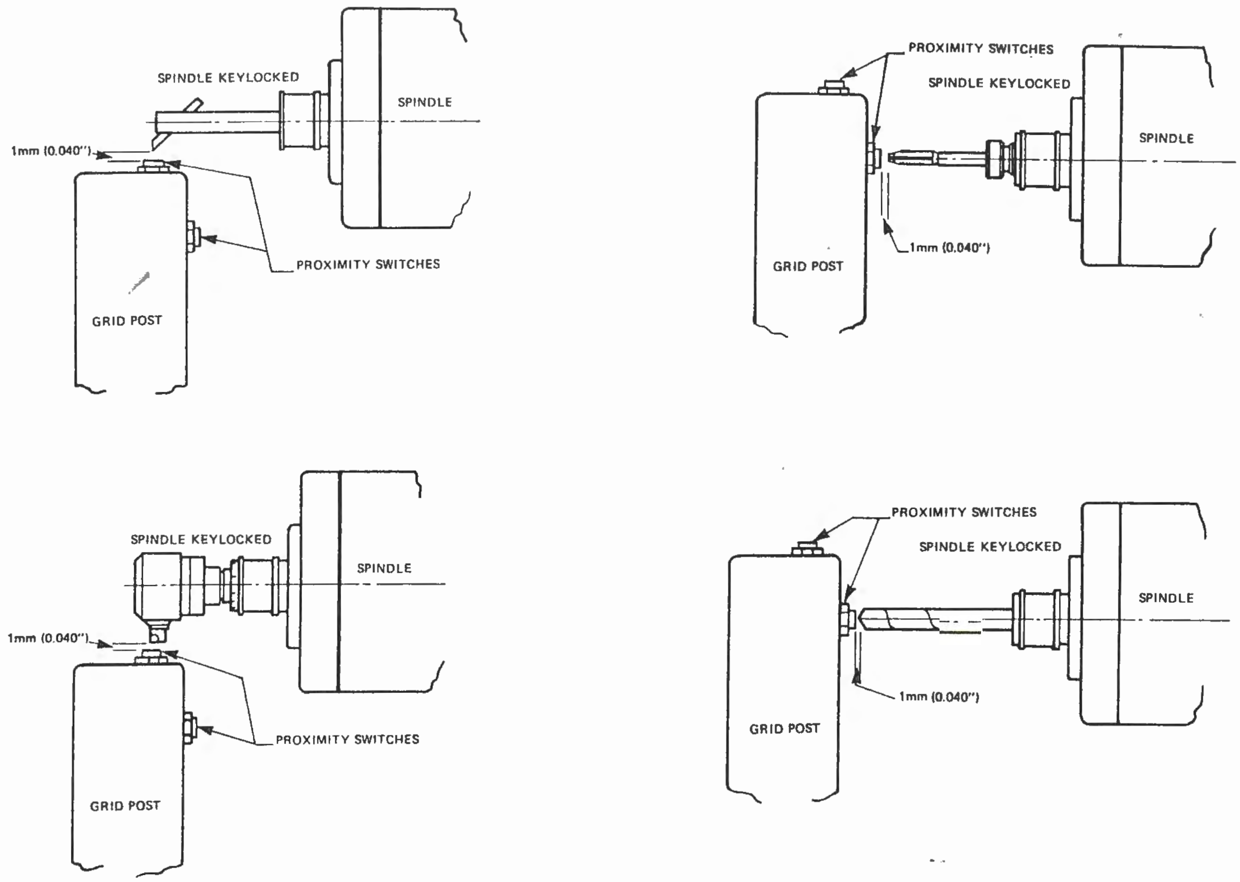


Fig. 7.3 SINGLE POINT EXAMPLES

If a Length and/or Diameter compensation is assigned to the tool being checked, their values must be in effect when positioning the tool to the checking station on the grid post.

2nd Block

A. Sequence number.
B. Enter the preparatory function G38. This code is non-modal and is active for only this one block.

Example

N200 X Y Z M19 S270 F	1st Block
N210 G38	2nd Block

D'ANDREAS U-HEAD

KT's Advanced Programming Language (KT-APL) supports an option for the D'Andreas U-Head facing and boring attachment. The U-head software option associates the effective diameter of the tool with the 6th axis (typically the U axis), therefore, the diameter of the tool may be modified by the part program. The U-head may be used to replace multiple boring bars, or to modify a bore by an amount that depends upon a previous probe cycle. The U-head may also be used for a lathe type facing operation, or for tapered bores or bores with circular cross sections.

The U-head option will set up the 6th axis parameters so that they are compatible with the mechanics of the head. For example, the rapid rate of the head is 9.843" /min, and the axis loop gain is 25. The rapid rate of the U axis may not be exceeded in linear or in ZU circular moves. The actual rate is the instantaneous combined vector velocity. In a circular move the velocity is limited to the rapid rate. In a linear ZU move the velocity may be any value that does not cause the component of the velocity along an axis to exceed the rapid rate *for that axis!* An attempt to program a rate that does not agree with the above rules will result in an input source error displayed, and the block will not be executable.

A machine that does not have a C axis will use the C axis select pad on the front panel to select the U axis.

The U-head option causes the control to force a "slow" state whenever it is selected to prevent the U-head from being jogged at any rate other than "slow".

The U-head option permits circular blocks in the ZU plane to be programmed with either center coordinates or radius. According to the rules governing circular planes, a ZU circle must be performed in G18 G21 mode. A [CLEAR ALL LOGIC] will return the control to the default plane (G17).

Diameter compensation may also be used with the U-head. KT-APL contains a corrected type of diameter comp that will work similar to the usual diameter comp in the XY plane except that the comp value will refer to the diameter of the tool tip of the U-head boring bar insert. This corrected version permits diameter comp to be used on tapered and circular bores as well as simple bores. It is necessary to program a "G18 G21" along with the G41 or G42 to enable the new version of diameter comp. Note that the type of compensation is also very similar to tool tip compensation as used on a lathe.

When a U-head tool record is added to the tool table the record must inform the control that the tool is a U-head. This is done by positioning the field marker in the tool table to the status field of the tool in question and hitting the letter "U". The "U" character will then appear in the status field for that tool.

With the U-head option, the control will not accept commands for the U axis unless the tool in the spindle is designated as a U-head. This restriction is necessary to prevent the U-head tool and its drive from losing synchroneity while the head is not in the spindle.

To automatically change the U-head into or out of the spindle the 6th axis first must be driven by PMI to the 6th axis tool change position (0.0"). This is essential because first, the head-drive engagement mechanism must be horizontal or vertical (depending upon the type of machine) which is the position at which a mechanical locking device on the head will keep it in position when it is released from the drive. Second, when more than a single head is used the control must use the same axis for both heads, so the same starting position for each head is necessary to prevent the control from forgetting the position of each head.

At initial setup of the U-head, U axis must be gridded without a head in the spindle. The head must be hand cranked to 0 before placement into the tool magazine.

The grid preset, lead screw compensation, and reverse compensation for the 6th axis must be carefully adjusted so that the tongue on the drive that engages the U-head is precisely horizontal when the U axis is driven to the tool change position.

Programming the U-head Option

Before programming the U-head, the programmer must be familiar with the construction of parameter statements, user defined subroutines and cycles and spindle probe programming. This section references these topics, but does not discuss their creation. For information on spindle probe programming see Pub 878 Spindle Probe Programming, Operating and Maintenance Manual. Information on subroutines and cycles is contained in Chapters 8 and 9.

The examples in this section contain text to explain the operation of the various commands. It is the programmer's decision to include this information in an actual program. Additionally, space has been inserted in the program text for greater clarity. This space would not appear in an actual program.

U-head Operating Suggestions

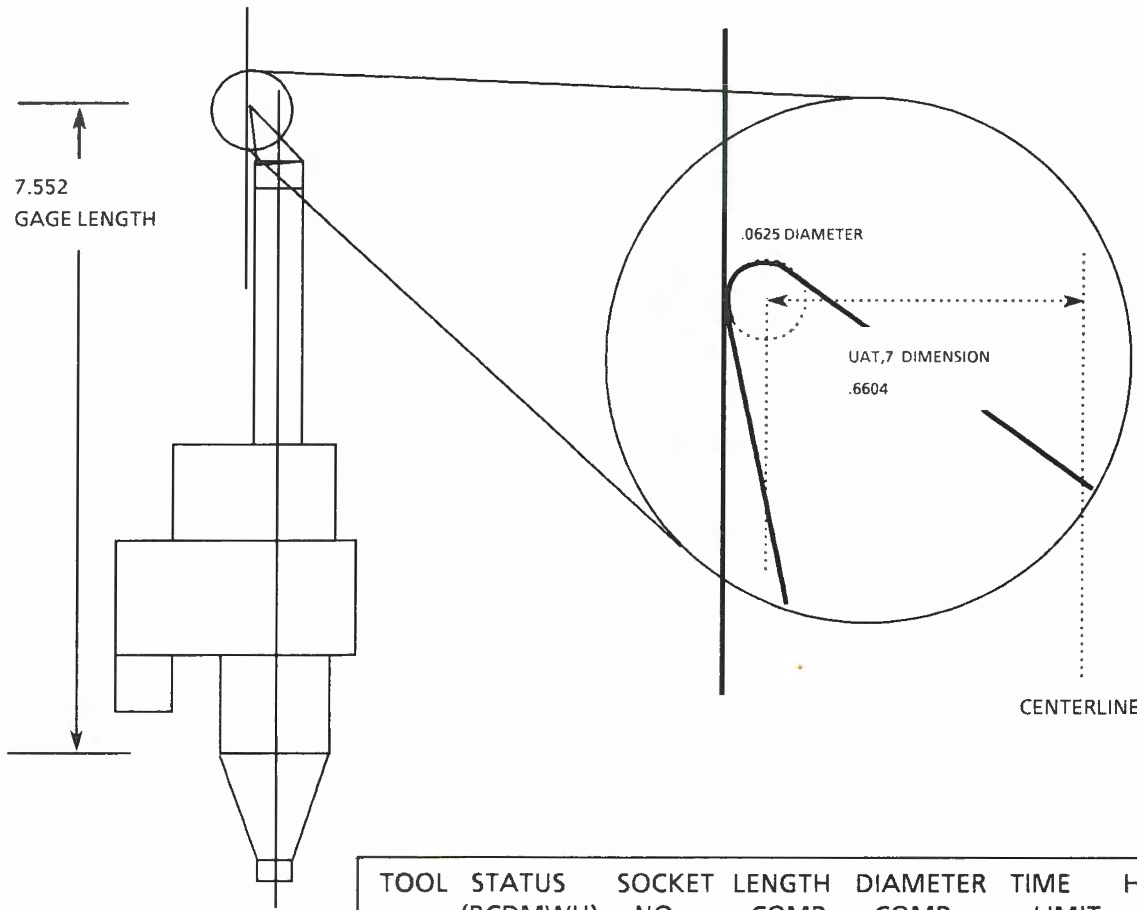
1. Both the programmer and the operator must be aware of the location of the U-head position and orientation in the tool magazine. Sufficient space must be left for adequate clearance between the U-head and the surrounding tooling. If the tool selection is of the progressive type it may require that dummy tool changes be performed to place the U-head in a safe storage location.
2. The programmed feedrate for the U axis may not exceed 10 IPM. Feedrates which are in excess of this feedrate will result in a error code.
3. There exist neither a software nor a hardware protection which will protect the tool from RPM in excess of that specified by the manufacturer. Consult the Kennametal U-head brochure for the proper RPM for each type of U-head.
4. Both the programmer and the operator should be aware of the existence of the travel limits of the tool and the possible collision zones between the tool, fixture and the machine.
5. It is recommended that the tool table be used to retain information regarding the tool insert radius. The tool nose radius information may be set used thru the use of a parameter. See the example in this section.
6. The U-head must be identified to the control by the letter "U" in the tool table status column.
7. The programmer should be aware that the correct axis commands are entered into the control (i.e. G18 and G21) when using either G2 or G3.
8. The correct cutter compensation (i.e. G41 or G42) must be commanded in the part program.
9. The G33 must be used in threading programs.

10. The spindle speed override selector switch is active during a program containing a G33 command. However, the spindle speed should not be altered while threading a part, because of a temporary loss of synchroneity between the axes and spindle.

The following example describes a subroutine which is used to access tool tip information from the tool table. It establishes the relationship between the tool tip in the U-head and the spindle centerline using a tool suffix number. In this example, suffix tool number 7 is used to store the information. The distance between the U-head tip, MINUS the insert radius is entered in the length comp column. The insert tip radius is entered in the diameter comp column as a primary compensation. This subroutine will automatically apply a G98 offset to the U-axis which is equal to the tool tip dimension minus the insert radius.

This example concerns cutting an ID. Were the program written for cutting an OD, the formula would be reversed and the distance would be the U-head cutting distance PLUS the insert radius. (See Fig.7.4)

```
(DFS,TIP-OFFSET,SAVE)
(UAT,7)
(PAR,L1=L3018"TIP-OFFSET")
(UAT,0)
M25U0
G98UL1     !U-axis is now offset equal to the value
           entered on suffix line 7
(END)
```



TOOL	STATUS (BCDMWU)	SOCKET NO.	LENGTH COMP.	DIAMETER COMP.	TIME LIMIT	H.P. LIMIT
1	U	21	7.5520	0.0625	0:59	1.5
.7			.6604			

This example of tool table information does not show all the columns of the tool table but only refers to the items discussed in the example.

Fig. 7.4 U-HEAD TOOL EXAMPLE

```
G70G90G94T1
G48H1
M6  !U-Head is loaded into the spindle
(CLS,TIP-OFFSET)   !Call subroutine to calculate the offset
Z20.M25S500M3M8     !Turn on spindle and coolant
X0Y0F0  !Position at rapid traverse
U2.  !Position the U axis for a 4.0" diameter cut
G18G21  !Selection of G18ZX plane and G21 switches 2nd axis with U
G41Z.1   !Ramp-on move for cutter comp left
Z-.05F5.  !Bore the 4.0" hole
U1.75     !The .05" deep face is made
U1.5Z-.3I1.75K-.3G02    !Contour CW.25 radius (this radius is made using I, K format for circular interpolation)
U1.25Z-.55R.25G03       !Contour CCW.25 radius (this radius is made using the R format for circular interpolation)
U.75      !Face the .55 step
U.5Z-.8   !Contour the 45 degree chamfer
Z,-(1.+.1)      !Bore thru 1.0" plus 0.1 clearance (note the use of the parenthesis is optional. This command could be expressed as Z-1.1)
U.4G42F0        !Retract U from the part surface and reverse cutter comp G42. (Note that the cutter will now be on the right side of the
                part surface during retraction.)
Z.1  !Retract to Z clearance position
Z20.M25G40  !Retract for the end of the program and cancel cutter comp
M30
```

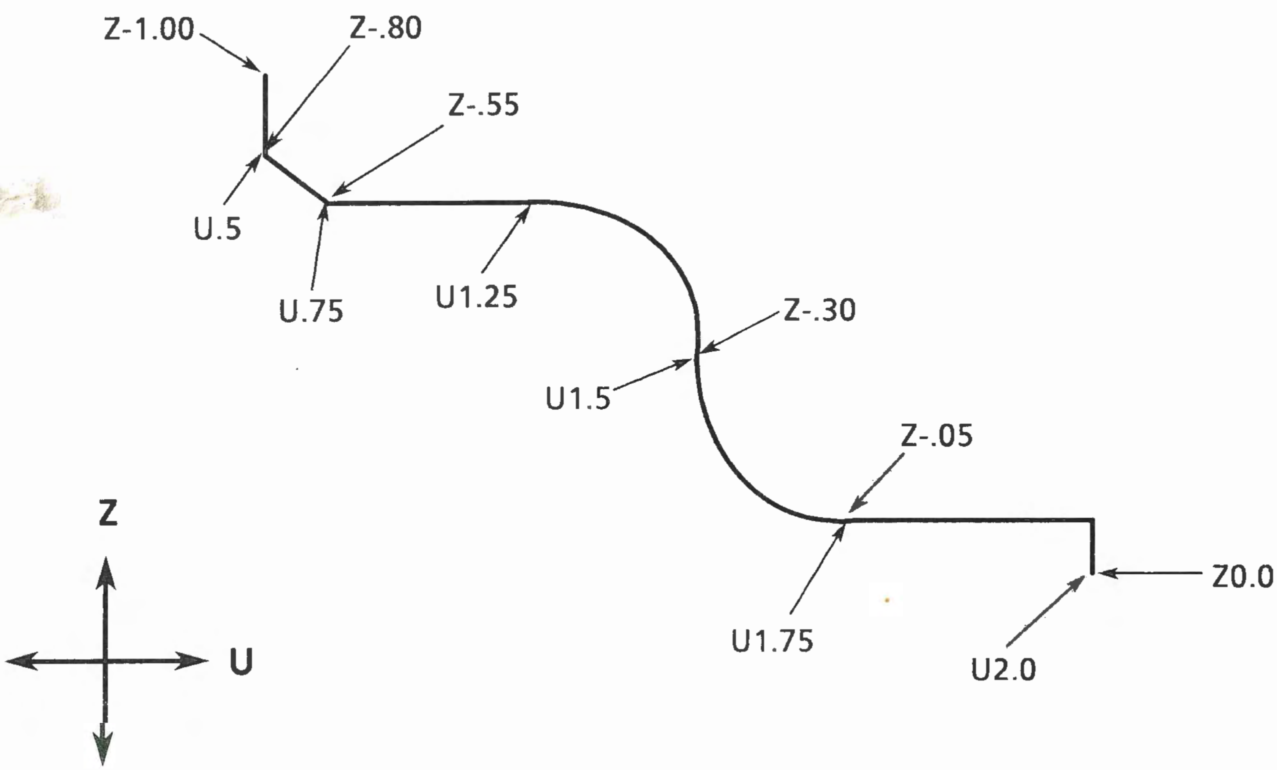


Fig. 7.5 EXAMPLE 2 - CHAMFERING, BORING, FACING and CIRCULAR INTERPOLATION

The second example demonstrates a U-head program that includes chamfering, facing, boring, and circular interpolation. Additionally, it shows the use of the tip offset subroutine. An illustration of the part follows the example. (Fig. 7.5)

THREADING WITH THE U-HEAD

Figure 7.6 illustrates cutting an inner diameter thread using a number of passes to achieve the final finished depth. The insert has a 10° feed into the part

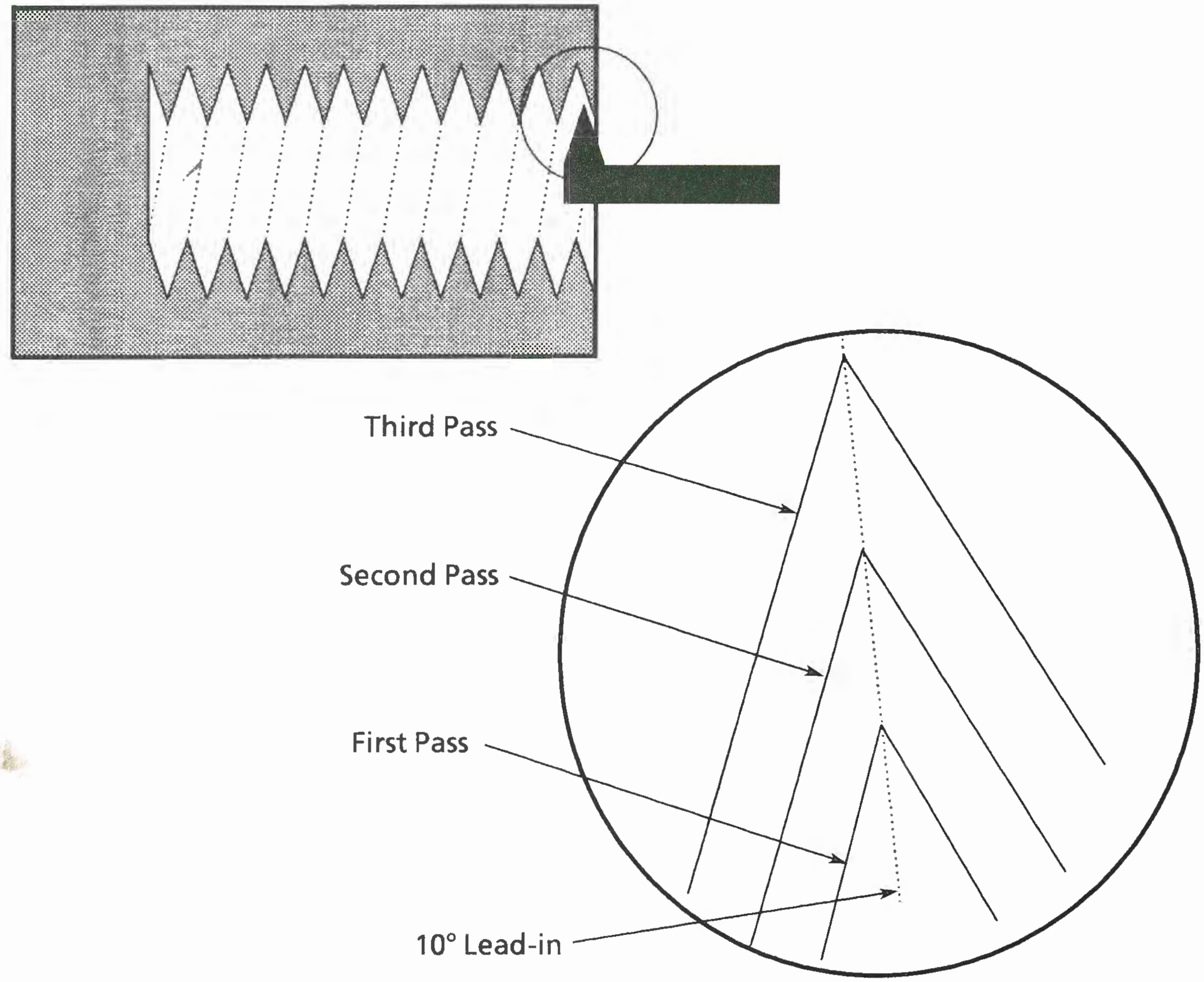


Fig. 7.6 THREAD-CUTTING EXAMPLE

surface. The 10° lead into the part is used to reduce cutting stress on the insert. The programmer can create a user-defined cycle to suit the needed program conditions, as shown in the examples that follow.

The programmer must coordinate the motions of the linear axis and the machine spindle thru the use of the G33 command.The use of the G33 command will cause the system to track the motion of the spindle, the Z axis, and coordinate it with the motion of the U axis. If an axis command is entered for some other axis other than the U or Z axis while the G33 is modal an error code "Illegal Threading Axis" message will be displayed. The axis velocity commanded in a program block containing a G33 command must not exceed 55% of the rapid value for that axis.

Synchronizing the Axis

When the programmer commands G33, axis motion will not begin until the programmed location has been met. Once the system arrives at the programmed location the control will then execute the G33 command. The G33 command may be cleared thru hitting the [CLEAR ALL LOGIC] key or thru programming a feedrate. If a feedrate command exists in the same block as a G33 command the error code "F COMMAND CANCELS G33" will be displayed.

If the programmer wishes to construct multiple lead threads the Type II data statement (SSA, angle in degrees) is used to command the control to move the spindle to the required starting angle in degrees. The

(SSA,) command must be given while the G33 is modal. The programmer should know that due to following error, axis and spindle acceleration, and the beginning of coordinated motion before reaching the part surface, the (SSA,) command produces an incremental angular position from the initial start position.

Example: (SSA,180)

This command would cause the control to start the second thread lead 180° from the first lead.

The distance that the U axis moves during each revolution is commanded in the block containing the G33 thru the use of the letter "E". If that block containing the G33 does not have an E value the control will generate an error message "LEAD SPECIFICATION REQUIRED". If a "D" command is contained in a G33 block the D information is the amount of change which will occur to the E value. If the following command is given:

Example: G33E.1D.01Z-2.000

The control will move the tool .11 during the first revolution,.12 during the second, etc. from the start point until the Z value has been met. If a negative D value is programmed the control will generate the error "FEEDRATE IS 0 BEFORE END POINT IS REACHED".

If successive program blocks containing D commands, following a block with an E command, do not contain E

```
(DFC,G833,SAVE)
(IFT,NOTDEF(L805)ORNOTDEF(L813)ORNOTDEF(L819),M0M99)      !Error Illegal Format
(PAR,L827 = 1.,L839 = L802,L840 = L803)
(IFT,G71,(PAR,L827 = 25.4))              !Metric conversion
(PAR,L829 = L802 + L803)                 !Total rough and finish passes
(PAR,L830 = L821/2.-L801)                !U clearance plane
(PAR,L831 = (L822-L821)/2.)              !Thread depth
(PAR,L832 = L802-2.)                     !Total rough passes-First rough pass + Last rough pass
(PAR,L833 = (L831-L809-L810)/L832        !Average depth of successive passes
(PAR,L834 = (L831-L809/TAN(80.))         !Total Z depth, Feed in at 10° angle
(PAR,L835 = L818 + L834 + .5*L827)       !Z start for G41 ramp-on
(PAR,L836 = L818 + L834 + .25*L827)      !Z Position for first cut at ramp-on
(PAR,L837 = L821/2.)                     !U position minor diameter
(PAR,L838 = L837 + L809)                 !U cut position
(IFT,L835>Z,ZL835F0)
G18G21G78XL824YL825SL819ML813
(LBL,1)ZL835
UL838
G41ZL836
G01G33EL805ZL826
UL837F0
G42UL830
(PAR,L839 = L839-1.)
(IFT,L839 = 1.,(GTO,2))
(IFT,L839<1.,(GTO,3))
(PAR,L838 = L838 + L833,L835 = L835-L833/TAN(80.),L836 = L836-L833/TAN(80.))
(GTO,1)
(LBL,2)(PAR,L838 = L838 + L810,L835 = L835-L810/TAN(80.),L836 = L836-L810/TAN(80.))
(GTO,1)
(LBL,3)(PAR,L840 = L840-1.)
(IFT,L840> = 0,(GTO,1))
G40ZL835
(END)
```

Fig. 7.7 EXAMPLE of INNER DIAMETER THREADING G833 CYCLE

```
G70G90G48H1
Z20.M25
B180.M25
(IFT,L3030<>1.,T1M6)      !If the active tool in the spindle is not tool 1, load tool 1.
(CLS, TIP-OFFSET)   !This subroutine has been defined previously (see Figures 7.4 and 7.5).
S1000M3M7
X0Y0F0
(PAR,L1 = 2.767"MINOR DIAMETER",L2 = 2.875"MAJOR DIAMETER")
G41U,L1/2.,Z.25       !Finish bore minor diameter(for this example only). Normally, the minor diameter is
                      finished with a different tool.
Z-1.5F10.
U,L1/2.-.1,F0
G40Z.5
G883A.1B8.C2.I.03J.001UL1VL2E,1./12.,R0X0Y0Z-1.S1000M33      !This command calls the user defined
                                                            G834 cycle and loads data (the parameter
                                                            values) into the cycle operation.
G80
G99H0
Z20.M25
B0F0
M30
```

Fig. 7.8 EXAMPLE PROGRAM USING G833 CYCLE TO MAKE AN INNER DIAMETER THREAD

commands, the control will assume that the initial E value is modal until canceled with some other value. The control will permit D commands in a block of information while the G33 command is modal. If a D is commanded when G33 is not modal an error message "D COMMAND ALLOWED IN G33 ONLY" will be displayed.

The next 2 examples are demonstrations of threading with a user-defined cycle (G833, G834) to cut inner and outer diameter threads. Both are straight 60° V-threads with a 10° feed into the part surface.

Example of an Inner Diameter Thread

Format: G833 A B C I J U V E R X Y Z S M

where:

- A is the U axis clearance position (PAR L801)
- B is the number of rough passes (PAR L802)
- C is the number of spring passes (PAR L803)
- S is the spindle speed (PAR L819)
- M is spindle on CW or CCW, (PAR L813)
- I is the first rough pass start depth (PAR L809)
- J is the finish pass depth (PAR L810)
- U is the minor diameter (PAR L821)
- V is the major diameter (PAR L822)
- E is the pitch (PAR L805)
- R is the part face (PAR L818)
- X is the X axis position (PAR L824)
- Y is the Y axis position (PAR L825)
- Z is the Z axis end position (PAR L826)

Example of an Outer Diameter Thread

Format: G834 A B C I J U V E R X Y Z S M

where:

- A is the U axis clearance position (PAR L801)
- B is the number of rough passes (PAR L802)
- C is the number of spring passes (PAR L803)
- S is the spindle speed (PAR L819)
- M is spindle on CW or CCW, (PAR L813)
- I is the first rough pass start depth (PAR L809)
- J is the finish pass depth (PAR L810)
- U is the minor diameter (PAR L821)
- V is the major diameter (PAR L822)
- E is the pitch (PAR L805)
- R is the part face (PAR L818)
- X is the X axis position (PAR L824)
- Y is the Y axis position (PAR L825)
- Z is the Z axis end position (PAR L826)

```
(DFC,G834,SAVE)
(IFT,NOTDEF(L805)ORNOTDEF(L813)ORNOTDEF(L819),M0M99)        !Error Illegal Format
(PAR,L827 = 1.,L839 = L802,L840 = L803)
(IFT,G71,(PAR,L827 = 25.4))                   !Metric Conversion
(PAR,L829 = L802 + L803)                      !Total rough and finish passes
(PAR,L830 = L822/2. + L801)                   !U clearance plane
(PAR,L831 = (L822-L821)/2.)                   !Thread depth
(PAR,L832 = L802-2.)                          !Total rough passes-First rough pass + Last rough pass
(PAR,L833 = (L831-L809-L810)/L832)            !Average depth of successive passes
(PAR,L834 = (L831-L809)/TAN(80.))             !Total Z depth, Feed in at 10° angle
(PAR,L835 = L818 + L834 + .5*L827)            !Z start for G41 ramp-on
(PAR,L836 = L818 + L834 + .25*L827)           !Z position for first cut at ramp-on
(PAR,L837 = L822/2.)                          !U position minor diameter
(PAR,L838 = L837-L809)                        !U cut position
(IFT,L835>Z,ZL835F0)
G18G21G78XL824YL825SL819ML813
(LBL,1)ZL835
UL838
G42ZL836
G01G33EL805ZL826
UL837F0
G41UL830
(PAR,L839 = L839-1.)
(IFT,L839 = 1.,(GTO,2))
(IFT,L839<1.,(GTO,3))
(PAR,L838 = L838-L833,L835 = L835-L833/TAN(80.),L836 = L836-L833/TAN(80.))
(GTO,1)
(LBL,2)(PAR,L838 = L838-L810,L835 = L835-L810/TAN(80.),L836 = L836-L810/TAN(80.))
(GTO,1)
(LBL,3)(PAR,L840 = L840-1.)
(IFT,L840>0,(GTO,1))
G40ZL835
(END)
```

Demonstration of the G834 threading cycle

```
G70G90G48H1
Z20.M25
B180.M25
(IFT,L3030<>1.,T1M6)      !If the active tool in the spindle is not tool 1, load tool 1.
(CLS, TIP-OFFSET)         !This subroutine has been defined previously (see Figures 7.4 and 7.5).
S1000M3M7
X0Y0F0
(PAR,L1 = 2.767"MINOR DIAMETER",L2 = 2.875"MAJOR DIAMETER")
G834A.1B8.C2.I.03J.001UL1VL2E,1./12.,R0X0Y0Z-1.S1000M3     !This command calls the user defined
                                                           G834 cycle and loads data (the parameter
                                                           values) into the cycle operation.
G80
G99H0
Z20.M25
B0F0
M30
```

Fig. 7.9 G834 EXAMPLE of OUTER DIAMETER THREADING CYCLE and PROGRAM

The fifth example shows the use of the spindle probe to update the tool tip radius length comp value of suffix number 7 in the tool table (Figures 7.10 & 7.11).

An example program (Fig. 7.12) using the spindle probe to update tool compensation follows the DFS, I. D. TIP-OFFSET UPDATE example below.

SPINDLE PROBE PROGRAMMING

This section covers spindle probe commands that are used in KT Advanced Programming Language. Information covering basic spindle probe programming is contained in Pub 878 Part Programming, Operating and Maintenance Manual.

The programmer has the option of creating custom designed probing cycles by the use of subroutines, defined cycles and Type II data statements.

The following Type II data statements are associated with probe cycle commands:

1. (OIF)-Open Inspection File -Used with a G36
2. (MSG,NAM)-same as OIF, used with a G36
3. (OPF)-same as OIF, but does not require G36
4. (FWR)-Write data into an open inspection file
5. (CLF)-Allows the user to close a file
6. (PRT)-Print any ASCII format file on a printer
7. (SER)-Stop part program on process device errors
8. (CER)-Continue program on process device errors
9. (CPY)-Copy the part program

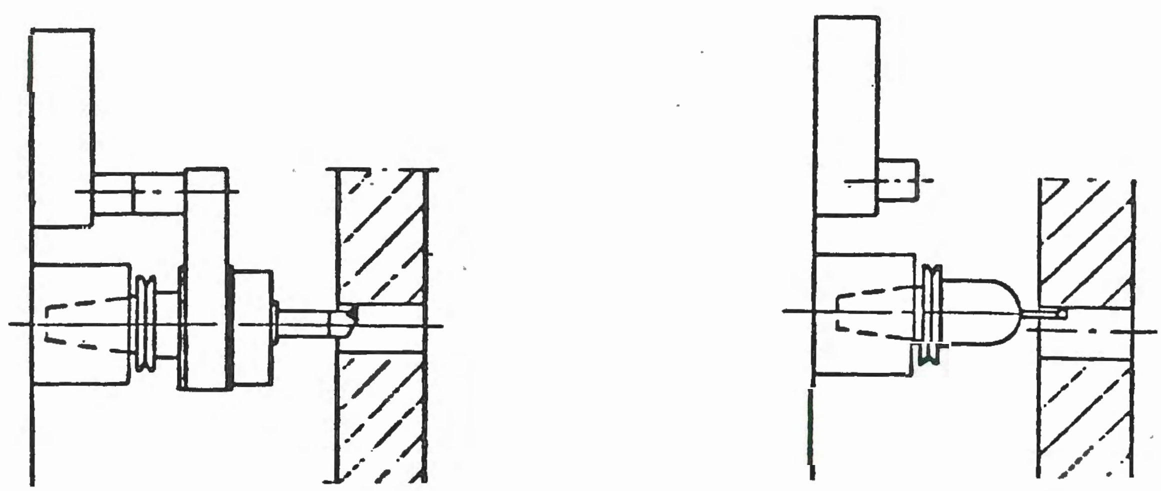


Fig. 7.10 U-Head and Spindle Probe

```
(DFS,I.D.TIP-OFFSET UPDATE,SAVE)
(UAT,7)
(PAR,L1 = L3018TIP-OFFSET)
(PAR.L100 = L3018"LAST TIP DIAMETER COMP")!SAVE CURRENT TIP DIAMETER COMP VALUE FOR UPDATE
(UAT,0)
M25U0
G98UL1
(END)
```

Fig 7.11 Tip Offset Subroutine used in the Probing Operation

```
G90G94G70G00B180.F0H0
G48H2
(PAR,L15=2.135"PROGRAM BORE DIAM")
(LBL,10)Z20.M25
(IFT,L3030<>1.,T1M6)            !T1 U-Head Tool
(CLS,I.D.TIP-OFFSET UPDATE)     !Calls previously defined subroutine
T2M7X3.Y3.F0S1000M3G18G21U,L15/2.
G41Z12.4
F10.Z11.2
U1.F9.
Z12.4F0G40M9
G99
Z20.U0M25
T2
M6
!The following program blocks are inserted to wash the chips out of the bore.
T77777G17G90G94G0X3.Y3.F0S30M3
Z6.M58
Y-3.
G04F2.
M9
Z20.M25M9

T77777
M6 !Load Spindle Probe
G90G94G00G17G48H2G70
G28X3.Y-3.F0        !The spindle probe is used to probe a bushing on the part fixture
G29
Z-.25F0
G27X20.       !The probe is commanded to move in X and then in Y
G27Y20.
G27X20.R2.1258      !Establish X axis probe ball radius
G27Y20.R2.1258      !Establish Y axis probe ball radius
Z.2G28F0
Y3.
G29
Z-.25
G27X20.
G27Y20.
G27X20.
(PAR,L17=L1005-L15"DIAM DEVIATION",L18=L100+L17/2."NEW TIP DIAMETER  COMP")
(PAR,L16=L1005"BORE DIAMETER")
(TLD,T1.7DL18)      !Re-establish T1.7 U-Head length comp.
H0
Z20.M25
(IFT,L17<-.0015,(GTO,10))      !If bore is undersize, recut with updated compensation.
(IFT,L17>.0015,M99M0)              !Bore oversize - Cycle Stop, sound alarm.  To continue, press Cycle Start
                                   and program will abort.
(IFT,L17>.0015,(GTO,100))      !Abort program.
(LBL,100)Z20.M25
H0G99X12Y19.
B0
M30
```

Fig. 7.12 UPDATING TIP RADIUS WITH SPINDLE PROBE

The following G commands may be used by the programmer to create custom designed cycles or subroutines:

NOTE

The G24, G25, G26 and G27 cycles cannot call or be called by other cycles. Only subroutines may call a cycle.

1. G22 to record all axis positions upon probe contact. Non-modal (NM). For G22 example, see Chap. 9.
2. G23 to record all axis positions upon probe non-contact. (NM)
3. G24 for tolerance verification. (NM)
4. G25 for automatic regrid. (NM)
5. G26 for single surface detection. (NM)
6. G27 for automatic position between surfaces. (NM)
7. G61 to record number of probe contacts. Modal (M)
8. G62 to record number of probe non-contacts. (M)

NOTE

These contacts or non-contacts are recorded in probe system parameter L1022. The count is reset to zero upon programming either a G61 or G62.

9. G60 is used to stop the count.

In addition to the spindle probe commands discussed in Chapter 2, KT Advanced Programming Language uses the following commands.

NOTE

All probe cycles must be programmed in a block by themselves. It is not permitted to have other G or M commands in the same block as a spindle probe command.

A tool length compensation equal to or greater than 4.5072 inches (114.483 mm) must be entered into the tool table before beginning the use of the spindle probe.

The control automatically defaults to 100 ipm in probe cycle operations unless some slower rate is programmed.

G24 TOLERANCE VERIFICATION

G24 X, Y, Z, A, B, C, XY is a tolerance verification for the indicated axes. It checks for deviation between programmed and absolute position within a programmed tolerance zone. This command is discussed in Pub 878 Part Programming and Operating Manual for the Spindle Probe in section 2.8.

G25 AUTOMATIC REGRID

G25 Automatic Regrid automatically resynchronizes all linear axes reference positions. The control is updated. This command is discussed in Pub 878 Part Programming and Operating Manual for the Spindle Probe in section 2.8.

G25R CALCULATE BALL RADIUS AND GRID PRESET

The G25R command calculates Ball Radius and Grid Preset. This command is used to determine the stylus ball radius and establish the grid presets. It is similar to the G25 command, but includes the ball radius calculations. This command requires the use of an R-word which is the accurately known distance between two surfaces. This command is discussed in Pub 878 Part Programming and Operating Manual for the Spindle Probe in section 2.9

G26TZ CALCULATE EFFECTIVE SECONDARY PROBE LENGTH

The G26TZ command is used to determine the effective probe length of a second probe that is being used in the tool magazine or when the primary probe stylus has been changed for any reason. The designation of T0 (zero) is used to trigger the operation of the cycle. The probe is automatically positioned to the X,Y coordinates of the G25RZ preset position, the Z command that is programmed is then given that will bring it into contact with the tool post or target bushing. The Z command is such a value that it is beyond the surface of the bushing or tool post. After the probe makes contact the secondary or primary probe length is recorded by the control in the tool table. The G26TZ command can only be used after a G25R command has been used to establish the position of either the tool post or the target bushing. It is a non-modal command.

Example: N100G26T0Z-5.

G26 SURFACE DETECTION

1. G26X is used for surface detection in the X axis.
2. G26Y is used for surface detection in the Y axis.
3. G26Z is used for surface detection in the Z axis.

G26 X,Y,Z programming examples are discussed in PUB 878 in section 2.10.

G26XY ANGULAR SURFACE DETECTION

The G26XY Angular Probe Cycle is used to detect an angular surface in the XY plane.

Example:

N110 X5.Y5	G26XY start point
N120G26X6.Y6.	(This point is beyond the part surface to ensure probe contact)

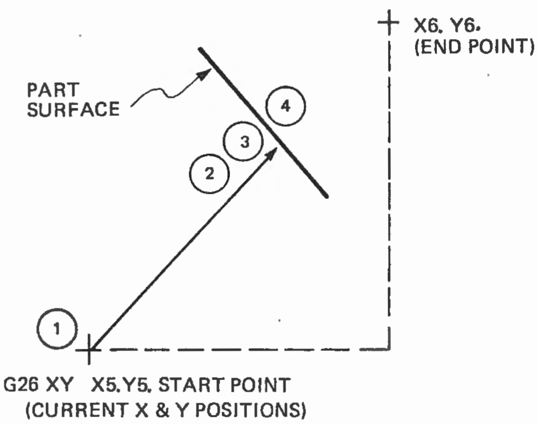


Fig. 7.13 G26XY VECTORED MOVE

In the illustration above, the start point is at X5.Y5. The probe is programmed past the actual part surface X6.Y6. The probe will come into contact with the part, touch the surface, back off, contact the surface again and stop with the spindle centerline on the part surface. For best results the programmed path should be at a right angle to the part surface being measured. Also, a G27XYR should be performed before a G26XY cycle to determine angular ball radius for parameter L1003. The G27XYR must include some accurately known radius of a target bushing and must be parallel to the vector path being used by the G26XY cycle.

G26RA ANGULAR SURFACE DETECTION

The G26RA is a command which is used to detect a surface at some specified distance (the R-word) at some specified positive angular displacement (the A-word). The angle that is used in this command is generated by using the 3 o'clock position as zero and moving in a counterclockwise direction to the desired point to generate an angular measurement that is always positive. The R-word is some incremental value that will take the probe stylus to a positive contact with the part surface.

Example: N125G26R2.A135.

In this example the probe would be directed to orientate 135° from the base line and then execute a move of 2 inches. Before it could complete that move it would come into contact with the part. Once again the probe path should be at a right angle to the part surface and the angular ball radius should be calculated with a G27XYR before the G26RA.

