

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

IURY JUSTO DE BARROS

**Projeto de uma fresadora CNC para materiais de baixa e média
densidade.**

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em
Engenharia de Computação.

Orientador: Prof. Dr. Eduardo Simões Lopes
Gastal

Porto Alegre

2019

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Oppermann

Vice-Reitor: Prof^a Jane Fraga Tutikian

Pró-Reitor de Graduação: Prof. Vladimir Pinheiro do Nascimento

Diretor do Instituto de Informática: Prof^a Carla Maria Dal Sasso Freitas

Coordenador do Curso de Engenharia de Computação: Prof. André Inácio Reis

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

RESUMO

Essa monografia descreve todo o projeto de uma máquina CNC, tanto o hardware como o software desenvolvido, e as decisões de projeto tomadas para a sua criação. Sempre levando em consideração obter o melhor custo benefício possível. A intenção desse projeto é ter uma máquina e um software que faça trabalhos de corte em materiais como MDF para serem utilizados para fins de arte ou criação de peças de madeira com uma boa precisão (artesanato). O software é de fácil utilização para que qualquer pessoa com conhecimentos básicos em CNC consiga operá-la. O programa gera o caminho de corte a partir de uma imagem digital de elevação, com o objetivo de minimizar o tempo de corte.

Palavras-chave: CNC. Projeto de Hardware e Software de uma CNC. Gerador de código Gcode.

CNC Project for low and medium density materials

ABSTRACT

This paper describes the entire design of a CNC machine, both the hardware and software developed, and the design decisions taken to create it. Always aiming to achieve the best possible cost benefit. The intent of this project is to have a machine and software that does cutting work on materials like MDF wood, to be used for art purposes or creation of pieces of wood with good precision (craftsmanship). The software is easy to use for anyone with basic knowledge of CNC. The software creates the cutting path from a digital elevation image, with the goal of minimizing the cutting time.

Keywords: CNC. Hardware and Software project for a milling machine. Gcode code generator.

LISTA DE FIGURAS

Figura 1. Homem esculpindo desenhos na madeira	11
Figura 2. CNC esculpindo desenhos em madeira	11
Figura 3. HSM-MODAL em operação	12
Figura 4. Imagem do projeto da CNC.....	13
Figura 5. Imagem com os eixos da CNC aparentes	13
Figura 6. Imagem da CNC montada	14
Figura 7. Fuso trapezoidal com castanha em bronze	15
Figura 8. Fuso de Esferas.....	15
Figura 9. Sistema de movimentação utilizando Correias	15
Figura 10. Pillowblock suspenso em uma guia linear	16
Figura 11. Tupia Makita RT0700C	18
Figura 12. Driver TB6600 com dissipador e proteção contra curto-circuito	19
Figura 13. Diagrama físico das conexões	20
Figura 14. Percurso quadrado com aresta de 10mm e centro em (5,5) em uma fresa de 3mm de diâmetro com corte por dentro, origem em (1.5,1.5)	22
Figura 15. Código escrito em Gcode para o percurso da Figura 14 – Fonte: Autor	22
Figura 16. Universal G-CodeSender.....	23
Figura 17. Máscara para a fresa de 3mm	24
Figura 18. Máscara para a fresa de 6mm	25
Figura 19. Na esquerda a imagem z-buffer, no centro a imagem de percurso no nível 4 (veja próxima seção) para a fresa de 3mm e na direita a mesma imagem para a fresa de 12mm	26
Figura 20. Imagem em z-buffer de uma esfera.....	27
Figura 21. Imagens de percurso nível 1 para fresa de 3 mm (esquerda) e 12 mm (direita)	27
Figura 22. Imagens de percurso nível 2 para as fresas de 3 e 12 mm	28
Figura 23. Imagens de simulação de corte: diâmetro de 3 mm com passo de 1 mm, diâmetro de 6 mm com passo de 6 mm e diferença entre os dois cortes	29
Figura 24. Representação de uma fresa com diâmetro da lâmina de corte menor do que a haste	30

Figura 25. Interface do Programa Gerador Gcode desenvolvido	31
Figura 26. Esfera no formato z-buffer	32
Figura 27. Imagem de percurso nível 1.....	33
Figura 28. Imagem de percurso nível 2.....	34
Figura 29. Imagem de percurso nível 3.....	34
Figura 30. Imagem de percurso nível 4.....	34
Figura 31. Imagem de percurso nível 5.....	35
Figura 32. Imagem de simulação de corte nível 1	35
Figura 33. Imagem de simulação de corte nível 2	36
Figura 34. Imagem de simulação de corte nível 3	36
Figura 35. Imagem de simulação de corte nível 4	36
Figura 36. Imagem de simulação de corte nível 5	37
Figura 37. Imagem gerada pela diferença de corte das fresas no primeiro nível	38
Figura 38. Imagem com um bloco pintado de vermelho	38
Figura 39. Pré-visualização gerada pelo G-CodeSender ao inserir o Gcode gerado para a Figura 37	39
Figura 40. Imagens dos cortes nos níveis 1 (esquerda) e 2 (direita) com a fresa de 12 mm e passo de 5 mm.....	40
Figura 41. Imagens dos cortes nos níveis 3 (esquerda) e 4 (direita) com a fresa de 12 mm e passo de 5 mm.....	40
Figura 42. Imagem do corte no nível 5 com a fresa de 12 mm e passo de 5 mm.....	41
Figura 43. Imagens dos cortes nos níveis 1 (esquerda) e 2 (direita) com a fresa de 3 mm e passo de 1 mm cortando por blocos	42
Figura 44. Imagens dos cortes nos níveis 3 (esquerda) e 4 (direita) com a fresa de 3 mm e passo de 1 mm cortando por blocos	42
Figura 45. Imagem do corte no nível 5 com a fresa de 3 mm e passo de 1 mm cortando por blocos	43
Figura 46. Resultado final após a última passagem	44
Figura 47. Imagem de uma xilogravura	45
Figura 48. Corte de xilogravura usando uma fresa de 3 mm.....	46

LISTA DE TABELAS

Tabela 1. Algumas funções da linguagem Gcode – Fonte: Autor.....	21
--	----

LISTA DE ABREVIATURAS E SIGLAS

CNC	Comando numérico computadorizado
MDF	Medium Density Fiberboard (Madeira com densidade média)
mm	Milímetros
cm	Centímetros

SUMÁRIO

1. INTRODUÇÃO	11
2. PROJETO MECÂNICO	13
2.1 Estrutura	13
2.2 Fusos trapezoidais	15
2.3 Rolamentos Pillowblock	16
2.2 Motores	16
2.2.1 Dimensionamento do motor de Passo	17
2.3 Ferramenta de Corte	18
3. PROJETO ELÉTRICO	19
3.1 Microcontrolador	19
3.2 Drivers dos Motores	19
3.3 Calculo de Potência e Fonte de Alimentação	20
4. PROJETO DE SOFTWARE	21
4.1 Grbl	21
4.1.1 Configuração do Grbl	21
4.2 GCode	22
4.3 Universal G-CodeSender	23
4.4 Programa desenvolvido	24
4.4.1 Proposta e Requisitos	24
4.4.2 Imagem de Percurso	24
4.4.2.1 Geração da Máscara para o Filtro	25
4.4.2.2 Aplicação do Filtro	26
4.4.2.3 Divisão da imagem de percurso em níveis	27
4.4.3 Imagem de Simulação de Corte	29
4.4.4 Geração do código GCode	30
4.4.5 Detector de colisão	31
5. RESULTADOS OBTIDOS	32
5.1 Exemplo com imagem de uma meia esfera	33
5.2 Imagens dos percursos	34
5.3 Imagens dos cortes	36
5.4 Processamento da diferença de imagens de corte	38
5.5 Enviando o Gcode para a CNC	40

5.6 A última passagem para corrigir imperfeições	44
5.7 Análise do tempo de corte	45
5.8 Criando xilogravuras	45
6. MELHORIAS FUTURAS	47
7. CONCLUSÕES	48
REFERÊNCIAS	49
APÊNDICE A – Lista de componentes	50
APÊNDICE B – Valores por componentes	51

1. INTRODUÇÃO

Originalmente, as máquinas para fazer cortes em madeiras funcionavam de forma manual. Até existiam suportes e gabaritos que quando acoplados davam maior precisão ao trabalho. No entanto, para conseguir um trabalho bem feito, era necessário que o operador tivesse uma enorme habilidade com o manuseio da máquina, ver Figura 1, e ainda assim não se pode comparar com o desempenho dos dispositivos atuais. Hoje, essas máquinas são amplamente utilizadas para a automação industrial e seu papel se tornou fundamental para o aperfeiçoamento e qualidade dos produtos por elas fabricados, ver Figura 2.

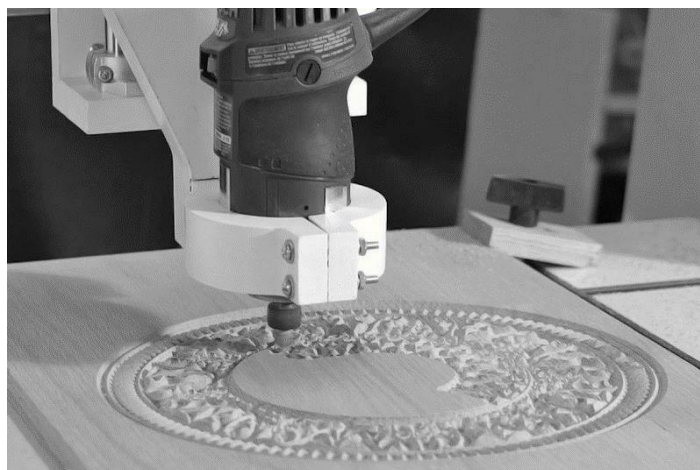
O culminar das três tecnologias (Máquinas ferramenta, Automação, Informática), e a ansiedade evolutiva humana trouxeram-nos hoje as possibilidades de construir peças e insumos de extrema complexidade e precisão (Azevedo, 2018). O tempo gasto para a confecção de um produto foi outro fator muito importante que foi aperfeiçoado com a automação. Enquanto um escultor demorava dias para produzir uma peça, a mesma pode ser feita em alguns minutos pela máquina, além de manter certo padrão de qualidade.

Figura 1. Homem esculpindo desenhos na madeira



Fonte: Nagy-BagolyArpad

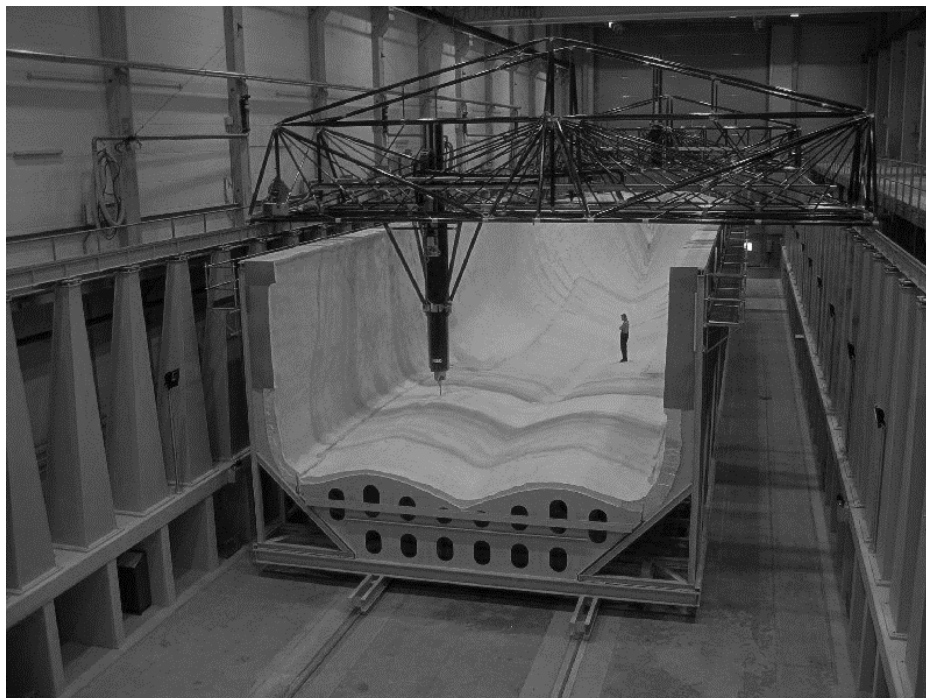
Figura 2. CNC esculpindo desenhos em madeira



Fonte: Mecânica Industrial

Comando numérico computadorizado (CNC) consiste em uma máquina ou mais máquinas que são comandadas por instruções expressas em números, que por sua vez representam posições no plano cartesiano da máquina, (Ryan, 2018). Esses equipamentos podem possuir 3, 4 ou 5 eixos e ter uma área de trabalho de alguns centímetros até centenas de metros como é o caso da HSM-Modal (CNC alemã com 150 metros de trabalho no eixo X). Na Figura 3, podemos ter uma noção das dimensões dessa máquina.

Figura 3. HSM-MODAL em operação



Fonte: EEW PROTEC

Com o aumento dos programas de CAD (*Computer Aided Design*) para a programação de CNC, o trabalho dos programadores de CNC foi facilitado. No entanto, a grande maioria desses softwares não são gratuitos ou não possuem uma interface muito amigável para o usuário. Para resolver esse problema, este trabalho desenvolveu um software que é de fácil utilização para desenhar na madeira o conteúdo de uma imagem digital.

2. PROJETO MECÂNICO

2.1 Estrutura

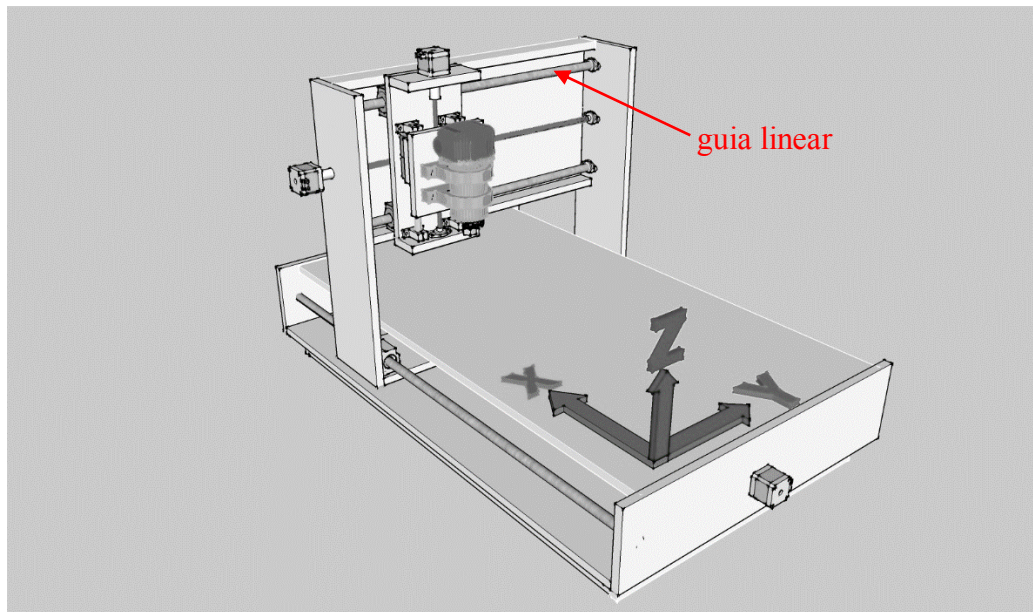
Para o projeto mecânico devemos definir o material da estrutura da máquina, onde as opções mais utilizadas são de madeira (MDF), polietileno e metal. Pelo custo benefício foi utilizado madeira, no entanto, o tempo de vida útil da máquina será reduzido por esse material ser menos resistente que os outros e pode haver rachaduras com o tempo de trabalho.

