

Computer Numerical Control Woodworking and Turning

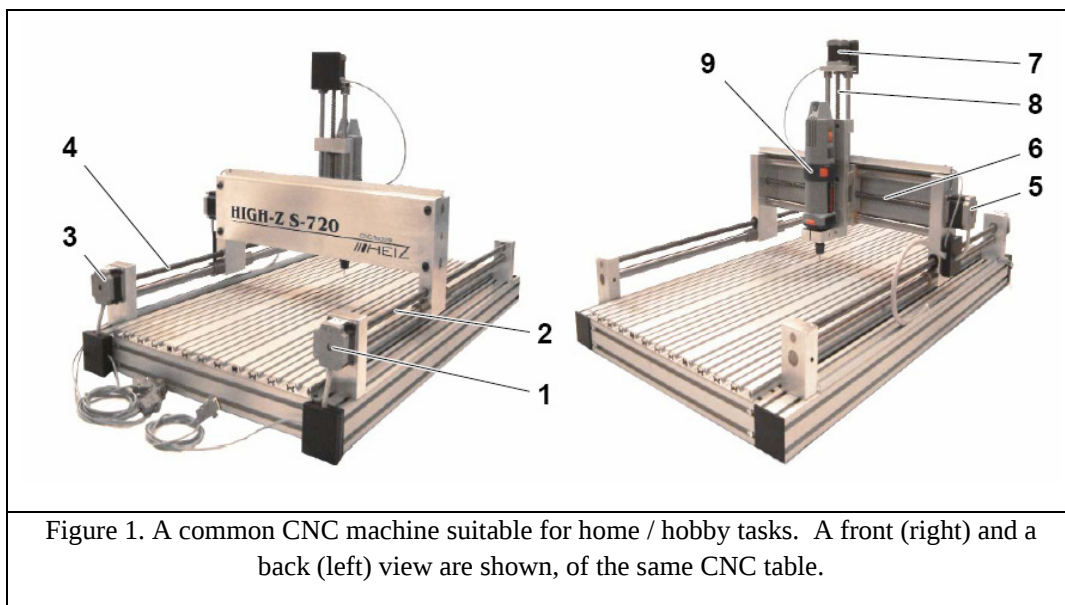
Gershon Elber

Introduction

For several decades, the use of computers for precise machining of metal parts has been a common practice in all the car, ship and aeronautic industries. In recent years, this computer numerical control (CNC) technology was also introduced into commercial wood working shops. This article examines the use of CNC technology for home – hobby purposes, mostly in the context of wood working and turning.

For the last few years, we have experienced a new flourishing wave of low end CNC devices. To first appreciate the benefits expected from using CNC in wood working, one must understand what CNC is all about. Consider a regular router hooked to an arm that can move in the three different directions of the main axes (X, Y, and Z). By making the motion control (amount, speed, acceleration, etc.) of the arm governed by a computer, one can move this router to virtually any location in the working space, and at any desired speed and acceleration. This computer controlled devices were used to be quite expensive but nowadays one can find a variety of products starting from a few thousand dollars. Moreover, the possibility of one building his/her own XYZ table is now better than ever (I will remark on this point later).

Figure 1 shows one such CNC example. This XYZ table controls the motion of the router (9 in Figure 1) using *step motors* (1, 3, 5, and 7 in Figure 1), or motors that could be commanded to rotate specific small angles. These motors operate *leading screws* (2, 4, 6, 8) that lead the arms. Here, motors 1 and 3 work in unison and control the X axis, moving the entire bridge above the table. Similarly, motor 5 controls the Y axis, moving the carrier of the router along the bridge and finally, motor 7 moves the router up and down, along the Z axis.