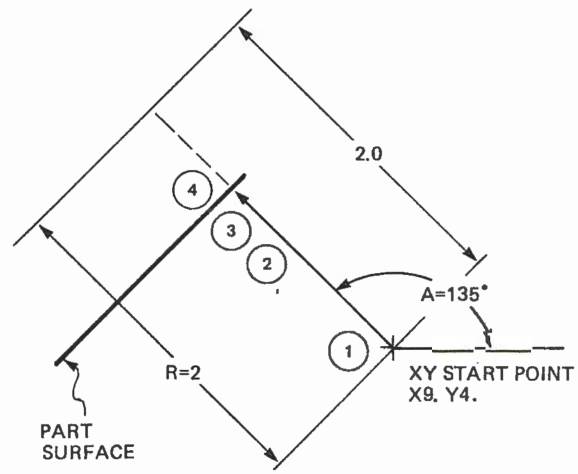


Fig. 7.14 G26RA ANGULAR SURFACE DETECTION

G26YZA (E) AUTOMATIC SURFACE ALIGNMENT A AXIS

This command permits the programmer to program a cycle which will automatically align the designated axis to be moved so that the work surface is parallel to the spindle face. The control recognizes information in the YZ plane and the A axis.

See the G26XZB (E) description for a detailed explanation of this cycle.

G26XZB (E) AUTOMATIC SURFACE ALIGNMENT B AXIS

This command permits the programmer to program a cycle which will automatically align the designated axis to be moved so that the work surface is parallel to the spindle face. The control recognizes information in the XZ plane and the B axis. This command is explained below.

Example: N230G90G70
N240G26X6.Z-5.B0E-1.5

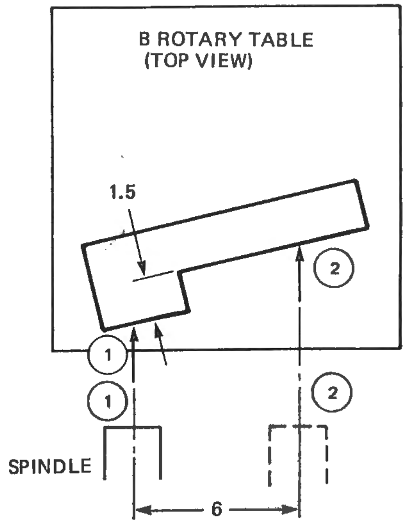


Fig. 7.15 AUTO B AXIS ALIGNMENT

This command is defined as follows:

1. X is the required incremental distance value.
2. Z is some value past the part surface.
3. The B0 command is used to trigger the cycle
4. The E command is optional, but is required when aligning parts with stepped faces or uneven surfaces. The E value is equal to difference between the surface of the part being probed and the elevation of the step above the probed surface.

The command is executed in this order:

1. The probe moves in the Z axis to contact the part surface from point 1.
2. After contact is made the probe reverses direction and contacts the part again at a reduced feedrate.
3. The probe retracts from the part surface to either the Z start position or the Z pullback position whichever is greater.
4. The probe moves in the X axis the commanded incremental distance to point 2.
5. The probe moves in the Z axis to contact the part surface.
6. Once contact has been made with the part surface probe reverses direction and contacts the part again at a reduced feedrate.
7. The probe retracts from the part surface to either the Z start position or the Z pullback position whichever is greater.
8. The B axis rotates so that the part surface is perpendicular to the spindle centerline.
9. If desired a fixture offset may be entered for the B axis.

G26YZC (E) AUTOMATIC SURFACE ALIGNMENT IN C AXIS

The G26 YZC (E) allows the programmer to align the part in the C axis. The operation of this command is similar to that of G26XZB (E) shown above except that the control will execute commands in the YZ plane and the C axis.

G27X AUTOMATIC POSITION BETWEEN SURFACE

The G27X is used to determine the centerpoint between two surfaces in the X axis. Programming instructions for this command are contained in the Part Programming and Operating Manual for the Spindle Probe Pub 878.

G27Y AUTOMATIC POSITION BETWEEN SURFACES

The G27Y is used to determine the centerpoint between two surfaces in the Y axis. Programming instructions for this command are contained in the Part Programming and Operating Manual for the Spindle Probe Pub 878.

G27R BALL RADIUS CALCULATION (X AXIS OR Y AXIS) (G27XR OR G27YR)

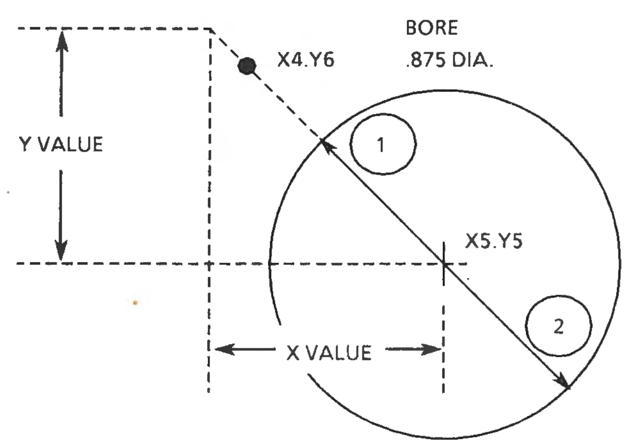


Fig. 7.16 G27XY ANGULAR PROBE

The G27 ball radius command is used to calculate the stylus ball radius in which the R-word is the accurately known diameter of either a target hole or bushing. A separate ball radius must be commanded for each axis. Programming instructions for this command are contained in the Part Programming and Operating Manual for the Spindle Probe Pub 878.

G27XY ANGULAR PROBE BETWEEN SURFACES

G27XY Angular Probe Cycle is used to detect an angular surface in the XY plane.

Example:

N110X5.Y5	!Start point
N120G27X4.Y6.	!This point is beyond the part surface to insure probe contact.

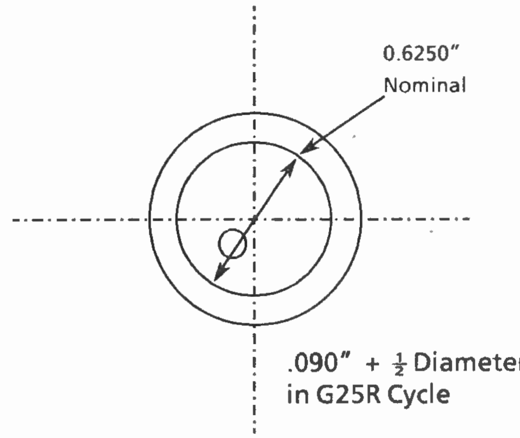
The procedure is described as follows:

1. The spindle probe is positioned near the midpoint of the surfaces to be checked in the XY plane and may be positioned at some Z depth below the part surface to insure contact with the probe stylus, but the probe is not in contact with the part surface.
2. The spindle probe is commanded to an XY position which is past a point that will bring the probe into contact with the part surface.
3. When the probe makes contact with the part surface at the programmed feedrate or default feedrate the probe will touch the part, back away, move toward the part again, and make contact.
4. The spindle will move away from the part surface returning to its start point, rotate 180°, move to the opposite part surface and repeat the touch, back away, touch procedure again.
5. Finally the probe will be positioned at the centerpoint between the part surfaces.
6. If desired a fixture offset or G98 value maybe entered into the control.

For best results the programmed path should be at a right angle to the part surface being measured. Additionally, a G27XYR should be performed before the G27XY cycle to determine the angular ball radius for parameter L1003. The G27XYR must include some accurately known diameter of a target bushing and must be parallel to the vector path being used by the G27XY cycle.

G27XYR ANGULAR BALL RADIUS CALCULATION

The G27XYR ball radius command is used to calculate the stylus ball radius in which the R-word is the accurately known diameter of either a target hole or bushing. To perform this command the XYR probe path should be parallel to that of the G27XY or G27RA command for which the ball radius is being established.



Probe will move 10.16mm (0.4") in positive X-axis direction prior to Z-axis motion into hole. Thus, the initial X-axis probe position must be near the left edge of the hole (as viewed from the spindle).

Fig. 7.17 G27XYR BALL RADIUS CALCULATION

Example:
N120 G27X5.Y4.R.6205

The G27XYR command operates in a similar fashion to the G27XY command with the addition of the R-word which is used to define the accurately known diameter of a bushing on a fixture or the broken tool detection post. Its operation is the same as a G27XR or G27YR except that a vectored path is generated by the XY command.

The sequence of operation is as follows:

1. The spindle probe is commanded to near the centerpoint of the bushing and given a Z command that will take it below the part surface.
2. The control executes the move to the programmed XY location which should be beyond the part surface where it goes thru the surface detection cycle of touching the part surface, backing away and touching the part surface again.
3. The spindle probe will then return to the centerpoint position.
4. The probe will rotate 180° and perform same operation on the opposite side of the part.
5. The probe will return to the centerpoint automatically.

G27RA ANGULAR PROBE BETWEEN SURFACES

The G27RA is a command which is used to detect a surface at some specified distance (the R-word) at some specified positive angular displacement (the A-word). The angle that is used in this command is generated by using the 3 o'clock position as zero and moving in a counterclockwise direction to the desired point to generate an angular measurement that is always positive. The R-word is some positive incremental value that will take the probe stylus to a positive contact with the part surface. See the example below:

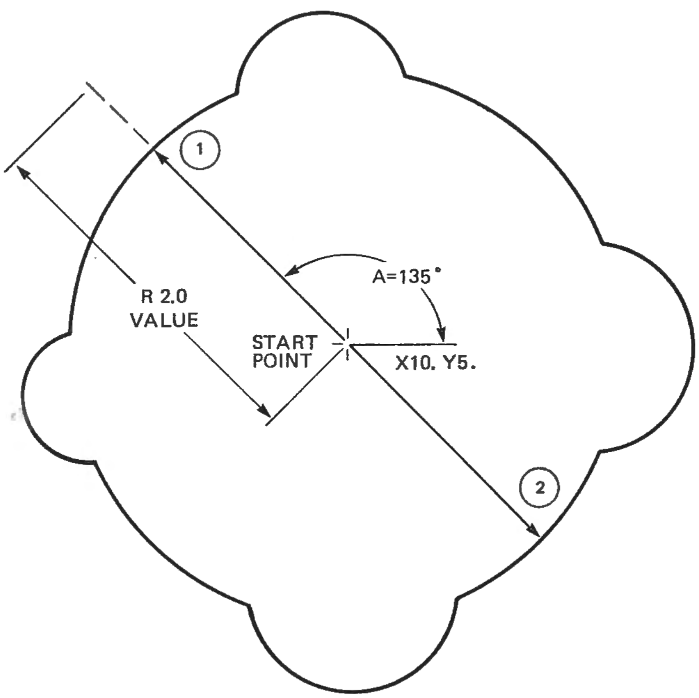


Fig. 7.18 G27RA ANGULAR PROBE between SURFACES

Example:
N125G27R2.A135.

Explanation of the G27RA command is as follows:

1. A previous XY command positions the probe to near the centerpoint of the surface to be probed at some specified Z depth.
2. The control rotates the probe to the commanded angle and moves the commanded positive incremental distance defined by the R-word.
3. Once contact is made the probe will execute the surface detection cycle of touching the part, backing away, and touching again.
4. After the surface detection cycle has been completed the probe will return to the centerpoint, rotate 180° and repeat the procedure on the opposite side of the part.
5. Upon completion of this cycle, the probe is moved to the centerpoint between the two surfaces.

G27B PROBE BETWEEN SURFACES IN THE B AXIS

NOTE

The G27B and G27C probe cycle commands will accept axial feedrate commands up the maximum possible for each individual machine system. If a feedrate command is not specified the control will default to 360 degrees per minute (DPM) for the first contact. The second contact will be made at 10.1 DPM. Additionally, the X axis must be at the centerline of the rotary axis.

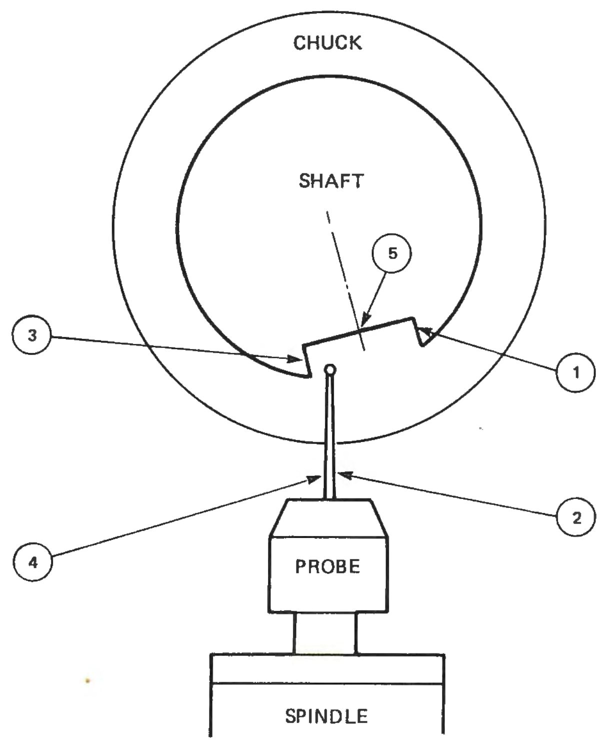


Fig. 7.19 OVERHEAD VIEW OF B AXIS TABLE

Example:
N430G27B90.F500.

The sequence of events in the above command would be as follows:

1. The B axis would first rotate to the zero° position at 500 DPM until contact is made. When contact is made the B axis will back off and a second contact will be made at 10.1 DPM.

2. The B axis will then rotate to the start position. The spindle probe will rotate 180°.
3. The B axis will rotate in the opposite direction until contact is made at the commanded feedrate. A second contact will be made at 10.1 DPM.
4. The control calculates the rotary deviation and commands the B axis to rotate so that centerline of the spindle will be in the center of measured surface.

G27C PROBE BETWEEN SURFACES IN THE C AXIS

This command is similar to the previous G27B axis command except it is executed in the C axis. Also the Y axis must be at the centerline of the C axis.

G27XZE PROBE A HUB IN THE X AXIS

This cycle is explained in the Part Programming and Operating Manual for the Spindle Probe Pub 878 Section 2.12.

G27YZE PROBE A HUB IN THE Y AXIS

This cycle is explained in the Part Programming and Operating Manual for the Spindle Probe Pub 878 Section 2.12 and is similar to the G27XZE cycle except it is operational in the Y axis.

G27XYZE PROBE A HUB IN THE X AND Y AXIS

The G27XYZE allows the programmer to command the spindle probe to a part at some XY coordinate as shown in the illustration below. A description of the cycle operation follows the illustration.

Example:
N450G27X7.Y12.Z3.12E2.34

The control would carry out this command in the following fashion:

1. A previous XYZ command would position the spindle to the approximate centerline of the obstruction, but would be above the part.
2. Next the control would execute the move to the commanded XY position at the specified E radius clearance position from the centerline of the obstruction.
3. The control would then move in the Z the specified distance.

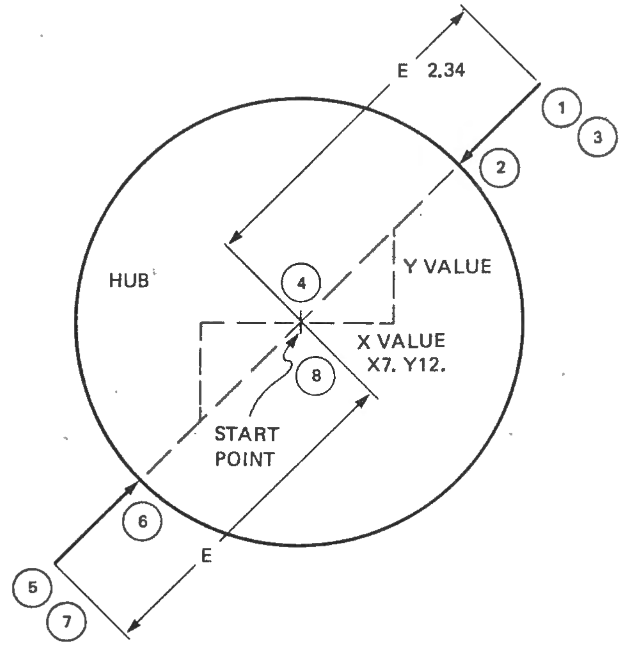


Fig. 7.20 G27XYZE VECTORED MOTION

4. The spindle probe then moves in the XY plane toward the obstruction until contact is made.
5. Once contact is made the control goes thru the surface detection cycle.
6. After completion of the detection cycle the control commands the spindle probe to retract in the reverse order of its entrance to the part surface.
7. The probe rotates 180° and repeats the procedure in opposite direction.
8. The E word is define as: some value greater than the radius of the obstruction, plus the radius of the stylus ball, plus some clearance value. It is always a positive value.

G27RAZE ANGULAR HUB DETECTION CYCLE

The G27RAZE command is used to detect a centerline and the surface separation of a hub or obstruction at some designated angle and radius from a commanded position. The G27RAZE is formatted as shown:

Example:
N670 G27R.75.A315.Z-.5E2.

The cycle is defined as follows:

1. The control recognizes the letter A as some angular dimension assuming the 3 o'clock position as zero.

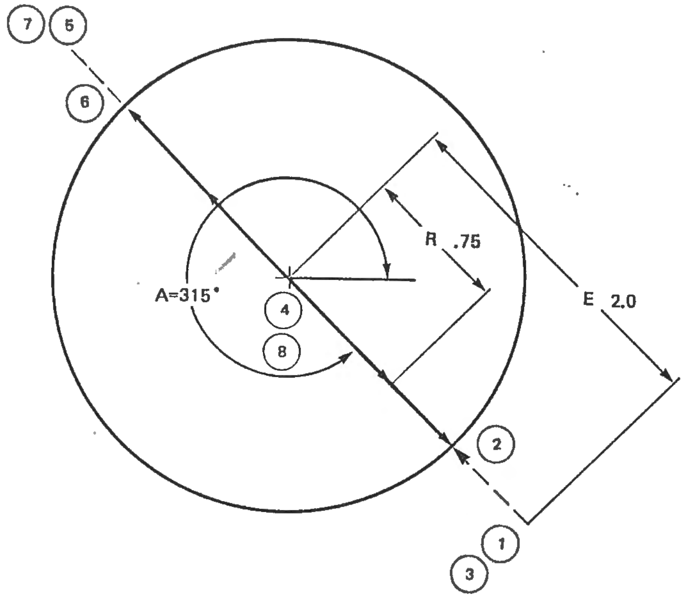


Fig. 7.21 G27 RAZE DETECTION CYCLE

2. The R word is some positive incremental positive distance past the surface being probed.
3. The E word is the clearance position selected by the programmer also a positive number.
4. The Z word is the depth to which the probe stylus is positioned.

This block would function in the following fashion:

1. The spindle probe has been positioned to near the centerline of the obstructed bore by some previous XYZ commands.
2. The control would move the probe to a radius point that is 2" from the centerpoint (if in the inch mode) of the bore at an angle of 315° from the 3 o'clock position as viewed from behind the spindle face looking outward to the part surface.
3. The probe moves the Z distance, in this case -.5".
4. Finally, the control moves the probe towards the target the commanded distance of 1.25". (The distance of 2" minus .75".)
5. Once contact has been made the control reverses direction.
6. The spindle probe rotates 180°.
7. The process is repeated on the opposite side of the part.

G27XZE TREPAN GROOVE AND OBSTRUCTED BORE DETECTION IN THE X AXIS

This cycle is explained in the Part Programming and Operating Manual for the Spindle Probe Pub 878 Section 2.12.4

G27YZE TREPAN GROOVE AND OBSTRUCTED BORE DETECTION IN THE Y AXIS

This cycle is explained in the Part Programming and Operating Manual for the Spindle Probe Pub 878 Section 2.12.4 and is similar to the G27XZE trepan cycle except it is operational in the Y axis.

G27XYZE ANGULAR TREPAN GROOVE E AND OBSTRUCTED BORE DETECTION

The G27XYZE allows the programmer to command the spindle probe to a part at some XY coordinate as shown in the illustration below.

Example:
N360G27X7.Y5.Z3.12E2.34

The control would carry out this command in the following fashion:

1. A previous XYZ command would position the spindle to the approximate centerline of the obstruction, but would be above the part.
2. Next the control would execute the move to the commanded XY position at the specified E radius clearance distance from the obstruction centerline.

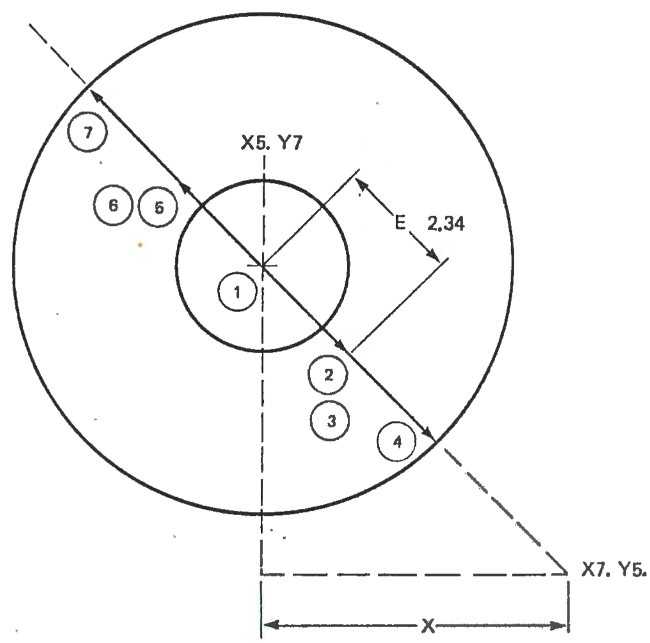


Fig. 7.22 XYZE ANGULAR TREPAN DETECTION

3. The control then moves Z the specified distance.
4. The spindle probe moves toward the obstruction until contact is made.
5. Once contact is made the control goes thru the surface detection cycle.
6. After completion of the detection cycle the control commands the spindle probe to retract in the reverse order of its entrance to the part surface.
7. The probe rotates 180° and repeats the procedure in opposite direction.
8. The E word is define as: some value greater than the radius of the obstruction, plus the radius of the stylus ball, plus some clearance value. It is always a positive value.

G27RAZE ANGULAR TREPAN GROOVE AND OBSTRUCTED BORE DETECTION

The G27RAZE command is used to measure a trepan groove at some designated angle and radius from a commanded position. The G27RAZE is formatted as shown in the following example.

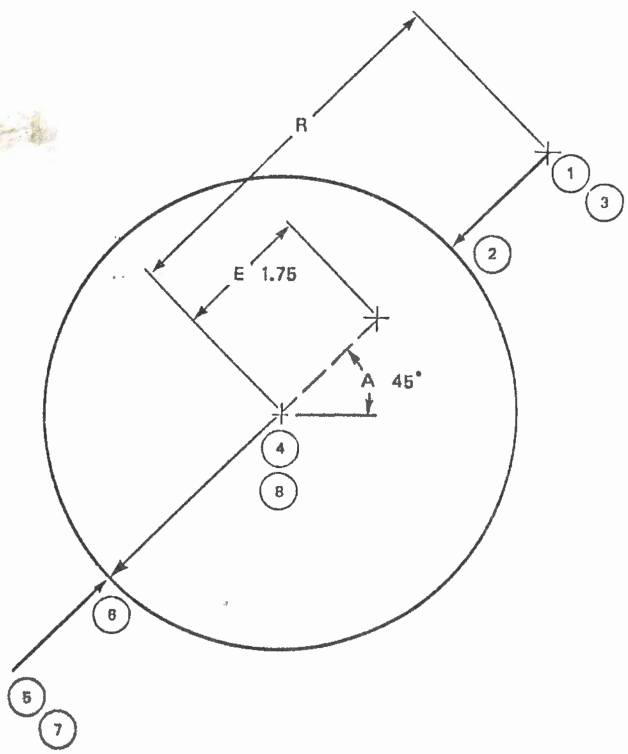


Fig. 7.24 RAZE ANGULAR TREPAN DETECTION

The cycle is defined as follows:

1. The control recognizes the letter A as an angular dimension assuming the 3 o'clock position as zero.
2. The R word is some positive incremental positive distance past the surface being probed.
3. The E word is the clearance position selected by the programmer also a positive number.
4. The Z word is the depth to which the probe stylus is positioned.

Example:
N390G27R3.5A45.Z-.65E1.75

This block would function in the following fashion:

1 The spindle probe has been positioned to near the centerline of the obstructed bore by some previous XYZ commands.
2. The control would move the probe to a radius point that is 1.75" from the centerpoint (if in the inch mode) of the bore at an angle of 45° from the 3 o'clock position as viewed from behind the spindle face looking outward to the part surface.
3. The probe then moves the Z commanded distance, in this case -.65.
4. Finally, the control moves the probe towards the target the commanded outwards away from the center of the part the distance of 3.5.
5. Once contact has been made the control reverses direction and retracts from the part surface in the reverse order from which it entered the part.
6. The spindle probe rotates 180°.
7. Process is repeated on the opposite side of the part.

TYPE II COMMANDS RELATED TO SPINDLE PROBE INSPECTION

K-T Advanced Programming Language will employ the use of several Type II command processors to perform the necessary disk, sequential, and remote I/O needed for the part program File Write feature. These commands can be used, without interrupting machine operation, to write/print/copy data directly to disk files from within a part program. The purpose of this section is to explain the operational description of each of these commands.

(OIF,filename)

The (OIF,filename) Type II command is used to create an ASCII disk file, with the given file name, for the purpose of recording process inspection data. A G36 command must first be programmed for the control to recognize an (OIF,filename) command. Note that the file name noted can contain a maximum of 32 alphanumeric characters, but can contain no commas, leading spaces, or trailing spaces.

Example:
N120G36
N130(OIF,PUMP341)

If no inspection file is open at the time this command is first encountered, the named file is created. Note, however, that if the named file is being opened for the second time, any previously stored information will be deleted. If this command is issued while an inspection file of the same name is already open and in use, the command is ignored; if an inspection file of a different name is already open at the time this command is encountered, the old file is first closed and the new file then created -- it is not possible to have more than one process inspection file open at any one time.

When an inspection file is created with this command, the following canned information is output to the file at the time it is first opened:

CONTROL SERIAL NUMBER = 345879

PALLET CODE/PART NUMBER = 3452

YY:MM:DD HH:MM
89:12:13 12:24 ←Example

The first entry indicates the serial number of the Control on which the file has been opened. The second entry indicates the pallet code number of the part on which the inspection process is being performed. The third entry indicates the date and time of day that the inspection file was created in the form YY:MM:DD HH:MM -- i.e., 89:12:13 would denote 13 December 1989 at 12:24 P.M. The dashed line is then output as a delimiter between the formatted output and any subsequent inspection data written to the file.

Any data can then be written to this file as is required. A file created with this command can be closed with either a G37 command or a (CLF) Type II command. Note that G37 cancels G36 (both are modal) and that if an inspection file is closed with a (CLF) command, the G36 will thus not need to be reprogrammed before the next (OIF,filename) command is executed.

(MSG,NAM,filename)

This Type II command produces the exact same results as the (OIF,<filename>) command, and can be used as a direct replacement. A G36 command must, again, be first programmed for this command to be recognized. The use of this command is retained to allow for downward compatibility with older systems.

(OPF,filename/type)

This command is used to create a disk file for the purpose of recording any type of formatted data. It differs from the (OIF,filename) command in that:

1. Either NC or ASCII files may be created.
2. No G36 command must first be programmed for this command to be recognized.
3. None of the formatted machine information output by the (OIF,filename) Type II processor is written to the file.
4. Variable file names are permitted through the insertion of program parameters directly into the filename, see following section.

This command is ignored if a file of the same name has already been opened and is in use, regardless of file type. Also, any previously opened file of a different name is first closed before the new file is created, and any existing and properly closed file of the same name and file type will be deleted before the new file is opened, thus erasing any previously stored information. Should a file with the same name, but different file type, exist at the time of the execution of this command, the message "FILE ALREADY EXISTS" will appear in the [MESSAGE] display, and the alarm will sound.

A file created with this command must be closed with a (CLF) command; see below.

Use of Variable File Types with the (OPF,filename/type) Command

The (OPF,filename/type) command may be used to create both NC and ASCII files; the /<type> field is optional.

Example: (OPF,PUMP123/AS)

NOTE

Should no field be supplied for /type, the file type will default to ASCII. Where the /type field is specified, an ASCII file must be denoted by "/AS", and an NC file by "/NC". Use of file types other than "/AS" or "/NC" will result in the message "ERROR - MUST BE EITHER NC prog OR ASCII" appearing in the [MESSAGE] display, and the alarm sounding.

Variable file names are permitted with the (OPF,filename/type) command through the insertion of program parameters directly into the filename.

Here, the filename noted may contain one or more program specified parameter, delimited by the ASCII characters < {left arrow} and > {right arrow}, and beginning with the character L. Any parameter specified with this command must have previously been defined in the part program or be a dedicated system parameter. The textual value of the parameter, as it appears in the actual filename, will be an 8-digit floating point representation.

Truncating with (LxxxI)

Besides simple parameter insertion, it is also possible to truncate the value of the associated parameter data that is used in the filename. Here, the character I is used to signify that the integer value of the parameter data is to be noted. The actual parameter value is not modified -- only that used in the file name is affected.

Parameter Rounding with LxxxR

If the programmer wishes to round parameter data to whole numbers for an OPF, DLF or FWR command, rather than using the rounding function described in Appendix A, an LxxxR may be used. For example:

```
(PAR,L23=3.9567,L24=5.)        !Defines the param
(OPF, PART<L23R>)           PART4.
(OPF, PART<L23R> <L24>)     PART45.000000
```

The use of the LxxxR format does not change the value of parameter held in the parameter table, but only modifies the data that is applied in the OPF, DLF or FWR command. For example, suppose that a part program includes the following lines:

```
N90 (PAR,L100=123)
N100(OPF,TESTFILE<L100>)    Output file name is
   :   :   :                TESTFILE123.00000
   :   :   :
N180 (CLF)
N190 (PAR,L100=123.456)
N200 (OPF,TESTFILE<L100>)   Output file name is
   :   :   :                TESTFILE123.45600
   :   :   :
N290 (CLF)
N300 (OPF,TESTFILE<L100I>)  Output file name is
   :   :   :                TESTFILE123
   :   :   :
N380 (CLF)
N390 (PAR,L200=987.6543)
N400 (OPF,<L100I><L200I>/NC)  Output file is
   :   :   :                  123987
   :   :   :
N480 (CLF)
```

Given this type of input, three ASCII files -- TESTFILE123.00000, TESTFILE123.45600, and TESTFILE123, respectively; and one NC file -- 123987 -- will be created and closed.

(OPF,filename,A/type)

This command is identical to the (OPF,filename/type) Type II command with the exception that if the named file is being opened for the second time, the previously stored information will not be deleted. Rather, the new information is appended onto the end of the old data. In this way, one file may be used a multiple number of times without the loss of any information. Note that if the named file does not exist the first time this command is encountered, a new file will be created just as if the append option had not been specified. The letter A is used to command the control to append the information to the previously opened file.

Example:
N150(OPF,PUMP341,A/AS)

A file created with this command must be closed with a (CLF) Type II command (see below).

(CLF)

This Type II command closes any file previously opened with the (OIF,filename), (OPF,filename/type), or (OPF,filename,A/type) Type II commands. Note again that the (CLF) command will not cancel any active G36 command, but that is does perform exactly as a G37 command in all other ways.

G37 functionality has been preserved to maintain tape compatibility with older systems. It is recommended that this command be used for all new part programs in lieu of G37, as G36 is modal and will thus never be removed once first programmed.

File access status parameter L1050 may be used with the (CLF) command to provide a way of allowing the part program to wait until after a file is closed before processing any more blocks. The part program would set parameter L1050 to 0 before the open file command. The part program would then wait after the (CLF) block until the file is closed by using an (LBL,10)(IFT,L1050=0,(GTO,10)) block. The system will set L1050 to non-zero after the file is closed. The usage of this parameter is optional. If the parameter L1050 is never created by the part program then the (CLF) will simply close the file and skip the setting of L1050 to non-zero.

Example:

```
N20(PAR,L1050=0)
    :   :   :
N100(OPF,PUMP123)
    :   :   :
INSPECTION PROGRAM
    :   :   :
N200(CLF)
N210(LBL,10)(IFT,L1050=0,(GTO,10))
```

(DLF,filename/type)

This command may be used to delete any NC or ASCII file from the system disk; this file not need be the same as was previously opened with an (OIF,filename) or (OPF,filename) command. The use of the file type identification is optional.

WARNING

THIS COMMAND WILL DELETE ANY NC OR ASCII FILE RESIDENT IN THE CONTROL DIRECTORY, INCLUDING PART PROGRAMS, SUBROUTINES, AND DEFINED CYCLES.

The file in question must first be closed. Should a file with the same name, but different file type, exist at the time of the execution of this command, the message "FILE ALREADY EXISTS" will appear in the Message Display, and the alarm will sound.

Example:

N250(DLF,PUMPPROGRAM0)	Deletes a program
N260(DLF,G881)	Deletes a user defined cycle
N270(DLF,DRILL80)	Deletes a subroutine

Use of Variable File Types with the (DLF,filename/type) Command

The (DLF,filename/type) command may be used to delete NC and ASCII files; the /type field is optional.

Should no field be supplied for /type, the file type will default to ASCII. Where the /type field is specified, an ASCII file must be denoted by "/AS",and an NC file by "/NC". Use of file types other than "/AS" or "/NC" will result in the message "ERROR - MUST BE EITHER NCprog OR ASCII" appearing in the Message Display, and the alarm sounding.

Variable file names are permitted with the (DLF,filename/type) command through the insertion of program parameters directly into the filename. Here, the filename noted may contain one or more program specified parameter, delimited by the ASCII characters < {left arrow} and > {right arrow}, and beginning with the character L. Any parameter specified with this command must have previously been defined in the part program or be a dedicated system parameter. The textual value of the parameter, as it appears in the actual filename, will be an 8-digit floating point representation. As is the case with the OPF command, the value of parameter data used may be truncated through the use of the "I" delimiter.

NOTE

Realize that the value of the actual parameter data must match that used in the original file name. If the data corresponding to the named parameter somehow changes between the time the file is created/closed and the time of the execution of this command, an error may occur as the name corresponding to the new value of the parameter data may not exist.

File access status parameter L1051 may be used with the (DLF,) command to provide a way for the part program to wait until after a file is deleted before continuing on. The part program would set parameter L1051 to 0 before the (DLF,). The part program would then wait after the (DLF) block until the file is closed with a (LBL,10) (IFT,L1051=0,(GTO,10)) block. The system will set L1051 to non-zero after the file is deleted. The usage of this parameter is optional. If the parameter L1051 is never created by the part program then the (DLF) will simply delete the file and skip the setting of L1051 to non-zero. For example:

```
N20(PAR,L1051=0)
    :   :   :
PART PROGRAM
    :   :   :
N200(DLF,ANYPART)
N210(LBL,10)(IFT,L1050=O,(GTO,10))
```

(MSG,<text>)

To allow for compatibility with older systems System 61 will recognize the MSG command. This Type II command is used to program any message text into a process inspection file. Note that a G36 command must first be entered for this command to be recognized by the Control. Here, the text included in the (MSG,text) command will appear at the next available line in the inspection file, followed by one empty line. If this command is programmed while no inspection file is open, the (MSG,text) command will be ignored and considered as simply an operator message.

As an example, suppose that a part program includes the following lines:

```
N90 G36
N100 (OIF,TESTFILE)
    :   :   :
N580 (MSG,THIS WRITTEN WITH "MSG".)
N590 (MSG,THIS IS A TEST.)
    :   :   :
N950 G37
```

Given this input, the file named TESTFILE will contain the following information:

```
CONTROL SERIAL NUMBER = 345879

PALLET CODE/PART NUMBER = 3452

        89:12:13  12:24

-------------------------------
  THIS WRITTEN WITH "MSG".
  THIS IS A TEST.
```

NOTE

The text specified can contain up to 80 alphanumeric characters, with the exception of Control Out and Control In characters, and that the system will truncate the output to 80 characters should more that 80 be specified in the (MSG,text) statement.

As another example, suppose that a part program includes the following lines

```
N100 (OPF,TESTFILE)
    :   :   :
N580 (MSG,THIS WRITTEN WITH "MSG".)
N590 (MSG,THIS IS A TEST.)
    :   :   :
N950 (CLF)
```

Given this input, an ASCII file named TESTFILE will contain the following information:

```
TESTFILE


THIS WRITTEN WITH "MSG".

THIS IS A TEST.
```

(FWR,text)

This command is used to program any text into a previously opened file. Unlike the (MSG,text) command, it is not necessary to program a G36 command prior to entering this command. Also, it is possible to insert program parameters and axis positions into these statements, and to append subsequent lines of (FWR) commanded text onto a single line in the file. Note that the system will truncate all data/text output on any single line of the file to 80 characters. Also note that if this command is

```
N100 (OPF,TESTFILE).
    :   :   :
N200 (FWR,"THIS WRITTEN WITH FWR.")         ! Text between single quotes
N210 (FWR,"ANY TYPE OF NEEDED TEXT.")       ! Note position of comma
N220 (FWR,"UPPER and lower case ALLOWED!")
N230 (FWR,"Insert a comma, just for demonstration.")
N240 (FWR,"Insert ""quotes"", just for demonstration.")
N250 (FWR,"(END)")
    :   :   :
N500 (CLF)
```

Given this input, an ascii file named TESTFILE will contain the following information:

```
  THIS WRITTEN WITH FWR.
  ANY TYPE OF NEEDED TEXT.
  UPPER and lower case ALLOWED!
  Insert a comma, just for demonstration.
  Insert "quotes", just for demonstration.
  (END)
```

Fig. 7.25 EXAMPLE OF THE USE OF (FWR,) STATEMENT

```
N700 (OPF,TESTFILE,A/AS)       ! Note the "append" option
    :   :   :
N870 (FWR,"THIS LINE CHOSEN TO TEST THE 'APPEND' OPTION.")
N880 (FWR,"Note that the previous data was not deleted!!")
    :   :   :
N950 (CLF)
```

Given this input, an ASCII file named TESTFILE will contain the following information:

```
THIS WRITTEN WITH FWR.
ANY TYPE OF NEEDED TEXT
UPPER and lower case ALLOWED!
Insert a comma, just for demonstration.
Insert "quotes", just for demonstration.

THIS LINE CHOSEN TO TEST THE 'APPEND' OPTION.
Note that the previous data was not deleted!!
```

Fig. 7.26 EXAMPLE OF THE USE OF AN (FWR,) STATEMENT WITH THE APPEND OPTION

programmed while no file is open, the warning message "PROCESS INSPECTION FILE NOT NAMED" is displayed, and the alarm is sounded.

For each type of (FWR,text) command, a comma (,) should be used to separate each field of the command statement. Commas, however, can also be included inside delimited text portions of (FWR,text) outputs. (See Figure 7.25).

Insertion of Free Format Text into (FWR,<text>) Commands

Any specified text may be written to an open file with the (FWR,text) command. Here, the quotation mark (") is a text delimiter. Any alphanumeric character may be used in the text stream. The double quotation mark ("") notes that one quotation mark is to appear in the text. (See Figures 7.25 and 7.26)

```
N100 (OPF,TESTFILE)
    :   :   :
N200 (PAR,L100 = 123.4567)
N210 (PAR,L200 = 9.8)
N220 (PAR,L300 = -45.67)
    :   :   :
N500 (FWR,L100)
N510 (FWR,"Parameter L200 equals ",L200)
N520 (FWR,"Parameter L300 equals ",L300," at this time.")
N530 (FWR,"Parameter L100, TRUNCATED, equals ",L100I)
N540 (CLF)
```

Given this input, an ASCII file named TESTFILE will contain the following information:

```
123.45670
Parameter L200 equals 9.8000000
Parameter L300 equals -45.670000 at this time.
Parameter L100, TRUNCATED, equals 123
```

Fig. 7.27 EXAMPLE OF THE USE OF AN (FWR,) STATMENT WITH PARAMETER SUBSTITUTION

Substitution of Program Parameters into (FWR,<text>) Commands

Any program specified parameter may be written to an open file with the (FWR,text) command (see Figure 7.27). Here, the character 'L' is used as a parameter delimiter. Note that parameter data and free format text may be included in one (FWR,text) command. Also note that any parameter specified in a (FWR) command must have previously been defined in the part program or be a dedicated system parameter.

In addition to simple parameter insertion, it is also possible to truncate the value of parameter data sent to a file. Here, the character 'I' is used to indicate that the *integer* value of the parameter data is to be noted. The actual value of the parameter is not modified, only the file data is affected. Note that leading zeros are suppressed in the display of parameter data.

Substitution of Axis Positions into (FWR,text) Commands

Any machine axis position may be written to an open file with the (FWR,text) command. Here, a comma followed by the letter designator of the requested axis indicates the output of that axis is desired. Note that the position written to the file will be in the units (inch or metric) active at the time the FWR statement is encountered in the part program, and that the position noted will include any G98, tool length/diameter comp, or fixture offsets that may be in effect. Thus, figures for axis positions stored in the file will be identical to those the main display. Also note that parameter data, free format text, and axis positions may all be included in one (FWR,text) command.

Continuation of Subsequent (FWR,text) Text Streams onto a Single Line

Multiple lines of (FWR,text) commanded text/data input may be written onto a single file line when the continuation option is specified. Here, the semicolon character ; signifies continuation. Note that this special delimiter MUST appear as the last input stream character before the final Control In character; it can, however, be used as a text character inside a pair of free format text delimiters.

(PRT,filename)

This Type II data statement can be used to send any ASCII or NC file to the printer from within a part program. The named file must reside on the MCU

```
N10 (OPF,TEST/NC)            !Open N/C file named TEST
    :   :   :
N50(PAR,L1 = 10000.)         !Truncation constant X.4 format for linear axis
N60(PAR,L1050 = 0)           !Status parameter for closing file
N70(PAR,L2 = (5. + 2.)*L1)   !X position truncated
N80(PAR,L3 = (6. + 3.)*L1)   !Y position truncated
N90(PAR,L4 = (12.-3.)*L1)    !Z position truncated
    :   :   :
        !  Write the following to TEST file
    :   :   :
N120(FWR,"N10G01G70G90G94X10.Y8.F0")
N130(FWR,"N20X",L2I,"Y",L3I,"Z",L4I,"F15.5S2000M3M8")        !Note position of " marks
N140(FWR,"(END)")
N150(CLF)
N160(LBL,11)(IFT,L1050 = 0,(GTO,11))        !Close file before continuation
    :   :   :
N170(CLS,TEST)          !Call the subroutine just created
N180M2

THE FOLLOWING WAS PROCESSED FROM PRECEDING (FWR,...) STATEMENTS:

N10G01G70G90G94X10.Y8.F0
N20X70000Y90000Z90000F15.5S2000M3M8
(END)
```

Fig. 7.28 EXAMPLE OF THE USE OF AN (FWR,) STATEMENT WITH AXIS POSITION SUBSTITUTION

```
N10 (OPF,TESTFILE)
    :   :   :
N190 (FWR,"FIRST FILE OPEN.")
N200 (FWR,"WRITE TWO LINES ONTO ONE LINE .... ";)
N210 (FWR,"THIS IS THE SECOND INPUT.")
N220 (FWR,"A SEMICOLON CAN BE WRITTEN, ;, INSIDE A TEXT STREAM.")
    :   :   :
N300 (PAR,L500 = 12.3456)
N310 Y67.9 F0
N320 (FWR,"PARAMETER/AXIS EXAMPLE .... ",L500,;)
N330 (FWR," ----",Y)
N340 (CLF)
    :   :   :
N380 (OPF,TESTFILE,A)
N390 (FWR,"SECOND FILE OPEN.")
N400 (FWR,"WRITE TWO LINES ONTO TWO LINES .... ")
N410 (FWR,"THIS IS THE SECOND INPUT.")
    :   :   :
N530 (CLF)

FIRST FILE OPEN.
WRITE TWO LINES ONTO ONE LINE .... THIS IS THE SECOND INPUT.
A SEMICOLON CAN BE WRITTEN, ;, INSIDE A TEXT STREAM.
PARAMETER/AXIS EXAMPLE .... 12.345600 ---- 67.900000

SECOND FILE OPEN.
WRITE TWO LINES ONTO TWO LINES ....
THIS IS THE SECOND INPUT.
```

Fig. 7.29

mass storage device, and must have been previously closed with a (CLF) or G37 if used with a OIF Type II statement. If an attempt to print a nonexistent file is made, the message "Error - PRT File Not Found or Not Closed" is displayed, and the alarm is sounded.

