

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

**Aplicação de um Sistema Robótico Utilizando Recursos de Sistemas  
CAD/CAM para o Processo de Fresamento**

Por

Gustavo Pizarro Meneghello

Dissertação para obtenção do Título de  
Mestre em Engenharia

Porto Alegre, Outubro de 2003

# **Aplicação de um Sistema Robótico Utilizando Recursos de Sistemas CAD/CAM para o Processo de Fresamento**

Por

Gustavo Pizarro Meneghello

Engenheiro Mecânico

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

Mestre em Engenharia

Área de Concentração: Processos de Fabricação

Orientador: Prof. Dr. Flávio José Lorini

Comissão de Avaliação:

Prof. Dr. Carlos Alberto Costa

Prof. Dr. Eduardo André Perondi

Prof. Dr. Leandro Costa de Oliveira

Prof. Dr. Jun Sérgio Ono Fonseca

Coordenador do PROMEC

## RESUMO

Sistemas produtivos industriais podem combinar, em células de manufatura, diferentes características de automação de processos permitindo interfaceamento e possibilitando a flexibilização e otimização da manufatura. Nos processos de fabricação, as dificuldades para a execução de processos de usinagem convencional ou mesmo através de comando numérico, quando se dispõe de máquinas adequadas para a fabricação de peças de geometrias complexas, podem limitar a criatividade, bem como dificultar o desenvolvimento do processo produtivo de forma otimizada. As dificuldades encontradas vão desde fixação das peças, necessidades de dispositivos especiais, restrições nos volumes de trabalho de máquinas CNC e mesmo a geração de trajetórias complexas para máquinas com limitações de eixos. Este trabalho propõe uma metodologia alternativa para usinagem, especialmente fresamento, com desenvolvimento de um algoritmo capaz de, através da utilização de recursos de sistemas CAD/CAM, traduzir arquivos gráficos para uma linguagem de programação utilizada em robôs. Na aplicação utilizam-se recursos dos sistemas CAD/CAM para gerar, numa primeira etapa, o código de programação para máquinas-ferramenta de comando numérico e posterior adaptação, para aplicação em sistemas robóticos.

Informações de caminhos de ferramenta para usinagem em máquinas CNC são convertidos, através de uma interface computacional, em trajetórias a serem seguidas por uma ferramenta guiada por um manipulador de robô industrial. Os parâmetros de processo são também adequados as restrições dos sistemas robotizados. A viabilidade do sistema proposto é confirmada através de testes realizados em modelos de superfícies complexas, onde o objetivo do referido trabalho foi alcançado.

## **ABSTRACT**

### **Application of a Robotic System Using CAD/CAM Systems Resources For The Milling Process**

Industrial productive systems can combine in manufacture cells different profiles in process automation allowing interfacing and making possible the flexibilization and optimization of manufacture process. Regarding the manufacturing processes, the difficulties to conventional process machining execution or even through numerical command when making available the right machines to manufacturing for complex shapes might bring limits to the creativity as well difficulties to the development of the productive process in an optimized way. The difficulties found out go from part fixing, needs of specials fixturing elements, restrictions on workspace for cnc machines and even the generation of complex tracks to machines with axle limitations. This assignment proposes an alternative methodology to machining, specially milling by developing an algorithm able to, through the application of CAD/CAM system resources, transform graphical files into instructions programming language applied in robots. The application uses CAD/CAM system resources to generate the programming code in numerical command to tool-machines. The next conversion occurs through a computational interface able to interpret the instructions generated by a CAM system.

The parameters and tracks for machining in CNC machines are converted into tracks to be followed by a tool guided by an industrial robot manipulator. The process parameters are converted in a way to adapt themselves to the restrictions of robotics systems. The possibility of the system offered is confirmed through tests applied in complex surface models where the objective of the assignment was achieved.

## ÍNDICE

1. INTRODUÇÃO	01
2. COMANDO NUMÉRICO	05
2.1 Sistema CN	06
2.2 Componentes de um Sistema de Comando Numérico	07
2.3 Tipos de controle em sistemas de comando numérico	08
2.4 Sistemas de Coordenadas	10
2.5 CNC	11
2.6 DNC	12
2.7 Programação CN	13
3. SISTEMAS CAD/CAM	20
3.1 A computação Gráfica	20
3.2 Tecnologia CAD	22
3.3 Integração entre sistemas e Padrões Gráficos	26
3.4 Sistemas CAM	33
4. ROBÓTICA INDUSTRIAL	36
4.1 Robô e Robótica	36
4.2 Aplicações de Robôs	37
4.3 Elementos Básicos	38
4.4 Espaço de Trabalho e Tipos de Manipuladores	47
4.5 Programação de robôs	52
5. DESENVOLVIMENTO DO SISTEMA SisPRob	58
5.1 Características Gerais do Sistema	58
5.2 A conversão do Padrão ISO para o Código RAPID	62
5.3 Implementação Computacional	76

6. APLICAÇÕES DO MODELO PROPOSTO	79
6.1 Características Gerais	79
6.2 Material Usinado	80
6.3 Equipamentos	80
6.4 Procedimento Experimental	81
6.5 Parâmetros de Operação	82
6.6 Estudo de Caso 1: Peça Exemplo 1	83
6.7 Estudo de Caso 2: Peça exemplo2	85
6.8 Estudo de Caso 3: Peça exemplo 3	87
6.9 Estudo de Caso 4: Peça exemplo 4	88
6.10 Resultados de Medições	89
7. CONCLUSÕES	90
7.1 Resultados e Discussões	90
7.2 Conclusões	91
8. TRABALHOS FUTUROS	93
9. REFERÊNCIAS BIBLIOGRÁFICAS	94
ANEXOS	97
Anexo 01 – Programa CN para execução da peça exemplo 2	97
Anexo 02 – Programa RAPID para execução da peça exemplo 2	103
Anexo 03 – Projeto da peça exemplo 2	110

## LISTA DE SÍMBOLOS

ABNT	Associação Brasileira de Normas Técnicas
AML	A Manufacturing Language
APT	Automacally Programmed Tools
BLU	Basic Length Unit
CAD	Computer Aided Design
CADCAM	Computer Aided Design/Computer Aided Manufacturing
CAE	Computer Aided Engineering
CAI	Computer Aided Inspection
CAM	Computer Aided Manufacturing
CAPP	Computer Aided Process Planning
CAX's	Computer Aided systems
CIM	Computer Integrated Manufacturing
CLData	Cutter Location Data
CLFile	Cutter Location File
CLP	Controlador Lógico Programável
CLU	Control Loops Unit
CRT	Tubo de Raios Catódicos
CN	Controle Numérico
CNC	Controle Numérico Computadorizado
DAC	Design Augment by Computer
DNC	Direct Numerical Control
DPU	Data Processing Unit
DWG	Data Exchange Drawing
DXF	Data Exchange File
ESPRIT	European Strategic Programme for Research into Information Technology
GPFAI	Grupo de Projeto e Fabricação em Automação Industrial
GT	Group Technology
IGES	Initial Graphics Exchange
ISO	Organização Internacional de Padronização
JARA	Japan Robot Association
MCL	Manufacturing Control Language
MCU	Machine Control Unit

MIT	Massachusetts Institute of Technology
MRPII	Manufacture Resource Planning
NC	Comando Numérico
PC	Personal Computer
RIA	Robotics Industries Association
SCARA	Selective Compliance Arm of Robot Assembly
SET	Standard d'Exchange Transfert
SIGGRAPH	Special Interest Group for Graphics
STEP	Standard for the Exchange of the Product Model Data
SISPROB	Sistema Pós-Processador para Robôs
STL	Stereolithography
TCP	Tool Central Point
UFMG	Universidade Federal de Minas Gerais
UFRGS	Universidade Federal do Rio Grande do Sul
UFSC	Universidade Federal de Santa Catarina
USP	Universidade de São Paulo
VAL	Victor's Assembly Language

## ÍNDICE DE FIGURAS

Figura 2.1	Sistema de coordenadas para Máquina-ferramenta.....	11
Figura 2.2	Fluxo de informações em DNC.....	13
Figura 2.3	Ambiente APT.....	18
Figura 3.1	Troca de dados sem formato padrão.....	27
Figura 3.2	Troca de dados com formato padrão.....	27
Figura 4.1	Tipos de juntas.....	41
Figura 4.2	Espaço de trabalho - Configuração cartesiana.....	47
Figura 4.3	Espaço de trabalho – Configuração cilíndrica.....	48
Figura 4.4	Espaço de trabalho – Configuração esférica.....	48
Figura 4.5	Espaço de trabalho – Configuração SCARA.....	48
Figura 4.6	Espaço de trabalho - Configuração antropomórfica.....	48
Figura 4.7	Robô Cartesiano.....	49
Figura 4.8	Robô Cilíndrico.....	50
Figura 4.9	Robô Esférico.....	50
Figura 4.10	Robô SCARA.....	50
Figura 4.11	Robô Antropomorfo.....	51
Figura 5.1	Estrutura Geral do Sistema Implementado.....	59
Figura 5.2	Sistema Robótico.....	65
Figura 5.3	Identificação de parada ou zona de passagem.....	66
Figura 5.4	Sistemas de coordenadas.....	67
Figura 5.5	Sistema de coordenadas da ferramenta.....	67
Figura 5.6	Sistemas de coordenadas (a) Fresadora CNC (b) Sistema Robótico.....	70
Figura 5.7	Segmentação da trajetória circular.....	71
Figura 5.8	Referenciamento em linguagem RAPID.....	74
Figura 5.9	Vetor unitário de orientação da ferramenta.....	75
Figura 5.10	Tela do SisPProb.....	76
Figura 6.1	Ferramenta pneumática.....	80
Figura 6.2	Mecanismo de fixação da ferramenta.....	81
Figura 6.3	Unidade processadora 3D.....	81
Figura 6.4	Procedimento experimental.....	81
Figura 6.5	Peça exemplo 1 – Modelo em CAD.....	83

Figura 6.6	Simulação das estratégias implementadas para peça exemplo 1.....	83
Figura 6.7	Referência zero peça.....	84
Figura 6.8	Operação de desbaste.....	84
Figura 6.9	Etapa intermediária de desbaste.....	84
Figura 6.10	Início de operação de acabamento.....	84
Figura 6.11	Etapa Intermediária de acabamento.....	84
Figura 6.12	Peça exemplo 1 concluída.....	84
Figura 6.13	Peça exemplo 2 – Modelo em CAD.....	85
Figura 6.14	Simulação da estratégia de usinagem implementada para peça exemplo 2..	85
Figura 6.15	Ciclo fixo de furação.....	85
Figura 6.16	Equipamento de testes experimentais.....	86
Figura 6.17	Início do processo de desbaste.....	86
Figura 6.18	Operação de desbaste.....	86
Figura 6.19	Execução de cavidade.....	86
Figura 6.20	Peça exemplo 2 concluída.....	86
Figura 6.21	Peça exemplo 3 – Modelo em CAD.....	87
Figura 6.22	Simulação das estratégias de usinagem.....	87
Figura 6.23	Início de operação.....	87
Figura 6.24	Operação de desbaste.....	87
Figura 6.25	Operação de acabamento.....	88
Figura 6.26	Fase intermediária de acabamento.....	88
Figura 6.27	Fase final de acabamento.....	88
Figura 6.28	Peça exemplo 3 concluída.....	88
Figura 6.29	Acabamento em modo 5 eixos.....	88
Figura 6.30	Execução de trajetória 5 eixos.....	88

## ÍNDICE DE TABELAS

Tabela 2.1	Símbolos utilizados para definição de parâmetros no código ISO 66025.....	15
Tabela 2.2	Algumas funções da norma ISO 66025.....	15
Tabela 3.1	Formato de arquivos de armazenagem e transferência de dados entre sistemas CAx's.....	28
Tabela 3.2	Representação de um arquivo IGES.....	29

Tabela 3.3	Exemplo de definição de uma entidade em formato STEP.....	32
Tabela 4.1	Marcos históricos da robótica industrial.....	36
Tabela 4.2	Fabricantes de robôs e linguagens de programação.....	56
Tabela 5.1	Principais funções do comando numérico ROMI – MACH8.....	64
Tabela 5.2	Funções da Linguagem RAPID.....	69
Tabela 5.3	Correlação das funções de movimentação ISO x Linguagem RAPID.....	72
Tabela 6.1	Parâmetros utilizados na operação do sistema robótico.....	82
Tabela 6.2	Registro de Medições.....	89

## 1. INTRODUÇÃO

O atual processo evolutivo dos componentes eletrônicos bem como dos computadores e microprocessadores possibilita grande evolução no que se refere à automação industrial. A generalização da utilização de computadores no processo de manufatura permitiu o auxílio destes nas funções de controle e gerenciamento de produção. Estes sistemas computacionais aliados à constante evolução das tecnologias de informação têm proporcionado o desenvolvimento de sistemas fabris integrados denominados “Fábricas do futuro”. A utilização de computadores em diferentes funções da manufatura tem crescido rapidamente nas mais diversas aplicações, seja desde o controle de estoques, na programação de comando numérico, no planejamento e controle da produção, até o monitoramento de máquinas e gerenciamento de sistemas de informação e integração de tarefas.

O desenvolvimento de sistemas computacionais de auxílio à manufatura iniciou com a evolução da computação gráfica. Sua história começa na década de 50 juntamente com o desenvolvimento das máquinas de comando numérico. Os recursos de auxílio à manufatura permitem que dados do produto, como geometria e dimensões, gerados a partir de sistemas CAD, possam ser usados diretamente na geração de programas de comando numérico para máquinas CNC, planejamento de processo e controle de produção.

O grau de sofisticação dos computadores e atuais tecnologias de comunicação permitem obter uma fábrica totalmente interligada e informatizada onde a automação é aplicada a todos os níveis da indústria e, através de sistemas eletrônicos e computadorizados integrados às máquinas e sistemas mecânicos, possibilitam o controle e a manipulação ampla sobre o processo de fabricação.

Atualmente, as limitações para a concepção de ambientes de manufatura automatizados não estão na tecnologia propriamente dita, mas na dificuldade de se fazer a integração de sistemas. Integração esta, fundamental para o desenvolvimento de ambientes fabris flexíveis e automatizados capazes de se adaptar a confecção de diferentes produtos possibilitando a otimização de processos de fabricação através da reprogramabilidade de máquinas e equipamentos.

A automação dos processos de fabricação é uma realidade que deve ser enfrentada pela indústria mundial, acelerando a competitividade em todos os setores, aumentando a produtividade, diminuindo custos e possibilitando novas perspectivas de produção, afetando a empresa em todos os aspectos produtivos.

Inserida num contexto mundial, a automação industrial é um dos caminhos para garantir um nível de competitividade que permitirá as empresas manterem-se ou até mesmo conquistar novos mercados. Como parte deste contexto pode-se citar a indústria metal-mecânica na qual os processos de usinagem de materiais são atividades primárias de desenvolvimento.

Parte das dificuldades encontradas nos processos de fabricação por usinagem convencional atualmente estão na fabricação de produtos de geometrias complexas e grandes dimensões. Nestes processos, pode-se citar dificuldades como problemas de fixação de peças, volumes de trabalho de máquinas CNC e geração de trajetórias complexas para máquinas de três ou mais eixos.

Desde a concepção dos produtos existe a preocupação quanto às possibilidades de fabricação de modo a atingir um custo compatível com a realidade do mercado além de qualidade satisfatória. Estas preocupações, muitas vezes, limitam a criatividade e a solução de problemas bem como dificultam o desenvolvimento do processo produtivo.

A partir da necessidade de desenvolver processos produtivos capazes de reduzir as dificuldades encontradas na fabricação de produtos de geometrias complexas e proporcionar flexibilização de máquinas e equipamentos, propõe-se neste projeto de pesquisa, a usinagem de superfícies complexas com auxílio de robô, especialmente na geração de protótipos ou peças especiais. A aplicação de um robô com seis graus de liberdade pode possibilitar a obtenção de geometrias complexas explorando os seis eixos para posicionamento e orientação de uma ferramenta de trabalho.

Fora das aplicações clássicas, poucos estudos têm sido realizados no sentido de automatizar o processo de programação de sistemas robóticos para utilização de robôs em diferentes funções. Dentre as áreas de pesquisa relacionadas ao tema deste trabalho pode-se citar a geração de algoritmos matemáticos para controle de movimentação através de juntas cinemáticas, a teleoperação de robôs via *Web* e o desenvolvimento de linguagens gráficas de programação de robôs orientadas a objetos.

Na área de teleoperação de robôs, um trabalho realizado na UFMG [14] apresenta uma aplicação desenvolvida para *Web* onde um manipulador de seis graus de liberdade esboça a caricatura de uma face contida numa imagem enviada via internet. Neste trabalho a idéia é permitir que uma pessoa envie uma imagem e monitore os movimentos de um robô via *Web*. O processo inicia com o processamento da imagem captada onde são utilizadas técnicas como detecção de bordas, afinamento e vetorização. Numa etapa seguinte são gerados dados e enviados ao controlador do robô para processamento de movimentos e geração de trajetórias para

o braço robótico. Nesta etapa, os movimentos e trajetórias são otimizados levando em consideração pontos de singularidade e área de trabalho [14].

Em 1997, Arai, Yago [2] apresentaram o trabalho de uma linguagem gráfica de programação de robôs que iniciou seu desenvolvimento em 1994 como um sistema protótipo na JARA – Japan Robot Association. O sistema consiste de uma interface gráfica amigável e de fácil aprendizado para edição de programas de robôs para soldagem por arco elétrico em meios produtivos de pequenos lotes. O objetivo deste trabalho era produzir um meio de programação de robôs utilizando uma plataforma de computadores pessoais (PC) com uma interface gráfica de alta performance.

Na área de programação de trajetórias de movimentação de robôs para usinagem pode-se citar alguns trabalhos desenvolvidos que tinham como objetivo detectar e evitar colisões de ferramentas com a peça e otimizar a trajetória a ser percorrida. Neste sentido, trabalhos como os de Benchetrit, Lenz e Shohan [5] e Kuczogi [21] contribuíram para o desenvolvimento desta área de pesquisa trazendo resultados que comprovaram a viabilidade de sua implementação. Dentro deste contexto, o trabalho de Vergeest [35] apresenta um software para geração automática de estratégias de remoção de material e planejamento de trajetória de ferramenta denominado MAOS. O trabalho foi desenvolvido em linguagem de programação C++, e tinha o objetivo de explorar a viabilidade da prototipagem rápida utilizando um robô. O trabalho apresenta testes virtuais e físicos que demonstram a viabilidade da geração automática de estratégias de remoção de material e planejamento de trajetórias de ferramenta. O método foi desenvolvido baseado na subdivisão de problemas em acessibilidade da ferramenta e planejamento de trajetória [37] onde um conjunto diminuto de orientações de ferramentas é determinado e o processo de remoção de material é realizado com a ferramenta muito próxima da peça permitindo a usinagem em regiões complexas [37]. Este trabalho foi implementado em uma célula de sete graus de liberdade denominada *Sculpturing Robot*. O método computacional apresentado demonstra a viabilidade de uma automação do processo de planejamento de trajetórias de usinagem para a fabricação de formas especiais.

Notoriamente, grande parte do complexo fabril nacional não dedica tempo para pesquisa e recursos financeiros para o desenvolvimento de tais sistemas ou até mesmo infra-estrutura para aplicações do gênero. Para tanto, é necessário utilizar-se dos recursos de integração disponíveis para o desenvolvimento de sistemas de automação customizáveis adaptando as necessidades do processo as características das máquinas e equipamentos disponíveis.

Neste sentido, o objetivo deste trabalho é desenvolver um sistema para auxiliar na geração de programas de instruções para robôs para sua utilização em operações de usinagem, mais especificamente fresamento, a partir de informações geométricas da peça a ser produzida, utilizando recursos de sistemas CAD/CAM.

O trabalho consiste da utilização de um robô industrial (ABB IRB 1400) existente no Laboratório de Robótica do Departamento de Engenharia Mecânica da Universidade Federal do Rio Grande do Sul para operações de usinagem de moldes ou modelos. A proposta deste trabalho é associar a geometria das peças às possibilidades de trajetórias que podem ser definidas com o robô do tipo antropomorfo, em termos de posicionamento e orientação para a ferramenta de trabalho utilizada.

Dentro deste contexto é realizada uma abordagem das principais tecnologias pertinentes com a pesquisa e desenvolvimento do trabalho apresentado, envolvendo fundamentos de robótica industrial; programação de robôs e máquinas de comando numérico, bem como uma abordagem de sistemas CAD/CAM. Esses tópicos são tratados nos capítulos I, II e III como fundamentação tecnológica para o sistema desenvolvido neste trabalho apresentado no capítulo IV.

## 2. COMANDO NUMÉRICO

O termo CN (Comando Numérico) refere-se a um dispositivo eletrônico que pode ser programado para coordenar a operação de uma máquina. Atualmente, este controle de máquinas é realizado através da utilização de microprocessadores capazes de receber instruções de programas, processar estas informações de controle, enviar sinais e receber respostas através de transdutores ligados às máquinas.

Pode-se definir CN como um sistema através do qual ações de um determinado equipamento são controladas pela interpretação de dados alfanuméricos e símbolos. Estes dados são codificados em um formato apropriado que define um programa de instruções para a fabricação de uma determinada peça ou execução de uma tarefa. Groover [18] define CN como uma forma de automação programável na qual o processo é controlado por números, letras e símbolos.

A tecnologia CN pode ser aplicada a várias atividades, desde o auxílio ao desenho e montagem até a fabricação de determinado componente. A utilização do CN em processos de fabricação pode ser dividida em máquinas-ferramenta e máquinas que não atuam como ferramenta [33]. Como exemplo do primeiro caso têm-se tornos mecânicos, fresadoras e máquinas de eletroerosão. Para o segundo caso tem-se aplicações como montagem, desenho e inspeção.

Com a utilização de equipamentos CN, pode-se obter uma seqüência de operações a serem realizadas pela máquina, escritas em linguagem própria desenvolvida para estes equipamentos denominada programa de Comando Numérico. Estes programas podem ser inseridos de modo direto (manual) ou indireto nas máquinas com auxílio de dispositivos de entrada de dados.

A busca por racionalizar e automatizar processos é um interesse comum na sociedade desde antigas civilizações. No final do século XIX surgiam as primeiras concepções de máquinas de comando numérico através de teares que poderiam ser programados para produzir um mesmo padrão repetidamente. Estas máquinas obtinham precisão e qualidade maiores que as manuais devido à programação em fitas largas de papel, que continham por meio de furos de pinos, instruções de comando.

Em meados dos anos 40, ainda durante a II Guerra Mundial [20], os processos de fabricação evoluíram tecnologicamente principalmente devido à necessidade de produção em larga escala e em tempo reduzido de peças e produtos para fornecimento a frentes de batalha.

Havia a necessidade de produção acelerada, de qualidade e precisão, fatores que forçaram os meios industriais a investir em desenvolvimento de alternativas tecnológicas.

O CN atualmente conhecido surgiu no final dos anos 40 através de John T. Parsons, que sugeriu um método de controle automático de máquina para uma fresadora na geração de perfis suaves. Um dos primeiros trabalhos neste sentido iniciou com a união da Força Aérea norte-americana, a empresa Parsons e o MIT (*Massachusetts Institute of Technology*), na utilização de uma máquina-ferramenta, fresadora de três eixos, substituindo os antigos controles e comandos convencionais por comando numérico dotado de leitora de fita de papel perfurada, unidade de processamento de dados e servomecanismos nos eixos. Em 1952 surge o servomecanismo de comando numérico [20]. A seqüência de desenvolvimento da tecnologia de comando numérico acompanhou a evolução da tecnologia dos computadores onde máquinas-ferramenta de CN passaram a incorporar microprocessadores e sistemas eletrônicos em sua estrutura de controle. A primeira máquina-ferramenta instalada no Brasil foi em 1967 e no ano de 1972, as primeiras máquinas-ferramenta de CN começaram a ser produzidas no país [25].

## 2.1 SISTEMA CN

O propósito da utilização de informações codificadas numericamente na automação de controles de equipamentos está em monitorar e controlar as operações da máquina, permitindo flexibilidade, precisão e repetibilidade no processo. Na utilização de CN em máquinas-ferramenta tem-se o controle da trajetória da ferramenta e seus parâmetros de operação como velocidade e profundidade de corte, refrigeração e avanço através de adequada programação. Qualquer eventual modificação de parâmetros ou de processos pode ser realizada por meio da edição de programas CN. A utilização do CN objetiva o controle do posicionamento da ferramenta, o controle e monitoramento da direção e velocidade da ferramenta de modo a prever sua trajetória e o controle de funções específicas como fluído de corte e avanço [23].

Dentre as vantagens obtidas na utilização de sistemas de CN destacam-se:

- Flexibilidade de produção;
- Repetibilidade na fabricação e redução de imprecisões;
- Possibilidade de edição e modificação de programas, minimizando a ocorrência de erros em processamento;
- Simulação de operações;
- Redução nos tempos de setup, diminuindo tempos de fabricação;

- Integração ente sistemas (CAD, CAM, CAx's, CIM).

## 2.2 COMPONENTES DE UM SISTEMA DE COMANDO NUMÉRICO

Um sistema de CN pode ser caracterizado por três componentes básicos [18]:

- Programa de instruções;
- Unidade de controle – MCU (*Machine Control Unit*);
- Máquina-ferramenta ou outro equipamento controlado;

O programa de instruções descreve, em linguagem interpretável pela unidade de controle, uma seqüência de comandos que controlam e direcionam o processo. A codificação deste programa é realizada em forma de números ou símbolos e este código é então armazenado em algum modo de entrada de dados que possa ser inserido na unidade de controle. O código recebe a denominação de programa CN.

A inserção de programas em máquinas operatrizes pode ser feita de duas formas: Manualmente (MDI) ou por conexão direta com um meio de entrada. Na entrada manual de dados, o programador escreve o código diretamente na interface de comando da máquina, enquanto que na conexão direta a máquina-ferramenta é conectada a um meio de acesso de dados como um computador (DNC).

O código pode conter informações de controle de movimentos para ferramenta ou mesa de fixação, parâmetros de operação e funções auxiliares além de mecanismos de comunicação com outros níveis da arquitetura de manufatura como diferentes células de produção automatizadas e computadores centrais de controle de produção.

A unidade de controle da máquina consiste de equipamentos eletrônicos e hardwares específicos que tem a função de ler e interpretar o código do programa de instruções e convertê-los em ações da máquina-ferramenta. Nas primeiras máquinas ferramenta, estas funções lógicas dos sistemas CN eram realizadas por circuitos de válvulas. Com a introdução da tecnologia computacional e incorporação de computadores nos sistemas CN estas funções passaram a ser realizadas por microprocessadores mais rápidos, eficientes e de menor custo substituindo as antigas válvulas e facilitando a popularização das máquinas CN passando a denominar-se CNC (*Computer Numerical Control*) [20].

A introdução de sistemas de controle digitais permitiu maior flexibilização e automação na utilização de máquinas-ferramenta, além de proporcionar a integração destas máquinas com

diferentes níveis da arquitetura de manufatura. Estes sistemas empregam pulsos elétricos que, em conjunto com motores de passo, controlam movimentos. Cada pulso de voltagem equivale à movimentação de uma unidade básica de movimentação denominada BLU (*Basic Length Unit*) em seu eixo correspondente. Quanto ao deslocamento equivalente a um BLU, este depende do sistema computacional e da resolução própria do BLU do comando utilizado. Computacionalmente, os dados são arranjados, manipulados e armazenados na forma de palavras binárias onde cada *bit* representa um BLU, para sistemas computacionais de 16, 32, 64, 128 e até mais *bits*. Assim, um sistema de 16 bits poderá representar ( $2^{16}$ ) ou 65.536 diferentes posições no eixo. Se o comprimento máximo de movimentação em uma direção for de 655 mm por exemplo, tem-se para um sistema de 16 bits uma resolução de 0,01 mm para movimentação de um BLU.

A unidade de controle ou MCU (*Machine Control Unit*) pode ser definida como o conjunto de todos os componentes eletrônicos associados ao CN que controlam diretamente o servossistema da máquina. Suas funções são coordenar a transmissão e distribuição de dados, processar as instruções de entrada e saída, interpretar e converter os dados através de operações de funções matemáticas, enviar sinais para os servomecanismos, monitorar e corrigir o sistema. Para a execução destas tarefas, a MCU conta com um *software* residente capaz de gerenciar e auxiliar os microprocessadores na realização de suas funções.

A MCU possui uma arquitetura constituída de duas unidades denominadas DPU (*Data Processing Unit*) e CLU (*Control Loops Unit*). A função do DPU é decodificar as informações recebidas, processá-las e enviar dados para o CLU, que deve fornecer um sinal informando que o segmento anteriormente recebido foi executado e que o DPU pode ler um novo bloco de informações [20]. Genericamente, os elementos típicos de uma unidade de controle são leitores de dados de entrada, *buffer* de dados, canais de saída para a máquina-ferramenta, canais de entrada de realimentação da máquina e controladores de seqüência [18].