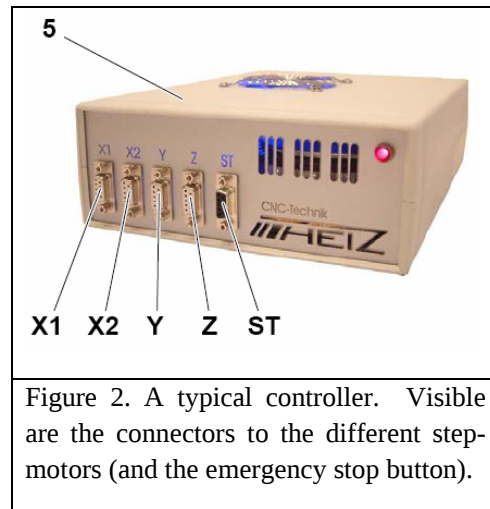


With this hardware, one needs a *controller* to command the step motors. The controller's duty is to provide a high level interface for motion-control. As an example if the router is now positioned at some location (x1, y1, z1) and we seek to move it to a new location (x2, y2, z2) in a maximal speed of V and maximum acceleration (deceleration) A, the controller will process these high level requests and will convert them to commands to all the motors. The controller will convert all these high levels requests to electronic pulses that are applied to the four motors in the table, concurrently. The simultaneous operation of all the motors is essential or otherwise no diagonal motion could be achieved, if you stop and think about it for a second. High end CNC devices are typically far more sophisticated, allowing control over features such as the router's (or spindle's) RPM, coolant, tool changers, etc. These features are of less value at the lower end market of CNC and are rarely available. On the other hand, the low end market of CNC devices still offers an excellent accuracy for the (hobby) wood working industry and accuracy and repeatability below 0.01mm is quite common.

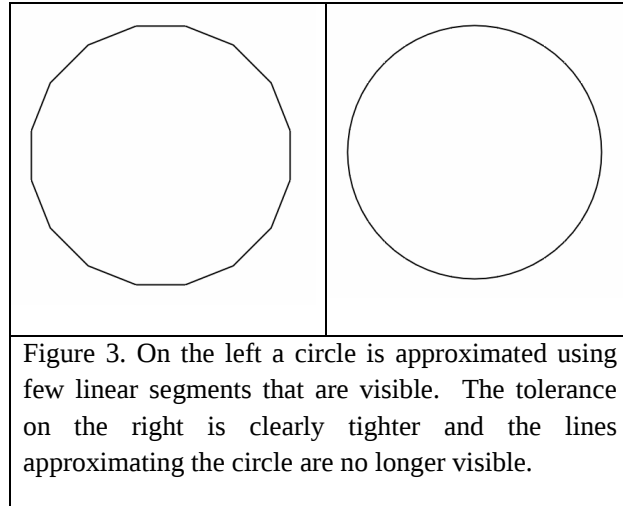
Getting the Ingredients Right

In order to operate a CNC device on your own you will need to:

1. Decide on the **hardware** you will use, either buy an XYZ table or build one. As I mention above, good low end CNC for home / hobby use can cost less than a thousand (US) dollars, building everything from scrap materials, and up to several thousand dollars for a pretty nice machine. In order to build an XYZ table of your own, you will need some mechanical engineering knowledge to handle the entire frame, support, and the motion mechanisms. The later includes decisions such as leading screws vs. ball screws, or step motors vs. servo motors.
2. Decide on the **controller** to use. In many cases, the controller will come with the XYZ table, if you purchased one (See Figure 2). Here again, you can build a controller on your own and you can find instructions and kits for CNC controllers on the WWW, supporting several motors. Here, if you like to build such a controller, the aid of some electronics expert might be desired.



3. Decide on the **software** to use. While you, in principle, can command your XYZ table via the controller to move the router to any location in the working space, try to draw a circle. Controllers can, typically, handle only linear motion (it is difficult enough, if you consider what is going on inside). To move the router along a circle, you need software that will break the circular motion into many small linear motions. Figure 3 shows you a circle approximated using linear motions but in two different tolerances. Using the second, tighter, tolerance, the linear motions are so small that the circle looks perfect.



In other words, you need to decide on the *solid modeling* or *geometric modeling* package you will use to create the geometry – you would like, probably, to design circles and other higher level shapes and let the solid modeling software convert the result to linear motion for the controller. You might want to consult some engineer or a software expert in the area of geometric modeling before you make your mind here.

