Este exemplar corresponde à redação final da Tese: "Desenvolvimento de um Editor/Simulador para tornos à Controle Numérico" defendida por Geraldo Nonato Telles e aprovada pela Comissão Julgadora em 11 de novembro de 1985.

PROF. NIVALDO LEMOS CUPINI
DEM FEC UNICAMP

DESENVOLVIMENTO DE UM EDITOR/SIMULADOR PARA TORNOS A CONTROLE NUMÉRICO

102/85

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA DE CAMPINAS
DEPARTAMENTO DE ENGENHARIA MECÂNICA

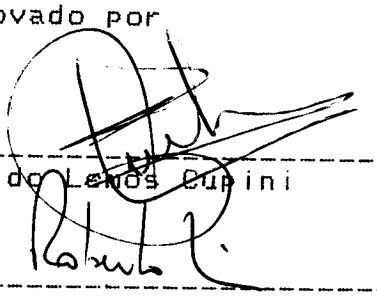
Tese de: Mestrado

Título da Tese: Desenvolvimento de um EDITOR/SIMULADOR para Tornos a Controle Numérico.

Autor: Geraldo Nonato Telles

Orientador: Nivaldo Lemos Cupini

Aprovado por



Prof. Dr. Nivaldo Lemos Cupini

Prof. Dr. Paulo R. Mei

Prof. Dr. Rezende Gomes dos Santos

Campinas, 11 de Novembro de 1985

AGRADECIMENTOS

Esse trabalho concretizou-se porque fui apoiado por:

-- Prof. Nivaldo Lemos Cupini, que acreditou na idéia.

-- Anselmo e Ronaldo, pelas sugestões valiosas.

-- Aristides pelo esmero nos desenhos.

O autor.

À minha esposa e filhas

que tantas alegrias me

têm dado

ÍNDICE

CAPÍTULO 1

INTRODUÇÃO

Item	Página
1.1-Localização dos Objetivos	1
1.2-Generalidades	1
1.3-Histórico do Controle Numérico	2
1.4-Estado da Arte dos Centros de Programação para NC	3
1.5-Microcomputadores	3
1.6-Equipamento escolhido para formação do Centro de Programação	4
1.7-Linguagem Utilizada	4
1.8-Tópicos Básicos de Programação Manual de Tornos NC	4-8
1.9-Bibliografia	9

CAPÍTULO 2

LINGUAGENS DE PROGRAMAÇÃO NC

Item	Página
2.1-Possibilidades básicas de Programação NC	10
2.2-Detalhamento dos Objetivos do EDITOR/SIMULADOR para tornos NC	10
2.3-Estrutura funcional do EDITOR/SIMULADOR	11
2.4-Exemplo de Programação NC	13-17
2.5-Utilização do EDITOR/SIMULADOR	18-25
2.6-Bibliografia	26

CAPÍTULO 3

METODOLOGIA UTILIZADA NO DESENVOLVIMENTO DO EDITOR/SIMULADOR

Item	Página
3.1-Generalidades	27
3.2-Escolha da Linguagem	27
3.3-Caracterização dos recursos do Armário de Controle	28
3.4-O EDITOR - Programa EDIT2	28
3.5-O SIMULADOR	29
3.6-Testes e Otimização do Sistema	31
3.7-Bibliografia	33

CAPÍTULO 4

CONCLUSÕES E PROPOSTAS

Item	Página
4.1-Conclusões	34
4.2-Propostas	34

APÊNDICE

A.1-Introdução	a-1
A.2-O Significado dos caracteres utilizados no MACH-3L	a-1
A.3-Estrutura e sequencia das palavras de programação	a-3
A.4-Função N: Número sequencial de blocos	a-4
A.5-Funções Preparatórias G	a-4/14
A.6-Funções Miscelâneas M	a-14/15

DESENVOLVIMENTO DE UM EDITOR/SIMULADOR

PARA TORNOS

A CONTROLE NUMÉRICO

R E S U M O

Foi desenvolvido um "software", para edição e simulação de programas para tornos NC. O sistema foi escrito em BASIC e utiliza microcomputadores do tipo Apple II.

O trabalho tem em princípio duas finalidades:

-- Auxilio aos profissionais que atuam na área de Controle Numérico.

-- Para formação de profissionais em Controle Numérico.

A B S T R A C T

Software was developed to edition and simulation for NC lathe programs, writtem in BASIC using an Apple II microcompu ter.

The system has two goals:

-- To aid profissionals working with Numerical Control.

-- To instruct NC programers.

GLOSSÁRIO DE SIGLAS

Optou-se nessa dissertação em manter-se as siglas internacionalmente aceitas.

APT	Automatically Programmed Tools
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CPU	Central Process Unit
DNC	Direct Numerical Control
NC	Numerical Control
PC	Personal Computer

CAPÍTULO 1

INTRODUÇÃO

1.1-Localização dos Objetivos

A Indústria Mecânica de nosso país tem apresentado nos últimos cinco anos, uma tendência bastante forte em fabricar e utilizar cada vez mais Máquinas-Ferramenta Controladas Numericamente, fenômeno esse já observado em países de maior desenvolvimento econômico e tecnológico.[1],[2]

Como toda tecnologia nova e emergente é requerido um tempo para assimilação e desenvolvimento. Consequentemente estamos atravessando uma fase deficitária tanto com relação a recursos humanos (através de profissionais competentes) quanto com relação a facilidades materiais.

O presente trabalho é uma tentativa modesta para preenchimento dessas lacunas, pois sua utilização poderá ser :

1º) Para auxílio aos profissionais que atuam na área de Controle Numérico.
2º) Para formação de profissionais em Controle Numérico.

1.2-Generalidades

Ao analisarmos o desempenho dos seres humanos nas mais distintas ocupações notaremos em todas, pontos de processamento de informações. O passo inicial de um tratamento médico é o levantamento de dados clínicos; o processamento dessas informações, juntamente com as informações fornecidas pelos exames clínicos produzirá um diagnóstico prévio. Similarmente um advogado primeiro obtém informações para depois preparar a sua argumentação de defesa ou ataque.

Se solicitarmos a um mestre ferramenteiro a execução de determinada peça mecânica, ele logo exigirá o desenho da mesma, e com este iniciará a compilação das informações nele contidas. Ele processará as informações ali contidas na sua tecnologia especial de trabalho: a especificação do material da peça (bitola, composição química); as máquinas requeridas para a execução da peça; a sequência de operações de usinagem; as condiçoes de corte,etc. Ele precisa pensar antes de iniciar a usinagem da peça. Provavelmente o melhor ferramenteiro seria aquele que conseguisse processar a maior quantidade de informações, dentro de sua especialidade.

A técnica de Controle Numérico consiste basicamente em novo método de organização das informações requeridas no processo de fabricação.

A denominação "CONTROLE NUMÉRICO" dá, à primeira vista, idéia de um outro sistema de controle, como os controles hidráulicos, pneumáticos ou elétricos; porém o Controle Numérico não abrange apenas o controle dos movimentos da máquina, mas também, e principalmente um novo conceito de organizaçao e produção.

O controle expande-se então largamente para fora da

Máquina-Ferramenta, por toda a estrutura organizacional da empresa, tanto nos meios de informações diretos (desenhos, processos, programas), como indiretos (Controle de Qualidade, Planejamento, Análise de Valores), etc.

O termo "CONTROLE NUMÉRICO" surgiu nos E.U.A. no início da década de 50 e sua conceituação mais simplória seria: todas as informações dimensionais de uma peça que já são há muito tempo definidas através de números, seriam então "entendidas" e, processadas automaticamente pelas máquinas NC.

1.3-Histórico_do_Controle_Numérico

A idéia de Controle Numérico surgiu com a necessidade de manufatura de peças de grandes aviões, logo após a 2a. Guerra Mundial.

Em 1949 a Força Aérea dos E.U.A. firmou contrato com a firma J. Parsons para fabricar máquinas programáveis capazes de usinar contornos complexos.[3]

Em 1952, agora já com a participação do Instituto de Tecnologia de Massachusetts (MIT) foi construído um protótipo de fresadora programável com 3 eixos. Essa máquina era dotada de leitora de fita de papel perfurada, unidade de processamento de dados e servo mecanismos acoplados aos fusos da mesa.[3]

Em 1956, logo após o lançamento da linguagem científica FORTRAN (FORmula TRANslation) foi desenvolvido no MIT a primeira linguagem automática para Controle Numérico que constituiu o conhecido APT (Automatically Programmed Tools), que possibilita a edição de programas para Controle Numérico com o emprego de Computadores de grande porte.[1]

No início dos anos 60 a BENDIX, desenvolveu o Controle Adaptativo, onde o controle não se restringe apenas à dimensões, mas também a potência, força, etc.[4]

Em 1967, foram implantadas no Brasil as primeiras máquinas à Controle Numérico inteiramente importadas dos E.U.A.[1]

No início da década de 70 surgiram as primeiras máquinas CNC (Controle Numérico Computadorizado) onde temos a possibilidade de armazenamento em memória principal de programas completos ou mesmo diversos programas.

A década de 70 assinala também o surgimento das primeiras máquinas à Controle Numérico com partes mecânicas fabricadas no Brasil e armários de Controle importados.

Observa-se então uma evolução contínua e notável das máquinas Controladas Numericamente concomitante com os computadores em geral.

Os controles modernos empregam em seu conceito físico (HARDWARE) semi-condutores, transistores, "flip-flops" e com a evolução da micro-eletrônica: circuitos altamente integrados e microprocessadores.

Com relação à parte de potência tem-se utilizado também componentes de estado sólido, tais como tiristores e transistores de potência.

A confiabilidade de cada componente eletrônico em si tem aumentado o que contribui para o aumento da confiabilidade do sistema com implementação cada vez maior de "software".

1.4-ESTADO DA ARTE DOS CENTROS DE PROGRAMAÇÃO PARA N.C.

Atualmente os Centros de Programação para NC existentes e em funcionamento no Brasil podem ser localizados em 2 extremos:

1º) Utilizando computadores do tipo mini ou de grande porte e pacotes de "software" que utilizam linguagens automáticas para programação, tais como: EXAPT,NUFORM,H100, etc.

2º) Sem auxílio de computadores, com a programação sendo feita manualmente e utilizando perfuradoras de fita de papel do tipo TELETYPE, FLEXOWRITER, etc.

Para o primeiro tipo de estrutura temos a comentar: investimento alto, necessidade de pessoal de altíssimo nível para "desempacotar" o "software" e dependência tecnológica com "soft-houses" do exterior. [5]

Para o segundo tipo de Centro de Programação temos evidente baixa produtividade e a incoerência de máquinas com altíssimo grau de automação serem alimentadas com programas produzidos sem um mínimo de infra-estrutura.

Os Centros de Programação para NC utilizando mini ou grandes computadores tem sua validade em empresas de grande porte, as quais muitas vezes já empregam grandes computadores em forma de "TIME-SHARING" e possibilitam inclusive (no campo da Engenharia Mecânica) a utilização de grandes sistemas CAD/CAM e DNC.

Para empresas de porte médio e pequeno (que empregam 80% da mão de obra utilizada em Mecânica) não é economicamente justificável o emprego de computadores de porte para aplicações em NC, e o que se viabiliza são Centros de Programação do tipo "manual" onde a produtividade fica bastante comprometida.

É de nosso entender que existe então uma demanda a ser suprida para Centros de Programação que ocupem uma posição intermediária entre esses dois extremos.[6]

Por outro lado existe a preocupação de desenvolvermos "Sistemas de Apoio ao Ensino em NC" pois muitas vezes não temos condição de demonstrações práticas aos nossos alunos dos "Cursos de Programação de Máquinas-Ferramenta NC", devido principalmente ao custo hora-máquina de tais equipamentos ser bastante elevado, e, pior ainda, devido muitas vezes a Entidade de Ensino não possuir tais equipamentos.

1.5-Microcomputadores

Com o advento dos microprocessadores, onde basicamente temos uma CPU em um único "chip", permitiu-se a utilização de microcomputadores que, de um lado automatizam tarefas anteriormente feitas manualmente e de outro lado executam trabalhos que antes só eram executados em mini ou grandes computadores.

Evidentemente o grande trunfo dos microcomputadores reside no seu baixo custo em termos de "hardware".

Os microcomputadores do tipo Apple e PC da IBM já estão sendo utilizados no exterior para constituiçaõ de "Micro-Centros de Programação para Controle Numérico".[7]

Felizmente existem Entidades (na maior parte dos casos, Universidades) que preocupam-se em preencher a lacuna existente

em termos de "software". Exemplo disso pode ser visto no sistema EDITE desenvolvido na Universidade Federal de Santa Catarina.[8]

1.6=Equipamento_escolhido_para_formação_do_"Centro_de_Programação"

A configuração necessária para constituição de tal Centro consiste em:

1 CPU Apple com 64 k
1 monitor de vídeo
2 unidades de disco-flexível de 5 1/4"
1 impressora matricial de baixa velocidade
1 unidade perfuradora/leitora de papel

Evidentemente, para o acoplamento dos diversos periféricos acima à CPU, são requeridas interfaces.

A escolha desse tipo de equipamento deve-se principalmente à sua capacidade de resolução gráfica ser considerada média e seu baixo custo

1.7=Linguagem_utilizada

O presente trabalho foi desenvolvido em BASIC devido principalmente aos fatores:

1º) Facilitar a implementação do mesmo por parte de terceiros
2º) O BASIC (APPLESOFT) possui recursos de computação gráfica poderosos e de utilização bastante simples.

Utilizou-se também o compilador TASC (MICROSOFT)[9] com a finalidade de aumentar a velocidade de execução.

1.8=Tópicos_Básicos_de_Programação_Manual_de_Tornos_NC

Uma programação padronizada refere-se ao relacionamento internacionalmente aceito entre letras, números e símbolos de um lado e funções das máquinas do outro lado. Deve-se ressaltar que as entidades normativas aparentemente não tiveram agilidade suficiente para acompanhar a notável evolução nesse campo. As normas DIN e ISO sôbre o tema são normas de caráter provisório e tem servido apenas como referência geral. [10]

Com o modesto intuito de facilitar o acompanhamento dos próximos capítulos, apresentamos a seguir um breve resumo das funções de programação do torno mecânico COSMOS-30 equipado com o armário de controle MACH 3L de fabricação das Indústrias ROMI. Um aprofundamento maior do tema poderá ser feito no Apêndice A.

1.8.1=Sistemas_de_Coordenadas_e_Códigos_de_Eixos

A norma VDI 3255 [11] prevê as seguintes regras:

a) O espaço de trabalho (interação ferramenta x peça) é referenciado a um sistema de coordenadas cartesianas ortogonais que seguem as mesmas convenções da Geometria Analítica.

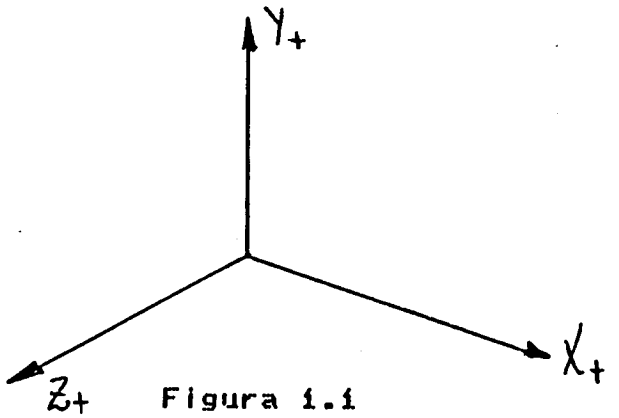


Figura 1.1

b) O eixo Z é perpendicular ao plano XY e sempre refere-se ao eixo-árvore da máquina ferramenta. No caso do torno COSMOS-30 teríamos a seguinte disposição: (visto de cima)

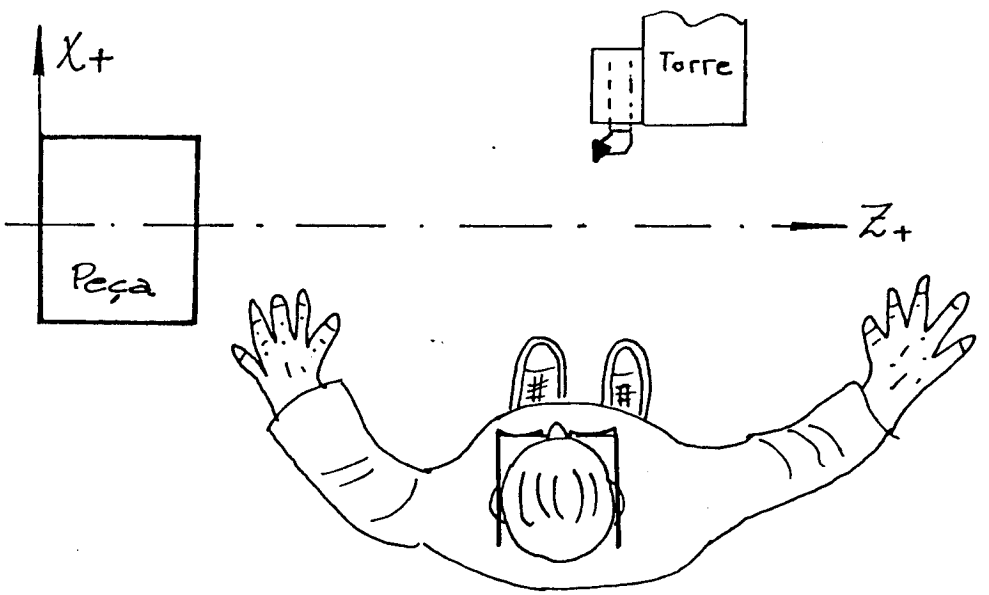


Figura 1.2

Os controles modernos permitem a utilização do "zero-peça" (origem) flutuante e definido no programa NC do usuário com o código G92, de maneira que o programador localize o "zero-peça" onde for mais cômodo.

