

SISTEMA DE VISUALIZAÇÃO DE DADOS GEOMÉTRICOS DE PROGRAMAS CN PARA OPERAÇÕES DE FRESAMENTO

Alexandre Dias da Silva

Universidade Federal de Santa Maria

Universidade Federal de Santa Maria, Avenida Roraima, sem nº, Bairro Camobi, Santa Maria – RS

adiass@ctlab.ufsm.br



II CONGRESSO NACIONAL DE ENGENHARIA MECÂNICA

II NATIONAL CONGRESS OF MECHANICAL ENGINEERING

12 a 16 de Agosto de 2002 - João Pessoa – PB

Denizaldo Soares Corrêa Junior

Universidade Federal de Santa Maria

Rua Cel. Niederauer, nº 947 ap. 408, Centro, Santa Maria – RS

denizaldo@nafa.ufsm.br

Resumo. Equipamentos com Comando Numérico Computadorizado (máquinas CNC) são empregados principalmente em processos de fabricação por usinagem. Em operações de fresamento, a geometria definida pelo programa da máquina pode ser visualizada pela conversão dos movimentos programados em entidades geométricas de um sistema CAD. O objetivo desse trabalho é propor uma metodologia para essa conversão, aplicada a uma fresadora com três eixos lineares de movimento. O sistema processa dados geométricos do programa da máquina, definidos com os seguintes recursos de programação: interpolação linear padrão; posicionamento linear polar; interpolação circular padrão; arcos polar; ciclo de furação; auto-rotina de círculo, quadrado ou retângulo de furos; e auto-rotina para fresamento de alojamento retangulares, quadrados ou circulares, reconhecendo também programas descritos na forma parametrizada. A metodologia foi implementada como aplicativo de um sistema CAD e exemplos de aplicação mostraram a eficiência do sistema para editoração de programas CN. A utilização da técnica em sistemas produtivos reduz os custos de fabricação porque elimina a necessidade de investimento em sofisticados programas computacionais para integração entre as fases de projeto e fabricação, quando o uso desses tem apenas o objetivo de editar programas de equipamentos CNC.

Palavras chave: Interação CAD/CAM, Programação CN, Comando numérico computadorizado, Programação Parametrizada.

1. Introdução

A indústria moderna tem desenvolvido e melhorado seus métodos de fabricação muito rapidamente, por conta de um mercado cada vez mais concorrido. A necessidade de métodos de fabricação mais rápidos, baratos e eficientes tem sido um fator bastante relevante quanto à sobrevivência de uma empresa de manufatura.

A automação de diversos processos tem sido a solução encontrada pela grande maioria das empresas, pois os processos automatizados permitem uma fabricação mais precisa quanto a dimensões e redução do tempo de produção. Ela proporciona maior repetibilidade das características do produto, reduzem a fadiga de operadores e permitem uma maior flexibilização da produção.

O fresamento é um processo de manufatura por usinagem, no qual peças prismáticas são produzidas através da retirada do excesso de material que corresponde à diferença entre a peça bruta e a superfície projetada, objetivando dar a forma e o acabamento desejados. A ferramenta de corte, ou fresa, possui movimento de rotação que define a velocidade de corte. A velocidade de avanço é obtida por movimentos lineares e/ou angulares da mesa de fixação da peça e/ou do cabeçote de fresamento. A versatilidade do processo quanto à forma da superfície usinada depende do número de eixos de movimentos (Rao et al, 1996).

Com o crescente desenvolvimento da tecnologia de automação, a indústria tem adotado, cada vez mais, equipamentos de comando numérico computadorizado na operação de fresamento, convencionalmente denominados de máquinas CNC. Os movimentos da fresadora são controlados

pelo comando da máquina através de uma sequência de operações que contém os dados geométricos e tecnológicos das trajetórias da ferramenta. Essas instruções de operação constituem o programa de comando numérico ou programa CN (Tlustý, 1999).

