

MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
DEPARTAMENTO DE ENGENHARIA MECÂNICA

Sistema CAD/CAM para usinagem com recursos de robótica industrial

por

Alberto Fernández Alonso

Monografia apresentada ao Departamento de Engenharia Mecânica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para obtenção do diploma de Engenheiro Mecânico.

Porto Alegre, Junho de 2012.



Universidade Federal do Rio Grande do Sul  
Escola de Engenharia  
Departamento de Engenharia Mecânica

Sistema CAD/CAM para usinagem com recursos de robótica industrial

por

Alberto Fernández Alonso

ESTA MONOGRAFIA FOI JULGADA ADEQUADA COMO PARTE DOS  
REQUISITOS PARA A OBTENÇÃO DO TÍTULO DE  
**ENGENHEIRO MECÂNICO**  
APROVADA EM SUA FORMA FINAL PELA BANCA EXAMINADORA DO  
DEPARTAMENTO DE ENGENHARIA MECÂNICA

Prof. Arnaldo Ruben Gonzalez  
Coordenador do Curso de Engenharia Mecânica

Área de Concentração: **Projeto e Fabricação**

Orientador: Flávio José Lorini

Comissão de Avaliação:

Prof. José Antonio Esmério Mazzaferro

Prof. Heraldo José de Amorim

Prof. Cintia Cristina Petry Mazzaferro

Porto Alegre, junho 2012.

## - AGRADECIMENTOS

*Muito obrigado, Prof. Flávio José Lorini, pela orientação na bibliografia, explicações da linguagem RAPID, estrutura e escritura do português neste TCC.*

*Colega e aluno da Engenharia Elétrica, Pablo Leonardelli, pela ajuda prestada na estrutura e linguagem de programação no software Visual Basic. Assim como a geração das interfaces do programa.*

ALONSO, A. F. **Sistema CAD/CAM para usinagem com recursos de robótica industrial.** 2012. 16 folhas. Monografia (Trabalho de Conclusão do Curso em Engenharia Mecânica) – Departamento de Engenharia Mecânica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012.

## RESUMO

O presente trabalho trata de um processo de usinagem *CAD/CAM*, utilizando recursos de robótica. O robô não é normalmente equipamento utilizado para fazer operações de usinagem, no entanto, neste projeto pretende-se demonstrar que a usinagem robotizada também é uma alternativa viável. A elaboração deste projeto, baseado em um sistema *CAD*, neste caso AutoCAD, onde é gerado qualquer desenho gráfico. Este desenho é armazenado em uma extensão de arquivo *DXF* e mediante uma interface *CAM*, programado em Visual Basic converte-se o arquivo *DXF* a um arquivo da linguagem *RAPID*, utilizada pelo robô. O sistema desenvolvido no Visual Basic faz uma leitura da informação do desenho e as suas trajetórias, representadas no arquivo *DXF*, exportando e convertendo os dados na linguagem *RAPID*, com a qual o robô executa a usinagem do desenho projetado.

**PALAVRAS-CHAVE:** *CAD/CAM*, automação da manufatura, usinagem robotizada

ALONSO, A. F. **Sistema CAD/CAM para usinagem com recursos de robótica industrial.** 2012. 16 folhas. Monografia (Trabalho de Conclusão do Curso em Engenharia Mecânica) – Departamento de Engenharia Mecânica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012.

## ABSTRACT

This work is a process of manufacture *CAD/CAM*, using robotic resources. The robot is not often used to produce manufacturing operations, however, this project intends to show that machining, is also a viable alternative. This project is made by a *CAD* system, in this case AutoCAD, which generates any graphic design. This design is stored in a *DXF* file extension and by a *CAM* interface, programmed in Visual Basic to turn the *DXF* file into an *RAPID* language file, used by the robot. The system developed in Visual Basic, reads the design information and its routes, represented by the *DXF* file, exporting and converting the information into a *RAPID* language, in which the robot executes the manufacturing process of the projected design.

**KEYWORDS:** *CAD/CAM*, manufacturing automation, robotised production

## ÍNDICE

	Pág.
1 INTRODUÇÃO .....	1
2 OBJETIVOS .....	1
3 REVISÃO BIBLIOGRÁFICA .....	2
3.1 Sistemas CAD/CAM e padrões gráficos .....	2
3.2 Robótica Industrial .....	3
3.3 Linguagens de programação de robôs industriais .....	4
3.3.1 Linguagem RAPID .....	4
4 METODOLOGIA .....	5
4.1 Desenho da Peça .....	6
4.2 Módulo de interpretação .....	7
4.3 Teste experimental .....	10
5 RESULTADOS E DISCUSSÃO .....	10
6 CONCLUSÕES .....	11
7 SUGESTÕES PARA TRABALHOS FUTUROS .....	12
REFERÊNCIAS BIBLIOGRÁFICAS .....	12

## 1. INTRODUÇÃO

O processo de fabricação por usinagem, realizado por máquinas-ferramenta, tem seu desenvolvimento contínuo baseado em soluções para aumentar a produtividade e a qualidade de produtos e processos. A redução do ciclo de projeto e fabricação de um produto e a competitividade exigem crescente flexibilização na aplicação dos equipamentos para a fabricação de peças de características variadas e em pequenos lotes além da capacidade de obtenção de geometrias complexas.

