ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά μαθήματα ΠΠ

Αριθμητικός Έλεγχος Εργαλειομηχανών

Ενότητα 9: Linear and Circular Interpolation

Δημήτρης Μούρτζης, Επίκουρος Καθηγητής
Πολυτεχνική Σχολή
Τμήμα Μηχανολόγων & Αεροναυπηγών Μηχανικών

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

Assistant Professor

Patras, 2015

Objectives of section 9

- 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 feedrate 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

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

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 feedrate 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**

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

- **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**

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 **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
- These chord segments are very small and practically indistinguishable from a true arc

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

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 9-8)
- 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

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

- IF 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

- 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

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**

- The cutter offsets for these situations can be found by using the formulas

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 section

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

End of Section

Funding

- This educational material has been developed in the teaching duties of the respective educator.
- The Project “**Open Academic Courses at the University of Patras**” has funded only the reformation of the educational material.
- The Project is implemented within the context of the Operational Programme "Education and Lifelong Learning" (EdLL) and is co-funded by the European Union (European Social Fund) and national resources.

European Union
European Social Fund

OPERATIONAL PROGRAMME
EDUCATION AND LIFELONG LEARNING
investing in knowledge society
MINISTRY OF EDUCATION, LIFELONG LEARNING AND RELIGIOUS AFFAIRS
MANAGING AUTHORITY

NSRF
2007-2013
programme for development
EUROPEAN SOCIAL FUND

Co- financed by Greece and the European Union

Reference Note

Copyright University of Patras, School of Engineering, Dept. of Mechanical Engineering & Aeronautics, Dimitris Mourtzis. Dimitris Mourtzis. «Computer Numerical Control of Machine Tools. Linear and Circular Interpolation». Version: 1.0. Patras 2015. Available at: https://eclass.upatras.gr/courses/MECH1213/

License Note

Notes Preservation

Any reproduction or modification of this material must include:

- the Reference Note
- the License Note
- the Notes Preservation statement
- the Third Parties’ Works Note (if exists)

as well as the accompanying hyperlinks.