O princípio de programação de tornos NC consiste em especificar-se o acesso da ponta teórica da ferramenta pelas coordenadas do ponto através das coordenadas X e Z do mesmo, referenciadas ao "zero-peça" (origem), previamente definido.

1.8.2-Estrutura_sintática_dos_blocos_de_informações_em_programas_NC

O programa NC é constituído de "blocos" (linhas) de programação. Os blocos são constituidos de palavras, que normalmente são dispostas na sequência.

1. Palavra para número de bloco: N...
2. Palavra para condição de percurso: G...
3. Palavra de coordenadas: X,Z,I,K, etc

4. Palavra para avanço: F...
5. Palavra para velocidade de corte: S...
6. Palavra para ferramenta: T...
7. Palavra para função auxiliar: M...

Devemos observar que não necessáriamente um bloco de programa NC apresenta todas essas informações sempre, sendo que a maioria das informações tem característica "modal", isto é continua válida enquanto não ocorrer uma que modifique-a.

1.8.2.1-Função N: Número sequencial de Blocos

É usualmente a primeira informação do bloco. Normalmente utilizada para definir a cronologia das operações mecânicas. O caractere N é então seguido de até 4 dígitos numéricos. Da mesma maneira que na programação BASIC os programadores de NC constantemente utilizam-se de numeração crescente com incrementos de 10, o que possibilita futuras inclusões de até 9 blocos, sem alterar a numeração dos blocos anteriormente escritos.

1.8.2.2-Funções Preparatórias G

As funções G, foram criadas para definir uma grande variedade dde condições de percurso. A norma DIN 66025 especificou-as de G00 à G99. Abaixo relacionamos algumas dessas funções necessárias para operações simples no torno COSMOS-30:

CóDIGO	SIGNIFICADO
G00	movimento rápido (aproximação ou afastamento)
G01	interpolação linear (usinagem)
G02	interpolação circular horária (usinagem)
G03	interpolação circular anti-horária (usinagem)
G20	programação da coordenada X em diâmetro
G21	programação da coordenada X em raio.
G33	corte de rosca (usinagem)
G61	definição de áreas de segurança
G71	programação em unidades métricas
G90	programação em coordenadas absolutas
G91	programação em coordenadas incrementais (relativas)
G92	definição do zero-peça
G96	especificação da velocidade de corte constante

1.8.2.3-Funções de coordenadas: X, Z, I, e K

A palavra de coordenada consiste das letras X,Z,I ou K seguida do valor algébrico da coordenada. A indicação fracionária é feita pelo ponto decimal. A resolução de programação é 0,0001 milímetro.

Devemos observar que na maior parte dos casos utiliza-se a coordenada X como especificação de diâmetro e que I e K referem-se indiretamente à X e Z respectivamente.

1.8.2.4-Palavra para avanço: F

Normalmente em tornos tem-se a indicação do avanço da ferramenta [mm/volta] especificado pela palavra F seguida do valor numérico do mesmo com indicação fracionária pelo ponto decimal. A resolução de programação é 0,001 [mm/volta].

1.8.2.5-Palavra para velocidade de corte: S

A especificação da velocidade de corte constante é obtida pela programação da função G96 seguida da palavra de programação S, e o valor numérico da mesma em [m/min] com resolução de 0,1.

1.8.2.6-Palavra de Especificação de Ferramenta: T

A ferramenta é especificada pela função T seguida de 4 dígitos numéricos, onde os dois primeiros indicarão a posição da mesma na torre (porta-ferramentas), e os 2 últimos o endereço de compensações X, Z, raio de arredondamento da pastilha e quadrante de trabalho.

Deve-se observar que a função T especifica a ferramenta e a função auxiliar M6 proporciona a indexação da mesma na posição de trabalho.

Observe-se também que as compensações de arredondamento de pastilha e quadrante de trabalho, normalmente são utilizadas para ferramentas de acabamento.

1.8.2.7-Funções Auxiliares: M

Analogamente as funções G, temos a especificação das funcões auxiliares (miscelâneas) M. Abaixo relacionamos as mais importantes no caso do Controle em estudo:

CóDIGO	SIGNIFICADO
M00	Parada Incondicional
M03	Liga eixo-árvore (sentido horário)
M04	Liga eixo-árvore (sentido anti-horário)
M05	Desliga eixo-árvore
M06	Indexação da ferramenta
M08	Liga fluido de corte
M09	Desliga fluido de corte
M30	Fim de Programa

1.8.2.8-Definição de Áreas de Segurança

O Controle MACH 3L permite a definição no programa NC de áreas de segurança, o que permite ao programador (ou operador NC) limitar e controlar o percurso da ferramenta em áreas consideradas críticas. A forma geral de programação dessas áreas é feita por:

G61 Ln Xx Zz Ii Kk

onde:

n = número da área (1<n<8)
x = diâmetro interno
z = encosto
i = diâmetro externo

k = comprimento

As áreas de segurança L1 à L4 são normalmente utilizadas para definir: castanha, placa, contra-ponto, etc. As áreas L5 à L8 são normalmente utilizadas para definir a peça em bruto.

Por exemplo para a castanha e peça indicadas abaixo teríamos as seguintes definições de áreas de segurança.

```
G61 L1 X115 Z-10 I135 K15
G61 L2 X95 Z-10 I115 K0
G61 L5 X0 Z0 I115 K87
```

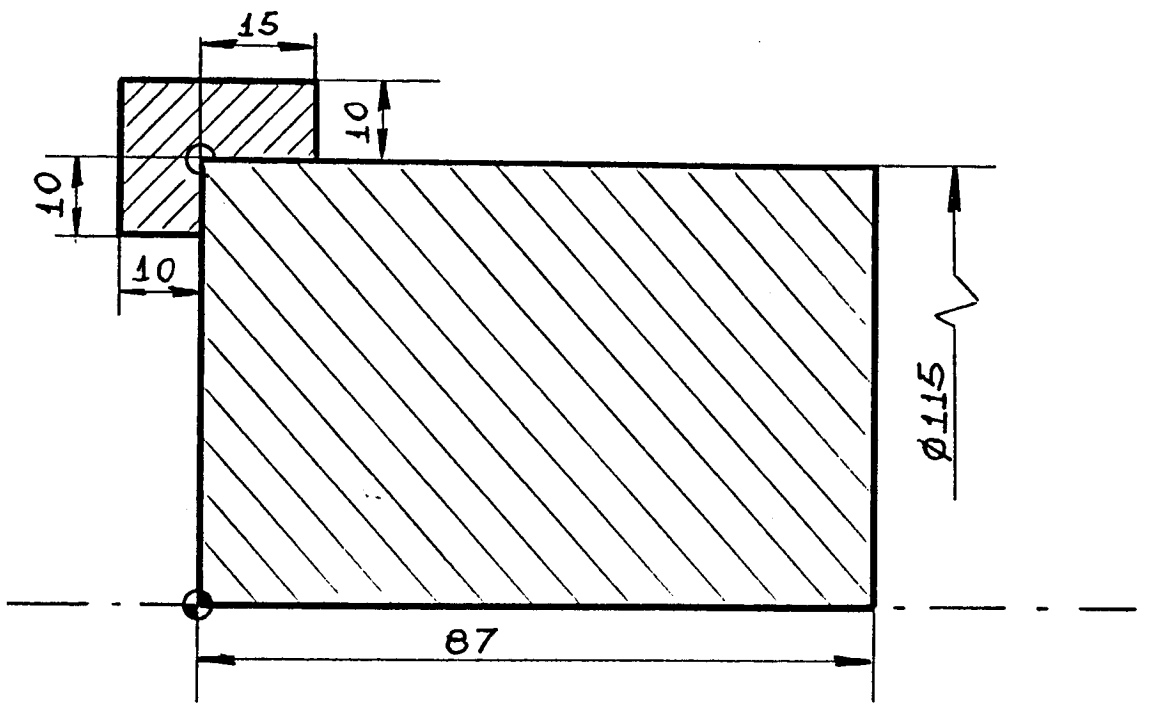


Figura 1.3

Devemos observar que a castanha do exemplo acima, para fins de definição foi dividida em dois retângulos definidos por L1 e L2.

1.2-Bibliografia

[1] STEMMER, Gaspar E. Estágio Atual do Comando Numérico no Brasil. São Paulo, Revista Máquinas & Ferramentas. Anais do I Seminário de C.N.-SOBRACON, maio 1981.

[2] HATSCHEK, R. L. NC PROGRAMING - Special Report. American Machinist February 1980. p. 119.

[3] PRESMAN, Roger S. & WILLIAMS, John E. Numerical Control and Computer-Aided Manufacturing. New York, John Wiley & Sons, 1977. 310p.

[4] BéZIER, P. Numerical Control, Mathematics and Aplications. [Emploi des Machines à Commande Numerique]. Translated by Forrest, A. R. London, John Wiley & Sons, 1972. 240 p.

[5] GENDRE, J. C. PROMO - An NC System for Small Computers.International Conference on Computer Aided Manufacture and Numerical Control. Glasgow, University of Strathclyde, 1974. p. 9-1.

[6] BRECHTEL, H.H. A Modular System and its Application. International Conference on Computer Aided Manufacture and Numerical Control. Glasgow, University of Strathclyde, 1974. p. 8-3.

[7] RYONO, Byron. Let your Apple do the Drafting. Philadelphia, A+ Magazine, dezembro 1984. 254 p.

[8] QUEIROZ, Abelardo A. Um programa com EDITE. São Paulo, Boletim da SOBRACON, março-abril 1985. 12 p.

[9] MICROSOFT. TASC User's Manual. Bellevue, Microsoft Inc, 1981. 97 p.

[10] DIN 66025: Programmaufbau fur numerisch gesteuerte Arbeiteismachinen, 1969.

[11] VDI 3255: Programmieren numerisch gesteuerter Werkezeugmaschinen - Festlegung der Koordinatenatenachsen und Zuordnung der Bewegungsrichtungen, Dusseldorf, 1968

CAPÍTULO 2

LINGUAGENS DE PROGRAMAÇÃO NC

2.1-Possibilidades Básicas de Programação NC.

Provavelmente o maior impacto proporcionado pelas Máquinas de Controle Numérico reside na **transferência de responsabilidade** da Oficina Mecânica para o Centro de Programação NC. Isto como consequência do **controle à distância**, onde o operador tem realmente a função de supervisionar o processo, e só interfere no ciclo de trabalho no caso de ocorrências de problemas. Por outro lado no Centro de Programação é feito antecipadamente todo o planejamento das operações que as máquinas deverão executar (especificação de ferramentas, condições de corte, trajetória, etc).[1]

A programação de máquinas NC, pode (analogamente com a computação em geral) utilizar-se de linguagens de baixo nível e de linguagens de alto nível. No caso de NC é comum referir-se à essas formas de programação como: **Programação Manual** e **Programação Automática**, e entende-se por:

-**Programação Manual**, quando não se utiliza nenhum equipamento computacional externo ao NC e a linguagem é constituída quase que de códigos fortuitos, e essencialmente numéricos.

-**Programação Automática**, quando frequentemente utiliza-se computadores de grande porte e linguagens de programação envolvendo palavras chave do idioma inglês.

O advento dos micro-computadores viabilizou uma possibidade intermediária entre esses extremos , que podemos classificar de **Programação Semi-Automática.**

2.2-Detalhamento dos Objetivos do EDITOR/SIMULADOR para tornos NC.

-"**gerar**" programas de maneira rápida e precisa.
-"**simular**" e "**depurar**" programas antes do encaminhamento para as máquinas NC.
-"**otimizar**" e modificar de forma rápida e racional programas já existentes .
-"**arquivar**" e gerenciar o acervo de programas de NC desenvolvidos no Centro de Programação, de forma a garantir uma estrutura ágil e funcional.

Para tanto foram desenvolvidos os seguintes módulos:

-EDIT2: dirigido à edição de programas novos; alteração de programas já existentes; gravação e leitura de programas em discos flexíveis.
-SIMULA2: dirigido à simulação de programas principais para tornos de NC.
-SUBROT1: direcionado à simulação de sub-rotinas de primeiro nível para tornos de NC.
-SUBROT2: dirigido à simulação de sub-rotinas de segundo nível em programas de tornos de NC.

2.3=Estrutura_funcional_do_EDITOR/SIMULADOR.

Uma unidade de disco flexível é utilizada com o "software" e outra unidade de disco é reservada para o arquivo de programas do usuário.

No esquema básico de trabalho (para programas novos) podemos distinguir as etapas seguintes de trabalho:

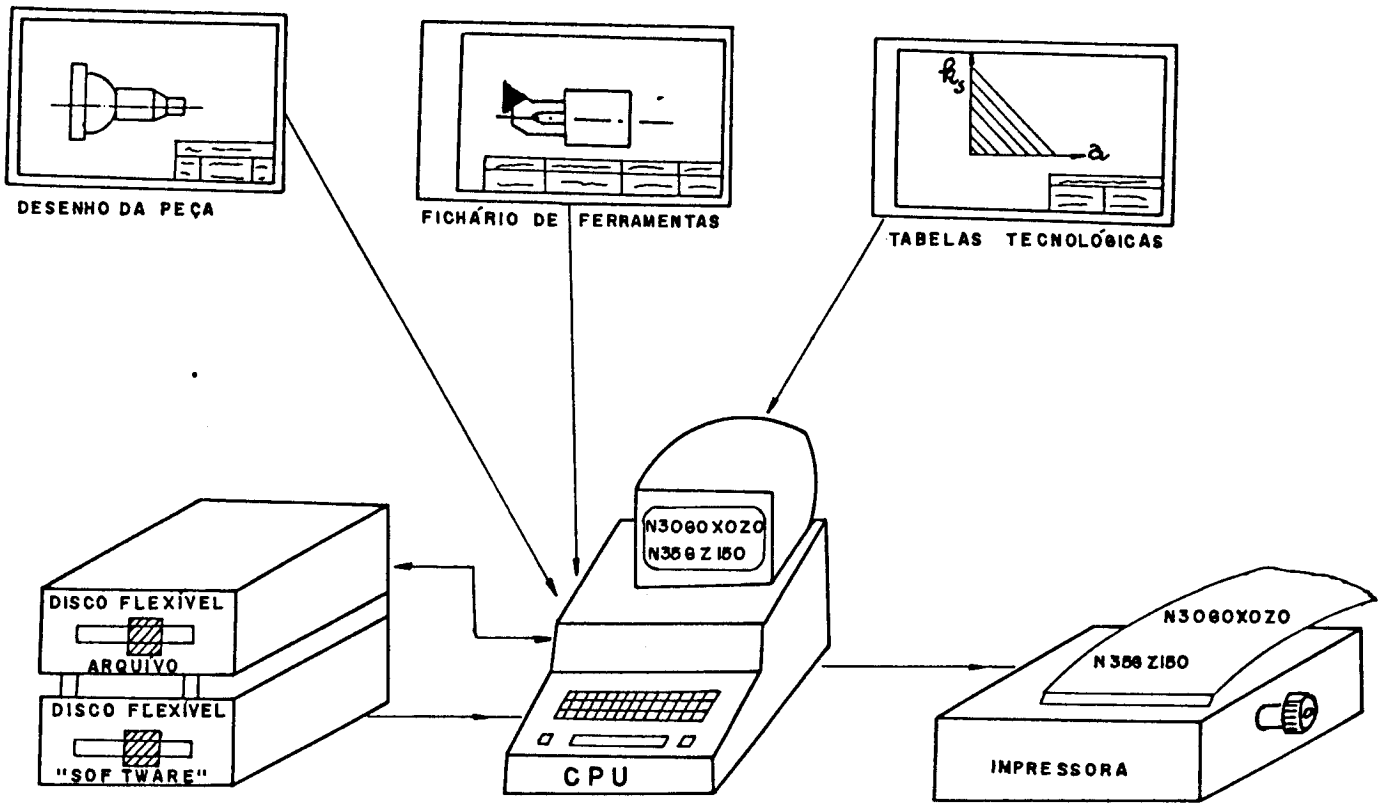
1º)EDIÇÃO: Utilizando o desenho do produto; tabelas tecnológicas e regras de semântica e sintaxe da linguagem, edita-se o programa de NC.