## 2.3 TIPOS DE CONTROLE EM SISTEMAS DE COMANDO NUMÉRICO

Existem diferentes definições e interpretações em tipos de controle de comando numérico e em cada comando, diferentes tipos de controle podem ser utilizados. Dentre as classificações definidas por Groover e Zimmers [18], Bedworth [4] e Koren [20] os tipos mais comuns de controles em comando numérico são caracterizados sob os seguintes parâmetros:

- fluxo aberto e fechado;
- ponto a ponto;

- corte reto;
- controle adaptativo;
- trajetória contínua;

### *Controle de fluxo aberto e fluxo fechado*

Segundo Koren [20], todos os sistemas de controle, incluindo sistemas de CN podem ser designados como de controle de fluxo aberto ou fechado. Num sistema aberto, não há retorno de informação, ou seja, a unidade de comando utiliza uma determinada seqüência de pulsos para comandar motores de passo e movimentar a ferramenta ou mesa até a posição desejada, sem que, para isso, haja um sensor de posição para retornar a informação e monitorar o processo. A unidade de controle não tem informação sobre o efeito do sinal que produz. Já em controle de fluxo fechado, existem sensores que medem a posição e velocidade do eixo e compara-os com a referência desejada, mantendo um monitoramento da ação corrigindo erros e permitindo maior precisão.

### *Ponto a ponto*

Neste sistema de controle também denominado sistema de posicionamento, a ferramenta ou peça é movida até uma posição definida para então iniciar a operação. A velocidade e a trajetória de posicionamento não são importantes neste tipo de controle visto que durante sua movimentação não há contato da ferramenta com a peça. Um exemplo deste tipo de controle é a operação de furação, onde o posicionamento da ferramenta para a operação é realizado por esse sistema. O objetivo deste tipo de sistema de controle de equipamentos CN é mover a ferramenta para uma posição pré-definida para a realização da operação. Os parâmetros de movimentação como velocidade e trajetória durante o posicionamento da ferramenta são definidos pelo comando da máquina. Estes são os sistemas de controle mais simples, conseqüentemente, de menor custo. Entretanto, adaptam-se perfeitamente a alguns tipos de processos como furação dispensando a utilização de sistemas mais avançados de controle [18].

### *Corte reto*

Este tipo de sistema de controle pode mover a ferramenta de corte paralelamente aos eixos principais a determinadas velocidades de corte. São apropriados a operações de fresamento para fabricação de peças de formatos retangulares, capazes de realizar o controle ponto a ponto porém incapazes de combinar movimentos em mais de uma direção [18].

### *Controle adaptativo*

O termo controle adaptativo define um sistema de controle que mede certas variáveis de processo e as utiliza para controlar velocidades e ou avanços. Surgiram por volta de 1962 [18] e integram às funções de controle, funções de monitoramento e realimentação de comandos. Estes comandos são dotados de servomecanismos que além de controlar uma determinada função, realizam medições de parâmetros de controle comparando-os com valores referenciais, promovendo realimentação de instruções para correções quando necessário.

Os dispositivos iniciais de medições eram baseados em sistemas analógicos sendo atualmente, substituídos por tecnologia digital e estão integrados em sistemas CNC. Sua utilização provê maior eficiência, precisão e automação de operações.

### *Trajatória contínua*

O sistema controla movimentos em diferentes eixos além de manter a ferramenta em operação durante sua movimentação. Os movimentos em cada eixo podem ser realizados simultaneamente com diferentes velocidades. A trajetória da ferramenta é controlada continuamente para gerar a geometria desejada. A maioria das formas que podem ser matematicamente definidas são possíveis de serem geradas utilizando-se de sistemas de trajetórias contínuas. É o sistema de controle mais complexo e flexível entre os tipos de controles de máquinas ferramenta e conseqüentemente, o de custo mais elevado. Sua característica mais importante é a capacidade de interpolar movimentos em diferentes eixos para gerar a geometria desejada.

## **2.4 SISTEMA DE COORDENADAS**

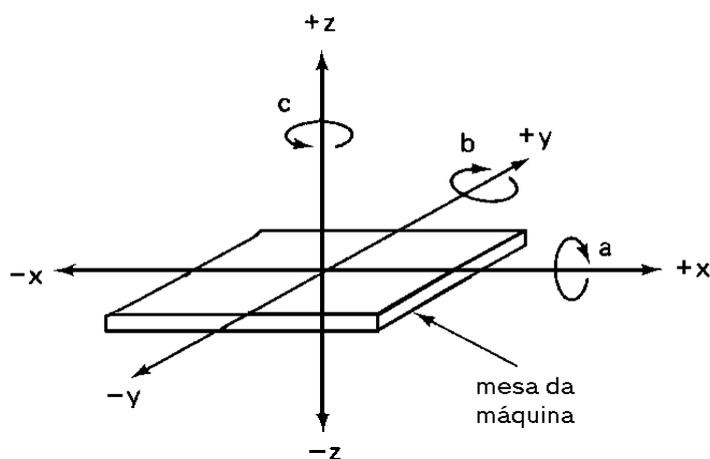
Quanto ao posicionamento, normalmente as máquinas-ferramenta apresentam sistemas de duas e três coordenadas cartesianas para posicionamento da ferramenta, bem como máquinas em que a ferramenta pode girar em um, dois ou três eixos de rotação (x, y ou z) o que proporciona ao equipamento seis graus de liberdade.

Em algumas máquinas-ferramenta, o eixo árvore executa o movimento enquanto a mesa de fixação permanece rígida. Em outras, a mesa é que executa o movimento. Existem também um número de possibilidades de orientação do sistema de coordenadas para cada equipamento, o que torna necessário estabelecer um sistema de coordenadas padrão para definição de

movimentos relativos. Com a definição de um padrão pode-se determinar uma seqüência de posições e movimentos da ferramenta de corte em relação à peça trabalhada.

Considerando-se uma máquina-ferramenta do tipo fresadora, o sistema de coordenadas de eixos é estabelecido em relação à mesa da máquina, adotando-se como referência à mesa fixa e a ferramenta móvel. Dois eixos principais ( $x$  e  $y$ ) são definidos sobre o plano da mesa de trabalho, enquanto o terceiro eixo principal é perpendicular a este plano. A movimentação na direção  $z$  é controlada pela movimentação vertical da torre que contém a ferramenta. Definem-se ainda três eixos rotacionais especificados como  $a$ ,  $b$  e  $c$ , cujos ângulos referem-se aos eixos  $x$ ,  $y$ , e  $z$  respectivamente. O sistema de coordenadas tem a função de localizar a ferramenta em relação ao espaço de trabalho.

Embora a grande maioria dos equipamentos e máquinas-ferramenta utilizarem no máximo seis graus de liberdade, considera-se ainda 14 possibilidades de eixos de movimentação para máquinas ferramenta de CN [4].



**Figura 2.1** - Sistema de coordenadas para Máquina-ferramenta [4].

## 2.5 CNC (*Computer Numerical Control*)

A evolução tecnológica computacional permitiu a utilização generalizada de computadores em muitas atividades. Computadores mais compactos e principalmente de menor custo tornaram-se atrativos para utilização em sistemas de comando numérico, proporcionando importantes avanços no início da década de 70, onde a introdução do uso de computadores no controle de sistemas de máquinas permitiu o surgimento do comando numérico computadorizado (CNC) e do comando numérico distribuído (DNC).

Por CNC entende-se um sistema computacional de controle de máquinas-ferramenta incluindo um microcomputador dedicado operando através de instruções armazenadas em sua memória para executar algumas ou todas as funções básicas do sistema de CN [20]. A utilização de microprocessadores dedicados no controle de funções CN em máquinas-ferramenta proporciona maior flexibilidade e capacidade computacional em relação aos sistemas de comando numérico.

Operacionalmente um sistema CNC inclui basicamente pelo menos três programas em seu software de controle: Programa da peça, programa de serviço e programa de controle. No programa da peça tem-se a descrição da geometria da peça a ser produzida e informações de parâmetros de operações como velocidade de corte, avanço, etc. As dimensões são expressas em números inteiros e convertidas pelo processador em correspondentes sinais de movimentação (BLU).

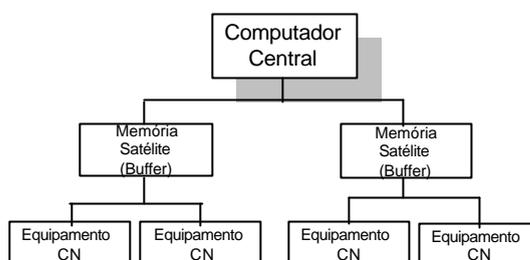
O programa de serviço disponibiliza meios para testar, editar e corrigir o programa da peça, minimizando a ocorrência de erros no processo. Quanto ao programa de controle, este recebe como dados de entrada, o programa da peça. Sua função é convertê-lo em sinais para os controladores de movimento dos eixos [20].

A interpretação e conversão dos dados em sinais de saída são realizadas através da interpolação numérica que coordena os movimentos relativos e simultâneos entre os eixos para gerar a trajetória desejada bem como controlar os parâmetros de operação. A trajetória é normalmente obtida de uma combinação de segmentos lineares e circulares de acordo com o programa de controle que deve conter sub-rotinas de interpolação linear e circular [20].

## 2.6 DNC

Com o objetivo de eliminar o processo de entrada de dados por meio de fita, em meados dos anos 60, surgiu o conceito de Comando Numérico Direto (*Direct Numerical Control*) [4]. Inicialmente a idéia era utilizar um computador no controle e processamento de informações para uma máquina-ferramenta. Porém, o custo para utilização deste tipo de controle tornou-se viável quando várias máquinas passaram a ser controladas por apenas um computador. A introdução dos conceitos de CNC e a tecnologia computacional de redes de comunicação modificaram o significado do comando numérico direto, introduzindo o conceito de controle numérico distribuído (*Distributed Numerical Control*) ou DNC.

Conceitualmente, DNC pode ser definido como um sistema de manufatura no qual um número de máquinas é controlado pelo computador através de uma conexão direta em tempo real. Neste tipo de sistema o programa da peça é transmitido do computador central diretamente à máquina-ferramenta para sua execução.



**Figura 2.2** - Fluxo de informações em DNC.

As instruções de controle e programas de peças são transmitidas via linhas de comunicação do computador central para as máquinas-ferramenta que também podem retornar informações relativas ao estado atual da máquina e dados do processo. Todo este fluxo de informações ocorre em tempo real, permitindo que toda requisição de instrução seja satisfeita quase que simultaneamente [18]. Estes sistemas operam em tempo real com um software supervisor que interliga os computadores e máquinas, controlando e estabelecendo prioridades de comunicação [20].

Atualmente, o DNC proporciona a integração das atividades de produção com diferentes sistemas de automação como CAD/CAM e sistemas gerenciadores de informação permitindo um ambiente fabril integrado. Alguns sistemas DNC permitem o comando centralizado e são utilizadas no controle de sistemas flexíveis de manufatura onde robôs e máquinas operatrizes estão presentes. A utilização do DNC proporciona maior velocidade e segurança na transferência de dados, racionalização do trabalho, controle de informações e integração.

## **2.7 PROGRAMAÇÃO CN**

Na programação CN considera-se um conjunto de símbolos, letras e números responsáveis pelo acionamento e controle de uma máquina de comando numérico. Esta programação tem a função de informar à máquina todas as etapas de determinada operação de uma peça através de uma linguagem padrão. Uma linha de comando de um programa CN pode conter informações sobre o movimento e posicionamento da ferramenta (movimento rápido, interpolação), parâmetros de operação (velocidade, avanço) e informações auxiliares (ligar

refrigerante, eixo árvore). Estes dados são obtidos da geometria da peça a ser usinada, das limitações da máquina-ferramenta e das características da ferramenta e controle CN.

Segundo Koren [20], um programa CN compreende o conjunto de todos os dados necessários para a produção da peça, além dos cálculos da trajetória da ferramenta que a máquina deve realizar e o arranjo destes dados em um formato padrão que possa ser convertido em instruções para um determinado equipamento com comando numérico. Groover [18] define a programação CN como o procedimento pelo qual a seqüência de passos de processo a serem realizados na máquina de comando numérico é planejada e documentada. O procedimento envolve a preparação do meio utilizado para transmitir instruções de processo para as máquinas-ferramenta.

Inicialmente o processo de programação CN era realizado manualmente através do trabalho de programadores que analisavam o desenho da peça e descreviam dentro de um sistema de coordenadas, dimensões e geometrias para a definição de uma trajetória de ferramenta apropriada conforme a máquina-ferramenta a ser utilizada. Com o aumento da complexidade geométrica das peças, o método manual passou a ser substituído pela programação auxiliada por computador que facilita a descrição da geometria do componente e auxilia no cálculo da trajetória e posicionamento da ferramenta. A introdução da tecnologia digital permitiu a programação de trajetórias de ferramentas para peças de geometria complexas que não poderiam ser executadas pela programação manual. A evolução da tecnologia computacional disponibilizou avanços significativos em sistemas de manufatura com o surgimento dos sistemas CAx's onde os sistemas CAM associados aos sistemas CAD possibilitaram recursos capazes de automatizar o processo de fabricação incluindo a programação CN.

Existem diferentes formas de classificação dos tipos de programação CN. Segundo Koren [20], a programação CN pode ser realizada manualmente ou com auxílio do computador, enquanto Queiroz e Stemmer [28] as subdividem em programação manual, programação do tipo APT (*Automatically Programmed Tool*), sistemas gráfico-interativos e sistemas CAD/CAM. Groover [18] divide os métodos de programação CN em programação manual da peça, programação auxiliada por computador, entrada manual de dados, programação utilizando sistemas CAD/CAM.

Embora em alguns métodos de programação auxiliada por computador, as linguagens computacionais de apoio ao processo tenham sido desenvolvidas com comandos e sintaxe próprias, o conjunto de letras, números e símbolos utilizado como programação no controle da máquina de comando numérico seguem os padrões ISO 66025 e DIN 66025. Este código é

responsável pelo acionamento e controle de uma máquina de comando numérico e sua estrutura compreende comandos de funções, parâmetros e instruções auxiliares.

O programa é estruturado em blocos ou sentenças onde as combinações destes termos formam rotinas e sub-rotinas de programa. O padrão tem letras reservadas para definição do tipo de operação, parâmetros auxiliares de operação, coordenada de movimentação e números que, junto às letras, definem a modalidade da operação, valores de coordenadas relativas aos eixos de referência e funções auxiliares.

Alguns comandos de funções e parâmetros da norma ISO 66025 são apresentados nas tabelas 2.1 e 2.2.

**Tabela 2.1** - Símbolos utilizados para definição de parâmetros no código ISO 66025.

N	Rótulo de linha
G, M	Funções
P	Parâmetros do programa
X, Y, Z	Deslocamento nos eixos cartesianos
I, J, K	Coordenadas de centro de circunferência em relação aos eixos de coordenadas
F	Avanço
T	Número de registro para troca de ferramenta
a, b, c	Eixo de rotação em torno das coordenadas X, Y, Z respectivamente
R	Raio de interpolação circular
S	Velocidade do eixo árvore principal

**Tabela 2.2** – Algumas funções da norma ISO 66025.

<b>G</b>	<b>Descrição</b>
G00	Função de posicionamento através de movimento rápido onde a ferramenta se desloca em linha reta até o ponto especificado pelos parâmetros da coordenada
G01	Movimento de interpolação linear
G02	Movimento de interpolação circular no sentido horário
G03	Movimento de interpolação circular no sentido anti-horário
G04	Tempo de espera ou atraso
G17	Seleção de plano de trabalho XY
G18	Seleção de plano de trabalho XZ
G42	Compensador do raio de ferramenta à direita

G70	Seleciona o sistema de unidades inglês (polegada)
G84	Ciclo de roscamento
G91	Define sistema de coordenadas incremental

### 2.7.1 PROGRAMAÇÃO MANUAL

O método de programação manual foi o primeiro método de programação de comando numérico utilizado. Consiste de um estudo da geometria do objeto onde o programador tem a função de definir a trajetória da ferramenta. Para preparar o programa da peça, as instruções de usinagem eram escritas em um formulário denominado manuscrito do programa da peça [18]. Este manuscrito era utilizado para a preparação do meio de entrada de dados nos sistemas CN antigos como fitas perfuradas. Atualmente, o avanço da tecnologia computacional e a introdução dos sistemas digitais na manufatura permitem que o programa da peça possa ser digitado diretamente na interface homem-máquina existente nas máquinas-ferramenta. É o que Groover [18] classifica como entrada manual de dados. Neste caso, a programação manual passa a ser a inserção direta do código de programação na máquina CN através de sua interface homem-máquina ou simplesmente a elaboração do código em um editor de texto em um sistema computacional qualquer para posterior inserção na máquina-ferramenta sem auxílio computacional para os cálculos de trajetória e geometria. Embora a programação manual ainda seja utilizada ela é reservada somente a geometrias e aplicações simples como processo de furação e aplicações de controle ponto a ponto.

### 2.7.2 PROGRAMAÇÃO AUXILIADA POR COMPUTADOR

O desenvolvimento e utilização de computação no auxílio a programação, iniciou com a generalização da utilização do comando numérico para controle de máquinas-ferramenta, e o aumento da complexidade geométrica das peças a serem produzidas [20].

Inicialmente, a utilização de computadores na programação CN limitava-se ao auxílio à realização dos cálculos de trajetória de ferramenta e descrição de geometrias mais complexas. Linguagens computacionais eram utilizadas para descrição de instruções de processamento e comunicação com o sistema de programação.

Algumas linguagens de programação de peças foram desenvolvidas na tentativa de promover uma completa descrição do processo de programação. Uma linguagem padronizada auxilia o programador a escrever o código fonte, onde a geometria da peça, o percurso da ferramenta e parâmetros auxiliares são definidos na sua sintaxe. O sistema digital processa o código fonte gerado seguindo as instruções nele contidas, realizando os cálculos geométricos e determinando o contorno da ferramenta gerando um arquivo neutro denominado CLData ou CLFile (*Cutter Location Data* ou *Cutter Location File*) independente de máquina ou pós-processador. Este arquivo é utilizado como entrada de dados num posterior pós-processamento onde o programa CN é gerado considerando-se características específicas de determinadas máquinas-ferramenta.

Segundo Groover [18], cabe ao programador a tarefa de definição da geometria da peça, especificação da trajetória de ferramenta e seqüência de operações enquanto que o processador do computador interpreta os dados de entrada (linguagem de programação) e realiza os cálculos aritméticos de trajetória e parâmetros de corte e pós-processamento necessários.

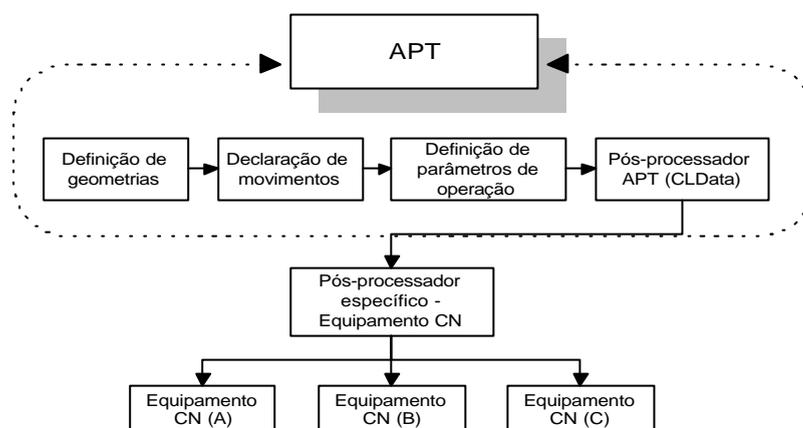
Na definição da geometria da peça, o programador utiliza-se de elementos básicos como linhas, pontos, planos, círculos, cilindros e outras superfícies matematicamente definidas para a descrição de formas complexas. Enumeram-se os elementos de que a peça é composta definindo suas dimensões e localização, além de especificar uma seqüência de movimentos entre os elementos de uma trajetória de corte [18]. O código gerado pelo programador é então tratado pelo sistema computacional para pós-processamento.

O pós-processamento tem a função de converter o arquivo neutro em um programa CN específico para uma determinada máquina-ferramenta, visto que existem diferentes tipos de comando numérico com conseqüentes diferenças em suas características e capacidades. O pós-processador CN é um software capaz de interpretar uma linguagem computacional de programação e converter os dados para o código padrão definido pela ISO 66025.

Groover [18] define pós-processadores como programas de conversão necessários para cada configuração de máquina-ferramenta, cuja função é gerar informações que podem ser diretamente aplicadas no controle de sistemas de comando numérico. A razão pela qual cada configuração particular de uma unidade de controle requer um pós-processador específico é as características individuais de cada sistema CN como especificações de tolerância e avanço programadas e seus efeitos dinâmicos no sistema.

## PROGRAMAÇÃO APT

Um dos métodos mais tradicionais de programação auxiliada por computador utiliza linguagens de programação para a descrição do código fonte. Diversas linguagens computacionais de auxílio a programação CN foram desenvolvidas desde as primeiras pesquisas realizadas no MIT em meados dos anos de 1950. Dentre elas, a APT (*Automatically Programmed Tool*) destaca-se por sua utilização e recursos, cujo esquema operacional é indicado na figura 2.3.



**Figura 2.3 - Ambiente APT.**

O APT não é somente uma linguagem de programação CN, mas também um programa que executa cálculos para gerar trajetórias de corte baseadas em informações de entidades geométricas e comandos da linguagem. Sua estrutura consiste em quatro tipos de declarações de geometria, movimentos, pós-processamento e declarações auxiliares. No primeiro tipo de declaração, são definidas as geometrias dos elementos utilizados na concepção do desenho identificando os símbolos, definindo os tipos de elementos (ponto, linha, círculo, etc.) e descrevendo dados para sua definição precisa (coordenadas, raio, etc.). Na declaração de movimentos são introduzidos comandos de movimento de ferramenta que explicitam o tipo de ação que deve ser realizada e os dados de descrição que definem a movimentação da ferramenta. As palavras reservadas do APT para definição de comandos de movimento são palavras comuns do meio industrial descritas em inglês, o que facilita sua compreensão.

Nas declarações de pós-processamento são definidos comandos para controle dos parâmetros de usinagem como rotação, avanço e velocidade de corte bem como outras características de máquinas-ferramenta como controle do fluido refrigerante e lubrificação. Para

completar a estrutura de uma programação em linguagem APT existem ainda as declarações auxiliares onde são definidos outros fatores como identificação da peça, lado a ser trabalhado, etc. O código gerado é convertido pelo sistema computacional para um arquivo neutro (CLData) para seu posterior pós-processamento. Dentre as vantagens da programação APT pode-se destacar a possibilidade de programar peças de geometria complexas com menor tempo e custo de programação.

### 2.7.3 PROGRAMAÇÃO AUTOMÁTICA POR SISTEMAS CAD/CAM

Os sistemas CAD/CAM são métodos de programação auxiliada por computador executados por algoritmos computacionais onde os dados de entrada do sistema são arquivos de transferência de dados de projeto ou percurso da ferramenta. Este sistema possibilita a interpretação da geometria das peças através da leitura de arquivos gráficos de transferência de dados e a geração de programas CN. O processo ocorre através da interatividade com o programador que permite a seleção de parâmetros de operação como tolerâncias, ferramentas e tipo de movimento em um banco de dados customizável. Com a interpretação dos dados geométricos e as informações do banco de dados resultante da interatividade, o sistema calcula e define as estratégias de usinagem como trajetórias e parâmetros de operação. Estas informações são então armazenadas em um arquivo de dados para posterior pós-processamento.

Nestes sistemas, a partir da representação geométrica da peça disponível em formato digital, é possível, além da programação, simular trajetórias de ferramentas e o processo de usinagem obtendo-se maior precisão e diminuição do tempo de programação e setup.

Os benefícios na utilização destes sistemas vão além da integração da manufatura. Sua utilização proporciona economia na definição da geometria, pois este processo passa a ser desenvolvido durante o projeto, possibilitando diminuição de erros através da simulação dinâmica do processo e a flexibilização de máquinas-ferramenta através da simples modificação da programação. A programação através de um sistema CAM é abordada de forma prática no aplicativo desenvolvido neste trabalho de pesquisa.

## 3 SISTEMAS CAD/CAM

### 3.1 A computação gráfica

O uso de imagens em sistemas computacionais teve seu início em 1950, logo após o surgimento dos primeiros computadores, quando, no *MIT (Massachusetts Institut of Technology)*, Ivan E. Sutherland reproduziu imagens simples por meio de uma interface de tubo de raios catódicos utilizando-se de um computador. Este projeto tem sua significância por ser uma das primeiras demonstrações de criação e manipulação de imagens em tempo real numa tela de CRT (Tubo de raios catódicos). Nessa época, a criação de gráficos e desenhos estava restrita apenas às instituições de pesquisa com aplicações de caráter exploratório, situação mantida até o final da década de 50. A computação gráfica pode ser definida como o 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.

No início de seu desenvolvimento, os recursos existentes de informática como hardware e software eram economicamente inacessíveis e tecnologicamente não permitiam uma expansão significativa da computação gráfica. Os computadores não tinham recursos para o uso interativo e eram desprovidos de dispositivos gráficos. Os progressos em hardware nesta área foram inicialmente lentos e pouco expressivos, destacando-se cronologicamente o surgimento dos seguintes equipamentos:

1951 – início da utilização de terminais gráficos;

1953 – “plotters”;

1958 – “light pens”;

1962 - terminais coloridos;

1964 – mesas digitalizadoras.

Em 1962, novamente Ivan E. Sutherland, contribuiu para o desenvolvimento da computação gráfica apresentando, como tese de doutoramento no MIT, um sistema de comunicação gráfica entre o homem e a máquina denominado SKETCHPAD [24]. Este sistema viria a provar a viabilidade do uso da computação como auxílio a desenhos e projetos.

Com os avanços tecnológicos na área computacional e o despertar do interesse das indústrias, que viam neste sistema um grande potencial, iniciou-se um intenso 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. Alguns projetos foram de significativa importância em meados de 1964, como o DAC-1 (*Design Augmented by Computer*) da GM, CADAM (*Computer Aided Design and Manufacturing*) da Lockheed e algumas iniciativas semelhantes como na BOEING, DOUGLAS, ROCKWELL, ROLLSROYCE, entre outras poucas empresas sempre apoiadas por um grande fornecedor de equipamentos computacionais como IBM, Control Data e DEC [18].

Com a disseminação do uso de sistemas gráficos surgiram inúmeros projetos de pesquisa e diferentes fabricantes de hardware, embora o custo desses equipamentos ainda fosse proibitivo, algumas aplicações especiais e empresas de grande capacidade de investimento começaram a implementá-lo.

Nos anos 70, surgiram os primeiros sistemas CAD também denominados sistemas 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.

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 de uma região retangular da tela [24], popularizou-se, fazendo com que a utilização destes sistemas multiplicasse as aplicações gráficas existentes proporcionando a milhões de novos usuários, acesso a editores de texto, planilhas e programas de desenho.

Entre os feitos relevantes da época, é preciso ainda citar a criação de grupos especiais de pesquisa em computação gráfica (SIGGRAPH – *Special Interest Group for Graphics*) pela ACM (*Association for Computing Machinery*) e o aparecimento do IGES (*Initial Graphics Exchange Specification*) – Padrão gráfico para especificação de formatos gráficos e linguagem para transmissão e armazenamento de dados de definição de produtos.

A década de 90 é caracterizada pela automação crescente dos processos industriais através da integração de técnicas de desenho, análise, simulação e fabricação (CAD,CAE,CAM).

Atualmente, os avanços nos sistemas de comunicação fazem com que as técnicas CAD/CAE/CAM disponibilizem a engenheiros e usuários ambientes fabris integrados. A utilização destas técnicas passa de uma opção a uma exigência no setor industrial, uma questão de sobrevivência em uma economia globalizada nas mais diversas aplicações.

Como vantagens na utilização destes sistemas gráficos pode-se citar [20]:

- síntese de grandes volumes de dados;
- facilidade na geração de imagens;
- facilidade na alteração de imagens previamente geradas, simplificando diversas atividades de projetos;
- produção de desenhos de melhor qualidade e tempo reduzido;
- base para o uso de sistemas de auxílio a projeto, análise e fabricação.

### **3.2 TECNOLOGIA CAD**

O projeto assistido por computador ou sistema CAD (*Computer Aided Design*), 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, é um fator fundamental na integração de sistemas e na automação de processos.

Como principal produto da computação gráfica na área industrial, estes sistemas gráficos tiveram seu princípio nas pesquisas de Ivan Sutherland no MIT em 1963, considerado o pai da computação gráfica e conseqüentemente dos sistemas CAD [4]. Pode-se conceituar genericamente nestes sistemas todo o envolvimento de computação nas atividades de projeto como criação, modificação, análise e otimização utilizando-se de sistemas gráficos interativos [18].

