Anilam

Anilam Electronics Corporation

CRUSADER II L

PROGRAMMING MANUAL

PART NO. 700-121

PART NO. 700-121

CRUSADER IIL PROGRAMMING MANUAL

TABLE OF CONTENTS

PART TWO: PROGRAMMING WITH THE CRUSADER IIL

BASIC SPECIFICATIONS

NO.		SPECIFICATION
1.	CONTROLLED AXES	Two Axes (X & Z).
2.	SIMUTANEOUSLY CONTROLLABLE AXES	Both Axes in RAPID, FEED & MANUAL operation.
3.	INCREMENT SYSTEM	Input: 0.0001"/.002mm, 0005"/.01mm
4.	MAXIMUM PROGRAMMABLE DIMENSIONS	+- 999.9999 inches +- 9999.999 mm
5.	DECIMAL POINT PROGRAMMING	Data can be input with decimal point.
6.	RAPID TRAVERSE RATE	RAPID traverse at 240 inches per minute (IPM)
7.	FEED RATE	0.0001" IPR 0.1" to 80" IPM
8.	ABSOLUTE/INCREMENTAL PROGRAMMING	Axis moves may be absolute or incremental, but may not be mixed in the same event.
9.	AXIS ZEROING	Any Axis may be zeroed at its present location, or preset to any dimension.
10.	POSITIONING IN RAPID	A RAPID command moves the machine at RAPID Traverse to the position specified. The machine deaccelerates to a stop, and an in position check is made before execution continues.
11.	LINEAR INTERPOLATION	A straight line move is made to the next X, Z, or XZ position at the programmed feedrate.
12.	CIRCULAR INTERPOLATION	Any ARC between 0 and 360 degrees may be programmed, CW or CCW.
13.	DWELL	Execution of a DWELL command stops the program from running until the start button is pressed. A timed DWELL will automatically restart. Time may be 0.1 to 999 9 seconds.

14.	CANNED CYCLES (G CODES)	A variety of miscellaneous canned cycles are accessed by a 2-digit G Code.
15.	AUXILIARY (AUX CODES)	Various features (absolute machine zero shift, turn on software limits etc.) are activated by a 4 digit AUX code.
16.	TOOL FUNCTION	Tool offsets are called by a 2-digit tool number (01-32). Tool offsets are set by a 4-digit number (2001 through 2032).
17.	HOLD	The movement of either one or both Axes may be stopped & resumed by using the HOLD & START buttons.
18.	SINGLE STEP	Commands may be executed event by event or move by move.
19.	EMERGENCY STOP button	When the EMERGENCY STOP button is pushed, both Servo motors and the spindle motor will stop.
20.	OVERTRAVEL	Both Axes are equipped with Limit switches, which will stop all Axes movement and show an error code when activated.
21.	MANUAL, FEED OR RAPID MOVES	Moves may be made by pressing MOVES and holding down the X+, Z+, X-or Z- keys.
22.	JOG MOVES	Positioning in incremental amounts can be made as follows: Jog 1: 0.0002 " or 0.002 MM. Jog 2: 0.002" or 0.02 MM. Jog 3: 0.02" or 0.2 MM.
23.	DO LOOP	A sequence of events may be repeated from 0 to 999 times.
24.	SUBROUTINES	A subroutine may call a subroutine to a nested depth of 32 levels.
25.	PROGRAM PROTECTION	10 minutes battery backup of program and position with ability to use program saver.

OPTIONS

1.	LINE VOLTAGE	110 VAC or 220 VAC.
2.	RS 232	Links control to other peripherals.

ABBREVIATIONS USED IN THIS MANUAL

ARC	Circular move
AUX	Auxiliary
CHAM	Chamfer
CCW	Counter Clockwise (Directions of an ARC)
CW	Clockwise (Directions of an ARC)
DISP POSN	Display Position
DO	Do Loop
FA	Feed Absolute
FI	Feed Incremental
INCH	Inches
INCR	Incremental
IN POSN	In Position
IPM	Inches per minute
IPR	Inches per revolution
MM	Millimeters
RAD	Radius
RA	Rapid Absolute
RI	Rapid Incremental
RPM	Revolutions per minute
SUBR	Subroutine
TPI	Threads per inch
V	Variable (V Code)

Explanations of the above terminology will become apparent when reading the manual.

QUICK REFERENCE LIST
for primary G CODES and AUXILIARY CODES

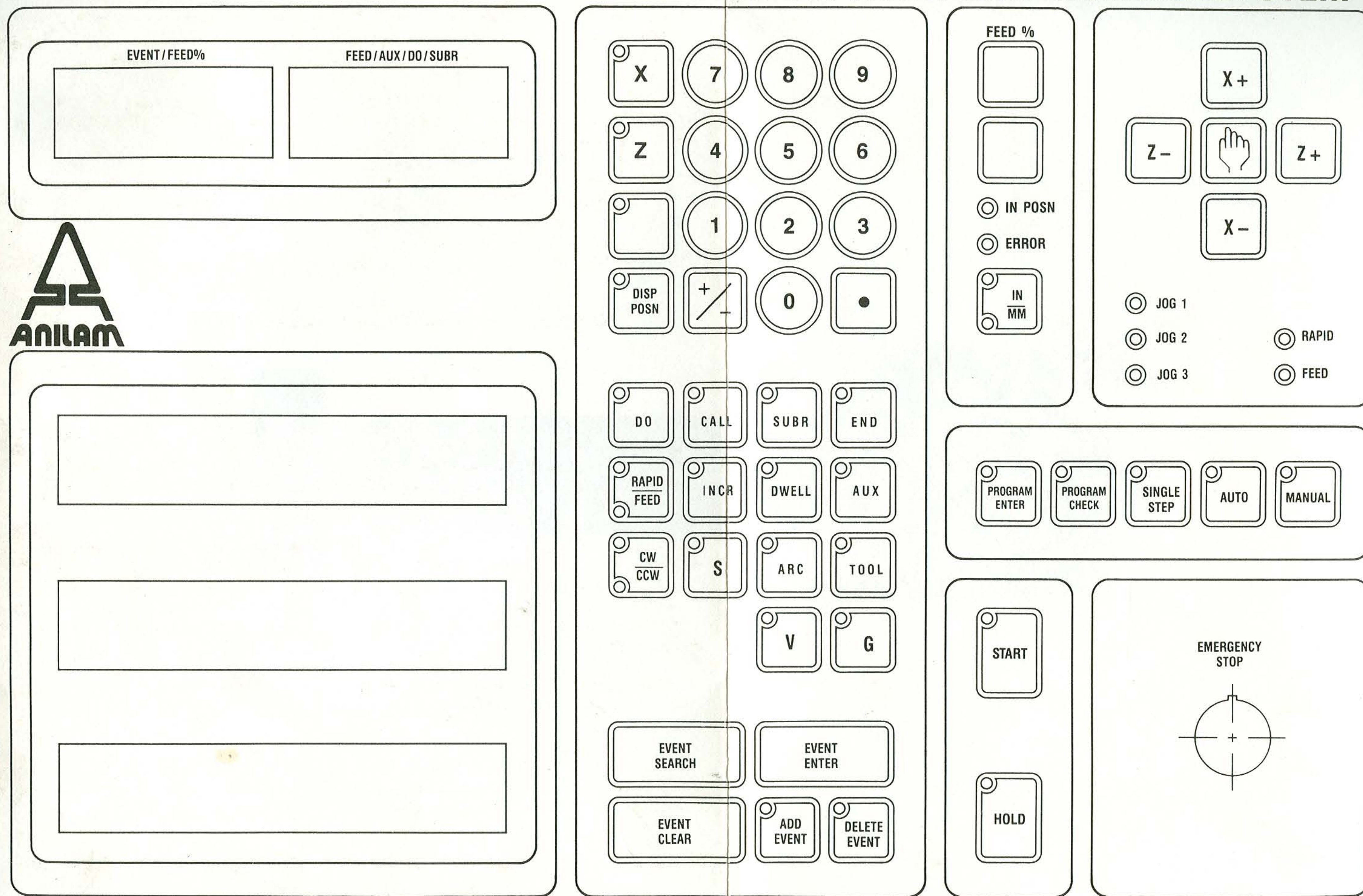
G-CODE_REFERENCE_LIST

G40:	Turns off tool nose radius compensation
G41:	Tool nose radius compensation (left)
G42:	Tool nose radius compensation (right)
G81:	Rough turning/boring canned cycle
G82:	Rough facing canned cycle
G84:	External/Internal longitudinal compound threading canned cycle
G85:	Face plunge threading canned cycle
G86:	External/Internal longitudinal compound threading canned cycle
G87:	Face compound threading canned cycle
G88:	External/Internal longitudinal grooving canned cycle
G89:	Face Grooving canned cycle
G94:	Inches/mm per minute programming
G95:	Inches/mm per revolution programming

AUXILIARY_CODE_REFERENCE_LIST

AUX 1000:	Turn on continuous path
AUX 1110:	Cancel software limits
AUX 1111:	Set software limits
AUX 1400:	FEED % override
AUX 1401:	Feed and RAPID % override
AUX 1900:	SINGLE STEP by event
AUX 1901:	SINGLE STEP by X or Z move
AUX 2000:	Cancel continuous path
AUX 0003:	Spindel CW
AUX 0004:	Spindel CCW
AUX 0005:	Spindel stop
AUX 0007:	Coolant on
AUX 0008:	Coolant off
AUX 1000:	Turns on continuous path programming
AUX 1101:	Zero shift
AUX 1110:	Software limits off (outer)
AUX 1112:	Enable software limits (outer)
AUX 1113:	Set software limits (outer)
AUX 1114:	Disable software limits (inner)
AUX 1116:	Enable software limits (inner)
AUX 1117:	Set software limits (inner)
AUX 1160:	Disable backlash compensation
AUX 1161:	Set & enable backlash compensation (x axis)
AUX 1162:	Set & enable backlash compensation (z axis)
AUX 1400:	Feedrate % override active
AUX 1401:	Feed & rapid % override active
AUX 1500:	Unlock program enter
AUX 1501:	Lock program enter
AUX 1605:	Beeper off
AUX 1606:	Beeper on
AUX 1900:	Single step one event
AUX 1901:	Single step one move
AUX 2000:	Turns off continuous path programming

LATHE MICROPROCESSOR CONTROL SYSTEM



SAFETY NOTICE

For clarity purposes, the drawings in this manual have been illustrated without protective guards in place.

When applying this control in a retrofit application, you are converting a manually controlled machine. Consideration must be given to meeting applicable safety and guarding standards. These standards can be obtained from:

1. "Safety Requirements for the Construction, Care and Use of Drilling, Milling and Boring Machines", ANSI Standards Institute, 1430 Broadway, New York, New York 10018.
2. "Point of Operation Guarding" Section 1910.212, U.S. Occupational Safety and Health Administration.
3. "Power Transmission Guarding" Section 1910.219, U.S. Occupational Safety and Health Administration.

Publications 2 and 3 are both available from the U.S. Department of Labor.

(Information in this manual is subject to change without notice.)

INTRODUCTION & GENERAL NOTES

The Crusader IIL Control you have purchased utilizes the latest advances in microprocessor technology. This, together with Anilam's machine shop orientation, enables you to begin production almost immediately. Very little schooling in programming techniques will be required.

The keyboard is easy to understand and simple to use. It is made of a sturdy, chip-resistant material. The switches are completely sealed and work via a newly devised method, combining the best of touch tone, with pressure techniques. When a button is depressed a tone is generated, telling the operator he has made the contract as required.

All programming may be carried out at the machine, or alternatively, an off-line programming system may be purchased.

NOTE:

As with any CNC machine, the operator should be familiar with correct machining practices and machine shop terminology. This manual has been written under these assumptions.

This manual has been arranged in two parts. The first part covers all functions of the Crusader IIL control.

The second part contains example programs and practical applications that are specific to the tooling arrangement on the Anilam Lathemate. This section should be used as a training aid by the new programmer. It is graduated from simple to more complex programs and covers basic knowledge (such as coordinate systems) that will be needed for programming.

We encourage you to study, enter and dry-run each example program. Use the front section for quick references to any function or command of which you are unfamiliar.

CNC programming is a never-ending learning experience. Once the basics are mastered, each new job will add to your expertise. Our toll-free telephone service is available to help you with any programming problem, whether elementary or highly complex. Please take advantage of this service:

TOLL-FREE SERVICE HOT LINE
1-800-327-6340

For your own safety and the protection of your equipment, always remember the programmer's cardinal rule: **Dry run every program thoroughly with one hand on the EMERGENCY STOP switch!**

TURNING THE SYSTEM ON

POWER SUPPLY

Power is supplied to the Crusader IIL through a 115 volt (230 volt optional) power supply cable. The cable outlet from the console is located on the lower right side when facing the control.

Power must be constant at a minimum of 105 volts AC and it must be properly grounded. The power cord can now be plugged in to an AC power source.

CRUSADER IIL START-UP

To gain access to the back of the Crusader IIL control, The operator should open the cabinet by swinging the front panel forwards. See figure 1.

To turn the control on, place the **ON/OFF** toggle switch (located on the back of the control) in the ON position.

Now, ensure the **Emergency Stop** button is out. This is checked by turning the button clockwise as indicated by the arrows.

Now push the **RESET** Switch. This is located on the cabinet above the Program Saver.

At this point, The MANUAL, IN/MM, HOLD, DISPLAY POSITION, RAPID, and IN POSITION lights should be on. The digits 1.00 will show in the FEED % display and the axis displays should read all zero.

If some lights are off or on which should not be, or if some of the digits are only partially lit, then turn the control off and call your Anilam representative. Or call the Anilam Service Hot-line at:

1-800-327-6340

If no lights at all appear, refer to the spare fuse listing inside the cabinet. Spare fuses are in the pocket located on the inside of the door at the rear of the cabinet.

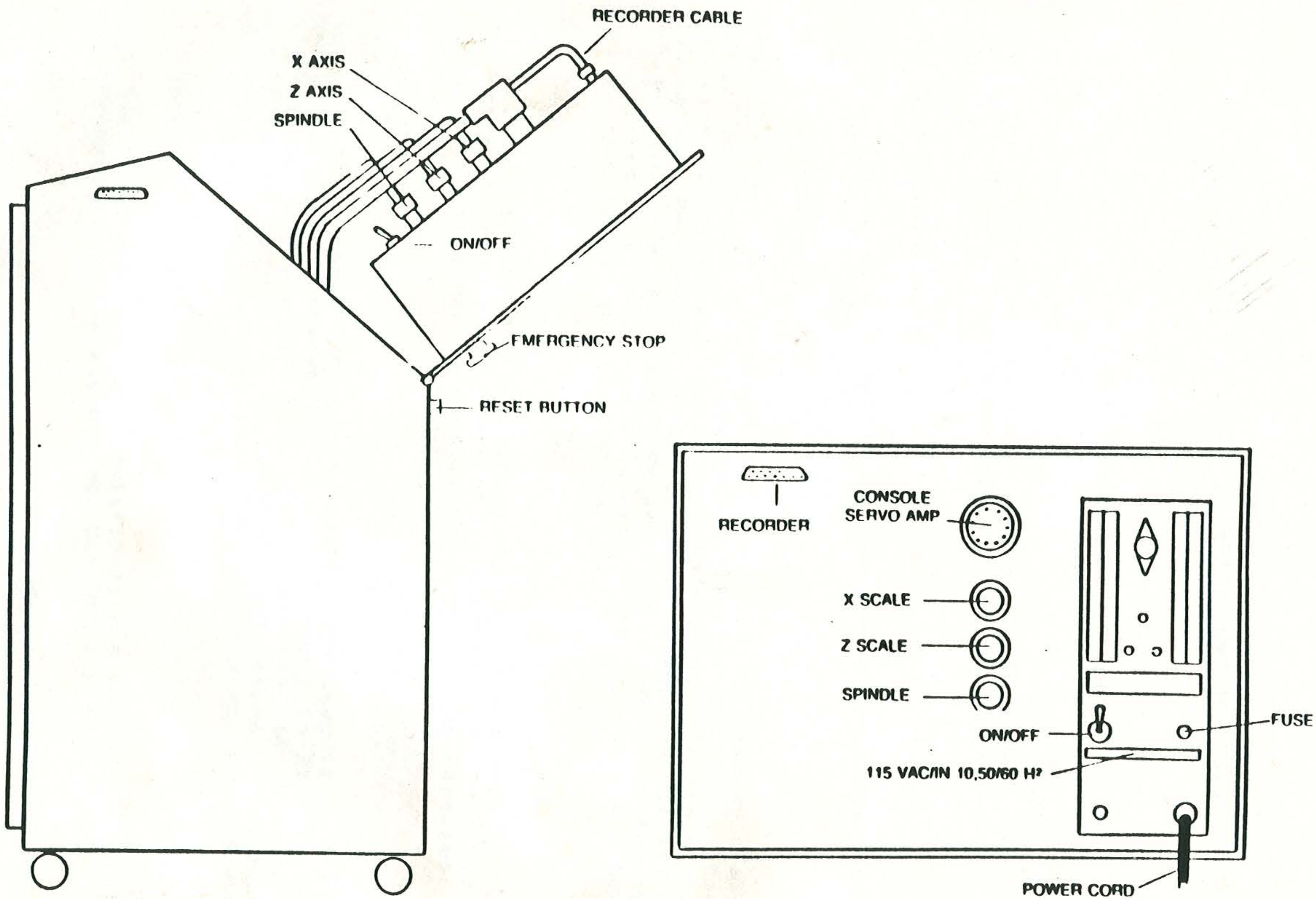

RECORDER CABLE
X AXIS
Z AXIS
SPINDLE
ON/OFF
EMERGENCY STOP
RESET BUTTON
RECORDER
CONSOLE SERVO AMP
X SCALE
Z SCALE
SPINDLE
ON/OFF
FUSE
115 VAC/IN 10,50/60 H²
POWER CORD

Figure 1

*** PART ONE: CONTROL FUNCTIONS ***

SECTION 1

THE ALPHA-NUMERIC KEYBOARD

When reading this manual refer to the foldout of the Crusader III keyboard (Figure 2) to keep yourself familiar with the key positions.

A. DIMENSIONING KEYS

The buttons X, Z, 0 to 9, +/- and the decimal point, are used for programming dimensions.

B. DISP POSN

When the light in the upper left corner of the display position button is on, the actual position of the tool is shown. This position is relative to Absolute Zero when tool length offsets are active (see Section 8). When not illuminated, the display shows the current commanded position. At any time-in any mode, the operator may depress this button to know either the program event position or the current actual position of the tool.

C. DO

This button is used for starting a repetitive sequence (DO LOOP). The DO command is always entered with a number of 1 to 999. This number determines how many times the loop will be repeated. See section 18.

D. CALL

This button is used to call a subroutine for execution. The CALL command is always entered with a number from 1 to 999. This number determines the subroutine that will be executed (or CALLed). After executing the subroutine, the program will return to the event following the CALL and continue the program execution. See section 18.

E. SUBR

This button is used with an identification number from 1 to 9999 to specify the start of a subroutine. See Section 18.

F. END

This button is used to define the end of a Do loop, subroutine, or the end of the main program.

G. RAPID/FEED

This button indicates in which mode the machine is being programmed. The upper light indicates RAPID moves and the lower light indicates FEED moves.

H. INCR

When lit, this button indicates that the move being programmed is in the incremental mode. When not lit, the system is in the absolute mode. For an explanation of absolute/incremental, see Section 3.

I. DWELL

This button causes the machine to remain stationary in its present location (when reached in AUTO mode). If a timed dwell is required, a number in seconds must be entered. For example:

DWELL 2.0 : A dwell of two seconds.
DWELL 0.5 : A dwell of one-half second.

J. AUX

The Auxiliary button and a four digit code number is used to access special features of the control. For example: AUX 1401, allows operator to control the RAPID rate of the machine.

K. CW/CCW

This button is used to specify the direction of the tool in an ARC command, either clockwise or counter clockwise. For an explanation of Arc programming see Section 7.

L. ARC

Used with the CW/CCW button, the ARC button enables the operator to program a cut in a circular motion.

M. TOOL

This button is used to set tool length offsets, tool radius compensation and to call tool numbers in the program. See Sections 8 and 9 for an explanation of tool length offsets and tool nose radius compensation.

N. V

This button is used to enter V-codes (or Variables) used in the canned cycles. The V is pressed, followed by two digits to identify the variable. Further key presses will be entered into the Z axis display. (For example: To enter V51 = 1.2500; press V 5 1 1 '.' 2 5).

O. G

The G button or G code is a preparatory code and prepares the machine to accept data in a particular form. For example:

G94: inches per minute feedrate programming.
G95: inches per revolution feedrate programming.

G codes are also used to identify the use of canned cycles. For a list of primary G codes, see the Quick Reference List in the beginning of this manual. For the use of canned cycles, see Section 10.

P. EVENT ENTER

This button only functions in the PROGRAM ENTER AND MANUAL modes of operation. Once all the information in an event is programmed, press EVENT ENTER and that block of information will be stored in the memory with the correct event number. The Crusader IIL will then automatically advance to the next event number for further information input.

The EVENT ENTER button is also used to preset any position into the displays in the MANUAL mode. To preset X and Z to zero (establish Absolute Machine Zero), simply press X Z 0 EVENT ENTER and this will reset the X and Z displays to zero.

Q. EVENT CLEAR

This button is used to clear an event within the program that is no longer required. EVENT CLEAR is also used to clear an entire program from the memory. Press PROGRAM ENTER and then press EVENT CLEAR five times. This will clear the memory.

R. ADD EVENT AND DELETE EVENT

These buttons are used for adding events or deleting events from a program in memory. For futher explanation see Section 17.

S. EVENT SEARCH

This button is used to look at information within a specific event without having to step through the program from Event 1. EVENT SEARCH can be used in any mode other than MANUAL. For further explanation and usage see Section 17.

SECTION 2

MODES OF OPERATION

When reading this section refer to the fold-out illustration of the Crusader IIL keyboard. This will help familiarize you with the key positions.

A. PROGRAM ENTER

This mode is utilized for programming and editing. It is in this mode that the operator enters or changes data in the memory.

B. PROGRAM CHECK

This mode is utilized to verify data in the program. In this mode, no data can be entered or changed.

C. SINGLE STEP

This mode is utilized when the operator wishes to execute one event of data. In the SINGLE STEP mode, when the START button is pressed, the machine will execute one event of the program then stop until the START button is pressed again to execute the next event.

D. AUTO

In this mode, when the START button is pressed, the machine will execute each event of the program automatically.

E. MANUAL

This mode is utilized to allow the operator complete flexibility in setting up the machine or moving any Axis individually.

Each of the above modes will be dealt with fully in the following sections.

SECTION 3

MACHINE AXIS FORMAT AND LAYOUT

Figure 3 illustrates the basic axis format for a manual engine lathe when viewed from the operator's position. All programs in this manual are referenced from a back-tooling position.

The arrows indicate the direction the tool will move for a plus or minus signed command.

A. DEFINING ABSOLUTE AND INCREMENTAL DIMENSIONING

Absolute and Incremental are the two modes in which the Crusader IIL can be programmed. The two modes differ in that absolute programming utilizes a set datum point from which all the programmed coordinates relate (see figure 4). Incremental uses the present tool position as a datum point and all coordinates are programmed from that point (see figure 5).

B. ABSOLUTE DIMENSIONING

As with Absolute Programming, Absolute Dimensioning also utilizes a known datum point from which all dimensions relate. In figure 4 the datum point is the center line of the component and the front face of the component. This datum point can be used on almost every component.

C. INCREMENTAL DIMENSIONING

Incremental dimensioning does not use a set datum point from which all dimensions relate. Instead, each dimension relates to a previous point (see figure 5). Therefore, if you wish to cut the diameter in figure 5 back to the farthest Z dimension, you must first calculate the distance from the front face of the component. The calculation for the example would be 2.5" + 2" + 0.5" + 0.5 = 5.5". If the dimensioning had been given in absolute the Z distance is known without having to calculate, therefore eliminating a possible error.

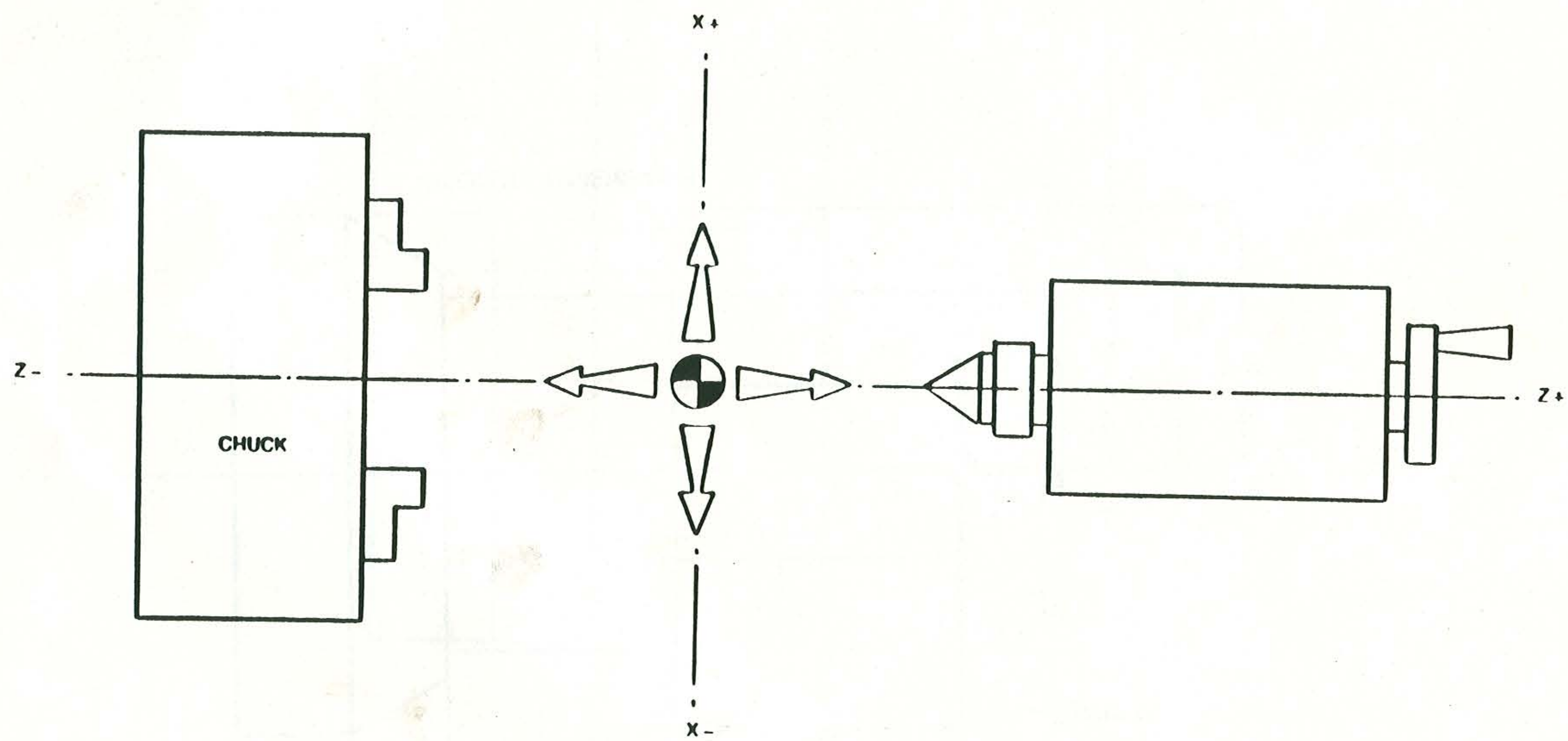
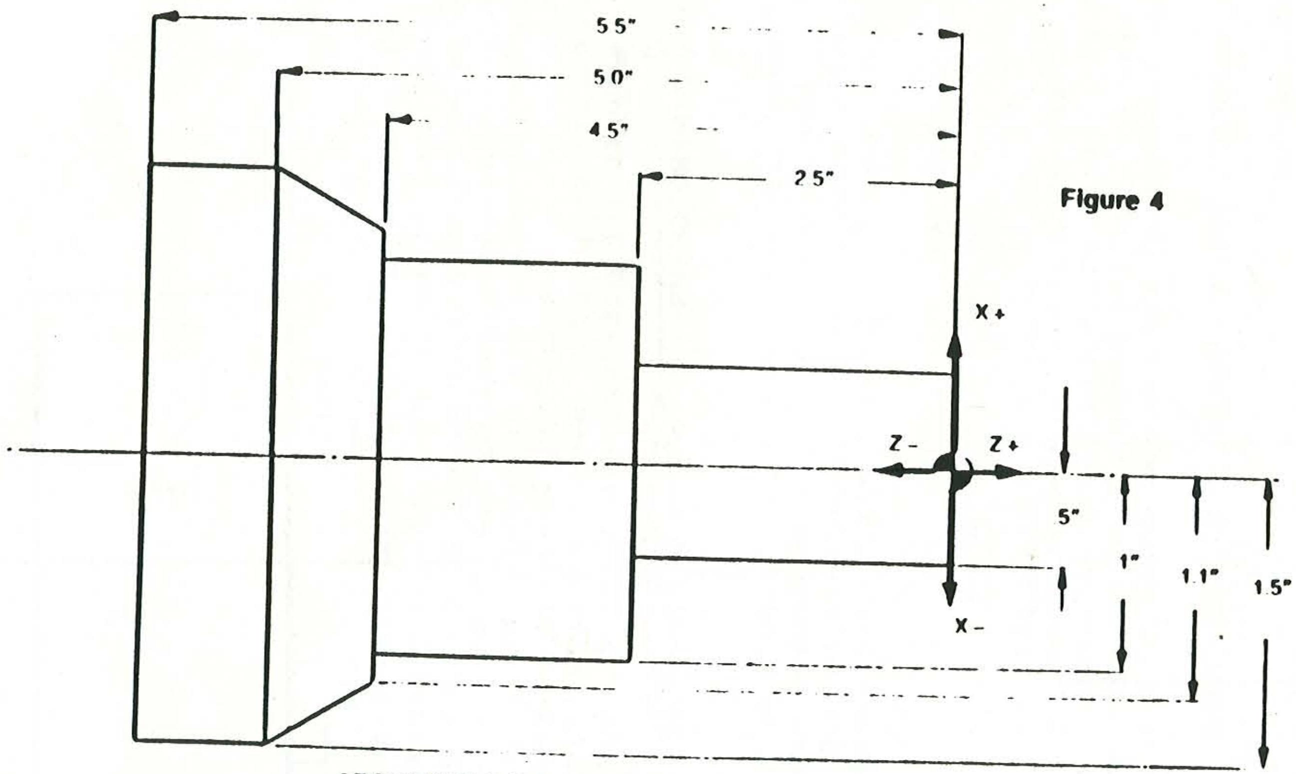


