

**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA**

**Concepção, Projeto e Implementação de
Células Automatizadas utilizando conceitos de
Programação Off-Line de Robôs**

Autor: Julio César de Almeida Freitas
Orientador: Professor Dr. João Maurício Rosário

07/04

**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA**

Concepção, Projeto e Implementação de Células Automatizadas utilizando conceitos de Programação Off-Line de Robôs

Autor: Julio César de Almeida Freitas
Orientador: Professor Dr. João Maurício Rosário

Curso: Engenharia Mecânica
Área de Concentração: Mecânica dos Sólidos e Projeto Mecânico

Dissertação de mestrado acadêmico apresentado à comissão de Pós-Graduação da Faculdade de Engenharia Mecânica, como requisito para a obtenção do título de Mestre em Engenharia Mecânica.

Campinas, 2004
S.P. - Brasil

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

F884c

Freitas, Julio César de Almeida

Concepção, projeto e implementação de células automatizadas utilizando conceitos de programação off-line de robôs / Júlio César de Almeida Freitas. --Campinas, SP: [s.n.], 2004.

Orientador: João Maurício Rosário.

Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica.

1. Robótica. 2. Automação – Aplicações industriais. 3. Simulação (Computadores). I. Rosário, João Maurício. II. Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica. III. Título.

**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA**

DISSERTAÇÃO DE MESTRADO ACADÊMICO

Concepção, Projeto e Implementação de Células Automatizadas utilizando conceitos de Programação Off-Line de Robôs

Autor: Julio César de Almeida Freitas

Orientador: Professor Dr. João Maurício Rosário

Prof. Dr. João Mauricio Rosário
Departamento de Projeto Mecânico - UNICAMP

Prof. Dr. José Manoel Balthazar
Departamento de Projeto Mecânico - UNICAMP

Prof. Dr. Alexandre Queiroz Bracarense
Universidade Federal de Minas Gerais - UFMG

Campinas, 16 de Julho de 2004

Dedicatória

Dedico este trabalho aos meus melhores “projetos”... Lucas, Marina e Filipe.

Pelos momentos implacáveis, o qual impus sacrifícios e que nunca serão compensados. O conforto vem do sorriso inocente, do choro manhoso, da prece em família. Gostaria que a vida desse a mesma oportunidade para meus filhos, de terem a mesma sorte que eu tive com seus avós.

Agradecimentos

Este trabalho não poderia ser terminado sem a ajuda de diversas pessoas às quais presto minha homenagem:

In memória, aos meus pais, Geraldo e Marina por deixarem exemplo de valores que norteiam a conduta da minha vida e dos meus filhos.

A minha profunda e sincera admiração ao meu orientador Professor Dr. João Mauricio Rosário, que desempenhou seu profissionalismo, empenho, apoio, incentivando para o desenvolvimento e concretização deste trabalho.

A todos os professores, funcionários, colegas de trabalho, de estudo, que de forma direta e indireta ajudaram na conclusão deste trabalho, não os citarei nominalmente para não cometer a indelicadeza de esquecer alguém e estou falando de vocês que estiveram de meu lado compartilhando os momentos e ajudando-me em vários caminhos, até a conclusão deste trabalho.

“Assim como falham as palavras quando querem exprimir qualquer pensamento,
Assim falham os pensamentos quando querem exprimir qualquer realidade.”

Resumo

FREITAS, Julio César de Almeida, *Concepção, Projeto e Implementação de Células*

Automatizadas utilizando conceitos de Programação Off-Line de Robôs, Campinas: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2004. 151 p. Dissertação (Mestrado).

A utilização de robôs em ambientes industriais exige a integração dos mesmos dentro do processo automatizado como um todo, através de um planejamento que considere aspectos relacionados ao custo / benefício, arquitetura de integração do equipamento e, planejamento de células automatizadas.

Este trabalho tem como principal objetivo a utilização de ferramentas de integração para integração de dispositivos automatizados, com ênfase na utilização de conceitos de programação Off-Line, possibilitando uma melhor utilização dos recursos disponíveis, formação profissionais mais direcionados a integração, garantindo ganhos eminentes da tecnologia existente e sua utilização racional.

Palavras Chave

Automação, Robótica, Simulação, Integração e Programação Off-Line

Abstract

FREITAS, Julio César de Almeida, Modeling, Design, Project and Implementation of Automation Cell using concepts of Off-Line Programming of Robots, Campinas: Faculty of Mechanical Engineering, University of Campinas, 2004. 151 p. Dissertation (Master).

The use of robots in industrial environment, demands its integration in the automation process through a planning that considers aspects of low cost and benefit, architecture of integration of peripheral and planning of automated cells.

This work have as main objective the use of tools for integration, of automated devices with emphasis in the use of the concept of Off-Line programming, making possible the better use of the resources available professional, formation addressing to the integration, guaranting the eminent earn of the existent technology and its rational use.

Key words:

Automation, Robotics, Simulation, Integration and Off-Line programming

Índice

Lista de Figuras	v
Lista de Tabelas	ix
Nomenclatura	x
Capítulo 1: Introdução	1
1.1 A utilização de robôs dentro do processo de Automação	2
1.2 Principais objetivos do trabalho	4
Capítulo 2: Posicionamento do trabalho em estudo – Revisão Bibliográfica	7
2.1 O surgimento da automação – Conceitos básicos	7
2.2 Níveis de automação	11
2.3 Fatores básicos para programação das tarefas	13
2.3.1 Comunicação utilizando entradas e saídas digitais (I/O)	13
2.3.2 Comunicação Field Bus	14
2.3.3 Protocolos de comunicação entre sistemas de controle e robôs	15
2.3.4 Gerenciamento simples e multi células de trabalho	16
2.3.5 Programação de tarefas de robôs	18
2.3.6 Programação de robôs industriais	23

2.4 Programação on-line de robôs industriais	23
2.4.1 Método de aprendizagem (Teaching)	23
2.4.2 Método Walk through	25
2.5 Programação off-line de robôs industriais	25
2.6 Conclusão	32
Capítulo 3: Programação Off-line de Robôs	33
3.1 A utilização de programação off-line de robôs	33
3.2 Convenção Denavit-Hartenberg “D-H”	35
3.3 Conceitos básicos	39
3.4 Modelagem cinemática do robô	42
3.4.1 Modelagem cinemática direta	43
3.4.2 Modelagem cinemática inversa	44
3.4.3 Modelagem dinâmica	45
3.5 A programação de robôs via teach – pendant	46
3.5.1 Movimentação angular das juntas	47
3.5.2 Movimento na direção cartesiana	48
3.5.3 Movimento de reorientação da ferramenta	49
3.6 Calibração da ferramenta	50
3.7 Linguagem de programação de robôs	51
3.8 Procedimentos básicos para implementação e execução de programas	52
3.9 Vantagens da programação off-line	53
3.10 Programação através de visualização gráfica	54
3.11 Exemplo de aplicação	55

