2021 ASEE ANNUAL CONFERENCE
Virtual Meeting | July 26–29, 2021 | Pacific Daylight Time
Paper ID #33439

MAKER: Design of a Virtual CNC Mill by Unity Software

Jose Diaz
Curtrell Trott
Mr. Fatahillah Iskandar
Prof. Ju Wang, Virginia State University
Dr. Zhenhua Wu, Virginia State University

Dr. Zhenhua Wu, is currently an Associate Professor in Manufacturing Engineering at Virginia State University. He received his PhD in Mechanical Engineering from Texas A&M University. His current research interests focus on cybermanufacturing, friction stir welding, sustainable manufacturing, and adaptive machining.

MAKER: Design of a Virtual CNC Mill by Unity Software

Abstract

It requires a lot of hands-on experience to learn how to operate a computer numerical control (CNC) mill. Virtual Reality (VR) can serve as a way to teach how to properly operate it. The goal of this research is to create a virtual CNC mill that can provide interactive training for students. The Unity software was used for this goal. Unity is a game development engine used to produce video games, utility software, and more. The functionality of the CNC simulation was created with C# scripting. The visual representation of the CNC mill was built through 3D modeling, and then transferred into FBX 3D models which are compatible with Unity. The virtual machine is able to take G-code inputs via either an input field or a text file. For the current version, it can simulate G00, G01, and G02/G03 commands and is able to cut sample workpieces per input. It also can save the coordinates of the cutting path via json file and use a python script to view its movement on a line graph. The virtual machine emulates the input commands very similar to the actual machine. This can serve as a good learning tool for CNC machine operators. The virtual machine created through Unity is proof that a digital simulation is achievable for the CNC machine. Future research will include implementing and incorporating mixed reality. Incorporating these kinds of technology will help a more immersive learning experience for students.

1 Introduction and Goal

Computer numerical control (CNC) mill is vital for many engineering and manufacturing practices. However, learning to use it, especially for a beginner with little to no experience, can be challenging. A virtual simulation can serve as a good learning tool for operating a real CNC machine. That way anyone could learn the process of using a machine such as this without the safety hazards that could happen if certain precautions are not taken. A virtual CNC machine would help students with learning how to operate a physical machine in practice. Not only is the simulation safer to learn, but it is more engaging. Allowing students to have interaction with the machine will vastly improve the learning process. Learning through practice is far more effective than trying to memorize a series of steps. That is our goal with this project: A safe, immersive learning experience for students who want to use a machine such as this for their own projects.

2 Related Literature

The authors searched the literature on virtual CNC systems for training and education. Unfortunately, there is not that many recent publications related to this topic. The most related publications are as below.

Chandramouli and Jin et al. introduced [1] the design and development process of VR education tool to simulate different additive manufacturing machines, and a CNC machine to allow the students experience the materials and equipment needed to create the same part using different

types of digital manufacturing technology. El-Mounayri and Aw et al. [2] had developed a virtual manufacturing laboratory of CNC milling. The laboratory environment provided the students: (a) access to a fully-functional virtual CNC milling machine, (b) training on the key operations of the CNC machine, (c) a lecture describing the components of the milling machine, and (d) a lecture describing the concepts of CNC milling. This virtual CNC machine was enabled by three software modules: (1) a CNC Milling machine simulator, (2) a virtual-environment display engine, and (3) an intelligent-agent engine. The three modules was running on a single computer in a seamless web-based framework, which allowed students to access and run the virtual CNC machining center on the web. The CNC Milling machine simulator simulated the machine physical components including the spindle, stages, manual controller interface, and controller software. It also simulated the cutting process including cutting tool motion, workpiece geometry during cutting, cutting forces, machining sounds, and chip-formation. Yao et al. [3] developed numerically controlled (NC) machining training systems based-on virtual machining technology to support training of trainee to acquire the theoretical know-how, practical skills, and the problem troubleshooting techniques on the usage of NC machines and NC programming.

In these publications, the aforementioned virtual CNC machines are all based on the technique of virtual reality. The authors propose a virtual CNC machine using the mixed reality technique. This proposed virtual CNC machine mixes physical machine and digital space, unlocking the interaction between human, computer, and physical CNC machines. It aims to provide a more immersive learning experience for students.