Um programa CN, na linguagem da máquina, pode se tornar de difícil interpretação e identificação, quando se trata de peças com um certo grau de complexidade e, principalmente quando se trata de programação parametrizada. Como esses programas contém um conjunto de códigos de funções preparatórias e de movimentação, uma interpretação mais adequada dos mesmos é feita por sistemas de simulação. A visualização gráfica da geometria da peça descrita no código facilita a identificação, possíveis correções de geometria e/ou modificação geométrica de um programa pré-existente. A simulação de um programa CN pode ser feita diretamente na tela gráfica do comando CNC ou em módulos de sistemas CAD/CAM. No primeiro caso, tem-se a desvantagem da redução da produtividade do equipamento, pois enquanto o operador simula um programa na máquina, ela não estará usinando. A simulação de programas em sistemas CAD/CAM pode não ser viável, principalmente para pequenas e médias empresas. Os custos de aquisição e manutenção desses sistemas não justificam sua aplicação apenas para esse fim.

O objetivo desse trabalho é apresentar uma metodologia de desenvolvimento de aplicativos em sistemas CAD para visualizar a geometria de peças, através da interpretação dos códigos do programa CN. O sistema foi implementado em uma ferramenta de projeto assistido por computador, AutoCAD 2000, (Autodesk, 2000) através de linguagem computacional, AutoLISP 2000, (Gaál, J., 2000) disponibilizada ao usuário para a criação de funções específicas. O programa foi desenvolvido para interpretar os códigos de programação CN convencionais e parametrizados de uma fresadora com três eixos programáveis.

2. Funções de Programação

A programação da fresadora CNC especificada para esse trabalho consiste de um conjunto de códigos normalizados EIA e ISO, conforme descrito por Groover, 2000, e de funções específicas do seu comando (ROMI, 1997).

Palavras iniciadas pela letra G são funções preparatórias. Definem para o controle os modos de operação do programa. Estes modos de operação podem ser modo de movimentação linear ou circular (G00, G01, G73, G41, G42, G02 e G03), auto ciclos de: furação (G81), conjunto de furos (G24 e G25) e fresamento de alojamentos (G26) ou Funções de Transformação: espelhamento (G31) e escala (G72).

As palavras iniciadas com a letra M são as chamadas funções de Miscelânea, e as significativas para o resultado gráfico do programa são: Fim de Programa (M02, M30), Parada do Programa para Troca de Ferramenta (M06).

Existem ainda as funções parametrizadas, nas quais, o comando da máquina efetua operações matemáticas com variáveis computáveis, aplica operadores e expressões aritméticas, possibilita desvios condicionais na sequência do programa, e executa chamadas de sub-rotinas ou sub-programas.

2.1. Interpolação Linear Polar

Além das funções convencionais de movimentação da ferramenta (funções X, Y e Z), o Comando também possibilita a programação linear polar, através das funções adicionais:

R – comprimento do raio polar

A – ângulo do próximo movimento, centrado no ponto final do ultimo movimento

C – ângulo cujo vértice está num centro de posicionamento polar

I – coordenada X do centro de posicionamento polar

J – coordenada Y do centro de posicionamento polar

As combinações possíveis dessas funções para a programação linear polar são apresentadas na Figura (1).

