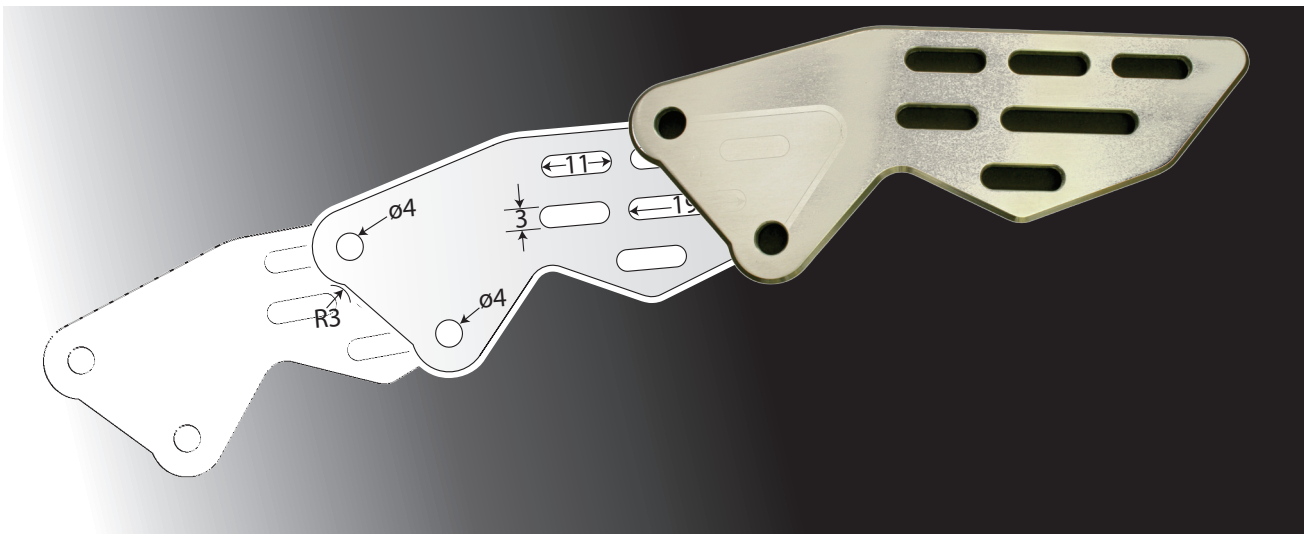
# CNC in the Workshop






# Part 7

# Part 7

*In this part of the series, we continue making the Vice Stop and illustrating many of the basics of using a CNC program.*

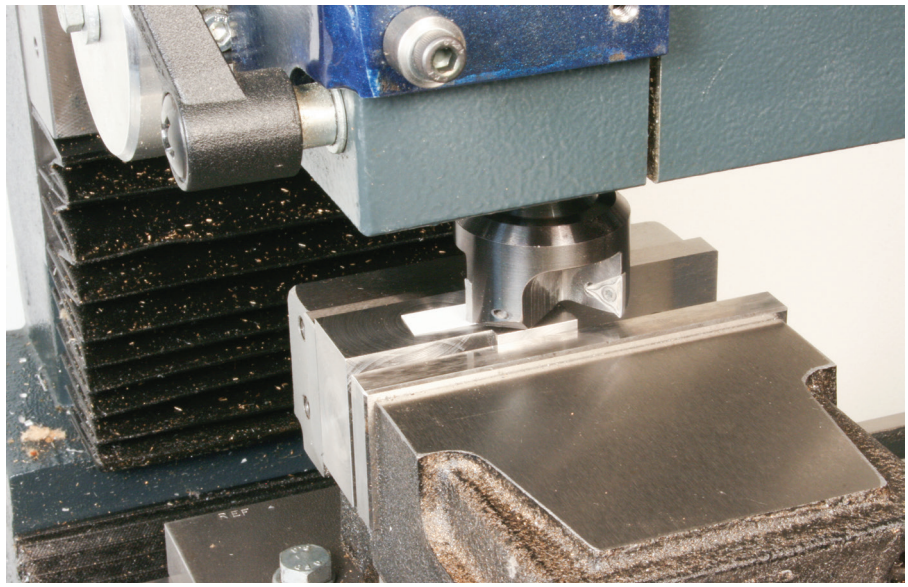


*Photo 57: Setting Z height using a piece of cigarette paper.*

## Setting the Z height

There are several ways in which we can set the Z height. For a job like this, and for many other jobs, it is convenient to set Z to 0 when the cutter is just touching the top surface of the work. Occasionally, it might be useful to set Z to 0 when the cutter is at the same level as the bottom surface of the work. It all depends on the job, and on your preferred style of working. In my experience, setting Z to 0 at the top surface is the most common situation, but, as usual, it's up to you.