Figure 3

Figure 4



ABSOLUTE DIMENSIONS

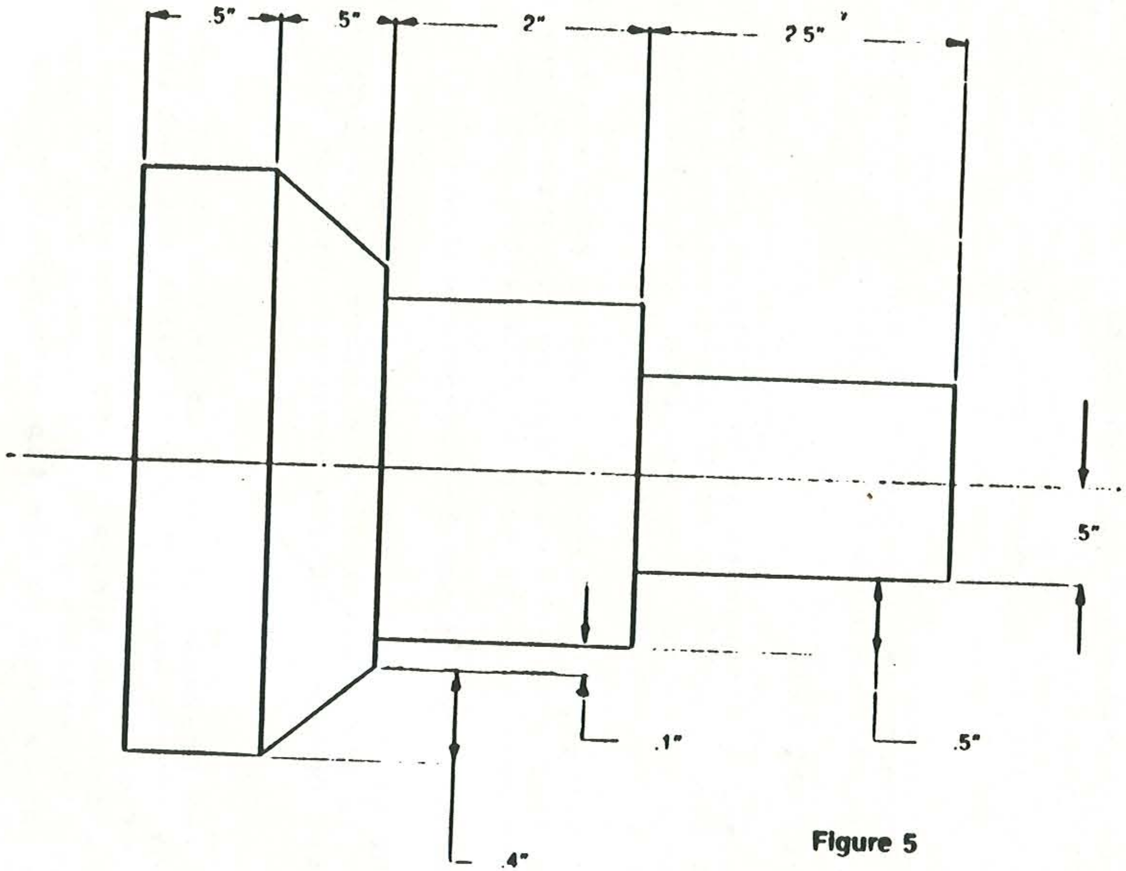


Figure 5

INCREMENTAL DIMENSIONS

SECTION 4

MANUAL AND JOG MODES

A. MANUAL OPERATION

The MANUAL operation mode of the Crusader IIL enables the operator to move the slides independently from any programmed moves. This mode is used mostly for setting purposes such as when a tool has to be moved to a position to set absolute machine zero or tool length offsets. See Section 8.

To operate the Crusader IIL in the MANUAL mode, press the MANUAL button, which lights up.

NOTE:
At start up, the Crusader is in MANUAL mode with feedrate in RAPID. Dimensioning is in ABSOLUTE and in INCH.

B. MOVEMENTS IN RAPID AND FEED

When MANUAL is selected, the feedrate is in RAPID as indicated by the light. The tool can be moved in RAPID along either axis by pressing the relevant axis button (X+, X-, Z+ or Z-). The tool will continue to move until the button is released.

By pressing the HAND button, the feedrate moves to FEED and the FEED light comes on. The feedrate percentage can be adjusted by the FEED percent buttons. Moves can then be made in FEED along either axis.

C. SLIDE JOGGING (Manual, Rapid and Indeterminate distances)

Simple jogging of the slides enables the operator to position the slides and tool post to the most convenient place for tool changes. It is also used to set Absolute Machine Zero (see Section 8).

D. MANUAL JOGGING

To move the slides, the Crusader IIL must be in the MANUAL mode of operation. Press the MANUAL key. This will light the red lamp, signifying the mode of operation. Set the feedrate to 100% for full rapid traverse (240 inches per minute) or to any other percentage of this rate. Settings are variable in 5% increments from 0-120%. The feedrate percentage can be read at the top left window of the control marked EVENT/FEED%.

NOTE:
AUX 1401 must be active in order to control the RAPID rate. See the Quick Reference List at the beginning of this manual for all AUX codes.

Press the appropriate key to move the slide in the required direction. The slide will continue moving until the key is released. Repeat as necessary.

Z- : Towards the spindle.
Z+ : Away from the spindle.
X- : Towards the machine center line.
X+ : Away from the machine center line.

E. JOGGING THE SLIDES TO SPECIFIC INCREMENTS

To move the slides a specific increment the control must be in the MANUAL mode of operation. Select the amount to be moved:

JOG 1 = 0.0002"/0.002 mm
or JOG 2 = 0.002"/0.02 mm
or JOG 3 = 0.02"/0.2 mm

Press the HAND symbol to select the desired increment distance. The red indicator lamp will show the selected increment (Jog 2 selected). Using the X and Z direction keys the slides can be moved the selected amount. Each strike of the key will give one incremental move of 0.002"/0.02mm. Repeat as necessary.

F. SINGLE STEP MOVES IN MANUAL

A single move in MANUAL can easily be entered and executed. For example, to move in 1.0" steps in the X+ direction in RAPID: First turn off DISP POSN so that information entered can be seen, and then press X 1.0 RAPID INCR. Then, each time START is pressed, the tool moves by 1.0" in the X positive direction. To clear this move press EVENT CLEAR. Any programmable move can be made as a Single Step, Manual move.

SECTION 5

FEEDRATE PROGRAMMING AND FEEDRATE OVERRIDE

A. FEEDRATE PROGRAMMING

In the Crusader IIL control there are two methods of programming a feedrate; G94 or G95:

G94: Feeding in Inches (or mm) Per Minute (IPM)
G95: Feeding in Inches (or mm) Per Revolution (IPR)

Since all manual lathes use IPR as their method of feeding, it is suggested the IPR be used when programming with the Crusader IIL control (IPR refers to the distance moved by the tool for each revolution of the spindle). The control monitors the spindle speed via a rotary encoder attached to the spindle, therefore it can accurately feed at any rate from .0001" to the maximum RAPID rate of the machine.

B. PROGRAMMING IN IPR

Inch Per Revolution (IPR) is set and activated by a G-code (G95). You must first enter Variable 21 which contains the amount of feed per revolution. The next event must be G95. This will remain in effect for any feed move until it is changed in the program. Here is an example of how to program in IPR:

Event #	Command	
12.	TOOL 1	Activate Tool #1 offsets.
13.	V21 .008	Feedrate selected at .008"
14.	G95	Feedrate selected in I.P.R.
15.	X-1.0 Z-1.0 FI	X & Z Axes to feed -1.0 at .008" per revolution of the spindle.

NOTE:
Always program a feedrate after a tool call.

C. PROGRAMMING IN IPM

Inch Per Minute (IPM) is set and activated by a G-code (G94). This will allow entry and change of feed rates using the FEED key and will remain in effect until V21 and G95 is programmed. The following is an example of how to program in IPM:

Event #	Command	
12.	TOOL 1	Activate tool #1 offsets.
13.	G94	Feedrate selected in I.P.M.
14.	FEED 20.0	Feedrate selected at 20.0"
15.	X-1.0 Z-1.0 FI	X & Z Axes to Feed -1.0 at 20.0 inches per minute.

D. FEEDRATE OVERRIDE

Within the Crusader IIL control is the ability to adjust the feedrate without having to edit the program. This is an important feature when proving a component program. For instance, if a cut is being taken and the feedrate appears to be too fast, the operator can lower the feedrate during the cut and obtain the correct chip control, by using the FEEDRATE % override. In the same manner the operator can also raise the FEEDRATE % override.

The FEEDRATE % override has a range of 0 to 120% For example, if 0.01" I.P.R. has been programmed and the override is set at 100%, the feedrate will be 0.01" IPR. With the override set at 60%, the feedrate will be 0.006" IPR. The override is adjustable in 5% increments using the FEED % buttons located at the top of the control panel to the left of the X and Z directional buttons.

When the FEED % button is pressed, the actual percentage of FEED is displayed in the EVENT/FEED % window at the top left of the control.

E. IN POSITION LIGHT

This light indicates that a command has been completed. For example, if a command causes the machine to move an axis, the IN POSN light will come on when the machine reaches its target position. This light is located below the FEED % buttons.

F. INCH/MILLIMETER BUTTON

The INCH/MM button and its lights indicate in which mode (inch or mm) the machine is being programmed and which mode the machine displays are showing. It is possible to program in INCH or MM, but the mode cannot be changed within a single program.

NOTE:

Do not change the mode during a cycle of the program or when in AUTO.

SECTION 6

SYSTEM STATUS SECTION AND POSITION DISPLAY

The left side of the Crusader IIL displays the appropriate information during programming or operation.

A. SYSTEM STATUS SECTION

The system status section at the top left hand panel of the Crusader IIL consists of two sets of displays. These are used throughout the programming process to show the status of the event being programmed.

The left half is used to show the event number during programming, Single Step and Auto operations. In MANUAL it displays the feedrate percentage. It can also display the feedrate percentage in other modes, if either FEED % button is pressed. After a few seconds it reverts to the event number.

The right half displays the actual feedrate at which the tool is moving. If an Auxiliary (AUX) number, Do Loop, Subroutine or Dwell has been programmed, this information is displayed instead of the feedrate.

An Auxiliary (AUX) number is displayed when the block containing it is current. If a Do Loop is being programmed, the number of times the loop is to be repeated is displayed. Subroutines are assigned a number and when they are initiated that number is displayed. The number of seconds remaining in a timed Dwell is also displayed here.

B. POSITION DISPLAYS

In the lower half the 3 large displays show, from top:

X Axis position: either actual or commanded position.
Z Axis position: either actual or commanded position.

Note:
DISP POSN sets either actual or commanded position.

The lower display shows two separate functions. The left of the display shows the active tool number. The right of the display shows the machine spindle speed.

SECTION 7

CIRCLES AND ARCS

A. CIRCULAR MOTION

When a tool is required to cut in a circular motion, the programmed events that enable the machine to move in this mode are known as an 'ARC Statement'. Four events are required to make up the Arc Statement:

1. The direction of the circular move, clockwise or counter clockwise (CW/CCW).
2. The center point, in both X and Z Axes, of the ARC radius.
3. The end point in both X and Z Axes, of the ARC or radius.
4. ARC (Used to end the ARC Statement.

The Crusader IIL requires these four events so the tool will move along the correct path (See Figure 6).

Without programming the direction of the arc the Crusader has four possible routes from Point A to Point B, so we tell the Crusader that the arc is to be cut counter clockwise (CCW). This still leaves two possible routes that the tool can take, so we give the Crusader the center point of the arc. The Crusader now knows the point about which to swing the arc. Because the tool is on the start point of the arc, the Crusader knows the radius of the arc.

Now the Crusader must be given the end point of the arc, otherwise it will assume the move is a complete circle. The Crusader now knows the direction of the arc, the point about which to swing the arc, and the point at which to end the arc. The last step is to end the 'Arc Statement' by programming ARC.

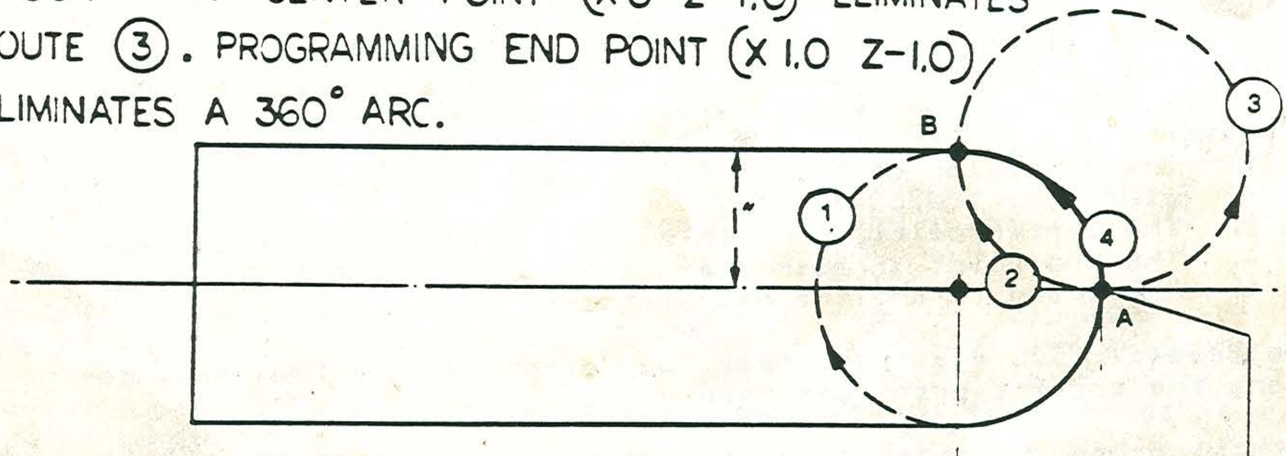
The same format is used in any type of radius or ARC programming, no matter if the radius is less than 90 degrees or more than 90 degrees. Study the examples to become accustomed to programming arcs and radii.

B. ABSOLUTE & INCREMENTAL ARCS

When programming an arc in absolute, the center point and end point of the arc must be programmed with coordinates relating to the component zero point. Normally this is the center line of the component and the front face (see figure 6).

However, when programming an arc in incremental, the center point and end point of the arc are coordinates relating to the starting position of the tool. Both absolute and incremental coordinates can be used within the Arc Statement, but cannot be programmed in the same event. The center point of the arc could be programmed in absolute and the end point programmed in incremental. The type of dimensioning used on the component drawing will dictate the most convenient method of programming the arc.

4 POSSIBLE ROUTES FROM POINT A—B ① THRU ④
PROGRAMMING (ARC CCW) ELIMINATES ROUTES ① & ②
PROGRAMMING CENTER POINT (X O Z-1.0) ELIMINATES
ROUTE ③. PROGRAMMING END POINT (X 1.0 Z-1.0)
ELIMINATES A 360° ARC.

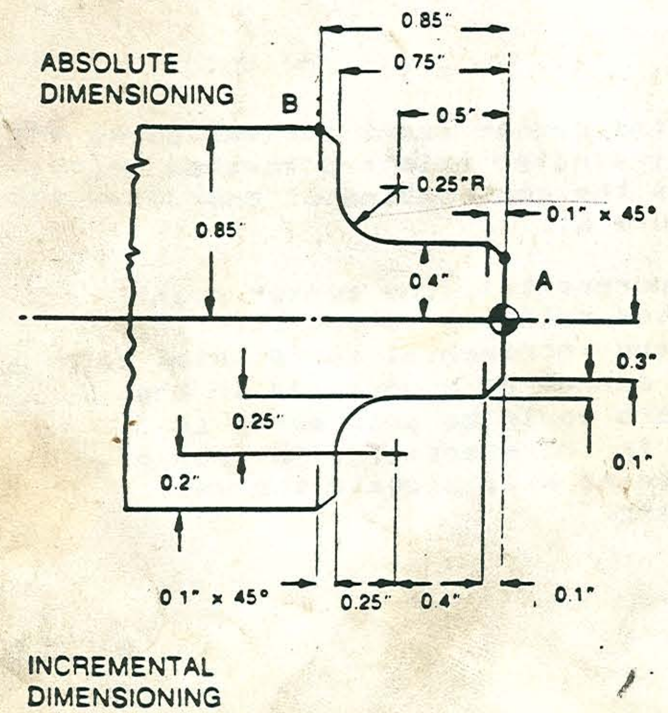


PROGRAMMING (ARC) ENDS THE ARC STATEMENT.

AS SEEN IN THE PROGRAM.

EVENT # COORDINATES

12. X O Z O FA. FEED TO START POINT.
13. ARC CCW. DIRECTION OF ARC.
14. X O Z-1.0 FA. CENTER POINT.
15. X 1.0 Z-1.0 FA. END POINT.
16. ARC. END OF ARC STATEMENT.



PROGRAM FROM A — B

	ABSOLUTE	INCREMENTAL
21.	X0.4 Z-0.1 FA	X0.1 Z-0.1 FI
22.	Z-0.5 FA	Z-0.4 FI
23.	ARC CW	ARC CW
24.	X0.65 Z-0.5 FA	X0.25 Z0 FI
25.	X0.65 Z-0.75 FA	X0.25 Z-0.25 FI
26.	ARC	ARC
27.	X0.75 FA	X0.1 FI
28.	X0.85 Z-0.85 FA	X0.1 Z-0.1 FI

Figure 6

SECTION 8

SETTING ABSOLUTE ZERO AND TOOL LENGTH OFFSETS

A. ABSOLUTE ZERO

Absolute Machine Zero is a position used for a tool change or component change. Absolute Zero can theoretically be set anywhere within the movement of the X and Z slides. Although it is possible to set Absolute Zero anywhere, the component dimensions will dictate the most advantageous setting position.

The factors that the operator must take into consideration are: the diameter and length of the component; the ease of access for changing both tools and components; and the distance that the tool must move before reaching its start point.

B. SETTING ABSOLUTE ZERO POSITION

When a position has been selected for the Absolute Zero position, manually move the X and Z slides to that point. Follow the steps below to set Absolute Zero:

1. Press MANUAL (even if the MANUAL lamp is lit) to clear any previous command.
2. Press TOOL 0 and START to deactivate any tool length offsets that may be active from a previous program.
3. Press X 0 Z 0. By pressing the above keys the existing figures in the X and Z Axes are re-defined as X zero and Z zero.
4. Press EVENT ENTER.

At this time, the X and Z Axis displays will read zero and the Absolute Zero position will be set. Any time when there are no tool length offsets active (deactivated by programming TOOL 0), the X and Z Axes can be programmed to return to Absolute Zero (i.e., when changing tools). For example, a move to Absolute Zero is programmed, before each tool change or at the end of a cycle.

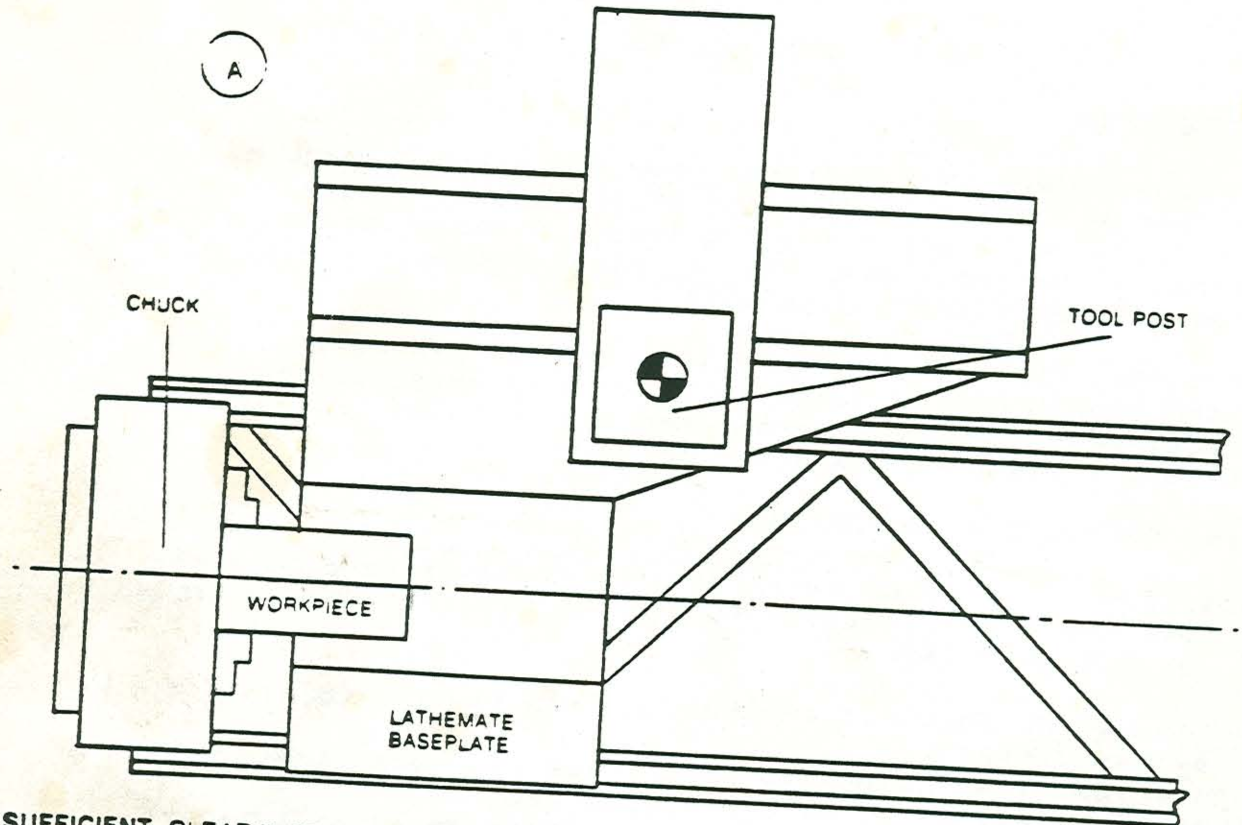
Figure 7 shows two positions at which Absolute Zero can be set. Example A indicates the correct setting position for the component shown.

* The slides are sufficiently clear of the component so the operator can safely change both the tool and part without risk of injury.

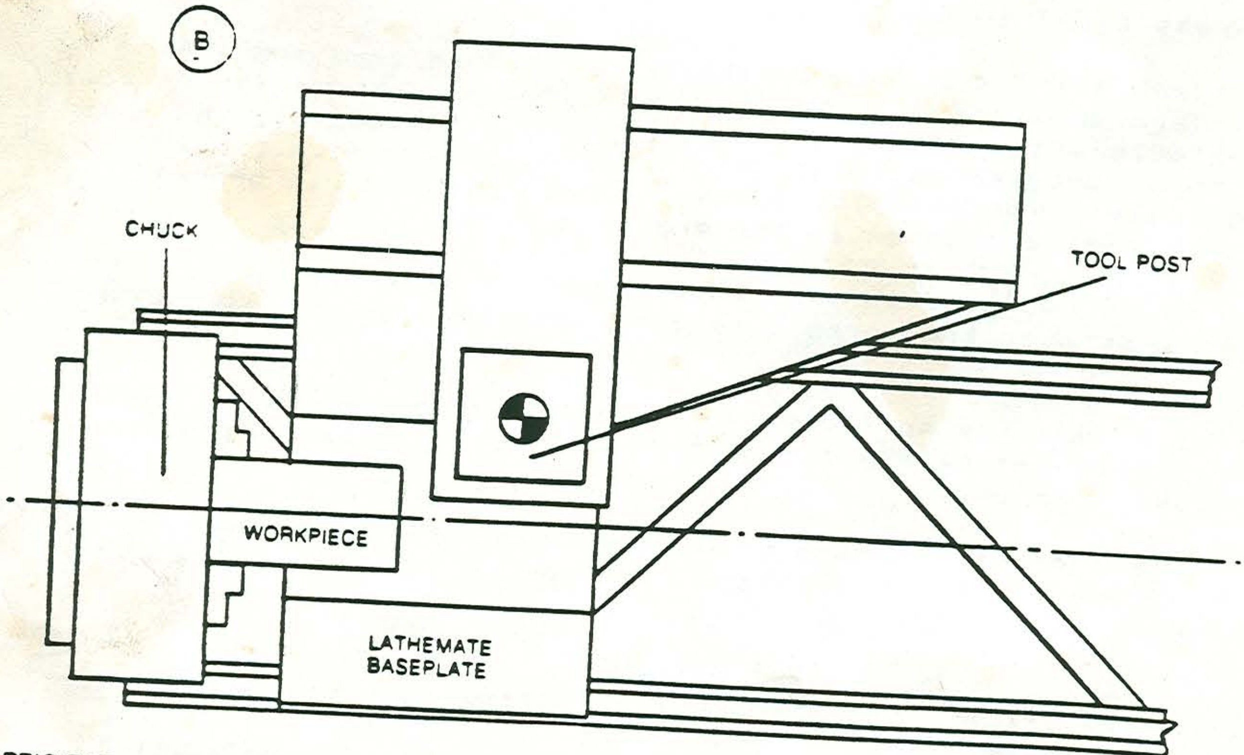
* The slides do not have to move an excessive distance before reaching the start position for a cut. This makes for a faster cycle time.

NOTE:

Figure 7 is illustraded showing the tooling position on the Anilam Lathemate. Whether your lathe is equipped with the Lathemate or not, the safety principles discussed here still apply.



SUFFICIENT CLEARANCE TO CHANGE TOOLS & COMPONENTS



NSUFFICIENT CLEARANCE TO CHANGE TOOLS & COMPONENTS

Figure 7

Example B indicates an incorrect setting for the Absolute Zero position.

* The slides are positioned too close to the component. The operator cannot change the tool without risk of injury.

* The X Axis is positioned below the diameter where the first cut would occur. Therefore, the operator would have to program an X+ move before positioning for the first cut. Similarly when returning to Absolute Zero, the tool would have to be programmed to move Z+ before moving in the X Axis, otherwise the tool would move directly into the component.