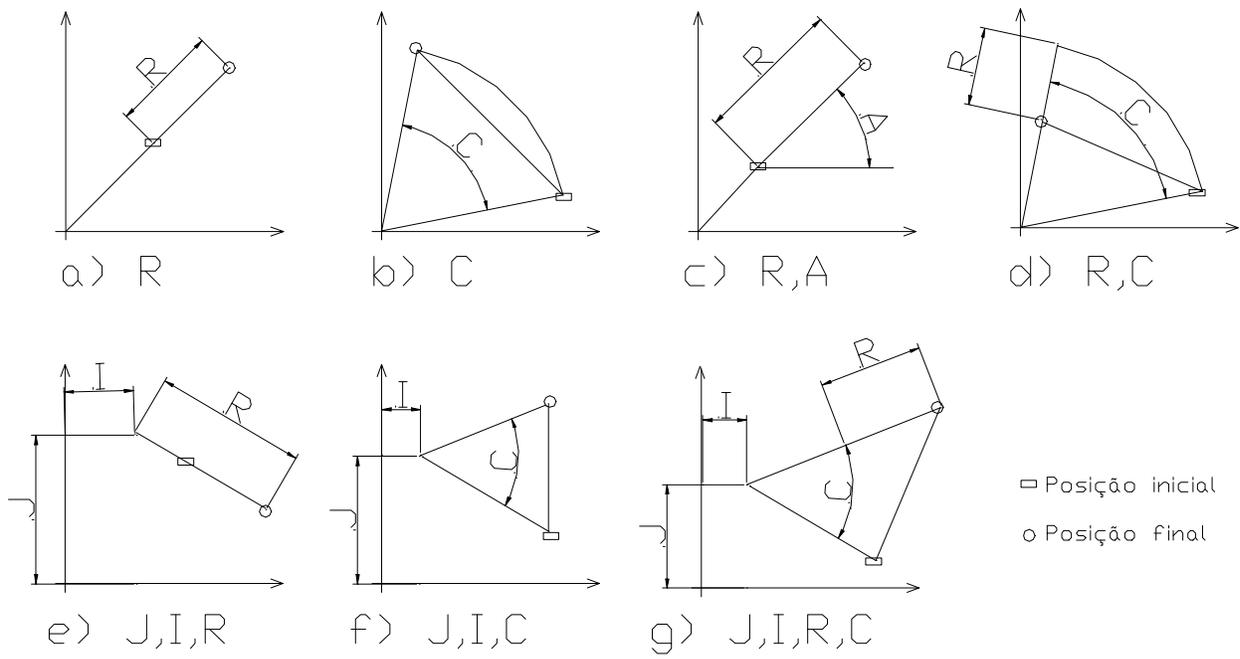


figura 1. Formas de programação linear polar.

Na programação linear utilizando apenas o valor do raio, como na Fig. (1.a), define-se um movimento polar de comprimento R do zero programa, passando pela posição atual da ferramenta. O valor do raio pode ser programado em absoluto ou incremental.

Na Fig. (1.b) tem-se programado somente o C portanto, haverá deslocamento da posição atual até o ponto final do ângulo C , que pode ser absoluto ou incremental, centrado no zero programa.

Quando R e A forem usados, a ferramenta irá mover-se para a distância R do ponto inicial, formando um ângulo A com o eixo X , como na Fig (1.c).

Caso R e C forem usados, a ferramenta será movida para a distância R do centro polar, formando um ângulo C com o eixo X , como na Fig (1.d).

Na programação linear utilizando valor do raio, I e J , como na Fig. (1.e), define-se um movimento polar de comprimento R do ponto definido por I e J , passando pela posição atual da ferramenta. O valor do raio pode ser programado em absoluto ou incremental.

Na Fig. (1.f) tem-se programado C , I e J , portanto, haverá deslocamento da posição atual até o ponto final do ângulo C centrado em I J , que podem ser absolutos ou incrementais.

Para o caso de R , C , I e J serem programados, o ponto final é conhecido pelo ângulo C , centrado em I J , e a uma distancia R do centro polar.

As possibilidades de programação acima descritas são aplicáveis as funções $G00$, $G01$, $G02$, $G03$, $G04$, $G73$, $G41$ e $G42$.

2.2. Interpolação Circular Polar

Na interpolação circular, também é possível utilizar funções polares para a definição do ponto final do arco e do centro do arco. Na Figura (2), demonstra-se as formas de programações circular polares.