Segundo Bedworth [4], algumas pessoas interpretam o termo CAD como uma ferramenta de apoio a desenho, outras, como auxílio à análise e outras ainda, como automação total de projeto envolvendo desenho, análise e otimização. Estes sistemas são baseados em computação gráfica interativa e a utilizam para a criação, transformação e visualização de dados na forma de figuras ou símbolos. Desse modo, qualquer aplicação que inclua uma interface gráfica na realização de tarefas de engenharia se considera um software de CAD.

A utilização destes sistemas proporciona inúmeros benefícios ao processo de projeto, refletido intrinsecamente em características como melhoria da qualidade e controle do trabalho. Outros benefícios diretamente mensuráveis da implementação de sistemas CAD [18] são o aumento na produtividade de projeto onde se verifica redução de tempo e *staff* de execução devido às facilidades de concepção disponibilizadas; a diminuição de erros, devido às disponibilidades de ferramentas de análise e informações técnicas gerando precisão e

minimização de erros; grande precisão de cálculos de projeto onde bibliotecas matemáticas provêm rapidez e precisão nos cálculos; padronização de procedimentos de projeto e a formação de uma base de dados que é utilizada na integração entre sistemas.

Os benefícios dos sistemas CAD propagam-se à manufatura e ao controle de processos. A base de dados gerada no desenho é utilizada no planejamento e controle de máquinas, equipamentos e processos de fabricação. Esta troca de informações entre sistemas iniciada pela base de dados gerada nos sistemas CAD é o fundamento da manufatura integrada e, por conseguinte, da automação de processos de fabricação.

A manipulação destas informações permite a aplicação em diversas atividades tais como [18]:

- geração de instruções para equipamento de controle numérico;
- planejamento de processo auxiliado por computador (CAPP);
- listagem de montagem para produção;
- inspeção auxiliada por computador (CAI);
- programação de robôs;
- tecnologia de grupo (GT);

Dentre as etapas de projeto que utilizam a tecnologia computacional como auxílio, Groover [18] subdivide-as em quatro áreas funcionais:

- modelagem Geométrica;
- análise de engenharia;
- revisão de projeto e avaliação;
- automação de desenhos;

A modelagem geométrica através da computação gráfica interativa é a fase em que a idéia inicial de projeto toma forma física. Quanto à análise de engenharia, embora seja também uma funcionalidade dos sistemas CAD, devido a sua especificidade em cálculos de tensão-deformação, transferência de calor, uso de equações diferenciais para descrever comportamentos dinâmicos, propriedades e requisitos de projeto, um módulo especial dos sistemas CAD incorpora estas funções, o CAE (*Computer Aided Engineering*). Este módulo tem a função de desenvolver análises de engenharia no modelo e sua dissociação aos sistemas CAD só é possível devido à integração entre sistemas proporcionado pela base de dados gerada na concepção do desenho. Ou seja, o grau de especialização de módulos computacionais que auxiliam no desenvolvimento do projeto toma a forma de novos sistemas integrados. Por isso essa associação natural que se faz atualmente a sistemas CAD como ferramenta de auxílio somente a desenho.

Na revisão de projeto, novamente a tecnologia computacional aplicada a sistemas CAD proporciona a manipulação de entidades gráficas de forma a verificar interferências entre peças, problemas de montagem, espaço de trabalho, dimensões, tolerâncias, enfim, uma automação capaz de minimizar os erros de projeto e proporcionar a visualização de uma simulação do produto. A modelagem de um projeto pode assumir diferentes formas de representação nos sistemas CAD, entendida como modelagem geométrica.

### *Modelagem Geométrica*

A modelagem geométrica é uma descrição matemática da geometria de um objeto denominada modelo. Este modelo permite sua manipulação e visualização em um terminal gráfico devido ao armazenamento desta descrição como um conjunto de dados.

Em suas aplicações iniciais em meados da década de 70, os sistemas CAD disponibilizavam apenas modelagem bidimensional, sendo então utilizados principalmente como ferramenta de automação do processo de desenho. A evolução dos sistemas gráficos e suas aplicações em CAD diretamente ligadas ao desenvolvimento do hardware permitiram um modelamento tridimensional aumentando sua potencialidade de manipulação de imagens e integração com outros sistemas. Nestes sistemas tridimensionais, as modificações e transformações nos modelos são realizadas interativamente entre o sistema e o projetista [16]. A modelagem propriamente dita é efetuada através da entrada de comandos computacionais pelo projetista onde, por efeito de um modelador matemático, a imagem gráfica é construída. O processo consiste na conversão dos comandos de entrada em modelos matemáticos armazenados em arquivos de dados que, por sua vez, podem ser visualizados e manipulados graficamente por intermédio do processamento computacional. Groover [18], classifica os esquemas de representação dos modelos geométricos tridimensionais em Modelagem por malhas ou *Wire-frame* e Modelagem por sólidos. Por sua vez, McMahon e Browne [24], acrescenta ainda nesta classificação a modelagem por superfícies.

### *Modelagem por Malhas (Wire-frame)*

No esquema de representação de modelos geométricos por malhas, uma geometria é definida por uma série de vértices, arestas, linhas e curvas interligadas representando as extremidades (bordas ou limites) de um objeto. A denominação deste modelo geométrico origina-se da aparência do modelo quando visualizado na tela onde a imagem assume a forma de uma estrutura de “arames”.

Em termos de modelagem tridimensional, este esquema de representação é o que requer menor tempo e espaço de memória para seu processamento e armazenagem respectivamente. Porém, apresenta certas deficiências quando utilizados para modelar objetos de engenharia como ambigüidade de representação, limitada capacidade de calcular propriedades mecânicas ou interseções geométricas, dificuldade na aplicação de cores e limitada capacidade de formação de dados para manufatura ou análise de engenharia. Devido a estas deficiências, são geralmente utilizadas em representações de layout, resolução de alguns problemas de geometria, estabelecimento de relações espaciais para projetos e em alguns casos onde uma manipulação dinâmica do modelo é requerida.

### *Modelagem por Superfícies*

Neste tipo de representação geométrica, o modelo é representado como um conjunto de superfícies poligonais fechadas, através da especificação de algumas ou todas as superfícies do objeto. As superfícies não regradas são criadas através de algoritmos especiais (Bezier, B-spline, etc.), onde a representação envolve uma série de entidades geométricas com cada superfície formando uma simples entidade. Muitas das ambigüidades dos modelos por malhas são superadas usando este esquema de representação, com as seguintes vantagens:

- reconhecimento de perfis curvos complexos;
- possibilidade de aplicação de variações de cores e sombras nas superfícies;
- maior capacidade de simulação de trajetórias para máquinas e robôs;

### *Modelagem por Sólidos*

Considerado como um avanço em modelagem geométrica em relação à modelagem por malhas ou por superfícies, em termos de realismo para o usuário e definição para o computador, a modelagem por sólidos possibilita uma completa descrição de um objeto sem ambigüidades devido à sua descrição em relação a sua forma volumétrica. A modelagem é feita através de poliedros fechados e não mais por arestas ou superfícies associadas resultando em um contínuo físico do sistema. Alguns fatores conduzem ao uso generalizado de modeladores sólidos, dentre eles:

- limitações dos sistemas por malhas e superfícies;
- completa descrição do objeto;
- detecção de interferências;
- geração automática de secções tridimensionais;

- disponibilidades de utilização de características como cores, animação e visualização;

Seus complexos recursos gráficos exigem um maior potencial em termos de processamento e memória computacional além de permitir a automação na construção de modelos e projetos, análise tridimensional de engenharia, avaliação de interferência, automação no processo de planejamento da manufatura e maior capacidade de integração entre sistemas.

### **3.3 INTEGRAÇÃO ENTRE SISTEMAS E PADRÕES GRÁFICOS**

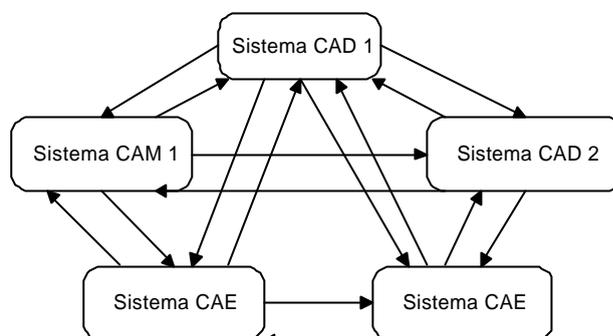
No enfoque da automação industrial, a integração entre sistemas computacionais de auxílio a projeto e manufatura iniciou com a junção da tecnologia de grupo e dos sistemas CAD. A base da tecnologia de integração de sistemas é a manipulação, armazenagem e compartilhamento de informações. Segundo Bedworth [4], 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. Ainda não há uma integração completa que dispense a intervenção humana, mas existem diferentes níveis de implementação desta tecnologia.

O processo de integração 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. A dificuldade encontra-se em tornar estes arquivos aptos ao compartilhamento e troca de informações com outros modeladores sólidos e estender sua utilização para todas as fases de desenvolvimento do produto (sistemas CAx's - CAM, CAPP, CAE).

Historicamente, a exigência de padronização de modelos de dados iniciou com a necessidade de diferentes aplicativos de sistemas CAD em compartilhar arquivos gráficos. O intercâmbio de dados durante os processos de desenvolvimento de produtos é uma realidade, a medida em que diferentes sistemas e processos são utilizados. O custo deste intercâmbio de informações resulta em tempo de processamento e custo de desenvolvimento.

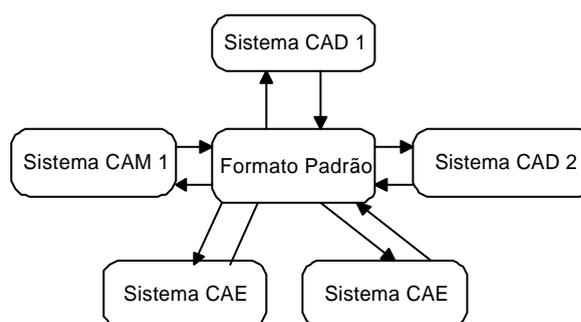
Existem duas formas de troca de informações entre sistemas, a tradução direta e a utilização de um formato neutro. A tradução direta converte a informação de um sistema diretamente para o formato nativo de outro sistema ou ainda lê os dados diretamente do arquivo nativo do sistema em que foi gerado. Sua estrutura apresenta melhores condições de captura de dados porém demanda a existência de tradutores capazes de interpretar os arquivos de outros

sistemas CAx's. A figura 3.1 apresenta uma o intercâmbio de dados entre diferentes sistemas CAx's sem a utilização de um formato padrão de transferência de dados.



**Figura 3.1** - Troca de dados sem formato padrão.

Na utilização do formato neutro têm-se uma especificação de formato de dados de domínio público, independente de software e hardware para intercâmbio de arquivos de engenharia, reduzindo a exigência de tradutores. A informação é convertida para um formato neutro para posterior tradução em formatos de sistemas específicos. A utilização de padrões de formatos reduz a necessidade de desenvolvimento de tradutores diminuindo custos de retrabalho e aumentando a qualidade das informações. A Figura 3.2 apresenta o esquema da transmissão de dados com a utilização de um padrão de formato de arquivo.



**Figura 3.2** - Troca de dados com formato padrão.

A base da integração entre sistemas é um interesse comum em todos os setores da indústria, onde uma variedade de projetos e comitês de pesquisa têm sido formados pesquisando padrões de formatos de arquivos de dados intercambiáveis. Nas últimas décadas, esforços têm sido feitos no desenvolvimento de pesquisas em estrutura de dados para sistemas CAD com atenção a formatação de padrões de arquivos de armazenamento de dados de modelos

geométricos para sua manipulação e transferência entre aplicativos correlacionados. Segundo Bedworth [4], a formatação de padrões de arquivos objetiva:

- permitir a comunicação de dados entre dois sistemas sem perda de informações;
- codificar os dados de modo tão compacto quanto possível para minimizar os custos de transmissão;
- Evitar processamentos complexos nos sistemas de envio e recebimento de informações para minimizar custos de desenvolvimento de processos;

A meta ideal para guiar o desenvolvimento de padrões de formato de dados é torná-los comum a todas as aplicações. Estes arquivos devem ter uma formatação capaz de ser interpretada por qualquer aplicativo através da conversão de formatos particulares de cada sistema em um formato “neutro” intercambiável, quando não aplicável uma tradução direta. Várias formatações de arquivos surgiram com o objetivo de solucionar a falta de um padrão “neutro” para a transferência de dados, sendo os principais indicados na tabela 3.1.

**Tabela 3.1** - Formato de arquivos de armazenagem e transferência de dados entre sistemas CAX's.

FORMATO	SISTEMA CAX'S	EXTENSÃO DO ARQUIVO	DIMENSÃO	COMENTÁRIO
ASCII		.txt / .doc	2 e 3 D	Converte as coordenadas bi e tridimensionais em pontos, linhas ou <i>splines</i> .
IGES	Padrão	.igs	3D	Padrão utilizado por grande parte dos sistemas CAD.
DXF	AutoCAD	.dxf	3D	Utilizado pelo <i>software</i> AutoCAD. É considerado um padrão “ <i>de facto</i> ” devido a sua generalizada utilização.
VDA	Padrão Alemão	.vda	3D	Padrão alemão utilizado pela indústria automobilística.
SET	Padrão Francês			Padrão francês utilizado pela indústria automobilística.
ACIS		.sat	3D	Núcleo de modelo geométrico adotado por vários sistemas CAX's.
DWG	AutoCAD	.dwg	2D	Formato criado pela <i>Auto Desk</i> para arquivos de desenho 2D.
STL		.stl	3D	<i>Stereolithography files</i> utilizado na prototipagem rápida.
TIF		.tif	2D	<i>Tag Image File Format</i> utilizado para armazenamento de imagens
CADL	Cadkey	.cdl	3D	Formato utilizado pela <i>CADKEY</i> . como forma de transferência de dados entre diferentes sistemas CAX's.

Em suas primeiras versões, as especificações de arquivos para transferência de dados contemplavam apenas dados geométricos. Alguns destes padrões se destacam por sua generalizada utilização, embora não exista um consenso em torno de um padrão único.

***IGES (Initial Graphics Exchange Specification)***

Desenvolvido em 1979 pelo *US National Bureau of Standards*, o IGES foi um dos primeiros trabalhos significativos em troca de dados, adotado como padrão em 1981 e atualmente controlado por uma organização denominada IGES/PDES. O *National Bureau of Standards* coordena trabalhos de diversos comitês e usuários no desenvolvimento do IGES desde 1980. Seu formato “neutro”, independente de pacotes gráficos é de domínio público e contém informações de geometria, estrutura e descrição de entidades [4]. O IGES foi desenvolvido com o objetivo de auxiliar a troca de dados entre diferentes sistemas CAD. Sua estrutura define um formato de arquivo para armazenar e transmitir primitivas geométricas como pontos, linhas e superfícies com uma especificação de estrutura e sintaxe em código ASCII, compacto ou binário, definindo um arquivo neutro. Esta estrutura apresenta-se dividida em 5 seções [24] que definem características do desenho e do sistema de origem, informações geradas pelo pré-processador do padrão para correta tradução dos dados, representação de tipos e subtipos de entidades e ponteiros além de determinar o fim do arquivo e identificar um subtotal de dados gravados para propósitos de transmissão. As seções são denominadas: *Inicial (Start Section)*, *Global (Global Section)*, *Directório (Directory Section)*, *Terminal (Termination Section)* e *Seção de Parâmetro de Dados (Parameter Data Section)*. Um exemplo de representação de entidades em arquivo IGES é apresentado na tabela 3.2.

**Tabela 3.2** - Representação de um arquivo IGES.

SolidWorks IGES FILE using analytic representation for surfaces	S 1
1H,,1H;,5HPart1,22HD:\Dissertação\IGE.IGS,41HSolidWorks 2001 by SolidWorks	G 1
ks Corporation,11HVersion 5.2,32,308,15,308,15,5HPart1,1.,2,2HMM,50,	G 2
0.125,13H020808.162515,1E-008,500000.,7HN estor.,,10,0,;	G 3
314 1 0 0 0	00000200D 1
314 0 0 1 0	0D 2
128 2 0 0 0	01010000D 3
128 0 0 4 0	0D 4
126 6 0 0 0	01010500D 5
126 0 0 2 0	0D 6
110 8 0 0 0	01010000D 7
110 0 0 1 0	0D 8
314,94.1176470588235,67.4509803921569,11.7647058823529,;	1P 1
128,1,1,1,1,0,0,1,0,0,0.,0.,1.,1.,0.,0.,1.,1.,1.,1.,1.,1.,	3P 2
26.462451362,0.,-15.80155642,-41.16381323,0.,-15.80155642,	3P 3
26.462451362,10.,-15.80155642,-41.16381323,10.,-15.80155642,0.,	3P 4
1.,0.,1.;	3P 5
126,1,1,1,0,1,0,0.,0.,1.,1.,1.,1.,1.,0.,0.,1.,1.,0.,0.,1.,0.,	5P 6
0.,1.;	5P 7
110,-41.16381323,10.,-15.80155642,-41.16381323,0.,-15.80155642;	7P 8
110,-41.16381323,10.,-15.80155642,26.462451362,10.,	11P 10
-15.80155642;	11P 11
S 1G 3D 144P 118	T 1

O formato IGES vem sofrendo revisões e implementações através de sua organização controladora, IGES/PDES, tornado-se um dos padrões de formatos mais utilizados para

transferência de dados de modelos geométricos. Paralelamente, muitos outros formatos surgiram de pesquisas desenvolvidas por outros comitês ou organizações. Formatos como o SET (*Standard d'Exchange Transfert*), desenvolvido pela Companhia Aeroespacial Francesa e mais tarde, adotado como padrão pela *French National Standards Body* em 1985, o VDA/FS, desenvolvido na Alemanha especialmente para representar superfícies de maior complexidade. Muitos destes formatos caíram em desuso ou não obtiveram uma aceitação universal como IGES, DXF, STEP e o STL [24].

O IGES não é um completo padrão de transferência de dados para integração entre sistemas de manufatura, mas ainda é um dos padrões mais utilizados para troca de dados de modelos geométricos entre sistemas CAD.

### **DXF/DWG (Data Exchange Files / DraWinG)**

DXF e DWG são formatos de arquivos para armazenagem e transferência de dados entre sistemas CAD desenvolvido pela *Auto Desk* 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. Estes formatos são considerados padrões pela sua aceitabilidade e uso sem uma adoção formal por qualquer organização oficial de padronização. A estrutura do arquivo em formato ASCII (DXF) é dividida em 4 partes [24] que definem valores para as variáveis do sistema e parâmetros como dimensionamento e *layers*, informações de estilos de linhas, sistema de coordenadas, 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.

### **STEP (Standard for the Exchange of the Product Model Data)**

Com a crescente integração entre sistemas computacionais de auxílio a projeto, manufatura, planejamento e controle de processos, o compartilhamento de informações não só de geometrias, mas de um modelo completo do produto tornou-se uma necessidade, passando a exigir uma metodologia mais completa para troca de dados de produto. Em meados de 1984, a organização IGES, com o objetivo de superar a insatisfação com as bases do seu próprio formato, iniciou o desenvolvimento de um padrão que pudesse suceder o IGES, o PDES. Até

então o IGES era tido como um padrão dominante para troca de dados entre sistemas CAD embora outras alternativas tenham surgido em paralelo. O objetivo era formatar um padrão apto a transferir um modelo completo de um produto com suficientes informações para sua utilização em diferentes sistemas.

Neste sentido iniciaram-se esforços por parte do subcomitê 4 da ISO *Technical Committee 184*, juntamente com atividades de pesquisa de outros formatos como o ESPRIT (*European Strategic Programme for Research into Information Technology*) para o desenvolvimento de um trabalho para padrão unificado. Como resultado deste esforço surge o padrão ISO 10303, conhecido como STEP (*Standard for the Exchange of Product Model Data*) que 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 utilização em todo o seu ciclo de fabricação, o que inclui, projeto, manufatura, qualidade, testes e suporte. Sua estrutura compreende um amplo conjunto de primitivas incluindo a topologia necessária de um modelo sólido em adição a outros atributos de projeto, tais como tipo de material, características de formas, especificações de tolerâncias, superfícies e materiais [4].

O formato utiliza uma linguagem interpretável específica de modelagem denominada EXPRESS, utilizada para especificar as informações dos modelos. A descrição destas especificações através da linguagem EXPRESS permite, além da armazenagem de dados em arquivos físicos, um mapeamento destes para visualização gráfica dispensando a definição da representação da entidade, pois a estrutura da linguagem EXPRESS inclui esta representação [24]. Sua estrutura tem uma arquitetura em três camadas permitindo aplicações de visualização e implementações. A primeira camada denominada de camada de aplicação, contém técnicas para implementações do STEP. A segunda, denominada camada lógica, consiste de informações dos recursos como descrição da geometria e estrutura do produto. A terceira camada denominada protocolo de aplicação tem a função de comandar um subconjunto padrão do STEP que garante sua implementação e intercambiabilidade entre diferentes sistemas, sendo mais compatível do que o IGES.

O desenvolvimento deste formato é dividido em classes como uma série de padrões separados em partes, cujo objetivo é abranger diferentes aspectos da integração. As pesquisas estão divididas em 7 classes [24] que desenvolvem, dentre outras, propriedades e características relacionadas à descrição da linguagem EXPRESS, representação de geometrias e estruturas para análise de elementos finitos e mapeamento para arquivos de armazenagem de dados. Estas

propriedades permitem a representação não apenas de dados, mas também de restrições, regras, funções e procedimentos. Esta linguagem possui um sistema de dados para entidades, similar ao conceito de classe no modelo orientado a objeto, e seu mecanismo de herança tem similaridade com a programação orientada a objetos. A tabela 3.3 apresenta um exemplo de definição de entidade geométrica em formato STEP.

**Tabela 3.3** - Exemplo de definição de uma entidade em formato STEP.

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION (( 'STEP AP214' ),
'1');
FILE_NAME ('ST.STEP',
'2002-08-08T19:24:59',
('Nestor' ),
('Wayne Corporation' ),
'SwSTEP 2.0',
'SolidWorks 2001039',
"");
FILE_SCHEMA (( 'AUTOMOTIVE_DESIGN' ));
ENDSEC;

