COMPUTER NUMERICAL CONTROL OF MACHINE TOOLS

Laboratory for Manufacturing Systems and Automation
Department of Mechanical Engineering and Aeronautics
University of Patras, Greece

Dr. Dimitris Mourtzis

Associate professor

Patras, 2017

Chapter 10:
Linear and Circular Interpolation

Table of Contents

Objectives of Chapter 10

- Learn about **linear** and **circular interpolation**
- Write programs using **linear interpolation** to cut simple angles
- Write simple programs using **circular interpolation** to mill arcs

Linear Interpolation

Linear interpolation:

- Means **cutting a straight line between two points**

- Sometimes this is referred to as a **feedrate move** since modern CNC controls **automatically** perform **linear interpolation** on any move made while in feed-rate mode

- Prior to modern CNC controls special codes were necessary to turn on the built-in linear interpolation system

- Some CNC controls also will **interpolate rapid moves** - while others simply move the axes drive motors at maximum speed in rapid traverse mode

Linear Interpolation

- The axis the spindle moves with **basic orthogonal movements** from the beginning to the end of the path
- The programmed rectilinear path is **divided into a large number of short length straight lines**
- **The more lines, the better approximation** is made of the actual path
- **The more lines, the more computational power required** - No longer used for non straight segments

Linear Interpolation

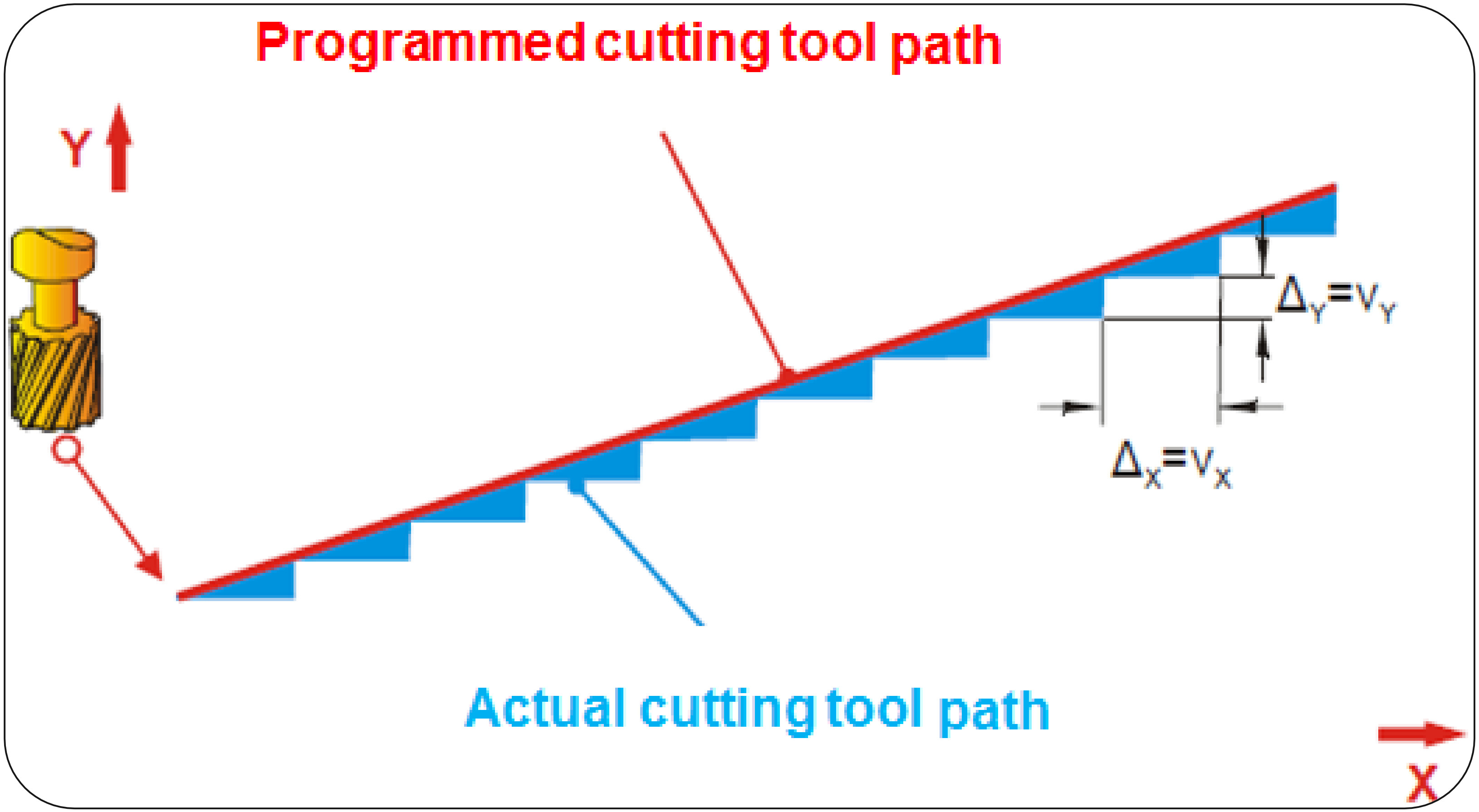


Figure 1:Linear Interpolation

(Source:Σύγχρονες μέθοδοι κατεργασίας υλικών και προγραμματισμός
με Ηλεκτρονικό Υπολογιστή (Η/Υ) ,Δ. Μούρτζης κ.α)

Linear interpolation:

- Machines capable of linear interpolation have a **continuous-path control system -** meaning that the drive motors on the various axes can operate at varying rates of speed
- Virtually all modern CNC controls utilize continuous path controls
- When cutting an angle the MCU calculates the angle based on the programmed coordinates
- Since the MCU knows the current spindle location, it can calculate the **difference in the X coordinate** between the current position and the programmed location

The change in the Y coordinate divided by the change in the X coordinate yields the slope of the cutter centerline path

Linear Interpolation

Calculating Cutter Offsets:

- The cutter has already been positioned at location #1
- A .500-inch diameter end mill is being used
- Before the angle can be cut it is necessary to first position the spindle at location #2
- The Y-axis coordinate for location #2 - as dimensioned on the part - is not the same point as the edge of the angle
- To determine this Y-axis cutter offset it will be necessary to determine the amount that must be added to the dimension on the part prior to place the spindle at location #2
- It will be necessary to calculate an amount to be subtracted from the point on the part designated as "P" to arrive at the X-axis coordinate for location #3

Calculating Cutter Offsets:

- Figure 2 shows a part on which an angle is to be milled

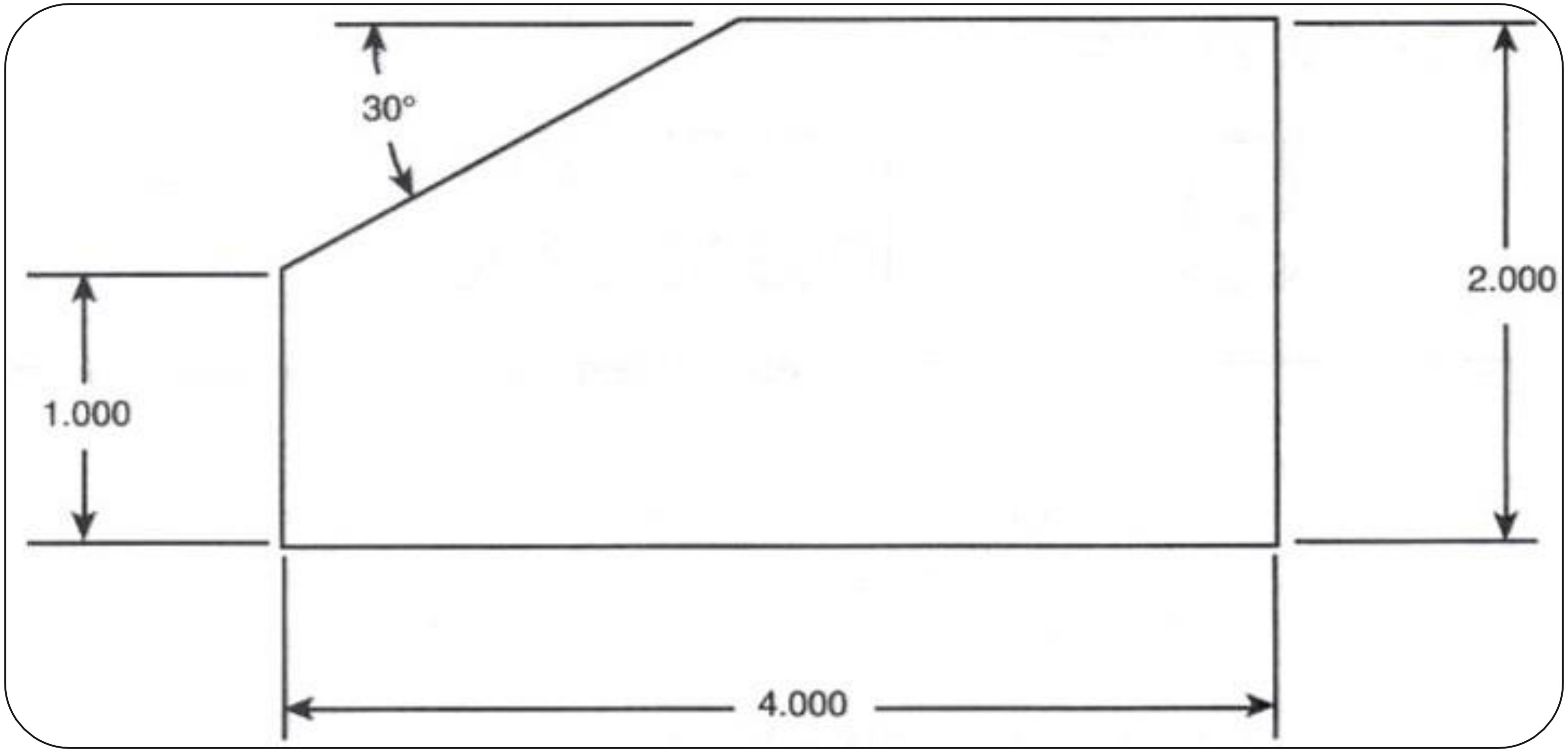


Figure 2: Part drawing

Seams W., "Computer Numerical Control, Concepts & Programming"

