

**UNIVERSIDADE DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE
PRODUÇÃO**

**COMPARAÇÃO DE DESEMPENHO ENTRE
PROGRAMAS CNC PARAMÉTRICOS E GERADOS
POR SISTEMA CAM**

DISSERTAÇÃO DE MESTRADO

LUIS FERNANDO MELEGARI

**Santa Maria, RS, Brasil
2011**

COMPARAÇÃO DE DESEMPENHO ENTRE PROGRAMAS CNC PARAMÉTRICOS E GERADOS POR SISTEMA CAM

por

Luis Fernando Melegari

Dissertação apresentada ao Curso de Mestrado do Programa de Pós Graduação em Engenharia de Produção da Universidade Federal de Santa Maria, como requisito parcial para a obtenção do título de **Mestre em Engenharia de Produção.**

Orientador: Prof. Dr. Alexandre Dias da Silva

Santa Maria, RS, Brasil

2011

**UNIVERSIDADE DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE
PRODUÇÃO**

A Comissão Examinadora, abaixo assinada,
aprova a Dissertação de Mestrado

**COMPARAÇÃO DE DESEMPENHO ENTRE PROGRAMAS CNC
PARAMÉTRICOS E GERADOS POR SISTEMA CAM**

elaborada por
Luis Fernando Melegari

como requisito parcial para obtenção do grau de
Mestre em Engenharia de Produção

Comissão Examinadora

Alexandre Dias da Silva, Dr.
(Presidente/Orientador)

Inácio da Fontoura Linberger, Dr. (UFSM)

Moacir Eckhardt, Dr. (CTISM)

Santa Maria, 15 de dezembro de 2011.

M519c Melegari, Luis Fernando

Comparação de desempenho entre Programas CNC Paramétricos e gerados por Sistema CAM / por Luis Fernando Melegari. – 2011.

109 p. : il. ; 31 cm

Orientador: Alexandre Dias da Silva.

Dissertação (mestrado) – Universidade Federal de Santa Maria, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia de Produção, RS, 2011

1. Engenharia de Produção 2. Usinagem 3. Programação 4. CNC, Programação Paramétrica 5. Sistemas CAM 6. Comando numérico I. Silva, Alexandre Dias da II. Título.

CDU 658.52

658.5:621.9

621.7

Ficha catalográfica elaborada por Simone G. Maisonave . CRB 10/1733
Biblioteca Central da UFSM

© 2011

Todos os direitos autorais reservados a Luis Fernando Melegari. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.

Endereço: Rua Doze, n. 2010, Bairro da Luz, Santa Maria, RS. CEP: 97110-680

Fone (0xx)55 32225678; Fax (0xx) 32251144; E-mail: ufesme@ct.ufsm.br

AGRADECIMENTOS

A minha esposa, meus pais, minha família, meus amigos.

A todos que contribuíram indiretamente nesta empreitada.

Ao Instituto Federal Sul-Riograndense – Campus Passo Fundo
pela possibilidade de realização dos experimentos.

Agradecimento ao Dr. Alexandre Dias da Silva pela orientação e
amizade desenvolvida nesse período, pela dedicação e suporte para o
sucesso do desenvolvimento deste trabalho.

RESUMO

Dissertação de Mestrado
Programa de Pós-Graduação em Engenharia de Produção
Universidade Federal de Santa Maria

COMPARAÇÃO DE DESEMPENHO ENTRE PROGRAMAS CNC PARAMÉTRICOS E GERADOS POR SISTEMA CAM.

AUTOR: LUIS FERNANDO MELEGARI

ORIENTADOR: ALEXANDRE DIAS DA SILVA

Data e Local da Defesa: Santa Maria, 15 de dezembro de 2011.

A utilização de sistemas CAM para a geração de trajetórias de ferramenta em contornos de geometria simples depende diretamente do conhecimento e da capacidade do usuário na alimentação das informações e na utilização dos recursos disponíveis desses sistemas. A programação paramétrica exige uma base sólida no conhecimento da programação CN e os resultados que se referem ao desempenho da máquina CNC estão diretamente ligados à forma de programação para a obtenção desses resultados. O objetivo deste estudo foi desenvolver uma análise de desempenho entre a programação paramétrica e um sistema CAM em uma máquina CNC específica, com base nos tempos de execução e velocidade de avanço da ferramenta para cada método de programação. A contagem de tempo foi realizada através do controlador CNC, que indica o tempo de execução de cada programa a partir do seu início até o recebimento ou leitura do comando de parada. A comparação entre esses métodos de programação gerou resultados que dependem diretamente da forma de programação, mas que recebem interferência dos recursos tecnológicos do controlador para o acionamento dos eixos da máquina CNC. Com essas experimentações, foi possível demonstrar, através de programas paramétricos, um aumento no desempenho em até 65% quando comparado a outros métodos de programação.

Palavras chave: CNC; programação paramétrica; usinagem; sistemas CAM

ABSTRACT

Dissertação de Mestrado
Programa de Pós-Graduação em Engenharia de Produção
Universidade Federal de Santa Maria

PERFORMANCE COPARISON BETWEEN PARAMETRIC PROGRAMS AND A CAM SYSTEM

AUTHOR: LUIS FERNANDO MELEGARI

ADVISOR: ALEXANDRE DIAS DA SILVA

Date and Location of Defense: Santa Maria, December 15, 2011.

The use of CAM system to create tool's trajectories in outline of simple geometry depends directly of knowledge and ability from user in how to maintain the information and make use of available riches from these systems. The parametric program requires a solid knowledge in CN programming and the results referring to CNC machine's performance are directly connect to the way of programming to obtain these results. The aim of this study was to develop a practice's analysis between parametric program and a CAM system in a specific CNC machine, based on acting time and speed of tool's progress to each programming method. The counting of time was accomplished through CNC controller, which shows the execution time of each program from its beginning until receiving or reading the stop command. The comparison between these programming methods created results that depend of the programming way, but who receive interference of controller's technologies recourses to put in action the CNC axis machine. With these testing, it was possible to demonstrate, by means of parametric programs, an increase of 65% in performance, when compared to other programming methods.

Keyword: CNC parametric programming; machining; CAM systems

LISTA DE TABELAS

Tabela 1 - Resultados Interpolação Helicoidal	77
Tabela 2 - Resultados Interpolação Circular	86
Tabela 3 - Resultados Elipse.....	94
Tabela 4 - Resultados Cavidade Circular	103
Tabela 5 - Comparação do tamanho dos programas em kilobytes	105

LISTA DE ILUSTRAÇÕES

Figura 1 - Modelamento sólido.....	24
Figura 2 - Determinação dos segmentos de reta.	25
Figura 3 - Agrupamento de peças.....	28
Figura 4 - Análise e preparação das geometrias em CAD/ CAM..	29
Figura 5 - Comparação do número de linhas geradas pelos dois processos.....	30
Figura 6 - Fixação da peça e variáveis analisadas.....	31
Figura 7 - Peças (a).....	32
Figura 8 - Peças (a).....	32
Figura 9 - Programação paramétrica para correção, peça (a)	33
Figura 10 - Programação paramétrica para correção, peça (b).....	33
Figura 11 - Resultados de precisão peça (a) e (b)..	34
Figura 12 - Principais trajetórias de usinagem.	35
Figura 13 - Comparativo de tempos - CAM.....	36
Figura 14 - Interpolação circular PO ₁ e linear PO ₂	37
Figura 15 - Velocidade de avanço real X programada com interpolação linear (3500 mm/min).	38
Figura 16 - Resultado interpolação circular (avanço de 3500 mm/min.)	39
Figura 17 - Formação da <i>Spline</i> através de pontos.....	40
Figura 18 - Divisão da interpolação em segmentos de reta.	41
Figura 19 - Precisão da velocidade de execução da interpolação	43
Figura 20 - Estimativa de tempo para cálculos de subprogramas.....	44
Figura 21 - Estimativa de tempo para cálculos.....	45
Figura 22 - Centro de Usinagem Vertical Romi Discovery 560.	46
Figura 23 - Plataforma de programação CAM.....	47

Figura 24 - Curva de aceleração e velocidade	48
Figura 25- Suavização de aceleração/desaceleração.....	48
Figura 26 - Exemplo de programação paramétrica.	53
Figura 27 – Programação da interpolação helicoidal CAD/CAM.....	54
Figura 28 - Fluxograma para interpolação helicoidal paramétrica.....	56
Figura 29 - Geometria para trajetória circular.....	57
Figura 30 - Trajetória programada na geometria circular.	58
Figura 31 - Cálculo do ponto X; Y através de trigonometria.....	58
Figura 32 - Fluxograma da programação paramétrica - interpolação circular.	61
Figura 33 - Geometria elíptica para geração da trajetória.	62
Figura 34 - Trajetória em arcos para a geometria elíptica.....	62
Figura 35 - Trajetória em segmentos de reta para geometria elíptica.....	63
Figura 36 - Exemplo Programa elipse – Arcos.....	64
Figura 37 – Determinação dos descolamentos em x e y sobre a geometria elíptica.....	65
Figura 38 – Fluxograma do programa paramétrico - Elipse.	66
Figura 39 - Trajetória da ferramenta para a cavidade circular.....	67
Figura 40 - Fluxograma Programa Paramétrico Cavidade Circular.....	70
Figura 41 - Programação CAM Helicoidal- Segmentos de retas.....	72
Figura 42 - Velocidade de Avanço – Programação Segmentos de retas.	73
Figura 43 - Tempo de execução – Programação Segmentos de retas.	73
Figura 44 - Velocidade de Avanço – Programação Segmentos de retas + G64.	74
Figura 45 - Tempo de execução – Programação Segmentos de retas + G64	74
Figura 46 - Forma de execução do programa pelo controlador NC.	75
Figura 47 - Programa paramétrico helicóide.	76
Figura 48 - Velocidade de Avanço – Programação Paramétrica.....	77
Figura 49 – Gráfico do desempenho das formas de programação.	78
Figura 50 - Influência da Tolerância na trajetória da ferramenta.....	79
Figura 51 - Programação CAM Circular - Segmentos de retas.	79
Figura 52 - Determinação do tamanho do segmento de reta utilizando programa CAD.....	80
Figura 53 - Tempo de execução – Programação Segmentos de retas.	81
Figura 54 - Programação CAM Circular - Segmentos de arcos.	82
Figura 55 - Programação CAM Circular Arcos – Tempo de execução.....	82
Figura 56 - Forma de execução do programa pelo controlador NC.	83

Figura 57 - Programa paramétrico circular.....	84
Figura 58 - Velocidade de execução - Programação paramétrica.....	85
Figura 59 - Tempo de execução - Programação paramétrica.	85
Figura 60 – Gráfico do desempenho para interpolação circular.....	87
Figura 61 - Programação CAM elipse – Segmentos de retas.	88
Figura 62 - Tempo de execução – Programa elipse segmentos de reta.	89
Figura 63 - Tempo de execução – Programa elipse segmentos de reta + G64.	90
Figura 64 - Programa CAM Elipse – Arcos.....	90
Figura 65 - Velocidade de avanço Elipse programação CAM – Arcos.....	91
Figura 66 - Tempo de execução Elipse programação CAM – Arcos.....	91
Figura 67 - Forma de execução do programa pelo controlador NC.	92
Figura 68 - Memória ocupada pelo programa paramétrico Elipse.....	93
Figura 69 - Tempo de execução programa paramétrico Elipse.....	94
Figura 70 - Comparação de desempenho para elipse.	95
Figura 71 - Cavidade Circular Tridimensional CAM.	96
Figura 72 - Programação CAM Cavidade Circular.	97
Figura 73 - Velocidade de avanço em tela – Cavidade Circular.....	98
Figura 74 - Tempo de execução cavidade circular – segmentos de reta.	99
Figura 75 - Programa CAM Cavidade Circular – Arcos.....	100
Figura 76 - Tempo de execução cavidade circular arcos.	101
Figura 77- Forma de execução do programa pelo controlador NC.	101
Figura 78 – Fragmentos do Programa Paramétrico Cavidade Circular.....	102
Figura 79 - Avanço Cavidade Circular Paramétrico.	103
Figura 80 - Comparação de desempenho Cavidade Circular.	104

LISTA DE ABREVIATURAS E SIGLAS

ANSI	American National Standards Institute
APT	Automatically Programed Tool (Ferramenta Programada Automaticamente)
CAD	<i>Computer Aided Design</i> (Projeto assistido por Computador)
CAE	<i>Computer Aided Engineering</i> (Engenharia assistida por Computador)
CAM	<i>Computer Aided Manufacturing</i> (Fabricação assistida por Computador)
CNC	Comando Numérico Computadorizado
CN	Comando Numérico
ISO	<i>International Organization for Standardization</i> (Organização Internacional para Normalização)
NC	<i>Numeric Control</i> (Comando Numérico)
PC	Computador Pessoal
UFSM	Universidade Federal de Santa Maria

SUMÁRIO

1 INTRODUÇÃO	15
1.1 Objetivo Geral	16
1.2 Objetivos Específicos	16
1.3 Delimitação do Tema	16
2 REVISÃO BIBLIOGRÁFICA	18
2.1 CNC – Comando Numérico Computadorizado	19
2.2 Métodos de Programação CNC	22
2.3 Programação Paramétrica	25
2.4.1 Tecnologia de Grupo.....	28
2.4.2 A programação paramétrica e os ciclos de usinagem.....	29
2.4 O Desempenho da Programação CNC	34
3 METODOLOGIA	46
3.1 Linguagem adotada na programação paramétrica	49
3.1.1 Operações/funções aritméticas.....	50
3.1.2 Operadores Lógicos e de Comparação.....	51
3.2 Interpolação Helicoidal	54
3.2.1 Interpolação Helicoidal CAM	55
3.2.2 Interpolação Helicoidal Paramétrico.....	55
3.3 Interpolação Circular	57
3.3.1 Interpolação Circular CAM – Segmentos de reta	59
3.3.2 Interpolação Circular CAM – Arcos	59

3.3.3 Interpolação Circular Paramétrica.....	60
3.4 Elipse.....	62
3.4.1 Elipse CAM – Segmentos de Reta.....	63
3.4.2 Elipse CAM – Arcos	63
3.4.3 Elipse Paramétrica	64
3.5 Cavidade Circular.....	67
3.5.1 Cavidade Circular CAM – Segmentos de reta.....	68
3.5.2 Cavidade Circular CAM – Arcos.....	68
3.5.3 Cavidade Circular Paramétrica	68
4 RESULTADOS E DISCUSSÕES	71
4.1 Interpolação Helicoidal.....	71
4.2 Interpolação Circular	78
4.3 Elipse	87
4.4 Cavidade Circular	96
5 CONCLUSÃO	106
BIBLIOGRAFIA	107

1 INTRODUÇÃO

As máquinas Comando Numérico Computadorizado (CNC) possuem uma unidade de controle, também referenciado como controlador, que segue uma linguagem de programação geralmente normalizada e ou, em alguns casos, específica, criada pelo fabricante da máquina. Os códigos de programação gerenciam as rotas e funções executadas pelos eixos de movimento da máquina, definindo sua velocidade e posição, formando trajetórias.

Nas execuções automáticas de movimentos da máquina, a fim de obter uma geometria definida, de uma peça ou da usinagem de partes desta, necessita-se gerar dados relacionados aos movimentos a serem executados pela máquina, através da programação CNC. O programa CNC deve conter as informações para a usinagem prevista, sendo a geometria e tecnologia aplicadas, propiciando redução do tempo de operacionalização e execução em comparação a uma máquina convencional, assim como melhor acabamento, boa qualidade e, conseqüentemente, redução do custo operacional, aumentando a repetibilidade e lucratividade.

A programação paramétrica é classificada como uma das formas de programação avançada. Este tipo de programação geralmente é utilizado para uma família de peças. O objetivo de tal programação é efetuar alterações de coordenadas de dimensão em um programa previamente preparado para uma geometria definida. Esse método de programação objetiva um menor tempo de preparação da máquina, simplificando a entrada de dados para uma mesma família de peças e reduzindo, muitas vezes, o tamanho do programa em comparação a outras técnicas de programação.

Em certos casos, dependendo da técnica utilizada em *Computer Aided Manufacturing* (CAM) para qualquer geometria, pode-se obter como resultado programas extensos e incompatíveis com a capacidade de armazenamento do controlador CNC. Esses programas podem também causar variações de aceleração e desaceleração dos eixos entre os pontos de ligação da trajetória da ferramenta. Isso se dá pelo fato de o controlador não possuir velocidade de processamento

suficiente em relação à velocidade de deslocamento dos eixos para os intervalos entre os pontos programados.

1.1 Objetivo Geral

O objetivo deste trabalho é analisar e comparar o desempenho de uma máquina CNC em relação ao tempo, velocidade e tamanho de programas, utilizando a programação paramétrica e a programação em software CAD/CAM.

1.2 Objetivos Específicos

Comparar os resultados de desempenho entre programas gerados em um sistema CAM e programas paramétricos construídos através de variáveis e funções que exigem processamento do controlador da máquina CNC, como cálculos trigonométricos e comparações lógicas.

1.3 Delimitação do Tema

A programação em sistema CAM é realizada através do software Solidcam (<http://www.solidcam.com/>) integrado a plataforma de trabalho do Solidworks (<http://www.solidworks.com/>). Os programas gerados pelo CAM são propositalmente programados de forma a gerar trajetórias através da ligação de pontos em linhas retas (interpolação linear – G1), assim como trajetórias compostas por movimentos circulares (G2/G3). Essas formas de programação são comparadas à programação paramétrica, que é formada por variáveis de dimensões que modificam o tamanho da trajetória da ferramenta. A trajetória também é formada pela utilização de cálculos e comparações lógicas, que fazem parte do corpo do programa.

Os dados de desempenho são fornecidos pelo controlador da máquina CNC. A velocidade real de execução é apresentada pelo controlador durante a execução do programa, bem como o tempo total, que é apresentado na parada de fim do programa. A velocidade real de execução é avaliada, de modo que os tempos de

execução na simulação dos programas que são apresentados pelo sistema CAM são descartados.

2 REVISÃO BIBLIOGRÁFICA

A programação CNC pode ser realizada por diversas técnicas, dependendo do nível de conhecimento do operador e dos recursos disponíveis. Pode se dar, assim, através do teclado numérico da máquina, do gerador de textos ou da programação CAD/CAM (Projeto Assistido por Computador/Fabricação Assistida por Computador) quando se tratarem de programações mais avançadas.

Algumas funções da programação manual auxiliam o operador a obter, através de ciclos de usinagem, geometrias regulares em um tempo menor que a programação CAD/CAM, com pequeno número de entrada de dados e rápida execução e com possibilidade de bom acabamento final. A programação paramétrica utiliza a entrada de dados manual no controlador.

Nikiel (2009) apresenta a programação paramétrica como recurso para as máquinas CNC obterem um maior rendimento. Em alguns casos, é relacionada para uma família de peças ou formatos específicos, otimizando o tempo para execução, assim como o acabamento final.

A otimização paramétrica dessas operações é realizada principalmente a fim de obter um baixo custo e tempo de usinagem. A alta produtividade das máquinas CNC depende de fatores também relacionados com a qualidade dos programas criados. (Grzegorz Nikiel, 2009).

O comando numérico controla automaticamente os movimentos de máquinas pela interpretação de códigos na forma numérica e de letras, sendo que o sistema interpreta esses dados e gera um sinal de saída aos atuadores e componentes da máquina. Tal máquina é denominada máquina CNC.

É importante saber quais as áreas de usinagem que serão beneficiadas com a utilização de uma máquina CNC, em que ainda se utiliza uma máquina convencional. Para tanto, as condições devem ser avaliadas. Caso as condições de usinagens e as ferramentas forem as mesmas, o tempo de corte será muito próximo se a potência permitir, havendo pequenos ganhos de tempo entre os intervalos das operações (SMID, 2000). São apresentadas por Smid (2000) algumas áreas em que o usuário pode e deve esperar melhorias com o CNC, se comparado aos processos convencionais:

1. Redução do tempo de *setup*
2. Redução do tempo de fabricação
3. Exatidão e repetibilidade
4. Contorno de formas complexas
5. Ferramentas simplificadas e seguras
6. Aumento da produtividade geral.

2.1 CNC – Comando Numérico Computadorizado

De acordo com Smid (2000), o controle numérico pode ser definido como a operação de uma máquina ferramenta por meio de instruções codificadas especificamente para seu sistema controlador próprio. Cassaniga (2000) define o CNC como um equipamento eletrônico receptor de informações da forma em que a máquina irá realizar determinados movimentos, por meio de uma linguagem própria que denomina o programa CNC, processando essas informações e devolvendo-as aos componentes do equipamento através de impulsos elétricos. Estes sinais elétricos são responsáveis pelos acionamentos de motores, válvulas e outros componentes, realizando operações sequenciais sem intervenção do operador.

Os fatores preponderantes que induziram à pesquisa, ao aparecimento e à introdução do uso de máquinas operatrizes comandadas numericamente foram o avanço tecnológico durante e após a segunda guerra mundial. Surgiu, assim, a necessidade de adaptação dos equipamentos à novos conceitos de fabricação, tais como baixo custo, pequenos lotes, baixo *lead-time*, produtos de geometria complexa e alta precisão e menor tempo entre projeto do produto e início da fabricação do mesmo.

Como precursoras das modernas máquinas de comando numérico, têm-se as máquinas de cartões perfurados desenvolvidas por Joseph Jacquard em 1801, que estabeleciam os padrões dos tecidos produzidos. As pianolas eram controladas por cilindros com pinos salientes ou rolos de papel perfurados através da utilização de sistemas pneumáticos. Muitas empresas e instituições trabalharam concomitantemente a fim de desenvolver máquinas de comando numérico durante a década de 40.

Provavelmente, a primeira aplicação de comando numérico deve-se à John C. Parsons, da Parsons Corporation de Traverse City, em Michigan, produtor de rotores de helicópteros. Na época, não era possível produzir gabaritos para os rotores na velocidade necessária, o que levou Parsons a conectar um “computador” da época com uma máquina operatriz. Inicialmente, Parsons utilizou cartões perfurados para codificar as informações para o sistema Digitron, como foi chamado.

Em 1949, a Força Aérea Americana (U.S. Air Force) contrata Parsons para realizar um estudo da aplicação dos sistemas de comando numérico com a finalidade de acelerar a produção de componentes de seus aviões e mísseis, cada vez mais complexos. Parsons, por sua vez, subcontratou o laboratório de Servomecanismos do Massachusetts Institute of Technology (MIT). Uma fresadora de três eixos - Hydrotel, da Cincinnati Milling Machine Company, foi escolhida como veículo para a experiência. Os controles de copiagem foram removidos e a máquina foi aparelhada com equipamento de comando numérico. O resultado foi um protótipo das máquinas de comando numérico atuais. Os pesquisadores do MIT criaram o termo *numerical control* ou comando numérico (MUNDO CNC, 2010).

Grande parte do desenvolvimento das máquinas de controle numérico deve-se à U.S. Air Force, produzindo estruturas de avançados aviões militares à jato, com construção num curto espaço de tempo entre o projeto e a fabricação, em pequenos lotes de peças, constituindo uma excelente oportunidade para o teste do comando numérico.

Wilson (1980) destaca que, no final da década de 50, os fabricantes de aviões incrementaram de maneira incisiva o uso do equipamento de comando numérico, com geração contínua de contornos. Tais indústrias asseguraram grande progresso naquele período pelo cumprimento do programa de produção, obtenção do grau de precisão exigido no produto a custos compatíveis, os quais não poderiam ter sido obtidos sem o comando numérico.

Ainda na década de 50, o método prático para aumentar o rendimento das máquinas existentes, em um curto período de tempo, era a adaptação de comando numérico através da instalação de controlador, motores e eixos. Entretanto, no fim dessa década, a iniciativa deixou de ser a mais viável, pois, a partir de 1955, as vendas desse tipo de máquina começaram a crescer, fazendo com que o preço caísse devido à contínua miniaturização dos componentes eletrônicos necessários. Componentes que incluem desde válvulas até circuitos integrados de grande escala

(VLSICs) diminuíram em tamanho e custo. A produção e a confiabilidade das máquinas aumentaram, uma vez que as máquinas comandadas numericamente continuaram a impressionar realizando operações previamente consideradas impossíveis ou impraticáveis, com melhor precisão e repetibilidade que os métodos convencionais.

Em 1957, juntamente com o início da guerra do Vietnã nos EUA, iniciou-se uma revolução no sistema de manufatura, intensificando o uso de máquinas de comando numérico. Surgiu um grande número de fabricantes de máquinas e de controles no mercado, sendo que alguns passaram também a fabricar seus próprios controles (CNC TECNOLOGIA, 2010). A partir de 1959, equipamentos com controles de posicionamento ponto a ponto e geração contínua de contornos foram sendo substituídos pelo trocador automático de ferramentas, o qual foi desenvolvido por uma fábrica de usinagem de metais para uso próprio.

Aplicações de controle de posicionamento começaram a crescer e logo o número de máquinas instaladas com este controle ultrapassou o daqueles de copiagem contínua de contorno. Em 1961, surgiu a primeira furadeira com posicionamento da mesa controlado por programa, notável por seu baixo preço quando comparada a outras máquinas de comando numérico da época. Em fins de 1962, todos os grandes fabricantes de máquinas-ferramentas estavam empenhados no controle numérico e, hoje, são poucos os que não oferecem esse tipo de produto (STEMMER, 1981).

Com o grande número de concorrentes e as diversificações existentes, houve a necessidade de padronização. Nos estágios iniciais, o comando numérico necessitava de estreita coordenação de esforços técnicos nas áreas de codificação, formatos de dados de entrada, terminologia e sistema organizacional, os quais eram necessários principalmente para facilitar o intercâmbio de lotes de encomendas entre os fabricantes de aviões. Assim a partir de 1958, houve a possibilidade de padronização do formato de dados de entrada conforme padrão RS-244.

Comumente, eram utilizadas fitas perfuradas como dados de entrada com as instruções dos dados referentes à peça e às condições de usinagem, definidas pelo programador. Essas fitas podiam ser executadas tanto pelo sistema manual como através de auxílio do computador.

Smid (2000) relata que, nos anos 70, foram introduzidas as máquinas CNC que passaram a depender menos da parte de *hardware*, essencial nos circuitos

daquelas anteriores aos anos 60, e a ter seu funcionamento baseado mais no *software*. Os avanços substituíram a entrada manual de dados e as fitas perfuradas por armazenamento em disquete dos programas ou comunicação remota, de modo que, atualmente, é possível inserir dados na máquina a partir de uma grande variedade de programas e linguagens.

Nos anos 80, a necessidade de elevar a produção com precisão chegou às empresas, sobretudo norte-americanas e europeias, que estavam fortemente preocupadas com os reduzidos aumentos de produtividade obtidos por seus trabalhadores. Tais fatos levaram a um aumento considerável na automação, principalmente nos EUA, numa tentativa de reconquistar uma posição competitiva no mercado global. Essas necessidades condizuram a uma maior confiança no *software* para programar equipamentos automáticos e máquinas CNC.

Smid (2000) afirma que o desenvolvimento da tecnologia CNC não significa o fim das máquinas convencionais. Existem muitos casos em que é preferível utilizar máquina convencional a máquina CNC, já que certos tipos de trabalho podem ser executados de forma mais rápida em máquinas convencionais ou semi-automáticas do que em máquinas numericamente controladas. Assim, máquinas CNC não significam a eliminação das máquinas convencionais, mas apenas o seu complemento.