3.12 Conclusão	60
Capítulo 4: Programação Avançada Direcionada a Aplicações Industriais	63
4.1 Descrição de Software de Programação Off-Line de Robôs	64
4.2 Integração de Sistemas Automatizados	65
4.3 Exemplo de implementação Utilizando Entradas-Saídas Digitais	67
4.4 Integração de Robôs Industriais	68
4.4.1 Célula Integrada de Manufatura implementada no SENAI	69
4.4.2 Célula Integrada de Soldagem implementada na UNICAMP	72
4.5 Exemplo de implementação utilizando Cooperação Robótica	75
4.5.1 Modelagem Off-Line da Célula Automatizada	84
4.5.2 Simulação e Programação Off-Line	85
4.5.3 Modelagem Cinemática Direta	88
4.6 Identificação de Parâmetros e posicionamento em relação ao referencial Zero do Robô	90
4.7 Conclusões	91
Capítulo 5: Conclusões Finais e Perspectivas Futuras	92
5.1 O ensino da programação off-line	94
5.1.1 Visualização gráfica	95
5.1.2 O profissional de nível médio	96

5.1.3 O profissional de nível técnico superior	96
5.1.4 A integração de sistemas automatizados	98
5.2 Exigências do profissional na área de robótica	99
Referências Bibliográficas	102
Anexo I: Principais Comandos e Exemplos de Programação de Robôs Utilizando Linguagem Textual	108
AI.1 Programação textual baseada na estruturação	113
Anexo II: Exemplo de Aplicação de Programação “Off-Line”	117
Anexo III: Proposta de Formação Profissional em Robótica Industrial utilizando a Programação Off-Line.	125
AIII.1 A formação profissional	125
AIII.2 A formação no Brasil	126
AIII.3 Alguns conceitos básicos de Mecatrônica	127
AIII.4 Níveis de automação	130

Lista de Figuras

2.1 Sistema de controle de realimentação	9
2.2 Sistema de controle sem realimentação	10
2.3 Exemplificação das categorias dos blocos de automação	11
2.4 Representação de arquitetura sistema “Aberto”, adaptado por Freitas 2004, de Fiedler, P. Schilb, C.J. 2003	15
2.4 Exemplos de redes de gerenciamento	17
2.5 Representação gráfica da situação	19
2.6 Representação gráfica em projeção da planta da célula	20
2.7 Grafcet Funcional	21
2.8 Exemplo de uma célula de trabalho	27
2.9 Programação off-line – Identificação de colisões	30
3.1 Representação da cinemática espacial de um robô 6R	35
3.2 Parâmetros a, a, d, T	36
3.3 Esquema de procedimento da referência dos “Link”s	38
3.4 Movimento angular das juntas	47
3.5 Movimento em Coordenadas Retangulares	48

3.6 Re-orientação da ferramenta	49
3.7 Procedimento de calibração da ferramenta terminal	51
3.8 Exemplo de uma célula de trabalho com solda robotizada	55
3.9 Ilustrando um programa escrito em linguagem Karel 2 pertencente a alguns controladores FANUC	56
3.10 Arquivo dos seis pontos ensinados (TP), que possui as coordenadas de posição e orientação	57
3.11 Célula de Manufatura Automatizada Robotizada	59
3.12 Detalhes da Célula de Manufatura Automatizada Robotizada	59
4.1 Dispositivo periférico, a considerar com quatro juntas	66
4.2 Dispositivo para troca automática da ferramenta (ATC)	66
4.3 Exemplo de Célula, Integração Real -Virtual. Integração de Robôs e dispositivos periféricos através de I/Os	67
4.4 Integração de robôs e dispositivos periféricos através de I/Os. Tratamento das informações	68
4.5 Célula Automatizada Robotizada, implementada na Escola SENAI e Faculdade SENAI de Tecnologia Mecatrônica – SP. “Armando de Arruda Pereira”	70
4.6 Célula Automatizada Robotizada, Virtual (Workspace®)	71
4.7 Célula Automatizada Robotizada Virtual, a ser implementada na FEM - UNICAMP	73
4.8 Célula Automatizada Cooperativa Virtual a ser implementada na UNICAMP	74
4.9 Detalhe do Dispositivo Mecatrônico a ser implementada na FEM-UNICAMP	74
4.10 Soldagem em Cooperação	76
4.11 Lógica Real da execução do Programa	77

4.12 Descrição lógica através do Grafcet Funcional	79
4.13 Modelo e situação proposta entre Robôs e Periféricos	81
4.14 1º Controle distinto e integrado	82
4.14 a Painel de Controle Virtual	82
4.15 2º Controle distinto e integrado	83
4.16 Espaço de trabalho do robô dentro da célula	84
4.17 Painel de conexão virtual entre robôs e dispositivos automatizados da célula	87
4.18 Modelo cinemático do Dispositivo – Parâmetros de Denavit Hartenberg	88
4.19 Sistemática proposta para calibração	90
5.1 Proposta de formação de profissional em Mecatrônica	99
5.2 Enfoque de profissional de engenharia do século XXI	101
AI.1 Instruções da linguagem de programação RAPID™ do robô ABB™	109
AI.2 Exemplos de programação	110
AI.3 Exemplo de implementação do programa (desenho de um círculo)	111
AI.4 Exemplo de Programação – Robô BB™	112
AI.5 Exemplo de utilização da instrução TPREADNUM	114
AI.6 Exemplo de programação	115
AI.7 Exemplo de programação - Sub-rotinas	116
AII.1 Bloco de deslocamento e a 1ª soldagem	118
AII.2 Bloco de deslocamento e a 2ª soldagem	119
AII.3 Bloco de deslocamento e a 3ª soldagem	119
AII.4 Bloco de deslocamento e a 4ª soldagem	120
AII.5 Utilização da Instrução For... EndFor	121
AII.6 Utilização da Instrução Repeat... Until	122

AII.7 Utilização da Instrução While... EndWhile	123
AIII.1 Definição de Mecatrônica	128
AIII.1a Definição de Mecatrônica nas Áreas de Automação	129
AIII.2 Complexidade de um Sistema Mecatrônico	131

Lista de Tabelas

3.1 Exemplos de softwares comerciais disponíveis no mercado para simulação e programação off-line de robôs	41
3.2 Exemplos de softwares de simulação e programação off-line desenvolvidos por fabricantes de robôs industriais	41
4.1 Estrutura de troca de informações I/O	86

Nomenclatura

Abreviações

ATC – Troca automática de ferramenta

CAD – Desenho Auxiliado por Computador

CAD/CAM – Desenho Auxiliado por Computador / Computador Auxiliando Manufatura

CIM – Computador Integrando a Manufatura

CNC – Controle Numérico Computadorizado

(CP) – Caminho controlado

CP – Caminho contínuo

Devicenet – protocolo de comunicação

FAS – Sistema de Automação de Fábrica

Field bus – protocolo de comunicação

FMS – Sistema Flexível de Manufatura

GMAW – Soldagem a Arco Metálico com Atmosfera Gasosa

GRAFCET – Gráfico de Etapa e Transição - Association Française de Cybernétique
Économique et Technique

Interbus - protocolo de comunicação

IHM – Interface Homem Máquina

I/O – protocolo de comunicação entrada / saída

JIG'S – Dispositivos de Fixação e Posicionamento

KAREL – Linguagem de programação de alguns controladores FANUC e Software de programação off-line Workspace® versão 4.01 .