Calculating Cutter Offsets:

- Figure 3 shows the cutter path and geometric relationships between the cutter and the part

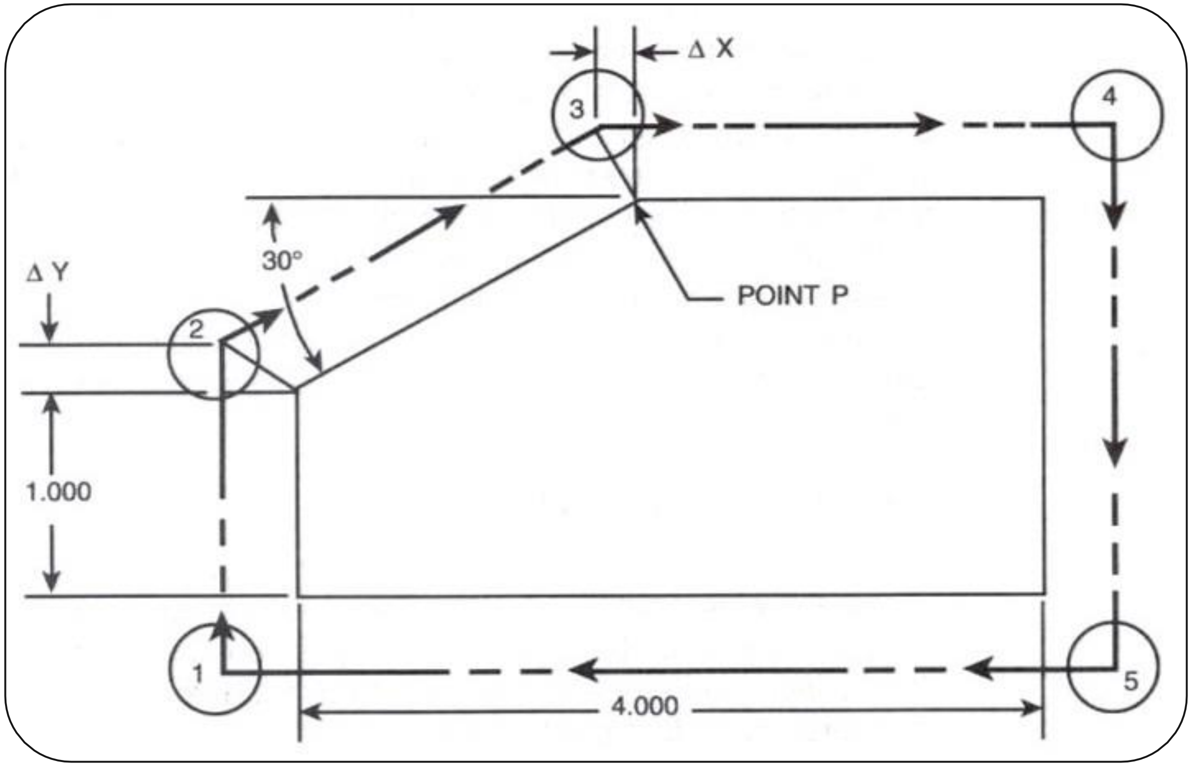


Figure 3:Cutter path for part in Figure 9-1

Seams W., "Computer Numerical Control, Concepts & Programming"

Calculating Cutter Offsets:

- Figure 4 represents an enlarged view of locations #2 and #3, illustrating the triangles involved in determining the offsets

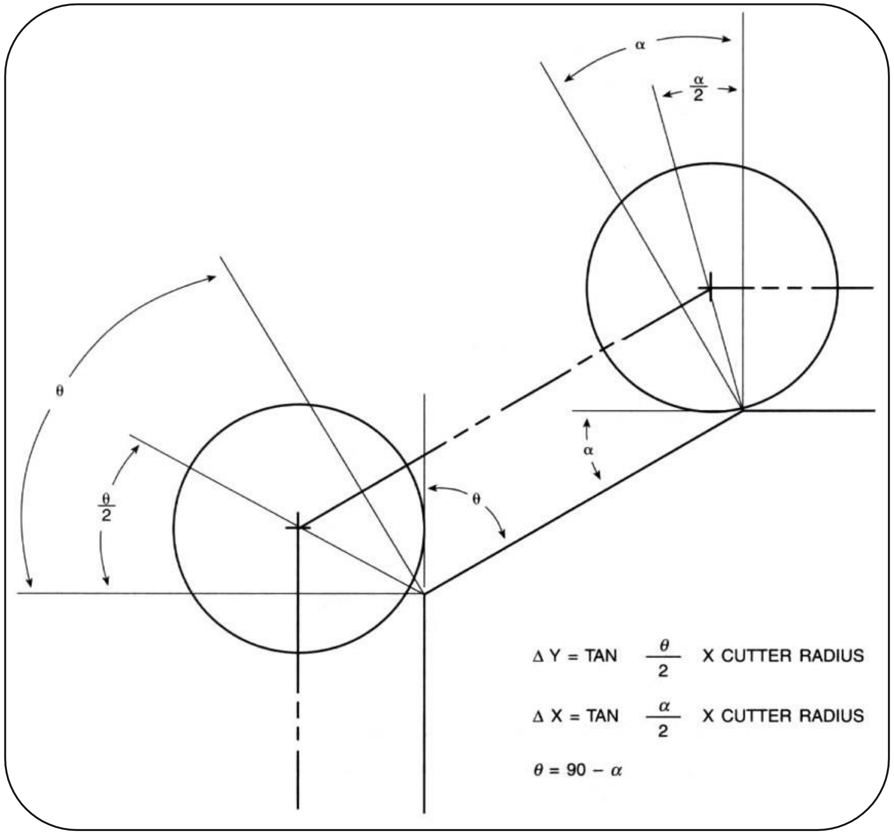


Figure 4:Determining cutter offset

Seams W., "Computer Numerical Control, Concepts & Programming"

Linear Interpolation

Calculating ΔY

ΔY = CR[TAN(θ/2)] where CR = cutter radius
ΔY = 0.25[TAN(30)]
ΔY = 0.25(0.5774)
ΔY = 0.144

Calculating ΔX

ΔX = CR[TAN(α/2)]
ΔX = 0.25[TAN(15)]
ΔX = 0.25(0.26794)
ΔX = 0.067

Before using this information to determine the X-axis coordinate calculate the coordinate location of point **"P" along the X axis:**

TAN(30) = 1.000/b
0.5774 = 1000/b
0.5774 X b = 1.000
b = 1.000/0.5774
b = 1.732

N…G01 Y1.144	**(move from #1 to #2)**
N…X1.665 Y2.25	**(move from #2 to #3)**
N…X4.25	**(move from #3 to #4)**

1. Subtracting .067 (the AX offset) from 1.732 produces the X-axis coordinate for the cutter of 1.665
2. The AY offset, which is .144 can be added to the 1.000 Y-axis dimension on the part to arrive at a Y-axis coordinate of 1.144

Linear Interpolation

Interpolating in Word Address Format

- Milling an angle with word address is not complicated
- The interpolator is **automatically turned on** when **feedrate mode** is commanded
- Milling becomes a matter of specifying the coordinates along with the **G01 feedrate mode code**
- In CNC shops **G01** is called: the **feedrate mode code** *or* the **linear interpolation code**
- With modern CNC controls the terms mean the same thing
- Any feed-rate move is considered an interpolated angled line move
- A move along the **X axis** only would cut an angled line of **0 degrees**
- A move along the **Y axis** would cut an angle of **90 degrees**

Linear Interpolation

Interpolating in Word Address Format (G01)

To illustrate linear interpolation, the following program lines would move the cutter from location #1 to locations #2, #3, and #4

N...G01 Y1.144	(move from #1 to #2)
N...X1.665 Y2.25	(move from #2 to #3)
N...X4.25	(move from #3 to #4)

- A **G01** is given to turn on linear interpolation (feedrate mode)
- The coordinate **Y1.144** moves the cutter to **location #2**
- The **X1.665**, **Y2.25** move the cutter from **location #2** to **#3**
- The **X4.25** coordinates then move the **cutter from #3** to **#4**
- Note: **G01** is a modal code: The machine remains in **linear interpolation mode (feedrate mode)** for all the coordinates specified. The **G01** is active until cancelled by another motion mode G code (**G00**, **G02**, **G03**, or **G04**)

Notes

- Linear interpolation is not difficult
- Aside from **calculating the cutter offsets** necessary to position the spindle - it is the same as straight line milling
- The real difference is that an **X and a Y coordinate are specified for the ending point of the angle** since there is a change in position of both axes
- Other cutter situations will present themselves in CNC part programming such as **arcs tangent to an angle or arcs tangent to other arcs**

Seams W., "Computer Numerical Control, Concepts & Programming"

Notes

Computer Aided Manufacturing (CAM) programming systems automatically calculate **cutter offsets** with **speed and accuracy no programmer can match**

- For this reason **CAM** systems have become the preferred programming system in many shops

- A good programmer or CNC operator must still know how to calculate cutter offsets in order **to edit programs in the machine control during the first piece setup**

Additional Example

- This part has two angles which intersect each other, a .500-inch cutter will be used.

