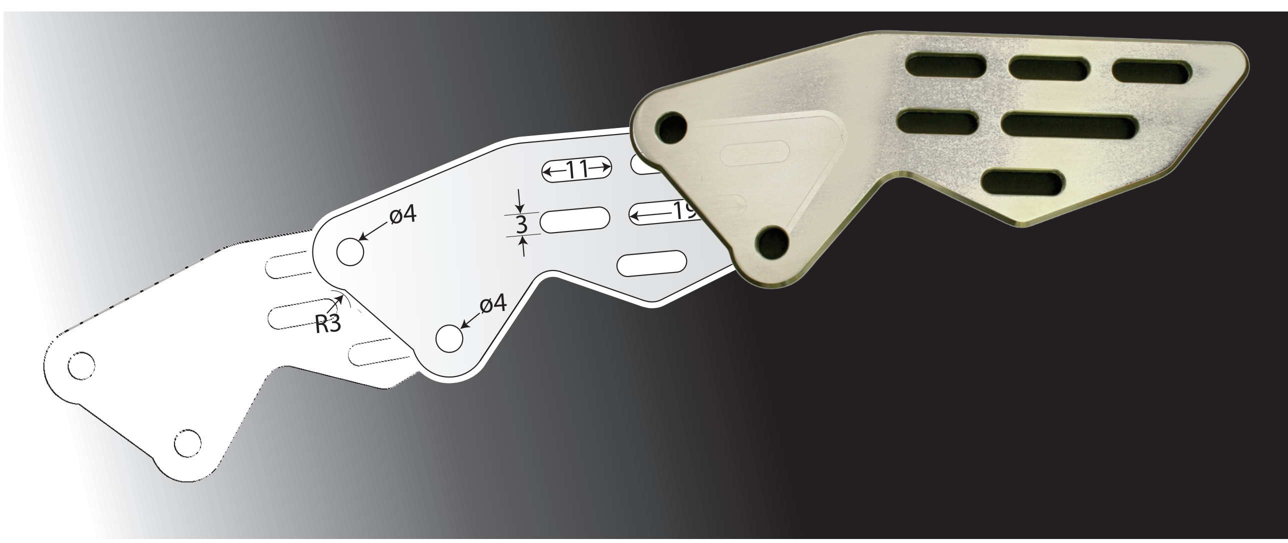
CNC in the Workshop



Part 9

Part 9

In this part of the series, we look at circular paths.

ALL WE HAVE

If all we had were straight line moves, we would still be able to do some good work with our CNC machines. That's just as well, because the straight line move *is* all we have. But its surprising where straight lines can take you. Remember that the control software, and the NIST core within that, can make the Controlled Point (CP) travel in a straight line between any two points.

A line is fine, and if we combine a sequence of moves we can travel around a square path. If we do a bit of trigonometry, we can use straight lines to travel around the periphery of any polygon. A polygon simply means a shape with many corners, and in a regular polygon the sides are all the same length and the angles are all the same. Fig 39 shows that if we start with a square and increase the number of sides, we soon get to the stage where the shape looks awfully like a circle. You might think a circle has a curved periphery, but unless we use a rotary table, a CNC circle is a polygon with a lot of small straight lines. The more lines there are, the shorter they will be, so the better the polygon will resemble a circle, and because we can easily cut lots of sides we can usually get a pretty good result. Certainly good enough for everyday use, and limited only by the effects of backlash and rigidity.

