



**SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL
FACULDADE DE TECNOLOGIA SENAI CIMATEC
CURSO SUPERIOR DE TECNOLOGIA EM MECATRÔNICA**

**FABRICAÇÃO DE PROTÓTIPOS POR FRESAMENTO UTILIZANDO
ROBÔS ANTROPOMÓRFICOS**

Salvador
2008

GUILHERME BOULHOSA RODAMILANS

**FABRICAÇÃO DE PROTÓTIPOS POR FRESAMENTO UTILIZANDO
ROBÔS ANTROPOMÓRFICOS**

Trabalho de conclusão de curso apresentado à Faculdade de Tecnologia SENAI Cimatec como requisito final para obtenção do título de Tecnólogo em Mecatrônica Industrial.

Orientador: Guilherme Oliveira de Souza.

Salvador
2008

Ficha catalográfica elaborada pela Biblioteca da Faculdade de Tecnologia
SENAI Cimatec

Rodamilans, Guilherme B.

Fabricação de protótipos por fresamento utilizando robôs antropomórficos/ Guilherme Boulhosa Rodamilans. Salvador, 2008. 67f.

1. CAM 2. Robôs 3. Usinagem. I. título

CDD 629.892

AGRADECIMENTO

Em primeiro lugar, agradeço aos funcionários do SENAI Cimatec pelo apoio que deram no andamento do curso. Agradeço ao núcleo de Automação, em especial Alexandre Ribeiro e Milton Bastos “Black”, que acompanharam e APOIARAM a nossa turma em praticamente toda a nossa caminhada. A todo o núcleo de Fabricação, em especial a Guilherme Souza, meu orientador de TCC, que topou esse projeto desde o momento em que fui conversar com ele pela primeira vez, e ainda me auxiliou de forma cautelosa em tudo que precisei para desenvolvê-lo. A Wilson, que me auxiliou no processo de usinagem e com o *software* NX. Um agradecimento especial para todo o núcleo de Microeletrônica, onde tive minha primeira experiência de trabalho e, em especial, a Cleber Vinicius, meu orientador de Iniciação Tecnológica, que me ensinou, entre muitas coisas, a pesquisar e me organizar. E, finalmente, aos meus colegas de sala que, com apoio mútuo, enfrentamos todas as dificuldades levando o nosso curso de forma prazerosa e proveitosa.

*“Gênio é 1% inspiração e 99%
transpiração.”*

(Thomas Edson)

RESUMO

A confecção de carcaça mecânica para protótipos funcionais de baixo custo, cujo *design* tem um papel fundamental, é uma tarefa complicada. Um dos fatores para esse problema está relacionado com os materiais normalmente empregados na confecção dos protótipos. Matrizes poliméricas reforçadas com fibra são uma ótima solução para associar design a baixo custo, porém necessitam de um molde para tal objetivo. Uma forma atraente para confecção de molde é gerá-lo através do seu modelo virtual. Isso pode ser feito com o auxílio de sistemas CAD/CAM associados à máquinas-ferramenta CNC. Em geral as máquinas usadas para tal tarefa, o fresamento, são os centros de usinagem e seu custo hora/máquina é elevado. Diante de um aumento de robôs antropomórficos em centros de tecnologia, tornar-se-ia interessante capacitá-los para executarem tarefas de usinagem com o objetivo de baratear o processo da confecção de moldes para fibra. Esse trabalho aborda o desenvolvimento de um pós-processador capaz de transformar a linguagem do arquivo neutro do CAM, que contém informações tecnológicas do processo de fresamento, em uma linguagem compatível com a do robô. Para tal utilizou-se as ferramentas do sistema CAD/CAM Unigraphics NX5 visando a comunicação com o robô ABB IRB 2400L. Por fim, foi realizado um teste de validação no qual uma peça em resina foi fresada utilizando-se o robô em questão. Erros dimensionais e de forma resultaram da usinagem da peça-teste, porém, julga-se que esses erros são desprezíveis diante do objetivo principal do trabalho, o de fazer com que um robô antropomórfico guiasse uma ferramenta de corte com instruções geradas a partir do modelo virtual de uma peça, fazendo-se uso de sistemas CAD/CAM.

Palavras-Chaves:

- pós-processador, CAM, robô, usinagem

Abstract

The manufacturing of mechanical casings for low-cost functional prototypes with esthetical value is a challenge and one of its reasons is closely related with the materials often applied. Fiber reinforced polymers are a good choice for good-looking design allied to low-cost but it brings the necessity of manufacturing a die. A attractive way for the manufacturing of a die is generate it from its virtual model. This can be made with the aid of CAD/CAM systems associated with CNC machines. The machines generally used for these tasks, milling, are known as machining centers and its hourly costs are usually high. Considering the raise of the number of anthropomorphic robots in technology centers, the use of them for machining operations could be an interesting choice to reduce the costs in the manufacturing of fiber molding dies. This work presents the development of a post-processor capable of convert the CAM neutral file language, containing milling process technological information, to a language that a robot understands. It was used the tools of the Unigraphics NX5 CAD/CAM system and the robot which was to run the programs generated were ABB IRB 2400L. At the end a validation test was performed in which a resin test-piece was milled with the cited robot. Dimensional and form errors resulted from the machining, nevertheless they were small, not affecting the achieving of the main goal of this work, making an anthropomorphic robot guide a cutting tool with instructions generated by a CAD/CAM system from the virtual model of a part.

Key words:

- post-processing, CAM, robot, machining

Lista de Figuras

Figura 1 – Exemplo de programa em Código G.	10
Figura 2 – Modos de concepção dos pós-processadores (SOUZA, 2006).....	19
Figura 3 – Exemplo de Juntas Rotativas.	22
Figura 4 – Robô Antropomórfico.....	23
Figura 5 – Abas do Post Builder.	27
Figura 6 – Sub-tópicos da aba <i>Program & Tool Path</i>	28
Figura 7 – Modificação do <i>G Codes</i>	29
Figura 8 – Suporte da Retificadora/Robô.	34
Figura 9 – Esquema do sistema para validação.	35
Figura 10 – Modelo Virtual da Peça a ser Usinada.	36
Figura 11 – Peça Usinada pelo Robô.	38
Figura 12 – Erro ocasionado pelo processo de referenciação.....	39
Figura 13 – Erro possivelmente ocasionado pelo robô.....	40

Lista de Tabelas

Tabela 1 – Exemplos de funções em código G (ISO 66025).....	9
-------------------------------------------------------------	---

Lista de Siglas

CAD – Projeto Auxiliado por Computador (<i>Computer Aided Design</i>)	2
CAM – Fabricação Auxiliada por Computador (<i>Computer Aided Manufacturing</i>)	2
CNC – Comando Numérico Computadorizado (<i>Computerized Numerical Control</i>)	2
CN – Comando Numérico	3
NURBS – <i>Non-Uniform Rational B-Splines</i>	3
MIG – <i>Metal Inert Gas</i> (Processo de Soldagem)	23
MAG – <i>Metal Active Gas</i> (Processo de Soldagem)	23

SUMÁRIO

1	INTRODUÇÃO	1
1.1	Justificativa.....	4
1.2	Objetivo	5
1.2.1	Objetivos Específicos	6
1.3	Impactos Esperados	6
2	COMANDO NUMÉRICO E CÓDIGO G	8
3	CAM	11
3.1	Etapas para Geração de Programa CN na Cadeia CAD/CAM	11
3.2	Parâmetros Gerais	14
3.3	Estratégia de Usinagem.....	15
4	PÓS-PROCESSADOR.....	18
5	ROBÔ INDUSTRIAL	21
5.1	Linguagem Robótica.....	23
6	MÉTODO DE EXECUÇÃO	25
7	IMPLEMENTAÇÃO E VALIDAÇÃO	26
7.1	Implementação	26
7.1.1	Machine Tool	27
7.1.2	Program & Tool Path.....	28
7.1.3	NC data Definitions	31
7.1.4	Output Settings	32
7.1.5	Post Files Preview.....	32
7.1.6	Considerações	32
7.2	Validação.....	33
7.2.1	Aparato para Validação	34
7.2.2	Peça a ser Usinada	36
7.2.3	Usinagem utilizando o Robô	36
7.2.4	Análise da peça usinada.....	38
8	CONCLUSÕES	41
	Sugestão para Trabalhos Futuros	43
	REFERÊNCIAS.....	44
	ANEXOS	45
	ANEXO 1 – Comandos utilizados do robô.....	46
	ANEXO 2 – Dados dos Programas de Usinagem.....	47
	ANEXO 3 – 5º Programa de usinagem do porta pilha escrito em código G... 51	
	ANEXO 4 – 5º Programa de usinagem do porta pilha escrito em RAPID.....	54

1 INTRODUÇÃO

Projetos que envolvam protótipos funcionais têm como um de seus principais desafios a confecção de sua carcaça mecânica e, quando é necessário um *design* arrojado, essa tarefa se torna mais difícil, devido, em parte, aos materiais normalmente empregados. Materiais como perfis de alumínio possuem ótima relação resistência/peso e boas propriedades térmicas, porém, trazem algumas dificuldades para serem trabalhados manualmente. Em geral, perfis de alumínio são ótimos para serem usados em peças com função estrutural, mas não são bons para produzir peças com *design* arrojado.

Outros elementos que são mais fáceis de trabalhar, como madeira e polímeros de baixa densidade, não trazem boa relação resistência/peso. Além disso, a madeira possui sérios problemas com a umidade do ambiente, necessitando de um tratamento prévio. Os polímeros de baixa densidade, a exemplo do Isopor®, são altamente reativos com elementos químicos como tintas e não suportam grandes temperaturas.

Em projetos de aeronaves ou carros, cuja aerodinâmica possui uma importância fundamental, a associação do *design* com materiais que possam ser usados como elementos estruturais traz, em geral, a necessidade do uso de ferramentas especiais, como moldes de injeção e matrizes de conformação. Porém, o preço muito elevado dessas ferramentas torna inviável a confecção de um número reduzido de protótipos ou peças.

As matrizes poliméricas reforçadas com fibra de vidro, carbono ou Kevlar®, trazem ótima relação resistência/peso, são fáceis de moldar e suportam de médias a altas temperaturas. A fibra de carbono se destaca por possuir ótima resistência mecânica e química, mesmo em altas temperaturas, e possuir preço acessível. Esses compósitos são ideais para a fabricação de peças únicas, protótipos funcionais e pequeno número de peças, pois possuem baixo custo no processo de produção.

O emprego de um molde torna-se essencial para se conseguir formas bem trabalhadas das fibras, economizando-as e reduzindo o peso da peça. Além de suas excelentes características, o uso desses materiais é bastante atrativo para a

construção de protótipos funcionais ou peças únicas devido à vasta variedade de processos na confecção do molde. Alternativas para a fabricação de moldes englobam desde um processo simples, como esculpi-los, até o uso de ferramentas mais modernas capazes de obtê-los a partir de um modelo virtual. Pode-se usar uma vasta gama de matérias-primas em ambos os processos. Independente do método da confecção do molde é possível obter acabamento satisfatório no objeto final, o protótipo ou peça.

Em geral, para conseguir transformar um modelo virtual em um molde físico, é necessário o uso de alguns tipos de ferramentas computacionais. As ferramentas do tipo CAD, *Computer Aided Design* ou Projeto Auxiliado por Computador, estão bastante difundidas no mercado e são usadas tanto por profissionais especializados como por iniciantes. Uma das maiores vantagens destas ferramentas é permitir a visualização em 3D de uma peça que está sendo criada, sem a necessidade da sua construção física. Com isso, é possível fazer um estudo completo da peça, verificando possíveis erros estruturais e volumétricos, deixando-a menos susceptível a erros na sua fabricação (SILVA FILHO, 2000).

Com o auxílio de outra ferramenta, o CAM, *Computer Aided Manufacturing* ou Fabricação Auxiliada por Computador, junto com uma máquina do tipo CNC, *Computerized Numerical Control* ou Comando Numérico Computadorizado, é possível fabricar uma peça diretamente do modelo virtual criado em CAD. Nesse processo, a máquina CNC se comporta como o braço de um escultor segurando uma ferramenta e o computador como sua mente. O computador, a mente, irá dizer quais os lugares que a ferramenta deve percorrer para retirar material e, com isso, conseguir transformar a matéria prima na peça que se imaginava (SILVA FILHO, 2000).

Dentre as máquinas-ferramenta CNC, os Centros de Usinagem são ideais para a fabricação de moldes, e isso é feito através do processo de fresamento (SENAI-BA, 2005). As máquinas mais comuns deste tipo são capazes de realizar operações de usinagem através de movimentos lineares coordenados e simultâneos entre ferramenta e peça em três eixos distintos, X, Y e Z. A junção desses movimentos faz com que a ferramenta consiga posicionar-se em qualquer lugar dentro do volume de trabalho da máquina.

Máquinas mais avançadas possuem, além dos eixos de posicionamento, outros eixos com movimentos rotatórios, que servem para orientar a fresa. As mais comuns são as de 4 e 5-eixos. A orientação da fresa possibilita uma inclinação constante relativa à normal da superfície usinada. Além disso, essas máquinas conseguem usinar peças mais complexas, pois podem ter acesso a regiões que os centros de usinagem em 3-eixos não alcançam (SOUZA, 2006).

Máquinas CNC precisam receber informações das trajetórias a serem descritas pela ferramenta de corte, para a remoção do material desejado. Essas informações podem ser geradas diretamente por um operador. Porém, em aplicações que envolvam superfícies complexas ou peças com grande número de detalhes é necessário o uso do computador para auxiliar na geração de trajetória devido ao grande número de pontos que a ferramenta deverá percorrer.

Uma das aplicações dos sistemas do tipo CAM é auxiliar na geração das trajetórias de ferramentas. *Softwares* CAM de geração de trajetória importam arquivos de sistemas CAD, o modelo virtual da peça, e oferecem ao operador alternativas de rotinas parametrizáveis que devem ser preenchidas para que o *software* execute seu algoritmo de geração de trajetória (CAVALHEIRO, 1998).

Após a geração dessas trajetórias, a última etapa é traduzir estas informações para uma linguagem que seja entendida pela máquina que irá executar a operação. Esse processo é conhecido como pós-processamento (SOUZA, 2006), e seu produto final é o programa CN (Comando Numérico). A linguagem mais comum de programas CN é o código G, definido pela norma ISO 6983 de 1982.

Programas CN são códigos alfanuméricos contendo informações sobre as trajetórias e os parâmetros de corte da operação de usinagem. Como estrutura básica, possuem um cabeçalho com instruções específicas de preparo de uma determinada máquina, um corpo com informações dos pontos que a ferramenta deve percorrer seguido de como será descrita a trajetória entre esses pontos (linear, circular, equação polinomial, NURBS), velocidade de avanço e velocidade de rotação da ferramenta, entre outras informações. Nos centros de usinagem que possuem 5-eixos, o programa ainda deverá contemplar informações necessárias para orientar a ferramenta (CAVALHEIRO, 1998).