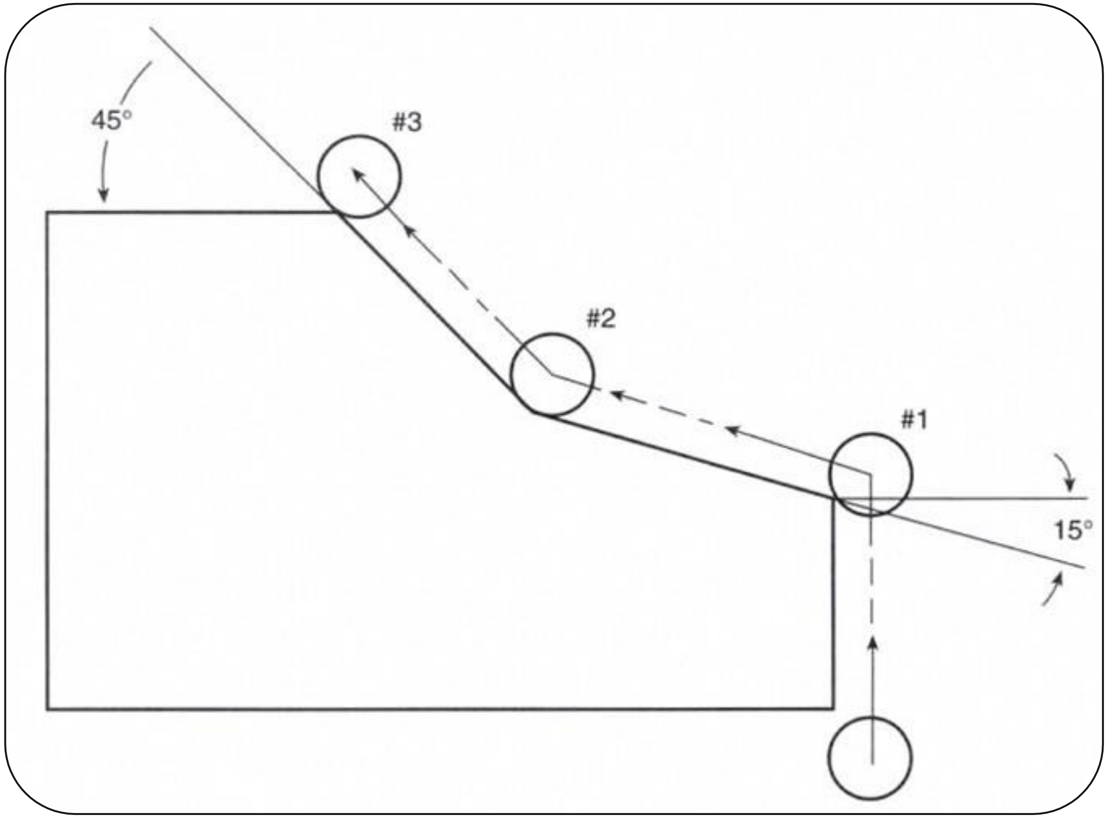


Figure 5: Part drawing with cutter path shown

Seams W., "Computer Numerical Control, Concepts & Programming"

LAB

Linear Interpolation

Offset For Location #1

ΔX = CR

ΔY = CR[TAN(75/2)]

ΔY = 0.25[TAN(37.5)]

ΔY = 0.25 (0.7673)

ΔY = 0.1918

Offset For Location #2

ΔX = CR X { [SIN((45+15)/2)] / [COS((45-15)/2)] }

ΔX = 0.25 X [SIN(30)/COS(15)]

ΔX = 0.1294

ΔY = CR X { [COS((45+15)/2] / [COS((45-15)/2)] }

ΔY = 0.25 X [COS(30)/COS(15)]

ΔY = 0.2241

Offset For Location #3

ΔX = CR X [TAN(45/2)]

ΔX = 0.25 X TAN(22.5)

ΔX = 0.1036

ΔY = CR

Linear Interpolation

- Other cutter situations will present themselves in CNC part programming such as **arcs tangent to an angle** or **arcs tangent to other arcs**

- **CAM** programming systems **automatically calculate cutter offsets** with speed and accuracy no programmer can match

- For this reason **CAM** systems have become the preferred programming system in many shops

- A good programmer or CNC operator must still know ***how to calculate cutter offsets in order to edit programs*** in the machine control during the first piece setup

Circular Interpolation

- In **cutting arcs**, the **M**achine **C**ontrol **U**nit **(MCU)** uses its ability to **generate angles to approximate an arc**
- Since the machine axes do not revolve around a centerpoint in a typical three-axis arrangement, **the cutting of a true arc is not possible**
- **Circular interpolation** is the term used to describe generating a **move consisting of a series of straight-line chord segments** by the **MCU** in two axes to simulate circular motion, as illustrated in Figure 7
- These **chord segments** are very small and practically indistinguishable from a true arc
- Figure 8 shows a part with a radius to be machined

Circular Interpolation

- **Limited to the main plane** of the machined surface
- **Unable participation of the rotary machining axis** of the machine-tool
- **Not used for interpolation in the space** due to requirement of the combined movement of three or more machining axes
- **Ideal** for moving the axes when **the path of the cutting tool in a plane contains circles, half circles or arcs**. In this case only the coordinates of the ends of the arc, the radius and center are required

Circular Interpolation

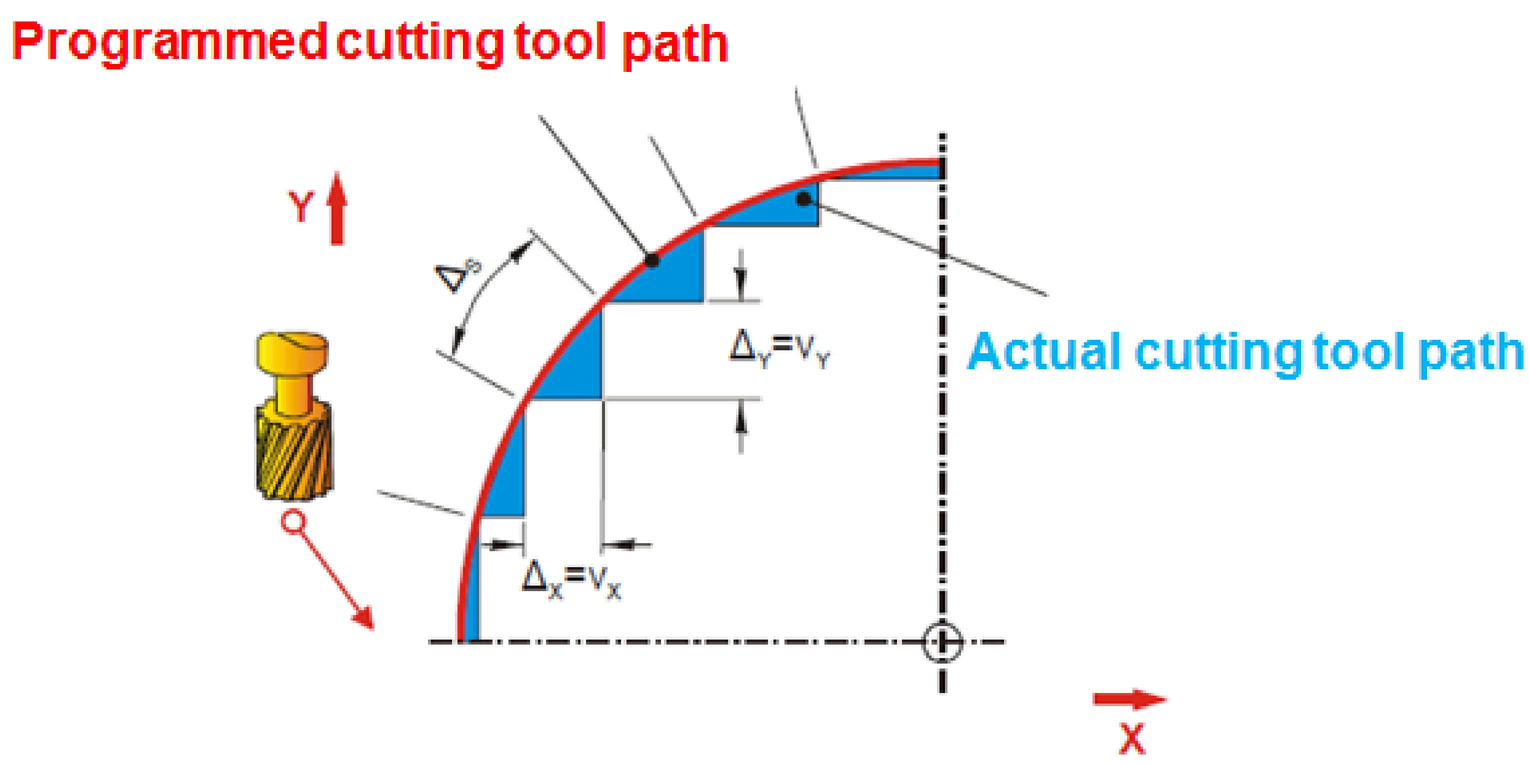
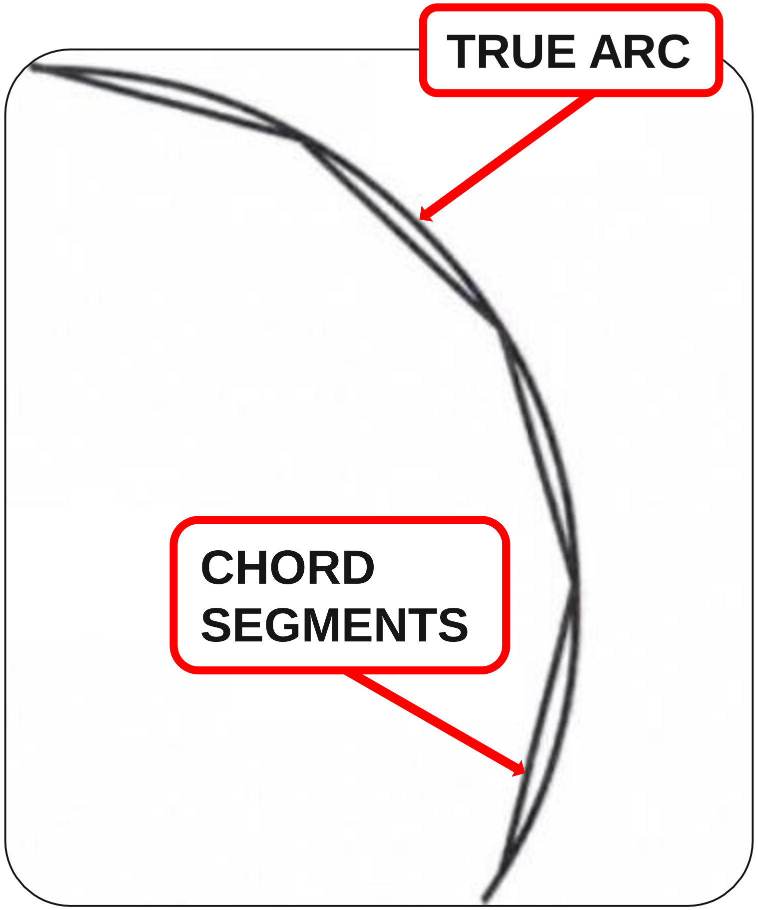