LAN – Rede de acesso local

MPM – Gerenciamento no processo de Manufatura.

Network – rede de trabalho

PLC – Controlador Lógico Programável

PLM – Gerenciamento no ciclo de vida do produto

PME – Pequena e média empresa

PMI – Pequena e média Indústria

Profibus – protocolo de comunicação

PTP – Ponto a Ponto

RAPID – linguagem de programação dos controladores ABB.

RdP – Redes de Petri

SA – Sistema Automatizado.

SFC – Fluxograma seqüencial – International Electrotechnical Committee

TCP/IP – protocolo de comunicação.

Capítulo 1

Introdução

A utilização de robôs industriais são cada vez mais frequentes em ambientes conhecidos, acessíveis e inacessíveis, como também em ambientes desconhecidos, exige-se a integração dos mesmos dentro de um processo que envolve não só o robô como também outros dispositivos associados à aplicação.

Em ambientes industriais, como não poderia deixar de ser, a exigência da integração dos mesmos impõe uma condição semelhante às citadas. A robotização vem da necessidade de se obter um processo, por exemplo, mais qualificado o qual garante repetições e precisões que fatalmente levaria à fadiga em ambientes insalubres ou não. Alguns atenuantes devem ser buscados, fazendo necessários para propiciar uma inserção consciente e correta na utilização de robôs em ambientes industriais.

1.1 A Utilização de robôs dentro do processo de automação

O processo de automação inicia-se através do planejamento de um novo processo ou de um já existente, a partir de um estudo de viabilidade, onde deverão ser considerados as limitações e ganhos na utilização de robôs industriais ou dispositivos automatizados dedicados. Dentre as principais vantagens e inconvenientes que envolvem a utilização de robôs como ferramenta de integração em automação podemos citar algumas, tais como:

i) Custo / benefício

Representa um fator preponderante, onde a amortização do robô é fator decisório na sua aquisição. No mercado brasileiro estima-se que o “pay back”, esteja em torno de 21 meses¹, em contra posto ao mercado norte americano que está entre 24 e 12 meses com uma forte tendência de queda (Nagler, 2002). Esse peso é negativo, para o mercado nacional e só haverá uma mudança do panorama de mercado a partir do instante em que o número de robôs instalados venha a aumentar e naturalmente haja queda do preço, conseqüentemente forçando a amortização de forma mais amena.

ii) Arquitetura de integração do equipamento

Toda unidade de controle possui um sistema “proprietário”, onde somente o fabricante possui liberdade de abri-lo e alterá-lo, a princípio esta limitação pode levar a uma dificuldade no processo de integração robô / periféricos.

Nos últimos tempos esse conceito vem ganhando um apelo mais forte no sentido de tornar a arquitetura padronizada, conceito de “aberta”, usufrui-se de sistemas padronizados disponíveis facilmente no mercado, como, por exemplo, o sistema operacional Windows®, utilizados em algumas interfaces homem-máquina e protocolos de comunicação como TCT/IP,

¹ período médio estimado no ano de 2003 , de forma informal pelos cálculos preliminares do Gerente de produto do departamento de robótica da ABB Brasil, em função da configuração de uma célula de soldagem com dois robôs, a mais comercializada pela ABB no Brasil.

devicenet, profibus, interbus entre outros, somando aí inúmeros ganhos técnicos e econômicos dos integradores e usuários.

Obviamente tomando-se as devidas proporções em relação à tecnologia desenvolvida, como os algoritmos de controle dos movimentos do robô, interface entre o braço do robô e seus acionamentos e etc. Este propósito vem a propiciar uma maior flexibilidade que possibilita maior facilidade de integração, principalmente, a questão da interface com outros equipamentos que possuem sistemas “abertos” através dos protocolos de comunicação que começam a dividir uma mesma tendência, o qual fortalece um sistema padronizado que naturalmente diminui o custo de atualização e novos implementos.

iii) Planejamento de células robotizadas

É decisivo dentro do processo de implementação ou alteração de células de manufatura automatizada, devendo ser analisada de forma detalhada para compreensão e planejamento de um novo processo ou alteração de um já existente, onde através do planejamento e concepção de células robotizadas é possível à simulação gráfica do robô e do ambiente, que tem como objetivo especificar equipamentos, envelope de trabalho, e gerar trajetórias para os elementos robóticos.

A partir do dimensionamento e projeto mecânico, se possibilita o processo de identificação de elementos, por meio do método gráfico para modelagem do ambiente. A descrição de situações de layout e integração com outros equipamentos periféricos permitem simular e programar de formas gráficas com a visualização, prevendo situações inusitadas a qual garante uma estratégia segura de implementação de um processo automatizado e robotizado.

iv) Custos envolvidos

Os inconvenientes mais significativos na utilização de robôs industriais são os altos custos envolvidos na aquisição, implantação, manutenção e principalmente em relação à mão-de-obra

qualificada no uso dessa tendência tecnológica cada vez mais real, mas que são sensivelmente amortizados, através do planejamento e popularização como contribuição para o processo que cada vez mais se faz necessário ao ambiente fabril de nosso cenário nacional.

1.2 Principais objetivos do trabalho

Existem trabalhos que enfocam a programação “off-line”, como por exemplo, (Sá, 1996), (Sá, 2000), que desenvolveu ferramentas para programação off-line baseadas na modelagem cinemática de robôs, com ênfase em redundâncias e tratamentos de colisões, implementando métodos numéricos para a resolução do problema cinemático inverso de manipuladores robóticos realçando o controle de posição, bem como sua supervisão.

Este trabalho tem como principal objetivo e justificativa a concepção, planejamento e operação de robôs industriais utilizando conceitos de simulação e programação off-line, preocupando-se em apresentar de forma abrangente e compreensível as ferramentas utilizadas na concepção, planejamento e integração de células automatizadas utilizando robôs industriais. Alguns dos pontos a serem explorados são as possibilidades de se utilizar melhor os recursos disponíveis, tais como simulação, programação e planejamento, a objetivar a formação e treinamento de profissionais mais direcionados a integração, garantindo uso racional, com a possibilidade dos ganhos eminentes da tecnologia e redução de custos envolvidos.

Alguns aspectos importantes serão abrangidos neste trabalho, tais como:

i) Estruturação de programas de integração: que requerem a utilização de conceitos de automação, tais como as ferramentas de descrições que permitem de forma funcional e tecnológica seqüenciar uma lógica de execução de uma tarefa (Sequential Flow Chart - SFC, Redes de Pétri - RdP), permitindo a implementação de programas off-line estruturados e posterior carregamento nos softwares de simulação a criar possibilidade da transferência para dispositivos robóticos (carregamento);

ii) Concepção de sistemática de calibração do sistema robótico, envolvendo robô, ferramentas terminais e processos. Um tratamento para identificação da posição de referência Zero deverá ser implementado a partir da utilização de sensores externos de medida, auxiliados por procedimentos matemáticos, envolvendo a identificação do posicionamento da base do robô em relação a um sistema de referência do sistema externo (através do cálculo do vetor posição inicial e matriz de orientação), a ser inserido automaticamente no programa off-line para carregamento posterior no dispositivo robótico.