Para a escolha das dimensões foi escolhido uma área de trabalho de 80 cm no eixo X, 45 cm no eixo Y e 10 cm no eixo Z, conforme Figura 4. Essas dimensões impactam diretamente sobre o custo da CNC, pois aumentam o peso, e

consequentemente, necessitam materiais mais resistentes como um diâmetro maior das guias lineares.

Existem diversos tipos de modos de trabalho das CNCs, mas os mais comuns são: a ferramenta de corte se movimentando sobre o material a ser cortado ou o inverso, a mesa de corte que é movimentada. Para esse projeto foi escolhido o primeiro modo, haja vista que para movimentar a mesa de corte ao invés da estrutura da fresadora é necessário mais torque, visto que o material a ser cortado pode ser pesado. E quanto maior o torque necessário, maior será a potência dos motores e, consequentemente, mais caros eles serão.

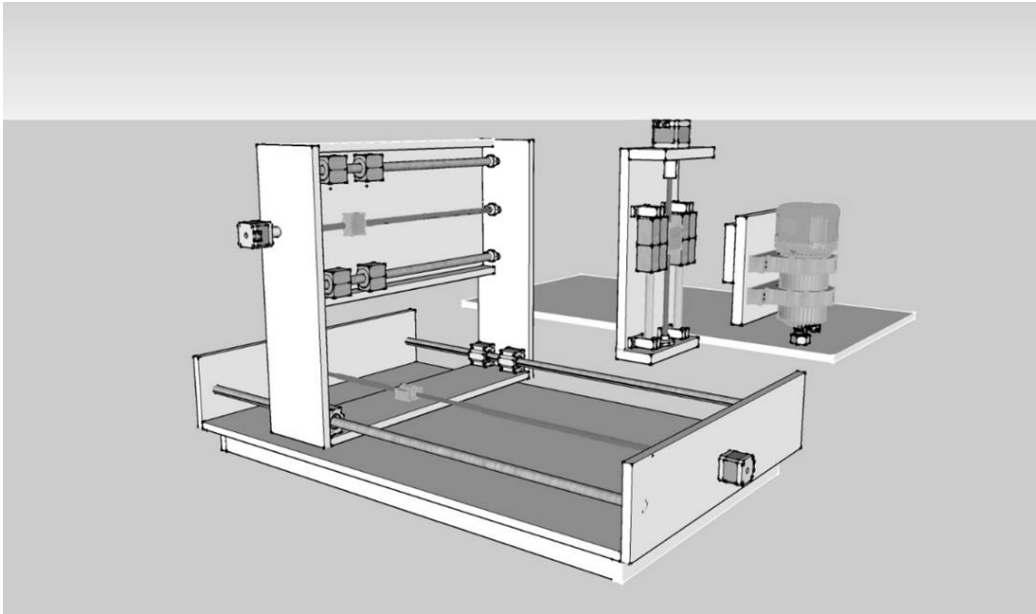
Figura 4. Imagem do projeto da CNC



Fonte: Autor

Na Figura 5, podemos ver o desenho do projeto em uma visão parcialmente explodida, no qual podemos perceber melhor como se dará o movimento dos eixos.

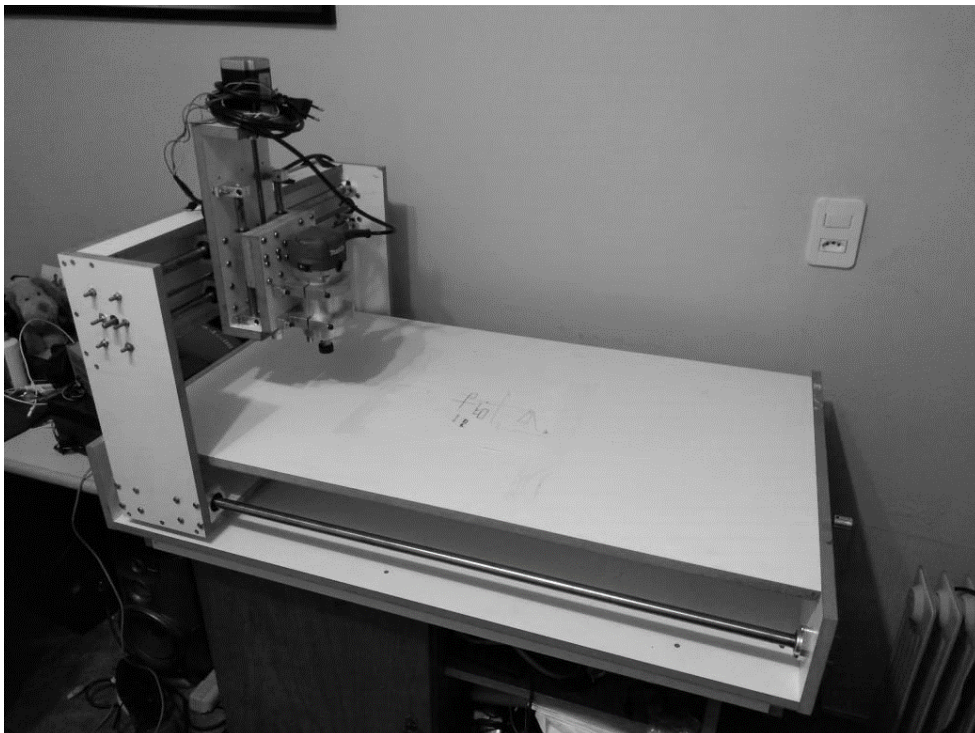
Figura 5. Imagem com os eixos da CNC aparentes



Fonte: Autor

Na Figura 6, podemos ver o resultado do projeto mecânico com a CNC construída.

Figura 6. Imagem da CNC montada



Fonte: Autor

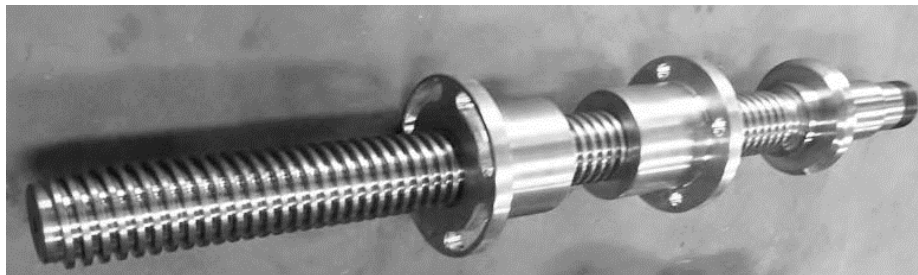
2.2 Fusos trapezoidais

Fusos trapezoidais são barras roscadas de metal que tem como objetivo transformar um movimento angular em um movimento linear de forma suave e uniforme (Galdino, 2014).

As opções analisadas foram os fusos trapezoidais, os fusos de esferas e as correias acopladas ao eixo. Embora os fusos trapezoidais sejam até cinco vezes mais caros do que utilizar correias (sistema utilizado em impressoras, Figura 9), eles apresentam menos folga e dão maior precisão na CNC, (Oriental Motor do Brasil Ltda, 2018). Já os fusos de esferas, ver Figura 8, possuem uma grande vantagem em relação aos fusos trapezoidais, uma vez que o rendimento de um fuso trapezoidal fica em torno de 40 a 50% do torque do motor, e o de esferas chega a ultrapassar os 90%. No entanto, o valor disponível para o projeto não foi suficiente para o uso desses fusos mais complexos.

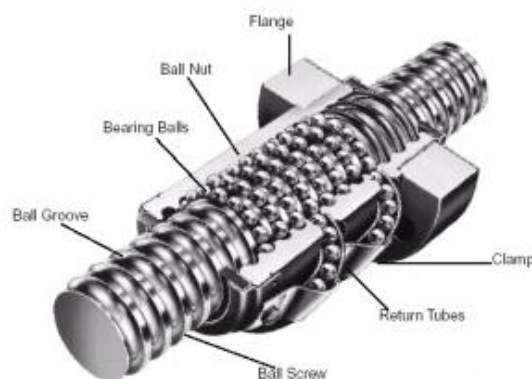
O que foi selecionado para o projeto foram fusos trapezoidais TR8 com passo de 8mm juntamente com duas castanhas em bronze, similar ao apresentado na Figura 7.

Figura 7. Fuso trapezoidal com castanha em bronze



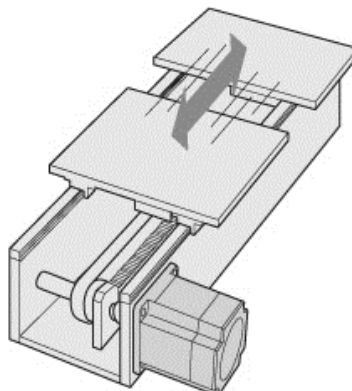
Fonte: Ava Linear System

Figura 8. Fuso de Esferas



Fonte: Kalatec Automação

Figura 9. Sistema de movimentação utilizando Correias

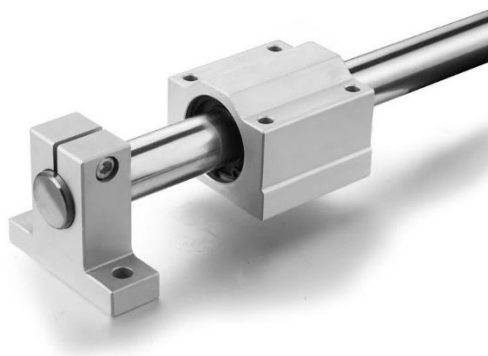


Fonte: Oriental Motor

2.3 Rolamentos Pillowblock

Para o suporte e deslize dos eixos foram utilizados rolamentos do tipo *Pillowblock* que restringem o deslocamento em uma única direção no modo duplo, ou seja, dois rolamentos para cada guia o que aumenta ainda mais a precisão da máquina. Seu sistema de rolamento com vedação diminui o nível de sujeiras entre os rolamentos garantindo uma boa vida útil para o equipamento.

Figura 10. Pillowblock suspenso em uma guia linear



Fonte: Uptech Brasil

Na Figura 10, vemos um Pillowblock do tipo SC16UU sobre uma guia linear de 16mm de diâmetro apoiada sobre um suporte Sk16. Sistema utilizado em todos eixos da máquina, exceto no eixo Z que o diâmetro dos componentes foi de 12mm.

2.2 Motores

Para o acionamento dos eixos da máquina foram utilizados motores de passo, pois esses motores possuem uma maior precisão e possuem uma vida útil bem grande, em torno de 12 anos, em condições normais de trabalho.

2.2.1 Dimensionamento do motor de Passo

Para o dimensionamento do motor de passo foram utilizadas algumas fórmulas para o cálculo do torque requerido. As fórmulas são baseadas em informações dos componentes do sistema (Hannifin, Parker, Outubro 2003).

Primeiramente calculamos a inércia da carga que leva em consideração a massa da mesa (m) em quilogramas e o tamanho do passo do fuso (p) em metros:

$$Inercia\ de\ carga = m * \left(\frac{p}{2 * \pi} \right)^2.$$

Após o cálculo da inércia da carga devemos calcular a inércia do próprio fuso. Para tanto, utilizamos a densidade do material do fuso (dF) em quilograma por metro cúbico, o diâmetro do fuso (D) em metros, e o comprimento total do fuso (L) também em metros:

$$Inercia\ do\ fuso = dF * D^4 * L.$$

Somando as duas inércias e a inércia do próprio motor (Jm), nós obtemos a inércia total do sistema:

$$Inercia\ total = Inercia\ de\ carga + Inercia\ do\ fuso + Jm.$$

Para o cálculo do torque necessário durante a aceleração iremos utilizar o valor encontrado pela inércia total do sistema e informações físicas da máquina como a quantidade de passos por segundo inicial do sistema (PPS), quantos passos por segundo final do sistema (PPF), a quantidade de passos por revolução do motor (PPR), o tempo de aceleração e a eficiência do fuso (n) em percentual.

Para isso calculamos primeiramente a velocidade inicial em radianos por segundo:

$$Velocidade\ inicial\ (rad/s) = \frac{2 * \pi * PPS}{PPR}.$$

Depois, a velocidade final em radianos por segundo:

$$Velocidade\ final\ (rad/s) = \frac{2 * \pi * PPF}{PPR}.$$

Com ambas as velocidades calculadas, basta informar o tempo necessário para que a máquina atinja a velocidade final para calcular o valor da aceleração em radianos por segundo ao quadrado:

$$Aceleração = \frac{Velocidade\ final - Velocidade\ inicial}{tempo\ de\ aceleração(s)}.$$

Para o cálculo do torque necessário para a aceleração, utilizamos os valores de inercia total e de aceleração previamente calculados, como mostra a equação abaixo:

$$Torque\ de\ aceleração = \frac{Inercia\ total * Aceleração}{2 * \pi * n}.$$

Ao obter o torque de aceleração, falta apenas obter o torque necessário para a carga do sistema. Para esse cálculo é necessário saber, a massa da mesa (m), o tamanho do passo no fuso (p), o coeficiente de fricção da guia linear com o *Pillowblock* e a eficiência do fuso (n):

$$\text{Torque de carga} = \frac{m * \text{aceleração da gravidade}(\sim 9,8) * p * \mu}{2 * \pi * n}$$

Feito o cálculo do torque de carga e o torque de aceleração, basta somá-los para obter o torque total necessário dos servos motores:

$$\text{Torque total} = \text{torque de carga} + \text{Torque de aceleração}.$$

Ao sabermos o torque total, multiplicamos por um fator de segurança (algo entre 1,5 e 2), para que o motor não trabalhe sobre muito estresse. Substituindo os valores reais da CNC projetada encontramos o valor de 0,64 N.m, ou seja, 6,4 Kg.f.cm. Para trabalhar com esse torque foi escolhido os motores Nema 23 de 15Kg.f.cm que em sua ligação no modo Unipolar possui um torque de 10Kg.f.cm (Neomotion, 2018).

2.3 Ferramenta de Corte

Existem diversos tipos de ferramentas para usinagem como tupias, retificas, *spindles*, *laser* ou jato de água. Cada um destes tipos possuem suas vantagens e desvantagens dependendo do tipo de material que irá cortar. Como nesse trabalho o propósito é usinar madeira com uma espessura acima de 15mm, os mais indicados são as tupias e os *spindles* dado que possuem uma boa relação entre potência e custo.

