Creative Learning of Analytic Geometry through NC Programming with a Virtual Lab Application

Pongrapee Kaewsaiha
p_kaewsaiha@hotmail.com
International College
Suan Sunandha Rajabhat University
Thailand

Abstract: *This paper presents the use of NC programming as a part of learning activities in analytic geometry class, with the use of a developed virtual lab application. The aim of the designed learning activity is to enhance the creativity of students in learning analytic geometry by visualizing the relationship between course materials and a real application in manufacturing industry.*

1. Introduction

Studying analytic geometry in ordinary ways involves with math formulas, coordinates, positioning, and graphic drawing. The usefulness of course contents are often asked by students since most textbooks do not provide enough real applications, or provide but no practical demonstration.

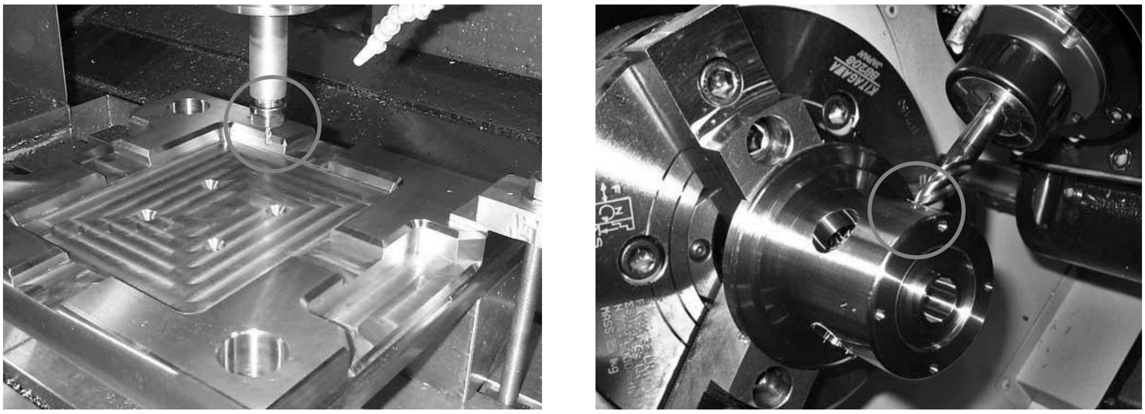
In manufacturing industry, NC machines are being used in many production lines mainly in automobile parts and furniture production. NC machines require input in the form of 2D or 3D coordinates with path positioning and some input parameters. An NC programming has become a required skill of mechanical engineers, robotic engineers, CAD-CAM engineers, manufacturing engineers, and many related fields. This technology makes many conceptual product designs become real, as seen in today’s market.

Basic NC programming is not too hard for high school students to learn since it has to do with only basic geometric shapes line lines or circular arcs. Unfortunately, an NC machine which is required to demonstrate this application is too expensive and may not be affordable for many schools. Even an NC simulator software, which is normally used by an engineer to verify the correctness of the code, is considered expensive. Some models of simulator software cost as much as real machines.

However, using NC programming as a part of learning activity in analytic geometry class may not need a real NC machine or a high-end simulator software. A simple application which can check the correctness of NC programs is enough to make students understand the concept. The designed learning activity and the use of a developed virtual lab application are discussed in this paper.

2. NC Machines and NC Programming

NC machines, often called CNC (Computerized Numerical Control) machines, being used in a manufacturing process can be categorized in many types. The type of the machine is specified by its moving axes and the attached tool. A milling machine (see Figure 2.1a) has a rotating sharp toolhead (specified by a red circle) which can shear out raw materials into the desired shape. A milling



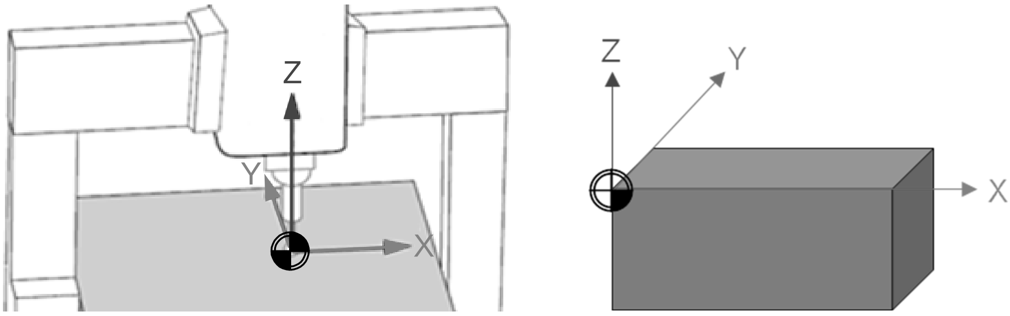
a. b.