Assim, os objetivos delineados anteriormente serão descritos nesse trabalho através dos seguintes capítulos:

No capítulo 2 é apresentada uma Revisão Bibliográfica detalhada direcionada a Integração de Robôs em Ambientes Industriais, onde é abordado um posicionamento do trabalho em estudo, com dedicação aos conceitos de sistemas automatizados e seus atributos, passando pela concepção de sistemas automatizados, modelagem de sistemas a eventos discretos e conceitos de estruturação onde a ênfase será mais focada em função de sua própria característica e possibilidade de uso mais flexível do robô. São apresentados também aspectos relacionados à programação de robôs industriais e redes de comunicação em ambientes industriais, envolvendo a utilização e programação de robôs, de dispositivos automatizados constituintes de células de manufatura a distância.

No capítulo 3 são abordados aspectos concernentes a Programação Off-line de Robôs, apresentação de métodos de programação, conceitos básicos de modelagem cinemática direta e inversa, problema de orientação da ferramenta, coordenadas de movimento do robô industrial, calibração de ferramenta, procedimentos de início de trabalho com o robô industrial e linguagens de programação textual. Enfatiza a estruturação, e simulação; mostrando novos paradigmas na utilização, análise e concepção de sistemas automatizados dentro da engenharia, baseada em modelos virtuais, propiciando análises e decisões mais próximas do real, com dados que são compilados de forma intensa e exaustiva, a uma relação de tempo baixa, agregada ao processo.

Neste capítulo é também apresentado um ambiente completo de programação off-line de um robô industrial que será utilizado nos próximos capítulos.

No capítulo 4 são apresentados através de exemplos práticos e implementados, a validação e testes dos conceitos apresentados anteriormente, Serão enfatizados dois exemplos de aplicações: uma Célula Automatizada já implementada no SENAI – São Caetano do Sul, SP, e um modelo de uma Célula Virtual a ser implementada no Laboratório de Automação Integrada e Robótica da Faculdade de Engenharia Mecânica da UNICAMP.

Através desses dois exemplos, foi enfatizado aspecto concernente aos processos de programação avançados, onde existe a possibilidade real de ganho no processo, enfocando uma utilização racional nos processos de programação e que venha a agregar de forma significativa a concepção de programação.

Neste capítulo ainda é apresentado um estudo relativo à calibração de robôs, a partir de obtenção de parâmetros cinemáticos durante a simulação, onde a partir desse estudo de casos será possível a realização de uma análise da aplicabilidade enfocando a concepção e planejamento no uso da simulação e programação off-line, dificuldades e facilidades, ganhos reais diretos e indiretos.

No capítulo 5 todo o trabalho desenvolvido sintetizado através de uma proposta de formação profissional para o Projeto de Células Robotizadas industriais, que utiliza conceitos de programação off-line, permite assim a viabilização e exequibilidade de implementação dessas ferramentas e procedimentos em empresas de pequeno e médio porte, constitui assim uma nova área de atuação profissional no ramo da engenharia do século XXI.

Finalmente no **capítulo 6** são apresentadas conclusões finais, tendências e perspectivas que poderão ser abordadas em trabalhos futuros.

Capítulo 2

Posicionamento do Trabalho em Estudo - Revisão Bibliográfica

Neste capítulo é apresentado uma revisão bibliográfica aprofundada direcionada a Integração de Robôs em Ambientes Industriais, sendo abordado inicialmente um posicionamento do trabalho em estudo, uma revisão dos conceitos de sistemas automatizados e seus atributos, passando pela concepção de sistemas automatizados, modelagem de sistemas a eventos discretos e conceitos de estruturação, enfatizando as diferentes possibilidades de uso mais flexível de um robô dentro de uma célula automatizada. São apresentados também aspectos relacionados à programação de robôs industriais e redes de comunicação em ambientes industriais, envolvendo a utilização e programação de robôs de dispositivos automatizados constituintes de células de manufatura a distância.

2.1 O surgimento da automação – Conceitos básicos

Os primeiros conceitos de automação surgiram no período da revolução industrial através da Mecanização, que consistia basicamente na melhoria das primeiras máquinas, substituindo-se a potência humana ou animal por outras formas de acionamento como, por exemplo, energia mecânica. Não deixando de lado a necessidade humana para controlar ou guiar as máquinas na

execução de uma função. Automação, por outro lado, tem a capacidade para a própria regulação, fazendo com que a máquina automatizada seja capaz de se controlar.

As definições de automação no decorrer de sua história deixam clara a identificação de quatro atributos básicos encontrados em Sistemas Automatizados, que designaremos S.A.. Estes atributos são tratados em (Groove, 1988); (Asfahl, 1992) e denominados blocos de automação, sendo eles:

i) Blocos de potência para executar alguma ação ou processo

Estes blocos correspondem às fontes de energia que geram as ações ou processos e podem coexistir duas possibilidades de atividades, a primeira associada aos processos, que por sua vez está associado a uma operação de manufatura e a segunda transferência e ou posicionamento, associado ao movimento da peça ou produto antes e após cada processo.

ii) Blocos de comandos programados para definir a ação ou execução do processo.

As atividades executadas estão contidas dentro de um programa. As ações definidas pelo programa são freqüentemente executadas dentro de uma maneira cíclica, dentro de muitas operações de produção. Estes ciclos de ações constituem o que é chamado ciclo de trabalho.

Para todo ciclo de trabalho, uma unidade de produção é completada no processo. Dentro de programas mais elaborados, o número de ações separadas executadas pelo S.A. podem ser grandes e as próprias ações podem ser complexas.

Algumas vezes o programa pode requerer sinais de comunicação entre diferentes partes dos equipamentos no S.A.. Estes sinais são referenciados para evitar que um dispositivo ou máquina inicie outra operação, antes do término da que está sendo processada, e são usados para transmitir coordenação de ações de vários componentes do sistema.

iii) Bloco de controle realimentado

É usado para determinar que os comandos programados foram corretamente executados. Como ilustrado, na figura 2.1, um sistema de controle de realimentação consiste de cinco elementos básicos (A) entrada, (B) saída, (C) início de processo controlado, (D) elementos sensoriados e (E) elementos atuantes e controlados.

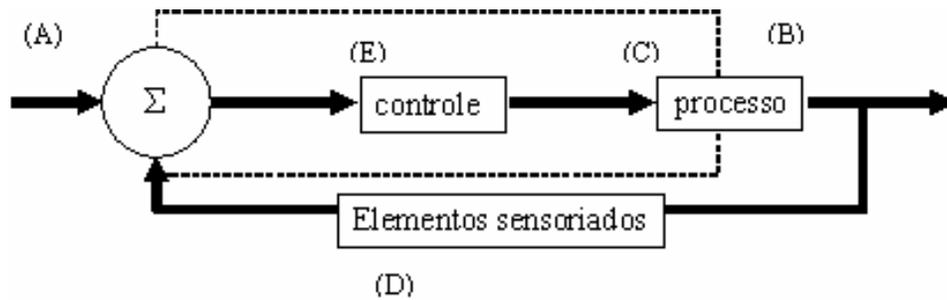


Figura 2.1 Sistema de controle de realimentação.