C. TOOL LENGTH OFFSETS

The tool length offset (TLO) is the distance from the tool tip when the slides are at the Absolute Zero position to the front face and center line of the component (or component zero point). The control needs these distances so that Absolute programming can be utilized.

Tool length offsets are entered at the beginning of the program during set up. The offset distances are entered using a 4 digit tool command. For example, if the offsets for the turning tool (Figure 8) were -3.000 in the X Axis and -2.000 in the Z Axis, the offsets entered into the control at the beginning of the program would be as follows:

Event #	Program Data
1.	TOOL 2001 (to be explained)
2.	X -3.0 Z -2.0 RA

If the offsets for the boring tool (Figure 8) were -3.500 in the X Axis and -1.000 in the Z Axis, the offsets entered into the control at the beginning of the program would be as follows:

Event #	Program Data
3.	TOOL 2002
4.	X -3.5 Z -1.0 RA

The 4 digits are interpreted as follows. The first 2 digits (20) are used to signify that tool length offsets are being set. The last 2 digits (01, 02, 03, etc., through 31), identify the TOOL number to which the offset is set. The actual tool offsets will be entered by the operator during set up.

If two tools are to be used for a given component, when the operator enters the program, events 2 and 4 are left empty for tool length

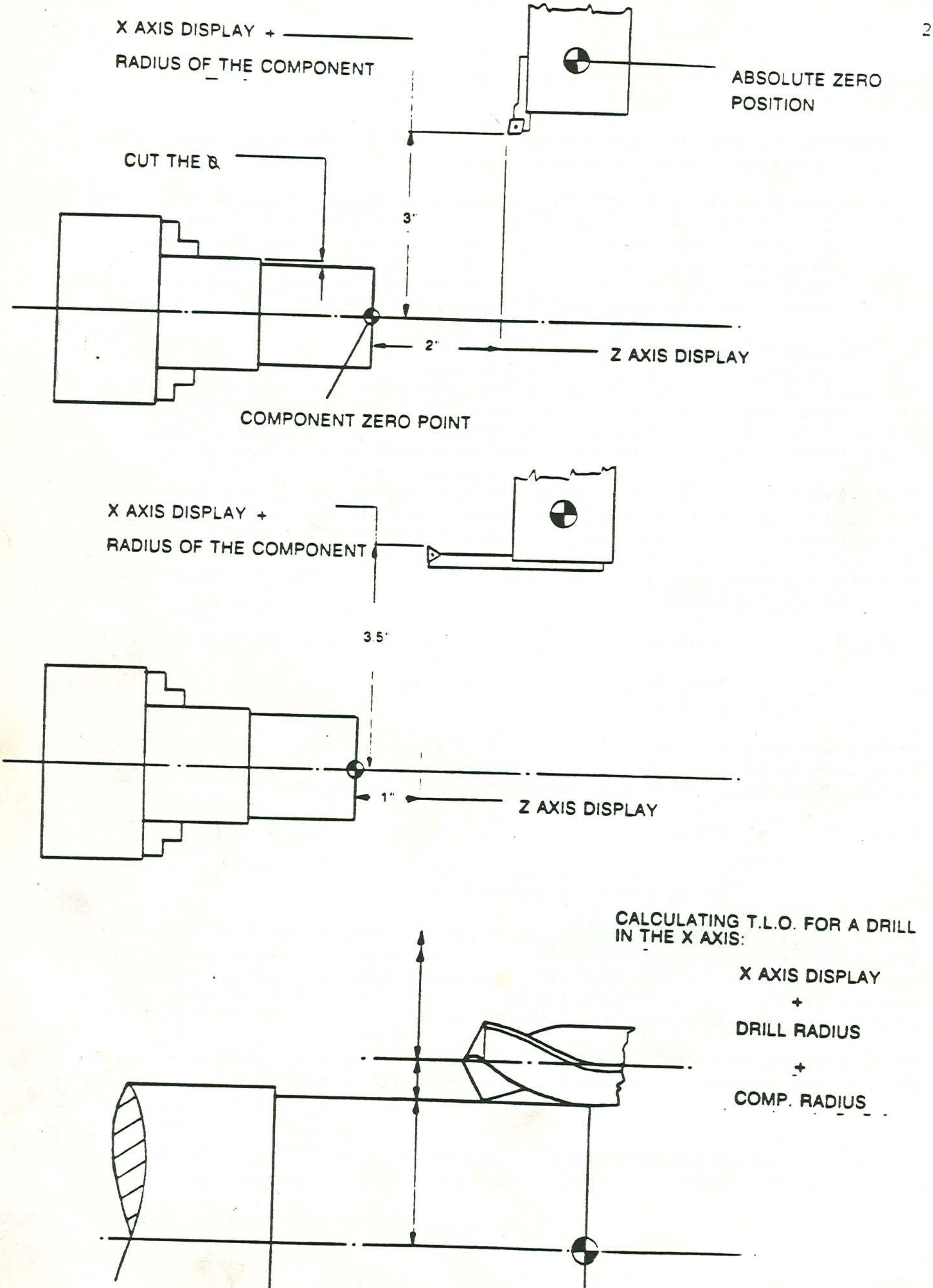

X AXIS DISPLAY +
RADIUS OF THE COMPONENT
ABSOLUTE ZERO POSITION
CUT THE ℄
3"
2"
Z AXIS DISPLAY
COMPONENT ZERO POINT
X AXIS DISPLAY +
RADIUS OF THE COMPONENT
3.5"
1"
Z AXIS DISPLAY
CALCULATING T.L.O. FOR A DRILL IN THE X AXIS:
X AXIS DISPLAY
+
DRILL RADIUS
+
COMP. RADIUS

Figure 8

offsets to be entered during set up. See the example below:

Event #	Program Data
1.	TOOL 2001
2.	(Empty; data to be entered during set up.)
3.	TOOL 2002
4.	(Empty; data to be entered during set up.)

D. SETTING TOOL LENGTH OFFSETS

The tool length offset (TLO) data should be the first information in the program. Each TLO is set using a 4 digit TOOL number followed by the TLO value (distance from the Absolute Zero position to the component zero point).

TLOs are established by the operator during set up as follows:

1. Load an unmachined component into the chuck.
2. Set the tools into the tool holders.
3. Clamp the tool holder on the tool post.
4. Set the Absolute Zero position (see Setting Absolute Zero Position).
5. Press MANUAL.
6. Using the slide directional keys, move the tool to a position to take a light cut on the diameter of the component.

   Note:
   Use FEED % to control the feedrate of the slides.

7. Write down the distance moved in the X Axis (the distance is displayed in the X Axis display). Measure the diameter that has been cut and divide by 2 to give the radius of the part. The radius can now be added to the distance in the X Axis. The answer will be the distance from the Absolute Zero position to the center line of the machine, which is the tool length offset for that tool in the X Axis.
8. Using the same tool, take a light cut down the front face of the component and write down the distance moved in the Z Axis (Z Axis display). The answer will be the distance from the Absolute Zero position to the front face or the component, which is the tool length offset for that tool in the Z Axis.
9. The same procedure is used to obtain the tool length offset values for subsequent tools but the cuts do not have to be taken. These tools can be brought into position using the 'slide jogging' procedure and just touched against the two machined surfaces.

NOTE:
When the tool length offsets are being set both the X and Z Axis are moving in a minus direction. Therefore, tool length offsets are always minus values and are entered in the program as minus values.

SECTION 9

TOOL NOSE RADIUS COMPENSATION

A. G40, G41 AND G42

The programmer always programs the path of theoretically sharp pointed tools. The difference between the theoretical sharp point and the actual point of the tool radius can be compensated for by using G41 and G42 (tool nose radius compensation codes). Both G41 and G42 are modal codes and stay in effect until cancelled by G40 (tool nose radius compensation off).

Two codes, G41 and G42, are necessary because the control must know if, when viewed in the direction of travel, the tool is to the left or to the right of the component. Figure 9 is an easy to follow guide giving the correct G code to use for a specified direction. When tool nose radius compensation is no longer required it is cancelled by G40.

B. WHY IT IS NECESSARY

If all the tools used in a cutting operation had perfectly sharp points there would be no need for tool nose radius compensation. A problem arises only when a tool has a tip radius, and even then only under certain situations.

A tool tip radius will not cause any problems if the work consists of straight facing, square shoulders and straight turning. However, when angles, chamfers, arcs or any simultaneous two-axis moves are required, then an error will be seen in the finished part. This error will depend on the size of the tool tip radius and the particular move being done. The larger the tip radius, the larger the error.

Figure 10 shows how an error occurs on a chamfer-type move. The cross-hatched area is material that will not be removed unless radius compensation is used.

Figure 11 shows how tool nose compensation would automatically correct this problem.

In a similar fashion, arcs programmed without using tool tip compensation, and combinations of arcs, tapers, etc., will all show some loss of true form. Whether compensation should be used will depend upon the tolerances involved and the size of the tool tip radius. When it is needed, it is a very powerful programming tool, automatically doing all the necessary trigonometry and mathematics for you.

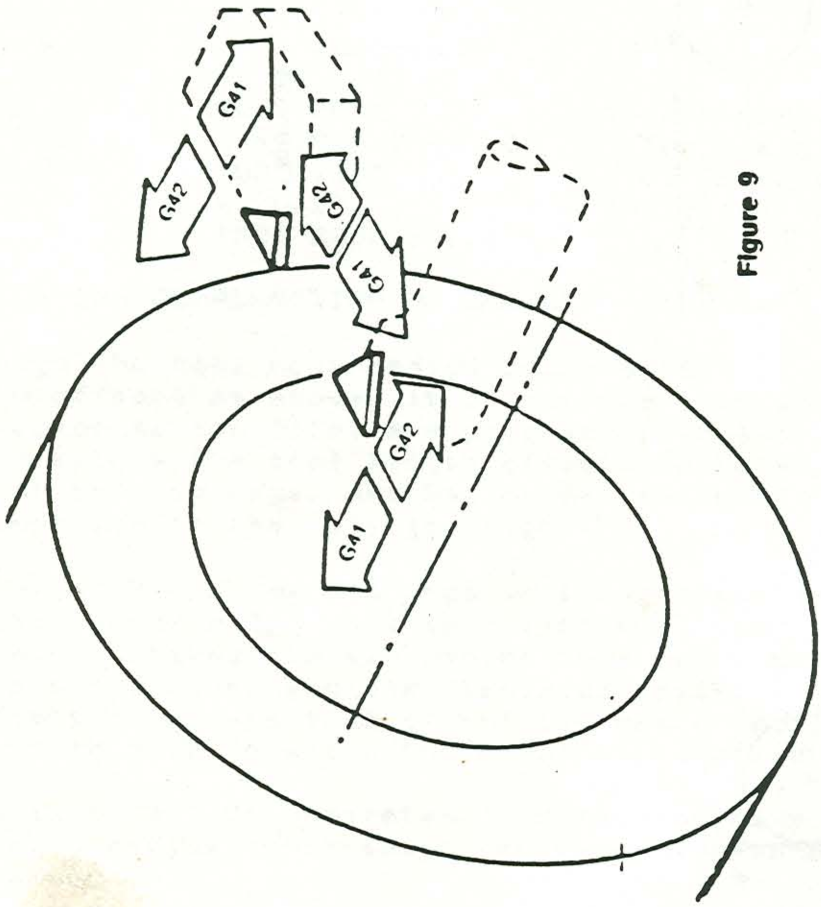


Figure 9

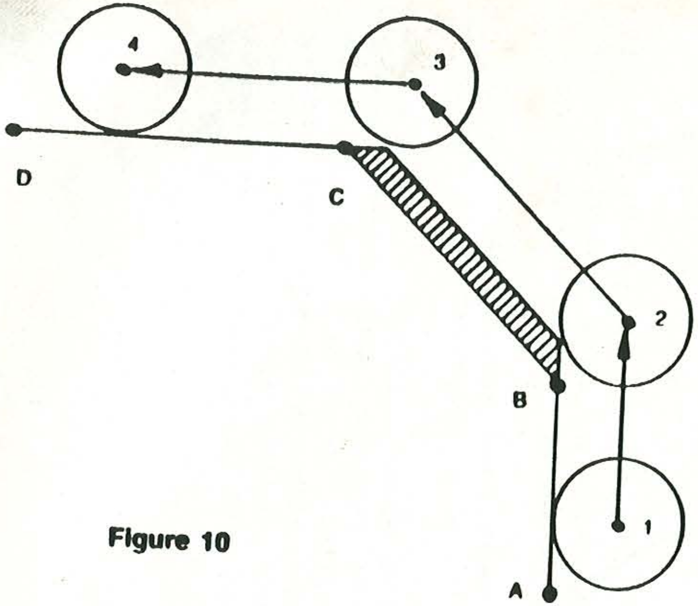


Figure 10

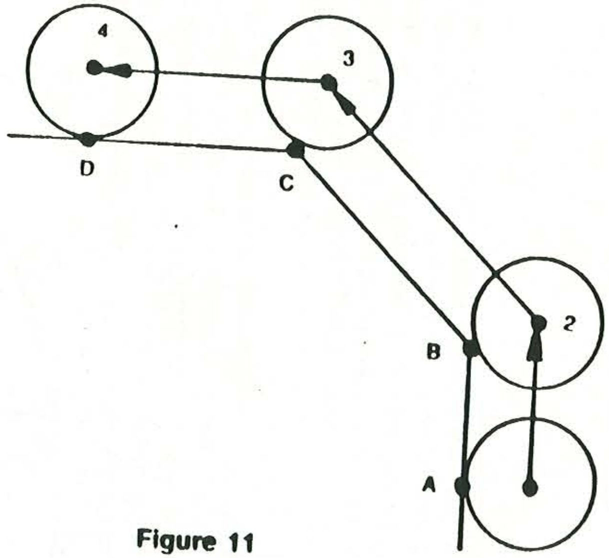


Figure 11

C. DESCRIBING THE TOOL

When compensation is used, more information about the tool is required than just the tool length offsets. For example, in TOOL 1301 the first digit tells the control that a tip radius is to be entered. The second digit is the TOOL location code number, and the last two give the TOOL number (01 to 32). The location code is the direction that the tool tip is pointed (see Figure 12).

This is followed by the tip radius, entered under the X address (always a positive number). Thus the complete entry for TOOL 1 with a tip radius of 0.032" is as follows:

Event #	Command	
1.	TOOL 2001	Tool number.
2.	X -3.945 Z -2.462	Tool length offsets.
3.	TOOL 1301	Tool location offset.
4.	X .032"	Tool tip radius.
5.	TOOL 2002.....etc.	

D. TURNING COMPENSATION ON AND OFF

Although the tool nose radius compensation is entered after the tool length offsets as above, it is not put into action until the tool nose compensation codes are programmed (G41 or G42). The location number alters the tool length offsets to be at the tip center rather than at the tip edge, and G41 or G42 instructs the control to make compensation to the left (or right).

The G41 or G42 command brings in the compensation during the next move in X and/or Z. This is called the 'ramp on' move. The ramp on move always takes the tip center to a point perpendicular to the next X Z move, and the displayed position while compensation is in effect is always that of the tip center position though the programmed move is along the component profile.

It is important to understand the ramp-on move when initiating tool nose radius compensation so safe ramp-on moves can be programmed.

When compensation is to be cancelled, G40 is entered to cancel either G41 or G42. The next X Z move 'cancels' the compensation, and is known as the 'ramp-off' move.

Study figure 13 and notice how the positioning of the tool tip in relation to the programmed coordinate dot is affected by G42 and G40. Figure 13-A shows how the part shoulder would be cut into by ramping off too soon. A general guide to use is to ramp on the work perpendicular to the first two compensated moves and ramp off perpendicular to the last two compensated moves.

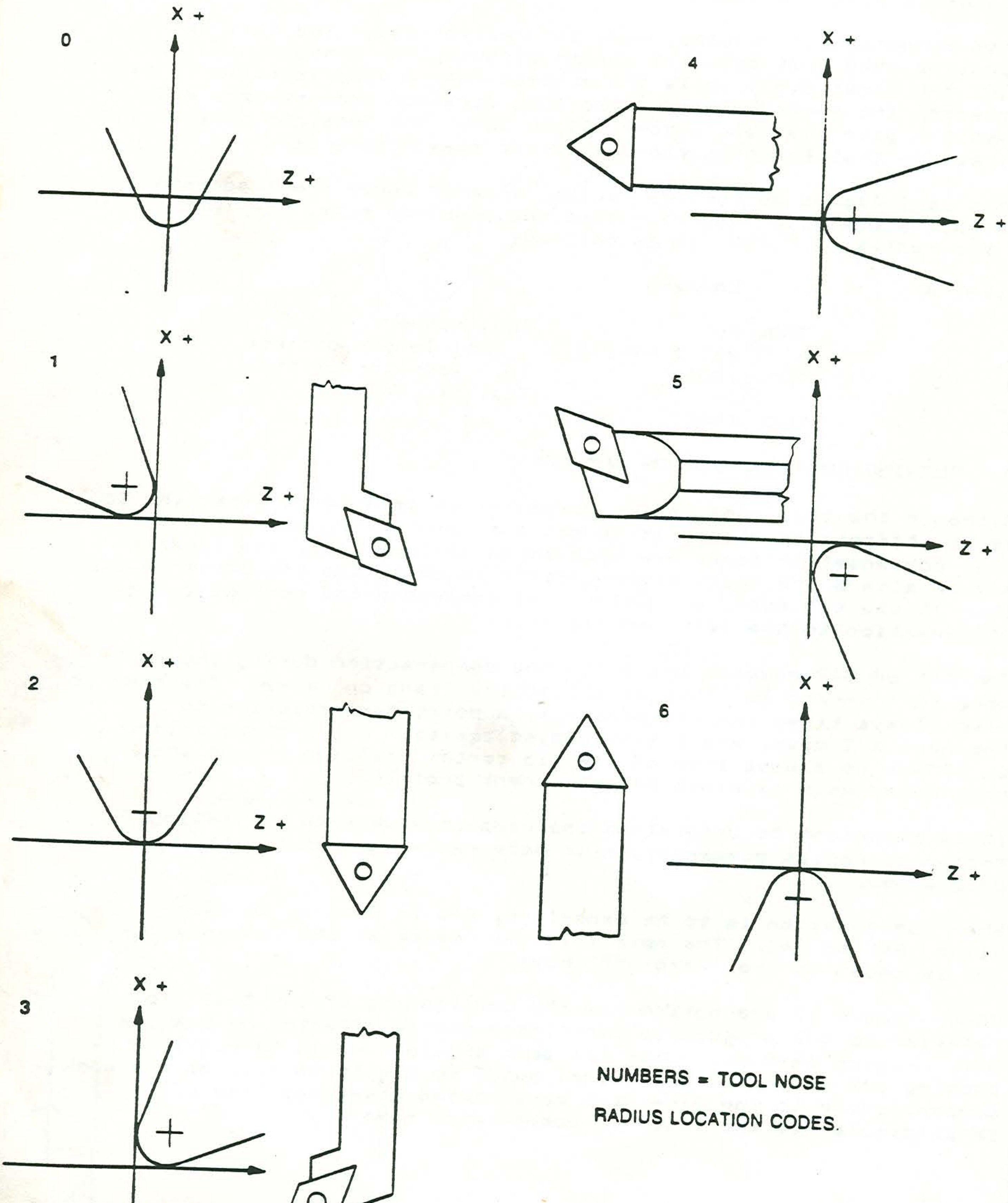

0
X +
Z +
4
X +
Z +
1
X +
Z +
5
X +
Z +
2
X +
Z +
6
X +
Z +
3
X +
NUMBERS = TOOL NOSE
RADIUS LOCATION CODES.

Figure 12

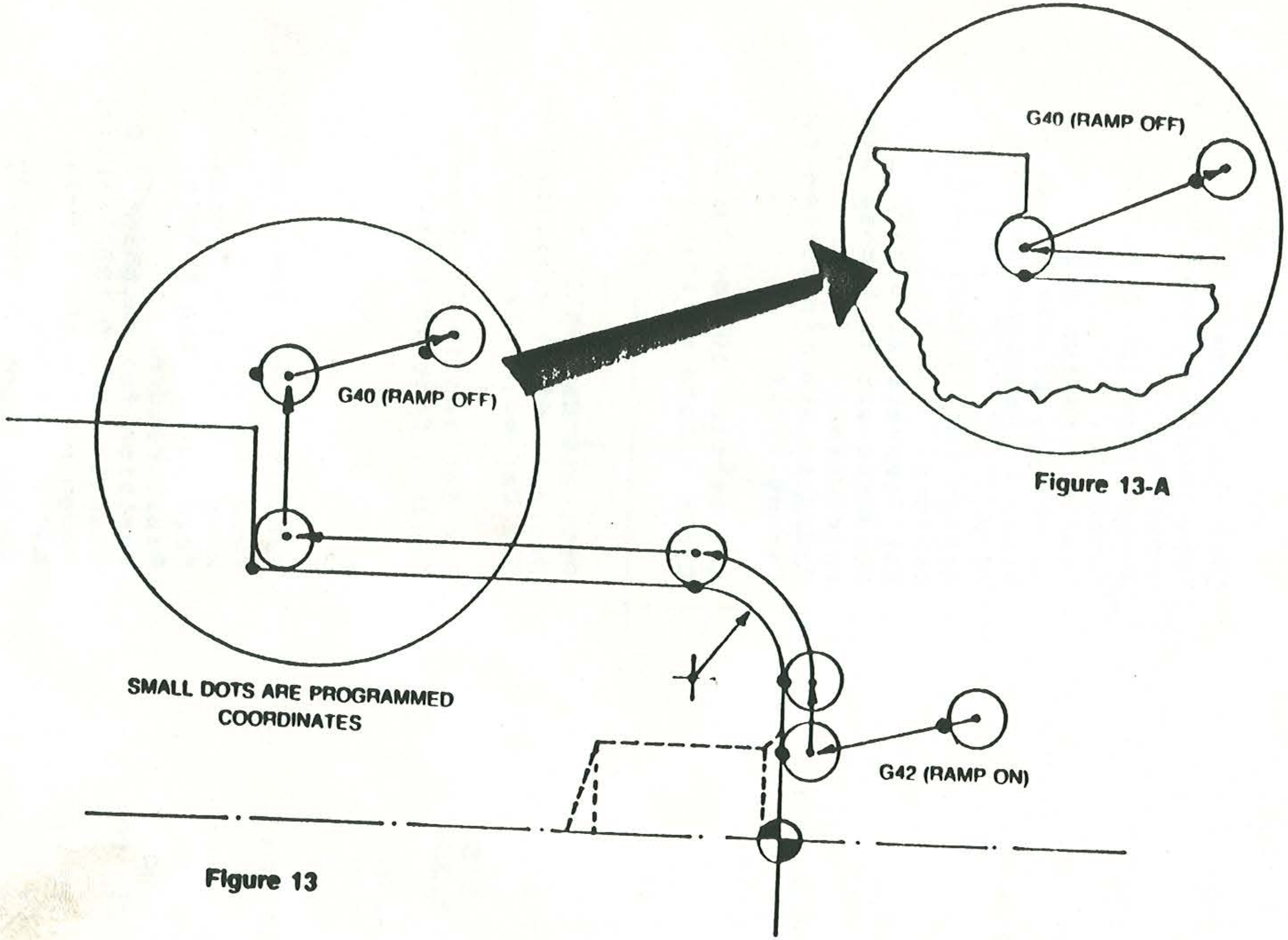


Figure 13

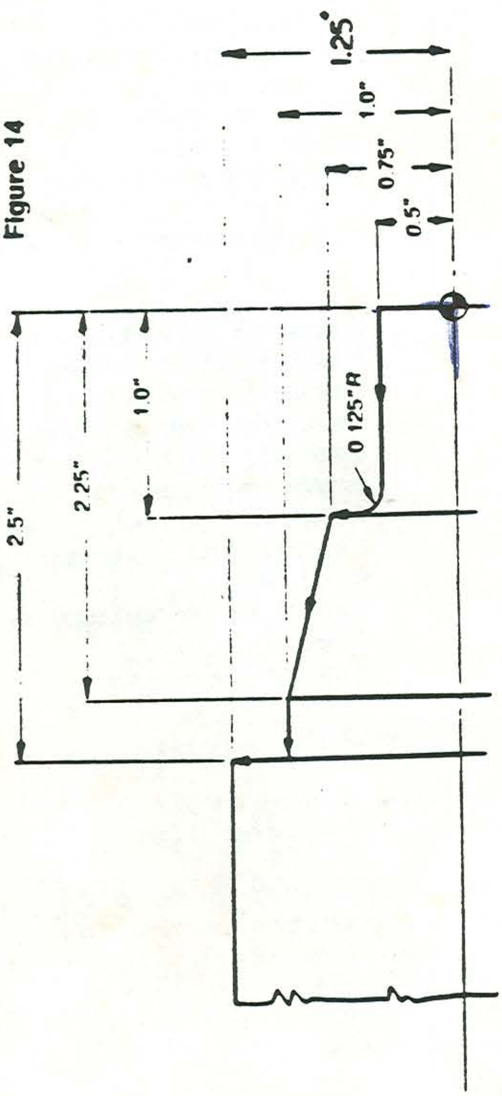
Figure 13-A

E. EXAMPLE OF TOOL NOSE RADIUS COMPENSATION

Figure 14 shows a component with an arc and a taper to be programmed using tool nose radius compensation. It is to be rough turned using 0.1000" cuts, followed by a finish cut using compensation. All TOOL and FEED information is to be included in the program. Following the part drawing is the program and description of the events.

Event #	Command	
1.	TOOL 2001	Call tool number 01.
2.	X Z RA	Tool length offsets for tool number 01.
3.	TOOL 1301	Assign location 3 to tool number 01.
4.	X .032 RA	Tool tip radius size.
5.	TOOL 0	Cancel any active T L O.
6.	X 0 Z 0 RA	Rapid to Absolute Machine. Zero.
7.	TOOL 1	Assign tool 1 tool length offsets.
8.	V21 .012	Set feedrate at 0.012.
9.	G95	Feedrate set in inches per revolution.
10.	X 1.35 Z .1 RA	Rapid to start position for Canned Cycle.
11.	V50 -.225	
12.	V51 -2.60	See Section 10 for Canned Cycle V Code definitions.
13.	V52 -.10	
14.	V53 -2.60	
15.	V54 0	
16.	G81	Activate Canned Cycle.
17.	X 1.1 Z .1 RA	Rapid to start Position for 2nd Canned Cycle.
18.	V50 -.225	
19.	V51 -2.35	See Section 10 for Canned Cycle V Code definition.
20.	V52 -.10	
21.	V53 -1.0	
22.	V54 0	
23.	G81	
24.	X .65 RA	Rapid to start Position for roughing cut.
25.	Z -1.0 FA	Rough cut in the Z Axis.
26.	X .75 FA	Feed out in the X Axis.
27.	Z .1 RA	Rapid return.
28.	X .525 RA	Position for roughing 0.5" diameter and 0.125" radius.
29.	Z -.875 FA	Rough cut in the Z Axis.
30.	ARC CW	ARC direction.
31.	X .125 Z 0 FI	Center point of the ARC incrementally.

Figure 14



32.	X .125 Z -.125 FI	End point of the ARC incrementally.
33.	ARC	End of ARC statement.
34.	Z .1 RA	Rapid Return.
35.	X .4 RA	Position in the X Axis for ramp on move.
36.	V21 .008	Re-set feedrate for finishing cut.
37.	G95	Feedrate set in inches per revolution.
38.	G42 Compensation	Call tool nose radius
39.	X .5 Z0 FA	Ramp on move.
40.	Z -.875 FA	Feed to start of radius and ramp on move.
41.	ARC CW	
42.	X .125 Z 0 FI	ARC statement in incremental.
43.	X .125 Z -.125 FI	
44.	ARC	
45.	X .75 FA	Feed to start of taper.
46.	X 1.0 Z -2.25 FA	Feed up the taper.
47.	Z -2.50 FA	Feed to shoulder.
48.	X 1.25 FA	Feed up the shoulder.
49.	G40	Cancel tool nose radius compensation.
50.	X 1.35 FA	Ramp off move.
51.	TOOL 0	Cancel tool length offset for tool #1.
52.	X 0 Z 0 RA	Rapid to Absolute Machine Zero
53.	END	End of Program.