Os *spindles* tem uma vantagem grande em relação as tupias que é a possibilidade de diminuir e aumentar a rotação através de um sinal enviado pelo microcontrolador. Isso é excelente, pois materiais com maior dureza devem trabalhar com rotação menor para preservar as fresas (laminas de corte). No entanto, o custo de um *spindle* chega a ser três vezes mais se comparado com uma tupia de mesma potência.

A solução com melhor custo benefício encontrada foi a tupia RT0700C (Makita do Brasil, 2018), pois além de possuir uma potência relativamente alta (710 Watts), ela possui um controle de rotações por minuto de 6 velocidades que vai de 10.000 até 30.000 rpm. Esse controle de velocidade é feito de forma manual na parte superior da tupia como podemos ver na Figura 11.

Figura 11. Tupia Makita RT0700C



Fonte: Makita do Brasil

3. PROJETO ELÉTRICO

3.1 Microcontrolador

Com as diversas opções de microcontroladores foi analisado o custo e a praticidade de programação. Como existem firmwares consolidados para trabalhar em CNCs utilizando a plataforma Arduino, foi selecionada a versão Arduino Uno. No entanto, as versões Arduino Nano e ArduinoMega também poderiam ser utilizadas respeitando a versão do Grbl para cada uma delas, conforme a sessão 4.1 Grbl.

3.2 Drivers dos Motores

Um driver de motor de passo funciona através de uma placa geradora que controla todos os pulsos utilizados para o procedimento. Dando a potência necessária para movimentar o motor. Sendo assim o motor funciona da seguinte forma: microcontrolador manda sinal para o driver, o mesmo é acionado e faz chaveamentos para dar a tensão e corrente necessária para o motor (Kalatec Automação Ltda, 2018).

Existem inúmeras soluções de drivers para atuar em motores de passo utilizando Arduino, podemos citar três: o A4988, o DRV8825 e o TB6600.

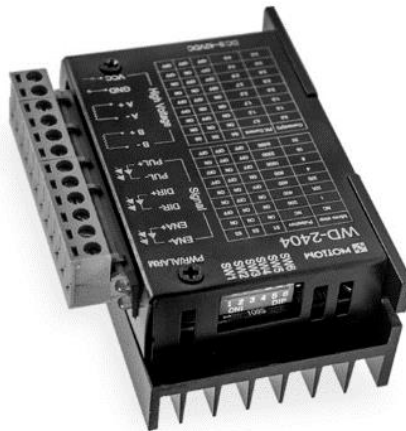
O driver A4988 é uma opção barata para as CNCs com motores que possuem tensão de 8V a 35V e uma corrente máxima de 2A e consegue aumentar a resolução, ou seja, consegue reduzir o tamanho de um passo em até 16 vezes utilizando sua configuração de micro passos.

O driver DRV8825 consegue trabalhar nas tensões de 8V até 45V e controlar a corrente até 2,5A e consegue aumentar a resolução em 32 vezes utilizando os micro passos.

O driver TB6600, Figura 12, é uma opção mais cara, porém trabalha com tensão de 12V a 48V e controla motores de até 4A. Sua configuração de micro passos consegue aumentar a resolução em 32 vezes.

Como em nosso projeto foi necessário utilizar motores que necessitam de até 3A, o driver selecionado foi o TB6600, desenvolvido pela Toshiba. A resolução nominal desse driver é de 200 passos por volta e se utilizar a configuração máxima de micro passos conseguimos uma resolução de 6400 passos por volta. Como nosso sistema de fusos possui uma distância de 8mm por volta do motor, o menor deslocamento possível da máquina será 800 passos por mm, ou 1,25 μ m por passo.

Figura 12. Driver TB6600 com dissipador e proteção contra curto-circuito



Fonte: Wotiom

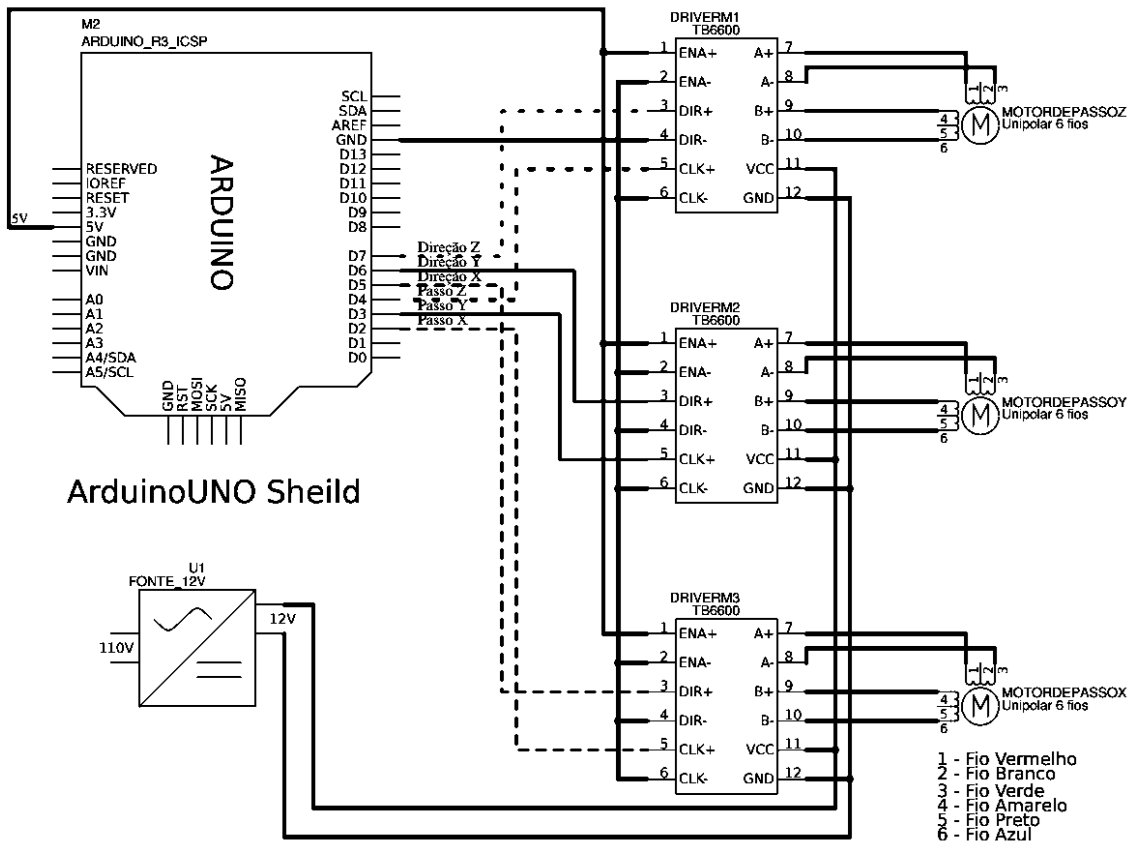
3.3 Calculo de Potência e Fonte de Alimentação

Após selecionarmos os atuadores e seus drivers, foi necessário escolher uma fonte de tensão adequada para alimentá-los. Como o driver trabalha com as tensões de 12V a 48V, optou-se por utilizar uma fonte de computador de 12V. No entanto, essa fonte deve possuir uma potência suficiente para trabalhar com três motores ao mesmo tempo, já que para esse projeto é necessário movimentar três eixos independentes (eixos X, Y e Z).

Cada motor possui uma corrente máxima de 3A, trabalhando a uma tensão de 12V, ou seja, nossa fonte de alimentação tem como requisitos na saída de 12V suportar no mínimo uma corrente de 9A. Foi utilizada uma fonte de computador ATX com potência de 422W e saída de até 18A na tensão 12V.

Na Figura 13, podemos ver como ficou diagrama físico das conexões entre os motores, os drivers, a fonte de alimentação e o microcontrolador.

Figura 13. Diagrama físico das conexões



4. PROJETO DE SOFTWARE

4.1 Grbl

O Grbl é um firmware gratuito, robusto e eficiente que interpreta a linguagem GCode, descrita na seção 4.2 GCode. Esse firmware está presente na comunidade *open source* desde 2009 e vem passando por melhorias desde então. Hoje, se tornou um dos interpretadores mais famosos quando o assunto se trata de CNCs (Jeon & Skogsrud, 2018).

Um grande desafio enfrentado por seus desenvolvedores foi otimizar bastante o código para que caiba nos 32Kbits de memória disponível no Arduino Uno. Além dos controles de movimento dos eixos, o Grbl controla algumas operações da ferramenta de corte e fluido de usinagem.

4.1.1 Configuração do Grbl

O Grbl possui várias configurações que necessitam ser ajustadas para o funcionamento correto da CNC. Essas configurações são mantidas na memória EEPROM do Arduino, ou seja, não serão apagadas após o desligamento do sistema.

Como essas configurações são específicas de cada máquina, velocidade de operação, por exemplo, não serão abordadas nesse trabalho mas são facilmente configuráveis utilizando a documentação disponível pelo Grbl (Jeon & Skogsrud, 2018).

4.2 GCode

Gcode é uma linguagem de programação voltada para as máquinas de controle numérico. Foi desenvolvida por volta de 1950 em um laboratório de mecânica no Instituto de Tecnologia de Massachussets.

Sua sintaxe é baseada em códigos para representar funções, e números que complementam esses códigos com valores. Por exemplo, para fazer a máquina andar sobre o sistema de coordenadas em sua velocidade máxima para a posição X=0 e Y=0 usamos o código G0x0.000y0.000 (Autodesk, Inc, 2018).

Na Tabela 1, temos uma lista simplificada de funções do Gcode:

Tabela 1. Algumas funções da linguagem Gcode

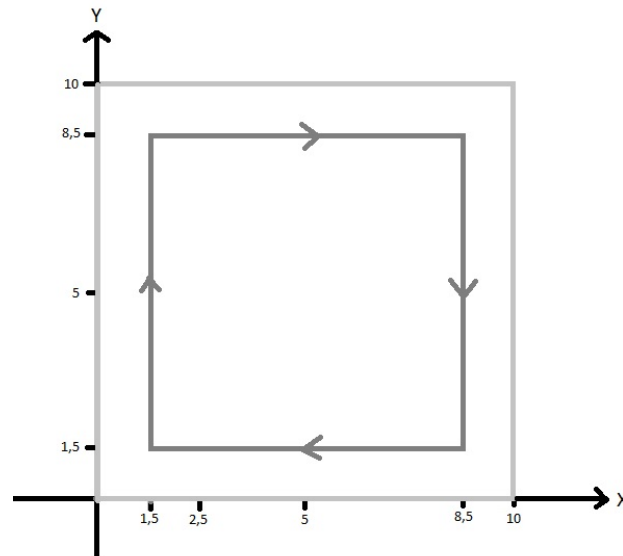
Código	Descrição	Tipo
G00	Posicionamento rápido	Modal
G01	Interpolação linear	Modal
G02	Interpolação circular horária	Modal
G03	Interpolação circular anti-horária	Modal
G04	Tempo de espera em segundos	Não modal
G70	Seleciona medidas em polegadas	Modal
G71	Seleciona medidas em milímetros	Modal
G90	Seleciona posicionamento absoluto	Modal
G91	Seleciona posicionamento incremental	Não modal
M06	Troca de ferramenta	Não modal
M30	Encerra o programa principal	Não modal

Fonte: Autor

As função classificada como modal é aquela que permanece ativa até que encontre outra função “G”, onde ela é cancelada para a execução da nova instrução.

Para exemplificar o uso da linguagem Gcode, iremos programar a CNC para desenhar um quadrado de 10mm de lado e centro na posição (5,5) utilizando uma fresa de 3mm de diâmetro com corte por dentro. Conforme Figura 14.

Figura 14. Percurso quadrado com aresta de 10mm e centro em (5,5) em uma fresa de 3mm de diâmetro com corte por dentro, origem em (1.5,1.5)



Fonte: Autor

O código na linguagem Gcode para que a CNC faça esse percurso tem a seguinte solução possível:

```
G0X1.500Y1.500Z10.000 // Posiciona a fresa nas coordenadas (1.5,1.5,10)
G1Z5.000F3000.0 // Avança para Z=5 com velocidade de 3000 mm/min
G1X8.500Y1.500F4560.0 // Avança para (8.5,1.5,5) a uma velocidade de 4560 mm/min
Y8.500 // Avança para (8.5,8.5,5)
X1.500 // Avança para (1.5,8.5,5)
Y1.500 // Avança para (1.5,1.5,5)
G0Z10.000 // Levanta para o Z seguro da máquina
G0X0.000Y0.000 // Volta para a origem
M30 // Encerra o programa
```

Figura 15. Código escrito em Gcode para o percurso da Figura 14 – Fonte: Autor

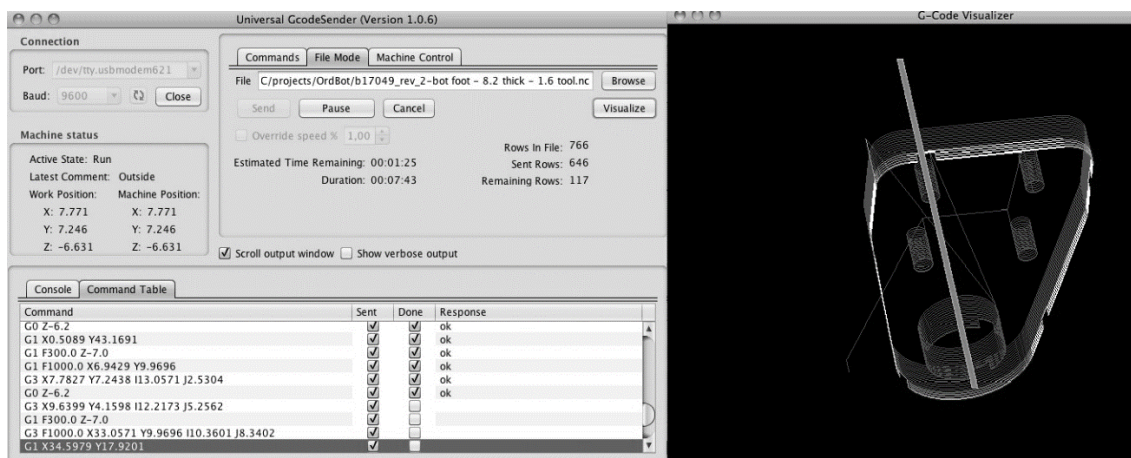
É possível perceber, na Figura 15, que as funções classificadas modais não precisam ser chamadas novamente, simplesmente chamando novos parâmetros, a máquina irá fazendo o percurso. Existem muitas funções definidas nessa linguagem, no entanto, para esse trabalho iremos nos limitar nas citadas anteriormente.

4.3 Universal G-CodeSender

Universal G-codeSender ou UGS (Winder, 2018), foi desenvolvido na plataforma Java e seu objetivo é fazer uma comunicação direta com o Grbl. Sua interface é de fácil utilização e nos dá um feedback do que está acontecendo com a CNC, conforme Figura 16.

A principal função do UGS será ler um arquivo com os códigos GCode e enviá-los para o Grbl. Uma função bem interessante implementada pelo programa é a pré-visualização do percurso efetuado pela CNC ao ler os códigos.