Figure 2.1 (a) CNC Milling, (b) CNC Lathe, Red circles indicate machine toolheads

machine usually has 3 or 4 moving axes controlled by a series of servo motors. A lathe machine has one rotating axis and is used to create cylindrical-shaped workpieces, see Figure 2.1b. A robot arm is also a kind of NC machines but its control is too complex and should not be included in any fundamental analytic geometry class.

2.1 NC Coordinate Systems

The concept of NC machine control is to specify the target coordinate where we need to move a toolhead to. Figure 2.2 shows the coordinate system being used in NC programming which is related to a fundamental 3D rectangular coordinate system where there are X, Y, and Z axes. In **machine coordinate** the origin is fixed according to the machine installation while in **workpiece coordinate** the position of the origin is adjustable according to the dimension of input material. Each NC machine has a default coordinate system which can be changed by a corresponding code in programming process. A "**<u>workpiece</u>**" coordinate is used for the entirety of this paper.



a. b.

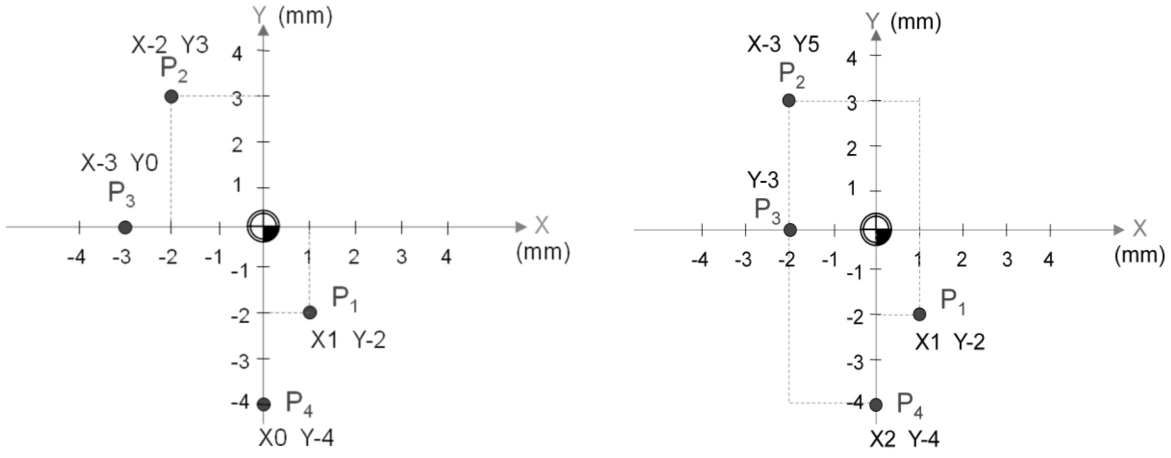
Figure 2.2 NC Coordinate Systems (a) machine coordinate (b) workpiece coordinate

2.2 NC Measurement Units

Two units are being used in NC programming, **millimeter** and **inch**. Each NC machine has a default measurement unit which can be changed by a corresponding code during a programming process. A "**<u>mm</u>**" unit is used for the entirety of this paper.

2.3 NC Positioning Systems

There are two NC positioning systems, an "**absolute**" positioning and an "**incremental**" positioning. In an absolute positioning system, points are measured from the origin, see Figure 2.3a. In an incremental positioning system, each point is measured from its preceding point, see Figure 2.3b. Each NC machine has a default positioning system which can be changed by a corresponding code during a programming process. An "**<u>absolute</u>**" positioning system is used for the entirety of this paper.



a. b.