Em centros de pesquisa, como faculdades de tecnologia, é possível encontrar Centros de Usinagem em 3-eixos, sendo as máquinas de 5-eixos mais raras devido ao seu alto custo. Porém, é comum também a presença de robôs antropomórficos, principalmente em instituições com cursos da área de mecatrônica. Esses robôs lembram um braço humano e são caracterizados por possuírem movimento rotacional em suas três primeiras juntas (SENAI-DR, 2004).

Em geral, robôs antropomórficos possuem seis juntas rotacionais e seu volume de trabalho lembra o formato de uma esfera. Esses robôs podem se comportar como um centro de usinagem em 5-eixos, porém, para trajetórias complexas, é conveniente o uso de um programa que auxilie na geração de trajetória como ocorre para as máquinas CNC. Os softwares CAM podem ser usados para fornecer esse auxílio desde que o arquivo de saída esteja com o código compatível com a linguagem de comando do robô que será utilizado. Os robôs possuem linguagem própria que depende do fabricante e de seu modelo.

Visto que a confecção de molde, para compósitos de fibra, aliado com a técnica de fabricação através de um modelo virtual, traz uma maior facilidade na confecção de carcaça mecânica estrutural e sabendo que o robô pode executar operações semelhantes aos centros de usinagem, torna-se interessante capacitar robôs para poderem executar operações de usinagem.

1.1 Justificativa

Com a dificuldade de se utilizar centros de usinagem em 3 e 5-eixos, devido ao elevado custo hora/máquina, torna-se interessante o estudo do uso de outras máquinas que possuam características e possam funcionar de forma semelhante. Apesar de sua cinemática ser diferente, o robô antropomórfico aparenta ser uma ótima alternativa para suprir a necessidade do uso dessas máquinas na usinagem de materiais de baixa resistência ao cisalhamento.

Além disso, o robô é concebido com o intuito de ser uma máquina versátil, teoricamente capaz de executar qualquer tarefa. Logo, o estudo do uso do robô na operação de fresamento o habilitará a executar mais um tipo de tarefa. A possibilidade de geração de sua trajetória através de programas CAM, possibilitaria também, o uso desse método para resolver problemas como o da fabricação de

protótipos funcionais, agregando, dessa forma, maior qualidade estética aos protótipos através da fabricação de modelos e moldes para serem utilizados com fibras de carbono e vidro, polímeros e resinas.

No caso específico do SENAI Cimatec, é possível encontrar quatro robôs antropomórficos e apenas um centro de usinagem disponível para os alunos dos cursos da área de mecatrônica estudarem. Assim, estudos nessa área podem trazer soluções para suprir a dificuldade de acesso aos centros de usinagem e até viabilizar o uso de uma máquina que possa se comportar como centro de usinagem em 5-eixos.

Além disso, o uso destes robôs proveria aos alunos do curso superior de mecatrônica industrial do SENAI Cimatec, que optarem por confeccionar projetos mecânicos usando essa tecnologia, a possibilidade de vivenciar todas as etapas do processo de fabricação de protótipos muito semelhante ao método real. Isso faz com que esses alunos adquiram e aprimorem seus conhecimentos em:

- uso da ferramenta CAM e sua interação com as ferramentas CAD, assunto não contemplado no curso, mas de ampla aplicação na indústria e que interage fortemente com a disciplina CNC;
- maior aprofundamento dos conhecimentos no processo de fresamento, expandindo assim o conteúdo da disciplina Processos Mecânicos de Fabricação;
- aprofundamento em programação de robôs e maior compreensão das limitações de trabalho dos mesmos, ampliando os conhecimentos das disciplinas Robótica I e II.

1.2 Objetivo

Este Trabalho de Conclusão de Curso está inserido no âmbito de um projeto que visa viabilizar o uso de robôs antropomórficos para a realização de operações de fresamento em materiais de baixa resistência ao cisalhamento. Seu principal objetivo é viabilizar a comunicação entre um sistema CAM e um robô antropomórfico comercial. Para tal, será confeccionado um pós-processador capaz de extrair do módulo *Manufacturing*, do *software* Unigraphics NX 5, informações de

trajetória de ferramenta para a criação de um arquivo de programação de robô compatível com o Robô ABB IRB 2400L localizado no Laboratório de Fabricação Robotizada do SENAI CIMATEC. Esse arquivo deve conter informações de um programa de usinagem com o código adaptado para que o robô execute-o levando em consideração suas limitações cinemáticas, mecânicas e sua forma de controle. Serão contempladas nesse arquivo trajetórias lineares, velocidades de avanço e de rotação, e funções de preparação do robô para executar essas tarefas.

1.2.1 Objetivos Específicos

Para o desenvolvimento do projeto foram traçados três objetivos específicos.

- Comparar e compatibilizar a linguagem do *software* NX5 com o Robô ABB IRB 2400L;
- Construir pós-processador para viabilizar a execução de trajetórias de ferramenta geradas no *software* NX5 no Robô ABB IRB 2400L;
- Validar o projeto através da execução de um programa de usinagem com o robô ABB IRB 2400.

1.3 Impactos Esperados

Este trabalho traz impactos diretos na área tecnológica e na área educacional.

Os impactos tecnológicos são:

- uma alternativa para a fabricação por fresamento de peças complexas em materiais de baixa resistência mecânica;
- tecnologia para a construção de pós-processadores com arquivos de saída escrito em código G ou em linguagem RAPID (Robô ABB IRB 2400L);
- confecção de pós-processadores para a interação do *software* NX 5 com as máquinas Discovery 1250, Discovery 560, Discovery 4022 e Hermle C 800 U;

Os impactos educacionais são:

- acrescentar mais uma funcionalidade aos robôs antropomórficos de uso didático;

- estudos Iniciais para solucionar problemas na confecção de carcaça mecânica para protótipos funcionais;
- possibilidade de expansão do conhecimento do aluno ao tentar solucionar o problema na construção de protótipos funcionais através dessa metodologia;
- outra possibilidade na geração de trajetória *off-line* para robô.

2 COMANDO NUMÉRICO E CÓDIGO G

Pode-se dizer que Comando Numérico (CN) é um equipamento eletrônico capaz de, sem intervenção de um operador, receber informações de entrada de dados, compilar estas informações e fazer com que os atuadores façam um utensílio seguir uma seqüência de ações pré-programadas.

Em 1808, Joseph M. Jacquard desenvolveu uma máquina industrial muito semelhante às máquinas CN, diferindo apenas por ser completamente mecânica. Através do uso de cartões de metal perfurado, Joseph conseguiu fazer com que um tear automaticamente ativasse ou desativasse os pontos através da presença ou ausência de furos, definindo assim o desenho do tecido. Pode-se considerar essa máquina a precursora das máquinas CN (PEREIRA, 2003).

O Comando Numérico foi criado para suprir as necessidades da confecção de peças complexas para Força Aérea dos Estados Unidos (*U.S. Air Force*) durante a II Guerra Mundial. O Laboratório de Servomecanismos do Instituto de Tecnologia de Massachusetts (MIT), junto com a Força Aérea dos Estados Unidos e a empresa Parsons Corporation of Traverse, em 1949, começaram os estudos para solucionar o problema armamentista (PEREIRA, 2003).

Para isso, os pesquisadores do MIT retiraram todos os comandos convencionais da fresadora de três eixos Hydrotel e adotaram um sistema de leitura de fita de papel junto com uma unidade de processamento. Essa unidade foi capaz de converter as informações da fita de papel para comandos elétricos compatíveis com os servomecanismos, responsáveis pelo acionamento dos eixos. A demonstração da máquina ocorreu em março de 1952 e a publicação do sistema em maio de 1953 (PEREIRA, 2003).

Com o advento e popularização dos sistemas microprocessados e dos microcomputadores, as máquinas CN incorporaram esses elementos surgindo assim o Comando Numérico Computadorizado, o CNC. Suas vantagens consistem na diminuição da central de controle, maior poder de processamento e maior flexibilidade. Este último fator é devido à possibilidade de alterar o programa de instrução na própria máquina, já que esse deixou de ser escrito em cartões perfurados.

Com o objetivo de padronizar um código para máquinas CNC, em 1958 a EIA (*Electronic Industries Association*) organizou estudos para padronizar a entrada dos Programas de Instruções (conhecidos como código G) feito conforme o Padrão RS-244, que depois passou a ser chamado de EIA244A ou ASCII. Posteriormente, a ISO também padronizou o código G, ISO 6983-88. Apesar de semelhantes, essas padronizações são distintas (PEREIRA, 2003).

Um arquivo de código G é composto basicamente por duas partes. A primeira é um cabeçalho com instruções para a preparação da máquina CNC e a segunda é o corpo com as instruções de trajetória, velocidades, mudança de ferramenta, etc.

Cada linha do programa começa com a letra N (rótulo da linha) seguida pelo número da linha, formando assim seu endereço. Após o número da linha, são colocadas as funções que se deseja executar (G ou M) ou o registro T para troca de ferramenta. A Tabela 1 mostra alguns exemplos de funções do código G.

Tabela 1 – Exemplos de funções em código G (ISO 66025).

G00	Função de posicionamento através de movimentação rápida onde a ferramenta se desloca em linha reta até o ponto especificado pelos parâmetros de coordenada
G01	Movimentação por interpolação linear
G02	Movimentação por interpolação circular no sentido horário
G03	Movimentação por interpolação circular no sentido anti-horário

Depois são inseridos os parâmetros necessários referentes à função. Na Figura 1 é mostrada uma parte de um programa em Código G (MENEGHELLO, 2003).

```
N0010 T02; "fresa de topo 19mm "  
N0020 O02 S1500 M03;  
N0030 M08;  
N0040 G00 X40. Y20. Z10. ;  
N0050 G01 X40. Y20. Z-4. F1000;  
N0060 G01 X110. Y20. Z-4. F1000;  
N0070 G03 X130. Y40. Z-4. I110. J40. F1000;  
N0080 G01 X130. Y140. Z-4. F1000;  
N0090 G03 X110. Y160. Z-4. I110. J140. F1000;  
N0100 G01 X40. Y160. Z-4. F1000 M3 S4000;
```

Figura 1 – Exemplo de programa em Código G.

O código G é um código incompleto e por isso os fabricantes complementam-no para atender as suas necessidades. Por esse, entre outros motivos, os programas em código G nem sempre são compatíveis, principalmente se os códigos são feitos para máquinas de fabricantes diferentes. Os programas podem ser gerados a partir de uma programação utilizando diretamente a máquina CNC (quando suportam esse tipo de programação) ou através de um PC.

Em PCs, pode-se usar editores de texto ASCII para programas mais simples, porém, para peças mais complexas, geralmente utiliza-se *software* de tipo CAM para auxiliar na confecção do programa. No processo da transformação da trajetória de ferramenta, contida no arquivo neutro do CAM, para o código G, faz-se necessário o uso de uma ferramenta computacional dedicada, conhecida por pós-processador¹, que deve ser compatível com a máquina para qual se pretende usar o código (PEREIRA, 2003).

¹ Vide Capítulo 4.

3 CAM

Segundo GROOVER *apud* MENEGHELLO (2003), sistemas CAM, *Computer Aided Manufacturing* ou Fabricação Auxiliada por Computador, podem ser definidos como ferramentas computacionais capazes de auxiliarem no planejamento, gerenciamento e controle da manufatura. Segundo esta definição, pode-se dividir os sistemas CAM em duas categorias:

Planejamento da manufatura – não existe uma interação direta, on-line, com o processo de manufatura. Nessa categoria, as principais funções são de auxiliar no planejamento de processo, programação da produção e criação de programas CN para máquinas CNC (MENEGHELLO, 2003).

Controle da Manufatura – o computador tem uma interação direta com a manufatura. Esses sistemas podem controlar, supervisionar e diagnosticar os processos de manufatura (MENEGHELLO, 2003).

Apesar dessa divisão, normalmente, quando se fala em sistemas CAM, associa-se aos *softwares* utilizados para a geração de trajetórias de ferramentas de corte para máquinas CNC. O processo de geração de trajetória é comumente associado a sistemas CAD. Um modelo geométrico virtual da peça criada em CAD é exportada para um *software* CAM. Após a configuração e inclusão de dados da manufatura pelo programador, o *software* CAM está pronto para gerar automaticamente o arquivo de trajetória de ferramenta, arquivo CN. A união do sistema CAD e o sistema CAM é conhecida como sistema ou cadeia CAD/CAM.

3.1 Etapas para Geração de Programa CN na Cadeia CAD/CAM

Para a geração de trajetórias e criação do programas para máquinas CNC com o uso de sistemas CAD e CAM pode-se distinguir sete passos.

1º passo: Criação do Modelo da Peça

O modelo da peça é confeccionado em *softwares* do tipo CAD. Existem muitos softwares no mercado, porém é necessário que o arquivo de saída seja

compatível com o software CAM que se deseja usar. Alguns *softwares* CAM são capazes de gerar um modelo virtual, através de um módulo CAD integrado, porém, normalmente possuem funções limitadas. É interessante frisar que sistemas CAD/CAM possuem essas duas funções divididas em módulos CAD e CAM, não sendo igual ao exemplo citado acima (CAVALHEIRO,1998).

2º passo: Exportação para CAM

Apesar de existirem extensões de arquivos CAD que são padronizadas, a exportação desses arquivos para *softwares* CAM está suscetível a anomalias geométricas causadas pelo mau “interfaceamento”. Em geral, esse fato ocorre quando os fabricantes dos *softwares* são diferentes. Quando o *software* CAM tem um módulo de edição geométrica é mais fácil solucionar esse tipo de defeito, porém quando não é dotado dessa ferramenta, é necessário voltar ao programa CAD e reeditar a geometria da peça. Apesar do seu alto custo, *softwares* que possuem os dois módulos, CAD e CAM, trazem como uma de suas vantagens a compatibilidade na transferência de geometrias (CAVALHEIRO,1998).

3º passo: Definição das Partes que Serão Fresadas

Após a importação, o programador define para o *software* CAM quais são as superfícies para as quais programas de usinagem serão gerados. Pode ser necessário separá-las, para que se estabeleçam diferentes estratégias e parâmetros de usinagem mais adequados a cada região.

4º passo: Definição de Parâmetros Gerais e Estratégia de Usinagem

O programador entrará com os dados dos parâmetros básicos e escolherá a estratégia de usinagem. Esses itens serão melhor abordados nos sub-capítulos posteriores.

5º passo: Execução do Cálculo da Trajetória

O programa executa o algoritmo de cálculo de trajetória definido pelo operador. Essa tarefa requer uma grande quantidade de processamento do computador e no final desse processo é gerado o arquivo neutro.