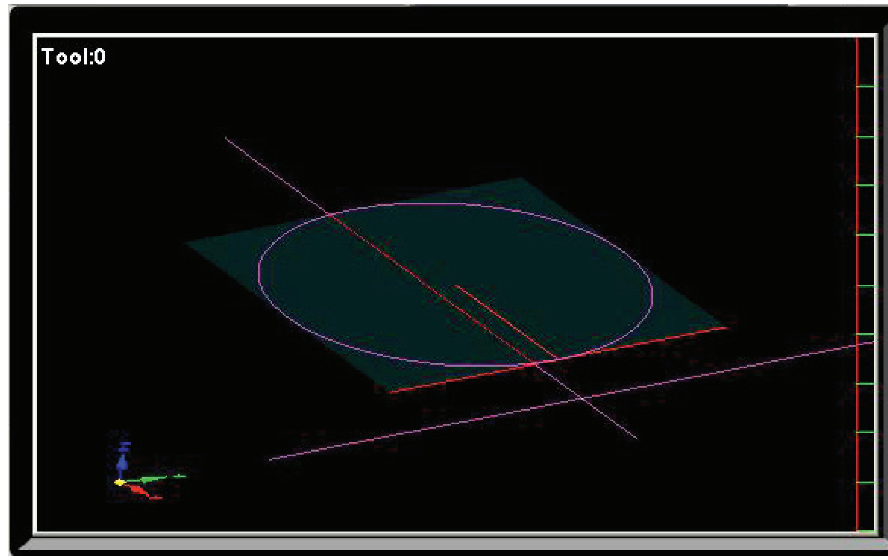


Fig 40: Flat circular tool path, showing CP travelling round in a circle at a constant Z height..

The trouble is, working out where the vertices (corners) are for a square is usually not too difficult, and doing the same for a polygon with a few more sides is do-able, even if it requires more troublesome calculations, but calculating the co-ordinates for lots and lots of sides leads to what the Victorians often called brain fag.

Fortunately, there are G code commands which can do that for us. In fact, we can simply use two commands designed specifically to move around a circular path, and all the relevant calculations are done while the machine moves the CP.

G1 moves the CP in a straight line. G2 moves the CP in what appears to be a circular path, clockwise (as viewed looking down through the Z axis and the mill spindle).

G3 moves in a "circular" path, anticlockwise. We will refer to these G2 and G3 paths as being circular, from now on, but they are not; they just use small straight lines to approximate a circular path. That should be good enough, under most circumstances. When we run into unexpected problems, we can return to the fundamental truth that the CP follows a polygonal path.

A circular path can be a part-circle (i.e. an arc) or a full circle, or more than a full circle. The path can be "flat", parallel to the mill table and in the X-Y plane (fig 40), or it can be a spiral, moving the Z axis up or down as the CP moves in an XY circle. That creates a spiral path which looks a lot like a helical spring (fig 41).

To be able to work out the path, Mach3 needs to know some basic information. If you were going to draw a circle on paper, you would want to know the position of the centre of the circle, and the radius of the circle (the distance from the centre of the circle to any point on the circumference of the circle) as shown in fig 42. Using G code, there are two

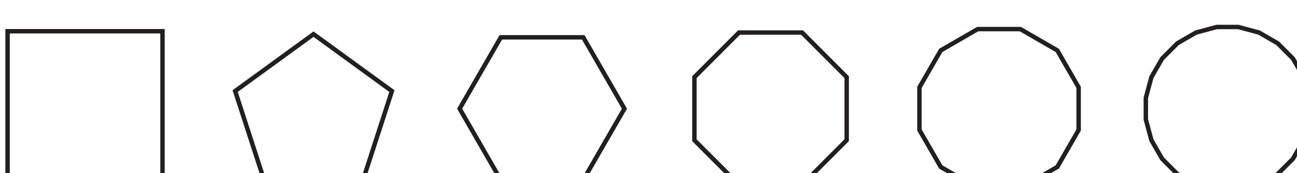


Fig 39: The more sides on a polygon, the closer it approximates to a circle..

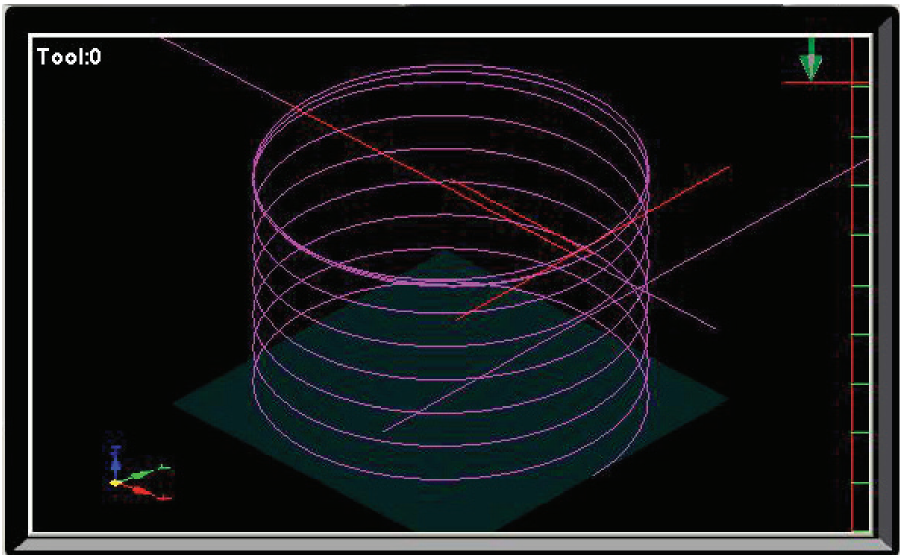


Fig 41: Helical tool path, showing the CP spiralling down into the work..