Cassaniga (2000) afirma que o CNC não é um simples sistema que atua diretamente no equipamento; devendo, antes, ser encarado como um processo de mudança de cultura dentro das empresas que tende a evoluir. Para o melhor aproveitamento desse tipo de equipamento, é necessária a boa organização no que se refere ao processo de fabricação, aos sistemas de medição e controle e à administração do tempo e métodos de trabalho.

2.2 Métodos de Programação CNC

As formas de programação CNC incluem uma variedade de procedimentos, que vão deste o método manual até o método automatizado.

A programação manual é feita através de teclados alfanuméricos conectados às máquinas de comando numérico, principalmente nos casos em que a

simplicidade do trabalho a ser feito e a natureza da operação não justificam gastos com sofisticação de métodos de programação. Por outro lado, o uso de programação com auxílio do computador proporciona, além da rapidez, uma maior segurança contra erros.

Era visível a falta de padronização nas empresas com mais de uma máquina, que possuíam até então uma linguagem própria, havendo necessidade de equipes técnicas especializadas para cada uma delas, para cada tipo de comando, refletindo diretamente nos custos de fabricação (CASSANIGA, 2000). Para que houvesse a possibilidade de uso de computadores no auxílio à programação das máquinas, foram desenvolvidos vários estudos visando um sistema de linguagem adequado, de maneira a facilitar a programação. A primeira linguagem desenvolvida para tal finalidade foi o *Automatically Programed Tool* (APT), criada pelo MIT em 1956. Para geração contínua de contornos, foram desenvolvidas novas linguagens como *Auto Prompt* (Automatic Programming of Machine Tools) - programa tridimensional. Os códigos das máquinas mais utilizadas no mundo foram padronizados, primeiramente, pela norma ISO 1056:1975 e, posteriormente, atualizados pela ISO 6983-1:198 (MUNDO CNC).

Os sistemas CAM (*Computer Aided Manufacturing*) auxiliam o usuário na geração de programas CNC em geometrias complexas, geralmente criadas a partir de modelagens 3D em sistemas CAD (*Computer Aided Design*). Segundo Foggiatto *et al* (2008), os sistemas CAD 2D foram desenvolvidos para substituir o desenho manual. De acordo com Benedito (2009), os recursos foram aprimorados passando de desenhos de duas dimensões (2D) para modelamentos tridimensionais em 3D, conforme representado na Figura 1.

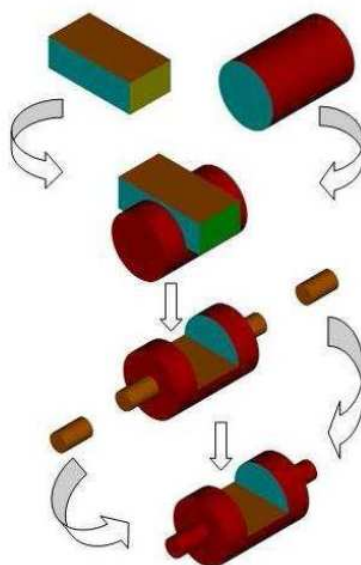


Figura 1 - Modelamento sólido - Benedito (2009).

As plataformas CAD/CAM podem trabalhar de formas distintas ou integradas em um único ambiente de trabalho computacional.

A utilização de sistemas CAD/CAM promove recursos como ciclos e formas de usinagem que permitem diferentes caminhos da ferramenta de forma pronta, uma vez que tais sistemas possuem uma plataforma de trabalho que possibilitam ao usuário criar o programa CNC a partir de um desenho definido de geometria mais complexa, o que permite gerar o programa automaticamente, sem necessidade real da digitação deste por parte do operador/programador. Para tanto, basta que o programador entre com as variáveis e as limitações necessárias para a forma usinagem da peça, integrando etapas de projeto e de fabricação. Os programas CAD/CAM permitem a simulação da execução final, promovendo virtualmente ao operador a geometria final da peça.

A interpolação linear determina a trajetória da ferramenta fazendo a ligação entre pontos e, formando, assim, segmentos de retas. Esses segmentos de reta são adaptáveis pelo programa CAM a uma faixa de tolerância previamente definida. Souza (2001) relata que a ligação entre dois pontos de forma retilínea são representados pelo comando G1 da linguagem de programação ISO 6983, deste modo a representação matemática é simplificada em relação a outros métodos.

Com base na Figura 2, apresentada por Souza & Coelho (2007), definida por geometrias circulares de raios distintos, contornadas por bandas de tolerâncias, são traçados segmentos de reta respeitando os limites destas. O tamanho do raio

dessas circunferências tem influência direta sobre o tamanho dos segmentos de retas gerados.

O método teórico em que os pontos unem a trajetória foi utilizado por Arias (2009) para verificar a dispersão dos pontos ao longo da trajetória da ferramenta. O método foi utilizado em diversos sistemas CAM para geração de programas CNC.

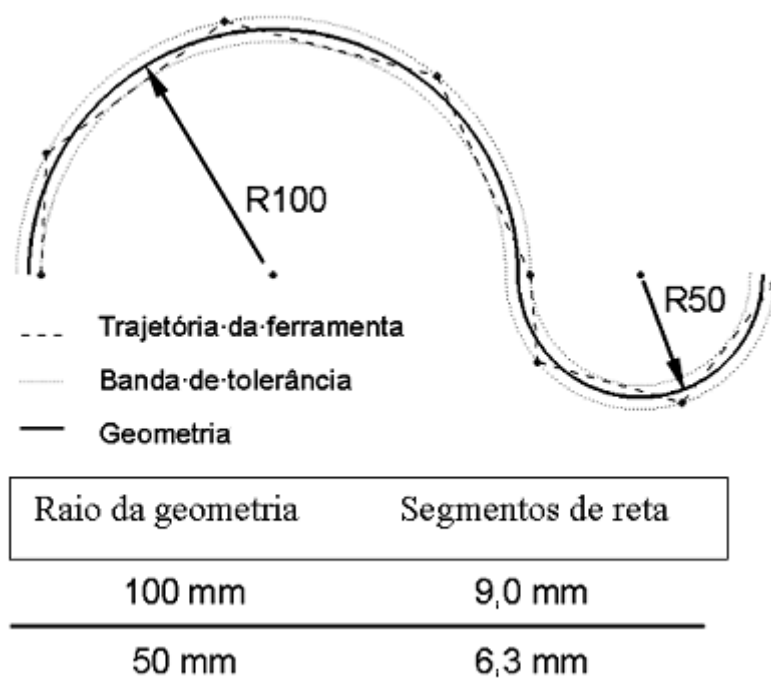


Figura 2 - Determinação dos segmentos de reta - Souza & Coelho (2007).

Na representação de superfícies complexas, a interpolação linear tem como característica a grande segmentação da trajetória que, associada às altas taxas de velocidade de avanço, resultará em significativos problemas de desempenho dinâmico na manufatura em máquinas de três e cinco eixos. Dentre as causas dos problemas, destacam-se o tempo de processamento do CNC e as características de aceleração da máquina-ferramenta (ARIAS, 2009).

2.3 Programação Paramétrica

As técnicas utilizadas para a programação CNC variam de acordo com a disponibilidade financeira e tecnológica. A entrada de dados, ou programa para a máquina CNC, pode ser fornecida diretamente de forma manual, pelo teclado

numérico do controlador (máquina) ou utilizando um editor de textos em um computador.

Em certos casos, algumas máquinas CNC não possuem capacidade de armazenamento suficiente para programas de geometrias simples com programação extensa gerada pelo CAD/CAM. Devido a uma limitação de hardware empregado no controlador, pode existir a dificuldade de transmissão de dados em tempo real de acordo com a execução do programa, o que causará paradas repentinas durante a movimentação da máquina a espera de recebimento de uma nova linha de comando por parte de um sistema de transmissão de dados auxiliar. Isso ocorre quando a transmissão de um programa comportando milhares de linhas em pequenos espaços percorridos torna a ocupar um tempo maior que a própria execução por parte da máquina, ou seja, a máquina executa esse movimento em pequeno espaço percorrido em um tempo menor que o necessário para o recebimento de nova linha de comando.

A programação paramétrica permite a utilização de cálculos que geram as coordenadas de movimentos ao sistema através de comandos para as funções específicas por meio do controlador CNC. Como exemplo, tem-se a substituição de um programa CNC comportando um grande número de linhas por um programa paramétrico sustentado por poucas linhas a fim de executar a mesma operação, favorecendo, assim, o acabamento e o tempo de usinagem.

As operações de usinagem em máquinas CNC de todos os tipos são o tema de muitas pesquisas orientadas para a otimização. Os pesquisadores tem abordado principalmente a estrutura da programação, bem como uma otimização paramétrica e seus resultados. A operação de usinagem CNC realizada em máquinas é caracterizada por sua qualidade superficial, precisão dimensional, complexidade do formato da superfície usinada, entre outras.

A programação de máquinas CNC comporta níveis de tecnologia em função da variação de geometrias aplicadas ao produto final, tornando-se um fator diferenciador para as empresas que utilizam esses recursos da tecnologia CNC, uma vez que isso reflete diretamente na taxa de desempenho e lucratividade. Sem o conhecimento da programação CNC e do seu rendimento, não é possível alcançar ou extrair de forma satisfatória os rendimentos máximos de uma máquina-ferramenta.

A maioria dos programas CNC atuais não são paramétricos, mas possuem uma estrutura simples, principalmente no que se refere aos programas gerados automaticamente pelos pós-processadores de CAD/CAM. Muitos dos blocos de tais programas descrevem o movimento apenas de usinagem (uma função de interpolação + as coordenadas) e sua análise não demanda grande eficiência computacional do controlador CNC. É diferente no caso de programas paramétricos (por exemplo, para usinagem de superfícies mais complexas), em que, em muitos blocos, os cálculos adicionais executados exigem um determinado tempo para o cálculo de coordenadas, podendo ser maior em comparação com o tempo de movimento da ferramenta, principalmente para controladores CNC de baixa eficiência.

Certos programas gerados de maneira simples pelas interfaces CAD/CAM podem, devido à configuração do pós-processador, gerar segmentos de retas, ou interpolações lineares, em distâncias entre coordenadas muito pequenas - centésimos a milésimos de milímetros - como no caso de um círculo formado por pequenos segmentos de reta, com resolução de cerca de 0,005mm. Esse fato pode resultar em uma movimentação não fluente da ferramenta, prejudicando a qualidade da superfície usinada, já que exige tempos de execução dos blocos menores não suportados pelo controlador.

A programação paramétrica avançada busca juntar ao controlador CNC novos ciclos de usinagem, partindo de cálculos matemáticos, atribuição de variáveis e desvios condicionais implementando a lógica algorítmica. Esse método de programação pode ser comparado a outros métodos que utilizam uma linguagem computacional definida, como Basic, C e Pascal (MUNDO CNC).

A técnica de parametrização de programas CNC é uma das técnicas de programação que pode contribuir para o aumento da eficiência operacional. A programação parametrizada auxilia a organização da manufatura, sendo que a aplicação desse método na Tecnologia de Grupo, bem como na criação de novos ciclos de usinagem está relacionada diretamente com o tipo e volume de peças a serem produzidas.

2.4.1 Tecnologia de Grupo

A Tecnologia de Grupo pode ser classificada genericamente como uma filosofia que define a solução de problemas explorando semelhanças. Lorini (1993) confirma a obtenção de vantagens operacionais e econômicas ao utilizar o tratamento de grupo, que funciona de modo a agrupar as peças e os recursos para a manufatura.

O agrupamento de peças em famílias segue uma metodologia que reúne semelhança de geometria e de processo. Assim, cada família de peças passa a ser tratada como uma peça composta contendo os atributos do grupo em que se insere. Busca-se, desse modo, obter uma manufatura de pequenos lotes e as vantagens, principalmente econômicas, de uma produção em massa (Lorini, 1993).

O autor admite que, em longo prazo, a Tecnologia de Grupo, além de proporcionar ganhos de produtividade, acaba por estabelecer bases para a implementação da fabricação assistida por computador. Os critérios utilizados para o agrupamento podem conduzir a famílias cujas peças tenham similaridade de forma ou de processo de fabricação. Assim, Lorini (1993) demonstra o agrupamento de peças segundo critérios geométricos e critérios de processos, conforme a Figura 3.

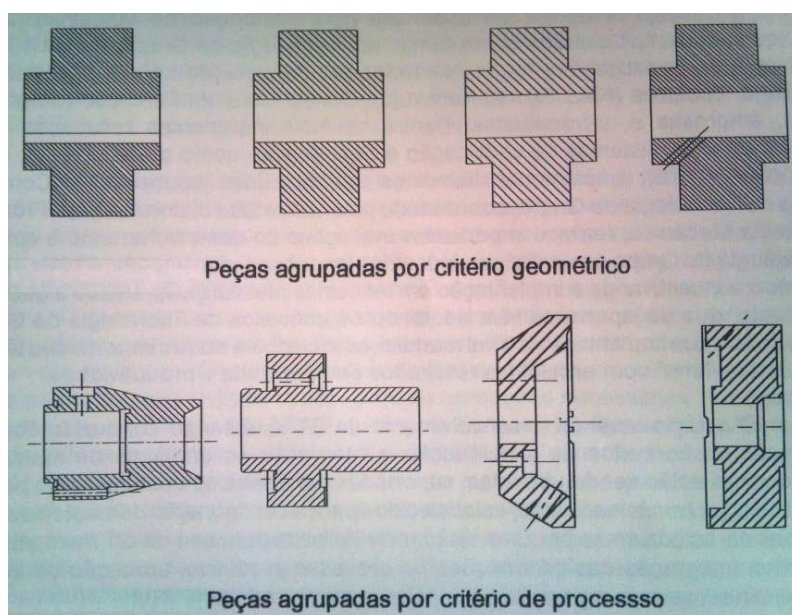


Figura 3 - Agrupamento de peças. Fonte: Lorini (1993).

2.4.2 A programação paramétrica e os ciclos de usinagem

A programação parametrizada permite ao usuário da máquina CNC a criação de novos ciclos de usinagem para funções específicas, tais como rosqueamento e furação. Com macros de usuários, podem-se desenvolver rotinas de propósitos gerais para operações como usinagem em linha, padrões de furos e roscas específicas ou entalhes (GONÇALVES, 2007).

Gonçalves (2007) comparou a programação CAM com a programação paramétrica. As geometrias estudadas foram analisadas em CAD, de modo a possibilitar a preparação de cada peça para um processo CAD/CAM (Figura 4).

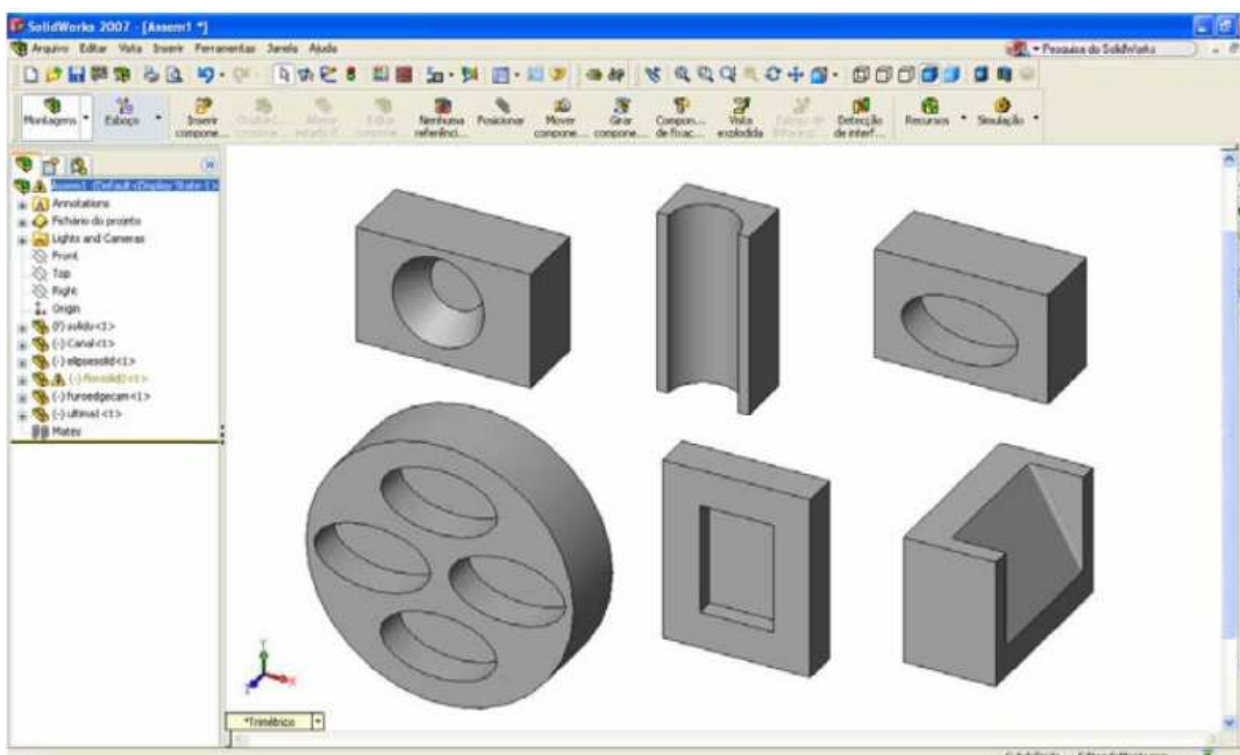


Figura 4 - Análise e preparação das geometrias em CAD/ CAM. Fonte: Gonçalves (2007).

A comparação foi realizada a partir do número de linhas de cada programa para a execução de uma mesma geometria. A Figura 5 apresenta as geometrias avaliadas juntamente com o tamanho do programa para cada método utilizado







GEOMETRIAS	CAD/CAM		PARAMETRIZADO
	Dimensões: X40.Y60.Z20. Para profundidade de corte U=2. Gerado <i>172 linhas</i>	Para profundidade de corte U=0.5 Gerado <i>607 linhas</i>	Dimensões: Qualquer Programado com <i>13 linhas</i>
	Dimensões: X80.Y30.Z53. Para profundidade de corte U=2. Gerado <i>1481 linhas</i>	Para profundidade de corte U=0.5 Gerado <i>6052 linhas</i>	Dimensões: Qualquer Programado com <i>15 linhas</i>
	Dimensões: R50.C25.Z10. Para profundidade de corte U=2. Gerado <i>645 linhas</i>	Para profundidade de corte U=0.5 Gerado <i>2462 linhas</i>	Dimensões: Qualquer Programado com <i>17 linhas</i>
	Dimensões: R200.C30.A0.B360. Para profundidade de corte U=2. Gerado <i>993 linhas</i>	Para profundidade de corte U=0.5 Gerado <i>4130 linhas</i>	Dimensões: Qualquer Programado com <i>23 linhas</i>
	Dimensões: R50.A0.B360.H30. Para profundidade de corte U=2. Gerado <i>433 linhas</i>	Para profundidade de corte U=0.5 Gerado <i>1778 linhas</i>	Dimensões: Qualquer Programado com <i>13 linhas</i>
	Dimensões: D40.Y100.Z30. Para profundidade de corte U=2. Gerado <i>1918 linhas</i>	Para profundidade de corte U=0.5 Gerado <i>7747 linhas</i>	Dimensões: Qualquer Programado com <i>17 linhas</i>

Figura 5 - Comparação do número de linhas geradas pelos dois processos. Fonte: Gonçalves (2007).

Liu (2001) apresenta uma metodologia para a redução de erros provocados pelas deflexões dos eixos em tornos CNC através da programação paramétrica. As variáveis independentes são adicionados ao início do programa em forma paramétrica. Após a reconstrução da trajetória da ferramenta, utilizando a programação paramétrica e aplicando os dados, a precisão final da usinagem é substancialmente melhorada. A Figura 6 apresenta a fixação da peça e as variáveis analisadas:

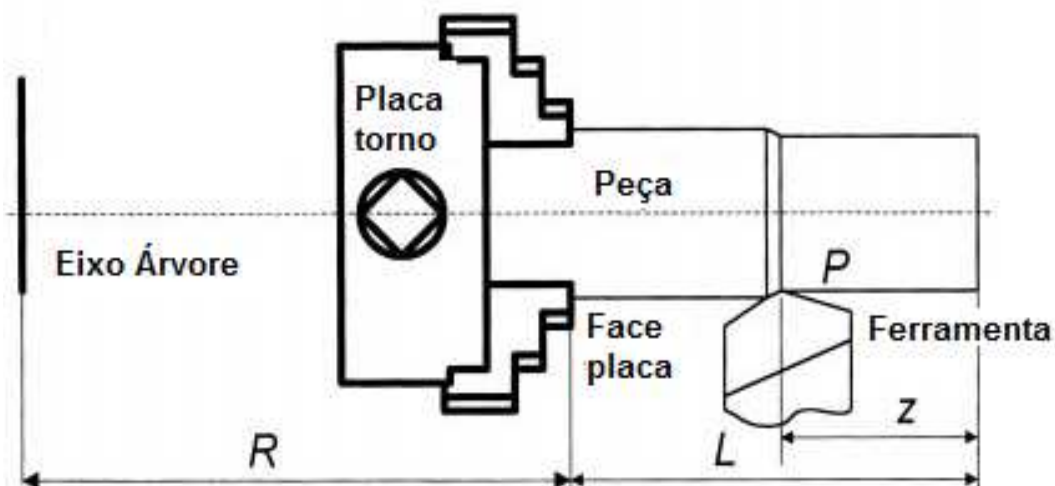


Figura 6 - Fixação da peça e variáveis analisadas. Fonte: Adaptado de Liu (2001).

Nos centros de torneamento que Liu (2001) utilizou em seu trabalho, os movimentos de corte da ferramenta limitam-se aos eixos X e Z. As variáveis #100 e #200 denotam os eixos X e Z, respectivamente. Essas variáveis representam a localização em um ponto definido da ferramenta de corte no programa. As variáveis #101, #102, #202... representam as dimensões da peça, no diâmetro e comprimento. As denotações $P(\#100,\#200)$ formam o caminho a ser percorrido pela ferramenta de corte.

As peças das figuras 7 (peça "a") e 8 (peça "b") respectivamente representam as geometrias e suas variáveis

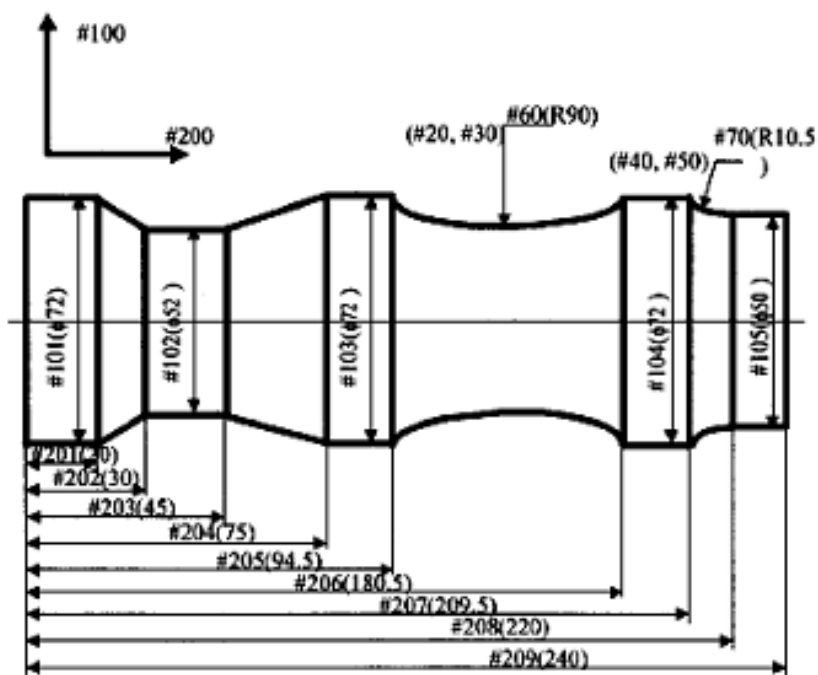


Figura 7 - Peças (a). Fonte: Liu (2001).

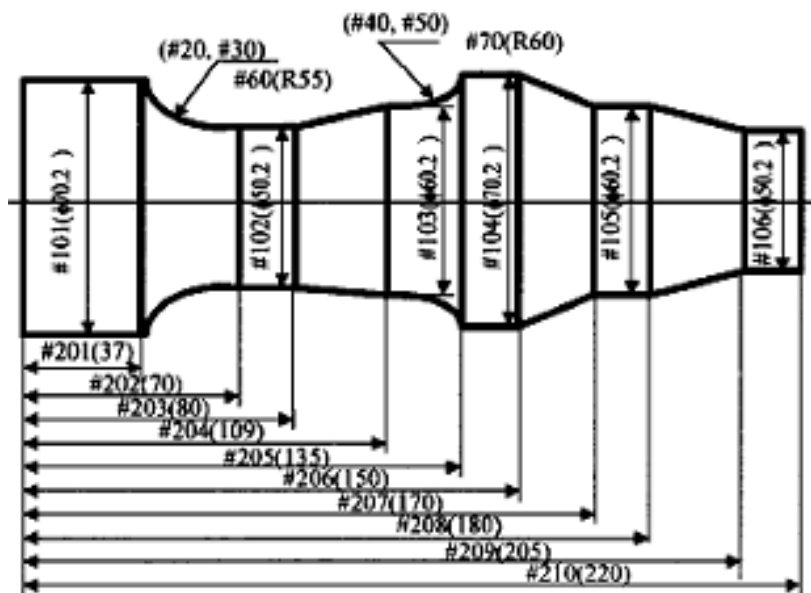


Figura 8 - Peças (a). Fonte: Liu (2001).

O caminho da ferramenta é descrito em programação paramétrica no início do programa, conforme as figuras 9 e 10, para as peças (a) e (b). As peças são divididas em setores que sofrem flexão diferenciada e as equações para cada divisão da peça são descritas na programação através de variáveis.

Para a peça (a):

$$P(\#100, \#200) = \begin{cases} \#101 & (0 \leq \#200 \leq \#201) \\ \#101 - [\#101 - \#102] \frac{\#200 - \#201}{\#202 - \#201} & (\#201 \leq \#200 \leq \#202) \\ \#102 & (\#202 \leq \#200 \leq \#203) \\ \#102 + [\#103 - \#102] \frac{\#200 - \#203}{\#204 - \#203} & (\#203 \leq \#200 \leq \#204) \\ \#103 & (\#204 \leq \#200 \leq \#205) \\ 2 \sqrt{([\#60]^2 - [\#200 - \#30]^2)} + \#20 & (\#205 \leq \#200 \leq \#206) \\ \#104 & (\#206 \leq \#200 \leq \#207) \\ 2 \sqrt{([\#70]^2 - [\#200 - \#50]^2)} + \#40 & (\#207 \leq \#200 \leq \#208) \\ \#105 & (\#208 \leq \#200 \leq \#209) \end{cases}$$

Figura 9 - Programação paramétrica para correção, peça (a) . Fonte: Liu (2001).