6º passo: Análise da trajetória Criada

Apesar de o computador calcular as trajetórias, ele também está susceptível a erro. Logo, após a geração das trajetórias pelo computador, é necessário que o programador verifique-as para que não ocorram erros e nem danos materiais e humanos na usinagem da peça, também deve-se verificar se as trajetórias podem ser otimizadas. Essas verificações podem ser feitas através de simulações no computador. Existem simuladores de todos os tipos, em geral os próprios softwares CAM já possuem simuladores. Porém, programas específicos para fazer essas tarefas podem comparar a geometria da peça após a usinagem virtual, com os dados do arquivo neutro ou com dados do arquivo pós-processado, com a geometria da peça gerada no CAD. Caso possuam uma modelagem da máquina a ser usada, esses programas podem também identificar colisões da máquina com a peça, dispositivos de fixação ou outros componentes da própria máquina. Exemplos de problemas mais freqüentes são (CAVALHEIRO,1998):

- *Overcut*, ferramenta retira mais material do que se deseja, podendo assim interferir nas dimensões da peça;
- *Undercut*, o processo não remove todo o material como desejado;
- Colisões;
- Entrada inadequada da ferramenta;
- Marcas de entradas, interrupções ou paradas bruscas.

7º passo: Pós-processamento do Arquivo Neutro

Após a análise concluída e erradicados possíveis erros, faz-se o *software* CAM rodar o seu algoritmo de pós-processamento. É necessário que o pós-processador escolhido seja compatível com a máquina-ferramenta que se deseja

usar. Nesse passo, é gerado o arquivo que será transferido para as máquinas CNC, em geral é escrito em código G (SILVA FILHO, 2000).

3.2 Parâmetros Gerais

Durante a geração de trajetórias é necessário fornecer para o software CAM alguns parâmetros que serão levados em consideração no momento em que o mesmo executar o seu algoritmo de geração de trajetória. São eles:

Peça Bruta: No software CAM, deve ser criada uma região que representará a matéria prima a ser usinada. Apesar de não ser necessariamente uma reprodução fiel da matéria prima, a peça bruta ou bloco (*block, Workpiece, Stock*), é criada para mostrar ao programa as delimitações volumétricas quando o mesmo estiver executando o seu algoritmo de geração de trajetória (SILVA FILHO, 2000).

Ferramenta: A geometria da ferramenta tem um papel fundamental na geração de trajetória. Como exemplo, nos processos de fresamento mais comuns têm-se fresas com o topo reto, com raio de ponta, esférico e toroidal, entre outros. Dependendo do tipo de ferramenta, a remoção de cavaco vai ser diferente.

Parâmetros de Corte:

- Velocidade de corte v_c ;
- Profundidade de corte a_p ;
- Avanço f e avanço por dente f_z ;
- Velocidade de avanço v_f – pode variar ao decorrer do programa, dependendo da operação que a ferramenta irá fazer (ex. entrar na peça) ou o quanto de material a mesma está removendo;
- Velocidade de rotação da fresa n ;
- Penetração de trabalho a_e (*step over, side step*);
- Sentido de corte (concordante ou discordante).

Movimentos globais da ferramenta

Parâmetros relativos ao posicionamento e aos movimentos locais da ferramenta:

- Ponto inicial (*home*);
- Zero-peça;
- Plano de segurança (*safe, clearance, traverse*);
- Plano de início ou saída (*start, z-up*);
- Incremento 2D e Incremento 3D;
- Direção e sentido das trajetórias;
- Ângulo de usinagem;
- Movimentos de entrada e saída;
- Movimento de conexão entre passadas adjacentes (*links, transition, connect, moves*).

3.3 Estratégia de Usinagem

O programador deve definir como será retirada a matéria-prima na confecção da peça. Em geral, primeiro faz-se uma operação de desbaste, aproximando assim o material que está sendo usinado às dimensões finais da peça. Nessa etapa, é feita a maior remoção de material. Ela é, de certa forma, grosseira, fazendo com que se economize tempo na usinagem. Depois, é realizado o pré-acabamento, dedicado à regularização do material sobressalente, de maneira a tornar o sobremetal constante para garantir a precisão da etapa posterior, o acabamento.

A lógica de distribuição das trajetórias também faz parte da estratégia de usinagem. São três os tradicionais métodos para a geração de trajetória em máquinas de 3-eixos para modelos 3D, são eles: Método Isoparamétrico, Método APT, Método Cartesiano. A partir dele os fabricantes dos sistemas CAM fizeram os seus algoritmos de geração de trajetória, onde se destacam (MASTELARI, 2004):

- Varredura de Área em 2½ eixos – em zigue-zague ou paralela ao contorno;
- Perfilamento em 2½ eixos;
- Usinagem por planos paralelos;
- Projeção Radial;
- Projeção Helicoidal ou Patamares em hélice;
- Usinagem por Offset de Contornos;
- Perfilamento em 3 eixos;
- Varredura em 3 eixos por patamar;
- Gravação (*engraving*);
- Follow Periphery;
- Ranhuras;
- Usinagens isoparamétricas para superfícies individuais ou seqüenciais;
- Usinagem isoparamétrica para superfícies regradas;
- Usinagem segundo projeção isoparamétrica;
- Re-usinagem ou estratégia para remoção de material remanescente.

As trajetórias são compostas por curvas geradas com base na equação da geometria da peça, levando-se em consideração uma tolerância estabelecida, e a forma de descrição dessas curvas pela ferramenta de corte é limitada pelo comando numérico da máquina. Os métodos utilizados para a adequação da trajetória à geometria da peça são:

Interpolação Linear – A trajetória é criada através da interpolação linear entre dois pontos da equação do modelo geométrico da peça, formando assim diversos seguimentos de reta. Esse método, apesar de sua simplicidade matemática, gera programas NC grandes, não sendo aconselhável para peças muito complexas devido ao tamanho do programa gerado.

Interpolação Circular – Nesse método, o sistema CAM gera a trajetória através da interpolação circular e linear da equação do modelo geométrico da peça, logo a trajetória será formada por segmentos de reta e semicírculos. É o método mais comumente utilizado.

Interpolações Complexas – Nessa metodologia, a trajetória é descrita através de equações matemáticas aproximadas do modelo geométrico da peça, sendo mais comum o uso de NURBS e equações Polinomiais. Não são todas as máquinas CNC que suportam este tipo de arquivo de trajetória, o que torna seu uso restrito.

Para finalizar o processo de geração de trajetória, faz-se necessário definir uma faixa de tolerância em relação ao modelo geométrico, para que dentro desses limites o programa CAM gere as interpolações necessárias na criação da trajetória de ferramenta.

4 PÓS-PROCESSADOR

Após o cálculo da trajetória desejada, o sistema CAM gera um arquivo neutro denominado CLData (*Cutter Location Data*) ou CLFile (*Cutter Location File*). Esse arquivo contém um banco de dados com características geométricas e tecnológicas, além da trajetória e características da ferramenta. O pós-processador utiliza esses dados e gera um arquivo de trajetória de ferramenta compatível com a máquina a ser utilizada. Para isso, aplica-se, em cima dos dados do arquivo neutro, um artifício matemático denominado “transformação cinemática inversa”, obtendo assim os movimentos dos eixos envolvidos na operação (MENEGHELLO, 2003).

Na interpolação linear ou circular, o arquivo neutro possui pontos de posicionamento relacionados com os eixos cartesianos (XP, YP e ZP). Para máquinas com mais de 3-eixos, é definido também nesse arquivo um versor de orientação de ferramenta (i, j, k). Já no método de interpolações complexas, o arquivo neutro possui equações para descrever o posicionamento da ferramenta $T(w)$. Em geral este último método é usado em centros de usinagem de 3-eixos. Para ser usado em máquinas com mais de 3-eixos é necessário dois sistemas de equações, um para a ponta da ferramenta e um para um ponto qualquer localizado no eixo da fresa acima de sua ponta, de maneira a definir sua orientação (SOUZA, 2006).

Os movimentos dos eixos podem, por exemplo, para um centro de usinagem em 5-eixos, ser descritos como (X_m, Y_m, Z_m, A, C) onde (X_m, Y_m, Z_m) descrevem os movimentos dos eixos convencionais de movimento linear e A e C os movimentos dos eixos de rotação em torno de X e Z respectivamente.

A Figura 2 mostra os possíveis modos de conversão de trajetória. No modo 1 o pós-processador, que está ligado diretamente à ferramenta CAM, gera todas as instruções de movimentação de eixos. Para isso é necessário que o pós-processador possua informações a respeito de toda a cadeia cinemática da máquina. No caso de máquinas de 3-eixos é o mais comum, porém, para máquinas com eixos rotativos só é usado quando a capacidade de processamento do CNC não consegue executar um grande volume de cálculo, tornando essa configuração muito rara (SOUZA, 2006).

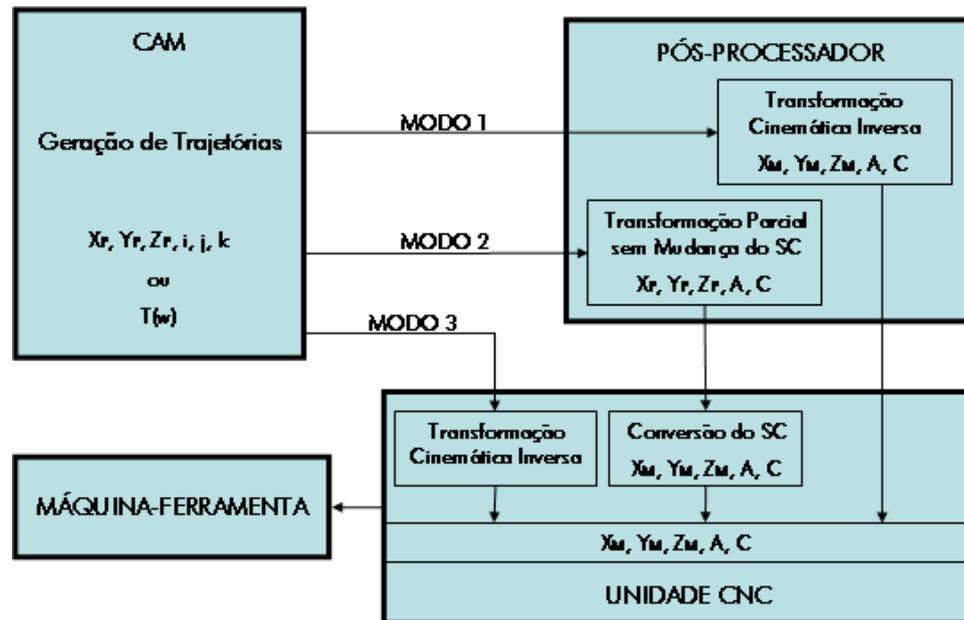


Figura 2 – Modos de concepção dos pós-processadores (SOUZA, 2006).

No modo dois, o pós-processador da ferramenta CAM faz uma adaptação do arquivo neutro para um código que a máquina CNC entenda. Geralmente, usa-se o código G. É conveniente falar que nessa configuração a máquina CNC possui também um pós-processador próprio, ao qual o usuário não tem acesso, contendo toda a cadeia cinemática dela, sendo assim, dedicada à mesma. Esse pós-processador gera os movimentos dos eixos da máquina através dos dados fornecidos pelo pós-processador do sistema CAM. Essa configuração é a mais usada e pode ser chamada de Híbrida, já que parte do pós-processamento é realizada pela ferramenta CAM e parte pelo CNC (SOUZA, 2006).

O modo 3 é aquele em que todo o pós-processamento é feito pela máquina CNC que recebe o arquivo CLData. Como principal vantagem, consegue-se uma maior fidelidade de forma na trajetória da ferramenta (SOUZA, 2006).

Como o robô tem um controlador próprio com algoritmos que calculam os movimentos dos eixos relativos à trajetória a ser descrita, levando em consideração aspectos como menor esforço entre juntas, o pós-processamento híbrido parece mais adequado para melhor eficiência e vida útil do robô. Nesse caso, o pós-processador do sistema CAM forneceria os dados de trajetória em uma linguagem compatível com o robô para que o controlador do mesmo escolhesse o melhor

posicionamento das juntas e descrevesse a trajetória desejada. Neste caso, o controlador do robô funcionaria como um pós-processador de uma máquina CNC.

Alguns sistemas CAM possuem ferramentas dedicadas à construção de pós-processadores capazes de gerar arquivos de saída contendo comandos peculiares às máquinas desejadas.

5 ROBÔ INDUSTRIAL

Segundo a *Robotic Industries Association* (RIA) o robô industrial é um “manipulador multifuncional re-programável, projetado para movimentar materiais, partes, ferramentas, ou peças especiais, através de diversos movimentos programados, para o desempenho de uma variedade de tarefas” (RIVIN *apud* ROMANO, 2002).

Introduzida por Karel Capex em uma de suas peças de ficção científica “*Rossum’s Universal Robots*” datada de 1921, a palavra *robots* em checo quer dizer trabalho árduo, trabalho forçado. Com essas influências hoje a palavra *Robot*, Robô, em português, é associada a máquinas que conseguem atuar seguindo uma lógica. Em 1954 G.C. Devol patenteia o conceito de robô industrial e em 1961 o primeiro robô industrial foi instalado pela Unimation na General Motors. Foi utilizado com o objetivo de manipular peças da produção de uma máquina de fundição (SENAI-DR, 2004).

Utilizado para trabalhos arriscados ou insalubres para o homem, os robôs industriais são também considerados convenientes para o uso em trabalhos repetitivos. Possuem basicamente duas partes. O manipulador é a parte mecânica/estrutural para a execução do movimento, e o controlador é a parte que contém a lógica e, através do processamento das informações, consegue controlar o movimento e atuação do robô (SENAI-DR, 2004).

Mecanicamente, o manipulador é formado por estruturas rígidas (corpos ou elos), conectados entre si através das articulações (juntas). As articulações têm uma real importância no tipo de movimento que o robô pode fazer, são classificadas como (SENAI-DR, 2004):

Prismáticas (P) – Geram movimentos lineares entre dois elos;

Rotativas (R) – Geram entre dois elos a variação da angulação em um único plano (Figura 3);

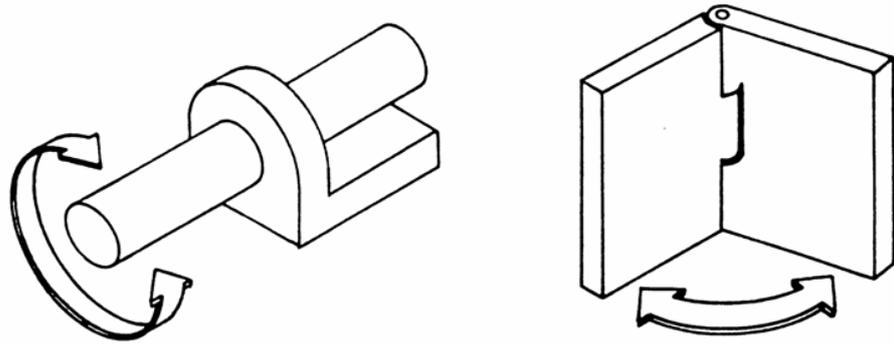


Figura 3 – Exemplo de Juntas Rotativas.