DATA;
#1 = ORIENTED_EDGE ( 'NONE', *, *, #239, .F. );
#2 = EDGE_CURVE ( 'NONE', #242, #230, #30, .T. );
#3 = ORIENTED_EDGE ( 'NONE', *, *, #244, .F. );
#4 = EDGE_LOOP ( 'NONE', ( #233, #240, #238, #173 ) );
#5 = ORIENTED_EDGE ( 'NONE', *, *, #155, .T. );
#6 = ORIENTED_EDGE ( 'NONE', *, *, #55, .F. );
#7 = ORIENTED_EDGE ( 'NONE', *, *, #19, .F. );
#8 = ORIENTED_EDGE ( 'NONE', *, *, #74, .T.
.
#242 = VERTEX_POINT ( 'NONE', #70 );
#243 = ADVANCED_FACE ( 'NONE', ( #71 ), #72, .T. );
#244 = EDGE_CURVE ( 'NONE', #230, #208, #79, .T. );
#245 = ORIENTED_EDGE ( 'NONE', *, *, #74, .F. );
ENDSEC;
END-ISO-10303-21;
```

### **STL** ( *Stereolithography* )

O formato STL é um padrão de fato para descrição de modelos sólidos, desenvolvido para sistemas de prototipagem rápida tridimensional por estereolitografia introduzido em 1988.

Um dos primeiros processos de fabricação rápida de protótipos surgiu por volta do ano de 1987. Pode-se definir a prototipagem rápida como um conjunto de processos tecnológicos que permitem fabricar modelos físicos tridimensionais diretamente a partir de um desenho em sistemas CAD onde a construção dos modelos é realizada progressivamente por camadas dispensando a utilização de ferramentas. O processo inicial utilizava resinas fotopolimerizáveis e apresentava uma série de limitações. Uma relevante limitação era a deficiência de interfaces

adequadas por parte dos sistemas CAD. O avanço tecnológico e o desenvolvimento de padrões de arquivos de transferência de dados permitem atualmente obter modelos por estereolitografia com elevada precisão partindo-se de qualquer modelador de sólido ou superfície.

O padrão STL utilizado em manufatura é um arquivo em código ASCII ou binário que consiste de uma lista de superfícies triangulares que descrevem um modelo sólido gerado por computador [13]. É o padrão de entrada de dados em máquinas de prototipagem rápida, e também, um formato padrão de arquivo de dados utilizado para armazenagem e transferência de informações entre sistemas CAX's. O conjunto de faces triangulares determina a superfície tridimensional do objeto. Cada triângulo deve compartilhar dois vértices com cada triângulo adjacente e nenhum vértice pode estar sobre o lado de um triângulo, definindo uma espécie de malha.

### **3.4 SISTEMAS CAM**

O auxílio de computadores à manufatura, ou sistemas CAM, pode ser definido como a utilização da ferramenta computacional e de recursos da 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 do projeto, através dos sistemas CAD, e os sistemas CAM, possibilitando a automação dos processos de monitoramento, controle e gerenciamento da manufatura.

Groover [18] define sistemas CAM como o efetivo uso da tecnologia computacional no planejamento, gerenciamento e controle das funções de manufatura através de interfaces diretas ou indiretas e divide suas aplicações em duas categorias: de Planejamento e de Controle.

#### **3.4.1 Planejamento da Manufatura**

Nesta categoria de aplicação, os sistemas CAM utilizam computadores indiretamente como ferramenta de suporte para as operações de produção. Não há uma interface direta do sistema computacional com o processo de manufatura. O sistema é utilizado *off-line* para auxiliar no planejamento, programação, previsão e controle de instruções e informações para o processo produtivo. Algumas das principais funções desta categoria de aplicação de sistemas CAM estão relacionadas com o planejamento de processos, a programação da produção, o balanceamento de linhas e principalmente a automação da programação de máquinas CNC.

*Planejamento de Processos Auxiliado por Computador - CAPP (Computer Aided Process Planning);*

No planejamento do processo encontra-se uma das maiores utilizações *off-line* dos sistemas CAM. Seus recursos auxiliam desde a seleção de equipamentos, ferramentas, dispositivos de fixação e determinação de operações até a automação da geração da trajetória de ferramentas, programação CN, simulação de processos e cálculo de custos. A função de planejamento pode ser definida como a atividade responsável pela conversão de dados de um projeto em instruções de trabalho através do detalhamento sistemático de uma seqüência de usinagem, além de métodos e parâmetros de operação que serão utilizados na manufatura da matéria bruta. Tanto a manipulação de bancos de dados como de ferramentas, condições de operação e máquinas operatrizes permite automatizar as operações de cálculo dos parâmetros de processo e estimativa de custos [7].

*Programação e Planejamento da Produção*

Esses sistemas, conhecidos como sistemas *MRPII (Manufacturing Resource Planning)* têm a função de controlar as necessidades da produção através da supervisão dos pedidos de materiais. O algoritmo computacional determina o programa de produção adequado para atender às necessidades da manufatura desenvolvendo uma programação de produção para cada centro de trabalho baseado na programação geral [7] com base nas informações de um banco de dados.

*Balanceamento de linhas de montagem auxiliado por computador*

O sistema tem a função de auxiliar no processo de distribuição de tarefas entre as estações de trabalho e linhas de montagem. A reorganização de máquinas e alocação de operadores ao longo da linha de montagem é planejada para atingir determinado tempo de ciclo em função da demanda predeterminada para as células de manufatura de peças componentes [7].

*Programação de máquinas de comando numérico auxiliado por computador*

Apesar da abrangência do significado, o termo CAM, está diretamente associado à programação CN, principalmente devido à difusão de sistemas CAD/CAM disponíveis no mercado, cujo propósito é de auxiliar na geração automática de programas para máquinas de comando numérico interagindo com desenhos gerados em sistemas CAD. Sua automação, permite que, através de ambientes gráficos interativos, a operação de programação de máquinas de comando numérico seja facilitada. A elaboração automática de programas CN se dá através de uma interpretação da geometria de peças geradas em arquivos gráficos (CAD) e uma interação

com o usuário do módulo CAM. Essa interação é responsável pelo fornecimento de informações para a geração do programa CN. A programação gerada é uma modelagem geométrica da peça seguida da seleção de parâmetros como tolerância, sobre-metal, ferramentas, velocidade de corte e avanço informados pela interação com o usuário ou, em alguns casos, determinados de modo automático pelo sistema. O resultado final do processamento destes dados é a geração automática de um arquivo contendo dados de trajetória de ferramenta associados a parâmetros de operação (CLData) ou seja, as instruções para fabricar a peça.

### **3.4.2 Controle da Manufatura**

Nesta categoria de aplicação de sistemas CAM, os recursos computacionais são diretamente ligados ao processo a ser monitorado e/ou controlado, isto é, o computador está conectado diretamente ao processo de fabricação. Estes sistemas podem controlar qualquer máquina CNC bem como equipamentos de transporte, robôs, linhas de transferência, CLPs, etc. Seus recursos ainda permitem a simulação de tarefas de diagnóstico, possibilitando a identificação de falhas e erros de planejamento.

Em monitoração, utiliza-se uma interface direta entre o computador e a manufatura apenas para coleta de dados do processo. O controle é realizado por operadores guiados pelas informações do processamento dos dados obtidos. No controle do processo, tanto a coleta de dados quanto à ação no processo são realizados pelo sistema computacional, com ou sem interferência humana. O processo produtivo é continuamente supervisionado e o controle de operações pode ser realizado segundo o processamento dos dados obtidos e a transmissão de instruções do sistema gerenciador para as máquinas operatrizes, portanto, o fluxo de informações é constante tanto no sentido controle – máquina quanto no sentido máquina – controle. Atualmente, uma das formas de comunicação e interligação de informações entre máquinas e equipamentos em ambiente fabril são as redes de chão-de-fábrica que permitem a transferência de dados entre diferentes equipamentos. Suas formas de conexão vão desde o cabo coaxial à fibra ótica.

A diversidade de equipamentos e software para controle disponíveis no mercado atualmente permite diversos níveis de automação e integração. Se existem limitações para a concepção de ambientes de manufatura automatizados, essas limitações estão na dificuldade de se integrar os diferentes sistemas de auxílio computacional à manufatura e não na disponibilidade tecnológica.

## 4 ROBÓTICA INDUSTRIAL

### 4.1 ROBÔ E ROBÓTICA

Na evolução dos processos produtivos e da automação industrial, fortemente relacionada ao comando numérico, está a robótica industrial. Na tecnologia de comando numérico controla-se o posicionamento de um ponto no espaço enquanto que em robótica, tanto o posicionamento quanto à orientação são manipulados, o que requer mais graus de liberdade e algoritmos de controle mais complexos [20]. O termo robô foi originalmente utilizado em uma peça de teatro de 1921 do escritor tchecoslovaco Karel Capek, denominada “Rossum`s Universal Robot” onde a palavra foi utilizada para designar seres mecânicos antropomorfos que se rebelavam contra os seres humanos [29]. O termo robô vem do tcheco *Robota*, que significa “trabalho forçado [27].

Um dos primeiros robôs industriais a serem instalados em chão-de-fábrica data de 1961. O equipamento foi desenvolvido pela empresa *Unimation Inc*, onde seus criadores, George Devol e Joe Engelberger, iniciaram a fabricação e comercialização destes equipamentos recebendo a denominação de “pais da robótica”. O projeto foi o resultado da combinação das tecnologias de comando numérico e atuadores hidráulicos. Alguns marcos da história da robótica são indicados na tabela seguinte 4.1 [31].

Tabela 4.1 - Marcos históricos da robótica industrial.

1801 – J. Jacquard inventou um tear programável;
1959 – Primeiro robô comercial introduzido no mercado. Controlado por fins-de-curso e excêntricos;
1960 – Primeiro robô “Unimate”. Utilizam princípios de controle numérico e atuadores hidráulicos;
1961 – Primeiro robô industrial instalado em chão de fábrica;
1968 – Desenvolvimento de um robô móvel com câmera de vídeo e sensores de contato denominado “ <i>Shakey</i> ” no <i>Stanford Research Institute</i> ;
1971 – Desenvolvimento de um elemento mecânico (braço) com atuação elétrica – <i>Stanford Research Institute</i> ;
1973 – Surgimento da primeira linguagem de programação de robôs: <i>WAVE</i> seguida em 1974 pela linguagem <i>AL</i> . Da união das duas, surge a linguagem comercial da <i>Unimation – VAL</i> ;
1978 – Desenvolvimento do robô <i>PUMA</i> , introduzido pela <i>Unimation</i> ;
1979 – Desenvolvimento na Universidade de <i>Yamanashi</i> do robô <i>SCARA</i> ;
1981 – Desenvolvimento da atuação direta ( <i>Direct Drive</i> ) para controle de movimentos;
1983 – Projeto de linhas flexíveis de montagem automatizadas com auxílio de robôs.;
1997 – A <i>Honda</i> anuncia o primeiro robô humanóide que sobe escadas;

As definições para robô industrial divergem entre especialistas ou entidades ligadas a área. A RIA (*Robotics Industries Association*) define robô como um manipulador reprogramável, multifuncional, projetado para movimentar materiais, peças, ferramentas ou dispositivos especializados. A ISO define robô como um manipulador multifuncional reprogramável com controle de posição automático, com vários graus de liberdade e base fixa ou móvel para utilização em aplicações de manipulação de peças, ferramentas e dispositivos especializados. Dentre as definições mais restritivas está a de Régie Renault que afirma ser um robô industrial uma máquina automática universal destinada à manipulação de objetos e dotada de uma capacidade de aprendizagem de um comportamento típico, da faculdade de observar o ambiente (percepção); da faculdade de analisar a informação assim obtida; e da possibilidade de modificar seu comportamento típico. Todas as definições apresentadas para a denominação de robôs têm em comum o fato da reprogramabilidade, que implica em controle computacional e a multifuncionalidade que determina o caráter flexível na utilização do equipamento. O conceito básico é a construção de um equipamento flexível o suficiente para executar diferentes tarefas automaticamente através de dispositivos facilmente programáveis.

A designação robótica industrial surge do estudo dos manipuladores robóticos que são usualmente conhecidos por robôs industriais.

## **4.2 APLICAÇÕES DE ROBÔS**

A utilização de robôs na indústria está ligada a um vasto campo de aplicações. Robôs industriais são amplamente utilizados em ambientes de trabalho nocivo a seres humanos, ciclos repetitivos onde se deseja precisão e repetibilidade, operações de manipulação com dificuldade operacional devido ao peso ou forma do objeto a ser manipulado e em atividades que requeiram operação contínua. No campo industrial os robôs são aplicados essencialmente nas atividades de manipulação de materiais, operações de processos, montagem e inspeção.

Na manipulação, a função do equipamento é mover materiais ou peças de uma posição a outra. Como exemplos básicos têm-se operações de movimentação de materiais entre equipamentos como esteiras transportadoras (*pick and place* – em operações de armazenamento e transporte, fundição e forjaria), paletização de materiais e operações de carga e descarga de máquinas operatrizes (centros de usinagem, prensas, máquinas de injeção de plástico). Para a realização destas tarefas o sistema robótico geralmente utiliza como efetadores as garras e

normalmente estará integrado a outros equipamentos com a finalidade de otimizar a produtividade.

Uma das maiores aplicações dos sistemas robóticos estão nas operações de processos. Entre os diversos tipos de aplicações de robôs em operações de processamento, podem ser destacados:

- soldagem por resistência elétrica;
- pintura e envernizamento de superfícies;
- acabamento superficial;
- limpeza através de jato d'água e abrasivos;
- corte através de processos por plasma, laser, oxicorte ou jato de água;
- fixação de peças com parafusos, rebites;
- operações de usinagem como rebarbação e polimento de superfícies, furação e fresamento.

Em atividades de montagem e inspeção, a utilização de robôs tem crescido rapidamente devido à evolução tecnológica na utilização de sensores e dispositivos de automação de sistemas robóticos como sistemas de visão. Como uma forma híbrida de utilização de robôs, a atividade de montagem e inspeção exige tanto manipulação de materiais quanto a operação de ferramentas de processamento em atividades como:

- fixação de circuitos integrados em placas;
- montagem de peças;
- inspeção por visão;
- verificação dimensional de peças através de sensores.

### **4.3 ELEMENTOS BÁSICOS DE UM ROBÔ**

Genericamente um robô é constituído de elementos mecânicos interligados por juntas, tendo em seu final uma estrutura semelhante a um pulso que permite ao equipamento o movimento em vários graus de liberdade. Sendo um equipamento reprogramável, seu movimento deve ser controlado por uma unidade de controle computacional. Basicamente, estes equipamentos são formados por elementos de controle, manipuladores e efetadores.

### **4.3.1 ELEMENTOS DE CONTROLE**

Os elementos de controle de um sistema robótico têm como componentes básicos computadores e periféricos que constituem o hardware do componente e programas residentes que gerenciam tarefas e funções. Esses elementos podem ser subdivididos em atuadores, unidade de controle e sensores. O sistema tem a função de interagir com o operador, controlar os movimentos dos elementos mecânicos e gerenciar a execução de programas de tarefas, através de algoritmos de controle capazes de gerenciar a utilização de atuadores, supervisionar os elementos através de sensores, além de realimentar o sistema, proporcionando certo grau de autonomia ao equipamento.

#### **Atuadores**

São dispositivos de conversão de energia elétrica, hidráulica ou pneumática em potência mecânica que impõem movimentos a um sistema mecânico para a realização de determinada tarefa. Pode-se citar como exemplo de atuadores os motores elétricos, cilindros pneumáticos e eletroímãs.

Os motores elétricos e hidráulicos permitem controle preciso de velocidade e posicionamento. A utilização de *encoders* (sensores de posição angular) e tacômetros possibilitam o controle com realimentação de dados proporcionando automação à movimentação dos elementos mecânicos. Os atuadores pneumáticos são mais utilizados para movimentação de cargas entre posições bem definidas, que caracterizam o controle denominado ponto-a-ponto.

#### **Sensores**

Os sensores são destinados à medição do estado interno do sistema robótico e do ambiente exterior. Suas leituras fornecem parâmetros para a unidade de controle, principalmente de posição, velocidade e torque dos elementos mecânicos.

#### **Unidade de Controle**

Como unidade de controle denominam-se os controladores de movimento como computadores e CLPs que gerenciam e monitoram a operação do equipamento.

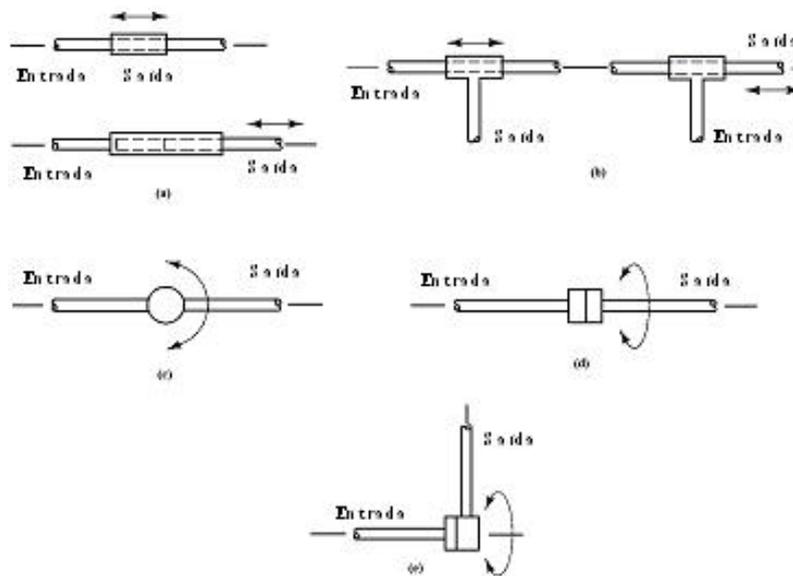
### 4.3.2 MANIPULADORES

Manipuladores são as estruturas mecânicas dos sistemas robóticos formados pela junção de elementos mecânicos de alta rigidez conectados entre si por juntas articuladas responsáveis pela capacidade de movimentação do conjunto. As juntas dos sistemas robóticos são similares às articulações humanas, proporcionando ao conjunto movimentos relativos entre os elementos mecânicos e definindo os graus de liberdade de movimentação do sistema. Cada junta pode ter até três eixos de movimentação com atuadores independentes para cada eixo. As juntas de articulação dos elementos mecânicos podem ser classificadas em rotacionais ou prismáticas, de acordo com o tipo de movimento relativo que permitem, respectivamente rotação ou translação.

As juntas de articulação dos sistemas robóticos são classificadas de acordo com as capacidades de movimento como [17]:

- LINEAR: Permite movimento relativo linear entre os elementos mecânicos e os eixos dos elementos estão em paralelo.;
- ORTOGONAL: É também uma junta de movimento linear porém com os eixos dos elementos mecânicos permanecem perpendiculares durante a movimentação;
- ROTACIONAL: Permite movimento relativo rotacional entre os elementos, com os eixos de rotação perpendiculares aos eixos dos elementos mecânicos;
- CILÍNDRICA: Também de movimento relativo rotacional, porém com os eixos de rotação e dos elementos mecânicos permanecendo paralelos entre si durante a movimentação;
- REVOLUÇÃO: Neste tipo de junta, o eixo de um elemento mecânico é paralelo ao eixo de rotação e o eixo do outro elemento mecânico é perpendicular.

Alguns exemplos de tipos de juntas são mostrados na figura 4.1.



**Figura 4.1** - Tipos de juntas: (a) Linear; (b) Ortogonal; (c) Rotacional; (d) Cilíndrica; (e) de Revolução; [16].

As juntas lineares são construtivamente mais complexas que as juntas prismáticas ou rotacionais e apresentam maior rigidez e precisão. Além destas características, os aspectos construtivos destes elementos devem priorizar a redução da resistência ao atrito por consequência dos sucessivos movimentos de partida e parada dos elementos.

O conjunto de elementos mecânicos, juntas e atuadores permite a movimentação do sistema e determina, por consequência, o espaço de trabalho do robô. Os robôs industriais são formados pela união de elementos mecânicos e juntas de articulação em forma de cadeia cinemática aberta determinando que o número de graus de liberdade de um sistema seja equivalente ao número de juntas do conjunto cinemático [29]. A base do sistema ou primeiro elemento da cadeia cinemática pode ser fixa ou móvel dependendo da necessidade de movimentação do conjunto e sua movimentação pode ser controlada por sua unidade de controle. Em contrapartida, o último elemento mecânico da cadeia é ligado somente ao seu precedente e sua extremidade serve para a fixação de dispositivos que irão trabalhar como efetadores conforme o tipo de aplicação. Neste último elemento, denominado punho, encontra-se um conjunto de articulações que permite dois ou três graus de liberdade de movimentação ao efetador. O punho tem a função de interligar os elementos mecânicos dos sistemas robóticos ao efetador final para a realização de determinada tarefa. Seus graus de liberdade de movimentação servem para orientar a posição final do efetador enquanto os graus de liberdade do braço mecânico têm a função de posicionar o dispositivo. A configuração que confere maior versatilidade ao punho é aquela em que os três eixos de rotação se interceptam em um ponto, e

recebe a denominação de punho esférico. Considerando que os três eixos de rotação do punho se encontram em um único ponto, a presença do punho não altera o espaço de trabalho.

Os manipuladores ainda contam com um sistema interno de sensores de controle e um sistema de transmissão de potência (torque/força e velocidade angular/linear) originada nos atuadores. A transmissão ocorre através da atuação de dispositivos como engrenagens, fusos de esferas recirculantes, correias e polias. Sua escolha depende de fatores de projeto como potência a ser transmitida e distância do atuador ao elemento e suas características mais importantes são rigidez e eficiência mecânica [29]. Os sensores internos têm a função de monitorar permanentemente o estado do robô, através de transdutores ligados a cada grau de liberdade dos elementos mecânicos, passando estas informações à unidade de controle para correções de posicionamento e movimentação.

### **4.3.3 EFETUADORES**

A característica de multifuncionalidade apresentada nas definições de robô, demonstra a variedade de possibilidades de utilização deste equipamento na indústria. Sua utilização e adaptação a diferentes funções dependem de fatores como sua configuração estrutural, características de desempenho, tipos de manipuladores e controle de tarefas. Para cada tarefa, diferentes efetuadores podem ser selecionados para promover a correta interação do equipamento com o ambiente operacional

Os efetuadores são dispositivos fixados junto ao último elo, órgão terminal ou punho de um robô, que permite ao mesmo realizar uma determinada tarefa. Responsáveis pelo objetivo final da operação do robô, o conjunto de todos os demais componentes deve habilitar a execução de operação do efetuador. Os efetuadores são responsáveis pela interação entre o órgão terminal do elemento mecânico e o objeto a ser trabalhado, sendo genericamente classificados como garras ou ferramentas especiais para processos..

#### **Garras Mecânicas**

As garras mecânicas têm o objetivo de simular os movimentos da mão humana que, segundo pesquisas de diferentes configurações de garras, são ditas como a configuração de maior versatilidade para realizar a manipulação de objetos. Apesar de não ser capaz de simular seus movimentos com exatidão, possuem características semelhantes com algumas limitações.

Em garras mecânicas os elementos mecânicos (dedos) que fazem o contato direto com o objeto a ser manipulado podem ser fixos ou intercambiáveis para compensar desgastes e

proporcionar flexibilização e multifuncionalidade em sua utilização. Sua aplicação está relacionada à manipulação de objetos em atividades como:

- Carregar e descarregar máquinas;
- Transportar peças;
- Paletização;
- Manipulação de caixas, garrafas, matérias primas, etc;
- Manipulação de ferramentas;

Existem diferentes formas de manipular objetos utilizando-se de garras, sendo a forma física do objeto, da garra, o tipo de controle e as características do robô de manipulação alguns dos aspectos relevantes para o êxito na operação. O formato da garra e a forma de controle devem ser projetados para realizar determinada função e o seu projeto deve levar em consideração aspectos como forma, massa, material e rigidez do objeto a ser manipulado, velocidade e aceleração do manipulador.

As garras podem prender um objeto através da constrição física ou atrito. Na construção física, os elementos mecânicos de contato (dedos) são projetados de acordo com a geometria da peça, não permitindo sua movimentação. No atrito, os elementos mecânicos devem aplicar uma força suficiente para segurar o objeto por atrito durante a operação de trabalho, com cuidado no controle da força para não provocar danos aos objetos manipulados. Estes dispositivos têm grande importância na obtenção do objetivo final na operação de robôs. Existem diversas formas de classificação de garras mecânicas, dentre outras relativas [9]:

- ao número de elementos mecânicos de contato (dedos);
  - Simples – apenas um elemento;
  - Duplo - dois elementos mecânicos de contato, etc.
- à forma de contato com o objeto;
  - Contato interno;
  - Contato externo;
  - Contato em forma de “V”;
  - Encaixe;
- ao tipo de movimentação dos elementos mecânicos;
  - Movimento Pivotante – elementos giram ao redor de pontos fixos na garra.

Normalmente é utilizado algum tipo de mecanismo articulado;

- Movimento Linear – Os elementos mecânicos deslocam-se entre si paralelamente, abrindo-se e fechando-se.

## **Garras Não Mecânicas**

São dispositivos com a função de prender e manipular objetos que utilizam princípios não mecânicos, tais como eletromagnetismo e sucção, podendo ser classificados em [9]:

- Garras articuladas;
- Ventosas;
- Garras magnéticas;
- Garras adesivas;
- Ganchos e cadinhos
- Adaptador automático de garras;

### *Garras Articuladas*

São projetadas para “agarrar” objetos de diferentes tamanhos e formas. Os vínculos são movimentados por pares de cabos, onde um cabo flexiona a articulação e o outro a estende. Sua destreza em segurar objetos de formas irregulares e tamanhos diferentes se deve ao grande número de vínculos.

### *Ventosas*

Ventosas ou garras a vácuo são projetadas para segurar uma superfície lisa por ação de vácuo. Estas garras possuem ventosas de sucção conectadas a bomba de ar comprimido, que seguram superfícies lisas como chapas metálicas e caixas de papelão. Para reduzir o risco de acidentes devido à perda de pressão da linha de vácuo, é comum usar mais do que uma ventosa. São indicados para objetos com planos lisos e limpos de modo a permitirem a formação do vácuo. Em objetos macios o copo de sucção deve ser rígido e em objetos rígidos o inverso. Nestes dispositivos a formação do vácuo pode ser através de bomba de vácuo ou venturi, apresentando como vantagens operacionais apenas uma superfície de contato, distribuição uniforme de pressão, garra relativamente leve e aplicação a uma variedade de materiais para a ventosa;

### *Garras Magnéticas*

Garras magnéticas são utilizadas para manusear objetos que podem ser magnetizados (ferrosos) através de um campo magnético, principalmente chapas e placas. Estas garras podem ser do tipo eletromagnéticas onde o controle de acionamento e desligamento necessita de uma fonte de corrente contínua ou de ímãs permanentes onde a operação exige um dispositivo separador entre peça e ímã. Ambos os tipos de garras são muito eficientes, uma vez que eles

podem segurar objetos de vários tamanhos e não necessitam de grande precisão no posicionamento da garra. Apresentam como vantagens e desvantagens [9].

Vantagens:

- menor tempo de pega;
- garra não precisa ser projetada para um determinado tamanho de peça;
- manuseio de peças ferrosas com furos;
- necessidade de apenas uma superfície de contato;

Desvantagens:

- magnetismo residual da peça de trabalho;
- possíveis deslizamentos naturais;
- menor precisão de posicionamento;

#### *Garras Adesivas*

São dispositivos que utilizam substância adesiva para operações de manuseio de objetos. Sua utilização é indicada para manusear materiais leves como tecido e outros. Para contornar o problema da perda de substância adesiva carrega-se o material adesivo na forma de fita contínua.

#### *Ganchos e Cadinhos*

Os ganchos são indicados para o manuseio de peças que tenham algum tipo de saliência que possa ser utilizada para encaixe. Utilizam-se cadinhos no manuseio de materiais de difícil controle de volume e/ou quantidade como líquidos e pós, produtos granulados, alimentícios, etc.

#### *Adaptador Automático de Garra*

Outros dispositivos podem ser projetados para manusear diferentes tipos de material. Na capacidade de projetar dispositivos para atender às expectativas de trabalho, reside a multifuncionalidade das garras de manipulação. Estes projetos devem levar em conta as características de desempenho do robô e a forma de controle utilizada.

Desenvolvidos a partir da necessidade de se ter uma garra capaz de segurar diversos tipos de objetos, um trocador automático vem a ser um adaptador que permite que uma garra seja rapidamente ligada ou removida do braço do robô. Esses sistemas, embora possam flexibilizar a operação, podem apresentar algumas desvantagens devido principalmente à:

- peso adicional na extremidade do braço do robô;

- complicações tecnológicas são uma fonte potencial de mau funcionamento;
- acréscimo no custo do robô;
- tempo gasto na troca das garras.

### **Ferramentas para Processos**

As ferramentas são dispositivos unidos ao órgão terminal do elemento mecânico manipulador (robô) junto ao seu punho, utilizadas para a realização de trabalho sobre um objeto, com função relacionada a operações de processamento [1].

Devido às suas funções de operação, estes dispositivos devem estar rigidamente fixos às extremidades dos robôs, impossibilitando a movimentação relativa entre eles onde o braço mecânico tem a função de posicionar e orientar a ferramenta em relação à peça a ser trabalhada. Sua utilização está relacionada às tarefas de operação desejada sobre uma trajetória planejada. Entre os diversos tipos de ferramentas disponíveis para operações de processamento temos:

- pistola de pintura;
- tochas para soldagem;
- corte a plasma;
- conjunto de pinças para soldagem por pontos;
- dispositivo para corte a plasma;
- pistola para corte por jato de água;

## **4.4 ESPAÇO DE TRABALHO E TIPOS DE MANIPULADORES**

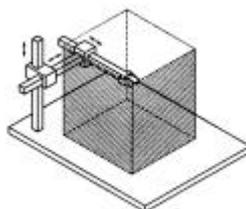
### **4.4.1 Espaço de Trabalho**

O espaço ou volume de trabalho de um robô pode ser definido como a região dentro da qual o manipulador pode movimentar e/ou posicionar o efetuator e é determinado pelo número, tipo e limite de movimentação das juntas utilizadas na configuração do sistema robótico. A forma e dimensões do espaço de trabalho dependem da estrutura construtiva e das limitações que são impostas pelas juntas. O objetivo da configuração da estrutura é posicionar o elemento terminal (ferramenta) do sistema para a realização de uma determinada tarefa enquanto o punho tem a função de orientar no espaço o dispositivo de operação.

O posicionamento no espaço requer um mínimo de três graus de liberdade e, de acordo com os elementos mecânicos e o tipo e forma de suas juntas, definem-se diversas formas de

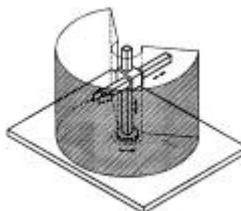
configurações de robôs, com os respectivos volumes de trabalho. Dentre as configurações de estrutura convencionais, as seguintes podem ser destacadas [22]:

*Cartesiana* – seqüência de três juntas prismáticas com eixos ortogonais entre si. Define um espaço de trabalho na forma de um paralelepípedo (figura 4.2);



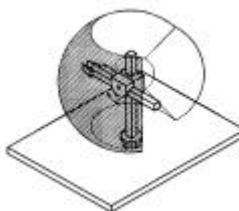
**Figura 4.2** - Espaço de trabalho - Configuração cartesiana.

*Cilíndrica* – duas juntas prismáticas e uma rotacional. A junta rotacional da base determina um espaço de trabalho na forma de setor de cilindro (figura 4.3);



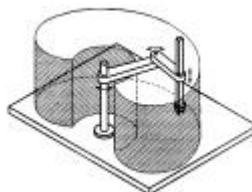
**Figura 4.3** - Espaço de trabalho – Configuração cilíndrica.

*Esférica* – uma junta prismática e duas rotacionais. Sua configuração de duas juntas rotacional define um espaço de trabalho na forma de uma porção de esfera (figura 4.4);



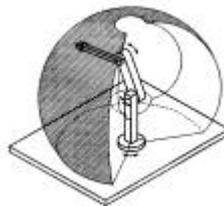
**Figura 4.4** - Espaço de trabalho – Configuração esférica.

*Articulada Horizontal ou SCARA* – duas juntas rotacionais paralelas e uma prismática com eixos de movimentação paralelos entre si. O espaço de trabalho é mostrado na figura 4.5;



**Figura 4.5** - Espaço de trabalho – Configuração SCARA.

*Antropomórfica* - três juntas rotacionais com o eixo de rotação da base, ortogonal em relação aos outros dois eixos que por sua vez são paralelos entre si. O espaço de trabalho tem a forma complexa mostrada na figura 4.6;



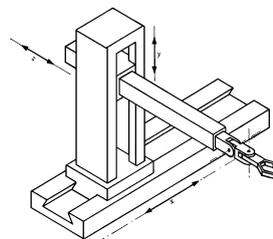
**Figura 4.6** - Espaço de trabalho - Configuração antropomórfica.

#### 4.4.2 Tipos de Robôs

Os diferentes tipos de robôs podem ser caracterizados pela configuração de elementos mecânicos e juntas que determinam o espaço ou volume de trabalho [22].

##### *ROBÔ CARTESIANO*

São robôs que realizam seus movimentos principais utilizando juntas prismáticas. A combinação de três juntas prismáticas resulta em movimentos de translação em cada um dos eixos de movimentação e num espaço de trabalho retangular.

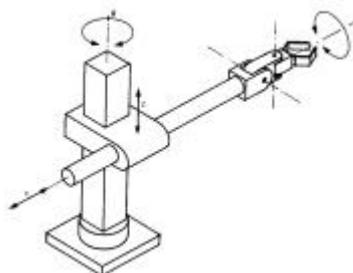


**Figura 4.7** - Robô Cartesiano.

A configuração com juntas lineares simplifica o controle de movimentação do sistema, porém eleva o custo devido à complexidade de construção das mesmas. Estes robôs são mais utilizados em operações de montagem que requerem posicionamento preciso em volumes de trabalho pequenos ou em operações de carga e descarga de máquinas-ferramenta. A estrutura do robô cartesiano do tipo pórtico permite uma precisão de posicionamento elevada e uniforme. Suas características construtivas permitem, além de precisão elevada, maior capacidade de carga e dimensões de trabalho.

### *ROBÔ CILÍNDRICO*

Esta configuração é formada por duas juntas prismáticas e uma rotacional, proporcionando dois movimentos de translação e um de rotação. O espaço de trabalho gerado tem a forma de um cilindro onde o movimento das juntas prismáticas define a altura e o diâmetro do volume de trabalho e o movimento rotacional define a forma cilíndrica. Este tipo de robô é utilizado basicamente na movimentação de peças e materiais para máquinas operatrizes onde se necessita de amplo volume de trabalho e pequena precisão.

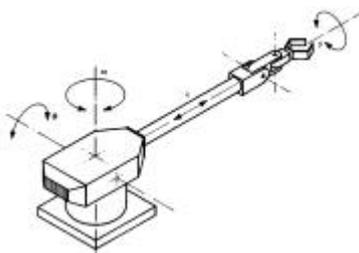


**Figura 4.8** - Robô Cilíndrico.

### *ROBÔ ESFÉRICO*

Neste tipo de configuração, o espaço de trabalho, gerado pelos dois movimentos de rotação e um de translação de suas duas juntas rotacionais e uma prismática, tem a forma de uma esfera. As duas juntas rotacionais descrevem dois arcos perpendiculares entre si, enquanto a junta prismática define o raio da superfície esférica.

Os robôs esféricos proporcionam alta precisão de posicionamento e são atualmente utilizados para atividades em ambientes insalubres como soldagem e manipulação de grandes cargas.

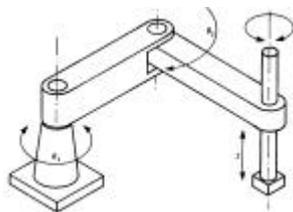


**Figura 4.9** - Robô Esférico.

### *ROBÔ ARTICULADO HORIZONTAL OU SCARA*

Este tipo de robô caracteriza-se pela disposição de duas juntas rotacionais paralelas, a fim de se obter uma movimentação sob um plano, e uma junta prismática, que permite movimento

perpendicular a este plano. O espaço de trabalho gerado nesta configuração é aproximadamente cilíndrico [29]. Devido à configuração de seus elementos mecânicos, é muito utilizado em atividades de montagem de componentes de dimensões pequenas.



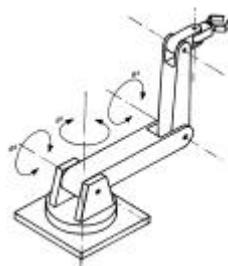
**Figura 4.10** - Robô SCARA.

O posicionamento em um plano horizontal é dado pelo movimento de rotação das duas juntas rotacionais, enquanto o movimento de translação da junta prismática possibilita deslocamentos verticais. Esta configuração apresenta elevada precisão de posicionamento e desempenho dinâmico. Surgiu no mercado japonês com a finalidade de suprir a necessidade de um robô de baixo custo com capacidade de operação para três ou quatro graus de liberdade. A aplicação principal é na área de montagem e movimentação de peças leves.

Sua denominação SCARA (*Selective Compliance Arm of Robot Assembly*) é devido a sua característica de estrutura rígida para esforços na direção vertical e flexibilidade no plano horizontal que o definem como um braço robotizado para montagem e permitem ao sistema executar tarefas de inserções em atividades de montagem.

### **ROBÔ ANTROPOMORFO**

Este tipo de configuração é formado exclusivamente por juntas rotacionais que permitem maior mobilidade ao manipulador. O espaço de trabalho gerado pode ser considerado como uma porção de esfera cuja configuração depende das efetivas possibilidades de movimentação das juntas. Sua denominação, antropomorfo, é devido à tentativa de reprodução com similaridade das possibilidades de movimentação de um braço humano. A estrutura articulada de um robô antropomorfo é mais compacta e versátil em relação a outra configuração qualquer de mesma dimensão. Sua característica construtiva possibilita um volume de trabalho mais amplo e maior versatilidade para atingir um determinado ponto no espaço de trabalho com diferentes configurações contornando obstáculos. Em geral, um robô antropomorfo pode realizar com maior facilidade todas as tarefas que os outros tipos de robôs executam.



**Figura 4.11** - Robô Antropomorfo.

As limitações deste tipo de estrutura são o custo, geralmente superior às demais configurações, e a capacidade de carga devido a sua forma construtiva em balanço. Por isso, para cada configuração de elementos mecânicos (braço) tem-se uma especificação de carga máxima, aceleração e velocidade.

## **4.5 PROGRAMAÇÃO DE ROBÔS**

### **4.5.1 Métodos de Programação**

Num contexto de sistemas flexíveis de manufatura, onde se busca otimizar sistemas produtivos de forma a obter alta produtividade com uma variedade de produtos, os sistemas robóticos têm destacada participação. A automação de processos que estes sistemas proporcionam, permitem flexibilidade através da reprogramabilidade de sistemas e multifuncionalidade de equipamentos por meio da adaptação destes as diferentes funções.

A programação de robôs pode ser definida como a descrição de procedimentos a serem tomados pelo manipulador sob a ação do controlador [29]. O processo de programação envolve a especificação de pontos a serem atingidos e a definição de trajetórias e parâmetros de operação como velocidade e controle de sinais de sensores externos quando utilizados. Essa programação pode ser efetuada em modo “*on-line*” ou “*off-line*”, considerando condições especiais.

#### **Programação *on-line***

O método de programação *on-line* é o método mais simples de programação de robôs, envolvendo diretamente o equipamento no processo de programação onde se utiliza uma interface do robô como forma de entrada de dados. O programador conduz o robô, via interface (*teach pendant*) ou manualmente (através da manipulação física do elemento mecânico), pela seqüência de posições e trajetórias desejadas. Nesta metodologia de programação, a principal função é a definição destas trajetórias e seqüências de posições para os elementos mecânicos. Ao

processo de programação associam-se parâmetros de velocidade a cada localização no espaço e define-se o tipo de movimentação (linear, ponto a ponto ou circular) que o efetuador deve realizar.

O controle de movimentação linear proporciona o movimento por pontos com pequenos incrementos de coordenadas através da memorização de posições, em intervalos constantes de tempo, além da possibilidade de monitoração de dados de sensores de posicionamento durante a movimentação do robô. Este tipo de movimento está relacionado ao controle de coordenadas de todas as juntas de movimentação do robô durante sua condução e memorização de trajetória. No controle ponto a ponto, não há memorização de posições intermediárias entre os pontos final e inicial. O robô move-se diretamente de um ponto a outro sem considerar o caminho a ser percorrido. Na movimentação circular, o controle de trajetória envolve a coordenação de eixos e juntas simultaneamente para a realização do movimento desejado. Os eixos se movem coordenadamente para gerar a trajetória definida. A posição de cada eixo é gravada pela unidade de controle em intervalos de tempo constante pelos sensores de posição dos eixos durante a condução de programação.

Segundo Groover [18], a metodologia de programação com a utilização direta do robô pode ser dividida em:

- programação manual;
- programação por condução (*walkthrough programming*);
- programação com a utilização da interface homem-máquina (*leadthrough programming*);

A programação manual pode ser definida como procedimentos de setup utilizados para a programação de robôs de baixa tecnologia para realização de tarefas simples como movimentação de peças entre pontos definidos. Na programação por condução o objetivo é a definição da correta seqüência de trajetórias. O programador conduz manualmente os elementos mecânicos do robô às posições desejadas, memorizando e definindo trajetórias e seqüências de movimentações desejadas para posterior reprodução em operações de processo onde os parâmetros de velocidade podem ser controlados independentemente. O controle de movimentação pode ser efetuado ponto a ponto ou por trajetória circular. O método de programação por interface homem-máquina utiliza-se do *teach pendant* para comandar os movimentos do robô através de um dispositivo de interação como um *joy-stick*. Cada seqüência de movimentação pode ser memorizada para a programação do ciclo de trabalho.

As vantagens do método de programação *on-line* estão na facilidade de acesso e memorização simultânea de trajetórias e pontos pré-determinados além da interação direta com o sistema de controle na consistência e verificação das instruções de programa. Como principais desvantagens pode-se destacar a utilização dos elementos mecânicos na programação o que interrompe suas atividades durante o processo, aumenta o tempo de programação e não permite lógicas mais complexas e cálculos na programação.

### **Programação *off-line***

O método de programação *off-line* pode ser definido como o processo de desenvolvimento de programas de robôs através da construção de um texto de programa, editado sem a necessidade da utilização do equipamento durante esta tarefa. O programa é editado sem a presença do robô e posteriormente inserido no sistema para controle do equipamento. Neste sistema de programação o código é gerado em um computador com instruções de trajetórias, parâmetros e seqüência de operações escritas em formato textual. Por ser gerado fora do controlador do robô, geralmente em um computador, tem-se um conjunto de ferramentas de auxílio a programação, desde simples editores de texto até complexos sistemas de simulação em três dimensões. A programação *off-line* surgiu em resposta à principal desvantagem do método *on-line*, permitindo que o robô permaneça em atividade durante o processo de programação. Contudo, por mais que este método dispense a utilização do robô na tarefa de programação, o equipamento é necessário para testar e memorizar pontos e posicionamentos a serem realizados na execução do programa.

Os avanços em tecnologia de robôs, tanto em hardware quanto em software, têm proporcionado o crescimento na utilização deste método, especialmente em aplicações complexas que exigem tempos longos de desenvolvimento de programas. Este método de programação *off-line* com a utilização de ferramentas computacionais de simulação e animação tem possibilitado algumas vantagens em relação aos outros métodos tais como:

- reutilização dos dados gerados em sistemas CAD, proporcionando uma programação rápida e a simulação de ambientes reais de operação;
- verificação e correção de erros de programação, evitando danos e paradas no processo produtivo e reduzindo tempos de testes;
- possibilidade de simulação de ambiente de trabalho sem a necessidade da disponibilidade destes equipamentos;

- redução de tempos de parada nos robôs devido à possibilidade de programação de diferentes tarefas enquanto o robô permanece ativo na linha de produção;
- possibilidade de programação de mais de um tipo de robô sem a necessidade de conhecimento de suas diferentes características de sintaxe e operação;
- integração com sistemas CAD/CAM, possibilitando a interação entre bancos de dados;

Existem ainda, nestes sistemas, ferramentas de suporte à programação de robôs na operação de ferramentas de processos de fabricação. Estes módulos suportam ferramentas de controle de processos como soldagem, pintura e montagem. Esta forma de programação gráfica interativa permite também a definição de trajetórias de robôs, realização de testes de colisão, verificação de erros de sintaxe e correção do código de programação.

As limitações dos processos de programação *off-line* encontram-se na dificuldade de compatibilidade entre tipos de sistemas de programação e características de diferentes fabricantes de robôs, além da complexidade de minimizar os efeitos dos erros e imprecisões entre os modelos reais e idealizados.

#### **4.5.2 Linguagens de Programação de Robôs**

A maioria das linguagens atuais de programação de robôs utiliza uma combinação entre programação textual e programação por condução (manual ou através da interface homem-máquina). Com a linguagem textual define-se a lógica e seqüência de programação, enquanto que, utiliza-se da programação por condução para a definição dos pontos específicos de localização no espaço de trabalho.

Um dos primeiros desenvolvimentos de linguagens de programação de robôs surgiu em 1973 no *Stanford Artificial Intelligence Laboratory*. A linguagem WAVE surgiu como resultado de pesquisas experimentais envolvendo um sistema de visão integrado a um robô de manipulação industrial [17]. Em 1974, iniciaram desenvolvimentos de uma nova linguagem de programação de robôs com a finalidade de controlar coordenadamente múltiplos elementos mecânicos em diferentes tarefas. Esta linguagem foi denominada AL. Da combinação de conceitos e definições das linguagens WAVE e AL surgiu a primeira linguagem comercial de programação de robôs, a linguagem VAL (*Victor's Assembly Language* – de Victor Scheinman). Introduzida em 1979 pela *Unimation* para a sua série de robôs PUMA, esta linguagem foi revisada e relançada em 1984 com a denominação de VAL II, estando atualmente na versão VAL+.

Com os avanços em hardware e software possibilitando maior flexibilização de sistemas, a programação despertou o interesse de fabricantes de robôs como objeto de estudos e pesquisas. A IBM através de seu *T. J. Research Labs*, iniciado em 1976, vem desenvolvendo duas linguagens de programação de robôs direcionadas para montagem e tarefas similares denominadas AUTOPASS e AML (*A Manufacturing Language*) [17]. A AML foi introduzida comercialmente em 1982 juntamente com os sistemas robóticos IBM. A *U.S. Air Force* em parceria com a *McDonnell-Douglas*, desenvolveu a MCL (*Manufacturing Control Language*), como evolução da linguagem de programação de comando numérico APT, enquanto que a *General Electric*, sob licença da empresa italiana DEA, introduziu em seus produtos a linguagem denominada HELP.

Outras linguagens foram surgindo como resposta às pesquisas desenvolvidas em empresas e universidades e sua utilização na indústria determinaram sua viabilidade. Como resultado de linguagens utilizadas na indústria pode-se citar ainda a RAIL, introduzida em 1981 pela *Automatix* para sistemas robóticos de montagem, soldagem e sistemas de visão, e a linguagem RAPID desenvolvida pela ABB para seus produtos. A tabela 4.2 mostra alguns fabricantes e as linguagens de programação utilizadas em seus produtos [29].

**Tabela 4.2 - Fabricantes de robôs e linguagens de programação.**

FABRICANTE	LINGUAGEM DE PROGRAMAÇÃO
ABB	ARLA, RAPID
Fanuc	KAREL
Reiss	IRL
Staubli	V+
Adept	V+
Comau	PDL2
Eshed	ACL
IBM	AML/2
Kawasaki	A5
Motoman	Inform 1, Inform 2
Nachi	SLIM
Panasonic	Parl-1, Parl-2
PSI	PSI
Samsung	FARL-II
Seiko	DARL 4
Toyota	TL-1
TQ	TQ

As linguagens de programação podem ser agrupadas em duas categorias ou gerações, em função das suas características e potencialidades [17]. A primeira geração de linguagens utiliza a combinação de estrutura textual (comandos) e procedimentos de condução do robô (posicionamento manual ou por interface). As características deste tipo de linguagem incluem a capacidade de definição da movimentação dos elementos mecânicos, a definição de sub-rotinas, interpolações lineares e circulares de trajetórias e comandos básicos de monitoração de sinais de sensores. A definição da movimentação dos elementos mecânicos se dá através de comandos para definição da seqüência de trajetórias e seus parâmetros e, para o posicionamento e definição da localização dos pontos, utiliza-se a condução manual ou por interface homem-máquina. Como exemplo de linguagem de primeira geração pode-se citar a linguagem VAL. As limitações deste tipo de linguagem estão na especificação de cálculos aritméticos para processamento durante execução do programa, impossibilidade de utilização de sensores complexos e na comunicação com outros computadores.

A segunda geração de linguagens de programação de robôs vem a superar as limitações encontradas na geração anterior através da adição de capacidades e características de programação. Estas novas características possibilitam a execução de tarefas mais complexas através do formato estruturado da construção do programa. Como exemplo deste tipo de linguagem, pode-se citar as linguagens comercialmente disponíveis como a AML, RAIL, MCL, RAPID e a VAL +.

As características para movimentação são basicamente as mesmas da geração anterior, porém a capacidade de controle de movimentação suporta geometrias mais complexas. A habilidade de manipular e monitorar dados de sensores avançados possibilita a execução de tarefas mais complexas e disponibiliza autonomia limitada ao sistema. A capacidade de monitorar informações do ambiente de trabalho e modificar comportamento mediante processamento de dados, além da habilidade de comunicação entre sistemas e computadores, são também características desta geração de linguagens de programação de robôs que estão disponíveis atualmente.

## **5 DESENVOLVIMENTO DO SISTEMA SisPRob**

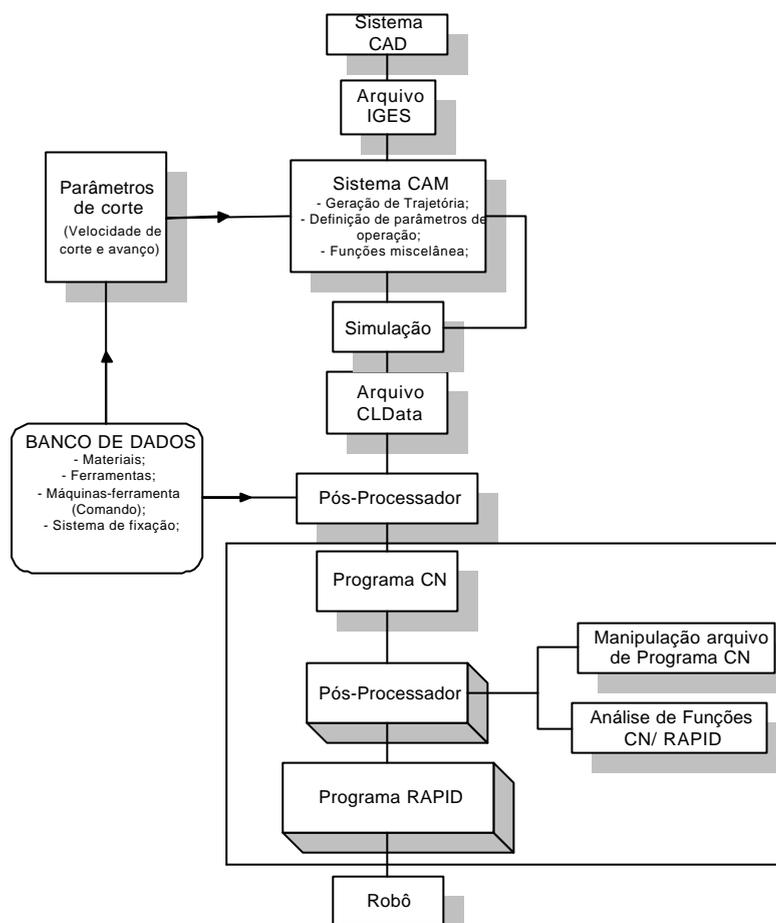
O processo de fabricação por usinagem realizado por máquinas-ferramenta tem seu desenvolvimento contínuo baseado em soluções para aumentar a produtividade e a qualidade de

produtos e processos. A redução do ciclo de projeto e fabricação de um produto e a competitividade exigem maior flexibilização de 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 (denominados CIM).

O presente trabalho apresenta o desenvolvimento de um sistema pós-processador para utilização de robôs em operações de usinagem, mais precisamente fresamento, denominado SisPRob – Sistema Pós-processador para Robô. Sua estrutura permite atender os objetivos propostos utilizando os recursos de integração entre sistemas para proporcionar flexibilidade na utilização de sistemas robóticos, além da automação do processo de programação de robôs para a execução de trajetórias e adequação de parâmetros de usinagem. O sistema objetiva a aplicação de um manipulador robótico para operações de usinagem através de programação automatizada de trajetórias.

## **5.1 CARACTERÍSTICAS GERAIS DO SISTEMA**

A estrutura geral do algoritmo implementado e sua integração aos demais sistemas utilizados é ilustrada na figura 5.1, onde as tarefas inseridas no quadro maior são as funções desenvolvidas para o aplicativo apresentado.



**Figura 5.1** - Estrutura Geral do Sistema Implementado.

O processo de automação da programação de trajetórias para usinagem através de robôs inicia-se em um software de CAD onde a informação geométrica do modelo gerado é representada analiticamente através de entidades geométricas como pontos, linhas, planos e superfícies em um arquivo gráfico de armazenagem de dados. A ligação entre os arquivos gráficos gerados em CAD e a programação CN é executada através da integração entre sistemas e funções de geração de trajetórias contidas no sistema CAM. Os arquivos gráficos gerados no sistema CAD são transferidos para os sistemas CAM através de um formato padrão de transferência de arquivos gráficos como IGES ou STEP. O software de CAM interpreta as informações deste arquivo e disponibiliza, através da manipulação do mesmo, ferramentas de auxílio à manufatura, como a geração de trajetórias e programas de operação de máquinas-ferramenta de comando numérico.

Os parâmetros de operação, como velocidade e profundidade de corte e avanço, são determinados pelo módulo de programação de comando numérico do sistema CAM, podendo ser

especificados com base em informações necessárias para o cálculo destes parâmetros presentes em um banco de dados de ferramentas e materiais com interação do operador. As trajetórias a serem percorridas pela ferramenta são geradas automaticamente sobre as entidades geométricas selecionadas através da interface gráfica do software de CAM conforme as opções de estratégias de usinagem selecionadas. O processo de geração das trajetórias e a interpolação de caminhos curvos de ferramentas em retas ou comandos de movimentação circular são realizados pelos algoritmos matemáticos implementados no módulo do sistema CAM.

O conjunto de informações de trajetórias e parâmetros de operação resultante do processamento dos dados fornecidos ao sistema com base na manipulação do arquivo gráfico é gravado num arquivo de localização de ferramenta denominado CL-Data (*Cutter Location Data*) gerado automaticamente após o final do processo interativo. Este arquivo é gerado em formato binário em sintaxe de linguagem de programação, o que impossibilita a leitura e interpretação sem o auxílio de funções de pós-processamento. O CL-Data contém as informações necessárias para a geração do programa de comando numérico através dos pós-processadores específicos para diferentes comandos de máquinas-ferramenta.

O módulo de simulação de operação reduz a possibilidade de erros no programa através da leitura e interpretação do arquivo CL-Data e verificação gráfica de inconsistências de programa e previsões de colisão de ferramenta com elementos de fixação, partes da máquina ou até mesmo com a própria peça. A simulação, além de minimizar erros permite otimizar o processo de usinagem através da automação da programação, aumento da confiabilidade e eficiência do processo e melhoria na qualidade final do produto comprovando as vantagens da programação *off-line*.

Após a verificação e correção de trajetórias, parâmetros e estratégias de operação, o pós-processador converte os dados do CL-Data em instruções e comandos para o controle de máquinas-ferramenta de comando numérico no formato da linguagem padrão ISO 66025, usualmente conhecida como código G com suas características específicas desenvolvidas para diferentes controles.

O software de CAM utilizado admite a manipulação do arquivo de pós-processamento, permitindo a customização de um pós-processador qualquer e o desenvolvimento de novos pós-processadores a partir de um padrão que contém a maioria das funções necessárias às operações convencionais de usinagem. Através da manipulação deste pós-processador é possível modificar o formato e o padrão do arquivo de texto do programa de comando numérico resultante do pós-processamento. A opção pelo desenvolvimento de um novo pós-processador em detrimento da

possibilidade de customização é justificada por limitações do processo como adaptação da sintaxe da linguagem robótica (RAPID) para o editor de pós-processador e a dificuldade de interpretação de cinco eixos para o sistema robótico.

A partir deste ponto, o sistema desenvolvido interage com os sistemas CAD/CAM. Este arquivo de texto com informações de trajetórias e parâmetros de operação é utilizado como dado de entrada no sistema apresentado para novo pós-processamento. O pós-processador desenvolvido interpreta o arquivo de texto de programa CN através da manipulação dos dados deste arquivo, correlaciona as funções do código ISO 66025 com as funções e limitações do sistema robótico utilizado gerando como resultado a automação do processo de programação de robôs. O algoritmo proporciona a conversão automática de trajetórias e adequação de funções de movimentação e parâmetros de operação.

O sistema desenvolvido consiste de funções e rotinas elaboradas em linguagem C que permite, devido às suas características de linguagem de baixo nível, portabilidade e integração com diversos sistemas e diferentes linguagens de interface gráfica. Para facilitar sua utilização, este aplicativo é associado a uma interface gráfica desenvolvida em Delphi.

Através do algoritmo implementado, são identificados comandos, funções e parâmetros de operação descritos para o comando numérico de uma máquina-ferramenta e relacionados com o sistema robótico. Uma outra possibilidade é a customização do pós-processador do próprio software de CAM que permite modificar ou recriar um novo formato de arquivo. A alteração deste módulo de pós-processamento permite redefinir características de formato de arquivo, comandos de movimentação e parâmetros de operação. Desta forma, é possível manipular os dados de entrada no sistema apresentado de forma a tornar o sistema flexível para a programação de qualquer comando numérico ou até mesmo adaptar o software de CAM para programação de outras linguagens de máquinas. Esta forma tem algumas limitações quando se refere à adaptação a programação de linguagens diferentes ao padrão ISO 66025, devido à incompatibilidade na sintaxe de comandos, parâmetros e funções de operação.

Após a geração do programa de operação do robô, o mesmo é transmitido ao equipamento através de um meio externo, pois o sistema robótico utilizado não dispõe de comunicação direta com o microcomputador.

Para o entendimento da metodologia seguida no módulo aplicativo, uma análise das características das linguagens e possibilidades de trajetórias associadas aos respectivos equipamentos é apresentada a seguir. Destacam-se ainda os sistemas de referência e as limitações e equivalências impostas para permitir a tradução entre as diferentes linguagens.

## 5.2 A CONVERSÃO DO PADRÃO ISO PARA O CÓDIGO RAPID

O sistema desenvolvido limita-se à conversão das funções e parâmetros de operação descritas no padrão ISO 66025 que são utilizados na operação do centro de usinagem (ROMI – Discovery 380) disponível no laboratório de usinagem da UFRGS em linguagem de programação de robôs RAPID para o sistema robótico utilizado (ABB – IRB1400).

### 5.2.1 O Padrão ISO 66025

As funções definidas no padrão e que são utilizadas pelo equipamento em questão são divididas em grupos de funções G e Miscelânea (M). As funções G são funções descritivas e preparatórias que definem para o comando o modo de operação do sistema. Os conceitos de modal e *default*, são utilizados na programação CNC. Um valor *default* é aquele que uma função ou parâmetro assume quando nenhum valor lhe é atribuído na programação. Já o valor modal é aquele que, quando definido, permanece ativo em blocos subsequentes até que uma nova função correspondente que a cancele seja executada. As funções também podem ser não modais, onde o modo de operação estabelecido permanece ativo apenas no bloco onde foi definido.

Para facilitar a programação, as funções G são divididas em diferentes grupos de acordo com sua aplicação. Segundo o manual da máquina-ferramenta utilizada [30], esses grupos são: Grupo de controle de coordenadas, modo dos eixos, controle de CNC, ciclos fixos, auto-rotinas e posicionamento/permanência. O grupo de controle de coordenadas referencia parâmetros que afetam o sistema de coordenada da máquina e, desta forma, o tipo de programação a ser utilizada no posicionamento dos eixos e incluem funções como G70, G71 e G90 descritas na tabela 5.1. O grupo de modos dos eixos referencia as condições sob as quais o controle interpreta o sistema de eixos da máquina definindo funções como G17, G18, G19, G40 e G42, enquanto que o grupo de controle do comando numérico computadorizado afeta as operações específicas do controle como edição de programa e interface gráfica. O grupo de ciclos fixos programa operações que ocorrem automaticamente após movimentos de posicionamento e permanecem ativas até que sejam canceladas. Suas funções incluem G80, G81 e G82 e estão descritas na tabela 5.1. O grupo de auto-rotinas, programa seqüências especiais de posicionamento ou operações, tais como fresamento de alojamentos ou círculo de furos, estas funções não foram utilizados na programação apresentada neste trabalho. As funções do grupo de posicionamento e permanência definem como o controle executa os posicionamentos e quando ocorrerão as pausas no

programa. Este grupo inclui as funções de movimentação G00, G01, G02 e G03. Dentre os grupos de funções de programação apresentados, aquele de posicionamento e permanência é o que requer maior atenção pois concentra as funções de movimentação que definem a geração da trajetória da ferramenta para a geometria requerida.

A sintaxe utilizada para este grupo de funções é apresentada no exemplo descrito a seguir e inclui além da definição da função, as coordenadas cartesianas do ponto de destino em relação ao ponto de referência da peça (zero-peça) e a velocidade de movimentação quando necessário. Para a função G00 a velocidade é previamente definida no controlador CNC, não aparecendo no bloco de instruções. As coordenadas de destino não necessitam ser descritas se não forem alteradas, assim, o comando elimina alguns caracteres da linha de instrução, diminuindo a quantidade de informação a ser transmitida. Nesse caso, mantém-se as coordenadas anteriores para os eixos que não estão descritos no bloco de instruções.

G01 X23.15 Y12. Z0.5 F500 → Exemplo de sintaxe para os comandos de movimentação G00 e G01.

Para as funções de movimentação circular a sintaxe define o sentido de movimentação, as coordenadas do ponto final e do centro da circunferência descrita. No caso da descrição de um círculo completo, o bloco de instruções define apenas o sentido de rotação e as coordenadas do centro da circunferência descrita. A seguir é descrito um exemplo de sintaxe para a função de movimentação circular.

G03 X55.491 Y19.959 I54.367 J18.305 → Sintaxe para os comandos de movimentação G02 e G03.

G03 I54.367 J18.305 → Funções G02 e G03 para circunferência completa.

As funções Miscelânea ou simplesmente funções M definem e controlam as ações de parâmetros auxiliares como troca de ferramenta, acionamento de fluido de corte e eixo árvore. Algumas destas funções são pré-definidas para o controle e execução de tarefas específicas, porém suas ações são determinadas de acordo com o comando numérico utilizado [30]. Neste sentido, a conversão destas funções para o sistema robótico referem-se a uma interpretação específica do controlador CNC utilizado.

Na tabela 5.1 é apresentado um quadro demonstrativo resumido das funções utilizadas na operação do comando numérico Romi – MACH8 instalado no laboratório de usinagem da UFRGS, utilizado neste trabalho.

**Tabela 5.1** - Principais funções do comando numérico ROMI – MACH8.

FUNÇÕES	DESCRIÇÃO
M00 / M01	Parada de Programa - As funções M00 e M01 causam uma parada na execução do programa. Quando um bloco do programa contendo as funções M00 e M01 é encontrado, a execução do programa é interrompida são mostradas no display as mensagens "PARADO" e "AGUARDANDO INICIO" .
M02 / M30	Fim de Programa - Estas funções são utilizadas para finalizar um programa ou sub-programa. Deve ser programada sozinha e no último bloco de dados do programa. Se o controle detectar sua ausência, a execução do programa é interrompida. Ambas as funções podem encerrar o carregamento do programa se seguidas de # (EOB).
M06	Parada do Programa e Troca Manual de Ferramenta - Esta função produz o mesmo efeito da função M00, porém é utilizada para gerar uma troca de ferramenta. A função desliga o eixo-árvore e movimenta os eixos para uma posição conveniente para efetuar a troca de ferramenta. Após a troca de ferramenta o programa executa o próximo bloco que retorna os eixos na posição de início de operação e religa o eixo-árvore.
M03	Rotação Direita. Aciona o eixo-árvore com rotação no sentido horário.
M04	Rotação Esquerda. Aciona o eixo-árvore com rotação no sentido anti-horário.
M05	Parada Eixo Árvore. Desliga o acionamento do eixo-árvore.
G00	Interpolação Linear Movimento Rápido - Informa aos eixos para se movimentarem ao longo de uma linha reta até o ponto programado com a velocidade rápida implementada.
G01	Interpolação Linear - Informa aos eixos para se movimentarem ao longo de uma linha reta a uma velocidade específica programada com uma função F.
G02	Interpolação Circular - Gera um arco no sentido horário. O arco é executado com movimento simultâneo dos eixos no plano de trabalho. É uma função não modal.
G03	Interpolação Circular - Gera um arco no sentido anti-horário. É também uma função não modal.
G17	Seleção do Plano de Trabalho XY
G18	Seleção do Plano de Trabalho XZ
G19	Seleção do Plano de Trabalho YZ
G40	Cancela Compensação de Ferramenta
G41	Compensação de Ferramenta à esquerda
G42	Compensação de Ferramenta à direita
G70	Sistema em Plogadas
G71	Sistema Métrico
G80	Cancela ciclos fixos
G81	Ciclo de Furação
G82	Ciclo de Rebaixo
G83	Ciclo de Furação com Tempo de Permanência
G90	Sistema de coordenadas absoluto

Um exemplo de programa CNC é apresentado no anexo 01, o mesmo corresponde à operação de desbaste da peça exemplo 2.

### 5.2.2 A Linguagem RAPID e o Sistema Robótico

O sistema robótico utilizado é composto de duas partes: O robô propriamente dito e o painel de controle. O robô consiste de elementos mecânicos (braço) interligados por juntas de movimentação que proporcionam deslocamento rotacional. O espaço de trabalho definido é uma porção de esfera e é apresentado na seção 4.4.1. O painel de controle consiste dos sistemas eletrônicos necessários para controlar o robô, eixos externos e equipamentos periféricos. Os elementos do sistema robótico são apresentados na figura 5.2.

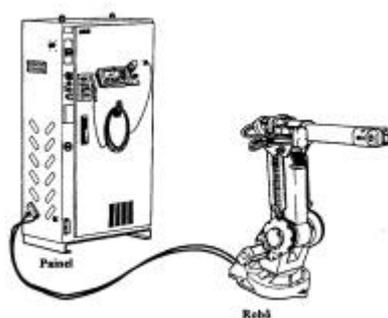
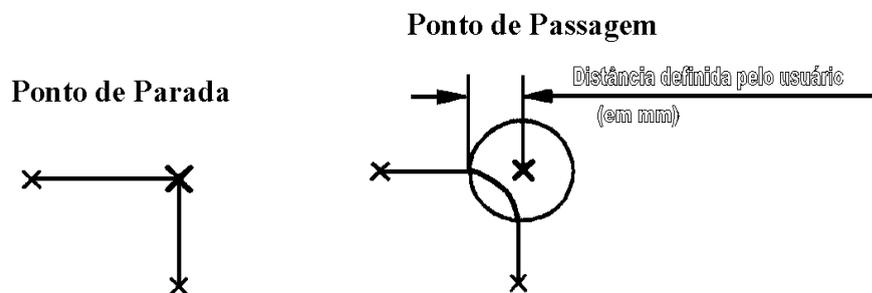


Figura 5.2 - Sistema Robótico.

A linguagem de programação de robôs denominada RAPID foi desenvolvida por um fabricante de robôs industriais para 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, como funções e procedimentos, rotinas de parâmetros, aplicações multitarefa (programas paralelos), 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 seqüência de movimentação é programada como um número de movimentos parciais entre os pontos da trajetória definida [1]. 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.



**Figura 5.3** – Identificação de parada ou zona de passagem.

Assim como em máquinas-ferramenta, a posição do robô e seus movimentos são relacionados a um sistema de referência. Para um sistema robótico existem dois sistemas de referência bem definidos. O primeiro, denominado sistema de base, corresponde a um sistema fixo, normalmente associado ao chassi do equipamento e coincidente com o sistema de coordenadas universais X – Y – Z. Um segundo sistema móvel está associado à extremidade do manipulador e refere-se ao sistema para ferramenta. Sua origem está definida num ponto denominado TCP (*Tool Central Point*), normalmente descrito como um ponto em algum lugar na ferramenta utilizada.

Na definição de trajetórias e posicionamentos dos objetos no espaço identifica-se sempre a posição e orientação do TCP em relação ao sistema de base (fixo). Esta definição é utilizada para relacionar o trajeto e a velocidade programada à ferramenta de trabalho. Para facilitar a programação, diferentes sistemas de coordenadas ainda podem ser definidos no sistema robótico.

O sistema de coordenada da base é definido na superfície da base do robô sendo o sistema de coordenadas utilizado no aplicativo desenvolvido para converter as coordenadas do sistema CNC. Neste sistema, o eixo Z é coincidente com o eixo da primeira junta do sistema robótico. Com o sistema de coordenadas da ferramenta, especifica-se o ponto central da ferramenta e sua orientação, sendo este o sistema utilizado para orientação da ferramenta no módulo de cinco eixos do pós-processador desenvolvido. O sistema robótico ainda define um sistema de coordenadas do usuário ou objeto que permite especificar como o objeto é posicionado em um dispositivo ou limitar a área de trabalho. Estes sistemas especificam cada posição em coordenadas de objeto em relação à posição da ferramenta e sua orientação, possibilitando, através de uma nova definição de ferramenta, a reutilização de programas para

diferentes aplicações. Na figura 5.4 são apresentados os diferentes sistemas de coordenadas definidos.

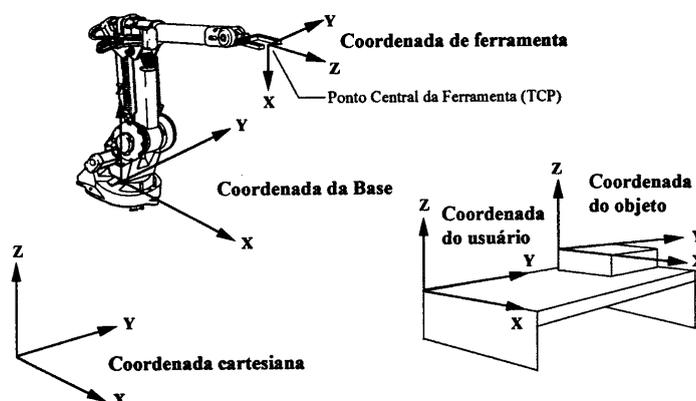


Figura 5.4 - Sistemas de coordenadas [1].

O tipo de trajetória possível para os elementos mecânicos do robô pode ser realizado através da movimentação por juntas, linear ou circular. Na movimentação por juntas, todos os eixos se movimentam individualmente e chegam na posição programada ao mesmo tempo. Nas movimentações linear e circular, os eixos se movimentam de tal forma a permitir o deslocamento do TCP em trajetória linear ou circular com aceleração e velocidade constantes. Neste tipo de movimentação, utilizada no aplicativo desenvolvido, o sistema robótico realiza uma série de cálculos de aceleração e velocidade para a cadeia cinemática com a finalidade de proporcionar um movimento constante do TCP.

Doravante, quando não for especificado nenhum sistema de coordenadas, o sistema padrão utilizado no sistema robótico é o sistema de coordenadas de base.

Os parâmetros de definição da orientação do TCP para o módulo de cinco eixos do aplicativo desenvolvido são dados em relação ao sistema de referência da ferramenta, apresentado na figura 5.5, para o ponto zero peça definido no início da operação. A orientação da ferramenta é executada com a função RELTOOL que define rotações e deslocamentos em relação ao sistema de coordenadas da ferramenta.

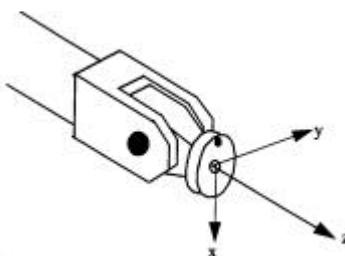


Figura 5.5 - Sistema de coordenadas da ferramenta [1].

Na linguagem do sistema robótico, os comandos para definir movimentos estão disponibilizados em três funções, respectivamente definidas como MOVEC para trajetória circular, MOVEJ para trajetória livre com movimentação definida pelas juntas, e MOVEL para trajetória linear imposta. A essas modalidades pode ser associada uma função auxiliar que permite um deslocamento incremental definido como OFFS, ou função OFFSET. Independente da modalidade, o comando deve ser completado indicando o ponto de destino, a velocidade, o grau de precisão no entorno do ponto a ser atingido, além da especificação da ferramenta de trabalho.

No trabalho desenvolvido, partindo-se sempre de um ponto de referência para a peça, às trajetórias são geradas utilizando o recurso de OFFSET, como o exemplo abaixo para a função circular.

#### **MOVEC**