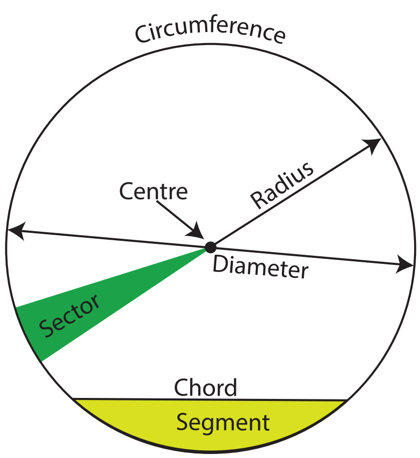


Fig 42: Parts of a circle.

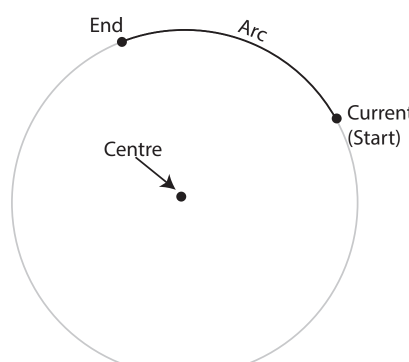


Fig 43: An arc, showing the Current (Start) and End points and the Centre..

basic ways to specify the size and position of the circle. For the moment, we will use just one of those methods, because it's more versatile. It's the "Centre Format Arc".

First, the CP must be on the circumference of the circle (fig 43), so this is a circular move from where the CP is right now. If you want to machine a circle or an arc somewhere else, you will have to go there first. We will refer to this starting position as the current location, or the current point.

Next, we need to know where the end of the path will be. If it is to be a full circle, that would be back at the point where we are now. But if the CP is to move in an arc and not a full circle, Mach3 needs to know where to stop. Sometimes it is easy to identify that point, and at other times we may need to do a bit of careful reasoning from a drawing, but we do need to know that point. We will call that the end point.

Finally, we need to know where the centre of the circle is, in relation to the current point. The G code interpreter can then work out the radius of the circle, which will be the distance from the current point to the centre of the circle. It is aware of the starting point because that's where the CP is right now, and it can simply travel around the circle until it gets to the end point.

There are two ways to specify the centre of the circle. One is to specify the actual co-ordinates of the centre, and it is called Absolute IJ mode. This is not too handy, because it nails down the centre of that circle quite firmly, and we will see later that it is useful to have a bit of flexibility so that we can cut the same circular path in a different place. If we specify the actual co-ordinates (i.e. the absolute co-ordinates) we will find it difficult to do that, because we have said where the centre of the circle is always to be.

The second way is to specify the X and Y distances from the current point to the centre of the circle. That means the centre is always placed in relation to where the CP is currently located. If we move the CP to another place, we can cut a similar circle there because the centre is located in relation to the current point. There is a wrinkle or two, but that's a big advantage. This second way is our recommended method, for now, and its called Incremental IJ mode. Such handy names.

We can specify which mode we want by using the command G91.1 to set Incremental IJ mode, or by using G92.1 to set Absolute IJ mode.

In a program, our recommended initialisation block already contains G91.1, so that takes care of that, while that program is running.

You can also type G91.1 in MDI mode. But to simplify things, in case you forget, Mach3 can be set up to use G91.1 by default. From

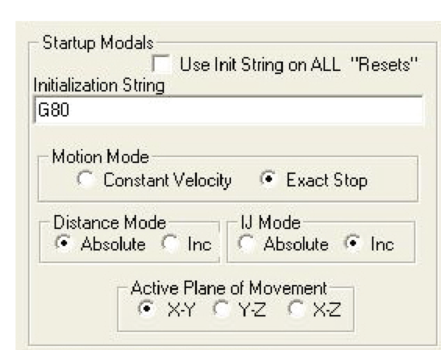


Fig 44: Set the IJ Mode from the Config>General Config menu..

the menus, choose Config > General Config; look in the second column from the right, about half way down, in Startup Modals, and set IJ Mode to Incremental (fig 44). When Mach3 starts up, it issues some commands to set itself up, and this is where it gets some of its settings.
Good practice means we should specify the IJ Mode within a program, for example, but having this set as the default condition means if we are working in MDI mode and not running a program, we can still use Incremental IJ mode without having to issue a G91.1 command every time.
For now, you should also check the box marked "Exact Path".