Esféricas – Geram entre dois elos a variação da angulação em qualquer plano.

Com isso o robô industrial também pode ser classificado segundo as suas três primeiras articulações, são elas as responsáveis por posicionar a ferramenta do robô no espaço, gerando assim uma região onde o robô pode alcançar, isto é, o seu volume de trabalho (ROMANO, 2002).

Robôs Cartesianos (PPP) – Possuem três juntas prismáticas cujos eixos de movimentos são coincidentes com as coordenadas cartesianas. Geram um volume de trabalho prismático.

Cilíndrico (RPP) – Uma base rotativa movimenta duas juntas prismáticas que estão em paralelos entre si. O volume de trabalho gerado é um cilindro.

Esférico (RRP) – Uma junta prismática é montada sobre duas juntas rotacionais defasadas de 90° . O volume de trabalho gerado é uma esfera.

Scara (RRP) – Uma junta prismática é montada sobre duas juntas rotacionais sem defasagem. O volume de trabalho gerado é um cilindro.

Antropomórfico(RRR) – Possui as suas três primeiras juntas rotacionais. Seu volume de trabalho é uma esfera (Figura 4).

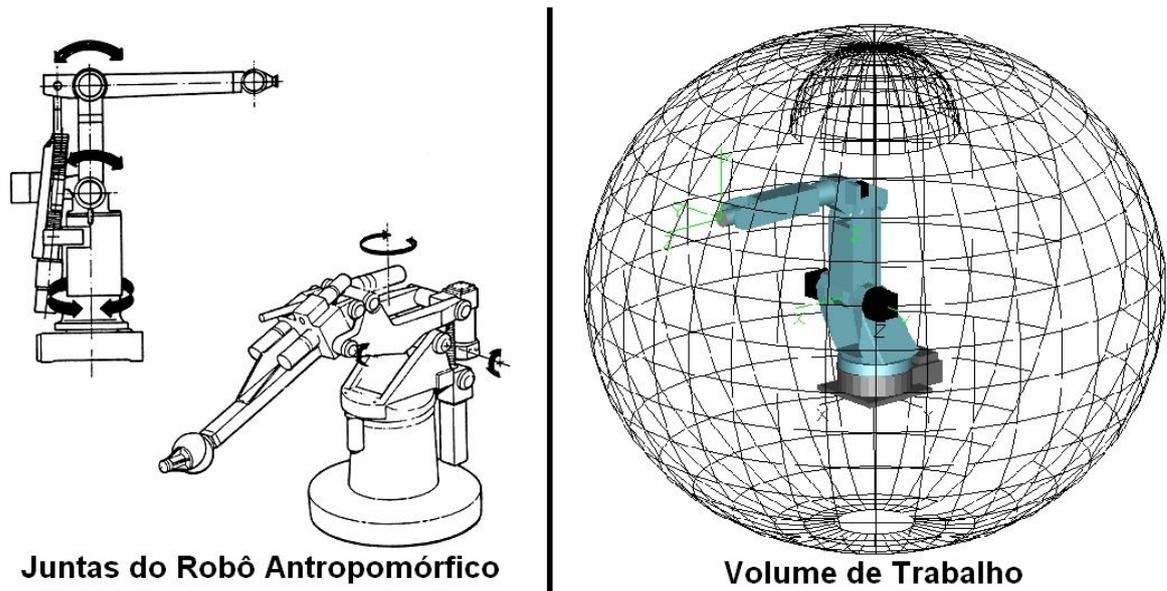


Figura 4 – Robô Antropomórfico.

Os robôs antropomórficos são apropriados para manipulação onde se exija alta velocidade e flexibilidade. Devido a sua forma construtiva tem sido largamente utilizado na indústria desde áreas de carga e descarga, até em áreas que necessitam de movimentos mais complexos, como em processos de soldagem (MIG e MAG).

Em geral esse tipo de robô possui seis eixos, dando-lhe uma liberdade de movimento excelente e em sua extremidade é colocada uma ferramenta destinada ao tipo de trabalho que irá executar. Como citado, os três primeiros eixos posicionam a ferramenta no espaço e os eixos subseqüentes orientam o manipulador. Logo travando a ponta da ferramenta em um ponto qualquer, é possível dispor ela espacialmente de diversas formas diferentes.

5.1 Linguagem Robótica

Além do seu funcionamento como um manipulador o robô antropomórfico geralmente consegue interagir com o meio através de entradas e saídas que podem ser analógicas e/ou digitais. Em seu funcionamento o robô lê e executa um arquivo contendo dados de trajetória, para o manipulador e dados de interação com o meio. Os dados contidos nesses arquivos, junto com a percepção do meio através dos

sensores, possibilitam a essas máquinas tomarem decisões e gerarem código de alerta, máquinas “semi-inteligentes”.

A linguagem desses arquivos é de propriedade de cada fabricante, mas, praticamente todas elas seguem a mesma lógica de uma programação estruturada, possuindo assim comandos para operações lógicas. O processo de criação desse arquivo chama-se programação e pode ser feito de forma *on-line* e *off-line*.

Na forma *on-line*, o programa é criado diretamente no robô e sua programação é feita através de sua interface homem-máquina, chamada de *Teach Pendant*. Ali são inseridos os tipos de comando desejados para uma certa tarefa, ou elementos de lógicas. Quando os comandos de trajetórias são programados por esse método *on-line*, é necessário que o operador marque um ponto inicial e o ponto final para que o programa interpole com o tipo de trajetória desejada (SENAI-DR, 2004).

A programação *off-line* é feita através de um computador, gerando o arquivo de programa para ser exportado posteriormente para o robô. Da mesma forma que a programação *on-line*, pode-se criar toda a estrutura do programa deixando apenas os pontos para serem marcados quando o programa estiver no robô. Porém, esse método permite que sejam criados os pontos virtualmente ou, dependendo do controlador, inserir trajetórias através de equações matemáticas. A grande desvantagem de se usar a criação virtual de trajetória é a dificuldade de se conseguir modelar com precisão as distâncias entre elementos que compõem o sistema onde o robô está inserido. Porém, esse tipo de programação faz-se útil quando se precisa fazer a ferramenta percorrer trajetórias complexas (SENAI-DR, 2004).

No Anexo 1 são mostrados os comandos da linguagem Rapid que foram utilizados na confecção do pós-processador e é feita também uma comparação com os comandos similares no código G.

6 MÉTODO DE EXECUÇÃO

O projeto foi iniciado com uma revisão bibliográfica a respeito de processos de usinagem, em especial o fresamento com o auxílio de ferramentas CAD/CAM e máquinas CNC, ampliando assim a visão geral do projeto.

Com a visão expandida, foi feito um primeiro contato com o software NX 5, possibilitando assim a ambientação e facilitando a compreensão nos estudos básicos para operar a ferramenta CAM, tais como: importação de arquivos do sistema CAD, preparação do arquivo CAD, parametrização de estratégias para geração de trajetória, verificação e simulação para avaliação de possibilidades de choques e colisões, e otimização das trajetórias criadas.

Dominadas estas funções básicas, foi feita uma imersão no assunto foco do projeto, o pós-processamento. Nesta etapa, confeccionou-se um pós-processador que permite a execução por um robô das trajetórias geradas. Para tal, foi necessário o domínio na criação de pós-processadores – atividade na qual o software NX 5 possui um aplicativo especial – que foi obtido através do contato diário com o software e com a criação de pós-processadores para aplicações comuns, como tornos e centros de usinagem em 3-eixos. Fez parte desta atividade, a confecção de pós-processadores para a interação do software NX 5 com as seguintes máquinas do SENAI CIMATEC: Discovery 1250, Discovery 560, Discovery 4022 e Hermle C 800 U.

Posteriormente realizou-se um estudo direcionado da cinemática do Robô ABB IRB 2400 e sua linguagem de programação, Rapid, com o objetivo de determinar quais comandos e informações seriam necessários para a confecção do pós-processador, como sintaxe de linhas de comando e limites do volume de trabalho.

Por fim, o pós-processador gerado foi validado através do fresamento de uma peça. Para isso acoplou-se ao robô um cabeçote fresador, e o fresamento da superfície complexa de teste foi realizado em um bloco de resina.

7 IMPLEMENTAÇÃO E VALIDAÇÃO

Inicialmente pretendia-se criar um pós-processador compatível com a linguagem do robô FANUC ARC Mate 50 iL. Apesar de ele suportar apenas 3 kg na ponta da ferramenta, era a opção mais favorável devido as suas menores dimensões, peso e disponibilidade. Porém, descobriu-se no andamento do projeto que o arquivo de programa dele não possui um padrão convencional, impossibilitando assim a exportação de arquivo de programa direto para o seu controlador.

Foi avaliada a possibilidade de importar o arquivo pós-processado no NX para o *software* de programação *offline* do robô FANUC, WINTPE, o que foi feito com sucesso devido ao fato desse *software* aceitar arquivos no padrão ASCII. No entanto, descobriu-se posteriormente que a licença dele estava vencida, impossibilitando assim a utilização de todos os seus recursos.

Como existia outro robô disponível, o ABB IRB 2400L, e o arquivo de entrada do seu controlador é em padrão ASCII, optou-se por trocar de robô no lugar de adquirir a licença para o WINTPE. A mudança do robô foi estratégica, pois, mesmo com a licença, havia dúvidas quanto à possibilidade de adequação dos códigos pós-processados para o robô FANUC devido à grande diferença de estrutura de programação entre o código G e a linguagem do robô FANUC.

Logo abaixo será demonstrado como foi feito o processo de implementação através do *software* Post Builder e, posteriormente, é descrito o processo de validação do pós-processador confeccionado.

7.1 Implementação

O Post Builder é uma ferramenta que auxilia na construção do pós-processador para o módulo de *Manufacturing* do NX5. A construção do pós-processador é feita a partir de um pós-processador genérico que pode ser usado para centros de usinagem, tornos ou máquinas de eletroerosão. Falando especificamente de pós-processadores genéricos para centro de usinagem, têm-se os que suportam máquinas de 3-eixos, 4-eixos com mesa ou cabeçote giratório e 5-

eixos com movimento rotativo somente na mesa, no cabeçote ou misto. Começou-se escolhendo o pós-processador genérico para centros de usinagem em 3-eixos, configurado para usar sistema internacional de medidas.

O Post Builder é dividido em cinco abas principais, são elas (Figura 5):

- *Machine Tool*;
- *Program & Tool Path*;
- *N/C Data Definitions*;
- *Output Settings*;
- *Post Files Preview*.

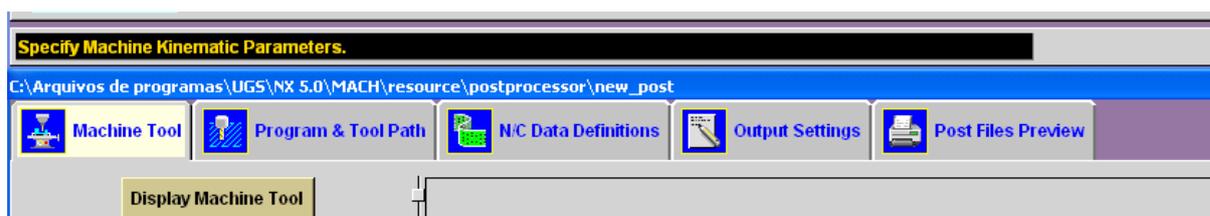


Figura 5 – Abas do Post Builder.

7.1.1 Machine Tool

Nessa aba são realizadas as configurações de parâmetros físicos da máquina. Aqui definem-se a configuração da resolução da máquina, a velocidade máxima de rotação do cabeçote, dimensões máximas dos eixos, e localização do *home* (origem ou zero-máquina). Define-se também se o programa deve gerar movimentos circulares ou não, entre outros elementos.

Dentre esses parâmetros, tomou-se cuidado para estabelecer a velocidade máxima e a resolução compatível com o robô, para que não aconteçam erros estruturais de programa quando o robô entrar em execução. A dimensão da máquina não foi limitada, pois não é o foco do trabalho lidar com a parte cinemática do robô mas, pode-se fazer facilmente essa determinação com as informações de volume de trabalho do robô que é fornecido no manual do fabricante. É importante frisar também que dependendo da altura e distância da peça a ser usinada em relação ao robô, as dimensões máximas em cada eixo irão mudar devido às

características volumétrica de trabalho do robô antropomórfico que é no formato de uma esfera.

Em primeira instância desejou-se construir dois pós-processadores, um que só gerasse trajetórias em movimentos lineares e outro que gerasse trajetórias em movimentos lineares e circulares, porém deparou-se com um problema do software NX em que as trajetórias circulares geradas eram incompatíveis, mas, esse fato será discutido posteriormente. Por esse motivo só foi construído pós-processador com a capacidade de gerar movimentos lineares.

7.1.2 Program & Tool Path

Esse tópico é onde se exige o maior tempo de trabalho, pois é nele que são feitas as principais modificações e adequações. Ele é subdividido em sete sub-tópicos (Figura 6).

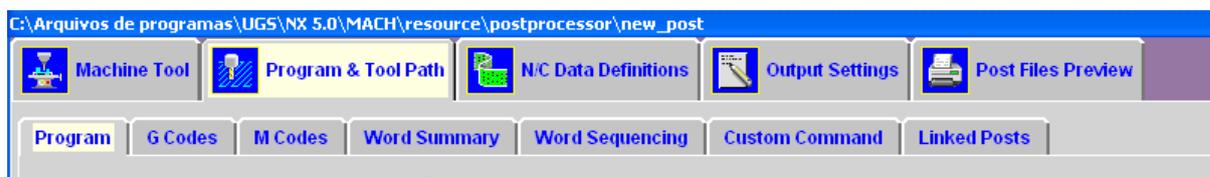


Figura 6 – Sub-tópicos da aba *Program & Tool Path*.

Program

Esse sub-tópico traz as informações da estrutura do programa após ser pós-processado. Aqui se configuram as funções que devem estar presente na etapa de geração de trajetória, mas para este trabalho, basicamente foi feita a retirada de funções do código G que o robô não necessita. Primeiro se define como deve ser o início do programa, cabeçalho, para isso teve que ser criado um comando personalizado (será abordado posteriormente) para poder compatibilizá-lo com o robô. Esse fato ocorreu porque necessitou-se criar algumas rotinas, como a de parametrização do zero peça. Nas máquinas CNC convencionais essa rotina já é inerente ao sistema, porém o robô não a possui. Foi também retirada a função que numera as linhas. No caso da linguagem robótica RAPID a numeração no início de cada linha não é necessária.