A quick and reasonably accurate method is to use a piece of cigarette paper; a slightly slower, but more accurate method is to user a roller or a slip gauge, and a third method which can lead to automated height setting is to use a probe. We'll leave the probe until a little later, as it's worth taking some time to consider the detail of hooking up various types of probe.

### *Using a cigarette paper*

Lick a piece of cigarette paper and stick it on top of the work (photo 57). With the cutter mounted, jog down almost to the top surface and set the spindle turning slowly. Set the jog mode to Step, and set the step interval to, say, 0.01mm. Carefully jog down until the cutter just snags the paper, and set Z0 there. If you prefer, this can be done with the spindle switched off, just rotating the cutter gently by hand. If the spindle is stationary, you can use the paper dry, teasing it back and forth in small movements until it tears on the descending cutter or is trapped by the cutter.

### *Using a roller*

Place a roller of known diameter on top of the work (photo 58). Take the cutter just to one side and jog down until it is below the top of the roller. Set the jog mode to step and set the jog interval to, say, 0.01mm. Jog upwards until the roller will just pass under the cutter. The ideal situation is to have work, roller and tool all touching as the roller rolls between the work and the tip of the tool. The best technique actually pivots the roller at the cutter, with the lower part of the roller slipping through against the top surface of the workpiece (photo 59). Be gentle with a fine cutter though; and don't force the roller through at any point. If you are bothered by backlash in the Z axis, reverse the procedure, starting with the cutter above the roller, then gently jog down until the roller is only just able to swing under the tool tip. Set Z0 at that height.

*Photo 58: Uisng a roller of known diameter for height setting.*

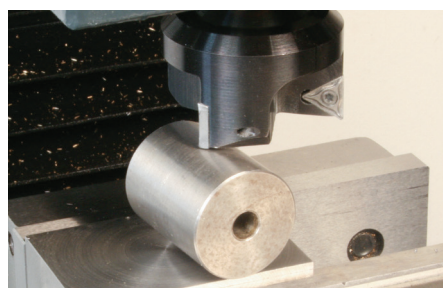


*Photo 59: Raise the cutter until the roller just passes underneath.*

This method is so frequently used that Mach3 has a setup for this. You can either:

click in the Z DRO and type the diameter of the roller then press Return; or

go to the Offsets screen. On the left, in the Gage Block Height section, where the diagram shows a block between the work and the cutter, enter the diameter of the roller, then click Set Z (fig 26). I normally use a 25mm diameter roller, so I would enter 25 in this section. When you go back to the Program Run screen, the Z DRO should read 25.000. Note that on the Offsets screen you should ignore the offsets shown in the DROs

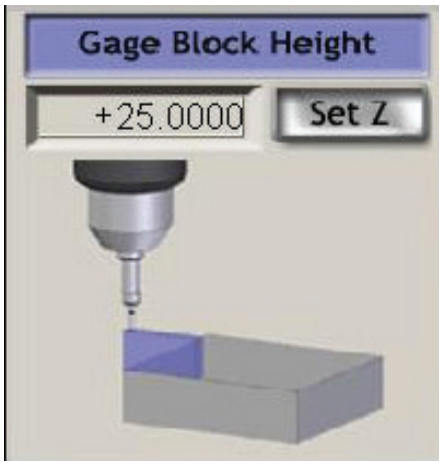



*Fig 26: Mach3 Z height set up for Touching Off..*

in the upper centre section of the screen. Mucking about with these, at this stage, leads to a whole world of confusion. Let's sort out the concepts involved with these offsets, separately, some other time (soon).

*Using a slip gauge*

Use the same technique as for a roller, but instead of jogging upwards until the roller can roll under the tool, jog upwards until a convenient size of slip gauge can just slide under the tool. Set Z to the thickness of the slip gauge using either of the two methods associated with the roller. Personally, I can cope more easily with scratches on a roller than with scuffs on my slip gauges. I prefer chicken to ham, too.

If you bring the cutter *down* onto the slip gauge, very very carefully, you will lose the sharp corners of an end mill, chip ceramic inserts and instantly destroy engraving cutters, as well as damaging your slip. I strongly advise against this.