```
MOVEC    OFFS(<PontoReferência>,<dx>,<dy>,<dz>),OFFS(<PontoReferência>,<dx>,<dy>,<dz>),<Velocidade>,<Zona>,<Tool>;
```

A função `offs` define os valores `dx`, `dy` e `dz` que são valores incrementais em relação ao ponto de referência “zero peça” previamente definido como variável do tipo `robotarget`. O valor incremental definido com a primeira função `offs` descreve o ponto de circularidade, sendo este uma posição na trajetória circular entre os pontos inicial e final do percurso definido. Para maior precisão de movimentação, este ponto é definido no meio da trajetória. O ponto de destino, definido pela segunda função `offs`, é a posição final da trajetória definida. O parâmetro de velocidade define, em milímetros por segundo, a velocidade de movimentação do TCP e em graus por segundo a orientação da ferramenta. Este parâmetro permite sua definição através de variáveis de velocidade do tipo `speedData`. A definição de Zona permite quantificar o desvio do ponto definido para uma mudança de trajetória com ou sem desaceleração (ponto de passagem ou parada). Também permite sua definição em forma de variável do tipo `ZoneData`. O argumento de ferramenta permite a definição de diferentes dados de ferramentas e informações do ponto TCP. Estes argumentos de velocidade, zona e ferramenta são definidos igualmente para as demais funções de movimentação. Para os comandos de movimentação MOVEL e MOVEJ, a função `Offs` define um ponto de destino que representa a posição final da trajetória definida.

#### **MOVEJ**