Posteriormente foram configuradas as operações que devem ser executadas no *Start* da máquina, como a troca de ferramenta. Não foi necessário configurar nenhuma operação nem para o início e nem para o final de programa.

Teve-se que voltar inúmeras vezes a esse processo para verificar se as configurações, anteriormente feitas em relação à sintaxe do código G, estavam compatíveis com o que se deseja quando o arquivo neutro fosse pós-processado. Foi a partir dessa etapa que pôde se visualizar como seria feita a compatibilização do código G e a forma de como deveria ser sua sintaxe. Para finalizar, foram adicionados dois blocos com as instruções necessárias para que o robô entendesse que o programa acabaria naquele ponto.

G Codes e M Codes

No programa, as funções G e M são agrupadas em relação ao tipo de tarefa a qual se destinam. Por exemplo, o grupo *G_motions* agrupa as funções do código G que tem relação à movimentação, como G00, G01, G02, entre outras. Deve-se reparar que o parâmetro G, prefixo, é constante para todas essas funções, porém a numeração, sufixo, muda mediante o tipo de movimento que se deseja fazer. Logo, nesses dois sub-tópicos é possível mudar o sufixo referente ao tipo de processo que a função vai executar.

Ainda analisando o caso acima, os valores de 00 e 01 foram trocados no projeto por J e L que, junto com o prefixo, significam para o robô respectivamente movimentos rápidos e movimentos lineares (Figura 7). Em determinadas funções M, os sufixos foram trocados por *Low* e *High* que, associados ao seu prefixo, terão a mesma sintaxe de um comando de controle de porta digital do robô.

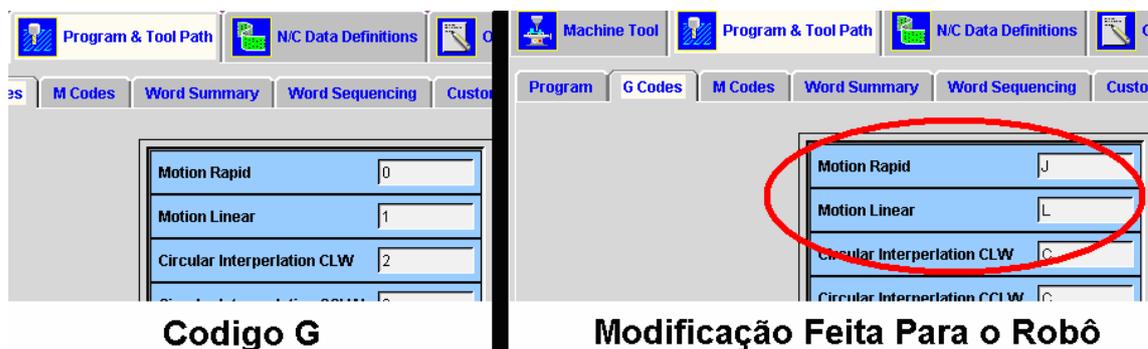


Figura 7 – Modificação do G Codes.

Word Summary

Nesse sub-tópico é feita a configuração das funções ou de um grupo de funções. Nele é possível mudar o seu nome, seu prefixo, definir se o sufixo vai ser um texto ou um número. Em muitos casos, como o grupo *G_Motion*, foi necessário trocar o tipo de dado do sufixo, de número para texto, possibilitando assim o uso dos sufixos J e L.

Quando definido o sufixo como um número, é possível definir a quantidade de casas decimais para direita e para a esquerda, se deve complementar com zeros, uso de ponto ou de vírgula, entre outras configurações. Voltando ao caso do *G_Motion*, o prefixo G foi trocado por Move, para que assim fosse formado, junto com o sufixo já configurado, as instruções MoveL e MoveJ.

Custom Command

Esse sub-tópico destina-se à definição de comandos personalizados, similar a uma área de programação do tipo que se usa na linguagem C. Para certos tipos de máquinas, faz-se necessário criar algumas rotinas que não são contempladas pelo programa, para solucionar esse problema, nesse sub-tópico é possível criar uma linha de programa utilizando a linguagem de programação do NX.

No caso específico do projeto, criaram-se alguns comandos personalizados. Um deles foi colocado na rotina de início de programa no sub-tópico *Program*. Nesse comando personalizado foi feito apenas o uso de comandos que mandam escrever um texto no arquivo quando é pós-processado. Fez-se necessária a utilização dessa ferramenta, pois o cabeçalho do programa requerido para o robô utilizado é relativamente grande, e ainda teve-se que criar a rotina de zero peça.

Assim, no início do programa pós-processado para o robô aparecem algumas linhas que contêm comandos do tipo, mostrar na tela do robô uma mensagem, “Vá até o ponto x inicial e aperte *Start*”, capturar o valor do ponto, guardar os pontos de X, Y e Z em uma variável.

Há ainda os campos **Word Sequencing** e **Linked Post** os quais não tiveram utilidade para o trabalho.

7.1.3 NC data Definitions

Nesse tópico definem-se basicamente as propriedades de cada função ou grupo de função, quais funções parâmetros devem estar presente em cada rotina e também é possível configurar em que ordem vão aparecer.

Na propriedade de cada função é possível configurar para que no final delas seja acrescentado um caractere. Foi muito usada essa opção para poder acrescentar “;” no final de algumas funções. Em funções como as X, Y e Z são criados valores e é nessa área que se pode definir qual o valor máximo, mínimo, quantidades de casas antes e depois da vírgula, entre outros elementos.

Foi necessário compatibilizar as informações que deveriam conter em cada rotina, como a rotina do movimento linear, e ajustar essas informações para que ficassem na mesma sintaxe do robô. Para isso teve-se que acrescentar blocos do tipo texto. No código G é possível fazer a supressão de algumas funções que não serão utilizadas para aquela linha de código pós-processado, já no robô isso não é possível.

Como exemplo podemos falar do movimento linear, no código G quando existem várias linhas que farão movimentos lineares, só é necessário acrescentar o comando G01 em uma linha e nas subseqüentes apenas apresentar o X, Y e Z sendo que se o movimento só variar em X, pode-se suprimir as variáveis Y e Z. Já no robô não é possível fazer isso pois ele necessita de todas essas informações em uma linha só, por isso torna-se necessário fazer com que em cada linha seja obrigado a aparecer um grupo de informações onde todas elas devem estar presentes.

Pode-se determinar se todas as funções e textos terão espaçamento ou se apenas algumas delas, entre outras modificações. Foi nessa área onde realmente se fez a compatibilidade de linguagem para que a saída do arquivo pós-processado tivesse a mesma estrutura do arquivo do programa do robô.

7.1.4 Output Settings

Nesse tópico podem-se fazer pequenas variações, como preencher espaço entre dois elementos, por exemplo, colocar o caractere “_” no lugar de espaço (“ ”). Pode-se definir também se será colocado algum caractere no final de todas as linhas. No caso do robô, essa foi a primeira estratégia pensada para poder colocar o caractere “;” no final das linhas, porém essa estratégia foi descartada porque há linhas no início e no final do programa do robô que não necessitam desse caractere.

Como substituição a essa estratégia esses caracteres foram colocados no final de algumas funções que estivessem localizadas na última posição de cada linha. A modificação mais importante feita nessa área foi mudar a extensão do arquivo pós-processado para “.prg”, fazendo com que o arquivo possa ser inserido diretamente no robô, uma vez que é esta a extensão de entrada de arquivo do robô ABB IRB 2400.

7.1.5 Post Files Preview

Nessa área não são feitas modificações, apenas é mostrado como fica o programa de acordo com as funções modificadas.

7.1.6 Considerações

No decorrer do processo da construção do pós-processador para o robô ABB IRB 2400 debateu-se com uma dificuldade devido à estrutura de programação fechada do programa Post Builder. Ao gerar movimentos rápidos necessitava-se colocar as coordenadas de X, Y e Z em lugares específicos, como se fez com os movimentos lineares, compatibilizando assim o arquivo gerado com o código do robô. Porém, apesar de que no pós-processador fosse possível colocar essas variáveis na posição desejada, toda vez que se pós-processava um arquivo de trajetória essas variáveis iam para o final da linha nas funções de movimentação rápida.

Demandou-se um pouco de tempo para poder solucionar o problema, e a solução encontrada não é a mais apropriada. O pós-processador é formado por 3 arquivos e descobriu-se que esse problema pode ser solucionado modificando-se o arquivo .def. Nesse arquivo tem-se a estrutura de todos os movimentos, toda vez que o pós-processador é salvo essa estrutura é re-escrita e percebe-se que sempre os pontos X, Y e Z vão para o final da linha nos movimentos do tipo rápido.

Para resolver esse problema é feito a modificação manual do arquivo .def ,após salvar o pós-processador, colocando esses elementos na posição desejada. É importante frisar que, toda vez que se altera o pós-processador utilizando-se o Post Builder, é necessário refazer essas modificações. Acredita-se que esse problema seja inerente à estrutura relativamente fechada do Post Builder na geração do arquivo .def.

Outra dificuldade encontrada na construção do pós-processador esta relacionado a geração de trajetória circular. Na aba *Machine Tool* desejou-se habilitar o pós-processador para executar tanto operações com interpolações lineares como circulares. Porém, ao gerar programas com interpolação circular utilizando um pós-processador genérico, observou-se que o NX5 estava gerando pontos que não correspondiam a um movimento circular. Essa conclusão pode ser tirada pelo fato de que na geração do ponto que indica o centro do raio, pode-se perceber que a distância entre esse ponto e ponto final e inicial do movimento eram diferentes, caracterizando assim um movimento não circular.

7.2 Validação

Para validar o projeto foi necessário habilitar fisicamente o robô com ferramentas para que o mesmo pudesse executar operações de usinagem. Os elementos físicos necessários para isso são descrito no sub-item Aparato para Validação. Precisava-se também de alguma geometria em CAD para poder gerar as trajetórias, com isso optou-se por confeccionar uma peça virtual para, a partir desta, criar trajetórias de ferramenta para o que o robô as seguisse, como descrito nos capítulos posteriores.

7.2.1 Aparato para Validação

Para poder validar o projeto, necessitava-se de um aparato, a ser acoplado no robô, capaz de fazer com que a fresa executasse seu movimento de corte. Para a confecção do aparato, a alternativa escolhida foi a mais simples e com menor custo devido à indisponibilidade de recurso e tempo. Com isso escolheu-se acoplar ao robô uma retificadora manual, também chamada de tupia, já que ela permite a utilização de uma ferramenta rotativa, como uma fresa, e já possui uma região adequada para acoplamento ao robô.

Utilizando esse equipamento o único trabalho despendido foi à criação de um suporte capaz de ser acoplado na retificadora manual e no robô (Figura 8). Esse suporte possui características muito simples e foi confeccionado através do fresamento de uma chapa de alumínio.



Figura 8 – Suporte da Retificadora/Robô.

Foi sugerido criar um mecanismo para a retificadora manual poder controlar a velocidade de rotação da fresa, como já contemplado no pós-processador. Esse mecanismo seria composto de um CLP com entrada analógica que estaria ligada a saída analógica do robô, saída essa responsável por mostrar um valor equivalente à velocidade de rotação da fresa descrita no programa de trajetória de ferramenta, uma entrada de alta velocidade para poder contar os giros da retificadora manual e uma saída para disparar o *triac* em um circuito do tipo *dimmer*. Com esses elementos formar-se-ia um sistema de malha fechada capaz de controlar a tensão de entrada da retificadora manual para assim alcançar a velocidade desejada para o programa CNC.

Devido à dificuldade de acoplar um sensor para contar o giro da retificadora manual, associada à escassez de tempo, foi utilizado um controle manual através de um transformador variável. A utilização desse elemento tinha o propósito de, apenas através da variação de tensão, definir um valor constante de velocidade de rotação.

A melhor forma de controlar a velocidade de um motor de corrente alternada é variar a frequência de alimentação, sendo geralmente necessário o uso de um equipamento específico, o inversor de frequência. No caso não se optou por usar esse último método porque seria necessário adquirir um inversor de frequência monofásico já que não se possuía.

Para o processo de validação, deixou-se disponível um aspirador de pó que foi usado tanto na limpeza em geral como também como um elemento para extrair cavaco do processo de usinagem executado pelo robô, muito necessário, já que a geometria a ser usinada era propícia ao acúmulo de cavacos, podendo assim influenciar negativamente nos resultados da usinagem. Foram usadas também uma mesa de seno e uma morsa para poder prender a peça a ser usinada. A Figura 9 apresenta um esquema sucinto do sistema feito para o robô executar as operações de usinagem.

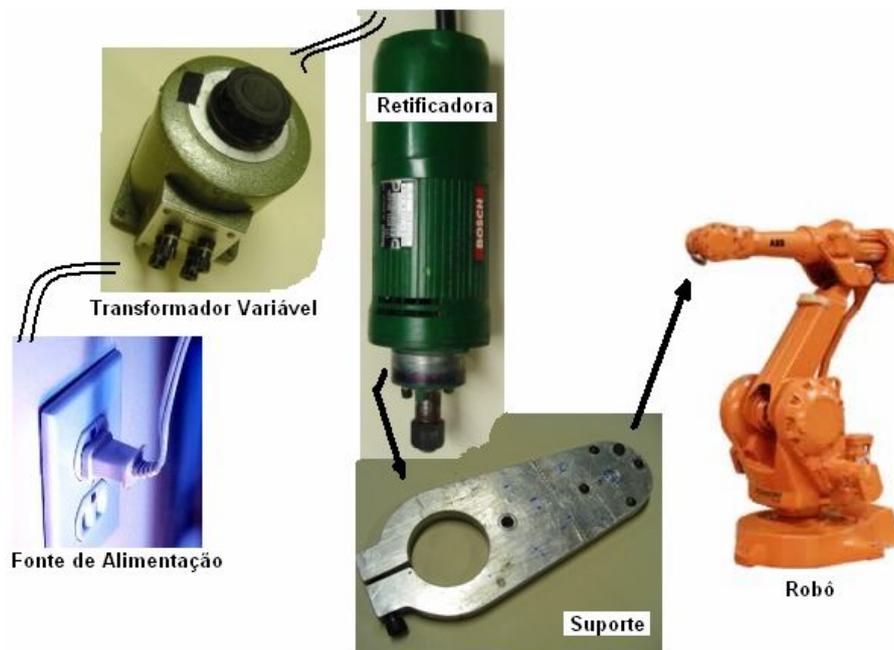


Figura 9 – Esquema do sistema para validação.

7.2.2 Peça a ser Usinada

Com o objetivo de suportar a idéia geral desse projeto, foi escolhido validá-lo através da usinagem de um molde de um porta-pilhas de um lampião de LED.

O porta-pilhas possui uma geometria complexa capaz de suportar três pilhas dentro de uma carcaça circular (Figura 10), e seu modelo virtual foi gerado no módulo *Modelling* do *software* NX5. Para a confecção do molde, optou-se por uma resina sintética, tanto por sua baixa resistência ao cisalhamento como por sua disponibilidade.

