

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

GERAÇÃO DE PROGRAMAS CNC ATRAVÉS DA IMPLEMENTAÇÃO DE FUNÇÕES
DIRECIONADAS ÀS CARACTERÍSTICAS DO PROCESSO PRODUTIVO

por

Marco Aurélio da Fontoura Gonçalves

Tese para obtenção do Título de
Doutor em Engenharia

Porto Alegre, agosto 2013

GERAÇÃO DE PROGRAMAS CNC ATRAVÉS DA IMPLEMENTAÇÃO DE FUNÇÕES
DIRECIONADAS ÀS CARACTERÍSTICAS DO PROCESSO PRODUTIVO

por

Marco Aurélio da Fontoura Gonçalves
Mestre em Engenharia de Produção

Tese submetida ao Programa de Pós-Graduação em Engenharia Mecânica, da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como parte dos requisitos necessários para a obtenção do Título de

Doutor em Engenharia

Área de Concentração: Fabricação Mecânica

Orientador: Prof. Dr. Flávio José Lorini

Co-orientador: Prof. Dr. Alexandre Dias da Silva

Aprovada por:

Prof. Dr. Álisson Rocha Machado..... LEPU/FEMEC/UFU

Prof. Dr. André João de Souza..... DEMEC-UFRGS

Prof. Dr. Leandro Costa de Oliveira..... CT-UFSM

Prof. Dr. Rogério José Marczak
Coordenador do PROMEC

Porto Alegre, 26, agosto 2013

*À Márcia, esposa e companheira.
Aos meus filhos, Igor Von Ende Gonçalves e
Lara Von Ende Gonçalves.*

AGRADECIMENTOS

Aos professores orientadores Flávio José Lorini e Alexandre Dias da Silva pelo apoio, confiança e dedicação na condução do desenvolvimento e realização deste trabalho.

A minha família, pelo apoio, compreensão, alegria e estímulo constante.

Aos meus pais, Marilú da Fontoura Gonçalves e Romeu Almeida Gonçalves, in memoriam.

Aos professores membros da banca de avaliação.

A direção e colegas do Colégio Técnico Industrial (CTISM), da Universidade Federal de Santa Maria (UFSM).

Ao programa de pós-graduação em Engenharia Mecânica PROMEC na forma de seus professores e funcionários.

A Universidade Federal do Rio Grande do Sul UFRGS.

RESUMO

Este trabalho descreve uma metodologia para a geração de programas CNC para otimizar um sistema produtivo. Quanto às informações geométricas, o sistema caracteriza-se por utilizar os ciclos de usinagem que o comando da máquina disponibiliza, por possibilitar a aplicação de ciclos desenvolvidos com recursos de programação parametrizada e por gerar novos ciclos, através de uma linguagem genérica, proposta neste trabalho, em um editor de programação parametrizada. Quanto aos dados tecnológicos, ou seja, as condições de usinagem definidas no código de programação, o sistema permite a otimização dos parâmetros de corte através de um banco de dados com informações realimentadas com resultados obtidos da produção. A análise da viabilidade da metodologia proposta é realizada através da implementação de um protótipo de um sistema dedicado à programação de tornos CNC. O protótipo é composto de aplicativos computacionais desenvolvidos e orientados à otimização do sistema produtivo. A metodologia para geração de funções CAD/CAM foi implementada e testada mostrando resultados eficientes. O protótipo implementado como um conjunto de ferramentas aplicativas integradas é de fácil operação e utilização unindo todos os módulos projetados.

Palavras-chave: CAD/CAM; banco de dados; usinagem; parametrização; CNC.

ABSTRACT

This study describes a methodology for the generation of CNC programs to optimize a productive system. Regarding geometrical information, the system is characterized by the use of machining cycles that the machine tool control provides because it allows the application of cycles developed with parameterized programming and by generation of new cycles, through a generic language, proposed in this study on a parameterized programming editor. As for technological data, i.e. the machining conditions set out in the programming code, the system allows the optimization of cutting parameters through a data base with information fed back with results of production. The feasibility analysis of the proposed methodology is accomplished through the implementation of a prototype of a system dedicated to programming CNC lathes. The prototype consists of computer applications developed and oriented to the optimization of the production system. The methodology for the generation of CAD/CAM functions has been implemented and tested showing effective results. The prototype implemented as an integrated set of applicative tools is of easy operation and use connecting all designed modules.

Keywords: CAD/CAM; database; machining; parameterization; CNC.

ÍNDICE

1	INTRODUÇÃO	1
1.1	<i>Objetivo do trabalho.....</i>	2
1.2	<i>Justificativa.....</i>	3
1.3	<i>Resultados esperados</i>	4
2	REVISÃO BIBLIOGRÁFICA	6
2.1	<i>Controle Numérico</i>	6
2.2	<i>Formas de programação CNC</i>	7
2.2.1	<i>Programação manual</i>	7
2.2.2	<i>Programação manual assistida.....</i>	8
2.2.3	<i>Programação MDI</i>	8
2.2.4	<i>Programação via CAD/CAM.....</i>	8
2.2.5	<i>Programação automatizada por computador</i>	9
2.3	<i>Programa CNC.....</i>	9
2.4	<i>Linguagem de alto nível</i>	10
2.5	<i>Sistemas CAD/CAM.....</i>	12
2.6	<i>Modelos de sistemas CAD/CAM</i>	14
2.7	<i>Programação Manual x CAD/CAM</i>	16
2.8	<i>Programação STEP-NC.....</i>	17
2.9	<i>Programação manual avançada</i>	18
2.10	<i>Trajetória da ferramenta.....</i>	19
2.11	<i>Pós-processador e sistemas aplicativos</i>	23
3	METODOLOGIA PARA CONCEPÇÃO DO TRABALHO.....	39
3.1	<i>Definição do sistema</i>	39
3.2	<i>Módulo de otimização de parâmetros de corte - MOPC</i>	40
3.2.1	<i>Curva de vida da ferramenta.....</i>	42
3.2.2	<i>Base de conhecimento considerada</i>	45
3.2.3	<i>Caracterização do sistema computacional.....</i>	47
3.3	<i>Módulo de programação parametrizada - MPP.....</i>	54
3.3.1	<i>Linguagem proposta</i>	55
3.3.2	<i>Caracterização da interface gráfica do algoritmo MPP</i>	58
3.4	<i>Módulo dos dados geométricos - MDG.....</i>	63
3.5	<i>Módulo dos ciclos de usinagem - MCU.....</i>	68
3.5.1	<i>Ciclo do cabeçalho padrão.....</i>	69
3.5.2	<i>Ciclo de posicionamentos.....</i>	70
3.5.3	<i>Ciclo de desbaste longitudinal.....</i>	71
3.5.4	<i>Ciclo do perfil da peça usinada.....</i>	73

3.5.5	Ciclo de rosqueamento automático.....	73
3.5.6	Ciclo de furação profunda	75
3.5.7	Ciclo de abertura de canais	77
3.6	<i>Caracterização do protótipo computacional</i>	78
4	APLICAÇÃO E AVALIAÇÃO DOS MÓDULOS DO SISTEMA	80
4.1	<i>Aplicação do Módulo de otimização de parâmetros de corte (MOPC)</i>	80
4.1.1	Estudos relacionados no cálculo do banco de dados CINFUS	80
4.1.2	Estudos relacionados ao cálculo do tempo de vida da ferramenta	83
4.1.3	Estudos relacionados ao cálculo das constantes da equação de Taylor.....	86
4.2	<i>Aplicação do Módulo de programação parametrizada (MPP)</i>	92
4.2.1	Ciclo para torneamento.....	92
4.2.2	Programa parametrizado com múltiplos ciclos	94
4.3	<i>Aplicação do Protótipo CAD/CAM</i>	97
4.3.1	Atualizando o MOPC para o projeto	99
4.3.2	Geração do programa para a peça especificada.....	101
4.3.3	Fabricação da peça modelo.....	108
5	CONCLUSÃO	112
	REFERÊNCIAS BIBLIOGRÁFICAS	114

LISTA DE FIGURAS

Figura 2.1 - Exemplo de programação ISO, Código “G/M”	10
Figura 2.2 - Ambiente da linguagem APT	11
Figura 2.3 - Fluxo generalizado do sistema CAD/CAM.....	13
Figura 2.4 - Exemplo de sistema CAD/CAM	14
Figura 2.5 - Modelos de programação CAD/CAM	15
Figura 2.6 - Comparativo do sistema “G/M” e STEP-NC	17
Figura 2.7 - Esquema geral do STEP-NC	18
Figura 2.8 - Comparativo programação CAD/CAM e Parametrizada	19
Figura 2.9 - Tolerância para cálculo dos programas CNC	21
Figura 2.10 - Interpolações, (a) linear, (b) linear e circular, (c) polinomial	22
Figura 2.11 - Interpolações normalmente usadas por CAM	22
Figura 2.12 - Assistente de códigos para desenvolvimento do pós-processador	24
Figura 2.13 - Pós-processador	25
Figura 2.14 - Interface do programa CNPP	27
Figura 2.15 - CNC/CAM integrador	29
Figura 2.16 - Processo normal e proposto	30
Figura 2.17 - Gerador de códigos	30
Figura 2.18 - Fluxograma de leitura DXF	31
Figura 2.19 - Cadeia mais comum de produção CAx	32
Figura 2.20 - Rede interoperável CAx	33
Figura 2.21 - Visão geral da plataforma universal	34
Figura 3.1 - Esquema do sistema.....	41
Figura 3.2 - Interface gráfica do protótipo CAD/CAM dedicado	41
Figura 3.3 - Fluxograma do MOPC.....	43
Figura 3.4 - Influência dos parâmetros na equação de Taylor	46
Figura 3.5 - Fluxograma do algoritmo do MOPC	47
Figura 3.6 - Estrutura relacional do MOPC.....	48
Figura 3.7 - Interface para cadastro da classe da ferramentas	49
Figura 3.8 - Interface para cadastro de ferramentas	50
Figura 3.9 - Interface para cadastro de materiais.....	50
Figura 3.10 - Interface para cadastro de máquinas	51

Figura 3.11 - Folha de dados experimentais	52
Figura 3.12 - Exemplo de dados tecnológicos extraídos do programa gerado.....	52
Figura 3.13 - Interface para inserir parâmetros de usinagem no sistema	53
Figura 3.14 - Interface para filtro de parâmetros de usinagem no sistema.....	54
Figura 3.15 - Interface de otimização do sistema.....	54
Figura 3.16 – Exemplo de programa com a linguagem proposta.....	58
Figura 3.17 - Interface gráfica do MPP	59
Figura 3.18 - Esquema do MPP.....	60
Figura 3.19 - Trecho do algoritmo interpretador para o código “G1”.....	60
Figura 3.20 - Módulo de Programação Parametrizada.....	61
Figura 3.21 - Sequência de fases para obter arquivo CNC.....	62
Figura 3.22 - Interface do módulo geométrico	63
Figura 3.23 - Ciclo MCS para abertura de canais.....	64
Figura 3.24 - Fluxograma da rotina <i>AutoLISP</i> para abertura de canal	64
Figura 3.25 - Códigos DXF do objeto “linha”	65
Figura 3.26 – Ilustração com trecho do programa ciclo de abertura de canal.....	66
Figura 3.27 - Arquivo texto gerado pela rotina abertura de canal.....	66
Figura 3.28 - Exemplos de rotinas desenvolvidas	67
Figura 3.29 - Exemplos de ciclos de desbaste em diferentes comandos CNC.....	68
Figura 3.30 - Funções miscelânea	69
Figura 3.31 - Cabeçalho padrão vinculado aos parâmetros calculados no MOPC.....	70
Figura 3.32 - Interface de posicionamento	71
Figura 3.33 - Desbaste longitudinal.....	72
Figura 3.34 - Interface para captura do perfil da peça.....	73
Figura 3.35 - Ciclo de rosqueamento automático.....	74
Figura 3.36 - Formulário de “Ajuda” do ciclo de rosqueamento	75
Figura 3.37 - Ciclo de furação profunda.....	76
Figura 3.38 - Formulário de “Ajuda” do ciclo de furação profunda	76
Figura 3.39 - Ciclo de canal.....	77
Figura 3.40 - Formulário de “Ajuda” do ciclo de canal	78
Figura 3.41 –Interface principal do sistema	79
Figura 3.42 - Interface principal com aplicativos executando.....	79

Figura 4.1 - Interface com filtro de dados CINFUS	81
Figura 4.2 - Cálculo dos coeficientes “C” e “G” de Taylor	82
Figura 4.3 - Filtro das condições de corte	84
Figura 4.4 - Interface de otimização com resultados	85
Figura 4.5 - Interface cálculo das constantes.....	88
Figura 4.6 - Tempo Máxima Produção (Método/MOPC)	89
Figura 4.7 - Tempo Mínimo Custo (Método/MOPC)	90
Figura 4.8 - Velocidade Máxima Produção (Método/MOPC)	90
Figura 4.9 - Velocidade Mínimo Custo (Método/MOPC)	91
Figura 4.10 - Desbaste em ângulo	92
Figura 4.11 - Variação da área de contato da ferramenta.....	92
Figura 4.12 - Fluxograma desbaste em ângulo.....	93
Figura 4.13 - Código comentado do ciclo de desbaste em ângulo	94
Figura 4.14 - Peça padrão representativa de uma família.....	95
Figura 4.15 - Fluxo do programa padrão	95
Figura 4.16 - Código do programa padrão	96
Figura 4.17 – Resultado da compilação no MPP do programa Padrão	97
Figura 4.18 - Aplicação dos ciclos desenvolvidos pelo MPP.....	98
Figura 4.19 - Modelo para gerar programa no Protótipo CAD/CAM.....	98
Figura 4.20 - Inserindo dados do sistema no banco de dados do MOPC	99
Figura 4.21 - Filtro dos dados para o projeto	100
Figura 4.22 - Interface de otimização do MOPC.....	100
Figura 4.23 - Posicionamento do zero peça.....	101
Figura 4.24 - Desenho do perfil da peça.....	102
Figura 4.25 - Cabeçalho do programa	102
Figura 4.26 - Posicionamento da ferramenta.....	103
Figura 4.27 - Extração de parâmetros para ciclo de desbaste longitudinal	104
Figura 4.28 - Programação do ciclo desbaste longitudinal.....	104
Figura 4.29 - Extração de parâmetros para ciclo de canal.....	105
Figura 4.30 - Programação do ciclo do canal	105
Figura 4.31 - Extração de parâmetros para o ciclo de rosca.....	106
Figura 4.32 - Programação do ciclo da rosca	106

Figura 4.33 - Extração do perfil da peça	107
Figura 4.34 - Ciclo captura o perfil automaticamente	108
Figura 4.35 - Programa executado no comando MCS.....	108
Figura 4.36 - Listagem do programa elaborado no Protótipo CAD/CAM.....	109
Figura 4.37 - Manufatura da peça modelo.....	110

LISTA DE TABELAS

Tabela 2.1 - Programa APT para definição de geometria	12
Tabela 2.2 - Comparações entre Programação Manual e CAM	16
Tabela 2.3 - Programa para FANUC e SIEMENS para furação	26
Tabela 2.4 - Relação Entre APT e CNC	28
Tabela 3.1 - Operadores aritméticos	56
Tabela 3.2 - Operadores relacionais	56
Tabela 3.3 - Código ISO 6983	56
Tabela 4.1 - Cálculo do tempo de vida da ferramenta para dados do CINFUS	81
Tabela 4.2 - Comparativo de resultados	82
Tabela 4.3 - Dados experimentais	83
Tabela 4.4 - Comparativo dos resultados	85
Tabela 4.5 - Tempo Calculado para os dados experimentais	87
Tabela 4.6 - Comparando método com MOPC	87
Tabela 4.7 - MOPC (f máximo e mínimo)	88
Tabela 4.8 - Variáveis para desbaste em ângulo	93

LISTA DE SIGLAS E ABREVIATURAS

ABNT	Associação Brasileira de Normas Técnicas
ANSI	American National Standards Institute
APT	Automatically Programmed Tool
ASCII	American Standard Code for Information Interchange
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CAPP	Computer Aided Process Planning
CAx	Computer Aided Systems
CBN	Cubic Boron Nitride
CLDATA	Cutter Location Data
CLFILE	Centerline Data File
CNC	Computer Numeric Control
CNPP	Numerical Control Program Processor
CT	Centro Tecnológico
CTISM	Colégio Técnico Industrial de Santa Maria
DIN	German Institute for Standardization
DNC	Direct Numerical Control
DOE	Metodologia de Superfície de Resposta
DWG	Autodesk Drawing Exchange Format
DXF	Drawing Exchange Format
EBNF	Extended Backus Naur For
EIA	Electronic Industries Association
G	General or preparatory
HSC	High Speed Cutting
HTML	HyperText Markup Language
IGES	Initial Graphics Exchange Specification
ISO	International Organization for Standardization
ITC	Icam Technologies Corporation
LAN	Local Area Network
MCU	Módulo de Ciclos de Usinagem

MDG	Módulo de Dados Geométrico
MDI	Manual Data Input
MIT	Massachusetts Institute of Technology
MOPC	Módulo de Otimização de Parâmetros de Corte
MPP	Módulo de Programação Parametrizada
NURBS	Non-Uniform Rational B-Spline
PROMECC	Programa de Pós-Graduação em Engenharia Mecânica
UFRGS	Universidade Federal do Rio Grande do Sul
UFSC	Universidade Federal de Santa Catarina
UFSM	Universidade Federal de Santa Maria
USAF	United States Air Force

LISTA DE SÍMBOLOS

a_p	Profundidade de corte, mm
C, G	Constantes de Taylor
C_m	Custo de Utilização da Máquina-Ferramenta, R\$
C_t	Custo por Gume, R\$
E, F, G, H	Expoentes Determinados Experimentalmente
f	Avanço, mm/rot
T	Tempo de vida da ferramenta, min
t_c	Tempo de Troca da Ferramenta, min
$T_{otm,c}$	Tempo de Vida Ótima Mínimo Custo, min
$T_{otm,p}$	Tempo de Vida Ótima Máxima Produção, min
VB_B	Largura da Marca de Desgaste, mm
v_c	Velocidade de corte, m/min
$v_{otm,c}$	Velocidade de corte Ótima Mínimo Custo, m/min
$v_{otm,p}$	Velocidade de corte Ótima Máxima Produção, m/min

1 INTRODUÇÃO

Máquinas que necessitam de posicionamento, velocidade e aceleração controlados podem ser automatizadas por meio de um Controle Numérico Computadorizado, também denominado comando CNC (*Computer Numeric Control*). Na atualidade, máquinas-ferramenta acionadas por comandos numéricos, tais como furadeiras, puncionadeiras, dobradeiras, tornos, fresadoras e mandriladoras, são alguns exemplos onde o CNC é aplicado. Empresas de pequeno porte e prestadoras de serviço já usufruem dessa tecnologia, não sendo mais privilégio das grandes indústrias.

Para produzir uma peça usinada com uma máquina-ferramenta comandada numericamente são necessários vários procedimentos, sendo um dos principais a elaboração do programa de instruções, ou seja, a geração dos códigos referentes à sequência de operações a serem executadas na máquina. Esse programa contém informações geométricas, que consistem no caminho percorrido pela ferramenta, e tecnológicas, que são as condições de usinagem. Os parâmetros tecnológicos devem propiciar boa qualidade de acabamento, redução de custo operacional e otimização do tempo de operação [Groover, 2011].

A programação CNC pode ser realizada por técnicas tradicionais como a Entrada Manual de Dados MDI (*Manual Data Input*). Nessa técnica, o programa é inserido diretamente no comando da máquina ou via sistemas computacionais integrados com a máquina. Os comandos das máquinas CNC oferecem grande variedade de funções e ciclos de usinagem com a finalidade de facilitar a forma de programação. Com esses recursos, o programador evita trabalhar com grande número de blocos de programação, pois os ciclos de usinagem disponíveis nos comandos refinam muitos processos repetitivos na elaboração do programa. A quantidade e qualidade dos ciclos dos comandos fazem parte do valor agregado para a comercialização das máquinas CNC. Como exemplo, na usinagem cilíndrica, os ciclos de usinagem são implementados para realizar operações como desbastes, abertura de canais, geração de roscas e chanfros ou outros procedimentos que são normalmente aplicados na manufatura de uma peça [Figueira, 2003].

A programação parametrizada é uma forma avançada de programação manual que permite incluir em programas cálculos matemáticos, variáveis computacionais e desvios condicionais. Isso permite programar a lógica algorítmica na operação da máquina de comando numérico. É uma linguagem de programação que reside no comando CNC e pode

ser acessada ao nível do código padrão. Combina técnicas de programação manuais com técnicas de programação paramétricas.

Técnicas mais avançadas, como sistemas de integração das etapas de projeto e fabricação, denominados de CAD/CAM (*Computer Aided Design/Computer Aided Manufacturing*), são aplicadas através de recursos computacionais. Esses *softwares* extraem as informações de sistemas CAD, através de processamento matemático, analisam ponto a ponto o desenho e geram arquivos de dados referentes aos posicionamentos da ferramenta interpretados da geometria. As informações podem ser armazenadas em arquivos no formato CLDATA (*Cutter Location Data*) [Groover, 2011]. Esses arquivos, também chamados de neutros, não são diretamente interpretados pelos comandos CNC. As instruções geradas por esses sistemas são adaptadas às respectivas máquinas para execução através de pós-processadores que consideram características operacionais de cada equipamento. Essa técnica normalmente dispensa os ciclos oferecidos pelo comando da máquina, o que caracteriza uma subutilização da potencialidade dos recursos de *software* embarcados no comando CNC.

A integração entre diferentes sistemas auxiliados por computador, generalizados pela sigla CAx (*Computer Aided Systems*), é relatada em publicações recentes por vários pesquisadores. Liu *et al.*, 2008, destacam sobre a necessidade de interação entre os sistemas CAM e comandos CNC. Zhang *et al.*, 2011, propuseram a padronização de uma linguagem específica aceita por todos os comandos com o objetivo de facilitar essa integração. Técnicas para ler diretamente o arquivo de desenho em formato padronizado são descritas por Wang e Su, 2010. A simulação virtual utilizando parâmetros de fabricante comparada com os dados de condições de corte reais pode gerar variações nos resultados dos processos [Zalewski, 2011]. Informações geométricas da peça podem também ser capturadas a partir de imagens digitais [Cheypoca *et al.*, 2010]. Problemas de pós-processamento CNC na indústria aeronáutica são relatados por Cheng e Jihong, 2011.

1.1 Objetivo do trabalho

Este trabalho propõe metodologias que possibilitem o uso dos recursos disponíveis em sistemas produtivos específicos, com a finalidade de otimizar a geração automática de programas CNC dedicados a esses sistemas.

Com o objetivo de analisar a técnica, um aplicativo computacional, inserido em um contexto de ferramentas para manufatura auxiliada por computador, é apresentado para

processar informações de capacidade do processo, quanto aos dados geométricos e tecnológicos.

Na definição dos recursos do protótipo, quanto à geometria, pretende-se explorar a utilização de funções de programação para a execução de ciclos de usinagem. No sistema proposto, os ciclos utilizados podem ser provenientes de funções de programação da linguagem dos próprios comandos CNC, de ciclos elaborados através de programação parametrizada ou de ciclos elaborados por um editor genérico de programação parametrizada, que é sugerido nesse trabalho através da definição de uma linguagem computacional universal. Em relação aos dados tecnológicos, a metodologia propõe a otimização dos parâmetros de corte através da elaboração de um sistema de gerenciamento interativo de banco de dados de usinagem. O conteúdo desse banco de dados tem como característica o seu aprimoramento através da realimentação de resultados obtidos do sistema produtivo e a utilização de filtros de dados para gerar informações adaptadas a aplicações específicas.

1.2 Justificativa

Na geração do programa CNC, com a técnica MDI, a inserção dos códigos é realizada diretamente no comando, interferindo na produção contínua da máquina. Utilizando um *software* CAM, a programação pode ser realizada em uma estação de trabalho sem a necessidade de parada do equipamento. O Código gerado pelo CAM, para ser interpretado pelo comando, necessita de um pós-processador, que normalmente compila para codificação ISO. Um pós-processador que utilize os ciclos das linguagens interativas características dos diferentes fabricantes seria de aplicação mais interessante. O trabalho propõe uma técnica de implementação de recursos para programação automática de equipamentos CNC para aplicação no desenvolvimento de sistemas dedicados. Essa metodologia caracteriza-se por ter funções CAD/CAM e gerar programas utilizando ciclos de comandos específicos sem necessidade de pós-processador.

Normalmente os sistemas de bancos de dados relacionados a parâmetros de corte de usinagem possuem como característica o armazenamento de informações e a possibilidade de recuperá-las quando necessário. O banco de dados do sistema proposto é caracterizado por uma técnica de filtragem dos parâmetros de corte, possibilitando combinações de dados para o cálculo das condições de corte convenientes para o processo.

Os comandos disponibilizam uma forma de programação avançada, com o intuito de facilitar a programação, implementar novos ciclos ou aprimorar trajetórias complexas. A

utilização da linguagem parametrizada é diferenciada para cada tipo de comando fazendo com que essa técnica seja pouco difundida. No trabalho proposto, define-se uma linguagem parametrizada universal e um algoritmo de interpretação e compilação, gerando o código CNC.

A implementação de sistemas computacionais dedicados, voltados à necessidade de um sistema produtivo, normalmente é dificultada por exigir uma gama de conhecimentos pontuais relativo a sistemas CAD/CAM. Neste trabalho, são disponibilizadas metodologias para o desenvolvimento de funções CAD/CAM relativas a aplicações em sistemas dedicados. Características como a facilidade de operação, o baixo custo e o direcionamento à tecnologia do sistema produtivo em que for aplicado são vantagens que podem ser obtidas. O método pode ser considerado como uma opção alternativa à aplicação de sistemas CAD/CAM para a geração do código CNC.

1.3 Resultados esperados

Um programa CNC gerado pelo sistema utilizando ciclos definidos do comando CNC terá como característica a forma compacta, com influência na variação do tempo de execução dos blocos do programa, alterando as acelerações e desacelerações dos eixos da máquina na execução das trajetórias. Uma análise dos tempos de execução e das velocidades será realizada no comando específico, onde serão aplicados os programas gerados pelo protótipo do sistema proposto.

A eficiência do banco de dados proposto deverá ser averiguada através de um estudo comparativo entre os resultados obtidos no sistema computacional e trabalhos científicos que utilizam técnicas alternativas no cálculo das constantes de Taylor e/ou tempo de vida da ferramenta, possibilitando a análise dos resultados obtidos após realimentações através da geração de dados relacionados a custos e produção.

Neste trabalho, propõe-se um compilador e uma linguagem genérica, ambos destinados à elaboração de programas parametrizados, elaborados e compilados para validar o sistema proposto.

A interface com o programador deverá ter um ambiente amigável, não exigindo do usuário conhecimento específico do formato e sintaxe das funções de programação referentes a ciclos de usinagem dos mais variados comandos, apenas o domínio da tecnologia de usinagem.

A análise da viabilidade dessa metodologia será testada com a implementação de um protótipo de um sistema dedicado à programação de tornos CNC, com comando MCS, através do desenvolvimento de aplicativos computacionais orientados à otimização do sistema produtivo. O sistema será aplicado para explorar os atuais recursos de programação disponíveis nos comandos das máquinas comerciais.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo, é apresentada uma revisão bibliográfica, abordando aspectos importantes à compreensão do tema proposto. Esses aspectos incluem conhecimentos relativos ao processo de programação em máquinas-ferramentas comandadas numericamente e assuntos relacionados.

2.1 Controle Numérico

Em 1949, a USAF (*United States Air Force*) descobriu uma pequena empresa fabricante de hélices e rotores para helicópteros, a *Parsons Corporation*. Com dificuldades para produzir gabaritos para os rotores na velocidade necessária de produção, havia experimentado uma forma rudimentar de controle por números em uma máquina de usinagem convencional [Groover, 2011].

A USAF, incentivada pelo grande avanço na fabricação de aviões e material bélico, contratou a *Parsons Corporation* e patrocinou estudos e o desenvolvimento do controle numérico. Planejaram e executaram as adaptações de controle numérico para uma fresadora de três eixos, a *Hydrotel*, da *Cincinnati Milling Machine Company*. Os controles e comandos convencionais foram retirados e substituídos pelo controle numérico, dotado de leitora de fita de papel perfurado, unidade de processamento de dados e servomecanismos nos eixos. Desse modo, foi criado um protótipo de uma máquina acionada por CNC, apresentado em 1953 no MIT (*Massachusetts Institute of Technology*). Os resultados foram excepcionais e demonstraram uma mudança revolucionária nas indústrias de transformação [Wilson, 1980].

Após esse período, a USAF teve um desenvolvimento extraordinário, pois as peças complexas e de grande precisão, empregadas na fabricação das aeronaves, principalmente nos aviões a jato de uso militar, passaram a ser produzidas de forma mais simples e rápida, reduzindo-se os prazos de entrega do produto, desde o projeto até o acabamento final. Em 1956, surgiu o trocador automático de ferramentas. Já em 1958, surgiram os equipamentos com controle de posicionamento ponto a ponto e a geração contínua de contornos foram melhorados por esse sistema em desenvolvimento.

A partir de 1957, houve nos Estados Unidos uma grande corrida na fabricação de máquinas comandadas numericamente. Até então, os empreendedores investiam em adaptações do CNC em máquinas convencionais. Esse novo processo foi cada vez mais

aplicado na rotina de manufatura, fazendo com que surgissem novos fabricantes, os quais já produziam seus próprios sistemas e linguagem de comandos. Devido ao grande número de fabricantes, começaram a surgir os primeiros problemas, sendo o principal a falta de uma linguagem única e padronizada [Figueira, 2003].

Em 1967, as primeiras máquinas controladas numericamente chegaram ao Brasil, vindas dos Estados Unidos. A partir daí, observou-se uma evolução contínua e concomitante ao desenvolvimento dos computadores em geral, fazendo com que os comandos CNC mais modernos empreguem hoje, em seu conceito físico (*hardware*), tecnologia de última geração.

Atualmente, a utilização do CNC é uma alternativa apropriada para a solução dos mais complexos problemas de usinagem. Onde anteriormente se exigia uma máquina ou uma ferramenta especial, atualmente é feito com o CNC de uma forma relativamente simples [Cassaniga, 2005].

2.2 Formas de programação CNC

Os códigos de um programa CNC são responsáveis pelo controle de uma máquina, informando todas as etapas de fabricação de uma determinada operação de usinagem. Um bloco de comando de um programa CNC pode conter informações sobre o deslocamento da ferramenta (movimento rápido, interpolação, etc.), informações tecnológicas (velocidade, avanço, etc.), ou informações que ativem funções auxiliares (ligar refrigerante, eixo da árvore, etc.). A obtenção dessas informações depende da geometria da peça usinada, considerando-se as limitações da máquina, as características do comando e da ferramenta [Figueira, 2003].

Os métodos ou técnicas de programação incluem uma variedade de procedimentos, que vão desde o método manual até o método automatizado. Lynch, 1997b, por exemplo, classifica as técnicas de programação em cinco tipos: programação manual, programação manual assistida, programação MDI (*Manual Data Input*), programação via CAD/CAM e programação automatizada por computador.

2.2.1 Programação manual

A programação manual refere-se à preparação de um programa, sem a utilização de recursos computacionais para determinar trajetórias de ferramentas, pontos de intersecção de perfis, avanços, velocidades, etc. Nesse modo de programação, as instruções de processamento são documentadas em um formulário manuscrito do programa, que é uma

listagem das posições da ferramenta em relação à peça, que a máquina precisa seguir para executar o processamento. A listagem também pode incluir outros comandos com informações sobre velocidades, avanços, ferramental, etc. Esse tipo de programação é ideal para aplicações ponto a ponto em trabalhos de torneamento e fresamento que envolvam o processamento de peças geometricamente simples e que requeiram poucas operações [Lynch, 1997a].

2.2.2 Programação manual assistida

Na programação assistida por computador, o trabalho de cálculo requerido na programação manual é automatizado. Para peças com geometrias complexas ou trabalhos com muitas etapas de processamento, o uso do computador resulta em economia significativa de tempo de programação. A escolha entre programação manual e programação assistida por computador depende do tipo e complexidade das peças produzidas, do número de eixos programados em cada máquina CNC, do volume de programas elaborados por período e do tempo médio de elaboração dos programas.

2.2.3 Programação MDI

O sistema MDI (*Manual Data Input*) é um procedimento no qual o programa é inserido diretamente no comando da máquina-ferramenta. Conseqüentemente, o uso de sistemas de transferência de dados é desnecessário. O procedimento de programação é simplificado para permitir que, além dos programadores, os operadores de máquina também façam ou alterem o programa. O operador/programador insere os dados do programa e comandos na unidade de controle da máquina.

2.2.4 Programação via CAD/CAM

Esse modo de programação é uma forma avançada da programação assistida por computador, onde um sistema gráfico interativo, equipado com *software* de programação CNC, é usado para facilitar a atividade de programação. Nesses sistemas, o programador trabalha numa estação CAD/CAM para selecionar ferramentas, métodos e procedimentos para usinar a peça projetada no CAD.

2.2.5 Programação automatizada por computador

Nesse modo automatiza-se a atividade de programação completa, usando um *software* que é capaz de tomar decisões lógicas sobre como a peça deva ser usinada. Na programação CNC usando o sistema CAD/CAM, muitos aspectos do procedimento foram automatizados. O objetivo é tornar possível automatizar o procedimento completo da programação. A interação entre esta atividade e Planejamento do Processo Assistido por Computador o CAPP (*Computer Aided Process Planning*) é fundamental para o sucesso da técnica [Lynch, 1997b].

2.3 Programa CNC

No início do desenvolvimento de máquinas comandadas numericamente, a falta de padronização era sentida nas empresas, com várias máquinas fabricadas por diferentes fornecedores, sendo que cada um tinha uma linguagem própria. Com isso, havia a necessidade de uma equipe técnica especializada para cada tipo de comando, elevando os custos de fabricação [Cassaniga, 2005].

Em 1958, por intermédio da EIA (*Electronic Industries Association*), organizaram-se estudos no sentido de padronizar os tipos de linguagem. A linguagem destinada à programação de máquinas era a APT (*Automatically Programmed Tool*) desenvolvida pelo Instituto de Tecnologia de *Massachusetts*, em 1956. Desde então foram desenvolvidas outras linguagens para a geração contínua de contornos como *AutoPrompt*, *ADAPT*, *Compact II*, *Action*, e outras que surgiram para novas aplicações. Os códigos de máquinas CNC mais utilizados no mundo foram padronizados inicialmente pela ISO 1056,1975, e posteriormente atualizados pela ISO 6983-1,1982, (*International Organization for Standardization*) e de forma muito semelhante pela ANSI NAS995,1996, (*American National Standards Institute*).

A programação manual é a base para a geração dos programas CNC, entre as mais utilizadas está a programação ISO 6983, escrita seguindo uma lógica básica. No início do programa são declarados códigos que compõem o cabeçalho do programa. São responsáveis pelas condições de usinagem, por exemplo, “G21” indica para a máquina que a unidade de medida é milímetro e “G90” ativa as coordenadas absolutas. Com essa lógica, cada código transmite para a máquina as condições de trabalho. As coordenadas “X e Z” correspondem ao deslocamento realizado da máquina. Na Figura 2.1, a linha “N30” define um deslocamento nas direções “X e Z” de 100 mm, a partir do ponto zero indicado por “G54”. O programador,

através desses códigos, pode definir deslocamentos rápidos “G00”, interpolações lineares “G01”, interpolações circulares “G02” e “G03”, entre outros.

N10	G21	G40	G90	G95
N20	G00			
N30	G54	X100	Z100	
N40	T01			
N50	G97	S545	M3	
N60	M8			
N70	X85	Z2		
N80	G00	X30		
N100	G01	Z0.0		
N110	G03	X40	Z-5	R5
N120	G01	Z-20		
N130	G01	Z-70		
N180	M30			

Figura 2.1 - Exemplo de programação ISO, Código “G/M”

Os códigos de funções miscelâneas são responsáveis por ligar e desligar algumas funções da máquina, por exemplo, “M40” avança e “M41” recua contra ponto, “M46” abre e “M47” fecha o sistema de fixação da matéria prima, “M8” liga e “M9” desliga fluido refrigerante, entre outros. Mesmo seguindo um padrão ISO 6983, a programação muda para comandos de marcas diferentes, cada um com suas particularidades.

2.4 Linguagem de alto nível

A linguagem APT (*Automatically Programmed Tool*) representa o método mais tradicional de programação auxiliada por computador. A linguagem APT foi desenvolvida no MIT para facilitar ao programador o fornecimento de informações sobre a peça e os movimentos da ferramenta para o computador [Koren, 1986]. Existem diferentes versões para a linguagem APT, como a APTURN (para tornos), APTMILL (para fresadoras e furadeiras) e APTPOINT (para operações ponto-a-ponto).

A função do programador, utilizando a linguagem APT, é descrever o programa fonte, onde é definida a geometria da peça e/ou o percurso da ferramenta via definição de forma padronizada pela linguagem de dados geométricos e funções auxiliares.

A função do processador é, através do programa fonte, realizar os cálculos geométricos, determinar o contorno da ferramenta e gerar um arquivo neutro CLDATA (*Cutter Location Data*) ou CLFILE (*Centerline Data File*), independente da máquina. Posteriormente esse arquivo é pós-processado, gerando um arquivo para a máquina específica.

Com o APT, a programação é realizada assumindo-se que a peça esteja parada e utilizando-se um sistema de coordenadas fixo na peça. A linguagem assume que a ferramenta se mova acompanhando uma reta. Logo, se a trajetória desejada é uma curva, tal trajetória é aproximada por um conjunto de segmentos de reta, com a precisão indicada pelo programador. As coordenadas obtidas são sempre as da ponta da ferramenta, ou seja, a intersecção do eixo da ferramenta com a superfície inferior. O esquema operacional da linguagem APT é mostrado na Figura 2.2.

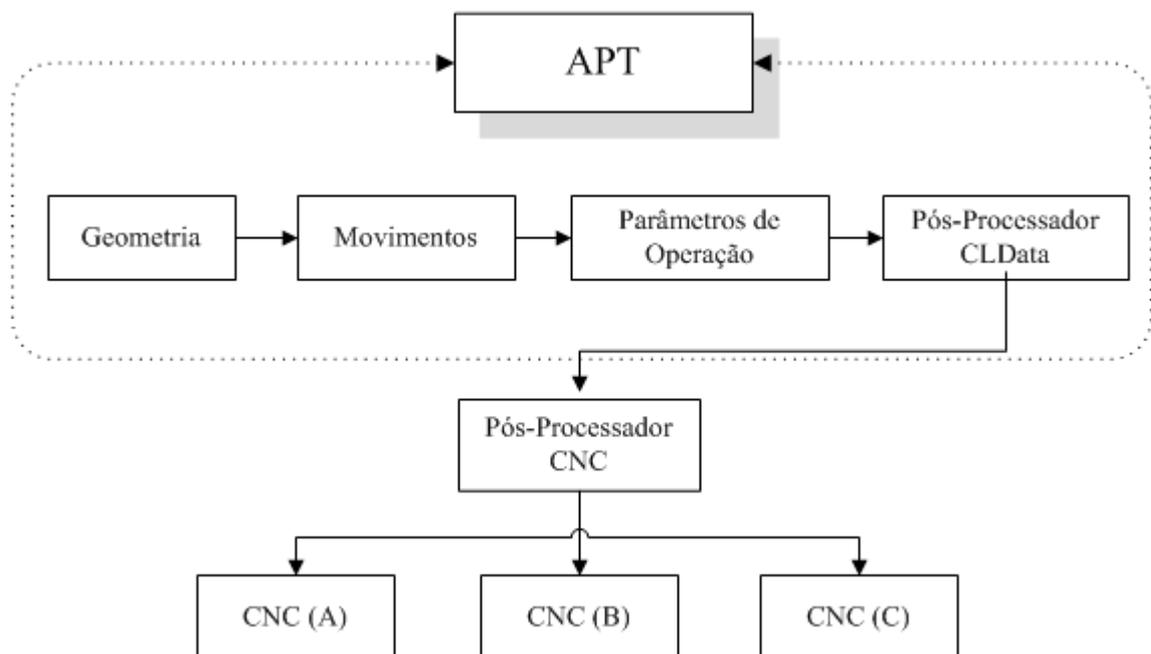


Figura 2.2 - Ambiente da linguagem APT [Adaptada de Meneghello, 2003]

Sua estrutura consiste em quatro tipos de declarações: geometria, movimento, pós-processamento e declarações auxiliares. No primeiro tipo de declaração, são definidas as geometrias dos elementos utilizados na concepção do desenho indicando os símbolos,

definindo os tipos de elementos (ponto, linhas, círculos, etc.) e descrevendo dados para sua definição precisa (coordenada, raio, etc.). Na declaração de movimento são introduzidos comandos de movimento de ferramenta que explicam o tipo de ação que deva ser realizada e os dados de descrição que definem o movimento da ferramenta. As palavras reservadas do APT para definição de comandos de movimentos são palavras comuns do meio industrial, o que facilita a compreensão [Groover, 2010].

Nas declarações de pós-processamento são definidos comandos para controle dos parâmetros de usinagem como rotação, avanço e velocidade de corte, bem como outras características de máquina-ferramenta como controle do fluido de refrigeração. A Tabela 2.1 mostra um exemplo de programação APT para definição de geometria.

Tabela 2.1 - Programa APT para definição de geometria [Adaptada de Groover, 2007]

<i>APT</i>	
P0	POINT/0,-1.0,0
P1	POINT/6.0,1.125,0,0
P2	POINT/0,0,0
P3	POINT/6.0,0,0
P4	POINT/1.75,4.5,0
L1	LINE/P2,P3
C1	CIRCLE/CENTER,P1,RADIUS, 1.125
L2	LINE/P4,LEFT,TANTO,C1
L3	LINE/P2,P4
PL1	PLANE/P2,P3,P4.

2.5 Sistemas CAD/CAM

A programação CNC com base em sistemas CAD/CAM foi criada inicialmente para solucionar as falhas da programação manual. Atualmente é concebida para programar tanto peças simples como complexas, de forma mais rápida, mais fácil e com um grau mais alto de precisão. Suas aplicações são bem sucedidas nas indústrias de fabricação de moldes e aeroespacial, que trabalham com usinagem de peças complexas e cujos programas são muito extensos [Lynch, 1994]. O arquivo CAD contém as informações necessárias ao processo como a geometria da peça. Na fase das configurações básicas é estabelecido o ponto de

origem das coordenadas necessárias para a movimentação dos eixos. Nessa etapa inicial gera-se o modelo sólido e o material a partir de blocos que representam a matéria-prima. Na sequência são definidos os parâmetros da máquina para confeccionar a peça e o pós-processador a ser usado. O processo CAD/CAM pode variar em relação aos diversos softwares. Um fluxo generalizado é apresentado na Figura 2.3.

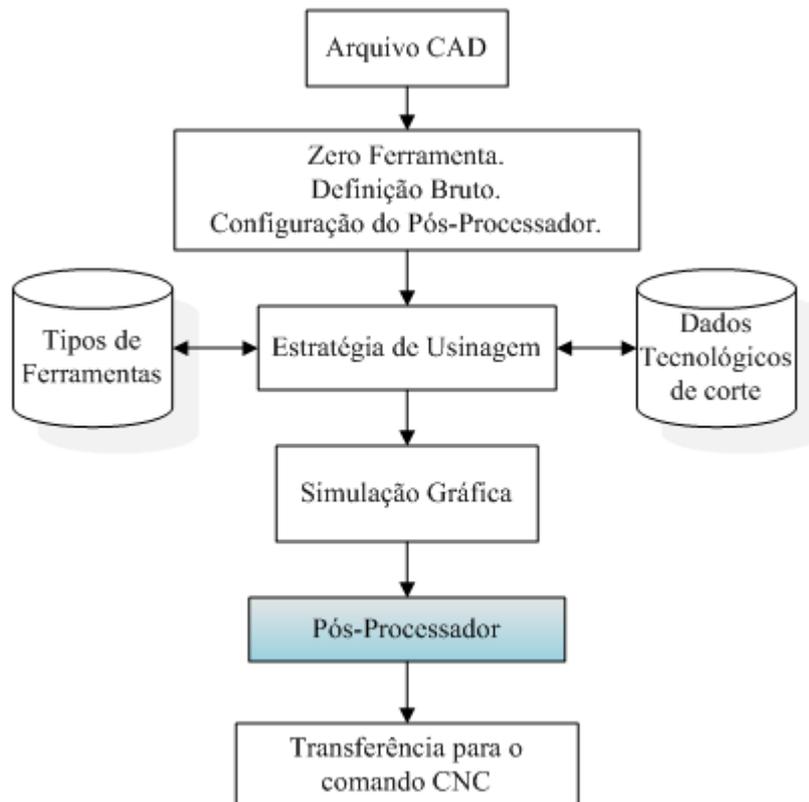


Figura 2.3 - Fluxo generalizado do sistema CAD/CAM

A escolha das estratégias de usinagem é uma fase que exige experiência e conhecimento do processo, onde são definidas as condições de acabamento, furações, desbaste, entre outros.