F. RULES FOR TOOL NOSE RADIUS COMPENSATION

1. The tool nose radius value is always given as an X Axis dimension and the location number is always the second digit in the TOOL 1000 code.
2. TOOL 1 (or TOOL 2, etc.) must be programmed before G41 or G42 so that the control knows which tool offsets are active.
3. The ramp-on move is always to a position perpendicular to the next X and/or Z move, and at least the tool nose radius size. Example: Using a .032" tool nose radius the ramp on move must be at least .032" in both the X and Z Axis.
4. The ramp-off X Z move follows G40.
5. There must be at least one straight line move after programming G41 or G42 before cutting an ARC.
6. An ARC can only be followed by a straight line move or another ARC. G40 is not permitted immediately following an ARC.
7. Changing the tool number (including changing to TOOL 0 or ending the program) is not permitted until after the ramp-off move following G40.
8. G40 causes the tool to stop perpendicular to the last X, Z move.
9. An uncompensated move must take place between G40 and the end of the program.

SECTION 10

CANNED CYCLES

Below is a list of the Canned Cycles available in the Crusader IIL.

A canned cycle is a series of cuts the machine can make in a predetermined manner. The canned cycles allow the operator to make the cuts without having to program every individual move. The lengths and depths of the cuts are determined by programming 'V-Codes' or variables. The assigned 'G' code determines the type of canned cycle to be used (i.e. G81: Rough Turning; or G83: Deep Hole Drilling; etc.).

The canned cycles can be used both externally and internally on any component, the signs (+ or -) give the direction the tool will move when taking the predetermined cuts.

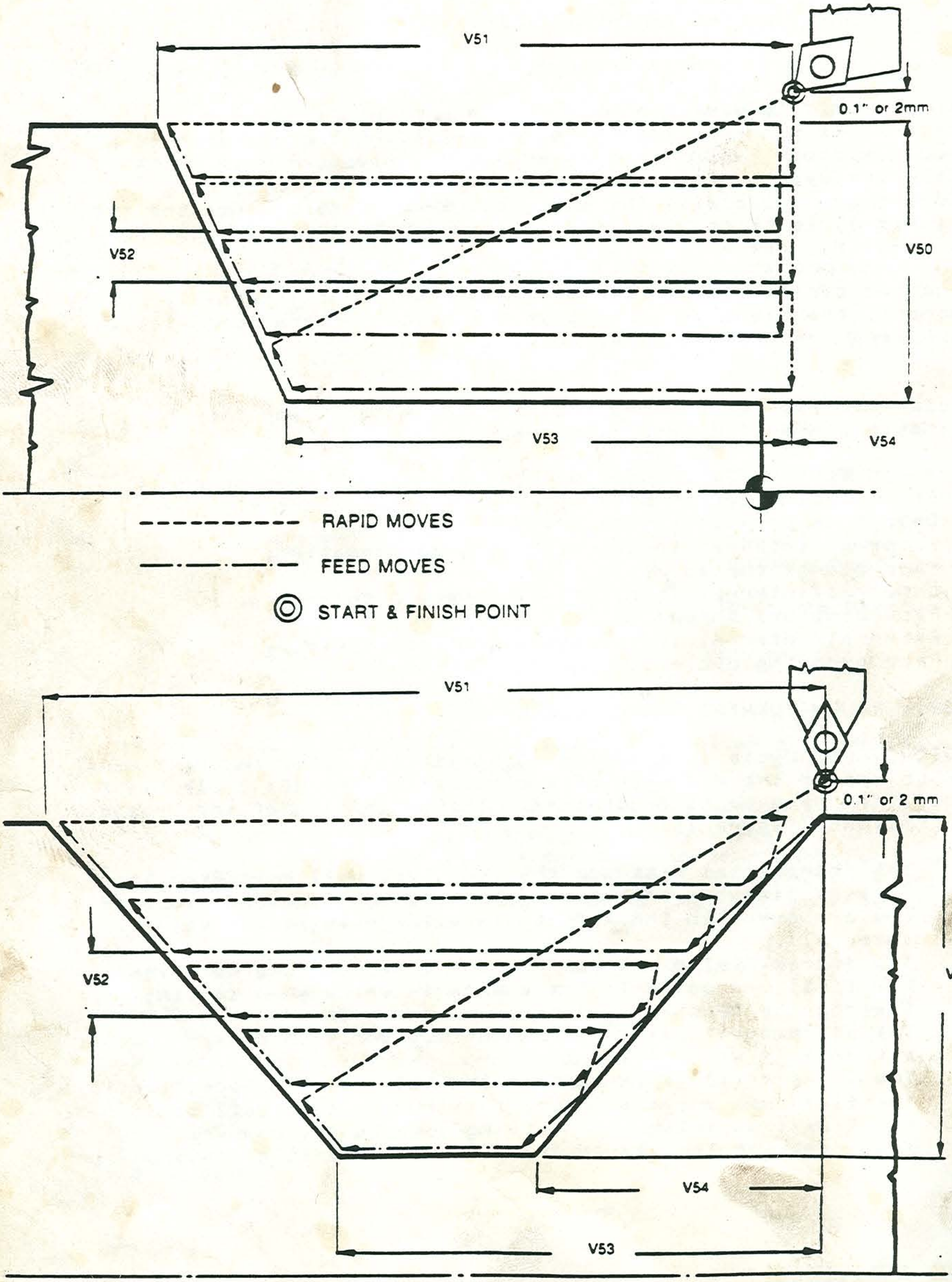
NOTE:
For clarification of V codes the operator should read the explanations and study the diagrams.

G81: Rough Turning/Boring cycle, with or without feed in angle.
G82: Rough Facing cycle, with or without feed in angle.
G83: Deep Hole drilling cycle.
G84: External/internal longitudinal plunge threading.
G85: Face plunge threading.
G86: External/internal longitudinal compound threading.
G87: Face compound threading.
G88: External/internal longitudinal grooving cycle.
G89: Face grooving cycle.

A. G81: ROUGH TURNING CANNED CYCLE

The G81 Canned Cycle is a series of horizontal cuts, in the Z Axis, taken to reduce the diameter of a component. The G81 cycle has 5 V-codes that have to be programmed. The definition of the V-codes is as follows (figure 15):

V50: The incremental distance that the tool will move from the outside diameter of the component to the finished diameter of the component in the X Axis (- value external, + value internal).
V51: The incremental distance from the tool starting position to the final end point in the Z Axis (- value when cutting towards the chuck).
V52: The incremental maximum depth of cut for each pass (- value external, + value internal).
V53: The incremental distance from the tool starting position to the final end point in the Z Axis before the 'pull out' in the X Axis (- value when cutting towards the chuck).
V54: The incremental distance from the tool starting position to


V51
0.1" or 2mm
V50
V52
V53
V54
RAPID MOVES
FEED MOVES
START & FINISH POINT
V51
0.1" or 2 mm
V52
V50
V54
V53

40

Figure 15

the Z Axis 'pull in' length (- value when cutting towards the chuck).

G81: Code programmed to activate the Canned Cycle.

NOTE:
When entering the Canned Cycle into the Crusader IIL, the above format must be followed V50 through V54 followed by G81.

B. G82 ROUGH FACING CANNED CYCLE

The G82 Canned Cycle is a series of vertical cuts in the X Axis taken to reduce the diameter of a component.

The G82 cycle has 5 V-codes that have to be programmed. The definition of these V-codes is as follows (figure 16):

V50: The incremental distance that the tool will move from the front face of the component to the finished face of the component in the Z Axis (- value when cutting towards the chuck).

V51: The incremental distance from the tool starting position to the final end point in the X Axis (- value external, + value internal).

V52: The incremental maximum depth of cut for each pass (- value when cutting towards the chuck).

V53: The incremental distance from the tool starting position to the final end point in the X Axis before the 'pull out' in the Z Axis (- value external, + value internal).

V54: The incremental distance from the tool starting position to the Axis 'pull in' length (- value external, + value internal).

G82: Code programmed to activate the Canned Cycle.

NOTE:
When entering the Canned Cycle into the Crusader IIL the above format must be followed: V50 through V54 followed by G82.

C. G83 DEEP HOLE DRILLING

The G83 Canned Cycle is a series of drilling 'pecks' in the Z Axis, taken to drill a hole in a component. The pecks allow for chip clearance when drilling a deep hole.

The G83 cycle has 2 V-codes that have to be programmed. The definition of the V-codes is as follows (figure 17):

V50: The incremental distance that the tool will move from the front face of the component to the bottom of the hole (always a - value).

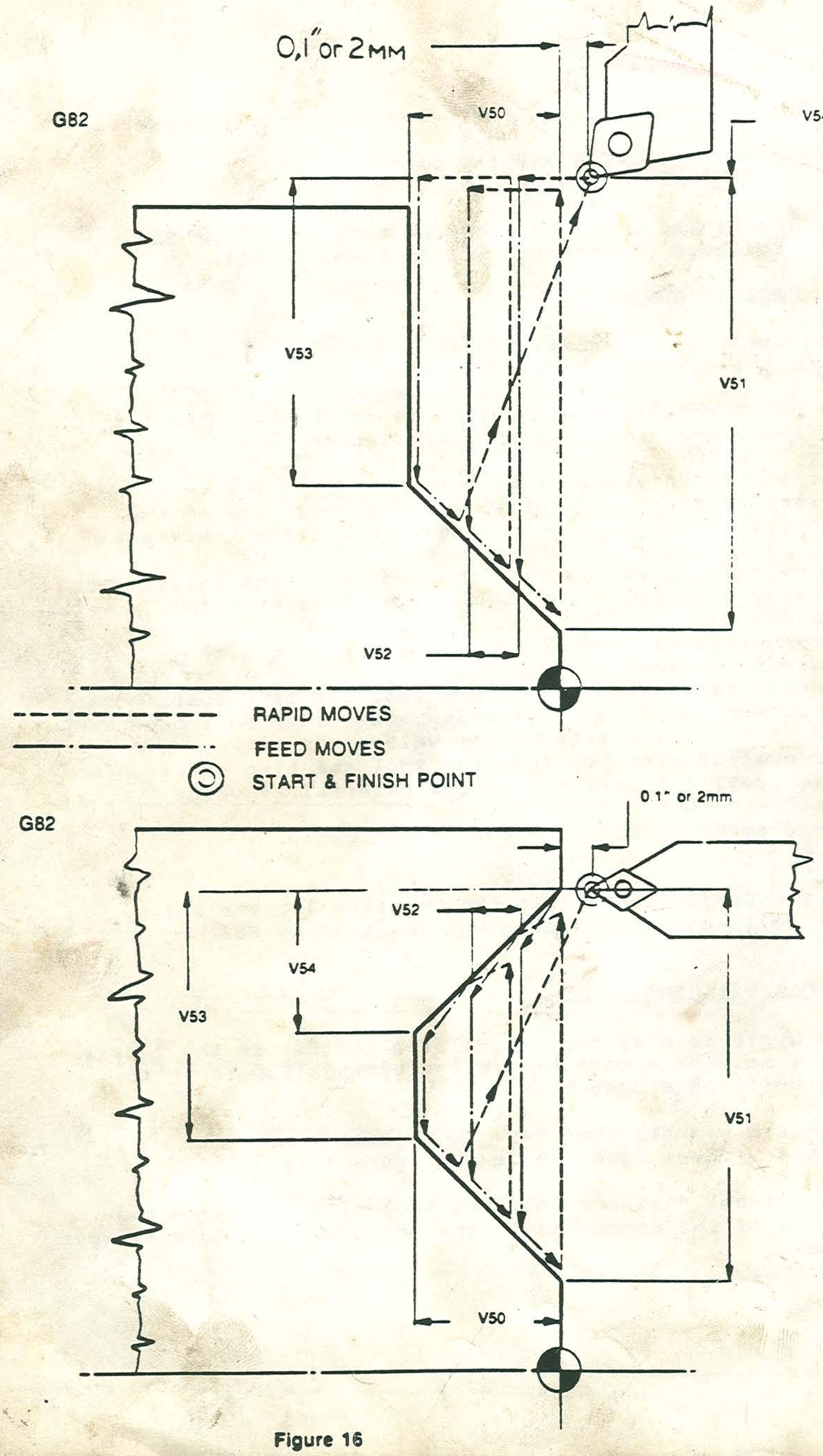

0,1" or 2MM
G82
V50
V54
V53
V51
V52
RAPID MOVES
FEED MOVES
START & FINISH POINT
G82
0.1" or 2mm
V52
V54
V53
V51
V50

Figure 16

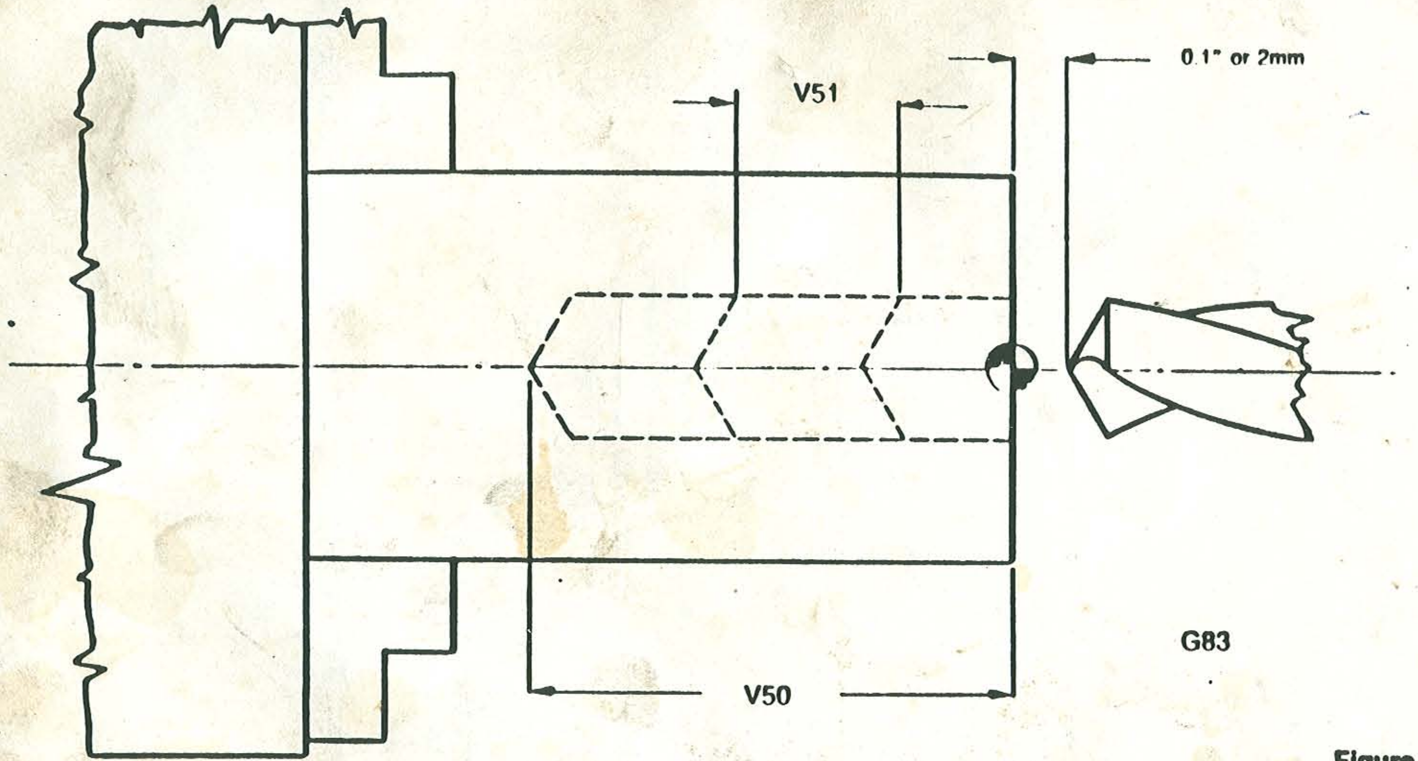


Figure 17

The acceleration distance is a length that is programmed preceeding the start of the thread cutting. This length is dependent on the feedrate. (the pitch of the thread) the faster the feedrate (a more coarse thread) the greater the acceleration distance needs to be. The chart below gives a general guide to follow:

F. **G84: LONGITUDINAL PLUNGE THREADING**

The G84 Canned Cycle is a series of passes on a diameter that progressively cut deeper to form a thread.

The tool must be positioned at the starting point shown. The G84 cycle has seven V-codes that have to be programmed. The definition of the V-codes is as follows (figure 18):

V41
or
V42: Either V41 or V42 can be programmed, but not both. V41: pitch of the thread. V42: Threads per inch. V41 should always be used when programming in metric, since metric threads are always given with pitch dimensions.

V50: The incremental distance in the Z Axis that the tool will move from the tool starting position, to the end point at full depth of thread. (- value when cutting towards the chuck).

V51: The incremental full depth of thread from the edge of the component (- value external, + value internal).

V52: The depth of the first pass, which is used to calculate the volume of each pass (- value external, + value internal).

V53: The incremental depth of the taper (zero if no taper) in the X Axis. Note: The angle of the taper cannot exceed 45 degrees (+ value external, - value internal).

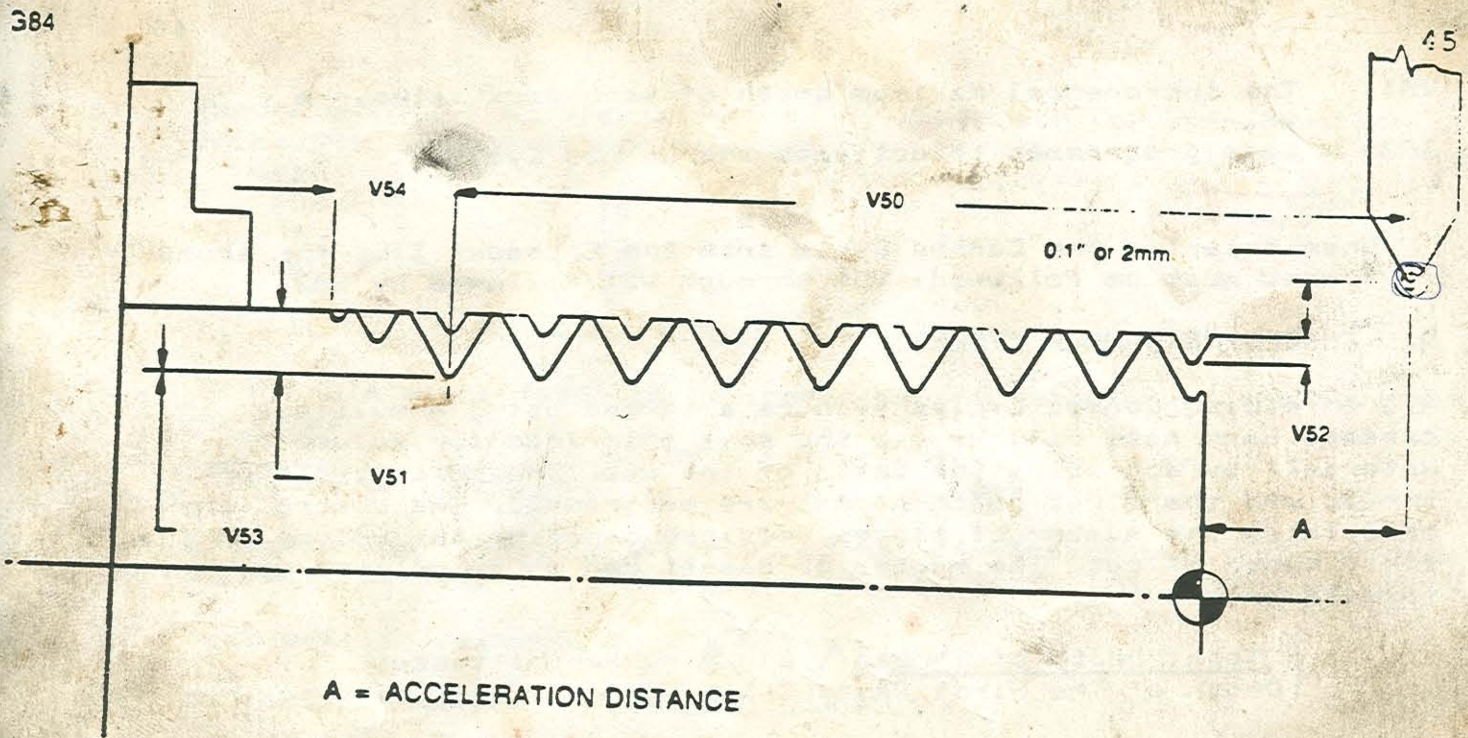
V54: The maximum incremental distance the tool will move along the Z Axis while pulling out of the thread (- value when cutting towards the chuck).

V55: The number of 'spring passes' the operator may program at full depth of thread, to remove tool deflection or burrs. THERE MUST BE A MINIMUM OF ONE.

G84: Code programmed to activate the Canned Cycle.

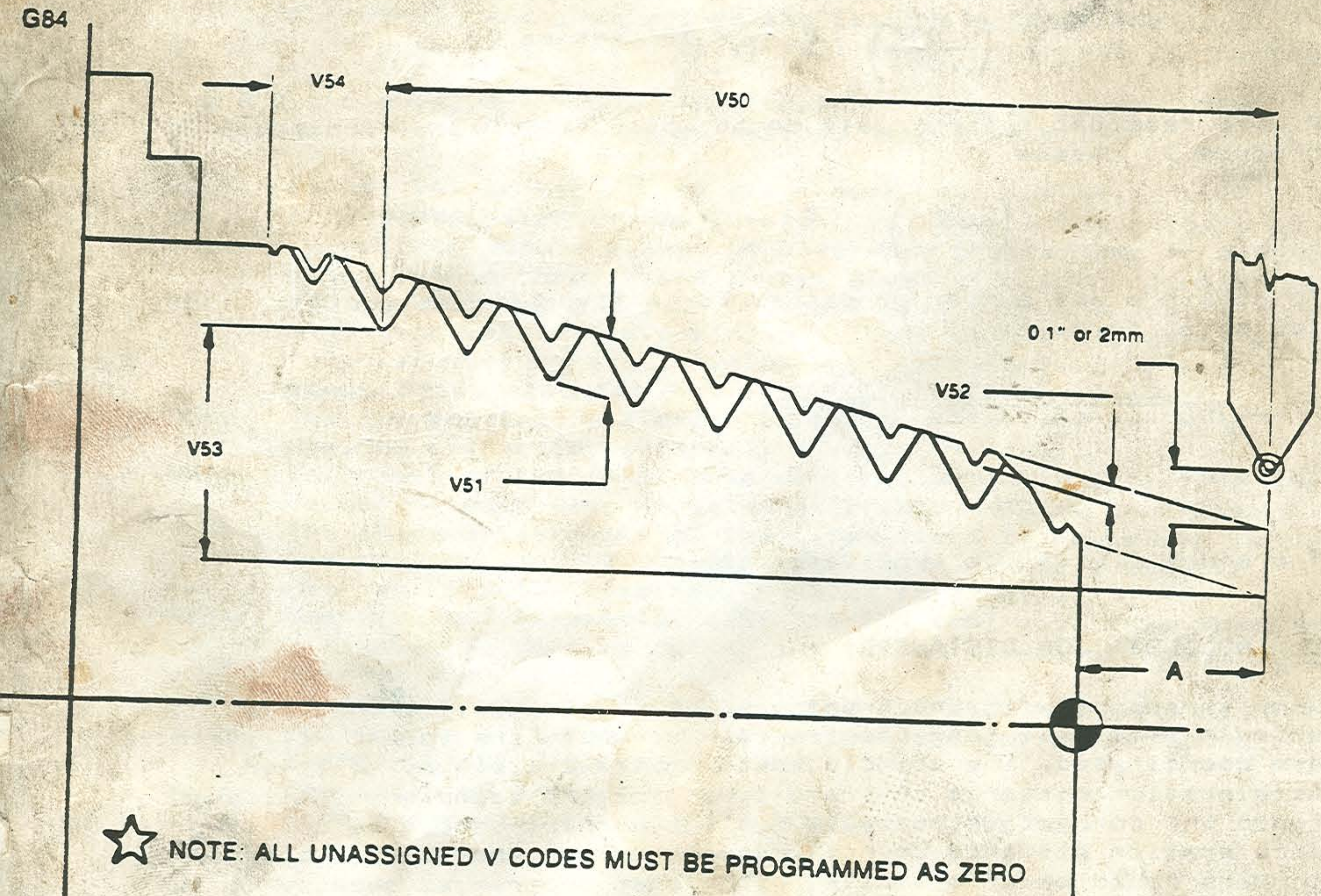
NOTE:

When entering the Canned Cycle into the Crusader II, the above format must be followed: V41 through V55 followed by G84.



A = ACCELERATION DISTANCE

V55 = NUMBER OF PASSES AT FINAL DEPTH



☆ NOTE: ALL UNASSIGNED V CODES MUST BE PROGRAMMED AS ZERO

Figure 18

V51: The incremental maximum depth of each peck (always a - value).

G83: Code programmed to activate the Canned Cycle.

Note

When entering the Canned Cycle into the Crusader IIL, the above format must be followed: V50 through V51 followed by G83.

D. THREADING CANNED CYCLES

All threading Canned Cycles produce a thread using a multiple passes. Each pass will remove the same chip load (or volume of material) by adjusting the depth of the cut. The total depth of thread and the first depth of cut are programmed. The control then calculates the number of passes required based on the volume of the first depth of cut. The number of passes can be calculated using the formula below:

$$\left(\frac{\text{Total Depth of Thread}}{\text{Depth of The First Pass}}\right)^2 = \text{number of passes}$$

The first pass should be deep enough to produce a reasonable number of passes. For example, standard 14 TPI thread should take approximately 8-12 passes to cut. Example: If the total depth of thread is 0.042" and the first pass depth is 0.003", then:

$$\left(\frac{0.042}{0.003}\right)^2 = 196 \text{ passes}$$

A more reasonable first pass depth would be 0.013" which would produce 11 passes.

$$\left(\frac{0.042}{0.013}\right)^2 = 10.44 \text{ or } 11 \text{ passes}$$

To obtain a set number of passes for a given thread use the following formula:

$$\frac{\text{Final Depth of Thread}}{\sqrt{\text{Number of Passes Required}}} = \text{1st pass depth of cut}$$

Example:

$$\frac{0.042}{\sqrt{11}} = 0.0126"$$

Therefore 0.0126" = V52 variable.

E. ACCELERATION DISTANCES

When threading with the Crusader IIL it is essential that the spindle revolution and the feedrate at which the thread will be cut are coordinated. The spindle speed and feedrate must be given an acceleration distance or 'ramp time' for the spindle and slide to reach the correct velocity to cut the correct thread pitch. If this acceleration distance is not programmed then the first two or three threads could be of a smaller pitch than programmed because it is still trying to attain the correct velocity.

Feedrate in Inches/Min.	Acceleration Distance (Inches)	Feedrate in MM/Min.	Acceleration Distance (M.M.)
120	0.22	3000	5.6
100	0.18	2500	4.5
80	0.14	2000	3.6
60	0.10	1500	2.5
40	0.07	1000	1.7
20	0.03	500	0.7

To calculate the feedrate in inches/mm. per minute use the following formula:

$$\frac{1}{\text{Threads per inch/MM}} \times \text{RPM} = \text{Thread feedrate in in/mm per minute.}$$

Example: The thread to be produced has 14 threads per inch and the RPM will be 900, therefore $\frac{1}{14} \times 900 = 64.28$ IPM

Acceleration distance equals 0.11".

G. G86: LONGITUDINAL COMPOUND THREADING

The G86 Canned Cycle is a series of passes on a diameter that progressively cut deeper to form a thread. When using compound threading the Crusader IIL performs similar to a manual machine. The tool will feed in to depth at an angle (normally 60 degrees for standard threads) and then cut on the leading edge of the tool, progressively getting deeper until full depth of thread is reached.

The G86 cycle has 8 'V-codes' that have to be programmed. The definition of the V-codes is as follows (see figure 19):

V41 or V42: Either V41 or V42 can be programmed, but not both. V41 equals the pitch of the thread. V42 equals the threads per inch. V41 should always be used when programming in metric, since metric threads are always given with pitch dimensions.

V50: The incremental distance in the Z Axis that the tool will move from the tool starting position, to the end point at full depth of thread.(- value when cutting towards the chuck).

V51: The incremental full depth of thread from edge of the component (- value external, + value internal).

V52: The depth of the first pass, which is used to calculate the volume of each pass (- value external, + value internal).