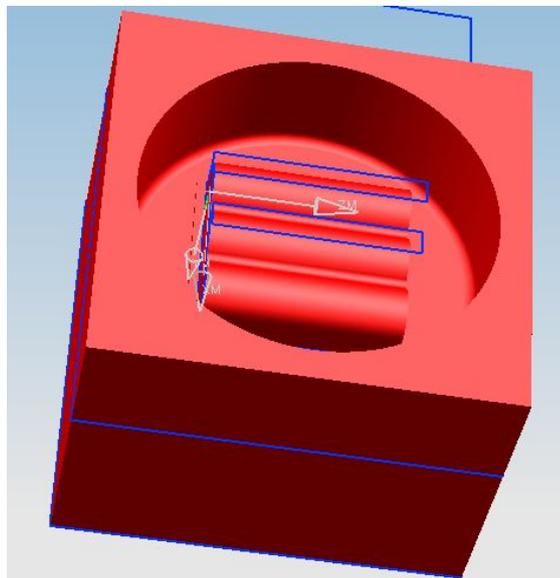


Figura 10 – Modelo Virtual da Peça a ser Usinada.

7.2.3 Usinagem utilizando o Robô

Para usinagem da peça foram feitos oito programas de trajetória de ferramenta. Inicialmente, os programas iniciariam de um ponto conhecido e, após a usinagem, parariam em um ponto conveniente ao *software* NX 5. Porém em testes preliminares foi notada a necessidade dos programas começarem e acabarem em um mesmo ponto, facilitando assim a referência da ferramenta na mudança do programa no robô. Todos os programas de desbaste utilizaram velocidade de avanço de 10400 mm/min e os programas de acabamento utilizaram a velocidade de

5000 mm/min, essas velocidades são relativamente altas para operações de usinagem.

No Anexo 2 é apresentado um resumo com os principais parâmetros tecnológicos preenchidos nas abas de programação do *software* CAM para os oito programas gerados. É interessante discorrer um pouco mais sobre o primeiro programa devido ao fato dele possuir o objetivo de testar um pouco da capacidade em relação ao esforço do sistema, robô-retificadora manual. Para isso optou-se por usar profundidade de corte muito superior em relação aos demais programas, 5 mm, e a fresa usada foi de 12 mm de diâmetro, fazendo com que esse programa demandasse, dentre todos os outros, maior esforço do sistema. Esse programa tem o objetivo de facear a superfície da peça fazendo com que o plano XY da superfície da peça fosse igual a o plano XY do robô e também esse processo corrigiu qualquer imperfeição na superfície da peça.

Após a confecção de todos os programas, os mesmos foram pós-processados utilizando o pós-processador confeccionado deixando assim os arquivos de saída prontos para serem executados no robô. No Anexo 3 é mostrado o programa 5 pós-processado em código G e no Anexo 4 é mostrado o mesmo programa, porém pós-processado em linguagem Rapid.

Inicialmente, foram testados todos os programas em “vazio” para verificar possíveis incompatibilidades de trajetórias. Com todos os programas verificados e aparentemente sem nenhuma incompatibilidade, fixou-se na morsa o bloco de resina e embaixo dele uma chapa de alumínio com o objetivo inicial de poder ajudar na referência da ferramenta no eixo Z. Depois se aproximou a ponta da ferramenta ao bloco e fez-se movê-la no eixo X com o objetivo de alinhar a o bloco em relação ao robô. Em seguida, foi colocada a ferramenta na posição inicial, ajustou-se o transformador variável para 70% de tensão e executou-se o primeiro programa.

Logo na primeira passada, a retificadora manual não agüentou o esforço e parou de rodar fazendo com que o robô estagnasse e não conseguisse continuar a sua trajetória e exibindo a mensagem erro por excesso de esforço nas juntas. Foi feito novamente o alinhamento da peça e a referenciação da ferramenta e executou-se o primeiro programa de ferramenta novamente. Com o transformador variável a 100% o robô conseguiu facear normalmente o bloco. Acredita-se que com a redução da tensão, e sua conseqüente redução da potência, houve uma sobrecarga no motor

da retificadora manual não atendendo à demanda da usinagem programada. Nos demais programas não ocorreu nenhum problema.

7.2.4 Análise da peça usinada

Apesar deste trabalho não se propor a fazer uma análise da capacidade e erro fornecido pelo robô no processo de usinagem, devido ao resultado final, fez-se interessante abordar esse assunto. Reparou-se que houve algumas falhas, porém, como a idéia desse projeto é mostrar a possibilidade do robô gerar peças tridimensionais sem se preocupar muito com as características de precisão dimensional, o processo saiu como esperado. A Figura 11 mostra a peça usinada.

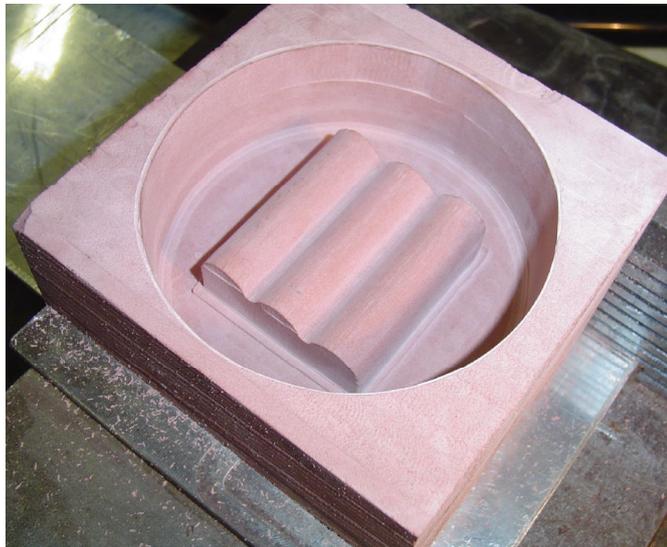


Figura 11 – Peça Usinada pelo Robô.

Verificaram-se duas fontes de erros que interferiram na precisão dimensional da peça. A fonte conhecida está relacionada com a referência da ferramenta. Toda vez que se mudava o programa era necessário zerar a ferramenta. Apesar de optar-se por uma estratégia na qual a ferramenta começaria e terminaria no mesmo ponto, tentando assim evitar a movimentação da ferramenta antes de referenciá-la, houve um problema inesperado.

O referenciamento era feito no início de cada programa com o objetivo de informar para o robô que é a partir deste ponto que ele deveria fazer os incrementos nos movimentos. Porém, para gravar um ponto, estava sendo necessário executar a

operação de gravar ponto, mover o robô, para depois executar novamente a operação de gravar ponto. Só assim o robô aceitava gravar o ponto. Com isso independente de onde a ferramenta parava no programa anterior, era necessário movê-la para conseguir zerá-la.

Devido ao fato de ter que zerar a ferramenta nos três eixos a cada programa associado a uma metodologia não adequada para essa tarefa, observou-se erros na usinagem que podem ter se originado por esse motivo (Figura 12).



Figura 12 – Erro ocasionado pelo processo de referênciação.

Observou-se também erros da peça que não foram rastreados, pois não era o foco do projeto. Porém acredita-se que esses erros são oriundos devido ao robô não suportar mudanças bruscas de trajetórias (Figura 13). Acredita-se nessa hipótese pela observação da região que possui esse erro com a trajetória da ferramenta.



Figura 13 – Erro possivelmente ocasionado pelo robô.

8 CONCLUSÕES

Com o objetivo de reduzir custos na fabricação de protótipos funcionais, esse trabalho buscou viabilizar a utilização de um robô para executar operações de fresamento. Para tal, criou-se um pós-processador capaz de converter informações de arquivos neutros gerados em um sistema CAM comercial em informações compreendidas pelo robô.

Para poder embasar esse trabalho com informações válidas para o entendimento do processo, foi realizada uma revisão bibliográfica abordando os principais conceitos. Começou-se por abordar o conceito do código G para depois discorrer sobre CAM, mostrando-se assim toda a cadeia necessária para a usinagem de peças utilizando essa ferramenta. Posteriormente discorreu-se sobre pós-processadores e robôs industriais possibilitando a visualização dos conhecimentos necessários para a criação do pós-processador que comunique o software CAM com o robô utilizado para o estudo.

A etapa de confecção do pós-processador se deu com o auxílio da ferramenta Post Builder do *software* NX 5. Partiu-se de uma de suas opções padronizadas, no caso a de um pós-processador para um centro de usinagem de três eixos, e as principais alterações foram: a definição dos parâmetros físicos da máquina (resolução, velocidade máxima do cabeçote, dimensões máximas dos eixos), escolha do tipo de interpolação (linear ou circular), estrutura do programa de trajetória de ferramenta (cabeçalho, corpo e fim de programa) e a compatibilização da sintaxe dos comandos do código G com os comandos do robô, como pode ser visto no Anexo 1.

As principais dificuldades encontradas nesta etapa foram na geração de movimento circular e nas linhas referentes a movimento rápido do programa de trajetória de ferramenta. Ao tentar gerar programas com interpolações circular utilizando um pós-processador genérico, observou-se que o NX5 estava gerando pontos que não correspondiam a um movimento circular. Por esse motivo optou-se em, apenas, confeccionar um pós-processador de interpolação linear. O problema que se pode ter em usar só interpolação linear em geometrias complexas está ligado

ao tamanho do programa de trajetória de ferramenta. Em testes viu-se que o robô ABB IRB 2400L suporta programas entre 10000 e 11000 linhas.

Outra dificuldade na criação do pós-processador foi observada ao gerar o primeiro programa de trajetória de ferramenta usando o pós-processador confeccionado. Observou-se que as linhas referentes a movimentos rápidos não estavam obedecendo à sintaxe descrita no programa de auxílio à confecção de pós-processador, o Post Builder. Percebeu-se que independente da posição que se colocavam as variáveis referentes às coordenadas, no arquivo de trajetórias de ferramenta elas sempre apareciam no final da linha de movimento rápida.

Para solucionar esse impasse foi necessário re-editar “manualmente” o arquivo *.def do pós-processador sem utilizar o Post Builder. Fez-se isso abrindo-o no editor de texto ASCII e re-ordenando as linhas referentes a movimentação rápida. Porém todas as vezes que se salva o pós-processador utilizando o Post Builder é necessário re-editar o arquivo *.def. Foi feita uma pesquisa para tentar solucionar esse problema de outra forma, mas acredita-se que a estrutura fechada do *software* não permita certo tipos de alterações.

Após a finalização da confecção do pós-processador foi necessário testá-lo para assim poder validar o método proposto. Inicialmente seria feita um teste virtual através do software *WorkSpace*, porém, com o entusiasmo do projeto, optou-se por validar o pós-processador através da usinagem de uma peça. Para tal, criou-se um dispositivo para montar uma tupia, destinada a acionar as fresas, na ponta do robô e usinou-se uma peça-teste em resina.

Durante a usinagem da peça defrontou-se com o problema da impossibilidade de manutenção do ponto de referência para programas distintos. Isso implicou na necessidade de que, para cada programa, fosse feita uma nova referência, o que compromete a precisão geométrica e de forma da peça em processamento.

Através da análise visual da peça após a usinagem percebeu-se que houve outros desvios de forma, oriundos de fontes não citadas acima. Acredita-se que esses desvios podem ser conseqüência de limitações tanto construtivas do robô (carência de rigidez e cinematismo em série) quanto de seu sistema de controle, fatores que podem agravados pelas altas velocidades de deslocamento utilizadas.

No entanto, considera-se que, apesar de haver desvios de forma visíveis, os erros geométricos não extrapolam o limite do aceitável, e o principal objetivo deste trabalho foi atingido, uma vez que foi possível usinar uma peça com distintas operações, utilizando-se um robô industrial, com programas gerados em um sistema CAM comercial.

Sugestão para Trabalhos Futuros

Durante a execução deste trabalho percebeu-se a necessidade de maior desenvolvimento de conhecimento em outras áreas. Abaixo são sugeridos alguns trabalhos:

- Construção de pós-processador de usinagem em 5-eixos para robô;
- Estudo da capacidade de força de usinagem que o robô pode fornecer junto com um rastreamento de erros ocasionados pelo robô;
- Estudo de materiais para serem usinados com robô;
- Simulação da dinâmica do robô em ambientes de simulação de máquinas CNC, com o objetivo de verificar possíveis choques entre robô e peça;
- Metodologia para confecção de protótipos funcionais com o uso de matrizes poliméricas reforçadas com fibra através da usinagem de moldes utilizando robô.

REFERÊNCIAS

CAVALHEIRO, A. Z.. **Sistematização do planejamento da programação via CAM do fresamento de cavidades de moldes para peças de injeção**. 1998. 215p. Dissertação (Mestrado em Engenharia Mecânica) – Universidade Federal de Santa Catarina, Florianópolis, 1998.

MASTELARI, N.. **Contribuição ao Processo de Integração de Informações da Manufatura para Empresas de Pequeno e Médio Porte**. 2004. 186p. Tese (Doutorado em Engenharia Mecânica) – Departamento de Engenharia de Fabricação, Universidade Estadual de Campinas, Campinas 2004.

MENEGHELLO, G. P.. **Aplicação de um sistema robótico utilizando recursos de sistemas CAD/CAM para o processo de fresamento**. 2003. 119p. Dissertação (Mestrado em Engenharia Mecânica) – PROMEC, Universidade Federal do Rio Grande do Sul, Porto Alegre. 2003

PEREIRA, A. G.. **Desenvolvimento e avaliação de um editor para programação CN em centros de usinagem**. 2003. 122p. Dissertação (Mestrado em Engenharia Mecânica) - Setor de Tecnologia da Universidade Federal do Paraná, Curitiba 2003.

ROMANO, V. F. **Robótica industrial: Aplicação na indústria de manufatura e de processo**, Edgard Blücher, São Paulo, 2002.

SENAI-DR BA. **Fresagem**. Salvador, 2005. (Rev.00)

SENAI-DR BA. **Robótica básica: Robô FANUC e ABB**. Salvador, 2004. 123f. (Rev.00)

SILVA FILHO, M. S.. **Fresamento por regiões como uma alternativa na usinagem de cavidades para moldes de injeção através de sistema CAD/CAM**. 2000. 133p. Dissertação (Mestrado em Engenharia Mecânica) – Universidade Federal de Santa Catarina, Florianópolis, 2000.