Os parâmetros de corte e o tipo de ferramenta são definidos conforme o processo. Por exemplo, para a operação de desbaste da peça são necessários os dados de avanço e velocidade relativa à ferramenta usada.

Na simulação da usinagem todo o projeto é posto em teste. É feita em ambiente computacional gráfico permitindo verificar colisões, tempos de usinagem, trocas de ferramentas e pequenos imprevistos, antes da fabricação final.

O pós-processador realiza a transformação da linguagem gráfica gerada pelo sistema CAD/CAM em linguagem CNC relativa ao comando da máquina especificado.

A transmissão do programa para o comando da máquina é realizado através de várias técnicas como porta serial, conexões via rede entre outras. A Figura 2.4 ilustra a interface de um sistema CAD/CAM, adaptada do *software EdgeCAM*, com os recursos de projeto, simulação e geração código CNC.

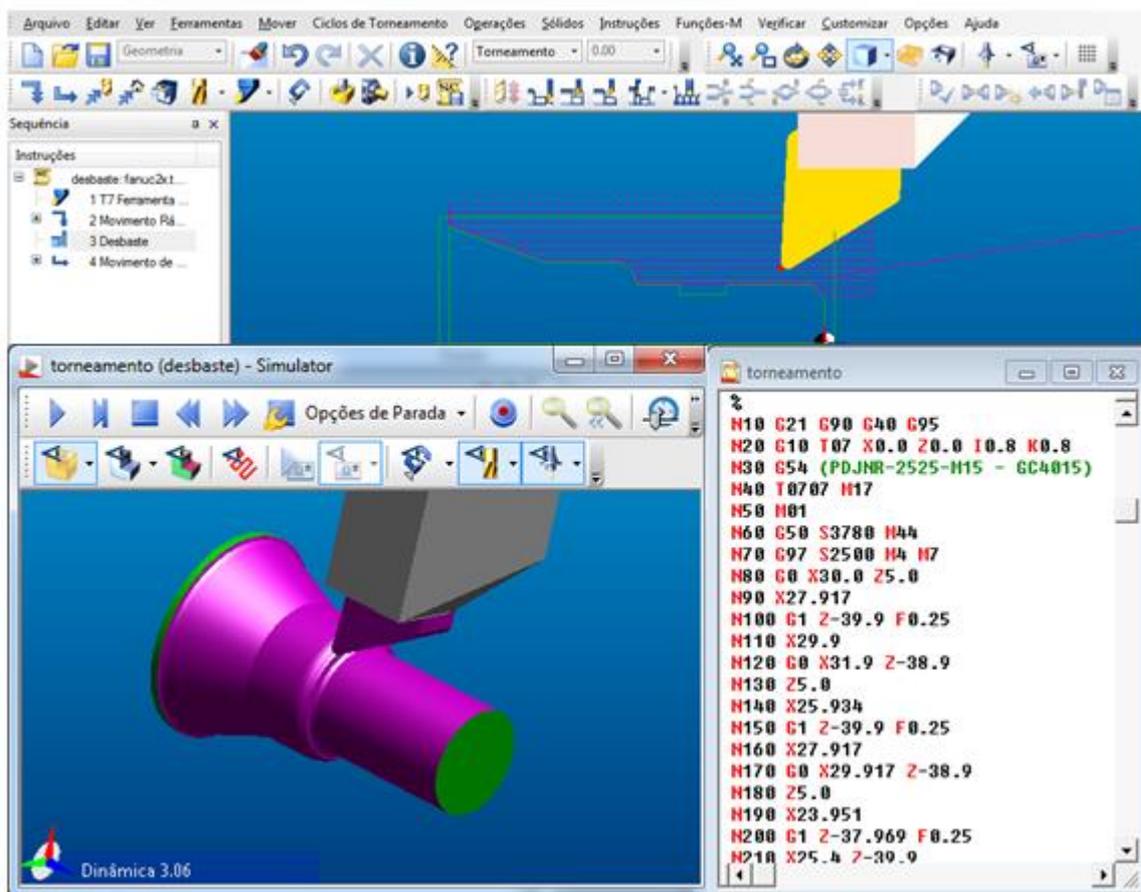


Figura 2.4 - Exemplo de sistema CAD/CAM [ECS, 2011]

2.6 Modelos de sistemas CAD/CAM

Um sistema CAM tem como base primária as informações provenientes de um desenho gerado em plataforma CAD e armazenado em um determinado formato padrão que permita sua importação, para processamento em ambiente integrado ou independente.

O modo como essa interação é implementada para atingir o objetivo principal do CAM, que é gerar as instruções de fabricação, pode ocorrer segundo diferentes abordagens,

seja em nível de automação ou interações com o usuário do sistema. A Figura 2.5 ilustra uma síntese de diferentes abordagens de sistemas.

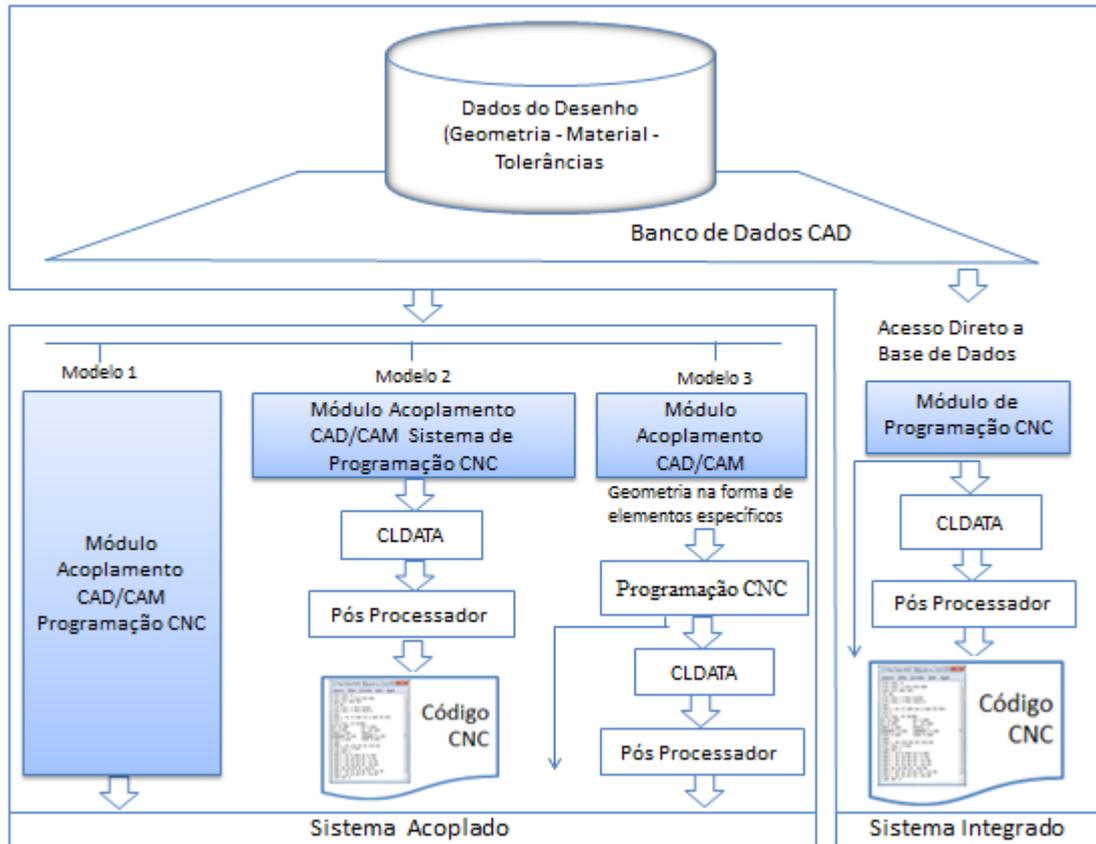


Figura 2.5 - Modelos de programação CAD/CAM [Adaptada de Gunnar, 2002]

A Figura 2.5 ilustra alguns modelos de programação CAD/CAM, tendo por definição um banco de dados onde se encontram os projetos em forma de desenhos CAD que carregam informações como a geometria da peça, material e dimensões.

No primeiro modelo, o sistema é independente do processo CAD, acoplando-se ao arquivo que armazena as informações geométricas necessárias para a análise do projeto. Gera o programa CNC em forma de arquivo CLDATA que é traduzido pelo pós-processador para o código final, tudo em um mesmo programa. O modelo acoplado representa um sistema único, independente das plataformas CAD, possuindo um banco de dados com os recursos dos comandos específicos, podendo gerar os códigos para vários comandos.

No segundo modelo, existe um sistema de interpretação e acoplamento com o sistema CAD, onde são obtidas informações geométricas. Com esses dados o sistema gera o arquivo CLDATA. Um programa independente de pós-processamento gera o código CNC para a

máquina. O modelo representa a maioria dos sistemas disponíveis comercialmente, que necessitam de um pós-processador referente ao comando utilizado, caso contrário, o programa não funcionará no comando da máquina.

No terceiro modelo, os sistemas são todos independentes. No momento da criação do programa CNC o resultado pode ser o código para a máquina ou um arquivo CLDATA.

A geração do programa CNC pode ser integrada ao sistema CAD, onde é utilizada a linguagem nativa do sistema, aproveitando a plataforma gráfica disponível para facilitar a obtenção das características geométricas do projeto. O algoritmo desenvolvido é responsável por gerar o código CNC através de um arquivo CLDATA. O pós-processador escreve o código CNC correspondente ao comando da máquina. O modo integrado corresponde, por exemplo, a sistemas gerados no software *AutoCAD*, com linguagem de programação *AutoLisp*. Essa linguagem, disponível no ambiente, facilita a obtenção de componentes e ferramentas gráficas.

2.7 Programação Manual x CAD/CAM

A programação manual possibilita desenvolver um programa mais qualificado, com redução de blocos de programação [Cassaniga, 2005]. O longo tempo de desenvolvimento do programa é um dos fatores negativos dessa técnica, que é justificada quando se busca uma programação para uma usinagem otimizada. A Tabela 2.2 mostra uma comparação entre as duas técnicas de programação, para determinada demanda de peças usinadas.

Tabela 2.2 - Comparações entre Programação Manual e CAM [Adaptada de Cassaniga, 2005]

Tipo de Processo	Lote Anual	Tipo de Programação	Tempo de Programação hora/lote	Tempo de Usinagem hora/peça	Tempo Total Programação + Usinagem hora/lote	Resultado
Produção Seriada	10000	MANUAL	40,00	0,10	1040,00	BOM
Produção Seriada	10000	CAM	1,00	0,20	2001,00	RUIM
Ferramentaria	1	MANUAL	80,00	6,00	86,00	RUIM
Ferramentaria	1	CAM	1,00	9,00	10,00	BOM

Se a necessidade for gerar rapidamente o programa sem a preocupação da otimização do processo ou onde exista uma complexidade da superfície geométrica ou os lotes são tão pequenos que o tempo de programação passa a ser mais importante que os tempos de usinagem, opta-se pela programação gerada pelo sistema CAD/CAM.

2.8 Programação STEP-NC

Uma nova opção de programação para comandos CNC, o STEP-NC, é desenvolvida atualmente pela ISO 14649, com o propósito de prover um padrão consistente e de qualidade para a fabricação. Ao contrário da ISO 6983, conhecida pelos códigos “G/M”, o STEP-NC não é um método de programação, nem descreve os movimentos da ferramenta para uma máquina CNC, mas sim um modelo de dados orientado a objetos para CNC [Rosso e Newman, 2003]. A Figura 2.6 mostra um comparativo do sistema “G/M” e STEP-NC.

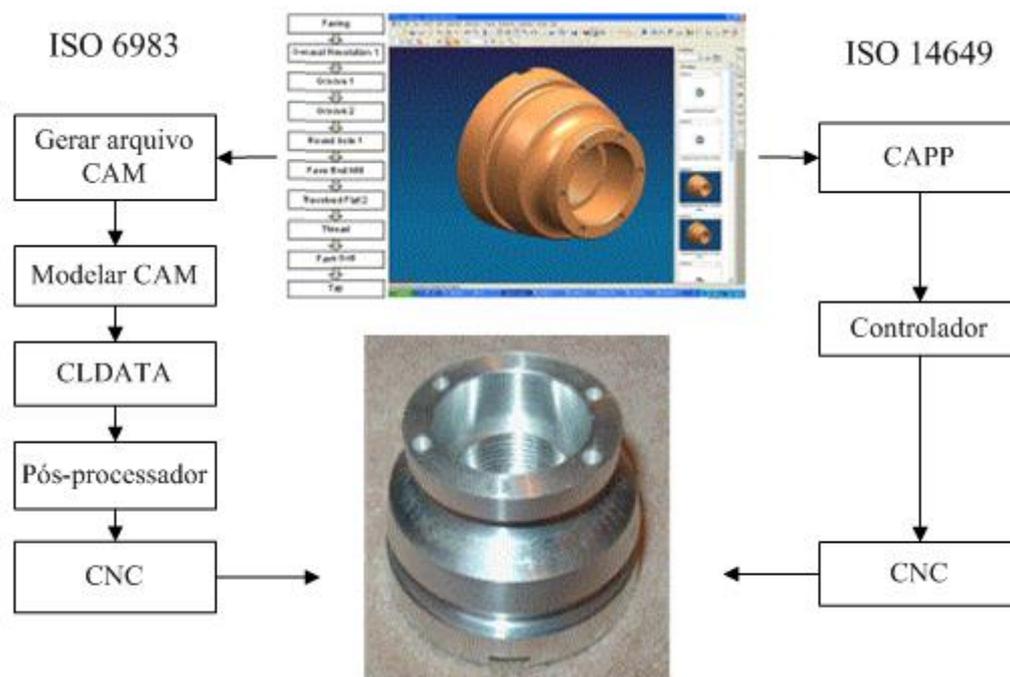


Figura 2.6 - Comparativo do sistema “G/M” e STEP-NC [Adaptada de Yusri e Keith, 2010]

O STEP-NC possui uma estrutura detalhada de interface de dados que incorpora a programação baseada em *features*, onde há uma gama de informações tais como: a *feature* a ser usinada, os tipos de ferramentas, as operações a realizar e o plano de trabalho. Para cada operação a realizar sobre uma ou mais *features*, é definida uma instrução chamada *Workingstep*. Esses *Workingsteps* fornecem as bases para o funcionamento dos planos de trabalho definidos em instruções chamadas *Workplan* que definem a fabricação do componente. A Figura 2.7 mostra um esquema geral do STEP-NC.

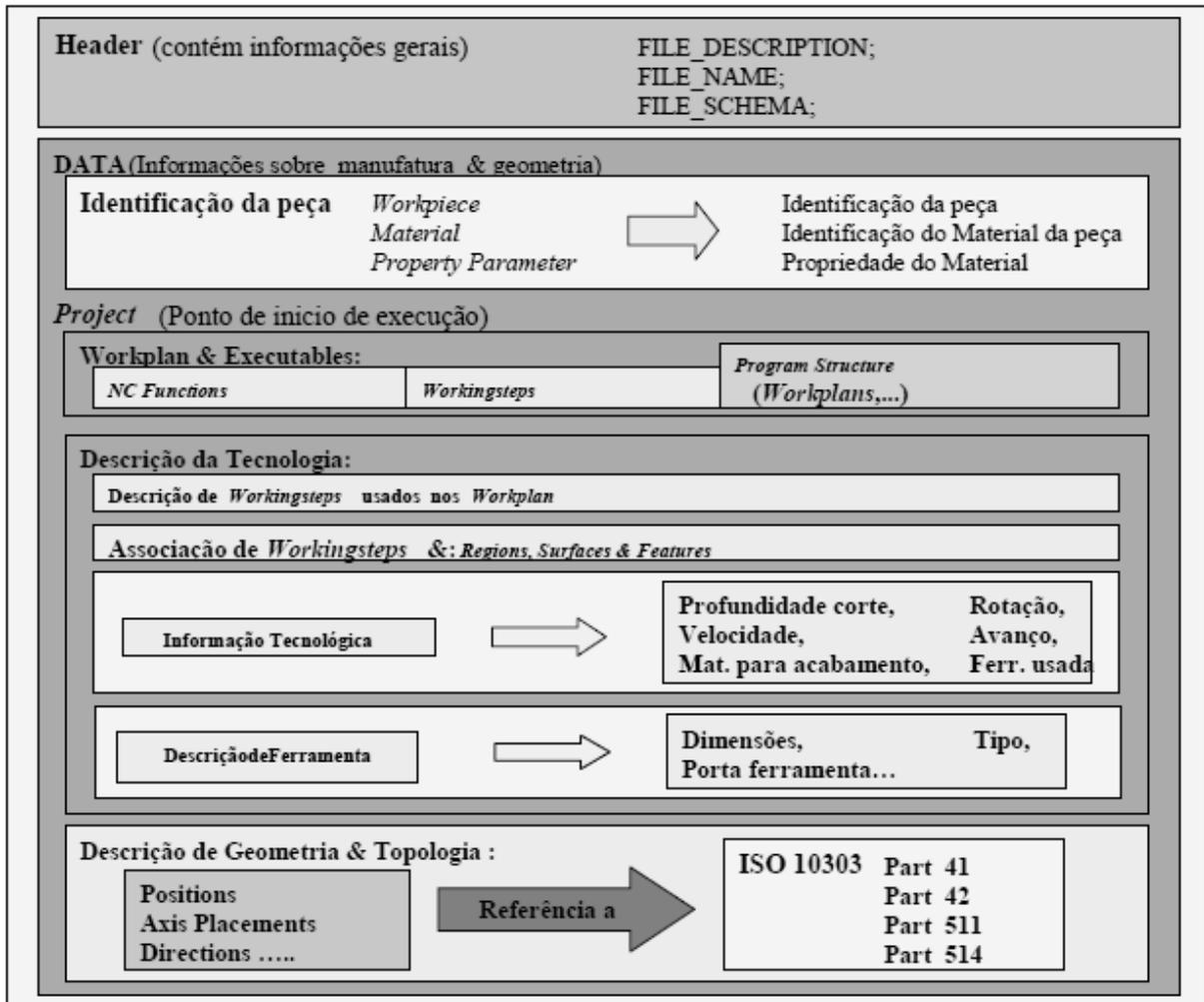


Figura 2.7 - Esquema geral do STEP-NC [Adaptada de Newman e Nassehi, 2007]

2.9 Programação manual avançada

A programação parametrizada pode ser comparada às linguagens de programação como *BASIC*, *C++* ou *PASCAL* [Lynch, 1997a]. Porém, essa linguagem de programação reside no controle do CNC, podendo combinar técnicas de programação manuais com técnicas de programação paramétricas. Características relacionadas aos computadores como: variáveis, aritmética e declarações de lógica estão disponíveis nessa linguagem. Como todas as linguagens de programação, a parametrizada possui várias versões. A mais popular é *Custom Macro B* (usado pela *Fanuc* e controles compatíveis). Outros incluem *User Task* (*Okuma*), *Q Routine* (*Sodick*) e linguagem de programação avançada. A Figura 2.8 mostra um comparativo entre programa gerado em CAD/CAM e programação parametrizada para vários modelos.

GEOMETRIAS	CAD/CAM		PARAMETRIZADO
	X40.Y60.Z20. ap=2. 172 linhas	ap=0.5 607 linhas	Programado com 13 linhas
	X80.Y30.Z53. ap=2. 1481 linhas	ap=0.5 6052 linhas	Programado com 15 linhas
	R50.C25.Z10. ap=2. 645 linhas	ap=0.5 2462 linhas	Programado com 17 linhas
	R200.C30.A0.B360. ap=2. 993 linhas	ap=0.5 4130 linhas	Programado com 23 linhas
	R50.A0.B360.H30. ap=2. 433 linhas	ap=0.5 1778 linhas	Programado com 13 linhas
	D40.Y100.Z30. ap=2. 1918 linhas	ap=0.5 7747 linhas	Programado com 17 linhas

Figura 2.8 - Comparativo programação CAD/CAM e Parametrizada [Gonçalves, 2007]

Além das muitas rotinas relacionadas ao computador, a maioria das versões de programação paramétrica tem rotinas relacionadas ao CNC com relativa profundidade. Por exemplo, macros que permitem ao usuário ter acesso a diversas propriedades do controle CNC (compensação de ferramenta, posicionamento de eixo, alarmes, geração e edição de código e proteção de programa) que permite a edição interna do programa.

2.10 Trajetória da ferramenta

Para a representação de qualquer geometria em sistema CAD e de controle e fabricação CAx, são aplicadas formulações matemáticas para desenvolver suas funções. A matemática clássica não é adequada para a representação de geometrias em sistemas computacionais, surgindo novos recursos de cálculos como os polinômios paramétricos [Souza e Ulbrich, 2009].

Matematicamente sugere-se denominar por funções *spline* o equacionamento capaz de representar uma curva em um sistema computacional, utilizando uma equação polinomial

paramétrica. As funções *spline* podem ser classificadas conforme seu desenvolvimento: curva de *Bézier*, *B-Spline*, *Spline de Alkima*, *NURBS (Non-Uniform Rational B-Spline)*, polinômios com diferentes graus de equação, dentre outros.

Esses modelos matemáticos foram desenvolvidos inicialmente por Pierre Bézier. Baseado em formulações inicialmente propostas por Charles Hermite, Bézier implementou o *software Unisurf*, em 1972, com o objetivo de auxiliar a fabricação de painéis de automóveis produzidos pela empresa *Renault*. Essa foi a primeira utilização de sistemas computacionais para modelamento de superfícies em projetos mecânicos. Atualmente, a formulação proposta por Bézier sofreu modificações, surgindo os modelos *B-Spline* e a mais recente *NURBS*, permitindo maior manipulação e controle da curva ou superfície gerada e conseqüentemente maior versatilidade na representação de formas complexas.

Curvas *Spline* estão sendo utilizadas com sucesso há algumas décadas para representar curvas e superfícies complexas em sistemas CAD. Uma vertente de desenvolvimento atual é aplicar uma *Spline* para descrever trajetórias complexas em ferramentas, substituindo a tradicional interpolação linear de segmentos de reta [Souza e Ulbrich, 2009].

O cálculo da trajetória da ferramenta deve ser considerado a principal função do software CAM, pois o resultado da operação de fabricação está diretamente relacionado com a qualidade da trajetória calculada, implicando alterações na superfície usinada, erros dimensionais e tempos de cálculo de usinagem.

Os algoritmos matemáticos e as metodologias para cálculo das trajetórias da ferramenta são criados em cada empresa desenvolvedora de sistemas CAM, e representam segredo industrial. O método tradicional para descrever uma trajetória de ferramenta para a usinagem de uma superfície complexa é a interpolação linear de segmentos de retas, de acordo com a norma DIN 66025 (*German Institute for Standardization*).

O CAM ajusta a trajetória da ferramenta dentro de uma banda de tolerância definida pelo usuário. Quanto menor a banda, mais próximo a ferramenta está da geometria CAD. A Figura 2.9 mostra a influência da curvatura da superfície dentro de uma mesma banda de tolerância.

Quanto menores as bandas de tolerâncias, mais próximas estarão as trajetórias da ferramenta da superfície a ser usinada, entretanto, menor serão os segmentos de retas. Os programas CNC possuirão maior volume de dados, pois mais linhas comporão o programa e essa característica poderá limitar as velocidades de avanço durante a usinagem, em especial de forma complexa. A máquina reduz a velocidade de avanço programada, como estudado por Souza, 2004, Souza e Coelho, 2007.

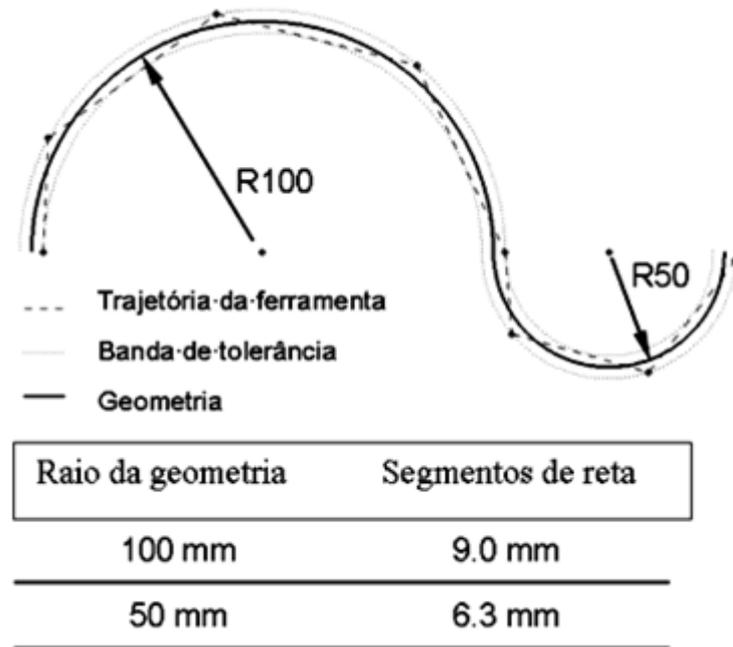


Figura 2.9 - Tolerância para cálculo dos programas CNC [Souza, 2004]

Devido às inconveniências do tradicional método de interpolação linear, Figura 2.10(a), para usinagem de forma complexa, uma alternativa é utilizar a interpolação circular associada com a interpolação linear, Figura 2.10(b), para descrever uma trajetória de ferramenta. Essa combinação pode manter a continuidade da curva e demonstra muitas vantagens sobre a utilização isolada de segmentos de reta [Souza, 2004]. Esse método só é aplicado para movimentos planares, interpolando dois eixos da máquina CNC, uma vez que a grande maioria dos sistemas CAD/CAM/CNC não tem recursos para realizar interpolação circular em três dimensões. Dessa forma, tem-se por objetivo utilizar polinômios *spline* para descrever trajetórias complexas de ferramenta. Esse método de interpolação faz uso de modelos matemáticos utilizados por sistemas CAD atuais para a representação de curvas e superfícies complexas. A possibilidade e as vantagens de utilizar uma *spline* para representar trajetórias complexas de ferramenta, tem como consequência a geração de programa CNC que não irá conter os comandos tradicionais “G01, G02 ou G03”, mas uma nova sintaxe que representará uma trajetória complexa. A Figura 2.10(c) ilustra a trajetória de ferramenta e o polígono de controle que define a curva *spline*. Os pontos cartesianos são utilizados para definir o polígono [Souza, 2004].

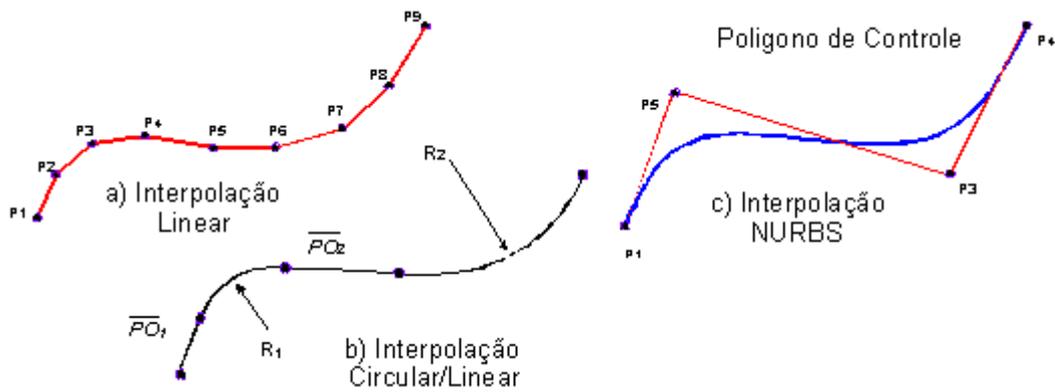


Figura 2.10 - Interpolações, (a) linear, (b) linear e circular, (c) polinomial [Souza, 2004]

A Figura 2.11 mostra as três metodologias de interpolação, normalmente usadas por sistemas CAM, aplicadas em uma banda de tolerância.

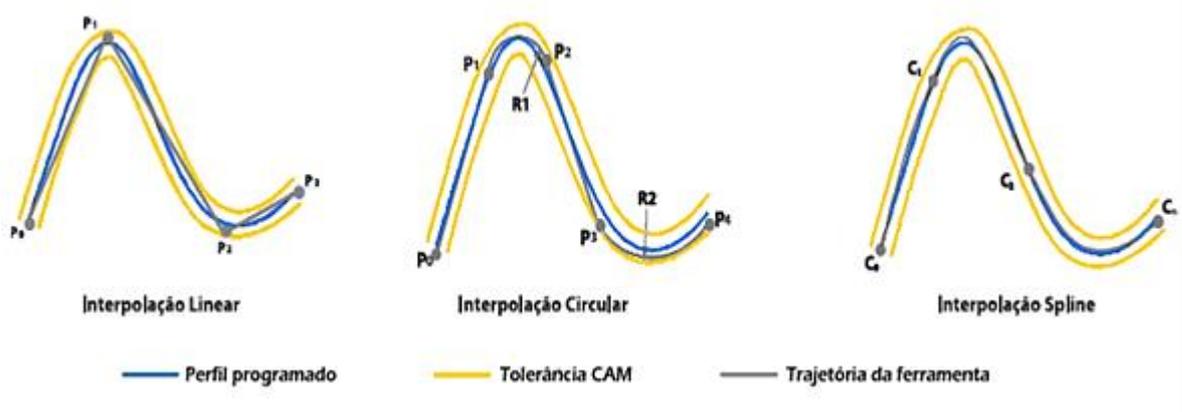


Figura 2.11 - Interpolações normalmente usadas por CAM [Schützer e Helleno, 2004]

Existem diferentes sintaxes para as entradas de dados *Spline*, conforme o comando numérico utilizado. Um exemplo para comando *Fanuc* é apresentado a seguir.

N500 G6.2 P K X Y Z R

Onde:

[P] – Sintaxe para Interpolação NURBS

[K] – Grau de equação polinomial

[X, Y, Z] – Coordenadas do ponto de controle

[R] – Peso do ponto de controle

Os sistemas CAM que permitem a geração de programas CNC utilizando polinômios *Spline* não o fazem diretamente. Esses são gerados partindo de um programa CNC linear de segmentos de retas. É utilizada a tolerância inicial para cálculo do programa linear adicionada à tolerância de cálculo do programa *Spline* [Souza, 2004; Souza e Coelho, 2007].

2.11 Pós-processador e sistemas aplicativos

Inicialmente um pós-processador era considerado uma ferramenta de interface entre a fabricação auxiliada por computador e os sistemas de controle numérico de máquinas, um simples tradutor, lendo as instruções de fabricação emitidas a partir de um sistema CAM e escrevendo em uma versão apropriada para uma máquina CNC. Hoje, porém, o pós-processamento evoluiu para incluir uma gama dinâmica de ferramentas que são responsáveis por emitir um código de máquina mais eficiente e produtivo possível [ITC, 2011].

A maioria dos sistemas CAM gera um ou mais tipos de arquivos de linguagem neutra que contêm instruções para uma máquina CNC. Esses estão em um formato binário chamado CLDATA ou algum formato ASCII (*American Standard Code for Information Interchange*), o qual é legível e geralmente escrito em linguagem APT. O formato ASCII aceita definições, geometrias simbólicas e instruções de usinagem, e gera o código CLDATA que descreve uma operação de usinagem passo a passo em condições absolutas. Alguns sistemas CAM possuem certo grau de flexibilidade, permitindo incluir procedimento no arquivo neutro, outros são bastante rígidos sobre o que pode e não pode ser incluído. O pós-processador é o software responsável por traduzir instruções neutras do sistema CAM para as instruções específicas requeridas pela máquina CNC. Esse software precisa responder às exigências e limitações do sistema CAM do comando da máquina CNC e do ambiente industrial [ITC, 2011].

Pós-processadores podem realizar outras tarefas, além de traduzir o código CLDATA ao código da máquina CNC. Por exemplo, um pós-processador pode limitar a alimentação e velocidade de pós-processamento, a qualidade da informação pós-processada, podendo assim minimizar o uso dos recursos disponíveis do CAM ou do CNC. Porém, pós-processadores mais sofisticados podem validar o programa antes de serem executados na máquina CNC. Há muitas regras simples que um pós-processador pode seguir, como colocar mensagens de advertência, que seriam exibidas quando regras são violadas, por exemplo:

- verificar se uma ferramenta não é selecionada próximo do início do programa;
- advertir quando movimentos de corte são implementados com fuso parado;
- sinalizar longa série de movimentos de posicionamentos;

- avisar que a ferramenta está fora do plano de usinagem;
- alertar se não forem ligadas as compensações de comprimento ou diâmetro quando forem solicitadas para uma ferramenta;
- comunicar a ocorrência ciclos fixos ativos durante uma mudança de ferramenta (eles deveriam ser temporariamente cancelados).

Pós-processadores podem auxiliar na correção de limitações e *bugs* do sistema CAM ou do comando da máquina CNC. Geralmente é muito mais fácil mudar o pós-processador do que implementar uma revisão do sistema CAM ou no comando da máquina CNC [ITC, 2011]. Os melhores códigos pós-processados são transparentes, em outras palavras, são aqueles que o usuário não toma conhecimento sobre a complexidade dos cálculos e ações que estão ocorrendo por trás dos mesmos. Os pós-processadores geram no ambiente CAM o programa CNC com base na geometria da peça, nas configurações de estratégias de usinagem, nas ferramentas e definições da máquina Figura 2.12.

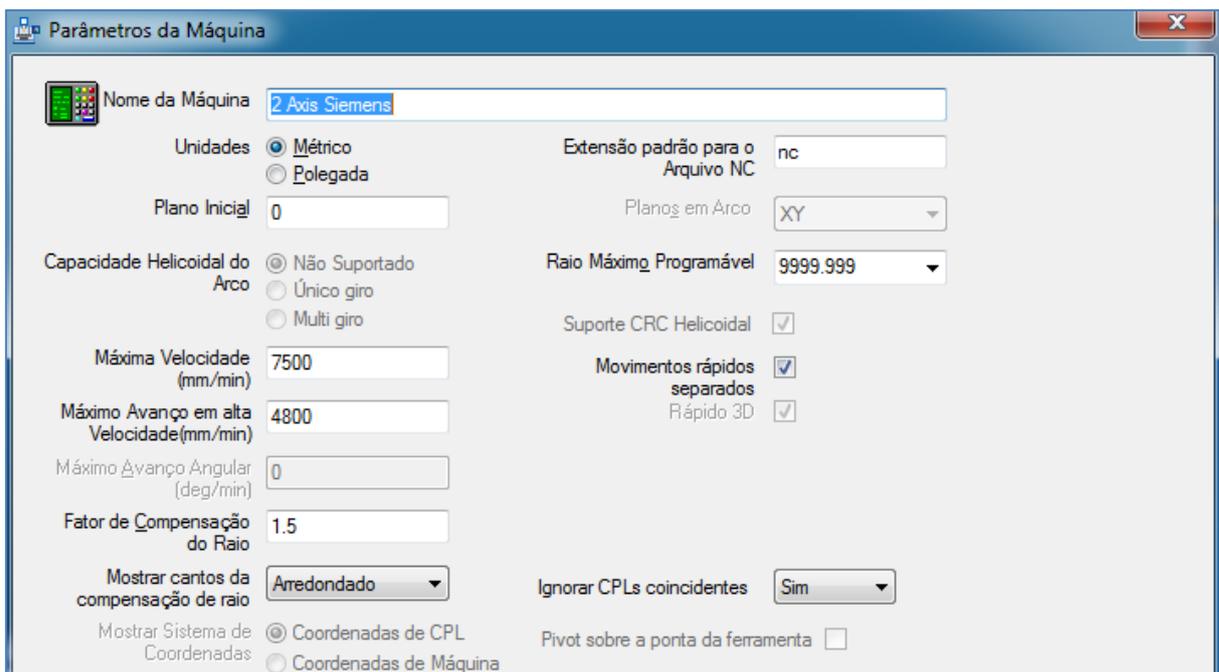


Figura 2.12 - Assistente de códigos para desenvolvimento do pós-processador [ECS, 2011]

Dependendo do tipo de software CAM, o pós-processador pode ser configurado através de um arquivo de parâmetros estabelecendo especificações para cada tipo de máquina. Esses arquivos são os responsáveis para que cada programa tenha características específicas da configuração do comando da máquina em cada empresa.

Na maioria dos softwares CAM, a linguagem de programação dos pós-processadores é exclusiva e não possui um padrão, levando os desenvolvedores dos pós-processadores a fazerem adaptações de exemplos desenvolvidos pelo fabricante. Essa técnica é dificultada por exigir grande conhecimento de lógica de programação e diferentes linguagens de programação. A Figura 2.13 mostra um trecho de pós-processador.

```

COMMENTSTART      :(
COMMENTEND        :)

***** EXAMPLES *****
VARIABLE : FECHAPORTA FECHAPORTA1 ROSCAPOSICAO ROSCAPOSICAO1
VARIABLE : Code
VARIABLE : TailStockon  ### 1= Tailstock on, 0= off
VARIABLE : ForceF      ### Used as a flag to force feeds

DEFINE Aux1      Nextclfile(631)

DEFINE OutputF   IF (ForceF) \\
                  IF (nextmisc(40) = 1) F__ * ENDIF \\
                  IF (nextmisc(40) = 2) F*  ENDIF \\
ELSE \\
                  IF (nextmisc(40) = 1) F__ ENDIF \\
                  IF (nextmisc(40) = 2) F  ENDIF \\
ENDIF

EX_FIRSTRAPID   : REMOVESPACESON
                  : "G00 X Z C
                  : REMOVESPACEOFF

EX_RAPID        : REMOVESPACESON
                  : "G00 X Z C
                  : REMOVESPACEOFF

EX_LINEAR       : REMOVESPACESON
                  : "FEEDUNIT G01 X Z FEEDRATECODE
                  : REMOVESPACEOFF

EX_CIRCLEZX    : REMOVESPACESON
                  : "FEEDUNIT CIRCLEDIRECTION X Z C K R* FEEDRATECODE
                  : REMOVESPACEOFF

```

Figura 2.13 - Pós-processador [Gonçalves, 2007]

O programa CNC para executar uma operação de furação, em diferentes comandos, necessita de pós-processadores respectivos a esses comandos. A Tabela 2.3 ilustra o programa para comandos *Fanuc* e *Siemens*, os quais correspondem à mesma operação de furação, mas cada comando entende de maneira diferente as instruções.

Tabela 2.3 - Programa para FANUC e SIEMENS para furação [ECS, 2011]

Comando FANUC	Comando SIEMENS
N10 G21 G40 G80 G90 G99	N10
N20 T0101	N30 M01
N30 G54	N40 T0101
N40 G97 S545 M3	N70 G54 D1
N50 M8	M90 G97 S600 M3
N60 G00 Z5.0	M100 G0 X0 Z5 M8
N70 X0.0	N110 CYCLE83
N80 G83 Z-44.507 R5.0 F1.0	(5,,1, -44.507,,0,,,0,0, F1.0,1)
N90 M30	N120 M30

Liu *et al.*, 2008, relatam que a maioria dos comandos obedecem a um programa específico. Os códigos gerados para um comando podem gerar conflito na sua reutilização em outras máquinas, com comandos diferentes. Essa situação é formada devido a alguns fatores:

- Nenhum padrão real de CNC está disponível, apesar de algumas normas existirem, RS274D (EUA), ISO (6983) e DIN (66025). No entanto as máquinas e as tecnologias de comandos têm sofrido grande desenvolvimento, sendo que novas funções e características incorporadas não são compatíveis com esses padrões.
- Os fabricantes de comandos CNC aumentaram o programa padrão para adaptar as suas novas funções específicas. Isso dificulta o intercâmbio dos programas gerados por diferentes comandos. Assim, mesmo o usuário que possui vários tipos de máquinas do mesmo fabricante, mas com diferentes comandos, tem que gerar diferentes programas para cada máquina.
- Para a reutilização de programas CNC em outros comandos são necessárias modificações, processo que geralmente requer conhecimento especializado em diferentes tipos de comandos, exigindo maior tempo na formação de programadores e operadores.
- Um CAM pode gerar diferentes programas através de diferentes pós-processadores. No entanto, não é viável para os casos de programação manual, que é de grande aplicação na indústria. Os sistemas necessitam de diferentes pós-processadores e o sistema CAM tem que ser preparado com o pós-processador com antecedência.

Verificando a dificuldade de reutilização de programas em diferentes comandos CNC, Liu *et al.*, 2008, realizaram estudos para elaboração de um algoritmo de processamento designado de processador de programa CNC (CNPP – *Numerical Control Program Processor*), ele funciona junto com os módulos do comando da máquina, podendo interpretar

variados programas, gerados por diferentes comandos. O projeto é composto por um motor de processamento e várias bibliotecas de comandos CNC.

Quando o programa é inserido nesse módulo, é feita uma análise sintática e semântica, usando o sistema EBNF (*Extended Backus Naur For*), que é uma metalinguagem para descrever uma linguagem de programação formal. Essa análise utiliza as bibliotecas dos comandos catalogados. Com o processamento realizado, é gerado o código final com funções canônicas de usinagem, desenvolvidas pelo NIST (*National Institute Standards and Technology*). A Figura 2.14 mostra a interface do CNPP.

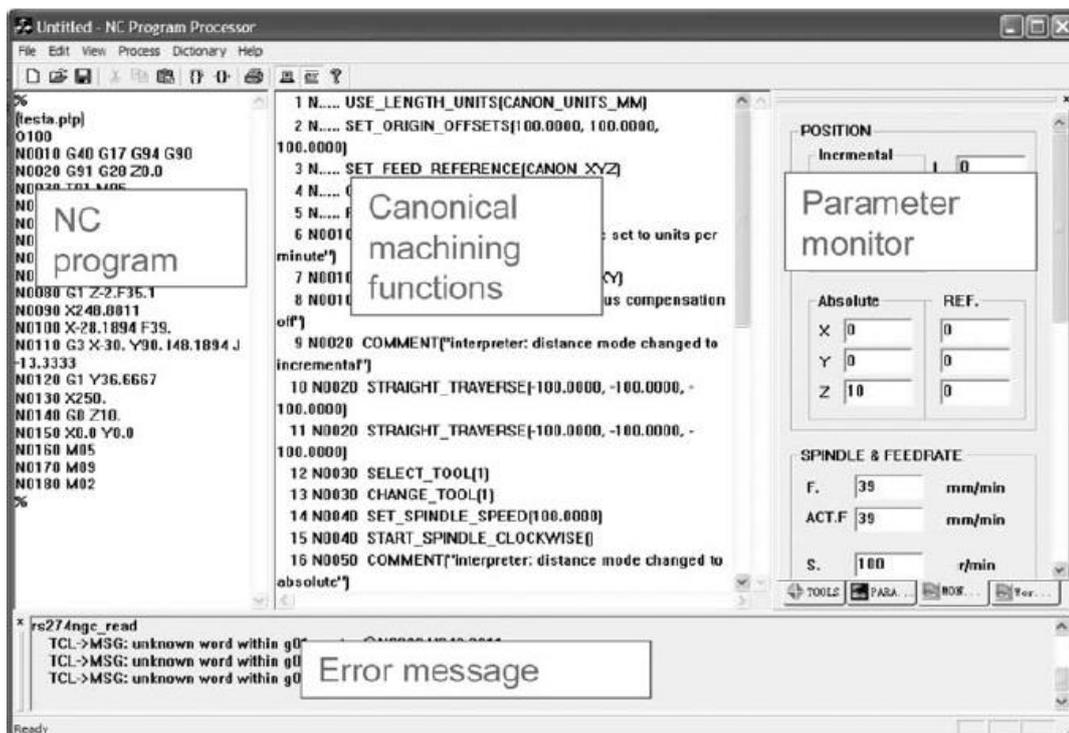


Figura 2.14 - Interface do programa CNPP [Liu *et al.*, 2008]

Cheng e Jihong, 2011, propuseram um sistema de furação automatizado para a indústria aeronáutica. Possuindo uma diversidade de sistemas CNC no processo, o aplicativo CAD/CAM utilizado não atendia totalmente as necessidades, gerando conflitos e perda nas modificações dos programas para sua reutilização. Desenvolveram um pós-processador CNC específico para qualificar e otimizar o trabalho de furação da asa de aviões com uma máquina de 5 eixos.