In summary, the CNC device has three major ingredients. How do these components communicate? The controller is the bridge between the XYZ table hardware and the geometric modeling software on the computer. On the one hand, it is hard-wired to the motors. On the other, it is connected to a personal computer, typically via an RS232 or a USB port. Your geometric modeling software should provide support for CNC machining and further be compatible with the controller. The software should be configured to access the port that is used by the controller when commands to operate the CNC are to be executed. This specific configuration depends on the type of geometric modeling software you have and some CNC manufacturers with also provide you with software and/or drivers to activate the controller and interface it.

Commanding the Controller

By now, you have all the necessary ingredients for CNC machining in place and we are almost ready to cut some chips. Before we do that, we better understand the protocol in which the computer (i.e. your geometric modeler) talks to the controller. When you asked your software to draw (and then cut) a circle, your solid modeling software will devise a sequence of many linear motions that approximates the circle. These linear motions are transmitted to the controller, via the designated port, in an agreed upon language of motion commands. In many cases, you will want to have some control over this stage and hence have some understanding of the language. In fact, for simple examples, you might be able to write or edit the file containing the motion commands.

Luckily, there are several common standards in this motion language. The most well known one uses G-codes (as the prefix of the language commands is 'G', while there are also some 'M' commands:-), a simple language that is found in all high end CNC devices, and many low-end ones.

Every command in the G-codes language is prescribing one motion. For instance G0 commands a linear rapid (non cutting speed) motion whereas G1 commands a linear cutting (speed) motion. These G commands will be followed by XYZ coordinates and other (optional) parameters such as federate (designated with F). Other commands control state. For example, G90 states that the coordinates are absolute, and G71 that units are in millimeters. Figure 4 shows a simple G-codes program to cut a square. All the lines start with N00xx line numbering prefix which is ignored by the controller and is for convenience only. Note Z is specified once only and by convention, if missing, the previous state (position, federate, etc.) is kept. The M30 command at the end terminates the program.

```
n0010 g90
n0020 g71 t1 m6
n0030 g1 x-1.000 y-1.000 z0.000 f10
n0040 g1 x-1.000 y1.000
n0050 g1 x1.000 y1.000
n0060 g1 x1.000 y-1.000
n0070 g1 x-1.000 y-1.000
n0990 m30
```

Figure 4. A simple G-Codes program to cut a square.

Another common motion language is HPGL (HP Graphics Language) that is an adaptation from a plotters' motion language invented by HP some decades ago. Other low end CNC devices offer propriety motion languages.

Cutting my Geometry

Use CNC machining deserves some word of warning. Needless to say, the router can be quite dangerous. Sending the router to wrong places, due to buggy motion commands can have grave consequences. You modeled a new part in your favorite geometric modeling package, and created the G-code motion program to cut the part. It is highly recommend to verify your motion program before you apply it to real wood. You can execute the program 'in the air' and inspect it, or, even better, cut light material like foam or Balsa wood first.

A more sophisticated verification alternative would be to simulate the G-code motion file using some CNC simulators. These simulators read in G-code and show the motion and the material off the stock, step by step. Figure 5 shows my own NCSim, a free (for non-commercial purposes) CNC simulator that I use.

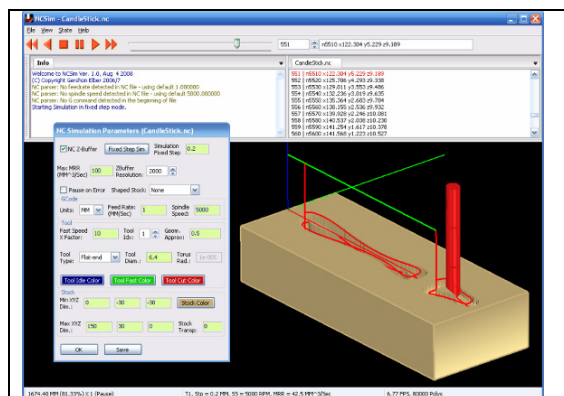


Figure 5. G-Code simulation on a computer, using the NCSim software package (<http://www.cs.technion.ac.il/~gershon/NCSim>). In red, the cutting motion is seen while in green the rapid motion is displayed. See also Figure 6.