Atualmente, com a tecnologia de integração de máquinas, equipamentos e sistemas em chão-de-fábrica, busca-se maior flexibilidade e produtividade utilizando-se tecnologias e recursos que incluem desde o comando numérico até os ambientes altamente automatizados como manufatura integrada por computador. Destacam-se nesse enfoque os sistemas de projeto auxiliado por computador, sistemas *CAD (Computer Aided Design)*, bem como de manufatura assistida sistemas *CAM, (Computer Aided Manufacturing)*.

O presente trabalho propõe o desenvolvimento de um sistema para auxiliar a manufatura, um módulo de pós-processador de arquivos de desenho para viabilizar utilização de robôs em operações de usinagem, mais precisamente fresamento, de modo automatizado.

Sua estrutura permite atender os objetivos propostos utilizando os recursos de integração entre sistemas *CAD/CAM*, para proporcionar flexibilidade na utilização de sistemas robóticos. A automação do processo de programação de robôs, através de um pós-processamento da geometria, viabiliza a execução de trajetórias com adequação de parâmetros de usinagem.

## 2. OBJETIVOS

O objetivo principal deste trabalho é viabilizar a execução de processos de usinagem, especificamente de fresamento, através da condução da ferramenta de corte por um braço robótico. A definição, de modo automatizado, para a geração da trajetória da ferramenta parte de um arquivo com informações gráficas, de onde são extraídas as coordenadas cartesianas da geometria de uma peça a ser usinada.

A partir das coordenadas do modelo *CAD 2D*, em uma etapa inicial gera-se um arquivo com todos os dados de trajetória, os quais posteriormente são processados através de um aplicativo desenvolvido em *Visual Basic*. No sistema proposto como software aplicativo, desenvolvido como um módulo de *CAM* são programadas as instruções de código de comando para um manipulador robótico.

O trabalho propõe-se como uma alternativa a usinagem tradicional para uma usinagem utilizando recursos de robótica, através da conversão dum arquivo gerado em software *CAD* e traduzido diretamente para uma linguagem de manipulador robótico.

Para testes da proposta utilizam-se como arquivos de desenho, aqueles gerados no formato *DXF (Drawing Exchange Format)* com o processamento das informações em aplicativo desenvolvido em *Visual Basic* e o código de instruções para robô definidos na linguagem *RAPID*, linguagem essa específica para controles de programação de equipamentos de fabricação *ABB*.

### 3. REVISÃO BIBLIOGRÁFICA

#### 3.1 Sistemas CAD/CAM e padrões gráficos

O uso de imagens em sistemas computacionais teve seu início em 1950, logo após o surgimento dos primeiros computadores. Nessa época a criação de gráficos e desenhos estava restrita apenas as instituições de pesquisa com aplicações de caráter exploratório, situação mantida até o final da década de 50 [Meneghello, 2003]. A computação gráfica pode ser definida como conjunto de algoritmos, técnicas e metodologias para tratamento e representação gráfica de informações através da criação, armazenamento e manipulação de figuras utilizando-se computadores e dispositivos periféricos gráficos.

Com os avanços tecnológicos na área computacional e o despertar do interesse das indústrias, que viam neste sistema um grande potencial, iniciou-se um grande desenvolvimento da computação gráfica já em meados da década de 60. Alguns setores da indústria, como o automobilístico e o aeroespacial, iniciaram a implementação de sistemas gráficos para apoio a projeto e fabricação. Nos anos 70, surgiram os primeiros sistemas *CAD* também denominados sistema de gráficos interativos. Com a evolução tecnológica disponibilizaram-se novos hardwares e os sistemas *CAD* consolidaram-se definitivamente surgindo os primeiros sistemas tridimensionais (protótipos), além de sistemas de modelagem de elementos finitos e análise de engenharia (*CAE - Computer Aided Engineering*).

No início dos anos 80, equipamentos e dispositivos computacionais tornaram-se mais acessíveis com o surgimento dos computadores pessoais produzidos em massa. O uso de sistemas gráficos através do mapa de bits (*bitmap*), definido como uma representação através de zeros e uns, popularizou-se. Assim multiplicou-se a utilização de aplicações gráficas, proporcionando a milhões de novos usuários acesso a editores de texto, planilhas e programas de desenho. Entre os feitos relevantes da época destaca-se o aparecimento do *IGES (Initial Graphics Exchange Specification)*, usado para especificação de formatos gráficos e linguagem para transmissão e armazenamento de dado de definição de produtos.

A década de 90 é caracterizada pela automação crescente dos processos industriais através da integração de técnicas de desenho (*CAD*), análise (*CAE*), simulação e fabricação (*CAM*). Atualmente os avanços nos sistemas de computação fazem com que as técnicas *CAD*, *CAE* e *CAM* disponibilizem a engenheiros e usuários ambientes de fabricação integrados. A utilização destas técnicas passa de uma opção a uma exigência no setor industrial.