**Finding the length of cut**

By jogging or using MDI commands, move the cutter above the work, and take it in the X direction to the right until it is clear of the right hand end of the work. Note the DRO reading, rounding it to some convenient and easily memorised figure. That's why you keep a pencil behind your right ear. The idea here is to know the X value for the starting and finishing positions as the cutter begins and ends each facing cut across the surface.

**Creating a program**

You remember the sequence for Teaching a set of commands (in part 3)? Creating a program is quite similar. Basically, we type the commands required for the sequence of moves we want the controlled point to make, then save it in a file. Once it is saved, it can be loaded into the computer at any time, then run, to make the machine carry out those moves. Technically, the program interprets a sequence of instructions, and turns those into appropriate signals for the stepper controllers.

One easy way to begin is to use File > Close to close any current sequence of instructions; then click Edit G-code. That should take you into the editor (probably Notepad) just as it did when you were editing the Teach sequence.

Type the instructions to take a cut across the surface. The exact distances will vary a little according to the start and finish positions of the cutter, and whether you need more than one pass to reach the whole width of the surface.

It's important that you sort out the method you will use, before diving into typing the commands. Here's a possible sequence for a one-pass cut (fig 27):

Set the feed rate
Set the spindle speed
Move in Z to Safe Z (a position well above the top surface and any clamps or obstructions).
Move to X0 Y0
Move down Z to create a depth of cut
Start the spindle
Cut along X until clear of the right hand end of the work
Move to Safe Z
Move back to X0
Stop the spindle

and the code for that might be:

F100
S300
G0 Z20
G0 X0 Y0
G0 Z-0.25
M3 (Start the spindle)
G1 X100
G0 Z20
G0 X0
M5 (Stop the spindle)

Now save your program (from within this editor). It's not quite finished, but it's a good idea to save it at this stage, in case we forget later, or make some other mistake.

File > Save As…

allows you to name the program and choose where to save it. Note that in the Save dialog window Save as Type defines the File Type as

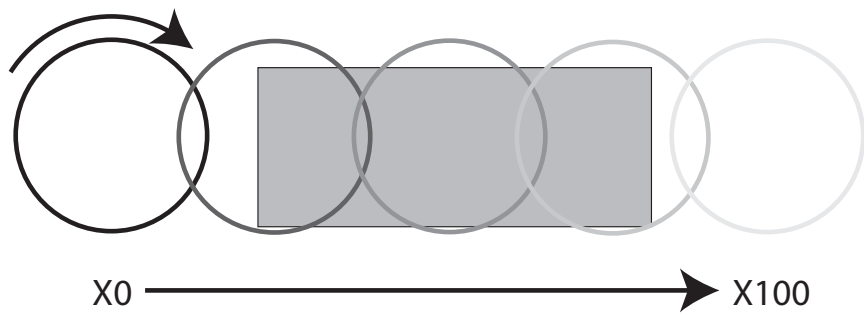



*Fig 27: One-pass cut.*

Text (*.txt where the * is the "wildcard" character which stands for "any characters" and in this context means "some unspecified name we will type in a minute".)
I keep all my Mach3 programs in one folder. Within that folder I have a mixture of single programs which do one job, and folders containing all the programs for a bigger job which is done in stages. I'm not a tidy person by nature, but I pass on my hearty recommendation to be disciplined in this respect. There are few things in this life as frustrating as knowing you have a program to do a particular job, but not being able to find it when you need it. It won't happen to you, of course.
You may wish to create a folder using the New Folder button, to hold all the examples for this series of articles. That way you will know, from month to month, where they are. Or use your own method.
The Save As… command only needs to be used once, when you initially create a program. After that, simply using File > Save saves the current version of the program over the top of the last one with that name, so you can quickly and frequently save updated versions as you go along. That's something I recommend. My own computer has never crashed in the middle of a session, but I live with the scars from the early days of computing, when crashes were more common than stable running, at times. Besides, our tumble drier is a cantankerous thing, prone to tripping the main electricity supply, and I don't have the luxury of an uninterruptible power supply for the workshop computer.
A saved program can be retrieved easily using File > Load… so if you want a break, just make sure the program is saved in its current version. When you restart the computer you can fire up Mach3 and Load the program once again, click Edit G-code and continue where you left off.
While making a program, you can switch in and out of the editor just as you did when Teaching a sequence. Edit G-code takes you into the editor, and File > Exit takes you back into Mach3. As you exit from the editor, the updated program will be taken into Mach3, *provided you have not changed the name of the program by using Save As.* This is an important proviso, and I recommend coming out of the editor using this sequence, to avoid tears:
File > Save (which saves the current version using the existing filename)
File > Exit