```
MOVEJ OFFS(<Ponto Referência>,<dx>,<dy>,<dz>),<Velocidade>,<Zona>,<Tool>;
```

#### **MOVEL**

```
MOVEL OFFS(<Ponto Referência>,<dx>,<dy>,<dz>), Velocidade, Zona,Tool;
```

O sistema robótico e a linguagem de programação utilizada ainda permitem uma variedade de funções e operacionalidades que não serão abordadas neste trabalho como controle externo de eixos, módulos de programação para aplicações específicas, controle de sinais externos para retroalimentação de programas e funções de programação de controle de fluxo.

Na tabela 5.2 é apresentado um conjunto de instruções e funções da linguagem RAPID utilizadas no desenvolvimento deste trabalho.

**Tabela 5.2** - Funções da Linguagem RAPID.

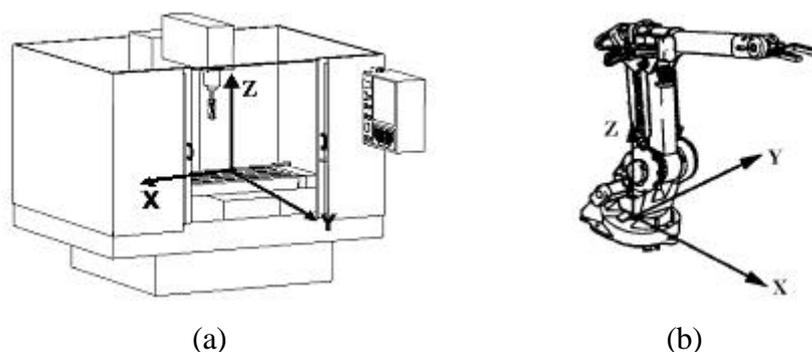
FUNÇÃO	DESCRIÇÃO	SINTAXE
WaitTime	Define Tempo de Espera - Este comando interrompe a execução do programa por um determinado tempo definido em milisegundos	WaitTime <tempo>;
Break	Interrompe a execução temporariamente - Para a execução do programa até que o operador reinicie o processo	Break;
Var	Define variáveis - Especifica o tipo e o nome das variáveis	VAR <tipo> <nome da variável>;
MoveC	Movimenta o TCP circularmente - Mantém a trajetória do TCP circular entre os pontos definidos	MoveC (<x>,<y>,<z>),<vel>,<zona>, Tool<num>;
MoveJ	Movimenta o TCP em coordenadas do robô - O robô segue de um ponto a outro através de uma trajetória definida pela adequação da cadeia cinemática para movimentação das juntas	MoveJ (<x>,<y>,<z>),<vel>,<zona>, Tool<num>;
MóveL	Movimenta o TCP em coordenada retangular - Mantém a trajetória retilínea entre os pontos especificados adequando a cadeia cinemática para manter a velocidade aceleração e orientação do TCP constante durante o movimento retilíneo	MoveL (<x>,<y>,<z>),<vel>,<zona>, Tool<num>;
Offs	Desloca uma posição - Permite um deslocamento incremental em relação ao ponto especificado e pode ser utilizado em conjunto com as funções de movimentação	Offs (<ponto de referencia>,<dx>,<dy>,<dz>);
RelTool	Desloca uma posição definida no sistema de coordenadas de ferramenta - Incrementa coordenadas e modifica a orientação da ferramenta. Utilizada em conjunto com funções de movimentação.	RelTool (<ponto de referência> <dx>,<dy>,<dz> \Rx:=<orientação em relação ao eixo X>\Ry:=<orientação em relação ao eixo Y> \Rz:=<orientação em relação ao eixo Z>
Crobt	Lê a posição corrente do robô (o "robotarget" completo) -	<variavel>:=Crobt();

	Identifica as coordenadas e orientação instantâneas do TCP com informação de estado de juntas para a posição corrente	
StopMove	Para o movimento do robô	StopMove;
StartMove	Reinicia o movimento do robô	StartMove;
TPWrite	Escreve um texto no display da interface de comunicação Homem-Máquina	TPWrite <"texto a ser exibido">;
TPReadNum	Lê uma informação numérica do teclado da interface H-M	TPReadNum <variável de armazenamento>, <"texto a ser exibido">;

Na sintaxe de instruções e funções de movimentação da linguagem RAPID, geralmente tem-se um número de argumentos associados que definem alguns parâmetros de operação ou ativam funções como visto anteriormente. Esta forma de estrutura permite definir e modificar parâmetros de funções em qualquer bloco de instruções desde que a variável do referido parâmetro esteja devidamente declarada anteriormente, o que diferencia esta linguagem das demais linguagens de programação de robôs como a VAL+. Um exemplo de programa RAPID para fresamento é apresentado no anexo 02. Este exemplo corresponde à conversão do arquivo de programa CN para execução da peça-exemplo 1.

### 5.2.3 Conversão Código G x Código RAPID

O sistema SisProRob correlaciona as funções do código ISO 66025 com as funções da linguagem RAPID de acordo com sua forma de execução. Algumas funções têm uma correspondência mais direta entre as linguagens como as funções de movimento linear G00 e G01, que são interpretadas como funções MOVE na linguagem RAPID. Esta conversão não se realiza diretamente devido à diferença nos sistemas de referência do CNC e do sistema robótico utilizado, além de particularidades de sintaxe entre os códigos de programação. Os diferentes sistemas de coordenadas para o CNC e o sistema robótico são apresentados na figura 5.6.



**Figura 5.6** - Sistemas de coordenadas: (a) Fresadora CNC; (b) Sistema Robótico.

As funções da linguagem de comando numérico como as de interpolação circular, ciclos fixos e funções miscelânea são analisadas segundo critérios adequados de conversão para sua adaptação ao sistema robótico.

O controlador CNC utiliza duas funções para gerar trajetórias circulares com movimentação simultânea de eixos (G02 e G03). Este movimento é denominado interpolação circular porque o controlador coordena movimentos individuais simultâneos dos eixos para produzir um arco. A sintaxe destas funções define, além da própria função, as coordenadas do ponto final de movimentação no plano de trabalho e as coordenadas do centro da trajetória circular. Para construção de um círculo completo, a sintaxe suprime a indicação do ponto final deixando apenas o valor das coordenadas do centro de giro como descrito no item anterior. A velocidade de avanço, como uma função modal, pode ou não ser modificada.

A função de trajetória circular para linguagem RAPID diferencia-se em sua sintaxe e forma de execução. A sintaxe desta função define, além da própria função MOVEC, as coordenadas de um ponto intermediário no arco descrito, as coordenadas do ponto final da trajetória e os parâmetros padrões deste tipo de função já descritos no item 4.2.2. Essa função na linguagem RAPID apresenta uma particularidade que dificulta a conversão do código G que é a não geração de trajetórias circulares completas. Para uma correta conversão de trajetórias circulares é necessário à segmentação da trajetória circular em dois arcos.

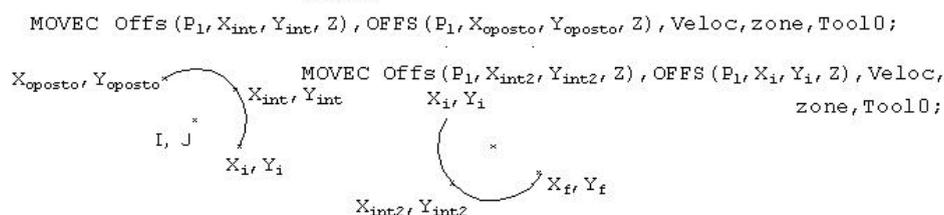
No sistema SisPRob realiza-se esta segmentação de trajetória circular em dois arcos complementares através de um algoritmo que identifica o tipo de trajetória circular, se completa ou incompleta (arco), e define diferentes fluxos de controle de programa. No caso de uma trajetória circular completa, o algoritmo utiliza as coordenadas do bloco anterior como coordenadas finais para o segundo arco e compara-as com as coordenadas do centro de rotação. Conforme a localização no sistema de coordenadas cartesianas do ponto central em relação à coordenada final e o sentido de rotação da função interpretada, pode-se obter, através da equação da circunferência, o raio da circunferência e as coordenadas de um ponto oposto ao inicial. Este ponto é então armazenado como ponto final na definição de um primeiro arco para descrição da trajetória circular. O resultado do procedimento descrito é ilustrado na figura 5.7.

ISO 66025

G03/02 I J



RAPID



**Figura 5.7 - Segmentação da trajetória circular.**

Desta forma se obtêm um ponto intermediário na circunferência que proporciona a segmentação da circunferência completa em dois arcos para definição da função MOVEC.

Para a descrição de trajetórias circulares incompletas (arcos) o processo de conversão da função CN em linguagem RAPID é similar ao processo para obtenção da trajetória circular completa. Para a obtenção do ponto de circularidade necessária para a descrição do comando MOVEC foram utilizadas propriedades trigonométricas para a obtenção da equação da circunferência. Através da solução de equações do segundo grau e testes lógicos, obteve-se o algoritmo de segmentação da trajetória circular. O ponto de circularidade obtido localiza-se próximo à posição mediana da trajetória circular definida, proporcionando maior precisão na execução da função de movimentação.

Para as funções M, foram implementados módulos complementares como interface de comunicação e interação do operador com o programa, uma vez que as funções de troca e acionamento de ferramentas, sentido de rotação e pausa de operação não são funções automaticamente realizáveis pelo sistema robótico utilizado. A cada função miscelânea, de movimentação e ciclos fixos interpretados, são adaptados os movimentos e comandos implícitos nestas funções para reprodução similar pelo sistema robótico. Estas adaptações são apresentadas no quadro comparativo com as relações das funções de movimentação, funções miscelânea e ciclos fixos interpretados e sua correspondência em linguagem RAPID na tabela 5.3.

**Tabela 5.3 - Correlação das funções de movimentação ISO x Linguagem RAPID.**

ISO 66025	Linguagem RAPID
M00 e M01 - Parada de Programa	TPWrite "Programa Parado"; TPWrite "Para recomencar aperte start ou fwr na barra de menu"; Stop;
M02 e M30 - Fim de Programa	TPWrite "Fim de Programa"; TPWrite "Para recomencar aperte start na barra de menu"; Stop;

M03 - Aciona Eixo árvore no sentido horário	TPWrite "Acione a Ferramenta"; TPReadNum opcao, "DIGITE 1 para OK"; if opcao <> 1 Break;
M04 - Aciona Eixo árvore no sentido anti-horário	Tpwrite "Atenção: Acionamento da ferramenta em sentido anti-horario"; TPwrite "Acione a Ferramenta"; TPReadNum opcao, "DIGITE 1 para OK"; if opcao <> 1 Break;
M05 - Pára o acionamento do eixo-árvore	TPwrite "Atencao: Pare o acionamento da ferramenta"; TPReadNum opcao, "DIGITE 1 para OK"; if opcao <> 1 Break;
M06 - Troca de Ferramenta	znum+=200; MoveL Offs (P1,x,y,znum), v2000, zona, tool0; TPWrite "Trocar Ferramentas"; TPReadNum opcao, "Digite 1 Qdo Ok"; If opcao <> 1 Break; znum-=200; MoveL Offs (P1,x,y,znum), v2000, zona, tool0;
M07 - Aciona Refrigeração	TPWrite "Refrigeracao por nevoa"; TPWrite "Acionar refrigeracao"; TPReadNum opcao, "Qdo Ok digite 1"; If opcao <> 1 Break;
M09 - Desliga Refrigeração	TPWrite "Desligar Refrigeração"; TPReadNum Opcao, "Qdo Ok Digite 1"; If opcao <> 1 Break;
G81 X Y Z R F Ciclo Fixo de Furação	MOVEJ Offs (P1,x,y,r), v2000, zona, tool0; MOVEL Offs (P1,x,y,zfura), veloc, zona, tool0; MOVEL Offs (P1,x,y,r), v2000, zona, tool0;
G82 X Y Z R F P Ciclo Fixo de Rebaixo	MOVEJ Offs (P1,x,y,r), v2000, zona, tool0; MOVEL Offs (P1,x,y,zfura), veloc, zona, tool0; WaitTime d; MOVEL Offs (P1,x,y,r), v2000, zona, tool0;
G83 e G73 X Y Z Q R F Ciclo Fixo de Furação com Tempo de Permanência	MOVEL Offs (P1,x,y,r), v2000, zona, tool0; zincr=r-iciclo; while (zincr <= z) { zincr-=iciclo; veloc.v_tcp:=fsai; MOVEL Offs (P1,x,y,zinc), veloc, zona, tool0; WaitTime delay; MOVEL Offs (P1,x,y,r), v2000, zona, tool0; } // fim do while
G00 X Y Z Interpolação Linear com Movimento Rápido	veloc.v_tcp:=f; MOVEJ Offs (P1,x,y,z), veloc, zona, tool0;
G01 X Y Z F Movimentação Linear	veloc.v_tcp:=f; MOVEL Offs (P1,x,y,z), veloc, zona, tool0;
G02 e G03 X Y Z I J Movimentação Circular (Exemplo para plano xy)	veloc.v_tcp:=f; MOVEC Offs (P1,valor1,valor2,z), Offs (P1,xoposto, yoposto,z), veloc, zona, tool0; MOVEC Offs (P1,valor1,valor2,z), Offs (P1,x,y,z), veloc, zona, tool0;
G00 X Y Z A B Movimentação Linear de posicionamento (Exemplo para 5 eixos)	MOVEJ Reltool (P1, -1*X, y,-1*Z\Rx:=()\Ry:=()), veloc, zona, tool0;

G01 X Y Z A B	MOVEJ Reltool (P1, -1*X, y, -1*Z\Rx:=()\Ry:=()),
Movimentação Linear	veloc, zona, tool0;
(Exemplo para 5 eixos)	

Para as funções de controle de coordenadas e modos dos eixos são geradas variáveis que definem rotinas a serem seguidas no controle de fluxo do programa. Estas funções alteram alguns parâmetros e a forma de algumas das funções de movimentação finais porém, a estrutura do algoritmo permanece a mesma.

As funções de modos dos eixos G17, G18 e G19 referem-se a planos de trabalho e interferem no cálculo das coordenadas de movimentação para interpolação circular. Neste caso, uma variável armazena o plano de trabalho referenciado e desvia o fluxo do programa para a sub-rotina do plano de trabalho especificado fazendo com que o cálculo seja realizado segundo o plano de trabalho selecionado. Para as funções de correção de ferramentas G40, G41 e G42, a opção é disponibilizada na geração da trajetória durante o processo interativo de programação CNC no software de CAM utilizado.

Para as funções de controle de coordenadas incrementais ou absolutas, G90 ou G91, o controle é realizado no próprio pós-processador do software de CAM. O pós-processador é customizado para somente operar com coordenadas absolutas para evitar o aumento significativo de linhas de programa, otimizando o tamanho do programa gerado. Isto é necessário devido à limitação de memória de armazenagem de dados do sistema robótico utilizado.

As funções de ciclos fixos, como os ciclos de furação G73, G81, G82 e G83 são interpretados segundo suas seqüências de movimentação e seus parâmetros de operação para realização de determinada função. As interpretações destas funções estão exemplificadas na tabela 5.3. Algumas das funções de ciclos fixos disponíveis para o controlador MACH-8 não foram interpretadas por não serem operações executáveis no sistema robótico, devido às limitações de ferramentas e rigidez do sistema.

O referenciamento da peça ou zero-peça é realizado similarmente ao processo conhecido em máquinas-ferramenta, e para tanto foi desenvolvida uma seqüência de programação na linguagem RAPID que permite a comunicação com o operador para definir o eixo a ser referenciado e o posicionamento manual do braço robótico para referenciamento. A seguinte rotina permite o referenciamento da peça.