Figure 6 :Circular interpolation
(Source:Σύγχρονες μέθοδοι κατεργασίας υλικών και προγραμματισμός με Ηλεκτρονικό Υπολογιστή (Η/Υ) ,Δ. Μούρτζης κ.α)

Circular Interpolation



- A **move consisting of a series of straight-line chord segments** by the MCU in two axes to simulate circular motion

Figure 7: Circular interpolation chord segments

Seams W., "Computer Numerical Control, Concepts & Programming"

Circular Interpolation

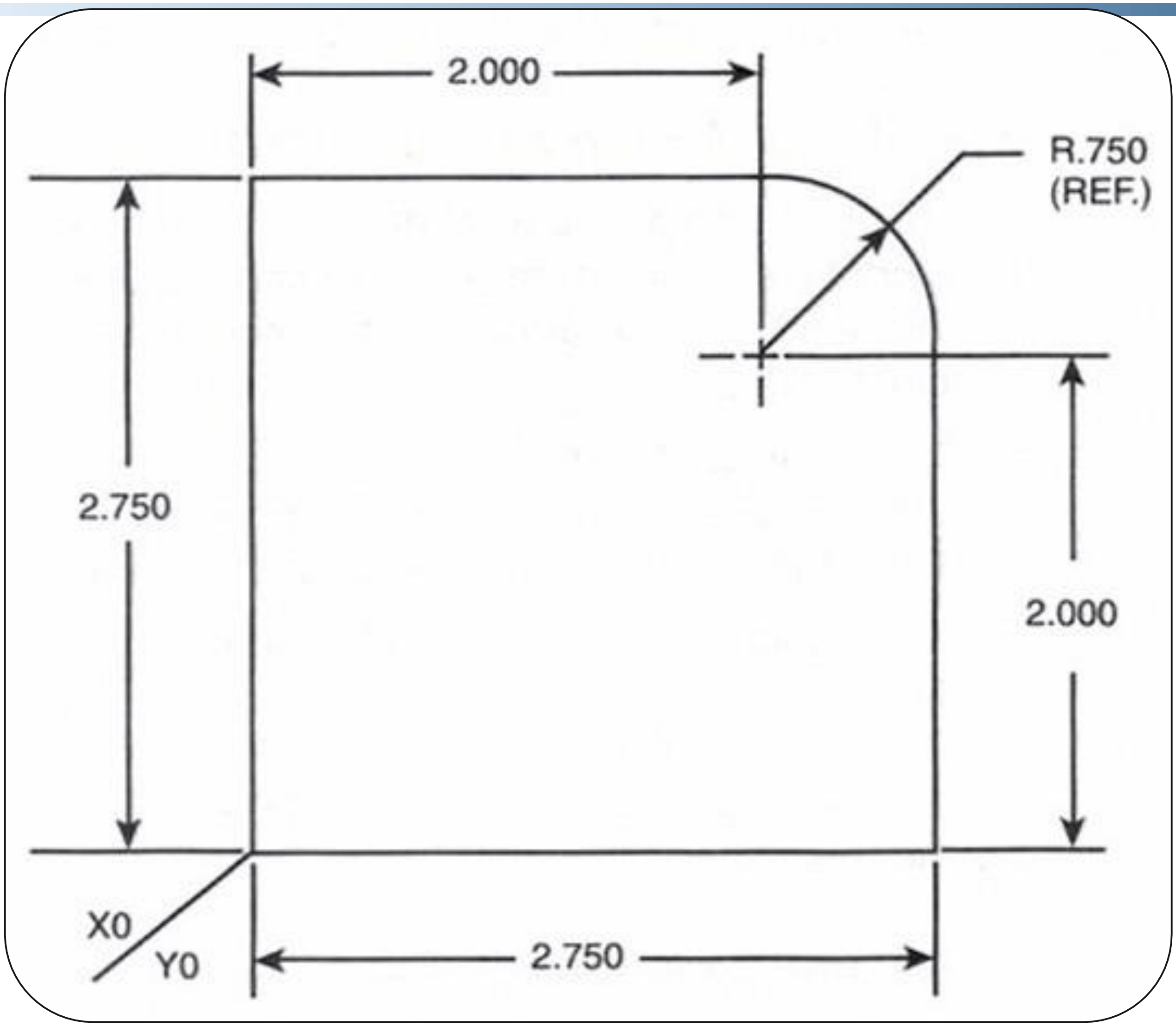
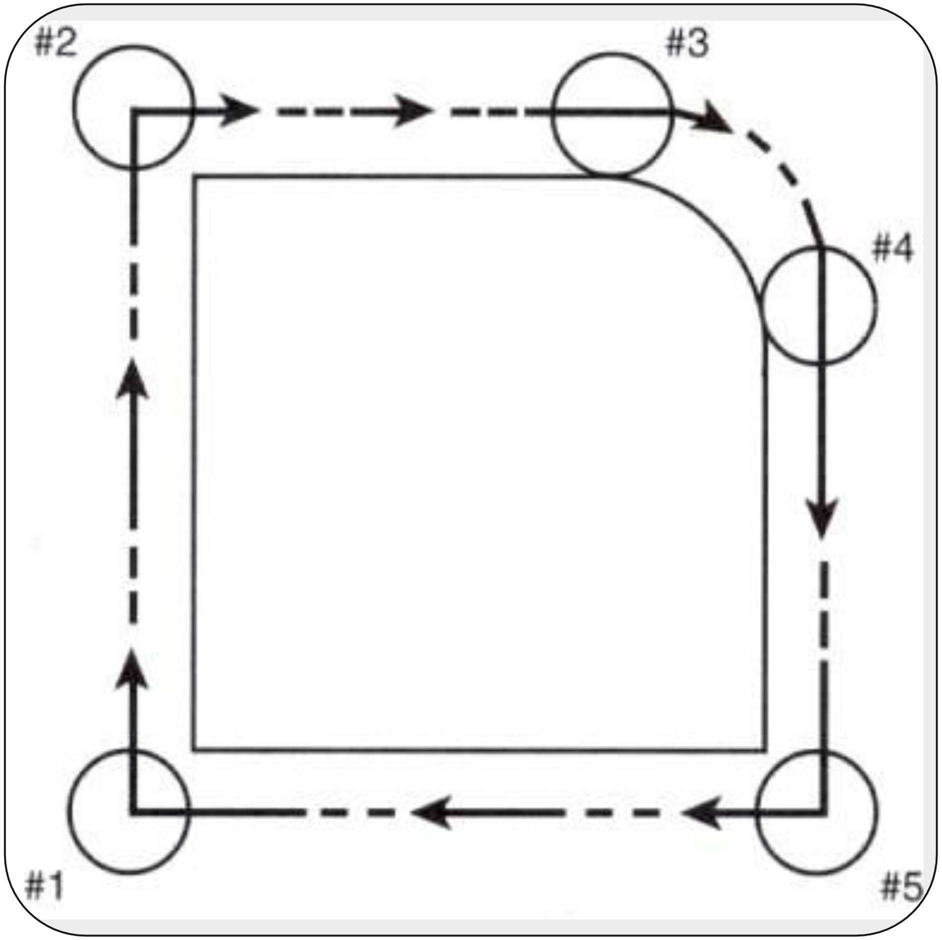