Peça (b):

$$P(\#100, \#200) = \begin{cases} \#101 & (0 \leq \#200 \leq \#201) \\ 2 \sqrt{([\#60]^2 - [\#200 - \#30]^2)} + \#20 & (\#201 \leq \#200 \leq \#202) \\ \#102 & (\#202 \leq \#200 \leq \#203) \\ \#102 + [\#103 - \#102] \frac{\#200 - \#203}{\#204 - \#203} & (\#203 \leq \#200 \leq \#204) \\ 2 \sqrt{([\#70]^2 - [\#200 - \#50]^2)} + \#40 & (\#204 \leq \#200 \leq \#205) \\ \#104 & (\#205 \leq \#200 \leq \#206) \\ \#104 - [\#105 - \#104] \frac{\#200 - \#206}{\#207 - \#206} & (\#206 \leq \#200 \leq \#207) \\ \#105 & (\#207 \leq \#200 \leq \#208) \\ \#106 - [\#107 - \#106] \frac{\#200 - \#208}{\#209 - \#208} & (\#208 \leq \#200 \leq \#209) \\ \#106 & (\#209 \leq \#200 \leq \#210) \end{cases}$$

Figura 10 - Programação paramétrica para correção, peça (b). Fonte: Liu (2001).

A proposta de Liu (2001) demonstrou redução da variação dimensional ao longo do corpo da peça utilizando-se a programação paramétrica para compensação da flexão da peça durante a usinagem. A variação em diâmetro, quando elevada era 79 μm , passou para aproximadamente 5 a 7 μm , muito próximo ao erro de repetibilidade da máquina adotada, que era de 4 μm (Figura 11).

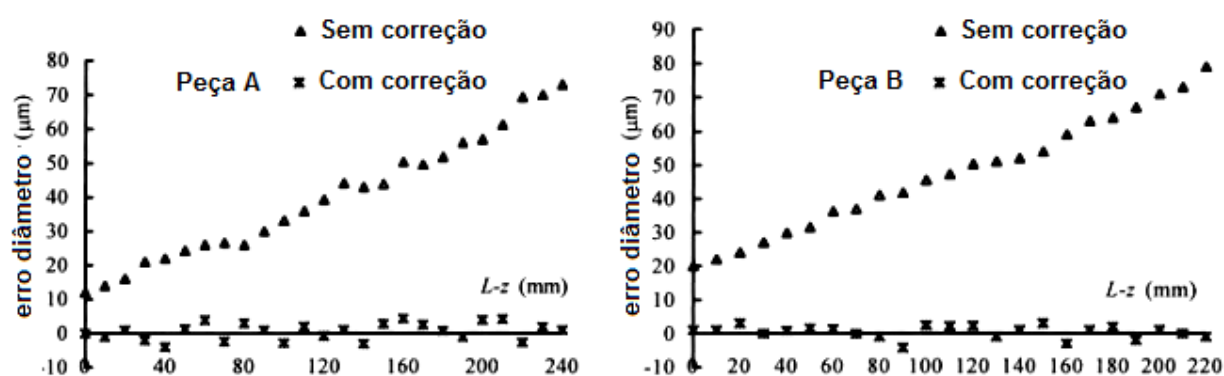


Figura 11 - Resultados de precisão peça (a) e (b). Fonte: Adaptado de Liu (2001).

A utilização da programação paramétrica para a compensação do erro foi utilizada em um modelo simples, sendo que as deflexões de máquina pela força radial de corte foram aplicadas ao programa paramétrico para a correção da trajetória da ferramenta em pontos definidos. Liu (2001) conclui, dessa forma, que a programação paramétrica, já que, ao ser utilizada para correção em uma máquina específica, obtém um reforço substancial na precisão da usinagem, pode ser introduzida em diversas outras máquinas.

2.4 O Desempenho da Programação CNC

Segundo Silva (2006), as limitações de tecnologia CAM são levadas em conta nas etapas iniciais da construção da geometria, ainda na utilização do sistema CAD. A partir da geometria criada em CAD, a mesma é transferida ao CAM onde são aplicados os recursos de desbaste e de acabamento seguindo as limitações e entrada de dados do usuário. As informações tecnológicas são alimentadas no CAM,

que permite a manipulação da biblioteca de ferramentas e estratégias de usinagem previamente definidas.

Souza (2007) apresenta os resultados da programação CAM com as principais trajetórias de usinagem (Figura 12) para a confecção de várias geometrias com o mesmo perfil, cada uma delas avaliada em função da estratégia definida.

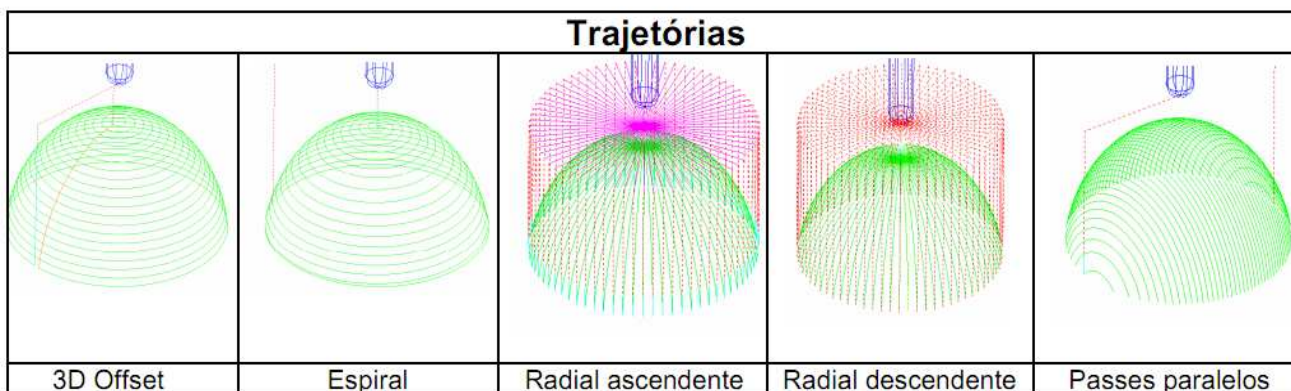


Figura 12 - Principais trajetórias de usinagem. Fonte: Bittencourt (2006).

Utilizando o recurso de simulação gráfica do CAM, realizou-se um comparativo do tempo estimado pelo sistema com o tempo real de usinagem de cada estratégia utilizada no experimento. Conforme exemplificado no gráfico da Figura 13, apenas na estratégia de usinagem em forma espiral o tempo de usinagem real foi muito próximo do tempo estimado pelo CAM. Com isso, pode-se concluir que quatro das cinco estratégias estudadas apresentaram variação de avanço programado, o que, além de aumentar o tempo de usinagem, prejudica o acabamento superficial do produto usinado.

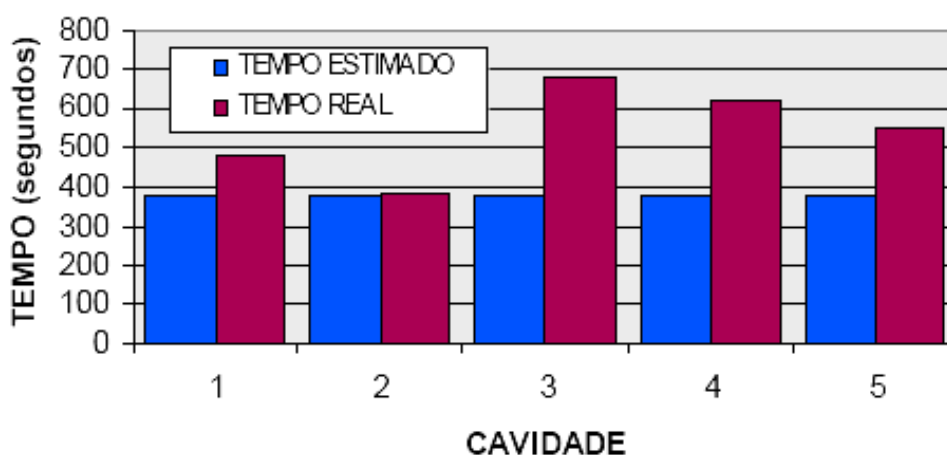


Figura 13 - Comparativo de tempos - CAM. Fonte: Souza (2007).

Os estudos realizados por Salami *et al* (2006) objetivaram otimizar as taxas de avanço em programas de acabamento utilizando ferramentas esféricas com um sistema CAM para o cálculo das trajetórias da ferramenta. Os testes foram realizados em um corpo de prova. Os arquivos de programa gerados pelo CAM foram editados, utilizando algoritmos matemáticos para a geração da trajetória. Os autores concluíram, assim, que é possível reduzir o tempo de acabamento em 10% através da otimização das trajetórias.

O tempo que o controlador CNC leva para o processamento de cada bloco programado é uma característica utilizada por fabricantes como parâmetro de desempenho, sendo definido como o tempo que o CNC leva para ler e processar um bloco de informação do programa CNC. Quando o tempo de processamento do bloco for maior que o tempo de usinagem do percurso do segmento, a máquina atingirá o ponto de destino e a informação para o próximo movimento ainda não estará disponível (ARIAS, 2009). Já quando o tempo de usinagem de um determinado percurso for menor que o tempo do processamento do bloco seguinte pelo controlador CNC, a máquina estará com sua tarefa de posicionamento concluído sem disponibilidade de informação para a trajetória seguinte.

Schützer *et al* (2007), consideram essa metodologia como uma forma inconveniente para a criação de programas em geometrias complexas, geralmente em usinagens de moldes e matrizes, devido aos seguintes fatores: programas CNC muito extensos; baixo acabamento pela geração de superfícies planas no perímetro de superfícies circulares (“facetamento” da superfície) e quantidade elevada de informações pela divisão de linhas circulares em pequenos segmentos de reta,

acarretando aceleração e desaceleração dos eixos entre cada um dos pontos programados, prejudicando a velocidade de avanço.

Souza (2000) apresenta a programação através da utilização dos comandos G01, G02 e G03, que descrevem a trajetória complexa da ferramenta. O método permite a associação da interpolação linear e da interpolação circular na representação da trajetória complexa, gerando programas NC que contêm comandos G01, G02 e G03. Os segmentos de reta PO_1 e PO_2 , conforme demonstrado na Figura 14, associados aos arcos R_1 e R_2 , são utilizados para descrever a mesma trajetória de ferramenta. Esse tipo de programação é utilizada para trajetórias em duas dimensões, sendo que os sistemas CAM são capazes de gerar programas com esta configuração e os controladores estão aptos a interpretação destes.

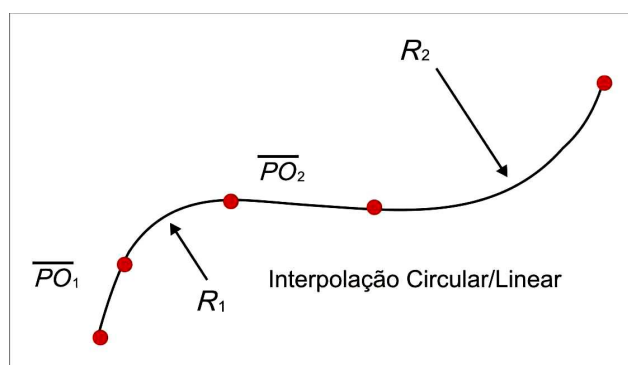


Figura 14 - Interpolação circular PO_1 e linear PO_2 . Fonte: Souza (2000).

Observem-se os estudos realizados por Souza (2004), com base na usinagem de três semicírculos. O primeiro experimento foi feito com avanço programado para 3500 mm/min, com a programação através da divisão do arco em segmentos de reta representado pelo comando G01. O segundo experimento foi usinado com a mesma taxa de avanço utilizando a interpolação circular, composta por comandos G02 e G03. Para fins de comparação, um terceiro programa foi construído com interpolação linear (G01) com avanço de 1000 mm/min. Na Figura 15 verifica-se grande variação de avanço para uma velocidade programada de 3500 mm/min utilizando interpolação linear como programação. O autor relata ainda que esse método gerou vibrações na máquina durante a experimentação.

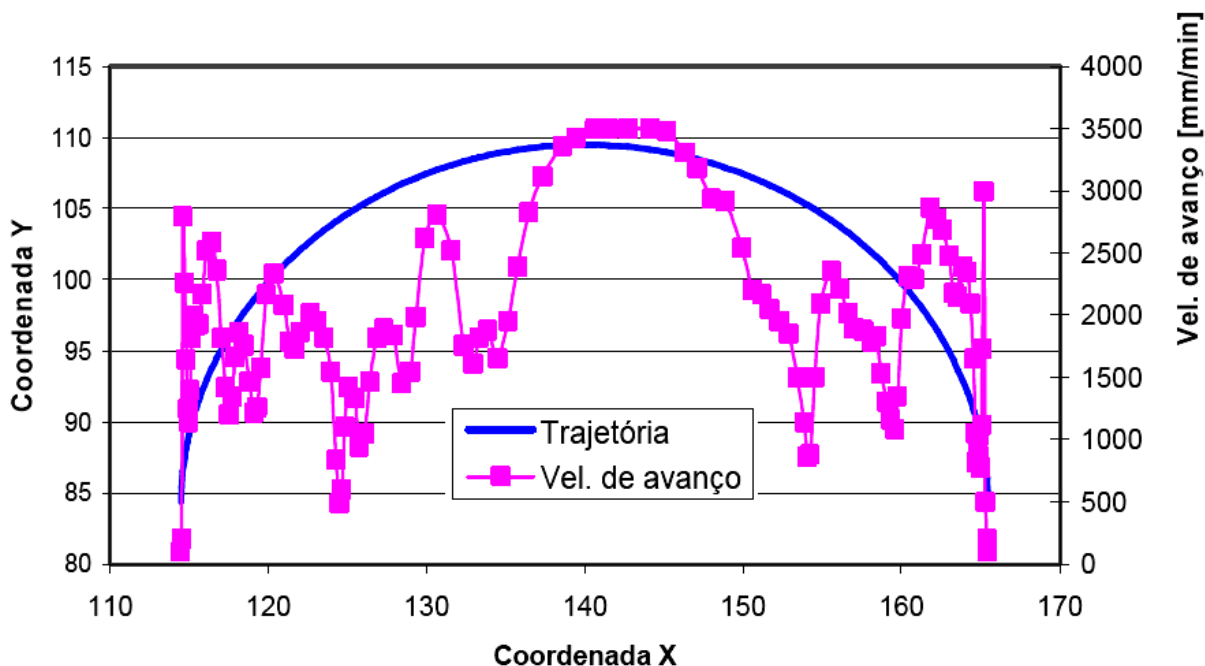


Figura 15 - Velocidade de avanço real X programada com interpolação linear (3500 mm/min). Fonte: Souza (2004).

Os resultados obtidos no primeiro experimento demonstram claramente as variações de velocidades de avanço na tentativa de reprodução de uma trajetória circular pela sua divisão em pequenos segmentos de reta (interpolação linear). As variações da velocidade de avanço geram perdas de produtividade e deficiências de acabamento superficial.

Souza, em sua segunda experimentação, baseada na utilização da programação através de interpolação circular, obteve um resultado diferente ao primeiro, sendo que não ocorreram variações da velocidade de avanço durante a execução da trajetória do semicírculo (Figura 16).

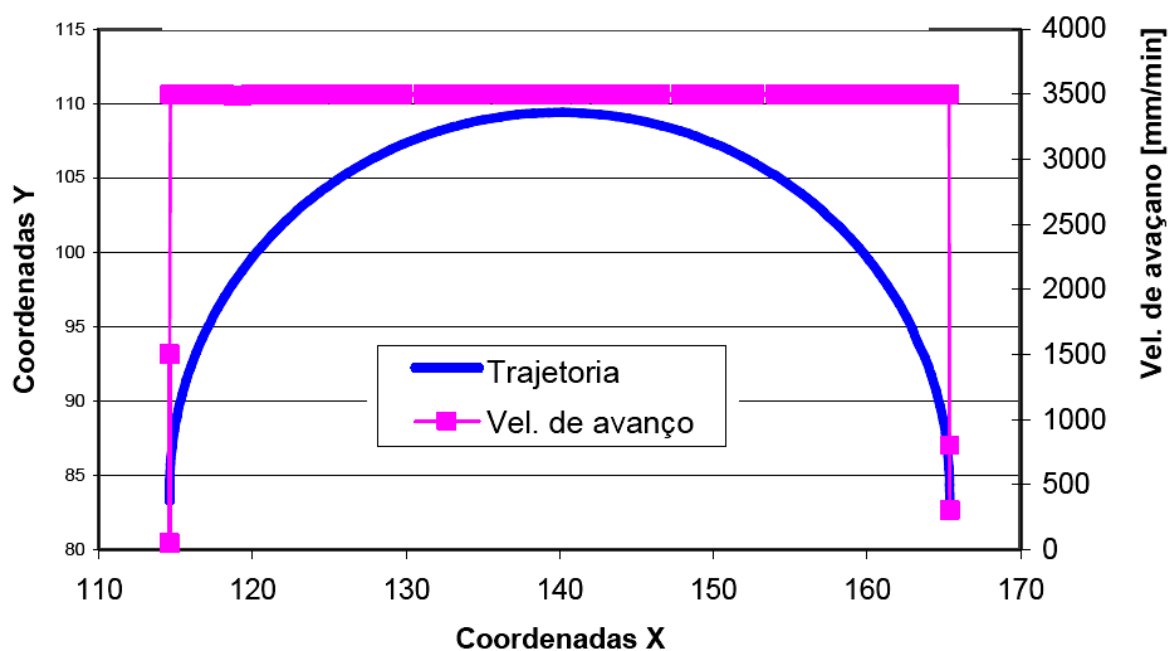


Figura 16 - Resultado interpolação circular (avanço de 3500 mm/min.) - Fonte Souza (2004).

Na avaliação do terceiro experimento com a utilização da interpolação linear e velocidade de avanço de 1000 mm/min, o avanço real encontrado foi coincidente com o programado. A média do avanço alcançado no primeiro experimento foi de 1978 mm/min enquanto que, com a utilização da interpolação circular, o avanço atingiu 3500 mm/min.

Schützer *et al* (2007) comparam a oscilação de avanço entre a utilização de trajetórias calculadas através de segmentos de retas G1 e através do uso de *spline*. Na utilização de *splines*, a trajetória da ferramenta não é calculada através de segmentos de reta ou por interpolação circular, mas por segmentos de curva, baseando-se em complexos modelos matemáticos realizados pelo controlador CNC ou pelo CAM.

Spline é uma curva não-concêntrica, não-reta e desenhada suavemente através de uma série de pontos, conhecida também como curva francesa (MUNDO CNC, 2011). A Figura 17 representa a formação da trajetória através dos pontos.

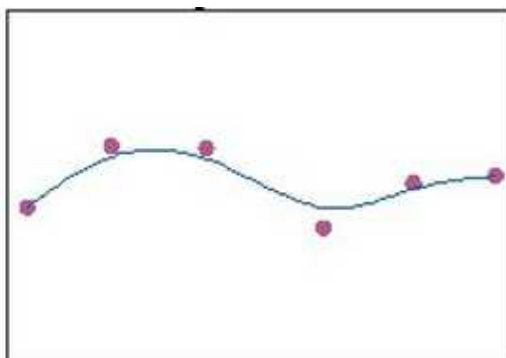


Figura 17 - Formação da *Spline* através de pontos. Fonte: <http://haydenpub.com/book2samples.html>

Com a utilização desse método, os sistemas CAM podem determinar a trajetória da ferramenta de forma mais suave e precisa, adaptando-se à tolerância desse sistema. Os autores analisam dois sistemas CAM, aplicando tolerâncias de cálculos entre 0,05 mm e 0,005 mm e três velocidades de avanço (2500 mm/min, 8000 mm/min e 15000 mm/min). Foram usinados corpos de provas em duas máquinas CNC.

Para uma velocidade de 2500 mm/min e tolerância de cálculo de 0,05 mm, os autores constataram uma pequena variação do avanço. Schützer e Helleno (2007) chegaram às seguintes conclusões:

- a) Melhor desempenho com relação ao comportamento da velocidade real das interpolações *splines* sobre a interpolação linear em relação à velocidade programada;
- b) Mesmo máquinas com alto desempenho dinâmico apresentam limitações tecnológicas decorrentes da utilização da interpolação linear na representação da trajetória da ferramenta na manufatura de superfícies complexas;
- c) Comportamentos distintos da velocidade de avanço real em algumas regiões do corpo de prova, decorrentes da utilização de diferentes estratégias.

Souza (2004) e Souza & Coelho (2007) sustentam que as trajetórias de ferramentas utilizando funções *splines* necessitam de dois valores de tolerância para cálculo. Sendo a tolerância para o cálculo inicial das trajetórias lineares, que segundo estas o *software* calcula as novas trajetórias *splines*.

Os algoritmos e os sistemas de cálculos para trajetórias *splines* ainda necessitam de desenvolvimento, uma vez que muitas equações para a trajetória

spline são geradas para representar movimentações lineares relativamente pequenas.

Estudos realizados por Boujelbene *et al* (2004) comparam as trajetórias de ferramentas compostas por segmentos de retas utilizando pontos a partir de trajetórias lineares e circulares. Nesses estudos, os autores concluíram que as trajetórias linear/circular apresentam vantagens no que se refere ao tamanho dos arquivos para os programas CNC e a um menor grau de erro geométrico do modelo com menor rugosidade superficial, o que gera, por sua vez, um menor tempo de polimento/acabamento de cavidades.

Chung (2004) apresenta de forma gráfica na Figura 18 a divisão do perímetro de parte de uma circunferência em um segmento de reta (Δs_k).

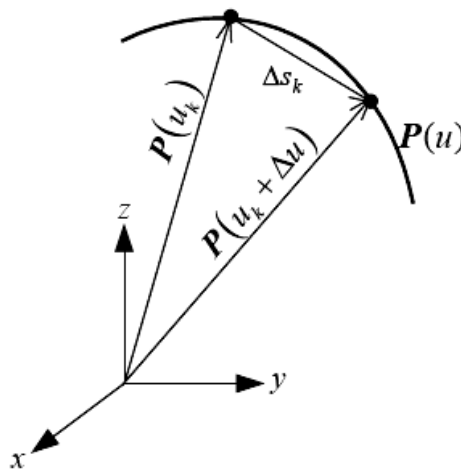


Figura 18 - Divisão da interpolação em segmentos de reta. Fonte: Cheng (2004).

Com o objetivo de explorar as variáveis que afetam a velocidade de trabalho, Chung (2004) iniciou suas análises através da equação 1:

$$Vt = \left| \frac{dP}{dt} \right| \quad (1)$$

Onde:

P = Ponto da trajetória

t = Tempo

Baseando-se nos dados tecnológicos, componentes do programa CNC, que determinam a velocidade de avanço, foi definida uma nova equação para esse tipo de aplicação, em que um novo incremento de velocidade (u) se faz necessário quando a interpolação é dividida em segmentos de reta definidos. O avanço programado $V(t)$ é conhecido e o objetivo é controlar uma possível taxa de aumento do parâmetro u para que esse avanço seja realmente alcançado. Uma vez que $u(t)$ é crescente, tem-se:

$$\frac{du}{dt} = \frac{V(t)}{\left| \frac{dP(u)}{du} \right|} = V(t)f(u) \quad (2)$$

Onde:

u = Incremento de velocidade

V = Avanço programado

Esta é uma equação diferencial para u . Dada uma corrente de valor u_k e os valores anteriores, objetivou-se resolver a equação diferencial ao longo de um intervalo de tempo de duração t , de modo a obter o próximo valor do parâmetro, formando u_{k+1} .

Chung (2004) utilizou o método preditor-corretor de Adams-Moulton, que permite aumentar a velocidade teórica de execução do programa para que a taxa de avanço real requerida seja alcançada. O método em questão utiliza incrementos de acelerações e desacelerações entre os intervalos dos pontos P . Dessa forma, o autor definiu uma velocidade de avanço de 50 mm/s e Δs_k de 0.0005 mm, chegando a resultados de acelerações média de 100 mm/s² e máxima de 150 mm/s², entre a partida e a chegada dos pontos, resultando em uma melhor uniformidade da velocidade programada. A Figura 19 apresenta o resultado para um experimento com tempo total de 8,22 s.

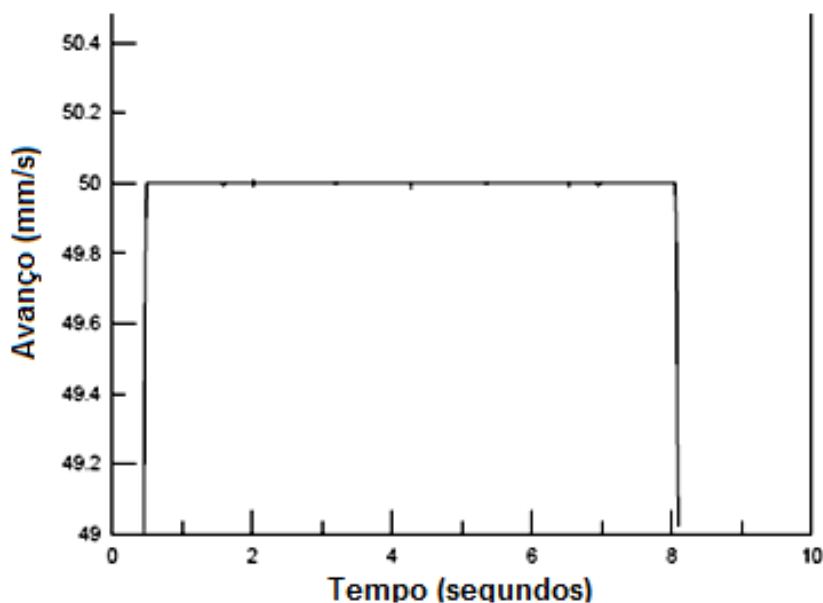


Figura 19 - Precisão da velocidade de execução da interpolação. Fonte: Adaptado de Cheng (2004).

Segundo Nikiel (2009), essa situação pode ser contornada utilizando uma programação paramétrica e levando em conta o projeto de um programa em que o tempo de execução dos blocos seja minimizado. Isso depende de dois fatores: da qualidade do programa ou algoritmo e do método de sua aplicação, sendo o primeiro fator o mais importante.

O autor ainda investiga acerca da velocidade de execução de programas paramétricos quanto à forma de programação. Os resultados foram avaliados em um controlador Siemens Sinumerik 810D, utilizado comumente nas indústrias. Apesar das avaliações serem realizadas em um controlador definido, as conclusões obtidas podem ser entendidas a outros controladores.

As avaliações centraram-se nas velocidades dos cálculos do controlador sobre as variáveis de parâmetro global R e na velocidade de cálculo no caso da utilização de funções complexas na programação, como função quadrada. Também são apresentados resultados quanto à velocidade de cálculos com expressões aritméticas complexas na utilização de variáveis estruturais como WHILE, ENDFOR, GOTO e na utilização de subprogramas.

A utilização da função \$AC_TIMER[14] acrescentada no corpo da programação permite ao usuário a apresentação dos dados do tempo de usinagem após o término da execução do programa. O autor realizou a contagem do tempo necessária para execução dos cálculos de cada intervalo definido de programação.

As formas de programação com objetivo de obtenção de um único resultado final contribuem diretamente no desempenho do controlador do comando numérico. O tempo de processamento das informações é tomado como base para a execução dos movimentos dos eixos da máquina. O intervalo de comando gerado pelo controlador para o acionamento dos eixos é influenciado pelo tempo do processamento do controlador. Utilizando-se de interpolações circulares divididas em pequenos segmentos de retas, o controlador pode levar um maior período para cálculo se comparado à alta aceleração e velocidade dos eixos para movimentação entre os pontos definidos. A Figura 20 apresenta os resultados comparativos entre programas gerados através da utilização de subprogramas em relação à forma paramétrica simples.