NOTE

This command will output only information contained in the named file to the printer, and not any header information with respect to file name, date, time, etc. It is thus the responsibility of the programmer to insert any such needed information into his file with MSG or FWR Type II commands.

File access status parameter L1052 may be used with the (PRT) command to provide a way of allowing the part program to wait after a file is completely printed out before processing any more blocks. The part program would set parameter L1052 to 0 before the print file command. The part program would then wait after the (PRT) block until the file is printed by using a (LBL,10)(IFT,L1052=0,(GTO,10)) block. The system will set L1052 to non-zero after the file is completely printed. The usage of this parameter is optional. If the parameter L1052 is never created by the part program then the (PRT) will simply print the file and skip the setting of L1052 to non-zero.

Example:

```
N20(PAR,L1052=0)
    :   :   :
N130(PRT,PUMP341)
N140(LBL,10)(IFT,L1052=0,(GTO,10))
```

(CPY,filename 1,TO/FROM,filename 2)

The CPY Type II command can be used to copy any disk resident source file to another file from within a part program. The source file can be of any file type -- ASCII, N/C ASCII, or binary -- and must be closed; the destination file, if a file of the same name already exists, must be closed and must be of the same file type as the source; if no file of the same name as the destination can be found, one will be created.

NOTE

If a file of the same name as the destination and same type as the source already exists, any old information in this file is lost as the contents of the source are copied to it.

The command syntax used here allows for either file in the input string to be designated as the source. If the copy direction is specified as "TO", the first file PUMP341 is noted as the source with the second file HOUSING678 being the destination; the opposite is true if the copy direction is specified as "FROM".

As an example, suppose that a part program includes the following line:

N10(CPY,PUMP341,TO,HOUSING678)

Given this input, one the following results will occur:

1. If a file named PUMP341 does not exist, the error message "SOURCE FILE READ ERROR" is sent to the [MESSAGE] display and the alarm is sounded.
2. If a file named HOUSING678 is found, but is of a different file type than PUMP341, the error message "FILE ALREADY EXISTS" is sent to the [MESSAGE] display and the alarm is sounded.
3. If PUMP341 is found to not be properly closed, the error message "SOURCE FILE READ ERROR" is sent to the [MESSAGE] display if HOUSING678 is found to not be properly closed, the error message "DESTINATION FILE WRITE ERROR" is sent to the [MESSAGE] display -- the alarm is sounded in either case.
4. If a file named HOUSING678 does not exist, one of the same file type as PUMP341 is created and the contents of PUMP341 are copied to it.
5. If a file named HOUSING678 is found, and is of the same file type as PUMP341, PUMP341 is copied to HOUSING678 and the original contents of HOUSING678 are lost.

Were the direction specified as "FROM" in the above example (CPY,PUMP341,FROM,HOUSING678) the same rules would have applied with the exception that HOUSING678 would have been noted as the source and PUMP341 the destination.

Example:
N210(CPY,PUMP341,FROM,HOUSING678)

File access status parameter L1053 may be used with the (CPY) command to provide a way of allowing the part program to wait until after a file is copied before processing any more blocks. The part program would set parameter L1053 to 0 before the copy file command. The part program would then wait after the (CPY) block until the file is copied by using an (LBL,10)(IFT,L1053=0,(GTO,10)) block. The system will set L1053 to non-zero after the file is copied. The usage of this parameter is optional. If the parameter L1053 is never created by the part program then the (CPY) will simply copy the file and not set L1053.

Example:

```
N30(PAR,L1053=0)
  :   :   :
N140(CPY,PUMP341,TO,HOUSING678)
N150(LBL,10)(IFT,L1053=0,(GTO,10))
```

(RCP,local file, TO / FROM, fms_node, remote_file/type)

The RCP Type II command may be used from within a part program to upload (specify "TO") or download (specify "FROM") any disk resident file to or from any remote node configured into an FMS cell. The remote node may be the FMS Host computer or another KT-MCS tied to it. The file names at each end must be specified, as well as type of file to be transferred.

Files uploaded or downloaded to or from other Machine Control Centers or the remote host can be of any file type -- ASCII, N/C ASCII, binary, or object.

NOTE

The file type specified in the Type II command line must match that of the file itself to insure that the remote host can manage the files' disk records properly; note an ASCII file by "/AS", a binary file by "/BI", an object file by "/OB", and NC files by "/NC".

Regardless of the designated receiver of a remote file transfer, all local files sent to a remote station must first be closed. If a file is downloaded to a previously existing or locked file of the same name and file type, any old information in this file is lost as the contents of the remote source file are copied to it.

Files local to each Machine Control Center are each assumed to be located in the control directory on the resident disk. However, the command line entry for "remote file name" allows the operator to specify not only a remote FMS Host computer file name, but also a subdirectory.

Example:
DAT:[170,1]PUMP341.123;56

Names of files local to Machine Control Centers can contain a maximum of 32 alphanumeric characters, but can contain no commas, leading or trailing spaces. This is also true for files at the remote FMS computer, with the exception that commas may be included.

When specifying the remote node name, it is necessary to use the station names for each machine. As an example, "KT01" denotes machine number one, "KT12" denotes machine number twelve; use "HOST" to indicate the remote Host computer.

As an example, suppose that a part program includes the following line:

N10(RCP,PUMP341,TO,HOST,HOUSING678.DAT/AS)

Given this input, the local ASCII file PUMP341 will be sent up to the remote FMS Host computer and stored in a file named HOUSING678.DAT. The original contents of a file are not affected by this command.

Again, suppose a part program includes the following:

N10 (RCP,PUMP341,FROM,HOST,FMS.CMD/NC)

Given this input, a file at the remote FMS Host computer named FMS.CMD will be transferred to a local NC file named PUMP341.

This command actually initiates a Remote File Transfer Task (RFTT) commonly used in present FMS systems. Upon being initiated, the RFTT within the PIC of the resident control will start up a remote file transfer which, at some later time, will be fielded by the remote RFTT. Upon completion of the file transfer, any error noted will be indicated by the error message "ERROR - REMOTE FILE COPY ERROR nnnn", where "nnnn" denotes an error number.

The Remote File Transfer Task is a relatively low priority task in the host computer, so completion may thus take up to several minutes, depending on the size of the file to be transferred and system loading. The RCP command processor will attempt the file transfer three times, waiting one minute between attempts, if an error is encountered, and that completion with error status will thus take even longer.

Use of this command from within a control not part of an FMS cell will yield the error message "TYPE II COMMAND SYNTAX ERROR".

File access status parameter L1054 may be used with the (RCP) command to provide a way of allowing the part program to wait until after a file is remotely copied before processing any more blocks. The part program would set parameter L1054 to 0 before the remote copy file command. The part program would then wait after the (RCP) block until the file is copied by using an (LBL,10)(IFT,L1054=0,(GTO,10)) block. The system will set L1054 to non-zero after the file is copied. The usage of this parameter is optional. If the parameter L1054 is never created by the part program then the (RCP) will simply copy the file and skip the setting of L1054 to non-zero.

Example:
N40(PAR,L1054=0)
: : :
N340(RCP,PUMP341,FROM,HOST,FMS.CMD/NC)
N350(LBL,10)(IFT,L1054=0,(GTO,10))

OPERATOR SPECIFIED I/O ERROR HANDLING

The programmer has, via Type II data commands, the ability to choose the action taken by the control should a device I/O error be encountered during the execution of one of the aforementioned Type II file/printer commands. Here, the programmer may specify that the machine be halted until the operator acts to correct the problem, or simply that a warning message be displayed and program execution continued on.

Machine control I/O devices are divided into three groups to allow for greater program flexibility: Mass Storage (disk) devices, Sequential I/O (printer/tape) devices, and Remote (FMS only) devices. When entering either of the two commands discussed below, denote mass storage devices with the mnemonic MAS, sequential devices with SEQ, and remote devices with the mnemonic REM. Note that specifying REM in a non-FMS system will not result in a command syntax error, but will produce no noticeable results.

NOTE

The default condition of the control is to stop program execution, display an error message, and sound the alarm upon detection of any type of file or device I/O error. The (CER) command can be used to turn off the block stop and alarm on a per-device,type basis -- error message continues to be displayed -- while the (SER) command restores to the default state.

(CER,specifier,specifier,specifier)

This Type II data statement is used to command the control to continue machine operation even if certain specified file or device I/O errors are encountered during program execution. Upon the detection of such an error, a warning message is displayed, there is NO ALARM while program execution is continued. This feature was incorporated into the control to allow for cases where the operator may not be present for the entire period the machine is in operation, and where is is mandatory that part production not be halted. The default condition of the control is to stop program execution on any type of file or device I/O error.

Example: N250(CER,MAS)
: : :
N370(CER,MAS,SEQ)
: : :
N490(CER,MAS,SEQ,REM)

To specify that machine operation be continued if mass storage (only) error is noted, enter the command (CER,MAS). Likewise, to specify that machine operation not be halted on sequential I/O error, specify (CER,SEQ); specify (CER,MAS,SEQ,REM) should program interruption never be desired when any type of I/O error is encountered. Note that any combination of two device type mnemonics is also valid.

(SER,specifier,specifier,specifier)

This Type II data statement is used to command the control to halt machine operation when specified file or device I/O errors are encountered during program execution. Should a control generated program stop be desired when a mass storage error occurs, the command (SER,MAS) should be entered. Likewise, a (SER,SEQ) would specify program stop on sequential I/O error, and (SER,MAS,SEQ,REM) would cause program stop on any type of error. Any combination of two device type mnemonics is also valid.

Example:
N500(SER,MAS)
: : :
N570(SER,MAS,SEQ)
: : :
N690(SER,MAS,SEQ,REM)

When an I/O error occurs at a device marked for program stop, the the machine is block stopped, a warning message displayed, and the alarm is sounded.

NOTE

NO FURTHER MACHINE OPERATION WILL OCCUR UNTIL THE OPERATOR TAKES ACTION TO CORRECT THE PROBLEM AND RESTART THE CONTROL.

CHAPTER 8

PARAMETER AND SUBROUTINE PROGRAMMING

INTRODUCTION

NOTE

Contouring is not recommended, when transferring control to other parts of the program. GOTOs and subroutine calls will normally delay program execution temporarily, while the control is locating the new entry point in the part program and a dwell mark may be made on the part. Perform calculations and GOTOs while the tool is away from the part when possible.

Blocks containing recognized Type II data statements can have only the following additional data in them:

1. N numbers
2. Labels

Example: N100(LBL,20)(PAR,...)

The exception to the above is for the cases of the Type II commands (IFT,), (CLS,), and (END). If an 'N' number or label is to be included, it must appear before the above Type II statements within the part program block otherwise an error will result.

The maximum number of subroutines or cycles that can be defined in any one program is 20. Memory space is used to keep these different names so a practical limit of 20 was chosen. This does not limit the number of *times* a particular subroutine is called, just the number of definitions in a program.

NUMBER RANGES and -0 PROCESSING

Constants range from -99999999.99999999 to 99999999.99999999. The constant -0 is processed as 0 and the sign is not retained by the system. If commands such as A-0, B-0 or C-0 are included in a call to a cycle, they will be converted to A-360., B-360., and C-360. Internal computational limits have been expanded to allow for operations like (99999999.+10.)/2.

Programmed and assignment ranges -99999999. thru 99999999.

For error processing, see the section "Parameter Calculation Errors" below.

TYPE II DATA

Spaces will be allowed between fields of data in Type II statements. The closing parenthesis is required. Previously defined Type II statements (before the subroutine feature) may not allow this however.

Recommended:
N10(DLP,L10,L12,L13,L(L15),L(L18))
Permitted:
N10(DLP, L10, L12 ,L13   ,L(L15))

NOTE

Spaces between fields are allowed but not recommended.

EXPRESSION CALCULATION LIMITS and ERRORS

There is a limit of 8 groups of *OPEN* parentheses allowed in arithmetic expressions. When the 9th level is entered, the message "Error - exceeds 8 open parentheses" is issued.

The maximum or minimum number that can be assigned to a parameter is +-99999999.99999999

When an assignment is made with a number that is outside that range, the message: "error - exceeds + or - 99999999.9999999" is issued for that block. The control will be Block Stopped and the block in error will not be allowed to go active.

These kinds of errors occur, for example, when division by zero is attempted, or when computing the tangent of 90 degrees.

PARAMETERS

There are three types of parameters: local, program, and machine related parameters.

Additionally, there are four levels that may exist in a part program:

1. Level 0 is the main part program
2. Level 1 is a subroutine or cycle called from Level 0
3. Level 2 is a subroutine or cycle called from Level 1
4. Level 3 is a subroutine or cycle called from Level 2

LOCAL PARAMETERS

Local parameters are a private set of parameters. The program has a set and each active subroutine has its own set of local parameters.

When a subroutine is called, a new set is enabled for the called subroutine. Thus, one subroutine that uses L1, can call another subroutine that also uses L1 in a different manner. Each L1 is private to the called subroutine. This allows subroutines to be independent of each other (one subroutine cannot inadvertently "pollute" the parameters of another). The new set of local parameters remain in existence until the subroutine or program terminates.

Example:
L3 = .157 in subroutine K5
L3 = 3.147 in subroutine K8

Local parameters may be accessed and modified by the Parameter Display and the subroutine or program owning them. These parameters can be assigned in the program or subroutine or by a subroutine call (CLS). Below is an illustration of local parameters.

Parameter	Value	Description
0	1.0000000	repeat count of sub.
1	4.5000000	X Axis Probe Limit
2	1.5000000	X Axis Probe Reading
7	3.1190000	Circle 1 Max. Dia
8	0.0	* enter insp limit *
99	6.0000000	Hole depth

Fig. 8.1 LOCAL PARAMETERS

Local parameters are identified by labels L0 through L99. L0 is a special parameter and is used to show the run count of a subroutine. The system decrements this count each time an END statement is executed in the subroutine. If L0 goes 0 or negative, the subroutine ends and returns to the caller or that level of the program which requested the subroutine to be implemented.

The parameter table will display the local parameters of the executing program or subroutine ONLY.

[CLEAR ALL LOGIC] will delete *all local parameters associated with active subroutines*. The set belonging to the MAIN program will not be deleted.

M02,M30, and [ASSIGN PROGRAM] will delete *all local and program parameters*.

PROGRAM PARAMETERS

Program parameters are used for program communications. Both the program and all the subroutines have access to these parameters. They are used to pass back parameters from subroutines and they ARE NOT PRIVATE to either the program or a subroutine. Within the program parameters L800-L899 are reserved for Defined Cycle use.

Program parameters differ from machine related parameters, because program parameters are deleted when an M02 or an M30 is executed or when a program is assigned.

100	14.000000	parts completed program
.	.	.
805	12.888800	E value - fixed cycle reserved for defined cycles
899	10.000000	last of fixed cycles
999	1.0000000	last of prog parameters

Fig. 8.2 PROGRAM PARAMETERS

Program parameters may be accessed and modified by the parameter display, program AND subroutines.

Program parameters are identified by the labels L100 through L999.

DEFINED CYCLE PARAMETERS

These parameters are reserved for defined cycle use ONLY. This keeps defined cycle parameters separate from subroutine parameters.

The range of defined cycle parameters is L800 thru L899. Defined cycle parameters (L800...L899) are deleted when G80 or G800 go active, upon clear all logic, and at the end of program (M2 or M30).

MACHINE RELATED PARAMETERS

Machine related parameters can be accessed by both the program and its subroutines but are not deleted when a program ends. Machine related parameters may be used for parts counts, PMI, and other program independent data.

Machine related parameters remain in existence until the system is reloaded, or until they are deleted by the parameter display, PMI, or the part program. These parameters may be accessed and modified by the program, PMI, a subroutine, or the parameter display.

```
1000   1.0000000  X axis post position
  :      :   :
1999   2.0000000
```

Machine related parameters are identified by labels L1000 thru L2999. Within this range L2000 thru L2999 are reserved for PMI use. L1000 thru L1022 are reserved for probe cycles. Customers with FMS systems should be aware that parameters L1200 thru 1299 are used for FMS applications and parameter definitions may vary from system to system. Customers should consult the PMI documentation for their systems for the definition of these parameters. Customers with stand alone machines have access to L1200 thru 1299. See Appendixes B and C. Additional reserved assignments may be made in the future.

PMI PARAMETERS

PMI parameters interface the programmable control to the program. They are assigned a range of the machine related parameters (L2000-2999) and are reserved for PMI/program/operator interaction.

```
2000   1.0000000  availability in hours
  :      :   :
2999   0.0        1 = hrs of operation
```

SYSTEM PARAMETERS

System parameters are used to access various system variables such as the active tool length and diameter compensations and other similar information.

These parameters are *read only*, that is, values *cannot* be assigned to them. In this way, system information is protected but is still accessible to the program.

System parameters are referenced using labels L3000 through L3999. System parameters are not shown on the parameter display.

```
3000   First axis absolute position (normally X)
  :      :   :
3999
```

The above are examples of system parameters and are not displayed. Their definitions are listed under System Parameter Definitions in Appendix C.

Additional assignments will be made in the future as necessary.

The adding, modification, and deletion of parameters and parameter data through the parameter display can be enabled or disabled through the use of the function enable/disable feature.

PARAMETERS ASSIGNMENT

Parameter values are assigned with the following Type II data statement. A *maximum* of 12 parameters can be assigned in any one (PAR,) statement.

Format:

(PAR,L<parameter no> = <expression> <"description">,L...)

↑ optional field

Parameter numbers : 0 thru 2999 or a (PAR) expression that rounds to 0..2999

An expression may consist of : <term> <term> <operator> <term>

The operator is a mathematical operation: +, -, *, /

The term is a : (,), constant, letter address value, function

A constant is := -99999999. < 0 < 99999999.

A letter address value is : A, B, C, E, F, I, J, K, L, P, R, S, T, U, V, W, X, Y, Z. Values used are commanded programmed positions (as displayed on main display..includes offsets).These values are valid only when the machine is configured to accept them.

The mathematical functions used are : SQ, SQRT, ABS, SIN, COS, TAN, ASN, ACS, ATN, INT, FRA, LOG, ALOG, LN, EXP

Description is an English text that may consist of from 1 to 20 displayable characters. The description must be enclosed within quotation marks and must be contained within the (PAR) expression. The use of double quotation marks "" causes the description field to be cleared.

Operator precedence (see Appendix F) :

1. positive and negative
2. multiplication and division
3. addition and subtraction

e.g.: 6+2*5 = 16

DELETING PARAMETERS

Parameters can be deleted with the following Type II data statement:

(DLP,L15,L100,L129,L(L123),L(L233))

This may become necessary when the 100 parameter limit is reached and previously defined parameters are no longer necessary.

TYPE I PARAMETER SUBSTITUTION

Any parameters (L0...L3999) may be referenced from Type 1 data by suffixing the letter address with a parameter identifier. All letter addresses can be used for substitution except for the letter L. Example:

```
        This is the current position of X axis ↓
N50(PAR,L10=10.5,L11=55.,L12=11.,L13=X)
N100 XL10 Y-L11 ZL12
                  ↑
         The value used for Z is in parameter L12.
```

NOTE: G, H, M, P, S, and T values are rounded to the nearest integer.

Letter addresses X, Y, and Z are assigned the values 10.500000, -55.000000, and 11.000000, respectively. The parameter values are not converted to inch or metric units but are used directly, without concern for what units the letter addresses are in.

G78 - IGNORE SUBSTITUTIONS

Type I letter commands that contain a substitution with a parameter that is undefined, are ignored in the block containing the G78. This is a non-modal G code that MUST PRECEDE all substitutions in the block.

This G code is mainly for subroutines and fixed cycles that need to execute conditionally "passed in" information without having to test to see if the parameter was actually passed into the control.

An example would be for a fixed cycle to move the other commanded axes after retracting the Z axis. No tests would have to be made for axes A, B, C, X or Y if the motion block contained a G78 as follows:

N30G78X5.19Y2.235ZL89F10.

Motion in the Z axis will occur ONLY if parameter L89 is present in the parameter table.

```
N100(PAR,L1 = 2.54"Centimeters per Inch")        ! simple example

N120(PAR,L10 = SQRT(L1*L2) + (X-L2) + 4.,L2 = L3/L4*L5*(X + L2))  !   complex example

N150(PAR,L2 = L2 + 1.,L3 = L2)      ! L3 gets L2 + 1. processed left to right

N170(PAR,L1 = 12.3081,L2 = INT(L1))       ! get integer part(12.) in L2

N210(PAR,L10 = 10.5,L11 = 55.,L12 = 11.,L13 = X,L6 = 7.)
                                            ↑ gets current X value as displayed on [MAIN] display (with
                                              offsets)
N230(PAR,L6 = 8.)
N240(PAR,L(L6 + 6.) = 10.5,L11 = 55.,L12 = 11.,L13 = X)
               ↑
               An expression defines the parameter no. If L6 contains 8., then this parameter becomes L14
               and is assigned 10.5. This is an example of creating a subscript parameter. The control
               automatically assigns the number by reading the expression L6 and carries out the function
               to create the parameter number.
```

Fig. 8.3

TYPE I DATA EXPRESSION SUBSTITUTION

Type I data may reference parametric expressions and mathematical functions directly through the use of a special form of Type I data. The placing of a comma after the Type I data letter address causes the control to interpret data after the comma as an expression. The value of the expression is assigned to the letter address specified. Examples:

N100 X,10.*SIN(45.),Y15.
Would result in X commanded to a position of 7.0711 and Y commanded to 15.

N200 X,Y,Z20.
Would result in X moving to a position currently held by Y axis and Z commanded to 20.

N300 X,X+.1
Would result in X commanded to the current position of the X axis plus .1.

N400 X,15.2*(COS(L50))
Would result in X commanded to 15.2 times the trigonometric cosine of parameter L50.

NOTE

COMMAS MUST BE USED TO PROPERLY FORMAT THESE STATEMENTS.

FXC - Fixture Offset Substitution

Fixture offset substitution format:

(FXC,HL####XL####YL####)
where #### is a decimal whole number ranging from 0...3999. Example:

(FXC,HL99XL10YL25)

In the above example, H will be assigned the value of L99, the X offset will have the value of L10, and the Y offset will have the value of L25.

The FXC parameter substitution allows the NC part programmer to substitute an H number or a fixture offset from the parameter table, within the FXC Type II statement. Each entry in the FXC statement may have an 'L' number or a value associated with it.

Error messages:

1. Input Source Error
2. Error - L number NOT in table
3. Error - L numbers range from 0..3999
4. Error - FXC 'H' codes range from 1..32
5. Error - axis fixture offset out of range

Rules pertaining to FXC:

1. H codes from the parameter table will be rounded to the nearest whole number (i.e. 2.4=2.; 2.5=3.).
2. H codes must range from 1 to 32; axis offsets must be valid for the designated axis.

TLD - Tool Table Load Substitution

Tool table load substitution format:

(TLD,TL####PL####DL####)
Where #### is a decimal whole number ranging from 0...3999

Example:
(TLD,TL111PL99DL10)

In the above example, T number of L111 in the tool table will have the value of L99 assigned to its tool length comp and L10 as its tool diameter comp.

The TLD parameter substitution allows the NC part programmer to substitute a tool number and / or tool length and / or diameter compensation from the parameter table within the TLD Type II statement.

Each entry in the TLD statement may have an 'L' number or an actual value associated with the tool number, length comp and diameter comp entry.

Error messages:

1. Input Source Error
2. Error - L number NOT in table
3. Error - L numbers range from 0..3999

Any attempt at parameter substitution with other entries will result in an Input Source Error.

SYSTEM PROCESSING OF PARAMETERS

Parameter values will be computed as "look ahead" and their values will be retained in an internal "look ahead" table as well as on the ready string. The values assigned to the parameters will be inserted into the main table (the one that is displayed to the operator) when the block containing the assignment goes active. This will allow cutter diameter compensation to function in the "look ahead" manner also.

SUBROUTINES

Subroutines may call other subroutines to a nesting depth of THREE. (See Figure 8.4)

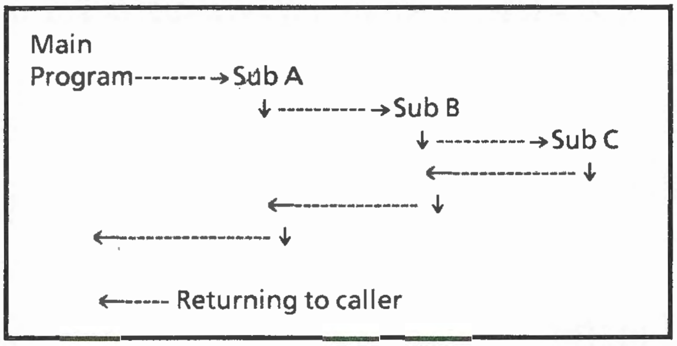


Fig. 8.4 SUBROUTINE NESTING LEVELS

The main program calls subroutine A, which calls subroutine B,which calls subroutine C. Subroutine C cannot call another subroutine.

Subroutine restrictions:

Subroutines/defined cycles which are locally defined in a program may not exceed a total of 20.

1. Subroutines can be recursive (call themselves).
2. Subroutines can't be *defined* in other subroutines.
3. Subroutine (END) statements cannot be deleted or modified from the ready string.
4. Subroutine names CANNOT be fixed cycle names (i.e. G80..G89 or G800..G899 or G24..G27 or G31).

DEFINING A SUBROUTINE

A subroutine is defined with the following Type II data statements:

N50(DFS,PUMPHOUSING134,SAVE) !Start sub

: : : :

The subroutine body would follow here. The SAVE part of the command saves the sub on the disk so other part programs can use it (optional).

: : : :

N200(END) ! End of subroutine

NOTE

Part program blocks containing (DFS,...) commands are not allowed to be cleared by the [CLEAR READY] function or through use of the MDI feature. Modification of a block containing a (DFS,...) command is also not allowed when using the MDI feature.

LOCAL SUBROUTINES

Local subroutines are always defined in the currently running part program. Local subroutines are retained by the control until an M02 or M30 is encountered or a new program is assigned. A maximum of 20 local subroutines can be defined in a program.

SAVED SUBROUTINES

If the subroutine is defined with the "SAVE", option, it is stored as a file on the disk using the subroutine's name for the file name. If the subroutine file already exists on disk the existing (old) file is deleted and the file being saved replaces it on the disk.

"Saved" subroutines can be defined within the part program or created using the file editor. The files associated with the "saved" subroutine are not deleted when the NC program terminates so they are available to other part programs. This feature allows a library of standard subroutines to be made that are available to all part programs.

NOTE

It is HIGHLY RECOMMENDED that a standard prefix be included in the name of these routines at each site. This will alert all users that these files are saved subroutines and are not to be routinely modified or deleted.

For example: $DRILL or $FACE or SUB-DRILL or SUB-FACE

SUBROUTINE CALL

Before a subroutine can be called, it must be defined by the DFS Type II data statement. A call to a subroutine has the following form:

(CLS,PUMP34,5,L5=12.7,L6=3.123,L7=L5*L6)

A *maximum* of 12 parameters can be assigned in the CLS statement (including the run count as 1 of the 12). The error message "Error - exceeds 12 L's" will be displayed ahead of the block in which it occurs.

The run count is displayed in the parameter table as L0 and is decremented when the subroutine executes the END statement. When L0 becomes less than or equal to zero, when the END statement goes active, the return to the caller is made and the subroutine terminates. When no run count is specified in the call, L0 is set to 1 automatically.

N100(CLS,BOLT,2,L5=7.) . . . a run count of 2 passed to the subroutine as an integer. The run count *cannot* be passed as an expression. It can , however, be passed as L0 like any other parameter.

N100(CLS,BOLT,L5=7.,L0=L3+2.) . . . a run count passed as an expression thru L0.

The parameter labels in the call correspond to the identical parameters used within the subroutine. Parameters L0 through L99 are reserved exclusively for subroutine use. Parameters on the right side of the assignment statement are current level parameters, while parameters on the left side are for the called subroutine level.

(CLS,BOLT,17,L41=.3,L34=43.,etc.... <parameter assignments>)

In the example in Figure 8.5 below, the call to subroutine "DRILLIT" will cause the subroutine to be executed four times before returning to the caller.

Parameter L1 within the subroutine has the value 5.0. Parameter L3 has the value the trigonometric sine of the program parameter L101 times 5.0.

Parameter L4 holds the value of parameter L1000 on the caller's level. Parameter L5 has the value 5.689

SUBROUTINE EXECUTION

Subroutines execute until the (END) statement is executed resulting in L0 (run count) becoming < or = to 0. Execution then continues with the block *after* the block that called the subroutine.

SUBROUTINE TERMINATION

When an operator wishes to exit an executing subroutine before the normal END occurs, he can MDI and execute the (END) statement. If 3 subroutines are active, for example, then three (END) statements are needed to return to the initial call in the main program assuming the run count of 1 in each level, otherwise the operator must set L0=0 then MDI the (END) statement for each level.

NOTE

[CLEAR ALL LOGIC] is not changed and will still set the control to begin at the first block of the main program.

MODIFYING SUBROUTINES

A subroutine (END) statement cannot be modified/deleted when it is on the ready string.

NOTE

Saved subroutines should be edited in the program that defines them, then the program which defines them should be run again.

SUBROUTINE STORAGE

If the subroutine is defined "save", it is stored as a single file otherwise it is NOT saved on the disk.

If a subroutine is called which is not defined in the program, a search is performed on disk for a saved subroutine. If it is found, the saved subroutine will be executed . If it is not found, a cycle stop is forced and an error message "Error - Subroutine or Defined Cycle Not Found" is presented.

MAIN DISPLAY

The [MAIN] display will indicate the level of the current subroutine call by freezing the calling block on the main display. This will allow all three subroutine levels to be displayed without changing the operator's current concepts of the [MAIN] display.

MDI modification of DFS subroutine blocks is not allowed. [CLEAR READY] is not allowed if the block on the top of the ready list of blocks contains a DFS subroutine command.

```
                PROGRAM  PARAMETERS PASSED TO SUBROUTINE
                  ↓       ↓              ↓          ↓
N100 (CLS,DRILLIT,4,L1 = 5.,L3 = SIN(L101*5.),L4 = L1000,L5 = 5.689)
                 ↑
                SUBROUTINE PARAMETER
```

Fig. 8.5 SUBROUTINE CALL EXAMPLE

SUBROUTINE CALL

Figure 8.6 is an illustration of a running program displayed on the CRT containing four active levels of operation. Figure 8.7 contains various CLS Type II data statements.

BLOCK/STRING SEARCH

N/C block/string searches are limited to the bounds of the active (executing) program level (main program or subroutine) and do not cross from one to another. The search function is used to read N/C part programs for a particular block number or a string of characters.

With this search function: upon successful completion of the search, the operator has the option of either executing the part program at the start of the block containing the search target or resuming the search for the next or previous occurrence of the target string.

DEFINED CYCLES

Defined cycles are similar to subroutines in that they can be programmed locally and also saved. The standard set is called "fixed cycles" (G81 thru G89), and are provided by the control. User programmed defined cycles are G801 thru G899 and follow the same operational rules as the G80 series.

When a defined cycle is activated, the cycle is AUTOMATICALLY run after each block that contains either an X, Y, Z, A, B, or C axis command.

A defined cycle will NOT be performed for blocks that do NOT contain an X, Y, Z, A, B, or C axis command.

Both G80 and G800 can be used to cancel the cycle. G77 can also be used to omit a cycle for 1 block, the one in which it appears.

G800 series cycles are defined in a similar manner as subroutines but must have a name which begins with the character "G" and be suffixed with a number from 801 to 899.

There are a set of parameters dedicated to exclusively to defined cycles. They are the global program parameters L800 thru L899. The letter addresses A thru Z are used to pass parameters to a defined cycle. They are transferred into the defined cycle parameters L801 thru L826 respectively (alphabetical order) when the cycle is activated. The cycle itself, references these

```
   X   20.9320      B  180.000     F  400.0  (380.0)     T 8607 (838)       H 32

   Y    8.8500      C   79.000     S 3000  (2800)  <359>       Length Comp.  -1.9790

   Z    7.2500      HP 12.0  (9.9)                             Diameter Comp.  0.9320

G: 00 17 29 37 40 48 50 70 80 90 94 99
M: 01 05 19 48

Program: ENGINE BLOCK NUMBER 85673                     Pallet: 099

N309(CLS,CYL BORES1,5,L5 = 3.)    ←an active subroutine (level 1)
N20(CLS,EXHAUST PORT2, 1,L2 = 4.)    ←an active subroutine (level 2)
N100(CLS,OIL WAY3 1,5,L1 = 3.3)   ←an active subroutine (level 3)
N10X10.YL5Z10.
N20G91ZL1F100.   ←the current active block (highlighted) in subroutine OIL WAY3 1

N30ZL3F0      ←Additional Program  Blocks
N40
N50
N60
N70
N80
N90
```

Fig. 8.6 MAIN SCREEN EXAMPLE OF SUBROUTINE CALL

Maximum number of parameters passed to a 32 character named subroutine: 4

```
N1000(CLS,OPERATION01234567890123456789012,99,L1 = L20,...,L4 = L50)
```

↑ 32 character name — ↑ Max of four parameter statements

Maximum number of parameters passed to a 10 character named subroutine: 8

```
N1000(CLS,OPERATION1,99,L1 = L20,L2 = L30,L3 = L40,...,L7 = L58,L8 = L56)
```

↑ 10 character name — ↑ Max of eight parameters

Maximum number of parameters passed to a 3 character named subroutine: 9

```
N1000(CLS,OP1,99,L1 = L20,L2 = L30,L3 = L40...,L9 = L57)
```

↑ 3 character name — ↑ Max of nine parameters

Maximum number of parameters passed to a 1 character named subroutine: 11

```
(CLS,A,99,L1 = 1.,L2 = 2.,L3 = 3.,L4 = 4.,..L8 = 8.,L9 = 9.,L10 = 0,L11 = 1.)
```

↑ 1 character name — ↑ Max of eleven parameters

Fig. 8.7 MORE EXAMPLES OF SUBROUTINE CALLS

parameter values with their L800 designations. (The letter L cannot be used to pass a parameter into a defined cycle. See the example and table below showing the letter addresses and their parameter values.)

1. When the defined cycle is canceled (G80 or G800), L800...L899 are deleted.
2. Subroutines CAN be called while a defined cycle is pending.
3. A defined cycle CAN be called by a subroutine.
4. A running defined cycle CANNOT call a subroutine.
5. A defined cycle CANNOT call another defined cycle.

```
N50G810X12.Y10.R6.6Z5.F60.S300M03
```

- F60. — passed to cycle as L806
- Z5. — passed to cycle as L826
- R6.6 — passed as L818
- G810 — Activates defined cycle on this block

A	B	C	D	E	F	G	H	I	J	K	M	
801	802	803	804	805	806	807	808	809	810	811	813	
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
814	815	816	817	818	819	820	821	822	823	824	825	826

Fig. 8.8 EXAMPLE OF A USER-DEFINED CYCLE

DEFINING A CYCLE

A cycle is defined with the following Type II data statements:

```
N50(DFC,G883,SAVE)     ! start of defined cycle
  :   :   :   :
```

The cycle body would follow here. The SAVE part of the command saves the cycle on the disk so other part programs can use it (optional).

```
  :   :   :   :
N200(END)              ! end of defined cycle
```

LOCALLY DEFINED CYCLES

Defined cycles are local when they are defined in the currently running part program and are not tagged with the "SAVE" option. These cycles are present in the control after they are executed and are removed when another program is assigned. The control will allow a maximum of 20 locally defined program cycles.

SAVING DEFINED CYCLES

If the cycle is defined with the "SAVE", option, it is stored as a disk file with the cycle name as the file name. Saved cycles can be defined within the part program or created using the file editor. The files associated with the saved cycle are not deleted when the part program terminates, so they are available to other programs. This feature allows a library of standard cycles to be made that are available to all part programs.

If a cycle is called which is not defined in the program, a search is performed on disk for a file with that name. If found, the file will be opened and executed. If it is not found, a cycle stop is forced and an error message presented.

CYCLE INHIBIT - G77

G77 is a non-modal code that inhibits the execution of any defined cycle in blocks that contain G77. The control will execute X,Y,Z movement commands, but will not perform G80 and G800 series cycle commands when the G77 command is active. This would allow the programmer to jump over clamps and fixtures without performing major modifications to the program or the defined cycle.

Example:

```
N200G81X2.Y3.Z-.350R.1F201500M3M8
N210G77X3.Y4Z2.      !Move to jump a clamp
N220X4.Y5.           !Resume drilling cycle
```

DIFFERENCES BETWEEN G80 and G800 CYCLES

In the G80 series of cycles, ONLY the letter commands A,B,C,E,F,R,X,Y,Z are passed to the cycle and the other letter commands are executed by the control BEFORE the first block of the G80 cycle is begun. The G80 cycle is responsible for the output of the passed in commands. Multiple G and M codes are allowed. This allows backward compatibility with previous systems.

For the G800 series cycles, all letter commands are passed in to the cycle (L801..L826) and NONE are executed before the cycle begins. Only ONE M or G code is allowed in this type of cycle and they are passed into L807 and L813. If more than one M or G code is programmed, *only the last one is acted upon.*

The format of passed parameters for defined cycles is as follows:

For letter addresses A, B, and C, the number is passed in X.3 format. Also note that a command of -0 for these letter address is passed as a -360.00000 because the value of -0 cannot be used as a parameter value.

Examples:

B180. passed in as parameter L802 = 180.00000
C344789 passed in as parameter L803 = 344.78900
A34 passed in as parameter L801 = .03400000
B-0 passed in as parameter L802 = -360.00000

For letter address F the number is passed in X.1 format when in inch mode and X.0 format when in metric mode

Examples:

G70 Inch mode
F222 passed in as parameter L806 = 22.200000
F10.5 passed in as parameter L806 = 10.500000

G71 Metric mode
F567 passed in as parameter L806 = 567.00000
F156. passed in as parameter L806 = 156.00000

For letter addresses D, O, and N the number is passed in X.4 format

Examples:

D123456 passed in as parameter L804 = 12.345600
O1.4567 passed in as parameter L815 = 1.4567000
N9876543 passed as parameter L814 = 987.65430

For letter addresses G, H, M, P, S, and T the number is passed in X.0 format

Examples:

G18 passed in as parameter L807 = 18.000000
H19 passed in as parameter L808 = 19.000000
M4 passed in as parameter L813 = 4.0000000
P3 passed in as parameter L816 = 3.0000000
S550 passed in as parameter L819 = 550.00000
S4000 passed in as parameter L819 = 4000.0000
T1234567 passed as parameter L820 = 1234567.0

For letter addresses E, I, J, K, Q, R, U, V, W, X, Y, and Z the number is passed in X.4 format when in inch and X.3 format when in metric mode. Examples:

G70 Inch mode
I333 passed in as parameter L809 = .03330000
X10.5 passed in as parameter L824 = 10.500000
R120000 passed as parameter L818 = 12.000000

G71 Metric mode
Z567 passed in as parameter L826 = .56700000
E156. passed in as parameter L805 = 156.00000
R155000 passed as parameter L818 = 155.00000

COMMENT STATEMENT

Because of the IFT..GTO and GTO statements (the capability of "jumping around" in the part program) the need to comment the part program is essential. A comment can be included in any block.

<Type I or Type II statement> ! comment text ... EOB

The comment sign ! indicates that characters between this symbol and the EOB are to be interpreted as comments only and are NOT to be acted upon. Therefore X, Y, Z and information to the right of the comment sign will not be acted upon by the control.

GO TO STATEMENT (GTO)

The GTO command is the unconditional transfer of control to the block containing the referenced label.

The search for the label occurs in the current level ONLY. That is, GTO searches in the main program are for labels in the main program. GTO searches in a subroutine, is conducted in that subroutine ONLY.

GTO FORMAT

(GTO,nnnn)
↑
Label number of label contained in block to be executed next.

Examples:

```
N25(GTO,1000)                 !A forward reference
N155(LBL,1000)X14.5678Y20.    !Label block
```

```
X   20.9320    B   180.000    F  400.0  (380.0)    T 8607 (838)    H 32

Y    8.8500    C    79.000    S 3000  (2800)  <359>    Length Comp.   -1.9790

Z    7.2500    HP 12.0  (9.9)                          Diameter Comp.  0.9320

G: 00 17 29 37 40 48 50 70 80 90 94 99
M: 01 05 19 48

Program: ENGINE BLOCK NUMBER 85673                 Pallet: 999
                                   Cycle: G810 E 10.000 R 12.0000 Z 20.0000
N300 X10.YL5Z10.                    ↑
N310 G810X14.Y14.E10.R12.Z20.     Present ONLY when a defined cycle has has been activated, and is
                                  highlighted when the cycle is running.
N10 X824.YL825ZL818F0
N20 ZL3F0
N30 G91ZL826F100.
N40 (END)    ! end of fixed cycle
N320 G80
N330
N340
```

Fig. 8.9 MAIN SCREEN EXAMPLE OF DEFINED CYCLE CALL

N200(GTO,1000) !A backward reference
GOTO command format: (GTO,####)
where #### is a decimal number ranging from 1...9999.

Example: (GTO,99)

The GTO Type II statement gives the NC part programmer the ability to alter the normal flow of an NC part program within the level that the part program is currently executing.

There are 4 levels defined:

Level 0 is defined as the main NC part program. Level 1 - 3 are defined as subroutine / cycle levels.

- Level 1 is a subroutine / cycle called from level 0.
- Level 2 is a subroutine / cycle called from level 1.
- Level 3 is a subroutine / cycle called from level 2.

When a GTO Type II statement is encountered the control system will search for the first program block containing the (LBL,####) defining the GTO destination label.