You might also wish to leave the previous version as it was when you last saved it, and create a new version (so that you can always go back to your previous version if your new modifications and additions don't work out as you intended). In that case, use:
File > Save As… and give the current version a new name.
File > Exit (which will take you back to Mach3 but retain the OLD version of the program, because Mach3 automatically loads the program which has the last filename it was using. The editor is outside of Mach3, so Mach3 will not have seen the details of the change to a new filename. A bit annoying, but we need to live with that.)
File > Load to load the newly saved version.
Once you have loaded a version using File > Load, the editor will expect to work with that version because it always works with the version currently in Mach3. Mach3 will continue to work with that version now that it knows its name.
If you go back from the editor to Mach3, and see the earlier version in the program window, just use File > Load to get the new version. And finally; it is always worth taking a moment to look at the program listing in Mach3 and check that it is the version you intended to work with. A quick glance at the Toolpath window is a good idea, too.

If you need to take additional cuts parallel to the first, to cut the rest of the surface, you will need to add one or more Y moves. With a 10mm diameter end mill, you may need 2 additional cuts:

Move in Z to Safe Z (a position well above the top surface and any clamps or obstructions).
Move to X0 then to Y0
Move down Z to create a depth of cut
Cut along X until clear of the right hand end of the work
Move to Safe Z
Move back to X0
Move in Y a little less than the cutter width
Move down Z to create the same depth of cut as last time
Cut along X until clear of the right hand end of the work
Move to Safe Z
Move back to X0
Move in Y a little less than the cutter width
Move down Z to create the same depth of cut as last time
Cut along X until clear of the right hand end of the work
Move to Safe Z
Move back to X0 then to Y0

Assuming a 10mm diameter cutter and a 70mm length of cut, the code for that might be:

```
F100
S300
M3

G0 Z20
G0 X0 Y0
G0 Z-0.25
G1 X70
G0 Z20
G0 X0

G0 Y9
G0 Z-0.25
G1 X70
G0 Z20
G0 X0

G0 Y18
```

```
G0 Z-0.25
G1 X70
G0 Z20
G0 X0

M5
```

If you want to add a second, deeper, pass, repeat the same basic sequence but with a deeper Z cut. Assuming a single pass 100mm long, add the following commands immediately before stopping the spindle (i.e. immediately before the M5 command):

```
G0 Z-0.5
G1 X100
G0 Z20
G0 X0
```

and to go deeper still, perhaps taking a finer finishing cut, add another set:

```
G0 Z-0.6
G1 X100
G0 Z20
G0 X0
```

If you are using a smaller diameter cutter, you will need to add the commands for the parallel cuts, at each Z height.

But wait; we're not quite finished.
If you look at the very top of the Program Run screen, towards the right hand end, you will see a set of commands marked Mill ->. These include some esoteric looking G code and M code commands. These apply automatically when Mach3 starts up. They can be modified to suit your own purposes, but they are there to ensure that Mach3 is set up to operate in a predictable, known, way when using MDI, jogging or running a Teach sequence. When creating a program we need to do exactly the same thing, by issuing a set of commands, from within the program, which set Mach3 into a known state so that it interprets the program instructions in a predictable way. That's an important concept, that a program should include all the commands necessary to ensure the behaviour of Mach3 is predictable when the program runs. Ideally, if I create a program on my computer, then give it to you, you should be able to run that program and get the same result, even though I don't know how your copy of Mach3 has been set up. In a perfect world, the program would then restore your settings before ending. That's actually quite difficult to do, and you can't depend on someone else's program doing that properly, which adds to the importance of having an initial set of commands which puts Mach3 into a known state. This is important when we begin using CAM software which generates a program for us.
Every program should have this initial set of commands, and we can call it an Initialisation Sequence, or Initialisation Block, for want of a better term.
For the moment, we can use the same set of codes listed at the top of the screen, provided they make Mach3 behave in an appropriate manner. We will discuss the various commands shortly, and you can make your own choice then, but in the meantime I would suggest something like this (but you can omit the comments in brackets):

```
G17 (Use the XY plane)
G21 (set the units to millimetres)
G40 (cancel cutter compensation)
G49 (cancel tool length offsets)
G54 (use co-ordinate system 1)
G80 (cancel canned cycle mode)
G90 (use absolute distances)
G91.1 (use incremental distances for arcs)
G92.1 (cancel co-ordinate system offsets)
G94 (set feed per minute mode)
G98 (set the canned cycle retract behaviour)
```