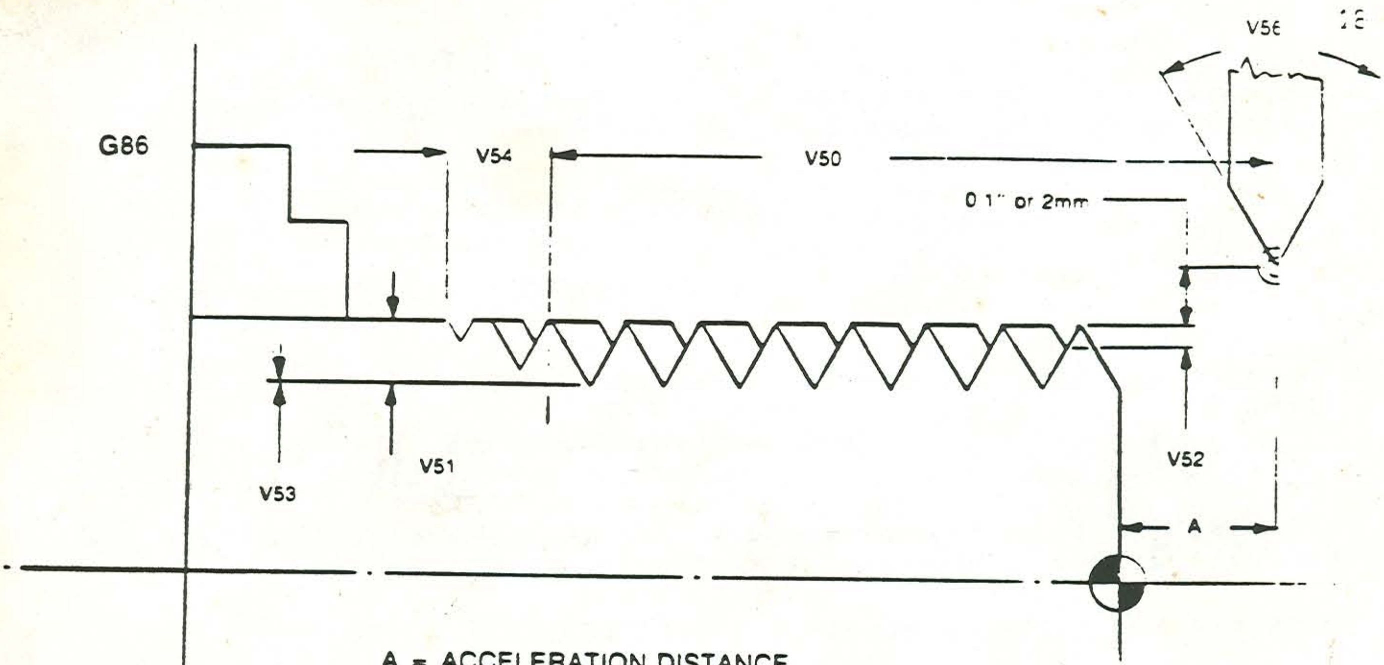
V53: The incremental depth of the taper (zero if no taper) in the X Axis. Note: The angle of the taper cannot exceed 45 degrees (+ value external, - value internal).

V54: The maximum incremental distance the tool will move along the Z Axis while pulling out of the thread (- value when cutting towards the chuck).

V55: The number of 'spring passes' the operator may program at full depth of thread to remove tool deflection or burrs. THERE MUST BE A MINIMUM OF ONE.

V56: The angle of thread.

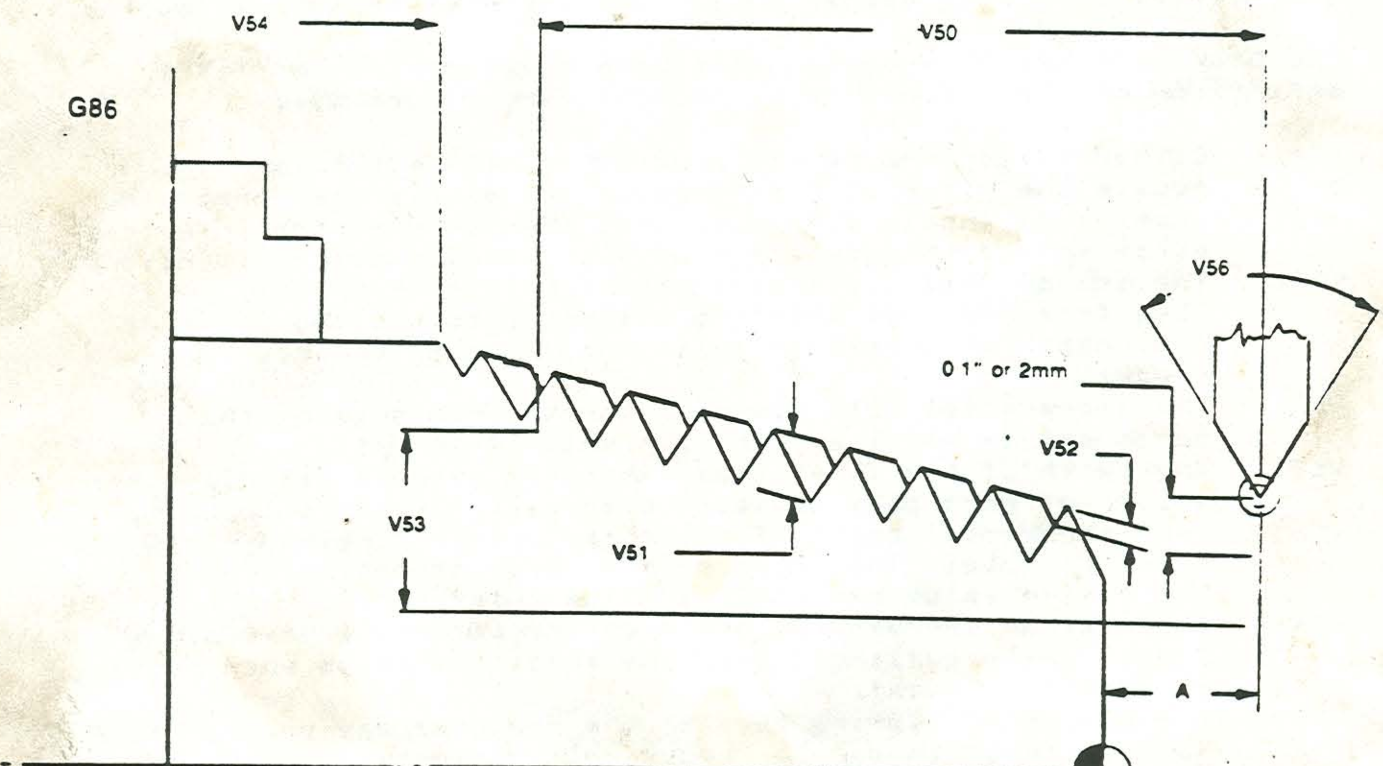
G86: Code programmed to activate the Canned Cycle.



A = ACCELERATION DISTANCE

V55 = NUMBER OF PASSES AT FINAL

V56 = ANGLE ... (60.0 ...



☆ NOTE: ALL UNASSIGNED V CODES MUST BE PROGRAMMED AS ZERO

Figure 19

NOTE:
When entering the Canned Cycle into the Crusader IIL the above format must be followed: V41 through V56, followed by G86.

H. G85: FACE PLUNGE THREADING

The G85 Canned Cycle is a series of passes on a face that progressively cut deeper to form a thread.

The tool must be positioned at the starting point shown. The G85 cycle has 7 V-codes that have to be programmed. The definition of the V-codes is as follows (see figure 20):

V41 or V42: Either V41 or V42 can be programmed, but not both. V41 equals the pitch of the thread. V42 equals the threads per inch (TPI). V41 should always be used when programming in metric, since metric threads are always given with pitch dimensions.

V50: The incremental distance in the X Axis that the tool will move from the tool starting position to the end point at full depth of thread (- value when cutting towards center line).

V51: The incremental full depth of thread from the face of the component (- value when cutting towards the chuck).

V52: The depth or the first pass which is used to calculate the volume of each pass (- value when cutting towards the chuck).

V53: The incremental depth of the taper (zero if no taper) in the Z Axis. Note: The angle of the taper cannot exceed 45 degrees.

V54: The maximum incremental distance the tool will move along the X Axis while pulling out of the thread (- value when cutting towards the center line.).

V55: The number of 'spring passes' the operator may program at full depth of thread to remove tool deflection or burrs. THERE MUST BE A MINIMUM OF ONE.

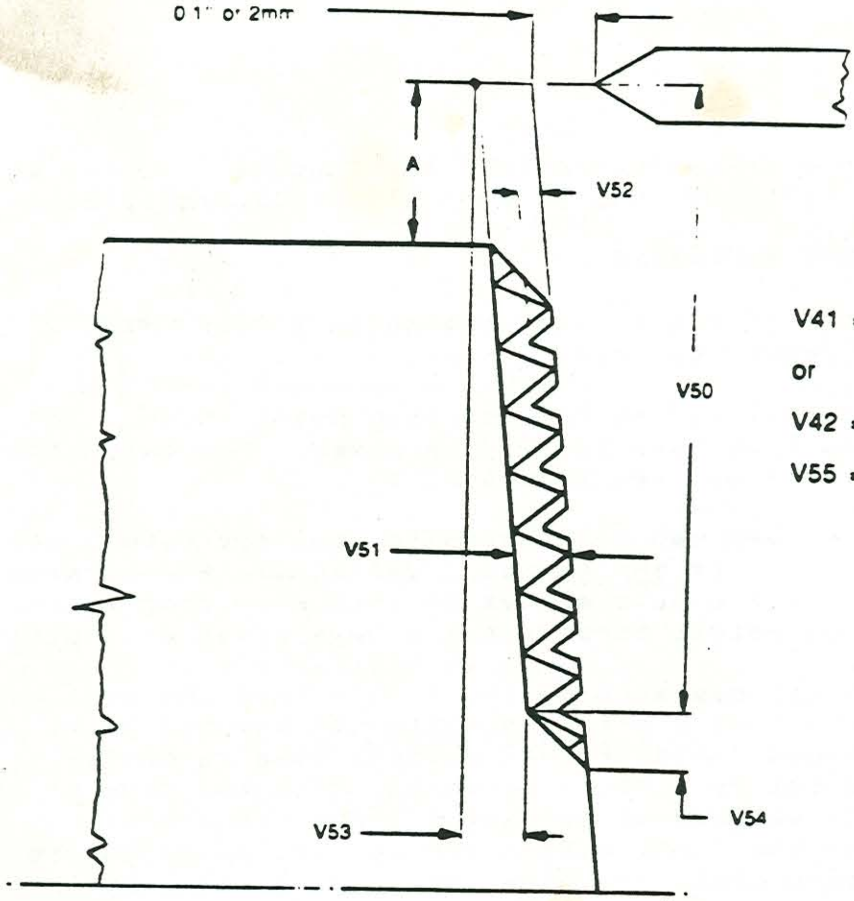
G85: Code programmed to activate the Canned Cycle.

NOTE:
When entering the Canned Cycle into the Crusader IIL the above format must be followed: V41 through V55, followed by G85.

I. G87: FACE COMPOUND THREADING

The G87 Canned Cycle is a series of passes on a face that progressively cut deeper to form a thread. When using compound threading the Crusader IIL performs similarly to a manual machine.The tool will feed in to depth at an angle (normally 60 degrees for standard threads) and then cut on the leading edge of the tool, progressively getting deeper until full thread depth is reached.

G85



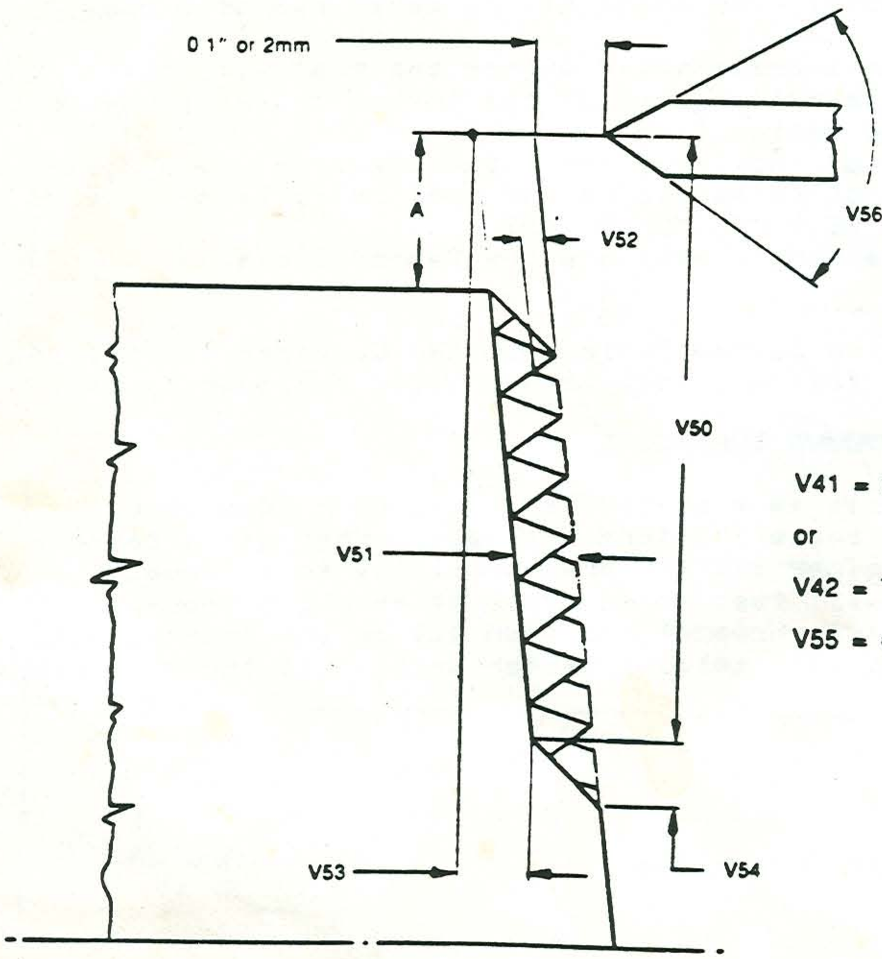
V41 = PITCH

or

V42 = T.P.I.

V55 = # OF SPRING PASSES

G87



V41 = PITCH

or

V42 = T.P.I.

V55 = # OF SPRING PASSES

Figure 20

The G87 cycle has 8 V-codes that have to be programmed. The definition of the V-codes is as follows (see figure 20):

V41 or V42: Either V41 or V42 can be programmed, but not both. V41 equals the pitch of the thread. V42 equals the threads per inch (TPI). V41 should always be used when programming in metric, since metric threads are always given with pitch dimensions.
V50: The incremental distance in the X Axis that the tool will move from the tool starting position, to the end point at full depth of thread (- value when cutting towards center line).
V51: The incremental full depth of thread from the face of the component (- value when cutting towards the chuck).
V52: The depth of the first pass, which is used to calculate the volume of each pass (- value when cutting towards the chuck).
V53: The incremental depth of the taper (zero if no taper) in the Z axis. Note: The angle of the taper cannot exceed 45o.
V54: The maximum incremental distance the tool will move along the X Axis while pulling out of the thread (- value when cutting towards the center line).
V55: The number of 'spring passes' the operator may program at full depth of thread to remove tool deflection or burrs. THERE MUST BE A MINIMUM OF ONE.
V56: The angle of thread.
G87: Code programmed to activate the canned cycle.

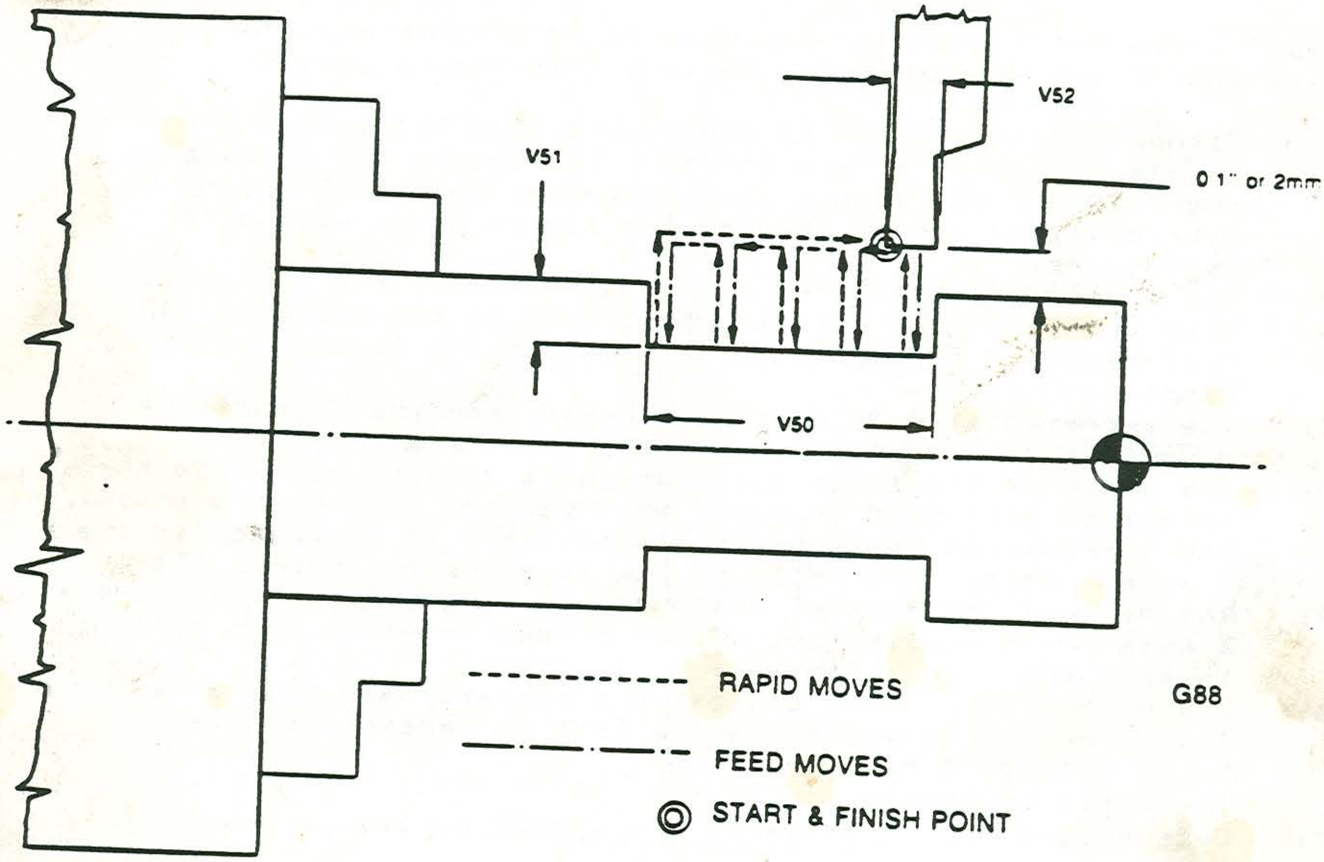
NOTE:
When entering the Canned Cycle into the Crusader IIL the above format must be followed, V41 through V56, followed by G87.

J. G88: LONGITUDINAL GROOVE CUTTING

The G88 Canned Cycle is a series of plunge cuts made on the diameter of a component to produce a groove. The Canned Cycle allows any width of groove to be cut with any known size of grooving tool. The G88 cycle has 4 V-codes that have to be programmed. The definitions of the V-codes is as follows (see figure 21):

V50: The incremental length of the groove (- value when cutting towards the chuck).
V51: The incremental depth of the groove from the outside of the component to the bottom of the groove (- value external, + value internal).
V52: The width of the grooving tool (no sign).
V53: A timed drill at the full depth of the groove. The dwell is entered in seconds (ie: 1.000 = 1 second).
G88: Code programmed to activate the Canned Cycle.

NOTE:
When entering the Canned Cycle into the Crusader IIL the above format must be followed: V50 through V53, followed by G88.


V51
V52
0.1" or 2mm
V50
RAPID MOVES
FEED MOVES
START & FINISH POINT
G88

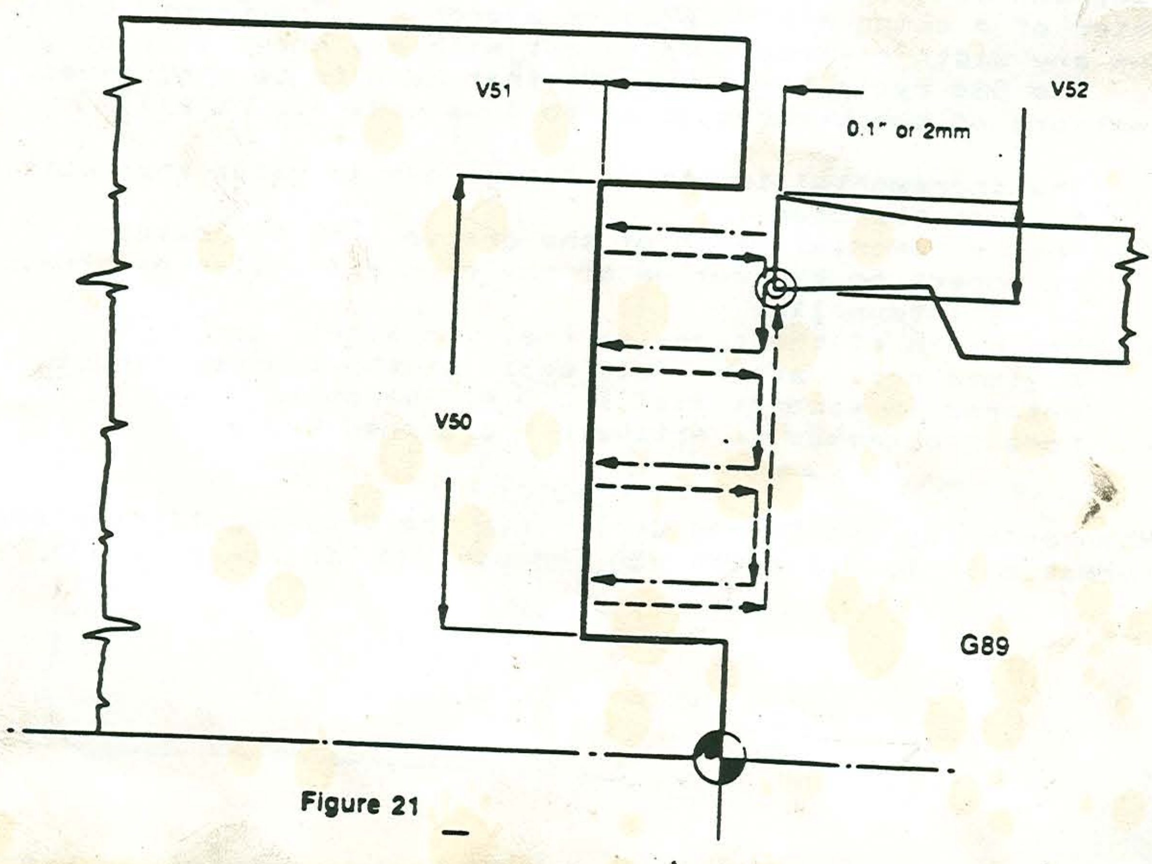

V51
V52
0.1" or 2mm
V50
G89

Figure 21

K. G89: FACE GROOVE CUTTING

The G89 Canned Cycle is a series of plunge cuts made on the face of a component to produce a groove. The Canned Cycle allows any width of groove to be cut with a known size of grooving tool. The G89 cycle has 4 V-codes that have to be programmed. The definitions of the V-codes is as follows (see figure 21):

V50: The incremental length of the groove (- value when cutting towards the chuck).
V51: The incremental depth of the groove from the face of the component to the bottom of the groove (- value).
V52: The width of the grooving tool (no sign).
V53: A timed dwell at the full depth of the groove. The dwell is entered in seconds (i.e., 1.000 = 1 second).
G89: Code programmed to activate the Canned Cycle.

NOTE:

When entering the Canned Cycle into the Crusader IIL the above format must be followed: V50 through V53, followed by G89.

L. CANNED CYCLE ERROR CODES

If an invalid value is programmed in a canned cycle, an error code will be displayed when the program is executed. The error light will come on and the number will be displayed in the upper right display.

ERROR 80: The values for turning, facing or threading variables have an input format error. Check the relationship between V50-V56 for incorrect entry.
ERROR 81: Drilling depth for G83 entered incorrectly.
ERROR 82: Tool width is greater than the groove width in the grooving cycle.
ERROR 83: Feed per revolution is less than or equal to zero, or is greater than 9999.

SECTION 11

AUXILIARY CODE LISTING

The Auxiliary (AUX) button is used in the main program to cause variations in the standard control functions. These variations are assigned numbers and are entered into the program prefixed by the AUX button (ie. AUX 7, AUX 400, AUX 2500, etc.). Below is a list of all the AUX codes and functions available in the Crusader IIL:

AUX 1000	Activate Continuous Path. This should only be used for contouring continuous tangent lines and circles and should be turned off immediately after use. It is not to be used in RAPID or when doing consecutive straight line moves or over 40 inches per minute in FEED. This command is modal and becomes active for the next motion command. It causes pre-processing to take place, so the machine does not pause between moves but executes continuous motion.
AUX 1400	FEED% override for feed moves only. Rapid moves will be made at 240 IPM. When the control is turned on, this function is automatically active each time.
AUX 1401	FEED% override for Feed and Rapid. Both Feed and Rapid moves will be affected by FEED% override.
AUX 1410	Causes rapid motion to be executed in a straight line. This is a modal command and is active at power up.
AUX 1411	Allows all axis to run at maximum rapid speed when a rapid move is programmed. The tool will not travel in a straight line when this AUX-code is active. This is a modal command.
AUX 1160	Deactivate Backlash Compensation (default).
AUX 1161	Set and activate Backlash Compensation. The following variables must be set before this AUX-code is activated: V01: .000 (The amount of backlash in the X Axis). V03: .000 (The amount of backlash in the Z Axis).
AUX 1162	Reactivate Backlash Compensation using variables established in AUX 1161.
AUX 1900	SINGLE STEP mode will execute one event each time the START button is pressed.
AUX 1901	SINGLE STEP mode will execute one X or Z move each time the START button is pressed (active at power up).
AUX 2000	Cancels AUX 1000 (active at power up).

The following AUX 2700 series are active with RS232-C option only:

AUX 2700	Write to RS232-C device in RS274 format.
AUX 2701	Read from RS232-C device in RS274 format.
AUX 2702	Write to RS232-C device in Anilam format.
AUX 2765	Set 5 bits per character.
AUX 2766	Set 6 bits per character.
AUX 2767	Set 7 bits per character.
AUX 2768	Set 8 bits per character.
AUX 2770	Set to no parity.
AUX 2771	Set to odd parity.
AUX 2772	Set to even parity.
AUX 2780	Set baud rate to 110 bits/second.
AUX 2781	Set baud rate to 150 bits/second.
AUX 2782	Set baud rate to 300 bits/second.
AUX 2783	Set baud rate to 600 bits/second.
AUX 2784	Set baud rate to 1200 bits/second.
AUX 2785	Set baud rate to 1800 bits/second.
AUX 2786	Set baud rate to 2400 bits/second.
AUX 2787	Set baud rate to 4800 bits/second.
AUX 2788	Set baud rate to 9600 bits/second.
AUX 2789	Set baud rate to 19,200 bits/second.

NOTE:

Any speed higher than 1200 bits/second will transfer data in bursts. Must use handshake.

AUX 2790	Set no handshake.
AUX 2791	Set software handshake (X on, X off)
AUX 2792	Set hardware handshake (DTR, DST)

SECTION 12

MISCELLANEOUS

A. ERROR LIGHT AND ERROR CODE LISTINGS

This light (along with a number displayed in the right half of the display at the top left of the console) indicates that an error has been made and identifies it. The following list defines what each number indicates:

1. Too many digits entered into the keyboard. The control will only accept dimensions from 999.9999 to 000.0001" (9999.99 to 0000.01mm) in the axis display registers.

2. No cassette recorder power. The cassette unit does not have power to turn the tape winding mechanism.

3. Cassette tape not in place, or no tape in cassette recorder unit.

4. Program cannot be recorded because the safety tab is missing. After a tape has been proven, the tape cannot be erased or changed once this tab is removed.

5. No program in the memory to record.

6 and 7: The tape has not played correctly. Look at the program using Program Check and replay the tape if it is not correct.

9. Tape stopped. Check for correct recording/playing procedure.

10: The program did not record on the cassette.

11: Arc coordinates are entered incorrectly. Either only one axis was described or the Arc Statement was omitted after the end point.