Figure 2.3 NC Positioning Systems (a) absolute positioning (b) incremental positioning

2.4 NC Programming

Any NC machine requires an input program to work, which is often called a "G-Code". This kind of user-defined program specifies a moving path of a toolhead eg. straight line, circular-arc, or cycle eg. drilling, or threading. The following codes are commonly used in NC programming:

- **G00** is a code for "**Rapid Positioning**". Executing this code a toolhead will move from its current position to the target coordinate at maximum speed. There must be a clear path between two points or a collision may occur. The following code shows how to move a toolhead to the coordinate (40,15,2). The illustration is shown in Figure 2.4.

G00 X40 Y15 Z2

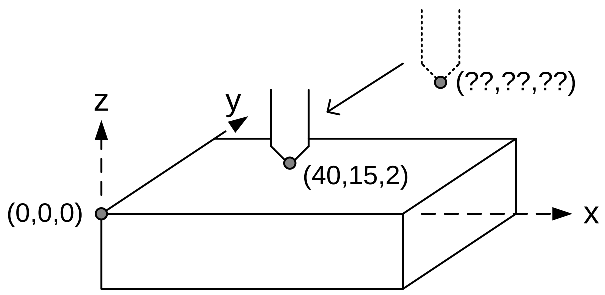


Figure 2.4 Execution of G00 command

Note: Coordinates can be decimals eg. X1.2965 Y-2.4837. The accuracy is different each machine.

- **G01** is the code for "**Linear Interpolation**". Executing this code a toolhead will move from its current position to the target coordinate in a straight line while shearing out material along the way at controllable speed (called a "feed rate"). The following code shows how to perform a straight-line cut from the current position at (5,5,-2) to the target coordinate (20, 15,-2) with a feed rate of 100 mm/min. The illustration is shown in Figure 2.5.

G01 X20 Y15 Z-2 F100

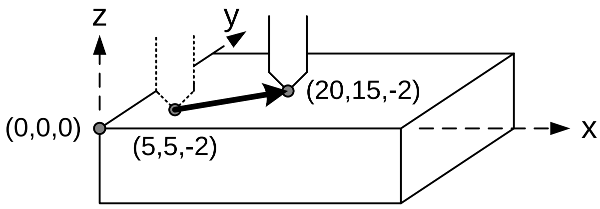


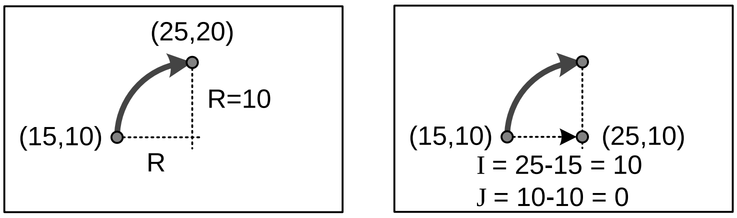
Figure 2.5 Execution of G01 command

- **G02** is a code for "**Circular Interpolation, Clockwise**". Executing this code a toolhead will move from its current position to the target coordinate in a circular arc of specified radius and feed rate. The following code shows how to perform a circular-arc cut (CW) from the current position at (15,10,-2) to the target coordinate (25,20,-2) with a radius of 10 mm and a feed rate of 75 mm/min. The illustration is shown in Figure 2.6a.

G02 X25 Y20 R10 F75

Another way to specify a radius of the arc is to specify a "relative distance" from the arc starting point to the center of the arc using "I" as a relative X, and "J" as a relative Y as follows, see Figure 2.6b.

G02 X25 Y20 I10 J0 F75



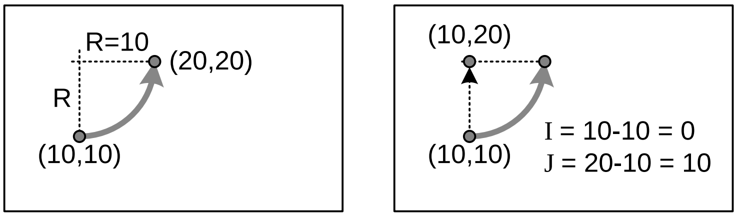
a. b.