2º)SIMULAÇÃO: A trajetória da ponta da ferramenta é visualizada no monitor de vídeo. Caso haja necessidade de alterações, retorna-se à primeira etapa.

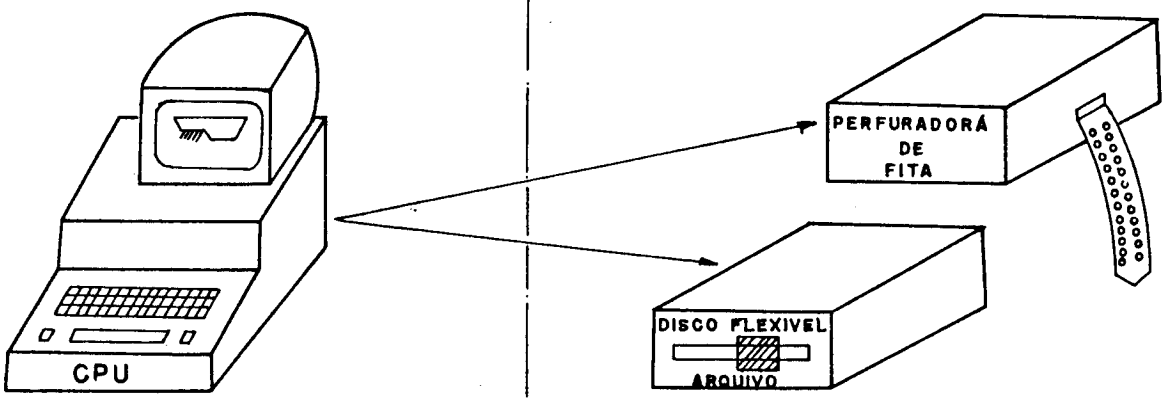
3º)TRANSFERÊNCIA DOS PROGRAMAS: Com o programa já otimizado, pode ser emitido listagem impressa do mesmo; poderá ser feita a gravação em disco flexível; emissão da fita perfurada, ou ainda a transmissão direta para o armário de Controle da Máquina-Ferramenta.

4º)TESTES NA OFICINA MECÂNICA: As primeiras peças do lote são executadas com atenção especial (normalmente com a assistência do programador) e se houver necessidade de algumas alterações, elas são executadas diretamente no painel do armário de Controle.

5º)EXECUÇÃO DO LOTE DE PEÇAS: Após a otimização do programa, a Máquina-Ferramenta entra em regime de produção normal.

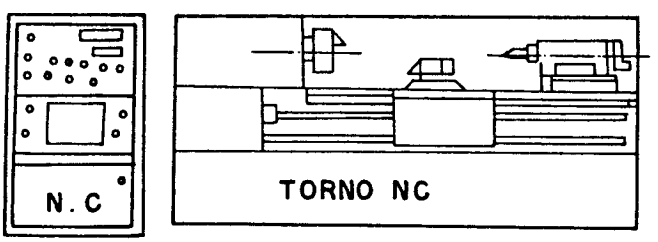


1º ETAPA EDIÇÃO



2º ETAPA SIMULAÇÃO

3º ETAPA TRANSFERÊNCIA



4º E 5º ETAPA TESTE E EXECUÇÃO

FIGURA 2.1

2.4-Exemplo de Programação NC utilizando o EDITOR/SIMULADOR

Suponha-se a necessidade de torneamento de peça conforme desenho:

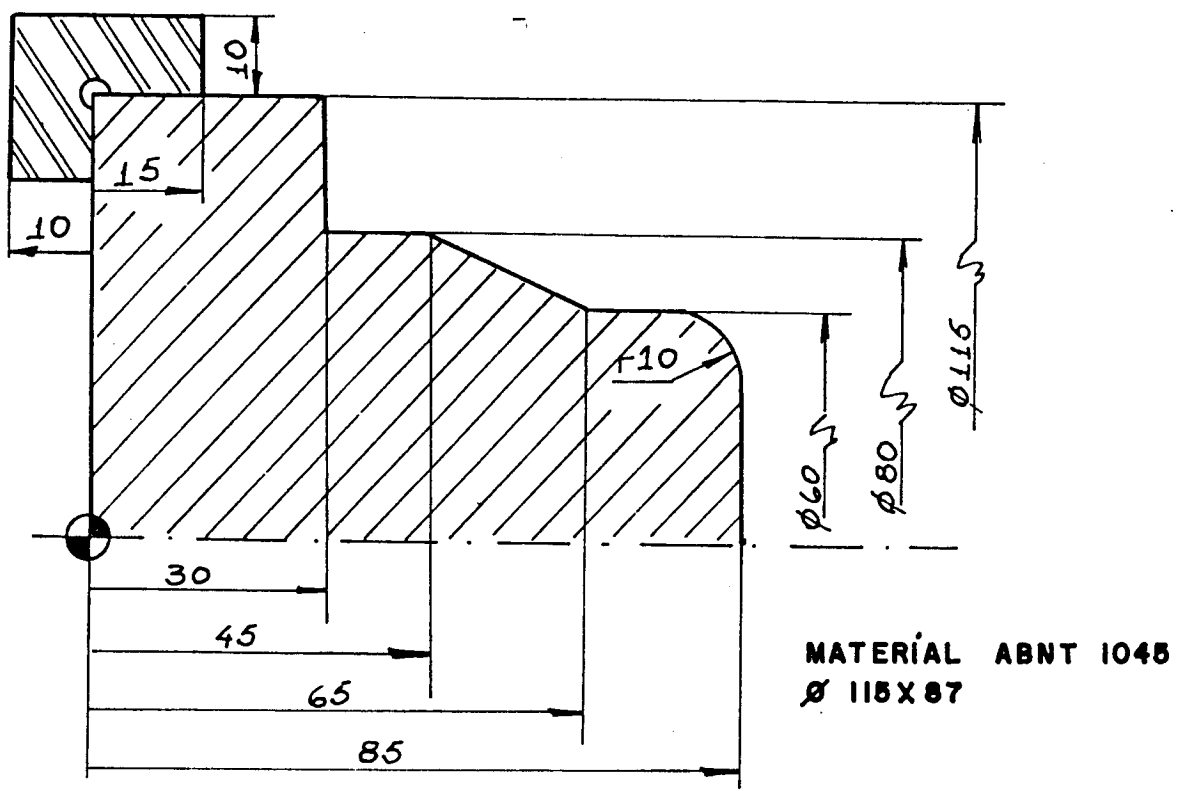


Figura 2.2

O processo de fabricação dessa peça envolve 3 operações distintas: faceamento, desbaste cilíndrico e acabamento conforme detalhamento e e correspondente programação NC apresentada a seguir.

Devemos ressaltar que a máquina base para esse trabalho é o COSMOS-30 com controle MACH 3L de fabricação das Indústrias Romi, cujas funções principais de programação NC estão relatadas no Apêndice A.

1a. Operação: Faceamento

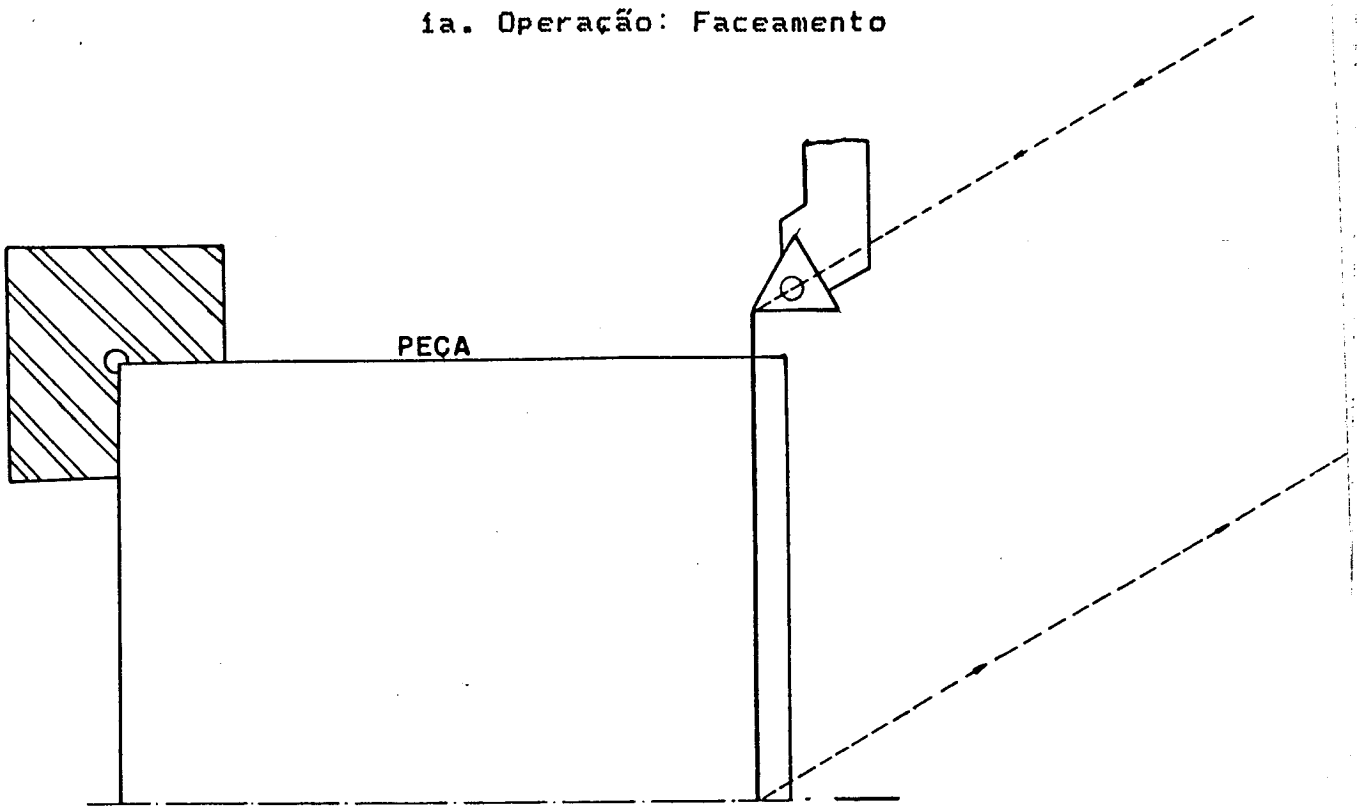


Figura 2.3

A parte do programa NC para execução dessa operação é relacionada a seguir com explicações da programação entre parentêses:

```
N10 G99;...FACEIA.    (referencia: zero máquina)
N20 T0101             (definição de ferramenta)
N30 G X Z             (movimento rápido para zero-máquina)
N40 M6                (indexação de ferramenta de faceamento)
N50 G92 X180. Z200.   (definição do zero-peça)
N60 G96               (definição da velocidade de corte constante)
N70 G47 S160.         (efetiva velocidade de corte)
N80 G92 S3000 M4      (definição de rotação máxima e sentido de giro)
N90 G46               (inibe a velocidade de corte constante)
N100 G X125. Z85.     (aproximação: movimento rápido)
N105 G47              (reativa velocidade de corte constante)
N110 G1 X F.2         (usinagem: faceamento)
N112 G X180. Z200.    (retorno ao ponto de troca)
N113 G46              (inibe a velocidade de corte constante)
```

2a. Operação: Torneamento (desbaste)

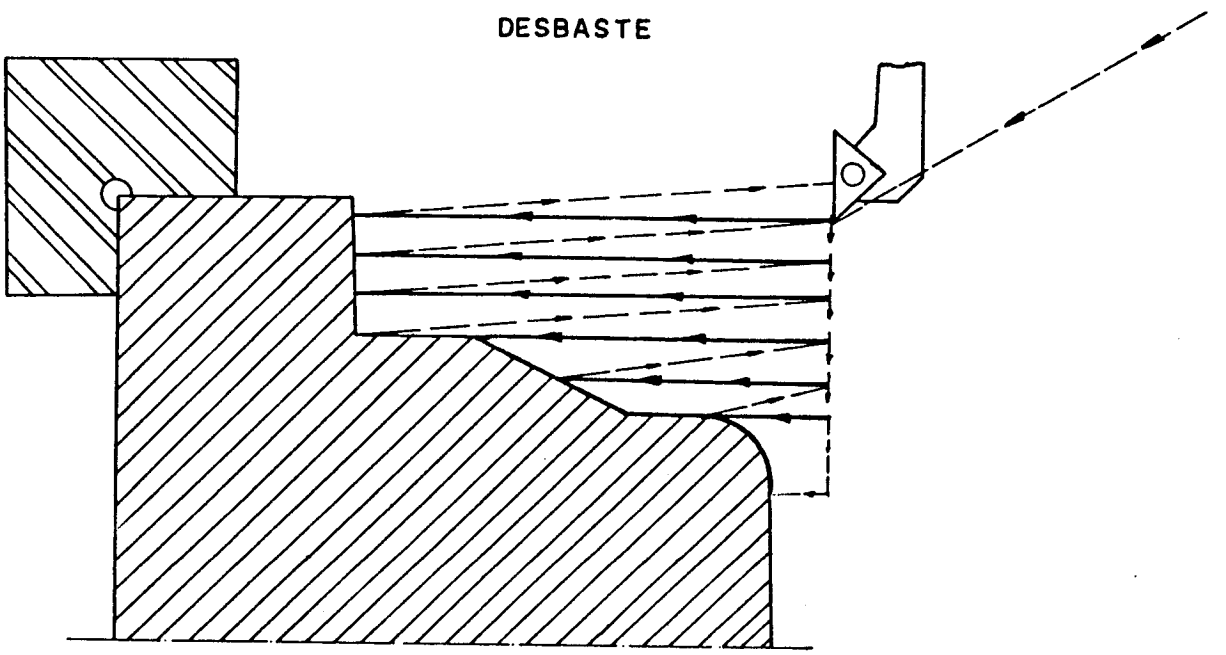


Figura 2.4

```
N114 G99;...DESBASTE.(referência:zero-máquina)
N115 T0202           (definição de ferramenta)
N116 G X Z           (retorno ao zero-máquina)
N117 M6              (indexação de ferramenta)
N118 G92 X180. Z200. (definição do zero-peça)
N119 G96             (definição da velocidade de corte constante)
N120 G47 S160. M4    (reativa a velocidade de corte; sentido de giro)
N121 G46             (inibe a velocidade de corte constante
N122 G X110.5 Z87    (aproximação: movimento rápido)
N123 G47             (reativa velocidade de conte constante)
N130 G74 X80.5 Z30.1 I10. U F.3   (ciclo-fixo de usinagem)
N140 G X70.5         (aproximação: movimento rápido)
N150 G1 Z55.5        (torneamento cilíndrico)
N160 G X75. Z87.     (retorno: movimento rápido)
N170 X60.5           (aproximação: movimento rápido)
N180 G1 Z75.1        (torneamento cilíndrico)
N190 G X65. Z87.     (retorno: movimento rápido)
N200 X50.            (aproximação: movimento rápido)
N210 G1 X60.5 Z81.75 (torneamento cônico)
N220 G Z87.          (retorno: movimento rápido)
N230 X40.5           (aproximação: movimento rápido)
N240 G1 Z85.1        (aproximação lenta)
N250 G3 X60.5 Z75.1 I40.5 K75.1 (interpolação circular anti-horária)
N260 G1 Z65.         (torneamento cilíndrico)
N270 X80.5 Z44.9     (torneamento cônico)
N275 G X180. Z200.   (movimento rápido: retorno ao ponto de troca)
N280 G46             (inibe velocidade de corte)
```