O projeto assistido por computador ou sistema *CAD*, é uma ferramenta computacional de auxílio a qualquer atividade de projeto para desenvolver, analisar ou modificar um modelo de engenharia, com o objetivo de diminuir custos, melhorar a qualidade e aumentar a produtividade. O projeto assistido por computador é um fator fundamental na integração de sistemas e na automação de processos.

Segundo Bedworth, 1991, algumas vezes o termo *CAD* é interpretado como uma ferramenta de apoio a desenho; outras, como auxílio à análise; e outras como automação total de projeto envolvendo desenho, análise e otimização. Estes sistemas são baseados em computação gráfica interativa e a utilizam para criação, transformação e visualização de dados na forma de figuras ou símbolos. Deste modo, qualquer aplicação que inclua uma interface gráfica na realização de tarefas de engenharia se considera um software de *CAD*.

No enfoque da automação industrial, a integração entre sistemas computacionais de auxílio a projeto e manufatura iniciou com a junção da tecnologia de grupo e dos sistemas *CAD*. A base de tecnologia de integração de sistemas é a manipulação, armazenagem e compartilhamento de informações. Integração implica automação e esta compreende que hardware e software podem criar uma base de dados em *CAD* e usá-la para analisar e fabricar uma peça automaticamente com pouca ou nenhuma intervenção humana. Embora não haja uma integração completa que dispense a intervenção humana, existem diferentes níveis de implementação desta tecnologia. O processo de integração *CAD-CAM* inicia na concepção do modelo geométrico através da geração de um arquivo de dados pelos sistemas *CAD*. Este

arquivo deve conter não somente dados da geometria do modelo, mas também informações e descrições da peça; como dimensões, tolerâncias e tipo de material.

Atualmente existe uma variedade de extensões de arquivos para transferência de dados. Alguns destes padrões se destacam pela sua utilização, embora não exista um consenso em torno de um padrão único. Dentre os padrões mais conhecidos, destacam-se o *IGES*, *DXF*, *DWG* (*DraWinG*) e *STEP* (*Standard for the Exchange of the Product Model Data*).

O *IGES* foi desenvolvido em 1979 [Meneghello, 2003], pelo *US National Bureau of Standards* com o objetivo de auxiliar a troca de dados entre diferentes sistemas *CAD*. Sua estrutura define um formato de arquivo para armazenar e transmitir geometrias com pontos, linhas e superfícies; com uma especificação de estrutura e sintaxe em código *ASCII* (*American Standard Code for Information Interchange*). O *IGES* não é um completo padrão de transferência de dados de integração entre sistemas de manufatura, mas ainda é um dos padrões mais utilizados para troca de dados entre modelos geométricos e sistemas *CAD*.

*DXF* e *DWG* são formatos de arquivos para armazenagem e transferência de dados entre sistemas *CAD* desenvolvido pela *AutoDesk* para seu software de *CAD*. Os formatos *DXF* e *DWG* estão entre os mais populares formatos de arquivos para troca de dados entre sistemas *CAD*. O *DWG* (*DraWinG*) apresenta sua estrutura na forma binária compacta enquanto o formato *DXF* (*Data Exchange Files*) apresenta-se na forma *ASCII*.

Segundo McMahon, 1988 [apud, Meneghello, 2003], a estrutura do arquivo em formato *DXF* é dividida em quatro partes, que definem valores para as variáveis do sistema e parâmetros: dimensionamento e *layers*, informações de estilos de linhas, sistema de coordenadas e definições de blocos de exemplos de modelos, além de entidades e dados. Este formato aplica-se melhor na interpretação de geometrias bidimensionais, enquanto entidades geométricas de maior ordem como *splines* e superfícies complexas podem causar problemas.

O padrão *STEP* originou-se a partir de atividades de pesquisa de outros formatos, como o *ESPRIT* (*European Strategic Programme for Research in Information Technology*) procurando assim o desenvolvimento de um trabalho com um padrão unificado. O *STEP* consiste de um conjunto de normas que suportam o desenvolvimento do produto disponibilizando informações e integração entre sistemas. O formato contempla uma estrutura de dados de produto para a utilização em tudo o seu ciclo de fabricação, o que inclui projeto, manufatura, qualidade, testes e suporte. Este formato utiliza uma linguagem interpretável específica de modelagem denominada *EXPRESS*, utilizada para especificar as informações dos modelos. Esta linguagem permite um mapeamento dos dados armazenados para visualização gráfica dispensando a definição da representação da entidade.

Os sistemas *CAM*, ou auxílio de computadores à manufatura, podem ser definidos como a utilização da ferramenta computacional e de recursos de tecnologia digital, no auxílio às funções de manufatura. A integração entre sistemas digitais e os padrões de arquivos gráficos permite a interação entre os dados gerados na concepção de projeto, através dos sistemas *CAD*, e os sistemas *CAM*, possibilitando a automação dos processos de monitoramento, controle e gerenciamento da manufatura. Os sistemas *CAM* como efetivo uso da tecnologia computacional no planejamento, gerenciamento e controle das funções de manufatura através de interfaces diretas ou indiretas [Groover e Zimers, 1984].

### 3.2 Robótica Industrial

O termo robô nasceu da palavra tcheca *robota* (operário). Ele apareceu pela primeira vez em 1921, no livro *RUR - Robôs Universais* de Rossum, do escritor Karel Capek [Groover et alli, 1986].