Figura 16. Universal G-CodeSender



Fonte Will Winder

4.4 Programa desenvolvido

4.4.1 Proposta e Requisitos

A proposta desse programa é gerar GCodes para fazer desenhos com base em imagens informadas pelo usuário. Essas imagens são transformadas em escala de cinza e a intensidade da cor de seus pixels irá representar a profundidade do corte (Figura 19, imagem da esquerda).

Quando essas imagens em escala de cinza representam um objeto em 3D elas são conhecidas como buffer de profundidade ou Z-buffer. Existem também aplicações que geram uma imagem Z-buffer a partir de um desenho em 3D.

Para uma boa qualidade nos desenhos utilizamos fresa pequena e percorremos todos os pixels da imagem e isso tem um tempo de corte muito alto. Para amenizar esse problema foi proposto um algoritmo que utiliza duas fresas com diâmetros diferentes para fazer o trabalho. Uma fresa de diâmetro maior para rápida retirada de material, e uma fresa de diâmetro menor para corte de detalhes na madeira.

Na fresa com diâmetro maior, ao invés da fresa passar por todos os pixels da imagem durante o corte, daremos “saltos” que pulam de uma única vez várias linhas da imagem, para que o tempo de corte seja reduzido. Todavia, ao darmos esses saltos, a qualidade do corte é reduzida e portanto após é necessário passar a fresa menor para retirar o restante do material.

Para que fosse possível fazer essa melhoria surgiu uma necessidade de se gerar novas imagens para se comparar o que seria o resultado de um corte com a fresa maior e o que seria o resultado de um corte com a fresa menor. Essas novas imagens chamamos de imagens de simulação de corte.

4.4.2 Imagem de Percurso

Uma fresa possui uma área de corte circular e quando posicionamos uma fresa sobre um determinado pixel de uma imagem estamos cortando uma vizinhança centrada neste pixel. Portanto, para cada pixel da imagem será analisado a intensidade de todos

os pixels vizinhos (que estão na área de corte) e definido que intensidade irá representar em uma nova imagem. Essa nova imagem gerada chamamos de imagem de percurso.

Essa imagem de percurso será gerada através da aplicação de um filtro na imagem de elevação original.

4.4.2.1 Geração da Máscara para o Filtro

Para a geração da máscara de aplicação do filtro é necessário informar o diâmetro da fresa e o tamanho em mm (sobre a superfície de corte) equivalente a um pixel da imagem. Dado o raio (r) e posição (x,y) da fresa, nós utilizamos a equação da circunferência para saber quais pixels vizinhos estarão “dentro” e portanto serão cortados pela fresa:

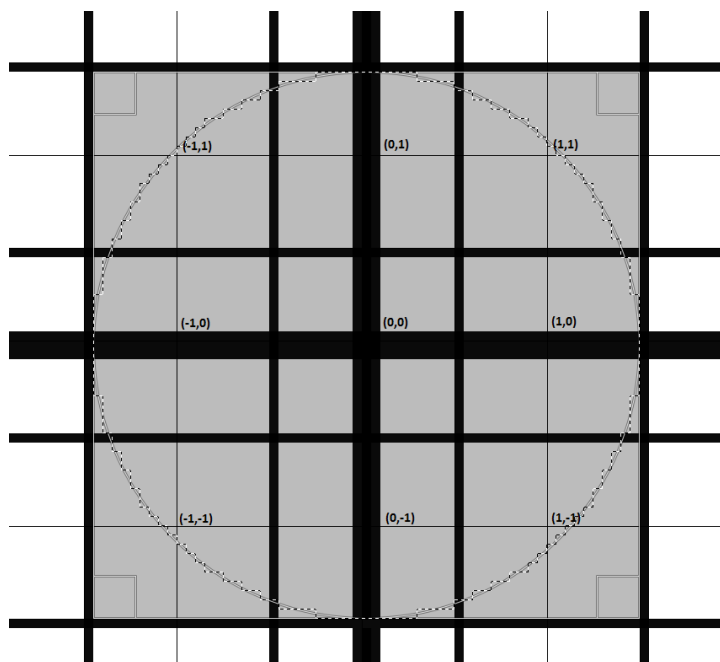
$$F(x, y) = x^2 + y^2 - r^2,$$

$$F(x, y) > 0: \text{Pixel fora da circunferência},$$

$$F(x, y) \leq 0: \text{Pixel dentro da circunferência}.$$

São analisados todos os pixels dentro de um quadrado com dimensões iguais ao diâmetro da fresa. O resultado será um conjunto de pontos no eixo cartesiano que irão corresponder a quais pixels estarão sob análise enquanto a fresa se movimentará sobre a imagem.

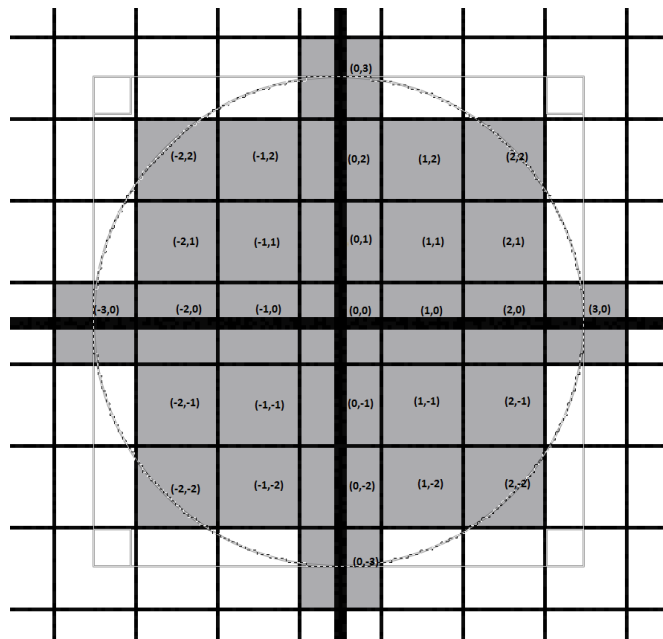
Figura 17. Máscara para a fresa de 3mm



Fonte: Autor

Na Figura 17, podemos ver quais pontos farão parte do filtro, sendo cada quadrado da imagem com lado de 0,5mm. Como a diferença entre o tamanho da fresa e o tamanho de um pixel é pequena, mais “quadriculada” será nossa representação de uma circunferência.

Figura 18. Máscara para a fresa de 6mm



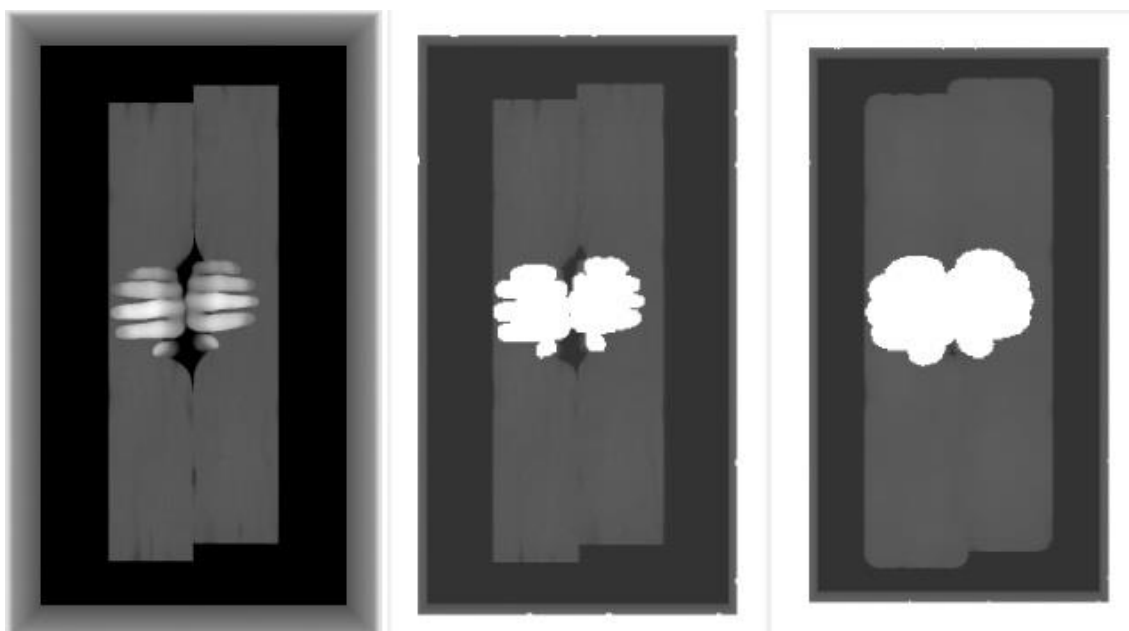
Fonte Autor

Por sua vez, na Figura 18, podemos ver essa diferença maior, pois dobrou o diâmetro da fresa. E conseguinte, mais próximo a uma circunferência será nossa representação.

4.4.2.2 Aplicação do Filtro

Para cada pixel (x,y) da imagem centramos a máscara definida na seção anterior sobre a posição do pixel. O algoritmo então busca pelo maior valor (equivalente a menor profundidade de corte) entre todos os pixels vizinhos que caem dentro da máscara, e este maior valor encontrado será colocado em uma nova imagem na respectiva posição (x,y). Essa nova imagem será chamada de imagem de percurso, conforme ilustrado na Figura 19 abaixo. Ao posicionarmos a fresa de corte com profundidade baseada nessa imagem de percurso, garantimos que não seja cortado mais material do que devido (lembre-se que a fresa possui uma área de corte circular que cobre vários pixels).

Figura 19. Na esquerda a imagem z-buffer, no centro a imagem de percurso no nível 4 (veja próxima seção) para a fresa de 3mm e na direita a mesma imagem para a fresa de 12mm



Fonte Autor

4.4.2.3 Divisão da imagem de percurso em níveis

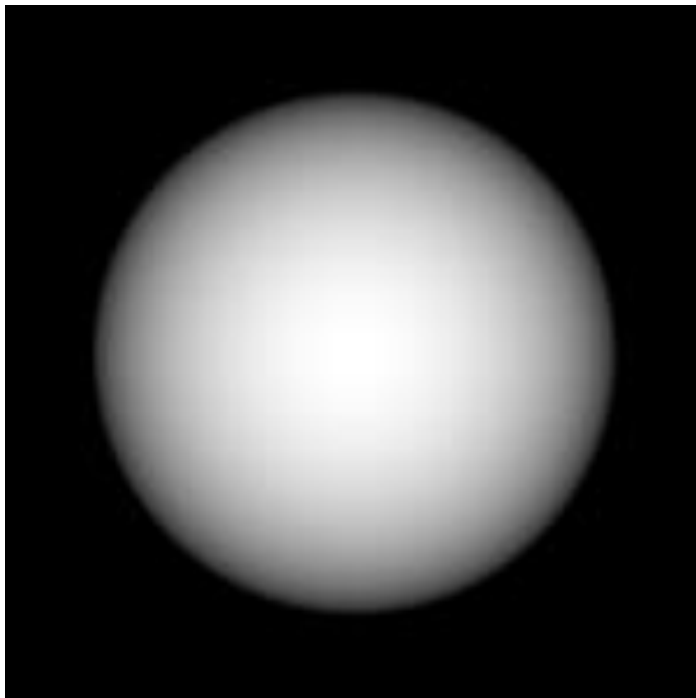
Por recomendação do fabricante da tupia a profundidade máxima de um corte não deve ultrapassar os 5 milímetros. Ou seja, para cortarmos a uma profundidade de 25 mm precisamos fazer no mínimo 5 cortes em sequência.

Para implementar essas múltiplas passadas, a partir da imagem de percurso com profundidade completa são computadas outras N imagens de percurso com profundidade reduzida, sendo N o número de níveis de profundidade que terá a imagem. Essa divisão é feita da seguinte maneira:

No primeiro nível, para todo pixel cuja profundidade for igual ou superior a 5 mm seu valor será fixado em 5mm na imagem gerada para esse nível. Quando for inferior 5 mm, o valor não será alterado.

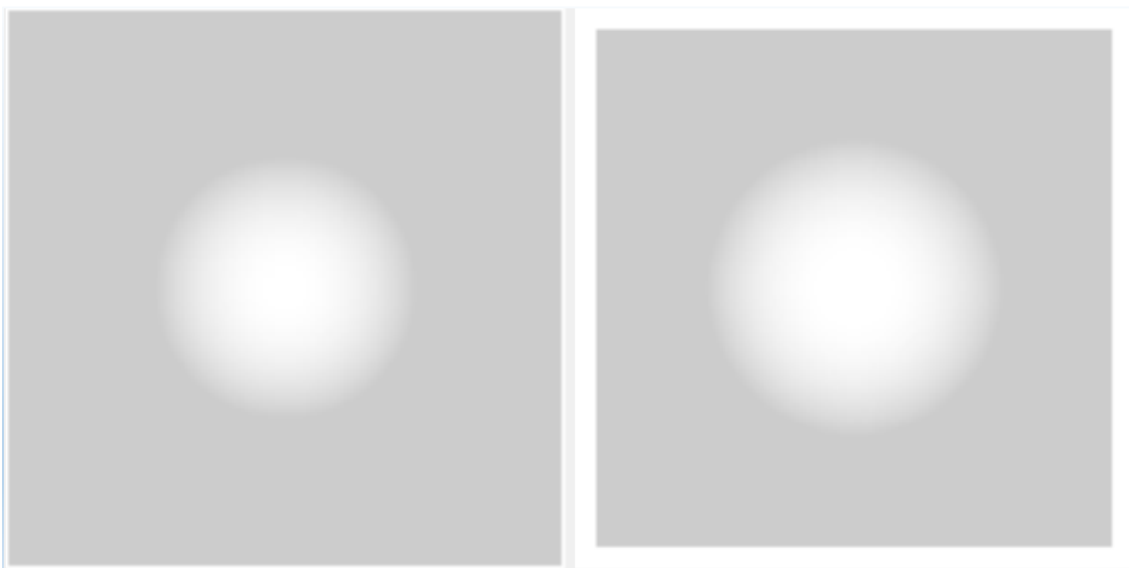
No segundo nível, será feita a mesma coisa, porém quando a profundidade for inferior a profundidade máxima do nível anterior o valor a ser colocado na nova imagem será um valor de intensidade correspondente a não cortar nada (255 na implementação atual). Isso é feito para otimizar o corte, pois a broca não passará onde não existe mais necessidade de corte. Isso pode ser visto na Figura 19, imagem central: note como os dedos das mãos estão em branco (valor 255), pois o corte relativo a estes pixels já foi realizado em níveis anteriores.