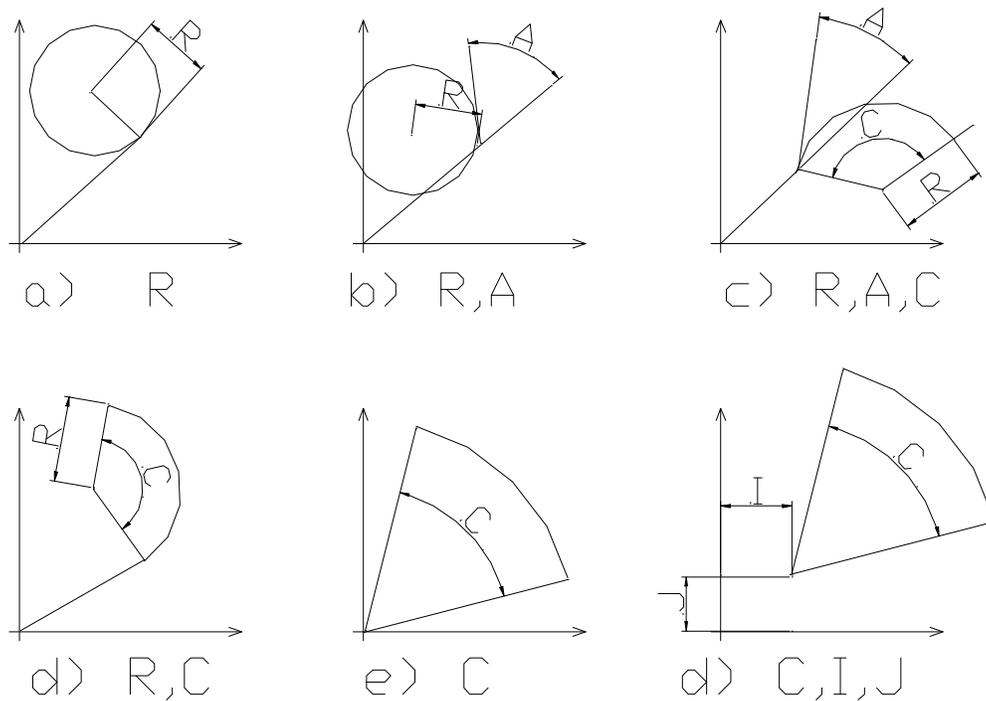


figura 2. Interpolações circulares polares.

Quando somente a função R é utilizada para definir uma interpolação circular, Fig (2.a), é definida uma função circular de raio R, tangente ao ultimo movimento.

Programando R e A, Fig (2.b), uma função circular de raio R é definida, tangente ao ângulo A centrado no ponto inicial da movimentação.

Quando a interpolação circular é programada conforme a Fig (2.c), define-se um arco, de comprimento C, centrado no centro do arco, raio igual a R e, formando uma tangente com o ângulo A, centrado no ponto inicial do arco.

Programando as funções R e C, tem-se como resultado um arco, tangente ao movimento anterior, de raio igual a R e, um comprimento C, Fig (2.d).

Caso somente C seja programado, é produzido um arco de comprimento C e, centrado no zero programa.

Na Fig (2.d), tem-se a programação utilizando C, I e J, Resultando em um arco de comprimento C, centrado em I J.

As formas de programação acima descritas estendem-se as funções G02 e G03.

2.3. Auto-Rotinas

As funções abaixo são as auto-rotinas significativas para o objetivo do trabalho:

A função G81 é a auto-rotina de furação. Quando ativada, determina a execução de uma série de furos idênticos para as diversas coordenadas programadas por X e Y, sendo desativada somente pela função G80. Para a sua programação, são necessários os parâmetros de profundidade (Z) e permanência no fundo do furo (D).

A função G26 define uma auto-rotina para fresamento de alojamentos retangulares, quadrados ou circulares. Esta função permite automaticamente desbastar e dar acabamento em alojamentos. Para alojamento retangular é possível especificar um raio de arredondamento nos cantos usando a função R, que deve ser maior que o raio da ferramenta utilizada para usinagem do alojamento. Requer as coordenadas da extremidade oposta do retângulo (X, Y), profundidade do alojamento (Z), sobremetal para acabamento (I, J e K) e, número de passes para os eixos X e Y (D) e para a profundidade (L).

A auto-rotina de círculo de furos é determinada pela função G24. Especifica a execução de uma série de operações em diversas posições dispostas em um padrão circular de pontos igualmente espaçados. Requer os parâmetros do centro do círculo, início do círculo, total de pontos do círculo completo e o total de auto-ciclos executados.

A função G25 determina um retângulo de furos igualmente espaçados. Necessita os dados de espaçamento entre furos e número de furações devem ser executadas.

2.4. Funções de Transformação

As funções de transformação são as funções capazes de modificar as coordenadas dos eixos programados.

A função G31 estabelece o espelhamento de eixos, ou seja, a inversão do sinal da coordenada programada para as próximas funções. Pode ser programada para o espelhamento em um ou dois eixos. Permanece ativa até a programação da função G30.

A função de Escala é definida por G72, esta, através do fator de multiplicação que poderá ser de 0,05 a 20 reduz ou aumenta os movimentos programados. Pode ser executado em um ou nos dois eixos.

2.4. Funções Parametrizadas

Funções parametrizadas são as funções nas quais, o comando da máquina pode efetuar operações matemáticas com variáveis computáveis, aplica operadores e expressões aritméticas como soma, subtração, multiplicação, divisão, logaritmos, exponenciais, etc.. Também pode possibilitar desvios condicionais na sequência do programa, e executa chamadas de sub-rotinas ou sub-programas de acordo com o resultado de expressões lógicas.

3. Metodologia

O programa desenvolvido nesse trabalho, denominado DECODIFICA PROGRAMA CN, é subdividido em 7 rotinas: *Decodifica*, *Analiza_dados*, *Desenha_retas*, *Desenha_arco_antihorário*, *Desenha_arco_horário*, *Desenha_retângulo*, *Parametrizada*.