Os projetos foram gerados pelo software CAD/CAM comercial CATIA, tendo uma precisão de medidas por leitura ótica. O programa final gerado pelo CAM descreve as trajetórias da ferramenta e as informações tecnológicas em formato APT, em que o pós-

processador proposto traduz para código CNC, específico para cada sistema, adequando todas as especificações e normas da gerência de projeto. A Tabela 2.4 mostra uma relação entre APT e CNC.

Tabela 2.4 - Relação Entre APT e CNC [Cheng e Jihong, 2011]

APT	CNC
FROM/x,y,z,a,b,c	G00
GOTO/x,y,z,a,b,c	G01
FEDRAT/n	F
CUTTER/d,r,e,f	T
SPINDL/n,cw/OFF	S-M03 M05
RAPID	G00
STOP	M0
FINI	M30

Zhang *et al.*, 2011, realizaram estudos na implementação de um sistema de planejamento de processo, baseado na norma STEP-NC para torneamento cilíndrico, tendo como parâmetro de análise a rugosidade. O algoritmo proposto possui um interpretador de dados, planejador do processo e o gerador do programa. Uma das características do STEP-NC é absorver informações do chão-de-fábrica. O parâmetro de rugosidade é a informação que estabelece a configuração do programa, conforme informações recebidas de formulação matemática, o programa é ajustado para melhorar a eficiência. Com essa técnica se elimina a necessidade de pós-processador que é mais uma característica desta linguagem.

Nunes *et al.*, 2008, estudaram a influência da trajetória da ferramenta na usinagem, gerada automaticamente por sistemas CAD/CAM. Comparando interpolações lineares com funções *Splines*, concluíram que há uma maior eficiência com menores tempos de usinagem, para altas velocidades, quando empregado às funções *Splines*. Um estudo equivalente realizado por Melegari, 2011, comparando os programas gerados com interpolações lineares e programas com linguagem parametrizada, tem como conclusão um melhor desempenho para os programas gerados com linguagem parametrizada. Yau e Kuo, 2001, desenvolveram um aplicativo que converte programas CNC tradicionais gerados por interpolações lineares em programas com formato *spline*.

Zalewski, 2011, parte de um contexto de que a maioria dos comandos CNC são programados por CAM. Tem como características a diminuição dos tempos de preparação, utilização de estratégias eficazes de movimento de ferramenta, a otimização de parâmetros e

outras qualidades que influenciam a ferramenta e a qualidade da superfície do objeto processado.

O estudo é baseado na simulação gerada por esses sistemas que não levam em conta as características das máquinas em chão de fábrica porque seus parâmetros de configuração não correspondem àqueles adequados para a máquina.

O projeto CNC/CAM integrador funciona como *feedback* entre CNC e CAM. É composto por um conjunto de sensores e um programa, onde os mesmos recolhem dados sobre as mudanças e orientações da peça e da ferramenta no curso da execução do programa. O software arquiva os dados, compara com os dados teóricos que serviram para o desenvolvimento do programa do controle e envia os resultados para análise do CAM. O resultado final é uma simulação do CNC compatível com o trabalho do CNC no sistema produtivo, porque o processo de aprendizagem desenvolvido pelo CNC/CAM integrador pode ser feito até o nível pretendido de compatibilidade entre simulação e produção.

O projeto apresenta três estágios. O primeiro é a coleta de dados de configuração do CNC para as necessidades de definição do pós-processador, como movimento dos eixos, pontos característicos, parâmetros importantes, entre outros. O segundo destina-se a ensinar ao CAM as qualidades efetuadas em CNC. O terceiro utiliza a base de dados do CNC/CAM integrador para ajustar o novo CAM. A Figura 2.15 mostra o princípio de funcionamento do CNC/CAM integrador.

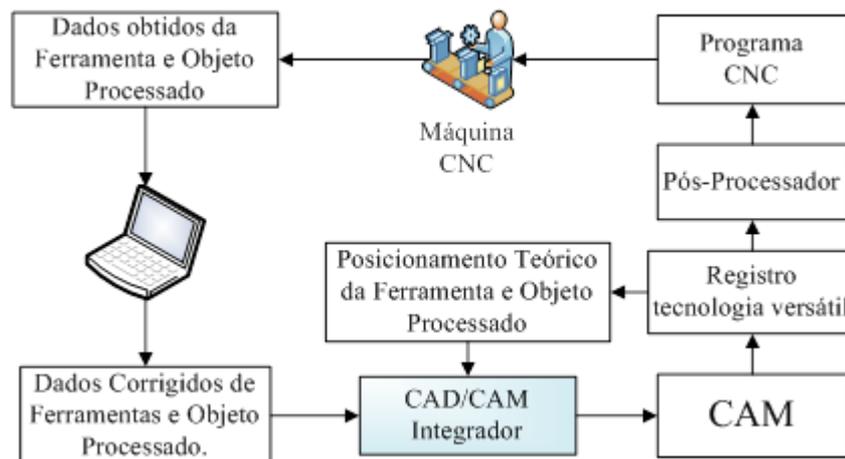


Figura 2.15 - CNC/CAM integrador [Adaptada de Zalewski, 2011]

Cheypoca *et al.*, 2010, verificam que no processo de geração de código CNC, realizado pelos sistemas CAM, são geralmente utilizados desenhos gerados nos sistemas CAD. A peça imagem pode ser obtida por outras ferramentas como de desenho ou imagens de

fotografias, que podem ser convertidas em formatos digitais por scanner de fotos ou câmeras fotográficas digitais, podendo ser exportados para arquivos de imagem de formato bitmap.

É proposto um pós-processador genérico para utilizar os dados de entrada oriundos de imagens *bitmap* e convertê-los em programação CNC. O primeiro módulo do software converte a imagem bitmap em sinais de nível “alto” e “baixo”, transformando o de “alto” em coordenadas “x, y”. A Figura 2.16 mostra um comparativo entre o processo normal e o proposto.

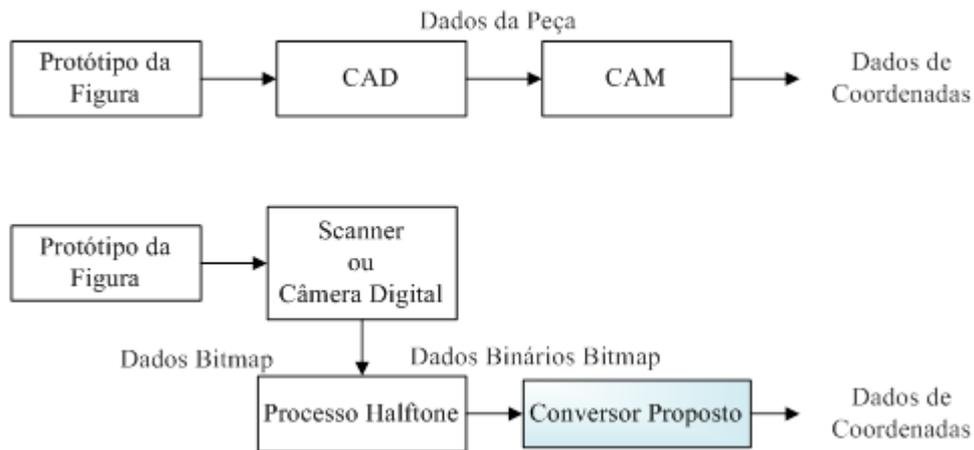


Figura 2.16 - Processo normal e proposto [Adaptada de Cheypoca *et al.*, 2010]

A Figura 2.17 ilustra a interface gráfica do pós-processador genérico.

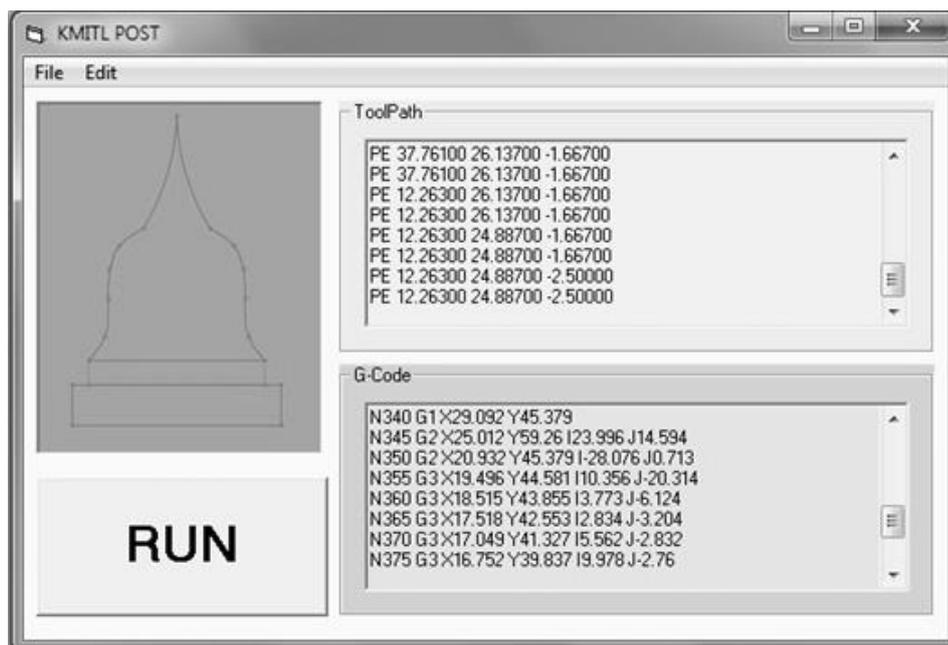


Figura 2.17 - Gerador de códigos [Cheypoca *et al.*, 2010]

O estudo é viabilizado pelo desenvolvimento de um algoritmo em *Visual C++*, que realiza o pós-processamento das informações convertidas do arquivo bitmap em linguagem de programação CNC. Os resultados relacionados a este trabalho mostram um menor uso da memória e conseqüentemente uma maior velocidade de processamento.

Wang e Su, 2010, desenvolveram um sistema de programação CNC gráfico, gerando os códigos automaticamente a partir de informações de contorno. Os módulos que compõem o sistema são a entrada de dados geométricos, o processamento matemático, o pós-processamento, a geração de código CNC e a plataforma de simulação gráfica.

O contorno da peça é extraído do arquivo DXF, oriundo de sistemas CAD. O módulo de processamento matemático ordena os dados oriundos do bloco entidades. A Figura 2.18 mostra o fluxograma para a leitura do arquivo DXF.

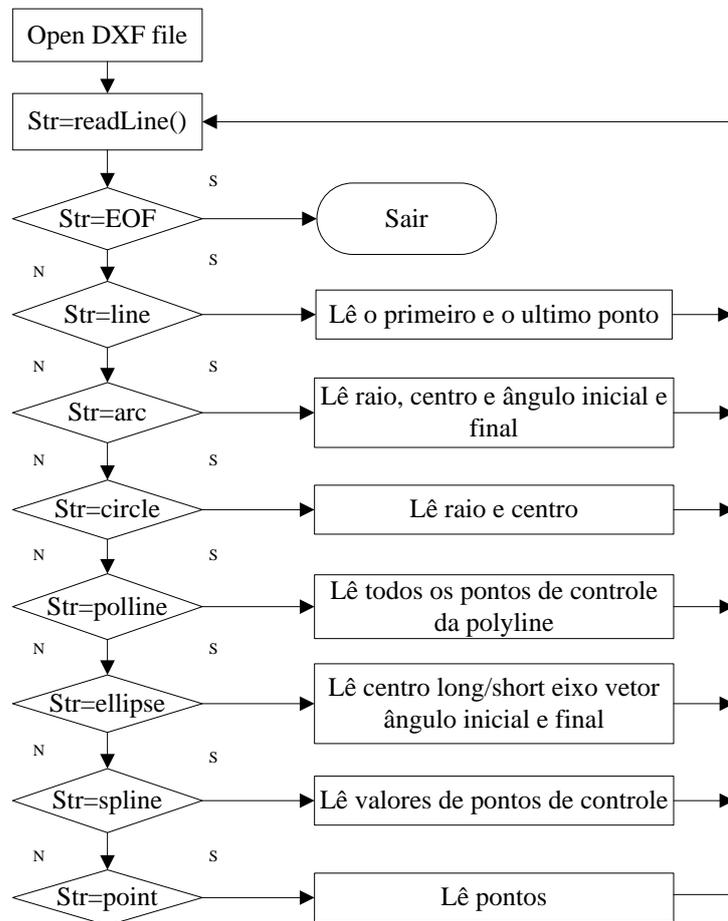


Figura 2.18 - Fluxograma de leitura DXF [Adaptada de Wang e Su., 2010]

A entrada dos dados geométricos pode ser por plataforma gráfica ou arquivo DXF, fornecidos pelo sistema CAD. O pré-processamento é realizado com parâmetros definidos pela interface homem-máquina. O módulo de programação CNC é utilizado para gerar o

código do arquivo CNC baseado no processamento dos parâmetros e entidades tratadas. O módulo de simulação pode mostrar o caminho da ferramenta dinamicamente em ambiente gráfico computacional.

Newman e Nassehi, 2007, verificaram que o avanço da fabricação com sistemas CNC resultou no desenvolvimento de grande variedade de sistemas e tecnologias relacionadas. Com os fabricantes tentando proteger a sua quota de mercado, com normas próprias, muitos sistemas não tem intercâmbio com outros. O resultado são ilhas de automação, que reduzem a fluidez de recursos de uma empresa. Para trocar informações entre essas ilhas é comum a utilização de pós-processadores.

Neste trabalho os autores fornecem uma visão de uma plataforma universal de produção que permita a transferência de informação sem fronteiras, e entre diferentes sistemas assistidos por computador (CAD, CAM, CAPP, etc.) conhecidos como CAx. A mais comum cadeia CAx utilizada na fabricação é mostrada na Figura 2.19.

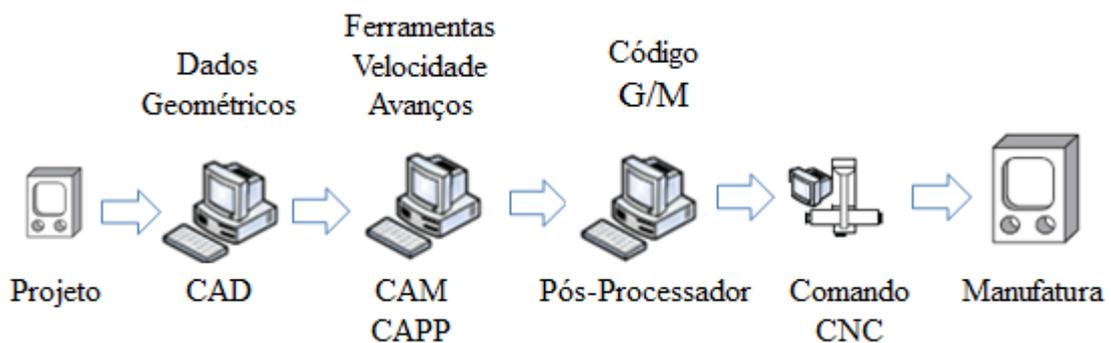


Figura 2.19 - Cadeia mais comum de produção CAx [Adaptado de Newman e Nasseli., 2007]

O projeto é transferido para um sistema assistido por computador, onde as informações geométricas sobre o componente são armazenadas em um arquivo CAD. Esse arquivo é transferido para um planejamento de processo assistido por computador CAPP ou uma fabricação auxiliada por computador CAM. Nessa etapa são inseridas informações auxiliares como definições de ferramentas, avanços, velocidades e estratégias de usinagem. Esses dados são armazenados em um formato proprietário de um arquivo CAM. Um pós-processador transforma dados do CAM para o CNC e o arquivo resultante é transferido para o comando.

Como o pós-processamento é um tradutor unidirecional, a integridade da informação transferida para a máquina pode ser comprometida logo que todas as modificações sejam feitas nos dados do controlador no chão-de-fábrica.

O fabricante do comando *Mazak* foi um dos primeiros a resolver esse problema através da introdução de *CAMWARE* (software de programação e simulação de usinagem de peças para as máquinas *Mazak*), unificando a abordagem para a cadeia CAx em que mesmo *off-line* o sistema de chão-de-fábrica utiliza a mesma interface de interação com o usuário.

Na cadeia atual, a informação é transferida através de pós-processadores e tradutores de forma unidirecional para o próximo sistema ao longo da cadeia. Em uma produção unificada, cada sistema CAx deve ser capaz de realizar o intercâmbio de informações com qualquer outro sistema CAx. A Figura 2.20 esquematiza os dois sistemas.

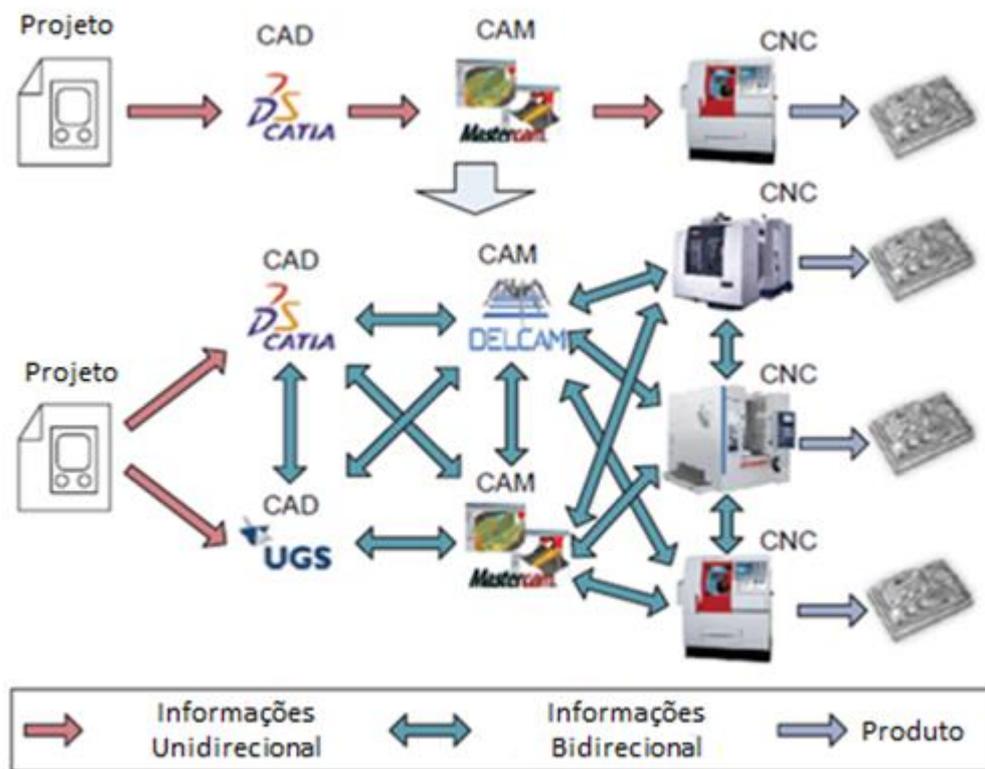


Figura 2.20 - Rede interoperável CAx [Adaptado de Newman e Nasseli., 2007]

Newman et al., 2007 propõem uma plataforma de suporte para produção universal para união dos CAx. A Figura 2.21 mostra a visão geral da plataforma universal.

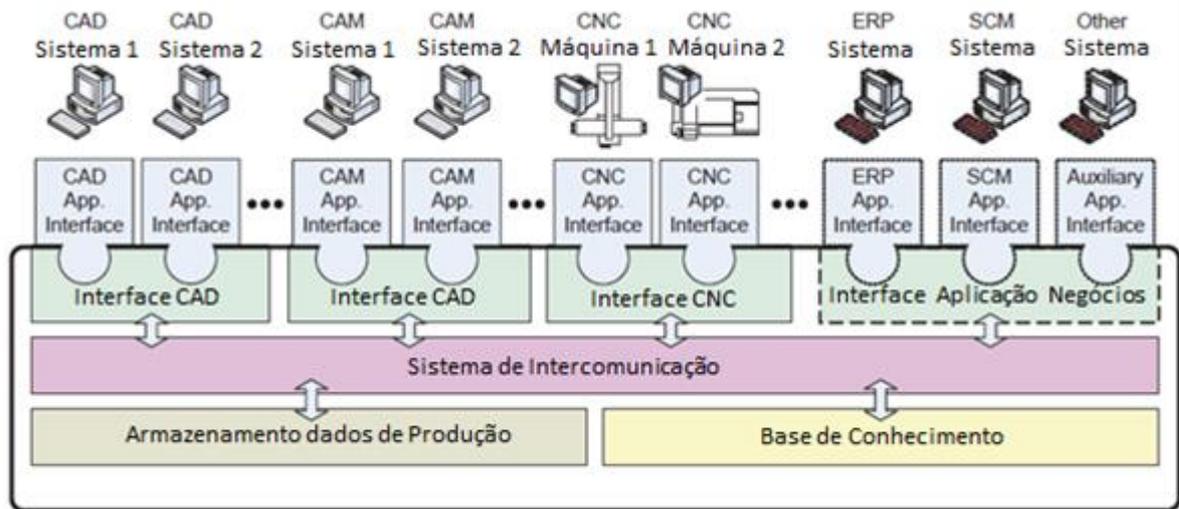


Figura 2.21 - Visão geral da plataforma universal [Adaptado de Newman e Nasseli., 2007]

Vários sistemas de comunicação, tais como estruturas web, podem ser usados para permitir que sistemas CAX tenham intercâmbio de informação com a plataforma. O *Intercommunication Bus* forma um agente móvel que permite que vários CAX se comuniquem uns com os outros, bem como com o *Warehouse*. Toda a transformação do conhecimento ocorre no domínio do padrão de *Data Warehouse*. Os níveis de interação podem ser realizados através do STEP NC, somente se todos fabricantes CAX fizerem o esforço de implementarem o padrão em seus produtos.

Yuliang *et al.*, 2010, mostram a criação de redes baseadas em intranet e sistemas de comunicação do código numérico CNC para o comando. O uso de máquinas-ferramenta comandadas numericamente na fabricação e a criação de redes para a integração dos comandos CNC tornaram-se uma tendência inevitável. As gestões atuais trabalham com documentos textos para interagir com os vários comandos. A proposta dos autores de integrar as máquinas por redes locais LAN (*local area network*) irá atingir a gestão automatizada, definindo um novo tipo de DNC (*Direct Numerical Control*). O sistema de gerenciamento CNC abrange o programa e o processamento da peça, bem como a programação para o comando da máquina. O sistema possui um banco de dados com biblioteca relativa às máquinas e aos procedimentos tecnológicos da peça. Cada estação de trabalho na LAN envia o serviço pedindo para o servidor DNC (servidor WEB) através de browser, solicitando informações de serviços de corte, de gestão de equipamento, de gerenciamento de documentos, etc. O servidor DNC finalmente envia o documento em formato HTML (*HyperText Markup Language*) de volta para a estação de trabalho.

Souza, 2007, desenvolveu ciclos avançados de usinagem com o objetivo de extrair pontos de entidades geométricas em 3D, definidas em CAD, no auxílio de geração automática de códigos CNC, para operações de desbaste e acabamento como um sistema CAD/CAM. Para essa alternativa pesquisada foi utilizada a linguagem *AutoLISP* e como plataforma a utilização do software *AutoCAD2000*. O programa CNC foi utilizado na usinagem de uma matriz associada a um sólido gerado por malhas de pontos, em um centro de usinagem CNC. Os resultados mostram que a utilização do sistema é viável para uso em pequenas e médias empresas. A justificativa da pesquisa dessa alternativa é relacionada ao custo considerado elevado de sistemas CAD/CAM, principalmente para pequenas e médias empresas, e também, à dificuldade de utilizá-los em setores industriais, pela carência de operadores qualificados. Superfícies de formas livres podem ser usinadas com os códigos CNC obtidos por pós-processamento. Isso leva, porém, à exigência de pequenas adaptações do modelo desenvolvido para aplicações em outros comandos numéricos de equipamentos CNC.

Canciglieri e Young, 2006, investigaram o potencial do sistema baseado no modelo do produto para representar múltiplas vistas no projeto orientado para a manufatura de produtos plásticos injetáveis e explorou, em particular, a vista da moldabilidade, do projeto do molde e da fabricação do molde. A abordagem explorou estruturas de informações particulares para auxiliar cada aplicativo do projeto orientado para a manufatura e um modelo de produto experimental foi implementado, utilizando um banco de dados orientado ao objeto. As atividades de projeto devem ser muito bem orquestradas e integradas, proporcionando a evolução da filosofia da Engenharia Concorrente, em que decisões sobre diferentes aspectos relacionados ao ciclo de vida do produto devem ser consideradas simultaneamente. Porém, quando todo o projeto orientado para a fabricação é considerado, existem muitos aspectos que devem ser dirigidos. Isso conduz à necessidade de sistemas de informações que sejam capazes de auxiliar múltiplas vistas do produto, onde cada vista provê a representação apropriada para auxiliar pelo menos uma perspectiva da manufatura.

Da Costa, 1997, descreve uma metodologia, na forma de um protótipo computacional, em que é possível a construção de um módulo especialista para escolha das condições de corte paralelamente às outras etapas do planejamento da usinagem. A metodologia tem um caráter generalista, permitindo ao usuário obter um sistema que contenha sua experiência pessoal e seja dirigido para um determinado chão-de-fábrica. A elaboração do trabalho é justificada devido às determinações das condições de usinagem raramente terem sido contempladas como um problema de planejamento, e, portanto, integrada às outras etapas. Observa-se que essa determinação vem sendo tratada isoladamente, como um problema de pesquisa

operacional (otimização de uma função objetivo) ou como uma simples consulta a um banco de dados.

Almeida, 2006, relata o desenvolvimento de um módulo computacional, acoplado a um software CNC para torneamento, que permita a predição de esforços de usinagem e rugosidade em tempo de programação CNC. Adota a técnica da redes neurais artificiais como núcleo, uma vez que permitem aproximações bastante satisfatórias do processo de torneamento. Os resultados indicam que os modelos de rede neurais artificiais adotados (multicamadas e função de base radial) aproximam de forma satisfatória o comportamento da força de corte e rugosidade, em função dos parâmetros de usinagem escolhidos (velocidade de corte, avanço e profundidade de corte) em uma série de casos de uso, utilizando-se o módulo computacional desenvolvido.

Rosso, 2005, propôs o desenvolvimento de um sistema de planejamento do processo de fabricação auxiliado por computador com base em novos padrões CNC que dê suporte à troca de informações entre CAD e CNC. Um novo modelo tem sido explorado para representar as informações necessárias, visando apoiar a atividade de planejamento de processo que utiliza a norma ISO 10303 STEP e ISO 14649 STEP-NC. O trabalho também enfoca a compreensão e desenvolvimento dos padrões CNC acima mencionados, com o objetivo de apoiar e integrar projeto assistido por computador, o processo de planejamento e a fabricação. A estrutura do modelo de produção é baseada em três classes, ou seja, a capacidade da *workstation*, a capacidade do processo e a capacidade de recursos. Esses fornecem a infraestrutura necessária para interoperabilidade de sistemas computacionais que suportam a cadeia do processo CAx. Os modelos foram testados, através de um estudo de caso, usando um protótipo computacional proposto, intitulado STEP-TM sistema CAPP. Esse protótipo mostra que essa abordagem de pesquisa oferece um potencial significativo para o planejamento de futuros processos interoperáveis de peças na geração de CAPP/CAM.

Souza, 2004, mostra uma revisão bibliográfica sobre usinagem HSC (*high speed cutting*), apresentando um estudo das forças de usinagem para fresamento de superfícies complexas e uma análise sobre as metodologias usadas em sistemas CAD/CAM e CNC para gerar e interpretar programas CNC que contenham movimentações complexas de ferramentas. Análises realizadas em um centro de usinagem HSC indicam que as limitações tecnológicas atuais na cadeia CAD/CAM/CNC restringem a usinagem de formas complexas com altas velocidades, reduzindo a velocidade de avanço programada. Ainda, essas análises apontam que as características intrínsecas desse processo de remoção de material produzem constantes alterações na força de usinagem.

Pereira, 2003, propôs o desenvolvimento de um sistema para edição e simulação de programas CNC específico para ambientes de manufatura, utilizando linguagem técnica com interface amigável e interativa, rodando em ambiente Windows. O aplicativo foi desenvolvido em Visual Basic e dividido em três módulos funcionais: editor, simulador de trajetória de ferramenta e transmissor para se efetuar a transferência do programa de usinagem do PC para a máquina CNC. O programa foi testado em ambiente fabril em simulação e usinagem de peças, gerando o código corretamente, com um nível de detalhamento na simulação superior ao do comando da máquina. Isso permitiu concluir que o protótipo poderá desempenhar um papel importante na tarefa de redução de tempos manuais em programação CNC.

Gonçalves, 2007, apresenta uma metodologia de aplicação dos recursos avançados de programação no desenvolvimento de rotinas parametrizadas dedicadas ao processo de fresamento. A programação parametrizada mostrou-se uma ferramenta eficaz para criar ciclos de usinagem a serem implementados como recursos adicionais às funções de programação disponíveis no comando da máquina. Por usar recursos tradicionais de programação, a técnica reduz o número de linhas de códigos em relação aos métodos computacionais de geração automática de programas de operação do equipamento. O trabalho mostrou que a técnica de programação parametrizada aumenta a eficiência operacional da máquina, através do aumento dos recursos de programação, agregando maior valor na utilização da mesma para realizar operações produtivas.

Meneghello, 2003, propõe uma metodologia alternativa para usinagem, especialmente fresamento, com desenvolvimento de um algoritmo capaz de, através de utilização de recursos de sistemas CAD/CAM, traduzir arquivos gráficos para uma linguagem de programação utilizada em robôs. Os recursos CAD/CAM são utilizados para gerar o código de programação para máquina-ferramenta de comando numérico e posterior adaptação, para aplicação em sistemas robóticos. Informações de caminho de ferramenta para usinagem em máquina CNC são convertidas, através de uma interface computacional, em trajetórias a serem seguidas por uma ferramenta guiada por um manipulador de robô industrial. A viabilidade do sistema proposto é confirmada através de testes realizados a partir de modelos com superfícies complexas, demonstrando que o objetivo do trabalho foi alcançado.

Coppini e Baptista, 1998, relatam que atualmente a otimização dos parâmetros de corte em usinagem está limitada à escolha correta da ferramenta com base em avanços, profundidades de usinagem e velocidades de corte indicados por fontes que publicam a experiência acumulada no assunto (geralmente catálogos de fabricantes). A principal preocupação, no que tange à otimização, é conseguir usinar a peça dentro das especificações

do desenho da mesma. São poucos frequentes os relatos práticos que mostram uma preocupação de aliar ao cuidado acima mencionado, o de otimizar as condições de corte. Desta forma, condições operacionais de usinagem são utilizadas através de extrapolações às existentes, mesmo sendo notório o fato de que este processo de fabricação depende de um número fantásticamente alto de fatores de influência e que tal prática é, por este motivo, geradora de possíveis grandes erros que se perpetuam em plantas fabris, gerando e avolumando eventuais grandes prejuízos que permanecem como se não existissem. Assim, a otimização mais frequentemente encontrada na prática de usinagem, consiste em “adotar as condições de usinagem” e considerando-as como otimizadas, gerenciar o tempo passivo que o operador da máquina teria a sua disposição para trocas de ferramentas e recuperação da fadiga devido às mais diversas causas.

3 METODOLOGIA PARA CONCEPÇÃO DO TRABALHO

Neste capítulo são abordados aspectos relevantes ao projeto e implementação do protótipo computacional, que integra as metodologias e ferramentas propostas como objetivo deste trabalho.

3.1 Definição do sistema

Para a geração do programa CNC utilizando a técnica MDI, os comandos disponibilizam para o usuário muitas opções de funções e ciclos de usinagem, definidos pelo fabricante, com o intuito de facilitar a elaboração do programa. Ciclos de usinagem complementares, necessários para implementar estratégias de usinagem, podem ser elaborados através de programação parametrizada, normalmente disponível no comando da máquina. Os sistemas CAD/CAM são uma técnica moderna para geração de programa CNC, integrando os processos de projeto e fabricação; porém, o programa gerado contém normalmente instruções básicas com interpolações lineares e circulares. A utilização dos ciclos disponíveis no comando normalmente não é realizada, pois necessitaria de pós-processadores especialistas e mão de obra qualificada.

Neste trabalho implementa-se uma metodologia para a geração de programas CNC para otimizar um sistema produtivo. Quanto às informações geométricas, o sistema caracteriza-se por utilizar os ciclos de usinagem que o comando da máquina disponibiliza, possibilitando a aplicação de ciclos desenvolvidos com recursos de programação parametrizada ou geração de novos ciclos, através de uma linguagem genérica proposta. Para os dados tecnológicos, ou seja, as condições de usinagem definidas no código de programação, o sistema permite a otimização dos parâmetros de corte através de um banco de dados com informações realimentadas com resultados obtidos da produção.

A análise da viabilidade da metodologia proposta é realizada através da implementação de um protótipo de um sistema dedicado à programação de tornos CNC, com comando MCS, através do desenvolvimento de aplicativos computacionais desenvolvidos com a linguagem *Delphi7* da *Borland*, *AutoLISP* da *Autodesk* e o banco de dados *firebirds*, orientados à otimização do sistema produtivo.

O *Delphi* é um compilador, uma IDE (*Integrated Development Environment*) e uma linguagem de programação. Utilizado no desenvolvimento de aplicações *desktop*, aplicações

multicamadas e cliente/servidor, compatível com os bancos de dados mais conhecidos do mercado [Cantù, 2003].

O *Firebird* é um sistema gerenciador de banco de dados, desenvolvido e mantido por uma comunidade de desenvolvedores e pela Fundação *Firebird*. É produto de código livre desenvolvido sob licença da *Interbase License*. A origem do *Firebird* é a versão *Open Source do Interbase 6.0* de Julho de 2000, quando a *Borland* liberou as fontes de seu produto [Cantu, 2006].

O *AutoLISP* é um dialeto da linguagem de programação *LISP* construído especificamente para uso com a versão completa do *AutoCAD* e seus derivados. Além da linguagem principal, a maioria das funções primitivas são relacionadas à geometria. Pode-se ter acesso ao banco de dados interno do *AutoCAD DWG*, ou a manipulação de entidades gráficas em *AutoCAD*. Também pode interagir com o usuário através de editor gráfico do *Autocad* pelo uso de funções primitivas que lhe permite escolher pontos, escolher objetos na tela, inserir números e outros dados [Matsumoto, 2001].

O sistema computacional implementado é integrado com módulos de aplicação específica, sendo um módulo gráfico com a finalidade de extração dos dados geométricos (MDG), módulo de ciclos de usinagem correspondentes ao comando específico (MCU), módulo de otimização de parâmetros de corte (MOPC) com processo de realimentação de dados oriundos do sistema produtivo, módulo de programação parametrizada (MPP) que possibilitará a implementação de novos ciclos. A Figura 3.1 mostra um esquema geral do sistema proposto, cujas finalidades são descritas na sequência.

A interface gráfica CAD/CAM integra todos os módulos em um editor de códigos, possibilitando acesso e alterações ao banco de dados de parâmetros de usinagem, do ambiente gráfico de desenho, da programação parametrizada e outras rotinas de suporte ao projeto. A Figura 3.2 ilustra a interface gráfica do protótipo CAD/CAM.

3.2 Módulo de otimização de parâmetros de corte - MOPC

A aplicação de sistemas computacionais como ferramenta de apoio a processos produtivos tem se mostrado indispensável para aumentar a qualidade e produtividade, notadamente na área de fabricação. A fabricação assistida por computador CAM envolve técnicas para apoiar e integrar as etapas do projeto, planejamento de processos, controle de produção, garantia de qualidade e programação de máquinas CNC.

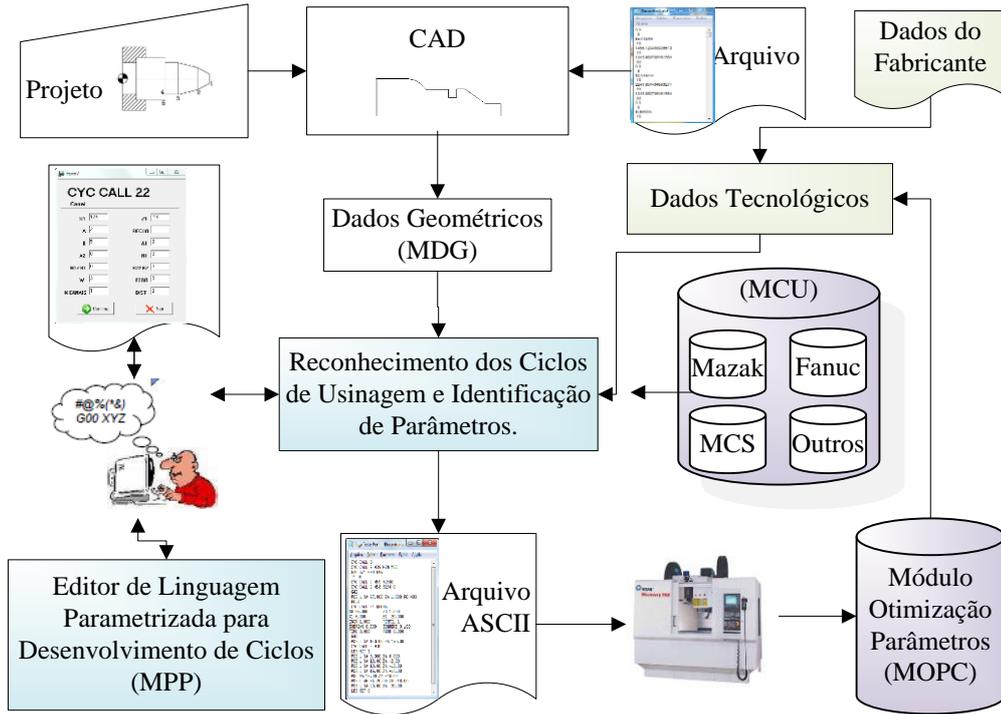


Figura 3.1 - Esquema do sistema

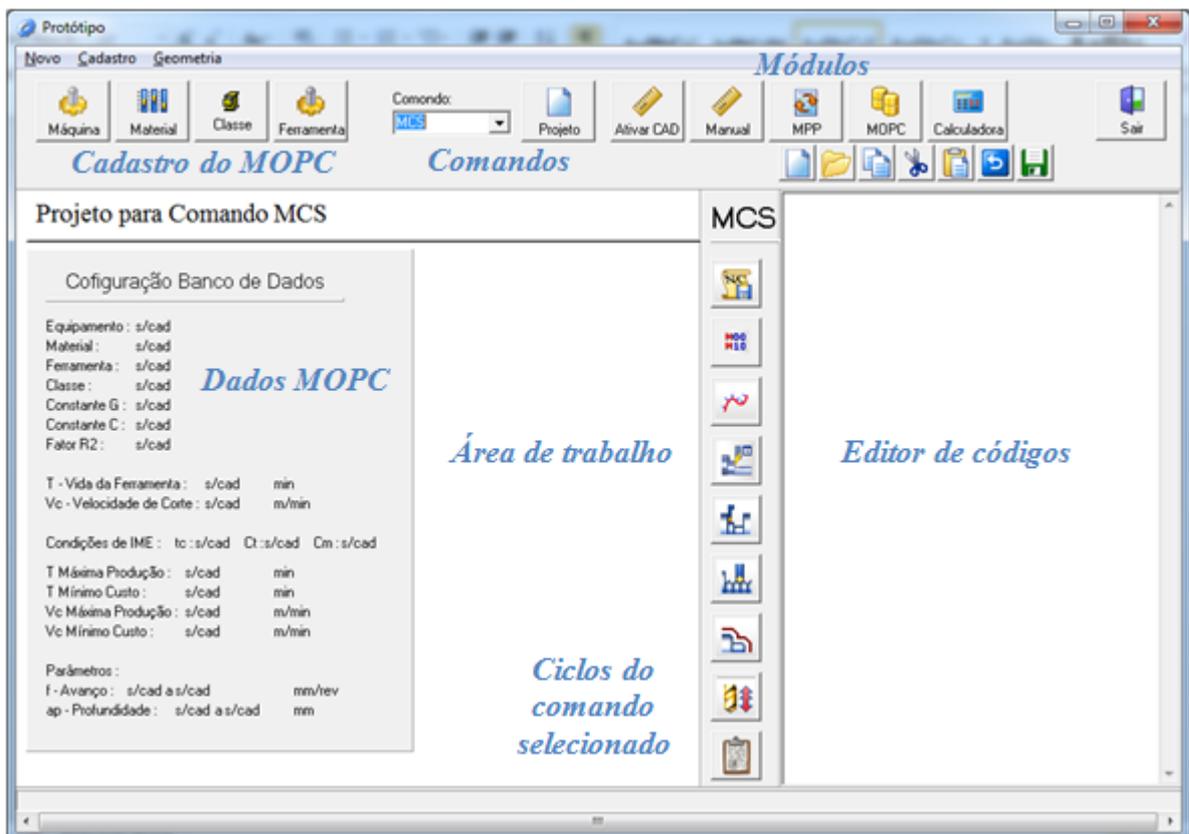


Figura 3.2 - Interface gráfica do protótipo CAD/CAM dedicado

Otimizar condições de corte em processos de fabricação por usinagem tem importante papel nessa cadeia. Embora pesquisas nessa área já tenham atingido um bom nível de desenvolvimento, nem sempre os resultados são empregados em ambiente industrial. A dificuldade em determinar valores ótimos das condições de corte limita-se a utilização de valores recomendados e adaptados aos processos em que serão aplicados e que raramente são melhorados pela realização de ensaios prévios de usinabilidade ou por consideração de critérios de otimização [Silva, 1994]. O conhecimento e a experiência acumulada sobre o processo e suas restrições permanecem, entretanto, como fatores decisivos nessa tarefa [Groover, 2011].

Davim, 2008, explica que o grande número de variáveis envolvidas na usinagem torna complexo o equacionamento da influência das condições de corte, tais como velocidade de corte, avanço e profundidade de corte, sobre as variáveis de estado do processo como vida da ferramenta, forças, forma de cavaco e acabamento superficial.

A literatura mostra que equações básicas relacionadas à economia em usinagem permitem estimar quantitativamente custos e tempos de produção para operações de usinagem. A velocidade de corte pode ser otimizada considerando aspectos de máxima produção ou mínimo custo [Diniz *et al.*, 2000; Ferraresi, 1982; Machado *et al.*, 2009].

O MOPC tem como característica, reunir em um banco de dados, informações de condições de usinagem, adotadas em um sistema produtivo específico, com os respectivos resultados. A metodologia aplicada na avaliação dos parâmetros de usinagem fundamenta-se na formulação de Taylor, aplicada especificamente para a operação de torneamento para definição da vida da ferramenta. O sistema computacional do módulo de otimização caracteriza-se pela implementação de algoritmo de análise formatado em uma base de conhecimento de dados de usinagem integrados a informações relativas a materiais, ferramentas e máquinas. A Figura 3.3 ilustra a metodologia empregada no desenvolvimento do módulo de otimização dos parâmetros de corte MOPC.

3.2.1 Curva de vida da ferramenta

Dentre as condições de processo a velocidade de corte é o parâmetro de maior influência no desgaste em uma ferramenta de corte de usinagem. A curva da vida da ferramenta define-se como sendo o gráfico de tempo de vida (T) em função da velocidade de corte (v_c). Foi primeiramente proposta por Taylor, 1906, que observou que o tempo de vida da ferramenta de corte em usinagem mantém relação direta com a velocidade de corte.