Figure 2.6 Execution of G02 command (a) with radius parameter (b) with center-of-the-arc relative distance parameters

- **G03** is a code for "**Circular Interpolation, Counter-Clockwise**". This code works mostly like G02 but with Counter-Clockwise movement instead of Clockwise. The following codes show how to perform a circular-arc cut (CCW) using the same parameters as the previous example. The illustrations are shown in Figure 2.7.

G03 X25 Y20 R10 F75

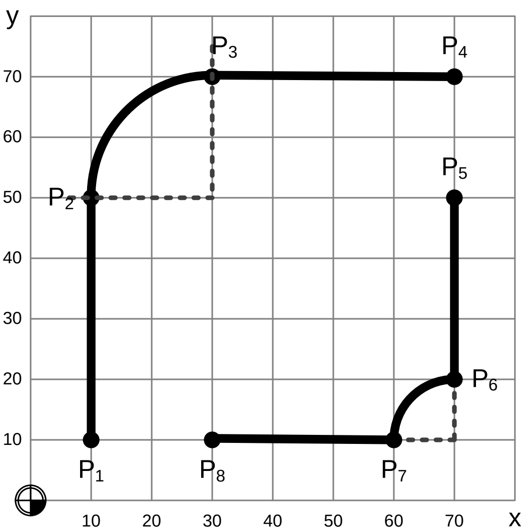
or G03 X25 Y20 I0 J10 F75



a. b.

Figure 2.7 Execution of G03 command (a) with radius parameter (b) with center-of-the-arc relative distance parameters

There are more commands in G-Code such as G28-Return to home position, G83-Peck drilling cycle, M05-Spindle stop, but only four basic commands mentioned before (G00-G03) are included in the designed classroom activity. The following example demonstrates the fundamental concept of NC programming. In this example a piece of square plastic is to be engraved like in Figure 2.8a using a CNC milling machine.




```
G00  X10  Y10  Z2
G01  Z-1  F100
G01  X10  Y50  F100
G02  X30  Y70  R20  F100
G01  X70  Y70  F100
G01  Z2  F100
G00  X70  Y50
G01  Z-1  F100
G01  X70  Y20  F100
G03  X60  Y10  R10  F100
G01  X30  Y10  F100
G01  Z2  F100
G00  Z50
```

```
G00  X10  Y10  Z2
G01  Z-1  F100
Y50
G02  X30  Y70  R20
G01  X70
Z2
G00  Y50
G01  Z-1
Y20
G03  X60  Y10  R10
G01  X30
Z2
G00  Z50
```

a. b. c.

Figure 2.8 (a) Tool paths (b) Semi-Full Program (c) Shorten Program

Figure 2.8b is a semi-full program which includes approaching, drilling, and retracting processes but omits all machine-control codes such as Spindle-ON or Spindle-Stop. Executing the first line a toolhead will "approach" to P_1 and wait at 2mm distance above the material surface. On the next line a toolhead will "drill" down 2mm from the surface, then head to P_2, P_3, and P_4 following the specified paths. On the 6^{th} line a toolhead is retracted to the safe distance at 2mm above the surface before approaching to the next point at P_5. The process end at point P_8 by retracting a toolhead along the way up to 50mm above the surface, allows a worker to securely stop the machine and remove a finished workpiece from its clamp.

The program shown in Figure 2.8c was shorted by removing duplicated parameters. For example, when only Z coordinate appears on the 2^{nd} line, it means the program will maintain the value of X and Y at X10 and Y10 as previously defined. Only Y50 appears on the 3^{rd} line means the program will maintain G01, X10, Z-1, and F100 as previously defined. Shortening the code is a challenge that makes student understand the concept of movement in one or two dimensions.

3. NC Virtual Lab Application

An NC virtual lab is an application software developed using JAVA. The software can generate and graphically display the result of a user-defined G-Code program. The concept of this virtual lab is from a CNC simulator software which is used to verify the result of G-Code programs before a real machine run. But the virtual lab was scaled down to meet the level of students and the designed learning activity described in the next section. The developed software has a file size less than 10kB and can graphically display toolpaths in 2D. To use the software just save a G-Code program in the same folder as an executable file, then run an application to check the result. Figure 3.1 shows a result display window of a virtual lab software when using codes in the previous example as an input.