Figura 20. Imagem em z-buffer de uma esfera



Fonte Autor

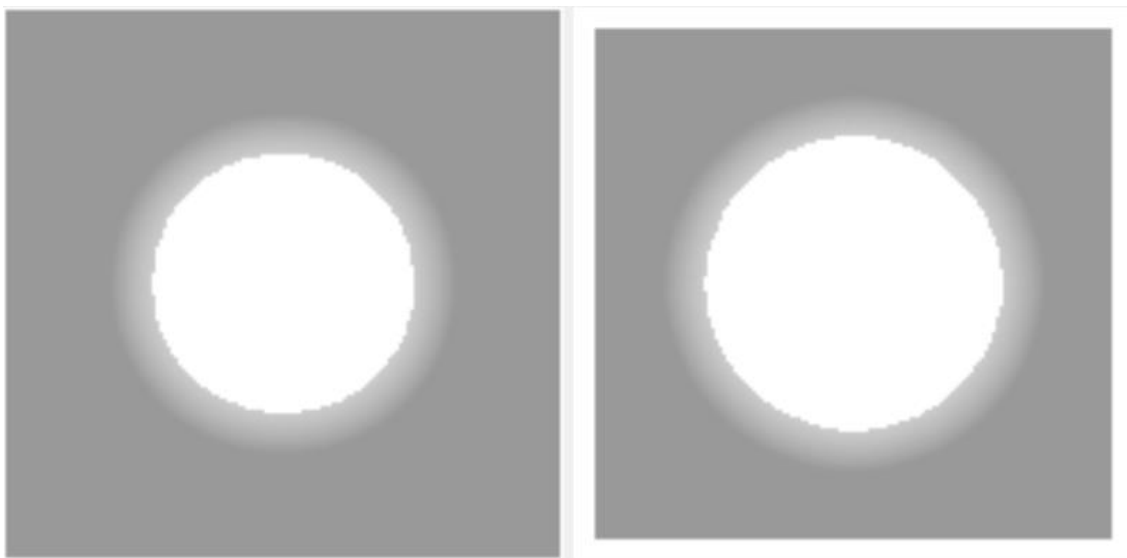
Figura 21. Imagens de percurso nível 1 para fresa de 3 mm (esquerda) e 12 mm (direita)



Fonte Autor

A Figura 21 mostra a imagem de percurso para o primeiro nível gerada para a Figura 20. Podemos perceber uma diferença nas imagens geradas pelas duas fresas. A “esfera central” é maior na imagem de percurso da fresa de 12 mm (direita) pois devido ao diâmetro da fresa ser maior a imagem deve compensar essa diferença para que o corte realizado seja igual.

Figura 22. Imagens de percurso nível 2 para as fresas de 3 e 12 mm



Fonte Autor

Comparando a Figura 21 e a Figura 22, podemos observar que a intensidade dos pixels no nível 2 é menor do que a do nível 1, o que significa que o corte é mais profundo. Podemos perceber também a área que já atingiu sua altura no nível 1 com a intensidade máxima no nível 2, isso significa que nesses locais (pixels em branco) a máquina não irá cortar e pode andar na velocidade máxima, diminuindo o tempo necessário para corte.

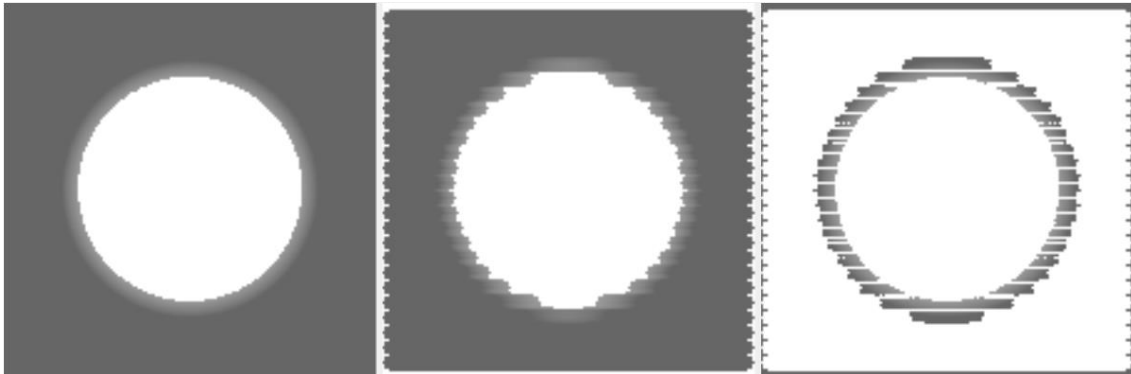
4.4.3 Imagem de Simulação de Corte

A imagem de simulação de corte representa como ficará o corte após a passagem da fresa pela imagem de percurso. Essa imagem de simulação é gerada com o intuito de comparar as imagens de corte da fresa maior e da fresa menor. Essa comparação é essencial dado que a diferença entre as duas imagens representa o que a fresa maior não conseguiu cortar e, conseqüentemente, o que precisa ser cortado pela fresa menor (Figura 23).

O programa irá percorrer todos os pixels, linha a linha, da esquerda para a direita em números de linhas pares, e da direita para a esquerda em número de linhas ímpares. Existe a possibilidade de se fazer “saltos” maiores, modificando a variável passo no programa. O passo indica de quantas em quantas linhas irá passar a fresa e quanto maior o passo mais material é retirado por vez em cada linha percorrida, e menor o tempo total de corte.

A parte mais complexa é a geração da imagem de simulação de corte quando o passo é maior do que 1 mm, ou seja, a fresa não passa por todos os pixels. Sendo assim, para saber a intensidade de um determinado pixel é necessário fazer uma conferência entre os pixels vizinhos somente de onde a fresa passou.

Figura 23. Imagens de simulação de corte: diâmetro de 3 mm com passo de 1 mm, diâmetro de 6 mm com passo de 6 mm e diferença entre os dois cortes



Fonte Autor

Ao fazermos essas simulações de corte pode ocorrer algum erro de arredondamento na intensidade de alguns pixels e esse erro poderá aparecer como uma diferença nas duas imagens de corte onde não deveria haver diferença. Para solucionar esse problema foi acrescentado um parâmetro que é o limiar aceitável de erro dos filtros. A diferença do valor do pixel de uma imagem para outra só será considerada se esse valor for superior ao limiar informado pelo usuário. Por exemplo, se o limiar de erro do filtro estiver como 0,5%, o algoritmo não irá considerar como diferentes dois pixels cujos valores são 254 e 255, pois a diferença de valor, igual a 1 unidade dentro de um intervalo de [0,255], é menor que 0,5% da largura do intervalo.

4.4.4 Geração do código GCode

Para a geração código Gcode é feito uma varredura nas imagens de percurso e gerado códigos do estilo: G0X – coordenada X – Y- coordenada Y – Z – profundidade do corte. Para cada pixel, será gerado um código G0 ou G1 indicando a coordenada X e Y desse pixel e a profundidade será calculada pela intensidade do pixel. Assim como representado nas Figura 14 e Figura 15.

Basicamente foram utilizadas duas instruções: a G0 e a G1. A instrução G0 representa o deslocamento da máquina quando não está fazendo nenhum corte, portanto ela atinge a velocidade máxima de deslocamento da máquina. Por sua vez, a G1 é a que representa o corte e para isso ela necessita que informe a velocidade.

Para cada tamanho de fresa devemos informar uma velocidade de deslocamento diferente. Fresas com menor diâmetro trabalham com menos rotação e podem operar com velocidade de deslocamento maior por retirar menos material. Já as fresas maiores trabalham em alta rotação e portanto velocidade de deslocamento menor, tirando grandes quantidades de material por vez, e forçam mais a CNC.

A profundidade dos cortes é uma variável obtida através da intensidade dos pixels da imagem de percurso com a seguinte formula:

$$Profundidade = \frac{(255 - I_{pixel})}{255} * Profundidade_{Max}.$$

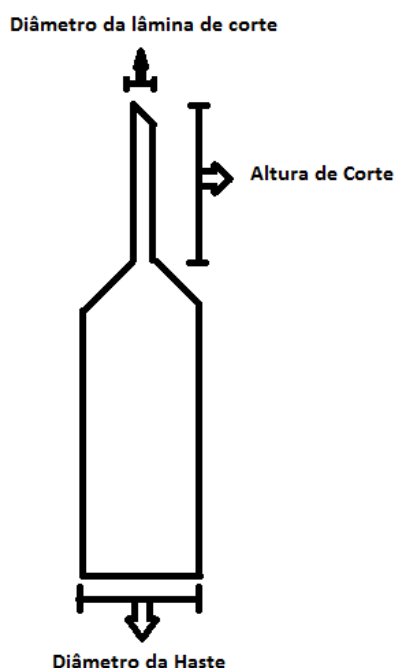
Sendo I_{pixel} a intensidade do pixel e $Profundidade_max$ a profundidade máxima que a máquina irá cortar quando o valor de I_{pixel} for zero. A $Profundidade_max$ será definida pelo usuário do programa.

Para o processo de desbaste (retirada de grande quantidade de material rapidamente) é feito uma varredura linear sobre o MDF usando uma fresa grande com passos grandes. O passo utilizado no desbaste pode ser no máximo igual ao diâmetro da fresa. Quanto maior for o passo, menor será o tempo para o desbaste; no entanto, maior será a diferença das imagens de uma fresa para outra e maior será o tempo de corte da fresa de diâmetro menor.

4.4.5 Detector de colisão

O detector de colisão foi uma necessidade que surgiu durante o projeto quando utilizamos uma fresa que tem o diâmetro da lâmina de corte menor que o diâmetro da haste, ver Figura 24.

Figura 24. Representação de uma fresa com diâmetro da lâmina de corte menor do que a haste



Fonte Autor

Para exemplificar o problema: vamos supor que a CNC tem que fazer um furo com profundidade maior que a altura de corte e esse furo tiver o diâmetro menor do que o diâmetro da haste. O programa irá aceitar o corte, todavia chegará um certo momento em que a haste colidirá com o material vizinho ao corte e poderá danificar a CNC.

Para evitar esse problema, o detector de colisão irá verificar a vizinhança de cada pixel e se houver vizinhos com profundidade maior que a altura de corte ele dará uma mensagem de alerta de colisão.

Se a haste tiver diâmetro igual ou inferior ao diâmetro da lâmina de corte, o detector de colisão não será necessário visto que não é possível que haja alguma colisão no corte.

5. RESULTADOS OBTIDOS

Para exemplificar o funcionamento do programa e seu resultado será apresentado uma sequência de ações desde a inserção da imagem até a geração do código GCode.

Figura 25. Interface do Programa Gerador Gcode desenvolvido

The screenshot shows a Windows-style window titled 'Form1'. It contains a series of input fields and buttons. On the left, there are labels for various parameters: 'Arquivo Imagem', 'Diâmetro da fresa', 'Tamanho da imagem X', 'Tamanho da imagem Y', 'Profundidade', 'Profundidade de corte', 'Pixel por mm', 'Passo em X e Y', a checkbox for 'Usar 2 fresas', 'Diâmetro da segunda fresa', 'Profundidade de corte' (repeated), 'Passo', and '% Erro dos filtros'. To the right of these labels are corresponding text input boxes. Some boxes contain pre-filled values: '3' for 'Diâmetro da fresa', '25' for 'Profundidade' and 'Profundidade de corte', '1' for 'Pixel por mm' and 'Passo em X e Y', '6' for 'Diâmetro da segunda fresa' and 'Passo', and '0.5' for '% Erro dos filtros'. A 'Procurar Img.' button is located to the right of the 'Arquivo Imagem' field. Below the input fields, there is a 'Gerar G-Code' button. At the bottom of the window, there is a large, empty rectangular area, likely intended for displaying the generated G-code or a visual representation of the process.

Fonte Autor

Primeiramente o usuário irá buscar um arquivo de imagem localizada no disco através do botão “Procurar Img” (Figura 25). Ao selecionar o arquivo os campos Tamanho da imagem X e Tamanho da imagem Y serão preenchidos com a quantidade de pixels em cada eixo da imagem. Esse valor irá representar o tamanho real da imagem em milímetros em cada eixo se cada pixel representar 1 milímetro na superfície de corte.

Em seguida, o usuário informa o diâmetro da fresa em milímetros e a profundidade máxima que representa o valor de intensidade zero no pixel da imagem. Se a fresa utilizada possuir o diâmetro de corte menor do que o diâmetro da haste deverá ser informado o valor da altura de corte da fresa em mm no campo Profundidade de corte para que passe pela conferência do detector de colisão. No entanto, se a fresa possuir a lâmina de corte com diâmetro igual ou superior ao diâmetro da haste não haverá colisão e esse valor deve ser o mesmo, ou superior, ao do campo profundidade.

No campo pixel por mm, o usuário deve informar o tamanho que um pixel da imagem representa no corte da máquina. Por exemplo, se utilizarmos o valor 1 pixel por mm cada pixel na imagem irá representar 1 mm no corte, ou seja, uma imagem com resolução de 150x150 pixels irá fazer um corte com tamanho de 150x150 milímetros na madeira.

O campo Passo em X e Y, deverá conter o tamanho do salto em mm que a fresa irá saltar a cada mudança de linha. Por exemplo, se esse valor estiver configurado com o valor 1 e cada pixel representar 1 mm será como se a fresa passasse por todos os pixels da imagem. Entretanto, se esse valor for maior a fresa dará saltos maiores na madeira e consequentemente tirará mais quantidade de material sob pena de não possuir uma resolução tão boa.

Se habilitar a caixa nomeada de Usar duas fresas, o programa irá gerar códigos para duas fresas diferentes. Essa função foi desenvolvida com o intuito de diminuir o tempo de corte dos desenhos. A ideia é utilizar uma fresa com diâmetro grande e passos largos para tirar grande quantidade de material e, logo em seguida, trocar por uma fresa menor com passos pequenos para fazer os detalhes da imagem e retirar o restante de material da madeira.

No campo erro dos filtros, o usuário informa o percentual (limiar) de erro dos filtros. Caso algum pixel esteja com valor adulterado por algum erro de arredondamento nas conversões das imagens de percurso para corte esse percentual irá ignorar essa diferença.

5.1 Exemplo com imagem de uma meia esfera

Para testes na máquina foi utilizado como entrada a seguinte imagem:

Figura 26. Esfera no formato z-buffer



Fonte Autor

Essa esfera possui um tamanho de 150x150 pixels e foram feitos dois tipos de cortes: (1) utilizando somente uma fresa com diâmetro e passos pequenos ou (2) utilizando duas fresas, inicialmente uma com diâmetro maior e passos mais largos para retirada rápida de material, seguida de outra com diâmetro e passos pequenos, para corte de detalhes.

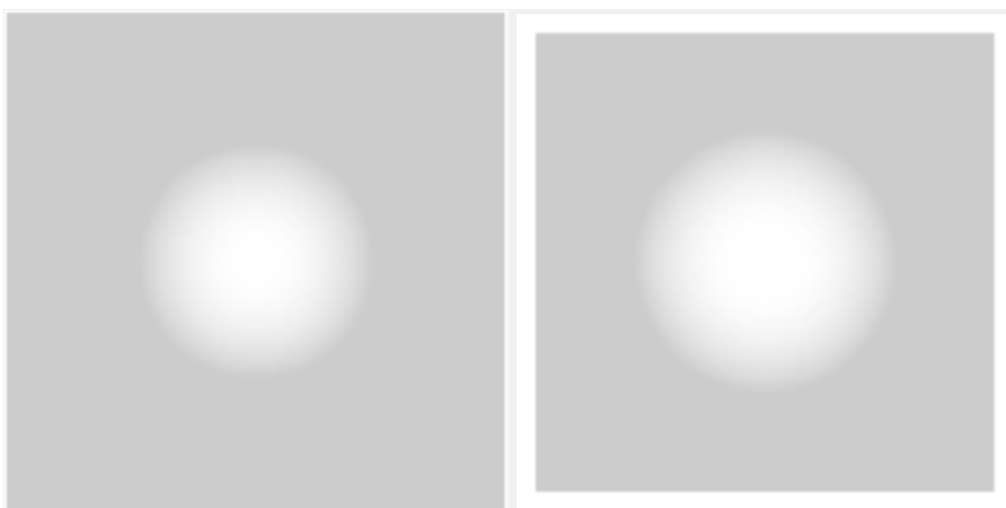
Para esse corte, serão utilizados duas fresas: uma de 3 mm e 12 mm de diâmetro com passo de 1 mm para a fresa menor e passos de 5 mm para a fresa maior e uma profundidade total de corte de 25 milímetros.