Descrição das Variáveis	Com Subprograma	Sem Subprograma
Fragmentos do programa testado.	<pre>FOR PP=1 TO 1000 CALL "TESTO" ENDFOR TESTO.SPF: R1=1 R2=R1+1 R3=R2+1 R4=R3+1 ...</pre>	<pre>FOR PP=1 TO 1000 R1=1 R2=R1+1 R3=R2+1 R4=R3+1 ... ENDFOR</pre>
Tempo médio	6,90 ms	1,48 ms

Figura 20 - Estimativa de tempo para cálculos de subprogramas. Fonte: Adaptado de Nickil (2009).

Esses programas paramétricos, ainda assim, podem ser afetados por um tempo de cálculo maior em relação a sua forma de programação. Nikiel (2009) apresenta, em seu experimento, a variação do tempo de execução da programação em função da sua forma de apresentação (Figura 21).

Descrição das Variáveis	Uso de Equação	Sem uso de Equação
Fragmentos do programa testado.	$R17 = (((R1 + R2) + (R3 - R4) - (R5/R6) - (R7/R8)) / R9) * R10$	$R11 = R1 + R2$ $R12 = R3 - R4$ $R13 = R5 / R6$ $R14 = R7 * R8$ $R15 = R11 + R12 - R13 - R14$ $R16 = R15 / R9$ $R17 = R16 * R10$
Tempo médio	5,78 ms	14,54 ms

Figura 21 - Estimativa de tempo para cálculos. Fonte: Adaptado de Nickil (2009).

3 METODOLOGIA

Para o desenvolvimento e execução dos programas, foi utilizado um *software* CAM e um centro de usinagem vertical ROMI 560 (Figura 22), dotado de um controlador SIEMENS Sinumerik 810D cedido pelo Instituto Federal Sul Riograndense - Campus Passo Fundo.



Figura 22 - Centro de Usinagem Vertical Romi Discovery 560.

A comparação do desempenho entre os programas gerados por um *software* CAM e os programas paramétricos construídos diretamente em um editor de textos e/ou teclado numérico do controlador é realizada através da utilização da maior velocidade de avanço disponível da máquina, limitada em 8000 mm/min. O tempo final de cada execução é comparado entre as formas de programação, assim como a utilização da memória do controlador por cada programa. Os tempos de execução dos programas são avaliados em função da capacidade de processamento do controlador, dos cálculos internos gerados por programas paramétricos e das velocidades de movimentação dos eixos para ambas as formas de programação,

baseados na forma de leitura do programa pelo controlador para o início e fim de cada ponto programado.

O *software* CAM utilizado permite, através de geometrias 2D e 3D, a criação de trajetórias de ferramentas de acordo com a necessidade do usuário. Sua plataforma é apresentada em conjunto ao *software* de modelagem sólida, apresentado na Figura 23.

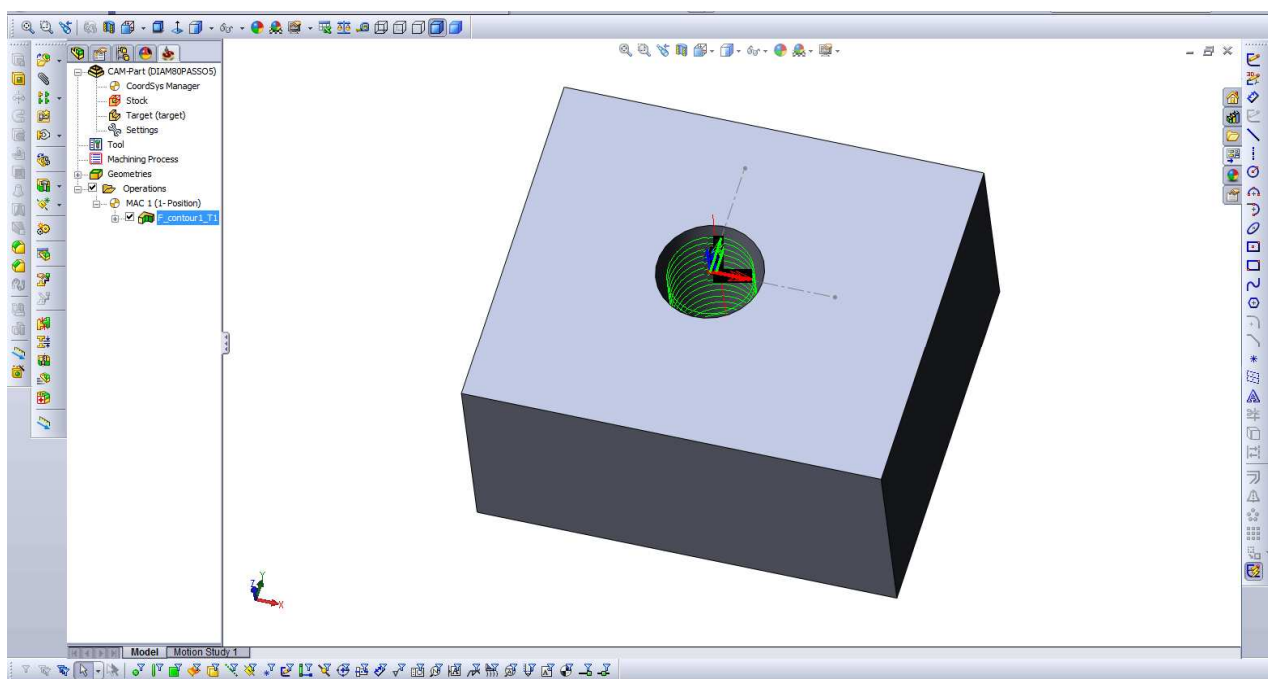


Figura 23 - Plataforma de programação CAM.

Os programas que são gerados pelo CAM têm seu desempenho avaliado em função da velocidade e do tempo de execução da trajetória de acordo com os códigos de comando que os compõem. Os programas gerados através da união de pontos por retas e a interpolação linear (G1) são também executados através do recurso do controlador CNC para a leitura de blocos a frente da execução (G64).

O controlador permite, através de recurso instalado, a leitura de até 100 blocos adiante da execução, suavizando os movimentos dos eixos no que se refere à parada exata a cada mudança de bloco, através da aplicação do comando G64 no cabeçalho de cada programa.

A suavização da aceleração e desaceleração dos movimentos dos eixos, quando o programa é executado a partir do arquivo gerado em CAM, em que os movimentos são gerenciados a partir de segmentos de reta entre 0,001 a 0,5 mm, gera, inicialmente, grandes acelerações e desacelerações entre os pontos, como apresentado na Figura 24, que compara a curva de aceleração e a velocidade teórica e a real. A curva de aceleração teórica apresenta “ T_a ” como o tempo de aceleração para a trajetória e “ T_d ” como o tempo de desaceleração para alcançar a trajetória teórica. Ao mesmo tempo, a Figura 24 demonstra as velocidades reais atingidas para cada mudança de bloco de programação (N1, N2,..... N100).

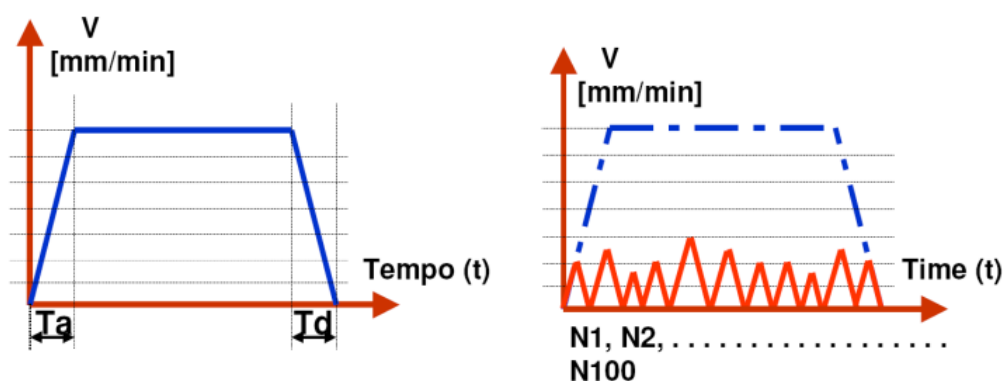


Figura 24 - Curva de aceleração e velocidade. Fonte: Manual Programação ROMI (2002)

A Figura 25 apresenta graficamente o recurso de suavização das acelerações/desacelerações dos movimentos executados pela máquina.

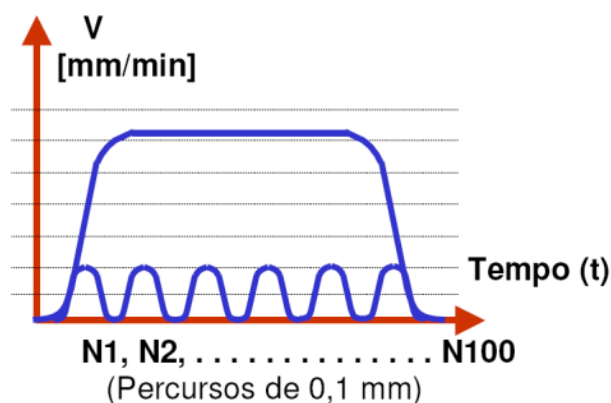


Figura 25- Suavização de aceleração/desaceleração. Fonte: Manual Programação ROMI (2002).

3.1 Linguagem adotada na programação paramétrica

A linguagem da programação paramétrica para o comando SIEMENS 810D utiliza variáveis “R”, ligadas diretamente no corpo do programa e modificadas pelo utilizador de acordo com suas necessidades (ROMI, 2002).

Parâmetros de cálculo “R” são registros fixos de R0 a R99 disponíveis para definição de variáveis na programação. O desenvolvimento de programas de família de peças que tem a mesma geometria, porém, valores dimensionais variáveis, permite, por sua vez, o desenvolvimento de perfis bidimensionais e/ou tridimensionais gerados ponto a ponto cujas coordenadas são calculadas através de algoritmos contidos dentro do programa com desvios condicionais. Aos parâmetros “R” podem ser atribuídos valores diretos ou indiretos cujo resultado deverá estar contido na seguinte gama:

Exemplo: R0=3.5678 R1=-36.4 R3=-3.65 R4=-6765.1234

Através da notação exponencial “EX”, pode ser alocada uma gama mais ampliada de números. Assim, escreve-se o valor do expoente após os caracteres EX.

Exemplo: R0=-0.1EX-5

Significado: R0=-0.000 001

R1=1.874EX8

Significado: R1=187 400 000

Nota: É permitido efetuar uma série de atribuição de valores em um único bloco, porém, o bloco deve ser exclusivo para atribuição.

Exemplo válido: R0=3.5678 R1=-36.4 R3=-3.65EX3 R4=3.23EX-2

Os parâmetros de cálculo e/ou expressões aritméticas poderão substituir valores em todos os endereços do programa, exceto N, G e L. Para tanto, escreve-se, após o caractere de endereço, o caractere “=” e a identificação do parâmetro, seguido ou não de expressão aritmética.

Exemplo:

N10 R5=24 R10=250 Efeito: Atribuído valor 24 ao parâmetro R5 e 250 ao R10.

N20 G1 X=R5 F=R10 Efeito: A máquina deslocará até a coordenada X24 numa $V_f = 250$ mm/min

3.1.1 Operações/funções aritméticas

Ao aplicar o cálculo funções aritméticas é necessário observar a notação matemática usual, isto é, a multiplicação/divisão tem prioridade sobre adição/subtração. Para definir prioridade arbitrária e/ou específica da execução, usam-se os “parênteses”. Em todas as circunstâncias em que são usados valores em graus, estes devem ser declarados em milésimos.

Exemplos:

N60 R1=8 R20=SIN(30.345) R9=R7*R8 R12=R10/R11

N70 R13=R1*R20-R9

N80 R15=SQRT(R13+R9*R1)

Os parâmetros de cálculo "R", conforme visto no capítulo anterior, podem ser submetidos a diversos tipos de cálculos. Os principais operadores são:

+ Adição

- Subtração

* Multiplicação

/ Divisão

SIN(*)

Seno

COS(*)

Cosseno

TAN(*)

Tangente

ASIN(*)

Arcoseno

ACOS(*)

Arcocosseno

ATAN2(*)

Arcotangente

SQRT(*)

Raiz quadrada

ABS(*)

Número absoluto

POT(*)

Potência de 2 (elevação ao quadrado)

TRUNC(*)	Truncar p/ inteiro
ROUND(*)	Arredondar p/ inteiro
LN(*)	Logaritmo natural
EXP(*)	Função exponencial

* = inserir valor

R1=R1+2 Resultado: Valor contido em R1 + 2.

R3=SIN(30) Resultado: R3 conterà o valor do seno de 30.

R4=R1+R20/R3 Resultado: R4 conterà o resultado da equação.

R5=(R1+R20)/R3 Resultado: R5 conterà o resultado da equação.

A função aritmética ATAN2(__,__) calcula o ângulo do vetor resultante (hipotenusa) tendo como dados outros dois vetores (catetos) de um ângulo reto. O resultado considera um dos 4 quadrantes ($-180 < 0 < 180^\circ$). A referência angular é sempre baseada numa linha originada no vértice do 2º valor seguindo na direção positiva.

Exemplos:

R20=ATAN2(30.5, 80.1) Resultado: 20.8455º

R21=ATAN2(30.5, -80.1) Resultado: 159.1545º

R22=ATAN2(-30.5, -80.1) Resultado: 200.8455º

R23=ATAN2(-30.5, 80.1) Resultado: 339.1545º

3.1.2 Operadores Lógicos e de Comparação

Operadores lógicos são usados para checar a condição de verdadeiro ou falso numa comparação entre dois valores, efetuando um desvio condicional.

Sintaxe: IF (comparação) GOTO? (LABEL DESTINO)

NOTA 1: “?” O desvio pode ser para um bloco (LABEL) que está para trás ou para diante do bloco condicional. Se estiver para trás, usa-se GOTOB (GO TO BACKWARD), se para diante GOTOF (GO TO FORWARD).

NOTA 2: LABEL DESTINO é a identificação do bloco para o qual a execução deverá ser desviada, caso o resultado da comparação seja verdadeiro.

NOTA 3: Caso o resultado da comparação não seja verdadeiro, não haverá desvio, logo, a execução continuará no próximo bloco.

Exemplo 1: IF R10>=R11 GOTOB INICIO

Efeito: Caso R10 seja maior ou igual a R11, a execução será desviada para o bloco (LABEL) nomeado como INICIO, que está programado para trás deste bloco.

Exemplo 2: IF R20= = (SIN(R31)) GOTOF POSICAO

Efeito: Caso R20 seja igual ao seno de R31, a execução será desviada para o bloco (LABEL) nomeado como POSICAO, que está programado para diante deste bloco.

Os operadores de comparação podem ser utilizados para formular uma condição de desvio. Expressões complexas podem também ser comparadas. São eles:

= = Equal to (Igual a);

< > Not equal to (Não igual a);

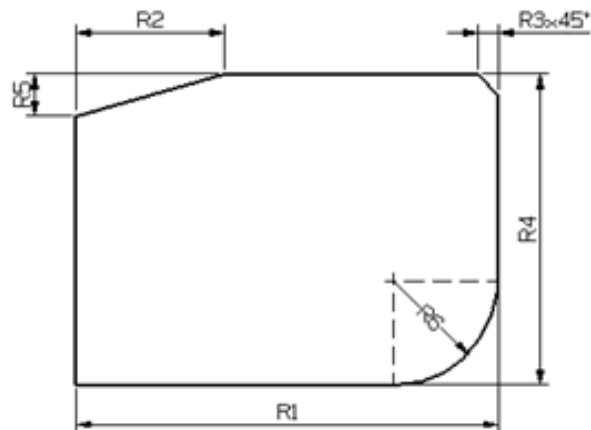
> Greater than (maior que);

< Less than (menor que);

> = Greater than or equal to (maior ou igual a);

< = Less than or equal to (menor ou igual a);

Um exemplo de programa paramétrico é apresentado na Figura 26.



```

G17 G64 G71 G90 G94
G53 G0 Z-110 D0
T1
M6
G54 D1 S1000 M3
R1=200; _____ COMPRIENTO TOTAL DA PECA
R2=70; _____ COMPRIENTO DO "CHANFRO" 1
R3=10; _____ CHANFRO 2 (45 GRAUS)
R4=150; _____ LARGURA TOTAL DA PECA
R5=20; _____ LARGURA DO "CHANFRO" 1
R6=50; _____ RAI0
R7=10; _____ DIAMETRO DA FERRAMENTA
R8=3; _____ FOLGA PARA A APROXIMACAO
R9=350; _____ AVANÇO
R7=R7/2; _____ CALCULO DO RAI0 DA FERRAMENTA
G0 X=-(R7+R8) Y=-(R7+R8)
Z5
Z-5
G42 G1 X0 Y0 F=R9
X=R1-R6
G3 X=R1 Y=R6 CR=R6
G1 Y=R4-R3
X=R1-R3 Y=R4
X=R2
X0 Y=R4-R5
Y0
G40 X=-(R7+R8) Y=-(R7+R8)
G53 G0 Z-110 D0 M5
M30;

```

Figura 26 - Exemplo de programação paramétrica. Fonte: Manual Programação ROMI (2002).

A otimização paramétrica dessas operações é realizada principalmente para obter um baixo custo e tempo de usinagem. Alta produtividade das máquinas CNC depende de fatores também relacionados com a qualidade dos programas criados.

3.2 Interpolação Helicoidal

A primeira análise do tempo de execução do programa do seu tamanho, abrange a criação e execução de um programa para gerar um perfil de interpolação helicoidal. Esse perfil requer que os três eixos da máquina CNC se movimentem simultaneamente, obedecendo a sincronização para o passo da hélice e o diâmetro a ser executado.

Conforme apresenta a Figura 27, a criação de um programa utilizando a plataforma CAD/CAM permite a visualização do perfil a ser executado. O diâmetro do perfil terá o diâmetro e a profundidade definidos igualmente para cada método de programação. O tempo para a usinagem será mostrado no painel da máquina como tempo de execução.

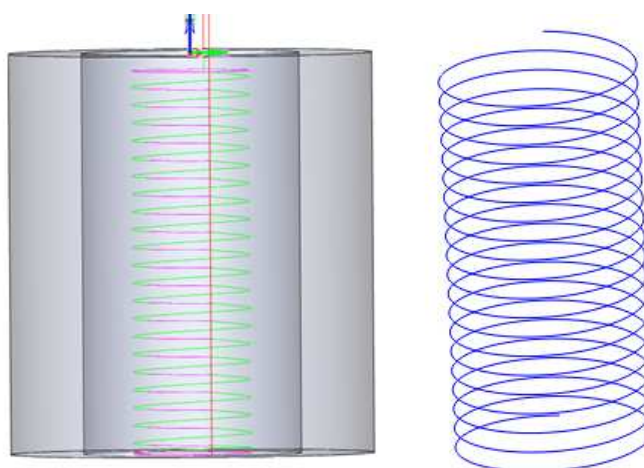


Figura 27 – Programação da interpolação helicoidal CAD/CAM.

A contagem do tempo para as diversas formas de programação considera uma única ferramenta partindo de um ponto específico, posicionada no centro da peça a uma altura definida no programa em relação a face. A posição inicial da ferramenta na máquina ficará localizada em seu ponto de altura máxima, zero máquina no eixo Z, no início de cada programa.

3.2.1 Interpolação Helicoidal CAM

Este programa apresenta em sua estrutura a geração da trajetória da usinagem em múltiplos segmentos de retas executados simultaneamente pelos três eixos (X, Y e Z). O número de caracteres gerados pelo programa CAM interfere diretamente no tamanho do programa em termos de ocupação da memória física do controlador CNC.

3.2.2 Interpolação Helicoidal Paramétrico

A interpolação helicoidal paramétrica apresenta em suas fases de execução, inicialmente, o posicionamento dos eixos até o ponto de início do raio da helicóide. Após esse posicionamento, o programa executa a interpolação circular com movimentação conjunta ao eixo (Z), formando uma helicóide. A posição do raio da helicóide é representada no programa pela variável (R5) e o incremento de profundidade a cada ciclo de volta completa é determinado por (R3). Após a execução da interpolação helicoidal até a profundidade programada, o programa incrementa o valor de profundidade (R3) e compara o valor com a profundidade máxima programada para o furo. Quando o valor é verdadeiro, o controlador converte para o final do programa, reposicionando os eixos até o centro da interpolação e retornando a ferramenta até o ponto inicial; caso contrário, executa-se novamente a interpolação para a nova posição de (Z).

As variáveis do programa são apresentadas a seguir:

- R1_____Diâmetro do furo;
- R2_____Profundidade do furo;
- R3_____Passo do helicóide;
- R4_____Diâmetro da ferramenta;
- R5= ((R1-R4)/2)_____Cálculo do raio;
- R6=R3_____Incrementa o valor do Z.

A simulação da programação paramétrica objetiva a execução da interpolação circular helicoidal com utilização do controlador para a realização dos cálculos, gerando o incremento de cada volta da helicóide e a posição de parada e viabilizando, durante toda a execução, o funcionamento regular dos motores dos

eixos de forma linear em velocidades constantes, como pode ser visto no fluxograma da figura 28.

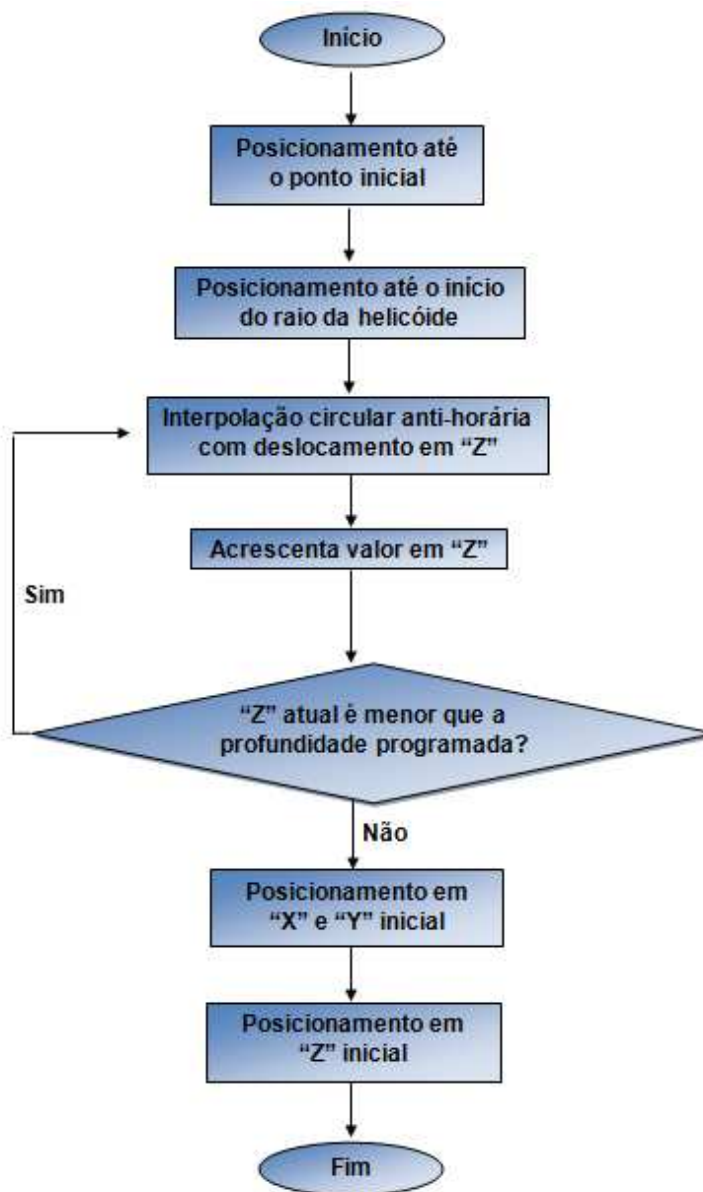


Figura 28 - Fluxograma para interpolação helicoidal paramétrica.

O programa paramétrico para construir uma helicóide de diâmetro e o passo da hélice proposto tem como base a utilização de uma linha de comando composta pelo comando (G3), que gerencia a movimentação dos eixos através de uma interpolação circular no sentido anti-horário, juntamente com o movimento do eixo (Z), formando uma interpolação helicoidal com resolução máxima do controlador e da máquina. A diferenciação principal para o programa gerado pelo CAD/CAM é a

utilização do controlador para obter a trajetória da ferramenta através das funções, das variáveis e dos cálculos apresentados em sua estrutura, ao contrário da programação inicial, que apenas permite ao controlador ler e executar o programa.

3.3 Interpolação Circular

A programação para a geração de uma trajetória circular (Figura 29), movimentando continuamente os eixos (X) e (Y) da máquina CNC, é apresentada de diferentes formas com o objetivo de verificar o desempenho da máquina e do controlador CN em relação às formas apresentadas com a máxima velocidade de avanço permitida pela máquina (8000 mm/min).

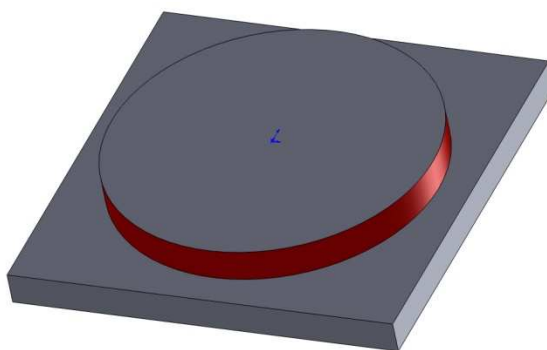


Figura 29 - Geometria para trajetória circular.

Os programas CN são apresentados de forma a extrair o desempenho da máquina a fim de executar um contorno de uma peça em uma altura e diâmetro externo iguais para as trajetórias programadas, conforme apresentado na Figura 30.

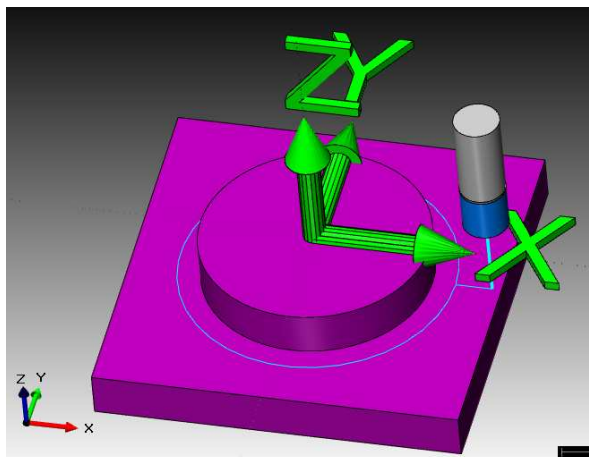


Figura 30 - Trajetória programada na geometria circular.

A forma de execução da trajetória da ferramenta é determinada de acordo com cada programa, de modo que é possível optar entre a geração de dois tipos de programas. O primeiro programa, gerado pelo CAM, contém trajetórias formadas por comando de interpolações circulares (G2) e (G3) e o segundo, formadas pelo perímetro da circunferência dividido em segmentos de retas. A comparação de desempenho é avaliada através do tempo utilizado pela máquina para a execução da trajetória.

O programa paramétrico é criado através da divisão do perímetro da circunferência em segmentos de reta, definidos pela variação do ângulo interno do raio da circunferência, sendo que a análise de desempenho da execução da trajetória em relação à variação angular programada para cálculo no programa paramétrico, varia de $0,1^\circ$ a 1° para a formação dos segmentos de reta (Figura 31).