Décadas atrás, os robôs faziam parte apenas da ficção científica, fruto da imaginação do homem. No início dos anos 60, os primeiros robôs começaram a ser aplicados com o objetivo de substituir o homem em tarefas que envolviam condições desagradáveis, tipicamente contendo altos níveis de calor, ruído, gases tóxicos, esforço físico extremo, ou trabalhos monótonos. Uma tendência crescente na aplicação dos robôs se justifica nos últimos



anos por razões importantes, quais sejam o constante aumento dos níveis salariais dos empregados e o avanço tecnológico no ramo de computadores que induz à redução dos preços do robô e uma significativa melhoria em seu desempenho.

Um robô consiste de um braço mecânico motorizado e um controlador, constituído de um computador industrial, sistema de potência e sensores que controla seus movimentos. O computador armazena em sua memória um programa que detalha o curso que o braço seguirá. Quando o programa está em execução, o computador envia sinais ativando motores que movem o braço e a carga no final dele, que é mantida pelo efetuador. Dois aspectos importantes do funcionamento de um braço mecânico correspondem ao sensoriamento do ambiente e como se realiza a programação do mesmo.

Segundo Niku, 2001, um robô sozinho não tem aplicação, este vem acompanhado com acessórios que pode ser dispositivos periféricos, algum tipo de sistema, máquinas de fabricação entre outros. O fator determinante de um dispositivo ser considerado um robô ou não é pelo simples fato de conhecer como é controlado. Um exemplo disso é se comparar-se um manipulador robótico e um guindaste convencional. No guindaste o controle é feito manualmente. Em um robô o controle é realizado por um computador executando algum tipo de programa.

### 3.3 Linguagens de programação de robôs industriais

Atualmente existem diversas linguagens de programação, sendo que cada fabricante faz o uso de uma própria linguagem associada ao seu equipamento. Essas linguagens, embora características de cada fabricante, incorporam a estrutura e sintaxe básica de linguagens como Basic, C e Fortran ou Pascal, linguagens comuns a sistemas computacionais. Linguagens de alto nível como Java, Pascal, C, entre outras, são baseadas em interpretador e executam as instruções de programa linha por linha, já as linguagens de baixo nível como *assembly* são baseadas no compilador utilizando linguagem de máquina, criando um código objeto antes de executar o programa.

As linguagens textuais para robôs foram criadas na década de 60. A primeira linguagem textual robótica foi a *WAVE* desenvolvida em 1963 como uma linguagem experimental para pesquisa no laboratório de inteligência artificial de *Stanford*. Em 1974 iniciou-se o desenvolvimento de uma nova linguagem em *Stanford*, essa linguagem foi chamada *AL* e ela possibilitou o controle de multiplex braços em atividades onde era necessária coordenação motora. Muitos dos conceitos da linguagem *WAVE* e *AL* fizeram parte do desenvolvimento da primeira linguagem textual robótica comercializável, *VAL* (*The Victor Assembly Language*, criada por *Victor Scheinman*). A *VAL* foi introduzida em 1979 pela *Unimation Inc.*, para os robôs em série *PUMA*. Esta linguagem foi aperfeiçoada para *VAL2* e lançada em 1984 [Groover, 1986]. Atualmente novas versões estão associadas aos robôs *ADEPT* e *STÁUBLI*.

O desenvolvimento de linguagem robótica também teve espaço no laboratório de pesquisa *Watson T.J.*, da *IBM*, no ano 1976. Duas das linguagens da *IBM* são a *AUTOPASS* e *AML* (*A Manufacturing Language*) sendo a segunda disponível comercialmente desde 1972 nos produtos robóticos da *IBM*.

Algumas das outras linguagens textuais robóticas são; a *RAIL* introduzida em 1981 pela *Automatix*; *MCL* (*Manufacturing Control Language*) desenvolvida pela *MC Donald Douglas* para força aérea americana como melhoramento do *APT* (*Automatically Programmed Tooling*), linguagem de programação numérica; e *HELP* disponível pela companhia elétrica geral sobre licença da firma italiana *DEA*.

#### 3.3.1 Linguagem *RAPID*

A linguagem de programação de robôs denominada *RAPID* foi desenvolvida por um fabricante de robôs industriais para a sua linha de produtos. Sua lógica de programação

estruturada contempla as instruções e parâmetros de controle das linguagens de programação tradicionais.

*RAPID* é uma linguagem de programação denominada de alto nível devido à sua característica de trabalhar com instruções escritas em linguagens interpretáveis como a língua inglesa. Com o objetivo específico de operação de sistemas robóticos, sua estrutura inclui recursos similares aos das modernas linguagens de programação estruturada, funções e procedimentos, rotinas de parâmetros, aplicações multitarefa, expressões lógicas e aritméticas, matrizes em até três dimensões, programas modulares, rotinas e definição de dados globais e locais. Um complexo conjunto de comandos específicos permite a programação e controle do manipulador para a execução das tarefas de movimentação.