Type those commands at the start of the program, so that they sit above the instructions for moving and cutting the workpiece. Some of those commands set behaviours directly, and some cancel any unwanted behaviours which may, or may not, be in operation as a result of some previously used commands. Some commands are termed "modal", and those commands are "sticky" so their settings persist until changed, and while that is helpful in some circumstances, we do need to be sure we either set or cancel modal settings at the start of every program.
Interestingly, if Mach3 seems to be behaving in a strange way, you can use MDI mode to type that Initialisation Sequence. It often gets you out of trouble.

**Two more things**

At the very start, above even the Initialisation Sequence, type a comment or two to remind yourself what the program does, where Z is set to 0, and the cutter you expect to be used. These comments are for human eyes only, and we would want Mach3 to skip over them without trying to interpret them as commands, so the convention is to enclose them in round brackets ( and ). Any comments in brackets will be ignored by Mach3. Any blank lines are created simply by pressing the Return key, as in a word processor, and these can be used to create visual breaks, to aid readability.

Finally, add the command M30 on its own, as the very last line of the program. This signals the end of the program. There are two choices here; M2 and M30, and M30 is the best command for general use.

So a complete program for a large cutter making a single pass at each height might look something like this:

```
(Surfacing program)
(1st August 2013)
(Cleans up the side of the vice stop)

(Initialisation Sequence)
G17 G21 G40 G49 G54 G80
G90 G91.1 G92.1 G94 G98

(Main program)
```

```
F100
S300
M3

G0 Z20
G0 X0 Y0

(First cut)
G0 Z-0.25
G1 X100
G0 Z20
G0 X0
(Second cut)
G0 Z-0.5
G1 X100
G0 Z20
G0 X0
(Finishing cut)
F80 (Slow down the feed rate for a fine finishing cut)
G0 Z-0.6
G1 X100
G0 Z20
G0 X0
M5
M30
```

The support website at www.cncintheworkshop.com has this completed example for a single pass as well as an example of a parallel pass program.
That saves repeating too many instructions here, and also means you can check my solutions.

Before we move on, it's worth just looking in detail at the Toolpath window, because it displays some useful information. The view on the window can be rotated, tilted and panned, using the mouse. The view in this window is a bit small, though, so click the Toolpath button at the top of the Mach3 screen, to go to a larger view. Holding the left mouse button and moving the mouse will tilt the view, as shown in fig 28. Red lines are moves at rapid feed rate (G0 commands), and blue lines are moves at cutting feed rate (G1 commands). Think of the work being inside an imaginary sphere, and the pointer travelling across the surface of the sphere, with your view being from the pointer down into the centre of the sphere. A simpler concept is simply that up and down tilts Y, left to right tilts X. The mouse scroll button zooms in and out. Holding the right mouse button grabs hold of the image, and moving the mouse while holding the right button pans the image. You should experiment with this until you are happy that you can more or less control the angle of view. Tilting the view is very useful because it allows you to see Z moves more clearly and helps you spot unexpected moves before you run the program.

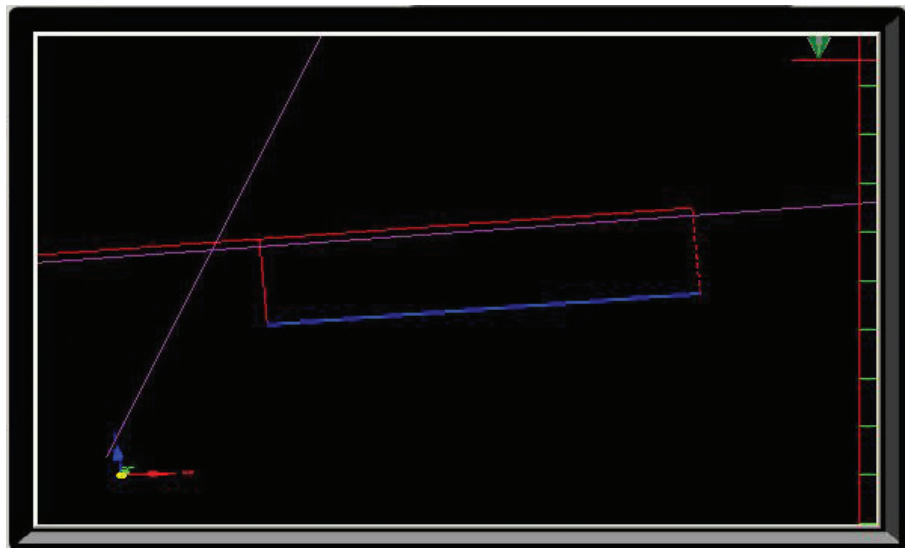


*Fig 28: Backplot window showing rapid feed moves (in red) and cutting speed moves (in blue).*