In this section, I present a few examples of my work, of wood turning combined with CNC machining. The G-Code motion files in all these examples were created using our own IRIT solid modeling package (see <http://www.cs.technion.ac.il/~irit> and <http://www.cs.technion.ac.il/~gershon/GuIrit>) that is free for non-commercial users but is not easy to use as it is a research oriented modeler. The created G-code was *simulated* before being used for actual cutting using NCSim (<http://www.cs.technion.ac.il/~gershon/NCSim>).

Figures 6 to 9 present four woodworking pieces that combined turning and CNC, in their creation. All figures show different stages in the construction of the piece, on the CNC XYZ table and on the lathe. Figure 6 shows a rotated candlestick with hollowed shapes that were carved using CNC. Figure 5 shows the CNC simulation of the hollowed forms, in NCSim.

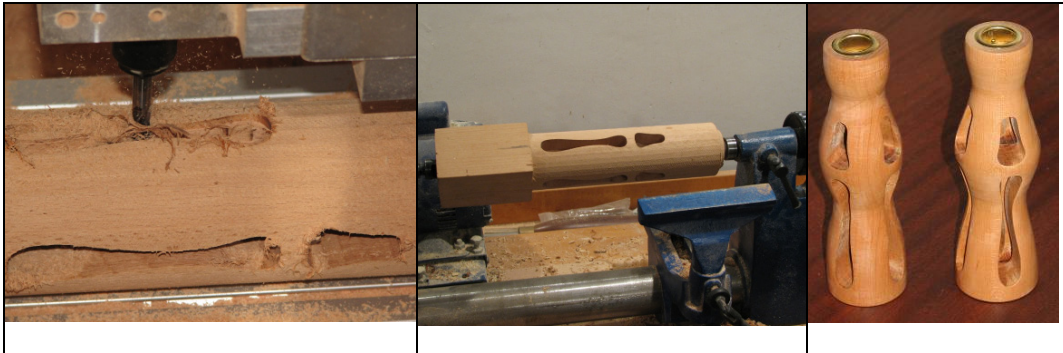


Figure 6. A candlestick with hollowed forms that were CNC machined out. On the left the initial rotated stock is machined on the XYZ table. The middle figure shows the candlestick after the CNC process was complete, back on the lathe. Finally, the image on the right shows (a pair) of final pieces. Figure 5 shows the simulation of this CNC machining process.

Figure 7 shows a vase with a wiggly neck and a square cross section. Figure 8 shows another variant of a vase, with an α -like shaped neck.



Figure 7. A vase with a neck that was CNC machined. On the left, the initial rotated stock is machined on the XYZ table. The middle figure shows the part after the CNC process was complete, back on the lathe. Finally the image on the right shows the final piece.



Figure 8. Another vase with a neck that was CNC machined. The left and middle images show the CNC steps on the XYZ table. Finally, the image on the right shows the final piece.

The final example is shown in Figure 9. Presented is a bowl with 8 ornamental cuts made around it, in the shape of two faces looking at each other (or a vase if you envision the removed material). Similarly, 8 rounded cuts are also introduced to the cover of the bowl.