12: Software limit reached.

13: Compensated elements do not intersect when cutter compensation is active.

16: Acceleration too high.

18: No motion seen by the control when a command is given. Probable causes are:
   - a. Encoder cable disconnected.
   - b. D/A signal disconnected.
   - c. Servo amp fuse blown.
   - d. Motor disconnected.
   - e. Pulley or belt not functioning.
   - f. Gib strip too tight.
   - g. Ballscrew inoperable.

19: Following error increasing too fast.

20: Velocity command too fast.

30: Non-existant AUX 2700 Code.

31: User abort of RS-232 data transfer by pressing Emergency Stop.

32: Program memory overflow while reading RS-232 data.

33: Input error while reading RS-232 data.

34: Input error while outputting the program using X on, X off, and using RS-232.

35: Buffer overflow when using RS-232.

80: V53 or V54 greater that V51, while using G81 and G82 cycles.

81: V50 has been programmed as a positive number while using G83.

82: Tool width is great than groove width.

83: Feed per revolution number is less than or equal to zero or greater than 9999.

84: Feed per revolution calculated to be more than the machine limit.

B. DIAGNOSTIC MODE

The Crusader IIL has a diagnostic or self test feature, which can check through all the functions of the system.

TO APPLY DIAGNOSTIC MODE

1. Make sure that any important program is recorded on cassette. A diagnostic check erases everything from the memory.
2. Press EMERGENCY STOP.
3. Disconnect the X, Z and Spindle Encoder cables from the rear of the Crusader console.
4. Press PROGRAM ENTER.
5. Press the DECIMAL POINT (5 times).
6. Press START.

The Diagnostics will run for approximately 30 seconds and then stop.

TO RESTORE NORMAL WORKING

1. Press DECIMAL POINT.
2. Press START.
3. Reset EMERGENCY STOP.
4. Press RESET button to start the drive motors. While console is running in the diagnostic mode, the displays show changing numbers as each function is checked. If a fault is detected, the same number flashes continuously. In such a case, please contact the Service Department at Anilam Electronics Corp., Miami and quote the flashing number.

C. CRUSADER IIL GUIDELINES

By following these simple guidelines, the programmer or operator should reduce costly programming errors and downtime.

1. After programming tool numbers and their assigned tool length offsets, always program TOOL 0, followed by X0, Z0 RAPID ABS. This ensures that the tool post is at the tool change position before any operation begins. This must also be used between tool changes, and at the end of the program.
2. Program a feedrate after each tool change, even if it is the same as the preceding feedrate.
3. Subroutines must be entered after the main program.
4. Always write the program down and update the written copy after editing.
5. Write RAPID or FEED and the movement mode (Absolute or Incremental) for every program event that causes motion.
6. Program tools in the sequence they will be used starting with TOOL 1. Keep a written record with the description of each tool on it and the spindle speed and feedrate used for that tool.
7. For ARC's which start and end on quadrants, incremental programming usually eliminates calculations.
8. Only ARC center and end points, ABS or INCR and FEED may be programmed between the first arc (CW or CCW) and the final arc (end of Arc Statement). Changes in feedrate or any other command must not be programmed between the arc commands.
9. After G40 has been programmed there must be an X or Z move before the end of the program.
10. Remember to program RAPID moves separately when retracting from a bore.
11. Always start the program either at Event #1, or a Tool 0, before the tool change at which it is required to start.
12. If in doubt, read the relevant section of the manual and fully understand it before starting to program.
13. THINK!

*** PART TWO ***
(Programming with the Crusader IIL)

SECTION 13

PROGRAM PLANNING

When planning a component it is recommended that a set format is followed. By following the recommended procedure, time consuming set-up errors can be avoided, thereby increasing productivity. Follow the steps detailed below:

1. From the component drawing, the programmer or operator decides the machining sequence, tooling requirements and speeds and feeds to be used. This information can then be documented on a Planning/Tooling sheet, as shown in the example.

2. Working from the Planning/Tooling sheet and the component drawing, the program is written.

3. The program is then loaded in the Crusader IIL control.

4. The tool offset values for each individual tool are then set and entered into the program.

5. The program can then be stepped through event by event (program proving in the SINGLE STEP mode) to ensure there are no obvious programming errors.

6. Next, a blank is loaded into the machine and cut. Again, the SINGLE STEP mode is used to check for errors. If any errors are found, the operator or programmer can stop the machine and make the corrections necessary to complete the component.

SECTION 14

PROGRAMMING

A. POSITIONING MOVES IN RAPID

Rapid moves can be used when a tool has to be positioned either before or after a feed move, or any time when the tool is not to contact the component (ie. returning to Absolute Machine Zero, positioning to take a cut, repositioning after a cut, etc). It is important when using RAPID that the operator or programmer does not rapid directly to the component, but allows clearance between the tool tip and component. An ideal and easily remembered figure is 0.1" or 2mm (see figure 22).

Tool 1: Turning position is .1"(2mm) from front face.
Tool 2: Facing position is .1"(2mm) above component diameter.
Tool 3: Grooving position is .1"(2mm) above groove diameter.
Tool 4: Boring position is .1"(2mm) from front face.

The distance .1"(2mm) is also ideal since all canned cycles have to begin .1"(2mm) from the cutting surface (see Section 10). Therefore, this figure can become a constant.

To program a positioning move to X 1.5 Z 0 use the following steps:

1. Press PROGRAM ENTER.
2. If the control is in the MM mode, select INCH.
3. Press X 1 (decimal point) 5 Z 0.
4. If the RAPID/FEED button is displaying FEED, press the button to change it to rapid.
5. Press EVENT ENTER. The event number (top left display window) will advance by one, signifying that the system has entered the command into memory.

NOTE: Absolute zero and tool length offsets must be set before programming the above steps.

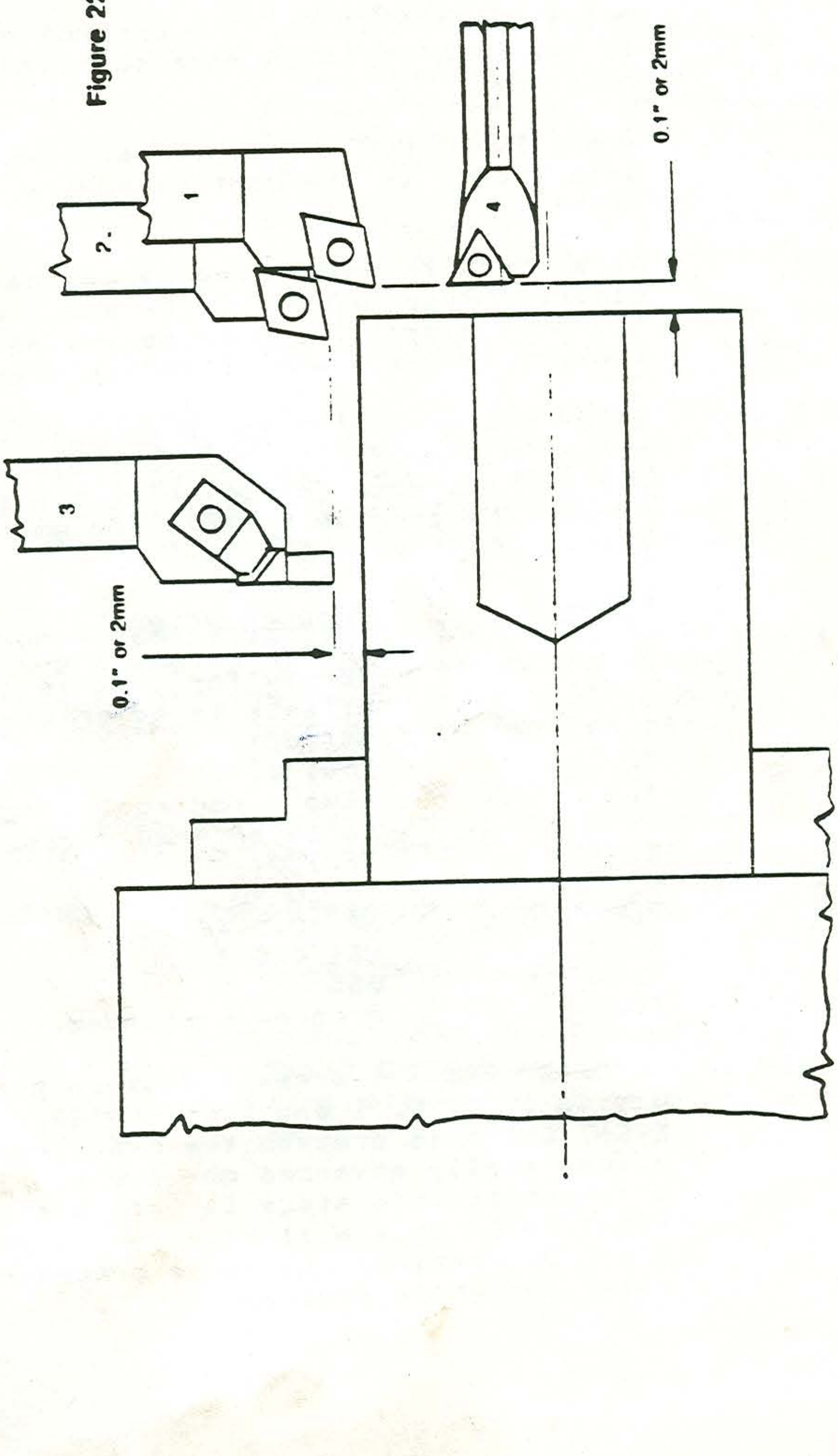


Figure 22

SECTION 15

ENTERING THE PROGRAM INTO THE CONTROL

Before attempting to enter a new program into the control manually, the first two steps from the 'Program Planning' section should be followed. These two steps are important because they help eliminate costly errors during the set-up procedure.

After the program has been planned and written, the procedure is as follows:

Step 1. Press PROGRAM ENTER. The control has to be in the PROGRAM ENTER mode to clear or load programs.

Step 2. Press EVENT CLEAR five times. Any previous data stored in the control memory is now deleted.

Step 3. As a check to ensure the memory has been cleared press PROGRAM CHECK and then START. The only display, should be the advancing event in numbers when START is pressed.

Step 4. After checking that the memory is clear press PROGRAM ENTER. The control is now in the correct mode for entering the program. Review the program example for manual entry below:

Event #	Program Data
1.	Tool 2001
2.	(Empty for tool length offsets to be entered during set up)
3.	Tool 2002
4.	(Empty for tool length offset to be entered during set up)
5.	Tool 0
6.	X0 Z0 RA
7.	X2 Z0.1 RA
8.	V21 0.012
9.	G95
10.	Program continued.

Step 5. To enter EVENT 1 press TOOL then press digits 2, 0, 0, 1 and EVENT ENTER. When EVENT ENTER is pressed the event number automatically advances one event. For event 2 (which at this stage is left empty) press EVENT ENTER this will advance the program to Event 3. Continue the above procedure, to enter the whole program.

A. RECORDING AND RE-ENTERING RECORDED PROGRAMS

After a program has been entered into the control manually, and proven to be correct, the Lathemate has the ability to record the data for subsequent run offs of the same component. Located at the front of the control console is the Program Saver mini-cassette recorder. The Program Saver is also utilized for the re-entry of recorded programs.

B. RECORDING PROCEDURE

Step 1.	Press PROGRAM ENTER.
Step 2.	Press Emergency Stop. If the EMERGENCY STOP is not depressed, the servo motors will automatically shut down when PROGRAM SAVER is activated.
Step 3.	Put a cassette tape into the recorder and close the cassette loading door.
Step 4.	Press RECORD in the PROGRAM SAVER unit. The red Indicator lamp will light.

At this time, the tape will rewind, then the MANUAL indicator lamp will light. The PROGRAM CHECK indicator lamp will blink, meaning that Channel 1 is being recorded on, and stop blinking when Channel 1 has completed its recording. The PROGRAM CHECK lamp will blink again when Channel 2 is being recorded on, the tape will then rewind.

Now, the PROGRAM ENTER indicator lamp will blink meaning that Channels 1 and 2 are being compared to the memory. The tape will then again rewind and the MANUAL lamp will shut off. A tone will sound signifying the recording sequence has been completed.

NOTE:

If an error code is displayed during the sequence, refer to the error code listings at the end of this section.

C. RE-ENTERING A RECORDED PROGRAM INTO MEMORY

Step 1.	Press PROGRAM ENTER.
Step 2.	Press EVENT CLEAR five times. By pressing EVENT CLEAR, any previous data held in the control memory is deleted, and the memory becomes clear.

Step 3. Press EMERGENCY STOP. If the EMERGENCY STOP is not depressed, the servo motors will automatically shut down when the PROGRAM SAVER unit is activated.

Step 4. Put the pre-recorded cassette tape into the recorder and close the cassette loading door.

Step 5. Press PLAY in the PROGRAM SAVE unit. The red indicator lamp will light.

At this point, there may be a slight pause then the PLAY light will come on and the tape will rewind. The MANUAL indicator lamp will then light. At this time the PROGRAM ENTER indicator lamp will blink while the recorded program is being entered into the memory. The tape will rewind again, and the MANUAL indicator lamp will shut off. A tone will sound signifying that the program has been played into the memory.

To check that the program has been entered, press PROGRAM CHECK and the first event of the program will be displayed on the control. Continuous key strikes advances the program event by event.

NOTE:

If during the sequence an error code is displayed refer to the error code listings in Section 12.

D. **ERROR LIGHT**

When certain errors are made a tone will sound, the error light will come on, and an error number will be displayed in the upper right window. To demonstrate this, press PROGRAM ENTER and press RECORD. Do not put a tape in the cassette loading door. The ERROR light will come on and error number **3** will be displayed.

SECTION 16

ABSOLUTE AND INCREMENTAL PROGRAMMING

For a basic knowledge of absolute and incremental dimensioning, please review Section 3. It is important that the operator have a solid understanding of absolute and incremental dimensioning and programming when working with the Crusader IIL.

A. ABSOLUTE PROGRAMMING

In figure 23 the component is shown with absolute dimensions. The program below shows how easily the co-ordinates can be taken from the print and entered into the program. In the example the tool is positioned at a tool change position (or Absolute machine zero). This position in the X Axis is 2.5" from the center line and 1.5" from the front face (Z Axis). These distances are known as tool length offsets. See Section 8 for an explanation.

The program will take the tool from absolute machine zero through points 1-9.

Event #	Command	
10.	TOOL 1	Activate Tool #1 offsets
11.	V21.008	Select a feedrate of 0.008"
12.	G95	Select feedrate in I.P.R.
13.	X0 Z0.1 RA	Rapid positioning move (see Section 9)
14.	Z 0 FA	Feed to point (1)
15.	X 0.25 FA	Feed to point (2)
16.	Z-1.0 FA	Feed to point (3)
17.	X 0.5 FA	Feed to point (4)
18.	Z-1.5 FA	Feed to point (5)
19.	X 0.75 FA	Feed to point (6)
20.	Z-2.75 FA	Feed to point (7)
21.	X 1.05 FA	Feed to point (8)
22.	X 1.5 Z-3.125 FA	Feed to point (9)

B. INCREMENTAL PROGRAMMING

The programming example below will show figure 23 programmed in the Incremental mode. The tool is positioned at Absolute Machine Zero (as in the previous Absolute example) and will move through the same points.

Event #	Command	
10.	TOOL 1	
11.	V21.008	
12.	G95	
13.	X-2.5 Z-1.4 RI	Rapid positioning move incrementally

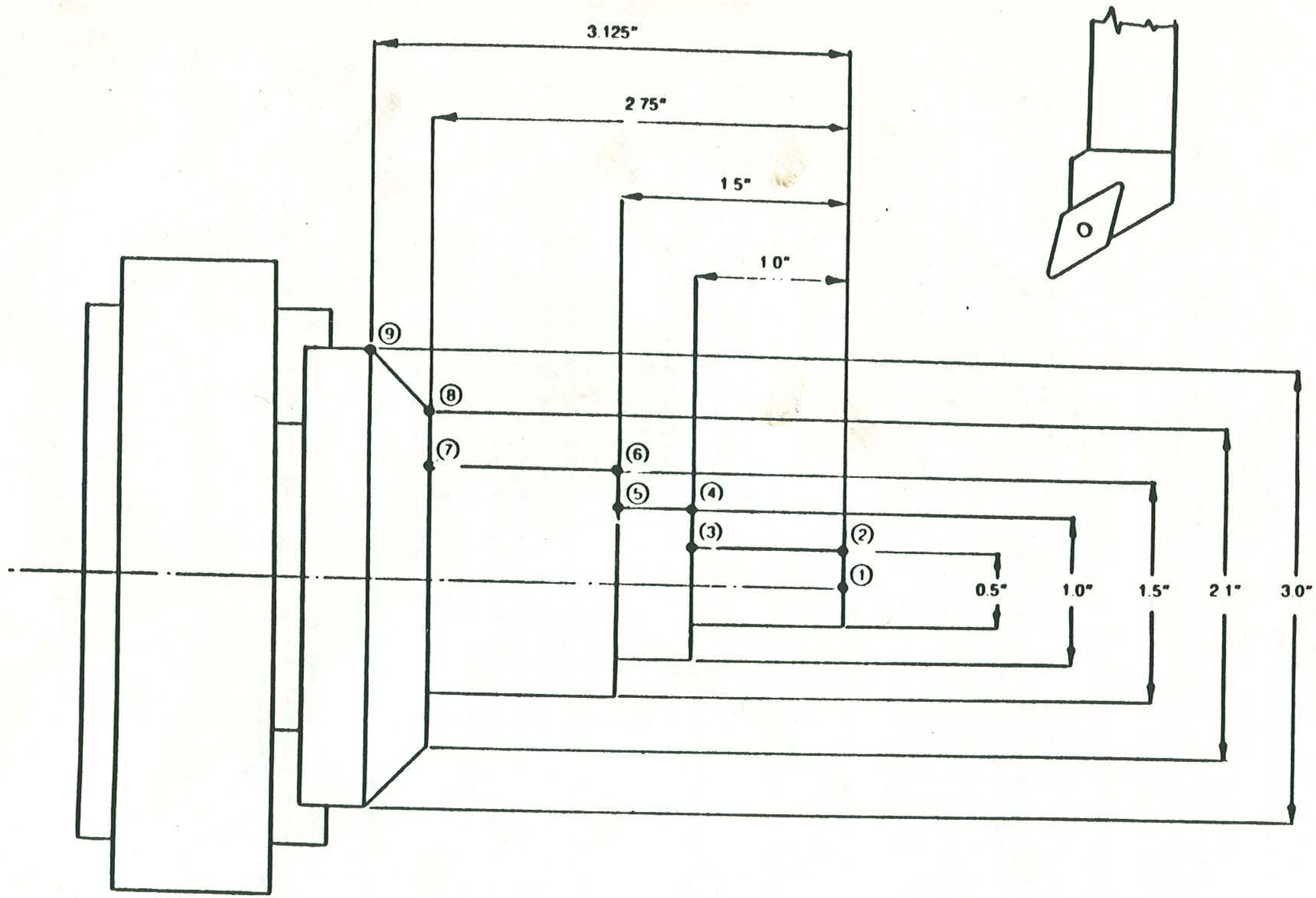
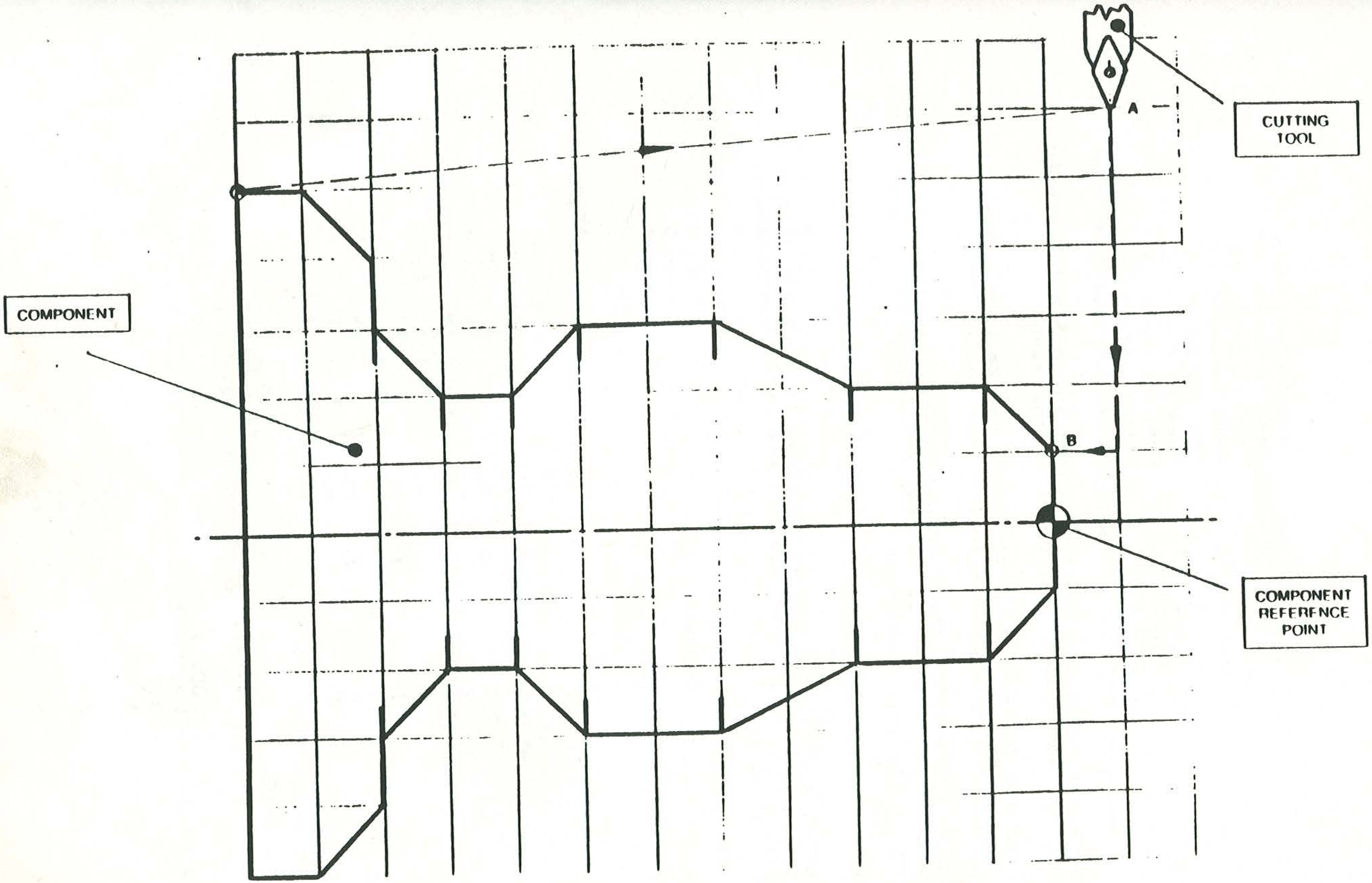


Figure 23



ASSUME EACH SQUARE IS EQUAL TO 0.1"

Figure 24

14.	Z-0.1 FI	Feed to point (1)
15.	X 0.25 FI	Feed to point (2)
16.	Z-1.0 FI	Feed to point (3)
17.	X 0.25 FI	Feed to point (4)
18.	Z-0.5 FI	Feed to point (5)
19.	X 0.25 FI	Feed to point (6)
20.	Z-1.25 FI	Feed to point (7)
21.	X 0.3 FI	Feed to point (8)
22.	X 0.45 Z-0.375 FI	Feed to point (9)

The program illustrates how the tool takes the present position as a datum point, and moves from that point to the next.

Both Absolute and Incremental programming can be used within the same program but cannot be mixed in the same event.

Programming in Incremental could create difficulties because any error in calculating a point is carried throughout the program. Each move relates to the previous move.

C. CUSTOMER PROGRAMMING EXAMPLE

Figure 24 shows a part that you can program in both Absolute and Incremental. Check your program against the correct programming shown in Section 16-D.

ABSOLUTE

Event #	Command		
10.	TOOL 1		
11.	V21.008		
12.	G95		
13.	X		RA
14.	Z		FA
15.	X	Z	FA
16.	Z		FA
17.	X	Z	FA
18.	Z		FA
19.	X	Z	FA
20.	Z		FA
21.	X	Z	FA
22.	X		FA
23.	X	Z	FA
24.	Z		FA
25.	X	Z	RA

INCREMENTAL

Event #	Command		
10.	TOOL 1		
11.	V21 .008		
12.	G95		
13.	X		RI
14.	Z		FI
15.	X	Z	FI
16.	Z		FI
17.	X	Z	FI

18.	Z		FI
19.	X	Z	FI
20.	Z		FI
21.	X	Z	FI
22.	X		FI
23.	X	Z	FI
24.	Z		FI
25.	X	Z	RI

D. ANSWERS FOR ABSOLUTE AND INCREMENTAL TEST PROGRAMS

ABSOLUTE

10.	TOOL 1	Activate Tool #1 offsets.
11.	V21 .008	Select feedrate of .008".
12.	G95	Select feedrate in IPR.
13.	X .1 RA	Rapid positioning move.
14.	Z 0 FA	Feed to point B.
15.	X .2	
16.	Z -.3 FA	
17.	X .3 Z -.5 FA	
18.	Z -.7 FA	Feed moves around the contour
19.	X .2 Z -.8 FA	of the component.
20.	Z -.9 FA	
21.	X .3 Z -1. FA	
22.	X .4 FA	
23.	X .5 Z -1.1 FA	
24.	Z -1.2 FA	Rapid return to Point A.
25.	X .6 Z .1 RA	

INCREMENTAL

10.	TOOL 1	Activate Tool #1 offsets.
11.	V21 .008	Select feedrate of .008".
12.	G95	Select feedrate of IPR.
13.	X -.5 RI	Rapid positioning move.
14.	Z -.1 FI	Feed to Point B.
15.	X .1 Z -.1 FI	
16.	Z -.2 FI	
17.	X .1 Z -.2 FI	
18.	Z -.2 FI	
19.	X -.1 Z -.1 FI	Feed moves around the contour
20.	Z -.1 FI	of the component.
21.	X .1 Z -.1 FI	
22.	X .1 FI	
23.	X .1 Z -.1 FI	
24.	Z -.1 FI	Rapid return to Point A.
25.	X .1 Z 1.3 RI	

SECTION 17

PROVING A PROGRAM

A. PROGRAM PROVING IN SINGLE STEP MODE

SINGLE STEP program proving is a safety procedure in which the operator systematically checks the validity of the program.

Program proving is carried out in the SINGLE STEP mode of operation so that when the START key is pressed, the control issues only one event of commands. On the completion of the single event the machine cannot proceed until the START key is again pressed. This sequence of operation is repeated until the end of the program is reached.

This procedure can initially be done without a component to provide visual confirmation that the commanded movements of the slides are in the intended manner of the program. The same procedure can then be used to cut the first component event by event.

Proving a program in the SINGLE STEP mode is not, however, simply a matter of repeatedly pressing the START key to see what happens. Before pressing START to action one event, the operator should always study a copy of the program, component drawing and...

1. Anticipate what will happen in the event.
2. Decide if the commanded movements can be executed safely.
3. Decide if the commands are in fact the commands required.
4. Alter incorrect commands if necessary.

The operator maintains complete control of the machine throughout the entire program proving procedure and can at any time...

1. Stop the cycle.
2. Override the control commands.
3. Monitor the displays of the X and Z Axis.
4. Modify the program.

B. PROGRAM MODIFICATIONS

After a program has been written and entered into the control it is essential that the program is checked and proven to be correct. Do not assume that a program is correct until it is proven. By following the program proving sequence, time consuming errors can be eliminated.

Below is a summary of operations that may be necessary when proving a program. A detailed explanation of each operation can be found on the following pages.

C. EVENT SEARCH

To initiate an EVENT SEARCH (i.e; jump to a specific event number) the console may be in any mode of operation but the following sequence must be adhered to:

1. Press EVENT CLEAR.

2. Insert the desired event number for the example we will use event number 241). The event number entered will be displayed in the top left corner marked EVENT.

3. Press EVENT SEARCH. The control will search for the desired event (241).

D. ADD EVENT

The ADD EVENT feature allows the operator to insert additional events within the program. When this is done the event numbers automatically move down to accomodate the new event, therefore keeping a numerically sequencial format. To add an event:

Step 1. Press PROGRAM ENTER. Program enter is the mode in which all program modifications are made.