Figure 8: Part with radius to be machined

Seams W., “Computer Numerical Control, Concepts & Programming”

Circular Interpolation



To generate an arc, the MCU needs to know the following information

1. The **axes** to be used in generating the arc
2. The **direction** of interpolation, **clockwise** or **counterclockwise**
3. The **starting X/Y/Z coordinate of the arc**
4. The **ending X/Y/Z coordinate of the arc**
5. The **X/Y/Z coordinates of the arc centerpoint**

Figure 9: Cutter path for part shown in Figure 8. In order to generate the radius, circular interpolation will be used to send the cutter from location #3 to location #4,a .500-inch diameter end mill will be used *(Seams W., "Computer Numerical Control, Concepts & Programming")*

Circular Interpolation

Specifying Axis for Interpolation

- Circular interpolation by definition involves only two axes.
- On FANUC-style controls, a plane designation code is used to select which pair of axis will be used to generate the arc motion.
- There are three G codes used to specify these planes:

G17 – Selects the X/Y plane (X and Y axis)

G18 – Selects the Y/Z plane (Y and Z axis)

G19 – Selects the Z/X plane (Z and X axis)

- These G codes are modal. A **G17**, for example, is cancelled only by a **G18** or **G19**
- The X/Y plane (using the X and Y axis) is the most common orientation for circular interpolation, therefore, **G17** will be used throughout the examples in this text

Circular Interpolation

Specifying Arc Direction

Circular interpolation can be accomplished in one of two directions: **clockwise, or counterclockwise**. There are two G codes used to specify direction:

G02 – Circular interpolation clockwise (CLW)

G03 – Circular interpolation counterclockwise (CCLW)

- **G02**/**G03** codes are modal
- They will cancel an active **G00** (rapid traverse) or **G01** (linear interpolation) codes
- **G02**/**G03** are feedrate mode codes, just as **G01** is.
- The difference lies in the type of interpolation used.
- **G01** generates straight-line interpolation motion. **G02**/**G03** generates arc simulation interpolation motion

Circular Interpolation

Specifying Beginning and Ending Arc Coordinates

- The **MCU** requires the spindle be positioned at the start of the arc when the **G02**/**G03** command is given
- The current spindle position is the beginning arc coordinates. The axis coordinates given on the **G02**/**G03** line are the spindle ending points of the arc motion

Specifying Arc Center points

- There are two methods used to specify arc centerpoints: **arc vector method** and **radius method** *(see* Figure 10)
- The arc vector method involves specifying the coordinates of the arc centerpoint as X/Y values
- In the radius method, the arc centerpoint is calculated internally by the **MCU**. The programmer simply specifies the radius value required

Circular Interpolation

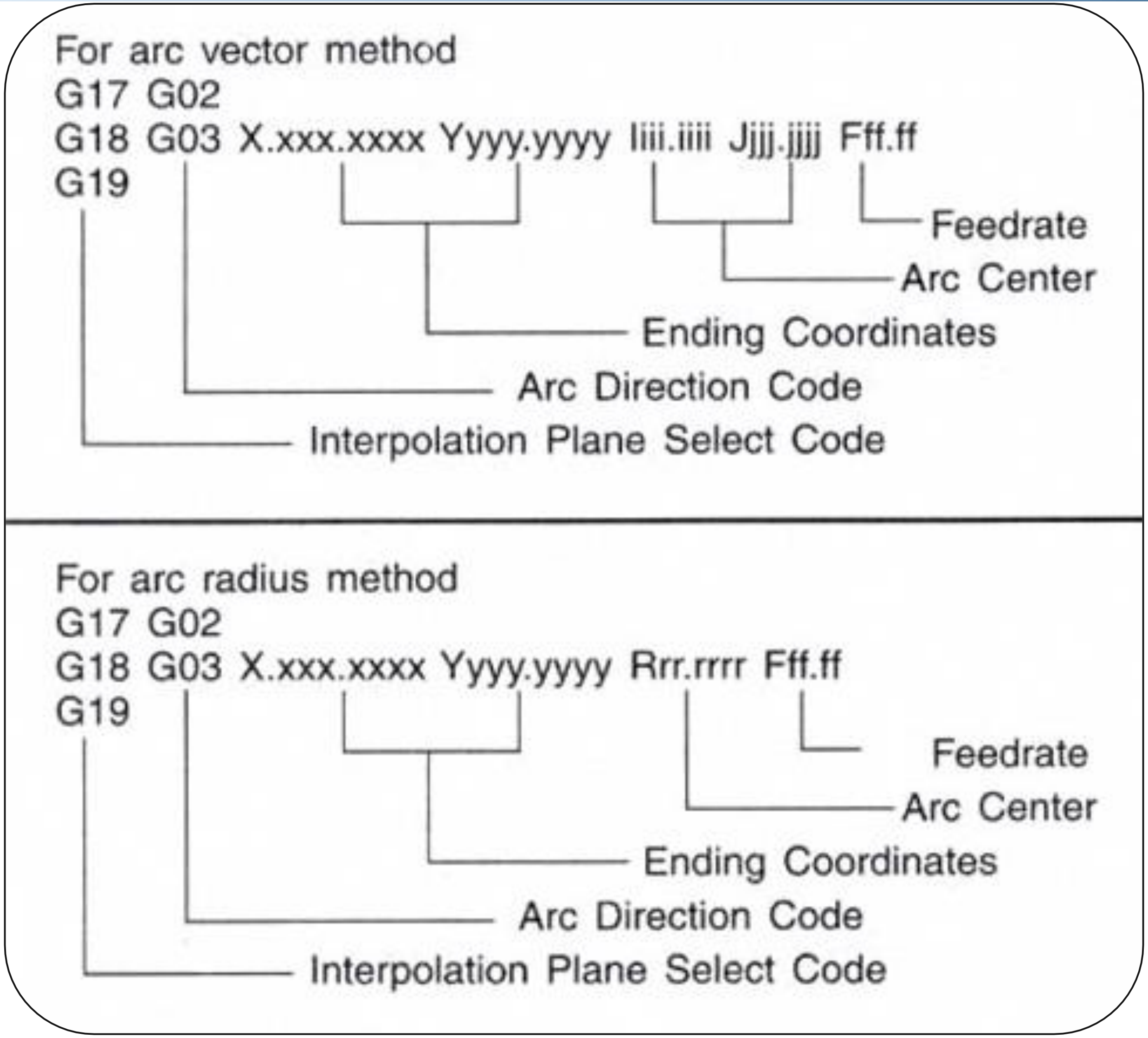


Figure 10:Arc vector method and radius method *(Seams W., "Computer Numerical Control, Concepts & Programming")*

Circular Interpolation

Arc Vector Method

- Since ***X, Y, and Z addresses are used to specify the end point of an arc***, **secondary addresses** are required to specify the **centerpoint of an arc**. The following addresses are used to designate arc center points

- **I—X-axis coordinate of an arc**. **J—Y-axis coordinate of an arc**. **K—Z-axis coordinate of an arc**

- Since **circular interpolation** occurs only in two axes, only two of these three codes will be required to generate an arc. When using the X/Y plane for milling arcs, as this text does, the **I** and **J** addresses are used

- The different ways controllers required the arc centerpoints to be specified complicate this matter: ***absolute coordinates***, ***to circle center***, or ***from circle center***. FANUC-style controls usually utilize the *to circle center* method

Circular Interpolation

Absolute Coordinates

- Some controls require the arc centerpoints specified by **I**, **J**, and/or **K** be the position of the **arc center relative to the coordinate system origin**

- In other words, the center of the arc is specified just as if it were a cutter coordinate using absolute positioning

- In Figure 8, the arc centerpoints are at X2.000, Y2.000. They would be specified as 12.0000 J2.0000 as in the following circular interpolation block:

N120 G17 G02 X3.Y2. 12. J2. F7

Circular Interpolation

To Circle Center

- Some controls require the **arc center points** be specified as an **incremental coordinate**, looking **from the center of the cutter to the center of the circle**
- In Figure 8, the radius of the arc is .750. The radius of the .500-diameter end mill is .25 inch
- To specify the centerpoint of the arc when the cutter is positioned at location #3, Figure 9 the incremental value of 0.0000 inch in X and -1.000 inch in Y would be specified as 10.0000 J-1.0000 as given in the following block of CNC code:

N120 G17 G02 X3.Y2. 10. J-1. F7.2

- The 1.000 incremental J value is calculated by adding the .250-inch cutter radius to the .750 part radius.
- A minus value is required since the direction from the cutter centerline to the arc centerline is in a minus direction
- The spindle is really generating a 1.000-inch arc when the cutter center is taken into account

Circular Interpolation

From Circle Center

- The **from circle center** method is the same as the **to circle center** **except the incremental coordinate is specified looking from the center of the arc to the center of the cutter**

- The signs associated with the **I**, **J**, and **K** addresses will be the **reverse** of the **to circle center** method

- The following line of code specifies the arc coordinates when the cutter is positioned at location #3, Figure 9

N120 G17 G02 X3.Y2. 10. J1. F7.2

Notice that the only difference between this block of code and the one given previously is the sign of the J address.