5.2 Imagens dos percursos

Após a seleção da imagem, configurar os parâmetros e clicar no botão Gerar Gcode o programa irá dividir a imagem em N imagens, sendo N o valor da profundidade dividido por 5. Isso é feito para que a máquina não faça cortes com mais de 5 milímetros de profundidade de cada vez. Ou seja, para esse desenho cuja profundidade foi definida com valor de 25, o valor de N será 5.

Após a divisão da imagem em 5 imagens com “alturas” diferentes, será gerado uma imagem de percurso para cada imagem e para cada fresa.

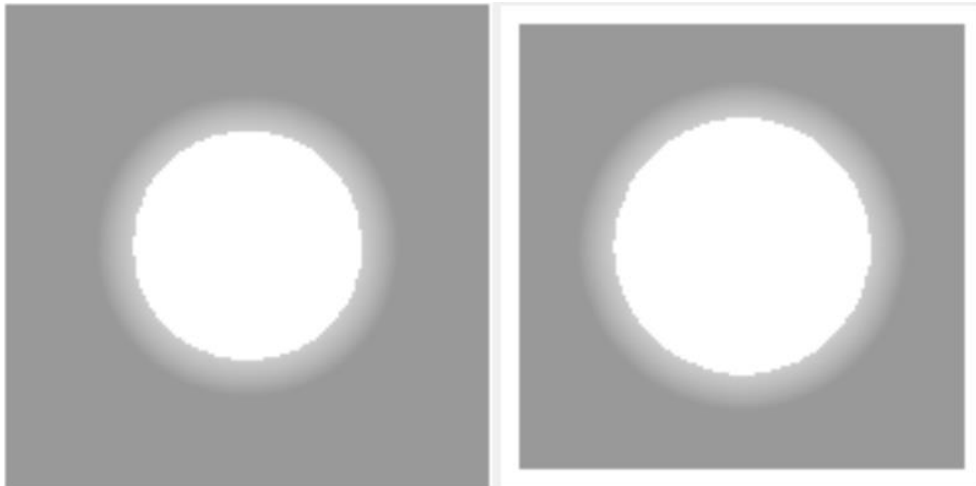
Figura 27. Imagem de percurso nível 1



Fonte Autor

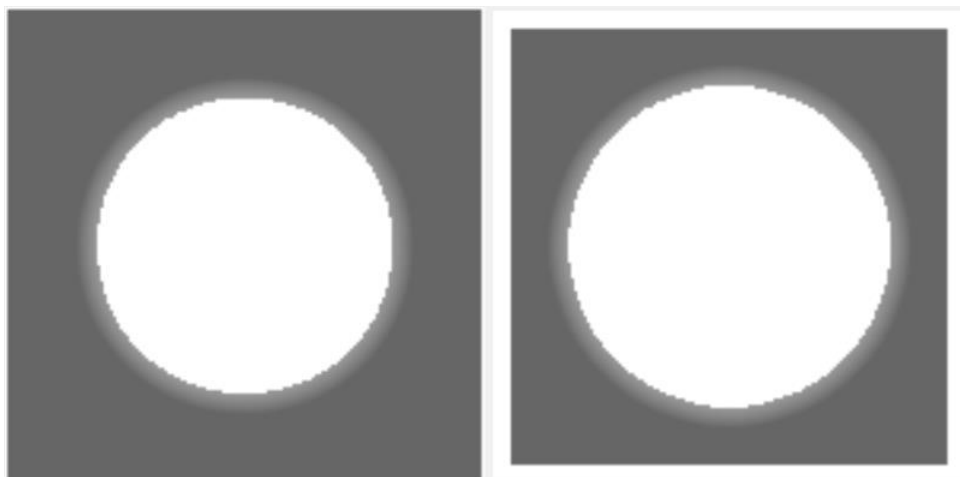
Na Figura 27, temos do lado esquerdo a imagem representando o percurso de nível 1 da fresa de 3 mm e do lado direito a fresa de 12 mm de diâmetro. A intensidade mais “escura” desse nível possui o valor correspondente a 5 mm de profundidade na madeira.

Figura 28. Imagem de percurso nível 2



Fonte Autor

Figura 29. Imagem de percurso nível 3



Fonte Autor

Figura 30. Imagem de percurso nível 4



Fonte Autor

Figura 31. Imagem de percurso nível 5



Fonte Autor

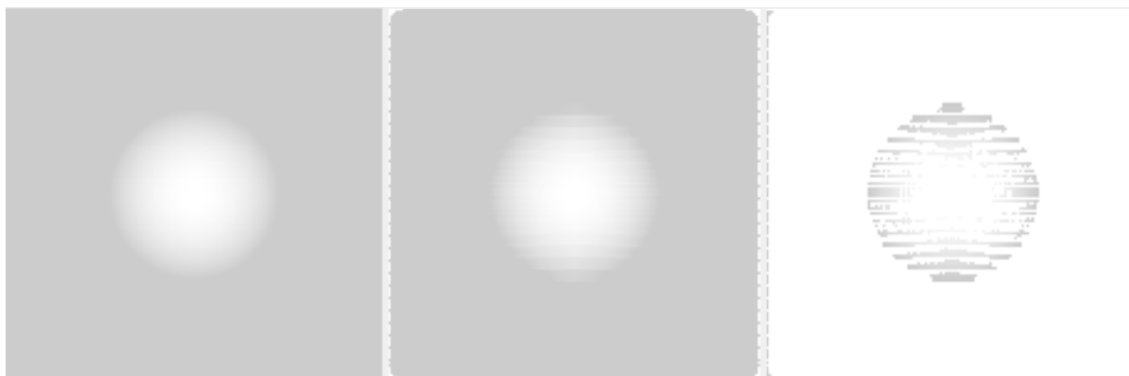
Na Figura 28, Figura 29, Figura 30 e Figura 31 as intensidades dos pixels representam um máximo de 10, 15, 20 e 25 mm de profundidade na madeira, respectivamente. Dessa forma o corte é realizado em 5 etapas de profundidade, respeitando o limite físico de profundidade máxima de corte.

5.3 Imagens dos cortes

Na sequência de ações tomadas pelo programa são geradas as imagens de simulação de corte para cada uma das imagens de percurso. Na esquerda será representado o corte da fresa de 3mm, no meio será o corte da fresa de 12 mm e na direita será a diferença entre os dois cortes. Essa imagem de diferença é utilizada para que o corte com a fresa de 3mm percorra somente regiões que não foram cortadas pela fresa de 12mm, acelerando assim a velocidade de corte.

Podemos perceber que na imagem de corte gerada para a fresa de 12 mm há alguns “degraus” na esfera, isso é esperado e acontece por causa dos passos maiores, igual a 5 mm, na fresa maior.

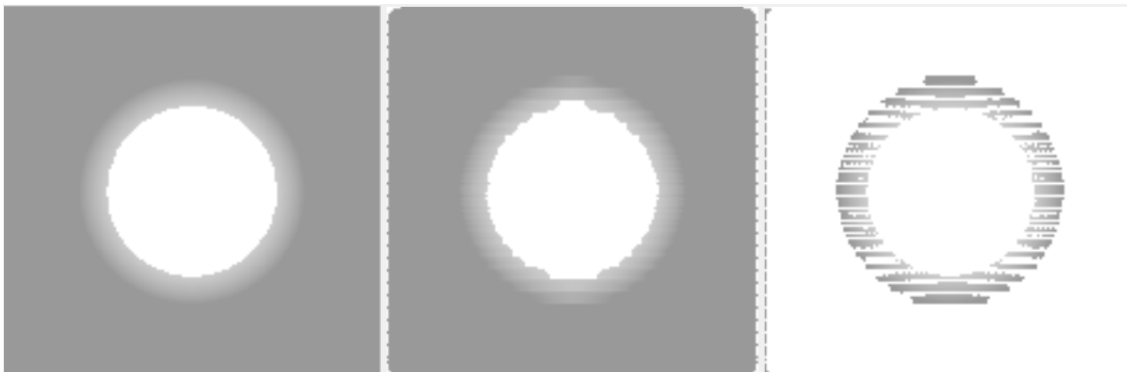
Figura 32. Imagem de simulação de corte nível 1



Fonte Autor

A partir da Figura 32, até a Figura 36, podemos ver a sequência de imagens de corte separadas por nível, de corte mais superficial até a mais profunda.

Figura 33. Imagem de simulação de corte nível 2



Fonte Autor

Figura 34. Imagem de simulação de corte nível 3



Fonte Autor

Figura 35. Imagem de simulação de corte nível 4



Fonte Autor

Figura 36. Imagem de simulação de corte nível 5



Fonte Autor

5.4 Processamento da diferença de imagens de corte

Com a imagem da diferença de cortes entre as duas fresas, foi utilizado uma função da biblioteca OpenCV chamada *ConnectedComponentsWithStats* que recebe como parâmetro uma imagem e nos retorna um vetor de blocos contíguos de pixels que compõe a imagem. Dessa forma as imagens das diferenças de corte não são mais contínuas, ou seja, se tornaram blocos. A função dessa biblioteca nos dá a possibilidade de fazer cortes por blocos e não mais de maneira totalmente sequencial como antes.

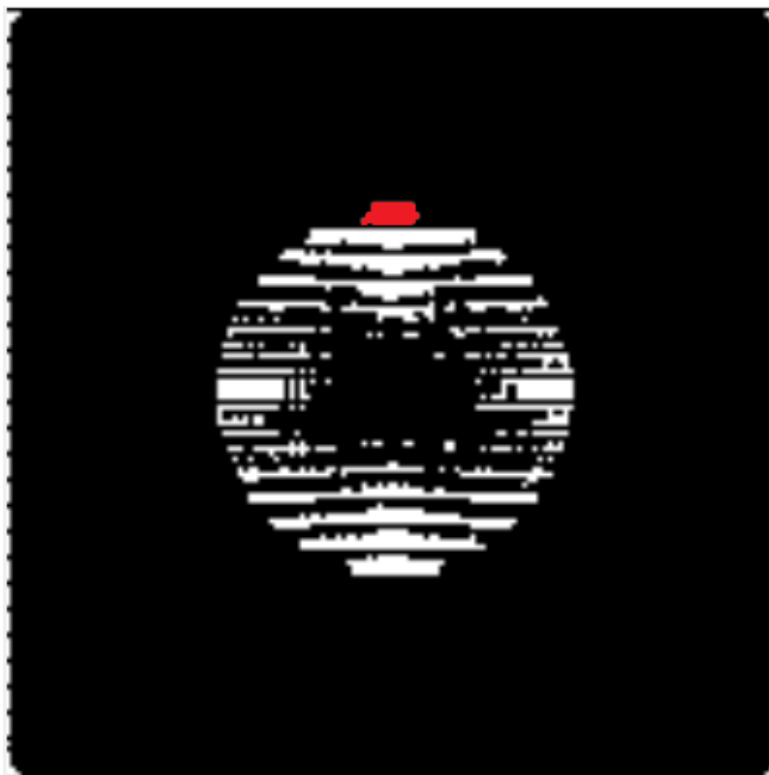
Desse modo, a CNC irá trabalhar com cada bloco individualmente e não irá varrer toda a imagem, tornando assim mais eficiente e mais rápido o corte. Podemos ver um dos blocos da imagem da Figura 37 na Figura 38, pintado de vermelho.

Figura 37. Imagem gerada pela diferença de corte das fresas no primeiro nível



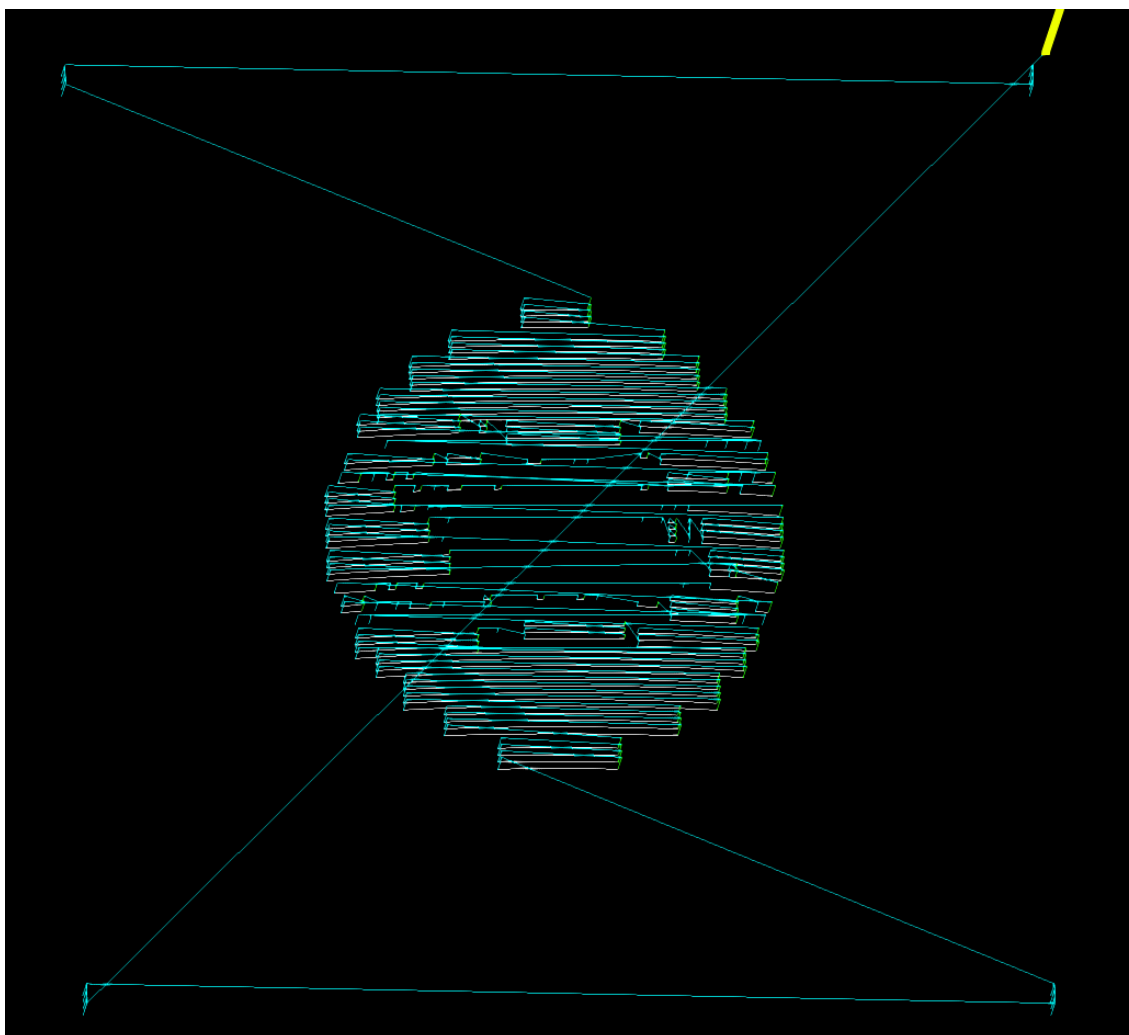
Fonte Autor

Figura 38. Imagem com um bloco pintado de vermelho



Fonte Autor

Figura 39. Pré-visualização gerada pelo G-CodeSender ao inserir o Gcode gerado para a Figura 37



Fonte Autor

Na Figura 37, vemos a imagem gerada pela diferença de cortes das fresas menor e maior e, na Figura 39, temos uma pré-visualização fornecida pelo G-CodeSender que mostra uma simulação do caminho tomado pela fresa.