Na linguagem *RAPID*, uma sequência de movimentação é programada como um número de movimentos parciais entre os pontos de trajetória definida [ABB Robótica, 1995]. Uma posição exata pode ser definida como um ponto de parada onde o robô encontra a posição programada precisa ou como um ponto de passagem onde o robô muda de direção, próximo ao ponto programado. A grandeza deste desvio é definida na descrição do comando de movimentação e dada em milímetros como uma dimensão de raio.

#### 4. METODOLOGIA

A metodologia seguida neste trabalho consiste na implementação dos módulos ilustrados na figura 4.1. Definido o desenho da peça no software *CAD*, o mesmo é analisado e transformado em instruções da linguagem *RAPID*, através de um módulo de interpretação, para execução como trajetória do robô. Cada módulo processo é descrito a seguir.

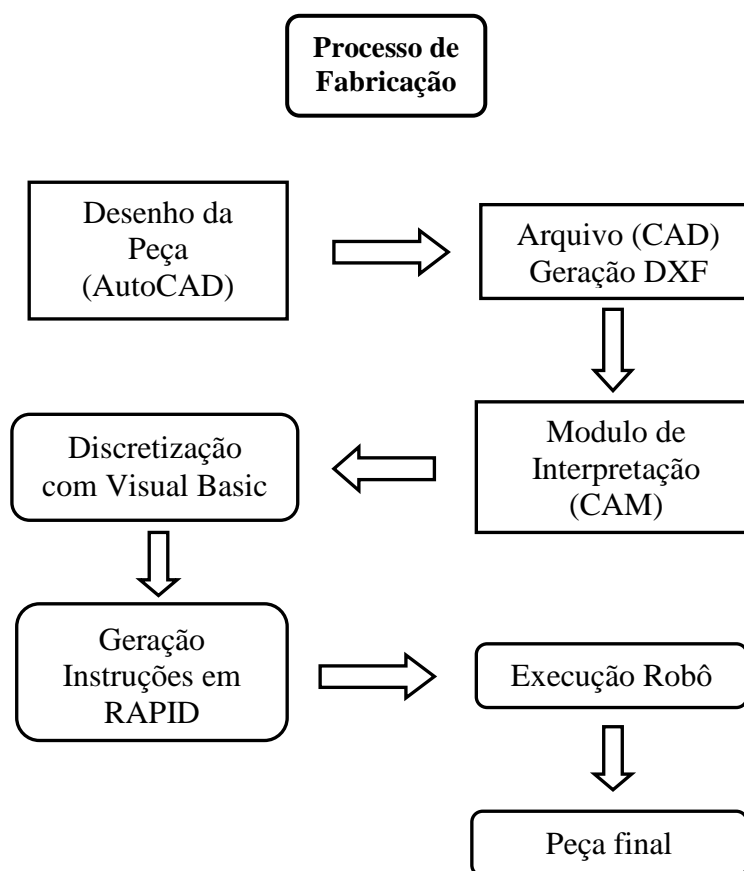


Figura 4.1 – Fluxograma geral do sistema.

## 4.1 Desenho da Peça

Um desenho realizado num sistema *CAD* pode ser armazenado em diversos formatos, conhecidos como os padrões gráficos sendo os mais comuns o *IGES*, *STEP*, *DXF* dentre outros. Esses padrões são armazenados na forma de texto que pode ser interpretado conhecendo-se sua estrutura. Na interpretação do texto do arquivo são identificadas entidades geométricas tais como: linhas, arcos, círculos, pontos, etc., através de expressões características compostas de palavras e códigos numéricos.

No presente trabalho utilizam-se informações do desenho partindo da leitura do arquivo *DXF*. Uma determinada geometria desenhada é armazenada no formato *DXF* segundo a seguinte estrutura: posição de cada entidade geométrica, componente de identidades e símbolo ou códigos que autenticam. Os arquivos são códigos *ASCII* em forma de instruções sequenciais onde em cada linha só se escreve um identificador ou valor correspondente ao elemento identificado.

O arquivo contém uma quantidade de dados iniciais sobre a configuração do programa no qual foi realizado o desenho. Estes dados não são utilizados para os fins desse trabalho. A parte útil do arquivo *DXF* começa depois de encontrar-se a palavra *entites* (entidades). Cada elemento do desenho identifica-se com um nome de entidade geométrica. Os mais comuns são ponto (*point*), linha (*line*), círculo (*circle*), arco (*arc*) e polilinha (*polyline*). Como exemplo, a figura 4.2 representa as características da entidade linha (*line*).

A estrutura da identificação de cada elemento contém dois tipos de caracteres numéricos. O primeiro é um identificador que cuja função é informar o tipo de atributo corresponde para respetiva entidade. O segundo identificador contém o valor do atributo. O significado dos identificadores mais comuns corresponde a tabela 4.1 [Matsumoto,2002].

Desenho Org novo, orden2: Bloc de notas	
Archivo Edición Formato Ver Ayuda	
AcDbLine	← Nome da entidade
10	← Identificador inicio coordenada "x"
2124.31429736269	← valor coordenada "x"
20	← Identificador inicio coordenada "y"
1305.473525202795	← valor coordenada "y"
30	← Identificador inicio coordenada "z"
0.0	← valor coordenada "z"
11	← Identificador fim coordenada "x"
2176.512227086683	← valor coordenada "x"
21	← Identificador fim coordenada "y"
1309.136365261734	← valor coordenada "y"
31	← Identificador fim coordenada "z"
0.0	← valor coordenada "z"
0	← concluí-o a descrição da entidade

Figura 4.2 – Identificadores de um arquivo DXF.