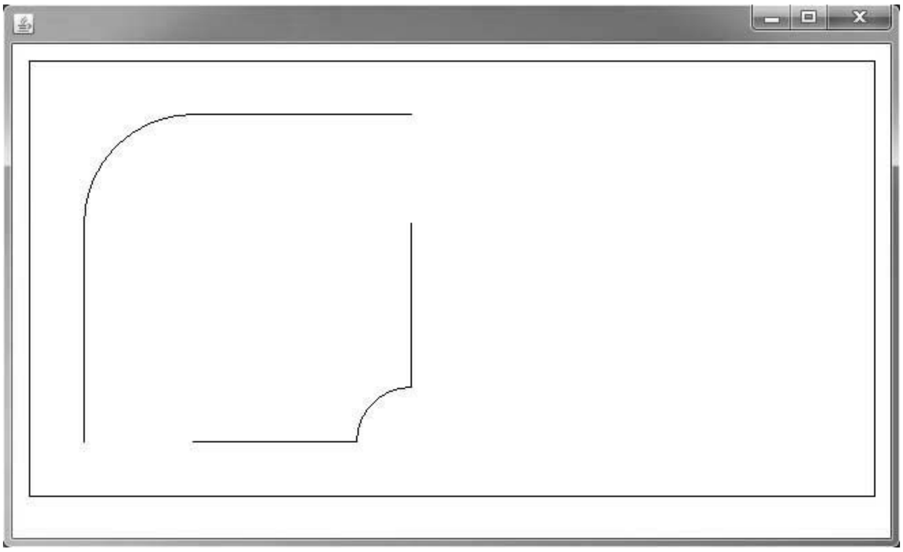


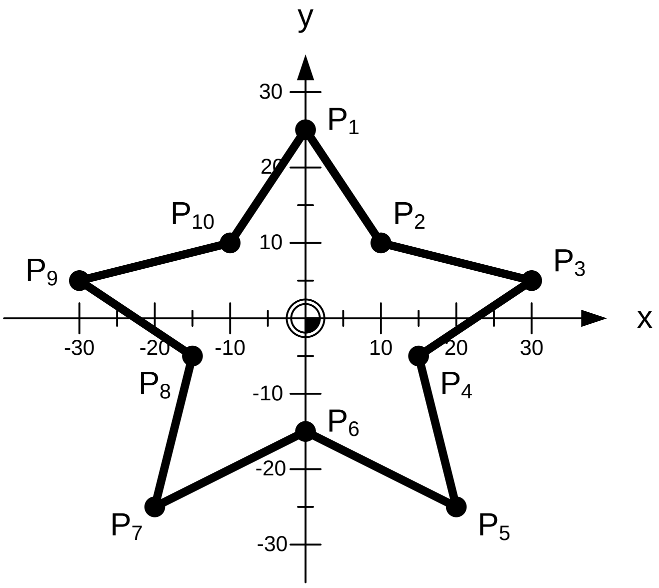
Figure 3.1 A result display window of a virtual lab application

4. Learning Activity

A learning activity described in this section has been designed to introduce students to NC technology and NC programming. The main purpose of the designed learning activity is to point out a relationship between course materials in analytic geometry and a real application. The following steps have been applied:

Step 1: Show some video clips about CNC machine run. Students will pay a lot of attention to the videos because it is their first time to see an industrial machine run at high speed, create a product in a very short time and precisely. They will realize how products they use in dayly life are created.

Step 2: Introduce student to NC coordinate systems and the way a toolhead moves around the grid. At this point students will realize that the productions of many household products are based on the same fundamental analytic geometry. Let students attempt a quiz about NC coordinates, see Figure 4.1.



a.

	Absolute		Incremental	
P1	X0	Y25	X0	Y25
P2	X10	Y10	X10	Y-15
P3	X30	Y5	X20	Y-5
P4	X15	Y-5	X-15	Y-10
P5	X20	Y-25	X5	Y-20
P6	X0	Y-15	X-20	Y10
P7	X-20	Y-25	X-20	Y-10
P8	X-15	Y-5	X5	Y20
P9	X-30	Y5	X-15	Y10
P10	X-10	Y10	X20	Y5

b.