```

TPWrite "Referencie X!";
Stop \noregain;
pontox:=Crobt();

TPWrite "Referencie Y!";
Stop \noregain;
pontoy:=Crobt();

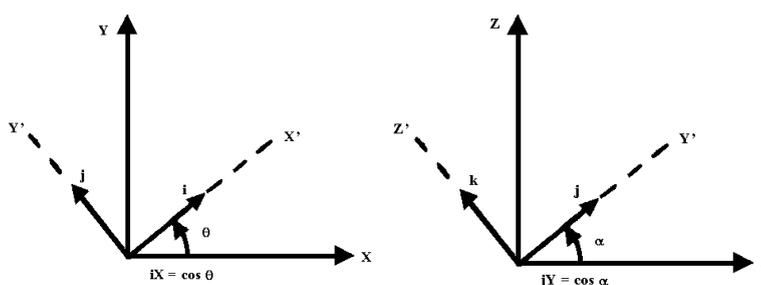
TPWrite "Referencie Z!";
Stop \noregain;
pontoz:=Crobt();

P1:= pontox;
P1.trans.y:= pontoy.trans.y;
P1.trans.z:= pontoz.trans.z;

```

**Figura 5.8 -** Referenciamento em linguagem RAPID.

Para o módulo de programação de cinco eixos as funções de movimentação em código ISO 66025 limitam-se as movimentações lineares G00 e G01. Neste pós-processamento a dificuldade de interpretação está em calcular os ângulos de deslocamento dos eixos e adequar a estrutura e a sintaxe dos comandos para as funções da linguagem RAPID. Para possibilitar o movimento de translação através de coordenadas incrementais em relação ao ponto de referência zero-peça e permitir rotações do pulso do robô para orientação da ferramenta foram utilizadas as funções RELTOOL e OFFS. O processo de conversão para cinco eixos implica na interpretação direta dos movimentos lineares G00 e G01 em MOVEJ para linguagem RAPID e a conversão dos ângulos de orientação da ferramenta. A função RELTOOL permite a orientação do ponto TCP em relação ao sistema de coordenadas da ferramenta definido no referenciamento da peça. Para a obtenção dos ângulos de orientação utilizam-se os parâmetros A e B do código ISO que se referem à projeção dos vetores unitários de orientação da ferramenta nas coordenadas cartesianas. Assim, através de propriedades trigonométricas obtém-se o ângulo de deslocamento da ferramenta para o sistema robótico como apresentado na figura 5.9.



**Figura 5.9 -** Vetor unitário de orientação da ferramenta.

O ângulo de movimentação em relação ao eixo X é calculado como o arco-seno do ângulo de deslocamento do sistema de referência incremental aqui denominado de ângulo  $q$  apresentado na figura 5.9. De forma semelhante é definido o ângulo de movimentação em relação ao eixo Y onde o ângulo de deslocamento é apresentado na figura 5.9 como ângulo  $a$ .

Desta forma, a sintaxe para os comandos de movimentação no módulo cinco eixos combinam posicionamento e orientação através da utilização conjunta das funções RELTOOL e OFFS no comando MOVE. Neste caso, o movimento de orientação da ferramenta é realizado segundo a função RELTOOL, e seus parâmetros são definidos segundo a orientação do sistema de coordenadas da ferramenta no instante da memorização do ponto de referência ou zero peça. Por conseguinte, para uma correta conversão de trajetórias, é necessário atentar para o fato de manter em zero graus o ângulo de movimentação do eixo seis do robô no momento do referenciamento do ponto zero peça. Desta forma garante-se a correlação entre os sistemas de coordenadas da ferramenta e o sistema de coordenadas da base permitindo uma correta conversão da orientação da ferramenta para a linguagem RAPID.

## **5.3 IMPLEMENTAÇÃO COMPUTACIONAL**

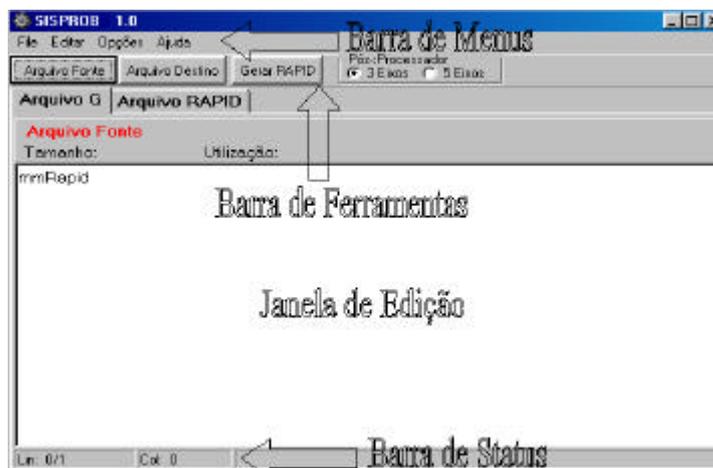
### **5.3.1 Software**

No desenvolvimento deste trabalho foram utilizados o SolidWorks e o SurfCam respectivamente como software de CAD e software de CAM. A escolha por estes aplicativos é justificada pela capacidade de integração entre sistemas, ferramentas de geração de trajetórias e estratégias de usinagem em conformidade com as necessidades de projeto e pela disponibilidade no laboratório de usinagem da UFRGS. Como software de programação para a interface visual foi utilizado o Delphi e o algoritmo de conversão foi implementado em linguagem C.

### **5.3.2 Interface de Pós-Processamento**

O módulo de software responsável pela conversão das linguagens tem como finalidade facilitar o processo de programação de trajetórias para sistemas robóticos proporcionando a sua adaptação e utilização em diferentes operações de manufatura mais precisamente operações de usinagem como fresamento. Para tanto foi desenvolvida uma interface gráfica interativa entre o usuário e o processo de fabricação, capaz de se integrar aos sistemas CAD/CAM facilitando o processo de programação e operação de sistemas robóticos. Esta interface é composta de menus e quadros de edição de texto.

Este item trata das instruções de operação do sistema apresentado. Para iniciar a operação do sistema, basta abrir o arquivo executável SisPRob e certificar-se que os aplicativos executáveis do algoritmo pós-processador desenvolvido estejam gravados junto ao diretório onde se encontra o arquivo SisPRob. A interface gráfica desenvolvida para o aplicativo SisPRob é apresentada na figura 5.10.



**Figura 5.10** - Tela do SisPRob.

A tela de apresentação do aplicativo SisPRob pode ser dividida em áreas distintas:

- Janelas de trabalho ou edição;
- Barra de Menus;
- Barra de Ferramentas;
- Barra de status;

A janela de trabalho é a área reservada para edição e manipulação de arquivos de texto. Este espaço é dividido em duas planilhas: Uma contendo o arquivo fonte ou arquivo a ser pós-processado e uma segunda planilha para exibição do programa convertido em linguagem RAPID. Nestas planilhas de edição pode-se abrir um arquivo de texto qualquer e manipulá-lo conforme necessidade de projeto ou simplesmente editar um arquivo novo para pós-processamento.

Na barra de menus encontram-se as opções de ARQUIVO, EDITAR, OPÇÕES e AJUDA. No menu ARQUIVO encontram-se as funções básicas de manipulação do arquivo de origem que permitem abrir, salvar e imprimir o arquivo fonte bem como a opção de sair do programa. A opção abrir carrega um quadro de diálogo para seleção do arquivo fonte a ser aberto. Na opção salvar novamente um quadro de diálogo é aberto para nomear o arquivo que deseja gravar e selecionar o diretório para sua armazenagem. Na função imprimir, o quadro de diálogo permite selecionar a impressora a ser utilizada e configurar as propriedades desejadas

para sua impressão. Todas estas funções são características comuns em aplicativos desenvolvidos para ambiente Windows.

No menu EDITAR da barra de menus têm-se as funções de edição de texto como copiar, colar e localizar. No menu OPÇÕES encontra-se a função fonte que permite alterar o tamanho e estilo da fonte de edição dos textos. E para finalizar o menu AJUDA onde se encontra a função ‘sobre’ que apresenta alguns dados do desenvolvimento deste aplicativo.

A barra de ferramentas permite acesso às funções de conversão. Neste aplicativo tem-se as funções de abrir arquivo de origem, salvar arquivo de programa RAPID, selecionar o módulo de três ou cinco eixos e a função de pós-processamento designada para gerar o código RAPID.

A barra de status localizada na parte inferior da tela, apresenta a informação do estado atual do cursor como linha e coluna em que o mesmo se encontra além do número total de linhas do texto da planilha selecionada. Como função auxiliar da barra de status tem-se também o tamanho total do arquivo em bytes. Esta função auxilia no controle da quantidade máxima de bytes disponíveis para o armazenamento no sistema robótico que neste caso está limitada em 130 Kbytes.

Para operação do aplicativo deve-se abrir um arquivo de texto com a programação em código ISO 66025, ou editar um novo programa na janela de edição do aplicativo, selecionar o módulo de três ou 5 eixos na barra de ferramentas, salvar o arquivo de destino e nomeá-lo e então gerar o programa RAPID através da função “Gerar RAPID” disponível na barra de ferramentas. Como resultado a planilha com a janela de edição do programa RAPID aparecerá exibindo o programa RAPID pós-processado. O programa pós-processador pode então ser diretamente executado no controle do robô.

### **5.3.3 Hardware**

O aplicativo foi desenvolvido e testado em um microcomputador PC com processador de 750 MHz Intel com 128 Mb de memória RAM. Porém, esta configuração de equipamento não é requisito para sua utilização. Configurações inferiores podem ser utilizadas para sua operação.

## **6 APLICAÇÕES DO MODELO PROPOSTO**

O sistema apresentado integra-se aos demais sistemas de auxílio a projeto e manufatura utilizando as características individuais de cada sistema independente, proporcionando a

flexibilização da utilização de sistemas robóticos. Um sistema CAD é utilizado para a modelagem geométrica da peça a ser produzida. Do sistema CAM são utilizadas as ferramentas de geração de trajetórias e parâmetros de operação através da manipulação do arquivo de dados geométricos obtidos na modelagem. O sistema, como desenvolvido, implementa um pós-processador capaz de converter os dados de programas de comando numérico em instruções de operação de robôs em linguagem RAPID, possibilitando a automação no processo de programação de trajetórias e parâmetros para utilização de robôs em operações de fresamento.

Conforme apresentado em capítulos anteriores, existem diferentes critérios e métodos de caracterização de um sistema computadorizado de auxílio à manufatura. O aplicativo desenvolvido caracteriza-se por ser um sistema interativo de programação que utiliza o método *off-line* de programação de robôs. O sistema apresentado insere-se na integração entre sistemas CAD/CAM permitindo maior flexibilização na utilização de equipamentos programáveis e caracteriza-se como interativo pois o processo de geração de trajetória, definição de estratégias e parâmetros de operação são realizados através do constante envolvimento de um operador no processo decisório para posterior pós-processamento.

Como verificação da viabilidade operacional deste sistema foram desenvolvidos estudos de casos na geração de superfícies complexas através da operação de fresamento com utilização de um robô.

## **6.1 CARACTERÍSTICAS GERAIS**

Com a finalidade de testar todas as etapas do sistema proposto, foram executados testes práticos, como estudos de caso, envolvendo os equipamentos de processo e o software necessário.

Todos os estudos de caso apresentados 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 para operação em sistemas robóticos e a fabricação por fresamento com a utilização de um robô antropomorfo. O objetivo destes estudos de caso é comprovar a viabilidade da conversão de trajetórias e adequação do sistema robótico a operações de usinagem através do sistema pós-processador SisPRob desenvolvido.

## 6.2 MATERIAL USINADO

Nos testes experimentais, para geração de protótipos, utilizaram-se materiais de baixa resistência mecânica devido às limitações de rigidez impostas pelo manipulador robótico. Alguns dos materiais testados foram o poliuretano expandido e materiais polímeros para explorar diferentes geometrias e trajetórias. Dentre esses materiais, os que apresentaram melhores resultados foram os polímeros denominados por seus fabricantes como Cibatool e Ren Shape. Estes materiais são utilizados para criação de modelos originais na indústria automobilística, aeroespacial, eletroeletrônica, etc. São materiais desenvolvidos para uma fabricação eficiente, com propriedades finais que proporcionam performance em aplicações específicas e boas características de usinabilidade.

Os estudos realizados apontaram o Cibatool como material de melhor resultado para a aplicação pretendida devido às suas propriedades mecânicas que possibilitaram uma usinagem com acabamento superficial de boa qualidade e a geração de baixos esforços de corte, não superando as limitações de rigidez do sistema robótico na realização dos testes propostos.

## 6.3 EQUIPAMENTOS

Os equipamentos de processo utilizados na realização deste trabalho foram um centro de usinagem vertical ROMI DISCOVERY 308, equipado com o comando MACH 8, potência de 5 Cv e capacidade de rotação máxima de  $4000 \text{ rpm}$  e um robô ABB modelo IRB-1400 do tipo antropomorfo com seis eixos de movimentação disponíveis no laboratório de Usinagem/Robótica da Universidade Federal do Rio Grande do Sul (GPFAl-UFRGS). Modelos muitos semelhantes foram gerados no equipamento CNC e no sistema robótico.

Ao último elemento mecânico do robô foi adaptada uma ferramenta de retífica pneumática REITZ modelo *ER-220 / 35 N-7* (figura 6.1) com rotação máxima de  $18000 \text{ rpm}$ , utilizada como acionamento para uma fresa de aço-rápido de dois gumes e  $4\text{mm}$  de diâmetro.



**Figura 6.1** - Ferramenta pneumática.

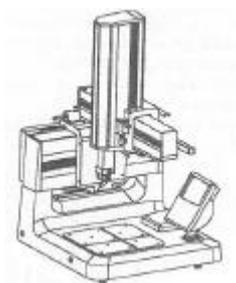
Para possibilitar o trabalho de usinagem com o robô foi preciso adaptar uma base de fixação de peças para o sistema, além de um mecanismo de fixação da ferramenta pneumática ao

pulso do robô. Para a base de fixação da peça utilizou-se uma mesa de fresadora com a finalidade de proporcionar maior rigidez ao sistema. O mecanismo de fixação da retífica foi projetado com o objetivo de adaptar a ferramenta pneumática ao braço robótico de forma a minimizar os efeitos de vibração, proporcionar rigidez ao cabeçote acionador e evitar colisões entre a ferramenta e os elementos mecânicos do robô. Devido à limitação de carga do sistema robótico, o material utilizado para a fabricação do dispositivo de fixação do cabeçote foi o alumínio. Sua escolha justifica-se pela satisfatória relação entre peso e rigidez para a sustentação da ferramenta. Esse mecanismo de suporte da ferramenta tem fixação direta no flange do robô através de parafusos e está demonstrado na figura 6.2.



**Figura 6.2** - Mecanismo de fixação da ferramenta.

Na aferição dimensional para as peças produzidas como estudos de caso utilizou-se uma unidade processadora de dados 3D Mitutoyo - QM-Measure 353 com resolução de 0,005 mm apresentada na figura 6.3.



**Figura 6.3** - Unidade processadora 3D.

## 6.4 PROCEDIMENTO

O procedimento experimental adotado para a realização dos estudos de caso foi o mesmo para todas as peças exemplo e segue o esquema apresentado na figura 6.4.



**Figura 6.4** - Procedimento experimental.

## 6.5 PARÂMETROS DE OPERAÇÃO

Na literatura consultada não se encontraram referências para parâmetros de corte relativos a operações de usinagem com a utilização de sistemas robóticos. No planejamento do processo e dos parâmetros de operação procurou-se compatibilizar a rigidez do sistema utilizado e as propriedades do material a ser usinado com a operação a ser realizada. Como princípio para o estabelecimento de parâmetros de operação para o sistema robótico, partiu-se da utilização dos mesmos parâmetros para máquinas-ferramenta, adaptando seus valores às limitações dos materiais e ferramentas utilizadas no desenvolvimento deste trabalho. O sistema robótico ainda apresenta outras limitações para operações de usinagem nas condições testadas, como a deficiência de controle gradual de rotações e o baixo torque da ferramenta pneumática, além da baixa rigidez do sistema, tanto de fixação da peça quanto dos elementos mecânicos do robô. Todas estas limitações foram superadas através de adequações de parâmetros de operação e escolha de materiais e operações compatíveis às aplicações desenvolvidas. A realização de testes experimentais só foi possível devido à opção por materiais de baixa rigidez que permitiram a operação sem a geração de elevadas forças de corte. O resultado da adequação de parâmetros para operação de usinagem com o robô para os estudos de casos desenvolvidos é demonstrado na tabela 6.1.

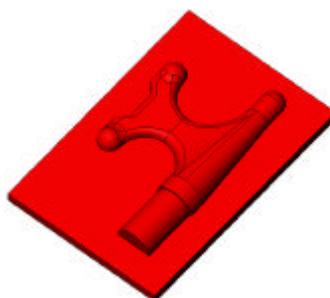
**Tabela 6.1** - Parâmetros utilizados na operação do sistema robótico.

OPERAÇÃO	ROTAÇÃO (rpm)	AVANÇO (mpm)	PROFUNDIDADE DE CORTE (mm)
DESBASTE	6000 – 7500	2,5 – 4,5	2,0 – 2,5
ACABAMENTO	7000 - 7500	2,0 – 2,5	0,8 – 1,5

A seleção destes parâmetros de operação se deu através de ensaios de usinagem testando-se diferentes condições para confecção de cavidades em um mesmo material adotando-se, em função do acabamento e limitações da ferramenta, os parâmetros indicados.

A seguir são apresentados alguns dos estudos de caso realizados experimentalmente.

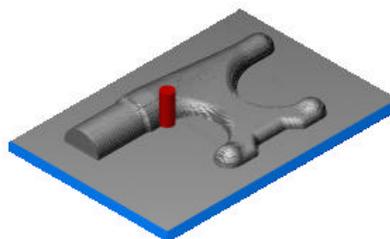
## 6.6 ESTUDO DE CASO: PEÇA EXEMPLO 1



**Figura 6.5** - Peça exemplo 1 – Modelo em CAD.

A primeira etapa no processo de geração do programa da peça para usinagem com auxílio de um robô corresponde a sua modelagem em um sistema CAD. O modelo gerado é então transferido ao sistema CAM através de um arquivo padrão de transferência de dados, no formato STEP para a geração de trajetórias e estratégias de operação. A opção pelo modelo apresentado na figura 6.5 se deve a complexidade geométrica que permite avaliar a conversão de trajetórias e parâmetros do sistema SisPRob. O processo de definição de trajetórias e parâmetros de operação se dá de forma interativa no módulo de programação do sistema CAM. Para a fabricação desta peça exemplo foram utilizadas uma operação de desbaste e duas operações de acabamento.

As operações foram testadas e visualizadas no módulo de simulação do sistema CAM e o resultado desta estratégia de operação é demonstrado na figura 6.6.



**Figura 6.6** - Simulação das estratégias implementadas para peça exemplo 1.

Após a simulação das estratégias de usinagem é gerado o programa da peça em código ISO 66025 (programa CN) através de um pós-processador customizado para o comando MACH-8. Este arquivo é utilizado como dado de entrada no sistema SisPRob para sua conversão em instruções de operação de robô na linguagem RAPID. Um exemplo de programa CN da peça exemplo 1 em código ISO 66025 (comando MACH 8) e Linguagem RAPID estão demonstrados

nos anexos 01 e 02 respectivamente. A etapa seguinte consiste em carregar o programa de operação na unidade controladora do sistema robótico.

A operação do robô inicia com o referenciamento da peça, processo semelhante ao realizado na operação de máquinas-ferramenta onde se informa ao controlador um ponto de referência para o controle de movimentos dos eixos coordenados. A figura 6.7 apresenta o ponto de referencia zero peça utilizado.

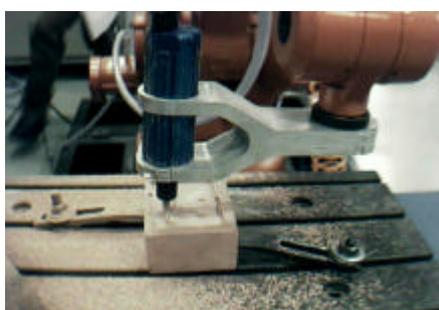


**Figura 6.7** - Referência zero peça.



**Figura 6.8** - Operação de desbaste.

Após o referenciamento o programa inicia a execução das trajetórias de usinagem como mostrado nas figuras 6.9, 6.10 e 6.11.



**Figura 6.9** - Etapa intermediária de desbaste.



**Figura 6.10** - Início de operação de acabamento.

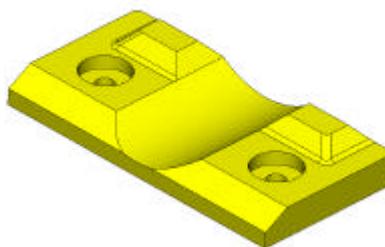


**Figura 6.11** - Etapa Intermediária de acabamento.



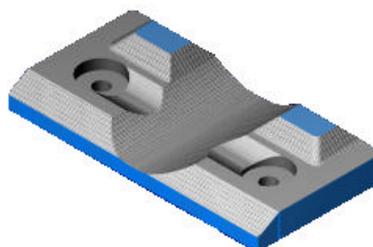
**Figura 6.12** - Peça exemplo 1 concluída.

## 6.7 ESTUDO DE CASO: PEÇA EXEMPLO 2



**Figura 6.13** - Peça exemplo 2 – Modelo em CAD.

As etapas do desenvolvimento deste estudo de caso são semelhantes ao estudo de caso 1 diferenciando-se pelas definições de trajetórias e estratégias de operação. O modelo apresentado na figura 6.13 permite avaliar a adaptação de funções de ciclos fixos como furação à linguagem RAPID através do pós-processador SisPRob. Foram utilizadas uma operação de desbaste, e duas operações de acabamento para a obtenção da geometria definida. O resultado da estratégia de usinagem utilizada pode ser visualizado no módulo de simulação do sistema CAM demonstrado na figura 6.14.



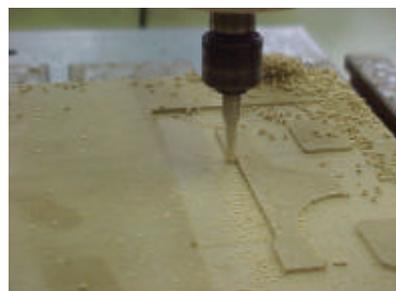
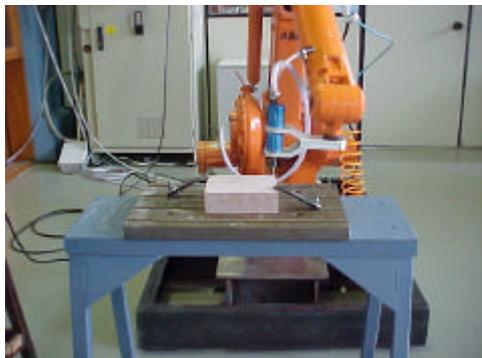
**Figura 6.14** - Simulação da estratégia de usinagem implementada para peça exemplo 2.

Na figura 6.15 é apresentada a operação de furação através da utilização do robô guiando a ferramenta com acionamento pneumático.

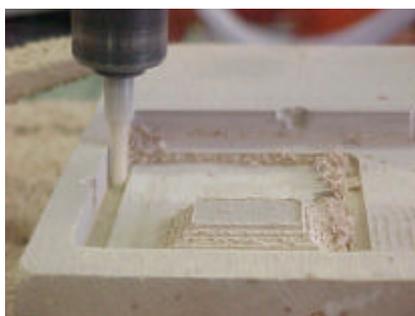


**Figura 6.15** - Ciclo fixo de furação.

Nas figuras 6.16, 6.17, 6.18 e 6.19 é ilustrado o processo de execução de trajetórias e o sistema montado para a realização dos testes experimentais.



**Figura 6.16** - Equipamento de testes experimentais. **Figura 6.17** - Início do processo de desbaste.



**Figura 6.18** - Operação de desbaste.



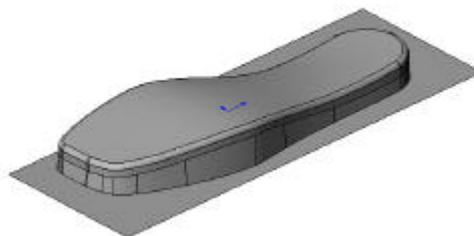
**Figura 6.19** - Execução de cavidade.

O resultado do processo de fabricação por fresamento com a utilização de um robô da peça exemplo 2 é mostrado na figura 6.20.



**Figura 6.20** - Peça exemplo 2 concluída.

### 6.8 ESTUDO DE CASO: PEÇA EXEMPLO 3



**Figura 6.21** - Peça exemplo 3 – Modelo em CAD.

O modelo geométrico apresentado na figura 6.21 ilustra outro modelo de peça que permite avaliar a conversão e a viabilidade de fabricação de superfícies complexas com a utilização de um robô industrial. Realizada em uma operação de desbaste, e três operações de acabamento para a obtenção da geometria definida. O resultado da estratégia de usinagem utilizada pode ser visualizado no módulo de simulação do sistema CAM demonstrado na figura 6.22.



**Figura 6.22** - Simulação das estratégias de usinagem.

Nas figuras 6.23, 6.24, 6.25 e 6.26 ilustra-se a execução das trajetórias de usinagem.



**Figura 6.23** - Início de operação.



**Figura 6.24** - Operação de desbaste.



**Figura 6.25** - Operação de acabamento.



**Figura 6.26** - Fase intermediária de acabamento.

O resultado final para a fabricação do modelo apresentado pode ser visualizado nas figuras 6.27 e 6.28.



**Figura 6.27** - Fase final de acabamento.



**Figura 6.28** – Peça Exemplo 3 concluída.

## 6.9 ESTUDO DE CASO 4

Este estudo de caso se refere à aplicação do sistema SISPROB para a geração de trajetórias e estratégias de operação utilizando cinco eixos para movimentação. Devido à limitação da capacidade de armazenamento de dados do sistema robótico e do significativo acréscimo de tamanho de arquivo quando se trabalha em cinco eixos, o modelo apresentado refere-se apenas a uma operação de acabamento na superfície da peça exemplo 3. Esta operação permitiu avaliar a viabilidade do sistema apresentado para a referida superfície.

Nas figuras 6.29 e 6.30 apresenta-se a execução das trajetórias para o estudo de caso.



**Figura 6.29** - Acabamento em modo 5 eixos.**Figura 6.30** - Execução de trajetória 5 eixos.

Para a execução deste estudo de caso, algumas limitações foram impostas pelo sistema utilizado. Entre elas a posição do acionamento da ferramenta que limita sua forma de fixação ao manipulador, tornando o processo de manipulação sem colisão mais complexo e a dificuldade em prever a configuração angular das juntas mecânicas que determinam o fim de curso e as possibilidades de obtenção da posição desejada através do cálculo da cadeia cinemática.

Como forma de superar estas limitações e realizar um exemplo prático de estudo de caso, foram definidas características de operações específicas tais como reduzido espaço de trabalho, baixas velocidades de corte e ponto zero peça definido o mais próximo possível dos valores zeros para o posicionamento angular de cada junta do manipulador robótico. Desta forma, para o estudo de caso apresentado, superou-se as limitações do sistema comprovando a viabilidade do pós-processador desenvolvido.

## 6.10 RESULTADOS DE MEDIÇÕES

Com o objetivo de prover dados que pudessem demonstrar a qualidade do resultado do procedimento experimental foram realizadas aferições dimensionais na peça exemplo 2 fabricada no estudo de caso. Levando-se em consideração os erros e imprecisões de medição e as limitações de rigidez do sistema de fixação da peça e ferramenta. Pode-se concluir como satisfatório o resultado obtido no desenvolvimento deste estudo de caso. O resultado da inspeção dimensional é mostrado na tabela 6.2. O projeto da peça com suas referidas cotas está no anexo 03.

**Tabela 6.2.** Registro de Medições.

REGISTROS DE INSPEÇÃO					
	Descrição	Peça Exemplo 2			
	Responsável	Gustavo Meneghello	Unidades: mm	Data	16/04/2003
ITEM	DESCRIÇÃO	NOMINAL	TOLERÂNCIA	MEDIDA ATUAL	DESVIO
1	Diâmetro Rebaixo 1	19,000		19,002	0,002
2	Diâmetro Rebaixo 2	19,000		19,036	0,036
3	Diâmetro Furo 1	7,000		7,078	0,078
4	Diâmetro Furo 2	7,000		7,186	0,186
5	Raio da Cavidade Central	34,410		33,493	-0,917
6	Cota 120.000	120,000		119,824	-0,176
7	Cota 57.000	57,000		57,946	0,946
8	Cota 8.920	8,920		9,27	0,4
9	Cota 11.550	11,550		10,847	0,703
10	Cota 6.230	6,230		7,13	0,9
11	Cota 47.000	47,000		47,933	0,933
12	Paralelismo A		0,01	0,068	0,058
13	Paralelismo B		0,01	0,139	0,012
15	Concentricidade C 1		0,01	0,161	
16	Concentricidade C 2		0,01	0,122	
17	Concentricidade Furo C 1		0,01	0,323	
18	Concentricidade Furo C 2		0,01	0,220	

## 7 CONCLUSÕES

### 7.1 RESULTADOS E DISCUSSÕES

Os procedimentos experimentais permitiram a fabricação dos modelos geométricos propostos através da usinagem por sistema robótico para as aplicações desenvolvidas. O pós-processamento do arquivo de programa CN gerou um programa com informações de trajetórias e parâmetros necessários para a fabricação dos modelos por fresamento através de um robô a partir de informações geométricas provenientes dos sistemas CAD e da utilização de recursos e funções dos sistemas CAM atingindo assim os objetivos para os quais o sistema se propôs.

Uma das dificuldades encontradas na execução dos procedimentos experimentais foi o limite de capacidade de armazenamento de dados do sistema robótico. Para contornar esta limitação foi necessário dividir os processos de desbaste e acabamento em diferentes segmentos de programas computacionais. As estratégias de operação como desbaste e acabamento foram segmentadas em programas menores com a capacidade máxima de armazenagem do controlador do robô, o que implica na necessidade de repetir o processo de referenciamento da peça a cada nova execução de programa. Para proporcionar o correto referenciamento para cada etapa segmentada foi desenvolvida uma rotina para enviar a ferramenta ao ponto de referência sempre ao final de cada etapa de programa.

Dentre as dificuldades encontradas, a diferença de sintaxe de funções de movimentação exigiu a criação de sub-rotinas de cálculo para a correta transcrição da trajetória a ser percorrida.

Algumas vantagens da utilização do aplicativo SisPRob podem ser relacionadas como:

- possibilidade de programação automática e integração entre sistemas;
- utilização de até cinco eixos para execução de operações de fresamento;
- grande capacidade no espaço de trabalho;
- flexibilização de operações de sistemas robóticos;
- programação do sistema robótico sem a necessidade de aprendizado da linguagem RAPID;

A análise da peça usinada permite definir como satisfatório o acabamento proporcionado pelo sistema. Embora algumas imperfeições ainda possam ser notadas nas peças, estas se devem muito mais à geometria da ferramenta utilizada do que às limitações do sistema uma vez que os resultados de medições aproximam-se das dimensões desejadas para a fabricação das peças.

No módulo para cinco eixos muitas limitações são impostas ao sistema. O aplicativo desenvolvido converte corretamente as informações geométricas e de trajetórias, porém, a viabilidade de execução do programa dependerá da configuração dos elementos mecânicos no referenciamento do ponto zero peça e das limitações mecânicas do sistema de acionamento da ferramenta condicionado pela fixação ao manipulador, de modo a evitar a possibilidade de colisão da ferramenta com a peça e o sistema robótico.

## 7.2 CONCLUSÕES

Os resultados dos estudos de caso apresentados mostraram a viabilidade de aplicação do pós-processador desenvolvido e sua integração aos demais sistemas computacionais de auxílio à manufatura. O pós-processador apresenta funções para conversão de programas CN em programas de operação de robôs proporcionando a utilização de até cinco eixos de movimentação para a usinagem de superfícies complexas.

Os recursos de integração permitiram a utilização de diferentes características dos sistemas CAD/CAM para a geração de um programa de usinagem em linguagem RAPID a partir de informações geométricas dos sistemas CAD. A utilização destes recursos de integração proporciona flexibilização e automação aos meios produtivos permitindo diminuir custos e otimizar a produção.

As vantagens do desenvolvimento de um pós-processador capaz de interpretar códigos de programação de trajetórias e parâmetros de operação de máquinas-ferramenta e convertê-los em programas de operação de robôs estão na versatilidade, flexibilização e integração entre sistemas proporcionado. O desenvolvimento de um pós-processador dedicado capaz de flexibilizar a utilização de um robô proporciona aplicações não convencionais diversificando a utilização de equipamentos e diminuindo o tempo de programação do sistema robótico.

O algoritmo elimina o processo de aprendizado da linguagem RAPID por parte do operador que pode programar trajetórias e parâmetros de operação através da linguagem de programação de comando numérico enquanto o aplicativo adapta os parâmetros para operação em sistemas robóticos.

A utilização do sistema robótico para operações de usinagem apresenta como vantagens a utilização de seis graus de liberdade para movimentação da ferramenta e a ampliação de dimensões de trabalho com as devidas limitações características de cada equipamento. Suas desvantagens devem-se à baixa rigidez do braço mecânico que limita a definição de materiais e

parâmetros a serem utilizados e a reduzida capacidade de armazenagem de dados do controlador do sistema que limita testes mais complexos, sem expansão de memória no controlador.

## **8 SUGESTÕES PARA TRABALHOS FUTUROS**

- Desenvolver o algoritmo tornando-o capaz de interpretar geometrias e gerar trajetórias a partir do reconhecimento e manipulação de arquivos de transferência de dados como IGES ou STEP, eliminando a etapa de programação CN na programação do robô;
- Ampliação do pós-processamento para a conversão em diferentes linguagens de programação de robôs;
- Análise de parâmetros de usinagem para sistemas robóticos buscando correlações entre processos em diferentes equipamentos;
- Desenvolver o módulo de 5 eixos, para detecção de colisões entre ferramenta, peça e manipulador e fim de curso de movimentação das juntas;

## 9 REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ABB Robótica, **ABB IRB 1400 User's Guide**, 1995.
- [2] ARAI, Tamio, ITOKO, T., YAGO, H. **A graphical Robot Language Developed in Japan**. Robótica, Vol. 5, pp 99-103, 1997.
- [3] AUTODESK. **AutoCAD Reference Manual** - Release 14, 1997.
- [4] BEDWORTH, David D. **Computer Integrated Design and Manufacturing**. USA: McGraw-Hill, 1991.
- [5] BENCHETRIT, E., LENZ, E., SHOHAM, M. **Collision Detection in Static and Dynamic Complex Scenes not Having Geometry Coherence**. Int. J. Advanced Manufacturing, Vol. 14, 70-76, 1998.
- [6] BESANT, C.B. **CAD/CAM Projeto e Fabricação com o Auxílio de Computador**. Rio de Janeiro: Campus, 1985.
- [7] BLACK, J. T. **O projeto da Fábrica com Futuro**. Tradução Gustavo Kannenberg. Porto Alegre. Bookman, 1998.
- [8] BRACARENSE, ALEXANDRE Q. **Robotic Welding Application Based on Off Line Simulation: Brazilian Cases**. Porto, Portugal, n. 42 (2001), p 84-87: il.
- [9] BRAGA, Rodrigo. **Robótica – Órgãos Terminais**. Disponível em: <<http://www.em.pucrs.br/~braga>>. Acesso em: 10 nov. 2001.
- [10] CASSANIGA, Fernando A. **Fácil Programação do Controle Numérico**. Sorocaba : F.A.C. Produções Editoriais, 1991.
- [11] CHIN, Jih-Hua, TSAI, Her-Chyi. **A Path Algorithm for Robotic Machining**. Robotics & Computer-Integrated Manufacturing, Vol. 10, No. 3, pp 185-198, 1993.
- [12] EIA/ANSI 494. **Numerical Control BCL Standards Association**. Disponível em: <<http://www.eia.org/eig/ncbsa>> Acesso em: 17 jun. 2002.
- [13] ENNEX CORPORATION. **The StL Format**. 1999. Disponível em: <<http://www.ennex.com/fabbers/stl.asp#ascII>> Acesso em: 22 out. 2002.
- [14] FILHO, João H. Valgas., CAMPOS, Mário F. M., NOGUEIRA, Régis. **Um Robô Caricaturista**. Anais III Semana de Pós-Graduação da UFMG.
- [15] FREITAS, P. H. F., BATOCCHIO, A. **Planejamento de processos a partir de um sistema que usa informações de CAD**. Revista Máquinas e Metais, 1997, Vol. 376, pp 91-97.

- [16] GROOVER, Mikell P. **Automation Production Systems and Computer Integrated Manufacturing**. New Jersey: Prentice-Hall, Englewood Cliffs, 1987.
- [17] GROOVER, P. Mikell; Weiss M., Nagel, R.N. **Industrial Robotics Technology, Programming and Applications**, Singapore, McGraw-Hill, 1986.
- [18] GROOVER, Mikell P., ZIMMERS Jr., E. W. **CAD/CAM Computer Aided Design and Manufacturing**. New Jersey: Prentice-Hall, Englewood Cliffs, 1984.
- [19] KERNIGHAN, Brian W., RITCHIE, Dennis M. **C A Linguagem de Programação Padrão ANSI**. Rio de Janeiro: Campus, 1989.
- [20] KOREN, Yoram. **Computer Control of Manufacturing Systems**. Singapore: McGraw-Hill, 1986.
- [21] KUCZOZI, Gy. **Collision avoidance for the sculpturing robot**. Technical report, Delft, University of Technology, Delft, 1998.
- [22] LORINI, F. J. **Fundamentos de Robótica Industrial** Notas de aula. Universidade Federal do Rio Grande do Sul, 2002.
- [23] LORINI, F. J. **Sistemas CAM**. Notas de aula. Universidade Federal do Rio Grande do Sul, 1999.
- [24] McMAHON, Chris, BROWNE, Jimmie. **CADCAM Principles, Practice and manufacturing management**. Harlow: Addison-Wesley, 1988.
- [25] OLIVEIRA, L. C. de. **Um sistema de geração de trajetórias de ferramentas em três eixos**. Dissertação de Mestrado, Universidade Federal do Rio Grande do Sul, 1997.
- [26] PAR SYSTEMS. Disponível em: <<http://www.par.com>> Acesso em: 20 jul. 2002.
- [27] PIRES, NORBERTO. **Codex Automaticus & Links**. Disponível em: <<http://robotics.dem.uc.pt/norberto/codex.htm>> Acesso em: 10 dez. 2002.
- [28] QUEIROZ, A. A., STEMMER, C. **Cenário da programação CN - Da programação Manual ao CAD/CAM - Simpósio sobre CAD/CAM - Revisão da situação Brasileira**, Anais, SOBRACON, São Paulo, 1986.
- [29] ROMANO, VITOR FERREIRA. **Robótica Industrial, Aplicação na Indústria de Manufatura e de Processos**. São Paulo: Edgard Blucher, 2002.
- [30] ROMI. **Manual de Programação –Mach-8MP**. Industrias Romi, 1996.
- [31] SANTOS, VÍTOR M. F. **Robótica Industrial**. Apostila. Universidade de Aveiro, Portugal, 2001.
- [32] SHIGLEY, J. E. **Mechanical Engineering Design**. New York: McGraw-Hill 3<sup>rd</sup> ed, 1977.

- [33] SIMÕES, Fábio M. S. **Implementação de um Sistema CAD/CAM para Fresadoras CNC a partir de Funções CAM Implementadas no CAD**. 152f. Dissertação de Mestrado da Universidade Federal de Uberlândia.
- [34] SPATIAL TECHNOLOGY. Disponível em: <<http://www.spatial.com>> Acesso em: 13 nov. 2001.
- [35] TANGELDER, J.S., VERGEEST, J. W., KUCZOZI, Gy, KOVÁCS, Zs., HORVÁTH, I. **Machining Large Complex Shapes Using a 7 DoF Device**. 1998. Disponível em : <<http://dutoa36.io.tudelft.nl/docs/ssm98007.pdf>> Acesso em: 10/10/2003.
- [36] TANGELDER, J.W. H. **Automated fabrication of shape models of free-form objects with a sculpturing robot**. PhD thesis, Delft University of Technology, Delft, 1998.
- [37] TEICHOLZ, E. **CAD/CAM handbook**. New York: McGraw-Hill, 1985. 423p.
- [38] ZEID, I. **CAD/CAM theory and practice**. New York: McGraw-Hill, 1991. 1052p.

**ANEXOS**

ANEXO 01 – Programa CN para execução da peça exemplo 2.

;LABORATÓRIO DE USINAGEM-UFRGS

;BASE\_MINI

G17

G71

G90

G99

O18 T18 M6

M3 S4000

G00 Z26.

G00 X10.257 Y61.585

M8

G00 Z-4.811

G01 Z-5.811 F400

Y62. Z-6.227 F1000

X10.757

Y61.22 Z-5.446

X11.257 Y60.743 Z-4.969

Y62. Z-6.227

X11.757

Y60.243 Z-4.469

X12.257 Y59.743 Z-3.969

Y62. Z-6.227

X12.757

Y59.243 Z-3.469

X13.257 Y58.743 Z-2.969

Y62. Z-6.227

X13.757

Y58.243 Z-2.469

X14.257 Y57.743 Z-1.969

Y62. Z-6.227

X14.757

Y57.243 Z-1.469

X15.257 Y56.743 Z-0.969

Y62. Z-6.227

X15.757

Y56.243 Z-0.469

X16.257 Y56. Z-0.227

Y62. Z-6.227

X16.757

Y56. Z-0.227

X17.257

Y62. Z-6.227

X17.757

Y56. Z-0.227

X18.257

Y62. Z-6.227

X18.757  
Y56. Z-0.227  
X19.257  
Y62. Z-6.227  
X19.757  
Y56. Z-0.227  
X20.257  
Y62. Z-6.227  
X20.757  
Y56. Z-0.227  
X21.257  
Y62. Z-6.227  
X21.757  
Y56. Z-0.227  
X22.257  
Y62. Z-6.227  
X22.757  
Y56. Z-0.227  
X23.257  
Y62. Z-6.227  
X23.757  
Y56. Z-0.227  
X24.257  
Y62. Z-6.227  
X24.757  
Y56. Z-0.227  
X25.257  
Y62. Z-6.227  
X25.757  
Y56. Z-0.227  
X26.257  
Y62. Z-6.227  
X26.757  
Y56. Z-0.227  
X27.257  
Y62. Z-6.227  
X27.757  
Y56. Z-0.227  
X28.257  
Y62. Z-6.227  
X28.757  
Y56. Z-0.227  
X29.257  
Y62. Z-6.227  
X29.757  
Y56. Z-0.227  
X30.257  
Y62. Z-6.227  
X30.757

Y56. Z-0.227  
X31.257  
Y62. Z-6.227  
X31.757  
Y56. Z-0.227  
X32.257  
Y62. Z-6.227  
X32.757  
Y56. Z-0.227  
X33.257  
Y62. Z-6.227  
X33.757  
Y56. Z-0.227  
X34.257 Y56.257 Z-0.484  
Y62. Z-6.227  
X34.757  
Y56.757 Z-0.984  
X35.257 Y57.257 Z-1.484  
Y62. Z-6.227  
X35.757  
Y57.757 Z-1.984  
X36.257 Y58.257 Z-2.484  
Y62. Z-6.227  
X36.757  
Y58.757 Z-2.984  
X37.257 Y59.257 Z-3.484  
Y62. Z-6.227  
X37.757  
Y59.757 Z-3.984  
X38.257 Y60.257 Z-4.484  
Y62. Z-6.227  
X38.757  
Y60.757 Z-4.984  
X39.257 Y61.257 Z-5.484  
Y62. Z-6.227  
G00 Z26.  
X80.257 Y61.743  
Z-4.969  
G01 Z-5.969 F400  
Y62. Z-6.227 F1000  
X80.757 Y62. Z-6.226  
Y61.243 Z-5.469  
X81.257 Y60.743 Z-4.969  
Y62. Z-6.227  
X81.757  
Y60.243 Z-4.469  
X82.257 Y59.743 Z-3.969  
Y62. Z-6.227  
X82.757

Y59.243 Z-3.469  
X83.257 Y58.743 Z-2.969  
Y62. Z-6.227  
X83.757  
Y58.243 Z-2.469  
X84.257 Y57.743 Z-1.969  
Y62. Z-6.227  
X84.757  
Y57.243 Z-1.469  
X85.257 Y56.743 Z-0.969  
Y62. Z-6.227  
X85.757  
Y56.243 Z-0.469  
X86.257 Y56. Z-0.227  
Y62. Z-6.227  
X86.757  
Y56. Z-0.227  
X87.257  
Y62. Z-6.227  
X87.757  
Y56. Z-0.227  
X88.257  
Y62. Z-6.227  
X88.757  
Y56. Z-0.227  
X89.257  
Y62. Z-6.227  
X89.757  
Y56. Z-0.227  
X90.257  
Y62. Z-6.227  
X90.757  
Y56. Z-0.227  
X91.257  
Y62. Z-6.227  
X91.757  
Y56. Z-0.227  
X92.257  
Y62. Z-6.227  
X92.757  
Y56. Z-0.227  
X93.257  
Y62. Z-6.227  
X93.757  
Y56. Z-0.227  
X94.257  
Y62. Z-6.227  
X94.757  
Y56. Z-0.227

X95.257  
Y62. Z-6.227  
X95.757  
Y56. Z-0.227  
X96.257  
Y62. Z-6.227  
X96.757  
Y56. Z-0.227  
X97.257  
Y62. Z-6.227  
X97.757  
Y56. Z-0.227  
X98.257  
Y62. Z-6.227  
X98.757  
Y56. Z-0.227  
X99.257  
Y62. Z-6.227  
X99.757  
Y56. Z-0.227  
X100.257  
Y62. Z-6.227  
X100.757  
Y56. Z-0.227  
X101.257  
Y62. Z-6.227  
X101.757  
Y56. Z-0.227  
X102.257  
Y62. Z-6.227  
X102.757  
Y56. Z-0.227  
X103.257  
Y62. Z-6.227  
X103.757  
Y56. Z-0.227  
X104.257 Y56.257 Z-0.483  
Y62. Z-6.227  
X104.757  
Y56.756 Z-0.982  
X105.257 Y57.255 Z-1.481  
Y62. Z-6.227  
X105.757  
Y57.754 Z-1.98  
X106.257 Y58.252 Z-2.479  
Y62. Z-6.227  
X106.757  
Y58.751 Z-2.978  
X107.257 Y59.25 Z-3.477

Y62. Z-6.227

X107.757

Y59.749 Z-3.976

X108.257 Y60.248 Z-4.474

Y62. Z-6.227

X108.757

Y60.747 Z-4.973

X109.257 Y61.246 Z-5.472

Y62. Z-6.227

X109.757

Y61.6 Z-5.827

G00 Z26.

M9

M5

M30

ANEXO 02 - Programa RAPID para execução da peça exemplo 2.