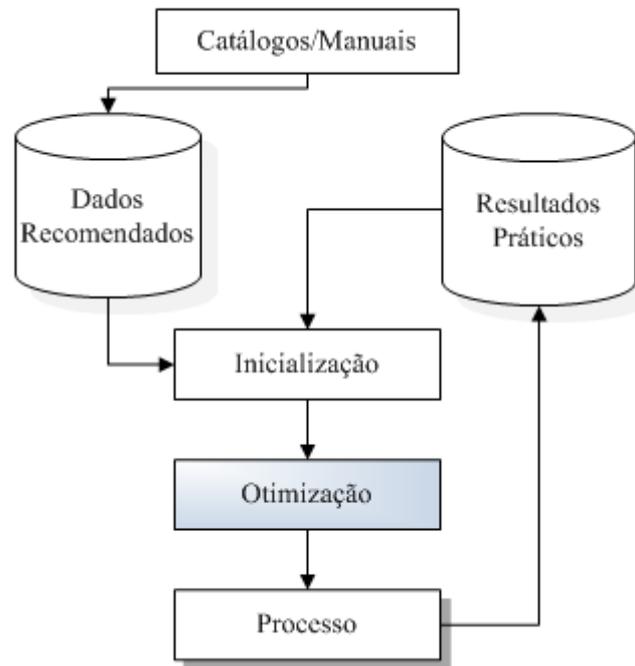


Figura 3.3 - Fluxograma do MOPC

A obtenção dessa relação é experimental e depende do par ferramenta-peça com valores fixos de avanço e de profundidade de usinagem [Machado *et al.*, 2009]. A curva de vida representada em um gráfico bi-logarítmico assume forma próxima da linear, o que indica uma relação exponencial entre T e v_c , Eq. (3.1).

$$T = K \cdot v_c^{-x} \quad (3.1)$$

Onde:

v_c = velocidade de corte

T = tempo de vida da ferramenta

K, x = coeficientes da equação

Embora a velocidade de corte seja fator preponderante para a vida da aresta de corte, deve-se considerar também a influência do avanço e da profundidade de corte. Assim, uma equação de Taylor expandida é aplicada considerando-se esses parâmetros Eq. (3.2).

$$v_c^x \cdot f^y \cdot a_p^z \cdot T = Q \quad (3.2)$$

Onde:

x, y, z, Q = coeficientes da equação

v_c = velocidade de corte (m/min)

T = tempo de vida da ferramenta (min)

a_p = profundidade de corte (mm)

f = avanço (mm/rot)

Considerando a influência de vários fatores na vida da ferramenta, o sistema utiliza uma equação, orientada para a prática, generalizando a equação de Taylor para torneamento, Eq. (3.3) [Stemmer, 2007; Dos Santos et al., 1999].

$$v_c = C_e \cdot T^G \cdot f^E \cdot a_p^F \cdot VB_B^H \quad (3.3)$$

Onde:

v_c = velocidade de corte

T = tempo de vida da ferramenta

f = avanço (mm/rot)

a_p = profundidade corte (mm)

VB_B = largura da marca de desgaste (mm)

E, F, G, H = expoentes determinados experimentalmente.

C_e = Coeficiente “C” da fórmula expandida.

Considerando como constantes os parâmetros f, a_p e VB_B em um determinado processo, conforme a Eq. (3.4), obtém-se a equação reduzida de Taylor, aplicada na base de cálculo do sistema, Eq.(3.5) [Stemmer, 2007].

$$C = C_e \cdot f^E \cdot a_p^F \cdot VB_B^H \quad (3.4)$$

$$v_c = C \cdot T^G \quad (3.5)$$

Onde:

$$C = K^{-1/-x}$$

$$G = 1/-x$$

3.2.2 Base de conhecimento considerada

O processamento de dados no módulo MOPC parte da especificação do material da ferramenta e da peça usinada. Para cada combinação ferramenta/peça, os parâmetros de usinagem implementados são velocidade de corte (v_c), profundidade de corte (a_p) e avanço (f). O tempo de vida da ferramenta (T) resultante dessas condições completam os dados de entrada. O algoritmo também está configurado para tratar dados complementares como tipo de máquina, critério de fim de vida da ferramenta - largura média da marca de desgaste de flanco (V_B), codificação da geometria da ferramenta, rugosidade obtida (R_a), força de corte (F_c) e força avanço (F_a).

Considerando que o aplicativo tem como modelamento do processo de usinagem a equação reduzida de Taylor, as informações mínimas a serem definidas correspondem à velocidade de corte e à vida correspondente. A profundidade de corte, o avanço e os dados complementares são restrições a serem manipuladas para selecionar os dados para processamento. Dessa forma, o algoritmo se caracteriza pela capacidade de oferecer ao processista a possibilidade de filtrar as informações do banco de dados, de acordo com os itens que julgar de interesse. O sistema processa somente as ocorrências que cumprem as condições especificadas e retorna como saída os valores de velocidade de corte e vida da ferramenta correspondente.

As condições de entrada selecionadas influenciam no cálculo das constantes C e G da equação reduzida de Taylor. Com a possibilidade de utilização das ferramentas de filtro, o sistema se aproxima da equação expandida de Taylor. Por exemplo, se for selecionado no filtro o avanço de corte com valor determinado, a constante “ C ” quando calculada terá um valor diferente da calculada sem esse valor estabelecido. Na Figura 3.4, os gráficos representam a equação de Taylor.

Como a equação de Taylor é uma função exponencial, em um gráfico bi-logarítmico representa-se como uma reta. Para uma linearização, quando vários pontos são considerados, o método de mínimos quadrados ajusta os dados para a definição dos parâmetros da equação da reta, cujos coeficientes correspondem as constantes da equação reduzida de Taylor.

Com o objetivo de dar uma orientação sobre a dispersão dos dados em relação à reta ajustada, o sistema indica o coeficiente de correlação r^2 , Eq. (3.6), que dá uma indicação da precisão do alinhamento dos pontos no gráfico $\log(V_c)$ por $\log(T)$ (quanto mais próximo de 1 o valor r^2 , melhor o ajuste).

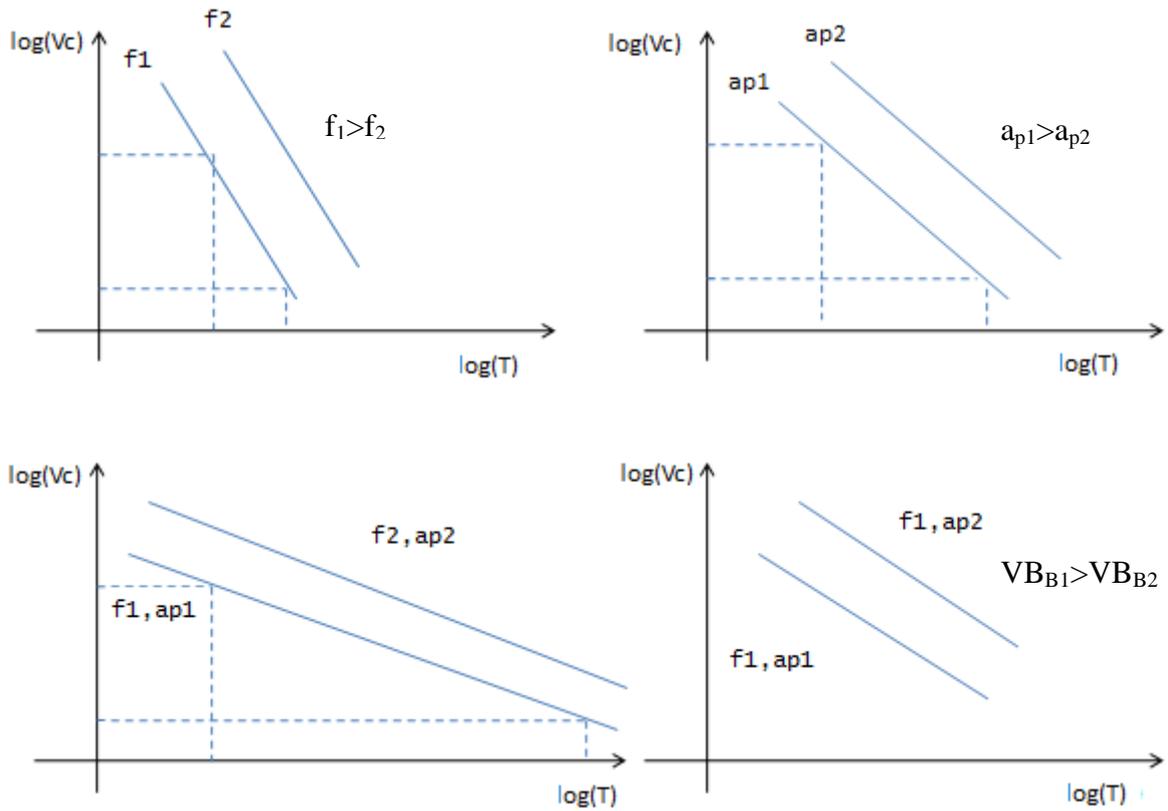


Figura 3.4 - Influência dos parâmetros na equação de Taylor [Stemmer, 2007]

$$r^2 = a \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum y_i^2 - (\sum y_i)^2} \quad (3.6)$$

Com as constantes da equação de Taylor calculadas, o algoritmo aplica equações básicas da economia em usinagem, que permitem calcular quantitativamente custos e tempos de produção para operações de usinagem. As Eq. (3.7) e Eq. (3.8) representam respectivamente a velocidade de corte e a vida da ferramenta para máxima produção e mínimo custo, [Stemmer, 2007].

$$v_{otm,p} = \left\{ - \left(1 + \frac{1}{G} \right) \cdot \frac{tc}{C^{-1/G}} \right\}^G ; v_{otm,c} = \left\{ - \left(1 + \frac{1}{G} \right) \cdot \frac{\left(tc + \frac{C_t}{C_m} \right)}{C^{-1/G}} \right\}^G \quad (3.7)$$

$$T_{otm,p} = - \left(1 + \frac{1}{G} \right) \cdot tc ; T_{otm,c} = - \left(1 + \frac{1}{G} \right) \cdot \left(tc + \frac{C_t}{C_m} \right) \quad (3.8)$$

Onde:

$v_{otm,p}$ = velocidade para máxima produção

$v_{otm,c}$ = velocidade para mínimo custo

$T_{otm,p}$ = tempo de vida para máxima produção

$T_{otm,c}$ = tempo de vida para mínimo custo

G e C = coeficientes da equação de Taylor

t_c = tempo de troca da ferramenta

C_t = custo por gume.

C_m = custo de utilização da máquina-ferramenta.

3.2.3 Caracterização do sistema computacional

O módulo de otimização de parâmetros de corte é fundamentado em um banco de dados alimentado com informações oriundas do processo produtivo. A Figura 3.5 ilustra o fluxograma do algoritmo do MOPC.

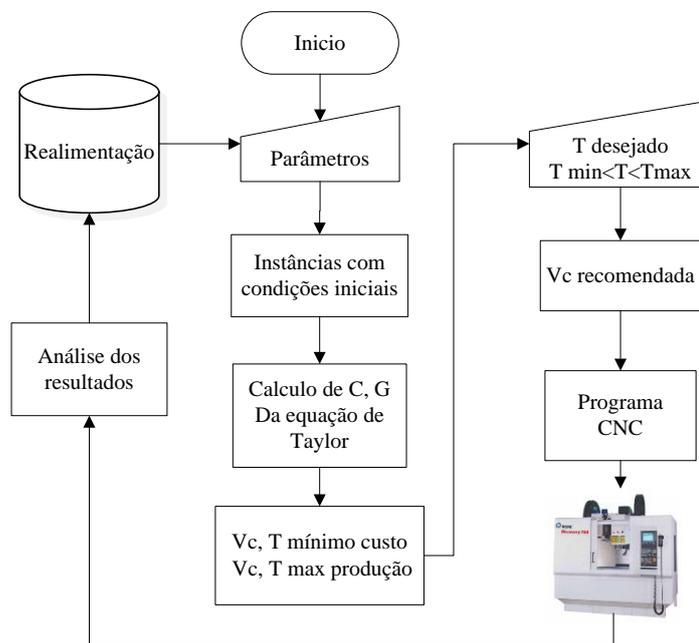


Figura 3.5 - Fluxograma do algoritmo do MOPC [Gonçalves, 2012]

Quando o sistema contiver um histórico suficiente de dados, o algoritmo poderá ser utilizado para calcular valores de velocidade de corte e tempos ótimos baseados nos dados arquivados. Selecionando-se parâmetros de interesse, o algoritmo filtra o banco de dados para

análise da influência da velocidade de corte na vida da ferramenta correspondente. Com os dados de vida de ferramenta e velocidade de corte obtém-se a curva de vida da ferramenta referente às constantes da equação de Taylor.

O MOPC é composto por um banco de dados com os campos das tabelas relacionados com a finalidade de organizar as informações inseridas. Cada tabela possui um campo de nome “código”. Esses campos são declarados com chave primária, que significa a impossibilidade de registros iguais. A Figura 3.6 ilustra a organização relacional das tabelas do banco de dados.

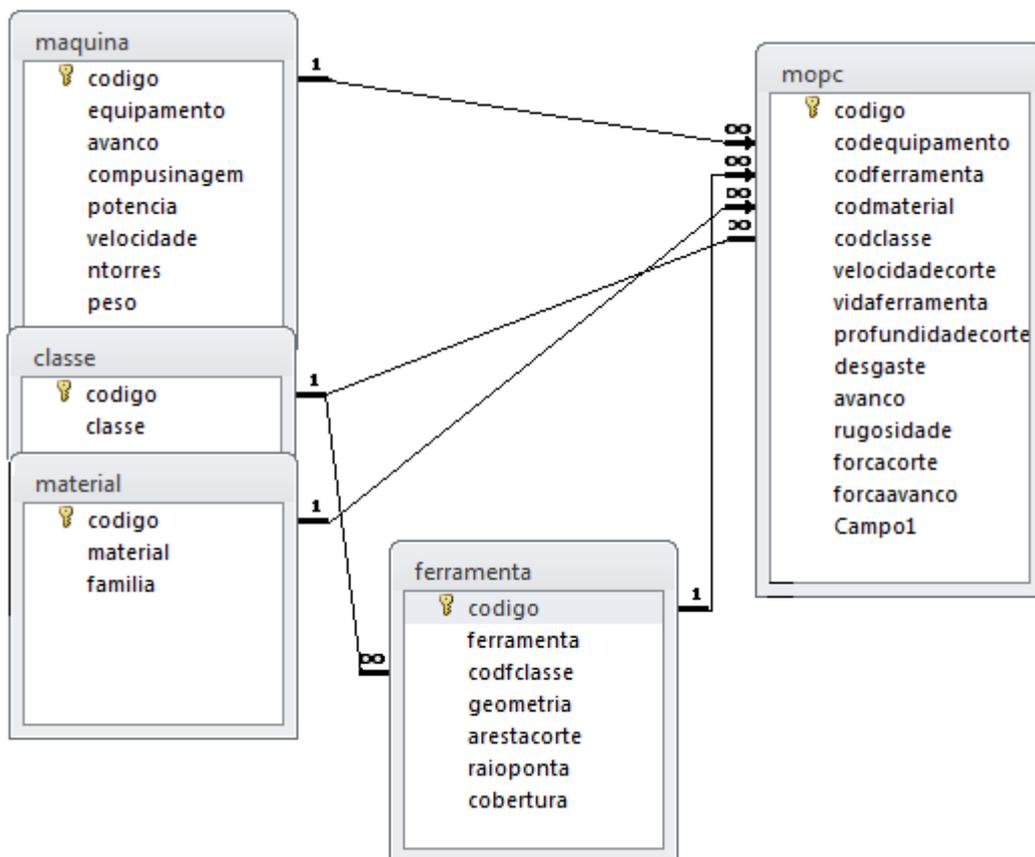


Figura 3.6 - Estrutura relacional do MOPC

Quando ocorrer um registro na tabela de dados “MOPC”, os campos relacionados serão preenchidos com os dados referentes ao campo “código” dessas tabelas, que possui dados numéricos não repetidos. Os campos que compõem as tabelas podem ser usados como variáveis no sistema computacional.

A manipulação dos campos das tabelas de registros do banco de dados é realizada por interfaces gráficas desenvolvidas no software de programação *Delphi7*. Para inserir dados na tabela MOPC, o tipo de material a ser usado e a classe das ferramentas usadas são itens

obrigatórios, sem os quais, não é possível gravar nova entrada de dados. Para correto funcionamento do sistema é necessário o cadastramento inicial do tipo de material e da classe da ferramenta a ser usado.

Dados complementares, como a codificação geométrica da ferramenta, indicada pelo fabricante e o tipo de máquina de trabalho, também são possíveis de inserção na configuração do sistema. A Figura 3.7 ilustra a interface de cadastro da classe da ferramenta.



Figura 3.7 - Interface para cadastro da classe da ferramentas

A tabela de ferramenta é vinculada ao código da tabela classe. Dessa forma não pode ser cadastrado um novo registro de ferramenta se a classe relacionada não estiver cadastrada na tabela classe.

As interfaces desenvolvidas possuem um conjunto de componentes na sua estrutura, similares ao formulário do cadastro de ferramentas ilustrado na Figura 3.8. Entre os componentes ressaltam-se caixa de edição para filtrar os elementos cadastrados, componente grade para exibir lista de registros, caixas de edição para cadastro de novos registros e um componente navegador. A tabela “MOPC” constituirá, no seu registro do valor numérico correspondente ao campo código das tabelas ferramenta, classe, máquina e material. A Figura 3.9 ilustra a interface para cadastro de material e a Figura 3.10 ilustra a interface para cadastro de máquinas.

Cadastro Ferramentas

Filtro:

Classe	Ferramenta	Geometria	Aresta Corte	Raio
K10	SNMM 190624			
▶ P10	TNMM 16 04 12 R4	R4	3	1,2
P15	TNMG 160404 PF GC 4015 P15			
H05	CNGA 12 04 08 501525			

Ferramenta:

Classe: Geometria: Raio da Ponta:

Aresta de Corte: Cobertura:

Figura 3.8 - Interface para cadastro de ferramentas

Cadastro Materiais

Filtro:

Material	Familia
▶ SAE 1020	Carbono
SAE 1040	Carbono
SAE 1045	Carbono
SAE 4140	Liga
SAE 5120	Liga
ABNT 1040	

Materiais: Familia:

Figura 3.9 - Interface para cadastro de materiais

Cadastro Máquinas

Filtro:

Equipamento
▶ Mori Seiki NL 3000/1250
Mori Seiki NI 3000/1250
Okuma LR 25
Index XGU 600
Promecor SMT 20
Romi Centur 30RV SENAI S. Leopoldo

Equipamento:

Avanço: mm/min Comp. Usinagem: mm Potência: KW

Velocidade: RPM Numero Torres: Peso: Kg

⏪ ⏩ + - ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ ⏺

Figura 3.10 - Interface para cadastro de máquinas

Nas fases mais avançadas, a inserção de dados práticos pode ser realizada disponibilizando para o operador da máquina uma planilha de aquisição de parâmetros de processo (folha de dados de campo), dividida por tipo de material e o tipo de ferramenta, conforme modelo proposto que é ilustrado na Figura 3.11.

Na implementação do sistema, os registros de inicialização do MOPC podem ser obtidos de dados fornecidos pelos fabricantes da ferramenta ou publicações técnicas. Com os dados de realimentação oriundos do sistema produtivo, os resultados se tornam mais refinados. Fixando valores nos parâmetros disponibilizados no filtro do sistema os cálculos da constante “C” podem ser aprimorados.

Outra possibilidade de extrair dados para alimentação do sistema é realizar uma leitura da listagem dos programas finais gerados pelo protótipo proposto. A Figura 3.12 ilustra exemplos de dados extraídos do programa. Nesse caso é obtido o número da ferramenta usada, expressa pela função “T”, a velocidade pela função “S”, a velocidade de avanço pela função ”F” e a distância percorrida pela função “X, Y”.

 Laboratório de Usinagem

		Ferramentas																				
		Ferramenta 1						Ferramenta 2						Ferramenta 3								
Materiais	Material 1	Vc	ap	f	T	α	C	Vc	ap	f	T	G	C	Vc	ap	f	T	G	C			
	Material 2	Ferramenta 1	Ferramenta 2	Ferramenta 3	Vc	ap	f	T	α	C	Vc	ap	f	T	G	C	Vc	ap	f	T	G	C

Figura 3.11 - Folha de dados experimentais [Gonçalves, 2011]

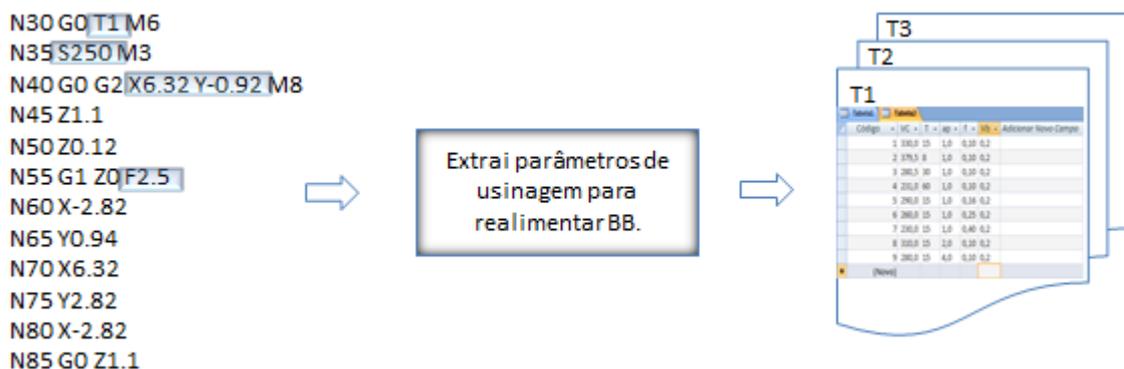


Figura 3.12 - Exemplo de dados tecnológicos extraídos do programa gerado

A figura 3.13 ilustra a interface de consulta e cadastramento de registros do MOPC, com os campos de inserção de dados de corte e a grade de visualização das informações gravadas no sistema. Os registros dos campos referentes à velocidade de corte e vida da ferramenta são obrigatórios. Os dados restantes são complementares e servem para definir os critérios do filtro.

Vc	T	ap	VB	f	Material	Class	Tools	Machine	Ra	Power Cut	Advance
260	15	1	0,2	0,25	SAE 1020	P10	SNMM 190624	Mori Seiki NL 3000/1250			
230	15	1	0,2	0,4	SAE 1020	P10	SNMM 190624	Mori Seiki NL 3000/1250			
310	15	2	0,2	0,1	SAE 1020	P10	SNMM 190624	Mori Seiki NL 3000/1250			
280	15	4	0,2	0,1	SAE 1020	P10	SNMM 190624	Mori Seiki NL 3000/1250			
425	41	0,8	0,31	0,15	ABNT 1040	P15	TNMG 160404 PF GC 40°	Romi Centur 30RV SENAI S.Le	14,58		
425	36	0,8	0,24	0,15	ABNT 1040	P15	TNMG 160404 PF GC 40°	Romi Centur 30RV SENAI S.Le	9,96	325	172,12
425	31	0,8	0,32	0,15	ABNT 1040	P15	TNMG 160404 PF GC 40°	Romi Centur 30RV SENAI S.Le	4,82	331,73	182,85
475	19	0,8	0,29	0,15	ABNT 1040	P15	TNMG 160404 PF GC 40°	Romi Centur 30RV SENAI S.Le	14,54	320,79	189,85
475	24	0,8	0,32	0,15	ABNT 1040	P15	TNMG 160404 PF GC 40°	Romi Centur 30RV SENAI S.Le	8,6	329,41	174,07
475	25	0,8	0,32	0,15	ABNT 1040	P15	TNMG 160404 PF GC 40°	Romi Centur 30RV SENAI S.Le	10,43	356,83	219,11
525	15	0,8	0,3	0,15	ABNT 1040	P15	TNMG 160404 PF GC 40°	Romi Centur 30RV SENAI S.Le	8,85	349,17	168,6
525	14	0,8	0,3	0,15	ABNT 1040	P15	TNMG 160404 PF GC 40°	Romi Centur 30RV SENAI S.Le	4,97	351,39	211,87
355	26,69	0,8	0,28	0,15	ABNT 1045	P15	TNMG 160404 PF GC 40°	Romi Centur 30RV SENAI S.Le	2,76	287,95	244,77

Figura 3.13 - Interface para inserir parâmetros de usinagem no sistema

O algoritmo processa a seleção dos dados, no banco de dados, relativos aos valores de velocidade de corte e tempo de vida da ferramenta correspondente às configurações de filtro realizadas. Esses dados são os necessários para o algoritmo realizar os cálculos, pois são baseado na equação reduzida de Taylor. O sistema caracteriza-se por possuir um dispositivo de filtro dos parâmetros cadastrados no banco de dados. Esse processo permite inserir intervalos de busca aprimorando a seleção. O método realiza a seleção dos dados respeitando as limitações estabelecidas no filtro. A classe da ferramenta e material da peça a ser usinado são dados iniciais de processamento, e os parâmetros na caixa de filtro ficam a critério do usuário. Na interface possibilita-se montar várias combinações de condições de usinagem para estudos. A Figura 3.14 ilustra a interface do filtro de dados.

Com os valores de velocidade de corte e vida da ferramenta extraídos dos dados do MOPC e considerando os parâmetros escolhidos na filtragem, a interface de otimização calcula as constantes da equação de Taylor, fornecendo os Intervalos de Máxima Eficiência e permite calcular a vida da ferramenta para uma velocidade de corte específica, e vice-versa. A Figura 3.15 mostra a interface de otimização.

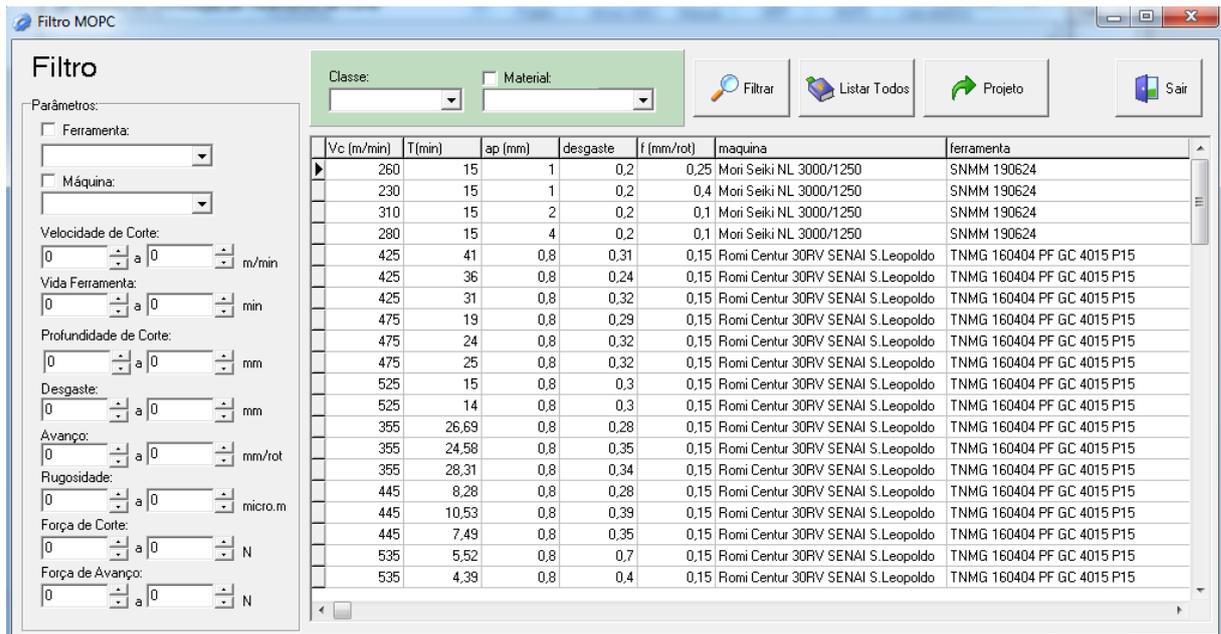


Figura 3.14 - Interface para filtro de parâmetros de usinagem no sistema

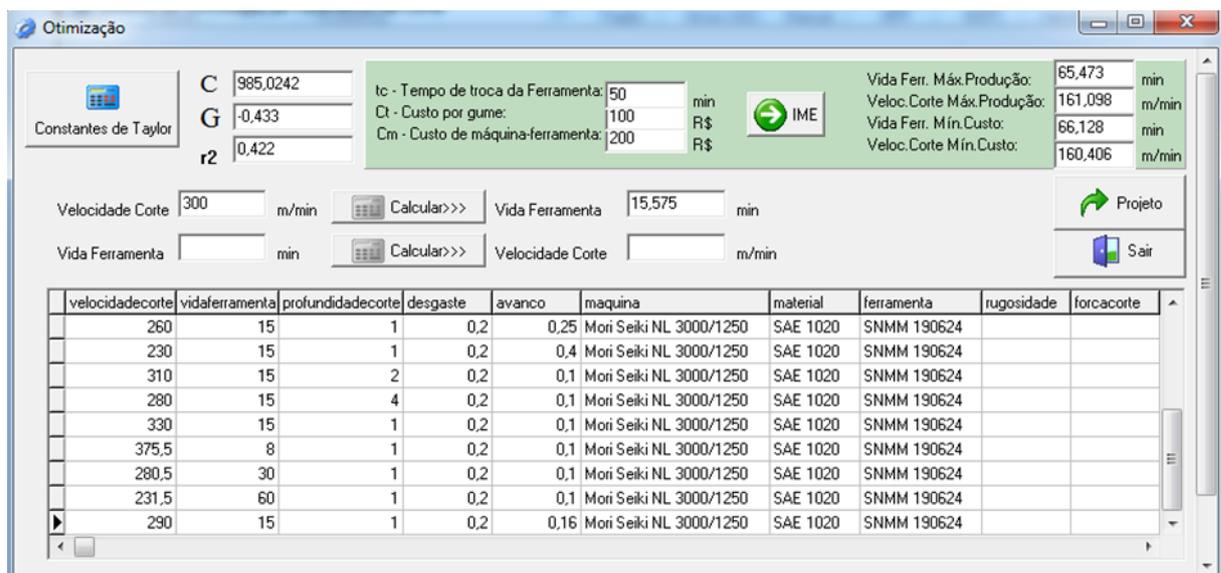


Figura 3.15 - Interface de otimização do sistema

3.3 Módulo de programação parametrizada - MPP

Apesar da eficiência dos sistemas CAM, a programação manual é bastante empregada, principalmente, na programação de geometrias regulares. Nessa forma de programação, existem funções que auxiliam o programador e estão disponíveis nos respectivos comandos CNC. Exemplos dessas funções são os ciclos de usinagem para operações de desbaste e execução de cavidades ou ressaltos regulares [Groover, 2011].

A programação parametrizada é uma forma de programação manual avançada que permite incluir em programas CNC cálculos matemáticos, variáveis computacionais e desvios condicionais. Isso permite implementar lógica algorítmica na operação da máquina de comando numérico. Os fabricantes dos comandos numéricos disponibilizam a metodologia da programação parametrizada, sendo específica para cada tipo de comando [Smid, 2003].

O Módulo de Programação Parametrizada (MPP) fundamenta-se em uma linguagem de programação híbrida universal, elaborada com fundamentos de programação Pascal [Manzano e Yamatumi, 2007] e incrementada com o código “G” (*General or preparatory*) pelo padrão ISO 6983, 1982.

O objetivo do desenvolvimento do MPP é facilitar a programação, não sendo necessário o conhecimento e domínio de múltiplas linguagens de programação parametrizada, pois cada comando possui sua linguagem própria. Com isso, a elaboração de operações de usinagem específica baseadas na parametrização torna-se mais acessível. As máquinas CNC nas quais ocorre à aplicação de recursos de programação parametrizada no desenvolvimento de ciclos de usinagem tendem a aumentar sua produtividade [Gonçalves, 2007]. Com a aplicação da parametrização tem-se mais uma referência à aplicação de uma técnica de programação que, embora pouco difundida, quando adequadamente utilizada permite gerar rotinas para geometrias padronizadas ou parametrizadas.

3.3.1 Linguagem proposta

Neste item é descrita a linguagem de programação híbrida, com funções básicas aplicadas às necessidades do movimento da ferramenta de usinagem. A linguagem proposta está baseada em fundamentos da linguagem Pascal com a inserção de Códigos “G”. Os recursos de programação utilizados são básicos e as sintaxes são demonstradas a seguir.

No contexto da linguagem as variáveis são do tipo real e declaradas no início do programa após a declaração “VAR”, como na sintaxe mostrada abaixo. O bloco de comando é representado por “begin” e “end”. Para atribuir valores nas variáveis é utilizado ” : = ” conforme a sintaxe:

```
Var  
Variavel1: real;  
Variavel2: real;
```

```
begin  
Variavel1:= 20;  
Variavel2:= 30;  
end;
```

Na linguagem proposta é possível realizar operações aritméticas sobre variáveis. A Tabela 3.1 mostra os operadores disponíveis.

Tabela 3.1 - Operadores aritméticos

Operador	Significado
*	Multiplicação
/	Divisão
+	Soma
-	Subtração

A linguagem pode realizar uma operação aritmética entre duas variáveis e atribuir o resultado a uma terceira variável, mediante a inclusão de uma linha de código com a seguinte sintaxe:

variável3:= variável1 operação variável2;

Operadores relacionais são admitidos na programação como indicados na Tabela 3.2.

Tabela 3.2 - Operadores relacionais

Operador	Significado
= (igual):	testar se dois valores são iguais.
<> (diferente):	testar se dois valores são diferentes
< (menor):	testar valor é menor do que outro.
<= (menor ou igual):	testar valor é menor ou igual a outro.
> (maior):	testar valor é maior do que outro.
>= (maior ou igual):	testar valor é maior ou igual a

O código de programação de comando numérico é inserido intercalado com a estrutura algorítmica, caracterizando a linguagem proposta como uma estrutura híbrida para interpretação do compilador. A Tabela 3.3 mostra exemplos de códigos aceitos na programação.

Tabela 3.3 - Código ISO 6983, 1982

ISO	Significado
G90	Coordenadas Absolutas
G91	Coordenadas Incrementais
G00	Movimento Aproximação
G01	Interpolação Linear
G02	Interpolação Circular H
G03	Interpolação Circular AH
X 100 Y 400 Z150	Posicionamentos
T2 D2	Troca Ferramentas
F200	Velocidade Avanço
S1200	Velocidade de rotação
M13 M14 M30	Funções Miscelâneas

Existem situações em que se deseja executar um trecho de código, apenas se uma determinada condição for verdadeira, e outro trecho de código, caso a condição testada seja falsa, a estrutura condicional *if-then-else* apresenta a seguinte sintaxe:

if (condição)	}	Funções da linguagem Pascal
then		
begin	}	Bloco de comandos executados com código CNC com a condição for verdadeira.
G0.....		
G1.....		
G3.....		
M14		
end	}	Funções da linguagem Pascal
else		
begin	}	Bloco de comandos executados com código CNC com a condição for falsa.
G0.....		
G1.....		
G2.....		
M14		
end;	}	Funções da linguagem Pascal
end;		

Essa estrutura de repetição é utilizada quando se deseja que a execução de um bloco de comandos seja repetida enquanto uma determinada condição for verdadeira. A estrutura de repetição apresenta a seguinte sintaxe:

while <condição> do	}	Bloco de comandos executados com código CNC enquanto a condição for verdadeira.
begin		
G0.....		
G1.....		
G3.....		
M14	}	
end;		

A Figura 3.16 mostra, como exemplo, a listagem de um programa desenvolvido com a linguagem proposta para a realização da operação de desbaste longitudinal. Os parâmetros iniciais são diâmetro inicial em X= 24 mm, diâmetro final em X= 12 mm, profundidade em Z= 40 mm, posição inicial em Z= 0 mm e profundidade de corte $a_p= 1$ mm.

```

var
xinicio : real;
xfinal : real;
zinicio : real;
zfinal : real;
xatual : real;
prof : real;

begin
xinicio := 24;
xfinal :=12;
zinicio := 0;
zfinal := -40;
prof := 1;
xatual := xinicio;

while (xatual>xfinal) do
begin
G0 X(xatual-prof) Z(zinicio+1);
G1 Z(zfinal);
G0 X(xatual);
G0 Z(zinicio+1);
xatual:= xatual - prof;

end;
if (xatual<=xfinal) them
begin
G00 x(xinicio) z(zinicio) + 5);
M30
end;
end;
end.

```

Declaração de Variáveis
Inicialização
Rotina de Repetição
Rotina Condicional

Figura 3.16 – Exemplo de programa com a linguagem proposta

3.3.2 Caracterização da interface gráfica do algoritmo MPP

Para interpretar a linguagem proposta foi desenvolvida uma estratégia de preparação e compilação do programa. A Figura 3.17 ilustra a interface gráfica desenvolvida em Delphi7 composta de duas áreas de edição. A primeira, à esquerda, é o ambiente para editar a linguagem parametrizada proposta no sistema. A segunda, do lado direito, é para editar o código CNC gerado.

A interface é composta de atalhos em forma de “botões” de acionamento, com os recursos de editoração mais utilizados. O botão de “Interpretação” executa o programa desenvolvido para ler, linha por linha, o programa parametrizado e gerar um arquivo *Pascal*.

O botão “Compilador” executa o aplicativo no ambiente de comando. A opção “Gera Código” mostra o programa gerado no editor de código CNC. O Programa pode ser transferido para o sistema principal através do acionamento da opção “Projeto”.

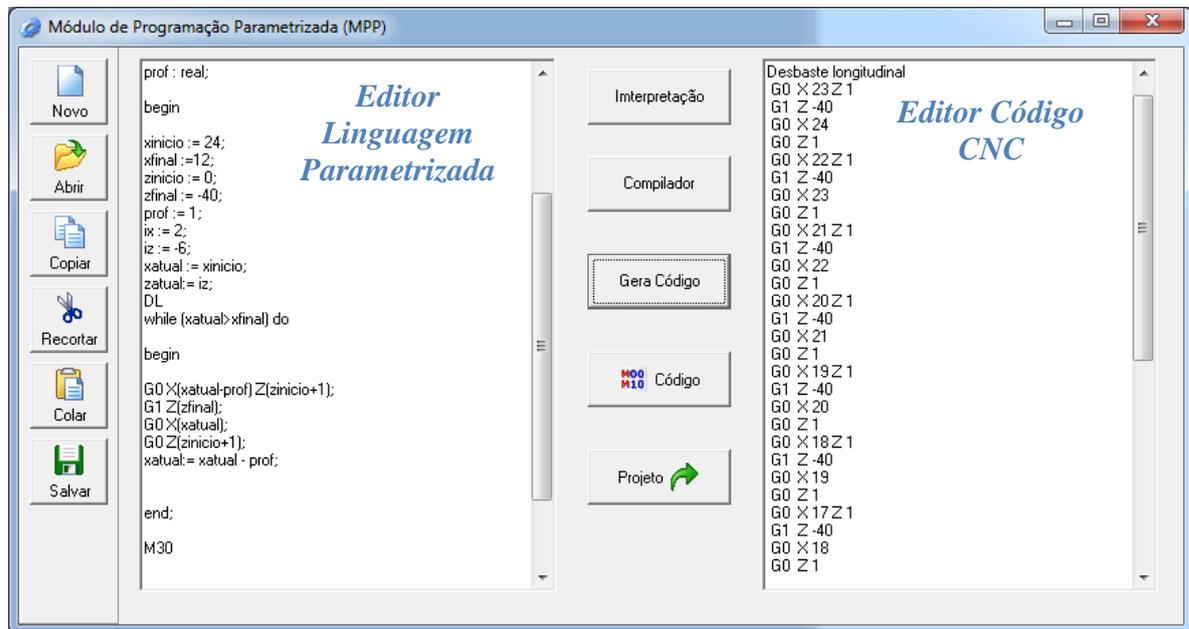


Figura 3.17 - Interface gráfica do MPP

O código de programação parametrizada elaborado é armazenado na forma de arquivo texto. Essa forma de armazenamento é a técnica empregada para transferência de dados entre os aplicativos.

A Figura 3.18 mostra um esquema da lógica empregada no desenvolvimento do módulo de programação parametrizada MPP. Nele pode ser observado o momento de aplicação do interpretador no arquivo de formato texto, contendo os códigos elaborados com a linguagem parametrizada proposta, gerando um arquivo1 em formato texto compilado pelo *Pascal* nativo, resultando o código CNC.

Um algoritmo foi desenvolvido com a finalidade de interpretar a sintaxe e a semântica dos blocos de programação da linguagem parametrizada proposta, arquivando os dados em unidade de compilação Pascal nativo. A Figura 3.19 ilustra um trecho desse algoritmo em linguagem de programação Pascal, responsável pela interpretação da linguagem proposta.

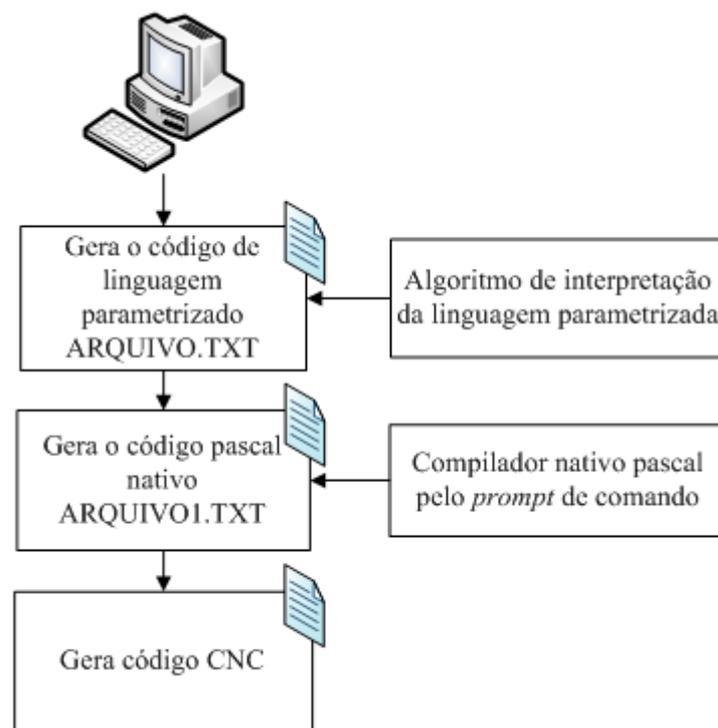


Figura 3.18 - Esquema do MPP

```

if termo = 'G1' then
begin
  if (pos('x',texto) <> 0) and (pos('z',texto) <> 0)then
  begin
    termo:= copy(texto,(pos('(' ,texto)),((pos(')',texto))-(pos
    ('(',texto))+1));
    termo1:= copy(texto,pos('z',texto),length(texto));
    termo2:= copy(termo1,(pos('(' ,termo1)),((pos(')',termo1))-(pos
    ('(',termo1))+1));
    writeln(arq2,'Memo1.Lines.Add('' G1 '' + '' x '' + floattostr'
    + termo + '' + '' z '' + floattostr' + termo2 + ');');
  end
  else
  if (pos('x',texto) <> 0) and (pos('z',texto) = 0)then
  begin
    termo:= copy(texto,(pos('(' ,texto)),((pos(')',texto))-(pos
    ('(',texto))+1));
    writeln(arq2,'Memo1.Lines.Add('' G1 '' + '' x '' + floattostr'
    + termo + ');');
  end
  else
  begin
    termo:= copy(texto,(pos('(' ,texto)),((pos(')',texto))-(pos
    ('(',texto))+1));
    writeln(arq2,'Memo1.Lines.Add('' G1 '' + '' z '' + floattostr'
    + termo + ');');
  end;
end;

```

Figura 3.19 - Trecho do algoritmo interpretador para o código "G1"

As informações do código gerado são transferidas para o sistema principal através da leitura dos dados do Arquivo CNC. As informações são inseridas automaticamente nas linhas de código do programa principal, montado com os ciclos de usinagem, tornando-se um programa híbrido.

Esse processo não é perceptível para o usuário do MPP. A sequência de geração dos arquivos necessários é automatizada, sem a interferência do programador. A Figura 3.20 ilustra as interfaces gráficas do MPP em tempo de compilação de um programa.