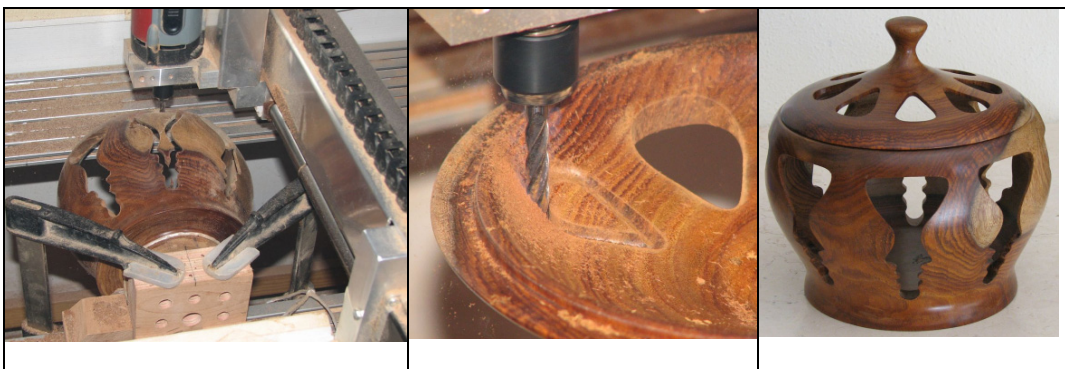


Figure 9. A bowl with ornamental cuts on the CNC table (left). Note the fixturing that is seen on the left side, which exploits the face plate fixture of the lathe. The cover is shown undergoing the CNC cuts in the middle and the final piece is shown on the right.

Some Additional food for thoughts

There are many directions one can go from here. One crucial component in any CNC machining work is fixturing, or properly restraining the stock while cutting it. In some cases, a good fixture requires human ingenuity intervention and is certainly beyond this article. In most cases, a simple fixture would do. The point is, when you design a new part you wish to make, do not forget to design the proper fixture for it.

Not everything could be made using an XYZ table. Having 3-axes of motions, such XYZ cutting process is also called *3-axis machining*. Consider an attempt to cut the inside of a vase with negative slopes or machine a surface of revolution. Such operations cannot be conducted using 3-axis machining in a single fixture. More degrees of freedom are needed to rotate the surface of revolution object. The figure on the right shows a 4th-axes add-on to an existing XYZ table that allows cutting rounded objects in one fixture.

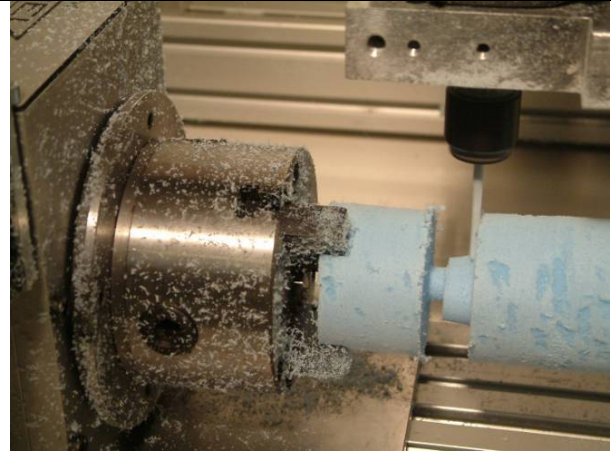


Figure 10. A 4th, rotated, axes for a CNC XYZ table.

Some more advanced controllers will also allow more complex motion, than just a linear (G0/G1) motion. Many high end CNC controllers also support circular (G2/G3 commands for clockwise and counter-clockwise motion) or even helical and general (spline) motion. These features are less likely to show up in low end CNC.

Happy CNC!

Additional WWW links:

http://lirtex.com/heb/index.php/cnc (In Hebrew)	http://www.imsrv.com/
http://www.cnczone.com/	http://www.machsupport.com/
http://www.cnc-step.com	http://www.majosoft.com/
http://www.minitechcnc.com/	http://www.shopbottools.com/
http://www.komo.com/	http://www.step-four.at/e_index.html
http://www.cnc-tables.com/	http://www.techno-isel.com/
http://www.digirout.com/	http://www.cncmasters.com/
http://www.editecnc.com/	http://www.thermwood.com/
http://www.dynacnc.com/	http://www.vectorart3d.com/
http://www.flashcutcnc.com/	http://www.xylotex.com/
http://www.hobbycnc.com/	http://www.carviewright.com/

Keywords for further WWW search: Computer Numerical Control, CNC, CNC controller, step motors, servo motors, G-code, HPGL, solid modeling, geometric modeling

Acknowledgement: Some of the images in the articles are used with permission from cnc-step, <http://www.cnc-step.com>