Figure 4.1 A quiz for basic NC coordinates (a) work paths (b) quiz table with solutions

Step 3: Teach students about basic NC programming, G-Code commands (G00-G03), then give some examples like in Figure 2.8. Then let students attempt a quiz about NC programming as in Figure 4.2. Students can use a virtual lab application to check their created programs. Notice that the answer is not unique. After this, let students discuss how to find which code has the following properties:

- Shortest overall moving distance
- Shortest lines of code
- Less approach/retract steps

According to work paths in Figure 4.2, the best answer should be a program that starts an approaching at the nearest point to the origin and finishes a process in a single approach/retract step.

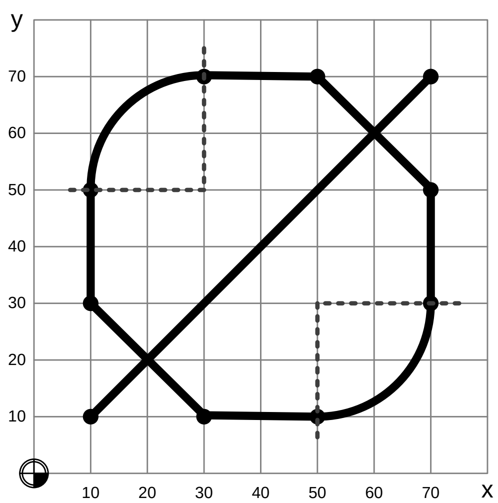


Figure 4.2 A quiz for basic NC programming

Step 4: Let students design their own workpieces on given grid papers, see Figure 4.3. Every design must be verified by an instructor to check that it is possible to program using basic G-Codes. At this point students may have many interesting ideas but some designs may be too complex to program manually. Some students may have a misunderstanding to geometric properties, especially arcs, which are needed to be clearified by an instructor. Some passed designs are shown in Figure 4.4. After the designs have passed the verification, then students have to create a G-Code program according to their designs. Students can use a virtual lab application to check their created programs.