Sistemas de controle realimentados são frequentemente referenciados a um sistema de controle de malha fechada. Nem todos os sistemas de controle são de malha fechada. Um segundo tipo de controle que é usado em S.A. é chamado de malha aberta. Neste tipo de controle, na figura 2.2, não há realimentação da malha para verificar que o comando de entrada foi executado.

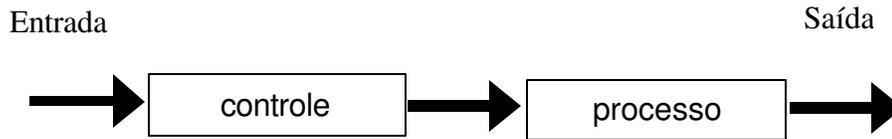


Figura 2.2 Sistema de controle sem realimentação.

iv) Bloco de tomada de decisões

São programados com a habilidade de tomar decisões durante operações. A capacidade de tomar decisões está na forma de uma lógica de sub-rotina no controle do programa. Através da união dessas sub-rotinas no controle do programa para sensores que monitoram o processo e seu ambiente, o sistema pode responder para as mudanças e irregularidades dentro de uma maneira lógica. Como efeito, o sistema de controle toma decisões. Há um numero de argumentos para dar a um S. A. a capacidade de tomar decisões. Estes argumentos incluem:

- a) otimização de processo;
- b) interação com humanos;
- c) detecção de erros e recuperação.

Uma outra abordagem em relação aos blocos de automação (Asfahl, 1992) também dividida em quatro classes básicas, é mostrada na figura 2.3, as classes são:

- a) *sensores*;
- b) *analísadores*;
- c) *atuadores*;
- d) *motores*:

Na parte superior um sistema automatizado e na parte inferior um sistema convencional. Os relacionamentos dessas quatro categorias são visíveis e o operador humano pode ser naturalmente substituído por um robô industrial.

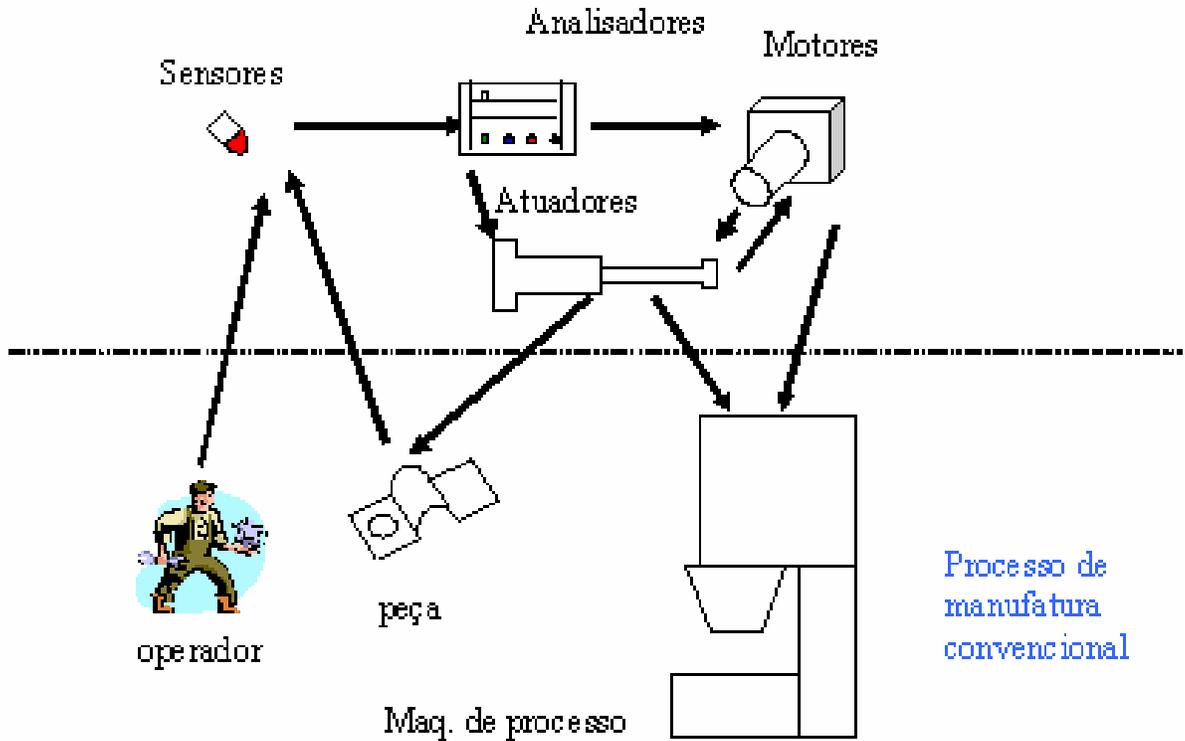


Figura 2.3 Exemplificação das categorias dos blocos de automação.

2.2 Níveis de automação

Os graus de automatismo do ciclo de produção industrial, principalmente na operação básica de fabricação, podem ser divididos em duas áreas distintas: Automação de Processos Contínuos e Automação da Produção em partes discretas. No primeiro caso, é encontrado principalmente em aplicações de indústrias que controlam processos, como, por exemplo, nas indústrias siderúrgicas e o segundo caso pode ser encontrados em indústrias que dividem os processos de fabricação em unidades, lotes ou série, como nas indústrias automotivas.

A Automação da Produção seriada comumente denominada Automação da Manufatura, pode ser dividida em diferentes concepções e principalmente em relação a quanto é automatizado, dependendo da quantidade de investimento realizado.

Para que seja direcionado ao entendimento da automação aplicada na produção, automação da manufatura, descrevendo as aplicações da mecânica, eletrônica e sistema baseados em computador para operar, controlar e gerenciar sistemas de manufatura se classificam no processo três tipos básicos associados ao tipo de produção (Groove, 1988), (Ferreira, 1991):

i) Automação Rígida

A automação rígida é limitada a grandes volumes de produção e é feita onde não há variações na produção. A seqüência do processo é fixada pela configuração do equipamento, são descomplicadas, porém torna-se complicada ao integrar operações múltiplas. Como principal característica pode-se dizer, adequado para produção contínua com alto volume de peças idênticas ou parecidas, equipamentos especializados com alta eficiência e inflexível para mudanças.

ii) Automação Programável

A automação programável tem a facilidade de ser programada e reprogramada conforme a necessidade, utilizados em processos com variações de produto com volumes baixos e médios. Indicada para produtos produzidos em grupos ou lotes, equipamento projetado para diferentes configurações, velocidade de produção menor do que a automação rígida.

iii) Automação Flexível

A automação flexível torna-se uma extensão da automação programável, porém há uma diferença com relação a produzir um “mix” de produtos sem a perda de tempo na mudança e ajustes físicos, executando com eficiência uma produção com combinações e planejamentos variáveis conforme a demanda sem a necessidade de operar em grupos e ou lotes. Duas características tecnológicas fazem possível a automação flexível:

1) A capacidade para reprogramar o equipamento para diferentes produtos sem tempo perdido de produção.