An example

After all that preamble, let's take an actual example.

Fig 45 and photo 73 show a circular plate with a pair of circular holes and an arc.
Let's assume a workpiece which is a 1.5mm thick piece of aluminium or brass sheet, 50 x 50, with the work origin (0, 0) at the front left corner.
The programs which follow assume the use of a 2mm diameter end mill or slot drill, but you could use a 3mm or 3.16mm diameter cutter if you adjust some of the sizes given in the example programs.

The largest arc is a shallow groove, 0.25mm deep. The two smaller circles are cut out of the workpiece, then the circumference of the outer circle is cut to full depth to release it from the sheet.

The centre of the large circle is at (24,26). You may wonder why it's not dead centre, at (25, 25). That's because this you will not learn as much if X and Y have the same value. Think of it as an excuse for more advanced mental gymnastics.

Taking the green arc first:
If the CP is at the left hand end of the arc, at (9, 34) and the arc ends at (39, 34) then the CP will need to move clockwise, so we need to use a G2 command.
The format of a circular move is:

G2 X~Y~Z~A~B~C~I~J~

where the ~character (the tilde character) stands for a number.
We will not be using the A, B or C axes for a while, so we can simplify that to:

G2 X~ Y~ Z~ I~ J~

The X, Y and Z values are the co-ordinates of the end point of the arc.
I is the distance along X from the current point to the X co-ordinate of the centre of the arc. Calculate it using the formula I = Xcentre – Xcurrent.
In this case, I = 24 – 9 = 15
J is the distance along Y from the current point to the Y co-ordinate of the centre of the arc. Calculate it using the formula J = Ycentre – Ycurrent.
In this case, J = 26 – 34 = -8
So our command becomes:
G2 X39 Y34 Z-0.2 I15 J-8

To try that out, set Z0 with the tool at the top surface of a convenient piece of material. Then, altering the following code to take account of any obstructions, cutting in air 10mm above the work:
G0 Z10
G0 X9 Y34
G2 X39 Y34 Z10 I15 J-8

How could we take that cut, for real? If you use a centre cutting end mill or slot drill, you could use

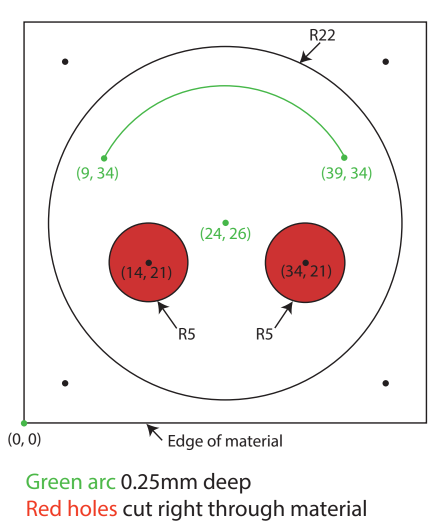


Fig 45: Circular workpiece with 2 through holes and a shallow arc.

Photo 73: The finished work.

G0 Z10 (take the cutter above the work)
G0 X9 Y34 (position the CP above the start of the arc)
G0 Z0.01 (lower the CP to just above the work)
G1 Z-0.2 (take a plunge cut to depth)
G2 X39 Y34 Z-0.2 I15 J-8 (cut at constant depth)

Or you could position the CP on the top surface, at the start of the arc; cut around the arc, descending as you go, reaching full depth at the end. Then cut back at constant depth to the start of the arc. That way, the cutter will gradually ramp into the work as it cuts, then remove the tapering bottom of the arc on the way back.