3 Methodology

The development flow for the virtual CNC machine is as Figure 1. It starts from creating 3D models for the real machine using standard CAD software. The 3D models were then converted to another 3D format which is compatible with Unity. The converted 3D model was imported into the Unity engine for developing functional G-codes and CNC machine operation. A user interface was created for human machine interaction, so the virtual machine can receive user input such as G/M codes and button pressing events. The virtual machine can generate optimized mesh cutting paths per the G-code input. The tool also records the toolpath coordinates for analysis.

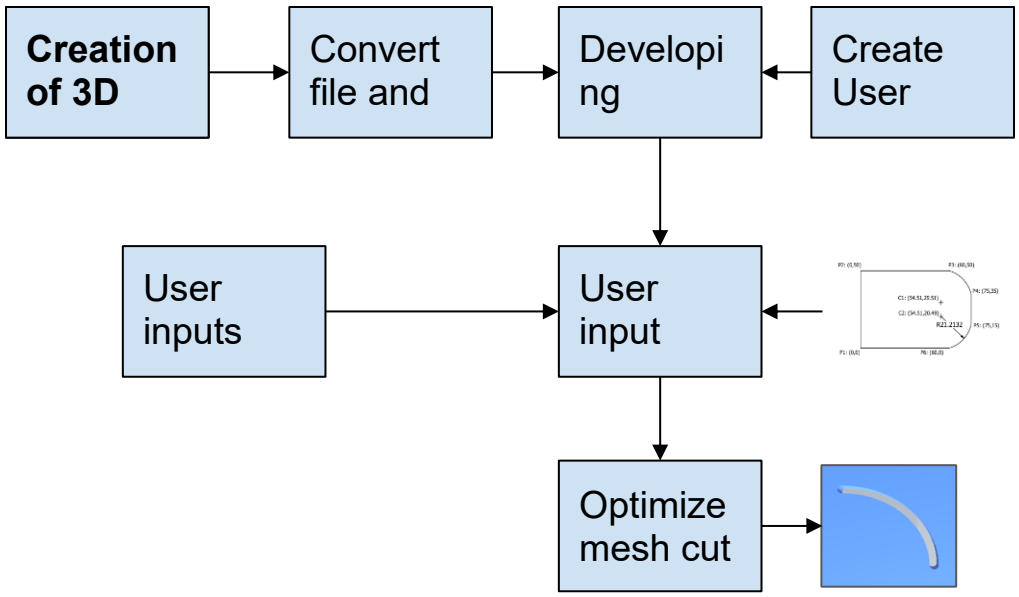


Figure 1. Development Flow for the Virtual CNC Machine

3.1 Creation of 3D Models

In order to create the immersion, a virtual representation had to be created. That way, when the user uses the real thing they have a better idea of what it looks like and how it would act. Solidworks was used to create 3D models for a Haas miniMill. However, to be able to display and use these 3D models in Unity, the models had to be converted to a compatible file format. For this reason the 3D models had to be changed to FBX models. The FBX models imported to Unity is shown as Figure 2.

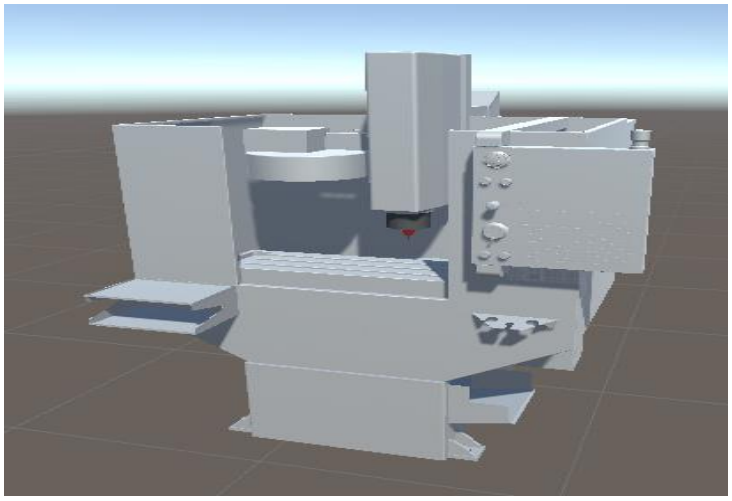


Figure 2. 3D FBX Model Imported in Unity

3.2 User Interface

The user interface (UI) allows the user to interact with the virtual machine. The current UI is as Figure 3. The basic input text field allows the user to type in the individual G-code commands.