2) A capacidade de mudança de ajuste físico do equipamento sem perda de tempo de produção.

2.3 Fatores básicos para programação das tarefas

2.3.1 Comunicação utilizando entradas e saídas digitais (I/O)

Num processo de manufatura é necessário sincronizar procedimentos e até trocar informações do processo da máquina e vários dispositivos. Para este propósito estes dispositivos (robô, máquina-ferramenta, controlador das células de trabalho, etc...) devem possuir interface comum de comunicação. Portanto, uma interface simples e robusta com alta confiança de transmissão se torna geralmente aceita, por exemplo, comunicação através de entradas e saídas digitais I/O.

Esta forma de comunicação é um método muito antigo e simples de comunicação entre sistemas automatizados industriais, é a troca de dados pelos meios de entradas e saídas digitais (I/O). Cada fio tem um determinado sentido e deverá ser ajustado para lógica 1 ou 0 a partir do lado onde o sinal foi enviado. O receptor do outro lado do fio pode testar este sinal digital e iniciar as ações apropriadas. Em nível industrial, o nível de lógica 1 é freqüentemente representado por um nível de sinal de corrente contínua de 24V, o nível 0 corresponde a 0 V.

Vários sinais podem ser combinados em uma ordem de transmissão de dados. Contudo, muitos sinais tem seu próprio par de I/O. Este tipo de comunicação, que pode ser também aplicado para a comunicação entre CLP, sensores e atuadores, acarretando uma complexa instalação de fios com possivelmente imensos cabos trançados e blindados. Contudo, não é aconselhável para comunicação de longas distâncias (superior a 20 metros).

Um exemplo deste tipo de comunicação é a interface I/O entre robôs e periféricos. Todos os dados para a sincronização de ambos os dispositivos são trocados sobre uma interface digital I/O. Por causa desta simplicidade e robustez. Mesmo limitada, pela sua simplicidade e baixo custo de implementação, esta interface atualmente é largamente utilizada.

2.3.2 Comunicação Field Bus

A comunicação I/O apresenta muitas dificuldades no cabeamento de grandes plantas industriais, envolvendo altos custos de cabeamento, calhas, e manutenção e reparo. Nos dias atuais, ocorre cada vez mais a substituição dessa forma de transmissão de sinais pelo barramento “Field Bus”. Este barramento utiliza somente um cabo de par trançado ao invés de um cabo de fios blindados. A instalação desse sistema Field Bus, é fácil e sem excesso de material, e de fácil configuração, exigindo ainda um alto nível de especificação do S.A. O sistema Field Bus é de expansão flexível e permite alta taxa de comunicação de dados.

Sob o ponto de vista de níveis de comunicação, o controle básico de robô é agora equipado com hardware para controlar seus próprios periféricos tal como “Field Bus” meios de barramentos tais como Devicenet, Interbus, Profibus, etc... Contudo, o controle de robô baseado em PC é também capaz de comunicação com periféricos mais complexos, como os sistemas de visão disponíveis comercialmente, leitores de códigos de barras, pagers, e etc...

Interfaces de comunicação estão usando softwares e hardwares padrões facilitando os seus interfaceamentos com outros sistemas externos, permitindo ao usuário melhorar os desenvolvimentos, diagnósticos, manutenção e ferramentas estatísticas. Estas possibilidades reforçam a concepção de integração entre sistemas “Proprietários” e “Abertos”, (Fiedler, 2003), ilustrado na figura 2.4, como uma representação de arquitetura sistema “Aberto” adaptado por Freitas(2004).

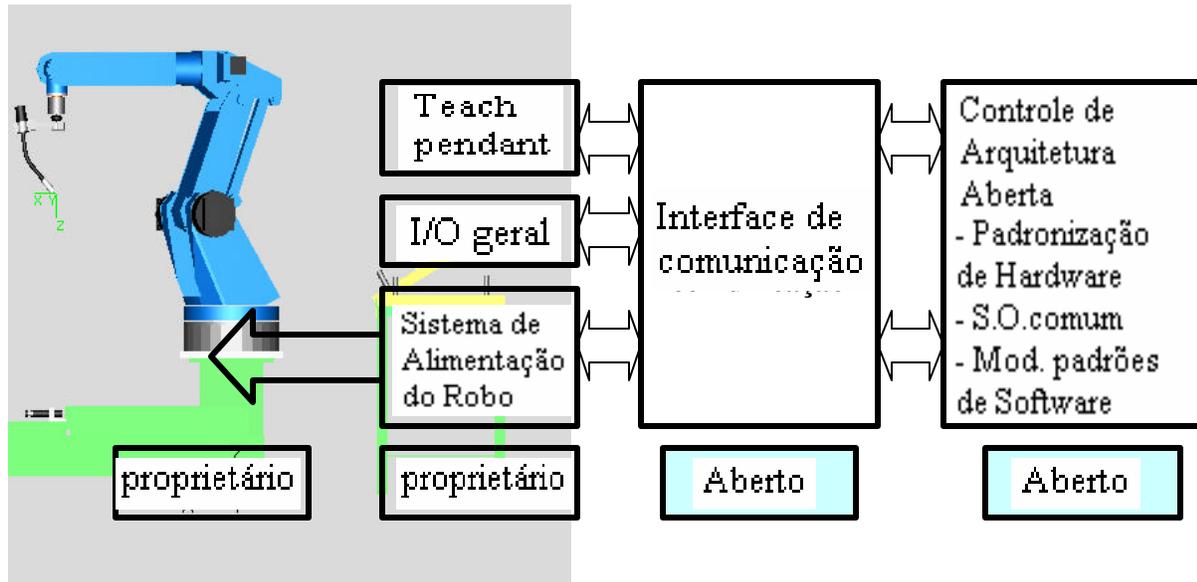


Figura 2.4 Representação de arquitetura sistema “Aberto”, adaptado por Freitas 2004, de Fiedler, P. Schilb, C.J. 2003.

2.3.3 Protocolos de comunicação entre sistemas de controle e robôs

O próximo nível de comunicação refere à de troca de informação entre sistemas de controle de robôs na mesma célula de trabalho e informações externas provenientes de computadores pessoais (PC) de escritórios. Na configuração de uma célula de trabalho, não está muito longe, de ter um controle mestre, como um Controlador Lógico Programável - CLP recebendo informação a partir de outro controle, processando e enviando para outros periféricos, exigindo a utilização de protocolos de comunicação padronizados.

Um dos principais inconvenientes é o fato que muitos sistemas utilizam formato de dados de sistemas proprietários. O protocolo de comunicação TCP/IP é um padrão de interfaceamento bem conhecido e validado. Existe um número de ferramentas baseadas em programação e diagnóstico TCP/IP disponíveis para quase todas plataformas concebíveis podem ser utilizadas para transferir informações entre dispositivos com uma dada célula de trabalho, e também entre

diferentes ações no chão de fábrica ou em escritórios. Por outro lado, para dispositivos ethernet, o controle de robô baseado em PC, normalmente possui um hardware padrão de comunicação (por exemplo, através da porta serial, paralela, barramento universal ou outros).