Tabela 4.1 - Parâmetros utilizados na leitura do arquivo DXF.

CODIGO	SIGNIFICADO DO IDENTIFICADOR
0	Concluí descrição entidade
6	tipo de linha
10	coordenada inicial em "x"
20	coordenada inicial em "y"
30	coordenada inicial em "z"
11	coordenada final em "x"
21	coordenada final em "y"
31	coordenada final em "z"
40	do radio da entidade
50	ângulo inicial de um arco
51	ângulo final de um arco
62	cor da entidade

#### 4.2 Módulo de interpretação

Partindo da informação do arquivo *CAD*, neste caso no padrão *DXF*, o programa realiza uma análise e interpretação para transformar as coordenadas das entidades geométricas em instruções de trajetória na linguagem específica para execução no robô. Deste modo obtém-se a definição da geometria a ser produzida através de uma ferramenta de usinagem. Na figura 4.3 representa-se a interface do programa implementado no software Visual Basic, que analisa as informações necessárias para fazer a interpretação e posteriormente a definição do código de instruções na linguagem do robô, neste caso linguagem *RAPID*.



Figura 4.3 – Interface de apresentação do programa.

O módulo de conversão é programado em Visual Basic. Caracteriza-se como um algoritmo para interpretar informações geométricas em pontos de trajetórias lineares ou circulares reescritas em um arquivo de instruções para um manipulador robótico.

O módulo de comunicação, ilustrado na figura 4.4, tem uma interface que possibilita ao usuário especificar a origem do arquivo com informações gráficas, bem como o destino do arquivo de instruções na linguagem *RAPID* para o manipulador robótico. Seleciona-se ainda, na interface, o número de passes e a profundidade de corte, em milímetros, no eixo "z" a ser executado pela ferramenta de corte.

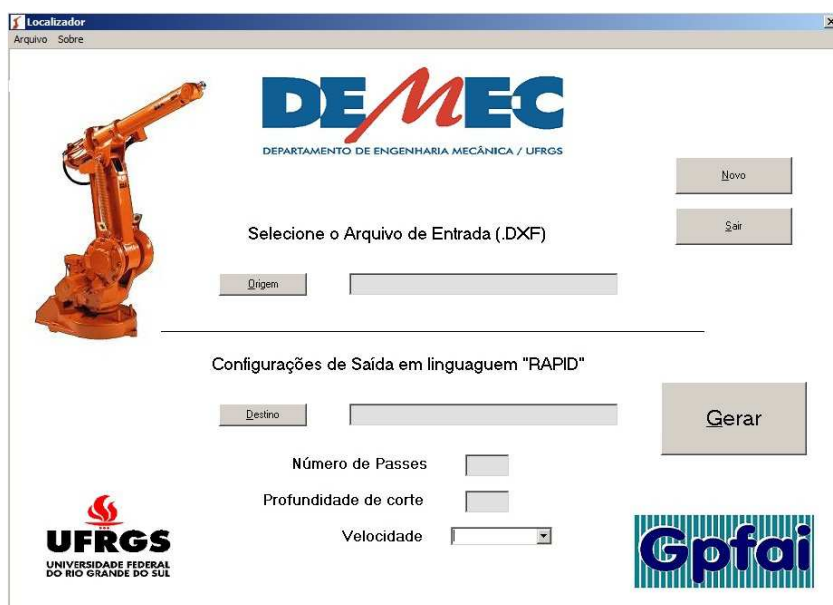


Figura 4.4– Programa gerador linguagem RAPID.

Nas instruções da linguagem do sistema robótico, os comandos para definir movimentos para ferramenta estão disponibilizados em três funções principais; respectivamente definidas como *MoveC* para trajetória circular, *MoveJ* para trajetória livre com movimentação definida pelas juntas, e *MoveL* para trajetória linear. A essas modalidades pode ser associada uma função auxiliar que permite um deslocamento incremental definido como função de *offset* (*OFFS*). Independentemente da modalidade, o comando é completado indicando a ponto de destino, a velocidade, o grau de precisão no entorno, além da especificação da ferramenta de trabalho. A sintaxe desses comandos é descrita com os respectivos parâmetros a seguir:

### Movimento Linear (*MoveL*)

*MoveL ToPoint, speed, zone, Tool;*

- *ToPoint*, é a posição de destino, definida como um nome da variável.
- *speed*, é a velocidade, em milímetros por segundo, a qual o robô executa o movimento, definida como uma variável com atributos de deslocamento linear e angular.
- *zone*, é um valor predeterminado que define a precisão, definida como um valor de raio que especifica uma região entorno da posição da trajetória. Permite quantificar o desvio do ponto definido para uma mudança de trajetória com ou sem desaceleração (ponto de passagem ou parada). Quando o valor dessa variável é nulo vem especificado como a declaração *Fine*.
- *Tool*, corresponde a especificação da ferramenta ou efetuator que se encontra na extremidade do robô.