If the return cut starts at the right hand end of the arc,
Xcurrent = 39, Ycurrent = 34
I = Xcentre – Xcurrent = 24 – 39 = -15
J = Ycentre – Ycurrent = 26 – 34 = -8
The cutter travels anticlockwise this time, so the return cut for the arc is:
G3 X9 Y34 Z-0.2 I-15 J-8
and the code becomes:

G0 Z10 (take the cutter above the work)
G0 X9 Y34 (position the CP above the start of the arc)
G0 Z0.01 (lower the CP to just above the work)
G2 X39 Y34 Z-0.2 I15 J-8 (cut clockwise, ramping into the work)
G3 X9 Y34 Z-0.2 I-15 J-8 (cut anticlockwise, at constant depth)
G0 Z10 (lift the CP clear)

Look at where you are going

If we were orienteering, I imagine it would be useful to look up from the map, from time to time, and use common sense to decide in which direction we should travel. Oh look; it's over there, to the left… 500 metres away.

I and J state how much the CP needs to travel from the current point to get to the centre of the arc. Imagine standing at (9, 34). How would you move the CP to get to (24, 26)? For X, you would move it 15 to the right, so I15. For Y, you would move it 8 towards the front of the table. That's travelling in the negative Y direction, and that's -8 units, so J-8.
This amounts almost to counting on one's fingers, but it gives a useful check on our calculations.

Arithmetic interlude

The arithmetic of all of this is quite simple, except that it is sometimes not. Negative numbers can cause local overheating of long dormant brain circuits, especially when there are multiple negatives in a subtraction. Forgive me for teaching my grandpa to suck eggs, but here's how I see it:
A sum like 39 – 24 should cause no trouble, because we are subtracting two positive numbers. But look at this arc (fig 46):
If the CP is at X-21 Y34
I = Xcentre – Xcurrent = -6 - -21 = 15
If you are comfortable with that calculation, just skip the rest of this arithmetic explanation. If not, please continue.
Just to be pedantic, every number has a sign. Where it is a positive number, we usually omit the sign, but it is useful to leave it in, sometimes. In that case, our sum becomes:
(-6) – (-21)
There are a couple of simple rules as to how to handle multiple signs which occur together:
+ next to + is the same as one + sign
+ next to – is the same as one – sign
- next to – is the same as one + sign

So our sum becomes
-6 +21
Now look at the size of the numbers. 21 is bigger than 6. How much bigger? 15. But the biggest number is positive, so the result will be +15 (written simply as 15, omitting the positive sign).
Check using the Look Where You Are Going method. Standing at the current point (-21, 34), is the centre of the arc 15 units to the right (or 15 units towards the back of the table, for Y movements)?
If the CP is at the right hand end of the arc, at (9, 34),
Xcentre – Xcurrent = -6 - +9 = -6 -9 = -15
Check: Is the X co-ordinate of the centre 15 units to the left of the current point?

CUTTING OUT A CIRCULAR HOLE

Now look at the left hand circle in fig 45. It's centre is at (14, 21) and its radius is 5,

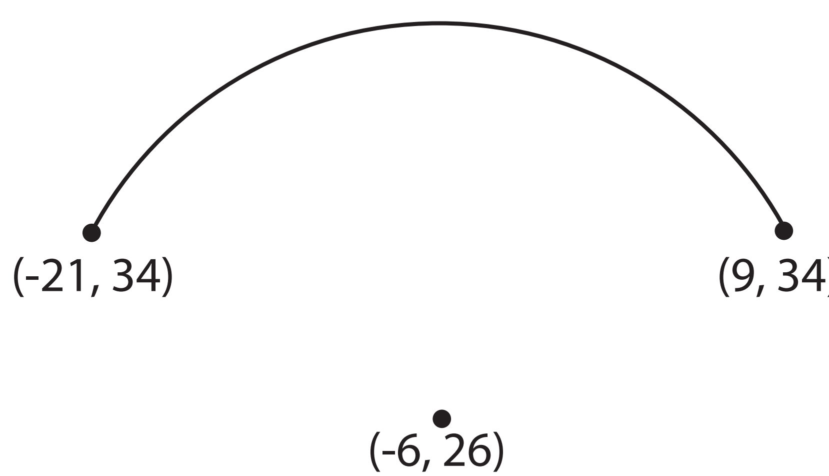


Fig 46: Another axample arc.

We could position the cutter over the circumference of the circle, and cut an arc all the way round, finishing where we started. Trouble is; that wouldn't give us what we want. Unlike the previous arc, where we simply cut along the path of the arc, this time we must take into account the radius of the cutter. Fig 47 shows that when we cut inside a circle, to leave the circumference at the final finished size, the path of the CP is inside the circumference, and its radius is the radius of the circle minus the radius of the cutter. That's because the cutter cuts on its outer edge, which is further away from the centre of the circle than the CP.