Um outro nível de comunicação com a célula de trabalho é através da transferência e utilização de informações relativas ao processo. Como exemplo, podem-se citar informações relativas aos processos de soldagem, ou informações provenientes da leitura e tratamento de sensores externos, usados para fazer controles mais inteligentes a partir de um toque sensoriado para encontrar juntas de solda e ou determinação do tamanho de peças.

Entretanto, muitos sistemas não coletam as informações das juntas a soldar quando realiza processamento simultâneo de um conjunto de informações para executar alguma decisão inteligente baseada nas estatísticas de uma junta paralela à outra junta. Por exemplo, um robô pode ser programado para inteligentemente sentir através de um toque, por vez, nas uniões das juntas selecionadas de uma dada peça, em uma ordem para melhorar o ciclo de trabalho, (por exemplo, por causa das variações nas peças pré-soldadas de um grupo em relação a outro).

Outro exemplo é o uso de informações compensadas estatisticamente, para coletar e comunicar esta informação com a gerência da célula de trabalho para dar então conhecimento da consistência das peças pré-soldadas. Similarmente, os métodos e informações trocadas aproximam a abordagem usada com o processo de toque sensoriado podendo ser usado com uma “costura” do caminho de soldagem e outros processos de células de trabalho.

2.3.4 Gerenciamento simples e multi células de trabalho.

O controle baseado em PC está facilitando o uso de padrões internos de comunicação e também ferramentas e protocolos de comunicação externas de células de trabalho. O uso de comunicação moderna e sistemas de ferramenta para simples células de trabalho, grupos de células de trabalho e linhas de produção estão revolucionando os sistemas de gerenciamento. Estas ferramentas podem prover dados estatísticos numéricos dentro do sistema operacional (por

exemplo, num processo de soldagem, a quantidade total do consumo de arame de solda utilizado no processo de soldagem), prover operações com mais informação e redução do tempo de parada.

Uma rede de comunicação é uma simples estrutura subjacente que é requisitada para automatizarem muitas destas concepções de gerenciamento. Existem muitas topologias de “network” capazes de executar as tarefas requisitadas de comunicação. Uma rede de comunicação pode ser tão simples como uma conexão de telefone até o controle do robô ou tão complexo quanto um sistema LAN (rede de trabalho de uma área local) com “firewalls”. A figura 2.5 mostra quatro exemplos de rede de comunicação, sendo três com gerenciamentos LAN.

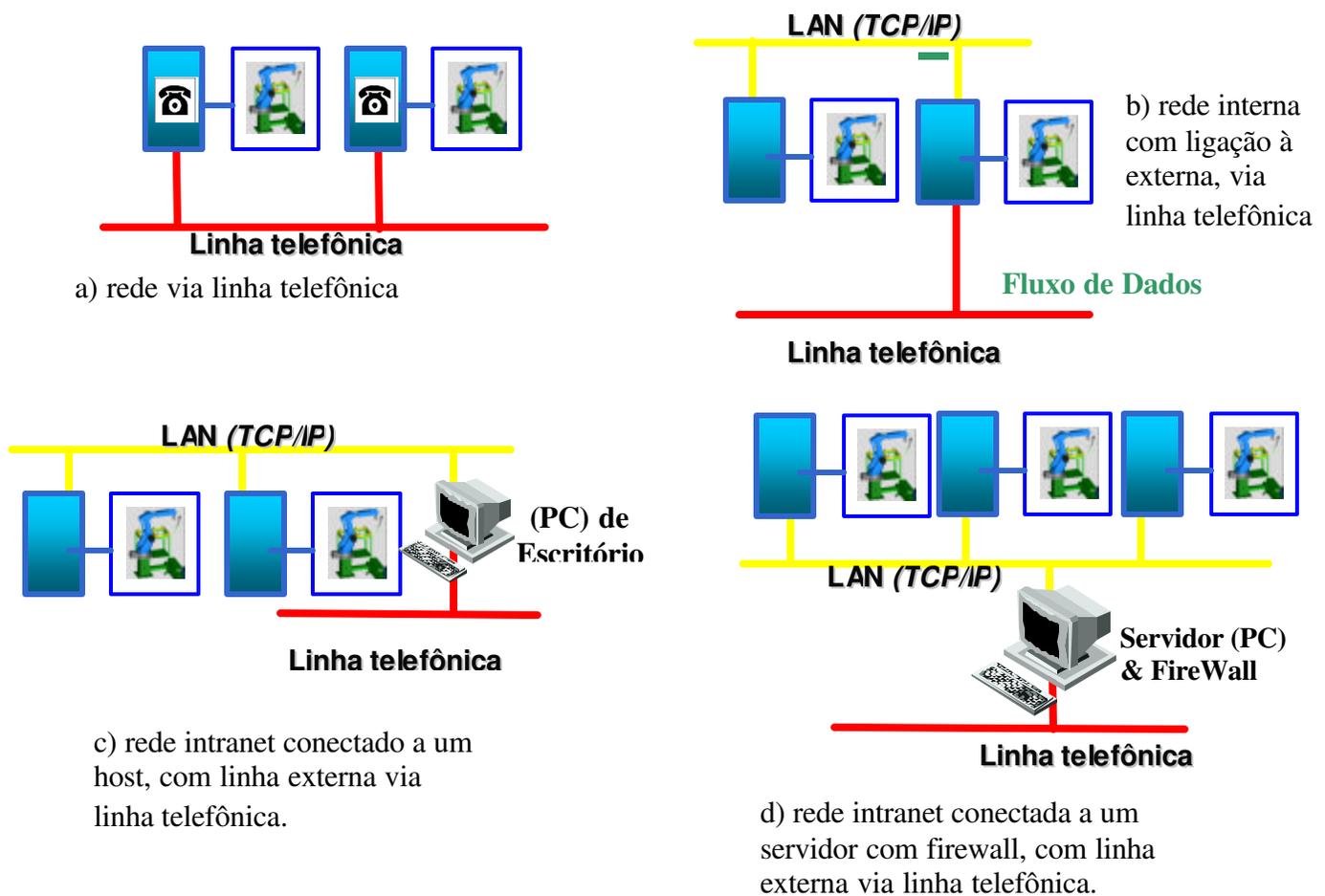


Figura 2.5 Exemplos de redes de gerenciamento.

2.3.5 Programação de tarefas de robôs

Além da tecnologia de comunicação com os seus diferentes níveis é necessário observar alguns fatores básicos para a programação de tarefas. A coordenada do ponto de movimento deve ser identificada e armazenada na unidade de controle. O ponto pode ser armazenado como coordenada de eixo da junta individual ou coordenada de posição. As funções devem ser identificadas e gravadas e serão executadas durante a execução de trajetórias.