The search is performed in the following steps:

1. It will first search the label reference table, which contains the last 10 labels referenced.
2. If the label is not found in the label reference table the part program will be searched from the current block to the end of the current level.
3. If the label is not found here, the part program will be searched from the beginning of the current level to the current block.
4. If the label is not found an error message "Error - GTO label not found" will be displayed on the main display.
5. If the label is found the part program will continue execution at the NC block containing the GTO destination label.

RULES PERTAINING TO GTO

1. The destination label must be defined within the level that contains the GTO statement. The reason for this is that the same label may be defined in every level, but only once within a level.
2. The GTO Type II statement must be the only statement in an NC program block.
3. The destination label must be a numeric value ranging from 1...9999.
4. Within levels 1-3 the END terminator must be in the last NC block of that level.

LABEL STATEMENT (LBL)

This is the target of the IFT...GTO and GTO statements. The block containing a label that is referenced by a "GOTO" will be executed in its entirety, after the GTO, even though the block may precede or succeed the GTO statement in the program.

A label appearing inside a Type II data statement or in a comment field will NOT be referenced by any GTO statements.

A label that is MDI'd will not be effective, as only the source program is searched for the label.

Label definition command format: (LBL,####)
where #### is a decimal number ranging from 1...9999. Example:

N30(IFT,L2=0,(GTO 50)) !Skip second part if none is fixtured.
N1000(LBL,50)X10.1001Y6.5432M5 !Enter here if only one part is fixtured.

The LBL Type II statement gives the NC part programmer the ability to reference an NC program block from anywhere within the level that the label is defined. There are 4 levels defined:

Level 0 is defined as the main NC part program. Level 1 - 3 are defined as subroutine / cycle levels.

- Level 1 is a subroutine / cycle called from level 0.
- Level 2 is a subroutine / cycle called from level 1.
- Level 3 is a subroutine / cycle called from level 2.

LABEL REFERENCE TABLE

The control system maintains a table of the last 10 labels that were referenced within the currently active part program. The table is arranged such that the most recently referenced label is the first label in the table, the previously referenced label is the second label in the table, and so on. This is done to decrease the time necessary to find the NC part program block containing the label being referenced.

RULES PERTAINING TO LABELS

1. The LBL Type II statement may be included in any NC program block and may be combined with any other Type I or Type II data.
2. The label number must be a numeric value ranging from 1...9999.

3. The same label may be defined in different levels (i.e. every subroutine / cycle and the mainline program may define label 10).

IFT,<Boolean expression>,TYPE I DATA

This operates similar to the IFT..GTO with Type I data replacing the GTO. The Type I data is executed *only if* the boolean expression is TRUE. It adds conditional block execution to NC programming.

CAUTION

EVEN THOUGH PARAMETERS MAY APPEAR TO BE EQUAL ON THE PARAMETER DISPLAY, THEY MAY NOT BE. TWO MORE SIGNIFICANT DIGITS EXIST THAN APPEAR ON THE DISPLAY. FOR EXAMPLE: TO COMPARE TWO EXPRESSIONS IN THE RANGE OF THE DISPLAY, THEY WOULD HAVE TO BE PROGRAMMED AS FOLLOWS: (IFT,ABS(L1)-ABS(L2)<.00000001)

Example:
N200 (IFT,G70,X10.Y10.M4)
↑
Boolean expression (in this case - G70). If the boolean expression is TRUE, when the block goes active, the spindle is started and the axes moved to X10.Y10.

If the boolean expression is FALSE, the next block following the IF statement is executed.

An IFT statement may contain numerous AND, OR, NOT, DEF expressions up to the maximum line length of 80 characters per line.

Example:
(IFT,G70ANDG94ANDG90,X10.Y10.M4).

The following items may be used in the construction of IFT Boolean expressions:

- Active code
- NOT Boolean expression
- DEF(parameter address)
- Boolean expression AND Boolean expression
- Boolean expression OR Boolean expression
- value expression <relational operator> value expression

Active code: Gnn,Mnn — TRUE if code is *active* (i.e. currently modal)

Relational operator: =, <>, <, >, <=, >= (<> is the not equal sign)

Value expression: +, -, *, /, (,), constant, letter address value, function (*multiply) (/divide)

Constant: 99999999. > 0 > -99999999.

Letter address value: A, B, C, E, F, I, J, K, L, P, Q, R, S, T, U, V, W, X, Y, Z
Values used are commanded programmed positions (as displayed on main display, includes offsets). These values are valid only when the machine is configured to accept them.

Function: SQ, SQRT, ABS, SIN, COS, TAN, ASN, ACS, ATN, FRA, INT, LOG, ALOG, LN, EXP

Parameter address: L1 . . . Ln

Type I data: as defined in EIA spec - RS274D

Operator precedence when parentheses are not used:
1. NOT and DEF
2. AND
3. OR
4. Relationals (=,<,>,<>...)

Example:

N100(IFT,G70,X10.F200.) ! Increase the feedrate if inch mode

N120(IFT,L10<50.2,X10.F200.)
! Increase the feedrate if L10 is below 50.2

IFT,<Boolean expression>,TYPE II DATA

The Type II format has been employed here to stay within the framework of EIA spec - RS-447. The following Type II mnemonics are not allowed in IFT statements: DFS, DFC, END, LBL, and IFT.

Type II mnemonic
↓
(IFT,<boolean expression>,(GTO,<label>))
↑
If the boolean expression is TRUE when the block goes active, the GTO will be done and the block containing the referenced label number will be executed next. A block search is performed from the beginning of the program or subroutine to find the block.

If the boolean expression is FALSE, the next block following the IF statement is executed.

The following items may be used in the construction of IFT Boolean Type II Data expressions:

- Active code
- NOT Boolean expression
- DEF(parameter address)
- Boolean expression AND Boolean expression
- Boolean expression OR Boolean expression
- value expression <relational operator> value expression

Active code: Gnn,Mnn — TRUE if code is *active* (i.e. currently modal)

Relational operator: =, <>, <, >, <=, >= (<> is the not equal sign)

Value expression: +, -, *, /, (,), constant, letter address value, function (*multiply) (/divide)

Constant: 99999999. > 0 > -99999999.

Letter address value: A, B, C, E, F, I, J, K, L, P, Q, R, S, T, U, V, W, X, Y, Z

Values used are commanded programmed positions (as displayed on main display, includes offsets). These values are valid only when the machine is configured to accept them.

Function: SQ, SQRT, ABS, SIN, COS, TAN, ASN, ACS, ATN, FRA, INT, LOG, ALOG, LN, EXP

Parameter address: L1 . . . Ln

Type I data: as defined in EIA spec RS274D

Operator precedence when parentheses are not used:

1. NOT and DEF
2. AND
3. OR
4. Relationals =,<,>,<>...

IFT, . . . ,GTO, STATEMENT

This statement gives the control a "simple and clear" way to execute conditional parts of a program. This one statement can replace the REPEAT-UNTIL and DO WHILE constructs. No complicated looping rules exist when only the IF-GTO statement is implemented. More important however, no complicated error reporting scheme is needed to inform the user when he has jumped into or out of a loop.

Looping can be accomplished however, by employing the modification of a parameter with the IF-GTO statement. Before the loop is to begin, set say L100 = 1. and begin the loop. In the loop set L100 = L100 + 1. At the end of the loop, test for the final value with the IF-GTO statement. The GTO should repeat the loop until the parameter reaches the final value in the IF test. The GTO references labels in the current level only. Only the main program is searched for the label when the GTO is in the main program. Only the subroutine is label searched when the GTO is in the subroutine.

(IFT,<boolean expression>,(GTO,<label>))

↑

If the boolean expression is TRUE, when the block goes active, the GTO will be done and the block containing the referenced label number will be executed next. A block search is performed from the beginning of the program or subroutine to find the block.

If the boolean expression is FALSE, the next block, the one following the IF statement, is executed.

The following items may be used in the construction of IFT Boolean Type II Data expressions:

- Active code
- NOT Boolean expression
- DEF(parameter address)
- Boolean expression AND Boolean expression

```
N100(IFT,G70 AND G90,(GTO,33))  ! Only if inch and absolute mode

N120(IFT,L10 = SQRT(L1*L2) + (X-L2) + 4.,(PAR,L88 = 14.55))    ! Complex example

N100(IFT,DEF(L802),(GTO,33))    ! Only if L802 is defined (in Parameter Table)
```

Fig. 8.10 IFT EXAMPLES

- Boolean expression OR Boolean expression
- value expression <relational operator> value expression

Active code: Gnn,Mnn — TRUE if code is *active* (i.e. currently modal)

Relational operator: =, <>, <, >, <=, >
(<> is the not equal sign)

Value expression: +, -, *, /, (,), constant, letter address value, function (*multiply) (/divide)

Constant: 99999999. > 0 > -99999999.

Letter address value: A, B, C, E, F, I, J, K, L, P, Q, R, S, T, U, V, W, X, Y, Z
Values used are commanded programmed positions (as displayed on main display..includes offsets). These values are valid only when the machine is configured to accept them.

Function: SQ, SQRT, ABS, SIN, COS, TAN, ASN, ACS, ATN, FRA, INT, LOG, ALOG, LN, EXP

Parameter address: L1 . . . Ln

Type I data: as defined in EIA spec. RS274D

Operator precedence when parentheses are not used:

1. NOT and DEF
2. AND
3. OR
4. Relationals =,<,>,<>...

IFT, . . . ,PARAMETER ASSIGNMENT

This operates similar to the IFT,..,GTO with PAR (parameter assignment) replacing the GTO. The assignment is made ONLY IF the boolean expression is TRUE.

(IFT,boolean expression,(PAR,L801=X,L803=14.5))
↑

If the boolean expression is TRUE, when the block goes active, L801 gets the X axis position and L803 is assigned 14.5.

If the boolean expression is FALSE, the next block, the one following the IF statement, is executed and no change to L801 and L803 is made.

The following items may be used in the construction of IFT Boolean Type II data expressions:

- Active code
- NOT Boolean expression
- DEF(parameter address)
- Boolean expression AND Boolean expression
- Boolean expression OR Boolean expression
- value expression <relational operator> value expression

Active code: Gn,Mn — TRUE if code is *active* (i.e. currently modal)

Relational operator: =, <>, <, >, <=, >=
(<> is the not equal sign)

Value expression: +, -, *, /, (,), constant, letter address value, function (*multiply) (/divide)

Constant: 99999999. > 0 > -99999999.

```
N100(IFT,G70,(PAR,L835 = 2.54))   ! If metric mode then assign 2.54

N120(IFT,G71,(PAR,L835 = .1))    ! If inch mode then assign .1
```

Examples showing placement of parentheses in (IFT,...) statements:

```
(IFT,DEF(L1),G91)
(IFT,NOTDEF(L1),G90)
(IFT,(L100 = 10.)AND(L200 = 5.),(PAR,L1 = 5.))
(IFT,NOTDEF(L5)ANDNOTDEF(L6),G91)
(IFT,NOTDEF(L5)ORNOTDEF(L6),G90)
```

Fig. 8.11 MORE IFT EXAMPLES

Letter address value: A, B, C, E, F, I, J, K, L, P, Q, R, S, T, U, V, W, X, Y, Z

Values used are commanded programmed positions (as displayed on main display, includes offsets). These values are valid only when the machine is configured to accept them.

Function: SQ, SQRT, ABS, SIN, COS, TAN, ASN, ACS, ATN, FRA, INT, LOG, ALOG, LN, EXP

Parameter address: L1 . . . Ln

Type I data: as defined in EIA spec RS274D

Operator precedence when parentheses are not used:

1. NOT and DEF
2. AND
3. OR
4. Relationals =,<,>,<>...

AUDIBLE ALARM (M99)

Turning on the audible alarm has become necessary, now that the part programs can make decisions (IFT statements). For example, if a spindle probe routine determines that a part is out of tolerance, a program stop, a message and an alarm is programmed to alert the operator. The alarm is handy when operators run more than one machine and may not be present.

An M99 will turn on the AUDIBLE ALARM when executed in a part program block and the message "M99 alarm - see MAIN display" will be written to the system message display. M99 is an "immediate" (with motion) function. Below is an example program segment where the alarm is used:

```
N140(CLS,BORE CENTER,1,L1=4.L2=3.) !get bore center in L100
N150(IFT,L100>=5.0000,GTO 250)   !check X axis against spec
N160
N170 M99 M0      !X Bore center < 5.0000 ... Please check fixture
N180(LBL,250).....
```

Execution of block N170 would stop the machine and turn on alarm so the operator will check the fixture.

(G79) Wait for Precompute of Maximum Number of Ready Blocks

To allow for improved performance during high speed contouring a non-modal G code "G79" will inhibit any further ready blocks to be transferred active until the control has precomputed ahead as far as possible. Additional memory resources will also be made available when the block containing the G79 code goes active. These additional memory resources will be available until the program terminates by a M2 or M30 or until a clear all logic is performed.

The first occurrence of G79 in a part program causes the additional memory resources to be made available and the transfer of ready blocks will be held up until the control has precomputed ahead as far as possible.

Subsequent occurrences of G79 in the part program will only cause the control to precompute ahead as far as possible, since additional memory resources had already been allocated on the first occurrence of G79.

NOTE

G79 should be used only when necessary to attempt to satisfy marginal contouring rates. System updates and additional features (new configuration tape) causing more memory resources to be used for software system overhead may cause a change in G79 contouring performance. Pausing may occur because of the lack of precomputing (less ready blocks available). Be aware of these marginal situations. These may require a change in the part program or process.

DAT Update Date / Time Parameters

The DAT Type II data command is used to use the current time/date information in a file in conjunction with the FWR command as in the following example. (DAT) is used to update parameters L3061...L3066 with the current year, month, day, hour, minute, and second. The format of this data is as follows:

L3061 - Year i.e.. 1988.0000 for 1988
L3062 - Month i.e.. 2.0000000 for February
L3063 - Day i.e.. 15.000000 for 15th day
L3064 - Hour i.e.. 17.000000 for 5:00 P.M.
L3065 - Minute i.e.. 12.000000 for 12 minutes
L3066 - Seconds i.e.. 5.0000000 for 5 seconds

```
(OPF, INSPECTION FILE120)
(DAT)
(FWR,"DAY="L3063)
(FWR,"HOUR="L3064)
(FWR,"MINUTE="L3065)
```

This would cause the output inspection file to show the current time at which the control had begun to record the inspection results.

PROBE CYCLES

The following G commands come with KT-APL when configured with probe cycles. These G commands will create user defined probe cycles. For examples of these commands, refer to the Program Listings for G24, G25, G26 and G27 in Appendix E and Chapter 9.

G22 - Record all axes positions upon probe "contact", halt axis motion and begin processing next NC part program block. No axis motion occurs if probe is in "contact" when G22 is processed. G22 is non-modal.

G23 - Record all axis positions on probe "non-contact", halt axis motion and begin processing next NC part program block. No axis motion occurs if probe is in "non-contact" state when G23 is processed. G23 is non-modal.

Axis "contact" or "non-contact" positions are loaded into parameters L3042 thru L3046 for axes X, Y, Z, B, A or C, respectively.

Probe status information is loaded into L1020 and L1021. L1020 should be loaded with a 0 by the part program before the G22 / G23 is called for. L1020 will be loaded with a non-zero value upon completion of the G22 or G23. L1021 is a report of the status of the last completed G22 or G23. L1021 will be loaded with an 8.000000 if last G22 or G23 completed successfully. It will be loaded with a 1.0000000 if probe was in "contact" at beginning of the last G22 or in "non-contact" at the beginning of the last G23. If this status was returned no axis motion occurred. It will be loaded with a 2.0000000 if "contact" did not occur during last G22 or "non-contact" during last G23.

G28 - Obstruction detection. Modal function which causes E-stop and message upon any probe contact.

G29 - Cancel Obstruction detection (G28).

G60 - Cancel (G61) or (G62)

G61 - Modal G code used to count the number of probe contacts. Total count is recorded in parameter number L1022. This count will be reset to 0 at the beginning of a block containing a G61. Probe in the contact state when G61 is processed will not contribute to the count.

G62 - Modal G code used to count the number of probe non-contacts. Total count is recorded in parameter number L1022. This count will be reset to 0 at the beginning of a block containing a G62. Probe in the non-contact state when G62 is processed will not contribute to the count.

G92 - Non-Modal G code to regrid the specified axes at the position specified by the letter address. (i.e. G92 X11. Y12. Z13. will cause the absolute position of X, Y, and Z to be loaded with 11., 12., and 13., respectively.)

M20 - Modal M code used to select spindle probe. It will be set as a default control mode state upon power on or a [CLEAR ALL LOGIC].

M21 - Modal M code to select auxiliary probe. It is cleared upon [POWER ON], M20, M2, M30, or [CLEAR ALL LOGIC].

The system will also provide "E-stop / Loss of Carrier" protection during G22 / G23 / G28 / G61 / G62 and display a message, turn on the audible alarm, and issue E-stop if Loss of Carrier signal is detected.

LOADING SUBROUTINE/CYCLE FILES

If subroutines or cycles are to be loaded unto the disk via a punched tape the tapes must terminate with a block which contains an M2 or M30. In actual execution of the subroutine or cycle, this block should never execute since all subroutines and cycles are normally terminated by a block with an (END) command. Attempting to load files which terminate simply with an (END) block will cause the copy function of the file handler to attempt to read beyond the end of the file (tape will run off the end). An error will result and no files will be loaded unto the disk.

LOADING SUPPLIED DEFINED CYCLES

The following defined cycles are provided on punched tape for every KT-APL control system: G81, G82, G83, G84, G85, G86, G87, G88, and G89. Optionally provided, if probe cycles feature is to be supplied: G24, G25, G26, G27, G28, and G29 and the subroutines for the Multipoint Tool Sensor. These cycles are provided as a part program on a punched tape and are loaded unto the disk as follows:

1. Using the file handler, copy the tape onto the disk and call it any convenient name.
2. Assign the program you have just put on the disk to run by using the assign program / pallet feature.
3. Run the part program. No axis motion should occurred. When the part program finishes, all the provided cycles should now be properly loaded onto the disk and ready for use.

MULTIPLE AXES

KT-APL can support 8 synchronous axes which may be configured as linear or circular, as desired. Linear axes, in addition to the usual 3 axes, must be configured as being parallel to the first 3 linear axes.

In order to program circles in any plane using parallel coaxes it is necessary to include G20,21 commands. The proper method for programming these circles is discussed in Appendix G.

CHAPTER 9

PROGRAMMING EXAMPLES

INTRODUCTION

The following are examples of programs written in KT Advanced Programming Language. The text that appears in the lower case is intended to explain the program text. That which is in the upper case is in the actual program. The system user must determine how much text is required to be used in each program. If the programmer wishes to include the text statements in the program, the comment sign ! must precede the text. Additionally, space has been introduced between some program blocks to improve comprehension.

PROGRAMMING GUIDELINES

1. Determine if user defined cycles or subroutines are required by the part program.
2. Determine what type of parameters are needed local, global, machine, etc.
3. If parameters are being created, make a list of the parameter numbers being used to avoid creating conflicting parameter numbers.
4. Avoid the use of non-motion blocks in areas where the control must be executing time critical decisions.
5. To decrease program execution time, make use of the disk to store defined cycles or subroutines.
6. Make the defined cycle or subroutine files as small as possible by combining as many commands into a single program block. Disk subroutines are loaded into to their own buffer areas and are kept there if 512 characters or less in length.
7. Combine multiple statements, like PAR statements, into single blocks.
8. If possible, before program motion begins as many parameter calculations should be done and stored in the parameter table. It is more effective for the control to access the result of a calculation than having to constantly recalculate the equation and then apply the result to the program.
9. Delete unnecessary parameters from the parameter table. It is faster for the control to search a short table than a long one.
10. It is faster for the control to process compact expressions like (PAR,L24=3.*3.1/5.6) rather than (PAR, L24 = 3. * 3.1 / 5.6).
11. Make use of decimal point placement to improve processing speed: X24. is faster than X240000.
12. When using mathematical expressions pay attention to the hierarchical rules and use the least number of parentheses required.
13. Try to keep LABEL and GTO commands in close proximity to each other. The control maintains a buffer storage of the last 10 label commands that it has accessed. It is quicker to call a label from the buffer than to search the program for it.
14. Keep the number of blocks inside a loop to the minimum required by the loop.
15. Comments (!) should be used as required to provide clarity, but they do contribute to the amount of information being processed by the control.
16. When creating IFT statements, attempt to make the boolean expression evaluate to false, if possible. However, do not substitute other slower statements to follow this rule.
17. Write the program to direct execution to specific areas that can handle specific cases of input.

The reader should be aware that the following are intended as format examples and other approaches exist to solve problems.

BOLT HOLE CIRCLE

Program name: Bolt Hole Circle
(The following table of information is shown to help define bolt hole parameters for the reader.)

L101= X CENTER OF BOLT CIRCLE
L102= Y CENTER OF BOLT CIRCLE
L103= DIAMETER OF BOLT CIRCLE
L104= NUMBER OF HOLES IN BOLT CIRCLE
L105= ANGLE TO FIRST HOLE
L106= ANGLE BETWEEN HOLES
L107= SPINDLE SPEED (CW)
L108= FEEDRATE
L109= PRIMARY CLEARANCE PLANE
L110= SECONDARY CLEARANCE PLANE
L111= DEPTH POSITION
L112= FEED INCREMENT
L113= FIXED CYCLE NUMBER
L114= INCH OR METRIC -G- CODE

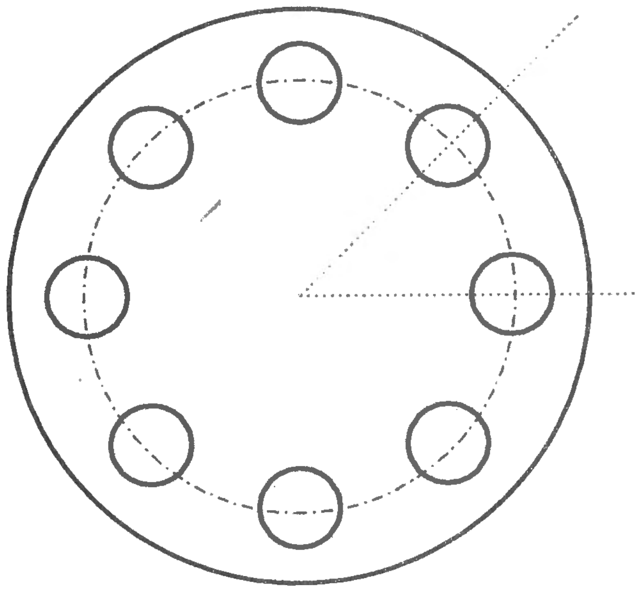


Fig. 9.1 BOLT HOLE CIRCLE

The following user defined subroutine is created by the programmer:

```
(DFS,B.C.,SAVE)
(IFT,L110<L109,(PAR,L110=L109))
(PAR,L1=1.,L6=L105)
(LBL,1)(PAR,L2=(L103/2.)*COS(L6),L3=(L103/2.)*SIN(L6))
! X & Y Change
G78GL114G0G90ZL110F0SL107M3
G78GL113X,L101+L2,Y,L102+L3,RL109ZL111EL112FL108
G80
G78ZL110F0
(IFT,L1=>L104,(GTO,2))
(PAR,L1=L1+1.,L6=L6+L106)
(GTO,1)
(LBL,2)(END)
```

DEMONSTRATION B.C.

```
N10G00G70G90G94X10.Y15.Z20.M25  !Restart block
N20T11111M6                     !Load tool
N30(PAR,L101=10."X CENTER OF B.C.")
N40(PAR,L102=15."Y CENTER OF B.C.")
N50(PAR,L103=6.0"DIAMETER OF B.C.")
N60(PAR,L104=8.0"NUMBER OF HOLES")
N70(PAR,L105=45."ANGLE TO FIRST HOLE")
N80(PAR,L106=45."ANGLE BETWEEN HOLES")
N90(PAR,L107=300."SPINDLE SPEED")
N100(PAR,L108=50."FEEDRATE")
N110(PAR,L109=18."PRIMARY CLEARANCE")
N120(PAR,L110=0"SECONDARY CLEARANCE")
N130(PAR,L111=16."DEPTH POSITION")
N140(PAR,L112=0"FEED INCREMENT")
N150(PAR,L113=81."FIXED CYCLE NUMBER")
N160(PAR,L114=70."INCH OR METRIC")
```

The above parameter statements are used to load information into the parameter table. The parameter table contains the L number,the numerical which appears after the equal sign and the verbal description that is contained between the quotation marks.

```
N180(CLS,B.C.)     !Calls the subroutine.
N190T22222M6       !A new tool is loaded.
N200(CLS,B.C.,L107=200.,L108=10.,L109=18.2,L1
13=84.)    !Some of the parameters are changed and
           the subroutine is performed again.
N210X10.Y15.Z20.M25   !Move to clearance position.
N220M30            !Program end.
```

B AXIS ROTATION-SHORTEST ROUTE

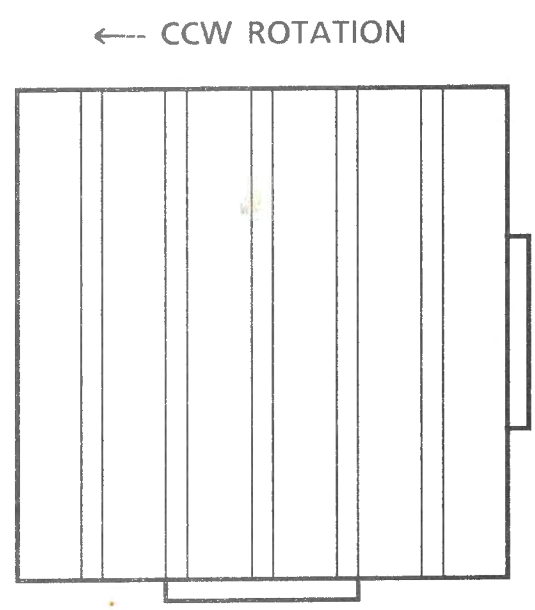


Fig. 9.2 B AXIS ROTATION

Program name: B Axis Rotation-Shortest Route

L1=COMMANDED B POSITION
L2=CURRENT B POSITION

The cycle is defined by the following:

```
(DFS,B,SAVE)   !Rotates B axis + or - (shortest route)
(IFT,L1=0,(PAR,L1=360.))
(PAR,L2=B,L3=L1)        !L2 is the B axis position
```

```
(IFT,L1<L2,(PAR,L3=L1+360.))
(IFT,180.<ABS(L3-L2),(PAR,L1=-L1))    !If L3-L2 is
                     greater than 180°, B rotation will be CCW.
BL1F0          !Rotate B axis at rapid
(END)

(DFS,120RBASE1,SAVE) !Part 120R fixture offset
                     station 1 (PAR,L101)
H1G48
(CLS,B,L1=90.)    !Subroutine B becomes nested in
                  subroutine called 120BASE1
(END)
(CLS,120RBASE<L200I>)
M2
```

POINT ROTATION EXAMPLE PROGRAM

Program name: DFCG802 Point Rotation

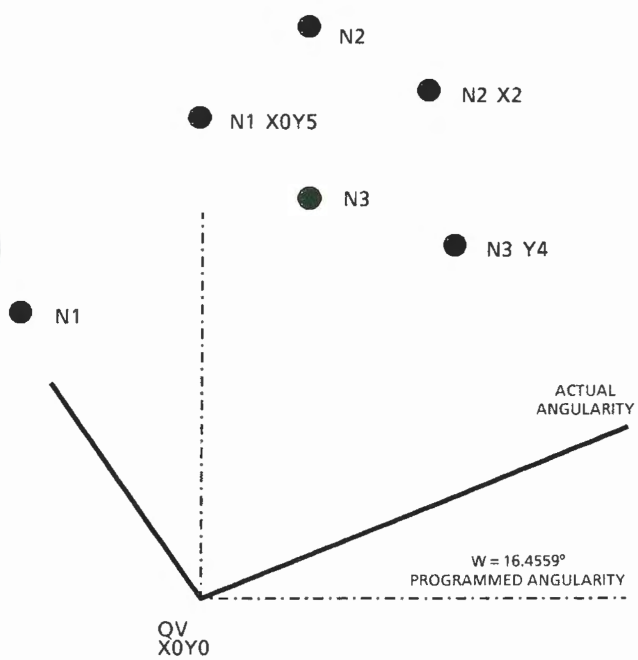


Fig. 9.3 POINT ROTATION

G802 Rotation Cycle (modal). This program is being created as user defined cycle. Once it is called it will remain in effect until canceled with a G80.

Functional Description: G802XYQVW

where:
X = X Coordinate of Point
Y = Y Coordinate of Point
Q = X Coordinate of Rotation Point
V = Y Coordinate of Rotation Point
W = Angle of Rotation (in degrees ±xxx.xxxx format)

When programmed with X, Y, Q, V and W-word information, the control automatically recalculates X and Y coordinates relative to Q and V rotation point plus W angle and positions to the calculated positions.

NOTE

Plus rotation = CCW, 0° = 3 o'clock position.

Programming application: used for rotation of points, for milling and positioning

```
(DFC,G802,SAVE)
(IFT,NOT(DEF(L806)),(PAR,L806=F))    !The blocks following
                      are the calculations required to adjust the
                      coordinates.
(PAR,L896=L824-L817,L897=L825-L822,L850=COS(L823),
                                         L851=SIN(L823))
(PAR,L898=L896*L850-L897*L851+L817,L899=L896*L851
                                         +L897*L850+L822)
(IFT,NOT(DEF(L809)),(GTO,1))  ! If "I" is not defined, go to 1
(PAR,L870=L809-L817,L871=L810-L822,L872=L870*L850-
                                         L871*L851+L817)
(PAR,L873=L870*L851+L871*L850+L822)
(LBL,1)G78GL807AL801BL802CL803XL898YL899ZL826IL872
                                  JL873FL806SL819ML813
(DLP,L807,L809,L810,L872,L873)
(END)
```

Example 1: Rotates three points 16.4559°.

```
Program coordinates   commanded angle
      ↓ ↓ ↓ ↓           ↓
N1G802X0Y5.Q0V0W16.4559F10.S100M3
    ↑__The cycle is called

N2X2.    !A new X position is commanded. The
         spindle is moved to the X,Y location.
N3Y4.    !A new Y position is commanded.
N4G80    !Terminate cycle with a G80 command
```

Blocks automatically performed in cycle:

N1 - X & Y will move to X-1.4164Y4.7952 at 10. IPM and spindle 100 RPM CW.
N2 - X & Y will move to X.5017Y5.3617 at 10. IPM
N3 - X & Y will move to X.7850Y4.4027 at 10. IPM
N4 - Cancel cycle

: : : :
Additional program blocks
: : : :

M30

Example 2:

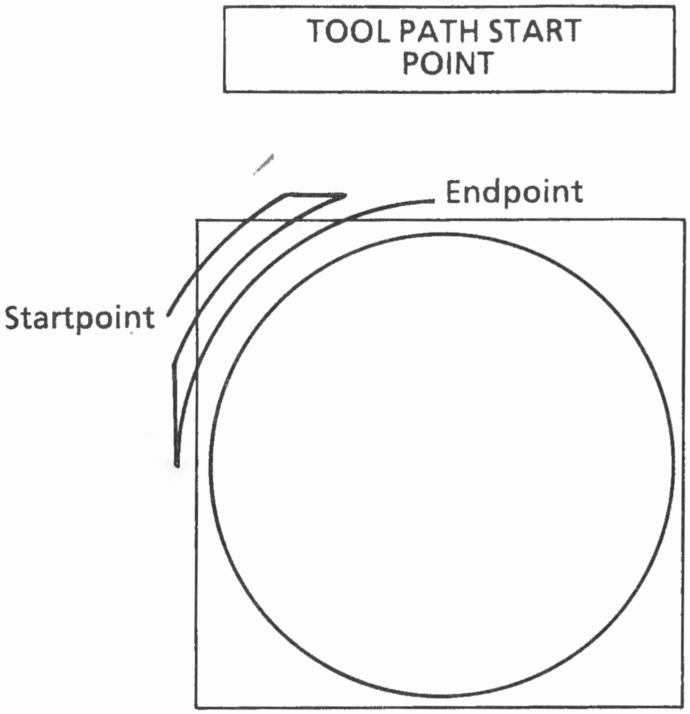


Fig. 9.4 MILLING A CIRCLE

The ALUM.CIR subroutine was developed to rough mill four corners of square, into a circle. The upper left X and Y coordinates were manually calculated for this corner. The remaining corners are rotated and recalculated within the subroutine.

```
(DFS,ALUM.CIR,SAVE)
(PAR,L1=0)
G17G90G94G70G01
(LBL,1)G802Q0V0WL1X-5.965Y3.6918SL188F0M3
```

The G802 rotation cycle is called and the coordinates are loaded into it.

```
Z-.46M7
G2X-3.9969Y5.765I0J0FL189    !Rough mill the circle
X-2.4525
G03X-5.765Y2.4526I0J0
X-5.615Y0
G02X0Y5.615I0J0
M9Z.25
(PAR,L1=L1+90.)
(IFT,L1<360.,(GTO,1))
(END)
```

Control is commanded to go to label 1 and the points are translated and rotated. The process is repeated three times to satisfy the label 1 definition of 360°.

```
N1(CLS,ALUM.CIR,L188=10000.,L189=100.)
```

CAUTION

This program was created for a machine with a high speed spindle.

```
N2X0Y5.6        !Finish mill circle
N3G2X0Y5.6I0J0
N4Z20.M25
N5M30
```

PATTERN ROTATION-TRANSLATION EXAMPLE

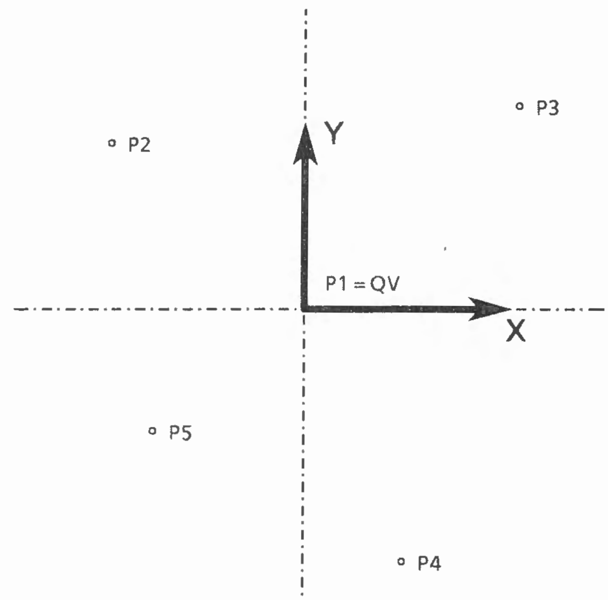


Fig. 9.5 G801 ROTATION-TRANSLATION

Program name: PAT-ROT-TRANS

Program PAT-ROT-TRANS is one method of defining, rotating and or translating patterns. Program was developed to be used with fixed cycles, G81-G89.

Functional Description: G801XYIJQVW

X = X Coordinate of point
Y = Y Coordinate of point
I = Incremental value of X translation
J = Incremental value of Y translation
Q = X Coordinate of rotation point

```
(DFC,G801,SAVE)
(PAR,L896 = L824-L817,L897 = L825-L822,L898 = (L896*COS(L823)-L897* SIN(L823)) + L817)
(PAR,L899 = (L896*SIN(L823) + L897*COS(L823)) + L822)          ! L898,L899 = Point rotated
(PAR,L898 = L898 + L809,L899 = L899 + L810)                    ! Point translated
G78GL807XL898YL899FL806SL819ML813
(END)
(DFS,PTS,SAVE)
G70G90G94G0
G801
G78XL101YL102WL103QL104VL105IL112JL113FL106
G80
G78GL107ZL108RL109EL110FL111
G80
(END)
(DFS,PAT1,SAVE)
(CLS,PTS,L101 = 0,L102 = 0)          ! Point 1
(CLS,PTS,L101 = -.75,L102 = .75)     ! Point 2
(CLS,PTS,L101 = 1.25)                ! Point 3
(CLS,PTS,L101 = .5,L102 = -1.5)      ! Point 4
(CLS,PTS,L101 = -.75,L102 = -1.)     ! Point 5
(DLP,L101,L102)
(END)
```

Fig. 9.6 ROTATION-TRANSLATION OF POINTS, WITH FIXED CYCLES G81-G89

V = Y Coordinate of rotation point
W = Angle of rotation (degrees, ±xxx.xxxx format)

Subroutine Program Parameter Descriptions:

L101 = X POINT COORDINATE
L102 = Y POINT COORDINATE
L103 = ANGLE OF ROTATION
L104 = Q
L105 = V
L106 = POSITIONING FEEDRATE
L107 = CYCLE
L108 = Z DEPTH
L109 = R CLEARANCE
L110 = E INCREMENT IF REQUIRED
L111 = FEEDRATE
L112 = X TRANSLATION
L113 = Y TRANSLATION

The programmer must build correct (PAR) statements

```
N1(PAR,L103 = 45.,L104 = 0,L105 = 0,L106 = 0,L108 = -1.)
N2(PAR,L109 = 0,L111 = 20.,L112 = 0,L113 = 0)
N3(CLS,PAT1,L107 = 81.)                      ! Drill PAT1 rotated 45°
N4(CLS,PAT1,L103 = 0,L107 = 81.,L112 = 2.,L113 = 1.)
                                   ! Drill PAT1 translated +2.X + 1.Y
N5(CLS,PAT1,L107 = 81.,L112 = 0,L113 = 0)    ! Drill PAT1 normal
N6(DLP,L103,L104,L105,L106,. . . ,L110,L111,L112,L113)
N7M30
```

USER-DEFINED PROBE CYCLE PROGRAM UNDERSIZE BORE G824

Program name: DFCG824 Undersize Bore

Format: G824A0ER

A0 = Used to trigger user-defined cycle
E = Minus tolerance
R = Expected bore diameter

If bore is undersize, parameter L101 = 1. If bore is on size or oversize, parameter L101 = 0. Parameter L101 is used to prebore a feature in another subroutine.

DEMONSTRATION G824 CYCLE

```
(PAR,L1005 = 2.)
G824A0R2.E.005                  The cycle is called.
M0     !OK L101 = 0             The part meets the spec.
G824A0R1.994E.005               The cycle is called.
M0     !OK-Oversize L101 = 0    The part is oversized.
G824A0R2.006E.005               The cycle is called.
M0     !Tolerance exceeded L101 = 1.  The part must
                                      be machined.
M2
```

```
(DFC,G824,SAVE)
(IFT,DEF(L807),G800M0M99)      ! Error - No other "G" codes allowed in probe cycle
(CER,MAS)
(IFT,G71AND(L3033<114.483),G800M0M99)     ! Error - Illegal Tool Compensation
(IFT,G70AND(L3033<4.5072),G800M0M99)      ! Error - Illegal Tool Compensation

 System parameters are set for probe length compensation.

(IFT,NOTDEF(L818),M0M99G80)      !Error - Need "R", bore diameter in G824 block
(IFT,NOTDEF(L805),M0M99G80)      !Error - Need "E", minus tolerance in G824 block

 Parameters are set for bore diameter and tolerance.

(IFT,DEF(L801),(GTO,100))
G800M0M99              ! Error - unrecognized G824 format if A is not in the command cycle stop.
(LBL,100)(PAR,L832=(L1005-L818))       ! The surface separation is determined. L1005 is defined in
                                         Appendix B.
(PAR,L101=0"PRE-BORE NOT REQUIRED")        ! The specification is checked.
(IFT,L832<-L805,(PAR,L101=1."PRE-BORE"))
G800
(END)
```

Fig. 9.7 USER-DEFINED PROBE CYCLE PROGRAM UNDERSIZE BORE G824

G892 BACKSPOTFACING USER DEFINED CYCLE

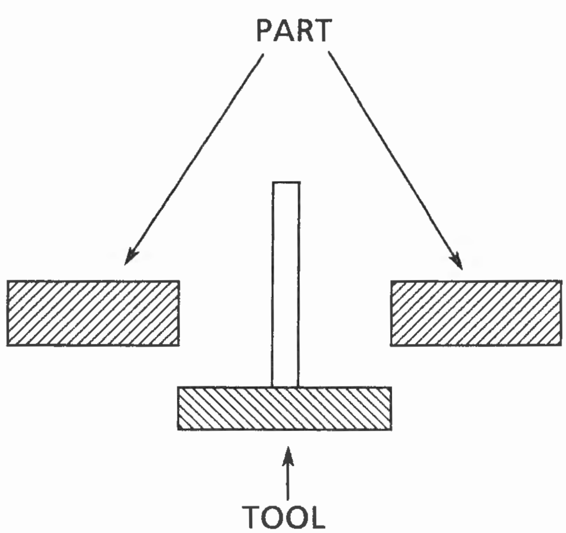


Fig. 9.8 BACKSPOTFACING

Program name: DFCG892

Format: G892XYZRFSMVW

X=PROGRAM COORDINATES
Y=PROGRAM COORDINATES
Z=PROGRAM COORDINATES
F=FEEDRATE
R=CLEARANCE POSITION
S=SPINDLE SPEED
M=M3 or M4
V=CENTERLINE OFFSET
W=SECONDARY CLEARANCE

```
(DFC,G892,SAVE) !Back Spotfacing cycle
(IFT,G91,M99M0G80)
(IFT,NOT(DEF(L806)),(PAR,L806=F))      ! Feedrate
(IFT,NOT(DEF(L818)),(PAR,L818=R))  ! R Clearance
(IFT,NOT(DEF(L822)),M0M99)!Error need centerline
(IFT,NOT(DEF(L823)),M0M99) !Error need secondary
                                            clearance
(IFT,NOT(DEF(L826)),(PAR,L826=Z))
(IFT,L818>Z,ZL818F0)      ! Parameters are defined.
G78AL801BL802CL803EL805RL818XL824YL825F0
M19
ZL818F0         !Move to clearance at F0.
X,L824+L822        !Move to centerline offset.
(IFT,G70,ZL823F100.)        !Move at 100 ipm.
(IFT,G71,ZL823F2540.)
XL824       !Move to commanded X axis.
(PAR,L835=L826-.01)         !Move in Z minus .01 of
                             commanded depth.
(IFT,G71,(PAR,L835=L826-.254))
```

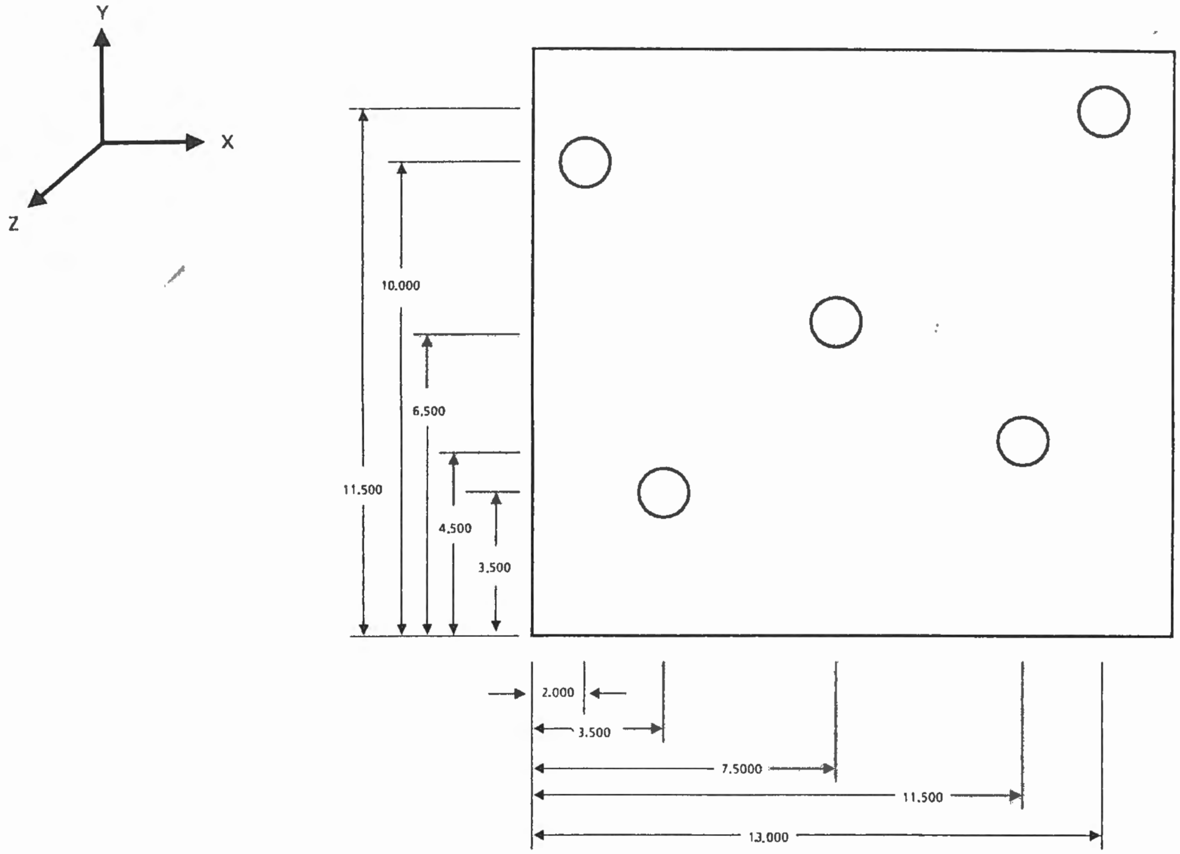


Fig. 9.9 PATTERN EXAMPLE

```
SL819ML813
ZL835FL806
S100
(IFT,G71,F12.7ZL826)      !Feed in 12.7 mmpm to Z
                          depth.
(IFT,G70,F.5ZL826)        !Feed in .5 ipm to Z depth.
G4F2.                   !Dwell 2 seconds.
(IFT,G70,ZL823F100.)
(IFT,G71,ZL823F2540.)
M19                       !Keylock.
X,L824+L822             !Move in X to centerline offset.
ZL818                   !Retract to clearance.
XL824F0
SL819FL806
(DLP,L801,L802,L803,L824,L825) !Delete parameters
(END)
```

PROGRAMMING SHORT CUTS FOR USER DEFINED CYCLES

This program is one method of programming patterns for G81-G89 cycles and user-defined cycles. The program was developed for absolute machine coordinates. No fixture or tool length offsets required.