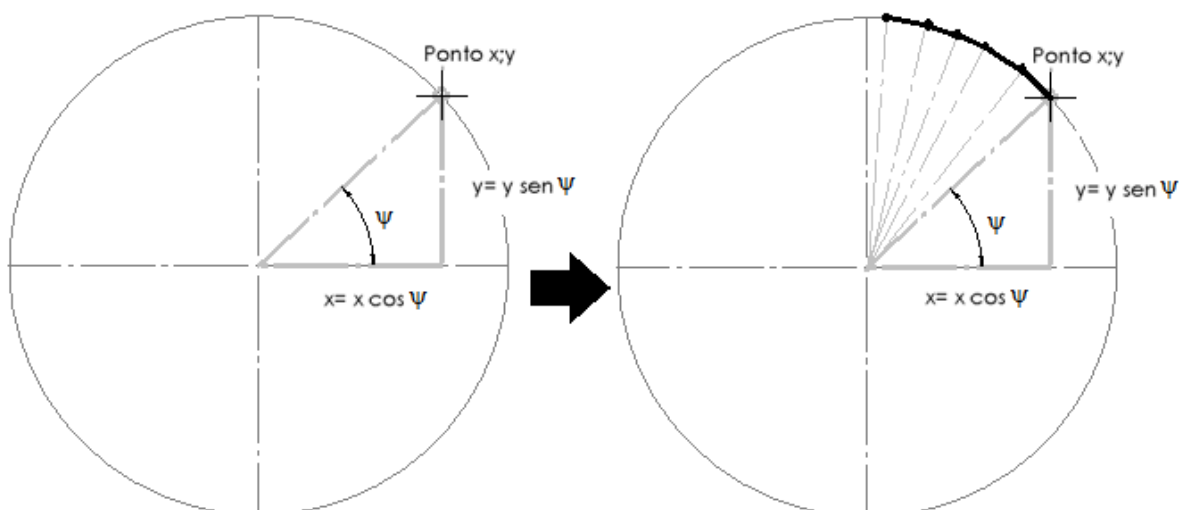


Figura 31 - Cálculo do ponto X; Y através de trigonometria.

Os programas são executados através da utilização em sua programação de uma ferramenta de diâmetro igual em todas as execuções com velocidade de avanço de 8000 mm/min. A rotação da ferramenta não tem influência sobre a velocidade de percurso da trajetória.

3.3.1 Interpolação Circular CAM – Segmentos de reta

A interpolação circular gerada pelo programa CAM é constituída por segmentos de reta variáveis de 0,1 a 0,5 mm para a formação do perímetro da circunferência. O programa CN é gerado a partir das seleções de programação oferecidas pelo *software*.

O tamanho de cada segmento de reta para a formação do contorno da geometria não pode ser definido no momento da programação em CAM, porém, pode ser identificado através da utilização das coordenadas geradas no programa e transferido a um programa CAD.

Os resultados de tempo e de velocidade são comparados, de modo a se verificar também o comportamento geral da máquina CNC em relação aos outros métodos de programação aplicados.

3.3.2 Interpolação Circular CAM – Arcos

O comando que executa a trajetória circular está localizado em um único bloco do programa, que contém a sintaxe (G2 X_ Y_ I_ J_). Os outros comandos são necessários para os posicionamentos dos eixos e por conterem informações tecnológicas do controlador. A programação da trajetória através de arcos é selecionada no *software* CAM e os arcos gerados são formados pelo *software* aleatoriamente, de forma que o usuário não controla no CAM a forma de obtenção do tamanho dos segmentos de reta.

Os dados tecnológicos para a formação da geometria circular, constituída por blocos de programação de interpolações circulares (G2). Com o objetivo de padronizar a execução do programa, são os mesmos utilizados nos outros métodos para esta mesma geometria.

3.3.3 Interpolação Circular Paramétrica

O programa CN, construído através da parametrização, conta com dados de entrada que estabelecem o raio da circunferência, os ângulos iniciais e finais do arco e a variação do incremento angular para definição do tamanho do segmento de reta, gerado no perímetro da circunferência.

As variáveis componentes do programa paramétrico são:

R1 _____ Raio do círculo;

R20 _____ Diâmetro da ferramenta;

R3 _____ Ângulo do início do raio/arco;

R4 _____ Ângulo do final do raio/arco;

R5 _____ Incremento angular/tolerância angular;

R6=R20/2 _____ Cálculo do raio da ferramenta;

R7 _____ Distância de aproximação da ferramenta;

R9 _____ Profundidade de corte;

R10 _____ Velocidade de avanço em mm/min.;

R1=R1+R6 _____ Cálculo do raio do arco em função do raio da ferramenta.

A trajetória da ferramenta para a interpolação circular paramétrica é definida por cálculos trigonométricos em função da tolerância angular, formando um ciclo de retorno e de incremento até que a condição programada seja atingida, conforme o fluxograma da Figura 32.



Figura 32 - Fluxograma da programação paramétrica - interpolação circular.

O programa paramétrico tem sua variável (R5) que define sua variação angular para o cálculo dos segmentos de reta com valor de $0,1^\circ$ e 1° avaliados, respectivamente.

3.4 Elipse

A elipse avaliada, apresentada na Figura 33, possui largura definida no eixo (X) e altura no eixo (Y); a profundidade é determinada no eixo (Z).

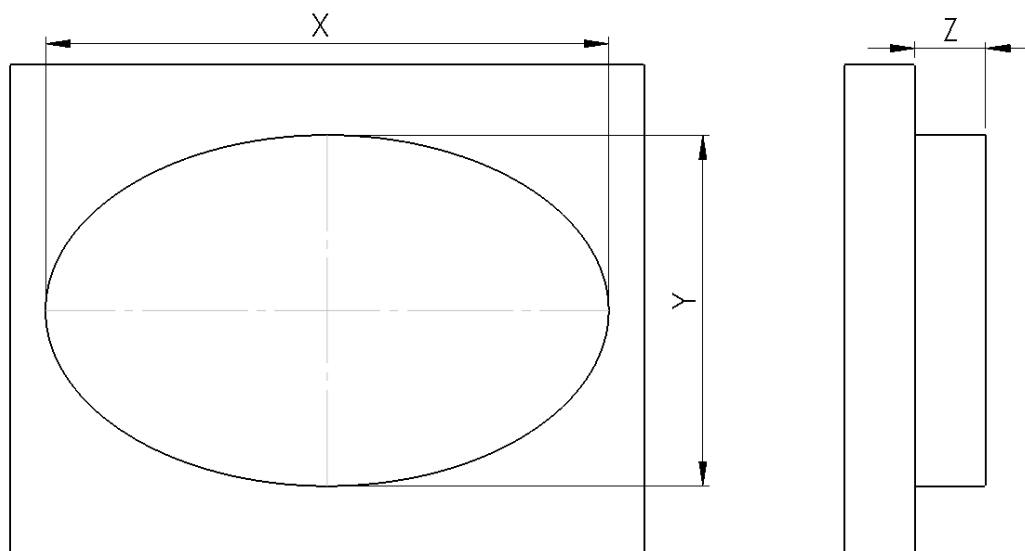


Figura 33 - Geometria elíptica para geração da trajetória.

A avaliação da geração da trajetória elíptica segue os mesmos princípios de programação da interpolação circular. A programação em CAM é avaliada com os programas que formam o contorno da elipse através de arcos (G2), conforme representado na Figura 34.

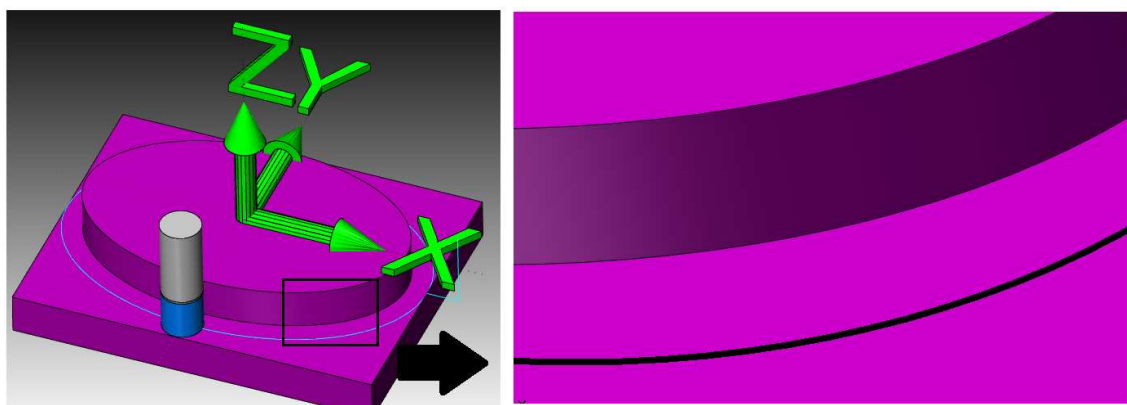


Figura 34 - Trajetória em arcos para a geometria elíptica.

A comparação de desempenho também é realizada com a utilização da programação CAM, através da divisão do perímetro da geométrica em segmentos de reta (Figura 35).

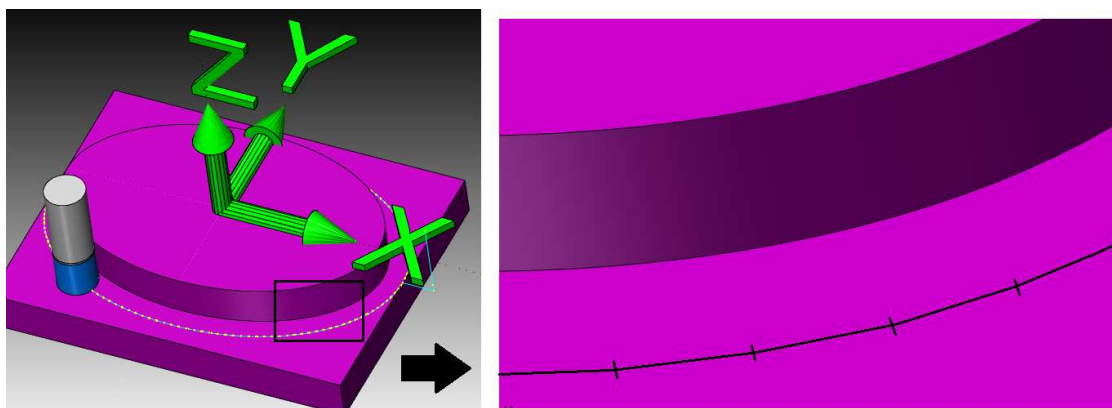


Figura 35 - Trajetória em segmentos de reta para geometria elíptica.

Para todos os ensaios, a geometria da ferramenta e da elipse são as mesmas.

3.4.1 Elipse CAM – Segmentos de Reta

O programa gerado pelo *software* CAM, para a trajetória da ferramenta na execução do contorno da elipse, é determinado para conter segmentos de reta com tamanho variável entre 0,1 mm a 0,5 mm, distribuídos aleatoriamente no perímetro da geometria. O controlador CN segue diretamente os comandos apresentados nos blocos do programa.

3.4.2 Elipse CAM – Arcos

A trajetória elíptica é também gerada pelo programa CAM com solicitações de segmentos circulares definidos por elementos de arcos gerados através de

interpolações circulares horárias (G2). Os blocos dos programas são gerados focalizando a criação dessas trajetórias circulares (G2), intercalando os blocos com os segmentos de retas (G1) resultantes da configuração de precisão de 0,01 mm para a trajetória.

Devido à tolerância de 0,01 mm aplicada ao CAM na superfície da geometria, visando a formação dos blocos de comando para a trajetória elíptica, composta pelos arcos (G2/G3), o programa é apresentado contendo partes em segmentos de reta e partes em arcos no sentido horário (G2), conforme o exemplo no programa da Figura 36.

G1 X-9.814 Y-29.293	—————▶	Segmento de reta
G2 X-17.934 Y-27.565 I9.221 J63.244	—————▶	Arco'
G1 X-17.944 Y-27.562	—————▶	Segmento de reta
G2 X-24.43 Y-25.283 I15.39 J54.167	—————▶	Arco
G1 X-24.441 Y-25.278	—————▶	Segmento de reta
G2 X-29.851 Y-22.572 I18.999 J44.756	—————▶	Arco
G1 X-29.865 Y-22.564	—————▶	Segmento de reta
G2 X-34.239 Y-19.615 I20.784 J35.542	—————▶	Arco

Figura 36 - Exemplo Programa elipse – Arcos.

3.4.3 Elipse Paramétrica

A criação do programa paramétrico baseia-se na criação da trajetória através da maior distância do centro da geometria para o arco no eixo (X) e no eixo (Y), utilizando cálculos trigonométricos que definem o tamanho do segmento de reta para o perímetro da elipse.

As equações trigonométricas que definem a relação entre os comprimentos dos eixos (X) e (Y) e o ângulo de incremento estão representadas a seguir, onde a equação (1) define o ponto de deslocamento para o eixo (X) e a equação (2) representa o deslocamento do eixo (Y). Ambas as equações estão em função da variação do incremento angular.

$$X = (R1 * \cos \psi) \quad (1)$$

$$Y = (R2 * \sin \psi) \quad (1)$$

A Figura 37 demonstra geometricamente o cálculo realizado pelo controlador CN através do programa paramétrico para a obtenção do ponto (X; Y), em função dos comprimentos dos raios da elipse no eixo horizontal e vertical baseando-se no ângulo incremental ($^{\circ}$), definido em 0,1 $^{\circ}$ e 1 $^{\circ}$ para as respectivas avaliações.

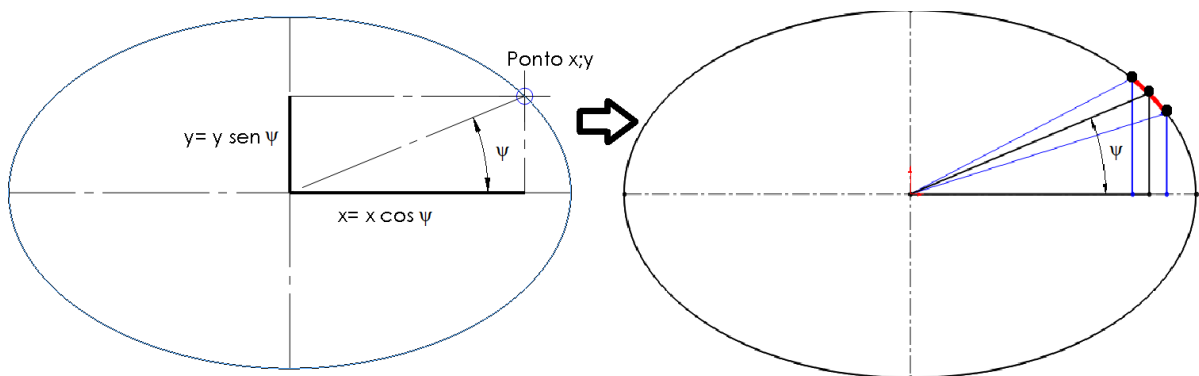


Figura 37 – Determinação dos descolamentos em x e y sobre a geometria elíptica.

O programa paramétrico é executado a partir dos dados inseridos para as seguintes variáveis:

- R1 _____ Maior comprimento;
- R2 _____ Menor comprimento;
- R20 _____ Diâmetro da ferramenta;
- R1 _____ Definição do raio dos eixos;
- R2= ((R2+R20)/2) _____ Definição do raio dos eixos;
- R3 _____ Ângulo inicial;
- R4 _____ Ângulo final;
- R5 _____ Incremento angular;
- R7 _____ Distância de aproximação;
- R10 _____ Velocidade de avanço.

O programa paramétrico assemelha-se em sua constituição ao programa para a trajetória de formação do círculo, também paramétrico, apenas com as condições modificadas para o raio no eixo (X), onde sua variável é definida como (R1) e no eixo (Y), que possui ligação com a variável (R2) para o tamanho do raio da elipse. A Figura 38 apresenta o fluxograma do programa.



Figura 38 – Fluxograma do programa paramétrico - Elipse.

O programa construído através da parametrização utiliza variáveis que delimitam a dimensão da elipse, da ferramenta utilizada e das compensações de aproximação da ferramenta através de cálculos programados, que somam o raio da ferramenta e o raio da elipse para a determinação da posição inicial.

3.5 Cavidade Circular

A geometria da cavidade circular é delimitada em um diâmetro de 80 mm com altura de 10 mm, executada por uma ferramenta de diâmetro de 10 mm com incremento de corte lateral de 70% do diâmetro da ferramenta. A taxa de avanço é de 8000 mm/min e a profundidade de corte é de 1 mm por passe.

A trajetória da ferramenta ocorre por interpolações lineares e circulares, a partir do centro da peça até o raio maior (Figura 39). O ciclo de desbaste é completado na região delimitada pelo diâmetro programado. A cada término da soma das trajetórias com os incrementos laterais, a ferramenta retorna ao início e parte para nova coordenada do eixo (Z).

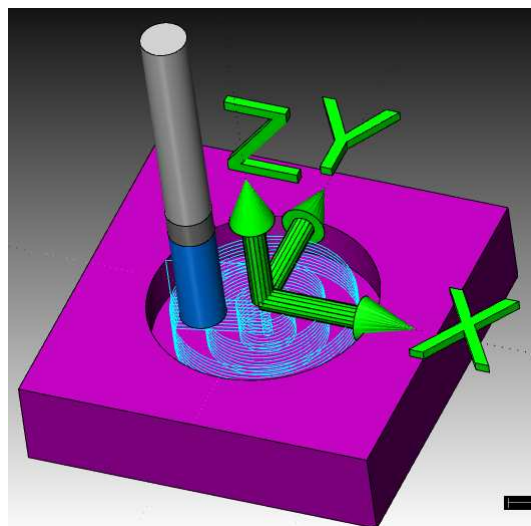


Figura 39 - Trajetória da ferramenta para a cavidade circular.

3.5.1 Cavidade Circular CAM – Segmentos de reta

O programa gerado pelo *software* CAM para o desbaste de uma cavidade circular, através da utilização de comandos de interpolação lineares, utiliza uma variação de comprimento de reta de 0,1 mm a 0,5 mm no perímetro das trajetórias que compõem o desbaste.

3.5.2 Cavidade Circular CAM – Arcos

A cavidade circular gerada em *software* CAM, utilizando o recurso de trajetórias circulares geradas por comandos de interpolação circulares (G2/G3) e tolerância de 0,01 mm, mescla blocos no comando NC com segmentos de reta (G1) e blocos com comandos de interpolação circular no sentido de usinagem anti-horário (G3).

3.5.3 Cavidade Circular Paramétrica

O programa paramétrico para a execução da cavidade circular parte do princípio de comparações entre o raio da ferramenta e o raio da cavidade, sendo que essas condições determinam a forma da execução da trajetória. A primeira condição compara se o diâmetro da ferramenta é maior que o diâmetro da cavidade, o que gera uma mensagem de erro e a parada do programa. A segunda condição verifica se o diâmetro da cavidade é equivalente ao diâmetro da ferramenta; se for atendida, a ferramenta seguirá uma trajetória de avanço em (Z) partindo do centro da cavidade até a altura programada. A terceira condição determina de que forma a cavidade com raio (R5) maior que o raio (R11) da ferramenta é executado. Para isso, a cavidade é dividida em alojamentos separadamente no programa paramétrico, de modo que condições de contorno determinem a seqüência da

execução e o momento de parada do programa, através de comparações do diâmetro da trajetória que está sendo executada com o diâmetro final indicado.

O raio da ferramenta utilizado é 5mm e o raio da cavidade gerada é 25mm com altura total de 10mm.

O programa paramétrico é composto pelas seguintes variáveis:

- R1_____ Plano de retorno em “Z”;
- R2_____ Plano de referência em “Z”;
- R3_____ distância de segurança em “Z”;
- R4_____ Altura da cavidade;
- R5_____ Raio da cavidade;
- R6_____ Coordenada do centro da cavidade em X;
- R7_____ Coordenada do centro da cavidade em Y;
- R8_____ Profundidade de corte;
- R9_____ Avanço em X (mm/min);
- R10_____ Avanço em Z (mm/min);
- R11_____ Raio da ferramenta;
- R12_____ Largura de corte (ae);
- R13=R2+R3_____ Cálculo do “Z” inicial.

Por fim, o programa paramétrico, como demonstrado através do fluxograma da Figura 40 é coordenado por condições que estabelecem a quantidade de alojamentos a serem executados com base no raio da ferramenta e no raio do círculo. O programa possui limitadores que acabam por identificar os dados inseridos pelo usuário e limitar as movimentações dos eixos através das condições previamente estabelecidas. Esse programa tem a metade do tamanho do menor programa gerado no CAM para a mesma função. Apesar de em sua construção a repetição do último passe de contorno no diâmetro externo da cavidade após toda a execução do desbaste.

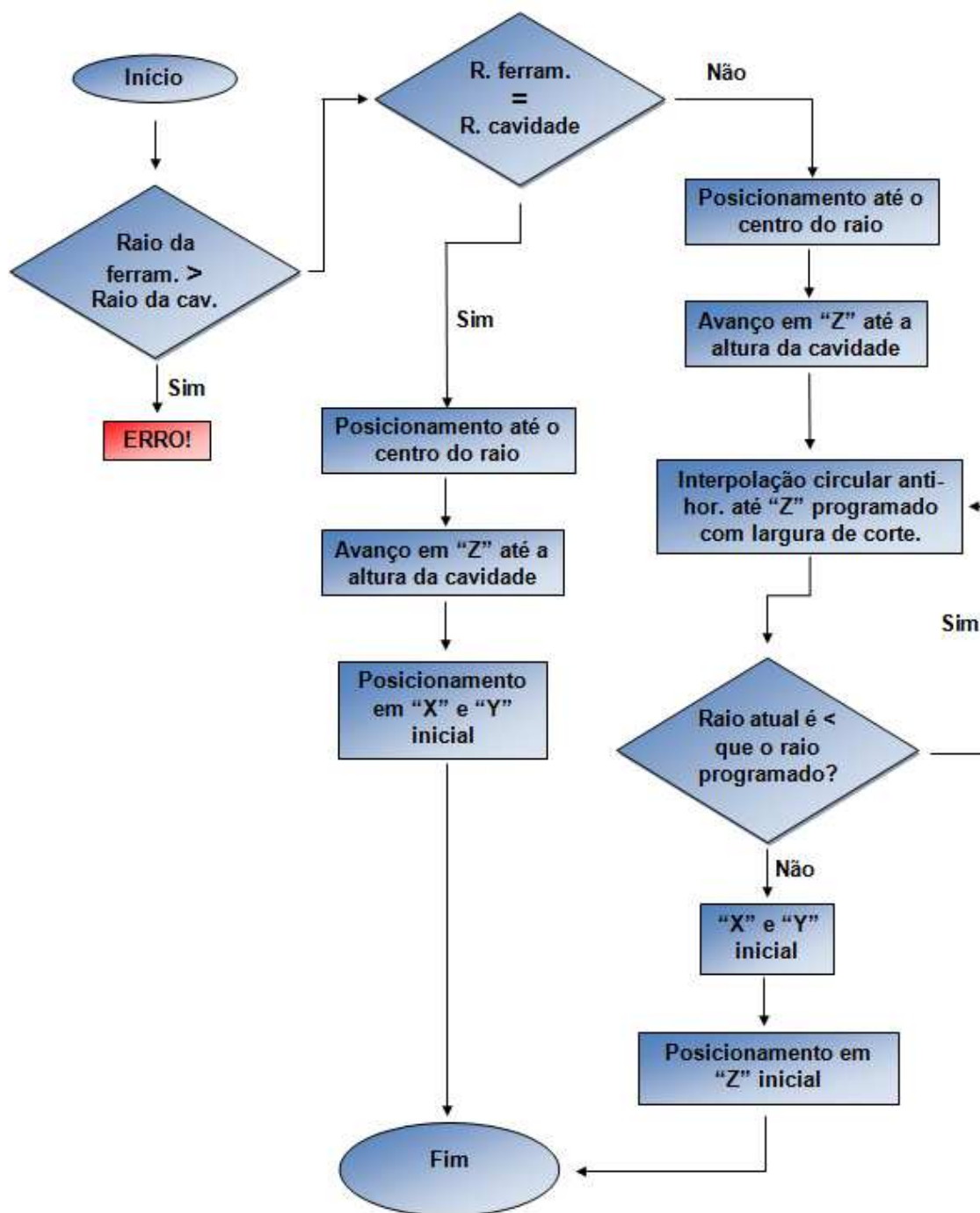


Figura 40 - Fluxograma Programa Paramétrico Cavidade Circular.

4 RESULTADOS E DISCUSSÕES

A execução e a comparação dos programas gerados em *software* CAD/CAM e manipulados via editor de textos, para o caso dos programas paramétricos, foram realizadas objetivando o máximo desempenho da máquina CNC com comando numérico SIEMENS Sinumerik 810D.

Os resultados apresentados a seguir, para cada um dos programas gerados em CAM e paramétricos, têm também seu desempenho comparado mediante a utilização ou não do recurso fornecido pelo controlador (G64), o que resultará em uma comprovação última de sua finalidade, bem como das diferenças entre as técnicas de programação empregadas.

4.1 Interpolação Helicoidal

O programa gerado pelo *software* CAM para uma interpolação helicoidal, utilizando uma ferramenta de diâmetro de 10 mm com rotação de 1000 min^{-1} e velocidade de avanço de 8000 mm/min, gerou um programa CNC de aproximadamente 6633 linhas, ocupando uma memória total na máquina de 168 kbytes. Esse programa conta com blocos de programação que utilizam a união entre pontos por segmentos de reta para a formação da trajetória. Os movimentos são denominados de interpolação linear (G1), conforme apresenta a Figura 41.

```

G17 G54 G90 G92 (6633 linhas)
T1 (THREAD MILL D10.0mm R0.mm)
S1000 M3
M8
G0 X6.252 Y-0.513
G43 H2 Z100.
G0 Z0.
G1 Z-2. F8000
.....
G1 X-6.707 Y0.633 Z-7.469
G1 X-6.728 Y0.336 Z-7.484
G1 X-6.739 Y-0.03 Z-7.501
G1 X-6.722 Y-0.483 Z-7.522
G1 X-6.662 Y-1.004 Z-7.546
G1 X-6.555 Y-1.56 Z-7.571
G1 X-6.418 Y-2.053 Z-7.593
G1 X-6.256 Y-2.504 Z-7.614
G1 X-6.069 Y-2.932 Z-7.633
.....
G1 X6.722 Y-0.5
G1 X6.739 Y-0.03
G1 Y0.
G0 X0
G0 Z100.
M9
G0 G53 Z0 H0
M30
%
```

Figura 41 - Programação CAM Helicoidal- Segmentos de retas.