3a. Operação: Acabamento

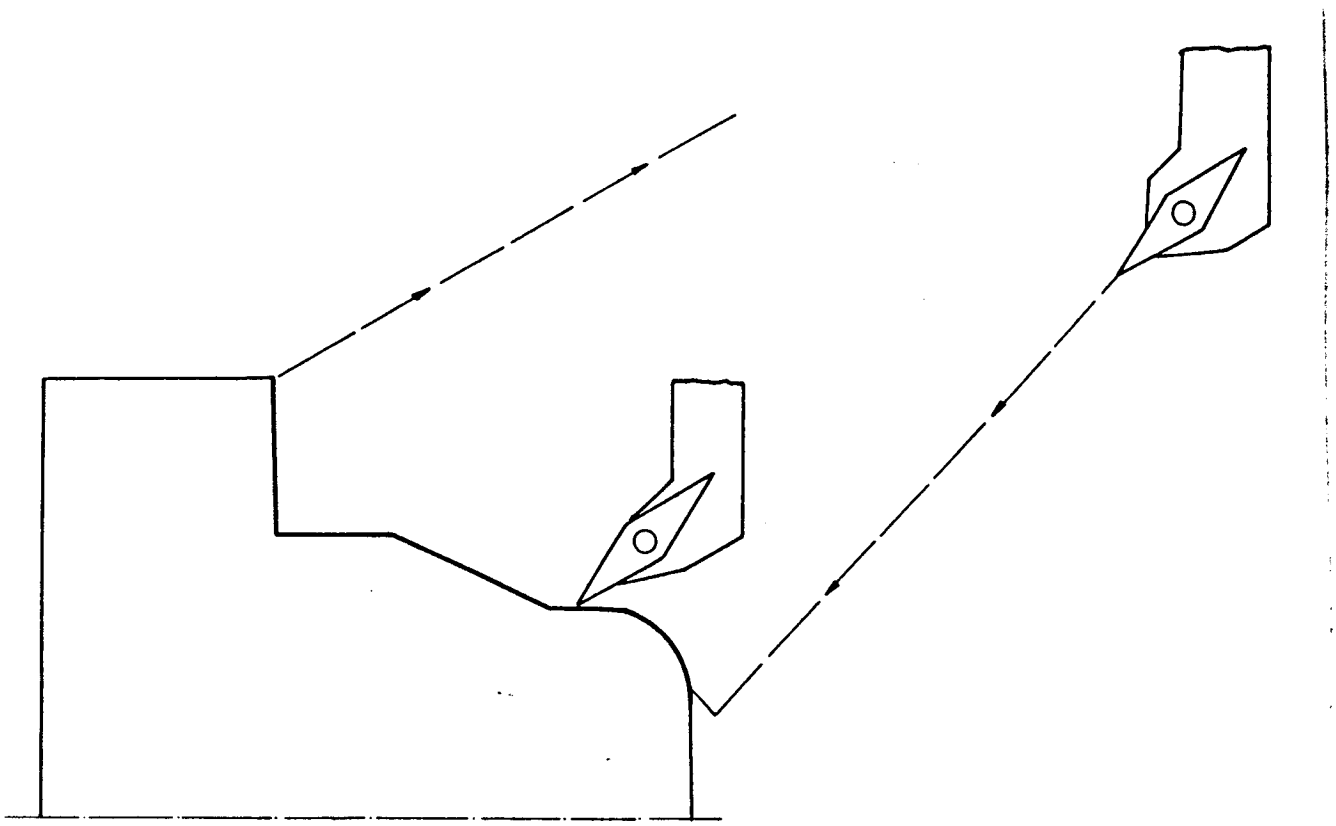


Figura 2.5

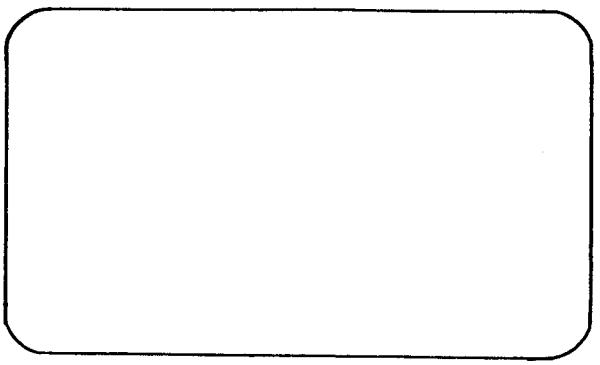
```
N290 G99;...ACABA.      (referência zero-máquina)
N300 T0404              (especificação da ferramenta de acabamento)
N310 G X Z              (movimento rápido ao zero-máquina)
N320 M6                 (indexação de ferramenta)
N330 G92 X180. Z200.    (definição: zero-peça)
N340 G96                (definição: velocidade de corte constante)
N350 G47 S190.          (ativação da velocidade de corte)
N360 G92 S3000 M4       (definição de rotação limite e sentido de giro)
N370 G46                (inibe a velocidade de corte constante)
N380 G X20. Z 87.       (movimento rápido: aproximação)
N385 G47                (ativa a velocidade de corte constante)
N386 G42                (compensação do raio da pastilha)
N390 G1 X4. Z85. F.1    (aproximação lenta)
N400 G3 X60. Z75. I40. K75. (interpolação circular anti-horária)
N410 G1 Z65.            (torneamento cilíndrico)
N420 X80. Z45.          (torneamento cônico)
N430 Z30.               (torneamento cilíndrico)
N440 X117.              (faceamento)
N445 G40                (cancela compesação do raio da pastilha)
N446 G1 X120. F1.       (afastamento)
```

```
N447 G X180. Z200.    (retorno ao ponto de troca de ferramenta)
N448 G46              (inibe velocidade de corte)
N450 G99;...FIM       (referência: zero-máquina)
N460 T0101            (especificação de ferramenta)
N470 G X Z            (movimento rápido: zero-peça)
N480 M6               (indexação de ferramenta)
N490 M30;FIM.         (fim de programa)
```

2.5-Utilização do EDITOR/SIMULADOR.

O sistema foi desenvolvido de forma a ter sua utilização simplificada ao extremo, assim sendo o único requisito prévio para sua utilização é o conhecimento de programação NC. Para facilitar sua utilização optou-se pelo emprego de "cardápios" de orientação.

A representação doravante utilizada para representação de informações apresentadas no monitor de vídeo do computador será:



As informações digitadas pelo usuário serão aqui representadas na forma sublinhada, para distinguir-se daquelas que o próprio computador apresenta. A posição indicada pelo cursor no vídeo será representada por: . Caso não seja perfeitamente evidente a tecla RETURN será representada por:

Todo o "software" está armazenado em um disco mestre, com partida automática ao ligar-se a chave de energia do micro computador. Após 12 segundos temos a seguinte mensagem apresentada no vídeo:

```
 SEJA BENVINDO AO
SISTEMA SIMULADOR P/
 CONTROLE NUMERICO

        DE
MAQUINAS FERRAMENTA
```

E após 10 segundos temos à disposição o cardápio de OPÇÕES PRINCIPAIS:

```
*****OPCOES PRINCIPAIS*****
<1> GERAR PROGRAMA NOVO
<2> GERAR SUB-PROGRAMA NOVO
<3> CARREGAR PROGRAMA DO DISCO
<4> CARREGAR SUB-PROGRAMA DO DISCO
<5> SIMULACAO
<6> OPCOES SECUNDARIAS

<>
```

Nessa situação o usuário obrigatoriamente fará a opção de 1 à 6, conforme necessidades e possibilidades que passaremos a explorar a seguir:.

2.5.1- OPÇÃO (Principal) (1): GERAR PROGRAMA NOVO.

O sistema apresenta então a mensagem:

```
DIGITE NUMERO DO PROGRAMA
      <          >
```

É possível a numeração de programas de 0 a 9999.
Caso o número do programa possua menos de 4 dígitos haverá necessidade de digitação da tecla RETURN após a entrada do número do programa. Por exemplo: 101 .
Após essa entrada poderá ser visualizado (alguns segundos) a mensagem:

```
PARA TERMINAR DIGITE  <FIM>
```

Sendo "FIM" a palavra chave utilizada pelo EDITOR para encerrar programas NC.
O sistema apresenta então a "tela":

```
N
```

Sendo que nessa situação o usuário passa a introduzir o programa NC, utilizando todos caracteres disponíveis no teclado e, evidentemente coerentes com a linguagem em questão.
Deve-se ter em mente que o EDITOR opera com o conceito de bloco (linha) de programação NC, de forma que qualquer alteração na linha pode ser feita utilizando normalmente os controladores de cursor: ← (esquerda) e → (direita), antes de digitar-se a tecla (RETURN).
Observe-se também que o próprio computador apresenta no início da linha o caractere N (numéro do bloco de programação NC), dispensando-se então a digitação repetitiva do mesmo pelo usuário.
A introdução do início do programa-exemplo relatado no item 2.3, resultaria na seguinte "tela":

```
N10G99.....EACEIA.↵
N20I0101↵
N30GXZ↵
N40M6↵
N50G92X180.Z200.↵
N60G96↵
NZ0G4ZS160.↵
N80G92S3000M4↵
N90G46↵
N100GX125.Z85.↵
N110G1XE.2↵
N112GX180.Z200.↵
```

Podemos observar que são apresentadas 12 linhas de em cada "tela". O final do programa teria o seguinte aspecto:

```
N430Z30.↵
N440X11Z↵
N445G40↵
N446G1X120.E1.↵
N447GX180.Z200.↵
N448G46↵
N450G99.....EIM.↵
N460I0101↵
N47Z0GXZ↵
N480M6↵
N490M30.....EIM.↵
NEIM
```

Ao digitar RETURN o micro-computador apresentará o cardápio de OPÇÕES SECUNDÁRIAS que serão comentadas no item 2.4.6.

2.5.2-OPÇÃO (Principal) (2) GERAR SUBPROGRAMA NOVO.

Atua exatamente como a Opção Principal (1) diferenciando-se apenas por tratar-se de "geração" de sub-programas NC.

2.5.3-OPÇÃO (Principal) (3) CARREGAR PROGRAMA DO DISCO.

Essa opção é utilizada quando há necessidade de termos acesso a programas NC anteriormente armazenados em disco.

Após a opção ter sido exercida o sistema pergunta qual o numero do programa a ser procurado em disco, através da mensagem:

```
DIGITE NUMERO DO PROGRAMA
        <      >
```

Após a indicação do programa desejado o sistema apresentará durante o acesso ao disco, a mensagem:

```
PROGRAMA N xxxx
   SENDO  LIDO
      DO DISCO
```

Caso a leitura em disco for bem sucedida, o computador apresentará automaticamente o cardápio de **OPÇÕES SECUNDÁRIAS** que será visto no item 2.5.6

Caso o programa não seja localizado o sistema apresenta a mensagem:

```
NÃO EXISTE PROGRAMA N. xxxx
      NO DISCO !!!
```

2.5.4-OPÇÃO (Principal) (4) CARREGAR SUB-PROGRAMA DO DISCO.

Atuação idêntica à OPÇÃO (3) apenas referindo-se à sub-programas NC.

2.5.5 - OPÇÃO (Principal) (5) SIMULAÇÃO.

Antes de iniciar a simulação o sistema solicita a definição das áreas de segurança através da mensagem:

```
DEFINA A: (FAULT AREA) E A : (SAFE AREA)
G61 L1
```

Nesta ocasião normalmente define-se as dimensões da castanha e da peça, através do comando **G61** (vide Apêndice A).

Para o programa-exemplo relatado no item 2.4, teríamos a seguinte definição de castanha e peça em bruto:

```
DEFINA A: (FAULT AREA) E A: (SAFE AREA)
G61 L1 X115 Z-10 I135 K15
G61 L2 X95 Z-10 I115 K0
G61 L3
G61 L4
G61 L5 X0 Z0 I115 K8Z
G61 L6
G61 L7
G61 L8
```

Observa-se que dos 8 lugares geométricos disponíveis utilizou-se 2 para definir-se a castanha (L1 e L2) e um para definir-se a peça em bruto (L5).

O sistema apresenta então a seguinte mensagem:

PROGRAMAÇÃO EM DIÂMETRO OU RAIO <D/R>?

Feita a opção por diâmetro <D> ou raio <R>, há necessidade de indicar :

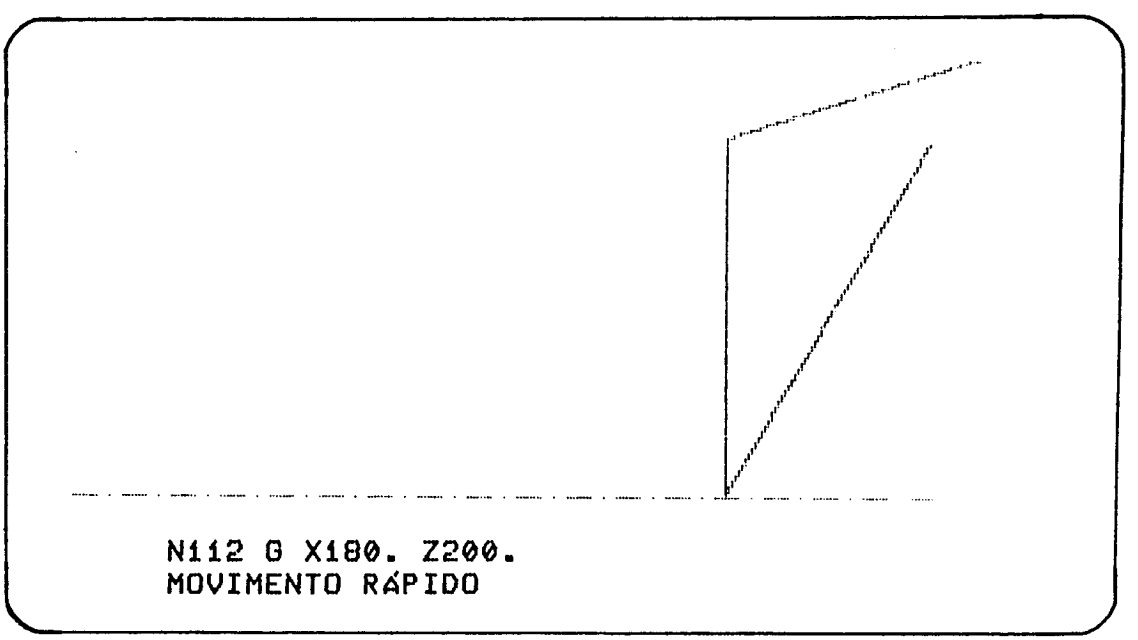
SIMULAÇÃO A PARTIR DO INICIO <S/N>?

Se optarmos por <N> haverá necessidade de indicação do número do bloco de início da simulação através da mensagem:

NUMERO DO BLOCO INICIAL ...N

Após essa indicação ocorre o início da simulação apresentando-se inicialmente as areas de segurança e a partir daí ocorrerá a simulação dinâmica da trajetória da ponta teórica da ferramenta.