The C# scripts behind the virtual machine will interpret the G-code command, and move the working table and spindle accordingly. Users can also click on the “Load” button to select a G-code file which contains a list of commands. After load, users can run the machine by clicking the “Run” button. These input features are loaded into the UI side of Unity for convenience.

There is also a control button that loads a control panel to start the machine. The “On” button creates an air pressure sound assuring the machine is on. The “Reset” button can clear alarms. The blue “Power Up” button moves the working-table to machine home position. From here, it prompts users to enter the size they desire for the mesh to generate. Then once that is finished, it prompts the user to place the mesh in a desired location of x and y. For convenience of the user there are additional buttons like “ Mesh Generator” to create the mesh, and a “Delete” button to remove the mesh and re-generate if the user chooses to. To also help users learn how to power up a CNC machine, there is a tutorial button that will guide him/her to power on the CNC machine like in a real life scenario.

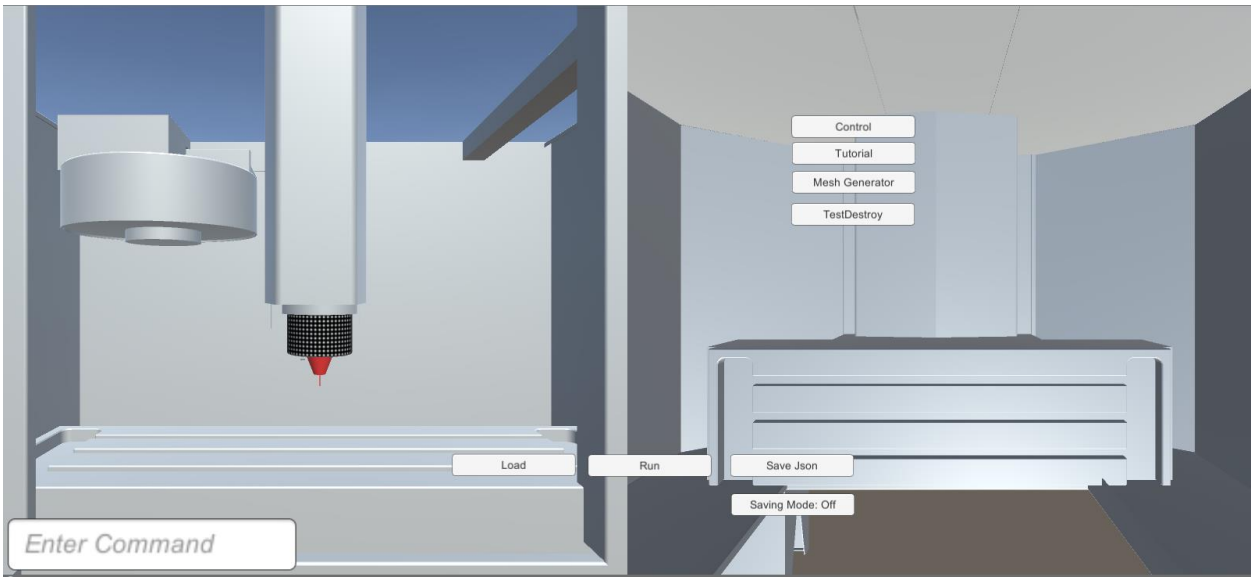


Figure 3. User Interface for the Virtual CNC Machine

3.3 G-code Implementation

G codes are command inputs for CNC machines to move in a specific direction. Each G-code is structured by a code type, followed by the variables needed to create the desired effect. Currently four functional G codes were implemented. G00 is the rapid movement that quickly positions the tool to the desired coordinate. G01 slowly moves the tool in linear interpolation when cutting. G02/G03 moves the tool in circular interpolation. The format that G00 uses is “G00 X## Y## Z##” where the first variable holds the G-code to use which in this case is G00. X-Y-Z coordinates are set up using the right-hand rule: X is meant to move the table from left to right, Y is meant to move the table forward and backwards, and Z is meant to move the spindle up and down from the working table. The Format that G01 follows is “G01 X## Y## Z## F##” which is similar to G00

but the difference with G01 is that it uses F which is known as the feed rate which determines the speed of the working table as the mill is cutting the workpiece material. G02 follows two formats which are ”G02 X## Y## R±## F##” or “G02 X## Y## I## J## F##” where R in the first format is the radius of the arc and I and J are the points for the center of the arc.