A primeira análise de simulação com o programa CAD/CAM, utilizando a programação da interpolação helicoidal gerada em segmentos de reta sem intervenção do controlador na suavização das acelerações e desacelerações dos eixos, apresentou vibrações durante a execução da trajetória da ferramenta, tempo total de execução de 76 segundos e avanço real máximo atingido de 3600 mm/min, conforme demonstram, respectivamente, as Figuras 42 e 43.

ativo		/_N_WKS_DIR/_N_TESTE_WPD	
Parada exata não alcançada		HELICOIDAL	
MCS	Posição [mm]	PercRes	T,F,S
X1	-263.811	0.000	T 1 D1 ø 10.000
Y1	-192.221	0.000	F 3602. 100% 8000. mm/min
Z1	-207.432	0.000	S 1005. 100% 1000. I
P.zero1		0% 80% 100%	
Bloco atual		HELICOIDAL.MPF	
G1 X-18.225 Y8.238 Z-12.162 ↑			
G1 X-19.355 Y5.04 Z-12.297 ↑			
G1 X-19.928 Y1.696 Z-12.432 ↑			
G1 Y-1.696 Z-12.568 ↑			
G1 X-19.355 Y-5.04 Z-12.703 ↑			
G1 X-18.225 Y-8.238 Z-12.838 ↑			
G1 X-16.57 Y-11.199 Z-12.973 ↑			

Figura 42 - Velocidade de Avanço – Programação Segmentos de retas.

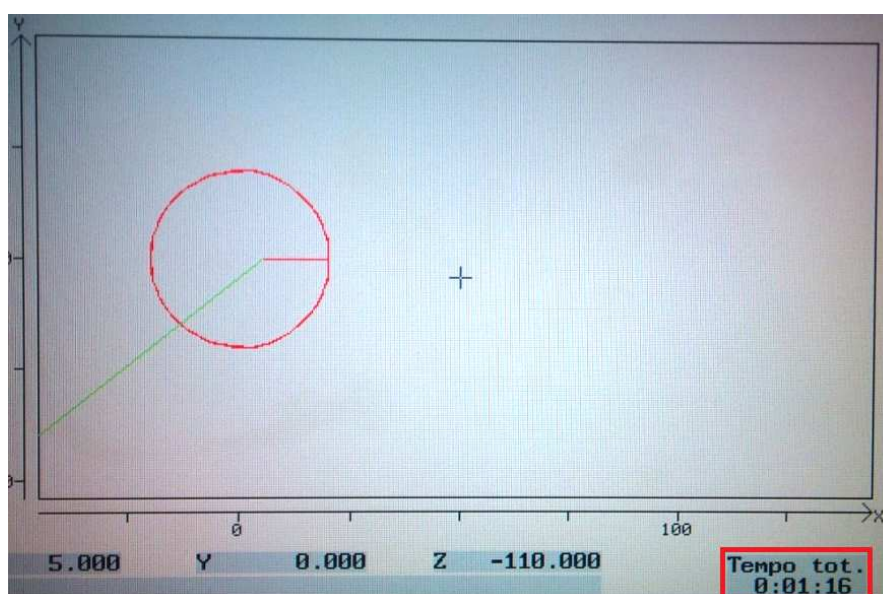


Figura 43 - Tempo de execução – Programação Segmentos de retas.

A aplicação do comando G64 para suavização das acelerações, posteriormente, em nova execução, teve como resultado o aumento da velocidade de avanço real e a redução significativa das vibrações geradas. O resultado após a simulação da aplicação demonstra um aumento na velocidade de avanço para aproximadamente 6700 mm/min, sendo que o valor programado era de 8000 mm/min.

A Figura 44 apresenta a visualização da tela do controlador com o velocidade de avanço programado e o atuante no momento da execução.

HELICOIDAL				
MCS	Posição [mm]	PercRes	T,F,S	
X1	-167.454	0.000	T 1	D1
Y1	-281.331	0.000	10.000	Z
Z1	-277.489	0.000	F 6723.	100%
			8000.	mm/min
			S 1005.	100%
			1000.	I
				100%
P.zero1				
Bloco atual HELICOIDAL.MPF				
G1	X0.849 Y-19.982 Z-38.784	↑		
G1	X4.214 Y-19.551 Z-38.919	↑		
G1	X7.457 Y-18.558 Z-39.054	↑		
G1	X10.486 Y-17.031 Z-39.189	↑		
G1	X13.213 Y-15.013 Z-39.324	↑		
G1	X15.561 Y-12.564 Z-39.459	↑		
G1	X17.46 Y-9.754 Z-39.595	↑		

Figura 44 - Velocidade de Avanço – Programação Segmentos de retas + G64.

O tempo necessário para executar o programa foi de 34 segundos (Figura 45).

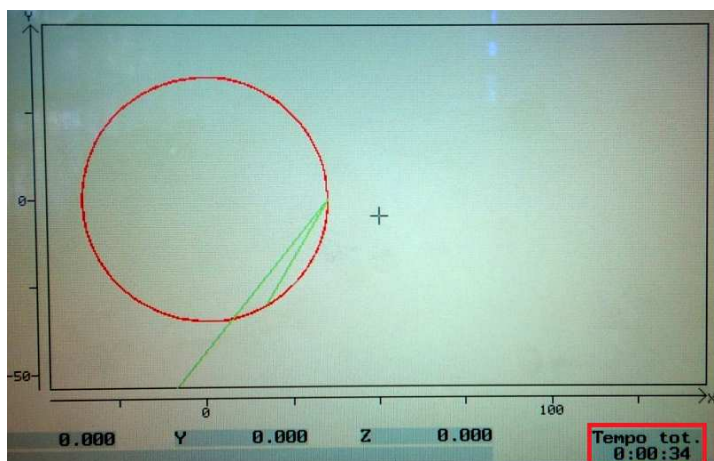


Figura 45 - Tempo de execução – Programação Segmentos de retas + G64

Por fim, realiza-se a entrada de dados no programa paramétrico no controlador NC. A partir da verificação do espaço físico utilizado em memória de armazenamento do controlador CNC, elaborou-se paralelamente um programa utilizando o editor de textos.

A aplicação da linguagem paramétrica teve como meta utilizar as funções do controlador para a geração da trajetória de usinagem através de variáveis e condições lógicas, como apresenta a Figura 46.

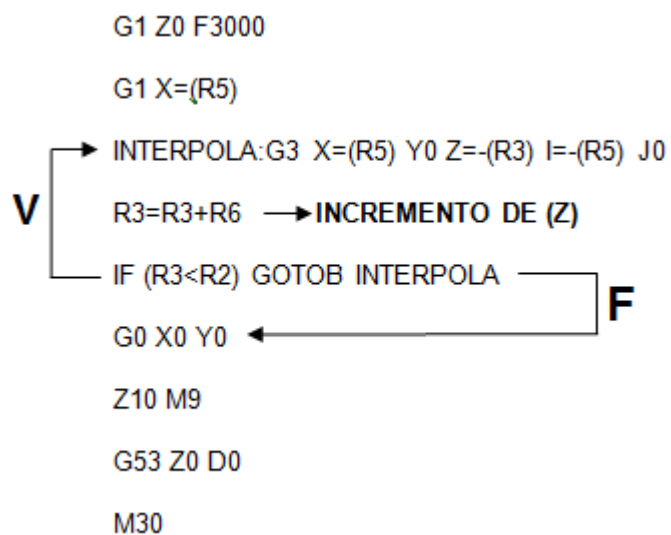


Figura 46 - Forma de execução do programa pelo controlador NC.

O programa construído ocupou um espaço de 394 bytes e 26 linhas, conforme apresentado na Figura 47.

```

(%_N_TESTEPARAMETRICO_MPF
; (INTERPOLAÇÃO HELICOIDAL PARAMETRICA)
R1=50           ;diâmetro do furo
R2=100          ;profundidade do furo
R3=5            ;passo do helicóide
R4=10           ;diâmetro da ferramenta
R5= ((R1-R4)/2) ;calculo do raio
R6=R3          ; incrementa o valor do z
G17 G64 G71 G90 G94
G53 G0 Z-110 D0
T1
M6
G54 D1
S1000 M3
G0 X0 Y0
Z2
G1 Z0 F3000
G1 X=(R5)
INTERPOLA:G3 X=(R5) Y0 Z=-(R3) I=-(R5) J0
R3=R3+R6
IF (R3<R2) GOTOB INTERPOLA
G0 X0 Y0
Z10 M9
G53 Z0 D0
M30
%
```

Figura 47 - Programa paramétrico helicóide.

O resultado é uma velocidade de avanço igual à velocidade programada de 8000mm/min com tempo de execução de 26 s (Figura 48).

HELICOIDAL				
MCS	Posição [mm]	PercRes	T,F,S	
X1	-201.862	58.922	T 1	D1
Y1	-238.752	-34.985	F 8000.	100%
Z1	-255.149	-3.727	S 1006.	100%
p.zero1			80%	100%
Bloco atual HELICOIDAL.MPF				
G0 Z2↑				
N140 G1 Z0 F200 ↑				
N150 G1 X=(R5) F8000 ↑				
N160 G3 X=(R5) Y0 Z=-(R3) I=-(R5) J0 F8000 ↑				
N170 R3= R3+ R6 ↑				
N180 IF(R3 <=R2) GOTO N160 ↑				
N190 G0 X0 Y0 ↑				

Figura 48 - Velocidade de Avanço – Programação Paramétrica.

A partir dos dados coletados, realiza-se a comparação do desempenho para a geração da trajetória helicoidal, através da aplicação de diferentes métodos de programação no *software* CAM, bem como a utilização do recurso da melhora do desempenho oferecido pelo controlador (G64). Os resultados de tempo e avanço da ferramenta na trajetória gerada pela programação paramétrica demonstram superioridade de desempenho em relação aos demais métodos, conforme apresenta a Tabela 1:

Tabela 1 - Resultados Interpolação Helicoidal

Interpolação Helicoidal			
Programa	Avanço real (mm/min.)	Tempo (s)	Variação (%)
CAM Retas	3600	76	0
CAM Retas +G64	6700	34	-45
PARAMETRICA	8000	26	-66

Graficamente, os valores são apresentados na Figura 49.

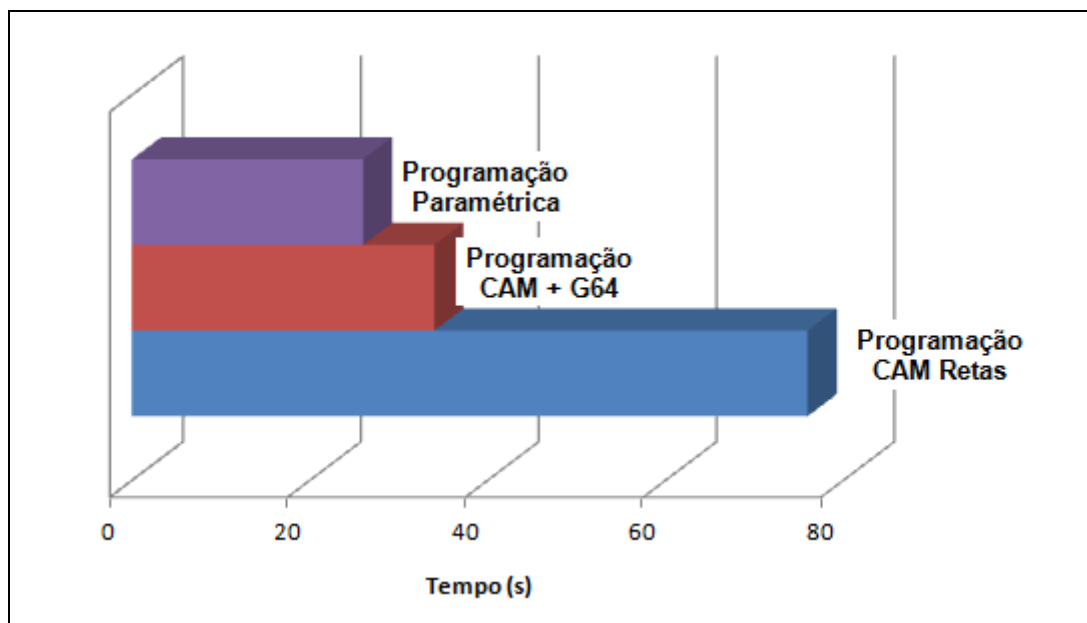


Figura 49 – Gráfico do desempenho das formas de programação.

4.2 Interpolação Circular

A interpolação circular, gerada em *software* CAM, foi executada, inicialmente, a partir do programa composto por blocos que formam a trajetória da ferramenta, através da ligação entre segmentos de reta no contorno do círculo programado. A geometria gerada é executada em função dos comandos que constam no programa e dependem dos dados de tolerância informados ao CAM, com a programação do contorno da peça a partir de uma superfície cilíndrica com altura definida, ou seja, um desenho em 3D. O CAM gera os segmentos de reta ligados entre si com variações de comprimento entre eles, que permanecem dentro da faixa de tolerância programada para a execução do contorno de 0,1 mm. A representação da influência da tolerância na trajetória da ferramenta é apresentada na Figura 50.



Figura 50 - Influência da Tolerância na trajetória da ferramenta.

O programa gerado mostrado na Figura 51 ocupa uma memória no controlador de 896 bytes e 51 linhas.

```
G17 G71 G90 G94
T01; END MILL D10.0mm R0.mm
S3000 M3
G54 D01
G0 X38. Y0.
G0 Z2.
G1 Z-10. F200
G1 X30. F8000
G1 X29.568 Y-5.07
G1 X28.286 Y-9.994
G1 X26.19 Y-14.631
G1 X23.341 Y-18.847
.....
G1 X28.286 Y9.994
G1 X29.568 Y5.07
G1 X30. Y0.
G1 X38.
G0 Z10.
M9
G53 G0 Z-110 D0 M5
M30
%
```

Figura 51 - Programação CAM Circular - Segmentos de retas.

O tamanho dos segmentos de retas gerados a partir desse método de programação foi determinado com a utilização de um *software* CAD, que demonstra graficamente parte da trajetória executada no programa CAM (Figura 52). Os pontos são traçados utilizando segmentos do início e do final do programa. O tamanho de cada segmento de reta gerado é de 5,088 mm com variação angular de 9,73°.

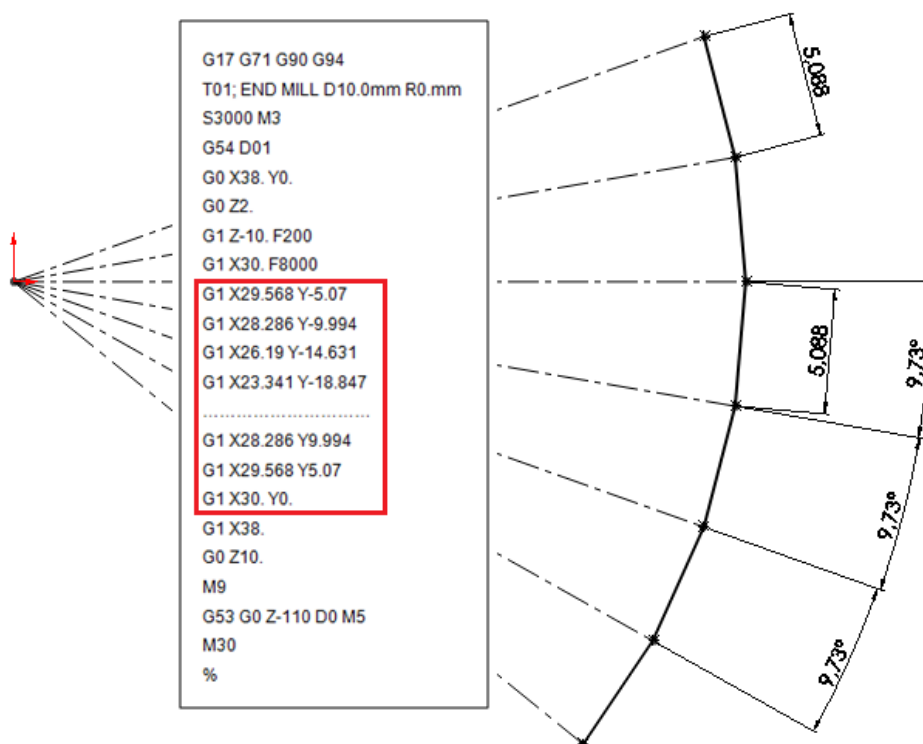


Figura 52 - Determinação do tamanho do segmento de reta utilizando programa CAD.

O tempo gasto na execução do programa para a trajetória circular, composta por blocos de comandos que executam a trajetória do círculo por ligação de pequenas retas através do seu perímetro, foi de 12 s (Figura 53) com uma velocidade de avanço de 5700 mm/min, para a velocidade programada de 8000 mm/min, com vibração da máquina e de seus componentes.

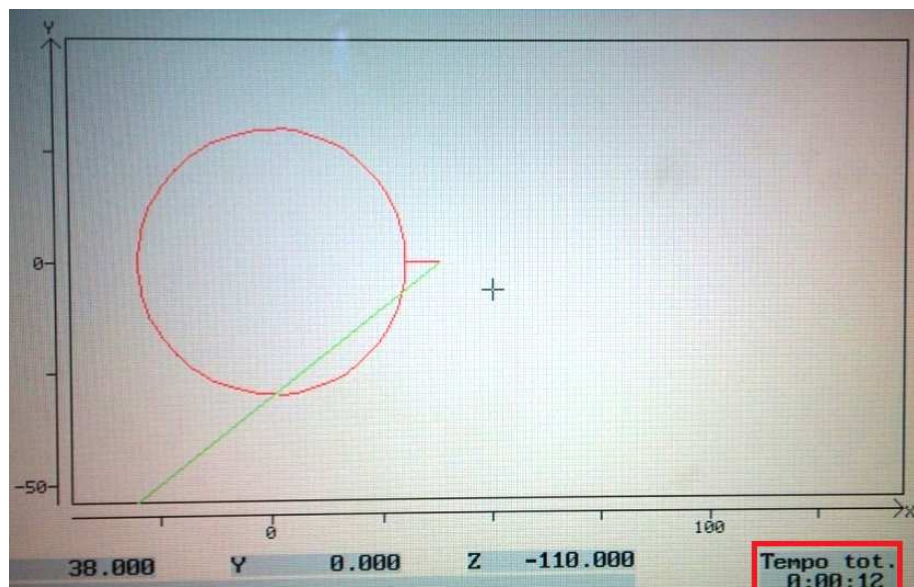


Figura 53 - Tempo de execução – Programação Segmentos de retas.

A utilização do mesmo programa, porém com a adição do comando de leitura de blocos à frente da execução (G64), reduziu a vibração da máquina em relação à condição inicial. O tempo de execução do programa também foi reduzido, passando de 12 s para 8 s com velocidade de avanço dos eixos (X) e (Y) igual à velocidade programada (8000 mm/min).

O programa CAM, ao utilizar o bloco de comando com interpolação circular (G2), executa a trajetória da ferramenta através da movimentação simultânea dos eixos (X) e (Y) durante o contorno do círculo. O processo não apresentou vibrações da máquina por conta da aceleração e desaceleração entre os pontos de ligação, já que esses não existem ao longo do perímetro para o método aplicado. A tolerância dimensional informada no CAM para esta formação é de 0,01 mm. A interpolação circular gerada pelo programa CAM (Figura 54), através de comandos de trajetória circular (G2) e (G3), gera um programa de usinagem com 15 linhas e 128 bytes.

```
G17 G71 G90 G94
T01;END MILL D10.0mm R0.mm
S3000 M3
G54 D01
G0 X38. Y0.
G0 Z2.
G1 Z-10. F200
G1 X30. F8000
G2 X30. Y0. I-30. J0.
G1 X38.
G0 Z10.
M9
G53 G0 Z-110 D0 M5
M30
%
```

Figura 54 - Programação CAM Circular - Segmentos de arcos.

A velocidade de avanço de execução foi de 8000 mm/min e o tempo utilizado foi de 7 segundos (Figura 55).

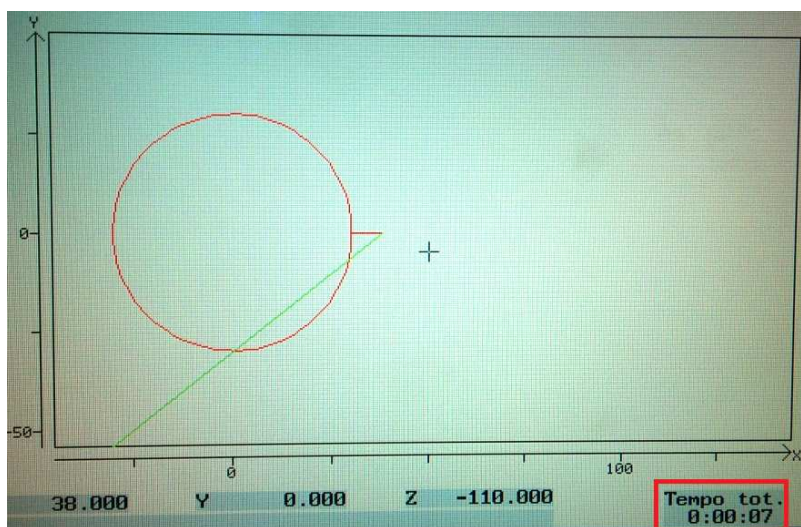


Figura 55 - Programação CAM Circular Arcos – Tempo de execução.

A programação paramétrica não sofreu alterações em sua velocidade e tempo de execução em função da aplicação do comando (G64), pois o comando atende o princípio de leitura de blocos a frente do bloco de execução atual.

A Figura 56 apresenta o conceito da programação paramétrica para a execução da trajetória circular, através do posicionamento da ferramenta até o ponto inicial em (Z), da aproximação até o raio do círculo, compensando o raio da ferramenta, e da execução da trajetória em segmento de reta, determinado pela variação angular de incremento e pelo cálculo trigonométrico com as variáveis (R1) e (R3) inseridas. A cada conclusão da trajetória, realiza-se o incremento da variação angular para definição de nova posição e compara-se a soma do ângulo com o ângulo final programado; se este for maior, o programa segue para o posicionamento (X) e (Y) inicial retornando à posição (Z) de partida.

```

G0 X=((R1+R7)*COS(R3)) Y=((R1+R7)*SIN(R3))
Z2
G01 Z-(R9) F150
V → INICIO: G01 X=(R1*COS(R3)) Y=(R1*SIN(R3)) F=R10
R3=R3+R5 → INCREMENTO ANGULAR 0.1°
IF R4>R3 GOTOB INICIO → F
G01 X=(R1*COS(R4)) Y=(R1*SIN(R4))
X=((R1+R7)*COS(R4)) Y=((R1+R7)*SIN(R4))
G53 G0 Z-110 D0
M30;

```

Figura 56 - Forma de execução do programa pelo controlador NC.

O programa paramétrico possui um tamanho de 427 bytes e 28 linhas e está descrito na Figura 57.

```

(%_N_TESTEPARAMETRICO_MPF
;(INTERPOLAÇÃO CIRCULAR PARAMETRICA – SEGMENTOS DE RETA)
G17 G64 G71 G90 G94
G53 G0 Z-110 D0
T1
M6
G54 D1
S2500 M3
R1=25           ; raio do círculo
R20=10          ; diâmetro da ferramenta
R3=0            ; ângulo início do raio/arco
R4=360          ; ângulo do final do raio/arco
R5=0.1          ; incremento angular/tolerância angular
R6=R20/2        ; calculo do raio da ferramenta
R7=3            ; distancia de aproximação da ferramenta
R9=1            ; profundidade de corte
R10=8000        ; velocidade de avanço em mm/min
R1=R1+R6        ; cálculo do raio do arco em função do raio da ferramenta
G0 X=((R1+R7)*COS(R3)) Y=((R1+R7)*SIN(R3))
Z2
G01 Z-(R9) F150
INICIO: G01 X=(R1*COS(R3)) Y=(R1*SIN(R3)) F=R10
R3=R3+R5
IF R4>R3 GOTOB INICIO
G01 X=(R1*COS(R4)) Y=(R1*SIN(R4))
X=((R1+R7)*COS(R4)) Y=((R1+R7)*SIN(R4))
G53 G0 Z-110 D0
M30;

```

Figura 57 - Programa paramétrico circular.

O tamanho dos segmentos de reta gerados no perímetro da circunferência de raio 25 mm para os dois ângulos. Com a divisão angular em intervalos de $0,1^\circ$, os segmentos de reta apresentam tamanho de 0,044 mm e, com 1° de variação entre os pontos programados, os segmentos possuem 0,393 mm de comprimento.

O desempenho do programa paramétrico, com incremento angular de $0,1^\circ$ para determinação do comprimento da reta que compõe o perímetro do contorno, foi avaliado em função do tempo de execução de 43 s, sem vibrações na máquina, com a redução da velocidade programada para 282 mm/min (Figura 58), traçando a trajetória com suavidade de movimentação entre os eixos (X, Y).

MCS	Posição [mm]	PercRes	T,F,S
X1	-204.326	0.022	T 1 D1 ø 10.000
Y1	-278.574	-0.040	F 282.7 100% 8000. mm/min
Z1	-239.300	0.000	S 2515. 100% 2500. I
P.zero1			0% 80% 100%
Bloco atual INTERPARAM50.MPF			
G0 X=((R1+R7)*COS(R3)) Y=((R1+R7)*SIN(R3)) ↑			
Z2 ↑			
G01 Z=-(R9) F8000 ↑			
INICIO: G01 X=(R1*COS(R3)) Y=(R1*SIN(R3)) F=R10 ↑			
R3=R3+R5 ↑			
IF R4>R3 GOTOB INICIO ↑			
G01 X=(R1*COS(R4)) Y=(R1*SIN(R4)) ↑			

Figura 58 - Velocidade de execução - Programação paramétrica.

Apesar de a ferramenta percorrer a trajetória de forma satisfatória, o controlador, embora apresentasse redução de velocidade, não efetuou a reprodução gráfica em tela de forma eficiente, traçando apenas a ligação entre alguns pontos (Figura 59).

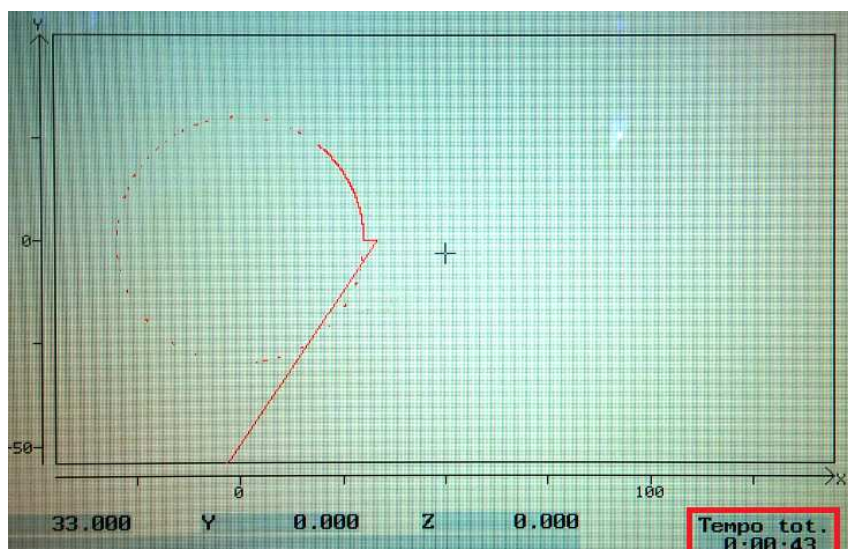


Figura 59 - Tempo de execução - Programação paramétrica.

Após da alteração do incremento angular do programa paramétrico para 1º, o tempo de execução do programa foi reduzido para 7s com velocidade de avanço

igual à velocidade programada. Esse resultado pode ser comparado ao desempenho obtido com a programação CAM, através da geração da trajetória da ferramenta por um único bloco contendo comando de trajetória circular (G2). Os resultados iguais indicam que o programa paramétrico com incremento angular de 1° gera o mesmo comprimento de reta no perímetro do contorno que o programa gerado em CAM; porém, o programa CAM, que contém a formação de arco (G2) em sua composição, evita o facetamento da superfície, influenciando no acabamento final da peça.