No caso do programa exemplo teríamos algumas telas típicas tais como:

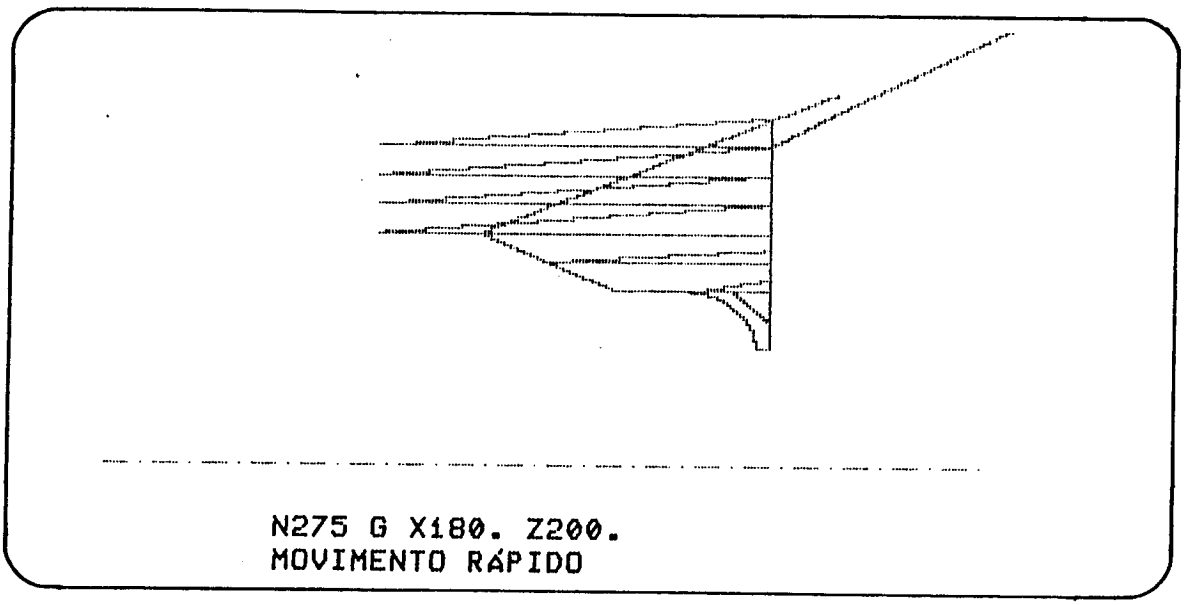


Observa-se então que o programa NC do usuário vai sendo interpretado e conforme ocorre a simulação da trajetória da ferramenta, tem-se disponível em português, a interpretação do bloco em

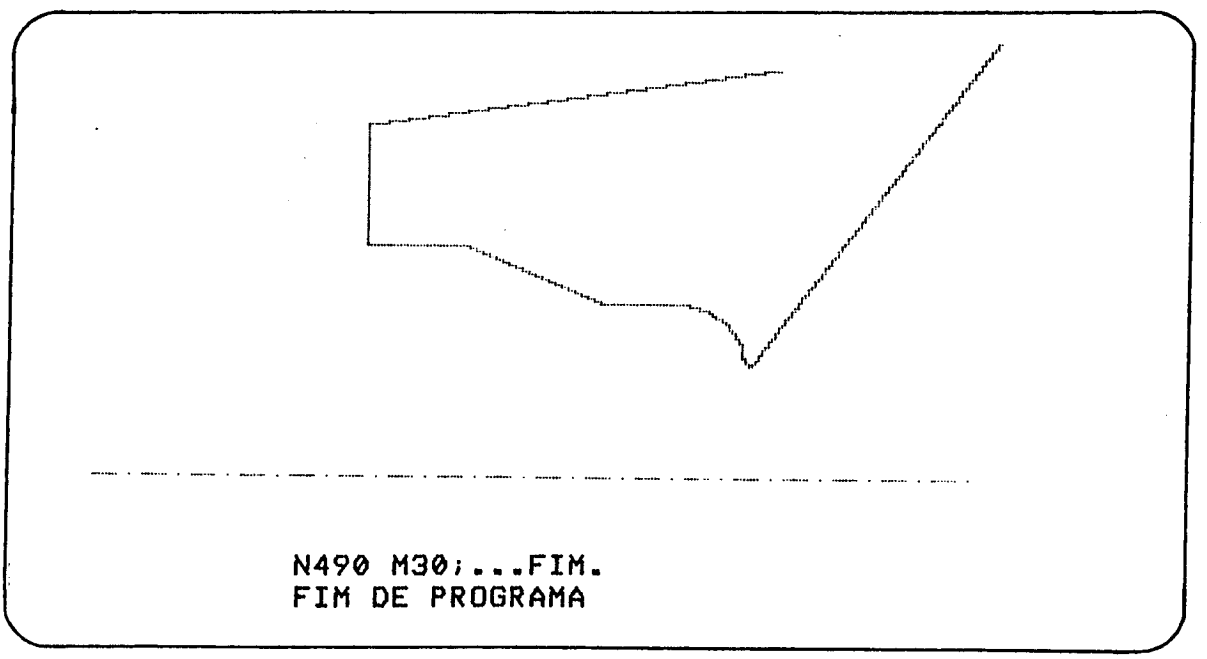
execução naquele momento.

Durante a simulação a velocidade da mesma pode ser alterada digitando-se as teclas "+" ou "-" respectivamente. O retorno ao EDITOR pode ser conseguido em qualquer instante da simulação digitando a tecla "R". O "congelamento" da imagem pode ser feito em qualquer instante digitando-se "C", e o consequente "descongelamento", digitando-se "D".

O percurso da ferramenta de desbaste (da peça-exemplo) é concluido no bloco de programação de número 275 e a imagem observada naquele instante é:



A ferramenta de acabamento tem seu percurso finalizado por:



2.5.6 - OPÇÃO (Principal) (6): OPÇÕES SECUNDÁRIAS.

é o meio de termos acesso as Opções Secundárias, que são:

```
****OPCOES SECUNDARIAS****

(1) ALTERAR BLOCO DO PROGRAMA
(2) INSERIR BLOCOS NO PROGRAMA
(3) CANCELAR BLOCO DO PROGRAMA
(4) IMPRESSAO DO PROGRAMA
(5) EXPOSICAO DO PROGRAMA (VIDEO)
(6) PERFURACAO DO PROGRAMA (FITA)
(7) ARMAZENAMENTO DO PROGRAMA (DISCO)
(8) SIMULACAO
(9) RETORNO AS OPCOES PRINCIPAIS
```

2.5.6.1 - OPÇÃO SECUNDÁRIA (1): ALTERAR BLOCO DO PROGRAMA.

Após esta opção tem-se a mensagem:

```
QUAL O BLOCO QUE DESEJA ALTERAR?
          (N      )
```

Feita a indicação do bloco é apresentado no vídeo o bloco e podemos inserir ou retirar caracteres principalmente através das teclas: " ← " (esquerda) e " → " (direita).

2.5.6.2 - OPÇÃO (Secundária) : (2): INSERIR BLOCOS NO PROGRAMA.

Após a apresentação da mensagem:

```
PARA ENCERRAR DIGITE :(FIM)
N
```

O usuário pode incluir blocos em qualquer ponto do programa, em qualquer ordem.

2.5.6.3 - OPÇÃO (Secundária): (3): CANCELAR BLOCO DO PROGRAMA.

Após a opção o micro-computador solicita o número do bloco a ser cancelado:

```
BLOCO A SER CANCELADO :(      )
```

Após a introdução do número do Bloco a ser cancela-

do o sistema expõe todo o bloco em questão e pede confirmação do cancelamento (S/N).

2.5.6.4 - OPÇÃO (Secundária): (4) : IMPRESSÃO DO PROGRAMA.

Caso o sistema possua uma impressora obtem-se a listagem do programa NC.

2.5.6.5 - OPÇÃO (Secundária): (5) : EXPOSIÇÃO DO PROGRAMA.

Páginas do progama são expostas no vídeo através desta opção. A mudança de páginas é feita digitando-se a barra de espaços do teclado. A última página do programa fica exposta por alguns segundos e após ocorre o retorno automático ao cardápio secundário.

2.5.6.6 - OPÇÃO (Secundária): (6): PERFURAÇÃO DO PROGRAMA.

O EDITOR foi desenvolvido de forma que se o Centro de Programação dispor de uma unidade de perfuração de fita de papel, o usuário poderá dispor dessa facilidade.

2.5.6.7 - OPÇÃO (Secundária): (7): ARMAZENAR PROGRAMA (DISCO).

Uma vez editado o programa é possível o armazenamento do mesmo em disco flexível. O usuário deverá organizar-se de maneira que não ocorra possibilidade de programas distintos receberem o mesmo número, pois será através do número de programa (definido no item 2.5.1) que o programa será acessado posteriormente. A possibilidade de arquivamento em disco flexível do acervo de programas NC do usuário é provavelmente o trunfo mais importante desse projeto, pois facilitará tremendamente o acesso à programas específicos e também otimizações do mesmo.

2.5.6.8 - OPÇÃO (Secundária) (8) : SIMULAÇÃO.

Essa opção proporcionará a simulação do programa exatamente da mesma forma relatada no item 2.5.5.

2.5.6.9 - OPÇÃO (Secundária) (9) : RETORNO ÀS OPÇÕES PRINCIPAIS.

Quando o usuário desejar o cardápio principal ele pode exercer essa opção (9) de maneira a dispor dos recursos relatados nos itens 2.5.1 a 2.5.5.

2.6 - Bibliografia:

[1] SIMON, Wilhelm. The Numerical Control of Machine Tools [Die Numerishe Steuerung von WWerkzeugmaschinen]. Londres, Edward Arnold, 1973, 544 p.

CAPÍTULO 3

METODOLOGIA UTILIZADA NO DESENVOLVIMENTO DO EDITOR/SIMULADOR

3.1-Generalidades.

O desenvolvimento do EDITOR/SIMULADOR foi considerado nesse projeto como "software" para gerar "software", pois o acervo de programas NC produzidos por esse Sistema de Apoio, pode ser considerado "software", apesar de sua aplicação bastante específica (controle de processo em tornos mecânicos).

As várias etapas do desenvolvimento desse projeto podem ser divididas em:

--Escolha da Linguagem

--Estudo e perfeita caracterização das regras de semântica e sintaxe para a programação manual de tornos NC.

--Levantamento das características particulares do armário de Controle em questão (MACH 3 L).

--Divisão do EDITOR/SIMULADOR em módulos com atribuições diversas.

--Desenvolvimento e teste de rotinas para algorítmos específicos.

--Testes e depurações das rotinas específicas.

--Testes e depurações do sistema integrando e harmonizando todas as rotinas específicas.

Outro aspecto prioritário na definição do EDITOR/SIMULADOR foi que ele deveria ser: conversacional e interativo, de forma que um programador NC experiente de posse do manual (de poucas páginas !), possa utilizá-lo sem precisar entrar no mérito do micro-computador.

3.2-Escolha_da_Linguagem.

Definidos os compromissos e objetivos já relatados (ítens 1.1; 2.2 e 2.3) foi necessário optar-se pela linguagem na qual o "software" seria desenvolvido. A escolha fundamentou-se principalmente nos recursos gráficos oferecidos pela linguagem a ser utilizada, relativamente à facilidade de seus comandos gráficos e velocidade de processamento.

Da triagem inicial dispensou-se logo de início o FORTRAN pela não disponibilidade de recursos gráficos, restando portanto: BASIC, PASCAL, e GRAFORTH. [1],[2],[3]

O desenvolvimento do SIMULADOR utilizando-se GRAFORTH teria sido bastante simples pois trata-se de linguagem essencialmente gráfica com recursos bastante poderosos em termos de computação gráfica. O GRAFORTH foi descartado por apresentar uma grande limitação: só opera com constantes e variáveis inteiras, fato esse que limitaria tremendamente as aplicações em Controle Numérico, onde devemos trabalhar com o Conjunto dos Números Reais sem nenhuma restrição.

A opção entre BASIC e PASCAL em favor do BASIC prendeu-se ao fato deste ser mais utilizado à nível de micro-computadores e portanto permitir o acesso de forma aberta à um universo mais amplo de usuários.

3.3-Caracterização dos Recursos do Armário de Controle.

Na etapa seguinte procedeu-se a um levantamento criterioso dos recursos de linguagem, e operacionais do armário de controle em questão.

Observou-se que apesar de existirem normas sobre o assunto, cada armário NC apresenta suas particularidades, principalmente devido à razões mercadológicas.

O controle MACH 3L utiliza 20 funções de programação que irão definir o fluxo de simulação do software desenvolvido. Essas funções aparecem na folha de programação (Vide Apêndice A).

Para as funções existentes foram desenvolvidas rotinas para simulação e emissão de mensagens interpretativas, sendo que para fins de simulação foram prioritárias as funções de controle de trajetória (funções G), que totalizam 35 tipos.

O sistema foi então desenvolvido por etapas. Explicando: foi inicialmente escrito a rotina para simulação da função G00, posteriormente incorporou-se a rotina de gerenciamento de G01, de maneira a cumprir todas as necessidades e harmonizar-se com G00.

Cuidou-se também para o aspecto de indicação (durante a simulação) de erros grosseiros de sintaxe, de maneira que na incidência de tais erros, eles podem ser eliminados ainda no Centro de Programação.

3.4-O EDITOR - Programa EDIT2.

O EDITOR para programas NC, foi desenvolvido com objetivo múltiplo de facilitar a geração de programas novos e alterar rapidamente programas já existentes, possibilitando também o arquivamento dos programas em disco flexível.

Com o intuito de facilitar a edição de programas o sistema dispensa o usuário da digitação em todo bloco de programação do caractere N (que existe em todo início de bloco!). Isto proporciona um ganho razoável de tempo, quando da digitação de programas.

A inserção de blocos intermediários à blocos já existentes, é automática e referenciada diretamente pelos próprios números dos blocos inseridos, de maneira que o usuário tenha total liberdade de inclusão de quantos blocos forem necessários e nos locais desejados.

A alteração dos blocos é feita utilizando-se todos os caracteres disponíveis no teclado e principalmente os controladores de cursor: ← (esquerda) e → (direita). Isto permite a alteração de forma simples, rápida e natural.

A impressão de programas NC é feita de maneira a facilitar a leitura e o acompanhamento do mesmo. Assim sendo, se um bloco for editado por exemplo na forma:

N110GX110.65Z3Z.52

Ele será exposto em vídeo, ou impresso com o EDITOR "separando" as funções de programação:

N110 G X110.65 Z37.52

Observa-se também que informações do tipo:G00 podem ser digitadas simplesmente: G , dispensando-se os zeros não significativos.

A exposição do programa no monitor de vídeo é feita em páginas de 10 linhas cada. A página exposta pode ser analizada quanto tempo for necessário e a exposição da página seguinte é feita digitando-se a barra de espaços, sendo que para a última página a exposição é feita por alguns segundos, após o que ocorre o retorno ao cardápio de opções secundárias.

A definição das áreas de segurança (L1 a L8) é feita no sistema "formatado", sendo que o EDITOR só permite a entrada dos caracteres na sequencia correta.

Devemos lembrar que no instante de gravação de programa ou sub-programa em disco flexível, o sistema deixa disponível apenas a última versão do programa ou sub-programa. O usuário deverá então cuidar de ter arquivado em outra unidade de disco a versão original do programa ou sub-programa.

O EDITOR foi desenvolvido de maneira a comportar programas NC com até 150 blocos. A limitação no número de caracteres por bloco é 255 para o EDITOR, mas devemos lembrar que o MACH 3L admite no máximo: 64 caracteres.

Na situação de entrada de dados o programa EDIT2 utiliza essencialmente o comando GET em detrimento do comando INPUT, isto devido o fato que durante a execução do comando INPUT, caso ocorra a entrada de dados incoerentes, tem-se a aborto dos trabalhos com mensagens indicativas de erro em inglês, que seriam bastante deselegantes para o usuário de NC. Foram então desenvolvidas rotinas especiais para entrada de dados de forma a contornar esse problema. [4]

No módulo de edição cuidou-se também para que a interação usuário x sistema fosse a mais natural possível, de maneira que em qualquer situação o usuário tenha uma "satisfação" por parte do sistema. Isto evita a monotonia de telas "mortas" onde o usuário não tem idéia do que está ocorrendo naquele momento. Por exemplo, após a instrução para leitura de programa de disco o sistema apresenta mensagens indicativas durante o processamento:

```
PROGRAMA xxx
SENDO LIDO
DO DISCO
```

Todas as vezes que o sistema EDITOR/SIMULADOR cumprir determinada tarefa e o controle voltar para o usuário haverá a emissão de uma mensagem sonora alertando o usuário.

3.5 - O SIMULADOR.