A figura 4.5 ilustra a execução de uma trajetória linear.

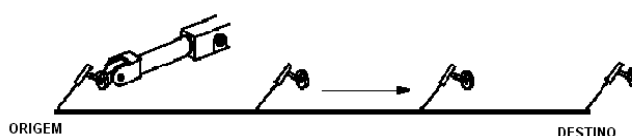


Figura 4.5 – Movimento Linear.

### Movimento Circular (*MoveC*)

Como ilustrado na figura 4.6, este movimento caracteriza-se por ser um movimento de interpolação circular, definido por o seu ponto de circularidade (*CirPoint*).

*MoveC CirPoint, ToPoint, speed, zone, Tool;*

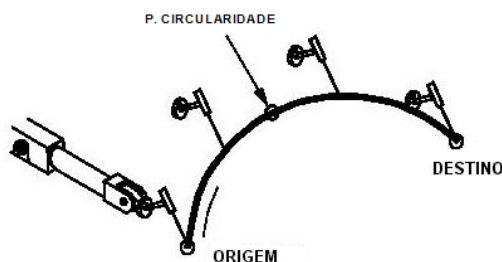


Figura 4.6 – Movimento Circular.

### Movimento Livre (*MoveJ*)

Na figura 4.7 representa-se esta opção utilizada para movimentação de trajetória livre entre um ponto e outro quando o controle da trajetória não é importante. Todos os eixos se movem com velocidade constante.

*MoveJ ToPoint, speed, zone, Tool;*

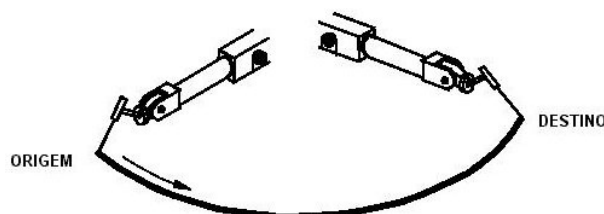


Figura 4.7 – Movimento Livre.

### Modalidade de movimento com *OFFSET*

O movimento *OFFSET* é um movimento com deslocamento incremental em cada um dos eixos cartesianos. Define os valores *dx*, *dy* e *dz* em relação ao ponto de referência “zero peça” previamente definido como variável do tipo *ROBOTARGET*.

*MoveJ OFFS (pto1, dx, dy, dz), speed, zone, Tool;*

*Exemplo:*

*MoveJ OFFS (pto1, 5, 10, 15), speed, zone, tool;*

A posição *pto1* é deslocada de 5 mm na direção *x*, 10 mm na direção *y* e 15 mm na direção *z*.

### 4.3 Teste experimental

Com finalidade de avaliar o sistema proposto, foram executados testes práticos, envolvendo os equipamentos de processo e o software necessário. Todos os estudos de teste foram realizados no laboratório de Usinagem/Robótica da UFRGS. As aplicações desenvolvidas envolvem o processo de desenho das peças, a geração de trajetórias de usinagem, a adaptação de parâmetros de corte (número de passes, profundidade de corte).

Nos testes experimentais, para geração de protótipos, utilizaram-se materiais de baixa resistência mecânica devido às limitações de rigidez impostas pelo manipulador robótico. O material testado foi poliuretano expandido. Os equipamentos de processo utilizados na realização deste projeto foram: um robô *ABB* modelo *IRB-1400* do tipo antropomorfo e uma ferramenta de corte acionada por um cabeçote pneumático *REITZ* modelo *ER-220 / 35N – 7E*. Para possibilitar o trabalho de usinagem foi preciso adaptar um mecanismo de fixação entre o robô e a ferramenta como se representa na figura 4.8.

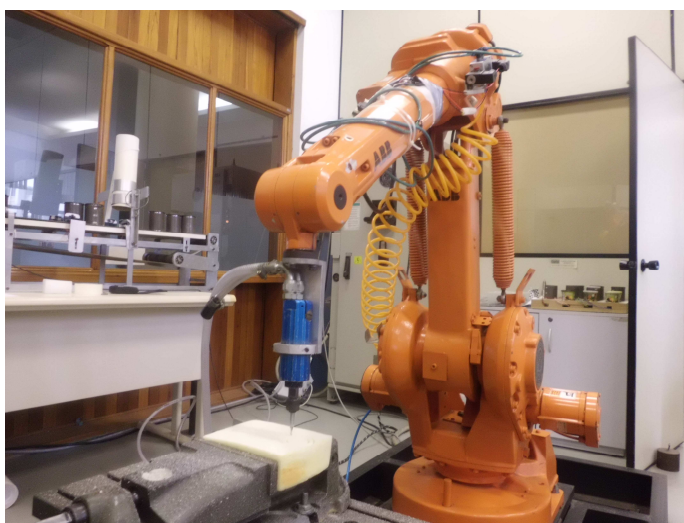


Figura 4.8 – Robô ABB com ferramenta de trabalho.

A definição das trajetórias da ferramenta são atreladas a um ponto de referência na definição do desenho. Para a execução da peça essa referência deve ser transferida como um ponto de referência para o robô no bloco da matéria-prima a ser usinada, figura 4.9.