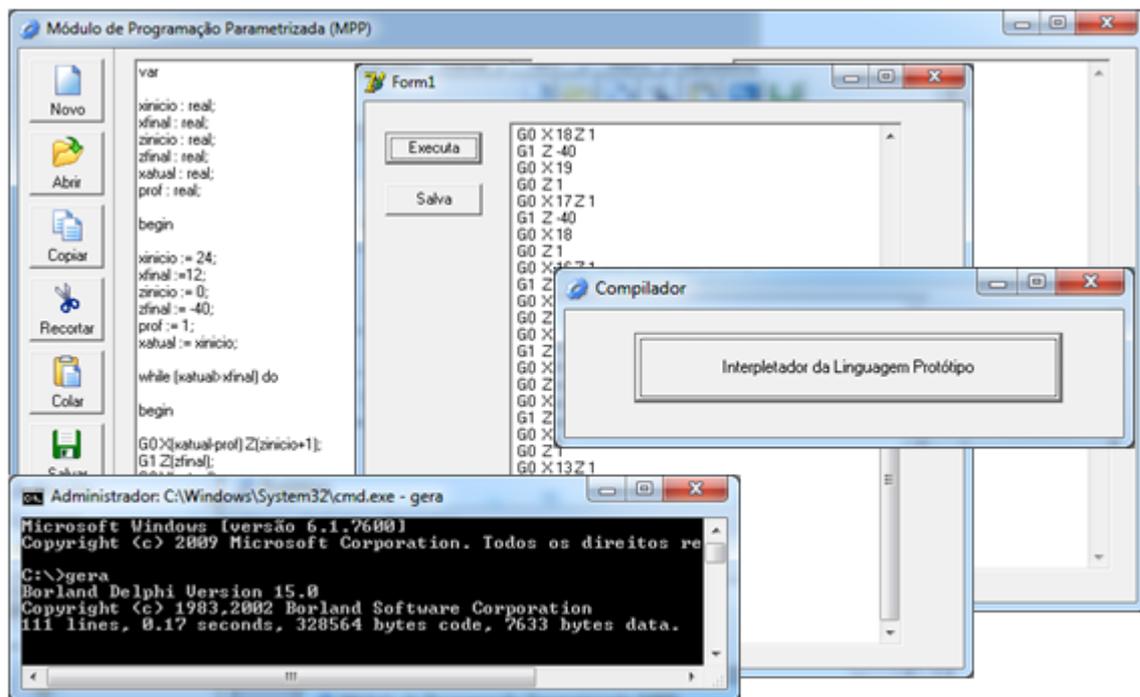


Figura 3.20 - Módulo de Programação Parametrizada

A Figura 3.21 ilustra os eventos ocorridos no processo de compilação da linguagem parametrizada e a geração do programa CNC. O programa com a codificação da linguagem genérica digitado no editor de linguagem parametrizada é armazenado num arquivo (nome.txt). O algoritmo (INTERPRETADOR) é executado gerando um código de formato nativo *Pascal*, armazenando o resultado em arquivo (nome1.txt). Um compilador nativo *Pascal* (COMPILADOR PASCAL) é executado através do comando “gera”, previamente programado em arquivo de execução no ambiente de comando. O código CNC é gerado e arquivado em (nomeCNC.txt).

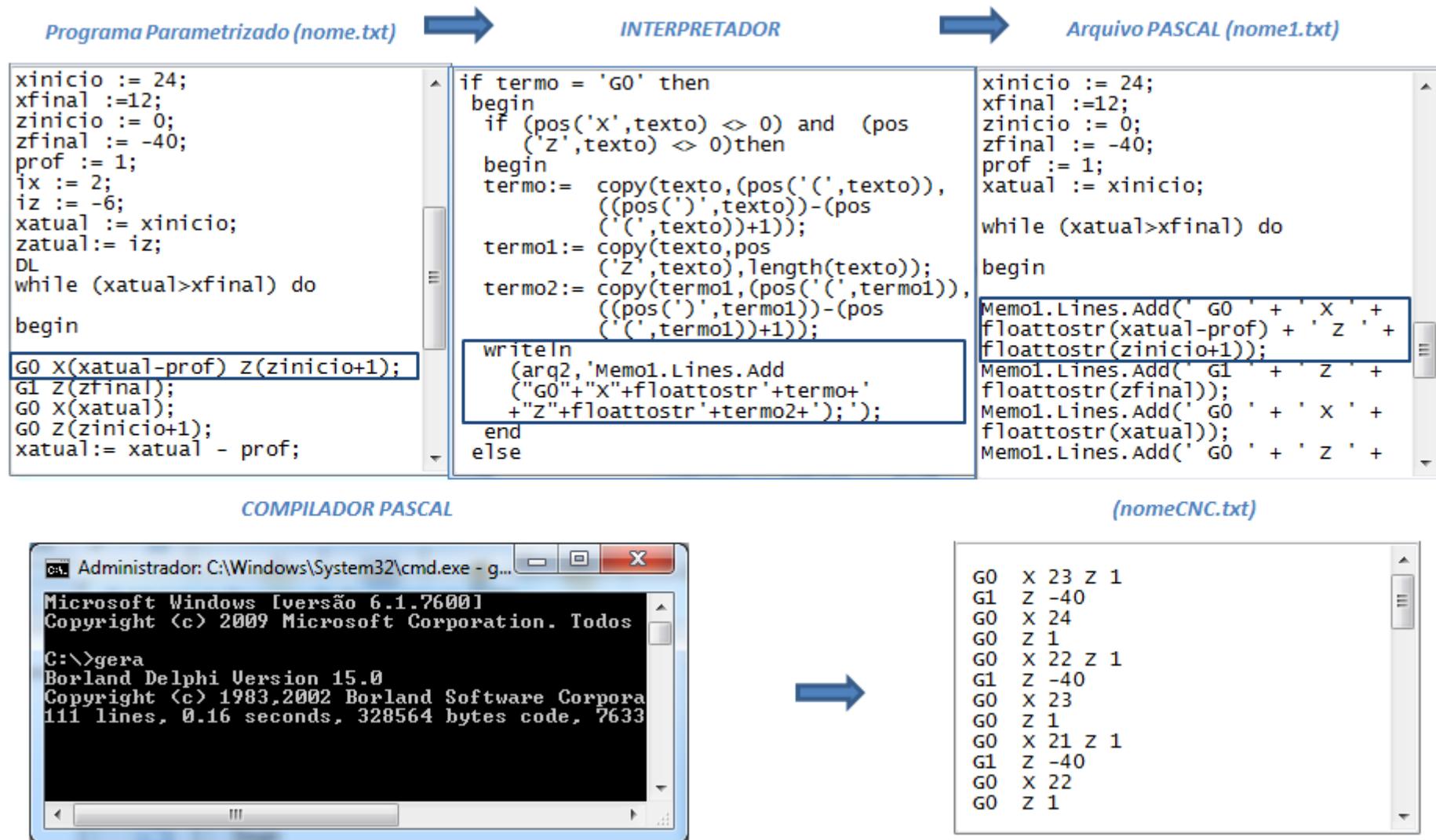


Figura 3.21 - Sequência de fases para obter arquivo CNC

3.4 Módulo dos dados geométricos - MDG

Para tratar dados geométricos, o sistema proposto necessita de um módulo gráfico vetorial e interativo com o usuário, com suporte à leitura de arquivos com formato DXF (*Drawing Exchange Format*), DWG (*Autodesk's Drawing Exchange Format*) ou IGES (*Initial Graphics Exchange Specification*). Esses tipos de dados são característicos para o armazenamento e transferência de informações entre softwares CAD. O módulo gráfico pode recuperar arquivos de desenhos gerados por ele próprio ou em ambientes CAD de terceiros. No ambiente gráfico vetorial, uma simples linha é considerada uma entidade do desenho possuindo características de um objeto de programação com suas propriedades armazenadas. Essas informações são os dados geométricos necessários para serem recuperados no sistema computacional proposto. Além dos recursos de edição de desenho, comuns em ambiente CAD, o módulo de desenho deve possuir arquitetura aberta para programação nativa, disponibilizando acesso às funções fornecidas pelo software. Dentro das características descritas para a criação do módulo de dados geométricos, optou-se pelo uso dos recursos gráficos do software *AutoCAD* que tem, como característica marcante, a possibilidade do uso da programação em linguagem interpretada, conhecida como *Auto LISP*. A Figura 3.22 ilustra o módulo MDG carregado em um formulário do protótipo CAD/CAM.

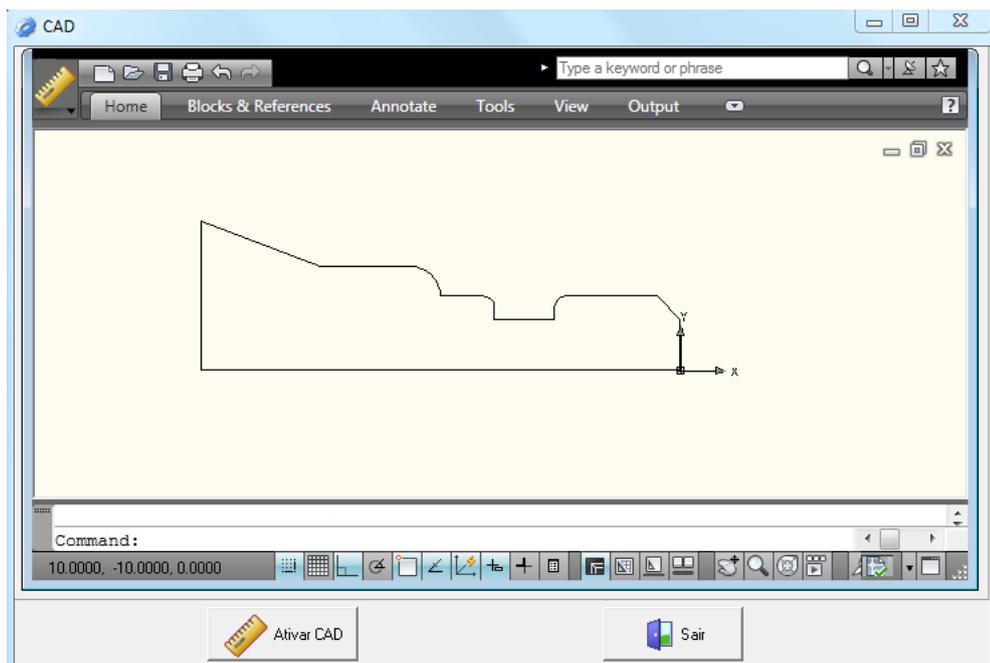


Figura 3.22 - Interface do módulo geométrico

A técnica para obtenção dos dados geométricos do módulo MDG consiste na elaboração de rotinas em linguagem *AutoLISP*, destinadas a reconhecer padrões de usinagem de torneamento, com a finalidade de associá-los aos ciclos em um comando específico. Como exemplo, o ciclo de abertura de canal externo para usinagem cilíndrica, da linguagem interativa do comando MCS, adaptada de MCS I, 2003, é ilustrado na Fig. 3.23.

: CYC CALL 22 ABERTURA DE CANAIS			
X1	50.000	Z1	80.000
A	20.000	RECUO	7.000
B	12.500	A1	15.000
A2	0.200	RI	1.000
R1/-B1	2	R2/-B2	-1
W	5	FERR	0
N CANAIS	1	DIST	30.000

Figura 3.23 - Ciclo MCS para abertura de canais

Os parâmetros desse ciclo obtidos automaticamente no módulo gráfico, através de rotina desenvolvida em *AutoLisp* são o diâmetro de referência para início de usinagem (X1), posição central da base do primeiro canal (Z1), profundidade do canal (A) e largura no fundo do rasgo (B). A figura 3.24 ilustra o fluxograma de uma rotina *AutoLISP*.

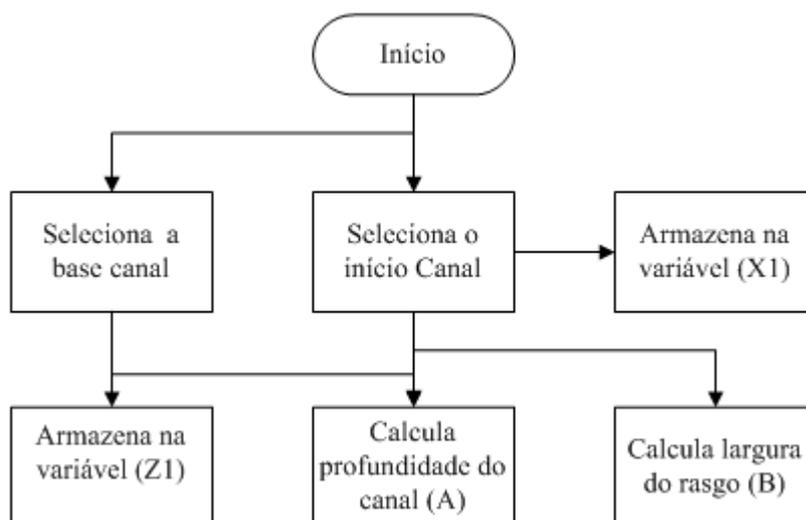


Figura 3.24 - Fluxograma da rotina *AutoLISP* para abertura de canal

As rotinas são executadas no ambiente CAD, na linha de comando, e carregadas no momento da inicialização do sistema. Tem como característica interagir com o usuário, através de mensagens indicando os procedimentos, bastando selecionar as entidades para serem extraídas as informações automaticamente. Os dados obtidos são armazenados em arquivo texto para serem acessados pelo sistema principal.

O acesso a objetos gráficos do *AutoCAD* pode ser realizado por meio de uma função chamada *Entity-Name* (código atribuído a cada um de seus elementos gráficos). Através dessa função é solicitada uma lista contendo todos os dados a respeito de um determinado objeto. Essa lista recebe o nome de *Entity-List* [Matsumoto, 2001]. A Figura 3.25 ilustra a exibição da *Entity-List*, para uma entidade “linha” do desenho, com os códigos *DXF*.

```

Command:
(setq el (entget en))
((-1 . <Entity name: 7ffffb05890>) // -1 é o código DXF
(0 . "LINE") // de Entity-Name
(330 .
  <Entity
  name: // "0" identifica tipo
  7ffffb039f0> // de elemento DXF
)
(5 . "1E9") // "8" é o código DXF
(100 . "AcDbEntity") // que identifica layer
(67 . 0)
(410 . "Model")
(8 .
  "0" // no caso de linhas, 10 é o
) // código DXF que identifica
(100 . "AcDbLine") // o primeiro vértice.
(10 2424.29 568.405 0.0)
(11 4277.84 1247.14 0.0)
(210 // no caso de linhas, 11 é o
  0.0 // código DXF que identifica
  0.0 // o segundo vértice.
  1.0
)
)

```

Figura 3.25 - Códigos DXF do objeto “linha”

O primeiro elemento do par associado de uma lista é conhecido como código DXF, pois segue a mesma regra de codificação definida pela *Autodesk* na geração de arquivos em formato DXF para troca de informação entre sistemas CAD. A função de acesso a objetos do CAD, que retorna a *Entity-List* do objeto identificado pelo *Entity-Name*, é “*entget*”.

A Figura 3.26 ilustra um trecho do código desenvolvido no editor de programação do *AutoLisp*, conforme fluxograma da Fig. 3.24, para captura dos pontos específicos usados como parâmetros de configuração do ciclo de abertura de canal da linguagem interativa do comando MCS. Os valores extraídos da entidade do desenho devido a aplicação da rotina são armazenados em arquivo texto. A Figura 3.27 mostra o arquivo texto gerado pela rotina.

```
(defun C:canal ()
  (setvar "OSMODE" 35)
  (setq pontoxmaior (getpoint "\nClique Inicio Canal: "))
  (setq X0 (rtos (* 2 (cadr pontoxmaior)) 2 3))
  (setq X01 (* 2 (cadr pontoxmaior)))
  (setq OBJ (entsel "\nSelecione base do canal: "))
  (setq EN (car OBJ))
  (setq EL (entget EN))
  (setq p1 (assoc 10 EL))
  (setq p11 (cdr p1))
  (setq p111 (car p11))
  (setq p2 (assoc 11 EL))
  (setq p22 (cdr p2))
  (setq p222 (car p22))
  (setq base (rtos (abs (- p111 p222)) 2 3))
  (setq Z1 (rtos (/ (+ p111 p222) 2) 2 3))
  (setq X02 (* 2 (cadr p22)))
  (setq prof_canal (rtos (- X01 X02) 2 3))
  (setq nome (findfile "canal.txt"))
  (setq fd (open nome "w"))
  (write-line X0 fd)
  (write-line Z1 fd)
  (write-line prof_canal fd)
  (write-line base fd)
  (close fd)
)
```

Figura 3.26 – Ilustração com trecho do programa ciclo de abertura de canal

```
Valores extraido para
ciclo de canal

Valor de X1 = 12.5
Valor de Z1 = -13
Valor de A = 4
Valor de B = 5
```

Figura 3.27 - Arquivo texto gerado pela rotina abertura de canal

A Figura 3.28 ilustra trechos de outras rotinas desenvolvidas em *AutoLISP* e os arquivos gerados.

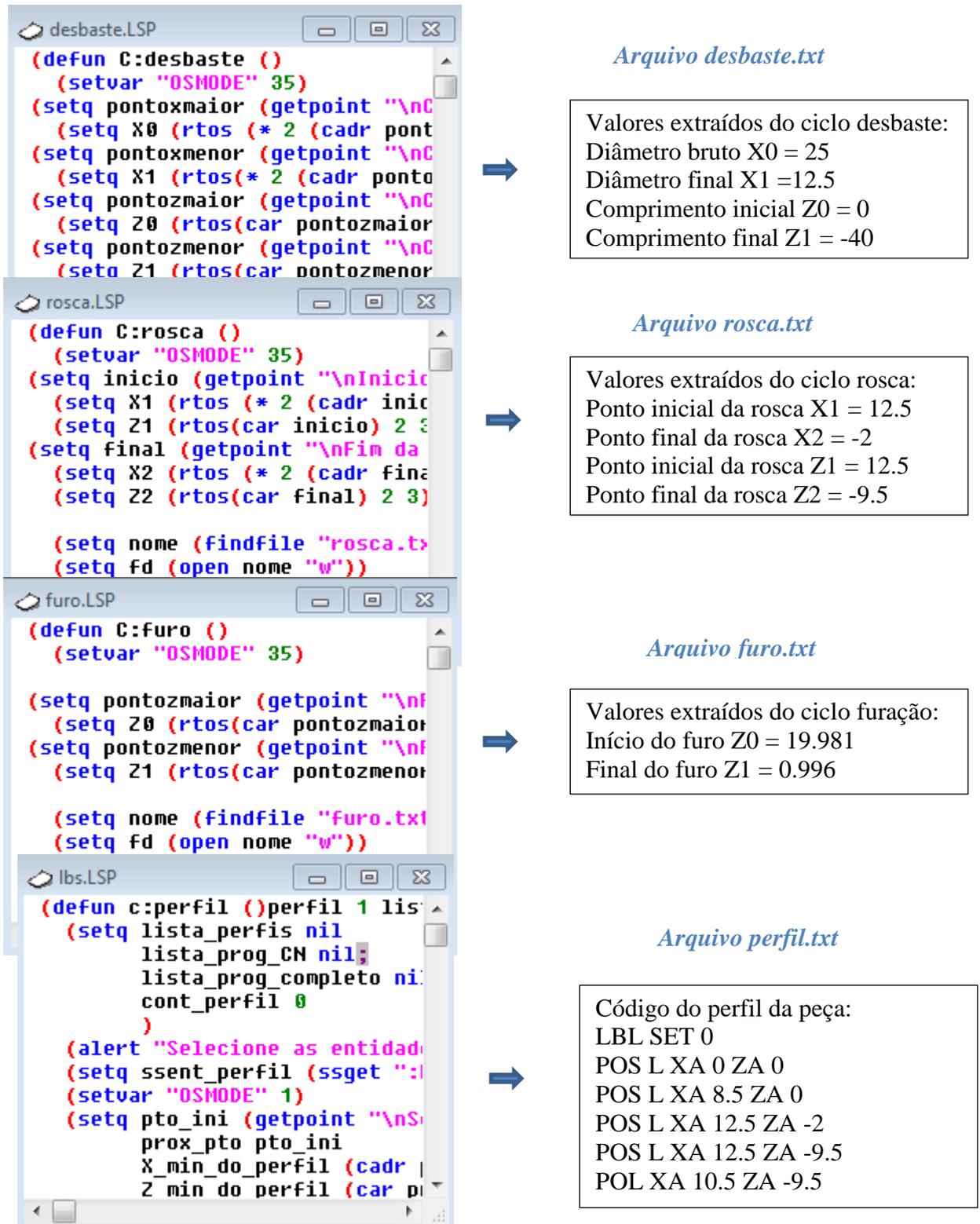


Figura 3.28 - Exemplos de rotinas desenvolvidas

3.5 Módulo dos ciclos de usinagem - MCU

No contexto do sistema computacional, o módulo MCU tem como objetivo a geração de programas CNC dedicados a um processo produtivo. Utiliza ciclos de usinagem disponíveis do comando especificado, dados geométricos obtidos automaticamente de ambiente CAD e dados tecnológicos disponibilizados por um banco de dados de parâmetros de corte, implementados no módulo MOPC.

Os ciclos de usinagem desenvolvidos para auxiliar a programação MDI podem ser diferenciados, em relação os parâmetros necessários para sua execução, mesmo realizando a mesma operação. A Figura 3.29 mostra alguns ciclos de comandos diferentes, que realizam o mesmo procedimento de usinagem, o desbaste longitudinal.

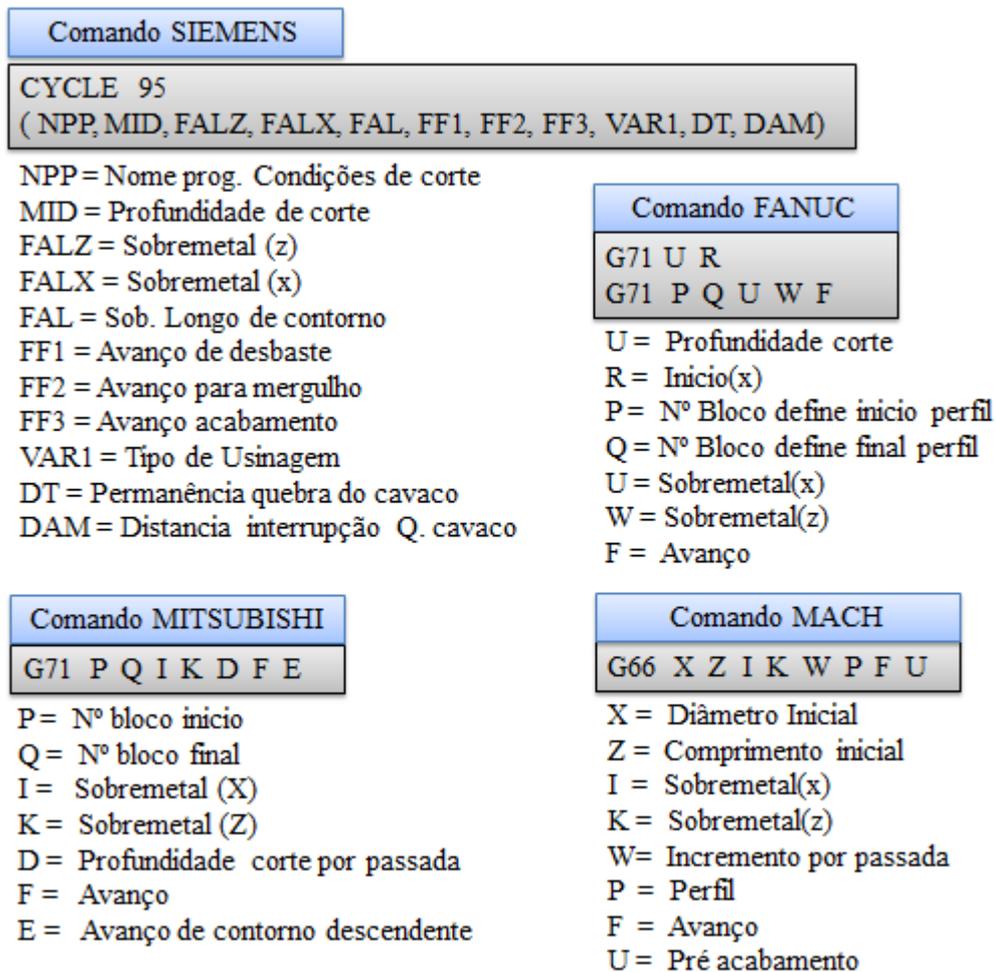


Figura 3.29 - Exemplos de ciclos de desbaste em diferentes comandos CNC

O aplicativo pode ser adaptado com ciclos de qualquer comando especificado, sendo necessária a configuração e programação no sistema. O protótipo desenvolvido para validação da metodologia foi configurado para comando MCS.

As interfaces gráficas desenvolvidas, correspondentes aos ciclos de usinagem disponíveis para a linguagem interativa no comando MCS, foram programadas em linguagem *Delphi*, mantendo-se a formatação descrita nos manuais do comando. Esse procedimento é realizado para adaptação do sistema a qualquer outro comando específico. Outra característica implementada nas interfaces gráficas desenvolvidas é a disponibilização automática de recursos oriundos do MOPC e do MDG. A seguir é abordada a metodologia aplicada no desenvolvimento dos ciclos para o protótipo computacional.

3.5.1 Ciclo do cabeçalho padrão

O programa CNC, para o comando MCS, necessita de uma linha inicial com a função “CYC CALL 0”. Essa função tem por objetivo zerar os parâmetros da máquina. A opção de utilizar cabeçalho padrão é normalmente usada por programadores, pois garante a segurança na troca de ferramenta, velocidades de operação, etc.

As funções miscelânea, declaradas no cabeçalho, alteram as condições da máquina. As mais utilizadas são “M39”, “M28” e “M68”, que significam, respectivamente, gama alta de rotações, contra ponto recuado e placa fechada. Foi desenvolvido um formulário no sistema relacionando às funções miscelâneas, com o objetivo de ajudar o programador. A Figura 3.30 ilustra esse formulário.

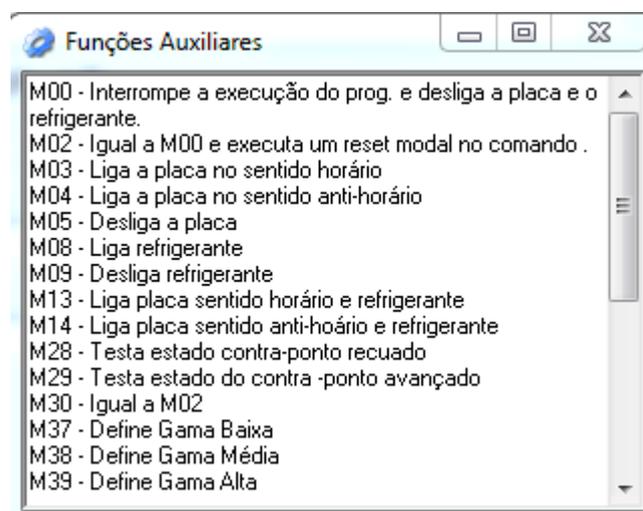


Figura 3.30 - Funções miscelânea

Outras funções devem ser definidas no cabeçalho do programa. Uma delas é a definição da ferramenta “T”, utilizada no processo de usinagem, juntamente com o corretor “D” correspondente. Para a rotação do torno, através da função “M59” ou velocidade de corte constante “M58”. A Figura 3.31 ilustra a interface gráfica desenvolvida para configuração do cabeçalho padrão.

Uma característica da interface gráfica do cabeçalho padrão é disponibilizar automaticamente os resultados das velocidades de corte calculadas pelo módulo MOPC. O programador pode aceitá-las ou interferir manualmente na configuração.

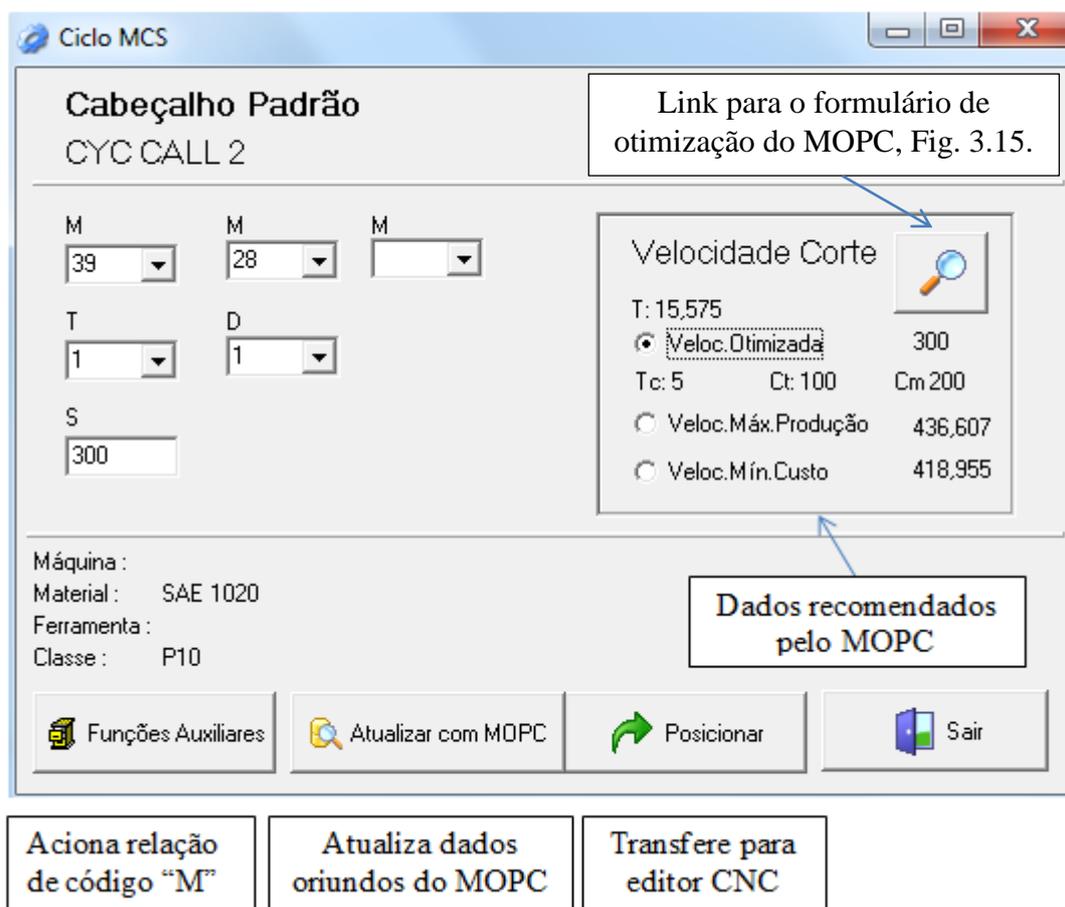


Figura 3.31 - Cabeçalho padrão vinculado aos parâmetros calculados no MOPC

3.5.2 Ciclo de posicionamentos

No comando MCS, as interpolações lineares e circulares são definidas pela função “POS”. A interface desenvolvida para posicionar a ferramenta de corte disponibiliza esses recursos com suas variações. Dados necessários para a programação, como o avanço de corte (f), são recomendados oriundos do sistema MOPC. A Figura 3.32 ilustra a interface de

posicionamento, com posicionamento circular, sentido horário, valores para coordenadas absolutas, avanço recomendado pelo módulo de otimização de parâmetros de corte e outros dados de configuração.

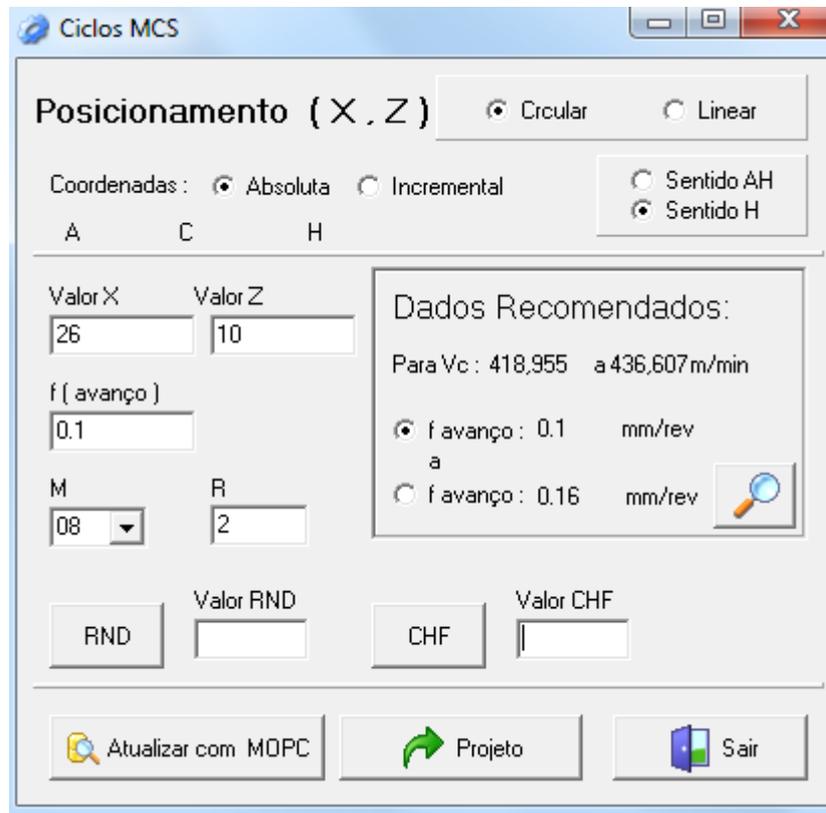


Figura 3.32 - Interface de posicionamento

3.5.3 Ciclo de desbaste longitudinal

Este ciclo tem por função gerar um perfil contínuo e externo no sentido longitudinal da peça. A geometria a ser usinada é gerada pelo movimento da ferramenta através de pontos descritos em um perfil e referenciado no final do programa.

Nas interfaces gráficas desenvolvidas, os parâmetros solicitados pelos ciclos que estiverem com a caixa de texto sombreada podem ser obtidos automaticamente do sistema gráfico disponível, bastando acionar o botão “CAD”. Os dados geométricos do desenho serão preenchidos automaticamente no formulário.

A variável “X0” representa o diâmetro da peça em bruto, “Z0” posição inicial do movimento, “X1” diâmetro final da peça e “Z1” comprimento final. Esses valores são obtidos pelo módulo gráfico através da execução da rotina específica e preenchidos da interface automaticamente. A Figura 3.33 ilustra a interface do ciclo de desbaste longitudinal.

The screenshot shows a software window titled "Ciclo MCS" with a subtitle "Desbaste Longitudinal" and "CYC CALL 24". The window contains several input fields for parameters:

- X0: 25
- X1: 12
- INCR: 1.5
- SOBREM X: 0.1
- TIPO: 1
- Z0: 2
- Z1: 40
- PERFIL: 1
- SOBREM Z: 0.2
- FERR: 3

There is also a "CAD" button with a folder icon. At the bottom, there are two buttons: "Projeto" with a green arrow icon and "Sair" with a blue window icon.

Figura 3.33 - Desbaste longitudinal

A variável “INCR” representa a quantidade de material no raio a ser removido por passe e depende principalmente da ferramenta que está realizando o desbaste, bem como do material usinado. Já a variável “PERFIL” indica o número do perfil que será usinado.

Os pontos do perfil estão descritos no final do programa CNC, após a função “M30” que determina o final do programa. Para que o ciclo de desbaste longitudinal interprete o perfil, é necessário que se escreva a função “LBS SET 1”. Essa função representa o início do perfil 1. Para que o sistema entenda qual é o último ponto do perfil é necessário que após esse ponto se escreva a opção “LBS SET 0”.

As variáveis “SOBREM X” e “SOBREM Z” determinam o sobremetal. O “Tipo” pode ser desbaste (0), desbaste e acabamento (1) e acabamento (2). A variável “FERR” indica que a ferramenta está posicionada no lado positivo do eixo “X”, caso for considerado o valor de (0) e (1) indicando lado negativo.

3.5.4 Ciclo do perfil da peça usinada

A extração do perfil a ser usinado no módulo gráfico é realizada exclusivamente por rotina de programação desenvolvida para o protótipo CAD/CAM. Foi elaborada uma interface no sistema para resgatar os dados e analisá-los antes de inserir no texto do editor de código CNC. A determinação do perfil é fundamental para o desenvolvimento do programa. O ciclo de desbaste longitudinal não funciona sem estar referenciado a um perfil da peça. A Figura 3.34 ilustra a interface de chamada e análise do perfil extraído do módulo gráfico.

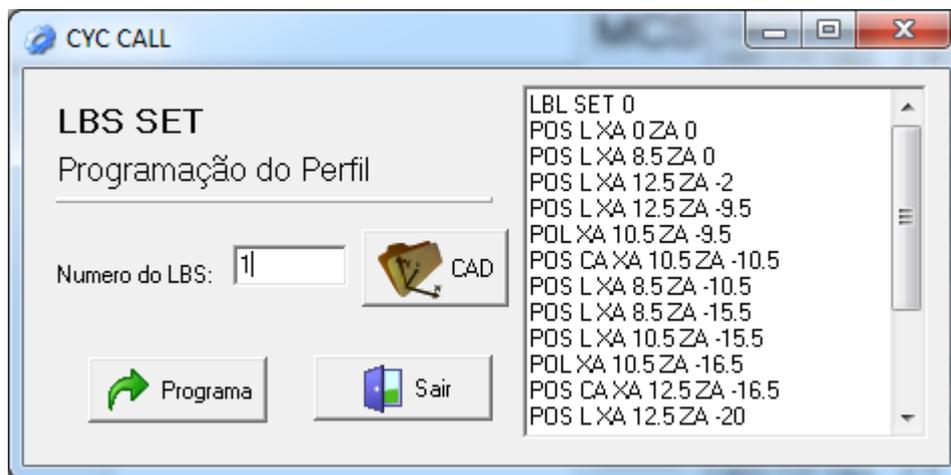


Figura 3.34 - Interface para captura do perfil da peça

3.5.5 Ciclo de rosqueamento automático

Este ciclo permite a abertura de roscas paralelas ou cônicas, internas ou externas, com movimento de corte no sentido positivo do eixo longitudinal “Z” e profundidade no eixo transversal “X”.

No ciclo de abertura de rosca, o passo é assimilado pela variável “PASSO”. A variável “ACABAMEN” representa a quantidade de material que será tirada no último passe da rosca. Esse valor irá depender principalmente do material que se está usinando e das condições de corte. A variável “PROF” representa a profundidade do canal da rosca, ela pode ser calculada a partir do passo. A variável “TIPO” permite três opções: ferramenta com entrada pelo centro da rosca representada pelo valor (0), com entrada pela lateral da rosca representada pelo valor (1) e com entrada da ferramenta, alternadamente, pelas laterais da rosca representada pelo valor (2). A variável “N PASSA” representa o número de passada que a ferramenta irá dar para retirar todo o material da profundidade da rosca. A variável “N VAZIO” representa o

número de passadas que a ferramenta realiza em vazio no canal, sem a retirada de rebarba. A variável “ANG SAI” representa o ângulo para saída no recuo da ferramenta. Para o valor (0), a ferramenta afasta-se em ângulo de 90°, (1) ângulo de 45° e (2) com 60°. A variável “FERR” tem a mesma função que no ciclo de desbaste longitudinal. Ela representa o lado que a ferramenta encontra-se. A Figura 3.35 ilustra a interface gerada para o ciclo de rosqueamento.

Dados extraídos automaticamente do módulo gráfico.

Aciona formulário de “Ajuda” para preenchimento dos parâmetros do ciclo

Parâmetro	Valor
X1	12
X2	12
Z1	20
Z2	20
AFAST	1
PASSO	0.214
PROF	0.64Xp
ACABAMEN	0
TIPO	0
ANG FERR	0
N PASSA	40
N VAZIO	2
ANG SAI	0
FERR	0

Figura 3.35 - Ciclo de rosqueamento automático

Os valores de dimensionamento para o início da rosca “X1”, “Z1” e para o final da rosca “X2”, “Z2” são extraídos automaticamente do módulo gráfico.

Os dados geométricos extraídos do módulo MDG são armazenados em arquivo texto, os quais são lidos posteriormente pelo protótipo CAD/CAM, quando acionado o botão “CAD” do formulário do ciclo.

A Figura 3.36, adaptada de MCS II, 2010, ilustra a interface desenvolvida para auxílio do programador no preenchimento do formulário do ciclo.

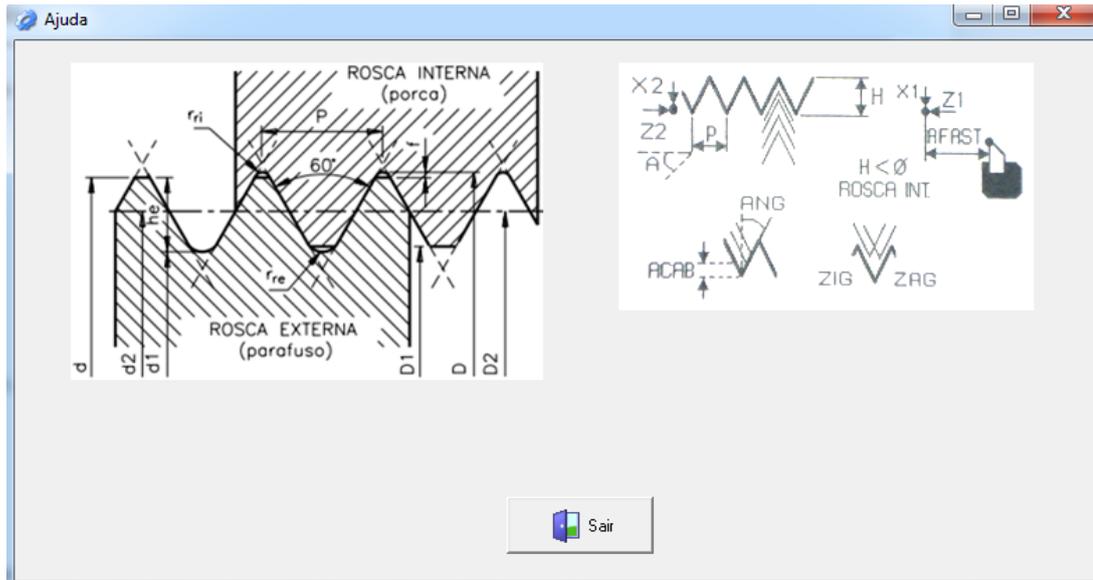


Figura 3.36 - Formulário de “Ajuda” do ciclo de rosqueamento

3.5.6 Ciclo de furação profunda

O ciclo de furação profunda permite a execução de furos no centro da peça, com incrementos para alívio de material e tempo de espera ao final da furação.

A variável “Z0” representa a posição de aproximação rápida da broca à peça. A variável “Z1” representa o ponto onde ocorrerá a primeira remoção de material da peça, nesse caso, a face da peça. Esses valores são extraídos do módulo gráfico e preenchidos automaticamente no formulário. A Figura 3.37 ilustra a interface gerada para o ciclo de furação. Essa interface também se caracteriza por estar interligada ao módulo MOPC, sendo recomendadas para o programador as velocidades oriundas do MOPC.

A variável “PROF” representa a profundidade total do furo. A variável “DIAM” o diâmetro da broca que irá realizar a furação. Já a variável “INCR” armazena a porção de material que a broca irá tirar sem parar em avanço de corte, realizando o retorno em avanço rápido após esse incremento. A variável “V. CORTE” representa a velocidade de corte da broca que estará sendo utilizada. Com essa variável e o diâmetro da broca, o ciclo encarrega-se de calcular a rotação necessária para a placa do torno. O “TEMPO” representa o tempo de espera para retorno ao final da execução do furo, em décimos de segundos. A Figura 3.38, adaptada de MCS II, 2010, ilustra a interface do formulário de “Ajuda” do ciclo de furação profunda.