O simulador foi desenvolvido de modo a facilitar ao máximo a sua utilização. A definição da escala da simulação é atribuição do próprio sistema, e ele o faz de modo a utilizar pelo menos 80% da tela do monitor de vídeo. Esta definição é feita em função da definição das áreas de segurança, onde basicamente define-se as dimensões do sistema de fixação (lugares geométricos L1 a L4) e a própria peça

(lugares geométricos L5 a L8). Caso a escala definida pelo sistema não satisfaça o usuário, ele poderá alterá-la manipulando com lugares geométricos fictícios. Esse recurso poderá ser utilizado caso o usuário necessite "ampliar" um determinado detalhe da peça.

A simulação é atribuição de 3 módulos do sistema: SIMULA2, SUBROT1 e SUBROT2, responsáveis respectivamente pela simulação de programas principais de NC, sub-rotinas de 1º nível e sub-rotinas de 2º nível. A lógica e o conteúdo desses módulos é essencialmente a mesma, de maneira que comentaremos apenas o módulo SIMULA2.

A primeira rotina do módulo SIMULA2 consiste em "separar" corretamente as funções em cada bloco de programa NC, pois o módulo EDIT2 incorporou simplesmente os blocos como um todo, através de variáveis tipo cordão ("string").

Tendo sido separadas as funções de cada bloco de programa NC, tem início o processamento das mesmas em função de rotinas diretivas. Devemos ressaltar que as funções prioritárias para definição da trajetória da ferramenta são as funções G, resultando como consequência que essencialmente os módulos de simulação dedicam-se à tais funções.

O APPLESOFT oferece basicamente duas maneiras de cômputo gráfico de alta resolução: através de tabelas de forma e através do comando HPLOT x,y . Na primeira situação define-se antecipadamente (através de conjunto de vetores) um lugar geométrico e posteriormente esse lugar geométrico pode ser apresentado como um todo na tela através do comando DRAW. Essa opção não foi utilizada porque a simulação da trajetória da ponta da ferramenta deve ocorrer progressivamente no vídeo, dando tempo para que o usuário tenha percepção de "onde para onde" , ocorre o movimento. Foi então utilizado essencialmente o comando HPLOT x,y, onde x e y são as coordenadas de um ponto no plano da tela, que deve ser "aceso" naquele instante, com cor determinada antecipadamente por HCOLOR n.

Utilizando-se então essencialmente de Geometria Euclidiana, Geometria Analítica, Trigonometria e cômputo, desenvolveu-se rotinas específicas para traçado de trajetória. Aproveitando-se do fato do sistema dispor de recursos poli-cromáticos, fez-se distinção entre as diversas situações de simulação através de cores distintas.

A utilização do comando HPLOT, permitiu "acender-se " pontos em sequência, com tempo de retardo controlados no pograma pela variável XU. Assim, para simulação de G00 (movimento rápido), XU tem um valor de retardo baixo e para G01 (usinagem), XU tem um valor de retardo alto, de tal forma a sensibilizar o usuário em termos de velocidade de simulações distintas.

A definição das velocidades foi experimental, através de um número razoável de opiniões. De qualquer maneira o usuário pode acelerar ou desacelerar o processo de simulação durante o processo, através das teclas: "+" e "-".

Outro ponto que acreditamos que deva ser ressaltado é a possibilidade de retorno (durante a simulação) ao EDITOR, isto porque muitas vezes o usuário percebe a necessidade de alteração do programa NC, e deseja fazê-lo antes de de analisar o restante da simulação. Esse retorno foi viabilizado pela inclusão de comando GET, em ponto estratégico nos módulos de simulação. Após o comando GET,"pergunta-se" se a tecla "R" foi acionada pelo usuário, sendo que na situação afirmativa ocorrerá a "chamada" do módulo EDITOR.

No caso de interpolações circulares o módulo de si-

mulação faz uma verificação da "consistência" do bloco de programação, de maneira que caso ocorra incoerência nos valores o sistema emite a mensagem: "VALORES INCOERENTES NA INTERPOLAÇÃO CIRCULAR" e retorna ao EDITOR onde o programador NC pode proceder a correção do bloco de interpolação circular e reiniciar a simulação no bloco que desejar.

O sistema permite simular até sub-rotinas de 2º nível, isto é, simula sub-rotina que foi "chamada" por sub-rotina (de 1º nível). Acreditamos que esse é o nível máximo de encadeamento utilizado por programadores NC, fato esse inclusive observado nos exemplos de programação no manual do fabricante.[5]

3.6-Testes e Otimização do Sistema.

É de nosso entendimento que não existe "software" perfeito. Normalmente ocorre o seu lançamento e posteriormente (principalmente devido a realimentação de usuários) surgem versões evoluidas do mesmo.

"Software" tem vida, isto é indubitável, atinge o grau máximo de otimização e posteriormente e inevitavelmente acabará envelhecendo e tornar-se-á obsoleto.

O sistema foi desenvolvido em módulos, sendo que cada módulo é composto de rotinas específicas. Desenvolvida uma determinada rotina, efetuou-se todos os testes possíveis com a mesma, de maneira a otimizá-la para um desempenho satisfatório. A rotina testada era então incorporada ao módulo e atacava-se a seguir uma nova rotina até fechar o módulo. O módulo era então exaustivamente testado de forma a não ocorrer incompatibilidades entre as diversas rotinas.

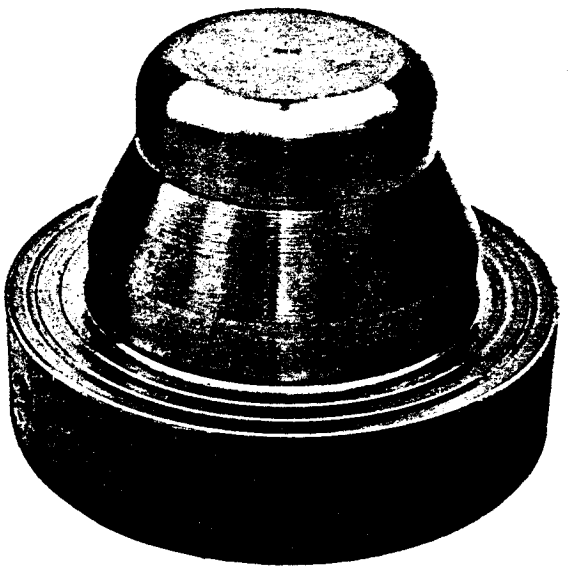
Com a versão inicial do sistema concluida efetuou-se testes demorados e detalhados do mesmo, com as correspondentes alterações para otimizações até conseguir-se um desempenho satisfatório do sistema.

Na fase de testes utilizou-se sobremaneira os exemplos de programação NC do manual do fabricante, e exercícios de programação aplicados à nossos alunos (1º semestre de 85) da disciplina EM-923 - Controle Numérico de Máquinas-Ferramenta.

A utilização na fase de testes de exemplos de programação de fontes diversas foi importante, pois cada programador utilizando as mesmas regras de semântica e sintaxe, impõe critérios e "estilo" próprio de programação, sendo que o sistema deve ser o mais universal possível.

A conclusão do desenvolvimento do EDITOR/SIMULADOR coincidiu com o recebimento no Setor de Engenharia de Fabricação do torno COSMOS, com execução de estréia (por parte do pessoal do SEF) da peça-exemplo indicada no item 2-4.

A execução da peça foi feita da segundo programação relatada no Capítulo 2 excetuando-se o fato (por carência de ferramental) de que a primeira e a segunda operações lá relatadas foram executadas pela mesma ferramenta.



Fotografia da peça-exemplo

Na fase de definição de diretrizes do EDITOR/SIMULADOR foram adotados alguns critérios que direcionam a utilização do mesmo. Acreditamos que esses critérios não tenham deixado o sistema rígido, dificultando a sua utilização. Por exemplo: para utilizar o sistema o usuário precisa obrigatoriamente definir as áreas de segurança, porque através dela o sistema definirá a escala para otimização do uso da tela durante a simulação. Um usuário que não irá utilizar o EDITOR/SIMULADOR pode (**mas_não_deve!**) empregar a máquina **sem** definir as áreas de segurança.

O EDITOR/SIMULADOR foi escrito de maneira que o usuário **não** necessite de nenhum conhecimento de programação de computadores, sendo requerido apenas evidentemente conhecimentos de programação NC. O relacionamento usuário x sistema é relativamente "amigável", através de sugestões e mensagens em português.

3.2-Bibliografia

[1] POOLE, Lon & MCNIFF, Martin & COOK, Steven. APPLE II: Guia do Usuário [Apple II, User's Guide].São Paulo, McGraw-Hill, 1984, 375 p.

[2] LUEHERMANN, Arthur & PECKHAM, Herbert. APPLE PASCAL: A Hands-On Approach. New York, McGraw-Hill, 1981, 431 p.

[3] LUTUS, Paul. GraFORTH. Portland, INSOFT, 1982.

[4] WYATT, Allen. BASIC Tricks for the Apple. Indianapolis, Howard W. Sams & Co, 1983, 139 p.

[5] ROMI, Indústrias. Comando Numérico MACH 3L - CNC - Manual de Programação. Santa Bárbara d'Oeste. 50 p.

CAPÍTULO 4

CONCLUSÕES E PROPOSTAS

4.1-Conclusões.

O presente trabalho constituiu-se no desenvolvimento de um produto de utilidade restrita aos programadores de NC e apoio à cursos de programação de tornos NC. A sua existência é palpável e bem definida. As aspirações iniciais em termos de performance do sistema foram plenamente satisfeitas. Durante o desenvolvimento do projeto foram adicionados recursos não previstos anteriormente (por exemplo o retorno ao EDITOR durante a simulação).

Após a iumplantação do Centro de Torneamento no Laboratório de Usinagem do Setor de Engenharia de Fabricação foi possível testar exaustivamente o sistema de maneira que algumas interpretações errôneas do catálogo do fabricante foram dirimidas (por exemplo a possibilidade da interpolação circular através do parâmetro R).

O sistema está implantado e em funcionamento no referido local onde tem sido utilizado e efetivado a validade da idéia da eliminação da fita de papel perfurada como meio de comunicação entre o Centro de Programação e o armário de Controle Numérico.

O EDITOR/SIMULADOR, também já foi aplicado em curso de Programação de Tornos NC, onde agilizou tremendamente o aprendizado através de exercícios de programação utilizando o sistema.

Acreditamos portanto que os objetivos iniciais (apoio à programadores e ao ensino de programação NC), foram plenamente alcançados.

4.2-Propostas.

Todo trabalho científico (felizmente) abre novos horizontes e apresenta novos desafios à Humanidade.

No caso específico, tem-se como pontos possíveis de serem desenvolvidos:

--um sistema simulador que proporcione a transformação da peça **durante** a usinagem e não, simplesmente o rastreamento da ponta da ferramenta

--um sistema simulador que proporcione o efeito de "ZOOM" em detalhes das peças.

--linguagem semi-automática com macro-instruções de tal forma que o usuário defina apenas o percurso da ferramenta de acabamento, com os passes de desbaste, determinados pelo próprio sistema.

--um simulador para máquinas com 3 eixos (fresadoras, centros de usinagem, etc).

--uma linguagem de programação NC de alto-nível utilizando palavras-chave do idioma português.

Como podemos observar o universo de propostas é bastante amplo, sendo que para efetivação de algumas delas, há necessidade de apoio de "hardware" mais poderoso que o que foi utilizado no presente desenvolvimento.

APÊNDICE A

PROGRAMAÇÃO MANUAL DE TORNOS DE CONTROLE NUMÉRICO

A.1-Introdução.

Apesar de existirem normas de programação manual para Controle Numérico, cada armário de Controle apresenta suas peculiaridades.

Relataremos aqui os recursos particulares de programação do torno COSMOS equipado com armário de Controle MACH 3L. Trata-se de Controle que foi liberado pela SEI para nacionalização.

A.2-O Significado dos Caracteres utilizados no Mach 3L.

Uma programação "standard" refere-se ao relacionamento internacionalmente aceito entre letras, números e símbolos de um lado e funções das máquinas NC do outro lado. Deve-se ressaltar que as discussões da ISO sobre o tema ainda não terminaram, e suas normas tem ainda caráter provisório.

"A"

Função: Ângulo de início da rosca
Formato: A3.3
Unidade: graus

"B"

Função: Ângulo de alimentação para roscamento
Formato: B3.3
Unidade: graus

"D"

Função: Tempo de permanência
Formato: D2.2
Unidade: Segundos

"F"

Função: Velocidade de avanço ou avanço por volta
Formatos: F2.4 ...polegada por volta
F3.3 ...milímetro por volta
F3.1 ...polegada por minuto
F4 ...milímetro por minuto

"G"

Função: Preparatória
Formato: G2

"H"

Função: Desvio incondicional
Formato: H4

"I"

Função: parâmetro para interpolação circular e roscamento
Formato: I +-2.5 ...polegada (medida no diâmetro)
I +-3.4 ...milímetro (medida no diâmetro)

"K"

Função:parâmetro para interpolação circular e roscamento, paralelo à Z
Formato: K +-2.5 ...polegada
K +-2.5 ...milímetro

"L"

Função: número de repetições na "chamada" de sub-rotina
Formato: L2

"M"

Função: miscelânea
Formato: M2

"N"

Função: número de bloco de programa
Formato: N4

"P"

Função: número de sub-rotina
Formato: P2

"R"

Função: raio de interpolação circular
Formato: R2.5 ...polegada
R3.4 ...milímetro

"S"

Função: Velocidade de corte ou rotação do eixo árvore
Formato: S4 ...rotação por minuto
S4.1 ...pés por minuto
S3.1 ...metros por minuto

"T"

Função: Especificação de ferramenta
Formato: T4
Composição: Taabb, onde:
aa = corresponde ao número da ferramenta
bb = corresponde ao número dos corretor de ferramenta

"U"

Função: parâmetro auxiliar em ciclo fixos de programação
Formatos: U+-2.5 ...polegada
U+-3.4 ...milímetro

"W"

Função: Incremento na direção X em ciclos fixos de programação
Formatos: W+-2.5 ...polegada
W+-3.4 ...milímetro

"X"

Função: coordenada referente a diâmetro
Formatos: X+-2.5 ...polegada
X+-3.4 ...milímetro

"Z"

Função: coordenada referente a comprimento
Formatos: Z+-2.5 ...polegada
Z+-3.4 ...milímetro

";"

Função: início de comentário

"."

Função: ponto decimal em grandezas fracionárias e separador de palavras em comentários

"/"

Função: supressor de sentença

"#" (line-feed)

Função: separador de blocos

"0" "1" "2" "3" 4" "5" "6" "7" "8" "9"

Função: dígitos numéricos do sistema decimal

"-"

Função: indicativo de grandeza negativa
Observação: As grandezas sem sinal, são interpretadas como positivas

A.3-Estrutura e sequencia das palavras de programação.

Um programa NC consiste em um conjunto de blocos (sentenças de programação). As palavras (funções de programação) são normalmente dispostas no bloco na sequencia:

1.Palavra para número de sequência	N----
2.Palavra para condição de percurso	G--
3.Palavra/s de coordenada/s	X,Z,U,W,A,B,I,K,R, etc
4.Palavra para avanço	F
5.Palavra para velocidade de corte	S----
6.Palavra para velocidade de corte	T----
7.Palavra para função miscelânea	M--

Comentários são habitualmente colocados no término do bloco, e são assim interpretados por serem precedidos do caractere ";". Não são permitidos espaços entre as palavras de um comentário, utiliza-se então o "." como separador.Exemplo:

N3450G99;DESBASTE.EXTERNO#

Existem palavras com atuação modal, isto é: elas

têm validade até o aparecimento de uma outra função que a cancele. Tais palavras não requerem a sua escrita em todo bloco de programação.

A.4-Função N : Número sequencial de Blocos.

É a primeira informação que deve constar na sentença de programação. Podemos programar de N0000 até N9999. Normalmente especifica-se uma numeração crescente de dez em dez, o que irá possibilitara inclusão posterior de até 9 blocos intermediários sem necessidade de alteração da numeração dos restantes.

No MACH 3L, a inclusão da palavra N é opcional, mas devemos lembrar que programas com numeração de blocos são mais fáceis de serem operacionalizados.

A.5-Funções Preparatórias G.