PARAMETER ASSIGNMENTS

L101 = X BASE.
L102 = Y BASE.
L103 = Z BASE.
L104 = FIX CYCLE NUMBER.
L105 = TOOL LENGTH.
L106 = Z DEPTH.
L107 = E INCREMENT.
L108 = FEEDRATE.
L109 = SPINDLE SPEED.
L110 = V CENTERLINE OFFSET.
L111 = W SECONDARY CLEARANCE.

```
(DFS,PTS,SAVE)
(PAR,L3=.1)                       !Z Tool clearance
(IFT,L104=84.,(PAR,L3=.25))  !Z Tap clearance
(PAR,L4=L101+L1,L5=L102+L2)
(IFT,L104<100.,(GTO,1))
GL104XL4YL5Z,L103+L105-L106,R,L103+L105+L3,VL110W,
                          L103+L105-L111,FL108SL109M3
(GTO,2)
(LBL,1)GL104XL4YL5Z,L103+L105-L106,R,L103+L105+L3,
                                  EL107FL108SL109M3
```

```
(LBL,2)(END)
(DFS,PAT1,SAVE)
!L1=X Print dimension of point, L2=Y Print dimension of point.
(CLS,PTS,L1=2.,L2=10.)          !PT.1
(CLS,PTS,L1=3.5,L2=3.5)         !PT.2
(CLS,PTS,L1=7.5,L2=6.5)         !PT.3
(CLS,PTS,L1=11.5,L2=4.5)        !PT.4
(CLS,PTS,L1=13.,L2=11.5)        !PT.5
G80
(DLP,L104,L105,L106,L107,L108,L109)
(END)
```

DEMONSTRATION PATTERN

```
N1(PAR,L101=15.-7.5"X BASE",L102=5.5"Y BASE",L103=3.+
                                              4.-6."Z BASE")
N2G90G94G00G70X30.Y11.5Z20.B0F0
N3T202M19M6          !1.25 Dia. Spotdrill
N4T55555
N5(CLS,PAT1,L104=81.,L105=10.,L106=.5,L107=0,L108=6.,
                          L109=500.) !Spotdrill PAT1
N6Z20.M25
N7T55555M19M6        !1.0 Dia. Drill
N8T77775
N9(CLS,PAT1,L104=87.,L105=10.5,L106=4.5,L107=.5,
                        L108=5.,L109=400.) !Drill PAT1
N10Z20.M25
N11T77775M19M6       !1.06 Dia. Backspotfacer
N12(PAR,L104=892.,L105=12.)
N13(CLS,PAT1,L106=3.97+.25,L107=0,L108=6.,L109=320.,
                             L110=.06,L111=4.+.1+.25)
N14Z20.M25
N15X30.Y11.5F0
N16M30
```

SHORT CUTS FOR G81-89 CYCLES EXAMPLE PATTERN 2

Program name:PATTERN-2

This program is another method of programming patterns for G81-G89 cycles. Also, see the above pattern example and illustration. The program was developed for absolute machine coordinates. No fixture or tool length offsets required.

PARAMETER ASSIGNMENTS

L101 = X BASE.
L102 = Y BASE.
L103 = Z BASE.
L104 = FIXCYCLE NUMBER.
L105 = TOOL LENGTH.
L106 = Z DEPTH.
L107 = E INCREMENT.
L108 = FEEDRATE.
L109 = SPINDLE SPEED.

```
(DFS,PTS,SAVE)
(PAR,L3=.1)!Z TOOL CLEARANCE
(IFT,L104=84.,(PAR,L3=.25))  !Z Tap clearance
GL104X,L101+L1,Y,L102+L2,Z,L103+L105-L106,R,L103+
                              L105+L3,EL107FL108SL109M3
(END)
(DFS,PAT1,SAVE)
! L1=X Print dimension of point, L2=Y Print dimension of point.
(CLS,PTS,L1=2.,L2=10.)          !PT.1
(CLS,PTS,L1=3.5,L2=3.5)         !PT.2
(CLS,PTS,L1=7.5,L2=6.5)         !PT.3
(CLS,PTS,L1=11.5,L2=4.5)        !PT.4
(CLS,PTS,L1=13.,L2=11.5)        !PT.5
G80
(DLP,L104,L105,L106,L107,L108,L109)
(END)
```

DEMONSTRATION PATTERN-2

```
N1(PAR,L101=15.-7.5"X BASE",L102=5.5"Y BASE",L103=3.+
                                              4.-6."Z BASE")
N2G90G94G00G70X30.Y11.5Z20.B0F0
N3T202M19M6          !.500 Dia. Spotdrill
N4T55555
N5(CLS,PAT1,L104=81.,L105=9.5,L106=.15,L107=0,L108=6.,
                          L109=500.) !Spotdrill PAT1
N6Z20.M25
N7T55555M19M6        !5/16 Dia. Drill
N8T77775
N9(CLS,PAT1,L104=87.,L105=10.,L106=2.094,L107=.5,
                       L108=10.,L109=800.) !Drill PAT1
N10Z20.M25
N11T77775M19M6       !3/8-16 Tap
N12(CLS,PAT1,L104=84.,L105=11.,L106=1.186,L107=0,
                       L108=20.,L109=320.)   !Tap PAT1
N13Z20.M25
N14X30.Y11.5F0
N15M30
```

FINDING A BORE SIZE EXAMPLE

Program name:(DFS,FINDBORE1)

"FINDBORE" subroutine was developed to establish X, Y and B fixture offset positions for workpiece or fixture bore. The following parameters must be programmed in the (CLS,FINDBORE) block:

L1=X program position of bore
L2=Y program position of bore
L3=B program position of bore

L4 = X contact
L5 = Y contact
L6 = H fixture offset line number

The programmer must construct a (PAR) statement which would load the coordinate values into the parameter table.

```
(DFS,FINDBORE,SAVE)
H0G29        !Fixture offsets are canceled.
G27XL4       !Probe X. Part is probed using a G27 command.
G27YL5       !Probe Y. Contact results are recorded in the
G27XL4       !Probe X. parameter table.
G27YL5       !Probe Y.
(PAR,L1=X-L1,L2=Y-L2,L7=L3,L3=B-L3)   !Calculate initial
                                       X,Y,B deviation
(FXC,HL6IL7XL1YL2BL3) !Establish X,Y,B initial offset values
(END)
```

DEMONSTRATION FINDBORE

```
N1G00G70G90G94
N2T77777M6
N3G28
N4           !** Position probe to probe bore. **
N5(CLS,FINDBORE,L1=10.,L2=11.5,L3=90.,L4=20.,L5=20.,
                                                L6=1.)
N6Z20.M25    !Retract for tool change
N7M6
N8           !** Activate H1 offset utilizing G48 or G49 mode.
N9           !** Continue part program.
N10M30
```

FIND THE Z SURFACE EXAMPLE

Program name:(DFS,FINDZ1)

"FINDZ" subroutine was developed to establish Zfixture offset position for workpiece or fixture reference. The following parameters must be programmed in the (CLS,FINDZ) block:

L1 = Z program position
L2 = Z position past surface for contact
L3 = H fixture offset line number
L4 = B program position

```
(DFS,FINDZ,SAVE)
H0G29                  !Fixture offsets are canceled.
G26ZL2                 !Probe Z surface.   Surface detection cycle.
(PAR,L1=(Z+L3033)-(L1+L3029))  !Calculate initial Z deviation.
(FXC,HL3IL4ZL1)        !Establish Z initial offset value.
(END)
```

DEMONSTRATION FINDZ

```
N1G00G70G90G94
N2T77777M6
N3G28
N4           !** Position probe to probe Z surface. **
N5(CLS,FINDZ,L1=6.,L2=-10.,L3=1.,L4=90.)
N6Z20.M25    !Retract for tool change.
N7M6
N8           !** Activate H1 offset utilizing G48 or G49 mode.
N9           !** Continue part program.
N10M30
```

USER DEFINED CYCLE G803

This command is used in the roughing and finishing cut example program that follows this example. For definition of the parameters see the listing for the G26 probe command.

```
(DFC,G803,SAVE) !Same as G26Z without pullback after contact
(CER,MAS)
IFT,G71 AND (L3033<114.483),G800M0M99)  !Error -illegal tool
                                         compensation
IFT,G70 AND (L3033<4.5072),G800M0M99)   !Error -illegal tool
                                         compensation
(PAR,L843=91.)
(IFT,G90,(PAR,L843=90))
IFT,G71,(GTO,50))
(PAR,L833=.040,L834=1.8,L844=100)
(GTO,60)
(LBL,50)(PAR,L833=1.016,L834=45.72,L844=2540)
(LBL,60)(IFT,DEF(L806),(GTO,65))
(PAR,L806=L844)
(LBL,65)(IFT,L806>844,(PAR,L806=L844))
(PAR,L831=0)
(IFT,DEF(L826),(PAR,L831=L831+32.))
(IFT,L831=32.,(GTO,100))G800GL843M0M99  ! Error -
                                         unrecognized
                                         format for G803
G800GL843M0M99        ! Improper probe status <>1., 2.,or 8.
(LBL,100)(IFT,L843=91.,(GTO,110))
(PAR,L826=L826-Z)
(LBL,110)(PAR,L832=0,L1020=0)
M19SL832
G91G22ZL826FL806
(LBL,120)(IFT,L1020=0,(GTO,120))
(IFT,L1021=8.,(GTO,125))
(IFT,L1021=1.,G800GL831M0M99)  ! Error - probe was in contact
                                at start of G22
(IFT,L1021=2.,G800GL831M0M99)  ! Error - probe did not make
                                contact with surface
G800GL831M99M0      !Improper probe status <>1.2. or 8.
(LBL,125)(PAR,L826=L3044+L833,L1020=0)
G23G90ZL826
(LBL,126)(IFT,L1020=0,(GTO,126))
```

```
(IFT,L1021=8.,(GTO,127))
(IFT,L1021=1.,G800GL831M0M99)  ! Error - probe was in contact
                                 at start of G23
(IFT,L1021=2.,G800GL831M0M99)  ! Error - probe did not lose
                                 contact with surface
G800GL831M0M99           ! Improper probe status <>1.2. or 8.
G800GL831M99M0
(LBL,127)ZL826
(PAR.L826=L833*-2.,L1020=0)
G91G22ZL826FL834M49
M48
(LBL,130)(IFT,L1020=0,(GTO,130))
(IFT,L1021=8.,(GTO,135))
(IFT,L1021=1.,G800GL843M0M99)  ! Error - probe was in contact
                                 at start of G22
(IFT,L1021=2.,G800GL843M0M99)  ! Error - probe did not make
                                 contact with surface
G800GL843M0M99           ! Improper probe status <>1.2. or 8.
(LBL,135)G90ZL3044
(IFT,DEF(814),(FWR,"N",L814))
(FWR,"Z SURFACE =",Z)
(FWR,)
G800
(END)
```

EXAMPLE OF ROUGHING AND FINISHING CUT PROGRAM

Program name: DFSNO.OFCUTS

Space has been inserted between the end of the (PAR) statement and the comment ! sign for greater clarity. It would not be required to have this space in an actual program.

```
(DFS,NO.OFCUTS,SAVE)
(PAR,L11=5."TOOL LENGTH")          !6.0 dia. face mill gage length
(PAR,L12=4.0022"Z FIN.- PROBE")    !Probe Z finish surface pos.
(PAR,L13=200."FEED-RATE")          !Roughing feedrate
(PAR,L14=500."SPINDLE SPEED")      !Roughing spindle speed
(PAR,L1=L12+L11"Z FIN.- MILL")     !Face mill finish Z surface.
(PAR,L2=L100+L11"Z STOCK - MILL")  !Probed part stock ref.
(PAR,L3=.25"MAX. CUT DEPTH")       !Maximum stock removal per
                                    stock cut
(PAR,L4=.03"FIN. CUT DEPTH")       !Finish stock removal
(PAR,L5=L2-(L1+L4)"RGH. STK. REMOVAL")  !Total roughing
                                         stock removal
(PAR,L6=INT(L5/L3+.9999),L7=L5/L6)  !L6=no. of cuts,
                                     L7=depth per cut
                                     <=.25
(PAR,L6=L6"NO. OF RGH. CUTS",L7=L7"RGH. CUT DEPTH")
(PAR,L8=L2-L7"Z CUT POSITION")  !First cut Z position
(PAR,L10=0)                     !Run counter
(LBL,10)(IFT,L6=1.,(PAR,L8=L1+L4,L6=L6"LAST RGH CUT"))
(IFT,L6=0,(PAR,L6=L6"FIN. CUT",L8=L1,L10=L10+1.,
                                       L13=30.,L14=600.))
! L13=Finish feedrate,L14=Finish spindle speed
G90G94G70G00SL14M3X10.Y11.F0  !Position X and Y for milling
ZL8                  !Position Z for milling
X20.FL13             !Mill surface
Z,Z+.1,F0            !Retract Z to .1 clearance
(IFT,L10>0,(GTO,30))  !No more cuts - End subroutine
(PAR,L6=L6-1."NO. OF CUTS LEFT",L8=L8-L7"Z CUT
                      POSITION")  !Next cut Z position
(GTO,10)             !Cut again - new Z
(LBL,30)(END)
```

DEMONSTRATION NO.OFCUTS

```
N1Z20.M25G90G94G00G70
N2B0F0
N3X30.Y11.5
N4T77777M19M6 !Spindle probe
N5X15.Y11.F0G28!Position to probe Z surface
N6G29
N7Z-20.G803 !Probe Z surface using a user defined cycle.
N8(PAR,L100=Z"Z SURF. PROBE") !Z surface position
N9Z20.M25
N10X30.Y11.5F0
N11T202M19M6 !6. diameter face mill
N12(CLS,NO.OFCUTS)!Rough and finish mill surface
N13Z20.M25
N14X30.Y11.5M25
N15M30
```

SPIRAL CUTTING PROGRAM EXAMPLE

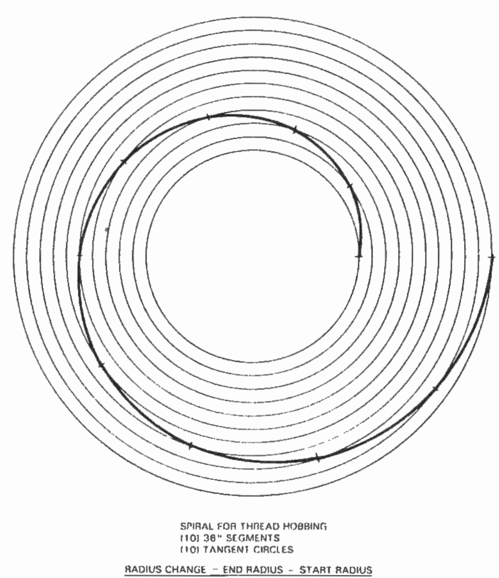


Fig. 9.10 SPIRAL THREADING

Program name: (DFS,SPIRAL,SAVE)

PARAMETER ASSIGNMENTS

L1 = X center
L2 = Y center
L3 = Start diameter
L4 = End diameter
L5 = Cutter diameter
L6 = Circular feedrate
L7 = Total Z travel
L40 = Angular displacement
L45 = Z start
L203 = Linear feedrate
L206 = Clearance plane

The programmer must construct the proper (PAR) statements to enter these values in the parameter table. In the subroutine, L60 = Rounding constant (.XXXXXXXXX ninth place).

```
(DFS,SPIRAL,SAVE)          !SPIRAL MILL THREADS
(PAR,L41=9.,L42=20.,L40=36.,L43=0,L8=(L4-L3)/20.,
                                        L60=.00000001/10.)
(PAR,L(L41)=((L3/2.-L5/2.+L8)/L60+.5)*L60)
(PAR,L7=L7/10.,L44=50.,L6=INT(L203*L3/(L3+L5)*10.)/10.)
(IFT,L6>10.,(PAR,L6=10.))
(LBL,1)(PAR,L43=L43+L40,L(L42)=((L(L41)*COS(L43)+L1)/
                                            L60+.5)*L60)
(PAR,L42=L42+1.,L(L44)=((L45+L7)/L60+.5)*L60)
(PAR,L(L42)=((L(L41)*SIN(L43)+L2)/L60+.5)*L60,L46=L(L41))
(PAR,L41=L41+1.,L(L41)=((L46+L8)/L60+.5)*L60,L42=L42+1.)
(PAR,L45=L(L44),L44=L44+1.)
(IFT,L41<19.,(GTO,1))
(IFT,L206>Z,ZL206F0)
G0XL1YL2F0
ZL206
Z,L45-L7/2.
G01G03G17G90X,L1+L3/2.-L5/2.,YL2ZL45R,(L3/2.-L5/2.)/2.,F,
                                                    L6/2.
G03XL20YL21ZL50RL9FL6
G03XL22YL23ZL51RL10
G03XL24YL25ZL52RL11
G03XL26YL27ZL53RL12
G03XL28YL29ZL54RL13
G03XL30YL31ZL55RL14
G03XL32YL33ZL56RL15
G03XL34YL35ZL57RL16
G03XL36YL37ZL58RL17
G03XL38YL39ZL59RL18
G03XL1YL2Z,L59+L7/2.,R,(((L38-L1)/2.)/L60+.5)*L60,F,L6/2.
ZL206F0
(END)
M0
```

DEMONSTRATION SPIRAL

```
(PAR,L203=20.,L206=.1)
(CLS,SPIRAL,L1=0,L2=0,L3=2.3662,L4=2.3716,L5=1.4778,
                                        L7=.087,L45=0)
M2
```

Program name: DFSSPIRAL1

This second example of a spiral routine uses the FWR command to preprocess the information.

PARAMETER ASSIGNMENTS IN CLS

L1 = X center
L2 = Y center
L3 = Start diameter
L4 = End diameter
L5 = Cutter diameter
L6 = Circular feedrate
L7 = Total Z travel
L8 = Z start
L203 = Linear feedrate
L206 = Clearance plane
L6,L27 = Feedrate
L9 = Incremental change of radius
L22 = X center
L23 = Y center
L10,L18,L25 = Radius
L11,L24 = X position of end point
L12 = Y position of end point
L13,L21 = Z position of end point
L14 = Angular step division of spiral
L15 = Angular start of spiral
L16 = Run counter
L17,L19 = Decimal place placement
L60 = Rounding position constant
L29 = Z clearance plane
L30 = Z start
L31 = Z end

```
(DFS,SPIRAL,SAVE)          !SPIRAL MILL THREADS
(PAR,L13=L8,L16=0,L14=36.,L15=0,L9=(L4-L3)/20.,
                              L17=10000.,L60=.00000001/10.)
(PAR,L7=L7/10.,L6=INT(L203*(L3-L5)/L5*10.)/10.,L10=L3/2.-
                                      L5/2.+L9, L19=.00001)
(IFT,L6>10.,(PAR,L6=10.))
(PAR,L27=(L6/2.)*10.+.5,L6=L6*10.,L25=((L3/2.-L5/2.)/2.)*L17
                                       +.5,L26=L8*L17-.5)
(PAR,L22=L1*L17+.5,L23=L2*L17+.5,L24=((L1+L3/2.-L5/2.)
                                       /L60+.5)*L19+.5)
(PAR,L29=L206*L17+.5,L30=(L8-L7*5.)*L17-.5)
(IFT,L22<0,(PAR,L22=L1*L17-.5))
(IFT,L23<0,(PAR,L23=L2*L17-.5))
(IFT,L24<0,(PAR,L24=((L1+L3/2.-L5/2.)/L60-.5)*L19-.5))
(IFT,L26>0,(PAR,L26=L8*L17+.5))
```

```
(IFT,L29<0,(PAR,L29=L206*L17-.5))
(IFT,L30>0,(PAR,L30=(L8-L7*5.)*L17+.5))
```

This portion of the subroutine uses the FWR to preprocess the information. Note the use of "I" in this text and quotation marks.

```
(FWR,"G00","X",L22I,"Y",L23I,"F0")
(FWR,"Z",L29I)
(FWR,"Z",L30I)
(FWR,"G01G03G17G90","X",L24I,"Y",L23I,"Z",L26I,"R",L25I,"F",
                                                        L27I)
(LBL,1)(PAR,L15=L15+L14,L11=((L10*COS(L15)+L1)/L60+.5)
                                 *L19+.5,L13=L13+L7)
(IFT,L11<0,(PAR,L11=((L10*COS(L15)+L1)/L60-.5)*L19-.5))
(PAR,L12=((L10*SIN(L15)+L2)/L60+.5)*L19+.5,L21=(L13/L60
                                            -.5)*L19-.5)
(IFT,L12<0,(PAR,L12=((L10*SIN(L15)+L2)/L60-.5)*L19-.5))
(IFT,L21>0,(PAR,L21=(L13/L60+.5)*L19+.5))
(PAR,L18=(L10/L60+.5)*L19+.5,L10=L10+L9,L16=L16+1.)
(FWR,"G03","X",L11I,"Y",L12I,"Z",L21I,"R",L18I,"F",L6I)
(IFT,L16<10.,(GTO,1))
(PAR,L31=(L13+L7*5.)*L17-.5,L18=L18/2.+.5)
(IFT,L31>0,(PAR,L31=(L13+L7*5.)*L17+.5))
(FWR,"G03","X",L22I,"Y",L23I,"Z",L31I,"R",L18I,"F",L27I)
(FWR,"Z",L29I,"F0")
(FWR,"(END)")
(END)
M0
```

*** TEST SPIRAL ***

```
(PAR,L203=20.,L206=.1)
(OPF,THRD/NC)          !Open a NC file called THRD
(CLS,SPIRAL,L1=0,L2=0,L3=2.3662,L4=2.3716,L5=1.4778,
                                     L7=.087,L8=-.85)
(PAR,L1050=0)
(CLF)                  !Close the file.
(LBL,10)(IFT,L1050=0,(GTO,10))   !Delay created to allow for
                                  program processing.
(DLP,L1050)            !Parameter deleted.
(CLS,THRD)             !Call up the subroutine called THRD and
                        the above information will be executed.
```

Use of the FWR command creates the following:

```
    G00X0Y0F0
    Z.1000
    Z-.8935
    G01G03G17G90X.4442Y0Z-.8500R.2221F5.
    G03X.3596Y.2613Z-.8413R.4445F10.
    G03X.1374Y.4230Z-.8326R.4447F10.
    G03X-.1375Y.4232Z-.8239R.4450F10.
    G03X-.3602Y.2617Z-.8152R.4453F10.
    G03X-.4456Y0Z-.8065R.4456F10.
    G03X-.3607Y-.2620Z-.7978R.4458F10.
    G03X-.1378Y-.4243Z-.7891R.4461F10.
    G03X.1379Y-.4245Z-.7804R.4464F10.
    G03X.3613Y-.2625Z-.7717R.4466F10.
    G03X.4469Y0Z-.7630R.4469F10.
    G03XX0Y0Z-.7195R.2235F5.
    Z.1F0
M2
```

CUTTING A CYLINDRICAL DIE

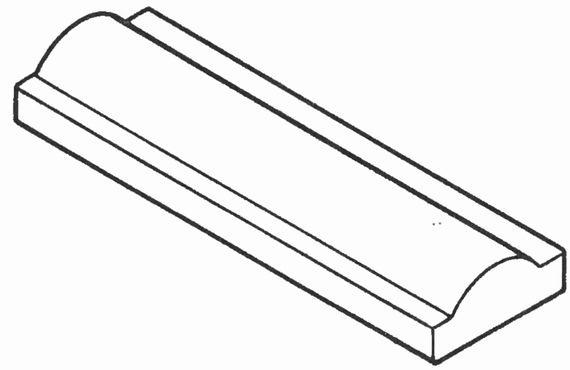


Fig. 9.11 DIE CYLINDER

Program name: (DFS, DIE CYLINDER)

L101 = Diameter of cylinder to be cut
L105 = Length of cylinder to be cut
L110 = Fix. offset no. where cyl. centerline is to start
L115 = Plane for cutting purposes (18[ZX] or 19[YZ])
L116 = Direction of cut (plus = 0, minus = 1)
L120 = Type of cut (1 = inside cyl. or 2 = outside cyl.)
L125 = Feedrate XXX.X IPM or XXXX. MMPM
L126 = Mode (70 = inch, 71 = metric)
L127 = Z positioning clearance
L130 = Diameter of ball-nosed end mill
L135 = Amount of cutting stepover

The programmer must construct the proper (PAR) statement for this information to be stored in the parameter table.

```
(DFS,CYL,SAVE)
(PAR,L10=2.,L11=3.,L41=0"CURRENT CUT LENGTH",
                          L31=L130/2.,L40=L101/2.-L31)
(IFT,L116=1.,(PAR,L10=3.,L11=2.))
G00G90G94GL1126
G48HL110
(IFT,L120=2.,(GTO,300))
X0Y0ZL127F0
```

```
Z-L31FL125
(IFT,L115=19.,(GTO,15))          !When cutting in G19 YZ plane
(MIR,X0Y0)
(IFT,L116=1.,(MIR,X0Y1))
X-L40
G18XL40Z-L31I0K-L31GL10
(GTO,90)
(LBL,15)(MIR,X0Y0)
(IFT,L116=1.,(MIR,X1Y0))
YL40
G19Y-L40Z-L31J0K-L31GL10
(LBL,90)(IFT,L105=0,(GTO,5000))
(LBL,100)(PAR,L41=L41+L135)      !Preceding cut plus stepover
(IFT,L41<L105,(GTO,110))         !When more cuts are required
(PAR,L41=L105)                   !Last cut
(LBL,110)(IFT,L120=2.,(GTO,320)) !For OD cuts
(IFT,L115=19.,(GTO,115))         !G19 cut
YL41
G18X-L40Z-L31I0K-L31GL11
(GTO,116)
(LBL,115)                        !G19 cut
XL41
G19YL40Z-L31J0K-L31GL11
(LBL,116)(IFT,L41=L105,(GTO,5000))   !End loop
(PAR,L41=L41+L135)               !Next cut increment
(IFT,L41<L105,(GTO,120))         !If more cuts are required
(PAR,L41=L105)                   !Last cut
(LBL,120)(IFT,L120=2.,(GTO,335)) !When cutting OD
(IFT,L115=19.,(GTO,125))         !G19 cut
YL41
G18XL40Z-L31I0K-L31GL10
(GTO,126)
(LBL,125)XL41
G19Y-L40Z-L31J0K-L31GL10
(LBL,126)(IFT,L41=L105,(GTO,5000))   !End loop
(GTO,100)                        !Restarts loop for more cuts
(LBL,300)(PAR,L40=L101/2.+L31)   !Radius of OD cutting CL
(IFT,L115=19.,(GTO,315))         !When cutting G19
(MIR,X0Y0)
(IFT,L116=1.,(MIR,X0Y1))
XL40Y0ZL1027F0
Z-L40FL125
G18X-L40Z-L40I0K-L40GL10
(GTO,90)
(LBL,315)(MIR,X0Y0)
(IFT,L116=1.,(MIR,X1Y0))
X0Y-L40ZL127
Z-L40FL125
G19YL40Z-L40J0K-L40GL10
(GTO,90)
(LBL,320)(IFT,L115=19.,(GTO,325))
YL41
G18XL40Z-L40I0K-L40GL11
(GTO,116)
(LBL,325)XL41
G19Y-L40Z-L40J0K-L40GL11
(GTO,116)
(LBL,335)(IFT,L115=19.,(GTO,340))
YL41
G18X-L40Z-L40I0K-L40GL10
(GTO,126)
(LBL,340)XL41
G19YL40Z-L40J0K-L40GL10
(GTO,126)
(LBL,5000)ZL1127F0
H0
(MIR,X0Y0)
(END)

M0          !Block delete off to load descriptions in parameter table
/(PAR,L101=0"CYL. DIA.",L105=0"TOTAL LENGTH OF CUT",
                                   L110=0"FIX-OFFSET NO.")
/(PAR,L115=0"CUTPLANE 18. OR 19.",L116=0"DIRECTION 0=
                                                   +1.=-")
/(PAR,L120=0"1.=I.D. 2.=O.D. CUT",L125=0"FEEDRATE",
                                L1126=0"70.=IN. 71.=MET.")
/(PAR,L127=0"Z CLEARANCE",L130=0"CUTTER-DIA.",
                                   L135=0"CUT STEPOVER")
```

DEMONSTRATION CYL

```
N1S1000M3
N2(PAR,L101=2.,L105=1.5,L110=1.,L115=18.,L120=1.,
                                   L125=100.,L126=70.)
N3(PAR,L127=.1,L130=.5,L135=.05,L116=0)
N4(CLS,CYL)
N5(PAR,L101=2.25,L105=.5,L110=2.,L135=.05)
N6(CLS,CYL)
N7(PAR,L101=1.75,L105=1.5,L110=3.,L135=.05)
N8(CLS,CYL)
N9(PAR,L101=1.,L105=1.75,L110=4.,L135=.05)
N10(CLS,CYL)
N11H0
N12M30
```

PROBE TEST FOR VARIOUS SIZE PARTS

Program name:MULTIBORES8A1

(DFS,5.I.D.HOUSING,SAVE)

NC program blocks (subroutine to create 5" ID bore)

(END) !End subroutine

(DFS,7.I.D.HOUSING,SAVE)

NC program blocks (subroutine to create 7" ID bore)

(END) !End subroutine

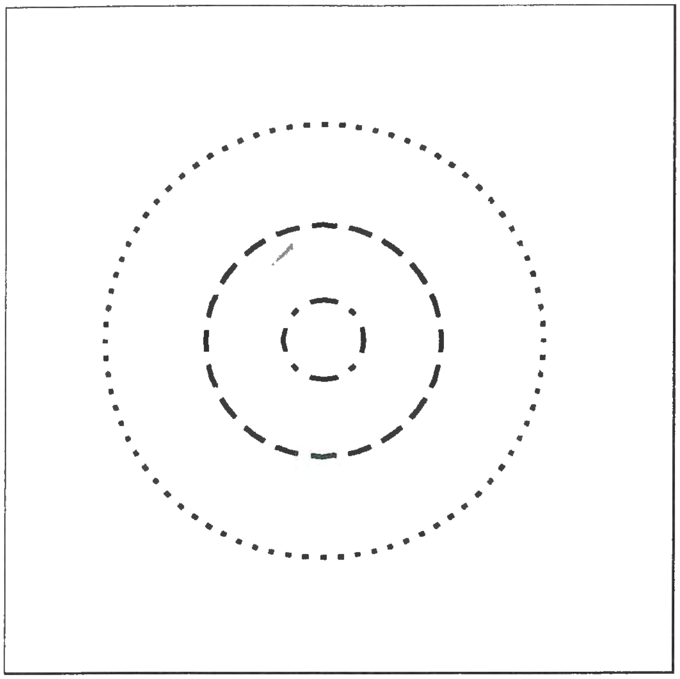


Fig. 9.12 VARIOUS SIZE BORES

(DFS,10.I.D.HOUSING,SAVE)

NC program blocks (subroutine to create 10" ID bore)

(END) !End subroutine

DEMONSTRATION PROGRAM

```
N1G90G94G00G70
N2Z20.M25G28
N3T77777M6        !Spindle Probe
N4B0F0
N5X21.Y20.
N6Z7.75           !Position Probe -.25 past Z surface
N7G27X27.         !Probe bore for identification
N8G27Y26.
N9G27X27.
```

The G27 cycle will automatically assign the surface separation to parameter L1005.

```
N10(IFT,L1005<6.,(GTO,10))     !5. ID Housing
N11(IFT,L1005<8.,(GTO,20))     !7. ID Housing
N12(GTO,30)                    !10. ID Housing
N13(LBL,10)(CLS,5.I.D.HOUSING)
(GTO,100)
N14(LBL,20)(CLS,7.I.D.HOUSING)
(GTO,100)
N15(LBL,30)(CLS,10.I.D.HOUSING)
N16(LBL,100)M30
```

USING THE SPINDLE PROBE TO ESTABLISH FIXTURE OFFSET

Program name:FIXOFF PROBE

"FIXOFF-PROBE" was developed to establish the X, Y and B positions of a workpiece loaded onto a machining center. After fixture offset values are established, subroutine "FIX" may then be used to recalculate values for subsequent B-axis positions.

The following predefined subroutines are used:
"FINDZ"
"FINDBORE"
See the Table of Contents to find these subroutines.

DEMONSTRATION PROGRAM

```
N1Z20.M25
N2X30.Y11.5M25
N3T77777M19M6                  !Spindle probe
N4S40M3G90G94G70G0
N5G04F5.            !Rotate spindle to seat bearings
N6M5
N7H0G99B180.F0  !B axis bore and Z surface position
N8(ZTB,H)
N9G28X19.Y17.14F0              !Position to probe Z
N10(CLS,FINDZ,L5=-20.,L3=0)  !L5=Z past
                                surface, L3=Z
                                program position
N11G48H1                       !Activate Z offset
N12F0G28X17.17Y17.14           !Position probe to bore
N13F0Z-.4                      !Position probe in bore
N14(CLS,FINDBORE,L1=17.17,L2=17.14,L4=180.)
          !L1=X,L2=Y,L4=B programmed positions
N15Z20.M25                     !Retract to clearance
N16X30.Y11.5M25
N17H0B0F0
N18M30
```

PROBE PART FOR IDENTIFICATION AND EXECUTE PART PROGRAMS

Program name:RUNWHICHPART

The G804 is a user defined cycle which is the same as the G26X command. However, it does not have the commands which produce an error message if there is non-contact with the part. It does not do a calculation for one-half the ball radius in determining the axis position. If a feedrate is not programmed, the cycle will be executed at 250 IPM.

For definition of the parameters see the description of the G26 cycle in Appendix E, Appendix B for the probe

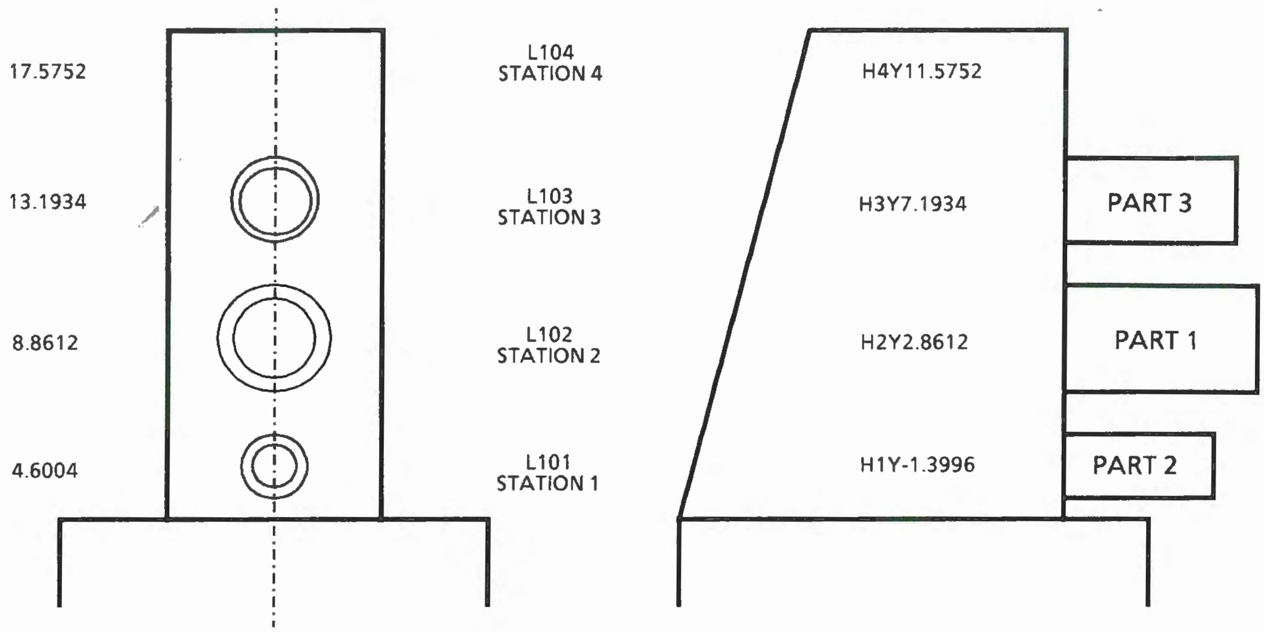


Fig. 9.13 FIXTURE PROFILE

cycle parameters and Appendix C for system parameters.

```
(DFC,G804,SAVE)
(CER,MAS)
(IFT,NOT(DEF(L1003)),(PAR,L1003=0))
(IFT,G71 AND (L3033<114.483),G800M0M99) !Error - Illegal tool
                                         comp
(IFT,G70 AND (L3033<4.5072),G800M0M99)  !Error - Illegal tool
                                         comp
(PAR,L843=91.)
(IFT,G90,(PAR,L843=90.))
(IFT,G71,(GTO,50))
(PAR,L833=.040,L834=1.8,L844=250.)
(GTO,60)
(LBL,50) (PAR,L833=1.016,L834=45.72,L844=6350.)
(LBL,60) (IFT,DEF(L806),(GTO,65))
(PAR,L806=L844)
(LBL,65)(IFT,L806>L844,(PAR,L806=L844))
(PAR,L831=0.)
(IFT,DEF(L824),(PAR,L831=L831+8.))
(IFT,L831=8.,(GTO,400))
G800 GL843 M0 M99          ! Error - Unrecognized format for G804
(LBL,400) (IFT,L843=91.,(GTO,405))
(PAR,L824=L824-X)
(LBL,405) (IFT,L824>=0.,(GTO,410))
(PAR,L832=180.,L835=L833,L839=-L1003)
(GTO,420)
(LBL,410) (PAR,L832=0.,L835=-L833,L839=L1003)
(LBL,420) M19 SL832
(PAR,L1020=0.)
G91 G22 XL824 FL806
(LBL,430) (IFT,L1020=0.,(GTO,430))
(IFT,L1021=1.,G800GL831M99M0)  !Error - Probe was in contact
                               at start of G22
GL843G800
(END)

(DFS,FINDWHICHPART,SAVE)       !Probe
(PAR,L1=7.54,L2=7.2,L3=1.,L100=0)
(LBL,10)HL200G48
X8.Y6.G28ZL1F0
ZL2
G29
G804X15.
X8.F0
(IFT,L1021=8.,(GTO,100))       !Probe made contact
(PAR,L3=L3+1.)
(IFT,L3=2.,(PAR,L1=4.6,L2=4.25))
(IFT,L3=3.,(PAR,L1=4.3,L2=3.95))
(IFT,L3=4.,(GTO,100))          !No parts loaded
(GTO,10)
(LBL,100)(PAR,L100=L3)         !L100=Part number
(END)

(DFS,TOOLCHANGE,SAVE)
L1=PART NUMBER.
L2=TOOL NUMBER.
L3=SPINDLE SPEED.
L4=NEXT TOOL NUMBER
```

The programmer must create the proper (PAR) statements to include the above information.

```
(PAR,L105=1.)
(IFT,(L101=L1)OR(L102=L1),(GTO,10))
(IFT,(L103=L1)OR(L104=L1),(GTO,10))
(PAR,L105=0)!Used in BOREALL cycle to establish no parts loaded
(GTO,100)      !No tool change - part not loaded
(LBL,10)Z20.M25
M25X30.Y11.5F0
TL2M6M19
(IFT,L4>0,TL4)
SL3M3M8
(LBL,100)(END)
```

```
(DFS,MILLPART1,SAVE)   !6. dia. face mill, 6. gage length
HL200G48
ZL150F0                !L150=Z clearance plane all parts
X9.Y6.
Z13.14
X21.F10.
Z13.1F0
X9.F15.
(END)
```

```
(DFS,MILLPART2,SAVE)   !6.dia. face mill , 6. gage length
HL200G48
ZL150F0
X10.Y6.
Z10.4
X20.F14.
(END)
```

```
(DFS,MILLPART3,SAVE)   !6.dia. face mill ,6. gage length
HL200G48
ZL150F0
X13.Y6.
Z10.17
X17.F15.0
(END)
```

```
(DFS,MILLALL,SAVE)
(PAR,L150=13.2)        !L150=Z clearance plane all parts
(IFT,L105=4.,(GTO,100))  !No Parts loaded
(IFT,L105=1.,(GTO,10))   !Part 1 loaded
(IFT,L105=2.,(GTO,20))   !Part 2 loaded
(IFT,L105=3.,(GTO,30))   !Part 3 loaded
(LBL,10)(CLS,MILLPART1)
(GTO,100)
(LBL,20)(CLS,MILLPART2)
(GTO,100)
(LBL,30)(CLS,MILLPART3)
(LBL,100)(END)
```

```
(DFS,BORES,SAVE)
HL200G48
ZL150F0
X15.Y6.
ZL300
ZL400FL500
ZL300F0
(END)
```

```
(DFS,BOREALL,SAVE)
(IFT,L105=0,(GTO,100))     !This part is not loaded in any station
(IFT,L101=L1,(GTO,10))     !L1=Part number
(GTO,15)
(LBL,10)(CLS,BORES,L200=1.)     !Bore part - Station 1
(LBL,15)(IFT,L102=L1,(GTO,20))
(GTO,25)
(LBL,20)(CLS,BORES,L200=2.)     !Bore part - Station 2
(LBL,25)(IFT,L103=L1,(GTO,30))
(GTO,35)
(LBL,30)(CLS,BORES,L200=3.)     !Bore part - Station 3
(LBL,35)(IFT,L104=L1,(GTO,40))
(GTO,100)
(LBL,40)(CLS,BORES,L200=4.)     !Bore part - Station 4
(LBL,100)(END)
```

DEMONSTRATION RUNWHICHPART

```
N1G90G94G70G00
N2(FXC,H1Y-1.3996,H2Y2.8612,H3Y7.1934,H4Y11.5752)
                    !Establish station locations thru fixture offsets
N3Z20.M25
N4M25X30.Y11.5F0
N5B180.
N6M19T77777M6      !Spindle probe
N7       !L200=Fixture offset number = Station number
N8(CLS,FINDWHICHPART,L200=1.)    !Station 1
N9(PAR,L101=L100"STATION 1")     !L100=Part number
N10(CLS,FINDWHICHPART,L200=2.)   !Station 2
N11(PAR,L102=L100"STATION 2")    !L100=Part number
N12(CLS,FINDWHICHPART,L200=3.)   !Station 3
N13(PAR,L103=L100"STATION 3")    !L100=Part number
N14(CLS,FINDWHICHPART,L200=4.)   !Station 4
N15(PAR,L104=L100"STATION 4")    !L100=Part number
N16Z20.M25
N17M25X30.Y11.5F0
N18M19T202M6       !6. dia. face mill, 6. tool length
N19S500M3
N20(CLS,MILLALL,L105=L101,L200=1.)!Fin. mill part - station 1
N21(CLS,MILLALL,L105=L102,L200=2.)!Fin. mill part - station 2
N22(CLS,MILLALL,L105=L103,L200=3.)!Fin. mill part - station 3
N23(CLS,MILLALL,L105=L104,L200=4.)!Fin. mill part - station 4
N24(CLS,TOOLCHANGE,L1=2.,L2=55555.,L3=900.,L4=104.)
                    !T2 1.0" diameter boring bar 7.0" length
```

```
N25 !************
N26            !L1=PART NO., L2=TOOL NO., L3=SPINDLE
               SPEED, L4=NEXT TOOL (READY POSITION)
N27 !************
```

The programmer must construct the proper (PAR) statement for the above information.

```
N28(CLS,BOREALL,L1=2.,L150=14.2,L300=11.5,L400=10.,
                                    L500=7.)    !Fin. bore part 2
N29 !************
N30            !L1=PART NO., L150=Z CLEAR. ALL,
               L300=PART Z CLEAR., L400=Z CUT DEPTH
N31            !L500=CUT FEED RATE.
N32 !************
N33(CLS,TOOLCHANGE,L1=1.,L2=104.,L3=300.,L4=100.)
                         !T3 3.0" diameter boring bar,7. 0" length
N34(CLS,BOREALL,L1=1.,L150=14.2,L300=14.2,L400=12.2,
                                    L500=3.)    !Fin. bore part 1
N35(CLS,TOOLCHANGE,L1=3.,L2=100.,L3=600.,L4=0)
                         !T4 2.0" diameter boring bar , 8. 0" length
N36(CLS,BOREALL,L1=3.,L150=15.2,L300=12.27,L400=11.,
                                    L500=5.)    !Fin. bore part 3
N37Z20.M25
N38X30.Y11.5M25
N39B0H0
N40(ZTB,H)             !Zero out Fixture Offset Table
N41M30
```

SAVE, LOAD AND DELETE PROBE GENERATED PARAMETERS

Program name: SAVE LOAD DELETE PAR

SUBROUTINE PROBE-PAR

PROBE-PAR was developed to save current probe post and ball radii parameters for each pallet in a multi-pallet system. Each pallet has a probe bushing in the fixture. The bushings are located at different places on the fixture. The bushings are to be used for probe cycle G25. Auto-regrid will be commanded every time a pallet is shuttled onto the table. The first two blocks of the program will load probe parameters related to the current pallet on the table.