CYC CALL

CYC CALL 23
Furação

Z0 Z1 

PROF INCR

DIAM V.CORTE

TEMPO

Velocidade Corte 

T: 15,575

Veloc.Otimizada 100

Tc: 0 Ct: 0 Cm: 0

Veloc.Máx.Produção 236,607

Veloc.Mín.Custo 160,45

 Ajuda  Atualizar com MOPC  Projeto  Sair

Figura 3.37 - Ciclo de furação profunda

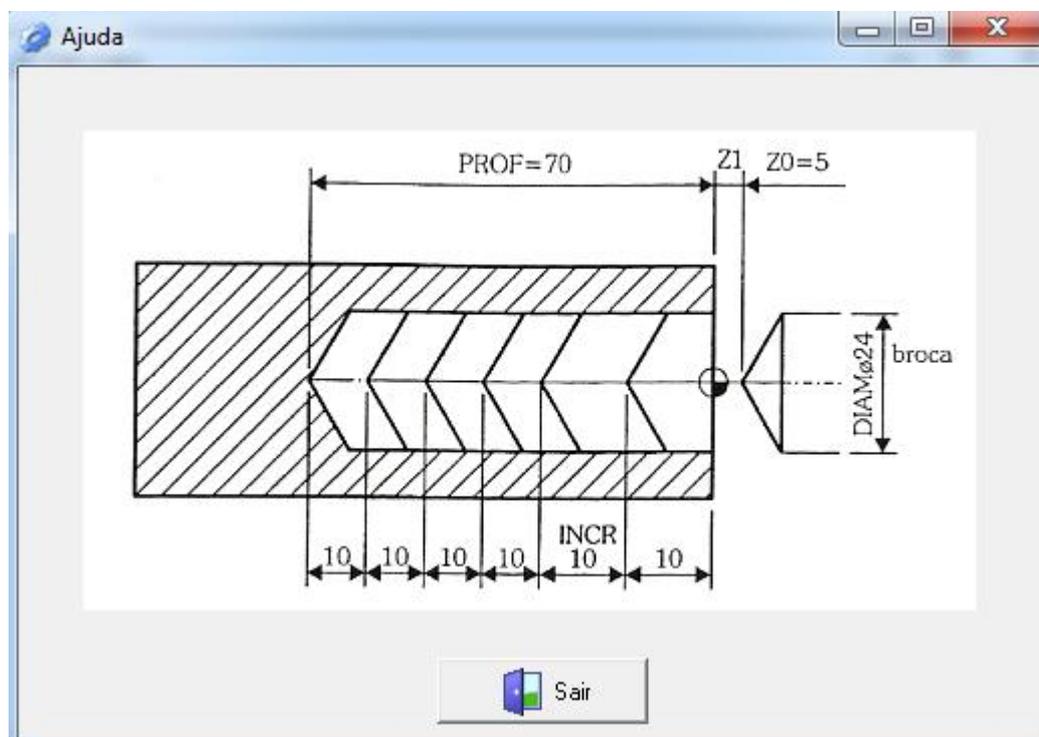


Figura 3.38 - Formulário de “Ajuda” do ciclo de furação profunda

3.5.7 Ciclo de abertura de canais

O ciclo de abertura de canais permite a abertura de um ou mais canais espaçados igualmente com a inserção de raios ao fundo, desde que sejam iguais para as duas laterais. Permite também a inserção de raios nas pontas superiores do canal ou diferentes ângulos de chanfros nas bordas.

A variável “X1” indica o diâmetro de referência para início da usinagem, ou seja, o ponto em relação as cota “X” onde inicia a remoção de material por parte do ciclo. Esse ponto pode ser interno ou externo dependendo da aplicação do ciclo. A variável “Z1” estabelece a posição central da base do primeiro canal em relação à cota “Z” do torno “A” define a profundidade do canal em milímetros e “B” representa a largura no fundo do rasgo, este valor refere-se à parte plana do fundo do canal somando-se os raios de arredondamento caso existam. Essas variáveis são extraídas do módulo gráfico, através da aplicação da rotina específica do canal e preenchidas automaticamente na interface do ciclo do canal. A Figura 3.39 ilustra a interface desenvolvida para o ciclo do canal.

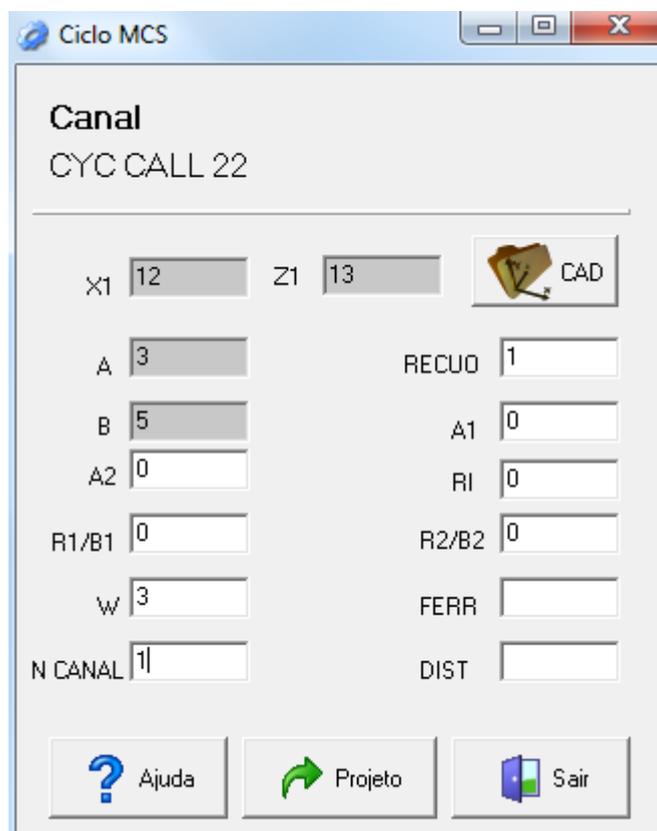


Figura 3.39 - Ciclo de canal

A variável “RECUO” estabelece o incremento inicial para a abertura do canal para alívio de material. “R1/-B1” e “R2/-B2” definem os raios nos cantos externos das faces esquerda e direita do canal, sendo que “R” representam valores para raios de arredondamentos, e B representam valores de chanfros. “W” indica a largura da ferramenta de corte. “N CANAL” representa o número de rasgos. “DISTANCIA” estabelece a distância entre centros de separação de canais. A Figura 3.40, adaptada de MCS II, 2010, ilustra a interface do formulário de “Ajuda”, do ciclo de furação profunda.

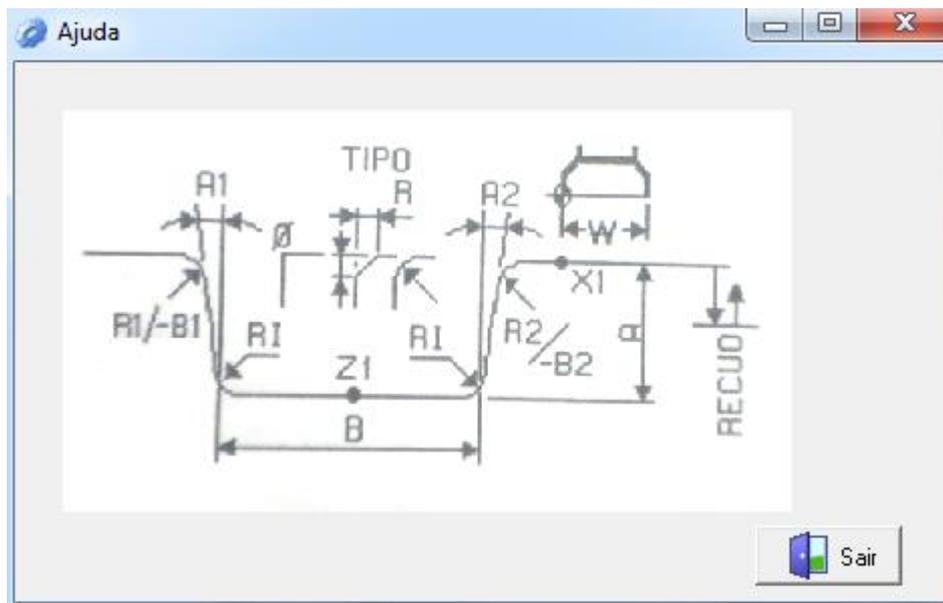


Figura 3.40 - Formulário de “Ajuda” do ciclo de canal

3.6 Caracterização do protótipo computacional

A interface principal do aplicativo computacional integra todos os módulos descritos anteriormente. É responsável pela abertura e fechamento dos bancos de dados, da execução dos módulos individualmente, através de atalhos dispostos na forma de botões de acionamentos. O editor do programa CNC, posicionado à direita da tela, é apresentado nessa interface principal, com seus recursos de edição. Os ciclos de usinagem são ativados depois da seleção do tipo de comando específico selecionado. Uma folha de dados referente aos itens calculados e selecionados no MOPC é posicionada a esquerda da interface principal. A Figura 3.41 ilustra a interface principal do protótipo. Todos os módulos do sistema estão integrados ao formulário principal. A Figura 3.42 ilustra uma etapa de trabalho, mostrando o módulo gráfico, um ciclo de usinagem e código de programação gerado no editor.

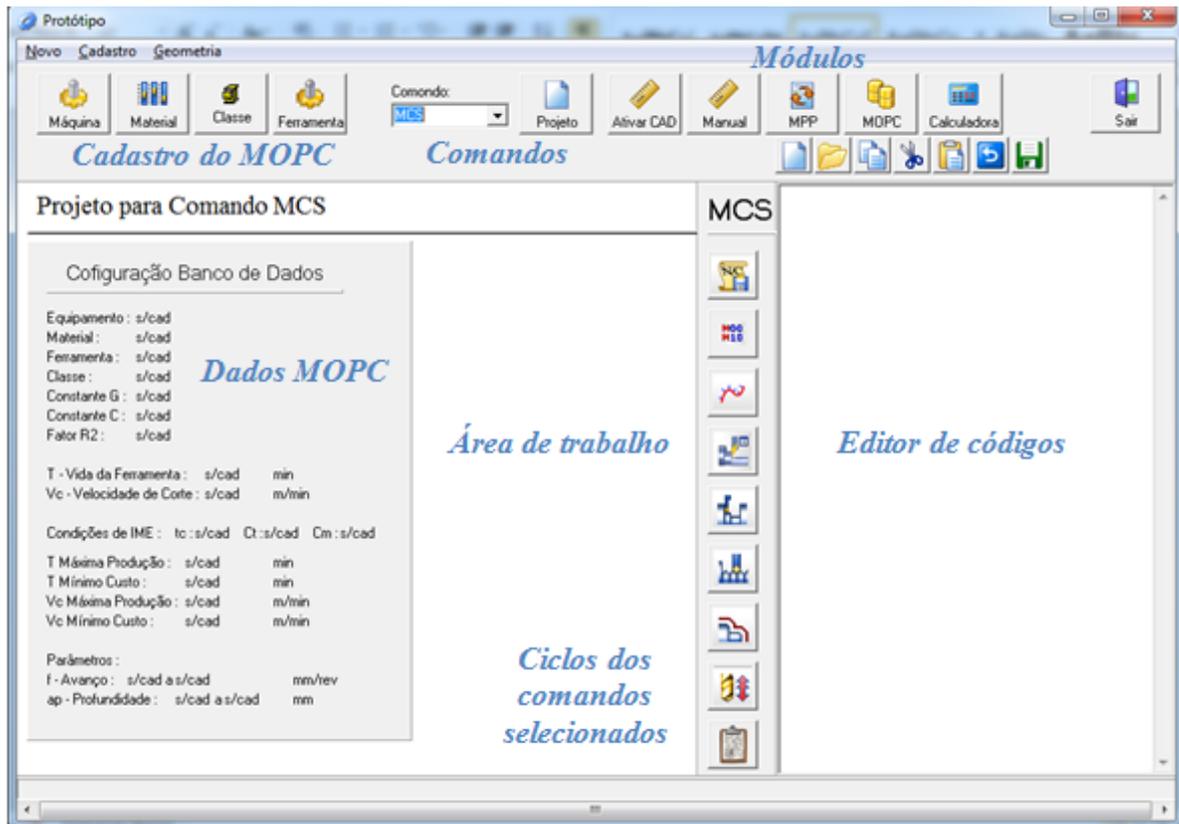


Figura 3.41 –Interface principal do sistema

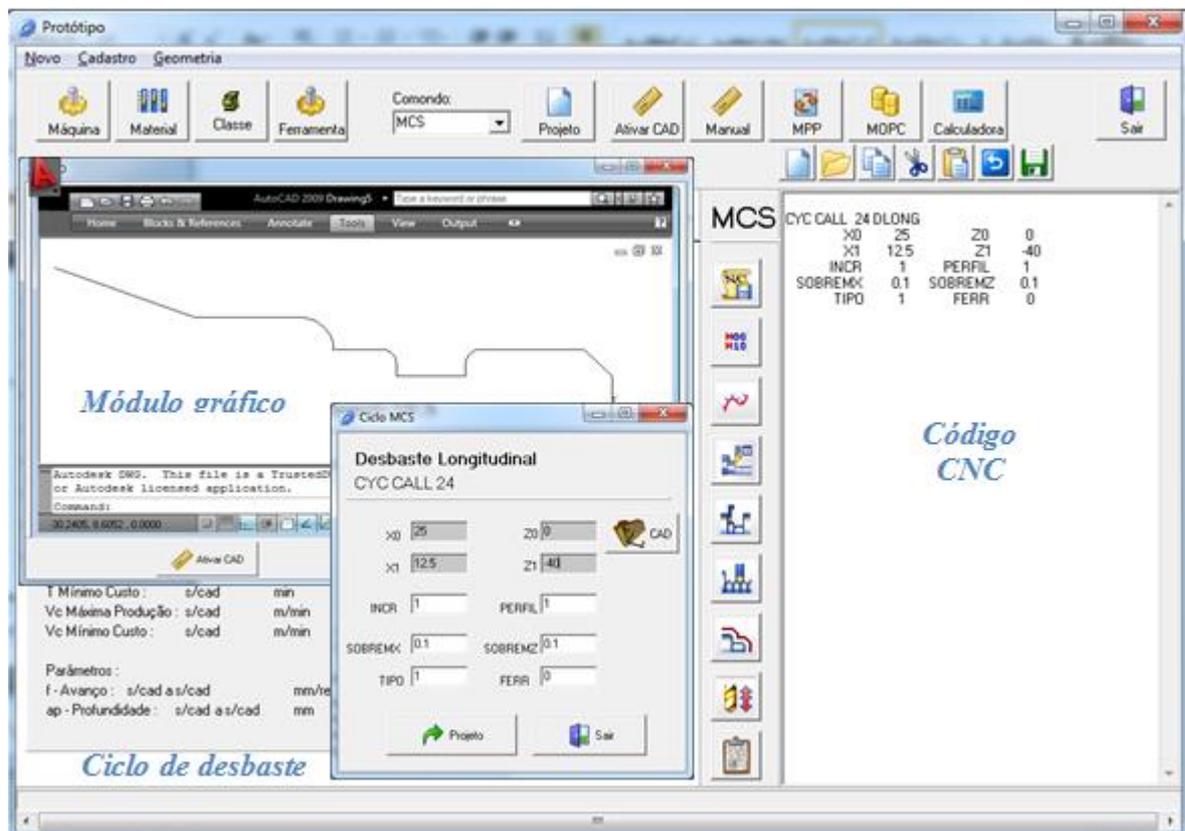


Figura 3.42 - Interface principal com aplicativos executando

4 APLICAÇÃO E AVALIAÇÃO DOS MÓDULOS DO SISTEMA

Neste capítulo, é analisada a aplicabilidade dos módulos individualmente e a operacionalidade do sistema em um processo produtivo

4.1 Aplicação do Módulo de otimização de parâmetros de corte (MOPC)

Como a realização de ensaios experimentais para alimentar o MOPC não faz parte do escopo deste trabalho, para a análise da eficiência desse módulo, sistemas produtivos foram simulados através de dados extraídos de publicações científicas. A título de validação do sistema implementado, apresenta-se a seguir um estudo comparativo entre os resultados obtidos através do sistema computacional proposto e os obtidos em trabalhos científicos que utilizam técnicas alternativas no cálculo das constantes de Taylor e/ou tempo de vida da ferramenta. Os trabalhos de pesquisa considerados na análise são dados do CINFUS (Centro de Informação de Usinagem) [Stemmer, 2007], e dados experimentais publicados por Pereira, 2004 e Dos Santos *et al.*, 1999.

4.1.1 Estudos relacionados no cálculo do banco de dados CINFUS

As diversas variáveis que envolvem os processos de usinagem podem levar a resultados diversificados. Além da possibilidade da especificação de condições de usinagem, através da orientação em catálogos de fabricantes de ferramentas, também pode-se recorrer a bancos de dados de usinagem, como METCUT (Machining Data Handbook - Metcut Research Associates Inc.) nos Estados Unidos, SANDVIK na Suécia e INFOS da Sociedade EXAPT na Alemanha [Stemmer, 2007]. Banco de dados análogo, para as condições brasileiras, foi desenvolvido por Boehs, 1988, no Departamento de Engenharia Mecânica da Universidade Federal de Santa Catarina. Esse banco, denominado CINFUS, baseia-se na fórmula generalizada de Taylor.

A Tabela 4.1 mostra dados de usinagem, obtidos no CINFUS, para material ABNT (FC-100), ferramenta de forma SNUN 120408 e classe K10. Os coeficientes para a Eq. (3.3), indicados para essas condições, são $C_e = 529.10$, $G = -0.199$, $E = -0.181$, $F = -0.144$, $H = 0.299$. O tempo de vida da ferramenta foi calculado utilizando a Eq. (3.3) e adicionado na última coluna. Observa-se que, somente na primeira linha da tabela, o valor do tempo de vida

da ferramenta é significativamente diferente. Nos demais itens, o resultado é coerente com as constantes indicadas para a equação de Taylor pelo sistema CINFUS.

Tabela 4.1 - Cálculo do tempo de vida da ferramenta para dados do CINFUS [Adaptada de Stemmer, 2007]

vc(m/min)	ap(mm)	f(mm/rot)	VB _B (mm)	T(min)	T(min) Calculado
250,00	1,00	0,25	0,30	20,00	25,01
236,50	2,00	0,25	0,30	20,00	20,02
223,10	3,00	0,25	0,30	20,00	20,01
214,10	4,00	0,25	0,30	20,00	19,99
207,30	5,00	0,25	0,30	20,00	20,00
201,90	6,00	0,25	0,30	20,00	20,02
197,50	7,00	0,25	0,30	20,00	20,00
193,70	8,00	0,25	0,30	20,00	20,02

O valor de “C”, utilizado na Eq. (3.5), que é base de cálculo do MOPC, é extraído da Eq. (3.4) quando se estabelece como constantes os demais termos, de acordo com a Eq. (4.1).

$$C = C_e \cdot f^E \cdot a_p^F \cdot VB^H = (529.10) \cdot 0,25^{-0.181} \cdot 4^{-0.144} \cdot (0.30)^{0.299} = 388,57 \quad (4.1)$$

A Figura 4.1 ilustra a interface de filtro dos dados do módulo MOPC, onde são selecionados os parâmetros para extrair os dados do banco CINFUS, recalculados para profundidade de corte de 4 mm.

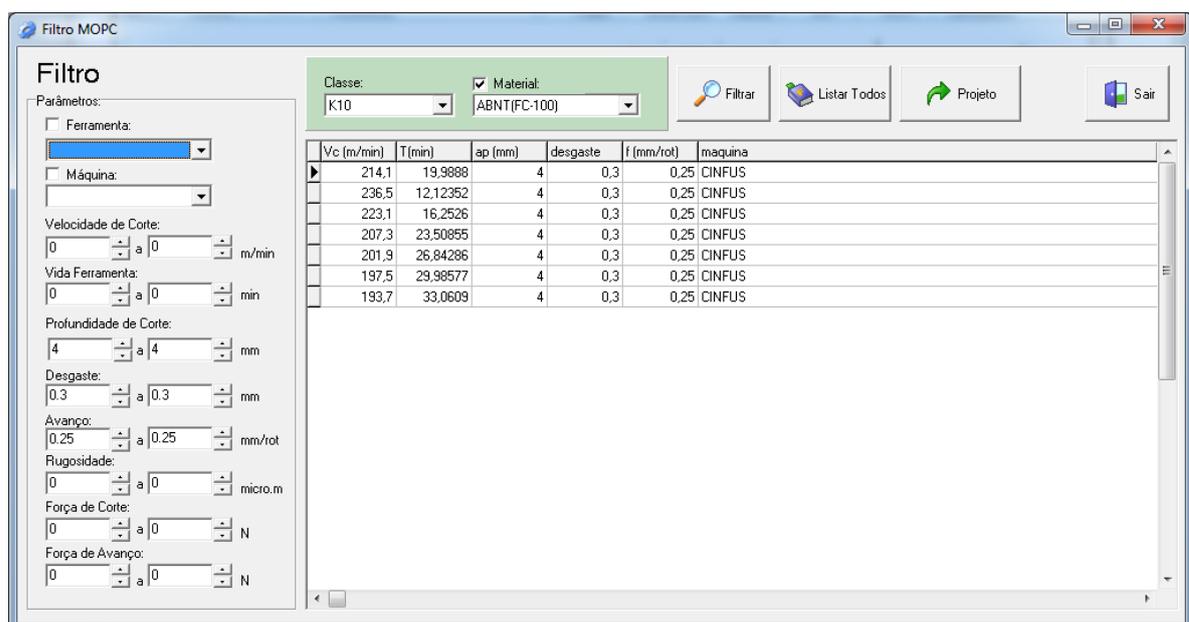


Figura 4.1 - Interface com filtro de dados CINFUS

A profundidade de corte está configurada para filtrar de 2 a 8 mm, eliminando a primeira linha da Tab. (4.1), que contém resultado significativamente diferente para a vida da ferramenta.

A Figura 4.2 ilustra detalhe da interface de otimização de dados do módulo MOCP, com os valores calculados de “C” e “G”, coeficientes de Taylor da Eq. (3.5), e o coeficiente de correlação r^2 , que nesse caso é 1,0 porque os valores apresentados na referência não são os dados experimentais para calcular os expoentes, mas sim os valores calculados para a velocidade de corte aplicando-se a equação expandida de Taylor com os expoentes processados. A Tabela (4.2) mostra um comparativo dos resultados.

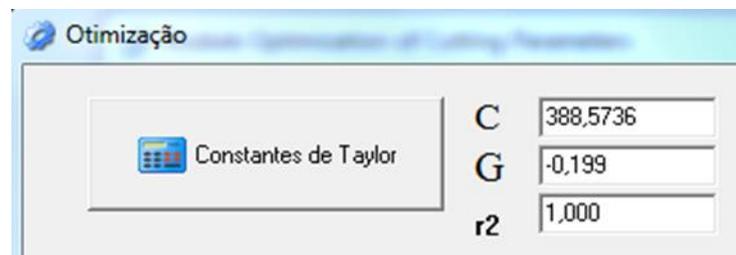


Figura 4.2 - Cálculo dos coeficientes “C” e “G” de Taylor

Tabela 4.2 - Comparativo de resultados

COMPARATIVO DOS RESULTADOS	
Valor da constante "Ce" calculado pelo CINFUS =	529,10
Aplicando a Eq. (4.1) para determinar "C" =	388,57
Valor calculado para "C" pelo MOPC =	388,57
Valor da constante "G" calculado pelo CINFUS =	-0,199
Valor calculado para "G" pelo MOPC =	-0,199

Os valores calculados pelo sistema, neste exemplo, mostram que, apesar de adotar a equação simplificada de Taylor para modular o processo, a metodologia de processamento adotada chega a resultados que equivalem à aplicação dessa equação na sua forma expandida.

4.1.2 Estudos relacionados ao cálculo do tempo de vida da ferramenta

Outros dados experimentais utilizados para análise do MOPC referem-se ao trabalho de Pereira, 2004, que determinou modelos de cálculo do tempo de vida de ferramenta no torneamento do aço ABNT 52100 endurecido utilizando a Metodologia de Superfície de Resposta (DOE), um conjunto de técnicas estatísticas que permite a análise de vários fatores de influência de um processo de maneira simultânea. Modelos de cálculos semelhantes foram utilizados por Sahin, 2009, para comparação da vida útil da ferramenta de CBN (*Cubic Boron Nitride*) e de Cerâmica, e por Asilturk e Akkus, 2011, para determinar o efeito de parâmetros de corte sobre a rugosidade de superfície dura no torneamento usando o método *Taguchi*.

Os ensaios de Pereira, 2004, foram realizados em um Torno CNC Nardini Logic 175, insertos intercambiáveis de cerâmica mista, *Sandvik*, recoberta com uma camada de Nitreto de Titânio, com geometria ISO CNGA 12 04 08 S01525, chanfrado para evitar grandes choques e quebra de ferramenta e classe H05. Como critério de fim de vida da ferramenta foi adotado o desgaste de flanco $VB_B = 0,3$ mm.

Através da elaboração de um arranjo fatorial completo com quatro *center points*, foi realizada a análise dos dados obtidos experimentalmente, mostrados na Tabela (4.3). Nesses ensaios, foram alternados os parâmetros de usinagem, com velocidade de corte (v_c) entre os níveis 200 m/min e 240 m/min, avanço (f) entre os níveis 0,05 mm/rot e 0,10 mm/rot e profundidade de corte (a_p) entre os níveis 0,15 mm e 0,30 mm.

Tabela 4.3 - Dados experimentais [Adaptada de Pereira, 2004]

vc	200	240	200	240	200	240	200	240	200	240	200	240	200	240	200	240
f	0,05	0,05	0,1	0,1	0,05	0,05	0,1	0,1	0,05	0,05	0,1	0,1	0,05	0,05	0,1	0,1
ap	0,15	0,15	0,2	0,2	0,3	0,3	0,3	0,3	0,15	0,15	0,15	0,2	0,3	0,3	0,3	0,3
T	15,8	10,9	9,2	8,5	12,4	7,2	7,9	6,8	17,7	12,1	10,5	8,5	10,6	7,7	8,5	5,7

A Equação (4.2) mostra a formulação obtida pelo modelo estatístico DOE para previsão da vida da ferramenta, a partir do modelo quadrático completo gerado pelo *Central Composite Design*. Substituindo as variáveis da Equação (4.2) por $v_c = 200$ m/min, $f = 0,05$ mm/rot e $a_p = 0,15$ mm (Tab. 4.3), que na tabela alcançou uma maior vida da ferramenta $T = 17,7$ min, obtém-se uma estimativa experimental [Pereira, 2004].

$$T = 115,1474 - 0,4687 \cdot v_c - 862,7411 \cdot f - 94,3462 \cdot a_p + 0,0006 \cdot v_c^2 + 2392,1875 \cdot f^2 + 77,2368 \cdot a_p^2 + 1,5 \cdot v_c \cdot f + 0,05 \cdot v_c \cdot a_p + 360 \cdot f \cdot a_p \quad (4.2)$$

Substituindo os valores por ótimos (Tab. 4.3), $v_c = 200$ m/min, $f = 0,05$ mm/rot e $a_p = 0,15$ mm, obtém-se

$$T = 15,3671 \text{ min}$$

A Figura 4.3 mostra a interface de filtro dos dados do módulo MOPC, onde estão selecionados os parâmetros para extrair dados, conforme experimento, fixando $a_p = 0,15$ mm e $f = 0,05$ mm/rot.

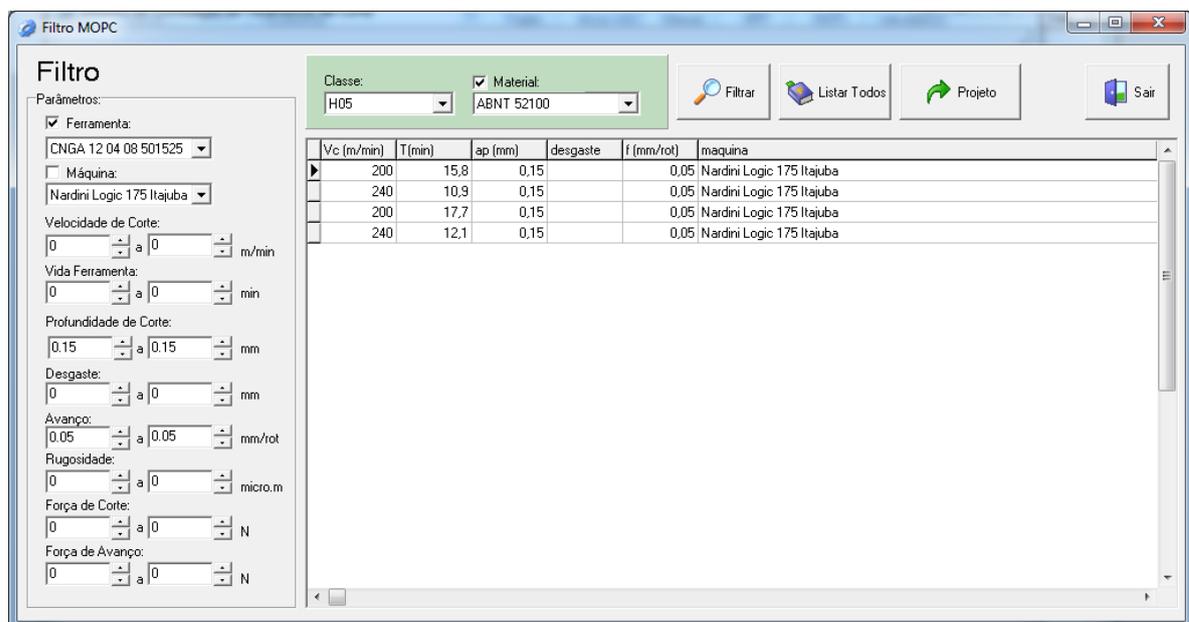


Figura 4.3 - Filtro das condições de corte

Com as condições de corte estabelecidas no filtro, é realizado o cálculo das constantes da equação de Taylor, possibilitando a entrada específica da velocidade de corte de 200 m/mm, tendo como resultado o tempo de vida estimado para a ferramenta de 16,738 min, como apresentado na interface da Fig. 4.4.

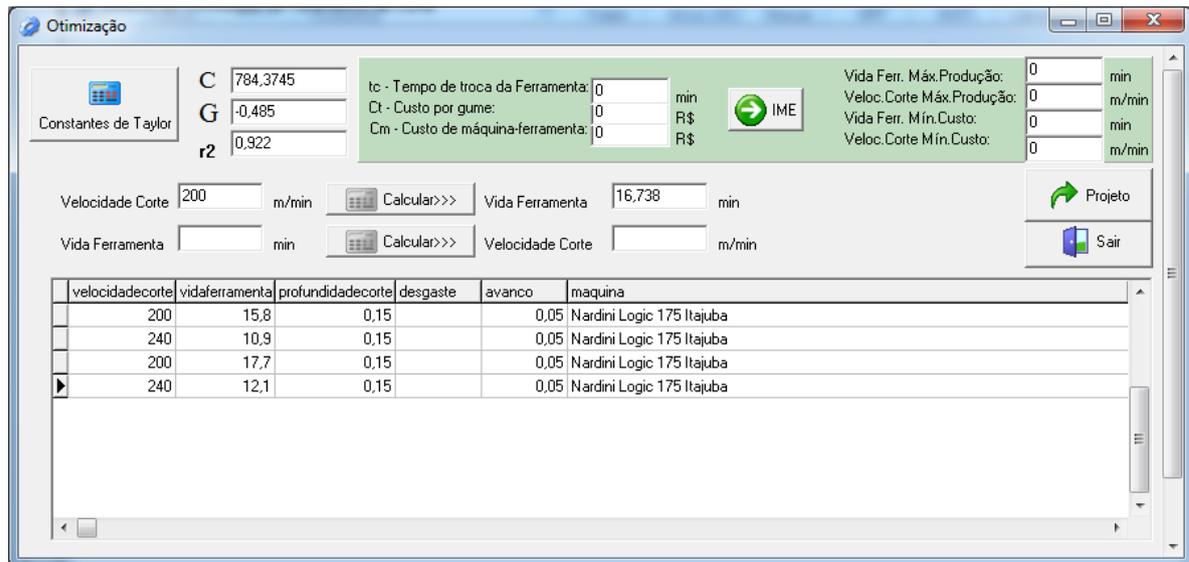


Figura 4.4 - Interface de otimização com resultados

A Tabela 4.4 compara o resultado do método proposto com valores para a condição de ótimo, ou seja, os níveis inferiores dos fatores V_c , f e ap que permitem que seja alcançada uma estimativa experimental para obter os valores de maior vida da ferramenta [Pereira, 2004], com valores obtidos pelo MOPC, para as mesmas condições de corte. Combinações de restrições na profundidade e velocidade de corte são realizadas através do recurso de filtragem do banco de dados.

Tabela 4.4 - Comparativo dos resultados

Parâmetros	Tempo de Vida da Ferramenta (minutos)		
	Valor Experimental Obtido p/ Pereira 2004	Método Modelo Pereira 2004	MOPC
$v_c = 200$ m/min $f = 0,05$ mm/rot $ap = 0,15$ mm	T = 17,7 min	T = 15,4 min	T = 16,7 min
$v_c = 240$ m/min $f = 0,05$ mm/rot $ap = 0,3$ mm	T = 7,2min	T = 5,71 min	T = 7,4 min
$v_c = 200$ m/min $f = 0,05$ mm/rot $ap = 0,3$ mm	T = 10,6 min	T = 15,4 min	T = 11,35 min

As análises dos valores mostram uma coerência entre os resultados obtidos por Pereira, 2004 e o sistema computacional MOPC. Mesmo sendo uma aproximação estatística do tempo de vida da ferramenta para os dados experimentais apresentados, a utilização da metodologia mostra uma convergência dos resultados.

4.1.3 Estudos relacionados ao cálculo das constantes da equação de Taylor

Dos Santos *et al.*, 1999, realizaram projeto experimental para calcular os coeficientes da equação de Taylor expandida para usinagem. Utilizaram um critério ideal experimental para identificação de parâmetros de usinagem de modelos não lineares de sistemas dinâmicos, baseado na minimização da razão entre máximos e mínimos valores singulares da matriz de sensibilidade [Duarte e Arruda, 1993]. Essa técnica permite calcular o melhor conjunto de parâmetros de usinagem (velocidade de corte, avanço e profundidade do corte) a ser utilizado para estimar os coeficientes da equação de Taylor. Logaritmos são tomados em ambos os lados da Eq. (3.4), a fim de transformá-la em uma estimativa linear Eq. (4.3).

$$Y = Xb = \begin{Bmatrix} \ln \text{VBMAX1} \\ \ln \text{VBMAX2} \\ \ln \text{VBMAX2} \\ \vdots \\ \vdots \\ \vdots \\ \ln \text{VBMAXn} \end{Bmatrix} = \begin{bmatrix} 1 & \ln v_1 & \ln f_1 & \ln a_1 & \ln T_1 \\ 1 & \ln v_2 & \ln f_2 & \ln a_2 & \ln T_2 \\ 1 & \ln v_3 & \ln f_3 & \ln a_3 & \ln T_3 \\ & & & \vdots & \\ & & & \vdots & \\ & & & \vdots & \\ 1 & \ln v_n & \ln f_n & \ln a_n & \ln T_n \end{bmatrix} \cdot \begin{Bmatrix} C \\ H \\ E \\ F \\ G \end{Bmatrix} \quad (4.3)$$

Utilizando-se o método dos mínimos quadrados em “b” da Eq.(4.3), tem-se o resultado estimado pela Eq. (4.4).

$$b = (X^T X)^{-1} (X^T Y) = V(X^T Y) \quad (4.4)$$

O problema da variável tempo “T”, na matriz, não ser conhecida para as condições de usinagem selecionadas é resolvido determinando “V_B” como limite do tempo de vida da ferramenta, transformando a variável em independente.

Quatro conjuntos de condições de corte são selecionados inicialmente, que é considerado o número mínimo de testes necessários para determinar os coeficientes da Eq. (3.4). Com os coeficientes determinados, é possível, para um determinado conjunto de

parâmetros de corte, o cálculo da estimativa da vida útil da ferramenta para um critério “ V_B ” dado.

Para o procedimento experimental, foi considerado aço AISI 1045, fresa *Sandvik R 260.22.080*, classe ISO P45-M35, CNC Romi-Interact IV fresadora com 15 kW de potência e o critério final de vida da ferramenta de $V_B = 0,7$ mm.

Para a análise realizada, são utilizados os dados experimentais com as constantes de Taylor calculadas e comparadas com resultados do sistema MOPC. A Tabela 4.5 mostra os parâmetros obtidos no experimento de Dos Santos *et al.*, 1999, com as respectivas constantes, os quais são aplicados a Eq. (3.4) resultando nos valores da coluna de tempos de vida das ferramentas, indicadas na tabela.

Tabela 4.5 - Tempo calculado para os dados [Adaptada de Dos Santos *et al.*, 1999]

Nº	Parâmetros				Constante Taylor Expandida					Vida Ferramenta
	V_c	f	a_p	V_B	H	E	F	G	C	T
1	116,52	0,07	2,25	0,70	0,46	-0,01	-0,20	-0,27	475,17	59,15
2	179,91	0,17	1,38	0,70	0,44	-0,03	-0,16	-0,28	474,15	17,73
3	250,00	0,12	1,60	0,70	0,43	-0,02	-0,17	-0,28	470,63	4,88
4	171,89	0,16	1,48	0,70	0,44	-0,05	-0,29	-0,29	482,44	19,21
5	204,47	0,14	1,54	0,70	0,43	-0,05	-0,30	-0,30	491,89	10,58
6	207,65	0,15	1,52	0,70	0,44	-0,05	-0,31	-0,31	523,89	10,86

Os dados da Tabela 4.5 são lançados no banco de dados do MOPC para cálculo das constantes da equação reduzida de Taylor. Para o primeiro cálculo pelo sistema, são tomadas todas as velocidades e tempos de vida da ferramenta correspondentes sem restrições dos outros parâmetros.

A Tabela 4.6 mostra um comparativo entre os valores calculados pelo método em análise [Dos Santos *et al.*, 1999] e os resultados obtidos pelo MOPC.

Tabela 4.6 - Comparando método com MOPC

Nº	Coeficiente "G"			Coeficiente "C"			
	Método	MOPC	ERRO%	Método	Corrigido/Reduzida	MOPC	ERRO%
1	-0,27	-0,31	12%	475,17	356,38	427,51	17%
2	-0,28	-0,31	10%	474,15	403,73	427,51	6%
3	-0,28	-0,31	9%	470,63	392,17	427,51	8%
4	-0,29	-0,31	7%	482,44	404,41	427,51	5%
5	-0,30	-0,31	5%	491,89	410,43	427,51	4%
6	-0,31	-0,31	2%	523,89	431,17	427,51	0%

As constantes obtidas pelo método matemático [Dos Santos *et al.*, 1999] são relacionadas à cada conjunto de parâmetros do experimento, e os valores das constantes geradas pelo método MOPC constituem um valor único geral para as condições estabelecidas no sistema de filtro. A Tabela 4.6 indica que os conjuntos de valores do experimento N° 1 apresentam o maior erro percentual.

Para o cálculo das constantes, restringindo-se no filtro de dados as condições de menores avanços entre 0,068 mm/rot e 0,142 mm/rot e maiores condições de avanço entre 0,148 mm/rot e 0,17 mm/rot, obtêm-se os resultados indicados na Tab. 4.7.

Tabela 4.7 - MOPC (f máximo e mínimo)

MÉTODO	Coeficiente "C"					
	MOPC	ERRO%	f = 0,068-0,142		f = 0,148-0,17	
			MOPC	ERRO%	MOPC	ERRO%
356,38	427,51	17%	415,57	14%	447,41	20%
403,73	427,51	6%	415,57	3%	447,41	10%
392,17	427,51	8%	415,57	6%	447,41	12%
404,41	427,51	5%	415,57	3%	447,41	10%
410,43	427,51	4%	415,57	1%	447,41	8%
431,17	427,51	0%	415,57	0%	447,41	4%

A Tabela 4.7 mostra uma diminuição do erro percentual para a condição em que foram filtrados os dados com um faixa de velocidade de avanço de valores mínimos. A Figura 4.5 ilustra a interface do MOPC com o cálculo das constantes obtidas, o filtro da velocidade de avanço com intervalo mínimo e valores calculados para o Intervalo de Máxima Eficiência com as condições de $t_c = 5$ min, $C_t = R\$ 10$ e $C_m = R\$ 200$.

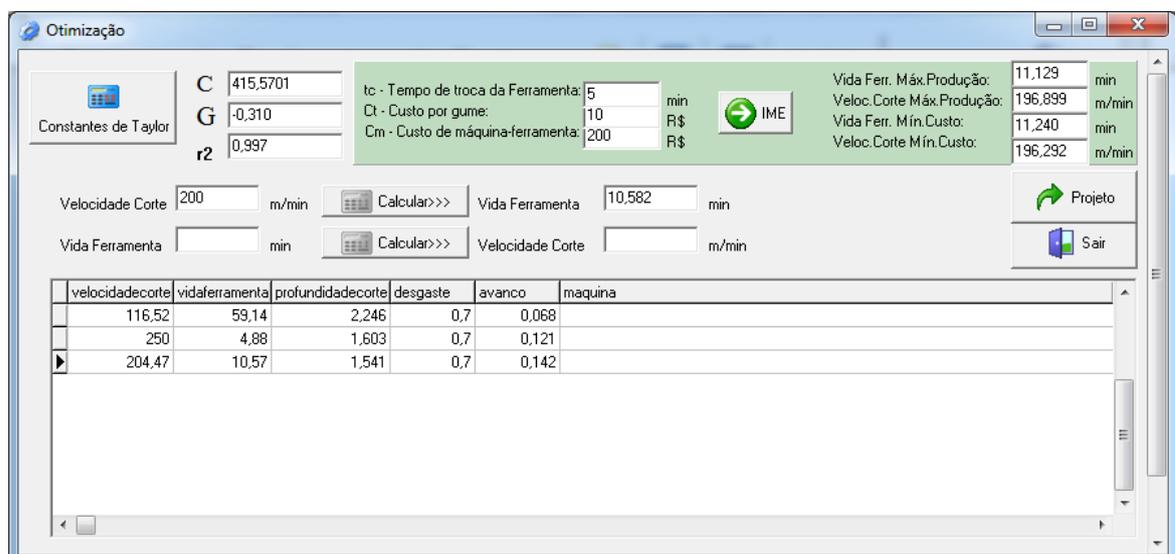


Figura 4.5 - Interface cálculo das constantes

Um comparativo é realizado entre os valores calculados para o Intervalo de Máxima Eficiência com as condições de filtro para menores avanços e os obtidos pelo método [Dos Santos *et al.*, 1999].

O aplicativo MOPC foi utilizado para realizar a análise, com valores de avanço compreendidos entre $f = 0,068$ mm e $0,142$ mm. Esse intervalo foi o que mais se aproximou dos resultados do método analisado. Utilizando o filtro do MOPC, com o intervalo da velocidade de avanço fixada nos valores mencionados anteriormente, determinaram-se os tempos e velocidades para máxima eficiência. Para o método analisado, os cálculos foram realizados individualmente para cada experimento. Os erros percentuais foram calculados relacionando os dados do MOPC e o método analisado. Esses dados são apresentados em gráficos analíticos para melhor visualização.

Para o tempo máximo de produção, a Fig.4.6 ilustra que o percentual de erro diminuiu entre os sistemas analisados. “O *software* empregado pelo método tende aprimorar os resultados” [Dos Santos *et al.*, 1999], sendo que com maior número de ensaios a tendência é aproximação com os resultados do MOPC.

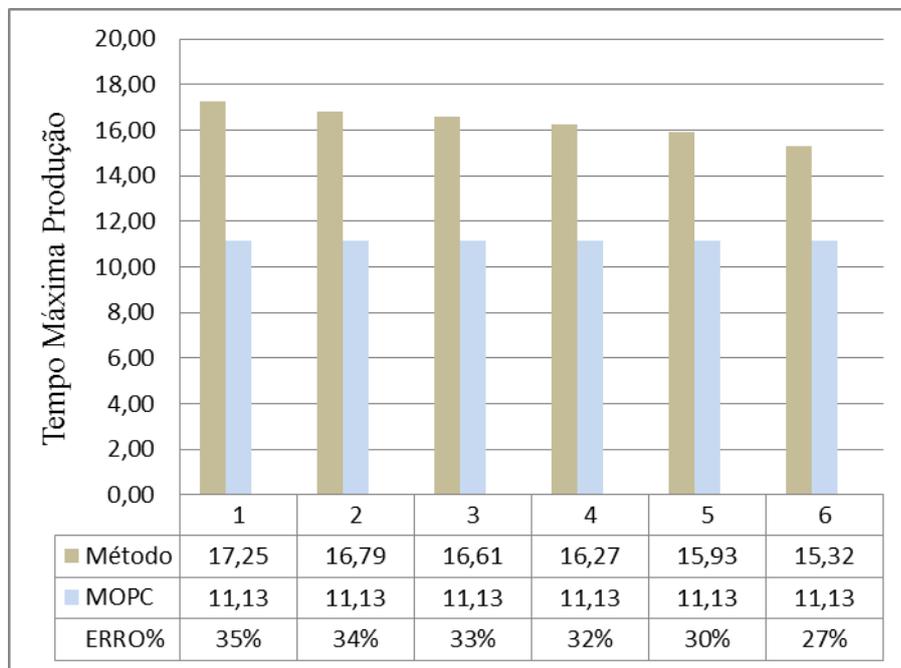


Figura 4.6 - Tempo Máxima Produção (Método/MOPC)

Para o tempo de vida da ferramenta para mínimo custo, a Fig. 4.7 ilustra novamente a tendência do método de aprimorar os resultados com o aumento dos experimentos, aproximando-se dos cálculos obtidos pelo MOPC.