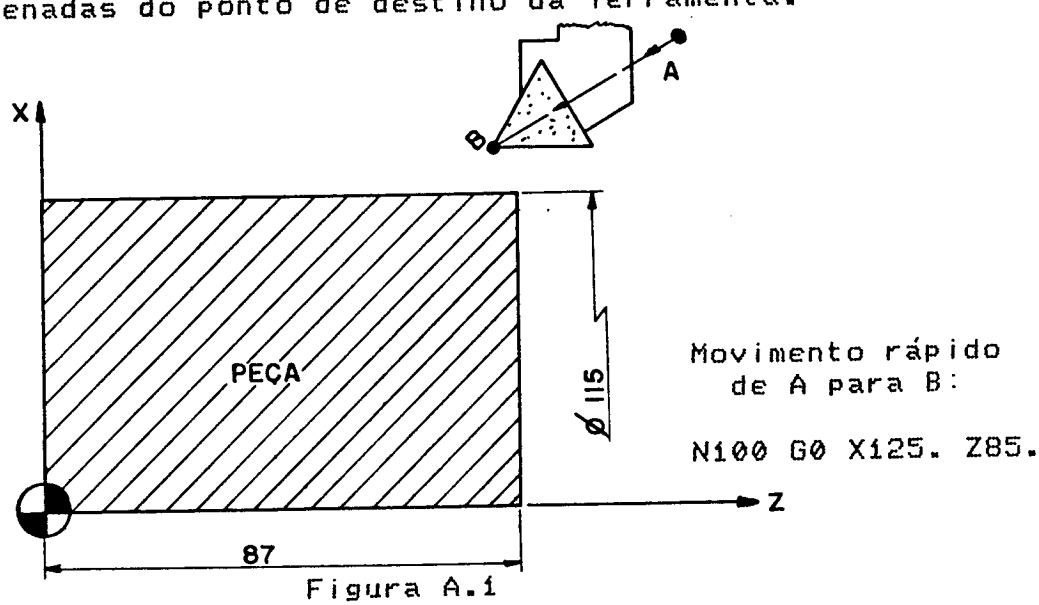
As funções G foram criadas para definir as condições de percurso. A seguir relacionaremos todas as funções G utilizadas por esse equipamento.

"G00"

MOVIMENTO RÁPIDO

Utilizado para aproximações e afastamentos da ferramenta. A torre porta-ferramentas movimenta-se em linha reta na velocidade máxima possível. (10 metros por minuto)

O movimento rápido é obtido programando-se a função G00 e as coordenadas do ponto de destino da ferramenta.



Movimento rápido de A para B:

N100 G0 X125. Z85.

Figura A.1

A função G00 tem atuação modal e cancela as funções: G01, G02, G03, e G73.

"G01"

USINAGEM

O movimento de usinagem é obtido programando-se G01 seguido das coordenadas do ponto de destino da ferramenta e do avanço.

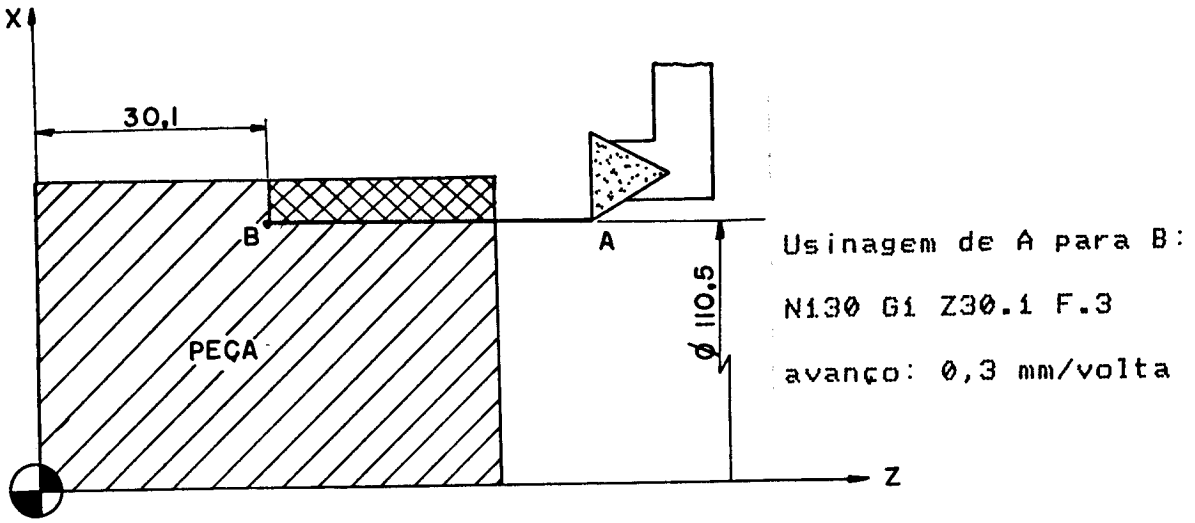


Figura A.2

A função G01 tem atuação modal e é cancelada por: G00, G02, G03, e G73.

"G02"

INTERPOLAÇÃO CIRCULAR HORÁRIA

A manufatura de arcos circulares horários é conseguida, programando-se G02 seguido das coordenadas do ponto de destino, as coordenadas do centro do arco (nos parâmetros I e K) e da especificação do avanço na função F.

Uma segunda possibilidade existe em especificar-se diretamente o raio do arco na função R:

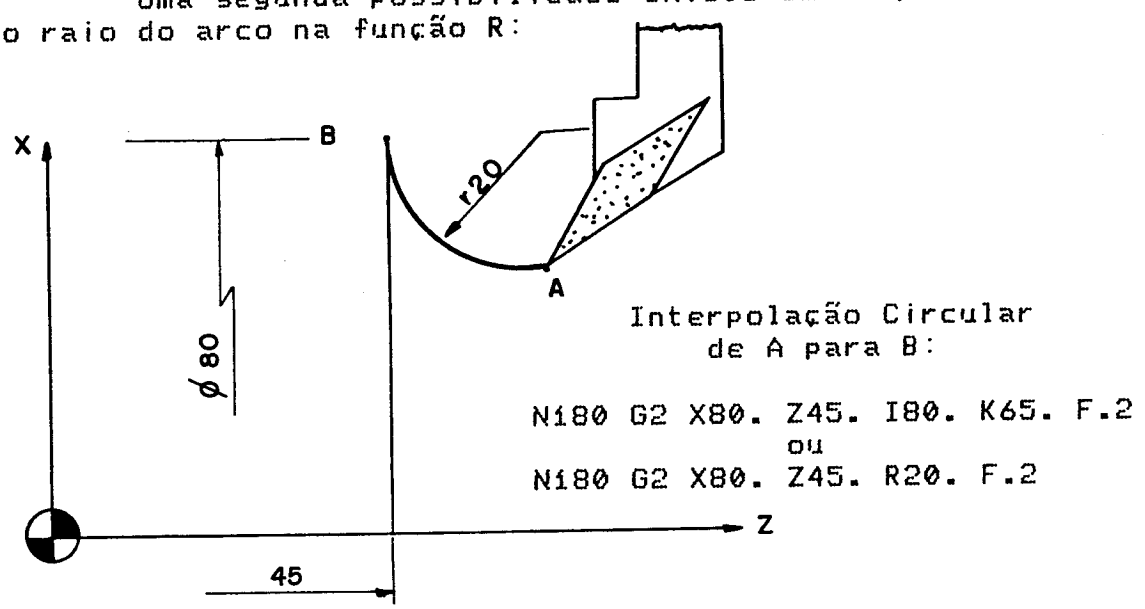


Figura A.3

A função G02 tem atuação modal e é cancelada por: G00, G01, G03 e G73. A definição de sentido horário deve ser feita olhando-se a máquina de cima.

"G03"

MOVIMENTO CIRCULAR ANTI-HORÁRIO

Programação análoga à G02, excetuando-se o sentido

de manufatura do arco. G03 tem atuação modal e é cancelada por: G00, G01, G02 e G73.

"G04"

TEMPO DE PERMANENCIA (DWELL)

Juntamente com a função D (tempo em segundos) inibe os avanços, permanecendo o eixo arvóre em rotação. Durante a permanencia, na página "status" do controle tem-se um cronometro decrescente.

"G20"

"PROGRAMAÇÃO EM DIÂMETRO

O diâmetro é associado à coordenada X. A função G20 tem atuação modal e é cancelada por G21.

"G21"

PROGRAMAÇÃO EM RAIO

O raio é associado à coordenada X. A função G21 tem atuação modal e é cancelada por G20.

"G30"

CANCELAMENTO DE IMAGEM ESPELHO

Cancela imagem espelho, a qual pode ser otida por G31 ou G32.

"G31"

IMAGEM ESPELHO RELATIVO AO EIXO X

Todo valor programado no eixo X sob efeito desta função será multiplicado por -1. Essa função é pouco utizada em tornos

"G32"

IMAGEM ESPELHO RELATIVO AO EIXO Z.

Todo valor programado no eixo Z sob efeito desta função será multiplicado por -1.

"G33"

CICLO DE ROSCAMENTO

Para utilizar-se esta função requer-se:
G33 X Z K [A] [I]

onde:
X...profundidade por passada (diâmetro-absoluta)
Z...posição final do comprimento da rosca (absoluta)
K...passo da rosca
[A]...ângulo de entrada da rosca
[I]...conicidade incremental no eixo X (diâmetro)

A função G33 tem atuação modal e se cancela por G80. Na programação cada bloco contendo uma função X produz um movimento rápido na profundidade X, em seguida a ferramenta move-se em Z executando um passe da rosca, seguido imediatamente em avanço rápido o retrocesso no eixo X, voltando ao diâmetro de partida e finalmente, em avanço rápido, a ferramenta retorna em Z na posição inicial, e assim sucessivamente para cada X programado.

"G40"

CANCELA COMPENSAÇÃO DO RAIO DA PASTILHA

O cancelamento da compensação automática do raio da pastilha é obtido através da programação de G40. A efetivação desse cancelamento é feito em um bloco subsequente com programação de G01.

"G41"

COMPENSAÇÃO DO RAIO DA PASTILHA (esquerda)

A função G41 seleciona o valor da compensação do raio da ponta da ferramenta, o qual é considerado localizado à esquerda da trajetória gerada pela ferramenta. Após a programação de G41 é obrigatório a programação de G01 ou G73.
Observação: durante a validade de G41 não é permitido a utilização de G00.

"G42"

COMPENSAÇÃO DO RAIO DA PASTILHA (direita)

Função idêntica à G41, mas com compensação automática do raio da pastilha à direita da trajetória da ferramenta.

"G46"

INIBIÇÃO DA VELOCIDADE DE CORTE CONSTANTE

A variação da rotação do eixo-árvore pode ser inibida através da função G46. Esse recurso normalmente é utilizado quando temos grandes variações de diâmetro em aproximações e afastamentos.

"G47"

ATIVAÇÃO DA VELOCIDADE DE CORTE CONSTANTE

Essa função cancela a inibição da velocidade de corte proporcionada por G46. Após a especificação de G47 temos efetivada a velocidade de corte especificada por G96.

"G53"

CANCELAMENTO DAS TRANSLAÇÕES DO ZERO-PEÇA

Essa função cancela as translações do zero-peça, especificadas por G54 e G55.

"G54"

PRIMEIRA TRANSLAÇÃO DO ZERO-PEÇA

Habitualmente utilizada em peças com duas fases de fabricação. Essa função possibita termos num mesmo programa dois "zeros-peça". O valor numérico da translação é introduzido no controle através da página "Tool Data File", com o título "chuck Offsets".

"G55"

SEGUNDA TRANSLAÇÃO DO ZERO-PEÇA

Atuação idêntica à função anterior para definição de uma segunda translação do "zero-peça".

"G60"

CANCELAMENTO DE ÁREAS DE SEGURANÇA

Áreas de segurança que foram anteriormente especificadas por G61, podem ser canceladas através de G60. Assim o cancelamento da terceira área de segurança, pode ser feito por:

G60 L3

Para cancelar-se todas áreas de segurança, especifica-se apenas G60.

"G61"

DEFINIÇÃO DE ÁREAS DE SEGURANÇA

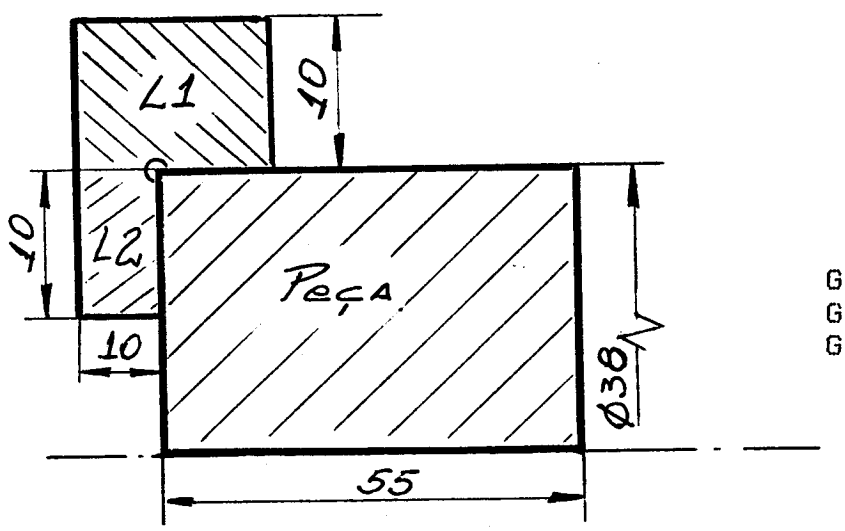
Podemos definir oito áreas de segurança. As quatro primeiras, são normalmente utilizadas para definir a placa, as castanhas e o contra-ponto. As quatro últimas, geralmente são empregadas para definir a peça propriamente.

Forma geral de programação:

G61 Ln Xx Zz Ii Kk

onde:

n...número da área (1 a 8)
x...diâmetro interno da área
z...linha de "encosto"
k...comprimento



G61 L1 X18. Z-10. I38. K0.
G61 L2 X38. Z-10. I58. K15.
G61 L5 X0 Z0 I38. K55.

Figura A.4

"G70"

PROGRAMAÇÃO EM POLEGADAS

A especificação de G70 proporciona a utilização de dimensões lineares em polegadas, da velocidade de corte em pés por minuto, do avanço em polegada por volta e da velocidade de avanço em polegada por minuto.

"G71"

PROGRAMAÇÃO EM UNIDADES MÉTRICAS

A especificação de G71 proporciona a utilização de dimensões lineares em milímetros, da velocidade de corte em metro por minuto, do avanço em milímetro por volta e da velocidade de avanço em milímetro por minuto.

"G73"

INTERPOLAÇÃO LINEAR PRECISA

Função de aplicação semelhante a G01 (interpolação linear), mas com acelerações e desacelerações proporcionando pontos de

transição mais precisos.

"G74"

CICLO-FIXO DE TORNEAMENTO E FURAÇÃO COM DESCARGA

Sucessivos passes de desbaste em torneamento, ou sucessivas entradas de broca, podem ser especificadas em um bloco apenas através de G74.