Seams W., "Computer Numerical Control, Concepts & Programming"

Circular Interpolation

Radius Method

- When using **the radius method**, the programmer only needs to **specify the radius to be cut** when programming the cut

- Instead of using the **I**, **J**, and/or **K** addresses **the R address** is used to specify the **arc radius**

- The following block of CNC code moves the cutter from location #3, Figure 9 to location #4 using the radius method

```
N120 G17 G02 X3.Y2. R1. F7.2
```

Notice that the radius to be cut is still 1.000 inch

- The controller is commanding motion of the spindle centerline. It does not know that there is a .500-inch diameter cutter in the spindle. The true cutter path is still a 1.000-inch arc

Circular Interpolation

- Although the radius method is easier to use than the arc vector method, the latter method is still common

- This is most likely because the radius method became available only with the advent of modern CNC controllers

- Many of today's programming practices have ties to the tape-controlled MCU of days gone by. This use of the arc vector method is one of these

Milling the Arc

- Putting together all these pieces, the following sections of CNC code will mill the part surface in Figures 8 and 9

Circular Interpolation

Arc Vector Method

N001 G80 G90 G00 G98	(optional stop code)
N100 T01 M06	(rapid to home zero)
N101 G00 X-0.25 Y-0.25	(cancel tool offset)
N102 G44 Z-1. H01	(retract Z)
N103 G01 Y3	(feed #5 to #1)
N104 X2.	(feed #4 to #5)
N105 G17 G02 X3. Y2. I2. J2. F7.2	(circular move to #4)
N106 G01 Y-0.25	(feed #2 to #3)
N107 X-0.25	(feed #1 to #2)
N108 G00Z0	(tool offset pickup)
N109 G49	(rapid to location #1)
N110 G91 G28 X0. Y0. Z0.	(tool change block)
N110 M01	(safety block)

Circular Interpolation

This example is typical **of FANUC, General Numeric**, and other **FANUC- style controls**:

N001 G80 G90 G00 G98	(optional stop code)
N100 T01 M06	(rapid to home zero)
N101 G00 X-0.25 Y-0.25	(cancel tool offset)
N102 G44 Z-1. H01	(retract Z)
N103 G01 Y3	(feed #5 to #1)
N104 X2.	(feed #4 to #5)
N105 G17 G02 X3. Y2. I2. J-1. F7.2	(circular move to #4)
N106 G01 Y-0.25	(feed #2 to #3)
N107 X-0.25	(feed #1 to #2)
N108 G00Z0	(tool offset pickup)
N109 G49	(rapid to #1)
N110 G91 G28 X0. Y0. Z0.	(tool change block)
N110 M01	(safety block)

Circular Interpolation

The following code uses the from **circle center method** to specify arc centerpoints:

N001 G80 G90 G00 G98	(optional stop code)
N100 T01 M06	(rapid to home zero)
N101 G00 X-0.25 Y-0.25	(cancel tool offset)
N102 G44 Z-1. H01	(retract Z)
N103 G01 Y3	(feed #5 to #1)
N104 X2.	(feed #4 to #5)
N105 G17 G02 X3. Y2. I2. J-1. F7.2	(circular move to #4)
N106 G01 Y-0.25	(feed #2 to #3)
N107 X-0.25	(feed #1 to #2)
N108 G00Z0	(tool offset pickup)
N109 G49	(rapid to #1)
N110 G91 G28 X0. Y0. Z0.	(tool change block)
N110 M01	(safety block)

Circular Interpolation

Radius Method

The following code mills the part using the **radius method** of specifying arc center coordinate:

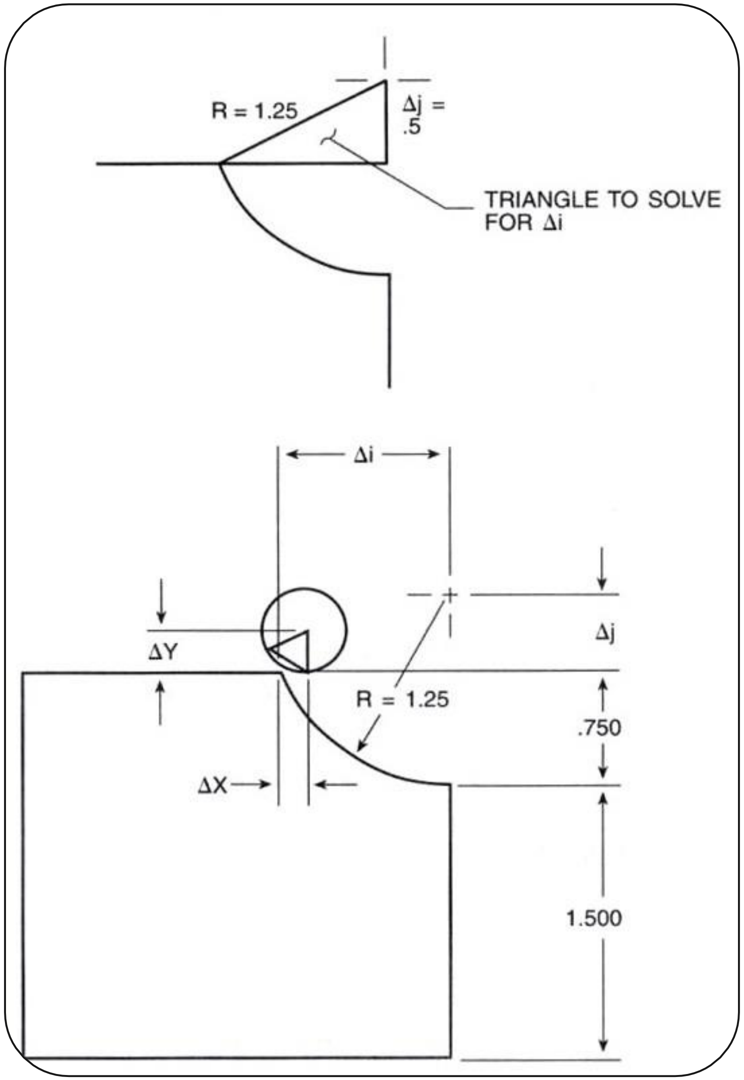
N001 G80 G90 G00 G98	(optional stop code)
N100 T01 M06	(rapid to home zero)
N101 G00 X-0.25 Y-0.25	(cancel tool offset)
N102 G44 Z-1. H01	(retract Z)
N103 G01 Y3	(feed #5 to #1)
N104 X2.	(feed #4 to #5)
N105 G17 G02 X3. Y2. R1. F7.2	(circular move to #4)
N106 G01 Y-0.25	(feed #2 to #3)
N107 X-0.25	(feed #1 to #2)
N108 G00Z0	(tool offset pickup)
N109 G49	(rapid to #1)
N110 G91 G28 X0. Y0. Z0.	(tool change block)
N110 M01	(safety block)

Circular Interpolation

Additional Circular Interpolation Examples

- The programs just discussed deal with **simple arcs which intersect a line parallel to a machine axis**
- In many cases, however, an **arc will intersect an angle or another arc**
- Figures 11 and 12 are examples of such cases
- The cutter offsets for these situations can be found by using the formulas from Appendix 6. The **cutter radius (CR)** in the following examples is .250 inch

Circular Interpolation



- Figures 11 and 12 are examples of cases of an arc intersecting an angle or another arc

Figure 11: Part drawing for additional example

(Seams W., "Computer Numerical Control, Concepts & Programming")

Circular Interpolation

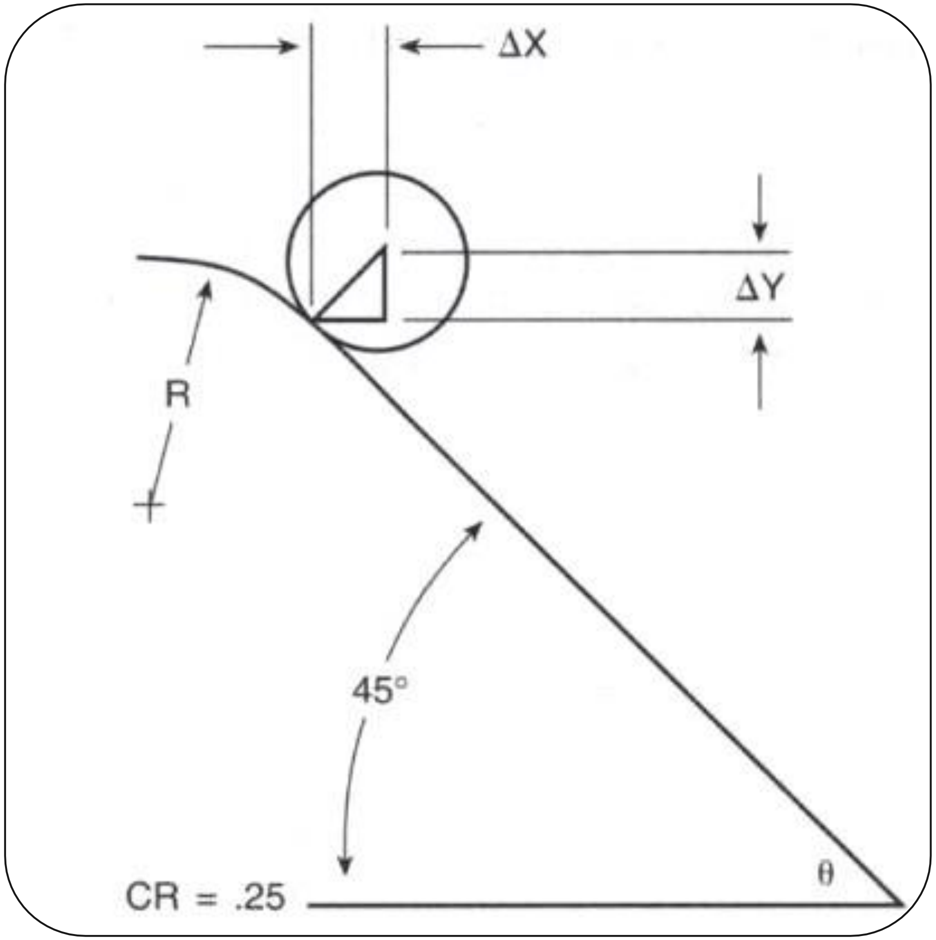


Figure 12: Circular Interpolation Example

Seams W., “Computer Numerical Control, Concepts & Programming”