Step 2. Search for the place in the program where the event needs to be added. Press ADD EVENT (Event 22 will be used in the example). The presently displayed event will move down by one event number. For example:

Event #	Programmed Data	Event #	Programmed Data
20.	X1.5 Z-2.0 FA	20.	X1.5 Z-2.0 FA
21.	X1.75 Z-2.25 FA	21.	X1.75 Z-2.25 FA
22.	TOOL 0	22.	(The Added Event)
23.	X0 Z0 RA	23.	TOOL 0
		24.	X0 Z0 RA

When ADD EVENT is pressed event 22 becomes empty and the following events move down by one event number. The data to be entered in the new event is: X2.0 RA.

Step 3. Input the new data. The red lamps will indicate RAPID or FEED. Absolute is determined by the lamp on the incremental key. If the lamp is not lit the mode of operation will be absolute.

Step 4. Press EVENT ENTER. This will enter the new data into event number 22. Example:

Event #	Programmed Data
20.	X1.5 Z-2.0 FA
21.	X1.75 Z-2.25 FA
22.	X2.0 RA
23.	TOOL 0
24.	X0 Z0 RA

E. DISPLAY CONTENTS OF EVENTS

The display of events contained in a program can be carried out by two different methods.

1. Display the content of each event in their numerical sequence. (i.e., event by event) or...
2. Display the contents of a specific event.

Method 1:

Step 1. Press PROGRAM CHECK.

Step 2. Press START. Pressing START advances the program by one event. Subsequent key strikes advance the program event by event.

NOTE:
When in the PROGRAM CHECK mode the program can be moved forwards or backwards event by event. The START key moves the program forwards. The HOLD key moves the program backwards.

Method 2:

Step 1. Press PROGRAM CHECK.

Step 2. Input event number to be displayed. Example: Event number 126.

Step 3. Press EVENT SEARCH. The control will now display the contents of event 126.

F. DELETE EVENT

The DELETE EVENT feature allows the operator to delete unwanted events within the program. When an event is deleted the event NUMBERS automatically move up, therefore keeping a numerically sequential format.

To delete an event:

Step 1. Press PROGRAM ENTER. PROGRAM ENTER is the mode in which all program modifications are made.

Step 2. SEARCH for the unwanted event. Press DELETE EVENT. The presently displayed event will be deleted and the following events will move up by one event number.

G. EVENT MODIFICATIONS

When an event modification is necessary the control must be in the PROGRAM ENTER mode of operation. The operator is unable to edit any information in the program unless he is in the PROGRAM ENTER mode.

Any changes that have to be made to an event can be listed under three headings:

1. Add data to an event.
2. Edit data in an event.
3. Delete data from an event.

H. ADD DATA TO AN EVENT

Step 1. Press PROGRAM ENTER.

Step 2. Search for the event to be modified. Input the event number to be changed (for example EVENT 73). The event number will be displayed in the top left side window marked EVENT/FEED %.

Step 3. Press EVENT SEARCH. When EVENT SEARCH is pressed the program searches for the event to be changed (i.e. EVENT 73).

Step 4. The additional data can now be entered into the control. See the example below:

Event_#	Program_Data
72.	X2.125 Z-3.75
73.	X2.25 (Z-4.3)
74.	Z-4.5

Additional data (Z-4.3) was entered into Event 73.

Step 5. Press EVENT ENTER. The additional data is now entered and stored in the program.

I. EDIT DATA IN AN EVENT

To edit data in an event follow steps (1-3) in the 'Add Data to an Event' section, then continue with the following:

Step 4. When steps 1-3 have been completed the control will display the event to be edited. For example: **EVENT 73 should read X2.25 Z-4.3. The Z figure is incorrect.**

Step 5. Press the Z key twice. The first strike deletes the existing incorrect value, and the second strike makes Z ready for the correct value to be inserted.

Step 6. To insert the correct Z value follow steps (4) and (5) from the 'Add Data' section. The editing sequence is now complete and the correct Z value is in the program.

J. DELETE DATA FROM AN EVENT

To delete data from an event follow the steps in the section 'Add Data To An Event' from 1-4. When step 4 is reached the operator only has to strike the Z key once, this deletes the unwanted data. The deletion from the event is now complete.

SECTION 18

PROGRAMMING EXAMPLES OF DO LOOPS & SUBROUTINES

In this section the examples cover the use of DO LOOPS and SUBROUTINES, which can simplify the program when various operations are repeated and help by reducing the chance of programming errors occuring in repetitive moves.

A. DO LOOPS

When the same operation or series of operations has to be done several times, the DO LOOP becomes important. Only the first complete sequence is programmed and this is then repeated as many times as required. This is done by pressing DO and then the number of times the routine is to be repeated. Next, the information is programmed that has to be repeated, and then END is programmed. Note that END is not the end of the program, but only of the DO LOOP. A DO LOOP must finish with an END statement. At any time when the same commands are to be used at equal increments a Do Loop can be used. See figure 25.

B. THE USE OF A DO LOOP

In the program portion below, Events 34-43 is the actual Do Loop sequence for the example shown.

Event #	Command	
30.	TOOL 1	Assign tool length offsets to tool #1.
31.	V21 .003	Select a feedrate.
32.	G95	Feedrate in inches per revolution.
33.	X 1.1 Z -.4 RA	Rapid to start position.
34.	DO 3	Initiate the DO Loop and number of cycles.
35.	X .85 FA	
36.	X 1.0 RA	
37.	Z -.025 FI	Moves within the DO Loop.
38.	X -.025 Z .025 FI	
39.	X 1.0 RA	
40.	Z .025 FI	
41.	X -.025 Z -.025 FI	
42.	X 1.1 RA	
43.	END	

C. SUBROUTINES

A SUBROUTINE can be used to repeat a mini-program which occurs several times during the main program. This mini-program is entered into the memory after the end of the main program, is labeled as a SUBROUTINE, and can then be called whenever it is required. For example, if the main program has 137 events the SUBROUTINE could start at event 138 or any later number. Any number of SUBROUTINES can be programmed within the memory capacity.

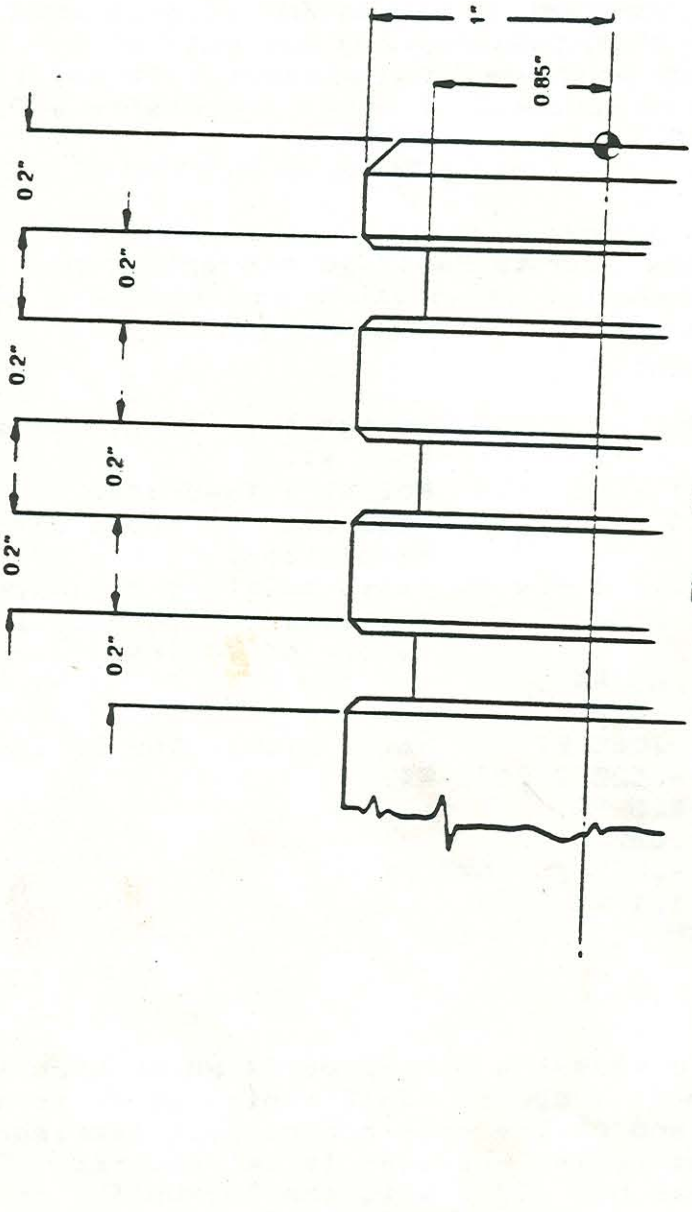


Figure 25

As a general rule, features which are repeated several times at set pitches can be programmed as Do Loops, and features which are repeated at arbitrary positions lend themselves to Subroutines. A Subroutine can be used in conjuction with a Do Loop.

Figure 26 is an example of a Subroutine for tapered grooves.

To call Subroutine 1 in the appropriate place in the main program, the CALL button is pressed with the Subroutine number. In the example, if a Subroutine is required immediately after events 24, 26 and 28, the events below (25, 27 and 29) are programmed CALL 1. At each event where the Subroutine is called the control jumps to the Subroutine and follows the programmed moves. The control reads END (last event in the Subroutine) and then jumps back to the main program and to the event following the CALL command (26, 28 & 30).

Event #	Command	
21.	TOOL 4	Assign tool length offsets to tool #4.
22.	V21.004	Select a feedrate.
23.	G95	Feedrate in inches per revolution IP.R.)
24.	X 1.1 Z -1.25 RA	Position to starting point for SUB 1.
25.	CALL 1	Call SUB 1 for first groove.
26.	Z -2.5 RA	Position to starting point for SUB 1.
27.	CALL 1	Call SUB 1 for second groove.
28.	Z -3.75 RA	Position to starting point for SUB 1.
29.	CALL 1	Call SUB 1 for third groove.
30.	TOOL 0	Cancel T L O.
31.	X 0 Z 0 RA	Rapid to Absolute Machine Zero.
32.	END.	End of program.
Subroutine 1 (Grooving):		
33.	SUB 1	Assign number 1 to subroutine.
34.	X .75 RA	
35.	X 1.0 RA	
36.	Z -.25 FI	
37.	X -.25 Z .25 FI	Grooving sequence of events, in both absolute and incremental.
38.	DWELL 2.0	
39.	X 1.1 RA	
40.	Z .25 FI	
41.	X .75 FA	
42.	X 1.0 RA	

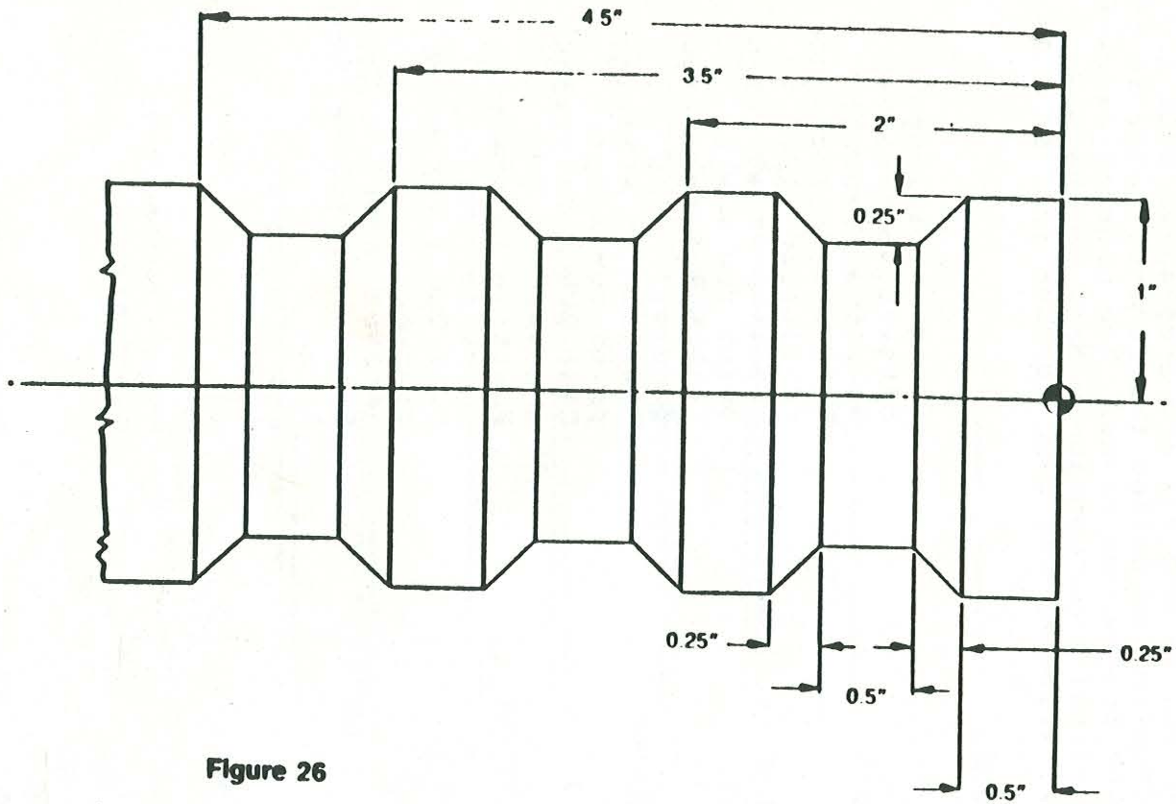


Figure 26

43.	Z .25 FI	
44.	X -.25 Z -.25 FI	
45.	DWELL 2.0	
46.	X 1.1 RA	
47.	END.	End of subroutine.

D. CASTING STOCK REMOVAL USING A SUBROUTINE

When it becomes necessary to machine a casting, a Subroutine can be utilized to remove the stock without having to program each individual move. Figure 27 below shows both a casting and the programming steps needed to rough and finish turn the diameters:

Event #	Command	
11.	TOOL 1	Assign tool length offsets to tool #1.
12.	V21 .015	Select a roughing feedrate.
13.	G95	Feedrate in IPR.
14.	X .7 Z .1 RA	Rapid to start point.
15.	CALL 1	Subroutine call for first pass.
16.	X .6 RA	Position for second pass.
17.	CALL 1	Subroutine call for second pass.
18.	V21 .008	Select a finishing feedrate.
19.	G95	Feedrate in IPR.
20.	X .55	Position for finishing pass.
21.	CALL 1	Subroutine call for finishing pass.
22.	TOOL 0	Cancel TLO.
23.	X 0 Y 0 RA	Rapid to Absolute Machine Zero.
24.	END	End of program.

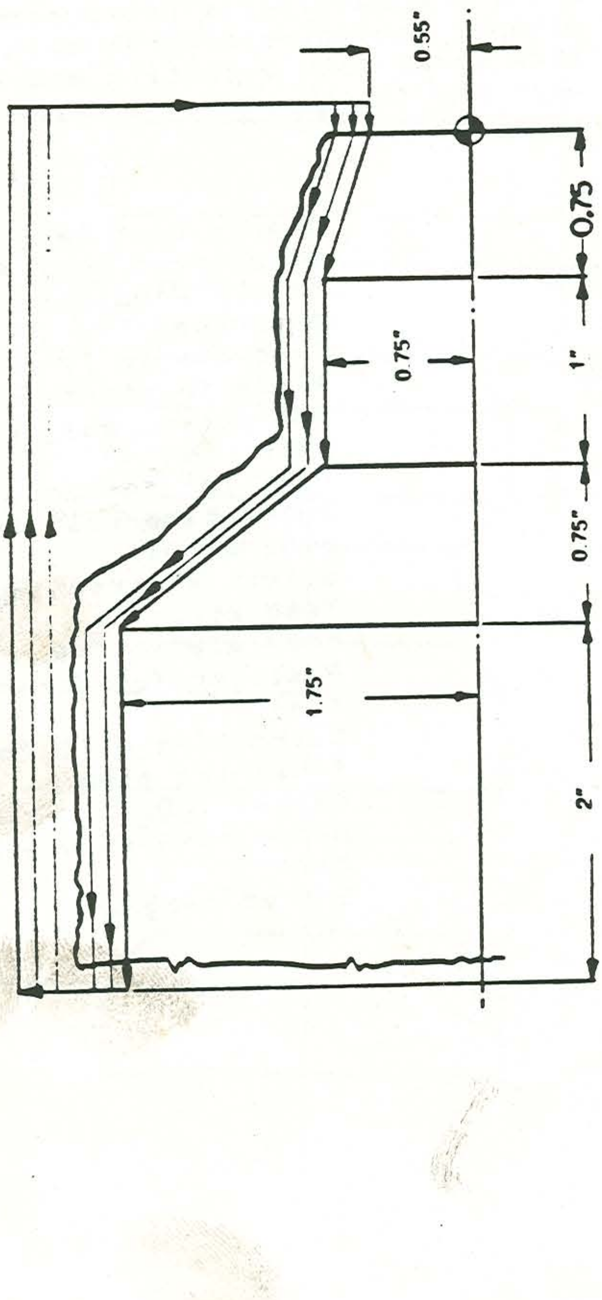


Figure 27

SUBROUTINE_1 (Casting Stock Removal):

Event_#	Command	
25.	SUB 1	Assign number 1 to subroutine.
26.	Z 0 FA	Feed in Absolute to front face.
27.	X .2 Z -.75 FI	
28.	Z -1.0 FI	Component profile in incremental moves.
29.	X 1.0 Z -.75 FI	
30.	Z -2.0 FI	
31.	X .2 FI	
32.	Z .1 FA	Z Axis return in Absolute.
33.	END	End of Subroutine.

E. CORNER BREAKING WITH A CHAMFER OR RADIUS USING A SUBROUTINE

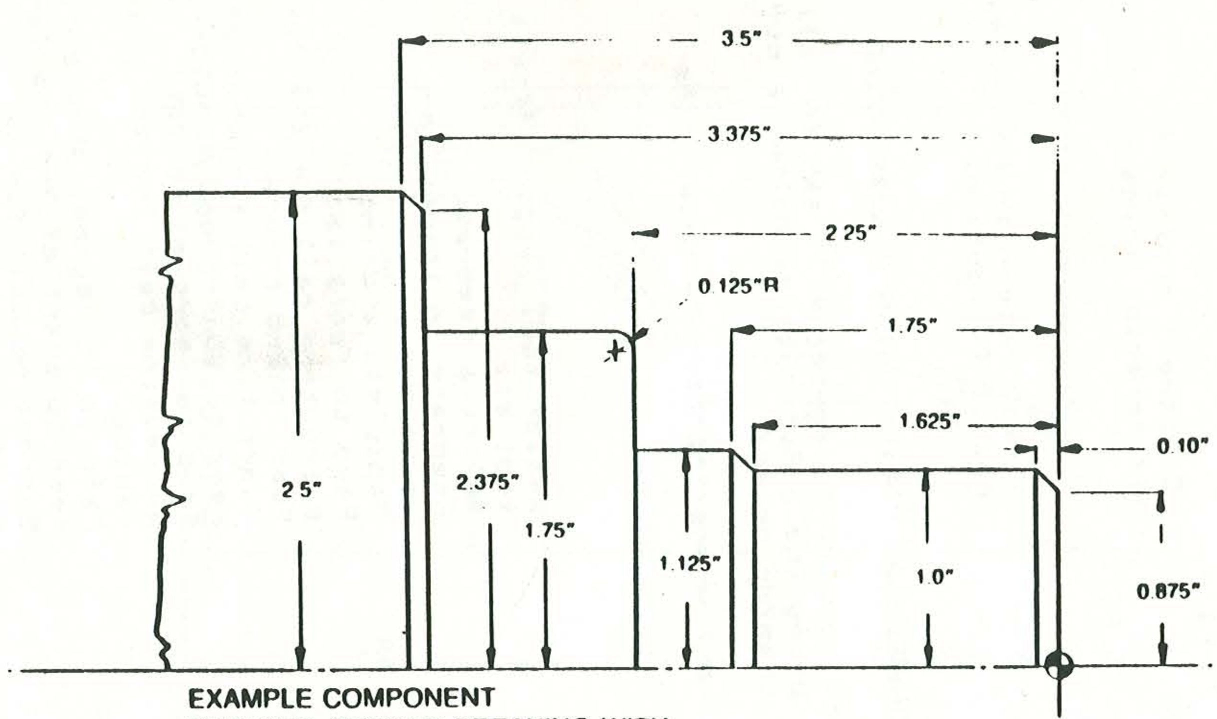
On almost every component, the component drawing will ask for all sharp corners to be removed. Using the 2 Subroutines below, a corner can be removed with either a chamfer or radius.

Figure 28 below shows a component where the corners have to be removed:

Event_#	Command	
11.	TOOL 1	Assign tool length offsets to tool #1.
12.	V21 .005	Select a feedrate.
13.	G95	Feedrate in inches per revolution.
14.	X .875 Z .1 RA	Rapid start point.
15.	Z 0 FA	Feed to front face.
16.	CALL 1	Subroutine call for chamfer.
17.	Z -1.625 FA	Feed to 2nd chamfer.
18.	CALL 1	Subroutine call for chamfer.
19.	Z -2.25 FA	Feed to 90 degrees shoulder.
20.	X 1.625	Feed to start of radius.
21.	CALL 2	Subroutine call for the radius.
22.	Z -3.375 FA	Feed to 90 degrees shoulder.
23.	X 2.375 FA	Feed to start of 3rd chamfer.
24.	CALL 1	Subroutine call for the chamfer.
25.	TOOL 0	Cancel TLO
26.	X 0 Z 0 RA	Rapid to Absolute Machine Zero.
27.	END	End of program.

SUBROUTINE_1: (Chamfer)

28.	SUB 1	Assign number 1 to subroutine.



EXAMPLE COMPONENT
SHOWING CORNER BREAKING WITH
BOTH CHAMFERS & RADII USING A
SUBROUTINE.

Figure 28

29.	X .1 Z -.1 FI	Feed at 45 degrees in incremental to chamfer.
30.	END	End of subroutine.

SUBROUTINE 2 (Radius)

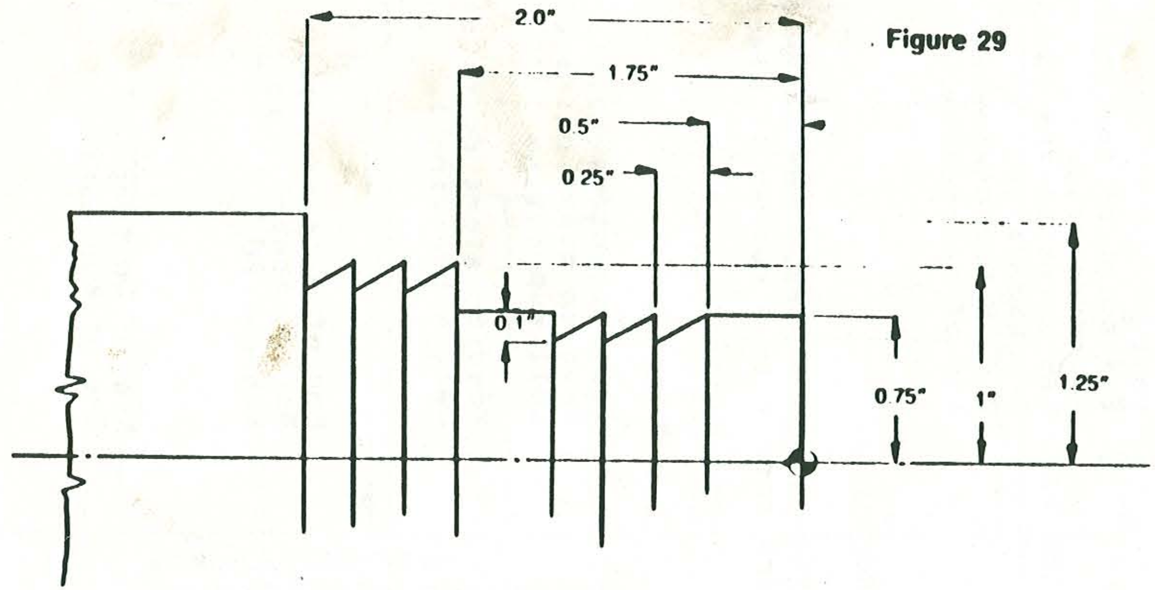
31.	SUB 2	Assign number 2 to subroutine.
32.	ARC CCW	Program direction of the radius.
33.	X 0 Z -.125 FI	End point of the radius incrementally.
34.	X .125 Z -.125 FI	End point of the radius incrementally.
35.	ARC	End of Arc Statement.
36.	END	End of subroutine.

F. COMBINING DO LOOPS AND SUBROUTINES:

A Do Loop can be programmed within a subroutine. For example, this would be done if a repeat feature occurs in a number of places along a shaft (see figure 29).

Event #	Command	
40.	X .75 Z .1 RA	Rapid to start point.
41.	Z -.5 FA	Feed to start point of sub 1.
42.	CALL 1	Call subroutine 1.
43.	Z -1.75 FA	
44.	X 1.0 FA	Feed to start point of Sub 1.
45.	CALL 1	Call subroutine 1.
46.	X 1.25 FA	
47.	TOOL 0	
48.	X 0 Z 0	
49.	END	End of main program.
50.	SUB 1	Assign number 1 to subroutine.
51.	DO 3	
52.	X -.1 Z -.25 FI	
53.	X .1 FI	
54.	END	End of Do Loop
55.	END	End of subroutine

Figure 29



DO LOOP WITHIN A SUBROUTINE

G. PROGRAM FOR EXAMPLE COMPONENT

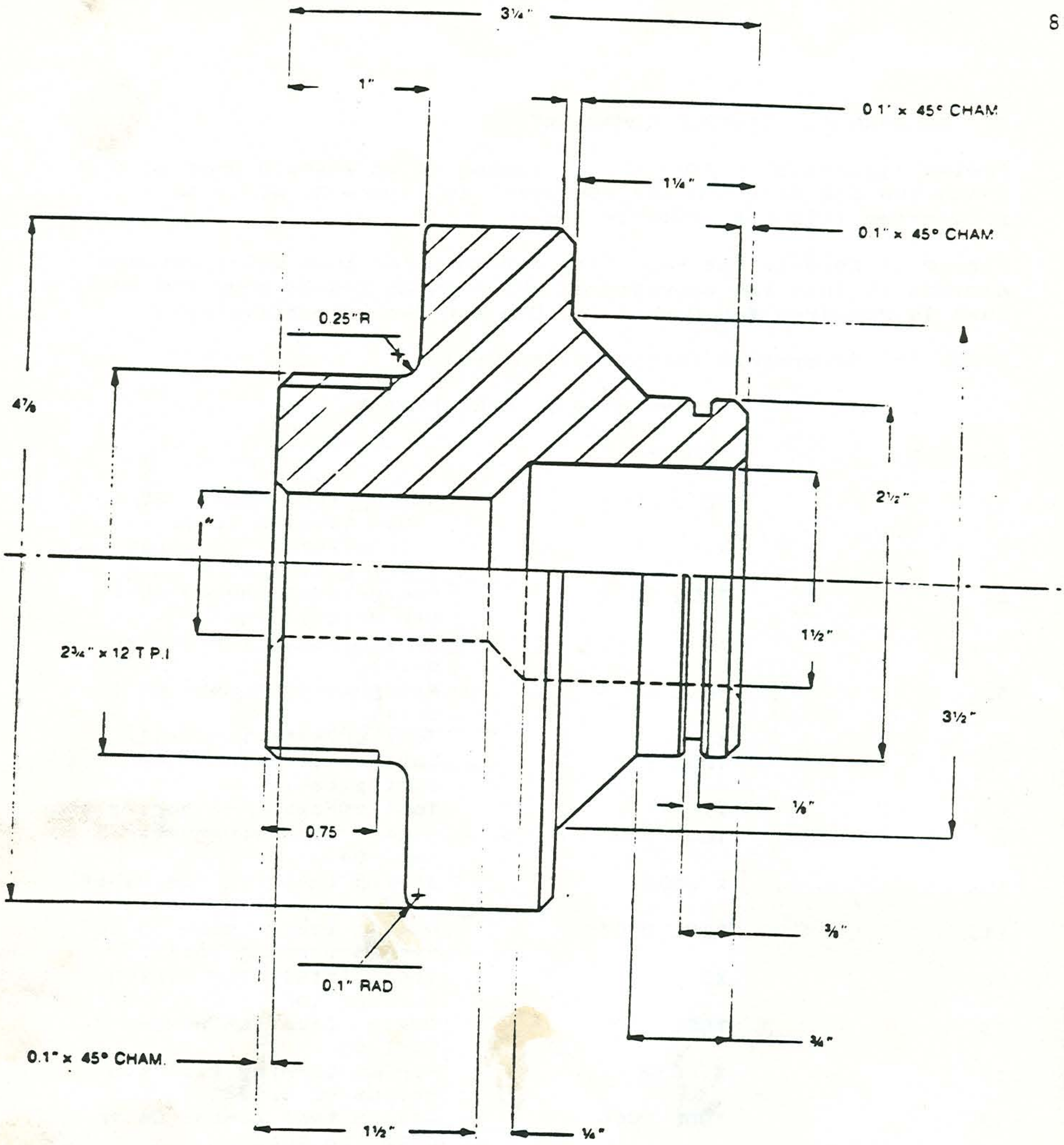
Review figures 30 - 33. The following is an example program that shows the operator how the component in figure 30 would be programmed into the Crusader IIL.