Naturally, the code for each must be different. G01 makes use of the “Lerp” function provided by Unity’s API [4]. This function moves the object from its current position to the destination in a linear fashion. The method for G02, however, is more complicated. In order to cut an arc, the G02 implementation first calculates the desired circle that fits the provided parameters. It does this by calculating the radius of the circle using the center point provided. Then, our algorithm calculates the current angle relative to the center of the circle and the target angle of the end point of the desired cutting path. Finally, the working table moves along this circle until it reaches the angle the end point is located. The G03 works the same way, it just moves in the counterclockwise direction.

3.4 Mesh Cutting and Optimization

The simulated cutting is performed as follows: a mesh that is broken into segments is created. The cutting algorithm moves the spindle down to the workpiece according to the G code. When the spindle touches the mesh on the table, a hole will be created at the contact point using Unity’s collision mechanism. The initial method involved an array of cubes to represent the work piece. When collision occurs, the cubes are deleted on contact with the mill. However, this was very processing intensive and the simulation slowed down significantly on larger work pieces and high resolution. The results were far from ideal whenever anything beyond a straight horizontal or vertical line was involved. As shown in Figure 4, the original cutting algorithm generated a very coarse cutting result.

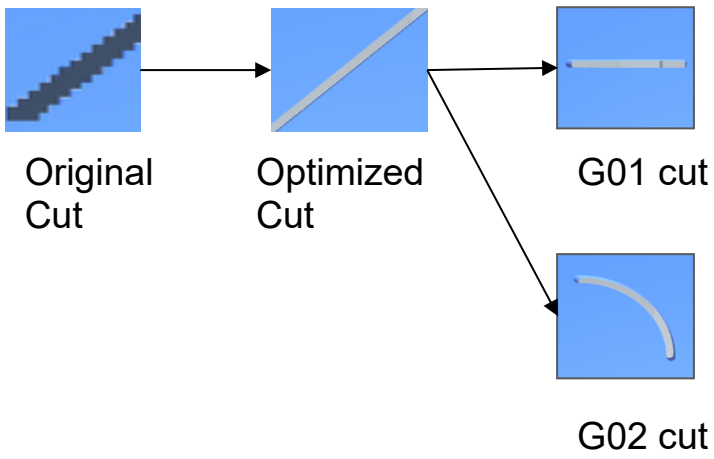


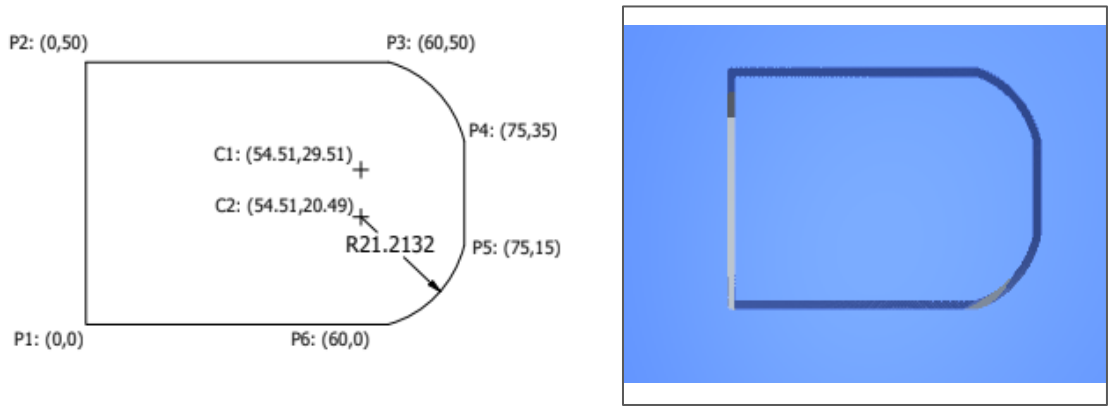
Figure 4. Original cut and re-topologizing optimization for G01 and G02