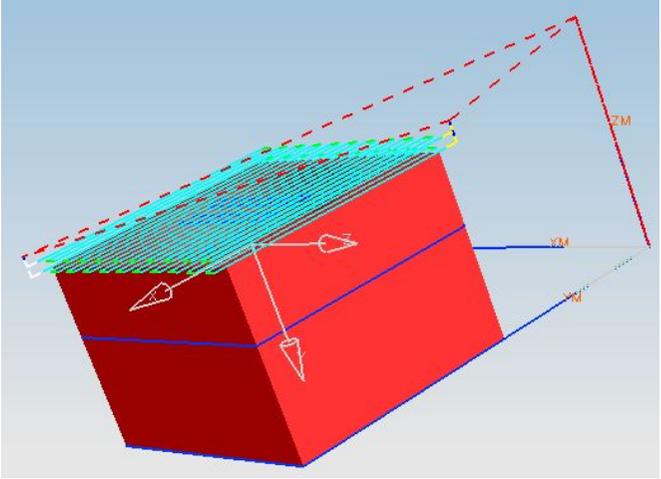
SOUZA, G. O.. **Fresamento em 5-eixos simultâneos de pás de compressores de turbinas a gás em tial6v4**. 2006. 169f. Dissertação (Mestre Em Engenharia Mecânica)- Universidade Federal De Santa Catarina, Florianópolis, 2006.

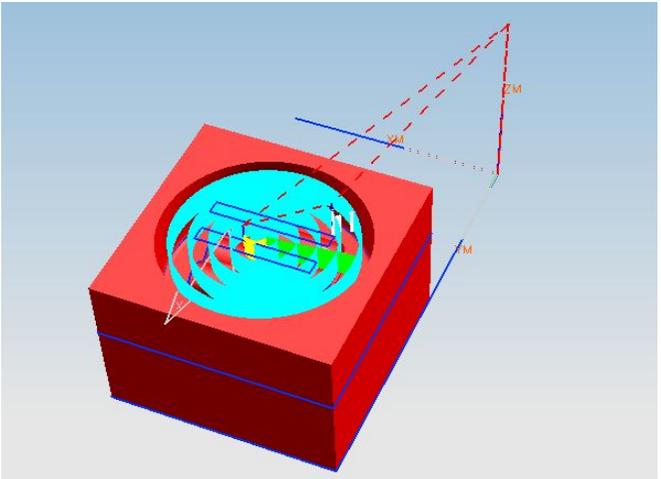
ANEXOS

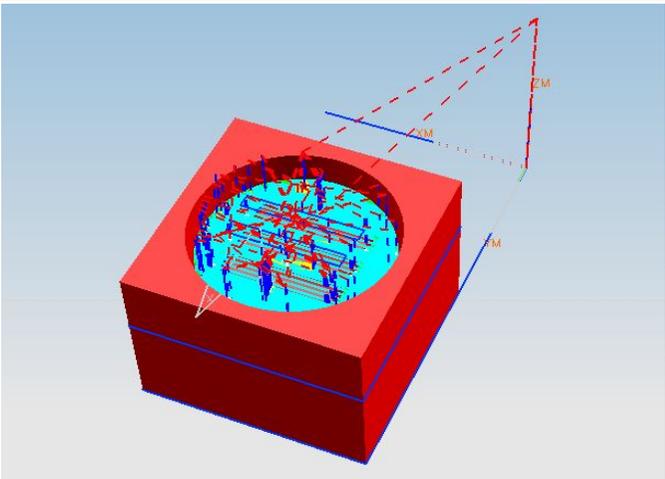
ANEXO 1 – Comandos utilizados do robô.

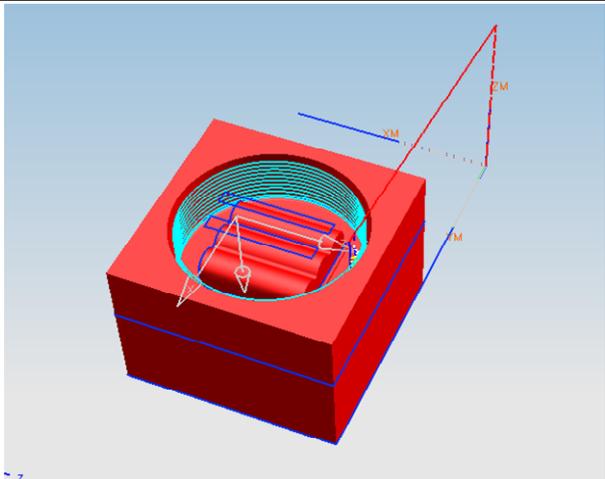
Movej	Similar ao G0 no código G, o comando faz com que o robô se mova em movimento rápido.
Movei	Similar ao G1 no código G, o comando faz com que o robô se mova em trajetória linear.
Waitempty	Similar ao G4 no código G, o comando faz com que o robô fique parado durante um tempo determinado.
Speeddata	Similar ao F (velocidade de avanço) no código G, refere-se à velocidade do ponto de controle.
Zonedata	O comando serve para suavizar a mudança de direção. Com tolerância acima de zero, a ponta da ferramenta muda a sua trajetória antes de chegar a o ponto.
Pulsedo	Comando que gera um pulso na saída digital do robô. Associado com uma variável esse comando substitui o M2 (liga Cabeçote em sentido horário) e o M5 (desliga cabeçote).
Setdo	Comando que muda o estado de uma saída digital. Associado com uma variável esse comando substitui o M8 (liga fluido de corte) e o M9 (desliga fluido de corte).
Setao	Comando que muda o sinal de uma saída Analógica. Associado a um circuito elétrico destinado para tal fim é possível variar a velocidade de giro do cabeçote. Foi usado para substituir S (velocidade de rotação) do código G.
Stop	Similar a o M0 no código G, o comando faz com que o programa pare de ser executado.
Robtarget	O comando captura as coordenadas da ponta da ferramenta. Foi usado para simular a operação de "zeramento" presente nas máquinas CNC.
Offs	Determina um novo sistema de coordenada em cima de um ponto. Logo qualquer movimento que o robô for fazer suas coordenadas vão estar em relação ao ponto e não mas a base do robô.

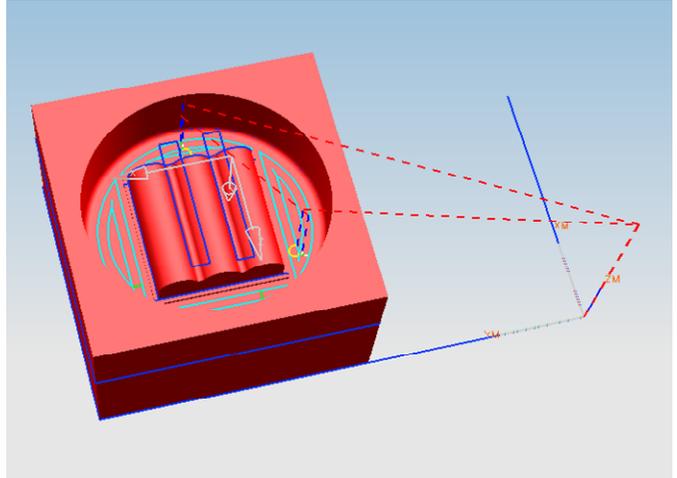
ANEXO 2 – Dados dos Programas de Usinagem.

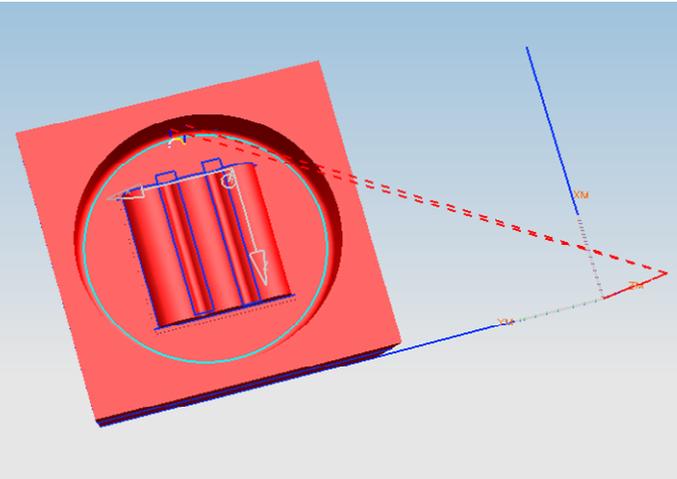
Programa 1		
	Descrição	Desbaste Superior
	Estratégia	Varredura 2 e 1/2 eixos em ZIGZAG
	Fresa	12mm Topo Reto / Z = 2
	Tolerância (mm)	+ 0,12
		- 0,03
	v_c (m/min)	980
	v_f (mmpm)	10400
	f_z (mm)	0,2
	a_p (mm)	5
	a_e	50%
	n (rpm)	26000
	Função NX 5	CAVIT_MILL

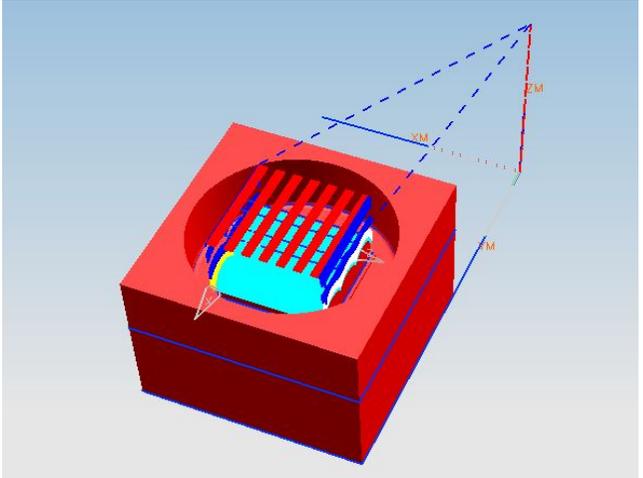
Programa 2		
	Descrição	Desbaste Cilindro Superior
	Estratégia	Perfilamento em 2 e 1/2 eixos
	Fresa	12mm Topo Reto / Z = 2
	Tolerância (mm)	+ 0,12
		- 0,03
	v_c (m/min)	10400
	v_f (mmpm)	980
	f_z (mm)	0,2
	a_p (mm)	0,5
	a_e	80%
	n (rpm)	26000
	Função NX 5	CAVIT_MILL

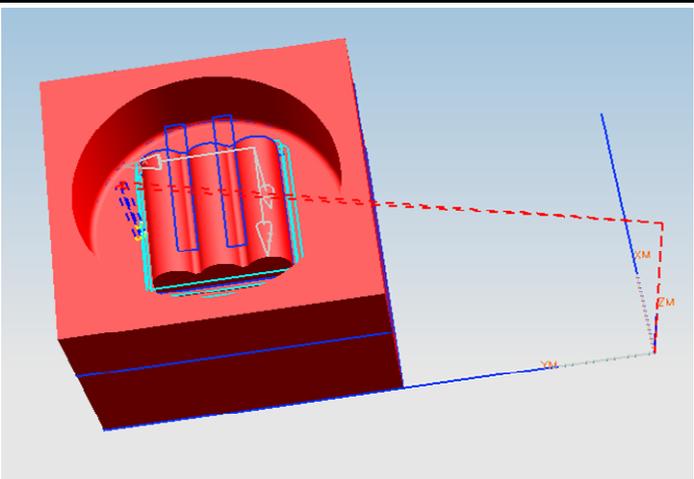
Programa 3		
	Descrição	Desbaste Cilindro Inferior
	Estratégia	Varredura 2 e 1/2 eixos em ZIGZAG
	Fresa	6mm Topo Reto / Z = 2
	Tolerância (mm)	+ 0,12
		- 0,03
	v_c (m/min)	10400
	v_f (mmpm)	980
	f_z (mm)	0,2
	a_p (mm)	0,5
	a_e	50%
n (rpm)	26000	
Função NX 5	CAVIT_MILL	

Programa 4		
	Descrição	Acabamento Lateral
	Estratégia	Perfilamento em 2 e 1/2 Eixos
	Fresa	6mm Topo Reto / Z = 2
	Tolerância (mm)	+ 0,03
		- 0,03
	v_c (m/min)	5000
	v_f (mmpm)	471
	f_z (mm)	0,1
	a_p (mm)	3
	a_e	-----
n (rpm)	26000	
Função NX 5	Planar_Profile	

Programa 5		
	Descrição	Desbaste Inferior
	Estratégia	Usinagem por Offset de Contorno
	Fresa	6mm Topo Reto / Z = 2
	Tolerância (mm)	+ 0,03
		- 0,03
	v_c (m/min)	5000
	v_f (mmpm)	471
	f_z (mm)	0,1
	a_p (mm)	-----
	a_e	70%
n (rpm)	26000	
Função NX 5	Rough_Follow	

Programa 6		
	Descrição	Acabamento Chanfro Inferior
	Estratégia	Perfilamento em 2 e 1/2 Eixos
	Fresa	6mm Topo esférico / Z = 2
	Tolerância (mm)	+ 0,03
		- 0,03
	v_c (m/min)	5000
	v_f (mmpm)	471
	f_z (mm)	0,1
	a_p (mm)	-----
	a_e	-----
n (rpm)	26000	
Função NX 5	Planar_Profile	

Programa 7		
	Descrição	Acabamento da Geometria Porta Pilha
	Estratégia	Isoparamétrica de Superfícies Sequenciais
	Fresa	4mm Topo Eférico / Z = 2
	Tolerância (mm)	+ 0,03
		- 0,03
	v_c (m/min)	5000
	v_f (mmpm)	471
	f_z (mm)	0,1
	a_p (mm)	-----
	a_e	-----
n (rpm)	26000	
Função NX 5	Streamline	

Programa 8		
	Descrição	Acabamento do Ressalto Inferior
	Estratégia	Perfilamento em 2 e 1/2 eixos
	Fresa	4mm Topo Reto / Z = 4
	Tolerância (mm)	+ 0,03
		- 0,03
	v_c (m/min)	5000
	v_f (mmpm)	471
	f_z (mm)	0,05
	a_p (mm)	-----
	a_e	-----
n (rpm)	26000	
Função NX 5	Zlevel_Profile	

ANEXO 3 – 5º Programa de usinagem do porta pilha escrito em código G.