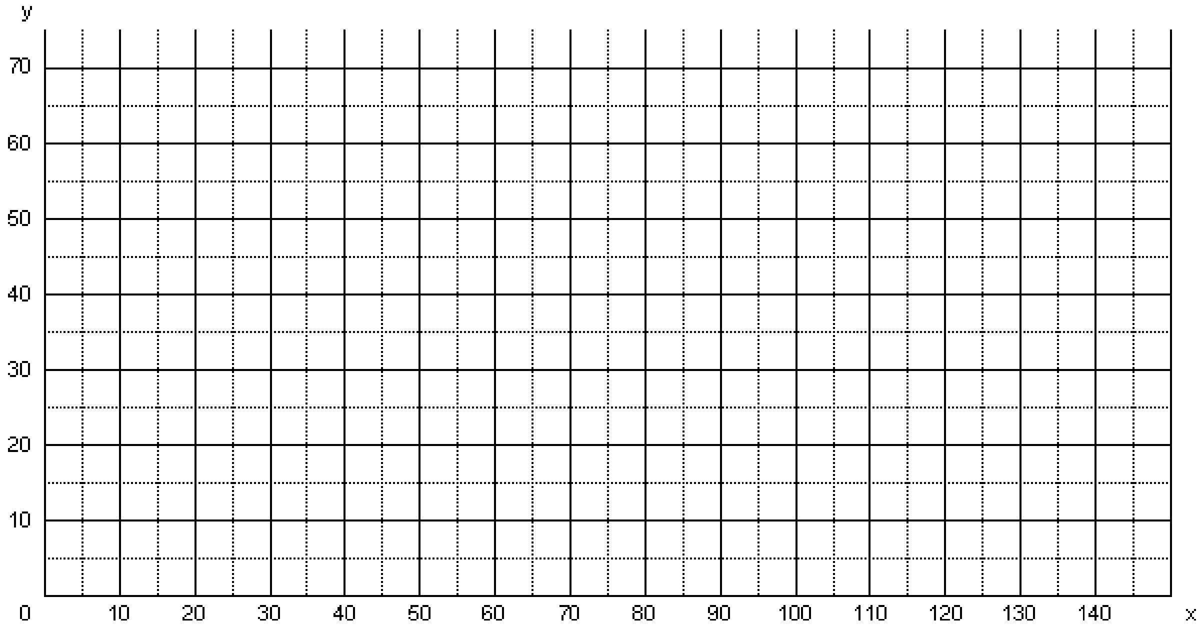


Figure 4.3 A grid paper for workpiece design

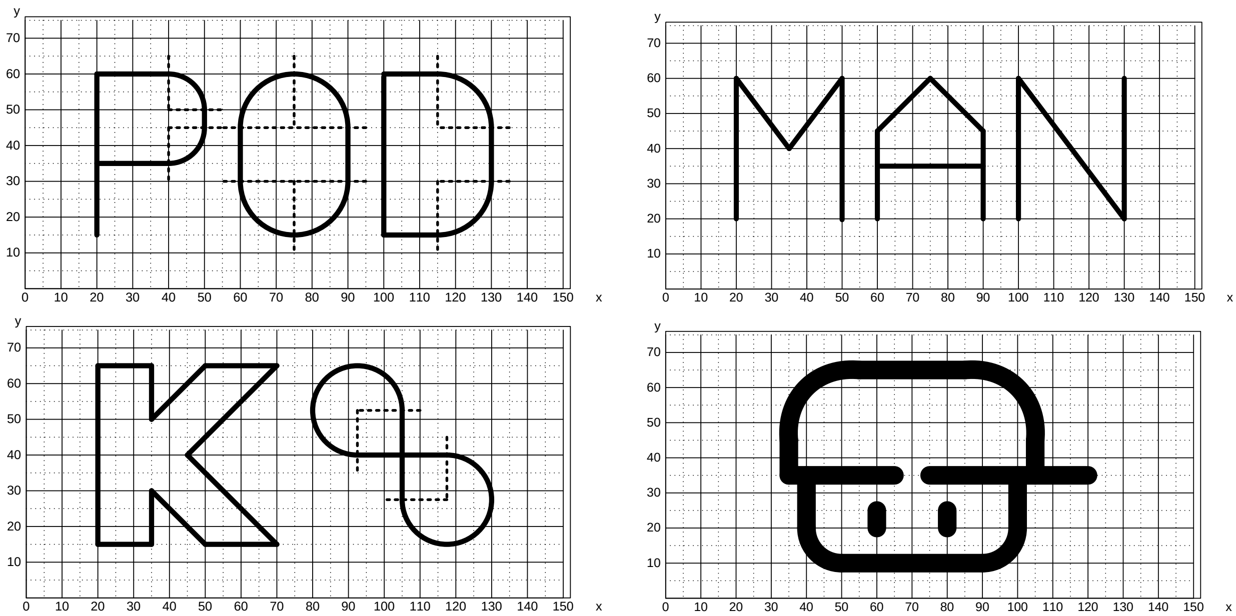


Figure 4.4 Some verified designs

After this step, an instructor has the option to continue to the next step if there are CNC simulator software or real CNC machines to run.

Step 5: Modify the finished codes into machine-ready programs and test-run with a CNC simulator software to verify the correctness of a machine run. The instructor may use experienced students to assist new students in this step because machine control and interface are complex and require several hours to train. Figure 4.5 shows the result of a G-Code program generated by a simulator software.

Figure 4.5 A result from a simulator software

Step 6: Let students operate a real CNC milling machine under the strict control of the instructor and assistance from trained students. A brief of CNC operating and safety policies must be given before the operation. The raw material used in this activity is a clear acrylic plastic, 75×150×5 mm, which has the same size as a given grid paper and a virtual lab display window. Figure 4.6 shows an example of a student's finished workpiece.