Forma geral:

G74 [X] Z [W] [I] [U]

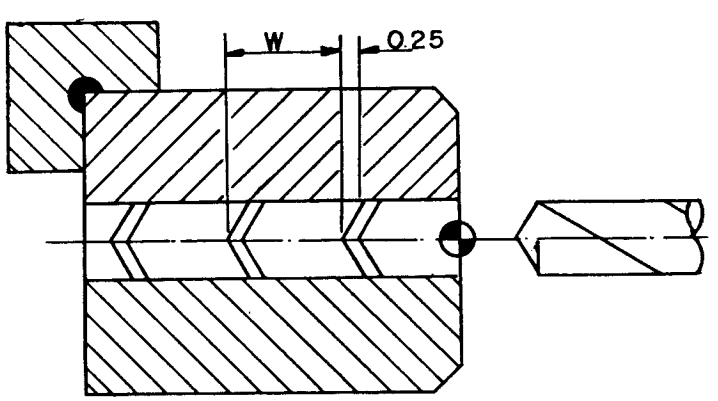
onde:

X... diâmetro final - Absoluto
Z... posição final - Absoluto
W... distância para quebra cavaco - Incremental
I... incremento por passada em diâmetro-Incremental
U... recuo para retorno - Incremental

Observações:

a) Não sendo programada a função X, a função I será ignorada e somente um passe será dado no X de início (anterior ao G74).
b) Na ausência da função W, tem-se a usinagem na direção Z até o valor programado.
c) Se houver a função X e não ocorrer a programação da função I, serão dados 2 passes: um no X inicial (anterior ao G74) e outro no X final (programado junto com G74)
d) Programando-se a função U o movimento de retorno será feito também na direção X no sentido de afastamento da peça com o valor igual à I.

```
G X0. Z6.
G74 Z-50. W20. F.15
```



Avanço [mm/volta]	W
Sai rápido	W
Posiciona rápido	W-.25
Avanço	W+.25
Sai rápido	2 x W
Posiciona rápido	2 x W - .25

Continua até Z ser alcançada e retorna ao ponto inicial

Figura A.5

G X70. Z66.
G74 X46. Z27. I6. U F.3

X diâmetro final

Z posição final

I incremento em X

U recuo para retorno

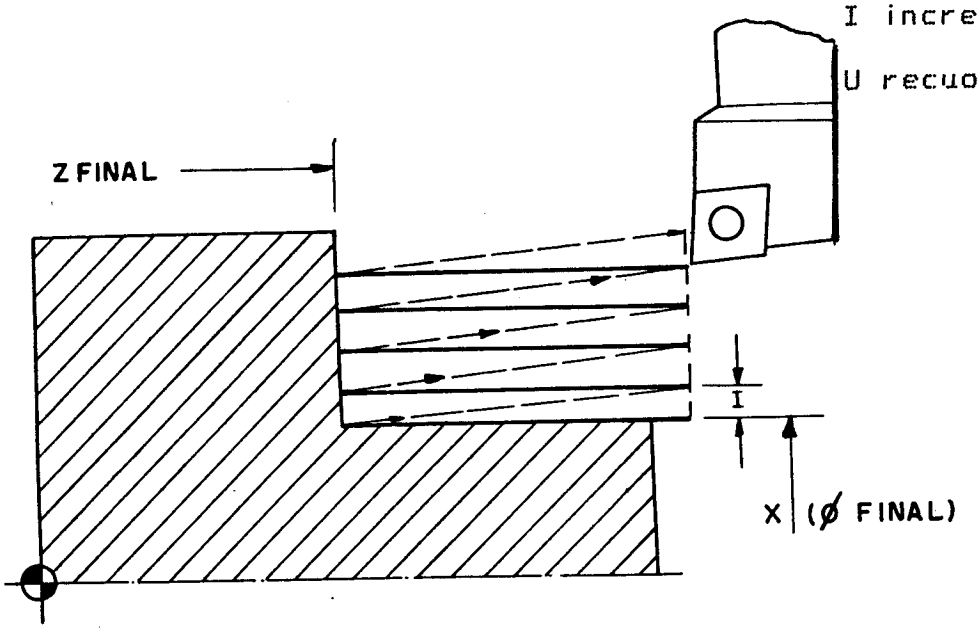


Figura A.6

G X40. Z66.
G74 X64. Z27. I6. U F.3

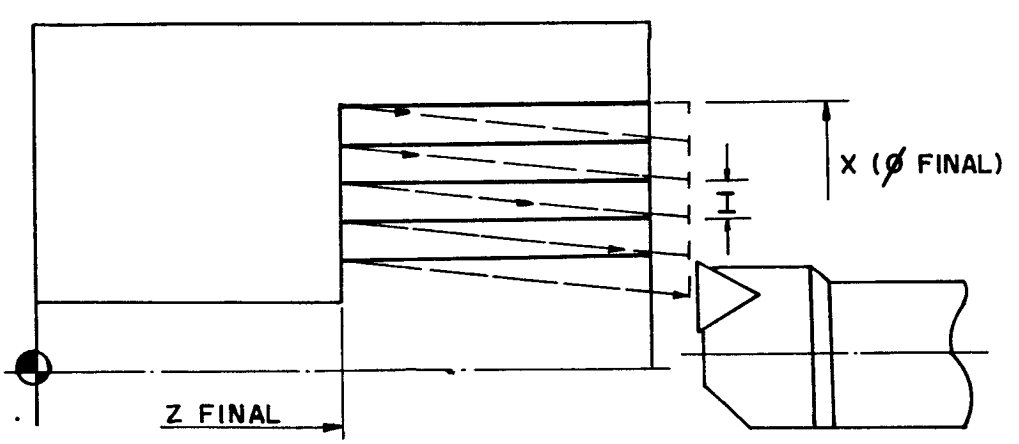


Figura A.7

"G75"

CICLO-FIXO DE FACEAMENTO E DE ABERTURA DE CANAIS

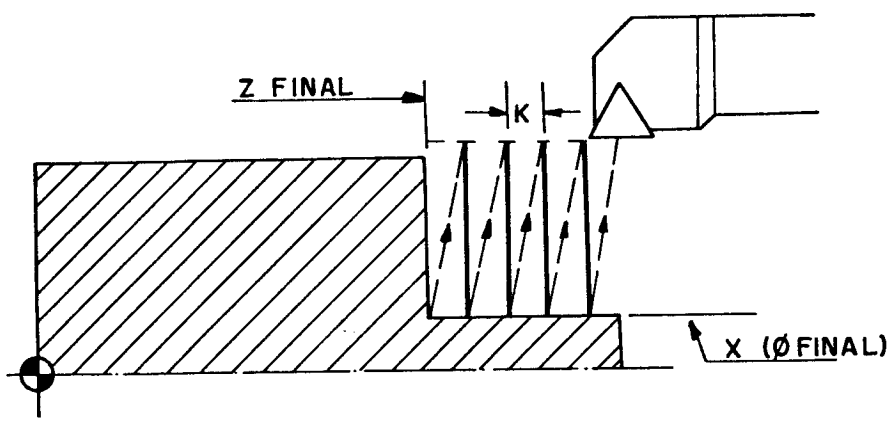
Similar ao ciclo anterior, mas com movimentos principais paralelos a X

Forma geral:

G75 X [Z] [W] [K] [U]

onde:

X... diâmetro final - Absoluto
Z... posição final - Absoluto
W... distância para quebra de cavaco - Incremental (diâmetro)
K... incremento para passadas em Z - Incremental
U... recuo para retorno de posição - Incremental



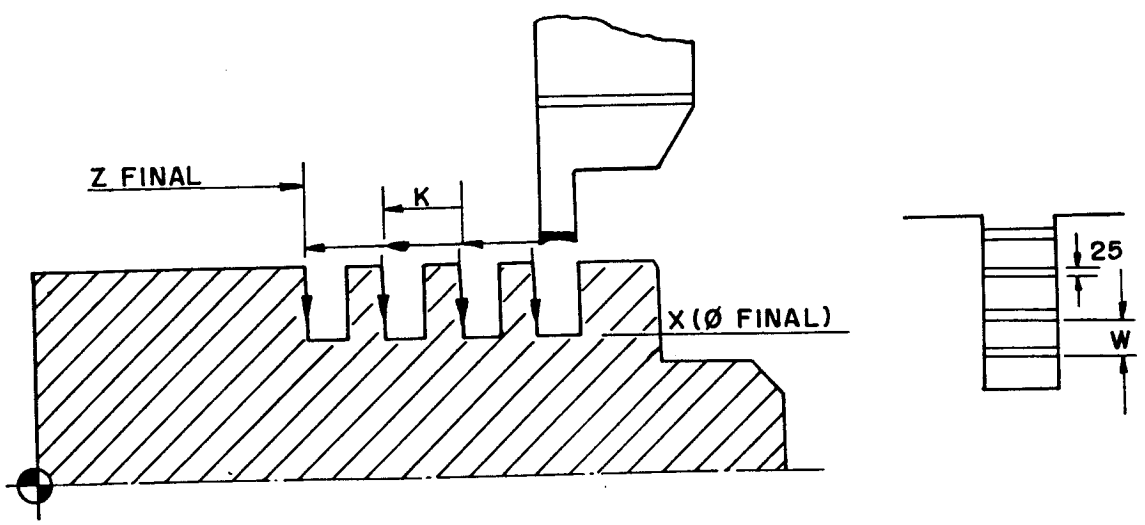


Figura A.8

"G76"

CICLO-FIXO DE ROSQUEAMENTO

Através dessa função é possível abrir roscas paralelas à X ou à Z ou mesmo roscas cônicas.

Forma geral:

G76 X Z K U W [A] [B] [I]

onde:

X... profundidade final do roscamento - diâmetro - absoluto
Z... posição final do comprimento da rosca - Absoluto
K... passo da rosca
U... altura total do filete - Diâmetro - Incremental
W... profundidade por passada - Diâmetro - Incremental
[A] abertura angular entre as entradas da rosca - graus
[B] ângulo de alimentação para sistema composto
[I] conicidade incremental no eixo X - diâmetro

Observações:

a) O número de passadas será definido através das funções X e W, sendo que o último incremento será subdividido em 4 passadas nos valores: W/2, W/4, W/8 e W/8.
b) Após a programação de G76 há necessidade de cancelamento da mesma através da programação de G80.
c) Lembrar que a posição de início (antes de programar-se G76) deve ser além do diâmetro externo da rosca (ou menor no caso de rosca interna)

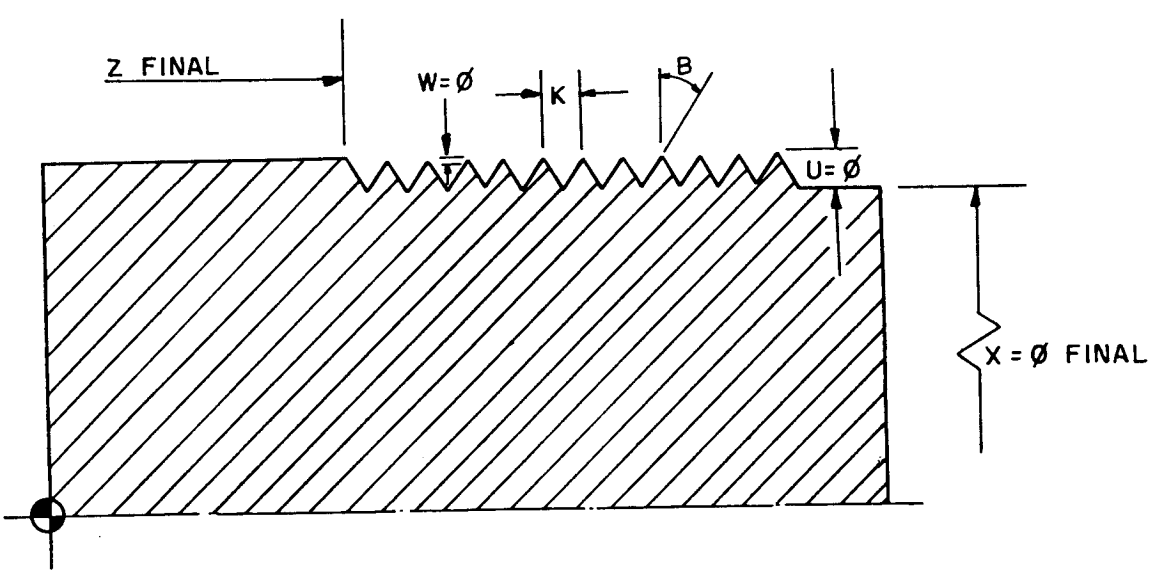


Figura A.9

"G80"

CANCELAMENTO DO CICLO FIXO DE ROSQUEAMENTO

É necessário a programação (em bloco à parte) de G80, para termos o cancelamento de G76.

"G90"

PROGRAMAÇÃO EM COORDENADAS ABSOLUTAS

Todas coordenadas especificadas após essa função são referenciadas ao zero peça. (desde que não seja programado: G91)

"G91"

PROGRAMAÇÃO EM COORDENADAS INCREMENTAIS

Todas coordenadas especificadas após essa função são referenciadas ao ponto de início do movimento. (desde que não seja programado:: G90)

"G92"

DEFINIÇÃO DO ZERO-PEÇA E LIMITE DE ROTAÇÃO

A função G92 é utilizada para definir o "zero-peça. A utilização pode ser vista no exemplo abaixo. Normalmente essa função é especificada após o indexamento da ferramenta.

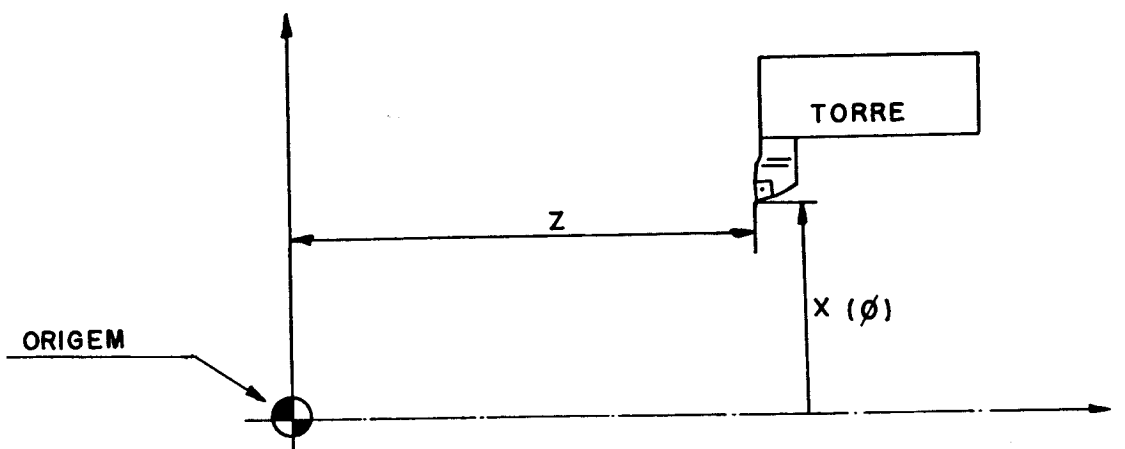


Figura A.10

Podemos limitar a rotação de trabalho, programando-se G92 seguido da função S.

"G94"

PROGRAMAÇÃO EM VELOCIDADE DE AVANÇO

Programação da função F em velocidade de avanço. [polegadas/min] ou [mm/min]

"G95"

PROGRAMAÇÃO EM AVANÇO

Programação da função F em avanço.[polegadas/volta] ou [mm/volta]

"G96"

PROGRAMAÇÃO EM VELOCIDADE DE CORTE CONSTANTE

Programação em velocidade de corte constante em [pés/min] ou [m/min]. A definição da velocidade de corte é feita em bloco subsequente ao que aparece G96, utilizando a palavra S.

"G97"

PROGRAMAÇÃO EM ROTAÇÃO CONSTANTE

Programação em rpm constante. A definição da rotação deve ser feita no bloco após ao de G97, através da função S.

"G99""

PROGRAMAÇÃO REFERENTE AO ZERO-MÁQUINA

Todas as coordenadas que aparecem após essa função são referenciadas ao zero-máquina. Essa função é cancelada por: G92 Xxx Zzz.

A.6-Funções_Miscelâneas_M.

As funções miscelâneas (ou auxiliares) normalmente são incluidas no final do bloco de programação. Aqui também podem ocorrer divergencias entre Unidades de Controle de procedências diversas.

"M00"

PARADA INCONDICIONAL

Essa função proporciona a parada de todos movimentos do torno. Normalmente é utilizada para o operador executar a virada da peça, quando tivermos duas fases de usinagem no mesmo programa.

"M01"

PARADA CONDICIONAL

Caso o operador tenha ativado no painel a mensagem: "OPTIONAL STOP", tem-se a parada dos movimentos ao atingir-se o bloco que contenha M01.

"M02"

FIM DE PROGRAMA

Normalmente utilizada quando os programas são lidos em fitas emendadas em forma de laço.

"M03"

LIGA EIXO-ÁRVORE NO SENTIDO HORÁRIO

Liga o eixo-árvore no sentido-horário olhando-se a placa frontalmente.

"M04"

LIGA EIXO-ÁRVORE NO SENTIDO ANTI-HORÁRIO

Idêntico ao anterior, mas com sentido inverso de rotação.

"M05"

DESLIGA EIXO-ÁRVORE

Proporciona a parada do movimento de rotação do eixo árvore.

"M06"

INDEXAÇÃO DE FERRAMENTA

Libera o movimento de indexação da torre para en-

trada na posição de trabalho de ferramenta especificada anteriormente através da função T.

"M08"
LIGA REFRIGERANTE DE CORTE
Proporciona o acionamento da bomba de fluido de corte.

"M09"
DESLIGA REFRIGERANTE DE CORTE
Desliga a bomba de fluido de corte.

"M30"
FIM DE PROGRAMA
Utilizado para encerrar programas.