Example:

```
N1(PAR,L1=L3020)
N2(CLS,PROBE PARAMETERS-PALLET<L1I>)
```

```
(DFS,PROBE-PAR,SAVE)
(PAR,L1=L3020)                    !L1=Current pallet no.
(OPF,PROBE PARAMETERS-PALLET<L1I>/NC)
                                  !Create NC file subroutine
(PAR,L1=L3)                       !L3=Save lowest PAR no.
(LBL,1)(IFT,NOTDEF(L(L1)),(GTO,10))
(PAR,L2=L(L1))                    !L2=value in PAR L1
(FWR,"(PAR,L",L1I,"=",L2,")")     !Write to file (PAR,Lno.=value)
(LBL,10)(PAR,L1=L1+1.)            !Increase PAR no. by 1
(IFT,L1<=L4,(GTO,1))              !L4=Save highest PAR no.
(FWR,"(END)")
(CLF)
(END)
```

SUBROUTINE DELETE-PAR

DELETE-PAR was developed to delete a range of parameters, by the part program.

```
(DFS,DELETE-PAR,SAVE)
(LBL,1)(IFT,DEF(L(L1)),(DLP,L(L1))) !If defined, delete par.
(PAR,L1=L1+1.)
(IFT,L1<=L2,(GTO,1))
(END)
M0
```

DEMONSTRATION SAVING PARAMETERS

```
N1(CLS,PROBE-PAR,L3=1000.,L4=1006.)    !Saves par. range
M0                                      L1000-L1006
```

DEMONSTRATION DELETE PARAMETERS

```
N2(CLS,DELETE-PAR,L1=1000.,L2=1006.)   !Delete par. range
M0                                      L1000-L1006
```

DEMONSTRATION LOADING PARAMETERS

```
N3(PAR,L1=L3020)
N4(CLS,PROBE PARAMETERS-PALLET<L1I>)
N5M2
```

SELECT PROPER TOOL BASED UPON SPINDLE PROBE RESULTS EXAMPLE

Program name: STARTWHICHTOOL

```
(DFS,TOOL)
G90G94G70G00
Z20.M25
M19TL1M6          !L1=Tool number
SL2M3M8           !L2=Spindle speed
X17.17Y17.14F0
Z,7.7+L4,F0       !L4=Tool length, 7.7=Prog. clearance
```

The programmer must create the proper (PAR) statements to load the required values into the parameter table.

```
Z,1.5+L4,FL3        !L3=Feedrate, 1.5=Prog. depth
Z,7.7+L4,F0
(END)
```

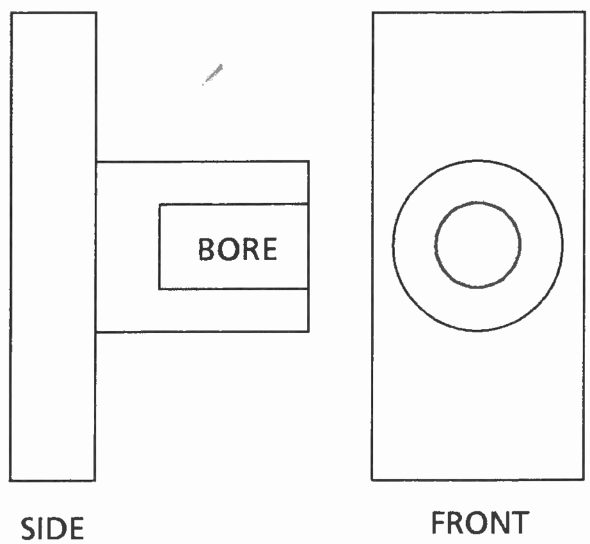


Fig. 9.14 STARTWHICHTOOL FIXTURING

DEMONSTRATION STARTWHICHTOOL

```
N1G90G94G00G70
N2Z20.M25
N3X30.Y11.5F0
N4M19T77777M6          !Spindle probe
N5(IFT,NOT(DEF(L1003))OR(L1003<=.001),M0M99)
                       !Error - No ball radius
N6B180.F0
N7X17.17Y17.14G28
N8Z7.0                 !Position probe past Z surface
N9G29
N10G27X20.             !Probe bore for size
N11G27Y20.
N12G27X20.
N13Z20.M25
N14X30.Y11.5F0
                       !L1005=Probed bore size
N15(IFT,L1005<1.-.03,(GTO,100))    !Bore 1.0 dia.
N16(IFT,L1005<1.5-.03,(GTO,200))   !Bore 1.5 dia.
N17(GTO,300)                       !Bore 2.0 dia.
N18(LBL,100)(CLS,TOOL,L1=55555.,L2=800.,L3=4.,L4=10.)
                                   !1. dia. boring bar - rough bore
N19(LBL,200)(CLS,TOOL,L1=77775.,L2=600.,L3=3.,L4=11.)
                                   !1.5 dia. boring bar - semi bore
N20(LBL,300)(CLS,TOOL,L1=202.,L2=400.,L3=2.,L4=9.)
                                   !2. dia. boring bar - fin. bore
N21Z20.M25
N22X30.Y11.5B0M25
N23M3
```

EXECUTE TOOL CHANGE BASED ON FEED AND SPEED CRITERIA

Program name: TEST TLCHG

This program is one method of programming a tool change macro. The program was developed to utilize Surface Feet per Minute and Feed per Revolution to determine feed and speed.

```
(DFS,TLCHG,SAVE)                    ! Tool change macro
M9                                  !Coolant off
Z20.M25M19                          !Position Z for tool change-keylock
TL1M6                               !Tool change
(IFT,L2>0,TL2)                      !Select ready tool if L2 > 0
(PAR,L3=INT((L4*12.)/(L3070*L5)+.5))  !L3=RPM-Note:
                                      L3070=PI(3.1415927)
(PAR,L100=INT(L3*L6*10.+.5)/10.)      !L100=Feedrate-IPM,
                                      3.1 format
SL3M3M8                             !Spindle on CW-Coolant on - flood
G90G94G00G70                        !Set state of control
(END)
```

DEMONSTRATION TLCHG

L1 = Spindle tool
L2 = Ready tool
L4 = SFM
L5 = Tool diameter
L6 = IPR

The programmer must create the proper (PAR) statements to load the required values into the parameter table.

```
N1(CLS,TLCHG,L1=55555.,L2=202.,L4=200.,L5=6.,L6=.01*8.)
                                 !6.0" dia. face mill-8 teeth
N2B0X5.Y3.F0                     !Position to mill, at rapid
N3Z15.
N4X15.FL100                      !Mill at feedrate
N5Y6.
N6X5.
N7(CLS,TLCHG,L1=202.,L2=0,L4=100.,L5=.75,L6=.005)
                                               !.75" dia. drill
N8G81X10.Y4.5Z12.R15.1FL100 !Drill point 1
N9X12.                           !Drill point 2
N10G80                           !Cancel drilling cycle
N11Z20.M25                       !Position for shuttle
N12X30.Y11.5B0M25                !Position for shuttle
N13M30                           !Shuttle
```

UPDATE TOOL COMPENSATIONS THRU PART SURFACE PROBING EXAMPLE

Program name:UPDATECOMPS

DEMONSTRATION UPDATECOMPS

```
N1G90G94G70G00B180.F0
N2(PAR,L10=1.156"PRG. SLOT DEPTH",L15=.75
                                        "PRG. SLOT WIDTH")
N3(LBL,10)Z20.M25
N4F0X30.Y11.5
N5M19T202M6                 !T202 .5 End mill 6. Gage length
                            finish mill .75 slot
N6(PAR,L19=L3018"LAST D-COMP.",L14=L3033
                                        "LAST P-COMP.")
N7T77777X26.Y6.975F0S900M3
N8Z8.3533
N9F100.X25.2G41
N10X5.
N11Y6.725
N12X25.2
N13X26.G40
N14Z20.M25
N15F0X30.Y11.5
N16T77777M6M19              !Spindle probe
N17G90G94G00G70
N18G28X14.8Y4.57F0
N19G29
N20Z0G803                   !Establish Z reference
N21(PAR,L1=Z"Z REF. COORDINATE")
N22Z4.25F0
N23G27X20.
N24G27Y20.
N25G27X20.R.8075            !Establish X probe ball radius
N26G27Y31.R.8075            !Establish probe ball radius
N27Z4.7G28F0
N28Y6.85
N29G29
N30Z0G803                   !Probe slot bottom
N31(PAR,L2=Z"Z SLOT COORDINATE")
N32(PAR,L11=L1-L2"SLOT DEPTH")
N33(PAR,L12=L11-L10"DEPTH DEVIATION",L13=L14+L12
                                        "NEW P-COMP.")
N34Z3.3F0
N35G27Y20.
N36G27Y20.
N37(PAR,L17=L1005-L15"WIDTH DEVIATION",L18=L19+L17
                                        "NEW D-COMP.")
N38(PAR,L16=L1005"SLOT WIDTH")
N39(TLD,T202DL18PL13)       !Re-establish T202 dia. and
                            length comps.
N40Z20.M25
N41(IFT,(L12<-.0015)OR(L17<-.0015),(GTO,10))
                            !Recut with updated comps
N42(IFT,L12>.0015,M99M0)    !Slot too deep
                  !See PAR L11 Press [CYCLE START] to continue.
N43(IFT,L17>.0015,M99M0)    !Slot oversize
                  !See PAR L16 Press [CYCLE START] to continue.
```

If part program is to be aborted if slot is too deep or width is oversize, see N44 below.

```
N44(IFT,(L12>.0015)OR(L17>.0015),(GTO,100))   !Abort program
N45F0X30.Y11.5                                !Continue program
N46T55555M19M6
N47(LBL,100)Z20.M25
N48X30.Y11.5M25
N49M30
```

USER GENERATED OPERATION USING A G22 COMMAND

G22 is a non-modal function that records all axis positions when the probe makes contact. The axes positions are recorded in system parameters L3042-49. All axis motion stops when the contact is made, (remaining block motion is skipped) and the next program block is processed. Neither axis motion or position recording occurs if the probe is in the contact state with a surface when the block containing the G22 goes active.

Probe status information is kept in parameter L1021 giving the results of the probing action. L1021 is set to 8.000000 before the G22 is commanded. This is required so that the block "look ahead" feature will NOT report lack of contact errors that can be generated by an incomplete probe status

Probe Status Information

Parameter Value	Status Information
L1021 = 8.0000000	contact made- axis motion halted, positions recorded
L1021 = 2.0000000	no contact made, axis motion completed, no positions recorded
L1021 = 1.0000000	contact made at the start of the cycle, no axis motion occurred, no positions recorded

Refer to parameters L3042-49 in appendix C for make / break positions for the axis

Example:

N5X10.Y10.Z12.31F100. !Move to starting position

N6(OPF, PUMP234) !Open a file called PUMP234 for probe results

N10(PAR,L1021=8.)

!Set L1021=8.0" prepare for probing operation

N20G22X11.F20. !Move to X11.

N30(IFT,L1021=8.,(GTO,3)) !If contact is made go to label 3 and process contact point

N40(IFT,L1021=1., M99M0) !If in contact at the start of the cycle, cycle stop and sound alarm.

N50(IFT,L1021=2., M99M0) !If in contact was not made cycle stop and sound the alarm.

N60(LBL,3)(PAR,L100=L3042-10.) !Subtract 10.0" from the contact position L3042 and record the value in the parameter table under the parameter L100.

For further explanation, the value of 10.0" might represent the distance from some absolute position and the difference might be some dimension required on the part drawing. A further application might be a comparison with a tolerance.

N70(IFT,L100>=10.55,(GTO,4)) !If this test is false the next program block will be executed.

N80 *Additional program blocks*

N90(LBL,4)(FWR,"TOLERANCE EXCEEDS SPEC. REWORK",L100) !This message would be included in the inspection report if the part exceeded the spec. The inspection program would not be interrupted if the part exceeded the spec., and the amount held in parameter L100 would be printed in the report.

Additional program blocks

(CLF) This command closes the file

G23 is a non-modal function that records all axis positions when the probe loses contact. All axis motion stops when the contact is lost, (remaining block motion is skipped) and the next program block is processed. Neither axis motion or position recording occurs if the probe is in the contact state with a surface when the block containing the G22 goes active.

USING A G62 TO COUNT HOLES

This example uses a G62 to count the number of holes in a part. It is assumed that if the probe does not make contact with the part at the commanded coordinate a hole exits at that point. Additionally, this program uses a user defined G826 cycle which is similar to the G26Z command but will not go into a stop condition if the probe does not make contact with the part surface.

N10T77777M6 !Load the probe into the spindle

N20G62 !Set L1022 to zero by commanding G62

N30X2.Y1.Z.1F100. !Position the probe

N40G826Z-.5 !Detect the surface hole 1

N50X3.Y1.Z.1 !Position the probe

N60G826Z-.5 !Detect the surface hole 2

N70X3.Y2.Z.1 !Position the probe

N80G826Z-.5 !Detect the surface hole 3

N90(IFT,L1022<3.,M0M99) !Missing a hole if the number of contacts is less than 3 the program will stop and the alarm will sound.

N100M2

APPENDIX A

MATHEMATICAL APPLICATIONS

TRUNCATION OF PARAMETERS

Truncating to a number of decimal places can be accomplished as shown in the examples below.

If parameter L1 has a value of 1.2345

1. Then the expression (PAR,L2=INT(L1)) will result in 1.0000 in parameter L2.
2. And the expression (PAR,L2=INT(L1*10.)/10.) will result in 1.2000 in parameter L2.
3. And the expression (PAR,L2=INT(L1*100.)/100.) will result in 1.2300 in parameter L2.
4. And the expression (PAR,L2=INT(L1*1000.)/1000.) will result in 1.2340 in parameter L2.

To truncate at additional places, apply the appropriate power of 10 as in the previous examples.

NOTE

The parameter display rounds to the eighth digit so the values on the display can sometimes be misleading. If a parameter has a value of 0.123456789 the value will show up as 0 .12345679. If a parameter has a value of 0.123449995 through 0.123449999 it will show up as 0.12345000 etc. Also, certain mathematical functions and certain repetitive operations (due to the nature of computers representing numbers) can result in results that are less than perfect (1.4999999 instead of 1.5). For these reasons it is recommended that close attention be paid to rounding and truncating at appropriate times. Rounding and truncating insure a value is as displayed.

ROUNDING OF PARAMETERS

Rounding can be accomplished using the "INT" function by taking the integer of an original number + .5 .

Examples:

1. If parameter L1 has a value from 1.5000 to 1.9999
   Then the expression (PAR,L2=INT(L1+.5)) will yield 2.0000 in parameter L2.

2. If parameter L1 has a value from 1.0000 to 1.4999
   Then the expression (PAR,L2=INT(L1+.5)) will yield 1.0000 in parameter L2.

Rounding can also be done to a number of decimal places.

1. If L1 has a value from 1.1500 to 1.1999
   Then the expression (PAR,L2=INT(L1*10.+.5)/10.) will result in 1.2000 in parameter L2.

2. If L1 has a value from 1.2250 to 1.2290
   The expression (PAR,L2=INT(L1*100.+.5)/100.) will result in 1.2300 in parameter L2.

3. If L1 has a value from 1.2335 to 1.2339
   The expression (PAR,L2=INT(L1*1000.+.5)/1000.) will result in 1.2340 in parameter L2.

To round at additional places, apply the appropriate power of 10 as in the previous examples.

NOTE

The parameter display rounds to the eighth digit so the values on the display can sometimes be misleading. If a parameter has a value of 0.123456789 the value will show up as .12345679. If a parameter has a value of 0.123449995 through 0.123449999 it will show up as .12345000 etc. Also, certain mathematical functions and certain repetitive operations (due to the nature of computers representing numbers) can result in results that are less than perfect (1.4999999 instead of 1.5). For these reasons it is recommended that close attention be paid to rounding and truncating at appropriate times. Rounding and truncating insure a value is as displayed.

ARITHMETIC FUNCTIONS

The functions SQ, SQRT, ABS, SIN, COS, TAN, ASN, ACS, ATN, INT, FRA, LOG, ALOG, LN, and EXP will be used in parameter assignments and IF expressions.

Decimal points must be specified by the programmer.

SQ - Returns the square of the argument.

SQ(SQn)   -10000.0000 < n < 10000.0000 if numeric constant

SQ(10.) will be 100.00000
SQ(100.) will be 10000.000

SQRT - Returns the square root of the argument.

SQRT(n)   0 <= n <= 99999999.

SQRT(25.) will be 5.0000000
SQRT(2500.) will be 50.000000

Error message: "Error: negative square root attempted"

ABS - Returns the absolute value of the argument.

ABS(n)   -99999999. <= n <= 99999999.

ABS(-10.) will be 10.000000

SIN (sine in degrees)

SIN(n)   -99999999. <= n <= 99999999.

SIN(25.) will be the SIN of 25.000000 degrees

COS (cosine in degrees)

COS(n)   -99999999. <= n <= 99999999.

COS(25.) will be the COS of 25.000000 degrees

TAN (tangent in degrees)

TAN(n) -99999999. < = n < = 99999999.

TAN(25.) will be the TAN of 25.000000 degrees
TAN(90.*n) where n = + or - odd integer will result in a internal overflow error.

ASN (arc sine in degrees) - Returns the angle in degrees whose sine is the argument.

ASN(n) -1.0000000 < = n < = 1.0000000

ASN(25.) will be the ARC SINE (in degrees) of 25.000000

ASN(1.) = 90.000000 Degrees
ASN(-1.) = -90.000000 Degrees

Error message: "ASN argument error: ASN(n) -1.0000 < = n < = 1.0000"

ACS (arc cosine in degrees) - Returns the angle in degrees whose cosine is the argument.

ACS(n) -1.0000000 < = n < = 1.0000000

ACS(25.) will be the ARC COSINE (in degrees) of 25.000000

ACS(1.) = 0.0 Degrees
ACS(-1.) = 180.00000 Degrees

Error message: "ACS argument error: ACS(n) -1.0000 < = n < = 1.0000"

ATN (arc tangent in degrees) - Returns the angle in degrees whose tangent is the argument.

ATN(n) -99999999. < = n < = 99999999.
ATN(25.) will be the arc tangent (in degrees) of 25.000000

ATN(-99999999.) = -89.99...
ATN(99999999.) = 89.99...

INT (integer) - Returns the portion of the argument that is to the left of the decimal point.

INT(n) -99999999. < = n < = 99999999.

INT(23.01) = 23.000000
INT(.25) = 0.0

INT(-9999.9999) = -9999.0000
INT(9999.9999) = 9999.0000

FRA (fraction) - Returns the portion of the argument that is to the right of the decimal point.

FRA(n) -99999999. < = n < = 99999999.

FRA(23.01) = .01000000
FRA(1.25) = .25000000

FRA(-9999.9999) = -.99990000
FRA(9999.9999) = .99990000

LOG (base 10 logarithm) - Returns the base 10 logarithm of the argument.

LOG(n) .00000001 <= n <= 99999999.

LOG(45.6845) = 1.6597688
LOG(9090.5432) = 3.9585898

LOG(.0001) = -4.0000000
LOG(99999999.) = 7.9999999

ALOG (base 10 antilogarithm) - Returns the base 10 antilogarithm of the argument.

ALOG(n) -7.9999999 <= n <= 7.9999999

ALOG(3.01) = 1023.2930
ALOG(1.25) = 17.782794
ALOG(-.5) = .31622776
ALOG(-2.5) = .00316227

ALOG(-3.9999) = .00010002
ALOG(3.9999) = 9997.6977

LN (natural base logarithm) - Returns the natural base logarithm of the argument.

LN(n) .00000001 <= n <= 99999999.

LN(23.01) = 3.1359289
LN(1.25) = .22314355

LN(.0001) = -9.2103403
LN(9999.9999) = 9.2103403

EXP (natural base antilogarithm) - Returns the natural base antilogarithm of the argument.

EXP(n) -18.420680 <= n <= 18.420680

EXP(3.89) = 48.910886
EXP(-9.) = .00012340
EXP(6.5) = 665.14163

EXP(9.2103) = 9999.5963

APPENDIX B

PARAMETER DEFINITIONS for PMI, PROBE, FILE and TOOL GAGE

PMI / PROBE PARAMETER DEFINITIONS

L1000 X axis post position (inch/metric)
L1001 Y axis post position (inch/metric)
L1002 Z axis post position (inch/metric)

L1003 X axis effective spindle probe ball radius (inch/metric)
L1004 Y axis effective spindle probe ball radius (inch/metric)
L1005 Surface Separation
L1006 Post Bushing Radius

L1020 Probe completion status
0.0 = G22 or G23 not complete
<>0 = G22 or G23 completed

L1021 Probe status
1.0000000 = Probe was in "contact" state at beginning of G22 in "non-contact" state at beginning of G23
2.0000000 = Probe"contact" did not occur during G22 or "non-contact" did not occur during G23.
8.0000000 = Probe "contact" occurred during G22 or "non-contact" occurred during G23

L1022 Number of probe contacts or non-contacts since last G61 (for contacts) or G62 (for non-contacts).

NOTE

Parameters marked (inch/metric) are converted to inch or metric values at the time of assignment only, depending on the modal state of the control; G70 (inch) or G71 (metric).

FILE STATUS READ / WRITE PARAMETERS

L1050 Status Parameter for (CLF,...) command - close file
L1051 Status Parameter for (DLF,...) command - delete file
L1052 Status Parameter for (PRT,...) command - print file
L1053 Status Parameter for (CPY,...) command - copy file
L1054 Status Parameter for (RCP,...) command - remote copy file

NOTE

These above parameters can be used optionally with the above commands to allow the part program to wait for any of these functions to be completed. The above parameters will be set to a value of 1.0000000 when the respective function completes if the parameter was previously defined. If the parameter was not previously defined, then no action is taken.

Example:

```
N50(PAR,L1050=0)                          ! Set close file status parameter to open
N100(OPF,CREATE SUB FILE)
N200(FWR,....................)
N300(FWR,....................)
:    :    :    :
:    :    :    :
N400(FWR,"(END)")
N500(CLF)
N600(LBL,10)(IFT,L1050=0,(GTO,10))        ! Wait until sub file is closed
N700(CLS,CREATE SUB FILE)                 ! Call the subroutine that was just created
N800 ......
N900 ......
```

TOOL GAGE PARAMETERS

L101 = Spindle Speed Initial Contact
L102 = Spindle Direction (Normally M4)
L103 = Expected Tool Length Programming Preset Lengths
L104 = Test Bar Diameter
L104 = Plus or Minus Variation for Expected Tool Length
L105 = Test Bar Diameter
L105 = Maximum Programmed Face Runout
L105 = Maximum Programmed O.D. Face Runout
L106 = Tool Diameter
L107 = Number of Teeth in Cutter
L108 = Programmed Preload On Sensor - Optional
L109 = G72 Contacting Feedrate
L110 = Programmed Allowable Length Wear
L110 = Programmed Allowable Diameter Wear

L120 = Number of Teeth Measured by Length Sensor
L121 = Low Tooth Measured by Length Sensor
L122 = High Tooth Measured by Length Sensor
L123 = Amount of Runout Measured by Length Sensor
L124 = Tool Length Measured by Length Sensor
L125 = Tool Length Compensation

L130 = Number of Teeth Measured by Diameter Sensor
L131 = Low Tooth Measured by Diameter Sensor
L132 = High Tooth Measured by Diameter Sensor
L133 = Amount of Runout Measured by Diameter Sensor
L134 = Tool Diameter Measured by Diameter Sensor
L135 = Tool Diameter Compensation

TOOL GAGE READ / WRITE PARAMETERS

L1101 = (MTD) completion status
non-zero = not complete,
zero = complete
L1102 = Start position of length sensor in Z
L1103 = Start position of diameter sensor in Y
L1104 = Preload dimension for length sensor
L1105 = Preload dimension for diameter sensor
L1106 = Sensing range of the length sensor
L1107 = Sensing range of the diameter sensor
L1108 = Center of target area for length sensor in X
L1109 = Center of target area for length sensor in Y
L1110 = Center of target area for diameter sensor in X
L1111 = Center of target area for diameter sensor in Z

APPENDIX C

SYSTEM PARAMETER DEFINITIONS

AXIS ABSOLUTE POSITIONS
(as seen on offset display)

L3000 First axis absolute position (normally X)
L3001 Second axis absolute position (normally Y)
L3002 Third axis absolute position (normally Z)
L3003 Fourth axis absolute position (normally B)
L3004 Fifth axis absolute position
L3005 Sixth axis absolute position
L3006 Seventh axis absolute position
L3007 Eighth axis absolute position

AXIS FIXTURE OFFSETS
(as seen on offset display)

L3008 First axis fixture offset (normally X)
L3009 Second axis fixture offset (normally Y)
L3010 Third axis fixture offset (normally Z)
L3011 Fourth axis fixture offset (normally B)
L3012 Fifth axis fixture offset

BORE CYCLE OFFSETS
(as seen on offset display)

L3016 First configured axis bore cycle offset U88 (normally X)
L3017 Second configured axis bore cycle offset U88 (normally Y)

PROGRAMMED TOOL DATA (READY TOOL)
(as seen on tool display)

L3018 Diameter Compensation
L3020 Active Pallet number (0..999)

AXIS GRID POSITIONS

L3021 First axis grid position (normally X)
L3022 Second axis grid position (normally Y)
L3023 Third axis grid position (normally Z)
L3024 Fourth axis grid position (normally B)
L3025 Fifth axis grid position
L3026 Sixth axis grid position
L3027 Seventh axis grid position

L3028 Eighth axis grid position

L3029 Third axis G26 pull back position (normally Z)

L3030 Spindle Tool Number
L3033 Length Compensation

LAST AXIS PROBE MAKE/BREAK POSITIONS

L3042 First axis probe make/break position (normally X)
L3043 Second axis probe make/break position (normally Y)
L3044 Third axis probe make/break position (normally Z)
L3045 Fourth axis probe make/break position (normally B)
L3046 Fifth axis probe make/break position
L3047 Sixth axis probe make/break position
L3048 Seventh axis probe make/break position
L3049 Eighth axis probe make/break position

L3060 Machine Serial Number
L3061 Year
L3062 Month
L3063 Day
L3064 Hour
L3065 Minute
L3066 Second

L3070 PI "Circumference divided by the Diameter of any circle"
L3071 E "Natural Logarithm Base"

L3080 3rd rotary axis (normally A) rotary center line with respect to 2nd linear axis (normally Y)
L3081 3rd rotary axis (normally A) rotary center line with respect to 3rd linear axis (normally Z)

L3082 1st rotary axis (normally B) rotary center line with respect to 1st linear axis (normally X)
L3083 1st rotary axis (normally B) rotary center line with respect to 3rd linear axis (normally Z)

L3084 2nd rotary axis (normally C) rotary center line with respect to 1st linear axis (normally X)
L3085 2nd rotary axis (normally C) rotary center line with respect to 2nd linear axis (normally Y)

L3090 3rd axis (normally Z) plus travel limit

NOTE

Where appropriate, parameter values are converted to inch or metric at the time of assignment only, depending on the modal state of the control; G70 (inch) or G71 (metric).

TOOL SENSOR PARAMETERS

(Read Only)(Additional Tool Sensor System Parameters are listed in Appendix B)

L3100 = Amount of tool sensor deflection after G72 command block.

L3102 = Number of cutting teeth detected during (MTD) tool measuring cycle.

L3103 = Lowest tooth detected (value is the incremental distance from L1102 position) during (MTD) tool measuring cycle.

L3104 = Highest tooth detected (value is the incremental distance from L1102 position) during (MTD) tool measuring cycle.

APPENDIX D

PROGRAM LISTINGS for DEFINED CYCLES G81-G89

G81

```
(DFC,G81,SAVE)
! KEARNEY AND TRECKER G81 CYCLE VERSION 1.0 INITIAL RELEASE 8-23-86
(IFT,G91,M99M0G80)
(IFT,NOT(DEF(L806)),(PAR,L806=F))
(IFT,NOT(DEF(L818)),(PAR,L818=Z))
(IFT,NOT(DEF(L826)),(PAR,L826=Z))
(IFT,L818>Z,ZL818F0)
G78AL801BL802CL803EL805RL818XL824YL825F0
ZL818F0
ZL826FL806
ZL818F0
FL806
(DLP,L801,L802,L803,L824,L825)
(END)
```

G82

```
(DFC,G82,SAVE)
! KEARNEY AND TRECKER G82 CYCLE VERSION 1.0 INITIAL RELEASE 8-23-86
(IFT,G91,M99M0G80)
(PAR,L819=S)
(IFT,NOT(DEF(L806)),(PAR,L806=F))
(IFT,NOT(DEF(L818)),(PAR,L818=Z))
(IFT,NOT(DEF(L826)),(PAR,L826=Z))
(IFT,L818>Z,ZL818F0)
G78AL801BL802CL803EL805RL818XL824YL825F0
ZL818F0
(PAR,L835=L826+.01)
(IFT,G71,(PAR,L835=L826+.254))
ZL835FL806
S100
(IFT,G71,F12.7ZL826)
(IFT,G70,F.5ZL826)
G4F2.
ZL818F0
SL819FL806
(DLP,L801,L802,L803,L824,L825)
(END)
```

G83

```
(DFC,G83,SAVE)
! KEARNEY AND TRECKER G83 CYCLE VERSION 1.0 INITIAL RELEASE 8-23-86
(IFT,G91,M99M0G80)
(IFT,NOT(DEF(L806)),(PAR,L806=F))
(IFT,NOT(DEF(L805)),M99M0G80)
```

```
(IFT,NOT(DEF(L818)),(PAR,L818=Z))
(IFT,NOT(DEF(L826)),(PAR,L826=Z))
(IFT,L818>Z,ZL818F0)
G78AL801BL802CL803EL805RL818XL824YL825F0
ZL818F0
(PAR,L835=3./(S/60.),L836=L818)
(LBL,10)(IFT,L836-L805<=L826,(GTO,20))
(PAR,L836=L836-L805)
ZL836FL806
G4FL835
(GTO,10)
(LBL,20)ZL826FL806
G4FL835
ZL818F0
FL806
(DLP,L801,L802,L803,L824,L825)
(END)
```

G84

```
(DFC,G84,SAVE)
! KEARNEY AND TRECKER G84 CYCLE VERSION 1.0 INITIAL RELEASE 8-23-86
(IFT,G91,M99M0G80)
(PAR,L819=S)
(IFT,NOT(DEF(L806)),(PAR,L806=F))
(IFT,NOT(DEF(L818)),(PAR,L818=Z))
(IFT,NOT(DEF(L826)),(PAR,L826=Z))
(PAR,L835=48.)
(IFT,M49,(PAR,L835=49.))
(IFT,L818>Z,ZL818F0)
G78AL801BL802CL803EL805RL818XL824YL825F0SL819M40ML835
ZL818F0
(PAR,L834=3.)
(IFT,M4,(PAR,L834=4.))
M49M5ZL826FL806
(IFT,L834=3.,M4ZL818)
(IFT,L834=4.,M3ZL818)
ML834ML835
(DLP,L801,L802,L803,L824,L825)
(END)
```

G85

```
(DFC,G85,SAVE)
! KEARNEY AND TRECKER G85 CYCLE VERSION 1.0 INITIAL RELEASE 8-23-86
(IFT,G91,M99M0G80)
(IFT,NOT(DEF(L806)),(PAR,L806=F))
(IFT,NOT(DEF(L818)),(PAR,L818=Z))
(IFT,NOT(DEF(L826)),(PAR,L826=Z))
(IFT,L818>Z,ZL818F0)
G78AL801BL802CL803EL805RL818XL824YL825F0
ZL818F0
ZL826FL806
ZL818FL806
(DLP,L801,L802,L803,L824,L825)
(END)
```

G86

```
(DFC,G86,SAVE)
! KEARNEY AND TRECKER G86 CYCLE VERSION 1.0 INITIAL RELEASE 8-23-86
(IFT,G91,M99M0G80)
(IFT,NOT(DEF(L806)),(PAR,L806=F))
(IFT,NOT(DEF(L818)),(PAR,L818=Z))
(IFT,NOT(DEF(L826)),(PAR,L826=Z))
(PAR,L834=3.)
(IFT,M4,(PAR,L834=4.))
(IFT,L818>Z,ZL818F0)
G78AL801BL802CL803EL805RL818XL824YL825F0
ZL818F0
ZL826FL806
M5
ZL818F0
ML834FL806
(DLP,L801,L802,L803,L824,L825)
(END)
```

G87

```
(DFC,G87,SAVE)
! KEARNEY AND TRECKER G87 CYCLE VERSION 1.0 INITIAL RELEASE 8-23-86
(IFT,G91,M99M0G80)
(IFT,NOT(DEF(L806)),(PAR,L806=F))
(IFT,NOT(DEF(L805)),M99M0G80)
(IFT,NOT(DEF(L818)),(PAR,L818=Z))
(IFT,NOT(DEF(L826)),(PAR,L826=Z))
(IFT,L818>Z,ZL818F0)
G78AL801BL802CL803EL805RL818XL824YL825F0
ZL818F0
(PAR,L834=L818,L837=L805)
(PAR,L835=.1,L836=100.)
(IFT,G71,(PAR,L835=2.54,L836=2540.))
(LBL,10)(IFT,L834-L837<=L826,(GTO,30))
(PAR,L834=L834-L837)
ZL834FL806
ZL818F0
(PAR,L834=L834+L835)
ZL834FL836
(PAR,L837=L805+L835)
(GTO,10)
(LBL,30)ZL826FL806
ZL818F0
FL806
(DLP,L801,L802,L803,L824,L825)
(END)
```

G88

```
(DFC,G88,SAVE)
! KEARNEY AND TRECKER G88 CYCLE VERSION 1.0 INITIAL RELEASE 8-23-86
(IFT,G91,M99M0G80)
(IFT,NOT(DEF(L806)),(PAR,L806=F))
(IFT,NOT(DEF(L818)),(PAR,L818=Z))
```

```
(IFT,NOT(DEF(L826)),(PAR,L826=Z))
(IFT,L818>Z,ZL818F0)
G78AL801BL802CL803EL805RL818XL824YL825F0
ZL818F0
ZL826FL806
(PAR,L834=3.,L835=-L3016,L836=-L3017)
(IFT,M4,(PAR,L834=4.))
M19
G91XL3016YL3017F0
G90ZL818
G91ML834XL835YL836
G90FL806
(DLP,L801,L802,L803,L824,L825)
(END)
```

G89

```
(DFC,G89,SAVE)
! KEARNEY AND TRECKER G89 CYCLE VERSION 1.0 INITIAL RELEASE 8-23-86
(IFT,G91,M99M0G80)
(IFT,NOT(DEF(L806)),(PAR,L806=F))
(IFT,NOT(DEF(L818)),(PAR,L818=Z))
(IFT,NOT(DEF(L826)),(PAR,L826=Z))
(IFT,L818>Z,ZL818F0)
G78AL801BL802CL803EL805RL818XL824YL825F0
ZL818F0
(PAR,L835=3./(S/60.))
ZL826FL806
G4FL835
ZL818FL806
(DLP,L801,L802,L803,L824,L825)
(END)
```

APPENDIX E

PROGRAM LISTINGS for G24, G25, G26 and G27 PROBE CYCLES

G24 PARAMETER ASSIGNMENTS

L831 = PROGRAMMED LETTER ADDRESS TOTAL
L832 = DIFFERENCE BETWEEN PROGRAMMED AND ACTUAL POSITION(S)

G24

```
(DFC,G24,SAVE)
! KEARNEY AND TRECKER G24 CYCLE VERSION 1.0 INITIAL RELEASE 8-23-86
(IFT,DEF(L807)ORDEF(L813),G800M0M99)        ! ERROR: NO OTHER M OR G CODES ALLOWED
(CER,MAS)
(IFT,G71AND(L3033<114.483),G800M0M99)       ! ERROR: ILLEGAL TOOL COMPENSATION
(IFT,G70AND(L3033<4.5072),G800M0M99)        ! ERROR: ILLEGAL TOOL COMPENSATION
(PAR,L831=0)
(IFT,DEF(L818),(PAR,L831=L831+1.))
(IFT,DEF(L824),(PAR,L831=L831+2.))
(IFT,DEF(L825),(PAR,L831=L831+4.))
(IFT,DEF(L826),(PAR,L831=L831+8.))
(IFT,DEF(L801),(PAR,L831=L831+16.))
(IFT,DEF(L802),(PAR,L831=L831+32.))
(IFT,DEF(L803),(PAR,L831=L831+64.))
(IFT,L831=1.,(GTO,100))
(IFT,L831=2.,(GTO,200))
(IFT,L831=4.,(GTO,550))
(IFT,L831=8.,(GTO,400))
(IFT,L831=6.,(GTO,500))
(IFT,L831=16.,(GTO,600))
(IFT,L831=32.,(GTO,700))
(IFT,L831=64.,(GTO,800))
G800M0M99                                   ! ERROR: UNRECOGNIZED G24 FORMAT
(LBL,100)(PAR,L832=ABS(L1005-L818))         ! G24R PROBE CYCLE
(GTO,8000)
(LBL,200)(PAR,L832=ABS(X-L824))             ! G24X PROBE CYCLE
(GTO,8000)
(LBL,400)(PAR,L832=ABS(Z-(L3029-L3033)-L826))      ! G24Z PROBE CYCLE
(GTO,8000)
(LBL,500)(PAR,L832=ABS(X-L824))             ! G24XY PROBE CYCLE
(IFT,L832>L805,G800M0M99)                   ! ERROR: G24 TOLERANCE EXCEEDED
(LBL,550)(PAR,L832=ABS(Y-L825))             ! G24Y PROBE CYCLE
(GTO,8000)
(LBL,600)(PAR,L832=ABS(A-L801))             ! G24A PROBE CYCLE
(GTO,8000)
(LBL,700)(PAR,L832=ABS(B-L802))             ! G24B PROBE CYCLE
(GTO,8000)
(LBL,800)(PAR,L832=ABS(C-L803))             ! G24C PROBE CYCLE
```

```
(LBL,8000)(IFT,L832>L805,G800M0M99)                 ! ERROR: G24 TOLERANCE EXCEEDED
G800
(END)
```

G25 PARAMETER ASSIGNMENTS

ABSOLUTE OR INCREMENTAL MODES, INCH OR METRIC MODES,FEEDRATE INPUT OPTIONAL

NOTE

USES AXIS POSITIONS RELATIVE TO OFFSETS. IF G25R IS RUN WITH ACTIVE OFFSETS AND/OR TOOL LENGTH COMP, THESE SAME OFFSETS MUST STAY IN PLACE FOR ALL FOLLOWING G25 AUTO-REGRID CYCLES.