O programa *Decodifica* define a sequência básica de processamento e sua lógica é apresentada no fluxograma da Fig (3). Os dados de entrada do sistema são: o endereço do arquivo onde está o código de comando numérico (programa CN), o ponto inicial para o desenvolvimento do desenho e o diâmetro da ferramenta inicial. A leitura do arquivo é feita de caracter por caracter, até encontrar o primeiro caracter significativo para a execução do desenho (G ou M). A identificação do tipo de função define a sequência de processamento para caracterização geométrica do programa CN (forma da trajetória da ferramenta) ou fim de execução (função fim de programa).

As opções de processamento são: Movimentação (G00, G01, G02, G03, G41, G42, G73), conjunto de furos (G24 e G25), Auto ciclos (G81 e G26) ou comando para a troca de ferramenta (M06) e, são diferenciados por uma estrutura condicional.

Caso a função de programação que está sendo processada seja uma função de movimentação, dois tipos de trajetórias são diferenciadas no ambiente de desenho do CAD: *avanço_rápido* e *avanço_de_trabalho*. Na sequência, o programa encontra, no arquivo, o caractere que descreve o nome de coordenada então, é solicitado o sub-programa *Analiza_dados* que armazena o valor da coordenada em uma variável. Esta rotina de análise de dados é repetida até encontrar o símbolo de fim de bloco “#”. Então é chamado o sub-programa de desenho (que pode ser *Desenha_retas*, *Desenha_arco_antihorário*, *Desenha_arco_horário*). Na sequência, é encontrado o próximo caractere interessante no arquivo, caso este seja uma nova função (G ou M), o programa identifica a função e retorna para a estrutura condicional de identificação da função, caso seja um caractere que defina uma coordenada, o programa executa uma nova análise de

dados e, repete a função de movimentação anterior.

Porém, se a função encontrada for um conjunto de furos, o programa procede à operação de análise de dados, seguindo do cálculo das coordenadas dos furos e armazenagem dessas coordenadas em uma lista. Então é lido o próximo caractere interessante, visando executar uma nova rotina (no caso de uma função) ou repetir a rotina de conjunto de furos (no caso de caractere de coordenadas).

Para o caso de a função ser um Auto-ciclo, a seqüência de processamento é a seguinte: análise de dados, localização do centro do furo através das coordenadas fornecidas, escolha da trajetória de avanço rápido, execução do avanço até o centro do furo, escolha da trajetória de avanço de trabalho e, representação adequada do furo. Na seqüência é identificada a próxima função, caso seja a função de cancelamento G80, o programa retorna para a estrutura de identificação de funções, caso contrário, será executado um novo furo determinado pelas coordenadas dadas.

Caso seja programada uma troca de ferramenta, o programa pede ao operador qual o diâmetro da ferramenta seguinte, calcula o raio desta e, identifica a próxima função.

A seqüência de identificação de comando e processamento de função repetida bloco a bloco do programa CN até encontrar o comando M02 ou M30 que simbolizam o fim de programa.

O sub-programa *Analiza_dados* é utilizado para armazenar o valor de uma coordenada em uma variável. Este sub-programa, quando solicitado pelo programa principal, recebe deste o caracter que denomina qual a coordenada está sendo dada pelo arquivo, o caracter em questão é o primeiro item da lista PAR. Primeiramente, através de uma estrutura condicional, é visto qual coordenada esta sendo dada. Então é lido o primeiro caracter que descreve o valor da coordenada e, verifica-se se este é um valor numérico ou o sinal de negativo, caso o caracter seja o símbolo de negativo, será ativada a variável “Negativo” e, será lido um novo caracter. Então os valores subseqüentes são lidos, processados e os armazenados em uma variável com o mesmo nome da coordenada dada. Esta rotina repete-se até que seja encontrada a denominação de uma nova coordenada ou fim de bloco. Existe dentro dessa rotina de repetição um contador que conta quantas casas decimais são fornecidas. Após todo o valor ter sido armazenado, existe um teste que verifica se existiu o sinal de negativo no inicio do valor, caso tenha existido, o valor armazenado na variável é multiplicado por -1 e, finalmente existe um teste caso tenha existido casas decimais, caso existam, o numero é corrigido por meio do contador de casas decimais citado anteriormente. Esta rotina retorna ao programa principal o valor da variável e, o nome do próximo valor de coordenada.