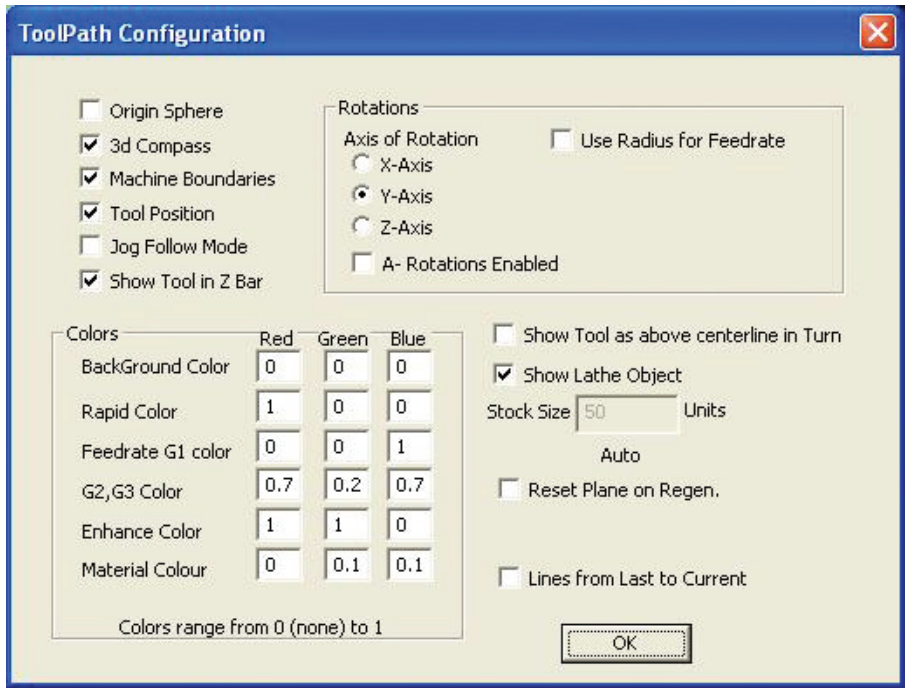



*Fig 29: Mach3 Toolpath Configuration window.*

For this program, you will need to zoom in close to distinguish the three individual blue lines, because the Z values are close to each other.
The contents of the tool path display can be customised via the Config > ToolPath menu (fig 29) and settings affect both the Tool Path screen and the tool path window on the Program Run screen.

**Origin Sphere:** creates a bounding box to indicate the size of the toolpath in X, Y and Z directions. I normally leave it off because I find it distracting.

**3D compass:** Removes or displays the YXZ axis indicator in the bottom left corner of the window. I usually have that on, but don't often make use of the information it provides.

**Machine Boundaries:** The limits on machine movement (the 'soft limits') are set using the Config>Home/Soft Limits menu (fig 30). Opting the show the Machine Boundaries on the Tool Path window means the image will be shrunk to fit the whole of the soft limits in the window. If, for example, your toolpath movements are from 0 to 100 in X, but your X soft limits are set to -500 and +500, the window will show a total X distance of 1000 units, so your 100 unit tool movements will look very small indeed. You can zoom in, of course, but you will be there for ages. I find this annoying so I leave this setting off.


*Fig 30: Mach3 Home and Soft Limits window.*

**Tool Position:** The axis indicator lines will move to follow the cut, or not. I usually leave this on, becasue I like to see where Mach3 thinks the cut should be. Unlike Jog Follow mode, this setting keeps the view of the workpiece stationary in the window and moves the axis indicator lines, rather than the other way around.

**Jog Follow Mode:** Jogging moves the view of the work so that the axis indicator lines remain centred in the window. I find this confusing; so I leave it off.

**Show Tool in Z Bar:** This shows the movement of the CP along the X axis, as a descending/ascending arrowhead moving against a scale on the right of the window. Somewhat useful.