In order to achieve a more realistic cutting with a smooth edge as shown in Figure 4, a re-topologizing script was implemented. the "Clipper" library was used for cutting path optimization. The library is developed by A. Johnson [5] which is an extension of B. Vatti's clipping algorithm to help the optimization of mesh cutting for the CNC Virtual Machine. The Vatti's clipping algorithm is used for general types of polygons regardless of their shape and size and is also very efficient compared to other clipping algorithms which are limited to certain types of polygons [6]. The Vatti's algorithm clips the subject polygon against another polygon which is known as the clip polygon [7]. The polygons have edges and when these edges touch each other, the algorithm starts to process the polygon to a non-intersecting polygon by giving it intersection points [7]. As the subject and clip polygon pass through each other once, all of their bounds are then created with a flag that indicates its type as either clip or subject of the edge [7]. Then the polygons are examined from the very bottom to the top through the use of scanbeams, which is the amount of area between a pair of horizontal lines and another pair of horizontal lines which go through each vertex [7]. Lastly, all the edges that have crossed with a scan beam are kept in order from the lowest to the highest depending on their x coordinates of each scanbeam and are updated when they are scanned [7].

The "Clipper" library also incorporates an algorithm of polygon offsetting with winding numbers. The winding number is the amount of times a closed curve goes around a point [8]. The algorithm first offset the edges of the shape [8]. Then these offset edges endpoints are connected to the original edges with an arc if they share a concave vertex [8]. For a shared convex vertex, the offset edges would be connected to a line segment [8]. The next step of the algorithm is to calculate the winding numbers of each region [8]. If the winding number is positive then the region would be the interior of the inner offset polygon [8]. Then lastly, getting the boundary of the inner offset polygon [8]. The "Clipper" library seems to use a combination of offsetting and clipping that helped create a smoother and optimized mesh cut for the virtual CNC machine. It effectively re-topologized the area of where it cuts.

4 Results of a complete machine part

Figure 5 shows a complete cutting result from a benchmark input file. The workpiece to be processed consists of four segments of straight line features and two arc features. The coordinates of each features is shown in Figure 5(a). The output of the continuous cutting from the benchmark is shown in Figure 5(b). The result is identical as the benchmark file. The processing is also in real-time due to the optimized meshing method.



(a) **(b)**

Figure 5. Benchmark input (a) and virtual machine output (b)

5 Conclusion and Future Work

The virtual CNC machine can serve an excellent learning tool for students learn real machine operation for any potential projects relating to engineering and manufacturing. The virtual CNC machine is able to simulate G code movement and takes input similarly to the real machine in which the spindle moves according to Z and the table moves according to X and Y. The authors hope to join the virtual CNC machine with mixed reality as the next step in development. Creating a mixed reality environment for this virtual 3D machine would make the experience more interactive and almost realistic for anyone who uses this software.

Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant No. 1818655. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Reference

[1] Chandramouli, M., & Jin, G., & Heffron, J. D., & Fidan, I., & Cossette, M., & Welsch, C. A., & Merrell, W. (2018, June), *Virtual Reality Education Modules for Digital Manufacturing Instruction,* Paper presented at 2018 ASEE Annual Conference & Exposition , Salt Lake City, Utah. 10.18260/1-2—31225

[2] El-Mounayri, H. (2005, June), *Virtual Manufacturing Laboratory for Training and Education,* Paper presented at 2005 Annual Conference, Portland, Oregon. 10.18260/1-2--15154

[3] Yingxue Yao, Jianguang Li, Changqing Liu, *A Virtual Machining Based Training System For Numerically Controlled Machining*, Computer Applications in Engineering Educations, Volume15, Issue1, 2007, Pages 64-72

[4] Unity API: https://docs.unity3d.com/ScriptReference/

[5] Johnson, Angus (2010-2013) Clipper - an open source freeware library for clipping and offsetting lines and polygons. : http://www.angusj.com
[6] Agoston, K., Max (2005, January 4) Computer graphics and geometric modeling: implementation and algorithms: http://books.google.com/books?q=vatti+clipping+agoston
[7] Vatti, R., Bala (1992, July) A generic solution to polygon clipping: https://dl.acm.org/doi/pdf/10.1145/129902.129906?casa_token=fQsVbBA4bIQAAAAA:KoZrlhghJG525glYFKldGJGdtQTpodEVxXfaa-1u4cvIWhPrpyehMDQHUWY7YN-GPDy2_uEnMxfm
[8] Chen, Xiaoru, McMains, Sara (2005) Polygon Offsetting by Computing Winding Numbers : http://www.me.berkeley.edu/~mcmains/pubs/DAC05OffsetPolygon.pdf