Os sub-programas *Desenha_retas*, *Desenha_arco_antihorário* e *Desenha_arco_horário*, são bastante semelhantes, todos iniciam com a verificação de quais as coordenadas são fornecidas pelo programa principal. Na seqüência os valores são direcionados para a opção de desenho que necessita estas coordenadas. Em seguida, a partir dessas coordenadas são calculados os pontos inicial, final e centro (para o caso de arcos) que são armazenados como listas dentro das coordenadas P1, P2 e CENTRO, os valores são todos em coordenadas cartesianas absolutas. Finalmente é desenhada a entidade, o ponto de localização da ferramenta é atualizado e as outras variáveis são zeradas.

O sub-programa *Desenha_retângulo* é bastante semelhante aos sub-programas comentados anteriormente. Este porém, resulta no desenho de um retângulo de cantos arredondados. Caso não seja fornecido um raio de arredondamento dos cantos, o sub-programa adota o raio da ferramenta como tal. Em seguida o sub-programa verifica quais as coordenadas são dadas, a partir dessas, são calculados os pontos finais das retas e arcos que irão descrever o retângulo de cantos arredondados.

A sub-rotina *Parametrizada* tem por finalidade a identificação e interpretação de funções parametrizadas. Sempre que alguma função parametrizada for identificada, o programa irá interpretar tal função, e reescreve-la na forma de programação CN convencional para que a mesma possa ser identificada pelas funções acima descritas.