Assuming a 1mm radius (2mm diameter) cutter and a 5mm radius (10mm diameter) circle, the CP needs to follow a path with 4mm radius (i.e. 5 – 1mm).

Where to start? It doesn't matter much, for this circle. Move the CP to (18, 21) which is as good as any other point. From the centre, moving to left or right, up or down takes us to a point whose co-ordinates are easy to calculate. It is unnecessarily difficult to calculate the co-ordinates of any other point on the circumference of the circle, unless you fancy a bit of recreational mathematics.

The cutter moves around the inside of the circle, and the spindle turns clockwise, so conventional milling means the teeth of the cutter should bite into the material as it moves forwards. That means the cutter should move clockwise.

The End of the path will be the same as the Start: X18 Y21

The Centre is at X14 Y21 so I = 14 -18 = -4 and J = 21 – 21 = 0

That makes sense, because standing at the current point (18, 21) which will be the Start, moving 4 units in the –X direction (to the left) takes the CP to the centre of the circle.

G0 Z10 (Safe Z)
G0 X18 Y21
G0 Z0.01
G2 X18 Y21 Z-0.25 I-4 J0

will spiral around the circle, descending as it goes, until Z is -0.25

Add some G2 commands with the same values except that Z should increase each time, perhaps in increments of -0.25, until Z is –1.50

That reaches full depth but leaves the ramped part created at the beginning of the last pass. There's a choice here:

Continue with the G2 commands, making the last two Z values -1.75 and -2.00, or take the cutter down just below the bottom surface of the work, then do a final pass at that Z height, to leave a flat bottom on the groove, removing the remaining tapered section left from the previous cut. The pass with Z-2.00 is simply to ensure that the cutter does cut all the remaining material and is right through the sheet, because the previous cut started at Z-1.50 and there may be small errors in the sheet thickness or our Z positioning.

G2 X18 Y21 Z-1.60 I-4 J0 (ramps down from the last height)
G2 X18 Y21 Z-1.60 I-4 J0 (flat bottom)

That frees the centre from the rest of the disc, but the cutter will cut into the surface under the workpiece, so that should be the same material, or softer, than the workpiece itself. It also means that we need to get into the habit of regarding the supporting material as sacrificial, because it is going to get cut ever time the CP descends below the bottom of the work; and that will be often.

The waste should break free, so be ready to step out of the way. I recommend blowing the waste away as it breaks free, using an airline or a puffer of some sort. The waste itself won't do much harm, but with a small cutter working in a small hole like this there is always the danger of damage if the waste jams between the circumference of the circle and the cutter.

The other circle, with centre at X34 Y21 can be cut in just the same way. In fact the commands will be the same except for the values of Xcurrent and Ycurrent.

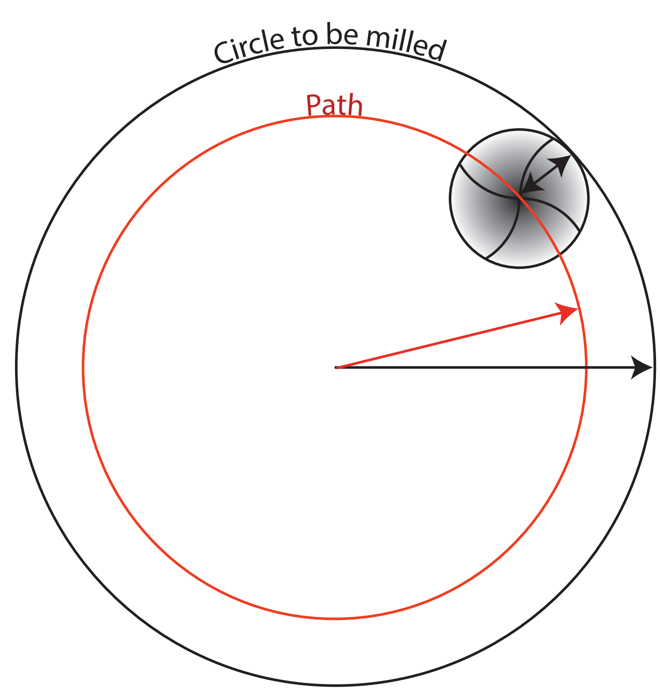


Fig 47: The CP lies inside the circle, displaced by the radius of the cutter.