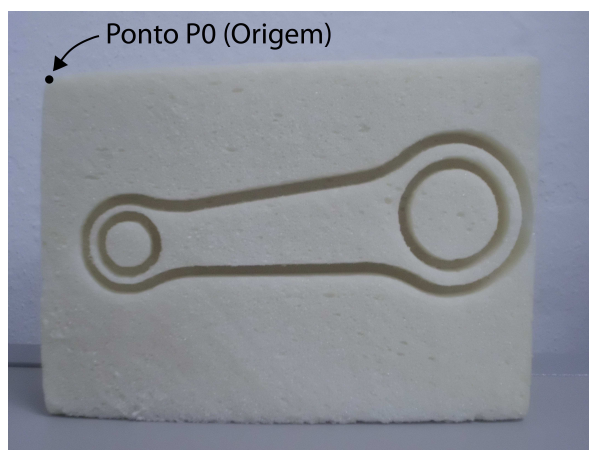


Figura 4.9 – Posição do ponto de origem.



A peça produzida, como protótipo, usinada com fresa de topo (diâmetro 3 mm), com material poliuretano expandido é ilustrada na figura 4.10.



Figura 4.10 – Geometria produzida como teste.

## 5. RESULTADOS E DISCUSSÃO

Os procedimentos experimentais permitem demonstrar a viabilidade da fabricação dos modelos geométricos propostos através da usinagem robotizada. O processamento do arquivo gerado no programa *CAM* possui as informações de trajetórias necessárias para a fabricação dos modelos através do robô.

Uma das dificuldades encontradas na execução dos procedimentos experimentais, no caso do arco (*arc*), foi decorrente da ordem na qual foi feito o desenho no programa *CAD*. Ou seja, quando no software *CAD* se faz uma sequência de arcos sucessiva, no arquivo *DXF* a mesma ordem é respeitada na sua montagem. Assim as instruções de usinagem também devem ocorrer nessa mesma ordem.

## 6. CONCLUSÕES

O resultado do estudo de caso apresentado mostra a viabilidade da aplicação do processo desenvolvido e sua integração à manufatura. Os recursos de integração permitiram a utilização de diferentes características no sistema *CAD* para a geração de um programa de usinagem em linguagem *RAPID*. A utilização destes recursos proporciona flexibilização e automação aos meios produtivos permitindo, por exemplo a realização de protótipos de grandes dimensões.

O sistema desenvolvido elimina o processo de aprendizado da linguagem *RAPID* por parte do usuário, que pode programar trajetórias e parâmetros de operação através da interface elaborada no Visual Basic.

A utilização do sistema robótico, para operações de usinagem, apresenta como vantagens na sua concepção mais ampla, com utilização de até seis graus de liberdade para movimentação da ferramenta, a possibilidade de geração de superfícies complexas. Essas possibilidades não foram exploradas neste trabalho, pois trata-se de um protótipo especial limitado a usinagem a duas dimensões.

Novos testes deverão ser executados com materiais alternativos para avaliação de acabamento e precisão dimensional, na continuidade do trabalho.



## 7. SUGESTÕES PARA TRABALHOS FUTUROS

O objetivo desse trabalho foi limitado a implementar um sistema de conversão de dados de desenho, em trajetórias de ferramenta para o robô executar a usinagem de uma peça por fresamento. Nesse sentido foram executados poucos testes experimentais não sendo avaliados problemas decorrentes de materiais nem aspectos de precisão dimensional. Assim, para trabalhos futuros recomenda-se:

- considerar compensação de raio de ferramenta para assegurar precisão dimensional;
- realizar testes com diferentes materiais para avaliar a usinabilidade no processo robotizado;
- trabalhar com diferentes padrões de arquivos gráficos, como *STEP*, *IGES*, etc;
- pode-se estender o sistema de conversão da linguagem *RAPID* a outras linguagens robotizadas.

## REFERÊNCIAS BIBLIOGRÁFICAS

ABB Robótica; “**ABB IRB 1400 User’s Guide**”, 1995.

Bedworth, D. D.; “**Computer Integrated Design and Manufacturing**”, McGraw-Hill, 1991.

Groover M.P.; Zimers E. W. Jr., “**CAD/CAM Computer Aided Design and Manufacturing**”, Prentice-Hall, 1984.

Groover M.P.; Weis M.; Nagel R. N.; Odrey N.G., “**Industrial Robotics Technology, Programming and Applications**”, McGraw-Hill, 1986.

Lorini, F. J.; “**Robótica Industrial**”, Notas de aula, Universidade Federal do Rio Grande do Sul, 2012.

Matsumoto, É. Y. “**Autolisp 2002 Linguagem de Programação do AutoCAD**”, Erica, 2002.

Meneguelo, G.; “**Aplicação de um sistema robótico utilizando recursos de sistemas CAD/CAM para o processo de fresamento**”. Dissertação de mestrado, Universidade Federal Rio Grande do Sul, 2003.

Niku, S. B.; “**Introduction to Robotics Analysis, Systems, Applications**”, Printice-Hall, 2001.

Perry, G.; “**Visual Basic 6**”, Campus, 2000.