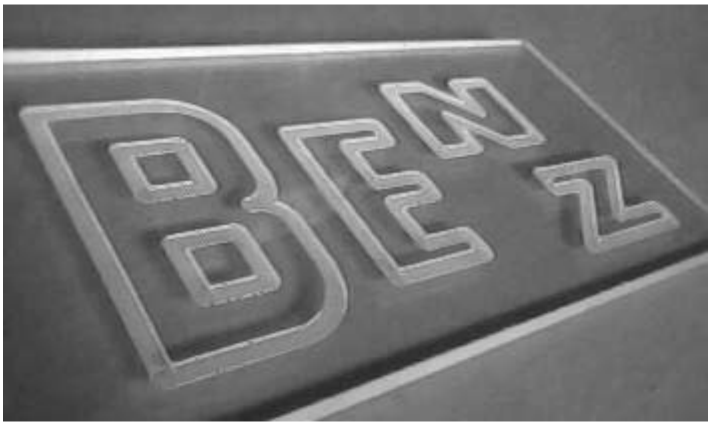
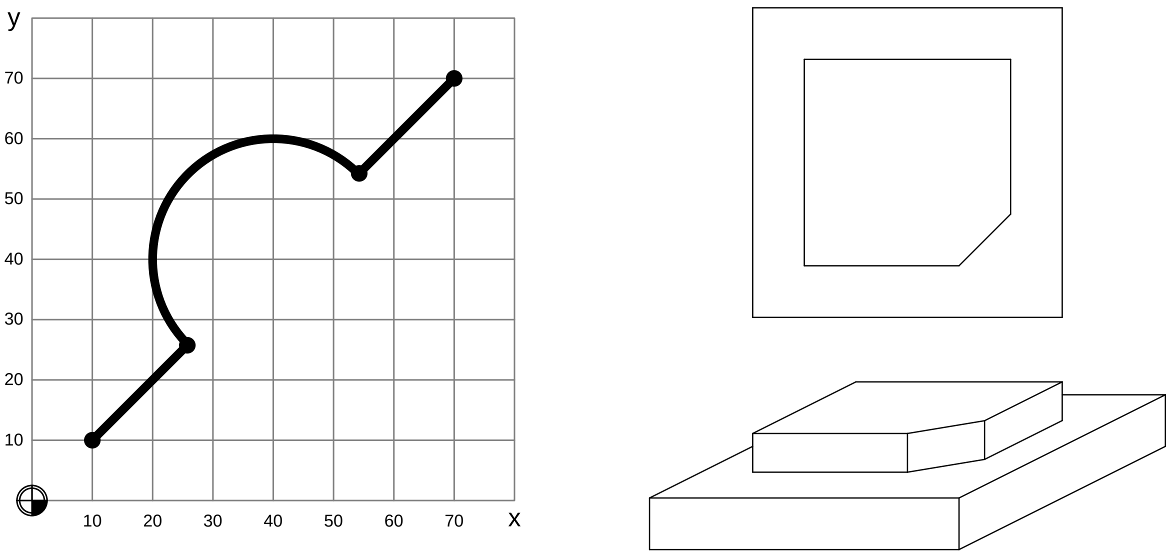


Figure 4.6 A finished workpiece created by a CNC milling

5. Conclusion and Discussion

Using NC programming as a part of analytic geometry class allows students to use their creativity along with course materials. The designed learning activity also encourages students better than learning analytic geometry in an ordinary way since they know how to apply their knowledge with an actual application. The developed virtual lab application works well with the designed activity and can save costs.

However, there are some limitations in manual NC programming. If the designed workpiece is too complex or has some points which cannot be measured directly from the grid as in Figure 5.1a, then an additional software (CAD/CAM) must be used. Anyway, it could be a challenge if any student can calculate for intersection points in Figure 5.1a. Another challenge is shown in Figure 5.1b. Since there are various types of cutting toolhead and each type has different diameter, creating such a workpiece in Figure 5.1b requires a tool diameter compensation which can be further discussed.



a. b.

Figure 5.1 Challenged workpieces

References

[1] K. Lee. (1999). *Principles of CAD/CAM/CAE Systems*. San Diego, CA: Addison-Wesley Longman Publishing.

[2] P. Smid. (2002). *CNC Programming Handbook, 2nd Edition*. New York, US: Industrial Press Inc.

[3] B. Eckel (2006). *Thinking in Java, 4th Edition*. CA: Prentice Hall.