L831 =	G90/G91 ENTRY STATE	
L832 =	X BUSHING EDGE LINE-UP DISTANCE	{INIT 0.0875"}
L833 =	Z BUSHING EDGE LINE-UP DISTANCE	{INIT 0.3"}
L834 =	Z DEPARTURE FOR BUSHING DETECTION	{INIT 0.7125"}
L835 =	X,Y,Z RETRACTION DISTANCE AFTER 1ST SURFACE DETECT	{INIT 0.04"}
L836 =	Z DEPARTURE FOR BUSHING HOLE ENTRY	{INIT 0.6"}
L837 =	SECOND DETECTION FEEDRATE	{INIT 1.1 IPM}
L838 =	CALCULATED X REGRID POS, XRG	
L839 =	CALCULATED Y REGRID POS, YRG	
L840 =	CALCULATED Z REGRID POS, ZRG	
L841 =	CALCULATED X POS BUSHING HOLE EDGE, XC+	
L842 =	CALCULATED X NEG BUSHING HOLE EDGE, XC-	
L843 =	CALCULATED Y POS BUSHING HOLE EDGE, YC+	
L844 =	CALCULATED Y NEG BUSHING HOLE EDGE, YC-	
L845 =	MAX ALLOWABLE FEEDRATE, F	{INIT 100 IPM}
L846 =	MIN ALLOWABLE PROBE LENGTH	{INIT 4.5072"}
L847 =	COMMANDED SPINDLE ANGLE	
L848 =	PASS COUNTER	
L849 =	CALCULATED EFFECTIVE BALL RADIUS, REFF	

G25

```
(DFC,G25,SAVE)
! KEARNEY AND TRECKER G25 CYCLE VERSION 1.0 INITIAL RELEASE 8-23-86
(IFT,DEF(L807)ORDEF(L813),G800M0M99)          ! ERROR: NO OTHER M OR G CODES ALLOWED
(IFT,G01,G800M0M99)          ! ERROR: G01 MODE NOT ALLOWED, G00 MODE MUST BE IN EFFECT
(CER,MAS)
(IFT,G71AND(L3033<114.483),G800M0M99)          ! ERROR: ILLEGAL TOOL COMPENSATION
(IFT,G70AND(L3033<4.5072),G800M0M99)           ! ERROR: ILLEGAL TOOL COMPENSATION
(IFT,G71,(GTO,50))
(PAR,L832=.1189,L833=.3,L834=.7125,L835=.04,L836=.6,L837=1.1,L845=100.)
(GTO,60)
(LBL,50)(PAR,L832=3.02,L833=7.62,L834=18.098,L835=1.016,L836=15.24,L837=28.)
(PAR,L845=2540.)
(LBL,60)(IFT,NOT(DEF(L806)),(PAR,L806=L845))
(IFT,L806>L845,(PAR,L806=L845))
(PAR,L848=0,L831=91.)
(IFT,G90,(PAR,L831=90.))
(IFT,NOT(DEF(L818)),(GTO,100))
(PAR,L1006=L818/2."BUSHING RADIUS",L838=X+L1006+L832,L839=Y)
(GTO,200)
```

```
(LBL,100)
(IFT,NOT(DEF(L1000)),G800M0M99)          ! ERROR: G25R POST POSITION NOT ESTABLISHED
(IFT,NOT(DEF(L1001)),G800M0M99)          ! ERROR: G25R POST POSITION NOT ESTABLISHED
(IFT,NOT(DEF(L1002)),G800M0M99)          ! ERROR: G25R POST POSITION NOT ESTABLISHED
(IFT,NOT(DEF(L1006)),G800M0M99)          ! ERROR: G25R POST BUSHING NOT ESTABLISHED
(PAR,L838=L1000,L839=L1001,L824=L1000-L1006-L832,L825=L1001,L826=L1002+L833)
G90XL824YL825ZL826FL806
(LBL,200)(PAR,L847=0)
M19SL847
(PAR,L826=-L834,L1020=0)
G91G22ZL826FL806
(LBL,210)(IFT,L1020=0,(GTO,210))
(IFT,L1021=8.,(GTO,215))
(IFT,L1021=1.,G800GL831M99M0)            ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL831M99M0)            ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL831M99M0                           ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,215)(PAR,L826=L3044+L835,L1020=0)
G23G90ZL826
(LBL,216)(IFT,L1020=0,(GTO,216))
(IFT,L1021=8.,(GTO,217))
(IFT,L1021=1.,G800GL831M99M0)            ! ERROR: PROBE WAS IN NON-CONTACT AT START OF G23
(IFT,L1021=2.,G800GL831M99M0)            ! ERROR: PROBE DID NOT LOSE CONTACT WITH SURFACE
G800GL831M99M0                           ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,217)ZL826
(PAR,L826=-L834,L1020=0)
G91G22ZL826FL837M49
M48
(LBL,220)(IFT,L1020=0,(GTO,220))
(IFT,L1021=8.,(GTO,225))
(IFT,L1021=1.,G800GL831M99M0)            ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL831M99M0)            ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL831M99M0                           ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,225)(PAR,L840=L3044,L826=L3044+L833)
G90ZL826FL806
G90XL838
(PAR,L826=L826-L836)
G90ZL826
(LBL,300)(PAR,L847=0)
M19SL847
(PAR,L824=L1006*10.,L1020=0)
G91G22XL824FL806
(LBL,310)(IFT,L1020=0,(GTO,310))
(IFT,L1021=8.,(GTO,315))
(IFT,L1021=1.,G800GL831M99M0)            ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL831M99M0)            ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL831M99M0                           ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,315)(PAR,L824=L3042-L835,L1020=0)
G23G90XL824
(LBL,316)(IFT,L1020=0,(GTO,316))
(IFT,L1021=8.,(GTO,317))
(IFT,L1021=1.,G800GL831M99M0)            ! ERROR: PROBE WAS IN NON-CONTACT AT START OF G23
(IFT,L1021=2.,G800GL831M99M0)            ! ERROR: PROBE DID NOT LOSE CONTACT WITH SURFACE
G800GL831M99M0                           ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,317)XL824
(PAR,L824=2.*L835,L1020=0)
```

```
G91G22XL824FL837M49
M48
(LBL,320)(IFT,L1020=0,(GTO,320))
(IFT,L1021=8.,(GTO,325))
(IFT,L1021=1.,G800GL831M99M0)          ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL831M99M0)          ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL831M99M0                         ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
(LBL,325)(PAR,L841=L3042)
G90XL838FL806
(PAR,L847=180.)
M19SL847
(PAR,L824=-10.*L1006,L1020=0)
G91G22XL824
(LBL,330)(IFT,L1020=0,(GTO,330))
(IFT,L1021=8.,(GTO,335))
(IFT,L1021=1.,G800GL831M99M0)          ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL831M99M0)          ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL831M99M0                         ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
(LBL,335)(PAR,L824=L3042+L835,L1020=0)
G23G90XL824
(LBL,336)(IFT,L1020=0,(GTO,336))
(IFT,L1021=8.,(GTO,337))
(IFT,L1021=1.,G800GL831M99M0)          ! ERROR: PROBE WAS IN NON-CONTACT AT START OF G23
(IFT,L1021=2.,G800GL831M99M0)          ! ERROR: PROBE DID NOT LOSE CONTACT WITH SURFACE
G800GL831M99M0                         ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
(LBL,337)XL824
(PAR,L824=-2.*L835,L1020=0)
G91G22XL824FL837M49
M48
(LBL,340)(IFT,L1020=0,(GTO,340))
(IFT,L1021=8.,(GTO,345))
(IFT,L1021=1.,G800GL831M99M0)          ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL831M99M0)          ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL831M99M0                         ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
(LBL,345)(PAR,L842=L3042,L838=L842+(L841-L842)/2.)
G90XL838FL806
(PAR,L847=270.)
M19SL847
(PAR,L825=L1006*10.,L1020=0)
G91G22YL825
(LBL,410)(IFT,L1020=0,(GTO,410))
(IFT,L1021=8.,(GTO,415))
(IFT,L1021=1.,G800GL831M99M0)          ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL831M99M0)          ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL831M99M0                         ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
(LBL,415)(PAR,L825=L3043-L835,L1020=0)
G23G90YL825
(LBL,416)(IFT,L1020=0,(GTO,416))
(IFT,L1021=8.,(GTO,417))
(IFT,L1021=1.,G800GL831M99M0)          ! ERROR: PROBE WAS IN NON-CONTACT AT START OF G23
(IFT,L1021=2.,G800GL831M99M0)          ! ERROR: PROBE DID NOT LOSE CONTACT WITH SURFACE
G800GL831M99M0                         ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
(LBL,417)YL825
(PAR,L825=2.*L835,L1020=0)
G91G22YL825FL837M49
```

```
M48
(LBL,420)(IFT,L1020=0,(GTO,420))
(IFT,L1021=8.,(GTO,425))
(IFT,L1021=1.,G800GL831M99M0)           ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL831M99M0)           ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL831M99M0                          ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,425)(PAR,L843=L3043)
G90YL839FL806
(PAR,L847=90.)
M19SL847
(PAR,L825=-10.*L1006,L1020=0)
G91G22YL825
(LBL,430)(IFT,L1020=0,(GTO,430))
(IFT,L1021=8.,(GTO,435))
(IFT,L1021=1.,G800GL831M99M0)           ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL831M99M0)           ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL831M99M0                          ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,435)(PAR,L825=L3043+L835,L1020=0)
G23G90YL825
(LBL,436)(IFT,L1020=0,(GTO,436))
(IFT,L1021=8.,(GTO,437))
(IFT,L1021=1.,G800GL831M99M0)           ! ERROR: PROBE WAS IN NON-CONTACT AT START OF G23
(IFT,L1021=2.,G800GL831M99M0)           ! ERROR: PROBE DID NOT LOSE CONTACT WITH SURFACE
G800GL831M99M0                          ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,437)YL825
(PAR,L825=-2.*L835,L1020=0)
G91G22YL825FL837M49
M48
(LBL,440)(IFT,L1020=0,(GTO,440))
(IFT,L1021=8.,(GTO,445))
(IFT,L1021=1.,G800GL831M99M0)           ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL831M99M0)           ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL831M99M0                          ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,445)(PAR,L844=L3043,L839=L844+(L843-L844)/2.,L849=L1006-(L843-L844)/2.)
(PAR,L850=L1006-(L841-L842)/2.)
G90YL839FL806
(PAR,L848=L848+1.)
(IFT,L848<>1.,(GTO,500))
(IFT,NOT(DEF(L818)),(GTO,300))
(PAR,L824=L838-L1006-L832,L825=L839,L826=L840+L833)
G90ZL826
G90XL824YL825
(GTO,200)
(LBL,500)(PAR,L1003=L850"X BALL RADIUS",L1004=L849"Y BALL RADIUS")
(PAR,L826=L840+L3029-L3033)
G90ZL826
(DLP,L1020,L1021)
(IFT,DEF(L818),(GTO,550))
G92XL1000YL1001Z,L1002+L3029-L3033
GL831G800
(LBL,550)(PAR,L1000=L838"X POST POSITION",L1001=L839"Y POST POSITION")
(PAR,L1002=L840"Z POST POSITION")
(DLP,L1020,L1021)
GL831G800
(END)
```

G26 PARAMETER ASSIGNMENTS

INCH OR METRIC MODES, ABSOLUTE OR INCREMENTAL MODES, FEEDRATE INPUT OPTIONAL, ANGLE PROBE CYCLES, ROTARY AXIS PROBE CYCLES

NOTE

1. **ASSUMES ACTIVE TOOL LENGTH COMP FOR G26Z.**
2. **ROTARY AXIS CYCLES ASSUME THAT EFFECTIVE BALL RADIUS IS POSITIVE; SERIOUS PROBE DAMAGE MAY RESULT IF THE PROBE USED HAS NEGATIVE EFFECTIVE BALL RADIUS.**

L831 = PROGRAMMED LETTER ADDRESS TOTAL
L832 = SPINDLE ORIENTATION ANGLE
L833 = RETRACTION DISTANCE AFTER 1ST DETECT {INIT 0.04"}
L834 = 2ND DETECTION FEEDRATE {INIT 1.1 IPM}
L835 = X RETRACTION DISTANCE AFTER 1ST DETECT
L836 = Y RETRACTION DISTANCE AFTER 1ST DETECT
L837 = X DEPARTURE FOR 2ND DETECTION
L838 = Y DEPARTURE FOR 2ND DETECTION
L839 = X COMPONENT OF EFF BALL RADIUS, REFFX
L840 = Y COMPONENT OF EFF BALL RADIUS, REFFY
L841 = X RETRACTION AND FINAL POSN COORDINATES
L842 = Y RETRACTION AND FINAL POSN COORDINATES
L843 = G90/1 STATUS WHEN PROBE CYCLE IS CALLED
L844 = MAX ALLOWABLE FEEDRATE {INIT 100 IPM}
L848 = INCREMENTAL X MOVE TO Z POST
L849 = SPINDLE TOOL LENGTH
L850 = SURFACE ANGLE

THE FOLLOWING PARAMETERS ARE USED ONLY FOR ROTARY PROBE CYCLES

L851 = INITIAL Z POSITION
L852 = Z COMMANDED POSITION
L853 = RUN COUNTER
L854 = FIRST Z CONTACT POSITION
L855 = RESTORE MODAL FEEDRATE WORD
L856 = SECOND CONTACT POSITION LINEAR LENGTH
L857 = ROTARY MOVE

```
(DFC,G26,SAVE)
!KEARNEY AND TRECKER G26 CYCLE VERSION 1.0 INITIAL RELEASE 8-23-86
(IFT,DEF(L807)ORDEF(L813),G800M0M99)          ! ERROR: NO OTHER M OR G CODES ALLOWED
(IFT,G01,G800M0M99)          ! ERROR: G01 MODE NOT ALLOWED, G00 MODE MUST BE IN EFFECT
(CER,MAS)
(IFT,G71AND(L3033<114.483),G800M0M99)          ! ERROR: ILLEGAL TOOL COMPENSATION
(IFT,G70AND(L3033<4.5072),G800M0M99)          ! ERROR: ILLEGAL TOOL COMPENSATION
(PAR,L843=91.)
(IFT,G90,(PAR,L843=90.))
(IFT,G71,(GTO,50))
(PAR,L833=.040,L834=1.1,L844=100.,L848=.1189)
(GTO,60)
(LBL,50)(PAR,L833=1.016,L834=28.,L844=2540.,L848=3.02)
(LBL,60)(IFT,DEF(L806),(GTO,65))
(PAR,L806=L844)
(LBL,65)(IFT,L806>L844,(PAR,L806=L844))
(PAR,L831=0)
```

```
(IFT,DEF(L801),(PAR,L831=L831+1.))
(IFT,DEF(L802),(PAR,L831=L831+2.))
(IFT,DEF(L803),(PAR,L831=L831+3.))
(IFT,DEF(L818),(PAR,L831=L831+4.))
(IFT,DEF(L824),(PAR,L831=L831+8.))
(IFT,DEF(L825),(PAR,L831=L831+16.))
(IFT,DEF(L826),(PAR,L831=L831+32.))
(IFT,DEF(L820),(PAR,L831=L831+128.))
(IFT,L831=49.,(GTO,70))
(IFT,L831=42.,(GTO,80))
(IFT,L831=51.,(GTO,70))
(IFT,L831=32.,(GTO,100))
(IFT,L831=5.,(GTO,200))
(IFT,L831=24.,(GTO,300))
(IFT,L831=8.,(GTO,400))
(IFT,L831=16.,(GTO,500))
(IFT,L831=160.,(GTO,900))
G800GL843M0M99                     ! ERROR: UNRECOGNIZED FORMAT FOR G26
```

G26 A, B, OR C ROTARY ALIGNMENT PROBE CYCLE

```
(LBL,70)(PAR,L856=L825)
(GTO,90)
(LBL,80)(PAR,L856=L824)
(LBL,90)(IFT,NOT(DEF(L805)),(PAR,L805=0))
(PAR,L851=Z,L852=L826,L853=0)
```

G26 Z PROBE CYCLE

```
(LBL,100)(IFT,L843=91.,(GTO,110))
(PAR,L826=L826-Z)
(LBL,110)(PAR,L832=0,L1020=0)
M19SL832
G91G22ZL826FL806
(LBL,120)(IFT,L1020=0,(GTO,120))
(IFT,L1021=8.,(GTO,125))
(IFT,L1021=1.,G800GL843M99M0)      ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)      ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                     ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,125)(PAR,L826=L3044+L833,L1020=0)
G23G90ZL826
(LBL,126)(IFT,L1020=0,(GTO,126))
(IFT,L1021=8.,(GTO,127))
(IFT,L1021=1.,G800GL831M99M0)      ! ERROR: PROBE WAS IN NON-CONTACT AT START OF G23
(IFT,L1021=2.,G800GL831M99M0)      ! ERROR: PROBE DID NOT LOSE CONTACT WITH SURFACE
G800GL831M99M0                     ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,127)ZL826
(PAR,L826=L833*-2.,L1020=0)
G91G22ZL826FL834M49
M48
(LBL,130)(IFT,L1020=0,(GTO,130))
(IFT,L1021=8.,(GTO,135))
(IFT,L1021=1.,G800GL843M99M0)      ! ERROR: PROBE WAS IN CONTACT AT START OF G22
```

```
(IFT,L1021=2.,G800GL843M99M0)          ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                         ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
(LBL,135)G90ZL3044
(IFT,L831>32.,(GTO,150))
(IFT,NOT(DEF(L814)),(GTO,140))
(PAR,L814=L814*10000.)
(FWR,"N",L814I)
(LBL,140)(FWR,"Z SURFACE=",Z)
(FWR,)
(PAR,L826=L3029-L3033)
G91ZL826FL806
(DLP,L1020,L1021)
GL843G800
(LBL,150)(PAR,L853=L853+1.,L826=L3029-L3033+L3044)
(IFT,L853=1.,(PAR,L854=L3044))
(IFT,L826<L851,(PAR,L826=L851))
G90ZL826FL806
(PAR,L826=L852)
(IFT,L853=2.,(GTO,160))
G78G91XL824YL825
(GTO,100)
(LBL,160)(PAR,L854=L854+L805,L857=ATN((L854-L3044)/L856),L855=F,L806=0)
(IFT,L831=49.,G91A-L857FL806)
(IFT,L831=42.,G91BL857FL806)
(IFT,(L831=51.)AND(B<180.),G91CL857FL806)
(IFT,(L831=51.)AND(B>180.),G91C-L857FL806)
(IFT,NOT(DEF(L814)),(GTO,170))
(PAR,L814=L814*10000.)
(FWR,"N",L814I)
(LBL,170)(IFT,L831=49.,(FWR,"A ALIGNMENT =",A))
(IFT,L831=42.,(FWR,"B ALIGNMENT =",B))
(IFT,L831=51.,(FWR,"C ALIGNMENT =",C))
(FWR,)
(DLP,L1020,L1021)
GL843FL855G800
```

G26 R, A PROBE CYCLE

```
(LBL,200)(PAR,L824=L818*COS(L801),L825=L818*SIN(L801),L850=L801)
(GTO,320)
```

G26 X, Y PROBE CYCLE

```
(LBL,300)(IFT,L843=91.,(GTO,305))
(PAR,L824=L824-X,L825=L825-Y)
(LBL,305)(PAR,L850=ATN(ABS(L825/L824)))
(IFT,L824>=0,(GTO,310))
(IFT,L825>=0,(GTO,315))
(PAR,L850=180.+L850)
(GTO,320)
(LBL,310)(IFT,L825>=0,(GTO,320))
(PAR,L850=360.-L850)
(GTO,320)
```

```
(LBL,315)(PAR,L850=180.-L850)
(LBL,320)(PAR,L832=INT((360.-L850)+.5))
(PAR,L835=L833*COS(L850),L839=L1003*COS(L850),L836=L833*SIN(L850))
(PAR,L840=L1003*SIN(L850),L1020=0)
M19SL832
G91G22XL824YL825FL806
(LBL,330)(IFT,L1020=0,(GTO,330))
(IFT,L1021=8.,(GTO,335))
(IFT,L1021=1.,G800GL843M99M0)        ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)        ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                       ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
(LBL,335)(PAR,L841=L3042-L835,L842=L3043-L836,L1020=0)
G23G90XL841YL842
(LBL,336)(IFT,L1020=0,(GTO,336))
(IFT,L1021=8.,(GTO,337))
(IFT,L1021=1.,G800GL831M99M0)        ! ERROR: PROBE WAS IN NON-CONTACT AT START OF G23
(IFT,L1021=2.,G800GL831M99M0)        ! ERROR: PROBE DID NOT LOSE CONTACT WITH SURFACE
G800GL831M99M0                       ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
(LBL,337)XL841YL842
(PAR,L837=2.*L835,L838=2.*L836,L1020=0)
G91G22XL837YL838FL834M49
M48
(LBL,340)(IFT,L1020=0,(GTO,340))
(IFT,L1021=8.,(GTO,345))
(IFT,L1021=1.,G800GL843M99M0)        ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)        ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                       ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
(LBL,345)(PAR,L841=L3042+L839,L842=L3043+L840)
G90XL841YL842FL806
(IFT,NOT(DEF(L814)),(GTO,350))
(PAR,L814=L814*10000.)
(FWR,"N",L814I)
(LBL,350)(FWR,"X SURFACE=",X)
(FWR,"Y SURFACE=",Y)
(FWR,)
(DLP,L1020,L1021)
GL843G800
```

G26X PROBE CYCLE

```
(LBL,400)(IFT,L843=91.,(GTO,405))
(PAR,L824=L824-X)
(LBL,405)(IFT,L824>=0,(GTO,410))
(PAR,L832=180.,L835=L833,L839=-L1003)
(GTO,420)
(LBL,410)(PAR,L832=0,L835=-L833,L839=L1003)
(LBL,420)M19SL832
(PAR,L1020=0)
G91G22XL824FL806
(LBL,430)(IFT,L1020=0,(GTO,430))
(IFT,L1021=8.,(GTO,435))
(IFT,L1021=1.,G800GL843M99M0)        ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)        ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                       ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
```

```
(LBL,435)(PAR,L841=L3042+L835,L1020=0)
G23G90XL841
(LBL,436)(IFT,L1020=0,(GTO,436))
(IFT,L1021=8.,(GTO,437))
(IFT,L1021=1.,G800GL831M99M0)          ! ERROR: PROBE WAS IN NON-CONTACT AT START OF G23
(IFT,L1021=2.,G800GL831M99M0)          ! ERROR: PROBE DID NOT LOSE CONTACT WITH SURFACE
G800GL831M99M0                         ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
(LBL,437)XL841
(PAR,L837=L835*-2.,L1020=0)
G91G22XL837FL834M49
M48
(LBL,440)(IFT,L1020=0,(GTO,440))
(IFT,L1021=8.,(GTO,445))
(IFT,L1021=1.,G800GL843M99M0)          ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)          ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                         ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
(LBL,445)(PAR,L841=L3042+L839)
G90XL841FL806
(IFT,NOT(DEF(L814)),(GTO,450))
(PAR,L814=L814*10000.)
(FWR,"N",L814I)
(LBL,450)(FWR,"X SURFACE=",X)
(FWR,)
(DLP,L1020,L1021)
GL843G800
```

G26Y PROBE CYCLE

```
(LBL,500)(IFT,L843=91.,(GTO,505))
(PAR,L825=L825-Y)
(LBL,505)(IFT,L825>=0,(GTO,510))
(PAR,L832=90.,L836=L833,L840=-L1004)
(GTO,520)
(LBL,510)(PAR,L832=270.,L836=-L833,L840=L1004)
(LBL,520)M19SL832
(PAR,L1020=0)
G91G22YL825FL806
(LBL,530)(IFT,L1020=0,(GTO,530))
(IFT,L1021=8.,(GTO,535))
(IFT,L1021=1.,G800GL843M99M0)          ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)          ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                         ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
(LBL,535)(PAR,L842=L3043+L836,L1020=0)
G23G90YL842
(LBL,536)(IFT,L1020=0,(GTO,536))
(IFT,L1021=8.,(GTO,537))
(IFT,L1021=1.,G800GL831M99M0)          ! ERROR: PROBE WAS IN NON-CONTACT AT START OF G23
(IFT,L1021=2.,G800GL831M99M0)          ! ERROR: PROBE DID NOT LOSE CONTACT WITH SURFACE
G800GL831M99M0                         ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
(LBL,537)YL842
(PAR,L838=L836*-2.,L1020=0)
G91G22YL838FL834M49
M48
(LBL,540)(IFT,L1020=0,(GTO,540))
```

```
(IFT,L1021=8.,(GTO,545))
(IFT,L1021=1.,G800GL843M99M0)          ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)          ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                         ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,545)(PAR,L842=L3043+L840)
G90YL842FL806
(IFT,NOT(DEF(L814)),(GTO,550))
(PAR,L814=L814*10000.)
(FWR,"N",L814I)
(LBL,550)(FWR,"Y SURFACE=",Y)
(FWR,)
(DLP,L1020,L1021)
GL843G800
```

G26TZ PROBE CYCLE

```
(LBL,900)(IFT,L843=91.,(GTO,905))
(PAR,L826=L826-Z)
(LBL,905)(PAR,L832=0,L1020=0,L824=L1000-L1006-L848,L825=L1001)
M19SL832
G90XL824YL825FL806
G91G22ZL826
(LBL,910)(IFT,L1020=0,(GTO,910))
(IFT,L1021=8.,(GTO,915))
(IFT,L1021=1.,G800GL843M99M0)          ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)          ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                         ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,915)(PAR,L826=L3044+L833,L1020=0)
G23G90ZL826
(LBL,916)(IFT,L1020=0,(GTO,916))
(IFT,L1021=8.,(GTO,917))
(IFT,L1021=1.,G800GL831M99M0)          ! ERROR: PROBE WAS IN NON-CONTACT AT START OF G23
(IFT,L1021=2.,G800GL831M99M0)          ! ERROR: PROBE DID NOT LOSE CONTACT WITH SURFACE
G800GL831M99M0                         ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,917)ZL826
(PAR,L826=-2.*L833,L1020=0)
G91G22ZL826FL834M49
M48
(LBL,920)(IFT,L1020=0,(GTO,920))
(IFT,L1021=8.,(GTO,925))
(IFT,L1021=1.,G800GL843M99M0)          ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)          ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                         ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,925)(PAR,L826=L3044+L3033-L1002,L849=L3033)
(IFT,NOT(DEF(L814)),(GTO,930))
(PAR,L814=L814*10000.)
(FWR,"N",L814I)
(LBL,930)(FWR,"SPINDLE PROBE LENGTH=",L826)
(FWR,)
(TLD,TL3030PL826)
(PAR,L826=L3044+L849+L3029-L3033-L826)
G90ZL826FL806
(DLP,L1020,L1021)
GL843G800
(END)
```

G27 PARAMETER ASSIGNMENTS

INCH OR METRIC MODES, ABSOLUTE OR INCREMENTAL MODES, FEEDRATE INPUT OPTIONAL, ANGLE PROBE CYCLES ROTARY AXIS PROBE CYCLES

NOTES

G27 B AND C CYCLES ASSUME THAT THE MEASURED SURFACE SEPARATION WILL BE LESS THAN 180°. INCORRECT RESULTS, AND POSSIBLE PROBE DAMAGE, WILL OCCUR IF THIS IS NOT THE CASE.

L831 = PROGRAMMED LETTER ADDRESS TOTAL
L832 = SPINDLE ORIENTATION ANGLE
L833 = RETRACTION DISTANCE AFTER 1ST DETECT {INIT 0.04"}
L834 = 2ND DETECTION FEEDRATE {INIT 1.1 IPM}
L835 = X (B,C) RETRACTION DISTANCE AFTER 1ST DETECT
L836 = Y RETRACTION DISTANCE AFTER 1ST DETECT
L837 = X (B,C) DEPARTURE FOR 2ND DETECTION
L838 = Y DEPARTURE FOR 2ND DETECTION
L839 = X COMPONENT OF EFF BALL RADIUS, REFFX
L840 = Y COMPONENT OF EFF BALL RADIUS, REFFY
L841 = X (B,C) RETRACTION AND FINAL POSN COORDINATES
L842 = Y RETRACTION AND FINAL POSN COORDINATES
L843 = G90/1 STATUS WHEN PROBE CYCLE IS CALLED
L844 = MAX ALLOWABLE FEEDRATE {INIT 100 IPM}
L845 = PASS COUNTER
L846 = CALCULATED X (B,C) POS EDGE, XC+ (BC+,CC+)
L847 = CALCULATED X (B,C) NEG EDGE, XC- (BC-,CC-)
L848 = CALCULATED Y POS EDGE, YC+
L849 = CALCULATED Y NEG EDGE, YC-
L850 = STARTING X (B,C) POSITION, XSTART (BSTART,CSTART)
L851 = STARTING Y POSITION, YSTART
L852 = STARTING Z POSITION, ZSTART
L853 = CALCULATED DIAMETER, R(CALC) {ANGLE CYCLES}
L854 = SIGN OF INITIAL OPERATOR COMMANDED ROTARY MOVE
L855 = ORIENTATION OF PROBE TIP TO ROTARY CENTERLINE { +1 ==> TIP IN FRONT OF CL : -1 ==> TIP BEHIND CL }
L856 = CENTER LINE OF ROTARY AXIS WHICH PROBE MUST CROSS TO CHANGE THE DIRECTION FROM WHICH THE PART WILL APPROACH THE PROBE TIP { I.E., A CW G26 A CYCLE IS BEING EXECUTED. IF THE PROBE IS ON THE (+) SIDE OF THE Z CENTERLINE OF THE A AXIS, THE PART WILL APPROACH THE PROBE FROM 270°. IF THE PROBE IS ON THE (-) SIDE, THE PART APPROACHES FROM 90°. }
L857 = SURFACE ANGLE
L858 = INCH/METRIC TRUNCATION CONSTANT IN SURFACE SEPARATION CALCULATION

```
(DFC,G27,SAVE)
! KEARNEY AND TRECKER G27 CYCLE VERSION 1.0 INITIAL RELEASE 8-23-86
(IFT,DEF(L807)ORDEF(L813),G800M0M99)          ! ERROR: NO OTHER M OR G CODES ALLOWED
(IFT,G01,G800M0M99)          ! ERROR: G01 MODE NOT ALLOWED, G00 MODE MUST BE IN EFFECT
(CER,MAS)
(IFT,G71AND(L3033<114.483),G800M0M99)          ! ERROR: ILLEGAL TOOL COMPENSATION
(IFT,G70AND(L3033<4.5072),G800M0M99)          ! ERROR: ILLEGAL TOOL COMPENSATION
(IFT,G71ANDNOT(DEF(L1003)),(PAR,L1003=.001))
(IFT,G70ANDNOT(DEF(L1003)),(PAR,L1003=.0001))
(IFT,G71ANDNOT(DEF(L1004)),(PAR,L1004=.001))
(IFT,G70ANDNOT(DEF(L1004)),(PAR,L1004=.0001))
(PAR,L843=91.)
```

```
(IFT,G90,(PAR,L843=90.))
(IFT,G71,(GTO,50))
(PAR,L833=.040,L834=1.1,L844=100.,L858=10000.)
(GTO,60)
(LBL,50)(PAR,L833=1.016,L834=28.,L844=2540.,L858=1000.)
(LBL,60)(PAR,L845=0)
(IFT,DEF(L802)ORDEF(L803),(GTO,70))
(IFT,NOT(DEF(L806)),(PAR,L806=L844))
(IFT,L806>L844,(PAR,L806=L844))
(GTO,80)
(LBL,70)(IFT,NOT(DEF(L806)),(PAR,L806=360.))
(PAR,L834=10.1)
(LBL,80)(PAR,L831=0)
(IFT,DEF(L801),(PAR,L831=L831+1.))
(IFT,DEF(L805),(PAR,L831=L831+2.))
(IFT,DEF(L818),(PAR,L831=L831+4.))
(IFT,DEF(L824),(PAR,L831=L831+8.))
(IFT,DEF(L825),(PAR,L831=L831+16.))
(IFT,DEF(L826),(PAR,L831=L831+32.))
(IFT,DEF(L802),(PAR,L831=L831+64.))
(IFT,DEF(L803),(PAR,L831=L831+128.))
(IFT,L831=8.,(GTO,100))
(IFT,L831=12.,(GTO,100))
(IFT,L831=16.,(GTO,200))
(IFT,L831=20.,(GTO,200))
(IFT,L831=5.,(GTO,300))
(IFT,L831=39.,(GTO,300))
(IFT,L831=24.,(GTO,400))
(IFT,L831=28.,(GTO,400))
(IFT,L831=42.,(GTO,500))
(IFT,L831=50.,(GTO,600))
(IFT,L831=58.,(GTO,700))
(IFT,L831=64.,(GTO,800))
(IFT,L831=128.,(GTO,900))
GL843G800M0M99                              ! ERROR: UNRECOGNIZED G27 FORMAT
```

G27 X (, R) PROBE CYCLE

```
(LBL,100)(PAR,L850=X)
(IFT,L843=91.,(GTO,110))
(PAR,L824=L824-X)
(LBL,110)(PAR,L824=ABS(L824),L832=0,L835=-L833)
(LBL,120)M19SL832
(PAR,L1020=0)
G91G22XL824FL806
(LBL,130)(IFT,L1020=0,(GTO,130))
(IFT,L1021=8.,(GTO,135))
(IFT,L1021=1.,G800GL843M99M0)          ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)          ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                         ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
(LBL,135)(PAR,L841=L3042+L835,L1020=0)
G23G90XL841
(LBL,136)(IFT,L1020=0,(GTO,136))
(IFT,L1021=8.,(GTO,137))
```

```
(IFT,L1021=1.,G800GL831M99M0)          ! ERROR: PROBE WAS IN NON-CONTACT AT START OF G23
(IFT,L1021=2.,G800GL831M99M0)          ! ERROR: PROBE DID NOT LOSE CONTACT WITH SURFACE
G800GL831M99M0                         ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,137)XL841
(PAR,L837=-2.*L835,L1020=0)
G91G22XL837FL834M49
M48
(LBL,140)(IFT,L1020≠0,(GTO,140))
(IFT,L1021=8.,(GTO,145))
(IFT,L1021=1.,G800GL843M99M0)          ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)          ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                         ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,145)(PAR,L845=L845+1.)
(IFT,L845<>1.,(GTO,150))
(PAR,L846=L3042)
G90XL850FL806
(PAR,L832=180.,L824=-L824,L835=L833)
(GTO,120)
(LBL,150)(PAR,L847=L3042,L824=L847+(L846-L847)/2.)
G90XL824FL806
(IFT,NOT(DEF(L818)),(GTO,160))
(PAR,L1003=(L818-L846+L847)/2."X BALL RADIUS",L1005=L818"X SURFACE SEPARATION")
(DLP,L1020,L1021)
GL843G800
(LBL,160)(PAR,L1005=L846-L847+L1003*2."X SURFACE SEPARATION")
(IFT,NOT(DEF(L814)),(GTO,170))
(PAR,L814=L814*10000.)
(FWR,"N",L814I)
(LBL,170)(PAR,L1005=INT(L1005*L858+.5)/L858)
(FWR,"X SURFACE SEPARATION=",L1005)
(FWR,"X CENTER=",X)
(FWR,)
(DLP,L1020,L1021)
GL843G800
```

G27 Y (, R) PROBE CYCLE

```
(LBL,200)(PAR,L851=Y)
(IFT,L843=91.,(GTO,210))
(PAR,L825=L825-Y)
(LBL,210)(PAR,L825=ABS(L825),L832=270.,L836=-L833)
(LBL,220)M19SL832
(PAR,L1020=0)
G91G22YL825FL806
(LBL,230)(IFT,L1020=0,(GTO,230))
(IFT,L1021=8.,(GTO,235))
(IFT,L1021=1.,G800GL843M99M0)          ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)          ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                         ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,235)(PAR,L842=L3043+L836,L1020=0)
G23G90YL842
(LBL,236)(IFT,L1020=0,(GTO,236))
(IFT,L1021=8.,(GTO,237))
(IFT,L1021=1.,G800GL831M99M0)          ! ERROR: PROBE WAS IN NON-CONTACT AT START OF G23
```

```
(IFT,L1021=2.,G800GL831M99M0)          ! ERROR: PROBE DID NOT LOSE CONTACT WITH SURFACE
G800GL831M99M0                         ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,237)YL842
(PAR,L838=-2.*L836,L1020=0)
G91G22YL838FL834M49
M48
(LBL,240)(IFT,L1020=0,(GTO,240))
(IFT,L1021=8.,(GTO,245))
(IFT,L1021=1.,G800GL843M99M0)          ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)          ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                         ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,245)(PAR,L845=L845+1.)
(IFT,L845<>1.,(GTO,250))
(PAR,L848=L3043)
G90YL851FL806
(PAR,L832=90.,L825=-L825,L836=L833)
(GTO,220)
(LBL,250)(PAR,L849=L3043,L825=L849+(L848-L849)/2.)
G90YL825FL806
(IFT,NOT(DEF(L818)),(GTO,260))
(PAR,L1004=(L818-L848+L849)/2."Y BALL RADIUS",L1005=L818"Y SURFACE SEPARATION")
(DLP,L1020,L1021)
GL843G800
(LBL,260)(PAR,L1005=L848-L849+L1004*2."Y SURFACE SEPARATION")
(IFT,NOT(DEF(L814)),(GTO,270))
(PAR,L814=L814*10000.)
(FWR,"N",L814I)
(LBL,270)(PAR,L1005=INT(L1005*L858+.5)/L858)
(FWR,"Y SURFACE SEPARATION=",L1005)
(FWR,"Y CENTER=",Y)
(FWR,)
(DLP,L1020,L1021)
GL843G800
```

G27 R, A (, Z, E) PROBE CYCLE

```
(LBL,300)(PAR,L850=X,L851=Y,L852=Z,L824=L818*COS(L801),L825=L818*SIN(L801))
(PAR,L857=L801)
(IFT,NOT(DEF(L805)),(GTO,420))
(GTO,700)
```

G27 X, Y (, R) PROBE CYCLE

```
(LBL,400)(PAR,L850=X,L851=Y)
(IFT,L843=91.,(GTO,405))
(PAR,L824=L824-X,L825=L825-Y)
(LBL,405)(PAR,L857=ATN(ABS(L825/L824)))
(IFT,L824>=0,(GTO,410))
(IFT,L825>=0,(GTO,415))
(PAR,L857=180.+L857)
(GTO,420)
(LBL,410)(IFT,L825>=0,(GTO,420))
(PAR,L857=360.-L857)
```

```
(GTO,420)
(LBL,415)(PAR,L857=180.-L857)
(LBL,420)(PAR,L835=L833*COS(L857),L832=INT((360.-L857)+.5))
(PAR,L836=L833*SIN(L857))
(LBL,430)M19SL832
(PAR,L1020=0)
G91G22XL824YL825FL806
(LBL,435)(IFT,L1020=0,(GTO,435))
(IFT,L1021=8.,(GTO,440))
(IFT,L1021=1.,G800GL843M99M0)          ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)          ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                         ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
(LBL,440)(PAR,L841=L3042-L835,L842=L3043-L836,L1020=0)
G23G90XL841YL842
(LBL,441)(IFT,L1020=0,(GTO,441))
(IFT,L1021=8.,(GTO,442))
(IFT,L1021=1.,G800GL831M99M0)          ! ERROR: PROBE WAS IN NON-CONTACT AT START OF G23
(IFT,L1021=2.,G800GL831M99M0)          ! ERROR: PROBE DID NOT LOSE CONTACT WITH SURFACE
G800GL831M99M0                         ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
(LBL,442)XL841YL842
(PAR,L837=2.*L835,L838=2.*L836,L1020=0)
G91G22XL837YL838FL834M49
M48
(LBL,445)(IFT,L1020=0,(GTO,445))
(IFT,L1021=8.,(GTO,448))
(IFT,L1021=1.,G800GL843M99M0)          ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)          ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                         ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
(LBL,448)(PAR,L845=L845+1.)
(IFT,L845< >1.,(GTO,450))
G90XL850YL851FL806
(PAR,L846=L3042,L848=L3043,L832=L832+180.)
(IFT,L832> =360.,(PAR,L832=L832-360.))
(PAR,L824=-L824,L825=-L825,L835=-L835,L836=-L836)
(GTO,430)
(LBL,450)(PAR,L847=L3042,L849=L3043,L824=L847+(L846-L847)/2.)
(PAR,L825=L849+(L848-L849)/2.)
G90XL824YL825FL806
(PAR,L853=SQRT(SQ(L846-L847)+SQ(L848-L849)))
(IFT,DEF(L801),(GTO,460))
(IFT,NOT(DEF(L818)),(GTO,460))
(PAR,L1003=(L818-L853)/2."ANGULAR BALL RADIUS",L1005=L818"SURFACE SEPARATION")
(DLP,L1020,L1021)
GL843G800
(LBL,460)(PAR,L1005=L853+2.*L1003"SURFACE SEPARATION")
(IFT,NOT(DEF(L814)),(GTO,470))
(PAR,L814=L814*10000.)
(FWR,"N",L814I)
(LBL,470)(PAR,L1005=INT(L1005*L858+.5)/L858)
(FWR,"X,Y SURFACE SEPARATION=",L1005)
(FWR,"X CENTER=",X)
(FWR,"Y CENTER=",Y)
(FWR,)
(DLP,L1020,L1021)
GL843G800
```