N0010 G40 G17 G90 G71
N0020 G91 G28 Z0.0
N0030 T30 M06
N0040 G0 G90 X0.0 Y0.0 S26000 M03
N0050 G43 Z115.5 H00
N0060 X104.9 Y157.
N0070 Z100.
N0080 Z58.
N0090 G1 Z55. F87. M08
N0100 X105.
N0110 X105.776 Y157.102
N0120 X106.5 Y157.402
N0130 X107.121 Y157.879
N0140 X107.598 Y158.5
N0150 X107.898 Y159.224
N0160 X108. Y160.
N0170 X107.883 Y163.348
N0180 X107.533 Y166.68
N0190 X106.951 Y169.98
N0200 X106.141 Y173.231
N0210 X105.105 Y176.417
N0220 X103.85 Y179.523
N0230 X102.381 Y182.535
N0240 X100.706 Y185.436
N0250 X98.833 Y188.214
N0260 X96.77 Y190.854
N0270 X94.528 Y193.344
N0280 X92.118 Y195.671
N0290 X89.552 Y197.825
N0300 X86.841 Y199.794
N0310 X84. Y201.569
N0320 X81.042 Y203.142
N0330 X77.981 Y204.505
N0340 X74.833 Y205.651
N0350 X71.612 Y206.574
N0360 X68.335 Y207.271
N0370 X65.017 Y207.737
N0380 X61.675 Y207.971
N0390 X58.325
N0400 X54.983 Y207.737
N0410 X51.665 Y207.271
N0420 X48.388 Y206.574
N0430 X45.167 Y205.651
N0440 X42.019 Y204.505
N0450 X38.958 Y203.142
N0460 X36. Y201.569
N0470 X33.159 Y199.794
N0480 X30.448 Y197.825

N0490 X27.882 Y195.671
N0500 X25.472 Y193.344
N0510 X23.23 Y190.854
N0520 X21.167 Y188.214
N0530 X19.294 Y185.436
N0540 X17.619 Y182.535
N0550 X16.15 Y179.523
N0560 X14.895 Y176.417
N0570 X13.859 Y173.231
N0580 X13.049 Y169.98
N0590 X12.467 Y166.68
N0600 X12.117 Y163.348
N0610 X12. Y160.
N0620 X12.117 Y156.652
N0630 X12.467 Y153.32
N0640 X13.049 Y150.02
N0650 X13.859 Y146.769
N0660 X14.895 Y143.583
N0670 X16.15 Y140.477
N0680 X17.619 Y137.465
N0690 X19.294 Y134.564
N0700 X21.167 Y131.786
N0710 X23.23 Y129.146
N0720 X25.472 Y126.656
N0730 X27.882 Y124.329
N0740 X30.448 Y122.175
N0750 X33.159 Y120.206
N0760 X36. Y118.431
N0770 X38.958 Y116.858
N0780 X42.019 Y115.495
N0790 X45.167 Y114.349
N0800 X48.388 Y113.426
N0810 X51.665 Y112.729
N0820 X54.983 Y112.263
N0830 X58.325 Y112.029
N0840 X61.675
N0850 X65.017 Y112.263
N0860 X68.335 Y112.729
N0870 X71.612 Y113.426
N0880 X74.833 Y114.349
N0890 X77.981 Y115.495
N0900 X81.042 Y116.858
N0910 X84. Y118.431
N0920 X86.841 Y120.206
N0930 X89.552 Y122.175
N0940 X92.118 Y124.329
N0950 X94.528 Y126.656
N0960 X96.77 Y129.146
N0970 X98.833 Y131.786
N0980 X100.706 Y134.564

N0990 X102.381 Y137.465
N1000 X103.85 Y140.477
N1010 X105.105 Y143.583
N1020 X106.141 Y146.769
N1030 X106.951 Y150.02
N1040 X107.533 Y153.32
N1050 X107.883 Y156.652
N1060 X108. Y160.
N1070 G0 X107.898 Y160.776
N1080 X107.598 Y161.5
N1090 X107.121 Y162.121
N1100 X106.5 Y162.598
N1110 X105.776 Y162.898
N1120 X105. Y163.
N1130 X104.9
N1140 Z58.
N1150 Z100.
N1160 Z115.5
N1170 X0.0 Y0.0
N1180 Z0.0
N1190 M02

ANEXO 4 – 5º Programa de usinagem do porta pilha escrito em RAPID.

```

%%
VERSION:1
LANGUAGE:ENGLISH
%%
MODULE Fresamento
VAR robtarget Ref;
PROC main()
VAR num zon:=201;
VAR zonedata zone:=z1;
VAR speeddata v:=v10;
VAR robtarget px;
VAR robtarget py;
VAR robtarget pz;
TPWrite "Coloque a Ferramenta";
Stop \noregain;
TPWrite "Referencie em X!";
Stop \noregain;
px:=Crobot();
TPWrite "Referencie em Y!";
Stop \noregain;
py:=Crobot();
TPWrite "Referencie em Z!";
Stop \noregain;
pz:=Crobot();
ref:=px;
ref.trans.y:=py.trans.y;
ref.trans.z:=pz.trans.z;
Erro:
TPReadNum zon, "Defina a aproximação: Valores de 0 a 200";
zone.pzone_tcp:=zon;
IF zon > 200 THEN
TPWrite "Valor Incorreto";
GOTO Erro;
ENDIF
IF zon < 0 THEN
TPWrite "Valor Incorreto";
GOTO Erro;
ENDIF
TPWrite "Pronto para Acionar a Ferramenta";
Stop \noregain;
MOVEJ Offs (ref, .000, .000, 115.500), v, zone, tool30;
veloc.v_tcp:=100.;
SetAO Ferramenta_A,2000;
PulseDO Ferramenta_Hr;
MOVEJ Offs (ref, 104.900, 157.000, 100.000), v, zone, tool30;
MOVEJ Offs (ref, 104.900, 157.000, 58.000), v, zone, tool30;

```

MOVEL Offs (ref, 104.900, 157.000, 55.000), v, zone, tool30;
veloc.v_tcp:=87.;

MOVEL Offs (ref, 105.000, 157.000, 55.000), v, zone, tool30;
MOVEL Offs (ref, 105.776, 157.102, 55.000), v, zone, tool30;
MOVEL Offs (ref, 106.500, 157.402, 55.000), v, zone, tool30;
MOVEL Offs (ref, 107.121, 157.879, 55.000), v, zone, tool30;
MOVEL Offs (ref, 107.598, 158.500, 55.000), v, zone, tool30;
MOVEL Offs (ref, 107.898, 159.224, 55.000), v, zone, tool30;
MOVEL Offs (ref, 108.000, 160.000, 55.000), v, zone, tool30;
MOVEL Offs (ref, 107.883, 163.348, 55.000), v, zone, tool30;
MOVEL Offs (ref, 107.533, 166.680, 55.000), v, zone, tool30;
MOVEL Offs (ref, 106.951, 169.980, 55.000), v, zone, tool30;
MOVEL Offs (ref, 106.141, 173.231, 55.000), v, zone, tool30;
MOVEL Offs (ref, 105.105, 176.417, 55.000), v, zone, tool30;
MOVEL Offs (ref, 103.850, 179.523, 55.000), v, zone, tool30;
MOVEL Offs (ref, 102.381, 182.535, 55.000), v, zone, tool30;
MOVEL Offs (ref, 100.706, 185.436, 55.000), v, zone, tool30;
MOVEL Offs (ref, 98.833, 188.214, 55.000), v, zone, tool30;
MOVEL Offs (ref, 96.770, 190.854, 55.000), v, zone, tool30;
MOVEL Offs (ref, 94.528, 193.344, 55.000), v, zone, tool30;
MOVEL Offs (ref, 92.118, 195.671, 55.000), v, zone, tool30;
MOVEL Offs (ref, 89.552, 197.825, 55.000), v, zone, tool30;
MOVEL Offs (ref, 86.841, 199.794, 55.000), v, zone, tool30;
MOVEL Offs (ref, 84.000, 201.569, 55.000), v, zone, tool30;
MOVEL Offs (ref, 81.042, 203.142, 55.000), v, zone, tool30;
MOVEL Offs (ref, 77.981, 204.505, 55.000), v, zone, tool30;
MOVEL Offs (ref, 74.833, 205.651, 55.000), v, zone, tool30;
MOVEL Offs (ref, 71.612, 206.574, 55.000), v, zone, tool30;
MOVEL Offs (ref, 68.335, 207.271, 55.000), v, zone, tool30;
MOVEL Offs (ref, 65.017, 207.737, 55.000), v, zone, tool30;
MOVEL Offs (ref, 61.675, 207.971, 55.000), v, zone, tool30;
MOVEL Offs (ref, 58.325, 207.971, 55.000), v, zone, tool30;
MOVEL Offs (ref, 54.983, 207.737, 55.000), v, zone, tool30;
MOVEL Offs (ref, 51.665, 207.271, 55.000), v, zone, tool30;
MOVEL Offs (ref, 48.388, 206.574, 55.000), v, zone, tool30;
MOVEL Offs (ref, 45.167, 205.651, 55.000), v, zone, tool30;
MOVEL Offs (ref, 42.019, 204.505, 55.000), v, zone, tool30;
MOVEL Offs (ref, 38.958, 203.142, 55.000), v, zone, tool30;
MOVEL Offs (ref, 36.000, 201.569, 55.000), v, zone, tool30;
MOVEL Offs (ref, 33.159, 199.794, 55.000), v, zone, tool30;
MOVEL Offs (ref, 30.448, 197.825, 55.000), v, zone, tool30;
MOVEL Offs (ref, 27.882, 195.671, 55.000), v, zone, tool30;
MOVEL Offs (ref, 25.472, 193.344, 55.000), v, zone, tool30;
MOVEL Offs (ref, 23.230, 190.854, 55.000), v, zone, tool30;
MOVEL Offs (ref, 21.167, 188.214, 55.000), v, zone, tool30;
MOVEL Offs (ref, 19.294, 185.436, 55.000), v, zone, tool30;
MOVEL Offs (ref, 17.619, 182.535, 55.000), v, zone, tool30;
MOVEL Offs (ref, 16.150, 179.523, 55.000), v, zone, tool30;
MOVEL Offs (ref, 14.895, 176.417, 55.000), v, zone, tool30;
MOVEL Offs (ref, 13.859, 173.231, 55.000), v, zone, tool30;

MOVEL Offs (ref, 13.049, 169.980, 55.000), v, zone, tool30;
MOVEL Offs (ref, 12.467, 166.680, 55.000), v, zone, tool30;
MOVEL Offs (ref, 12.117, 163.348, 55.000), v, zone, tool30;
MOVEL Offs (ref, 12.000, 160.000, 55.000), v, zone, tool30;
MOVEL Offs (ref, 12.117, 156.652, 55.000), v, zone, tool30;
MOVEL Offs (ref, 12.467, 153.320, 55.000), v, zone, tool30;
MOVEL Offs (ref, 13.049, 150.020, 55.000), v, zone, tool30;
MOVEL Offs (ref, 13.859, 146.769, 55.000), v, zone, tool30;
MOVEL Offs (ref, 14.895, 143.583, 55.000), v, zone, tool30;
MOVEL Offs (ref, 16.150, 140.477, 55.000), v, zone, tool30;
MOVEL Offs (ref, 17.619, 137.465, 55.000), v, zone, tool30;
MOVEL Offs (ref, 19.294, 134.564, 55.000), v, zone, tool30;
MOVEL Offs (ref, 21.167, 131.786, 55.000), v, zone, tool30;
MOVEL Offs (ref, 23.230, 129.146, 55.000), v, zone, tool30;
MOVEL Offs (ref, 25.472, 126.656, 55.000), v, zone, tool30;
MOVEL Offs (ref, 27.882, 124.329, 55.000), v, zone, tool30;
MOVEL Offs (ref, 30.448, 122.175, 55.000), v, zone, tool30;
MOVEL Offs (ref, 33.159, 120.206, 55.000), v, zone, tool30;
MOVEL Offs (ref, 36.000, 118.431, 55.000), v, zone, tool30;
MOVEL Offs (ref, 38.958, 116.858, 55.000), v, zone, tool30;
MOVEL Offs (ref, 42.019, 115.495, 55.000), v, zone, tool30;
MOVEL Offs (ref, 45.167, 114.349, 55.000), v, zone, tool30;
MOVEL Offs (ref, 48.388, 113.426, 55.000), v, zone, tool30;
MOVEL Offs (ref, 51.665, 112.729, 55.000), v, zone, tool30;
MOVEL Offs (ref, 54.983, 112.263, 55.000), v, zone, tool30;
MOVEL Offs (ref, 58.325, 112.029, 55.000), v, zone, tool30;
MOVEL Offs (ref, 61.675, 112.029, 55.000), v, zone, tool30;
MOVEL Offs (ref, 65.017, 112.263, 55.000), v, zone, tool30;
MOVEL Offs (ref, 68.335, 112.729, 55.000), v, zone, tool30;
MOVEL Offs (ref, 71.612, 113.426, 55.000), v, zone, tool30;
MOVEL Offs (ref, 74.833, 114.349, 55.000), v, zone, tool30;
MOVEL Offs (ref, 77.981, 115.495, 55.000), v, zone, tool30;
MOVEL Offs (ref, 81.042, 116.858, 55.000), v, zone, tool30;
MOVEL Offs (ref, 84.000, 118.431, 55.000), v, zone, tool30;
MOVEL Offs (ref, 86.841, 120.206, 55.000), v, zone, tool30;
MOVEL Offs (ref, 89.552, 122.175, 55.000), v, zone, tool30;
MOVEL Offs (ref, 92.118, 124.329, 55.000), v, zone, tool30;
MOVEL Offs (ref, 94.528, 126.656, 55.000), v, zone, tool30;
MOVEL Offs (ref, 96.770, 129.146, 55.000), v, zone, tool30;
MOVEL Offs (ref, 98.833, 131.786, 55.000), v, zone, tool30;
MOVEL Offs (ref, 100.706, 134.564, 55.000), v, zone, tool30;
MOVEL Offs (ref, 102.381, 137.465, 55.000), v, zone, tool30;
MOVEL Offs (ref, 103.850, 140.477, 55.000), v, zone, tool30;
MOVEL Offs (ref, 105.105, 143.583, 55.000), v, zone, tool30;
MOVEL Offs (ref, 106.141, 146.769, 55.000), v, zone, tool30;
MOVEL Offs (ref, 106.951, 150.020, 55.000), v, zone, tool30;
MOVEL Offs (ref, 107.533, 153.320, 55.000), v, zone, tool30;
MOVEL Offs (ref, 107.883, 156.652, 55.000), v, zone, tool30;
MOVEL Offs (ref, 108.000, 160.000, 55.000), v, zone, tool30;
MOVEJ Offs (ref, 107.898, 160.776, 55.000), v, zone, tool30;

```
veloc.v_tcp:=100.;  
MOVEJ Offs (ref, 107.598, 161.500, 55.000), v, zone, tool30;  
MOVEJ Offs (ref, 107.121, 162.121, 55.000), v, zone, tool30;  
MOVEJ Offs (ref, 106.500, 162.598, 55.000), v, zone, tool30;  
MOVEJ Offs (ref, 105.776, 162.898, 55.000), v, zone, tool30;  
MOVEJ Offs (ref, 105.000, 163.000, 55.000), v, zone, tool30;  
MOVEJ Offs (ref, 104.900, 163.000, 55.000), v, zone, tool30;  
MOVEJ Offs (ref, 104.900, 163.000, 58.000), v, zone, tool30;  
MOVEJ Offs (ref, 104.900, 163.000, 100.000), v, zone, tool30;  
MOVEJ Offs (ref, .000, .000, 115.500), v, zone, tool30;  
MOVEJ Offs (ref, .000, .000, .000), v, zone, tool30;  
TPWrite "Fim de Programa";  
PulseDO Ferramenta_Desl;  
Stop;  
ENDPROC  
ENDMODULE
```