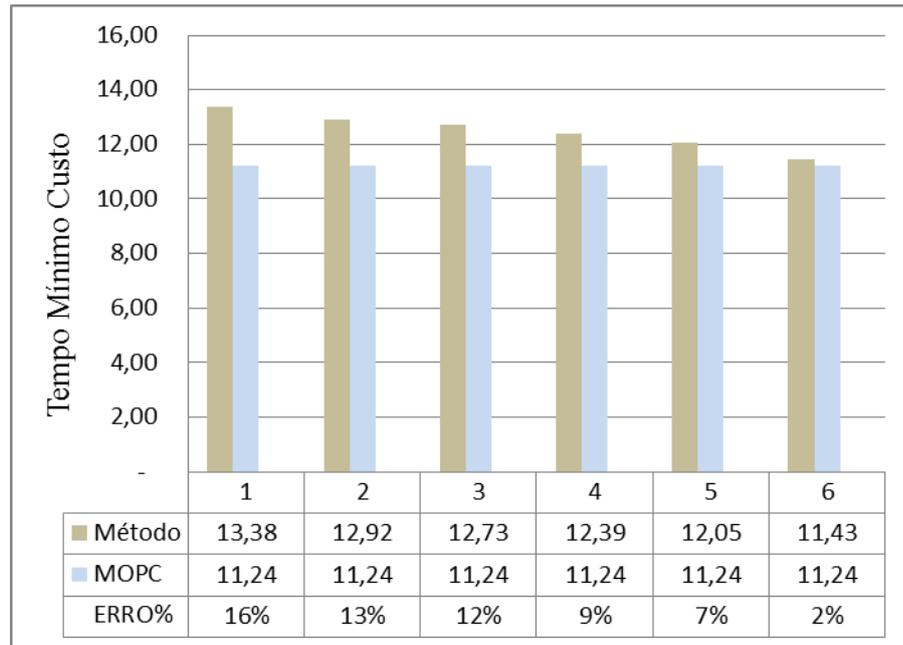


Figura 4.7 - Tempo Mínimo Custo (Método/MOPC)

Para a velocidade máxima produção, a Fig. 4.8 ilustra uma aproximação nos resultados dos métodos nos ensaios 2 e 6. A Fig.4.9 ilustra para os ensaios 2, 4 e 5 erros percentuais praticamente nulos para a velocidade mínimo custo.

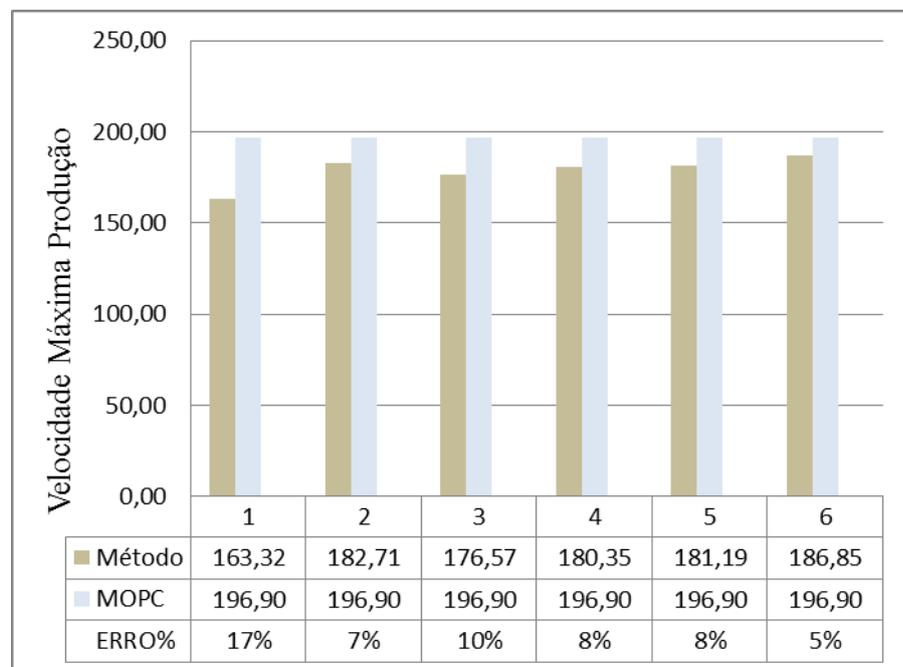


Figura 4.8 - Velocidade Máxima Produção (Método/MOPC)

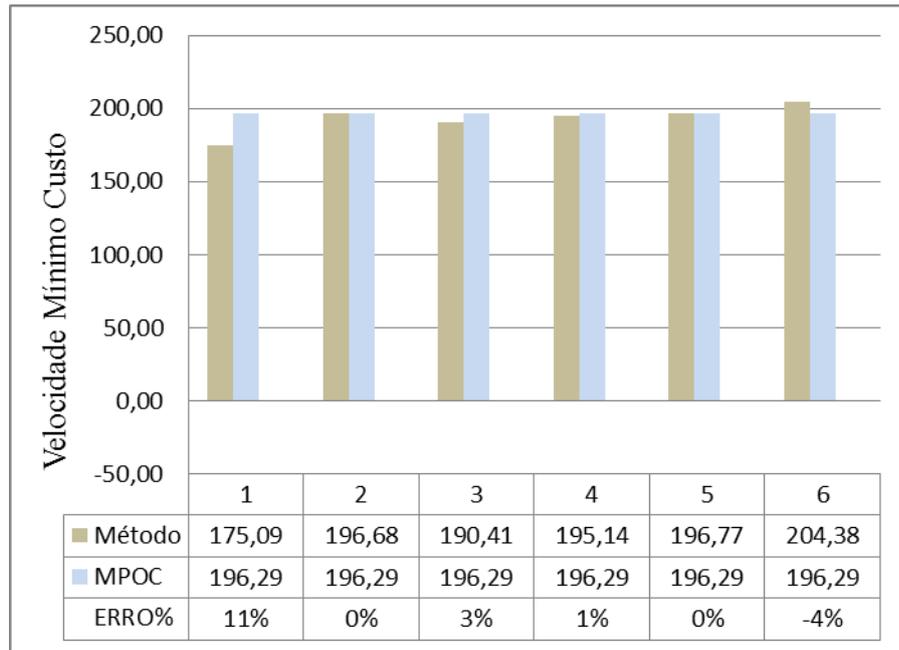


Figura 4.9 - Velocidade Mínimo Custo (Método/MOPC)

Com base nas análises realizadas, considera-se que o propósito do algoritmo MPOC mostra-se eficiente para manter o histórico de uma máquina operando em produção industrial, onde a otimização pela fórmula de Taylor reduzida pode servir de instrumento para aprimoramento de estratégias de usinagem para a máquina. O algoritmo tem estrutura para armazenamento de outros parâmetros de corte que compõem a equação expandida de Taylor, podendo, por outros métodos estatísticos, calcular os coeficientes.

Trabalhos científicos publicados com propostas de técnicas diferentes de cálculo das constantes de Taylor foram comparados ao sistema MOPC, sendo que os resultados obtidos foram de valores aproximados aos mostrados nos métodos analisados.

O sistema mostra uma variedade de opções para filtrar do banco de dados os valores do tempo de vida de ferramenta e velocidade de corte, com uma vasta combinação de parâmetros. Com essa característica é possível a combinação de condições de usinagem, tendo como resultado a disponibilidade de uma ferramenta para estudos de projeções do comportamento da vida da ferramenta.

Os resultados demonstram a viabilidade dos cálculos do aplicativo computacional MOPC, pois o método é coerente com as técnicas analisadas.

4.2 Aplicação do Módulo de programação parametrizada (MPP)

Com o objetivo de demonstrar a aplicação da metodologia proposta no módulo de programação parametrizada (MPP), são apresentados exemplos de criação de rotinas de usinagem com a linguagem parametrizada genérica híbrida para ciclos específicos de programação.

4.2.1 Ciclo para torneamento

A necessidade de diferentes estratégias de usinagem é frequente, mas nem sempre possível, pois são limitadas as possibilidades de ciclos oferecidas pelos comandos CNC. A Figura 4.10 mostra uma estratégia de usinagem pouco convencional - o desbaste em ângulo, não encontrado na maioria nos comandos. Os ciclos normalmente usados para esse procedimento são o desbaste longitudinal e desbaste transversal.

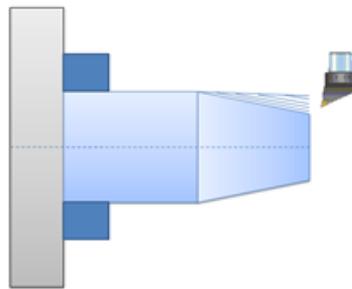


Figura 4.10 - Desbaste em ângulo

A Figura 4.11 mostra o resultado da ampliação do desbaste longitudinal, formando os “degraus”, retirados na última passada.

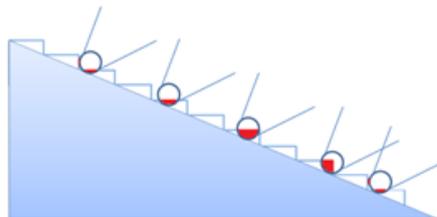


Figura 4.11 - Variação da área de contato da ferramenta

Esse tipo de operação pode trazer benefícios ao acabamento da peça usinada, reduzindo a variação da força de corte na retirada do sobremetal para acabamento. Além

disso, a redução da formação do escalonamento pode ser vantajosa para minimizar a trepidação da máquina. A Figura 4.12 representa, em forma de fluxograma, o algoritmo implementado para o desbaste em ângulo.

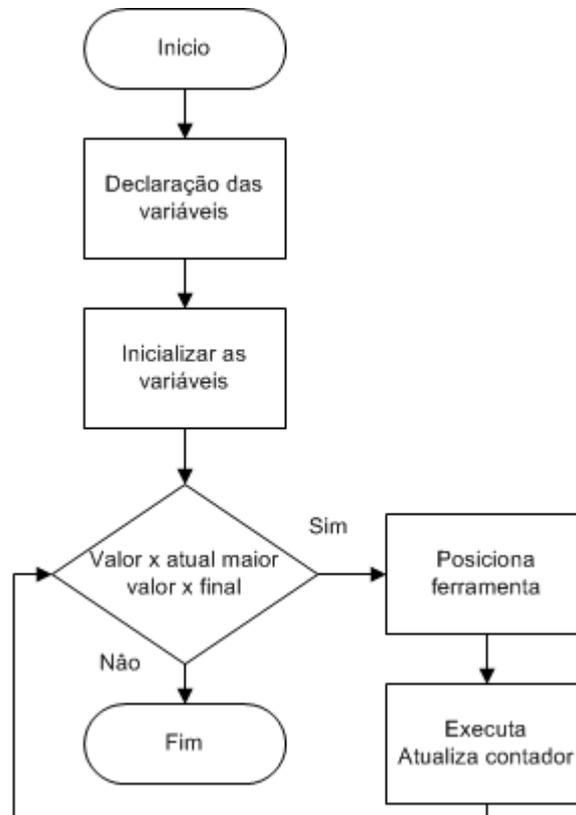


Figura 4.12 - Fluxograma desbaste em ângulo

Para a elaboração do programa CNC paramétrico, com o objetivo de realizar o desbaste em ângulo, são estabelecidas as variáveis necessárias ao sistema, Tab. 4.8.

Tabela 4.8 - Variáveis para desbaste em ângulo

Variáveis	Definições
xinicio	Diâmetro maior peça
xfinal	Diâmetro menor peça
zinicio	Comprimento inicial
zfinal	Comprimento final
xatual	Contador em x
ix	Incremento em x
iz	Incremento em z
zatual	Contador em z

O código do programa elaborado com a linguagem proposta, com comentários, é ilustrado na Fig. 4.13.

```

var
xinicio : real;           //Declaração das
xfinal  : real;           //variáveis
zinicio : real;
zfinal  : real;
xatual  : real;
ix      : real;
iz      : real;
zatural : real;

begin
xinicio := 24;           //Inicialização
xfinal  :=12;           //das variáveis
zinicio := 0;
zfinal  := 40;
ix      := 2;           //valores de inclinação
iz      := -6;
xatural := xinicio;
zatural:= iz;

while (xatural>xfinal) do //Repetir até a
                           //condição verdadeira
begin
G0 X(xatural-ix) Z(zinicio+1); //Posição inicial
G1 X(xinicio) Z(zatural);     //Primeira passada
G0 X(xatural) Z(zinicio +1);  //Posição inicial
xatural:= xatural - ix;       //Corrige contador x
zatural:= zatural + iz;       //Corrige contador z

end;

end;
end.

```

Figura 4.13 - Código comentado do ciclo de desbaste em ângulo

Na execução das instruções do programa, a ferramenta é posicionada em (xatural-ix) e (zinicio+1), ponto inicial para o deslocamento da ferramenta. O movimento em ângulo é realizado determinando o ponto de deslocamento (xinicio e zatural). Os corretores são atualizados e a ferramenta volta ao ponto inicial e realiza o próximo movimento.

4.2.2 Programa parametrizado com múltiplos ciclos

Uma das características da programação parametrizada é a possibilidade de uso da lógica algorítmica, como, por exemplo, as condições relacionais e condicionais. É possível um programa parametrizado conter vários procedimentos ou ciclos na forma de subrotinas, e executá-los conforme forem acionados.

A Figura. 4.14 ilustra as operações de desbastes longitudinal, transversal e em ângulo de uma peça padrão representativa de uma determinada família de peças produzida em uma célula de fabricação. Um programa parametrizado pode ser desenvolvido para executar todas as operações ou somente as que forem solicitadas. A Figura 4.15 ilustra o fluxograma para o programa.

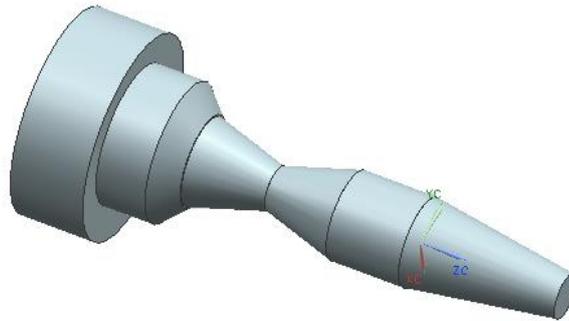


Figura 4.14 - Peça padrão representativa de uma família

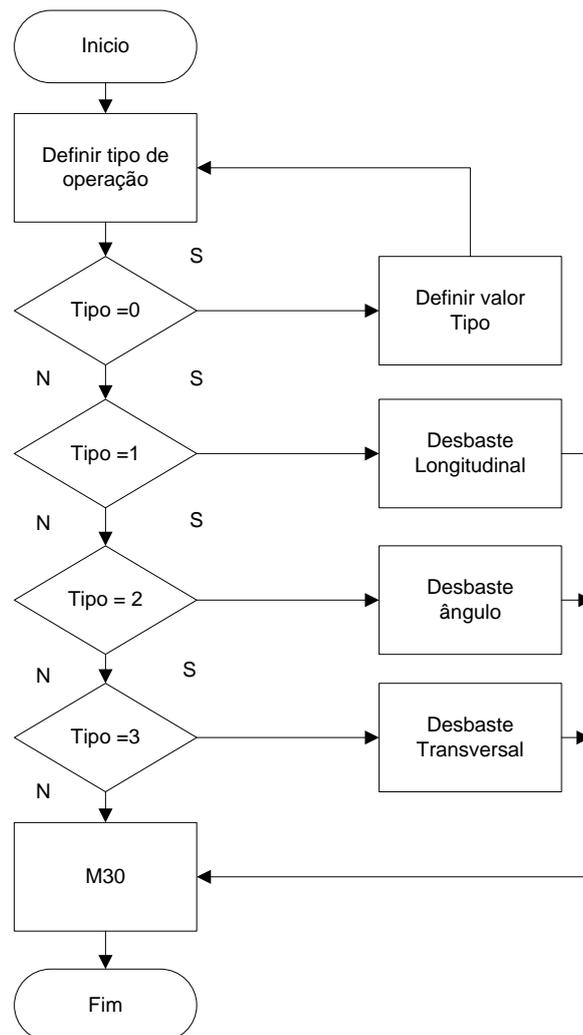


Figura 4.15 - Fluxo do programa padrão

Com o objetivo de demonstrar a metodologia para a programação da peça padrão, foi elaborado um programa parametrizado. Considerando que essas rotinas desenvolvidas são de relativa simplicidade, em que o foco está no posicionamento da ferramenta, não foi acrescentada, nos módulos, a compensação da ferramenta e ciclos de acabamento.

A Figura 4.16 ilustra o programa com os blocos condicionais para cada tipo de desbaste selecionado. O código é compilado no MPP, gerando o programa CNC correspondente à operação de desbaste selecionada, conforme a Figura 4.17.

```

if (tipo=1) then
begin
DL
while (xatual>xfinal) do
begin
G0 X(xatual-prof) Z(zinicio+1);
G1 Z(zfinal);
G0 X(xatual);
G0 Z(zinicio+1);
xatual:= xatual - prof;
end;
end;

if (tipo=2) then
begin
DA
while (xatual>xfinal) do
begin
G0 X(xatual-ix) Z(zinicio+1);
G1 X(xinicio) Z(zatual);
G0 X(xatual) Z(zinicio +1);
xatual:= xatual - ix;
zatual:= zatual + iz;
end;
end;

if (tipo=3) then
begin
DT
while (zatualt>zfinal) do
begin
G0 X(xatual+1) Z(zatualt);
G1 X(xfinal);
G0 X(xatual+1) Z(zatualt);
zatualt := zatualt - prof;
end;
end;
M30
end;

```

Desbaste Longitudinal

Desbaste Ângulo

Desbaste Transversal

Figura 4.16 - Código do programa padrão

Os ciclos elaborados pela linguagem parametrizada do MPP podem ser armazenados na forma de arquivo texto para serem utilizados pelo editor de programação no momento da elaboração do programa. São considerados como ciclos adicionais para o comando da máquina, complementando os recursos disponíveis na elaboração da programação.

Desbaste longitudinal	Desbaste em ângulo	Desbaste Transversal
G0 X 23 Z 1	G0 X 22 Z 1	G0 X 25 Z 0
G1 Z -40	G1 X 24 Z -6	G1 X 12
G0 X 24	G0 X 24 Z 1	G0 X 25 Z 0
G0 Z 1	G0 X 20 Z 1	G0 X 25 Z -1
G0 X 22 Z 1	G1 X 24 Z -12	G1 X 12
G1 Z -40	G0 X 22 Z 1	G0 X 25 Z -1
G0 X 23	G0 X 18 Z 1	G0 X 25 Z -2
G0 Z 1	G1 X 24 Z -18	G1 X 12
G0 X 21 Z 1	G0 X 20 Z 1	G0 X 25 Z -2
G1 Z -40	G0 X 16 Z 1	G0 X 25 Z -3
G0 X 22	G1 X 24 Z -24	G1 X 12
G0 Z 1	G0 X 18 Z 1	G0 X 25 Z -3
G0 X 20 Z 1	G0 X 14 Z 1	G0 X 25 Z -4
G1 Z -40	G1 X 24 Z -30	G1 X 12
G0 X 21	G0 X 16 Z 1	G0 X 25 Z -4
G0 Z 1	G0 X 12 Z 1	G0 X 25 Z -5
G0 X 19 Z 1	G1 X 24 Z -36	G1 X 12
G1 Z -40	G0 X 14 Z 1	G0 X 25 Z -5
G0 X 20	M30	G0 X 25 Z -6

Figura 4.17 – Resultado da compilação no MPP do programa Padrão

Com essa visão, o sistema pode ser utilizado também dentro da metodologia da tecnologia de grupo considerando-se uma programação relativa a uma peça “padrão” em que comparada com características geométricas presentes ou não na mesma, o programa é simplificado representando uma peça específica [Lorini, 1993].

Os ciclos parametrizados que executam operações pertencentes a uma família de peças podem ser armazenados em grupos de arquivos. Quando houver a necessidade de executar a programação de uma peça pertencente a esse grupo, pode-se recorrer a esses ciclos armazenados mudando apenas os parâmetros. Um esquema para ilustração da aplicação dos ciclos desenvolvidos no MPP é mostrado na Figura 4.18.

4.3 Aplicação do Protótipo CAD/CAM

Para demonstração integrada dos diversos módulos do sistema CAD/CAM, adota-se uma peça modelo a ser produzida em um torno *Nardini logic* 195, com comando numérico MCS. O código de programação CNC, gerado pelo protótipo, baseado na linguagem interativa do comando MCS refere-se a uma peça que contém operações básicas de usinagem cilíndrica externa. Esses procedimentos são desbaste longitudinal, acabamentos, abertura de canal e abertura de rosca. A metodologia proposta possibilita a adaptação do protótipo a qualquer comando numérico, necessitando para isso, realizar a configuração do sistema.

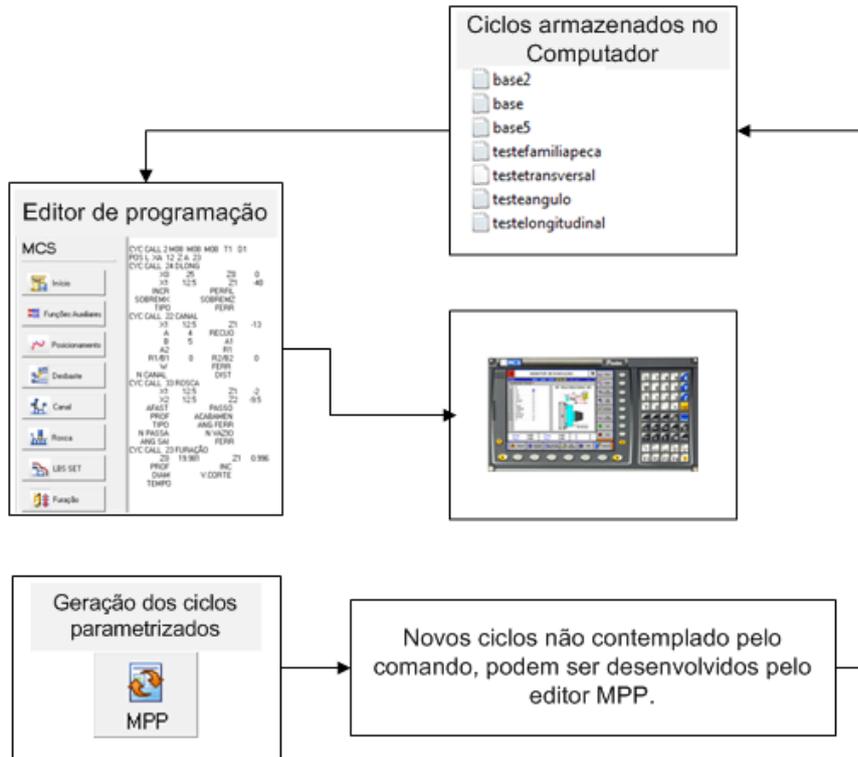


Figura 4.18 - Aplicação dos ciclos desenvolvidos pelo MPP

Os equipamentos testados pertencem aos laboratórios do Colégio Técnico Industrial de Santa Maria (CTISM) e do Centro de Tecnologia (CT), ambos da Universidade Federal de Santa Maria (UFSM). O protótipo CAD/CAM aplicado nos tornos agrega mais uma ferramenta disponível para a estrutura didática. O modelo da peça proposta para o ensaio é ilustrado na Fig. 4.19.

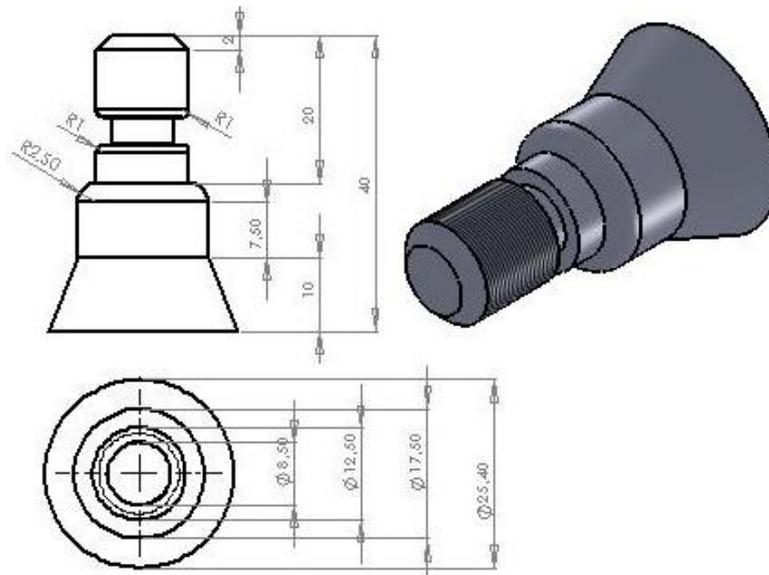


Figura 4.19 - Modelo para gerar programa no Protótipo CAD/CAM

A seguir é descrito os procedimentos utilizados com o protótipo CAD/CAM para a geração do programa CNC para a peça modelo.

4.3.1 Atualizando o MOPC para o projeto

Para que o sistema forneça dados de corte otimizados para a operação a ser executada, parâmetros de corte de usinagem do sistema produtivo dedicado, relativos ao material e ferramenta utilizados, são inseridos no banco de dados, como mostrada na Fig. 4.20.

Os dados inseridos no banco de dados para alimentar o MOPC, no caso, são relativos ao aço ABNT 1045 com diâmetro de 25 mm e comprimento de 60 mm com uma ferramenta de desbaste longitudinal PCLNL 2020K12, classe P10.

codigo	codequipam	codferrame	codmaterial	codclasse	velocidadec	vidaferram	profundidac	desgaste	avanco
59	4	5	11	4	0,96	0,95			
60	4	5	11	4	1,02	1,02			
61	4	5	11	4	1,07	1,14			
62	4	5	11	4	1,11	1,25			
64	17	4	9	3	525	14	0,8	0,32	0,15
65	1	1	1	2	290	15	1	0,2	0,16
66	19	7	12	5	214,1	19,9888	4	0,3	0,25
67	19	7	12	5	236,5	12,12352	2	0,3	0,25
74	19	7	12	5	223,1	16,2526	3	0,3	0,25
75	19	7	12	5	207,3	23,50855	5	0,3	0,25
76	19	7	12	5	201,9	26,84286	6	0,3	0,25
77	19	7	12	5	197,5	29,98577	7	0,3	0,25
78	19	7	12	5	193,7	33,0609	8	0,3	0,25
79				6	116,52	59,14	2,246	0,7	0,068
80				6	179,91	17,73	1,379	0,7	0,17
81				6	250	4,88	1,603	0,7	0,121
82				6	171,89	19,2	1,482	0,7	0,155
83				6	204,47	10,57	1,541	0,7	0,142
84				6	207,65	10,85	1,516	0,7	0,148
85	20	8	10	2	214,1	19,9	4	0,3	0,25
86	20	8	10	2	236,5	12,12	2	0,3	0,25
87	20	8	10	2	223,1	16,25	3	0,3	0,25
88	20	8	10	2	207,3	23,5	5	0,3	0,25
89	20	8	10	2	201,9	26,84	6	0,3	0,25
90	20	8	10	2	197,5	29,98	7	0,3	0,25

Figura 4.20 - Inserindo dados do sistema no banco de dados do MOPC

Conforme mostrado na interface gráfica ilustrada na Fig. 4.21, é realizado o filtro das informações correspondentes aos itens de interesse do projeto. As informações estão indicadas na imagem, listadas na grade, para um avanço estabelecido no filtro de 0,25 mm/rot.

A Figura 4.22 ilustra a interface de otimização, com os resultados dos cálculos das constantes da equação de Taylor e velocidades ótimas para $t_c = 5$ min, $C_t = 100$ R\$ e $C_m = 200$ R\$, bem como o cálculo do tempo de vida da ferramenta para a velocidade de 230 m/min.



Figura 4.21 - Filtro dos dados para o projeto

Esses dados serão recomendados automaticamente nas interfaces de programação dos ciclos do comando e selecionados quando necessários. Existindo a possibilidade, através de um link, no momento da programação, pode-se voltar ao MOPC para modificar parâmetros estabelecidos no filtro de dados, aprimorando os resultados obtidos.

Constantes de Taylor calculadas

Velocidades e tempos ótimos

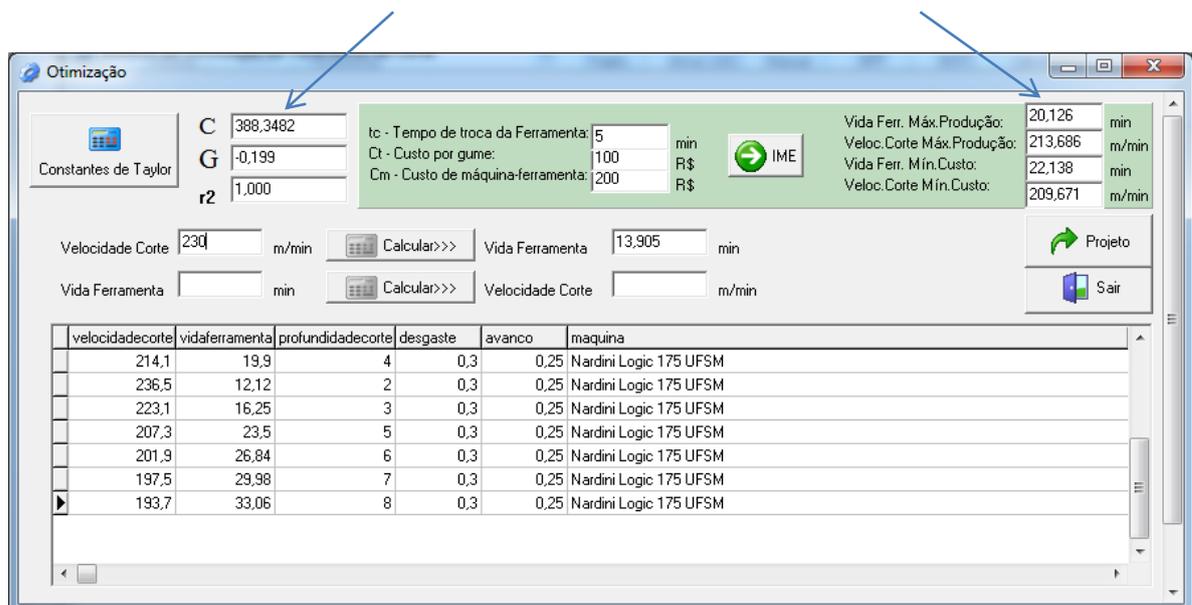


Figura 4.22 - Interface de otimização do MOPC

4.3.2 Geração do programa para a peça especificada

O perfil da peça usinada é desenhado na plataforma CAD do módulo gráfico, de onde são extraídos os dados geométricos automaticamente, com a origem de coordenadas fixadas na face da peça. Importando um desenho, o sistema disponibiliza uma rotina desenvolvida para transferir o zero peça para a face da peça. A Figura 4.23 ilustra o posicionamento do zero peça na face da peça, através da rotina “zero”, implementada no sistema.

Todas as rotinas desenvolvidas em *AutoLISP*, descritas neste trabalho, são carregadas automaticamente quando ativado o módulo gráfico e executadas quando selecionadas. A Figura 4.24 ilustra a interface principal com o módulo gráfico.

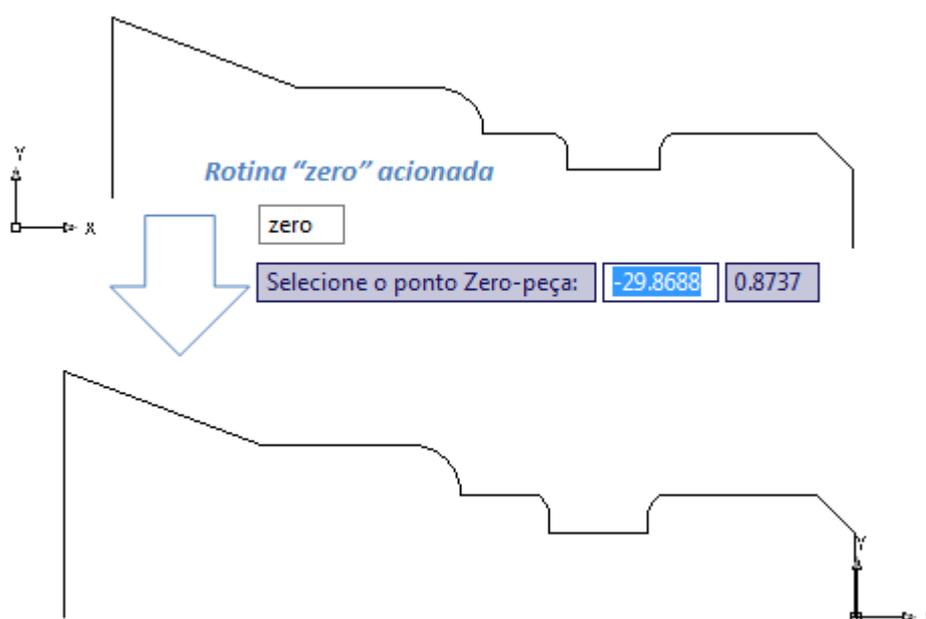


Figura 4.23 - Posicionamento do zero peça

A definição das instruções de programa inicia com o cabeçalho, onde as condições básicas de operação da máquina são definidas. A função miscelânea “M39” define trabalho em gama alta, “M28” testa contra ponto recuado e “M68” verifica se a placa está aberta. Nesse bloco, declara-se a identificação e o corretor da ferramenta usada (“T” = 1, e o corretor “D” = 1). Com a instrução “M59” é determinada a máxima rotação para VCC (velocidade de corte constante) e “M58” sua ativação. A Figura 4.25 ilustra a elaboração do cabeçalho do programa. Mostra a interface de cabeçalho padrão configurada para escrever a ultima linha,

com o uso da velocidade recomendada pelo MOPC para “M58”. Na área de edição do programa, o código é montado.

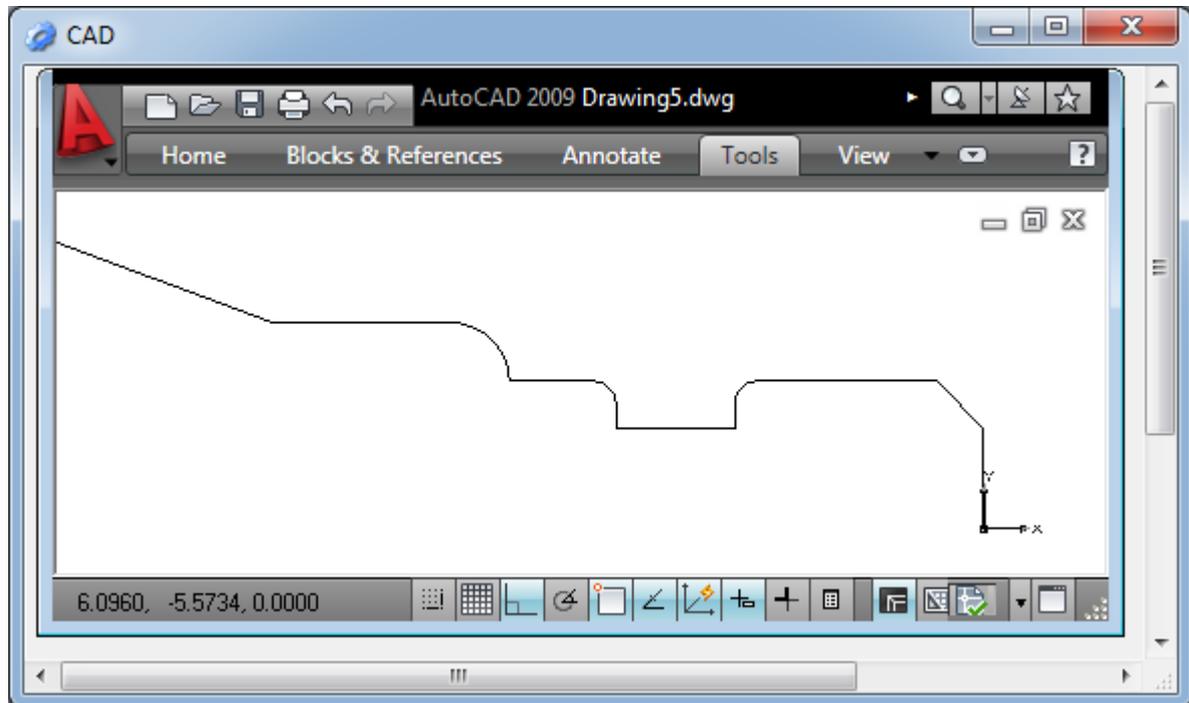


Figura 4.24 - Desenho do perfil da peça

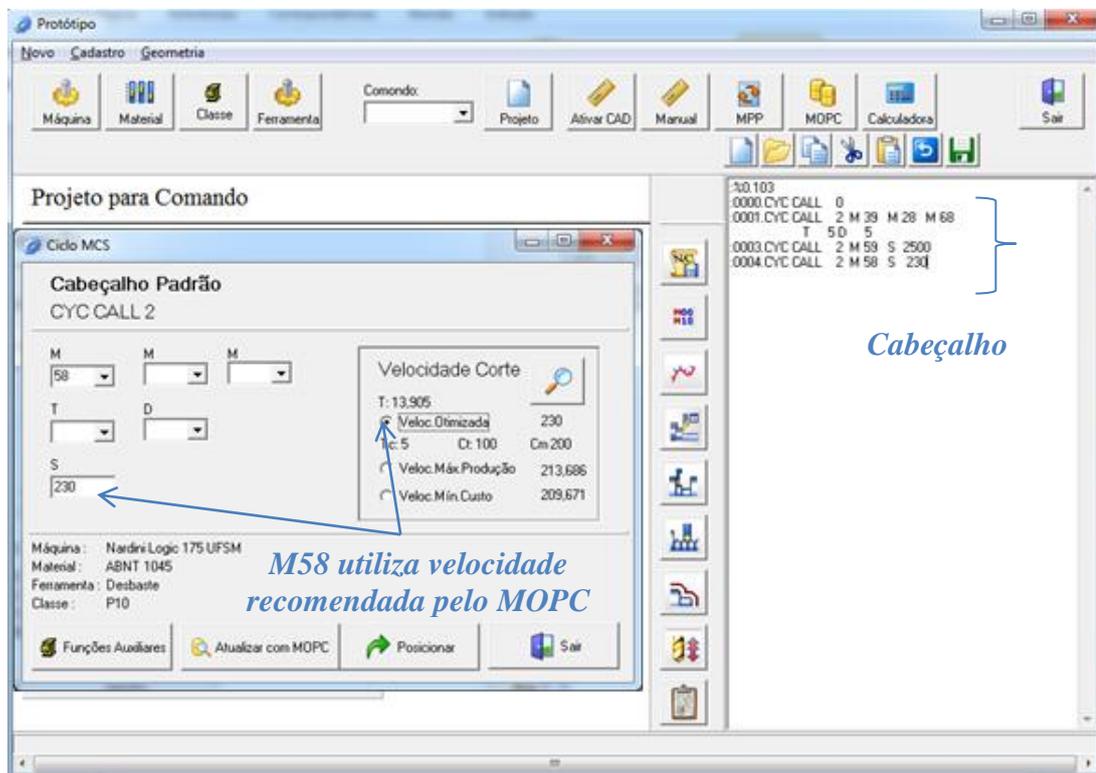


Figura 4.25 - Cabeçalho do programa

O próximo bloco do programa determina o posicionamento inicial da ferramenta. O ponto de troca de ferramenta (nesse projeto é definido com “X” = 30 mm e “Z” = 50 mm). A velocidade de avanço rápido “F” = 0, que representa a maior velocidade de avanço da máquina. Adota-se a função miscelânea “M13” que liga a rotação da placa com sentido anti-horário com fluido refrigerante. Na próxima linha é indicado o posicionamento de aproximação da ferramenta com a velocidade de avanço para “F” = 0.250 mm/rot. Com esses dados de posicionamento de ferramenta e velocidade de avanço, o programa entra no ciclo de desbaste. A Figura 4.26 ilustra a interface de posicionamento e os códigos escritos no editor.

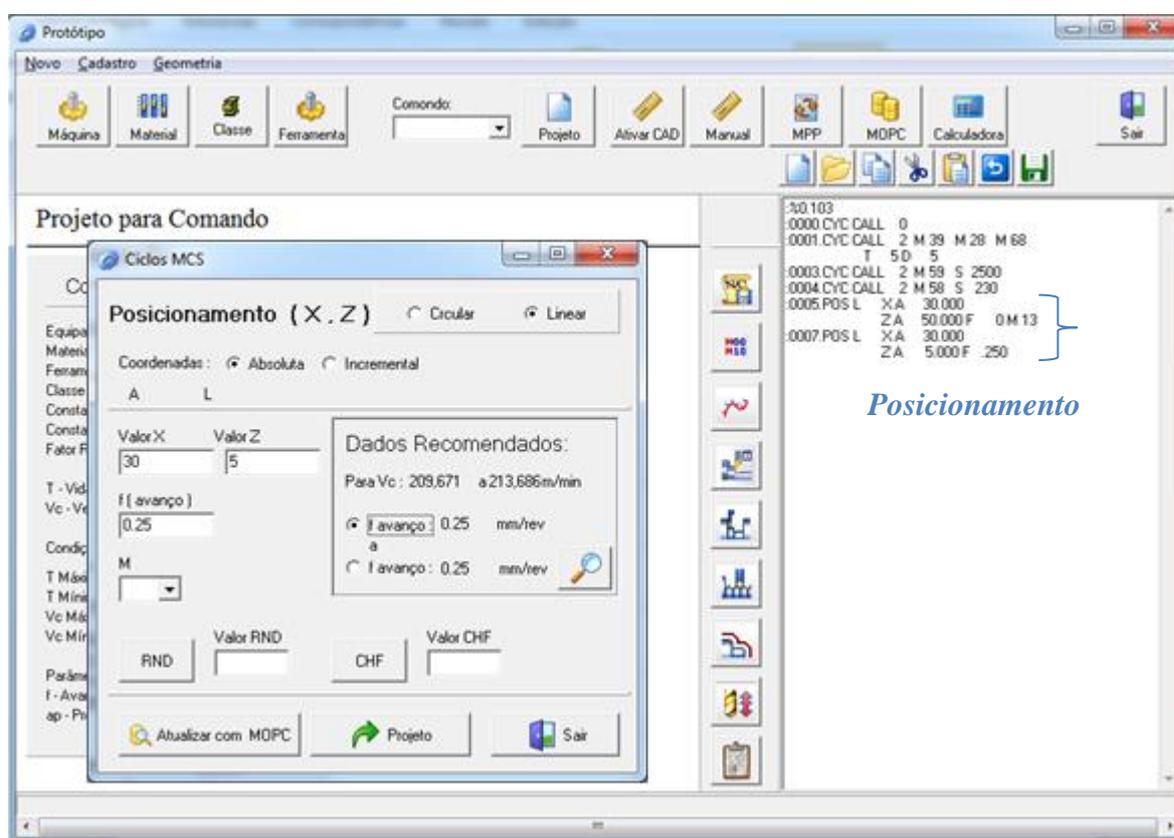


Figura 4.26 - Posicionamento da ferramenta

Na interface de desbaste longitudinal, os parâmetros de dimensionamento necessários para sua configuração são extraídos do módulo gráfico, através da execução da rotina “desbaste”, como é ilustrado na Figura 4.27.

Os demais parâmetros necessários para a configuração do ciclo do desbaste longitudinal são preenchidos manualmente no sistema. A Figura 4.28 ilustra esse procedimento e a inserção do código gerado pelo programa no texto do editor CNC.