Δj = 1.25 – 0.75
Δj = 0.5

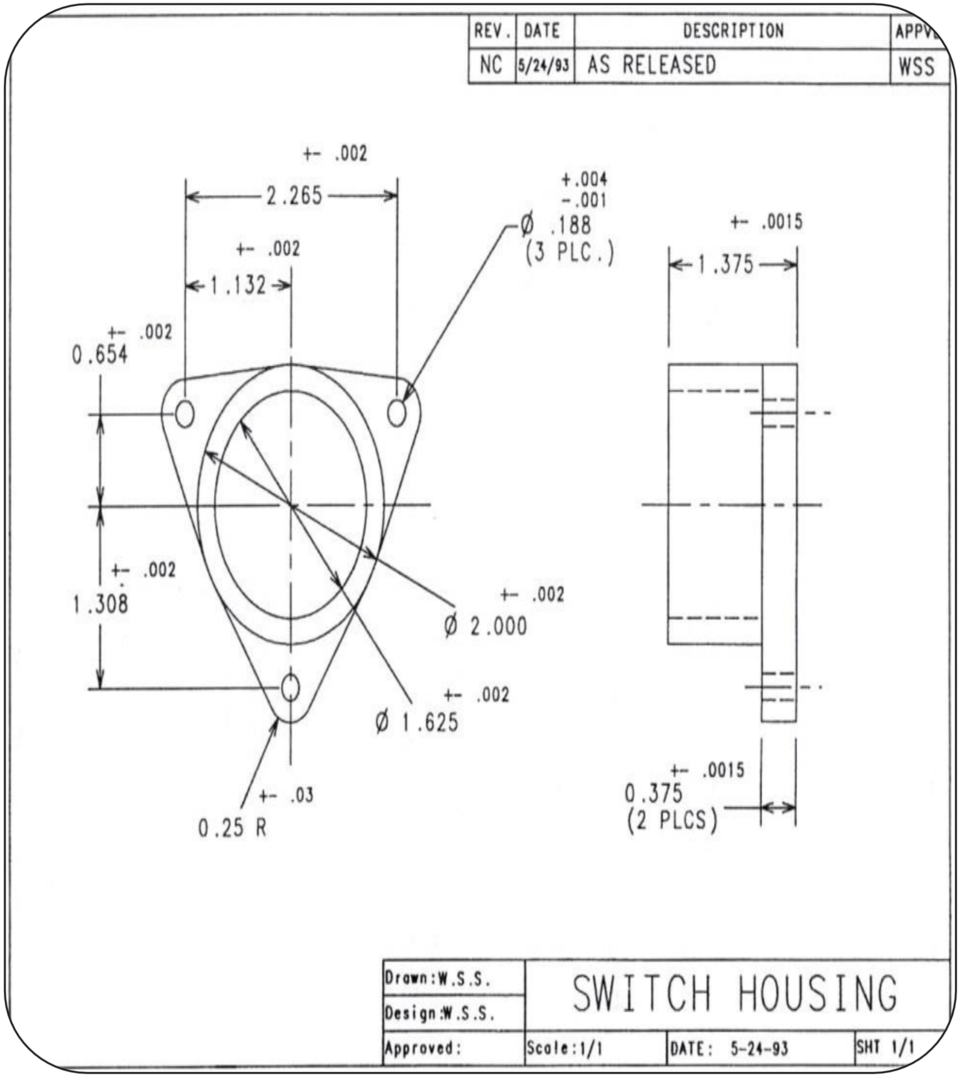
Δi = SQRT [(1.252 – 0.52)]
Δi = SQRT [(1.562 – 0.25)]
Δi = 1.1454

ΔY = CR
ΔX = Δi – SQRT [(R-CR)2 – (Δj – CR)2]
ΔX = 1.1454–SQRT[(1.125–0.25)–(0.5-.25)]
ΔX = 0.17715

ΔX = CR x SIN(45)
ΔX = 0.25 x (0.7071)
ΔX = 0.1769

ΔY = CR x COS45)
ΔX = 0.25 x (0.7071)
ΔX = 0.1769

A Comprehensive Example



- Figure 13 depicts a switch housing that is to be milled on a three-axis vertical machining center with a FANUC 6M control

Figure 13: Part drawing for comprehensive example *(Seams W., "Computer Numerical Control, Concepts & Programming")*

A Comprehensive Example

- The accompanying program serves as an example of a program using both linear and circular interpolation
- There are **no straight lines** on this part. All contouring moves will involve **angles** or **arcs**
- The setup for this part involves two fixtures as shown on the setup sheet in Figure 14
- The first fixture involves holding a slug of round stock in an OD collet . The second fixture holds the part in an OD collet on the 2.000 diameter which has been machined on the first fixture
- The summarized sequence of operations follows:

A Comprehensive Example

First Fixture

1. Rough and finish the top of the part, using a circular motion with a 1.0 – diameter end mill to achieve cutter coverage
2. Rough and finish the 1.625–diameter ID with 1.0 end mill
3. Rough and finish the 2.000–diameter OD with 1.0 end mill
4. Rough drill the 0.188–diameter holes to 0.177–diameter with 0.177–diameter stub drill
5. Finish ream the 0.188–diameter holes to 0.188–diameter with a 0.188–diameter reamer

Second Fixture

1. Rough and finish the top of part to establish 1.375 overall dimension with a 1.0–diameter end mill
2. Rough and finish the outside periphery of the part using a 0.625–diameter end mill

The program manuscript is given in Figure 9-13. The circular interpolation moves in this program use the cutter vector method of specifying arc center coordinates. The arc centers are specified using *to circle center* coordinates, as is commonplace with FANUC-style-controls.

A Comprehensive Example

STA. NO.	TOOL DESCRIPTION
1.	1" END MILL, 4-FLT CENTER CUTTING (Ø1.000)
2.	Ø.177 STUB DRILL
3.	Ø.1880 REAMER
4.	5/8 END MILL, 2FLT (Ø.625)

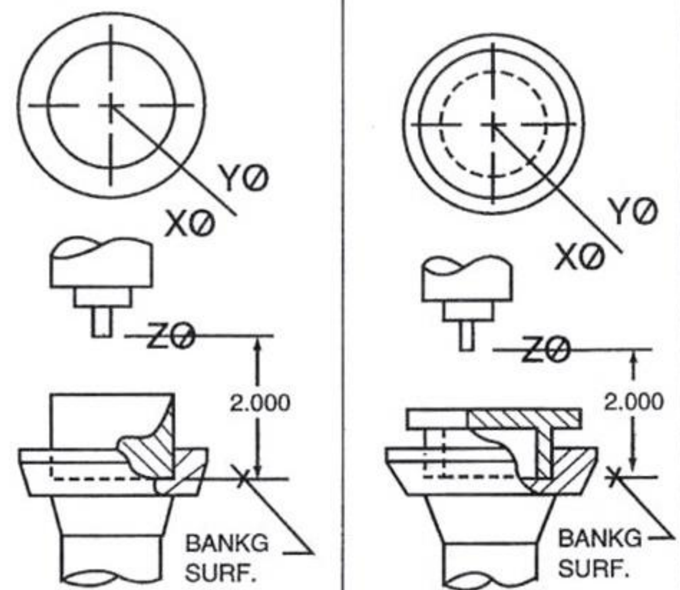
REMARKS

TAPE NUMBER: 9012

FIXTURE: C-1232 & C-1233 COLLETS

WORK COORDINATES/FIXTURE OFFSETS

92 FIX 1		92 FIX 2	
C/L COLLET		C/L COLLET	
C/L COLLET		C/L COLLET	
2" ABOVE BANK		2" ABOVE BANK	



FIX #1 USES OFFSETS HØ1, HØ2, HØ3

FIX #2 USES OFFSETS H12 & HØ4

DRWN: WSS	PART NUMBER: SWITCH HOUSING	OP: 20
PRGM: WSS	MACHINE: UNIVERSAL VNC	
APVD: 7/20/93		

Figure 14: Setup sheet for part in Figure 13

Seams W., "Computer Numerical Control, Concepts & Programming"