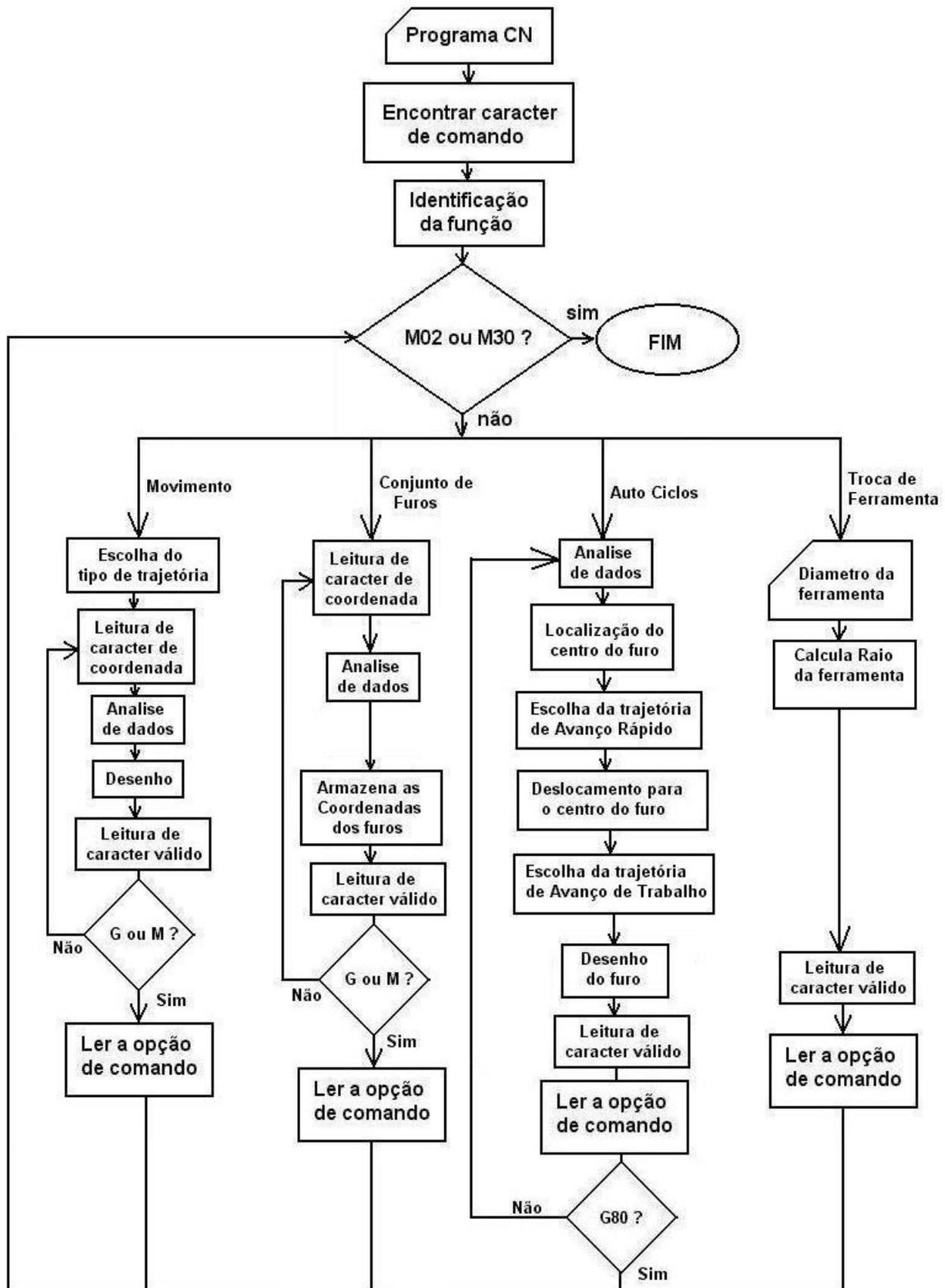


figura 3. Fluxograma do programa principal

4. Resultados

O programa CN apresentado no quadro foi utilizado como exemplo para análise de resultados do sistema desenvolvido. O conteúdo do programa apresenta variadas funções de programação, incluindo interpolações linear e circular polares e auto-rotinas. Os blocos iniciais do programa (N10 a N40). O bloco N050 apresenta o movimento inicial de aproximação da ferramenta. A função de avanço rápido é diferenciada pela característica de cor da trajetória. O primeiro avanço programado (N60) é uma interpolação linear na forma convencional. O bloco seguinte contém uma interpolação linear polar através da informação do raio polar e do ângulo do movimento em relação à direção X. Esse bloco executa o chanfro do canto superior esquerdo da peça.

O bloco N100 contém uma interpolação circular polar para executar o arredondamento no canto superior direito. Essa função utiliza uma combinação de parâmetros que definem o raio do arco e o respectivo comprimento. As funções contidas nos blocos N120 e N130 são na forma incremental (G91 no bloco N110). Após redefinir as coordenadas para valores absolutos (G90 no bloco N140), uma interpolação circular polar sentido anti-horário é especificada através das coordenadas do centro (I320.7 e J50.7) e comprimento do arco (C180.).

A função G26 especificada no programa CN estabelece a usinagem da cavidade retangular que aparece na figura. Após a execução da cavidade, a ferramenta desloca-se em movimento rápido para a primeira posição do conjunto de ponto definidos pela função preparatória G25 no bloco N200. A função G81 no bloco anterior especifica que furos iguais com profundidade até a coordenada Z40.0, plano de retração distante 2mm da superfície da peça e avanço de trabalho de 150mm/min, devem ser executados nessas posições. As circunferências representativas dos furos são geradas no desenho através da informação sobre o diâmetro da ferramenta solicitada ao usuário como decorrência da compilação do bloco N

Partindo do programa colocado abaixo foi obtido o resultado demonstrado na Fig (4).

```
N0010 G99#
N0020 G90#
N0030 G71#
N0040 G17#
N0042 X10. Y-50 T01 M06;fresa d10#
N0044 O01 S1500 M08#
N0046 M03#
N0050 G00 X200. Y150.#
N0055 G41#
N0060 G01 X200. Y300. F800#
N0070 r100. A45#
N0090 x500#
N0100 G02 R70 C90#
N0110 G91#
N0120 G1 Y-250.#
N0130 X-440.#
N0140 G90#
N0150 G03 I320.7 J50.7 C180.#
N0160 G01 X200 Y150#
N0165 G40#
N0170 G X250. Y220.#
N0180 G26 X400. Y300. Z-25. L10 D16#
N0190 G X500. Y280.#
N0194 X10. Y-50 T02 M06;broca d20mm#
N0196 O01 M08#
N0195 G81 R2. Z40. F150#
N0200 G25 X100. Y-80. I4 J3#
N0210 G80#
N0220 G X100. Y100.#
N0230 M02#
```