Figrue 31 details the machining sequence for this component and divides it into two operations. Figures 32 and 33 show the tooling that is required for Operations One and Two respectively.

Below is the program for Operation One:

Program 1, Operation 1

Event #	Command	
1.	TOOL 2001	Assign tool number 01 to rough turning tool.
2.	XZ	Tool offsets for rough turning tool.
3.	TOOL 2002	Assign tool number 02 to center drill.
4.	XZ	Tool offsets for center drill.
5.	TOOL 2003	Assign tool number 03 to drill.
6.	XZ	Tool offsets for drill.
7.	TOOL 2004	Assign tool number 04 to the boring tool.
8.	XZ	Tool offsets for boring tool.
9.	TOOL 1504	Assign location number 5 to tool 04.
10.	X .032	Boring tool tip radius of 0.032".
11.	TOOL 2005	Assign tool number 05 to finish turning tool.
12.	XZ	Tool offsets for finish turning tool.
13.	TOOL 1305	Assign location number 3 to tool 05.
14.	X .032	Finish turning tool tip radius of 0.032".
15.	TOOL 2006	Assign tool number 06 to grooving tool.
16.	XZ	Tool offsets for grooving tool.
17.	CALL 1	Subroutine to return tool to Absolute Zero position.
18.	TOOL 1	Activate tool #1 offsets (Rough Turning tool).
19.	V21 .010	Set the feedrate
20.	G95	Feedrate selected in inches per revolution.
21.	X 2.6 Z .10 RA	Rapid to start position (5.2" diameter, 0.01" from finish face).
22.	X -.035 FA	Feed down front face .035" past center line.



GROOVE DIA. 2½". GROOVE CHAMFERS 0.02" x 45°.

STOCK SIZE 5" DIA x 3⅜" LONG

MATERIAL = LEADLOY.

NOT TO SCALE.

Figure 30

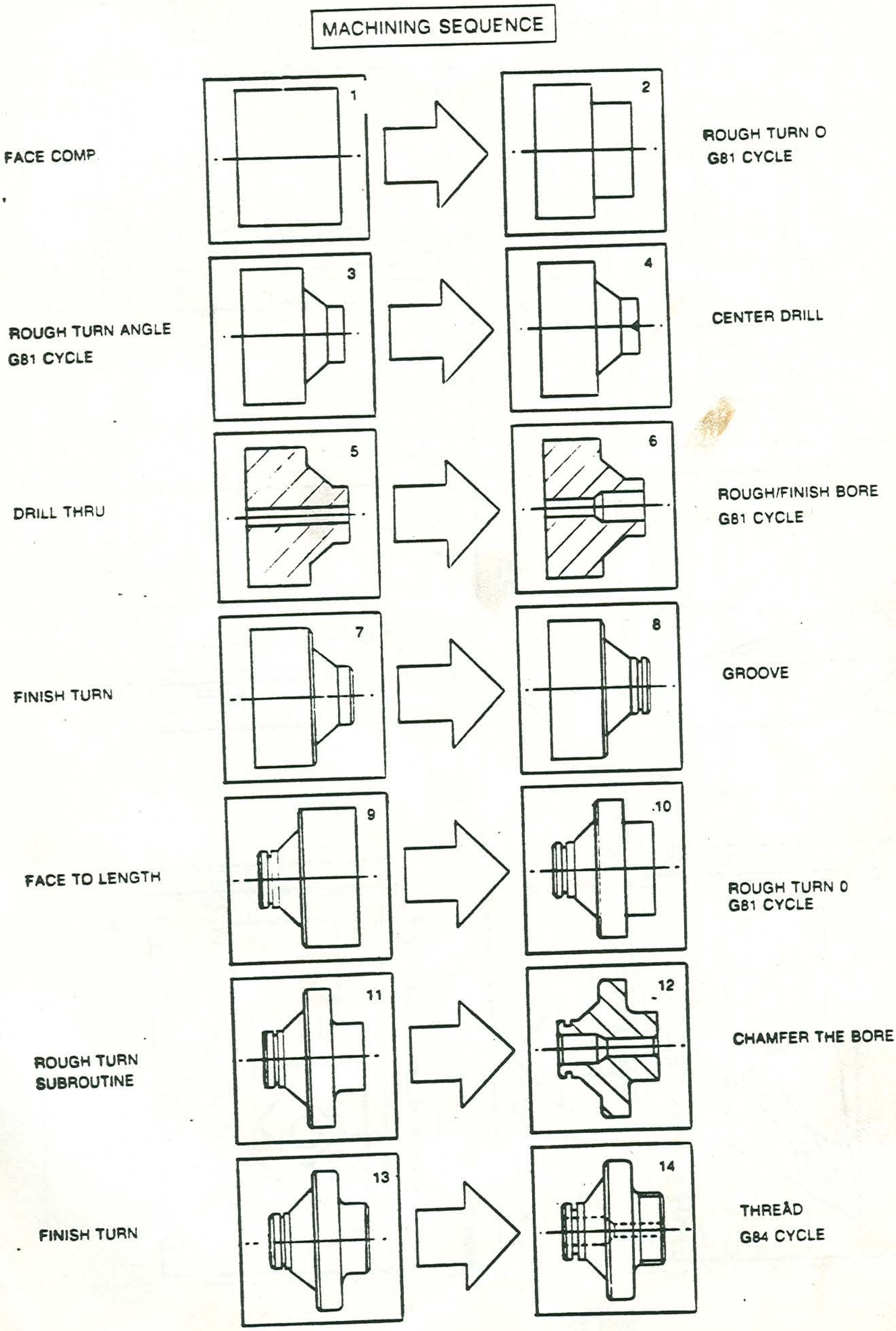

MACHINING SEQUENCE
1
FACE COMP.
2
ROUGH TURN O
G81 CYCLE
3
ROUGH TURN ANGLE
G81 CYCLE
4
CENTER DRILL
5
DRILL THRU
6
ROUGH/FINISH BORE
G81 CYCLE
7
FINISH TURN
8
GROOVE
9
FACE TO LENGTH
10
ROUGH TURN 0
G81 CYCLE
11
ROUGH TURN
SUBROUTINE
12
CHAMFER THE BORE
13
FINISH TURN
14
THREAD
G84 CYCLE

Figure 31

TOOLING FOR OPERATION ONE

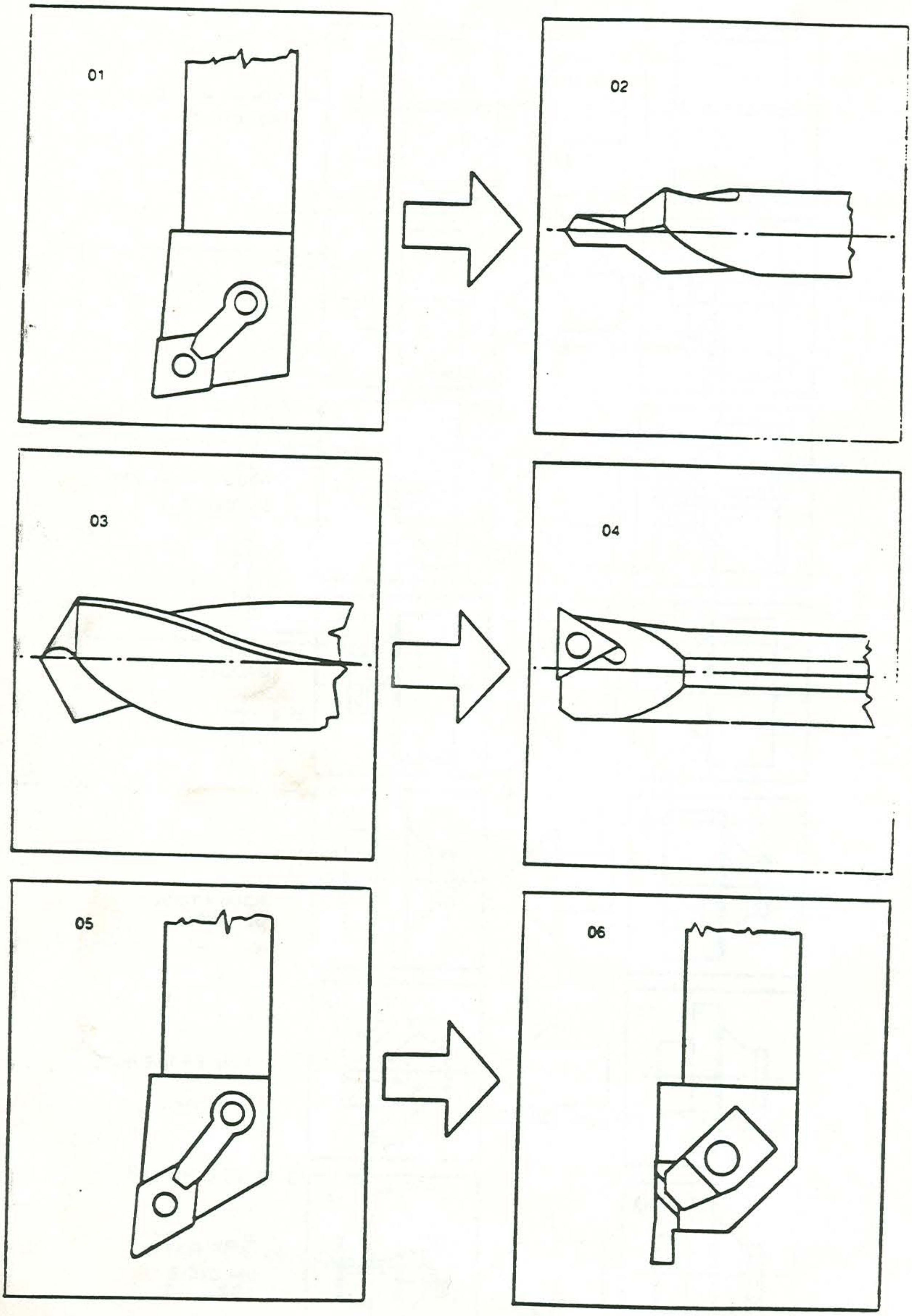


Figure 32

ANILAM ELECTRONICS CORP.

5625 NW. 79th AVE. MIAMI, FL. (305) 592 2727

LATHEMATE PLANNING SHEET.

COMPONENT No.	0001		
DESCRIPTION.	DEMONSTRATION PART.		
MATERIAL.	LEADLOY.		
ENGINEER.	N.WHEWAY.	DATE.	1.1.85

OPERATION. #1	TOOL DESCRIPTION.	TOOL #.	INSERT RADIUS.	LOCAT. No.	FEED RATE.	SPEED.	SPINDLE DIRECT.
FACE, ROUGH TURN DIA &							
ANGLE.	80° DIAMOND L.H.	01	.032	3	.010	1000	REV.
CENTER DRILL.	#3 C' DRILL.	02	—	—	.005	1000	FOR.
DRILL THRU'.	7/8" DIA. DRILL.	03	—	—	.005	1000	FOR.
ROUGH & FINISH BORE.	3/4" DIA. B. BAR.	04	.032	5	.010	1000	REV.
FINISH FACE & TURN.	55° DIAMOND L.H.	05	.032	3	.006	1000	REV.
CUT 1/8" WIDE GROOVE.	L.H. 1/8" GROOVE.	06	—	—	.003	1000	REV.

TOOLING FOR OPERATION TWO

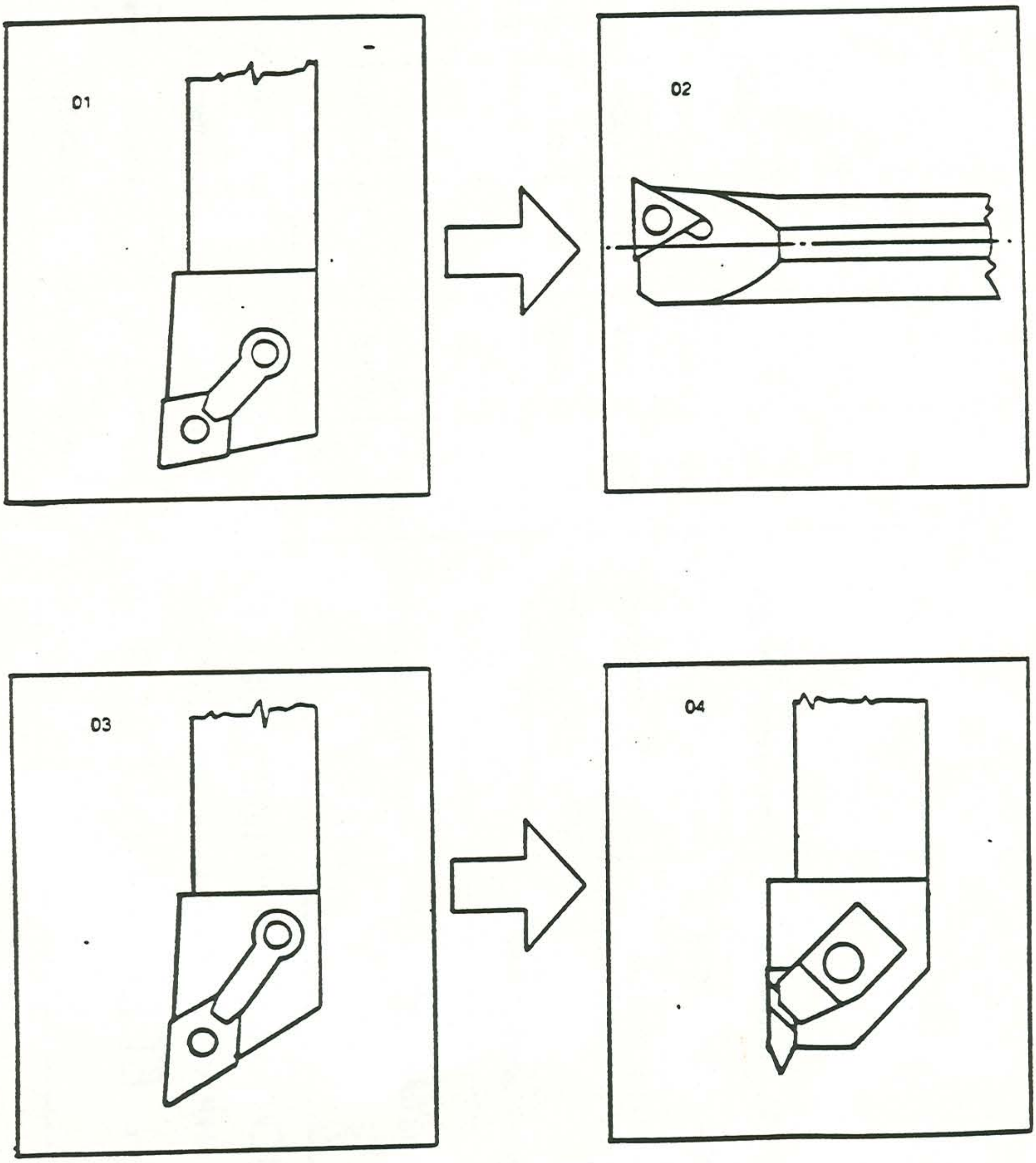


Figure 33

ANILAM ELECTRONICS CORP. 5625 NW. 79th AVE. MIAMI, FL. (305) 592 2727 LATHEMATE PLANNING SHEET.			
COMPONENT No.	0001		
DESCRIPTION.	DEMONSTRATION PART.		
MATERIAL.	LEADLOY.		
ENGINEER.	N.WHEWAY	DATE.	1.1.85

OPERATION. #2	TOOL DESCRIPTION.	TOOL #.	INSERT RADIUS.	LOCAT. No.	FEED RATE.	SPEED.	SPINDLE DIRECT.
FACE, ROUGH TURN DIA &							
RADIUS.	80° DIAMOND L.H.	01	.032	3	.010	1000	REV.
CHAMFER THE BORE.	3/4" DIA. B. BAR.	02	.032	5	.006	1000	REV.
FINISH FACE & TURN.	55° DIAMOND L.H.	03	.032	3	.006	1000	REV.
SINGLE POINT THREAD.	R.H. THREAD TOOL.	04	—	—	.0833	1000	FOR.

23.	X 2.6 Z .01 RA	Rapid to start position for canned cycle (G81).
24.	V50 -.75	
25.	V51 -1.35	
26.	V52 -.125	G81 canned cycle variables.
27.	V53 -1.35	
28.	V54 0	
29.	G81	Activation of canned cycles.
30.	X 1.85 RA	Rapid to start position for canned cycle (G81).
31.	V50 -.5	
32.	V51 -1.0	
33.	V52 -.125	
34.	V53 -.85	Canned cycle variables.
35.	V54 0	
36.	G81	Activation of canned cycle.
37.	CALL 1	Return tool to Absolute Zero position using subroutine 1.
38.	TOOL 2	Activate tool #2 offsets (center drill).
39.	V21 .005	Set feedrate.
40.	G95	Feedrate selected in inches per revolution.
41.	X 0 Z .1 RA	Rapid to center line and 0.1" from front face.
42.	Z -.35 FA	Feed in Z Axis to center drill component.
43.	Z .1 RA	Rapid out in Z Axis.
44.	CALL 1	Return tool to Absolute Zero position using subroutine 1.
45.	TOOL 3	Activate tool #3 offsets (drill).
46.	V21 .005	Set feedrate.
47.	G95	Feedrate selected in inches per revolution.
48.	X 0 Z .1 RA	Rapid to center line and 0.1" from front face.
49.	V50 -3.5	G83 Canned Cycle variables.
50.	V51 -1.0	
51.	G83	Activation of canned cycle.
52.	CALL 1	Return tool to Absolute Zero position using subroutine 1.
53.	TOOL 4	Activate tool #4 offsets (boring tool).
54.	V21 .01	Set feedrate.
55.	G95	Feedrate selected in IPR.
56.	X .3844 Z .1 RA	Rapid to start point for canned cycle.
57.	V50 .25	
58.	V51 -1.85	
59.	V52 .10	G81 canned cycle variables.
60.	V53 -1.6	
61.	V54 0	
62.	G81	Activation of canned cycle.
63.	V21 .006	Set feedrate.
64.	G95	Feedrate selected in IPR.
65.	X .9 RA	Rapid to start position for finishing cut of the bore.

66.	G41	Activate Tool Nose Radius Compensation.
67.	X .85 Z 0 FA	Feed to cut 45o chamfer.
68.	X .75 Z -.1 FA	Feed to cut 0.1 " chamfer.
69.	X -1.5 FA	Feed down the bore.
70.	Z .5 Z -1.75 FA	Feed down the angle of the bore.
71.	X -3.3 FA	Feed through the bore.
72.	G40	Cancel tool nose compensation.
73.	X .45	Ramp off bore.
74.	Z .1	Rapid out of the bore.
75.	CALL 1	Return tool to Absolute Zero position using subroutine 1.
76.	TOOL 5	Activate tool #5 offsets (finish turning tool).
77.	V21 .006	Set feedrate.
78.	G95	Feedrate seleted in IPR.
79.	X .7 Z .1 RA	Rapid to start position for finishing cut.
80.	G42	Activate Tool Nose Compensation.
81.	X .75 Z 0 FA	Feed on to front face ramp on.
82.	X 1.15 FA	Feed up the face to start of the chamfer.
83.	X .1 Z -.1 FI	Cut the .1" x 45o chamfer in incremental mode.
84.	Z -.75 FA	Feed across the 2.5" diameter.
85.	X 1.75 Z 1.25 FA	Finish cut the 45o angle.
86.	X 2.3375 FA	Feed up the face to start of the chamfer.
87.	X .1 Z -.1 FI	Cut the .1" x 45o chamfer in incremental mode.
88.	Z -2.125 FA	Feed across the 4.875" diameter.
89.	G40	Cancel tool nose compensation.
90.	X 2.6 FA	Move off the component.
91.	CALL 1	Return tool to Absolute Zero position using subroutine 1.
92.	TOOL 6	Activate tool #6 offsets (grooving tool).
93.	V21 .003	Set feedrate.
94.	G95	Feedrate selected in IPR.
95.	X 1.35 Z -.374 RA	Rapid to start position for grooving tool.
96.	X 1.15 FA	Feed to groove diameter.
97.	DWELL 2.0	Dwell to clean up bottom of the groove (2 seconds).
98.	X 1.25 RA	Rapid up to 2.5" diameter.
99.	Z -.02 FI	Feed to position for chamfering left side of groove.
100.	X -.02 Z .02 FI	Cut chamfer .02 x 45o.
101.	X 1.25 RA	Rapid up to 2.5 diameter.

102.	Z .02 FI	Feed to position for chamfering right side of groove.
103.	X -.02 Z -.02 FI	Cut chamfer .02" x 45o
104.	X 1.35 RA	Move out in X axis.
105.	CALL 1	Return tool to Absolute Zero position using subroutine 1.
106.	END	End of Operation #1.

(SUBROUTINE #1)

110.	SUB 1	Assign number 1 to subroutine.
111.	TOOL 0	Cancel any active tool length offset.
112.	X 0 Z 0	Rapid to Absolute Zero position.
113.	END	End of subroutine.

H. PROGRAM FOR EXAMPLE COMPONENT

Again, review figures 30, 31 and 32. These figures will detail the dimensions, machining sequences and required tooling for the following program example:

Program 2, Operation 2

Event #	Command	
1.	TOOL 2001	Assign tool number 01 to rough turning tool.
2.	X Z	Tool offsets for rough turning tool.
3.	TOOL 2002	Assign tool number 02 to the boring tool.
4.	X Z	Tool offsets for boring tool
5.	TOOL 1502	Assign Location number 5 to tool 02.
6.	X .032	Finish boring tool tip radius of 0.032".
7.	TOOL 2003	Assign tool number 03 to the finished turning tool.
8.	X Z	Tool offsets for finish turning tool.
9.	TOOL 1303	Assign location number 3 to tool 03.
10.	X .032	Finish turning tool tip radius of 0.032".
11.	TOOL 2004	Assign tool number 04 to the threading tool.
12.	X Z	Tool offsets for threading tool.
13.	CALL 1	Subroutine to return tool to Absolute Zero Posn.
14.	TOOL 1	Activate tool #1 offsets (rough turning tool).
15.	V21 .010	Set the feedrate.
16.	G95	Feedrate selected in inches per revolution.
17.	X 2.6 Z .01 RA	Rapid to start Posn. ready to face Comp. to size.
18.	X -.035 FA	Feed down front face 0.035" past center line.
19.	X 2.6 Z .1 RA	Rapid to start Posn. for Canned Cycle (G81).
20.	V50 -.8125	
21.	V51 -1.1	G81 Canned Cycle variables
22.	V52 -.1	
23.	V53 -1.1	
24.	V54 0	
25.	G81	Activation of Canned Cycle
26.	X 1.5 RA	Rapid to start Posn. to rough the 0.25" radius.
27.	CALL 2	1st roughing pass using a Subroutine.
28.	Z .1 RA	Rapid back to 0.1" in the Z Axis.

Event #	Command	
29.	X 1.375 RA	Posn. in the X Axis for 2nd roughing pass.
30.	CALL 2	Second roughing pass using a Subroutine for the radius.
31.	CALL 1	Return tool to Absolute Zero Posn. using Subroutine 1.
32.	TOOL 2	Activate tool #2 offsets (boring tool).
33.	V21 .006	Set the feedrate.
34.	G95.	Feedrate selected in inches per revolution.
35.	X .55 Z .1 RA	Rapid to Posn. ready to chamfer the bore.
36.	G41	Turn on tool nose radius compensation.
37.	X .60 Z 0 FA	Ramp on to the front face (beginning of the chamfer).
38.	X -.15 Z -.15 FI	Feed at 45o to chamfer in incremental.
39.	G40	Cancel tool nose radius compensation
40.	X .40	Ramp off the bore.
41.	Z .1 RA	Rapid clear of the bore.
42.	CALL 1	Return tool to Absolute Zero Posn. using Subroutine 1.
43.	TOOL 3	Activate tool #3 offsets (finish turning tool).
44.	V21 .006	Set the feedrate.
45.	G95	Feedrate selected in inches per revolution.
46.	X .45 Z .1 RA	Rapid to start Posn. for finishing cut.
47.	G42	Turn on tool nose radius compensation.
48.	X .50 Z 0 FA	Ramp on to front face.
49.	X 1.275 FA	Feed up the face to the start of the chamfer.
50.	X 1.375 Z -.1 FA	Cut the 0.1" X 45o chamfer.
51.	CALL 2	Use Sub 2 to finish the 2 3/4" Dia. & the 0.25" radius.
52.	X 2.3375 FA	Feed up the face to the start of the 0.1" radius.
53.	ARC CCW	Select the ARC direction of counter clockwise.
54.	X 2.3375 Z -1.1 FA	Define the center points of the radius.
55.	X 2.4375 Z -1.1 FA	Define the end points of the radius.
56.	ARC	Complete the ARC statement for the radius.

57.	X 2.5 FA	Move off the diameter.
58.	G40	Cancel tool nose radius compensation.
59.	X 2.55 FA	Ramp off for tool compensation.
60.	CALL 1	Return tool to Absolute Zero Posn. using Subroutine 1.
61.	TOOL 4	Activate tool #4 offsets (threading tool).
62.	V21 .05	Set the feedrate for positioning when threading.
63.	G95	Select feedrate in inches per revolution.
64.	X 1.475 Z .14 RA	Rapid to Posn. for threading Canned Cycle.
65.	V42 12.0	
66.	V50 -.89	
67.	V51 -.048	G84 Canned Cycle variables.
68.	V52 -.014	
69.	V53 0	
70.	V54 0	
71.	V55 1.0	
72.	G84	Activation of Canned Cycle.
73.	CALL 1	Return tool to Absolute Zero Posn. using Subroutine 1.
74.	END	End of program & operation #2.

SUBROUTINE 1

100.	SUB 1	Assign number 1 to Subroutine.
101.	TOOL 0	Cancel any active tool length offset.
102.	X 0 Z 0 RA	Rapid to Absolute Zero Posn.
103.	END	End of Subroutine.

SUBROUTINE 2

200.	SUB 2	Assign number 2 to Subroutine.
201.	Z -.75 FA	Turn Dia. to beginning of the radius.
202.	ARC CW	Assign direction of radius.
203.	X 1.625 Z -.75 FA	Define the center points of the radius.
[illegible]4.	X 1.625 Z -1.0 FA	Define the end points of the radius.
	ARC	Complete the ARC statement.
	END	End of Subroutine.

SECTION 19:

O.E.M. CONTROL FUNCTIONS

NOTE:

All machine functions that are specific to particular O.E.M. requirements may be described and explained within this section by the O.E.M. company. Anilam requests that all programming explanations be as clear and concise as possible.

ANILAM ELECTRONICS CORP.

5625 NW. 79th AVE. MIAMI, FL. (305) 592 2727

LATHEMATE PLANNING SHEET.

COMPONENT No:			
DESCRIPTION.			
MATERIAL.			
ENGINEER.		DATE.	

OPERATION.	TOOL DESCRIPTION.	TOOL #.	INSERT RADIUS.	LOCAT. No.	FEED RATE.	SPEED.	SPINDLE DIRECT.

ANILAM ELECTRONICS CORP.

5625 NW. 79th AVE. MIAMI, FL. (305) 592 2727

LATHEMATE PLANNING SHEET.

COMPONENT No:	
DESCRIPTION.	
MATERIAL.	
ENGINEER.	DATE.

OPERATION.	TOOL DESCRIPTION.	TOOL#.	INSERT RADIUS.	LOCAT. No.	FEED RATE.	SPEED.	SPINDLE DIRECT.