A comparação do desempenho (Tabela 2), para a geração da trajetória circular na aplicação de diferentes métodos de programação, apresentou grande variação no que se refere ao tempo de execução e velocidade de avanço para a programação paramétrica, utilizando baixo incremento angular ($0,1^\circ$). Esse resultado ocorre devido à limitação da capacidade de processamento do controlador NC, que, ao executar o movimento para o pequeno segmento de reta gerado (0,044 mm) em velocidade programada, depende do cálculo para o novo movimento, de modo que o controlador reduz a velocidade de execução a espera do processamento do programa para seqüenciar a trajetória. Com o aumento do incremento angular para 1° , as linhas retas que formam o contorno da trajetória circular ficam maiores, sendo que o tempo para percorrer esses segmentos é maior que o tempo utilizado pelo controlador para o processamento dos cálculos e o programa é executado com desempenho equivalente ao programa CAM composto por arcos (G2).

Tabela 2 - Resultados Interpolação Circular

Interpolação Circular			
Programa	Avanço real (mm/min.)	Tempo (s)	Variação (%)
CAM Retas	5700	12	0
CAM Retas +G64	8000	8	-33
CAM Arcos	8000	7	-41
PARAMÉTRICA $0,1^\circ$	282	43	258
PARAMÉTRICA 1°	8000	7	-41

O desempenho da interpolação circular em função do tempo é apresentado graficamente na Figura 60:

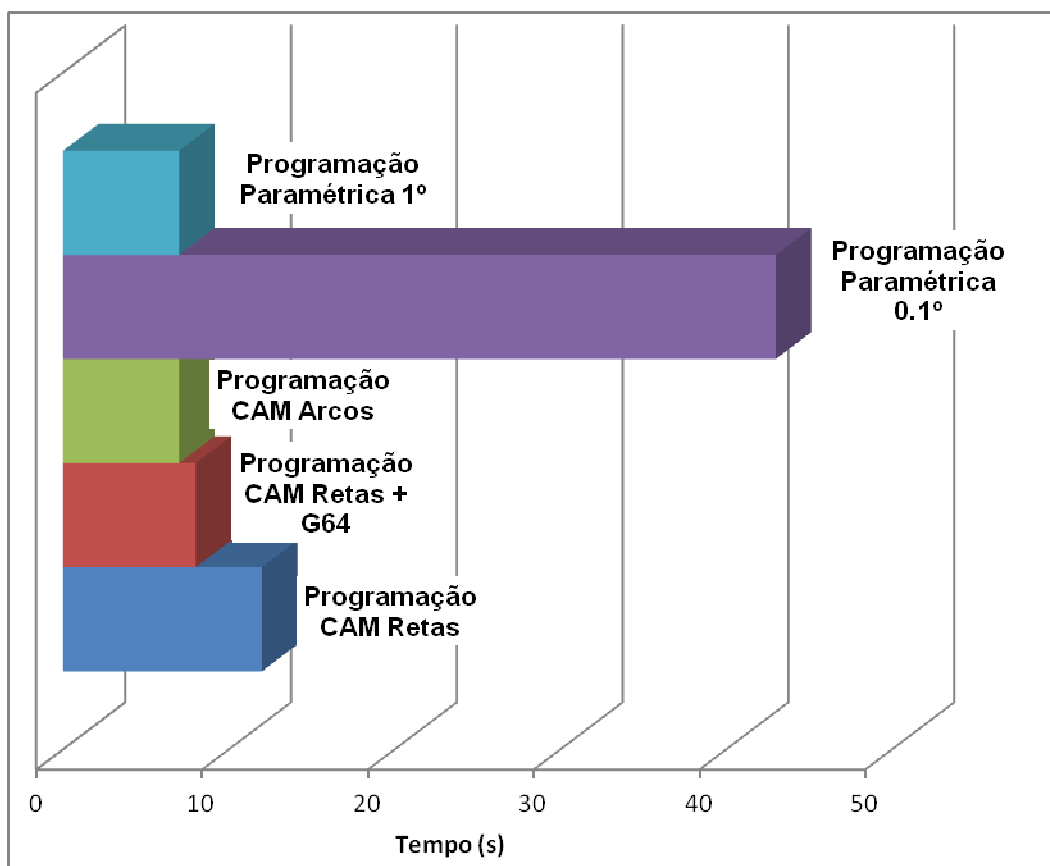


Figura 60 – Gráfico do desempenho para interpolação circular.

4.3 Elipse

A geração da trajetória elíptica para valores de geometria de 80 mm no eixo (X) e 50 mm no eixo (Y) foi avaliada em função do tempo de execução dos programas, bem como suas velocidades iniciais a partir do programa gerado em CAM, sem a utilização de recurso de aumento de desempenho de leitura oferecido pelo controlador da máquina CNC.

A figura 61 apresenta o programa CN gerado por blocos que executam movimentos lineares, com variação de comprimento limitada a manter a tolerância dimensional de 0,01 mm, definida no momento da programação no CAM.

Os blocos de programação contêm comandos especificamente formados por trajetórias lineares (G1), que formam a geometria do perímetro da circunferência. A divisão do perímetro pelo CAM em diversos segmentos faz com que o número de

linhas do programa seja maior se comparado ao programa formado por arcos apenas.

A memória ocupada do controlador pelo programa CN é de 10,5 kbytes, contendo 526 linhas. Parte do programa é apresentado na Figura 61.

```
G17 G71 G90 G94
T01; END MILL D10.0mm R0.mm
S3000 M3
G54 D01
G0 X53. Y0.004
G0 Z-8.
G1 Z-10. F200
G1 X45. Y0.002 F8000
G1 Y-0.004
G1 Y-0.045
.....
G1 X45. Y0.002
G1 X53. Y0.004
G0 Z10.
M9
G53 G0 Z-110 D0 M5
M30
%
```

Figura 61 - Programação CAM elipse – Segmentos de retas.

Durante a leitura e execução do programa CN, da forma que foi constituída pelo *software* CAM, a máquina apresentou vibração excessiva e limitação de avanço em relação ao avanço máximo programado. O avanço atingido teve dificuldades em ser coletado e a mudança dos blocos do programa para leitura em função dos pequenos segmentos de reta de percurso da trajetória fez com que a taxa de atualização dos valores no painel do controlador fosse inferior à velocidade de execução. Em alguns pontos da trajetória, o controlador apresentou valor de avanço próximo a 330 mm/min para o avanço máximo programado de 8000 mm/min, sendo que o tempo total para o percurso foi de 46 s (Figura 62).

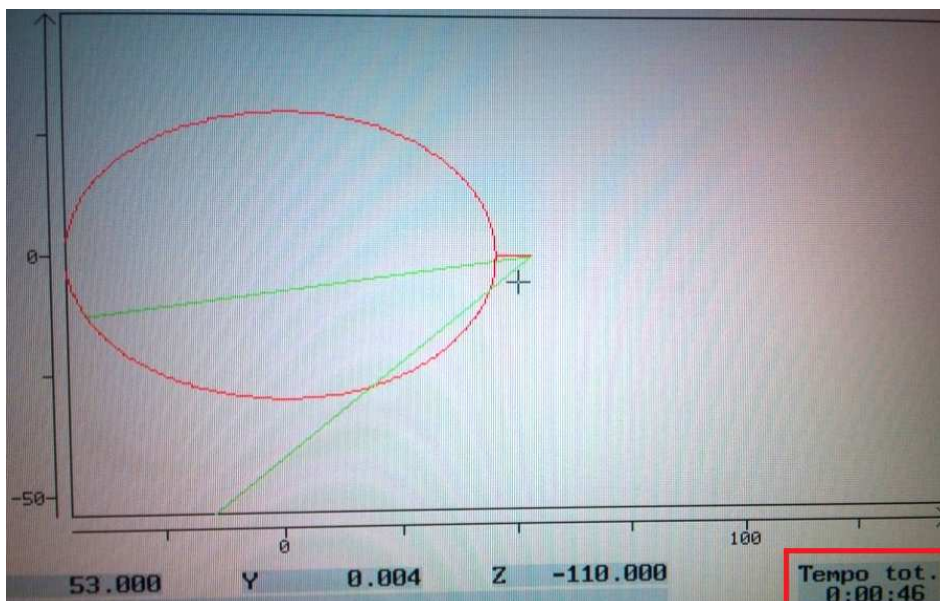


Figura 62 - Tempo de execução – Programa elipse segmentos de reta.

A adição ao programa do comando (G64) aumentou consideravelmente a velocidade de avanço para 3022 mm/min, indicada no painel do controlador. Mesmo aplicando essa função que fez com que o controlador obtivesse as informações de até 100 blocos a frente da leitura atual, a vibração não foi amenizada. O tempo de execução foi de 17 s, conforme a Figura 63.

Apesar da velocidade de avanço ter sido aumentada em uma escala de 10 vezes, ela não se manteve constante durante a execução da trajetória helicoidal, o que é comprovado pelas diferenças de tempo para os métodos, que não foi reduzida na mesma proporção. Esse fato ocorre devido à atualização dos dados pelo controlador em tela e ao tamanho dos segmentos de reta que compõe o programa, uma vez que o tempo para a informação chegar à tela é maior que a atualização do próximo bloco a ser executado pelo controlador.

Outro fato observado é que, ao utilizar a programação por segmentos de reta, a cada ponto programado, o controlador fez com que a máquina tendesse a fazer a parada exata a cada posição, com aceleração e desaceleração entre os pontos.

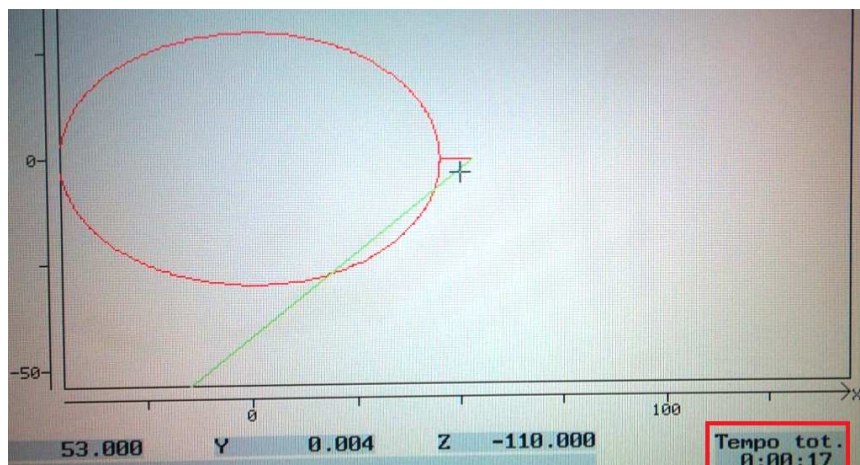


Figura 63 - Tempo de execução – Programa elipse segmentos de reta + G64.

O programa CAM contendo interpolações circulares (Figura 64) ocupa um espaço de 1,97 kbytes e 73 linhas.

```

G17 G71 G90 G94
T01; END MILL D10.0mm R0.mm
S3000 M3
G54 D01
G0 X52.92 Y-2.15
G0 Z2.
G1 Z-10. F200
G1 X44.942 Y-1.555 F8000
G2 X44.076 Y-6.137 I-20.745 J1.549
.....
G1 X24.883 Y25.088
G2 X30.248 Y22.337 I-19.179 J-44.024
G1 X30.262 Y22.329
G2 X34.587 Y19.341 I-20.861 J-34.819
.....
G1 X44.742 Y3.257
G2 X45. Y0.002 I-20.51 J-3.263
G2 X44.942 Y-1.555 I-20.803 J-0.007
G1 X52.92 Y-2.15
G0 Z10.
M9
G53 G0 Z-110 D0 M5
M30
%
```

Figura 64 - Programa CAM Elipse – Arcos.

O valor de velocidade de avanço alcançada pela máquina durante o percurso da trajetória foi de 4196 mm/min, contando com uma significativa redução da vibração, sem sua anulação por conta dos segmentos de reta que constam no programa. A Figura 65 demonstra a execução e os dados obtidos no painel do controlador.

MCS	Posição [mm]	PercRes	T,F,S
X1	-213.635	0.000	T 1 Ø 10.000
Y1	-171.580	0.000	F 4196. 100% 8000. mm/min
Z1	-205.000	0.000	S 3019. 100% 3000. I

P.zero1

Bloco atual ELIPSECIRCULAR.MPF

G1 X18.424 Y27.422 ↑
 G2 X24.871 Y25.093 I-15.692 J-53.505 ↑
 G1 X24.883 Y25.088 ↑
 G2 X30.248 Y22.337 I-19.179 J-44.024 ↑
 G1 X30.262 Y22.329 ↑
 G2 X34.587 Y19.341 I-20.861 J-34.819 ↑
 G1 X34.599 Y19.332 ↑

Figura 65 - Velocidade de avanço Elipse programação CAM – Arcos.

A Figura 66 demonstra a tela do controlador após a execução do programa composto por arcos (G2). A tela apresenta graficamente a trajetória da ferramenta, bem como o tempo total de 13 s para a formação da elipse.

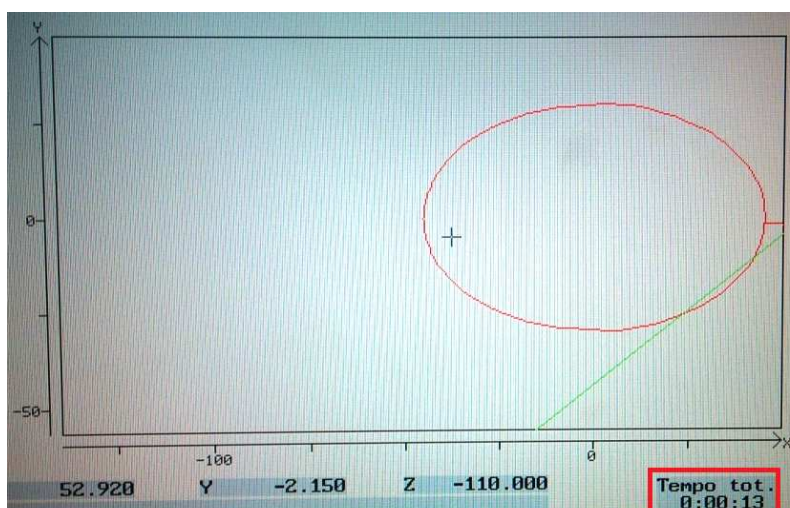


Figura 66 - Tempo de execução Elipse programação CAM – Arcos.

A figura 67 apresenta o programa paramétrico para a obtenção da geometria elíptica. Inicia-se pela aproximação dos eixos até uma altura próxima a peça, após faz-se a aproximação da ferramenta até o início do raio da interpolação, no ângulo inicial definido como 0° e no raio maior da elipse no eixo (X). Então, o eixo (Z) é deslocado até a posição definida como altura do contorno no programa. A trajetória da ferramenta ocorre por um deslocamento retilíneo, com o incremento angular indicado na variável (R5) do programa paramétrico baseado nas variáveis de dimensão da elipse (R1) e (R2).

A programação soma o incremento angular ao ângulo atual e compara com o ângulo final (R4). Se o ângulo calculado for maior que o ângulo final, o programa segue para sua continuação até o reposicionamento ao ponto inicial; caso contrário, ele retorna à execução do novo segmento de reta em torno do perímetro da geometria, baseado no ângulo definido.

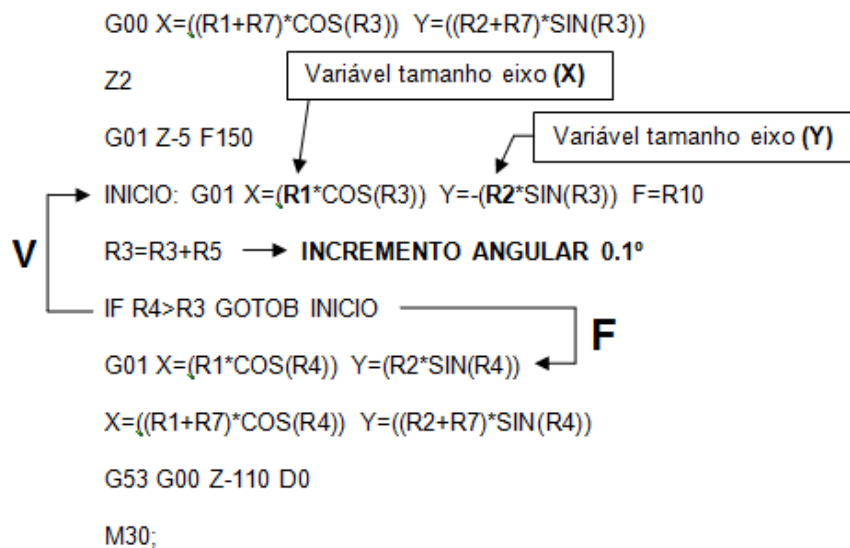


Figura 67 - Forma de execução do programa pelo controlador NC.

A memória ocupada pelo programa paramétrico (Figura 68) é de 628 bytes e 27 linhas.

```

%_N_ELIPSEPARAM_MPF
; $PATH=/_N_WKS_DIR/_N_TESTE_WPD
G17 G64 G71 G90 G94
T1
M6
G54 D1
S3500 M3
R1=80           ; maior comprimento
R2=50           ; menor comprimento
R20=10          ; diâmetro da ferramenta
R1=((R1+R20)/2)
R2=((R2+R20)/2) ; definição do raio dos eixos
R3=0            ; ângulo inicial
R4=360          ; ângulo final
R5=0.1          ; incremento angular
R7=3            ; distancia de aproximação
R10=8000        ; velocidade de avanço
G00 X=((R1+R7)*COS(R3)) Y=((R2+R7)*SIN(R3))
Z2
G01 Z-5 F200
INICIO: G01 X=(R1*COS(R3)) Y=-(R2*SIN(R3)) F=R10
R3=R3+R5
IF R4>R3 GOTOB INICIO
G01 X=(R1*COS(R4)) Y=(R2*SIN(R4))
X=((R1+R7)*COS(R4)) Y=((R2+R7)*SIN(R4))
G53 G00 Z0 D0
M30;

```

Figura 68 - Memória ocupada pelo programa paramétrico Elipse.

O tempo necessário para a ferramenta percorrer o contorno da elipse, utilizando incrementos angulares de $0,1^\circ$ para ligação entre os pontos da trajetória, foi de 45 s com um avanço reduzido pelo controlador de 8000 mm/min (programado) para cerca de 332 mm/min. A redução do avanço foi realizada pelo controlador com o objetivo de movimentação dos eixos de forma contínua, sem variações de aceleração e desaceleração entre os pontos calculados no programa, o que gerou movimentação suave com ausência de vibrações, refletindo, naturalmente, no acabamento superficial de uma peça durante a usinagem.

A Figura 69 apresenta a informação do tempo total na tela do controlador CN. Nota-se que o controlador não reproduziu graficamente em tela o caminho da ferramenta devido à sua velocidade de processamento dos cálculos e repetição dos

ciclos do programa paramétrico, o que fez com que a máquina utilizasse o mesmo procedimento de geração da trajetória para a execução e para a demonstração em tela. A taxa de avanço coletada permaneceu constante na tela do controlador durante a experimentação utilizando o programa paramétrico.

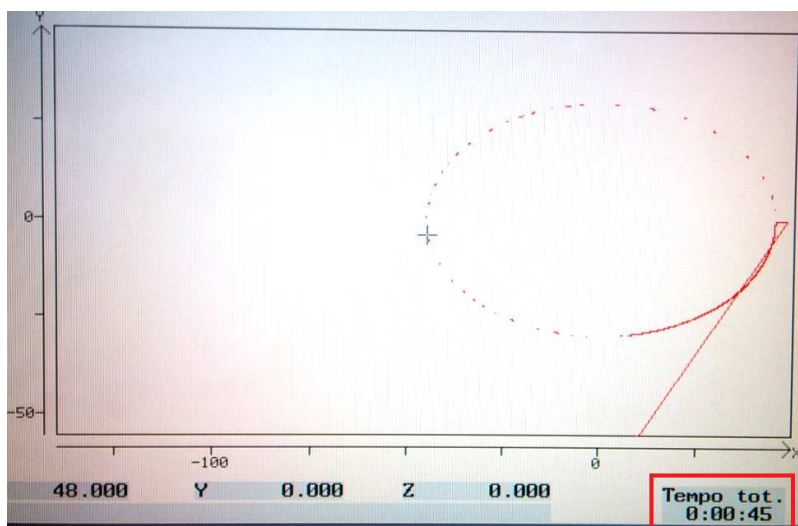


Figura 69 - Tempo de execução programa paramétrico Elipse.

Posteriormente à experimentação, utilizando como incremento angular $0,1^\circ$, realizou-se a alteração do dado de entrada do programa paramétrico para 1° de incremento angular. Dessa forma, o programa foi executado em um tempo de 9 s com um avanço constante de 6300 mm/min e sem vibrações na máquina CNC, sendo, no entanto, ainda inferior ao avanço programado.

A Tabela 3 apresenta os dados de desempenho para comparação entre as formas de programação para formação da trajetória da elipse.

Tabela 3 – Resultados Elipse

Elipse			
Programa	Avanço real (mm/min.)	Tempo (s)	Variação (%)
CAM Retas	330	46	0
CAM Retas +G64	3022	17	-63
CAM Arcos	4196	13	-72
PARAMÉTRICA 0.1°	332	45	-2
PARAMÉTRICA 1°	6300	9	-80

No gráfico da Figura 70, observa-se que a programação paramétrica com incremento angular de $0,1^\circ$ e a programação CAM por segmentos de retas possuíram desempenho equivalente em função do tempo, o que significa que a taxa de avanço, que não foi devidamente atualizada no painel da máquina na programação CAM pelo controlador, pode ser comparada com o avanço constante apresentado na programação paramétrica. O resultado de menor tempo para a programação paramétrica por incrementos angulares de 1° aumenta a distância entre os pontos a serem traçados pelo controlador no perímetro da elipse. Dessa forma, a máquina ainda não atingiu a velocidade de avanço de 8000 mm/min programada. Para que tal resultado seja obtido, é necessário o aumento do incremento angular, o que reduz o acabamento superficial e gera o facetamento da superfície de contorno.

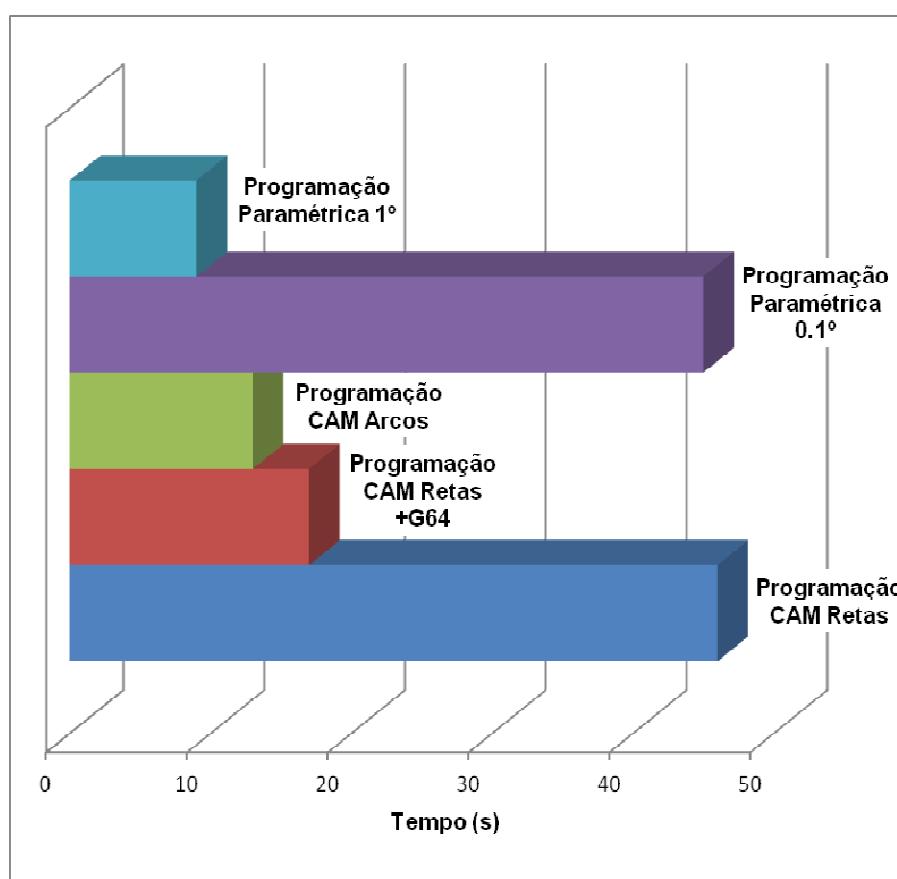


Figura 70 - Comparação de desempenho para elipse.

4.4 Cavidade Circular

A programação da cavidade circular em CAM, por tratar-se de uma geometria construída em CAD, teve de seguir os mesmos princípios da programação paramétrica para que a comparação dos resultados fosse possível.

O incremento de corte lateral e a profundidade, a cada passe, foram os fatores a serem observados durante o preenchimento das informações tecnológicas no CAM.

A primeira experimentação foi do programa composto pelo desbaste da cavidade circular através de blocos de programação por segmentos de reta, utilizando uma tolerância dimensional de 0,01 mm na superfície da geometria CAD, com avanço de 8000 mm/min programado e incremento de corte lateral de 70% em relação ao diâmetro da ferramenta. A rotação da ferramenta não gera influência sobre o tempo e velocidade de avanço para o percurso das trajetórias. A Figura 71 mostra as linhas de trajetória da ferramenta gerada pelo CAM na geometria tridimensional.

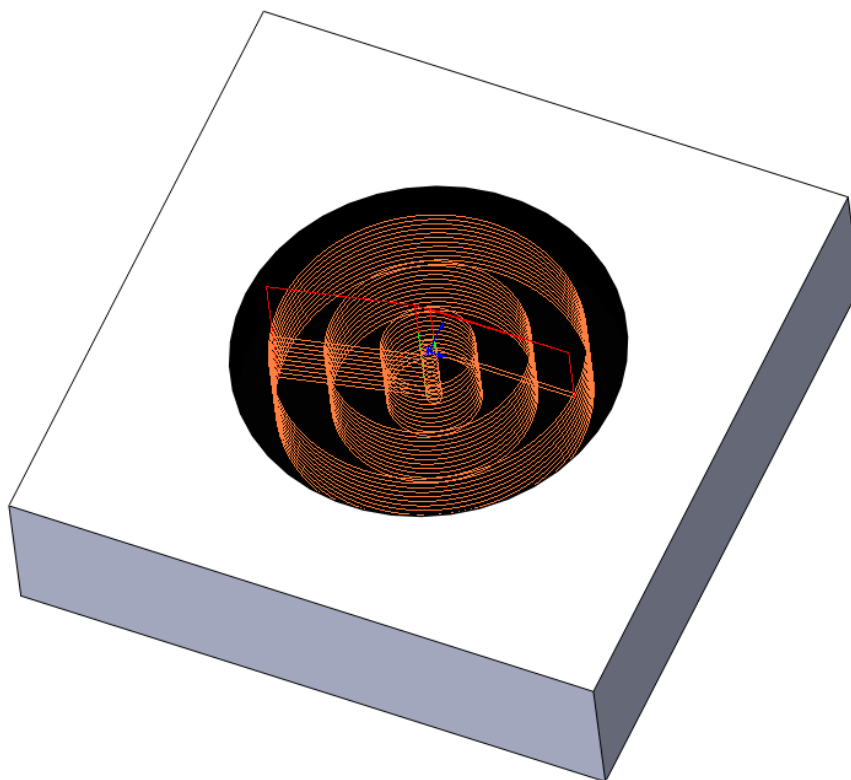


Figura 71 - Cavidade Circular Tridimensional CAM.