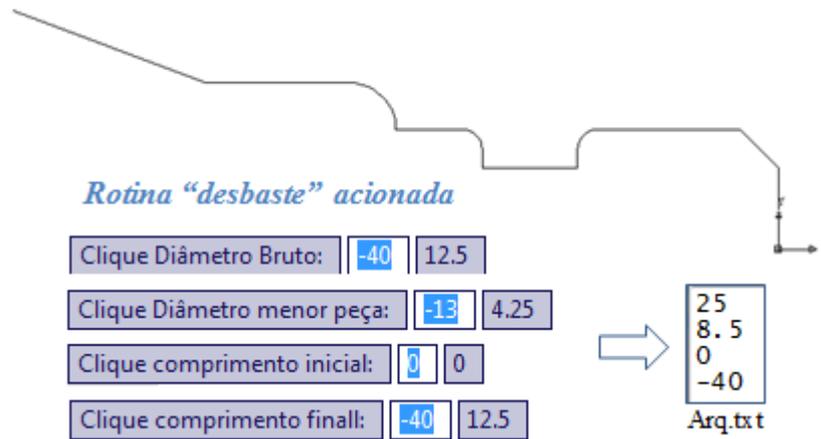


Figura 4.27 - Extração de parâmetros para ciclo de desbaste longitudinal

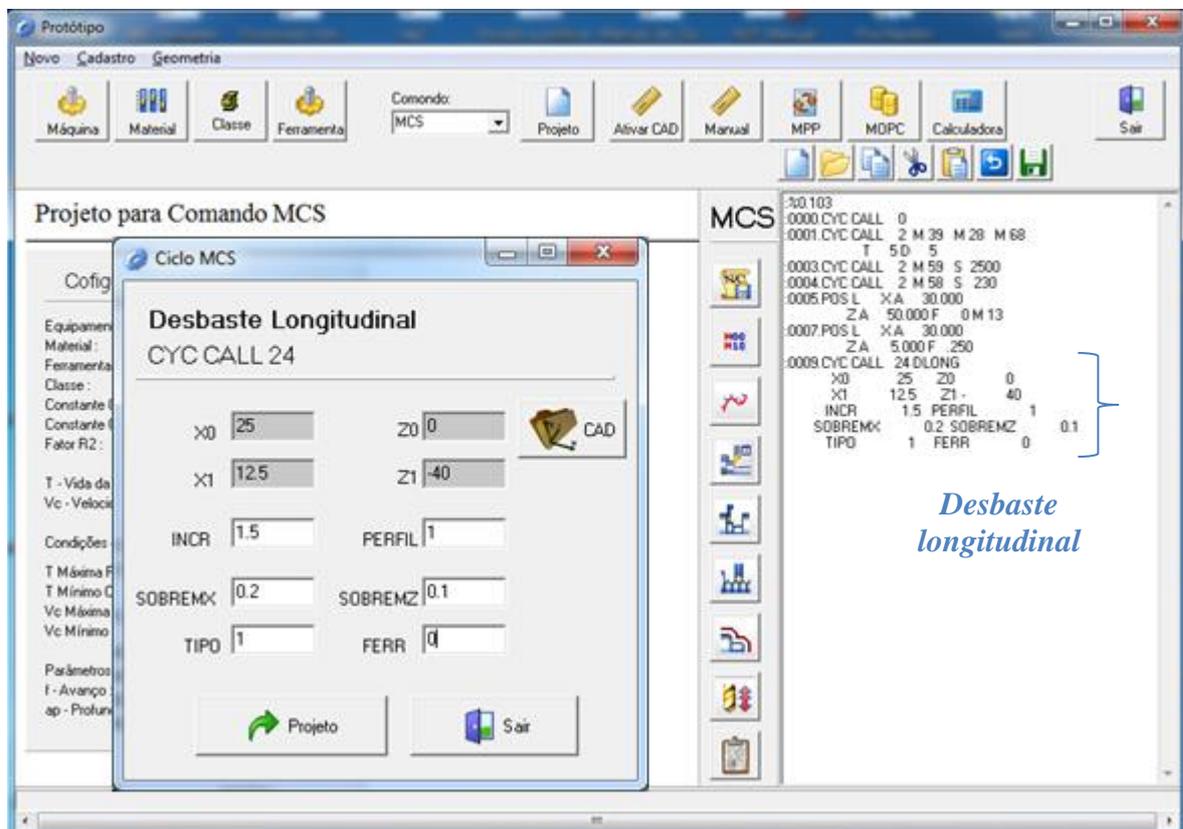


Figura 4.28 - Programação do ciclo desbaste longitudinal

O próximo procedimento para a fabricação da peça é a programação da abertura de canal. Na preparação do procedimento, a ferramenta tem que ser trocada no exemplo para “T” = 6, que representa o local de fixação, resultando na reconfiguração das velocidades e avanços da ferramenta.

Na interface do ciclo de canal, os dados geométricos referentes aos parâmetros, como “X1” - diâmetro para início da usinagem, “Z1” - ponto central na base do canal, “A” - profundidade do canal e “B” - largura do fundo do canal, são extraídos automaticamente, executando rotina específica desenvolvida no módulo gráfico. A Figura 4.29 ilustra a extração dos parâmetros no módulo gráfico.

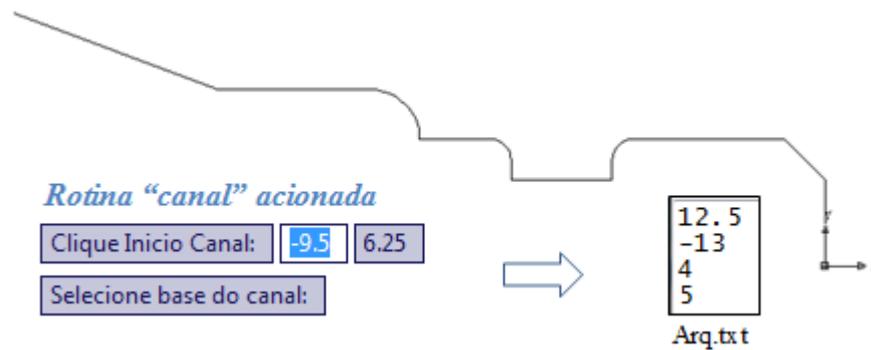


Figura 4.29 - Extração de parâmetros para ciclo de canal

Os demais parâmetros necessários para a configuração do ciclo do canal são preenchidos manualmente. A Figura 4.30 ilustra esse procedimento e a inserção do código gerado pelo programa no texto do editor CNC.

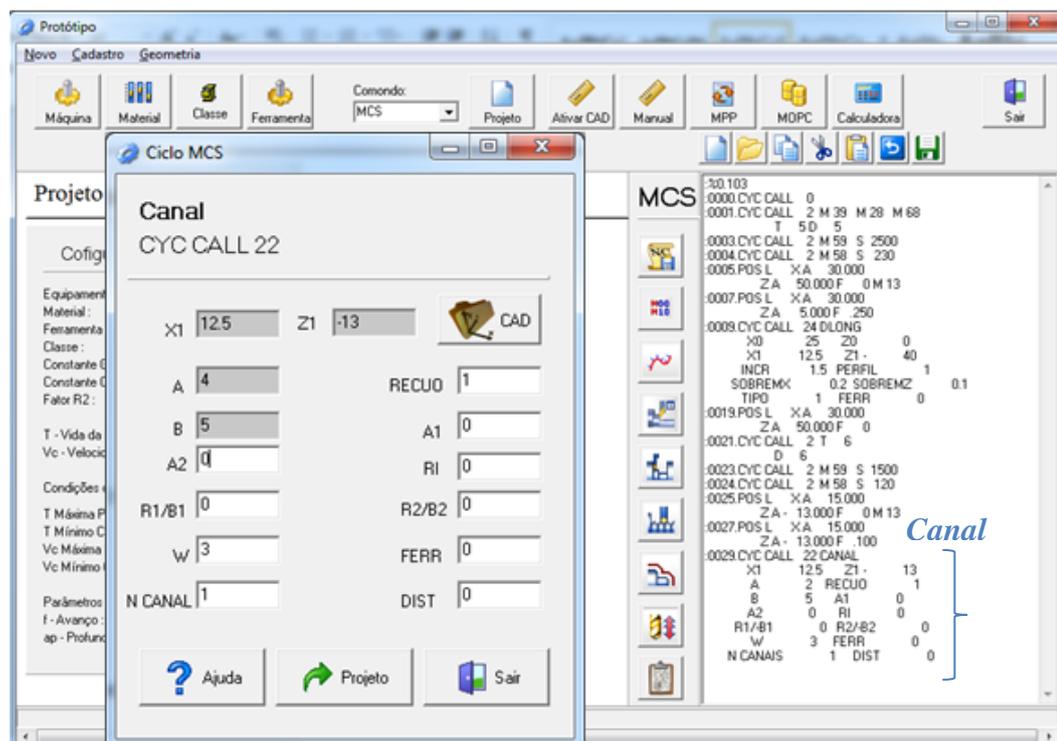


Figura 4.30 - Programação do ciclo do canal

Na sequência define-se programação da abertura de rosca. Na preparação do procedimento, a ferramenta é redefinida para (“T” = 7), resultando em nova reconfiguração das velocidades e avanços da ferramenta.

Na interface do ciclo de rosca, os dados geométricos referentes aos parâmetros, como “X1” e “Z1”, ponto inicial da rosca, e “X2” e “Z2”, ponto final da rosca, são extraídos, automaticamente, executando rotina específica desenvolvida no módulo gráfico. A Figura 4.31 ilustra a extração dos parâmetros no módulo gráfico.

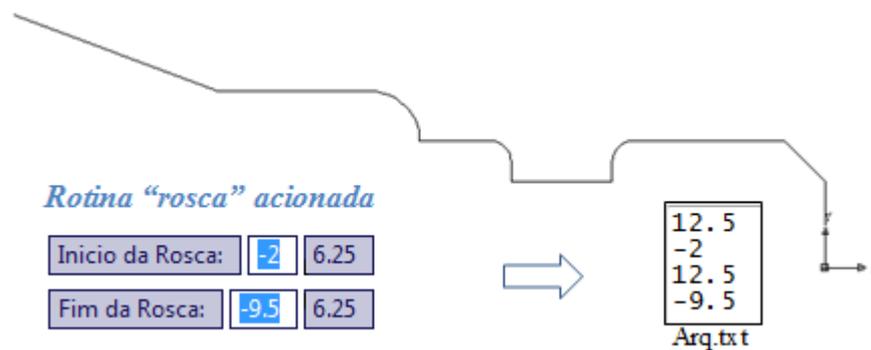


Figura 4.31 - Extração de parâmetros para o ciclo de rosca

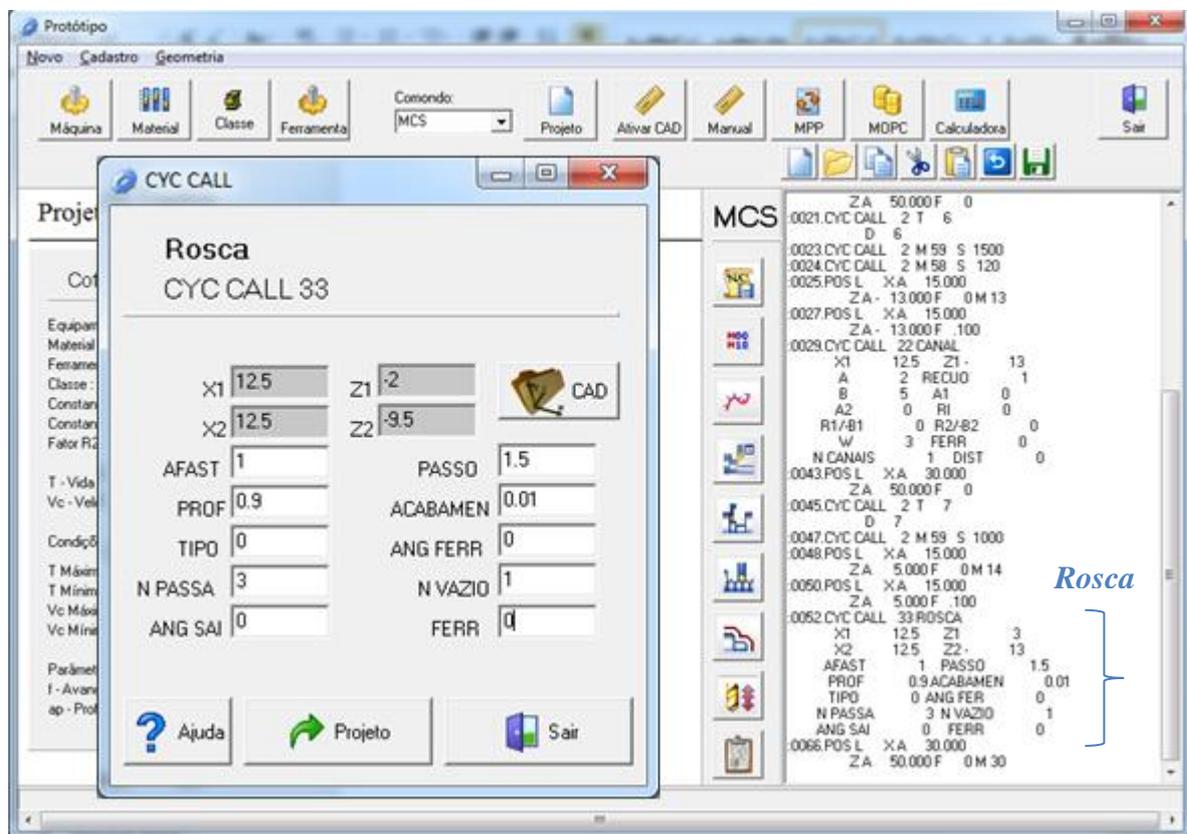


Figura 4.32 - Programação do ciclo da rosca

Os demais parâmetros necessários para completar a configuração do ciclo de rosca são preenchidos manualmente. A Figura 4.32 ilustra esse procedimento e a inserção do código gerado pelo programa no texto do editor CNC.

No final do programa, acrescenta-se o perfil da peça a ser usinada. Esse perfil é extraído basicamente pela rotina “perfil” desenvolvida. A declaração de início deve ser “LBL SET” e a de final com “LBL SET 0”. A Figura 4.33 ilustra a extração dos parâmetros no módulo gráfico.

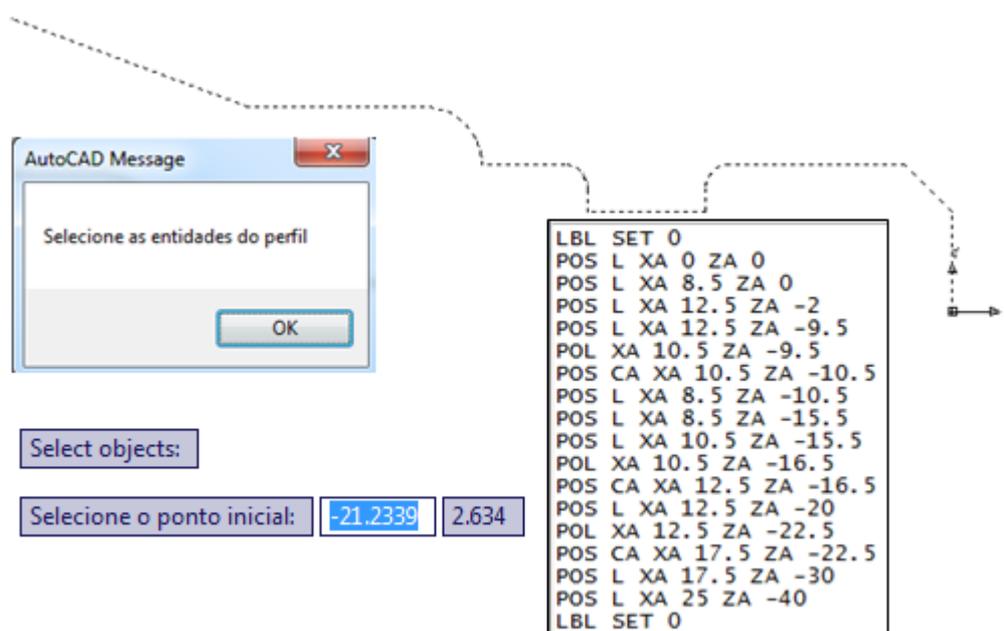


Figura 4.33 - Extração do perfil da peça

A interface relacionada ao perfil lista o código gerado antes de inseri-lo no código do editor CNC, com a finalidade de permitir o ajuste dos códigos extraídos do módulo gráfico, antes de inserir definitivamente no editor CNC.

O módulo desenvolvido para extrair o perfil é automático, bastando indicar as linhas do desenho. O algoritmo analisa, nas entidades do desenho, o sentido e inclinação das linhas, com seus pontos iniciais e finais, armazenando as informações para montar o perfil da peça a ser usinada. A geração dessa informação é necessária para o funcionamento correto dos ciclos de desbaste e acabamento. A Figura 4.34 ilustra esta operação.

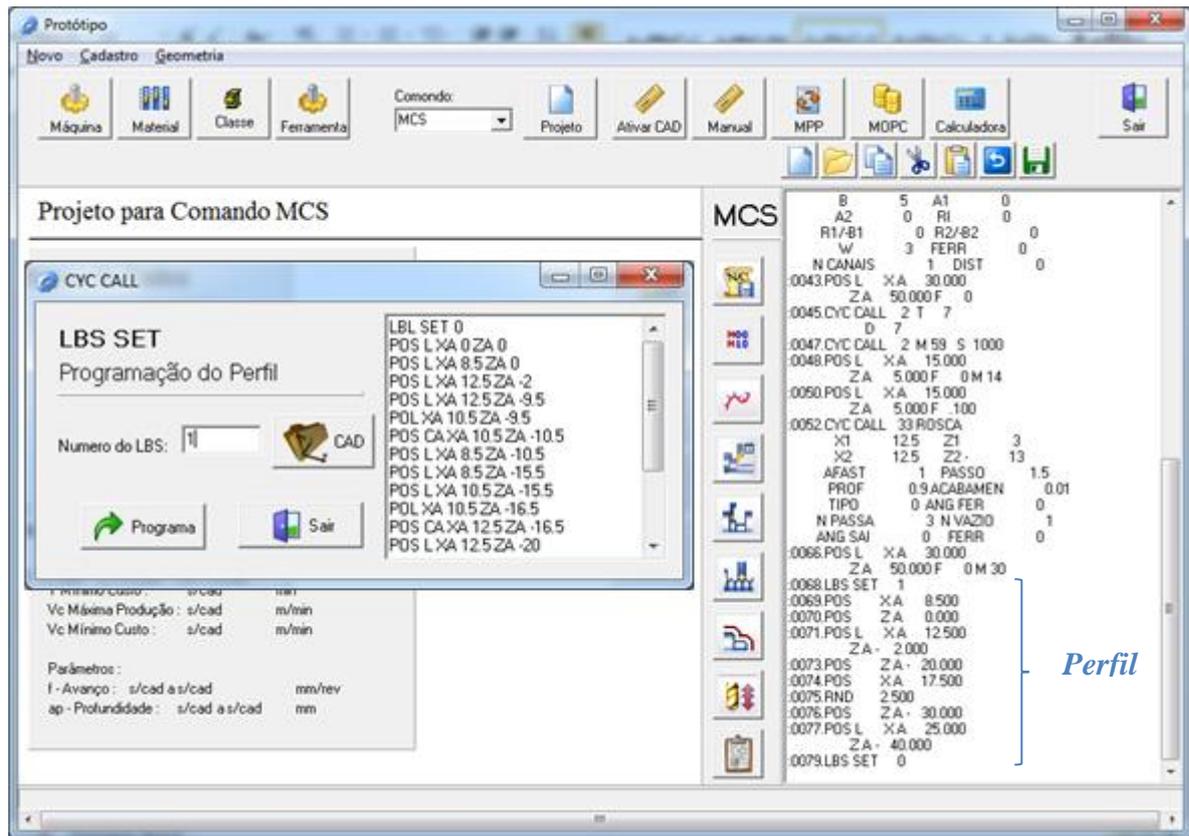


Figura 4.34 - Ciclo captura o perfil automaticamente

4.3.3 Fabricação da peça modelo

A manufatura da peça modelo foi realizada após o programa gerado pelo protótipo CAD/CAM ter sido simulado graficamente no comando MCS. A Figura 4.35 ilustra a imagem do programa carregado no comando da máquina e a imagem da simulação gráfica.

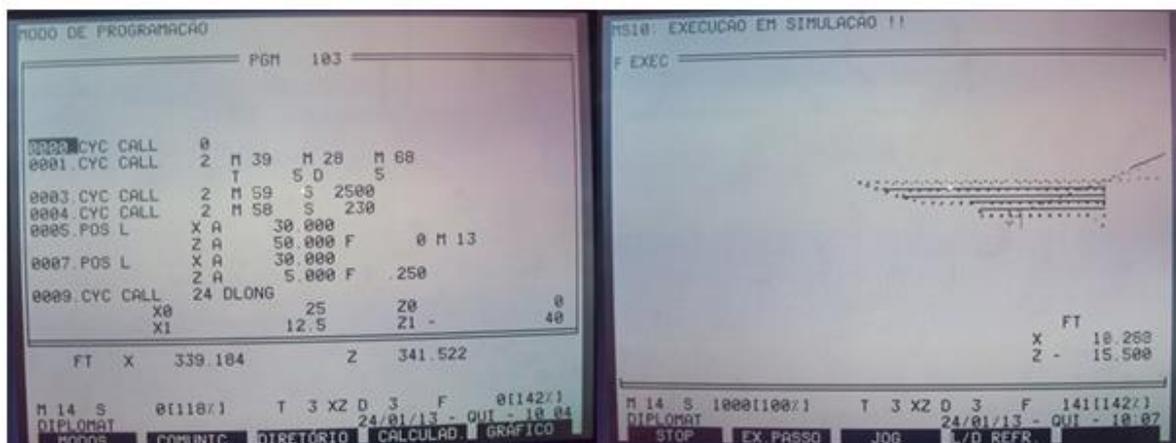


Figura 4.35 - Programa executado no comando MCS

A Figura 4.36 ilustra a listagem final do programa gerado pelo protótipo CAD/CAM. Esse código é enviado para o comando CNC através de um software específico.

Protótipo CAD/CAM	SERIE 500	:0043.POS L X A 30.000	
		Z A 50.000 F 0	
:%0.103		:0045.CYC CALL 2 T 7	
:0000.CYC CALL 0		D 7	
:0001.CYC CALL 2 M 39 M 28 M 68		:0047.CYC CALL 2 M 59 S 1000	
T 5 D 5		:0048.POS L X A 15.000	
:0003.CYC CALL 2 M 59 S 2500		Z A 5.000 F 0 M 14	
:0004.CYC CALL 2 M 58 S 230		:0050.POS L X A 15.000	
:0005.POS L X A 30.000		Z A 5.000 F .100	
Z A 50.000 F 0 M 13		:0052.CYC CALL 33 ROSCA	
:0007.POS L X A 30.000		X1 12.5 Z1 3	
Z A 5.000 F .250		X2 12.5 Z2 - 13	
:0009.CYC CALL 24 DLONG		AFAST 1 PASSO 1.5	
X0 25 Z0 0		PROF 0.9 ACABAMEN 0.01	
X1 12.5 Z1 - 40		TIPO 0 ANG FER 0	
INCR 1.5 PERFIL 1		N PASSA 3 N VAZIO 1	
SOBREM X 0.2 SOBREM Z 0.1		ANG SAI 0 FERR 0	
TIPO 1 FERR 0		:0066.POS L X A 30.000	
:0019.POS L X A 30.000		Z A 50.000 F 0 M 30	
Z A 50.000 F 0		:0068.LBS SET 1	
:0021.CYC CALL 2 T 6		:0069.POS X A 8.500	
D 6		:0070.POS Z A 0.000	
:0023.CYC CALL 2 M 59 S 1500		:0071.POS L X A 12.500	
:0024.CYC CALL 2 M 58 S 120		Z A - 2.000	
:0025.POS L X A 15.000		:0073.POS Z A - 20.000	
Z A - 13.000 F 0 M 13		:0074.POS X A 17.500	
:0027.POS L X A 15.000		:0075.RND 2.500	
Z A - 13.000 F .100		:0076.POS Z A - 30.000	
:0029.CYC CALL 22 CANAL		:0077.POS L X A 25.000	
X1 12.5 Z1 - 13		Z A - 40.000	
A 2 RECUO 1		:0079.LBS SET 0	
B 5 A1 0			
A2 0 RI 0			
R1/-B1 0 R2/-B2 0			
W 3 FERR 0			
N CANAIS 1 DIST 0			

Figura 4.36 - Listagem do programa elaborado no Protótipo CAD/CAM

O primeiro ciclo executado pelo programa é o de desbaste longitudinal. No comando testado quando o programa aciona o ciclo com a linguagem interativa, a tela do comando mostra o ciclo interno em execução. O segundo ciclo executado é a abertura de canal. No programa, são feitas a troca de ferramenta e ajuste de velocidades e avanço para satisfazer o

novo ciclo. O terceiro ciclo executado é a abertura de rosca. A Figura 4.37 ilustra a sequência de execução dos ciclos no comando e a manufatura da peça.

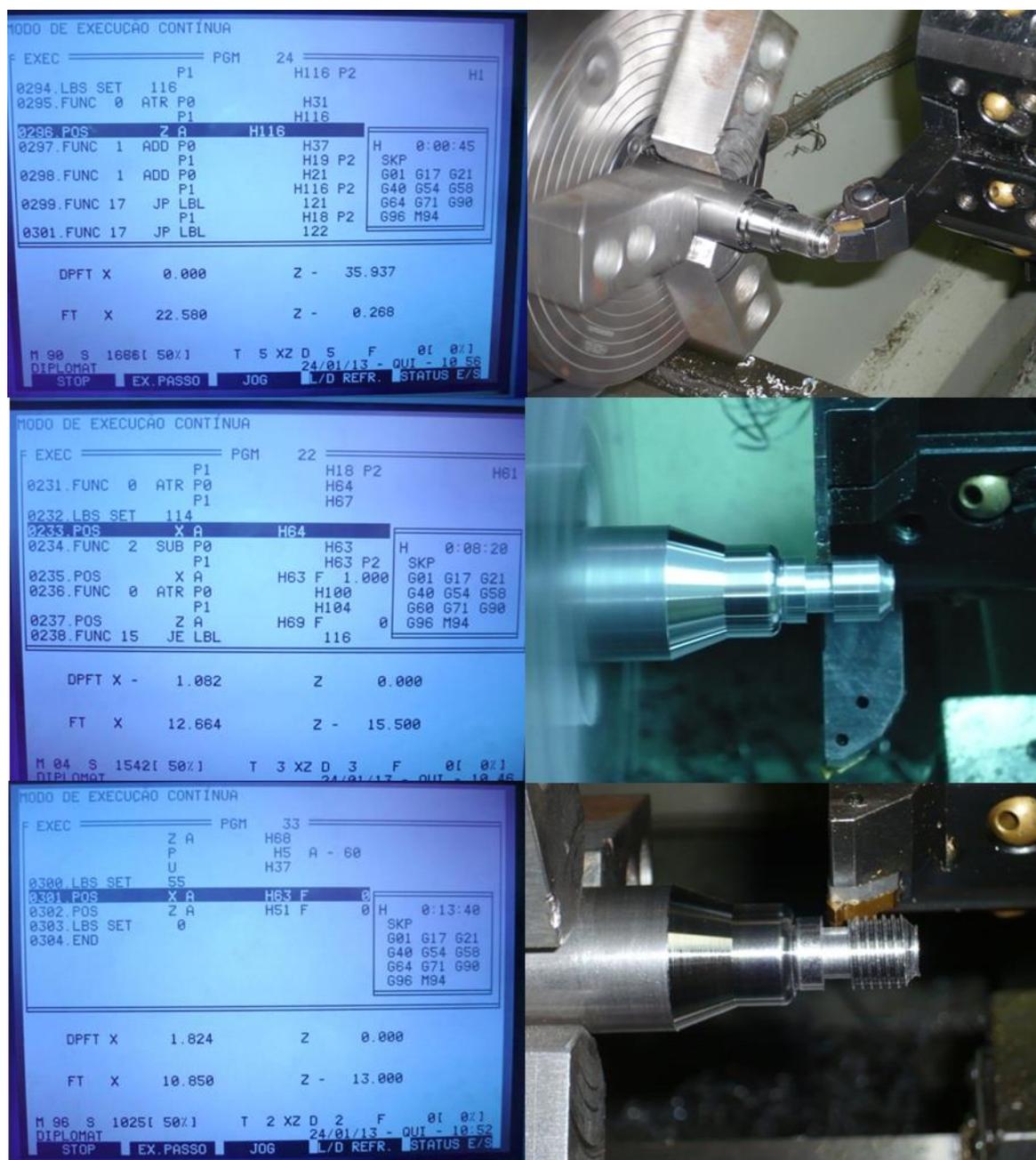


Figura 4.37 - Manufatura da peça modelo

O objetivo do trabalho é propor uma metodologia de desenvolvimento e aplicações de funções CAD/CAM para sistemas dedicados e o desenvolvimento de um aplicativo computacional para averiguação do método. A manufatura da peça foi realizada como uma complementação prática no desenvolvimento do trabalho. Muitos procedimentos

convencionais de usinagem não foram avaliados e implementados, sendo que o objetivo é a aplicação da lógica de programação. Para uma produção seriada alguns procedimentos de usinagem necessitam de ajustes.

5 CONCLUSÃO

O protótipo computacional desenvolvido no contexto do trabalho é composto de módulos com funções, geradas com as metodologias propostas, que podem ser aplicadas individualmente ou interligadas em um sistema de planejamento de atividade de fabricação, especialmente na melhoria e automatização das operações de usinagem. Foi realizado um estudo da aplicabilidade individual dos módulos e do aplicativo completo.

Quanto ao Módulo de Otimização de Parâmetros de Corte (MOPC), que tem como característica processar resultados da aplicação de dados de usinagem especificados para o processo de fabricação, permite disponibilizar condições de corte otimizadas, através do modelo matemático proposto pela equação de Taylor. Embora essa equação seja utilizada na sua forma simplificada, onde relaciona-se apenas velocidade de corte com vida de ferramenta, o sistema possui filtros para configuração dos demais parâmetros de corte que permitem obter resultados equivalentes à aplicação da equação expandida de Taylor.

Como a realização de ensaios experimentais para alimentar o MOPC não faz parte do escopo deste trabalho, para a análise da eficiência desse módulo, sistemas produtivos foram simulados através de dados extraídos de publicações científicas. Embora os modelamentos matemáticos desses trabalhos possam divergir do sistema adotado, os resultados da simulação mostraram que a precisão na determinação de dados de corte otimizados é diretamente proporcional à quantidade e diversificação de informações inseridas no processamento através dos filtros.

O Módulo de Programação Parametrizada (MPP) propõe uma metodologia de geração dos códigos para comando numérico que tem características técnicas equivalentes às funções disponíveis em linguagens semelhantes existentes em comandos de máquina modernas, que representam um recurso adicional de programação. As vantagens alcançadas com o método são a disponibilidade de uma linguagem universal e a compilação independente do comando CNC. Os artifícios computacionais propostos para permitir a utilização do compilador de uma linguagem computacional convencional (PASCAL) como plataforma desse sistema mostrou a viabilidade da técnica.

Para a análise da eficiência desse módulo, são apresentados dois exemplos de programas parametrizados, um para realizar operações de desbastes em ângulo e outro para executar múltiplas tarefas, conforme requisitos especificados pelo usuário. No primeiro caso, trata-se da criação de um novo ciclo de usinagem que pode aumentar a potencialidade do

processo produtivo como um todo, pois pode ser utilizado em qualquer máquina CNC disponível. O segundo exemplo pode ser considerado como característico para aplicação em processos de fabricação que envolvem técnicas de produção como a Tecnologia de Grupo.

Os programas apresentados como exemplo para análise do módulo MPP mostram ainda que a utilização de recursos básicos de programação, como desvios condicionais, repetições e uso de variáveis, são suficientes para o desenvolvimento de uma variedade significativa de programas parametrizados.

A técnica de implementação de uma interface entre o protótipo computacional e um sistema CAD originou o Módulo de Desenho Gráfico (MDG). Os resultados apresentados através de exemplos mostraram que não se trata apenas de uma forma de alimentar o sistema com informações geométricas, mas, principalmente, comprovam a viabilidade de se explorar recursos avançados de automatização de processamento geométrico, disponíveis no CAD (aplicativos AUTOLISP), para alimentar o sistema com dados necessários às funções CAD/CAM dedicadas.

No Módulo de Ciclos de Usinagem MCU, a implementação de funções específicas de cada equipamento é realizada através de rotinas no protótipo CAD/CAM. Embora isso tenha a desvantagem de requerer um estudo prévio da formatação dos ciclos disponíveis no comando, a técnica possibilita um uso simplificado de funções de programação desenvolvidas e otimizadas pelo próprio fabricante do equipamento.

A utilização do protótipo computacional na elaboração de uma peça exemplo mostrou ainda, a flexibilidade do sistema quanto ao interfaceamento entre os diferentes módulos. A consequência disso é a potencialidade para gerar programas CNC com maior eficiência quanto aos dados geométricos e tecnológicos.

As metodologias para desenvolver as funções CAD/CAM, propostas no trabalho, são uma alternativa para sistemas de produção dedicados, em que o objetivo é criar soluções pontuais, simples e eficientes. A metodologia proposta representa ainda uma alternativa à aplicação de sistemas CAD/CAM comerciais. O trabalho mostrou que, apesar da necessidade de conhecimento técnico especializado para implementação, um sistema dedicado pode explorar com maior eficiência as potencialidades do processo produtivo disponível.

Por fim, para futuros trabalhos, o sistema pode absorver um módulo de configuração automática de ciclos de usinagem e um módulo voltado ao sistema CAPP, pois os dados já estão em banco de dados.

REFERÊNCIAS BIBLIOGRÁFICAS

Almeida, S. L. R. **Módulo Computacional, Baseado em Redes Neurais, para a Força de Corte e para a Rugosidade, em Torneamento**, Tese Doutorado, Universidade Estadual de Campinas, UNICAMP, Campinas, 2006.

American National Standards Institute, **ANSI NAS 995** Specification For Computerized Numerical Control (cnc), 1996.

Asilturk, I. Akkus H., “Determining the effect of cutting parameters on surface roughness in hard turning using the Taguchi method”, **Journal Elsevier Measurement**, 2011.

Boehs, L. **Projeto e implantação de um sistema computadorizado de banco de dados de usinagem (CINFUS)**, Tese (Doutorado em Engenharia Mecânica), Universidade Federal de Santa Catarina, 1988.

Canciglieri, O. J.; Young, R. M. Estrutura de informação como suporte em múltiplas vistas do produto num projeto orientado para a manufatura, **SciELO Prod. [online]**, vol.16, n.2, p. 354-374. ISSN 0103-6513, 2006.

Cantu, C. H. **Firebird 2.0 - O Banco de Dados do Novo Milênio**, Ciência Moderna, 2006.

Cantù, M. **Dominando o Delphi7 – A Bíblia**, Makron Books, 2003.

Cassaniga, F.A. **Fácil Programação do Controle Numérico**, CNC Tecnologia Livraria e Editora, São Paulo, 2005.

Cheng, Z.; Jihong, L. An off-line programming system for flexible drilling of aircraft wing structures, **Emerald Current, Assembly Automation**, Vol. 31 Iss: 2, p.161–168, 2011.

Cheypoca, T.; Lekthamrong, C.; Koomgaew, C. High efficiency postprocessor for generic NC machine, **Control Automation and Systems (ICCAS), International Conference on [1-4244-7453-1]**, p.511, 2010.

Coppini, N. L., Baptista, E. A., **Alternativas de otimização dos parâmetros de corte para cenários de fabricação em usinagem**, ENEGEP, 1998, Niterói – RJ.

Da Costa, D.D. **Proposta de uma Metodologia para Construção de um Módulo Especialista Dirigido à Escolha das Condições de Corte**, Tese Doutorado, Universidade Estadual de Campinas, UNICAMP, Campinas, 1997.

Davim, J. P. **Manufacturing: Fundamentals and recent advances**, London, Springer, 361 p., 2008.

Deutsches Institut für Normung, **DIN 66025** Complete G-Code NC Programming as per ISO (DIN 66025) and RS274.

Diniz, A. E., Marcondes, F. C., Coppini, N. L **Tecnologia da Usinagem dos Materiais**, Artliber, São Paulo, 244 p., 2000,

Dos Santos, A.L.B., Duarte, M.A.V., Abrão A.M., Machado, A.R. An optimisation procedure to determine the coefficients of the extended Taylor's equation in machining, **International Journal of Machine Tools & Manufacture**, 1999.

Duarte, M.A.V., Arruda, J.R.F Updating rotor-bearing finite element models using experimental frequency response functions, **RBCM-Revista Brasileira de Ciências Mecânicas** p. 136–149., 1993

ECS - EdgeCam Student Edition_10.0, **User Guide da Pathtrace Engineering Systems**, 2011.

Ferraresi, D. **Fundamentos da Usinagem dos Metais**, Edgard Blücher, São Paulo, 1982.

Figueira, R.J.C.M. **CAD/CAE/CAM/CIM**, Instituto Politécnico do Porto, Instituto Superior de Engenharia, Departamento de Engenharia de Informática, Portugal, 2003.

Gonçalves, M. A. F.; Silva, A. D.; Souza, A. J.; Lorini, F. J. . **Metodologia para otimização automática de parâmetros de usinagem**. In: 6º Congresso Brasileiro de Engenharia de Fabricação, 2011, Caxias do Sul - RS. Anais do VI COBEF (CD-ROM), 2011.

Gonçalves, M. A. F.; Silva, A. D.; Souza, A. J.; Lorini, F. J. . Metodologia para otimização automática de parâmetros de usinagem. **Revista Máquinas e Metais**, v. 560, p. 52-63, 2012.

Gonçalves, M. A.F. **Um Estudo Sobre Implementação de Ciclos de Usinagem Através de Programação Parametrizada em Máquinas de Comando Numérico Computadorizado**, Dissertação de Mestrado, Universidade Federal de Santa Maria, UFSM, Santa Maria, 2007.

Groover, M. **Automação Industrial e Sistemas de Manufatura**, Pearson Prentice Hall, 3ª ed, São Paulo, 2011.

Groover, M.P. **Automation, Production Systems, and Computer-Integrated Manufacturing (3rd Edition)**, Prentice-Hall, Hardcover, 2007.

Groover, M.P. **Fundamentals of Modern Manufacturing: materials, processes, and systems** – 4rd ed. John Wiley & Sons Inc., EUA, 2010.

Gunnar, P. **CAD-Ausbildung für die Konstruktionspraxis. Tl. 3. CAD CAM: Konstruktionsdaten für die Fertigung**, Hanser, Germany, 2002.

International Standards Organization, **ISO1056** miscellaneous general or preparatory, 1975.

International Standards Organization, **ISO6983** Numerical control of machines

program format and definition of address words Part 1: data format for positioning, Line motion and contouring control systems, 1982.

ITC - **Icam Technologies Corporation**, Quebec, Canadá. Disponível em <http://www.icam.com> Acesso em: 15 dezembro 2011.

Koren, Y. **Computer Control of Manufacturing System**. McGraw-Hill, Singapore, 1986.

Liu, Y.; Guo, X.; Yamazaki, K.; Kashihara, K.; Fujishima, M. A study of a universal NC program processor for a CNC system, **The International Journal of Advanced Manufacturing Technology**, Volume 36, Number 7-8, p.738-745, DOI: 10.1007/s00170-006-0890-y, Springer, 2008.

Lorini, F. J. **Tecnologia de Grupo e Organização da Manufatura**, UFSC, Florianópolis, 1993.

Lynch, M. **Computer numerical control: Machining**, McGraw-Hill, New York, 1994.

Lynch, M. **Parametric programming for computer numerical control machine**, Society of Manufacturing Engineers, Michigan, 1997a.

Lynch, M. **The key concepts of CNC. Modern Machine Shop**, McGraw-Hill, Cincinnati, 1997b.

Machado, A. R.; Silva, M. B.; Coelho, R. T.; Abrão, A. M. **Teoria da Usinagem dos Materiais**, Edgard Blücher, São Paulo, 2009.

Manzano, N.G. e Yamatumi, W.Y. **Free Pascal: Programação de Computadores**, Érica, São Paulo, 2007.

Matsumoto, E. Y. **AutoLISP – Linguagem de programação para AutoCAD**, São Paulo, Érica, 2001.

MCS I, Operação e programação – SX 570, Versão 2.2, Novembro, 2010.

MCS II, Descrição de Ciclos Para Tornos – Série 500, Versão 1.3, Novembro, 2010.

Melegari, L. F. **Comparação de desempenho entre programas CNC paramétricos e gerados por sistema CAM**, Dissertação de Mestrado, Universidade Federal de Santa Maria, UFSM, Santa Maria, 2011.

Meneghello, G. P. **Aplicação de Sistema Robótico Utilizando Recursos de Sistema CAD/CAM para o Processo de Fresamento**. Dissertação de Mestrado, Universidade Federal do Rio Grande do Sul, UFRGS, Porto Alegre, 2003.

Newman, S.T.; Nassehi, A. Universal Manufacturing Platform for CNC Machining, **Annals of the CIRP** Vol. 56/1/2007.

Nunes, J. M.; Gomes, J. O.; Souza, G. G.; Sutério, R. Influência da interpolação da trajetória da ferramenta no fresamento a altas velocidades de superfícies complexas, **Tecnologia em Metalurgia e Materiais**, São Paulo, 2008.

Pereira, A. G. **Desenvolvimento e Avaliação de um Editor para Programação NC em Centros de Usinagem**, Dissertação de Mestrado, Universidade Federal do Paraná, UFPR, Curitiba, 2003.

Pereira, J. C **Determinação de Modelos de Vida de Ferramenta e Rugosidade no Torneamento do Aço ABNT 52100 Endurecido Utilizando a Metodologia de Superfície de Resposta (DOE)**, Dissertação de Mestrado, Engenharia Mecânica Universidade Federal de Itajubá, 2004.

Rosso, R. S. U. J. **STEP Compliant CAD/CAPP/CAM System For Rotational Asymmetric Parts**, Tese Doctor of Philosophy of Loughborough University, 2005.

Rosso, R.S.U.; Newman, S.T. Estrutura de Dados para Sistemas CAD/CAM aderente à STEP, **Anais Congresso Ibero-Americano de Engenharia Mecânica**, Portugal, 2003.

Sahin, Y. Comparison of tool life between ceramic and cubic boron nitride (CBN) cutting tools when machining hardened steels, **Journal of Materials Processing Technology**, 2009.

Schützer, K.; Helleno, A. L. Fatores que influenciam a usinagem de moldes e matrizes com alta velocidade. **Revista de Ciência & Tecnologia**, V.12, n.23, p. 7-14, 2004.

Silva, A. D., **Otimização Automática de Parâmetros de Usinagem**, Tese Doutorado, Universidade Federal de Santa Catarina, UFSC, Florianópolis, 1994.

Smid, P. **Cnc Programming Handbook: A Comprehensive Guide to Practical Cnc Programming**, Industrial Press Inc, 2ª ed, New York., 2003.

Souza, A, F; Ulbrich, C, B, L, **Engenharia Integrada por computador e sistemas CAD/CAM/CAE**, Artlibre, São Paulo, 2009.

Souza, A. F. **Contribuições ao fresamento de geometrias complexas aplicando a tecnologia de usinagem com altas velocidades**, Tese Doutorado, Escola de Engenharia de São Carlos, 2004.

Souza, A. F.; Coelho, R. T. Experimental investigation of feed rate limitations on high speed milling aimed at industrial applications, **Int. Jou. Adv. Manufacturing Technology**, DOI 10.1007/s00170-006-0445-2, 2007

Souza, D. J. A. Usinagem de superfícies definidas em CAD, **Estudos Tecnológicos Unisinos**, Vol. 3, V. 1, p.58-73, ISSN 1808-7310, 2007.

Standard for numerically controlled machines, **RS274D** developed by the Electronic Industry Association. The revision was approved in February, 1980.

Stemmer, C. E. **Ferramentas de Corte I**, Universidade Federal de Santa Catarina UFSC, Florianópolis, 2007.

Taylor, F. W. **On the art of cutting metals, an address made at the opening of the annual meeting in New York**, The American Society of Mechanical Engineers, New York, 1906.

Wang, P.; Su, L. Design and Realization of NC Graphic Programming System, **WRI World Congress on Software Engineering**, DOI: 10.1109/WCSE.2010.54, China, 2010.

Wilson, F. **Numerical control in manufacturing**. USA: Mc-Graw Hill Book Company, 1980.

Yau, H. T; Kuo, M. j. Nurbs machining and feed rate adjustment for high-speed cutting of complex sculptured surfaces, **International Journal of Production Research, London**, v.39, n.1, 2001.

Yuliang, C.; Wei, Z.; Wenjuan, H. Research of intranet-based networking and communication systems of the Computer Numerical Control machine tools, **Advanced Computer Control (ICACC), 2nd International Conference on**, 2010.

Yusri, Y., Keith, C. Design of a STEP compliant system for turning operations, **Robotics and Computer-Integrated Manufacturing**, Volume 26, Issue 6, , p. 753–758, 2010

Zalewski, A. The concept of feedback between numerical controlled machine tool and the cam program, **Advances in Manufacturing Science and Technology**, Vol. 35, No. 2 Warsaw, Poland, 2011.

Zhang, X.; Liu, R.; Nassehi, A.; Newman, S.T. A STEP-compliant process planning system for CNC turning operations, **Elsevier ScienceDirect, Robotics and Computer Integrated Manufacturing**, Volume 27, Iss 2, p. 349-356, 2011.