5.5 Enviando o Gcode para a CNC

Com os códigos Gcode gerados para todos os níveis primeiramente utilizamos o que foi chamado de Segunda_fresa1 que é o código para o primeiro nível da segunda fresa (a maior).

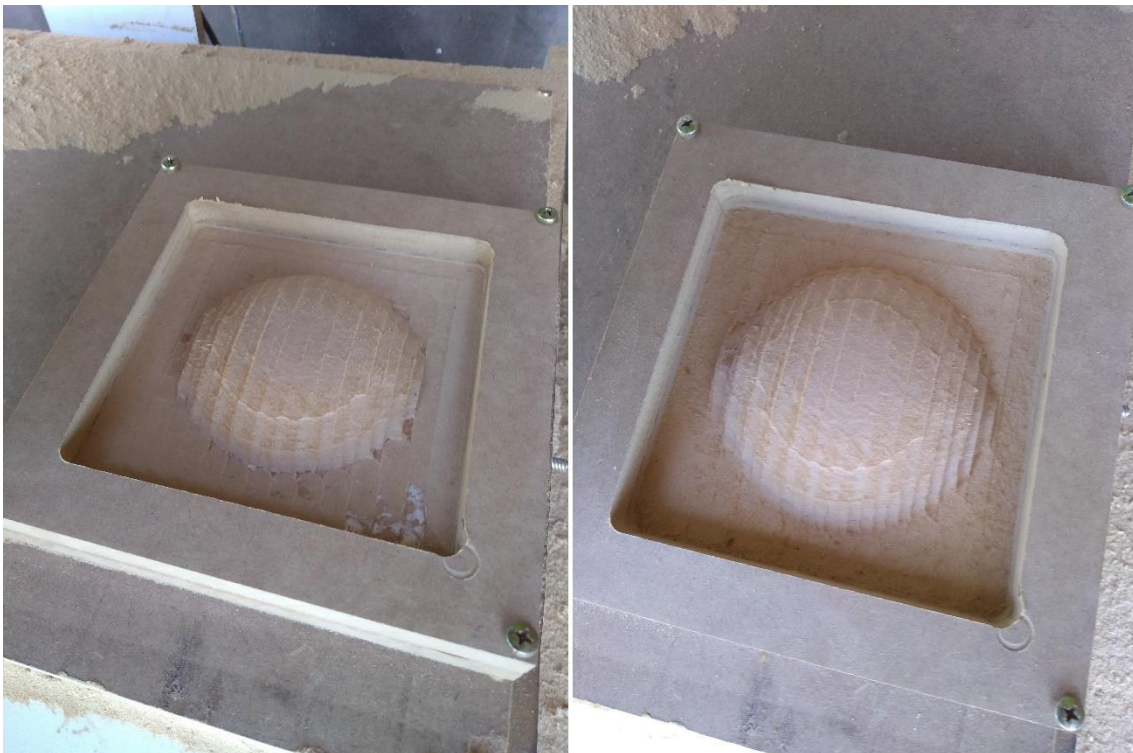
Esse código então é enviado para o Grbl no Arduino através do software G-CodeSender e irá começar a movimentar a fresa e começar a cortar a madeira.

Figura 40. Imagens dos cortes nos níveis 1 (esquerda) e 2 (direita) com a fresa de 12 mm e passo de 5 mm



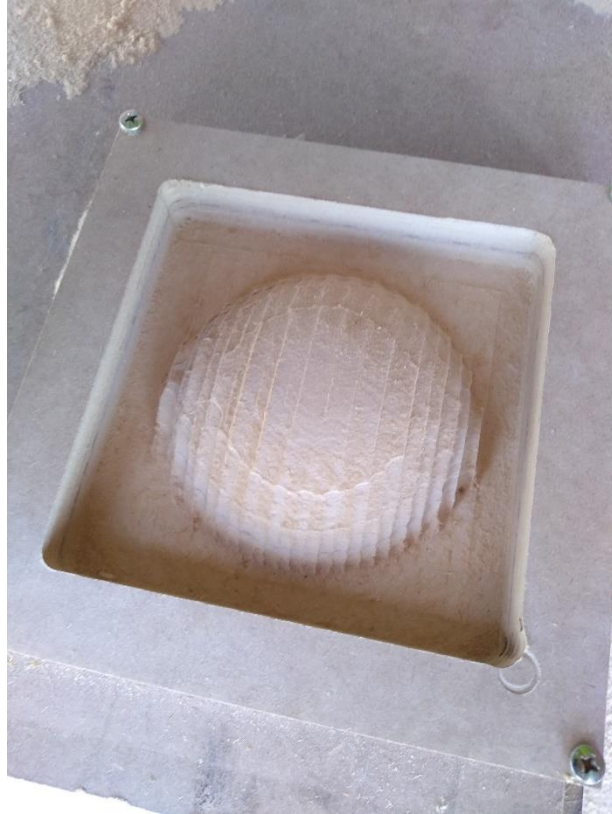
Fonte Autor

Figura 41. Imagens dos cortes nos níveis 3 (esquerda) e 4 (direita) com a fresa de 12 mm e passo de 5 mm



Fonte Autor

Figura 42. Imagem do corte no nível 5 com a fresa de 12 mm e passo de 5 mm



Fonte Autor

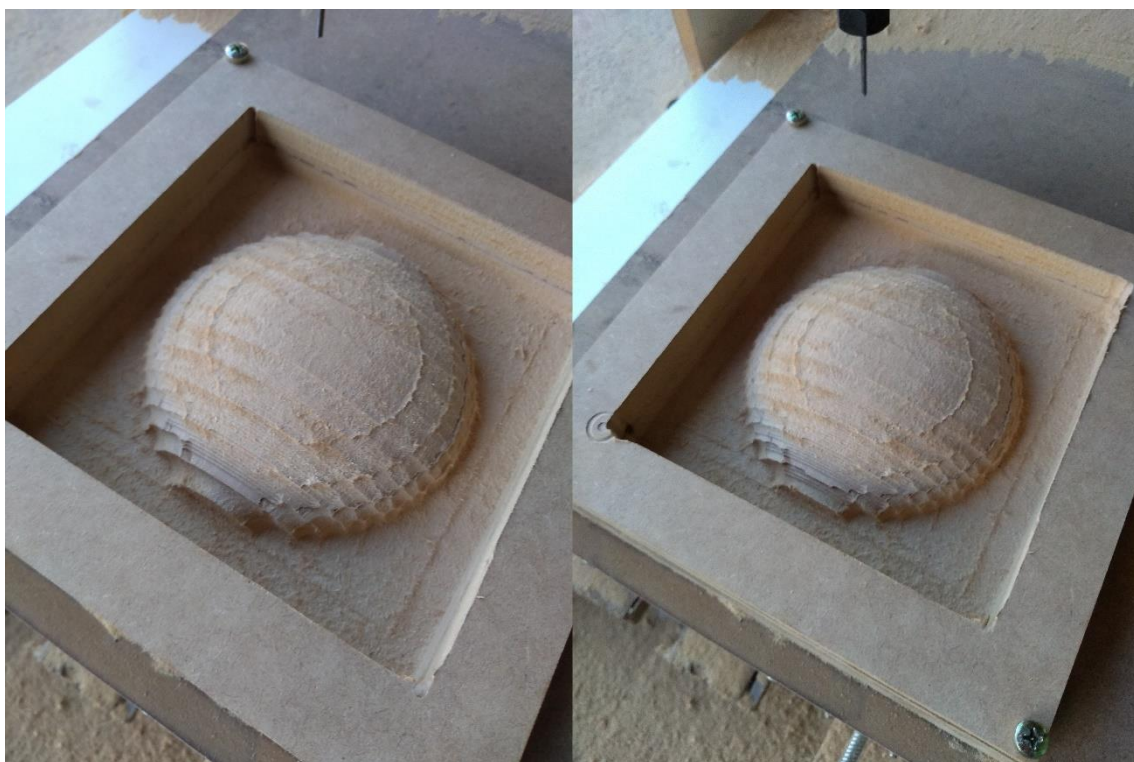
A sequência de imagens: Figura 40, Figura 41 e Figura 42 mostram o resultado da execução dos códigos Gcode na CNC. O próximo passo é passar a fresa pequena para que tire o material que a fresa maior não consegue. Por exemplo, se houver um furo pequeno no centro da esfera, ao passar o filtro da fresa maior esse furo não será feito pois nos pixels onde há esse furo sempre haverá um vizinho dentro do diâmetro da fresa que irá possuir uma altura que não é a do furo e consequentemente o furo não irá aparecer na imagem de percurso da fresa grande. No entanto, esse furo irá aparecer na imagem de percurso da fresa pequena, ver novamente a Figura 19.

Figura 43. Imagens dos cortes nos níveis 1 (esquerda) e 2 (direita) com a fresa de 3 mm e passo de 1 mm cortando por blocos



Fonte Autor

Figura 44. Imagens dos cortes nos níveis 3 (esquerda) e 4 (direita) com a fresa de 3 mm e passo de 1 mm cortando por blocos



Fonte Autor

Figura 45. Imagem do corte no nível 5 com a fresa de 3 mm e passo de 1 mm cortando por blocos



Fonte Autor

Podemos perceber na Figura 43, Figura 44 e Figura 45 que após passar a fresa menor nas diferenças entre as imagens de corte ainda não está com o acabamento bom pois ao cortar os blocos ainda há imperfeições na esfera. Essas imperfeições serão corrigidas na última passagem, ver sessão 5.6.

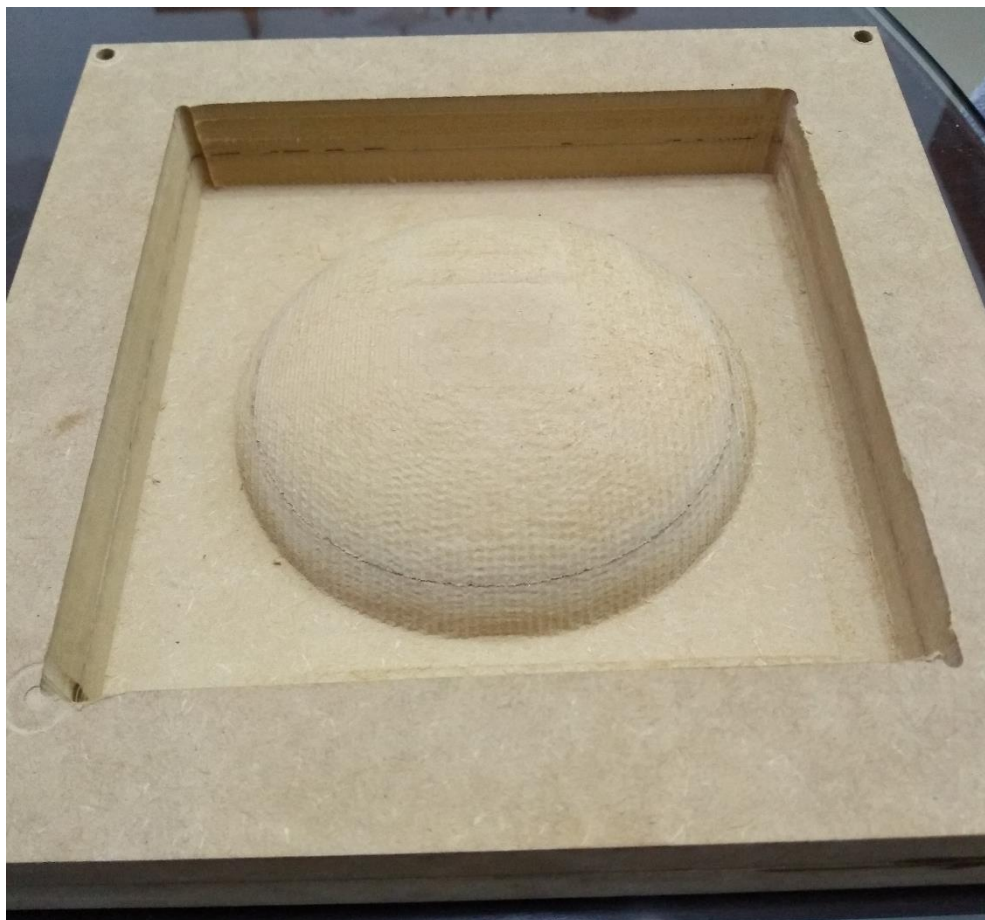
No entanto, essas múltiplas passagens com a fresa menor são necessárias porque se houver locais onde a fresa maior não consiga chegar e a fresa menor consiga, haverá locais onde a fresa menor terá que tirar muito material de uma única vez na última passagem.

5.6 A última passagem para corrigir imperfeições

Na sequência de ações o próximo passo é fazer um último corte na outra direção (invés da varredura ser feita de linha em linha ela será feita de coluna em coluna, a máquina inverterá o eixo que vai percorrer), com a fresa pequena, percorrendo todo o desenho não necessitando mais dividir a imagem em níveis, ou seja, a CNC irá passar por todos os pontos da imagem em uma única vez.

Essa última passagem é a mais demorada e corrige a maior parte das imperfeições do desenho, ver Figura 46.

Figura 46. Resultado final após a última passagem



Fonte Autor

5.7 Análise do tempo de corte

Analizando visualmente o resultado de corte usando somente uma fresa pequena e comparando com o resultado usando duas fresas praticamente não há diferença. O desenho cortado por duas fresas teve uma qualidade tão boa quanto o cortado somente com uma fresa.

No que diz respeito ao tempo de corte, o tempo de corte utilizando duas fresas foi menor do que quando usamos somente uma. O tempo total de corte para o sistema de duas fresas foi de 1 hora, 53 minutos e 50 segundos. Já o tempo para fazer o mesmo desenho utilizando somente a fresa pequena foi de 3 horas.

Portanto, o aperfeiçoamento da programação para possibilitar o uso de 2 fresas nos deu uma redução de 1/3 no tempo total de corte, o que se mostrou satisfatório visto que esses cortes levam um tempo considerável para serem feitos.