O programa NC constituído de segmentos de reta (Figura 72) possui 24,6 kbytes e aproximadamente 1262 linhas.

```
G17 G71 G90 G94
T01; END MILL D10.0mm R0.mm
S1000 M3
G54 D01
G0 X0.193 Y-0.973
G0 Z2.
G1 Z-1. F200
G3 X0.193 Y-0.973 I-0.193 J0.974 F8000
G1 X1.167 Y-5.877
G3 X1.167 Y-5.877 I-1.167 J5.878
G1 X2.53 Y-12.743
G3 X2.53 Y-12.743 I-2.53 J12.744
G1 X3.89 Y-19.608
G1 X4.446 Y-19.497
G1 X5.556 Y-19.21
G1 X6.633 Y-18.865
G1 X7.707 Y-18.452
.....
G1 X2.222 Y-19.874
G1 X3.333 Y-19.718
G1 X3.89 Y-19.608
G0 Z10.
M9
G53 G0 Z-110 D0 M5
M30
%
```

Figura 72 - Programação CAM Cavidade Circular.

O tempo para a execução das trajetórias pela máquina CNC foi de 188 s, sendo que ocorreu grande vibração da máquina e não foi possível a coleta de dados de velocidade de avanço em tela do controlador, devido às variações de aceleração e desaceleração entre os pontos que não permitiram em algum momento uma velocidade constante. A Figura 73 apresenta em destaque a forma em que é exibida a velocidade de avanço pelo controlador no momento da execução do programa, bem como a velocidade programada entre outros dados tecnológicos.

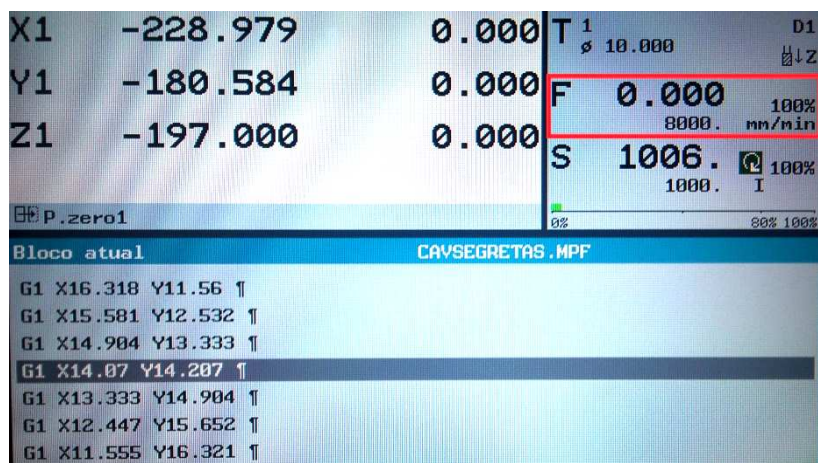


Figura 73 - Velocidade de avanço em tela – Cavidade Circular.

A utilização do recurso do controlador para a suavização da execução no que se refere às acelerações e desacelerações entre os pontos, resultou em uma diferença de desempenho considerável para o programa da cavidade circular. Por tratar-se do maior programa experimentado em tamanho e tempo de execução, os valores de tempo caíram substancialmente de 188 s para apenas 69 s. Com base nesses dados, se o comando (G64) fosse aplicado a esse tipo de programação, com uma cavidade em aço em maiores dimensões, como um molde de injeção ou modelo de fundição, por exemplo, o tempo total do processo de usinagem poderia ser reduzido em algumas horas.

A Figura 74 demonstra o valor obtido na tela do controlador CNC. Esse programa ainda gerou vibrações durante a execução e a velocidade de avanço não ultrapassou 1360 mm/min.

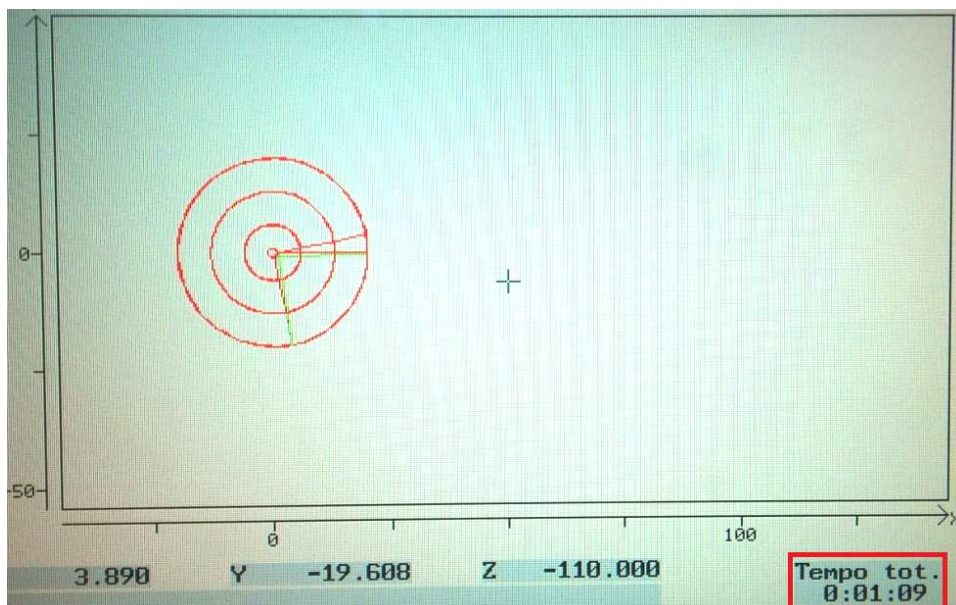


Figura 74 - Tempo de execução cavidade circular – segmentos de reta.

A programação gerada em CAM, configurada para a criação de blocos com comandos de arcos especificamente para a trajetória de desbaste circular, reduziu o tamanho do programa de 1262 linhas para 143 linhas, compostas principalmente por blocos com comandos de trajetória (G3).

O programa apresentado (Figura 75) permitiu, assim, um tamanho total de ocupação de memória do controlador da máquina CNC de 2,96 kbytes.

```

G17 G71 G90 G94
T01;END MILL D10.0mm R0.mm
S1000 M3
G54 D01
G0 X-0.972 Y-0.209
G0 Z2.
G1 Z-1. F200
G3 X-0.972 Y-0.209 I0.973 J0.209 F8000
G1 X-5.861 Y-1.257
.....
G3 X-19.55 Y-4.192 I19.55 J4.192
G0 Z10.
G0 X-0.972 Y-0.209
G0 Z2.
G1 Z-7. F200
G3 X-0.972 Y-0.209 I0.973 J0.209 F8000
G1 X-5.861 Y-1.257
G3 X-5.861 Y-1.257 I5.861 J1.257
G1 X-12.706 Y-2.725
G3 X-12.706 Y-2.725 I12.706 J2.724
G1 X-19.55 Y-4.192
G3 X-19.55 Y-4.192 I19.55 J4.192
G0 Z10.
M9
G53 G0 Z-110 D0 M5
M30
%
```

Figura 75 - Programa CAM Cavidade Circular – Arcos.

A utilização dos comandos de interpolação circular reduziu o tempo de experimentação do programa para 65 s (Figura 76) com ausência de vibração de máquina durante o trajeto da ferramenta. Apesar da redução ser baixa em relação ao teste anterior, o fato de ausentar a máquina de vibrações influencia diretamente na qualidade final da peça usinada. A velocidade de execução no momento da execução do programa apresentada pelo controlador foi de 3930 mm/min.

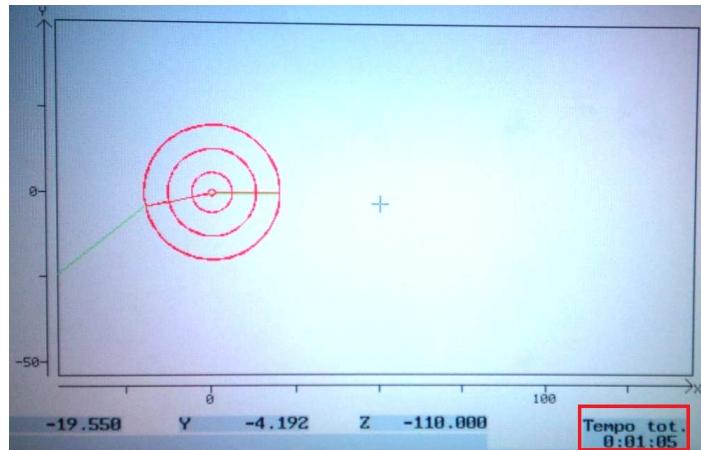


Figura 76 - Tempo de execução cavidade circular arcos.

A Figura 77 apresenta as condições estabelecidas no programa paramétrico para a geração da trajetória da ferramenta.

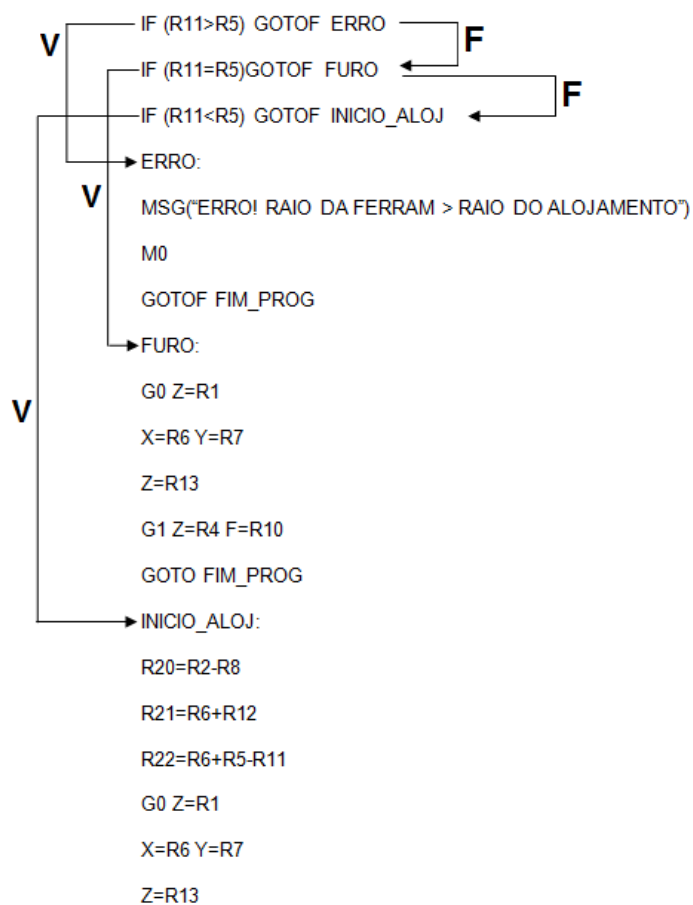


Figura 77- Forma de execução do programa pelo controlador NC.

O programa paramétrico é composto por 71 linhas e ocupa uma memória total do controlador de 1.02 kbytes, representado de forma fragmentada na Figura 78.

```
(%_N_TESTEPARAMETRICO_MPF
; (CAVIDADE CIRCULAR – INTERPOLAÇÃO CIRCULAR)
R1=5 ; plano de retorno em "Z"
R2=0 ; plano de referencia em "Z"
R3=2 ; distancia de segurança em "Z"
R4=-10 ; altura da cavidade
R5=25 ; raio da cavidade
R6=0 ; coordenada do centro da cavidade em X
R7=0 ; coordenada do centro da cavidade em Y
R8=1 ; profundidade de corte
R9=8000 ; avanço em X (mm/min.)
R10=200 ; avanço em Z (mm/min.)
R11=5 ; raio da ferramenta
R12=7 ; incremento de corte lateral (AE)
R13=R2+R3 ; calculo do "Z" inicial
IF (R11>R5) GOTOF ERRO
IF (R11=R5)GOTOF FURO
IF (R11<R5) GOTOF INICIO_ALOJ
ERRO:
MSG ("ERRO! RAI0 DA FERRAM > RAI0 DO ALOJAMENTO")
M0
GOTOF FIM_PROG
FURO:
G0 Z=R1
X=R6 Y=R7
Z=R13
G1 Z=R4 F=R10
GOTO FIM_PROG
INICIO_ALOJ:
R20=R2-R8
R21=R6+R12
.....
REPEAT ALARG FIM_ALARG
FIM_PROG:
G0 Z=R1
G53 G0 Z-110 D0
M30
```

Figura 78 – Fragmentos do Programa Paramétrico Cavidade Circular.

O tempo utilizado para a experimentação foi de 32 s sem vibrações na máquina e com velocidade de avanço constante de 8000 mm/min. A Figura 79 demonstra a tela do controlador com os dados referentes ao avanço programado e obtido durante a execução do programa, bem como a linha de comando que é executada no momento.

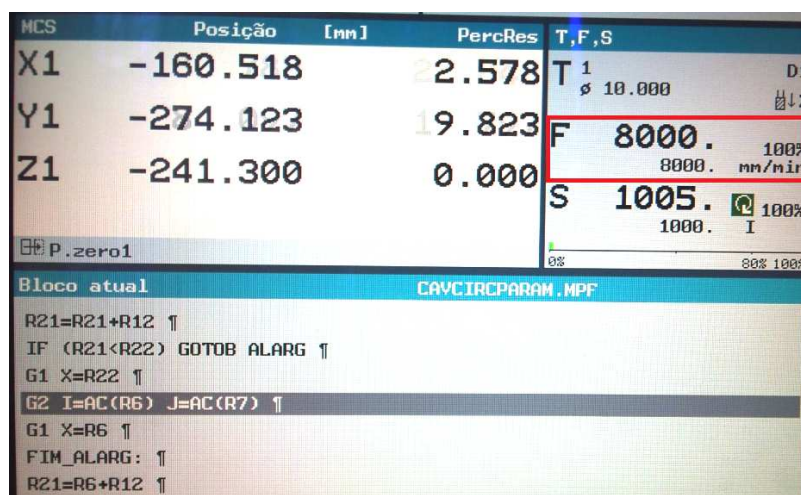


Figura 79 - Avanço Cavidade Circular Paramétrico.

Através da Tabela 4, foi realizada a comparação do desempenho em função da velocidade alcançada e o tempo para a execução das trajetórias circulares de formação da cavidade.

Tabela 4 – Resultados Cavidade Circular

Cavidade Circular			
Programa	Avanço real (mm/min.)	Tempo (s)	Variação (%)
CAM Retas	1360	188	0
CAM Retas +G64	3710	69	-63
CAM Arcos	3930	65	-72
PARAMETRICA	8000	32	-2

Considerando-se a tabela acima, o gráfico da Figura 80 pode ser avaliado em função do melhor desempenho obtido pela programação paramétrica na geração

das trajetórias, através do abastecimento das variáveis do programa pelo usuário e dos cálculos realizados pelo controlador da máquina CNC. Em função do tamanho do programa e da forma geométrica construída pela movimentação da ferramenta de diâmetro de 10 mm, pode-se avaliar o desempenho da programação paramétrica sendo 5,8 vezes superior em relação à programação CAM em sua forma básica inicial.

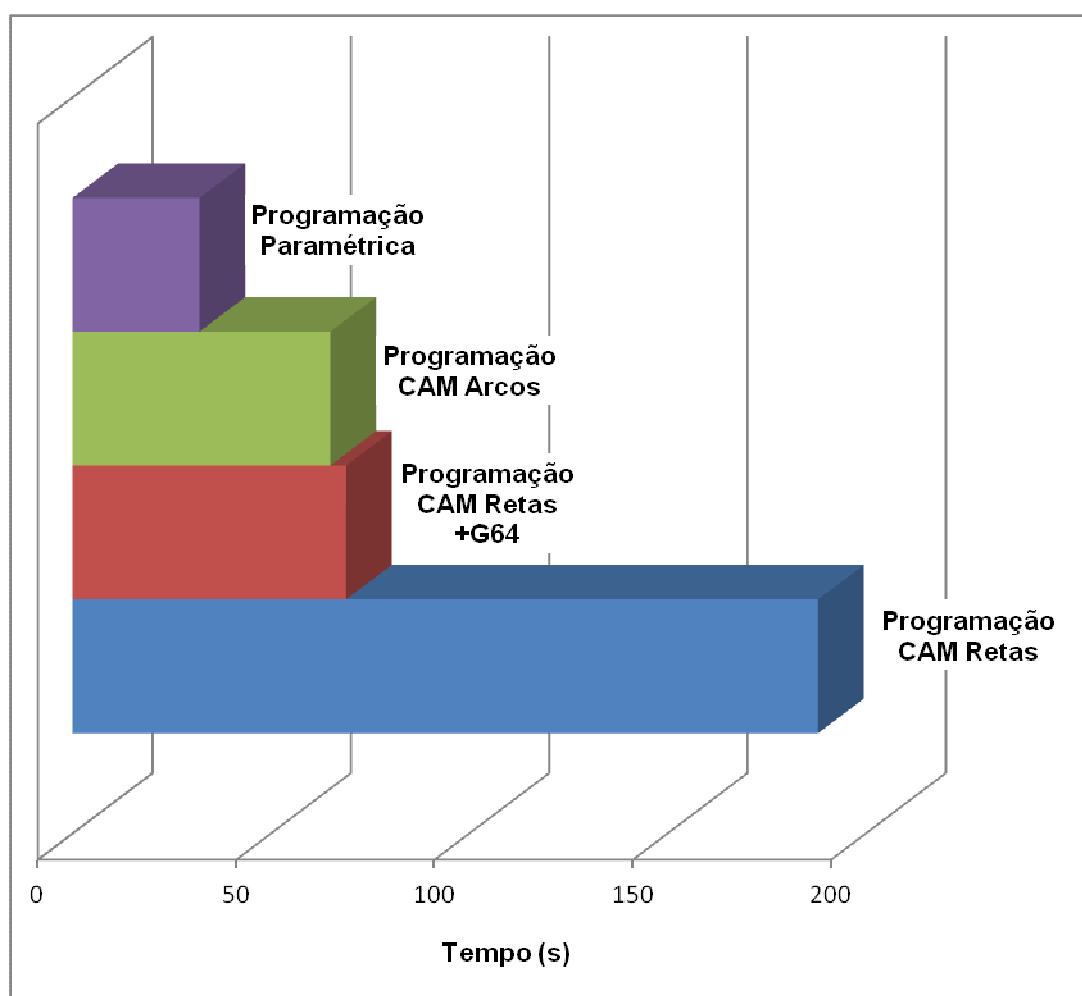


Figura 80 - Comparação de desempenho Cavidade Circular.

Finalmente, através da Tabela 5, são comparados os tamanhos dos programas em kilobytes. A comparação é realizada entre cada método de programação. A geometria que gerou mais disparidade entre o tamanho do programa foi a interpolação helicoidal, em que a programação CAM utilizou um espaço 426 vezes maior que a programação paramétrica.

A programação circular apresentou o menor programa na programação CAM por arcos (G2), com um tamanho próximo a programação paramétrica para a geometria, devido à da simples construção do programa. A programação paramétrica, devido à descrição das variáveis no programa, utilizou-se de um número maior de caracteres, aumentando, assim, a memória ocupada.

Tabela 5 - Comparação do tamanho dos programas em kilobytes

Interpolação Helicoidal	
Método de Programação	Tamanho (kbytes)
Programação CAM Retas	168
Programação PARAMÉTRICA	0,394
Interpolação Circular	
Método de Programação	Tamanho (kbytes)
Programação CAM Retas	0,896
Programação CAM Arcos	0,128
Programação PARAMETRICA	0,427
Elipse	
Método de Programação	Tamanho (kbytes)
Programação CAM Retas	10,5
Programação CAM Arcos	1,97
Programação PARAMÉTRICA	0,628
Cavidade Circular	
Método de Programação	Tamanho (kbytes)
Programação CAM Retas	24,6
Programação CAM Arcos	2,96
Programação PARAMÉTRICA	1,02

5 CONCLUSÃO

Através da análise dos diferentes resultados obtidos, observando-se as variações de tempo de execução de cada programa pode-se perceber a interferência dos métodos de programação sobre as acelerações e desacelerações dos eixos da máquina CNC.

Dentre as formas de programação, a maioria das programações paramétricas foi o método que demonstrou maior velocidade de execução com menores vibrações, mantendo velocidade constante dos eixos, de forma que o programa não foi constituído de pequenos segmentos de retas, o que provocaria acelerações e desacelerações bruscas durante a trajetória da ferramenta.

Percebe-se que a tecnologia do controlador CNC pode contar com alguns recursos que contornam e auxiliam na execução de programas gerados por CAM, em que a geração do programa CNC depende não só do programador, mas também da configuração do próprio pós-processador do *software*. Esse tipo de recurso também é limitado pelo próprio *hardware* da máquina CNC.

Assim, o melhor desempenho é atingido minimizando o uso de pequenos segmentos de retas em interpolações circulares, dispensando cada vez mais a leitura de linhas de programa e utilizando seu processamento e cálculos internos para execução das trajetórias de ferramenta.

Para futuros trabalhos, sugestiona-se a avaliação de vibrações geradas bem como a análise do acabamento superficial em função dos recursos do *software* CAM em comparação a programação paramétrica.

BIBLIOGRAFIA

- ARIAS, M. L. **Análise do comportamento de diferentes sistemas CAM na usinagem de superfícies complexas utilizando altas velocidades**. 2009. 100f. Dissertação (Mestrado em Engenharia Mecânica) - Instituto Superior Tupy, Joinville, 2009.
- ARIAS, M. L. ; SOUZA, A. F. Descrição matemática das funções Spline. **Máquinas e Metais**. v. 507, p. 290-311, 2008.
- BENEDITO, S. R. Benefícios da Implantação do Projeto 3D na Indústria de Moldes. Qualificação para mestrado em engenharia mecânica, 2009.
- BITTENCOURT, M. **Influência das estratégias de usinagem na fabricação de moldes**. 2006. Monografia. Instituto Superior Tupy. Joinville, 2006.
- BOUJELBENE, M. et al. Productivity enhancement in dies and molds manufacturing by the use of C1 continuous tool path. **International Journal of Machine Tool & Manufacture**. Amsterdam, v. 44, p. 101-107, 2004.
- CASSANIGA, F.A. **Fácil Programação do Controle Numérico**. São Paulo: F. A. C. Promoções Editoriais, 2000. 310 p.
- CHENG, C.-W.; TSAI, M.-C. Real-time variable feed rate NURBS curve interpolator for CNC machining. The International Journal of Advanced Manufacturing Technology. v. 23, n. 11-12, p. 865-873, 2004.
- CHENG, C.-W.; TSAI, M.-C.; MACIEJOWSKI, J. Accurate Feedrate Control of CNC Machine Tools along NURBS Curves. **43rd IEEE Conference on Decision and Control**. Atlantis, Paradise Island, Bahamas. December 14-17, p. 3053-3058, 2004.
- FOGGIATTO, J. A.; VOLPATO, N.; BONTORIN, A. C. B. Os projetos em CAD 3D exigem e merecem cuidados especiais. **Máquinas e Metais**, São Paulo, v. 1, p. 132-147, 2008.
- GONÇALVES, M. A. **Um estudo sobre implementação de ciclos de usinagem através de programação parametrizada em máquinas de comando numérico computadorizado**. 2007. 96f. Dissertação (Mestrado em Engenharia Mecânica) - Universidade Federal de Santa Maria, 2007.
- HELLENO, A. L.; SCHÜTZER, K. **Influência da Interpolação Spline sobre a velocidade de avanço da manufatura de superfícies complexas**. O mundo da usinagem. São Paulo, v. 3, p. 9-16, 2007.
- HISTÓRICO do CNC. In: CNC Tecnologia. Disponível em: <<http://www.cnctecnologia.com.br/historico.htm>.> Acesso em: out. 2011.

LIU, Z. Q. Methodology of Parametric Programming for Error Compensation on CNC Centres. **The international Journal of Avanced Manufacturing Technology**. Springer-Verlag London Limited. p. 570–574, 2001.

LORINI, F. J.. **Tecnologia de grupo e organização da manufatura**. Florianópolis: UFSC, 1993. 105 p.

MUNDO CNC. Disponível em: <<http://www.mundocnc.com.br/avan2.htm>>. Acesso em: dez. 2010.

NIKIEL, G. Optimization of Execution Speed of the CNC Parametric Part Programs. **Advances in Manufacturing Science and Technology**. v. 33, n. 2, 2009.

PROCESSOS E MANUFATURA. In: PROGRAMAÇÃO CNC Parametrizada. Disponível em: <<http://processosemanufatura.com/blog/programacao-cnc-parametrizada/>>. Acesso em: dez. 2010.

ROMI. **Manual de operação e programação Siemens 810D**. Discovery, 2002. 102 p.

SALAMI, M. H.; SADEGHI, B. M. Feed rate optimization for 3-axis ball-end milling of sculptured surfaces. **International Journal of Machine Tools & Manufacture**. v,47. p. 760–767, 2007.

SILVA, A. S. A. **Desenvolvimento Integrado CAD/CAM de Componentes para turbinas a gás**. 2006. 133f. Tese (Doutorado em Sistemas Aeroespaciais e Mecatrônica) Instituto Tecnológico de Aeronáutica, São José dos Campos, 2006.

SILVA, S. D. **CNC – Programação de Comandos Numéricos Computadorizados**. 8. ed. São Paulo: Editora Érica Ltda, 2008.

SMID, P. **CNC Programming Handbook**, New York: Industrial Press, 2000, 620 p.

SOLIDCAM. Disponível em: <www.solidcam.com>. Acesso em: nov. 2011.

SOLIDWORKS. Disponível em: <www.solidworks.com>. Acesso em: nov. 2011.

SOUZA, A. F. Conceitos Avançados. **Mundo CNC**. Disponível em: <www.mundocnc.com.br/avan2.htm>. Acesso em:

SOUZA, A. F. **Contribuições ao Fresamento de geometrias complexas aplicando a tecnologia de usinagem com altas velocidades**. 2004. Tese (Doutorado em Engenharia Mecânica) - Universidade de São Paulo, São Carlos. 2004.

SOUZA, A.; STANIK, M. (2000). **Aplicação da usinagem com altíssima velocidade de corte na manufatura de moldes e matrizes**. Congresso de usinagem.

SOUZA, A. F.; COELHO, R. T. **Experimental investigation of feed rate limitations o high speed milling aimed at industrial applications**. Int J Adv Manuf Technol, v. 32, p. 1104-1114, 2007.

SOUZA, A. F. **Spline Polynomial to Describe a Free Form Tool Path for High Speed Milling**. In: INTERNACIONAL CONFERENCE ON HIGH SPEED MACHINING, San Sebastian, 2007.

SOUZA, C.E. Anais do XII ENCITA 2006. ITA, p. 16-19, 2006

STEMMER, G. E. Estágio Atual do Comando Numérico no Brasil, **Máquinas e Metais**. São Paulo, v.190, 1981.

WILSON, F. **Numerical Control in Manufacturing**. USA: Mc – Graw Hill Book Company, 1980. 399p.