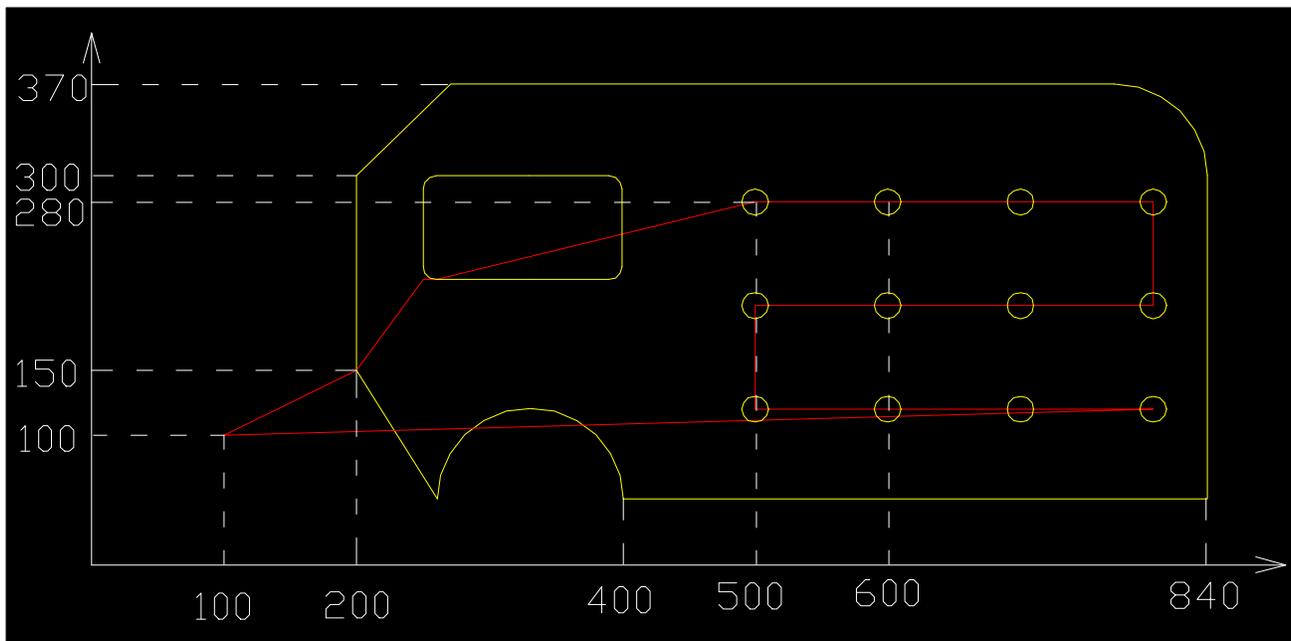


figura 4. Representação gráfica do programa colocado anteriormente.

5. Conclusão

A técnica apresentada nesse trabalho se distingue de métodos convencionais de simulação de programas CN em diferentes aspectos. No sistema proposto, maior rapidez e clareza na identificação de programas CN são obtidos através da visualização apenas da geometria da trajetória da ferramenta em operações que definem a forma final da peça. A apresentação gráfica dispensa linhas de trajetórias irrelevantes da ferramenta, como operações de desbaste em ciclos fixos de usinagem, e cria entidades de desenho que não podem ser visualizadas por essas trajetórias, como círculos representativos de furos.

O sistema proposto para visualização geométrica de programas CN representa uma metodologia que pode ser adaptada a outros sistemas computacionais. No CAD, ele foi implementado através da aplicação de recursos que permitem aos usuários a criação de funções personalizadas. Além de serem comuns esses recursos em diferentes sistemas CAD, a técnica também pode ser aplicada em compiladores de linguagens computacionais que apresentem funções gráficas compatíveis. Assim, a metodologia proposta representa uma alternativa ao investimento para aquisição de pacotes comerciais de integração CAD/CAM, ou mesmo de sistemas CAD. Essa opção pode representar uma importante redução dos custos de produção, principalmente para empresas de pequeno e médio porte.

6. Agradecimentos

Os autores agradecem à FAPERGS pelo apoio financeiro através da concessão de bolsa de Iniciação Científica.

7. Referências

- Rao, N., Bedi, S. and Buchal, R., 1996, "Implementation of the Principal-Axis Method for Machining of Complex Surfaces", *The International Journal of Advanced Manufacturing Technology*, Vol. 11, pp. 249-257.
- ROMI, 1997, "Manual de Programação MACH 8MP – Discovery 308".
- Autodesk, 2000, "AutoCAD 2000 Reference Manual".

Thusty, G., 1999, "Manufacturing Processes and Equipment", Ed. Prentice Hall, New Jersey, USA, 928p.

Gaàl, J. 2000, "Curso de AutoLISP"