G27 X, Z, E PROBE CYCLE

```
(LBL,500)(PAR,L850=X,L852=Z)
(IFT,L843=90.,(GTO,510))
(PAR,L826=L826+Z,L824=L824+X)
(LBL,510)(PAR,L824=L824-(X+L805))
(IFT,L824>0,(GTO,520))
(PAR,L832=180.,L835=L833,L839=-L1003)
(GTO,530)
(LBL,520)(PAR,L832=0)
(PAR,L835=-L833,L839=L1003)
(LBL,530)M19SL832
G91XL805FL806
G90ZL826
(LBL,532)(PAR,L1020=0)
G91G22XL824
(LBL,535)(IFT,L1020=0,(GTO,535))
(IFT,L1021=8.,(GTO,536))
(IFT,L1021=1.,G800GL843M99M0)        ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)        ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                       ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,536)(PAR,L841=L3042+L835,L1020=0)
G23G90XL841
(LBL,537)(IFT,L1020=0,(GTO,537))
(IFT,L1021=8.,(GTO,538))
(IFT,L1021=1.,G800GL831M99M0)        ! ERROR: PROBE WAS IN NON-CONTACT AT START OF G23
(IFT,L1021=2.,G800GL831M99M0)        ! ERROR: PROBE DID NOT LOSE CONTACT WITH SURFACE
G800GL831M99M0                       ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,538)XL841
(PAR,L837=-2.*L835,L1020=0)
G91G22XL837FL834M49
M48
(LBL,540)(IFT,L1020=0,(GTO,540))
(IFT,L1021=8.,(GTO,548))
(IFT,L1021=1.,G800GL843M99M0)        ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)        ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                       ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,548)(PAR,L841=L850+L805)
G90XL841FL806
G90ZL852
(PAR,L845=L845+1.)
(IFT,L845<>1.,(GTO,550))
G90XL850
(PAR,L846=L3042,L832=L832+180.)
(IFT,L832>=360.,(PAR,L832=L832-360.))
(PAR,L824=-L824,L835=-L835,L805=-L805)
M19SL832
G91XL805
G90ZL826
(GTO,532)
(LBL,550)(PAR,L847=L3042,L1005=L846-L847+2.*L839"X SURFACE SEPARATION")
(PAR,L824=L847+(L846-L847)/2.)
G90XL824FL806
(IFT,NOT(DEF(L814)),(GTO,560))
(PAR,L814=L814*10000.)
```

```
(FWR,"N",L814I)
(LBL,560)(PAR,L1005=INT(L1005*L858+.5)/L858)
(FWR,"X SURFACE SEPARATION=",L1005)
(FWR,"X CENTER=",X)
(FWR,)
(DLP,L1020,L1021)
GL843G800
```

G27 Y, Z, E PROBE CYCLE

```
(LBL,600)(PAR,L851=Y,L852=Z)
(IFT,L843=90.,(GTO,610))
(PAR,L826=L826+Z,L825=L825+Y)
(LBL,610)(PAR,L825=L825-(Y+L805))
(IFT,L825>0,(GTO,620))
(PAR,L832=90.,L836=L833,L840=-L1003)
(GTO,630)
(LBL,620)(PAR,L832=270.,L836=-L833,L840=L1003)
(LBL,630)M19SL832
G91YL805FL806
G90ZL826
(LBL,632)(PAR,L1020=0)
G91G22YL825
(LBL,635)(IFT,L1020=0,(GTO,635))
(IFT,L1021=8.,(GTO,636))
(IFT,L1021=1.,G800GL843M99M0)       ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)       ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                      ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
(LBL,636)(PAR,L842=L3043+L836,L1020=0)
G23G90YL842
(LBL,637)(IFT,L1020=0,(GTO,637))
(IFT,L1021=8.,(GTO,638))
(IFT,L1021=1.,G800GL831M99M0)       ! ERROR: PROBE WAS IN NON-CONTACT AT START OF G23
(IFT,L1021=2.,G800GL831M99M0)       ! ERROR: PROBE DID NOT LOSE CONTACT WITH SURFACE
G800GL831M99M0                      ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
(LBL,638)YL842
(PAR,L838=-2.*L836,L1020=0)
G91G22YL838FL834M49
M48
(LBL,640)(IFT,L1020=0,(GTO,640))
(IFT,L1021=8.,(GTO,648))
(IFT,L1021=1.,G800GL843M99M0)       ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)       ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                      ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
(LBL,648)(PAR,L842=L851+L805)
G90YL842FL806
G90ZL852
(PAR,L845=L845+1.)
(IFT,L845< >1.,(GTO,650))
G90YL851
(PAR,L848=L3043,L832=L832+180.)
(IFT,L832>=360.,(PAR,L832=L832-360.))
(PAR,L825=-L825,L836=-L836,L805=-L805)
M19SL832
```

```
G91YL805
G90ZL826
(GTO,632)
(LBL,650)(PAR,L849=L3043,L1005=L848-L849+2.*L840"Y SURFACE SEPARATION")
(PAR,L825=L849+(L848-L849)/2.)
G90YL825FL806
(IFT,NOT(DEF(L814)),(GTO,660))
(PAR,L814=L814*10000.)
(FWR,"N",L814I)
(LBL,660)(PAR,L1005=INT(L1005*L858+.5)/L858)
(FWR,"Y SURFACE SEPARATION=",L1005)
(FWR,"Y CENTER=",Y)
(FWR,)
(DLP,L1020,L1021)
GL843G800
```

G27 X, Y, Z, E PROBE CYCLE

```
(LBL,700)(PAR,L850=X,L851=Y,L852=Z)
(IFT,DEF(L818),(GTO,701))
(IFT,L843=90.,(GTO,702))
(LBL,701)(PAR,L841=L824,L842=L825,L824=L824+X,L825=L825+Y)
(IFT,L843=90.,(GTO,705))
(PAR,L826=L826+Z)
(GTO,705)
(LBL,702)(PAR,L841=L824-X,L842=L825-Y)
(LBL,705)(PAR,L853=SQRT(SQ(L841)+SQ(L842)))
(IFT,DEF(L818),(GTO,720))
(PAR,L857=ATN(ABS(L842/L841)))
(IFT,L841>=0,(GTO,710))
(IFT,L842>=0,(GTO,715))
(PAR,L857=180.+L857)
(GTO,720)
(LBL,710)(IFT,L842>=0,(GTO,720))
(PAR,L857=360.-L857)
(GTO,720)
(LBL,715)(PAR,L857=180.-L857)
(LBL,720)(PAR,L841=L805*COS(L857),L842=L805*SIN(L857),L832=INT((360.-L857)+.5))
(IFT,L853>L805,(GTO,725))
(PAR,L832=L832+180.,L857=L857+180.)
(IFT,L832>=360.,(PAR,L832=L832-360.,L857=L857-360.))
(PAR,L839=-L1003)
(GTO,727)
(LBL,725)(PAR,L839=L1003)
(LBL,727)M19SL832
G91XL841YL842FL806
G90ZL826
(LBL,730)(PAR,L835=L833*COS(L857),L836=L833*SIN(L857),L824=L824-X,L825=L825-Y)
(LBL,740)(PAR,L1020=0)
G91G22XL824YL825
(LBL,741)(IFT,L1020=0,(GTO,741))
(IFT,L1021=8.,(GTO,742))
(IFT,L1021=1.,G800GL843M99M0)        ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)        ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
```

```
G800GL843M99M0                          ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
(LBL,742)(PAR,L837=L3042-L835,L838=L3043-L836,L1020=0)
G23G90XL837YL838
(LBL,743)(IFT,L1020=0,(GTO,743))
(IFT,L1021=8.,(GTO,744))
(IFT,L1021=1.,G800GL831M99M0)           ! ERROR: PROBE WAS IN NON-CONTACT AT START OF G23
(IFT,L1021=2.,G800GL831M99M0)           ! ERROR: PROBE DID NOT LOSE CONTACT WITH SURFACE
G800GL831M99M0                          ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
(LBL,744)XL837YL838
(PAR,L837=2.*L835,L838=2.*L836,L1020=0)
G91G22XL837YL838FL834M49
M48
(LBL,745)(IFT,L1020=0,(GTO,745))
(IFT,L1021=8.,(GTO,746))
(IFT,L1021=1.,G800GL843M99M0)           ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)           ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                          ! ERROR: IMPROPER PROBE STATUS< >1.,2.,OR 8.
(LBL,746)(PAR,L837=L850+L841,L838=L851+L842)
G90XL837YL838FL806
G90ZL852
(PAR,L845=L845+1.)
(IFT,L845< >1.,(GTO,750))
G90XL850YL851
(PAR,L846=L3042,L848=L3043,L832=L832+180.)
(IFT,L832>=360.,(PAR,L832=L832-360.))
(PAR,L841=-L841,L842=-L842,L824=-L824,L825=-L825,L835=-L835,L836=-L836)
M19SL832
G91XL841YL842
G90ZL826
(GTO,740)
(LBL,750)(PAR,L847=L3042,L849=L3043,L824=L847+(L846-L847)/2.)
(PAR,L825=L849+(L848-L849)/2.)
G90XL824YL825
(PAR,L853=SQRT(SQ(L846-L847)+SQ(L848-L849)))
(PAR,L1005=L853+2.*L839"SURFACE SEPARATION")
(IFT,NOT(DEF(L814)),(GTO,760))
(PAR,L814=L814*10000.)
(FWR,"N",L814I)
(LBL,760)(PAR,L1005=INT(L1005*L858+.5)/L858)
(FWR,"X,Y SURFACE SEPARATION=",L1005)
(FWR,"X CENTER=",X)
(FWR,"Y CENTER=",Y)
(FWR,)
(DLP,L1020,L1021)
GL843G800
```

G27 B ROTARY AXIS PROBE CYCLE

```
(LBL,800)(PAR,L850=INT(B*1000.+.5)/1000.)
(IFT,L850=0,(PAR,L850=360.))
(IFT,G70,(GTO,810))
(PAR,L856=L3083)
(GTO,820)
(LBL,810)(PAR,L856=L3083)
```

```
(LBL,820)(IFT,(L3002-L3033)>L856,(GTO,840))
(PAR,L855=-1.)
(IFT,L802<0,(GTO,850))
(LBL,830)(PAR,L832=180.,L835=1.*L855,L854=-1.*L855)
(GTO,860)
(LBL,840)(PAR,L855=1.)
(IFT,L802<0,(GTO,830))
(LBL,850)(PAR,L832=0,L835=-1.*L855,L854=1.*L855)
(LBL,860)(IFT,L843=91.,(GTO,875))
(IFT,L802>0,(GTO,870))
(PAR,L802=B-L802)
(IFT,L802<0,(PAR,L802=L802+360.))
(PAR,L802=-L802)
(GTO,875)
(LBL,870)(PAR,L802=L802-B)
(IFT,L802<0,(PAR,L802=L802+360.))
(LBL,875)M19SL832
(PAR,L1020=0)
G91G22BL802FL806
(LBL,880)(IFT,L1020=0,(GTO,880))
(IFT,L1021=8.,(GTO,882))
(IFT,L1021=1.,G800GL843M99M0)          ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)          ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                         ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,882)(PAR,L841=(L3045+L835)*-L854,L1020=0)
G23G90BL841
(LBL,883)(IFT,L1020=0,(GTO,883))
(IFT,L1021=8.,(GTO,884))
(IFT,L1021=1.,G800GL831M99M0)          ! ERROR: PROBE WAS IN NON-CONTACT AT START OF G23
(IFT,L1021=2.,G800GL831M99M0)          ! ERROR: PROBE DID NOT LOSE CONTACT WITH SURFACE
G800GL831M99M0                         ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,884)BL841
(PAR,L837=2.*ABS(L835)*L854,L1020=0)
G91G22BL837FL834M49
M48
(LBL,885)(IFT,L1020=0,(GTO,885))
(IFT,L1021=8.,(GTO,887))
(IFT,L1021=1.,G800GL843M99M0)          ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)          ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                         ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,887)(PAR,L845=L845+1.)
(IFT,L845<>1.,(GTO,890))
(PAR,L846=L3045,L850=L850*-L854)
G90BL850FL806
(PAR,L832=L832+180.,L802=-L802,L835=-L835,L854=-L854)
(GTO,875)
(LBL,890)(PAR,L847=L3045,L802=L847+(L846-L847)/2.)
(IFT,ABS(L802-L846)>90.,(PAR,L802=L802-180.))
(IFT,L802<0,(PAR,L802=360.+L802))
(PAR,L802=L802*-L854)
G90BL802FL806
(IFT,NOT(DEF(L814)),(GTO,895))
(PAR,L814=L814*10000.)
(FWR,"N",L814I)
(LBL,895)(FWR,"B ALIGNMENT=",B)
```

```
(FWR,)
(DLP,L1020,L1021)
GL843G800
```

G27 C ROTARY AXIS PROBE CYCLE

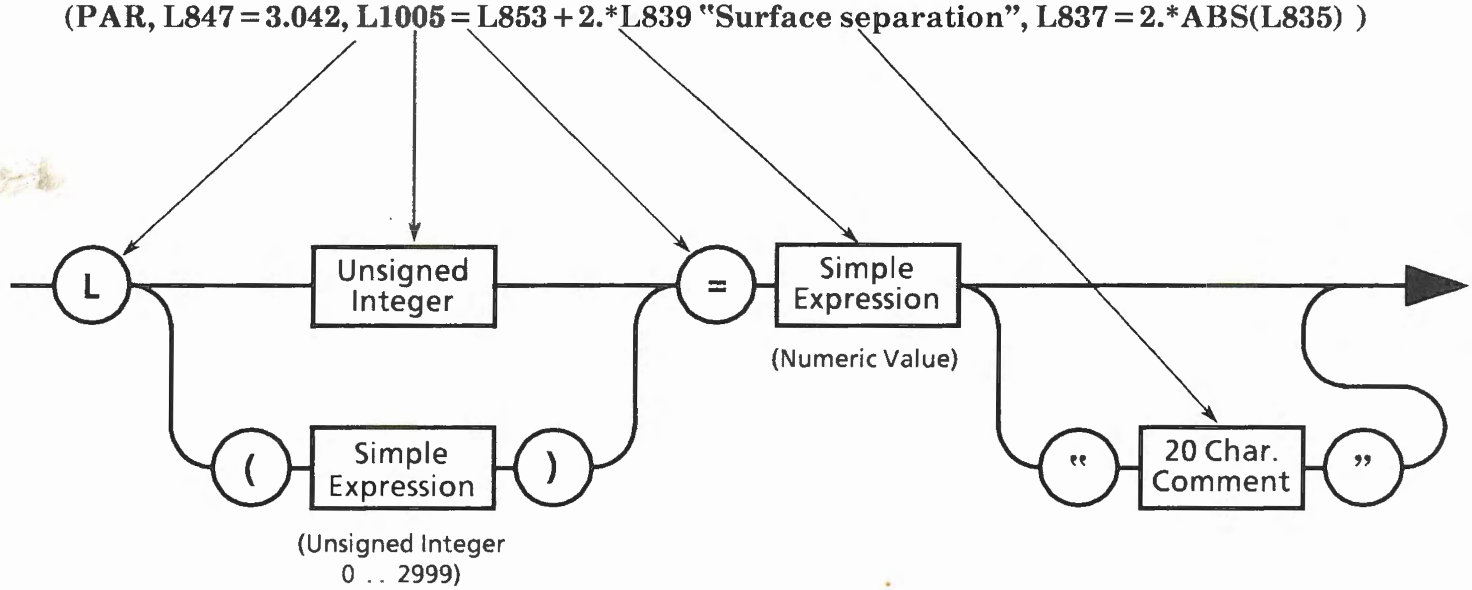
```
(LBL,900)(IFT,G70,(GTO,910))
(PAR,L824=L3000-L3084,L825=L3001-L3085)
(GTO,920)
(LBL,910)(PAR,L824=L3000-L3084,L825=L3001-L3085)
(LBL,920)(PAR,L850=INT(C*1000.+.5)/1000.,L832=ATN(ABS(L825/L824)))
(IFT,L850=0,(PAR,L850=360.))
(IFT,L824>=0,(GTO,930))
(IFT,L825>=0,(GTO,935))
(PAR,L832=180.+L832)
(GTO,940)
(LBL,930)(IFT,L825>=0,(GTO,940))
(PAR,L832=360.-L832)
(GTO,940)
(LBL,935)(PAR,L832=180.-L832)
(LBL,940)(IFT,L803<0,(GTO,950))
(PAR,L832=L832+90.,L835=-1.,L854=1.)
(GTO,960)
(LBL,950)(PAR,L832=L832-90.,L835=1.,L854=-1.)
(LBL,960)(IFT,L832<0,(PAR,L832=360.+L832))
(IFT,L832>360.,(PAR,L832=360.-L832))
(IFT,L843=91.,(GTO,972))
(IFT,L803>0,(GTO,970))
(PAR,L803=C-L803)
(IFT,L803<0,(PAR,L803=L803+360.))
(PAR,L803=-L803)
(GTO,972)
(LBL,970)(PAR,L803=L803-C)
(IFT,L803<0,(PAR,L803=L803+360.))
(LBL,972)M19SL832
(PAR,L1020=0)
G91G22CL803FL806
(LBL,974)(IFT,L1020=0,(GTO,974))
(IFT,L1021=8.,(GTO,975))
(IFT,L1021=1.,G800GL843M99M0)          ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)          ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                         ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,975)(PAR,L841=(L3046+L835)*-L854,L1020=0)
G23G90CL841
(LBL,976)(IFT,L1020=0,(GTO,976))
(IFT,L1021=8.,(GTO,977))
(IFT,L1021=1.,G800GL831M99M0)          ! ERROR: PROBE WAS IN NON-CONTACT AT START OF G23
(IFT,L1021=2.,G800GL831M99M0)          ! ERROR: PROBE DID NOT LOSE CONTACT WITH SURFACE
G800GL831M99M0                         ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,977)CL841
(PAR,L837=2.*ABS(L835)*L854,L1020=0)
G91G22CL837FL834M49
M48
(LBL,979)(IFT,L1020=0,(GTO,979))
```

```
(IFT,L1021=8.,(GTO,980))
(IFT,L1021=1.,G800GL843M99M0)          ! ERROR: PROBE WAS IN CONTACT AT START OF G22
(IFT,L1021=2.,G800GL843M99M0)          ! ERROR: PROBE DID NOT MAKE CONTACT WITH SURFACE
G800GL843M99M0                         ! ERROR: IMPROPER PROBE STATUS<>1.,2.,OR 8.
(LBL,980)(PAR,L845=L845+1.)
(IFT,L845<>1.,(GTO,990))
(PAR,L846=L3046,L850=L850*-L854)
G90CL850FL806
(PAR,L832=L832+180.)
(IFT,L832>360.,(PAR,L832=L832-360.))
(PAR,L803=-L803,L835=-L835,L854=-L854)
(GTO,972)
(LBL,990)(PAR,L847=L3046,L803=L847+(L846-L847)/2.)
(IFT,ABS(L803-L846)>90.,(PAR,L803=L803-180.))
(IFT,L803<0,(PAR,L803=360.+L803))
(PAR,L803=L803*-L854)
G90CL803FL806
(IFT,NOT(DEF(L814)),(GTO,1000))
(PAR,L814=L814*10000.)
(FWR,"N",L814I)
(LBL,1000)(FWR,"C ALIGNMENT=",C)
(FWR,)
(DLP,L1020,L1021)
GL843G800
(END)
```

APPENDIX F

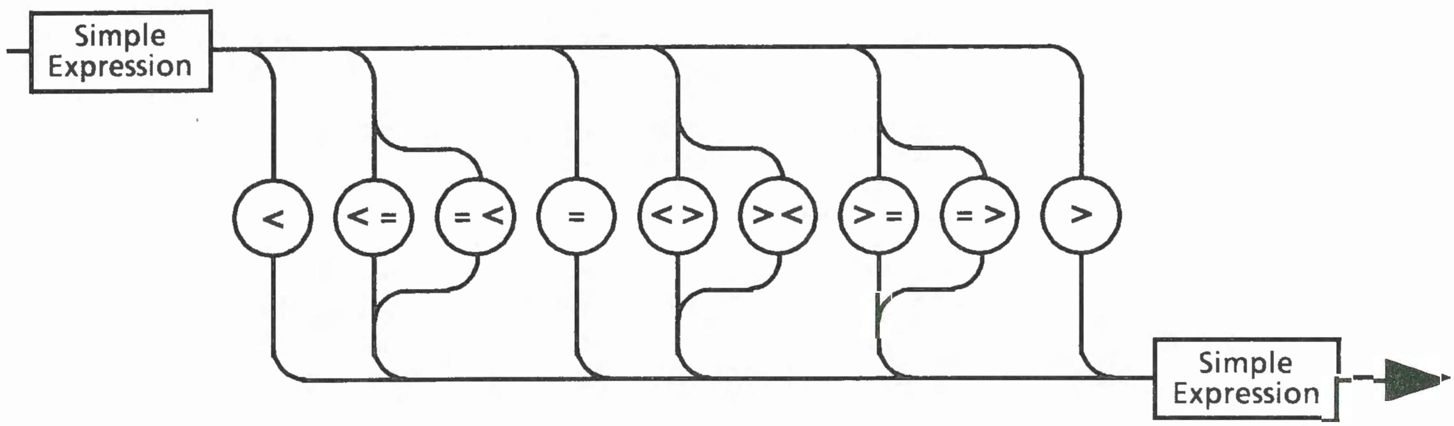
PARAMETER ASSIGNMENT EXAMPLE
AND
EXPRESSION EVALUATION SYNTAX DIAGRAMS

(PAR, L847 = 3.042, L1005 = L853 + 2.*L839 "Surface separation", L837 = 2.*ABS(L835))

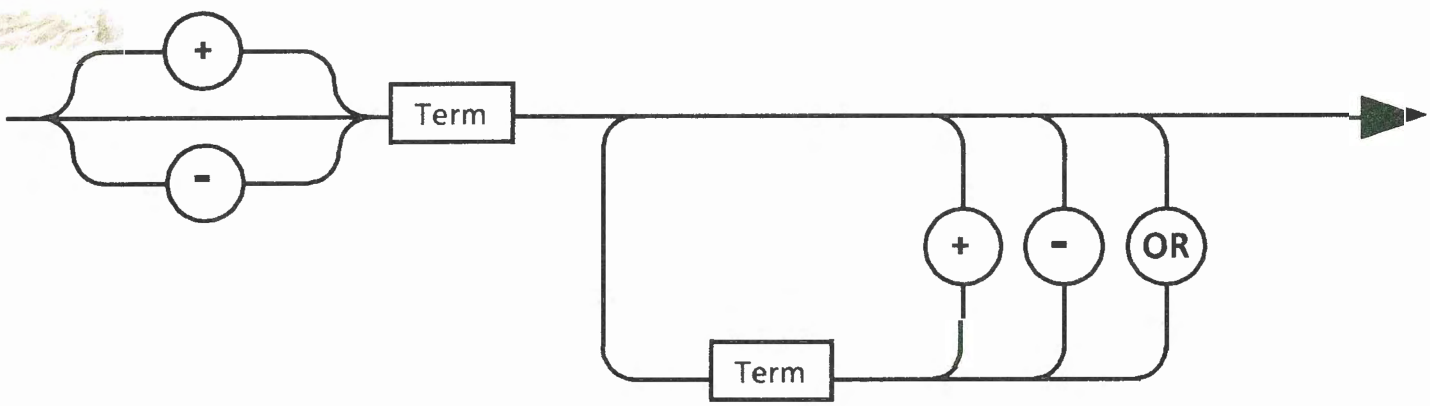


EXPRESSION EVALUATION SYNTAX DIAGRAMS

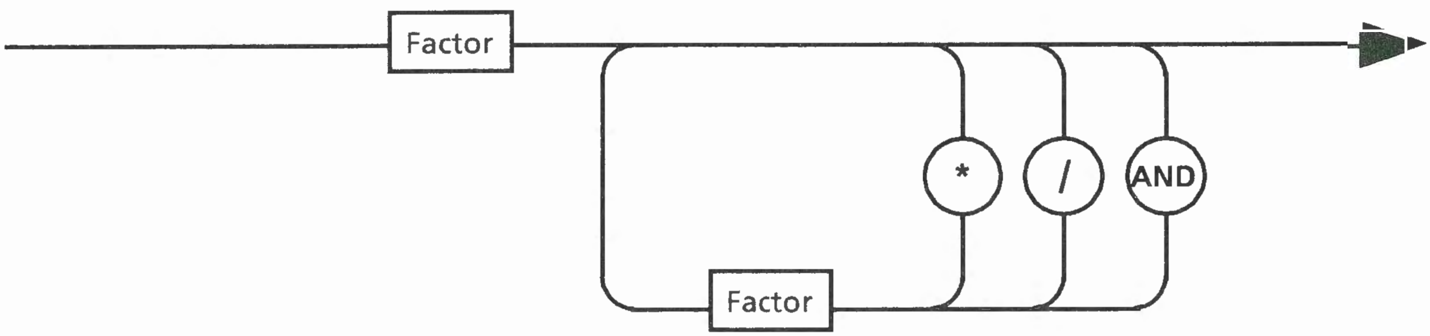
EXPRESSION



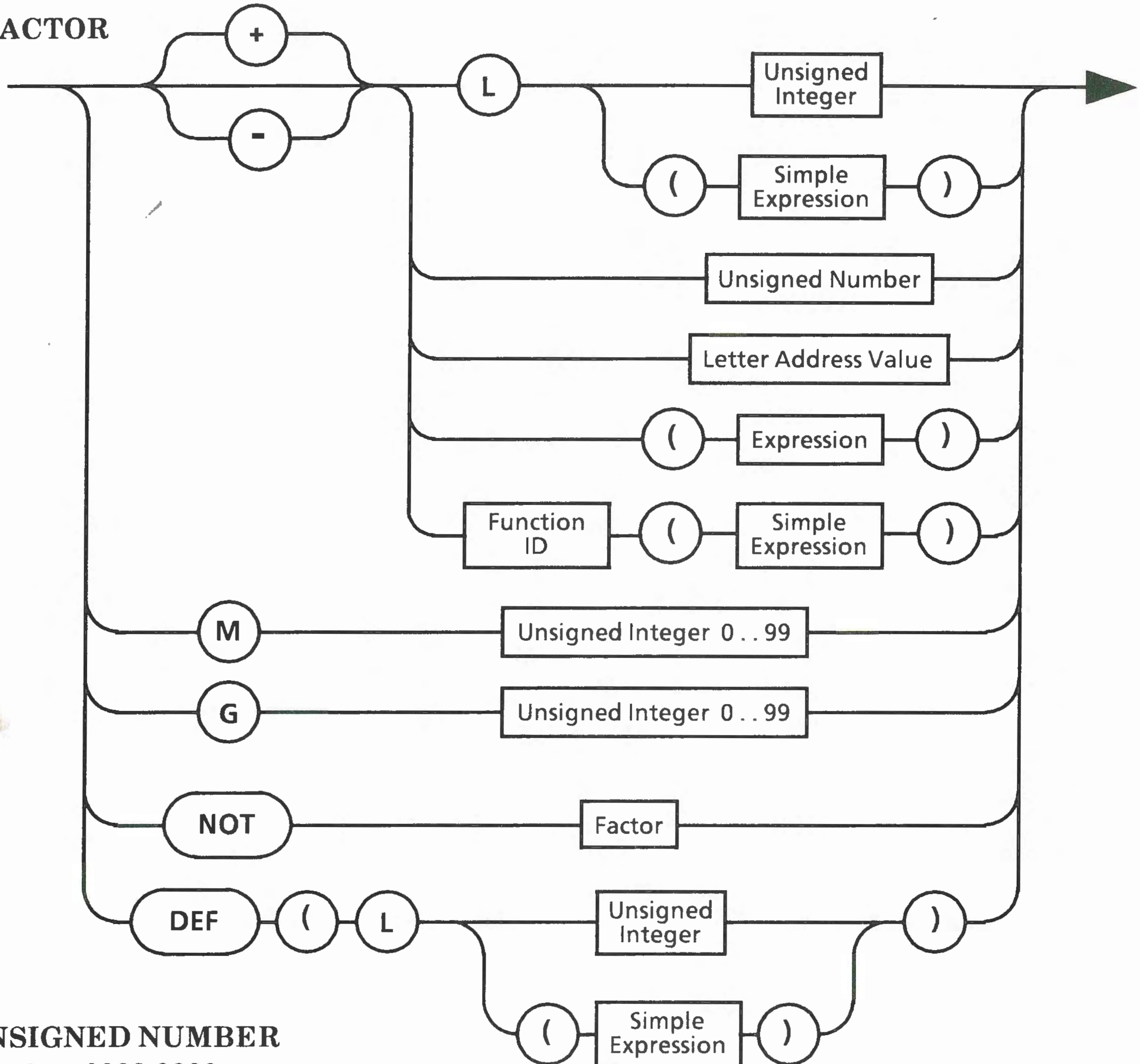
SIMPLE EXPRESSION



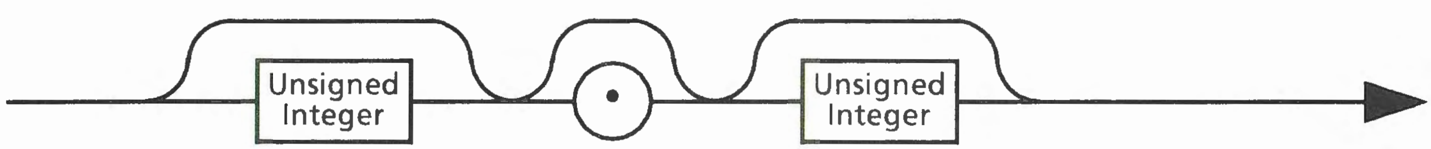
TERM



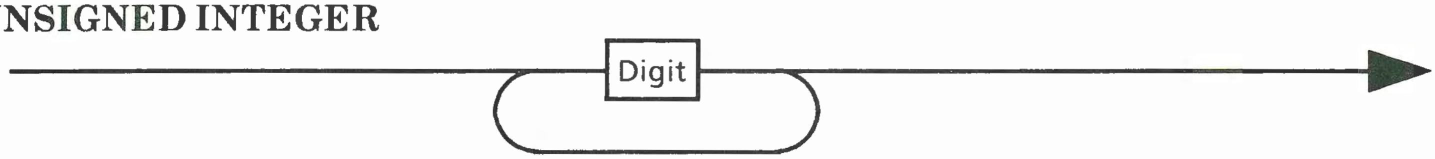
FACTOR



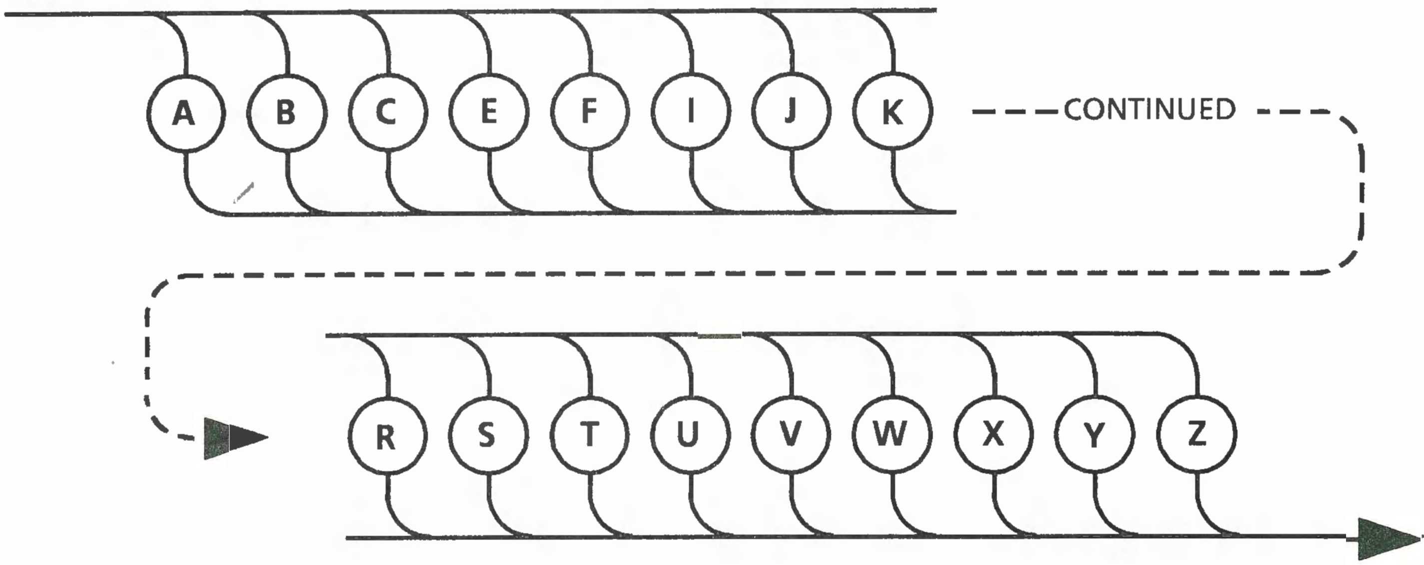
UNSIGNED NUMBER
0 . . 9999.9999



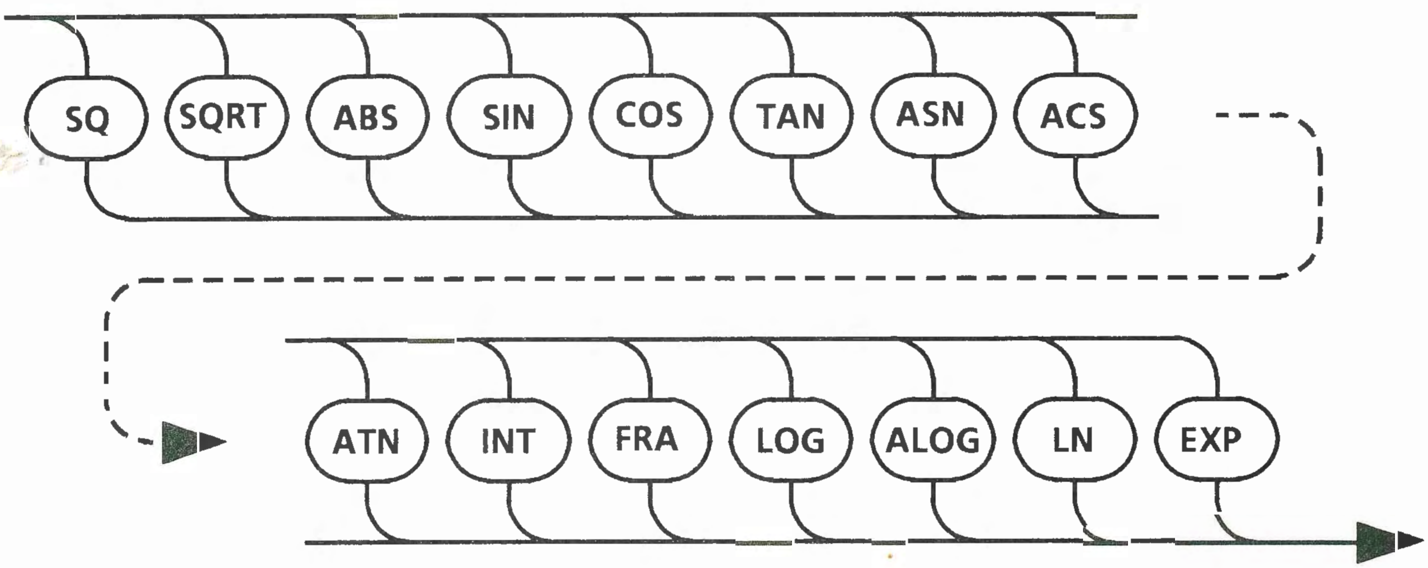
UNSIGNED INTEGER



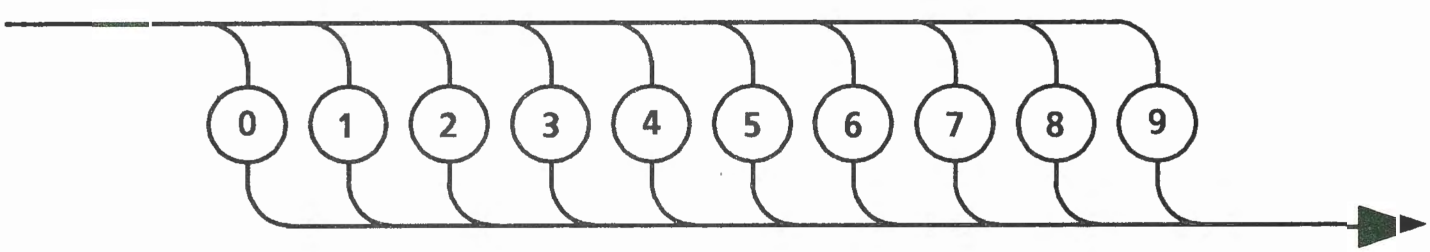
LETTER ADDRESS VALUE



FUNCTION ID



DIGIT



APPENDIX G

SUMMARY OF TYPE II DATA STATEMENTS

TYPE II MNEMONICS (ALPHABETIC ORDER)

(ACP) - Specify & Modify Adaptive Control
(CER) - Continue Part Program Execution Upon Device Errors
(CLF) - Close File
(CLS) - Call Subroutine
(CPY) - Copy a File
(DAT) - Update Date and Time
(DFC) - Define a Cycle
(DFS) - Define a Subroutine
(DLF) - Delete File
(DLP) - Delete Parameter
(END) - End Statement
(EVT) - Record Part Program Event
(FWR) - Write to File
(FXC) - Fixture Offset Table Load
(GTO) - Go To Label
(IFT) - Conditional "IF True"
(LBL) - Label Definition
(MIR) - Axis Inversion - Mirror Image
(MSG) - Programmable Operator Messages
(NAM) - Name a Part Program
(OIF) - Naming Spindle Probe Inspection File
(OPF) - Open File
(PAR) - Parameter Assignment Statement
(PRT) - Print File
(RCP) - Copy a file to or from a remote station
(SER) - Stop Part Program Execution Upon Device Errors
(SET) - Set Output
(TLD) - Load a Tool Table with Tool Parameter
(U88) - Programmable G88 Boring Cycle
(UAT) - Select Alternate Tool Data
(UCY) - Maximum Part Program Execution Time - FMS
(UFX) - Calculate Fixture Offset & Load Table
(ULF) - Loads Fixture Offset File From Mass Storage Device
(USF) - Saves Fixture Offset Table to a Mass Storage Device
(UTL) - Identifies all Tools used in a Part Program - FMS
(WAI) - Wait on Input
(ZTB) - Zero Table - T, H, Table

APPENDIX H

DEFINING CIRCULAR PLANES with SIX AXES - G20, G21

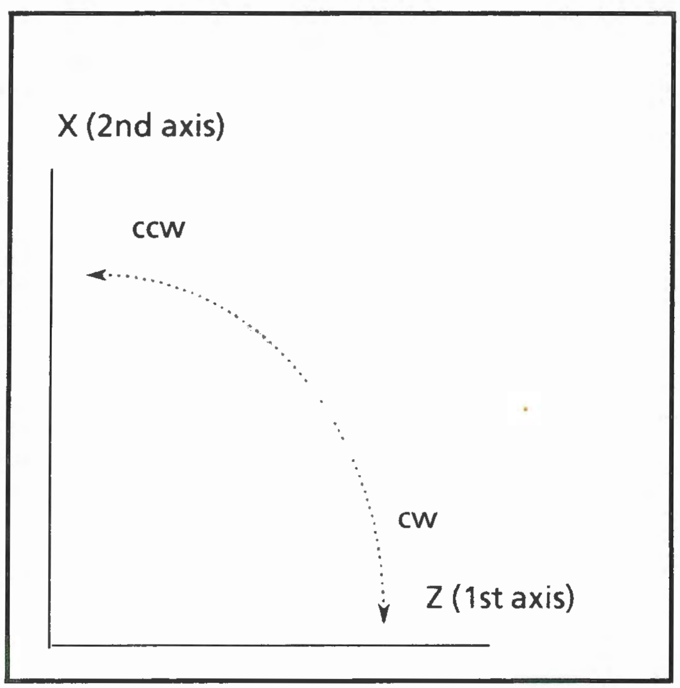
DEFINING CIRCULAR PLANES WITH 6 AXES

The following section describes a way of defining the planes of circular blocks when a machine has 6 linear axes. Plane selection currently is defined in the following way:

X,Y plane G17
X,Z plane G18
Y,Z plane G19

Now, three additional coaxes (U,V,W) exist where U is parallel to the X axis, V is parallel to the Y axis, and W is parallel to the Z axis, then additional information is needed to specify which two axes are involved in the circular move. This additional information is now provided by two additional G functions (G20 and G21).

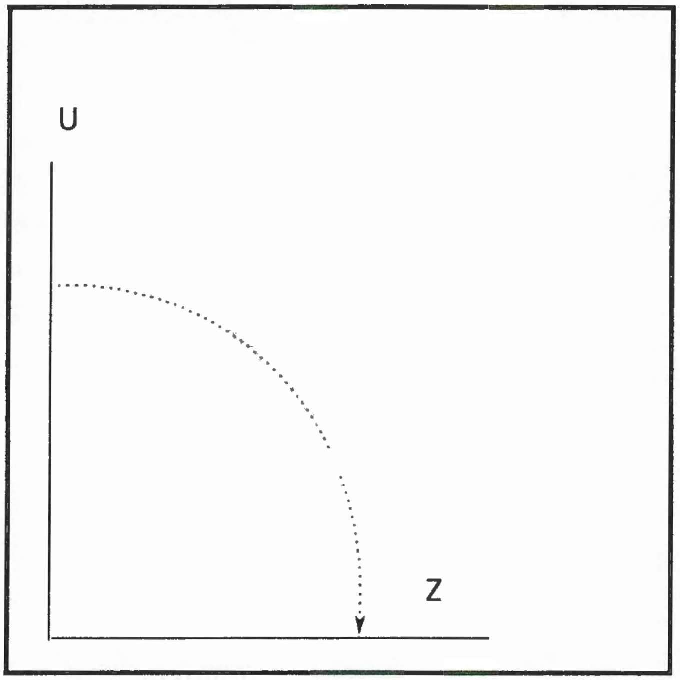
Recall that when an axis plane is specified plane think of one axis as the "first axis" and the other as the "second axis". For example, in G18 Z is the first axis and X is the second axis. This concept is useful for defining the direction of circular moves (i.e clockwise (G02) and counterclockwise (G03)). Furthermore, plot a plane so that the first axis is directed to the right and the second axis is directed upward. The G18 plane is shown below along with the defined directions of clockwise and counterclockwise.



The concept of clockwise only makes sense if one uses a consistent definition of axis direction.

For example, to specify a circle including one or more of the additional axes (U,V,W), use G20 and G21 to indicate which axes have been replaced by their corresponding parallel coaxes. G20 implies replacement of the first axis, G21 implies replacement of the second axis, and G20 and G21 in the same block implies that both axes are replaced.

Consider a G02 block in the Z,U plane. Since the second axis (X) has been replaced by its corresponding parallel coaxis (U), the modal state should be G21 as well as G18. The direction of the circle is shown below.



Note that G20 and G21 do not define cutting planes. They only selectively replace the circular axes with their parallel coaxes. G20 and G21 must be used in conjunction with one of the circular planes G17, G18, or G19.

To return to the basic axes, it is necessary to restate the circular plane without a G20 or a G21, or hit [CLEAR ALL LOGIC]. For example, if it is desired to cut a circle with axes Z and X, after cutting circles in Z and U, the programmer must eliminate the G21 state by restating G18 without G21.

INDEX

G

J

K

L

M

Q

R

Y

Z

Kearney & Trecker Corporation

A Cross & Trecker Company