```

%%%
VERSION:1
LANGUAGE:ENGLISH
%%%
MODULE W0

VAR robtarget P1;
PROC main ()

VAR num opcao;
VAR num opcaofinal;
VAR zonedata zona;
VAR num zincr;
VAR robtarget pontox;
VAR robtarget pontoy;
VAR robtarget pontoz;
VAR speeddata veloc:=[2000,500,5000,1000];
TPWrite "Referencie X!";
Stop \noregain;
pontox:=Crobot();

TPWrite "Referencie Y!";
Stop \noregain;
pontoy:=Crobot();

TPWrite "Referencie Z!";
Stop \noregain;
pontoz:=Crobot();

P1:= pontox;
P1.trans.y:= pontoy.trans.y;
P1.trans.z:= pontoz.trans.z;
zona:=z10;
zona.pzone_tcp:=0;
zona.pzone_ori:=0.5;

MoveL Offs (P1,0, 0, 200.000),veloc,zona,tool0;
TPWrite "Trocar Ferramentas";
TPReadNum opcao,"Digite 1 Qdo Ok";
If opcao <> 1 Break;
MoveL Offs (P1,0, 0, 0.000),v2000,zona,tool0;
TPWrite "Acione a Ferramenta";
TPReadNum opcao,"DIGITE 1 para OK";
if opcao <> 1 Break;
veloc.v_tcp:=2000;
MOVEJ Offs (P1,0, 0, 26), veloc, zona, tool0;
veloc.v_tcp:=2000;

```

```
MOVEJ Offs (P1,61.585, -10.257, 26), veloc, zona, tool0;
TPWrite "Refrigeracao por nevoa";
TPWrite "Acionar refrigeracao";
TPReadNum opcao, "Qdo Ok digite 1";
If opcao <> 1 Break;
veloc.v_tcp:=2000;
MOVEJ Offs (P1,61.585, -10.257, -4.811), veloc, zona, tool0;
veloc.v_tcp:=6.680;
MOVEJ Offs (P1,61.585, -10.257, -5.811), veloc,zona, tool0;
veloc.v_tcp:=16.700;
MOVEJ Offs (P1,62, -10.257, -6.227), veloc,zona, tool0;
MOVEJ Offs (P1,62, -10.757, -6.227), veloc,zona, tool0;
MOVEJ Offs (P1,61.22, -10.757, -5.446), veloc,zona, tool0;
MOVEJ Offs (P1,60.743, -11.257, -4.969), veloc,zona, tool0;
MOVEJ Offs (P1,62, -11.257, -6.227), veloc,zona, tool0;
MOVEJ Offs (P1,62, -11.757, -6.227), veloc,zona, tool0;
MOVEJ Offs (P1,60.243, -11.757, -4.469), veloc,zona, tool0;
MOVEJ Offs (P1,59.743, -12.257, -3.969), veloc,zona, tool0;
MOVEJ Offs (P1,62, -12.257, -6.227), veloc,zona, tool0;
MOVEJ Offs (P1,62, -12.757, -6.227), veloc,zona, tool0;
MOVEJ Offs (P1,59.243, -12.757, -3.469), veloc,zona, tool0;
MOVEJ Offs (P1,58.743, -13.257, -2.969), veloc,zona, tool0;
MOVEJ Offs (P1,62, -13.257, -6.227), veloc,zona, tool0;
MOVEJ Offs (P1,62, -13.757, -6.227), veloc,zona, tool0;
MOVEJ Offs (P1,58.243, -13.757, -2.469), veloc,zona, tool0;
MOVEJ Offs (P1,57.743, -14.257, -1.969), veloc,zona, tool0;
MOVEJ Offs (P1,62, -14.257, -6.227), veloc,zona, tool0;
MOVEJ Offs (P1,62, -14.757, -6.227), veloc,zona, tool0;
MOVEJ Offs (P1,57.243, -14.757, -1.469), veloc,zona, tool0;
MOVEJ Offs (P1,56.743, -15.257, -0.969), veloc,zona, tool0;
MOVEJ Offs (P1,62, -15.257, -6.227), veloc,zona, tool0;
MOVEJ Offs (P1,62, -15.757, -6.227), veloc,zona, tool0;
MOVEJ Offs (P1,56.243, -15.757, -0.469), veloc,zona, tool0;
MOVEJ Offs (P1,56, -16.257, -0.227), veloc,zona, tool0;
MOVEJ Offs (P1,62, -16.257, -6.227), veloc,zona, tool0;
MOVEJ Offs (P1,62, -16.757, -6.227), veloc,zona, tool0;
MOVEJ Offs (P1,56, -16.757, -0.227), veloc,zona, tool0;
MOVEJ Offs (P1,56, -17.257, -0.227), veloc,zona, tool0;
MOVEJ Offs (P1,62, -17.257, -6.227), veloc,zona, tool0;
MOVEJ Offs (P1,62, -17.757, -6.227), veloc,zona, tool0;
MOVEJ Offs (P1,56, -17.757, -0.227), veloc,zona, tool0;
MOVEJ Offs (P1,56, -18.257, -0.227), veloc,zona, tool0;
MOVEJ Offs (P1,62, -18.257, -6.227), veloc,zona, tool0;
MOVEJ Offs (P1,62, -18.757, -6.227), veloc,zona, tool0;
MOVEJ Offs (P1,56, -18.757, -0.227), veloc,zona, tool0;
MOVEJ Offs (P1,56, -19.257, -0.227), veloc,zona, tool0;
MOVEJ Offs (P1,62, -19.257, -6.227), veloc,zona, tool0;
MOVEJ Offs (P1,62, -19.757, -6.227), veloc,zona, tool0;
MOVEJ Offs (P1,56, -19.757, -0.227), veloc,zona, tool0;
```



MOVEL Offs (P1,62, -32.257, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -32.757, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,56, -32.757, -0.227), veloc,zona, tool0;  
MOVEL Offs (P1,56, -33.257, -0.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -33.257, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -33.757, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,56, -33.757, -0.227), veloc,zona, tool0;  
MOVEL Offs (P1,56.257, -34.257, -0.484), veloc,zona, tool0;  
MOVEL Offs (P1,62, -34.257, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -34.757, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,56.757, -34.757, -0.984), veloc,zona, tool0;  
MOVEL Offs (P1,57.257, -35.257, -1.484), veloc,zona, tool0;  
MOVEL Offs (P1,62, -35.257, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -35.757, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,57.757, -35.757, -1.984), veloc,zona, tool0;  
MOVEL Offs (P1,58.257, -36.257, -2.484), veloc,zona, tool0;  
MOVEL Offs (P1,62, -36.257, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -36.757, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,58.757, -36.757, -2.984), veloc,zona, tool0;  
MOVEL Offs (P1,59.257, -37.257, -3.484), veloc,zona, tool0;  
MOVEL Offs (P1,62, -37.257, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -37.757, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,59.757, -37.757, -3.984), veloc,zona, tool0;  
MOVEL Offs (P1,60.257, -38.257, -4.484), veloc,zona, tool0;  
MOVEL Offs (P1,62, -38.257, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -38.757, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,60.757, -38.757, -4.984), veloc,zona, tool0;  
MOVEL Offs (P1,61.257, -39.257, -5.484), veloc,zona, tool0;  
MOVEL Offs (P1,62, -39.257, -6.227), veloc,zona, tool0;  
veloc.v\_tcp:=2000;  
MOVEJ Offs (P1,62, -39.257, 26), veloc, zona, tool0;  
veloc.v\_tcp:=2000;  
MOVEJ Offs (P1,61.743, -80.257, 26), veloc, zona, tool0;  
veloc.v\_tcp:=2000;  
MOVEJ Offs (P1,61.743, -80.257, -4.969), veloc, zona, tool0;  
veloc.v\_tcp:=6.680;  
MOVEL Offs (P1,61.743, -80.257, -5.969), veloc,zona, tool0;  
veloc.v\_tcp:=16.700;  
MOVEL Offs (P1,62, -80.257, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -80.757, -6.226), veloc,zona, tool0;  
MOVEL Offs (P1,61.243, -80.757, -5.469), veloc,zona, tool0;  
MOVEL Offs (P1,60.743, -81.257, -4.969), veloc,zona, tool0;  
MOVEL Offs (P1,62, -81.257, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -81.757, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,60.243, -81.757, -4.469), veloc,zona, tool0;  
MOVEL Offs (P1,59.743, -82.257, -3.969), veloc,zona, tool0;  
MOVEL Offs (P1,62, -82.257, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -82.757, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,59.243, -82.757, -3.469), veloc,zona, tool0;



MOVEL Offs (P1,62, -95.257, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -95.757, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,56, -95.757, -0.227), veloc,zona, tool0;  
MOVEL Offs (P1,56, -96.257, -0.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -96.257, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -96.757, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,56, -96.757, -0.227), veloc,zona, tool0;  
MOVEL Offs (P1,56, -97.257, -0.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -97.257, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -97.757, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,56, -97.757, -0.227), veloc,zona, tool0;  
MOVEL Offs (P1,56, -98.257, -0.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -98.257, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -98.757, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,56, -98.757, -0.227), veloc,zona, tool0;  
MOVEL Offs (P1,56, -99.257, -0.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -99.257, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -99.757, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,56, -99.757, -0.227), veloc,zona, tool0;  
MOVEL Offs (P1,56, -100.257, -0.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -100.257, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -100.757, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,56, -100.757, -0.227), veloc,zona, tool0;  
MOVEL Offs (P1,56, -101.257, -0.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -101.257, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -101.757, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,56, -101.757, -0.227), veloc,zona, tool0;  
MOVEL Offs (P1,56, -102.257, -0.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -102.257, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -102.757, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,56, -102.757, -0.227), veloc,zona, tool0;  
MOVEL Offs (P1,56, -103.257, -0.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -103.257, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -103.757, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,56, -103.757, -0.227), veloc,zona, tool0;  
MOVEL Offs (P1,56.257, -104.257, -0.483), veloc,zona, tool0;  
MOVEL Offs (P1,62, -104.257, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -104.757, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,56.756, -104.757, -0.982), veloc,zona, tool0;  
MOVEL Offs (P1,57.255, -105.257, -1.481), veloc,zona, tool0;  
MOVEL Offs (P1,62, -105.257, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -105.757, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,57.754, -105.757, -1.98), velbc,zona, tool0;  
MOVEL Offs (P1,58.252, -106.257, -2.479), veloc,zona, tool0;  
MOVEL Offs (P1,62, -106.257, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,62, -106.757, -6.227), veloc,zona, tool0;  
MOVEL Offs (P1,58.751, -106.757, -2.978), veloc,zona, tool0;  
MOVEL Offs (P1,59.25, -107.257, -3.477), veloc,zona, tool0;  
MOVEL Offs (P1,62, -107.257, -6.227), veloc,zona, tool0;

```

MOVEL Offs (P1,62, -107.757, -6.227), veloc,zona, tool0;
MOVEL Offs (P1,59.749, -107.757, -3.976), veloc,zona, tool0;
MOVEL Offs (P1,60.248, -108.257, -4.474), veloc,zona, tool0;
MOVEL Offs (P1,62, -108.257, -6.227), veloc,zona, tool0;
MOVEL Offs (P1,62, -108.757, -6.227), veloc,zona, tool0;
MOVEL Offs (P1,60.747, -108.757, -4.973), veloc,zona, tool0;
MOVEL Offs (P1,61.246, -109.257, -5.472), velbc,zona, tool0;
MOVEL Offs (P1,62, -109.257, -6.227), veloc,zona, tool0;
MOVEL Offs (P1,62, -109.757, -6.227), veloc,zona, tool0;
MOVEL Offs (P1,61.6, -109.757, -5.827), veloc,zona, tool0;
veloc.v_tcp:=2000;
MOVEJ Offs (P1,61.6, -109.757, 26), veloc, zona, tool0;
TPWrite "Desligar Refrigeracao";
TPReadNum Opcao, "Qdo Ok Digite 1";
If opcao <> 1 Break;
TPwrite "Atencao: Pare o acionamento da ferramenta";
TPReadNum opcao,"DIGITE 1 para OK";
if opcao <> 1 Break;
TPWrite "Fim de Programa";
TPWrite "Para recomencar aperte start na barra de menu";
TPReadNum opcaofinal,"Continuar Programa - 1 Programa novo - 2";
if opcaofinal=1 then
MOVEL Offs (P1,0,0,50),v200, zona, tool0;
MOVEL P1,v200, zona, tool0;
endif
stop;
TPWrite "Fim de Programa";
TPWrite "Para recomencar aperte start na barra de menu";
TPReadNum opcaofinal,"Continuar Programa - 1 Programa novo - 2";
if opcaofinal=1 then
MOVEL Offs (P1,0,0,50),v200, zona, tool0;
MOVEL P1,v200, zona, tool0;
endif
stop;

ENDPROC
ENDMODULE

```

ANEXO 03 - Projeto da peça exemplo 2.