LAB A Comprehensive Example

```
%
O9013
(SWITCH HOUSING OPERATION 30 06/18/93 07:11:54)
N1 G91 G28 Z.0 G28 X.0 Y.0
N2 G28 X.0 Y.0 Z.0
N3 G92 X-12.752 Y-7.453 Z.0
(* **********)
(* TOOL NO. 1)
(* 1.000 DIA. END MILL)
(* ROUGH/FINISH TOP OF PART)
(* **********)
N4 T01M06
N5 G90G00G80
N6 G00X.0Y.0S1800M03
N7 G44Z-.5H01M08
(ROUGH THE TOP OF THE PART)
(LEAVE .010 STK. TO FINISH)
N8 G01 Z-1.615 F8.
N9 Y.5 F20.
N10 G17 G03 X.0 Y.5 I.0 J-.5
N11 G01 Y.0
(FINISH TOP OF PART)
N12 S3000 M03
N13 Z-1.625 F16.
N14 Y.5
N15 G03 X.0 Y.5 I.0 J-.5
N16 G01 Y.0
(ROUGH 1.625 DIA. BORE)
N17 S1800 M03
N18 Z-.615 F7.
N19 Y.3025
N20 G03 X.0 Y.3025 I.0 J-.3025
N21 G01 Y.2025
(FINISH 1.625 DIA. BORE)
N22 Z-.625 F15.
N23 Y.3125
N24 G03 X.0 Y.3125 I.0 J-.3125
```

```
N25 G01 Y.0
N26 G00 Z.0
(ROUGH 2.000 DIA.)
N27 Y-3.
N28 Z-.615
N29 G01 Y-1.51
N30 G02 X.0Y-1.51 I.0 J1.51
N31G01Y-1.61
(FINISH 2.000 DIA.)
N32 Z-.625
N33 Y-1.5
N34 G02 X.0 Y-1.5 I.0 J1.5
N35 G01 Y-3.
N36 G49M09
N37 G91 G00 G28 Z.0
N38 G28 X.0 Y.0
N39 M01
(* **********)
(* TOOL NO. 2)
(* .177 DIA. STUB DRILL)
(* DRILL HOLES IN FLANGE)
(* **********)
N40 T02 M06
N41 G90 G00 G80
N42 G00 X1.132 Y.654 S2200 M03
N43 G44 Z-.525H02M08
N44 G81 G98 X1.132 Y.654 Z-2.125 R-1.525
        F8.8
N45 X-1.132
N46 X.0 Y-1.308
N47 G80
N48 G49 M09
N49 G91 G00 G28 Z.0
N50 G28 X.0 Y.0
N51 M01
(* ***********)
```

```
(* TOOL NO. 3)
(* .188 DIA. REAMER)
(* FINISH REAM HOLE PATTERN)
(* ***********)
N52 T03M06
N53 G90G00G80
N54 G00 X1.132 Y.654 S1200 M03
N55 G44 Z-.525 H03 M08
N56 G85 G98 X1.132 Y.654 Z-2.075 R-1.525 F9.6
N57 X-1.132
N58 X.0 Y-1.308
N59 G80
N60 G49 M09
N61 G91 G00 G28 Z.0
N62 G28 X.0 Y.0
N63 M01
(* ***********)
(* SET COORDINATE SYSTEM FOR 2ND
        FIXTURE)
(* ***********)
N64 G28 X.0 Y.0 Z.0
N65 G92 X-8.253 Y-7.253 Z.0
(* ***********)
(* RECALL TOOL 1)
(* 1.000 DIA. END MILL)
(* MILL TOP OF PART ON FIXTURE 2)
(* USES LENGTH OFFSET H11)
(* ***********)
N66 T01 M06
N67 G90 G00 G80
N68 G00 X.0 Y3. S2000 M03
N69 G44 Z-.615 H11 M08
(ROUGH TOP OF PART)
```

Figure 15: Program to machine part in Figure 13

(Seams W., "Computer Numerical Control, Concepts & Programming")

A Comprehensive Example

```
(ROUGH TOP OF PART)
N70 G01 Y.5 F16.
N71 G02 X.0 Y.5 I.0 J-.5
N72 G01 Y.0
(FINISH TOP OF PART)
N73Z-.625
N74 Y.5
N75 G03 X.0 Y.5 I.0 J-.5
N76 G01 Y3.
N77 G49 M09
N78 G91 G00 G28 Z.0
N79 G28 X.0 Y.0
N80 M01
(* ***********)
(* TOOL NO. 4)
(* .625 DIA. END MILL)
(* MILL OUTSIDE PERIPHERY OF PART)
(* ***********)
N81 T04 M06
N82 G90 G00 G80
N83 G00 X.0 Y3. S2200 M03
N84 G44Z-1.15H04M08
(ROUGH OUTSIDE)
N85 G01 Y1.3225 Z-1.1F12.
N86 G02 X.1151 Y1.3175I.0J-1.3225
N87 G01 X1.1818 Y1.2243
N88 G02 X1.651 Y.4123I-.0498J-.5703
N89 G01 X1.1988 Y-.5584
N90 G02 X1.0835 Y-.7583 I-1.1988 J.5584
N91 G01 X.469 Y-1.6363
N92 G02 X-.469 Y-1.6363 I-.469 J.3283
N93 G01 X-1.0835 Y-.7583
N94 G02 X-1.1988 Y-.5584 I1.0835 J.7583
N95 G01 X-1.651 Y.4123
```

```
N96 G02 X-1.1818 Y1.2243 I.519 J.2417
N97 G01 X-.1151 Y1.3175
N98 Y1.4175
(FINISH OUTSIDE)
N99 G00 X-1.6945 Y1.2699
N100 G01 Y1.1695
N101 X-.1142 Y1.3075
N102 G02X.1142Y1.3075 I.1142 J-1.3075
N103 G01 X1.1809 Y1.2144
N104 G02 X1.6419 Y.4165 I-.0489 J-.5604
N105 G01 X1.1898 Y-.5542
N106 G02 X1.0753 Y-.7526 I-1.1898 J.5542
N107 G01 X.4608 Y-1.6305
N108 G02 X-.4608 Y-1.6305 I-.4608 J.3225
N109 G01 X-1.0753 Y-.7526
N110 G02 X-1.1898 Y-.5542 I1.0753 J.7526
N111 G01 X-1.6419 Y.4165
N112 G02 X-1.1809 Y1.2144 I.5099 J.2375
N113 G01 X.1 Y1.3262
N114 G00 Z.0
N115 G49 M09
N116 G91 G00 G28 Z.0
N117 G28 X.0 Y.0
N118 M01
N119 G91 G00 G28 X.0 Y.0 Z.0
N120 M30
%
```

Figure 16: Program to machine part in Figure 13 *(Seams W., "Computer Numerical Control, Concepts & Programming")*

Summary 1/3

- **Linear interpolation** is the ability to **cut angles**. It is simply a feedrate move, in a straight line, between two points

- **Circular interpolation** is the ability to **cut arcs** or **arc segments**. Arcs are cut by means of a series of choral segments generated by the **MCU** to approximate the arc curvature

- It is necessary to calculate the **cutter offset coordinates** when using **linear** and **circular interpolation**

- **G01** is the code to institute **linear interpolation**. It also is referred to as the feedrate move code

- **G02** and **G03** are used to **institute circular interpolation**

Summary 2/3

- **G02** turns on clockwise interpolation. **G03** turns on counterclockwise interpolation

- There are two methods used to specify the arc center-points to the **MCU**: the **arc vector method** and the **radius method**

- When using the **arc vector** to specify **center-points**, some controls require the center-points to be given in **absolute coordinates**, some in **incremental coordinates** from the **cutter center to the circle center**, and other in incremental coordinates from the **circle center to the cutter center**

- The format for circular interpolation for the arc vector method is:

G17 G02
G18 G03 X… Y… Z… I… J… K…
G19

For most uses (X/Y plane interpolation) the format is:

G17 G02/G03 X… Y… I… J…

Summary 3/3

- The format for **circular interpolation** for the radius method is:

G17 G02
G18 G03 X… Y… Z… R…
G19

- For most uses (**X/Y plane interpolation**) the format is:

G17 G02/G03 X… Y… R…

Vocabulary Introduced in this chapter

- Arc center-points
- Arc vector method
- Circular interpolation
- From circle center
- Linear interpolation
- Radius method
- To circle center

References

1. Chryssolouris G., «Manufacturing Systems: Theory and Practice», 2nd Edition, 2006, Springer-Verlag

2. http://www.dptechnology.com/

3. Kalpakjian S., «Manufacturing Engineering and Technology», 2nd Edition, 1992, Addison-Wesley Publishing company

4. Mattson M., “CNC Programming, Principles and Applications”, Delmar, 2002

5. Seams W., “Computer Numerical Control, Concepts & Programming”, 4th Edition, Delmar, 2002

6. Γ. Χρυσολούρης, «Συστήματα Παραγωγής Θεωρία και Πράξη» Μέρος I και II, Εκπαιδευτικές Σημειώσεις, Πανεπιστήμιο Πατρών, 2001,

7. Δ. Μούρτζης, “Αριθμητικός Έλεγχος Εργαλειομηχανών” Εκπαιδευτικές Σημειώσεις, Πανεπιστήμιο Πατρών, 2011 (3η Έκδοση)

8. Πετρόπουλου Π.Γ., «Μηχανουργική Τεχνολογία – II. Τεχνολογία κατεργασιών κοπής των μετάλλων», 1998, Εκδόσεις Ζήτη

9. Σύγχρονες μέθοδοι κατεργασίας υλικών και προγραμματισμός με Ηλεκτρονικό Υπολογιστή (Η/Υ) ,Δ. Μούρτζης ,Κ. Σαλωνίτης