5.8 Criando xilogravuras

Uma outra aplicação que a CNC é capaz de fazer são xilogravuras.

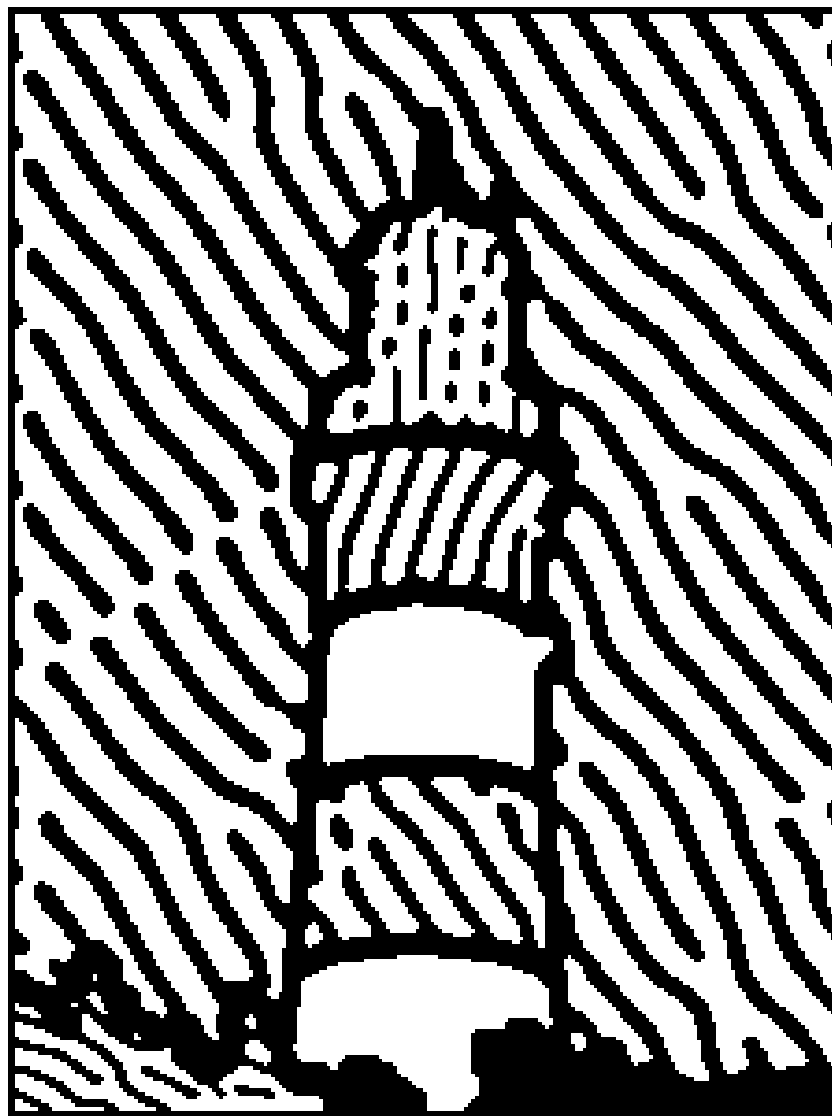
A xilogravura é uma forma tradicional de arte onde passa-se tinta na superfície de um bloco de madeira entalhado que será usado como superfície de impressão sobre uma folha de papel, de modo que apenas as partes que não foram entalhadas serão impressas no papel (Mesquita & Walter, 2019).

Como entrada para o programa utilizamos um dos desenhos gerados pelo trabalho de mestrado do autor Davi Mesquita, que gera xilogravuras a partir de imagens reais.

Na Figura 47, vemos a xilogravura de uma imagem onde há um farol em uma paisagem. E na Figura 48, temos o corte realizado pelo algoritmo proposto da xilogravura na CNC com uma fresa de 3mm de diâmetro usando passo de 1mm e uma profundidade de 2 mm.

O tempo de corte da xilogravura foi de uma hora e gerou um desenho na madeira de 30 cm de altura por 22 cm de largura. Podemos notar que existem pontos onde não foi cortado e isso acontece por que as linhas na imagem eram mais finas do que o diâmetro da fresa e só seriam cortadas se tivesse uma fresa menor do que 3 mm ou se a imagem tivesse um tamanho maior.

Figura 47. Imagem de uma xilogravura



Fonte: Mesquita e Walter 2019

Figura 48. Corte de xilogravura usando uma fresa de 3 mm



Fonte Autor

6. MELHORIAS FUTURAS

Existem muitas melhorias que podem aperfeiçoar o tempo e a qualidade dos cortes na CNC:

- Fazer novas peças para a CNC usando a própria CNC. Como não havia furadeira de bancada e outros instrumentos para fazer as peças da máquina de uma maneira mais precisa, ficaram alguns desalinhamentos entre os eixos o que provoca um certo atrito a mais nos eixos e consequentemente diminui a velocidade de movimento.
- Trocar a mesa de sacrifício (parte sobre a qual é fixado a madeira a ser cortada) por uma chapa de diâmetro maior. Como a mesa de sacrifício foi feita da mesma chapa de mdf que os outros componentes da CNC, ela possui um diâmetro inadequado. Isso fez com que a mesa de sacrifício

sofresse uma deformação no centro o que interfere nas peças que serão cortadas pois estão apoiadas sobre ela.

- Otimização do código GCode. Existem ainda trechos de código Gcode que são gerados desnecessariamente, o que pode provocar uma piora no tempo de corte.
- Fazer um sistema de calibragem de altura. No atual estado da máquina, a calibragem de altura zero é feita de forma manual utilizando-se um multímetro na função de curto-circuito e uma chapa metálica de 2 mm. A fresa então é descida até tocar a chapa metálica e fechar o curto-circuito com o multímetro. O novo sistema seria ligar a chapa metálica no Arduino e conectar a fresa também ao Arduino e fazer a calibragem de altura de forma automática.
- Colocar um sistema de soprador de ar próximo à fresa para que remova os “cavacos” da madeira. Quando fazemos um corte, o material depositado por onde passou a fresa vai se acumulando e provoca um superaquecimento da fresa e, por conseguinte, a perda do fio da lâmina.

7. CONCLUSÕES

O trabalho feito demonstra um passo-a-passo para qualquer pessoa a fazer sua própria CNC e consiga operá-la de forma intuitiva e funcional. Todo material foi feito com produtos nacionais e foram utilizadas ferramentas comuns para a montagem, ou seja, somente com as ferramentas que o autor possuía em casa.

Primeiramente o software desenvolvido utilizava somente uma fresa e tinha um tempo de corte bem demorado. Então foram feitas propostas para se otimizar esse tempo como mudar a forma de varredura da imagem ou utilizarmos mais de uma fresa.

Foi escolhido então utilizarmos duas fresas. O nível de complexidade no desenvolvimento do algoritmo aumentou substancialmente e não sabíamos se essa melhoria iria reduzir o tempo do corte. No entanto, ao concluir o projeto, percebemos uma boa melhora no tempo e uma qualidade tão boa quanto antes.

Foram feitos muitos cortes errados e a cada tentativa íamos corrigindo o código até chegar ao resultado apresentado, que se mostrou satisfatório pois a esfera ficou com uma boa simetria e com qualidade visual boa.

Mesmo não utilizando ferramentas como a furadeira de bancada, a CNC ficou com um bom alinhamento e precisão como podemos observar no corte da esfera. A velocidade de 12000 mm/min foi outro ponto positivo porque ficou próximo a velocidade de 15000 mm/min que possuem algumas CNCs construídas em metal disponíveis no mercado.

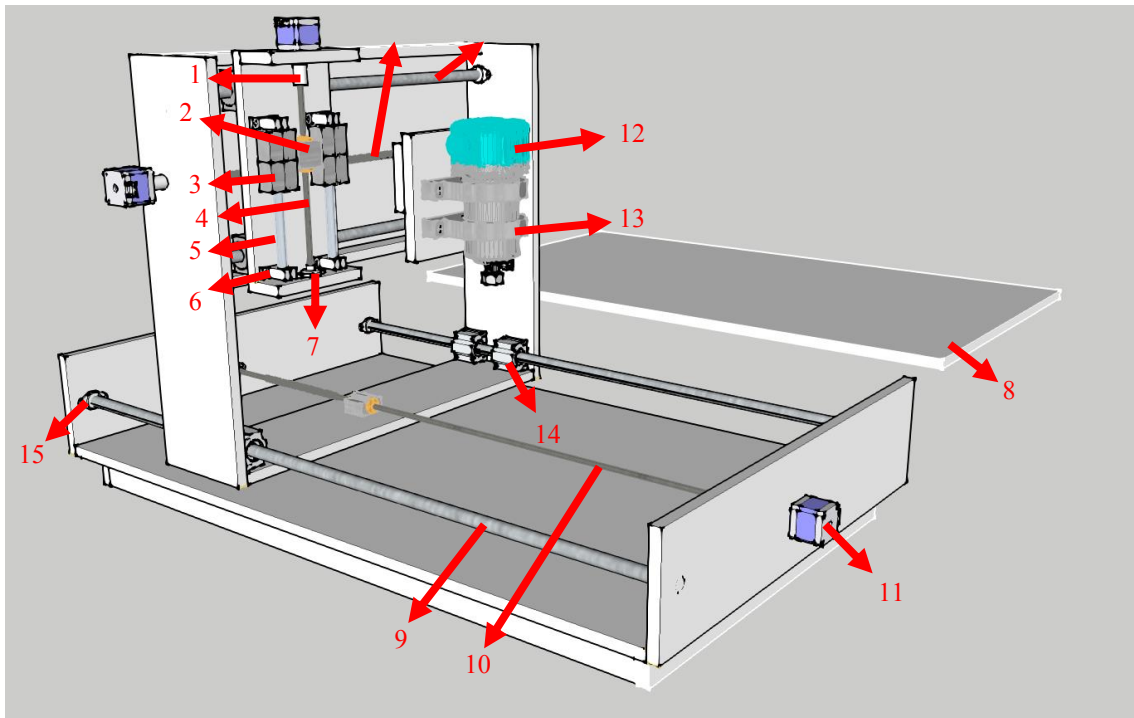
O custo total do projeto foi de 2.750,00 reais e o valor de uma máquina com a mesma área útil é em média de 15.000 reais.

Em síntese, por mais que pareça fácil a produção de uma CNC, existem muitas dificuldades que irão aparecer no decorrer do desenvolvimento. No entanto, essa monografia será de grande ajuda para qualquer pessoa que queira montar sua própria CNC ou se interesse pelo assunto. Todo o projeto será disponibilizado no site: https://github.com/ijbarros/CNC_INF_UFRGS

REFERÊNCIAS

- Autodesk, Inc. (26 de Outubro de 2018). Fonte: Titans of CNC Academy: http://academy.titansofcnc.com/files/Fundamentals_of_CNC_Machining.pdf
- Azevedo, A. L. (26 de outubro de 2018). *Os Primórdios do controle numérico*. Fonte: Mundo CNC: <https://www.mundocnc.com.br/historico.php>
- Foltran, T. (25 de agosto de 2018). *Akiyama Tecnologia em Componentes Eletrônicos Ltda*. Fonte: <https://www.guiacnc.com.br>: <https://www.guiacnc.com.br/akiyama-244/dimensionamento-do-motor-de-passo/>
- Galdino, L. (Dezembro de 2014). Cálculo da rotação, do torque e da potência de motores elétricos para transmissão por parafusos de potência. *Augusto Guzzo Revista Acadêmica, São Paulo, n. 14.*, pp. 215-227.
- Hannifin, Parker. (Outubro 2003). Tecnologia Eletromecânica. Jacarei, SP.
- Jeon, S. K., & Skogsrud, S. S. (2 de setembro de 2018). Fonte: Tudo sobre Grbl: <https://github.com/grbl/grbl>
- Kalatec Automação Ltda. (18 de setembro de 2018). Fonte: <http://www.kalatec.com.br/>: <http://www.kalatec.com.br/vamos-falar-sobre-driver-motor-passo/>
- Makita do Brasil. (26 de Outubro de 2018). *Informações sobre a tupia*. Fonte: <http://www.makita.com.br>: <http://www.makita.com.br/produtos/produto.asp?cod=RT0700C>
- Mesquita, D. P., & Walter, M. (2019). Reaction-Diffusion Woodcuts.
- Neomotion. (26 de outubro de 2018). *Datasheet de motores de passo e informações úteis*. Fonte: www.neomotion.com.br: <https://www.neomotion.com.br/wp-content/uploads/2017/07/Cat%C3%A1logo-Datasheet-dos-motores-de-passo-R01.pdf>
- Oriental Motor do Brasil Ltda. (18 de Setembro de 2018). Fonte: Site da Oriental Motor: <https://www.orientalmotor.com.br/tecnologia/tudo-que-voce-precisa-saber-sobre-motores-de-passo.html>
- Ryan, V. (25 de Setembro de 2018). Fonte: TechnologyStudent: <http://www.technologystudent.com/cam/camex.htm>
- Skogsrud, S. S., Jeon, S. K., & Geisler, J. (s.d.). *Repositório do GitHub*. Acesso em Junho de 2019, disponível em Grbl: <https://github.com/gnea/grbl/wiki>
- Winder, W. (10 de setembro de 2018). Fonte: Repositório GitHub: <https://github.com/winder/Universal-G-Code-Sender>
- Wood, G. a. (2005). *Digital image processing*.

APÊNDICE A – Lista de componentes



Legenda:

1. Acoplamento Flexível 6,35x8mm
2. Castanha Tr8 8mm com suporte
3. Rolamentos Pillow Block Scs12uu
4. Fuso trapezoidal Tr8 Passo 8mm comprimento
5. Guia linear aço temperado 12mm 300mm
6. Suporte guia linear Sk12
7. Mancal KP08 8mm
8. Mesa de sacrifício e estrutura de MDF
9. Guia linear aço temperado 16mm 1000mm
10. Fuso trapezoidal Tr8 Passo 8mm comprimento 1000mm
11. Motor de Passo Nema23 de 15Kg.f.cm
12. Tupia Makita Rt0700c 710w
13. Suporte Tupia em Alumínio
14. Rolamento Pillow Block Scs16uu
15. Suporte guia linear Shf16
16. Fuso trapezoidal Tr8 Passo 8mm comprimento 600mm
17. Guia linear aço temperado 16mm 600mm

APÊNDICE B – Valores por componentes

Nome	Quantidade	Valor Unitário	Total
Motores Nema23 15Kg + Driver TB6600	3	200	600
Tupia rc0700C	1	600	600
Mesa e Estrutura	1	200	200
Guia linear 16mm 1000mm	2	100	200
Guia linear 16mm 600mm	2	75	150
Rolamento Scs16uu	8	18,62	148,96
Suporte guia linear shf16	8	17,62	140,96
Castanha Tr8	3	45	135
Fuso Tr8 600mm	1	100	100
Guia linear 12mm	2	48	96
Rolamento Scs12uu	4	20	80
Suporte tupia alumínio	2	40	80
Mancal KP08	3	16,9	50,7
Fuso 8mm 300mm	1	50	50
Suporte Sk12	4	12,5	50
Acoplamento Flexível	3	15	45
		TOTAL	2726,62