As informações relativas aos pontos e funções são organizadas dentro de um caminho de seqüência lógica e subseqüência estabelecendo caminhos realizados sobre condições específicas. Por exemplo, uma seqüência lógica pode ser definida de forma simples através de uma ferramenta que auxilie a visualização lógica de um processo, ilustrado de forma adequada através da representação “GRAF CET” (tratado como SFC - Sequential Flow Chart, norma IEC 1131-3).

Neste exemplo, é necessário um robô, para pegar uma nova peça em uma “esteira” de transporte, onde o mesmo é integrado a uma máquina de usinagem, sistema de transferência de peças através de esteira e um sistema de segurança.

Um programador conduzira o robô passo a passo através de um ciclo de operação manual e gravará cada movimento no controle do robô (programação ponto a ponto). Adicionalmente, dados relativos às funções e parâmetros de movimento serão inseridos quando os pontos forem programados. A interface homem máquina (IHM) é usada para posicionar o robô, contudo os dados de entrada podem ser requisitados e necessários de serem especificados através da interface de controle, como mostra a representação gráfica da situação apresentada na figura 2.6.

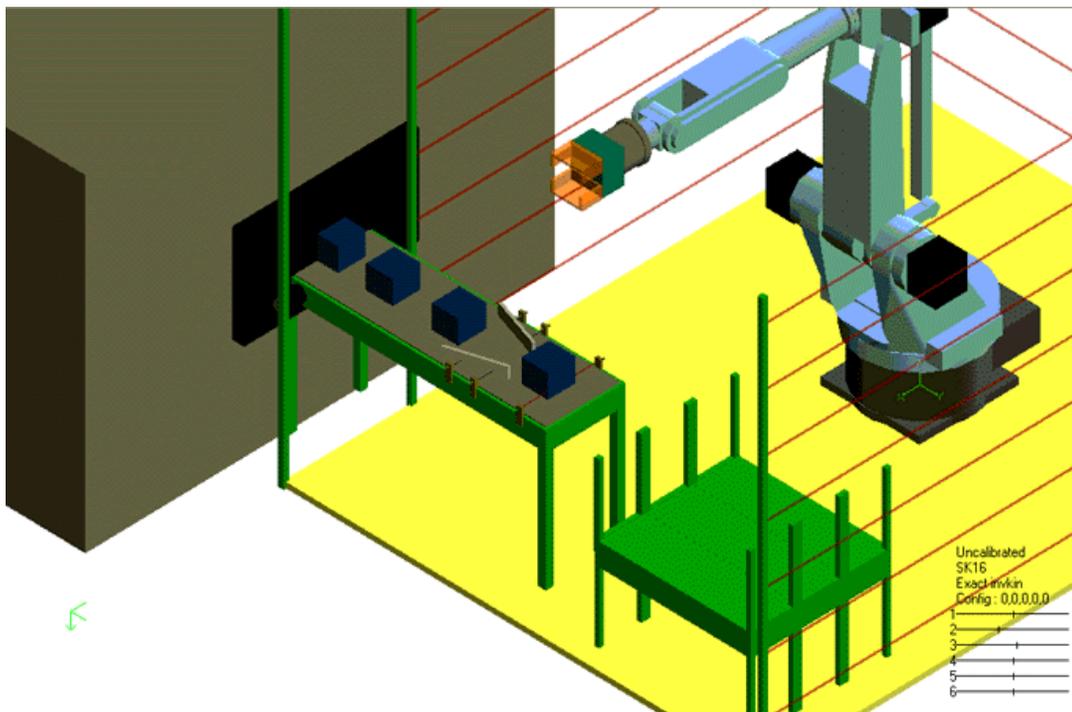


Figura 2.6 Representação gráfica da situação

Informações para o desenvolvimento da programação:

- a) As peças são transferidas através de uma esteira do processo de usinagem;
- b) Um sensor de infravermelho na esteira informa a chegada da peça;
- c) A célula tem um sensor de cortina de infravermelho para garantir segurança. A entrada de uma pessoa, por exemplo, faz o controlador acionar uma parada de emergência;
- d) O gerenciamento da célula é feito pelo controlador do robô;
- e) Os comandos de entrada e saída de sinais (I/O) são utilizados para fornecer as informações de chegada da peça (sensor infravermelho), da abertura e fechamento da garra do robô e do sistema de segurança (cortina de infravermelho);
- f) Os comandos de movimento para a trajetória do robô devem levar em consideração que o robô se aproxima da peça na esteira (o que implica em movimentos de aproximação), transporte da peça para o "pallet" e se afasta do "pallet", voltando à posição inicial.

A seguir, tem-se a representação gráfica em projeção da planta da célula, figura 2.7, que será usada como referência para a programação da tarefa. Ela pode facilitar a definição dos pontos de movimento e a visualização geral da tarefa a ser realizada.

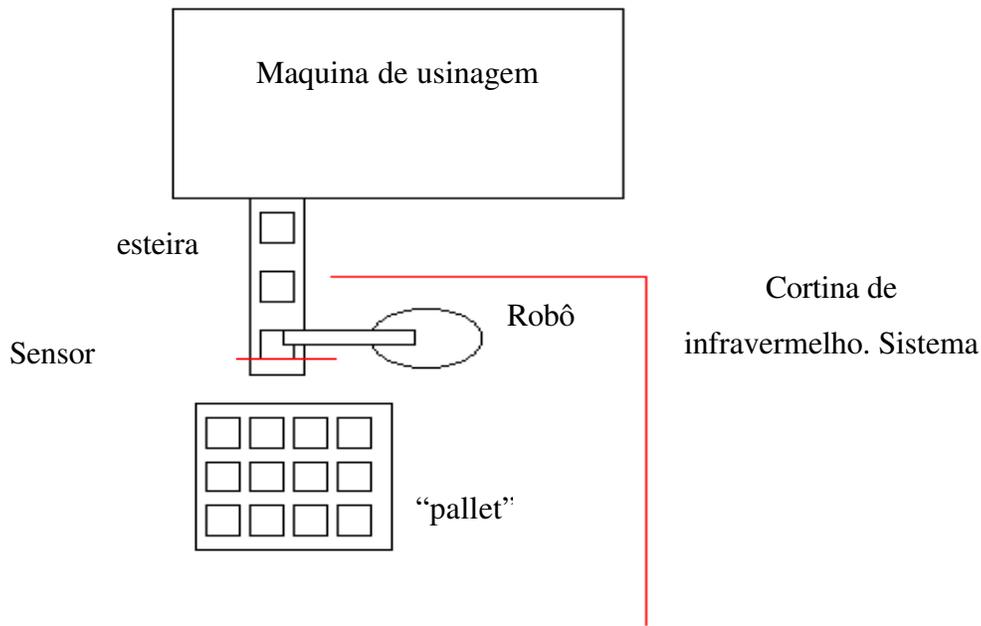


Figura 2.7 Representação gráfica em projeção da planta da célula

A figura 2.8 apresenta o Grafset Funcional, descritivo na situação em estudo. A utilização do Grafset funcional permite de forma clara, visualizar a lógica de um sistema automatizado seqüencial, com suas possíveis evoluções ou uma seqüência que se desenvolve simultaneamente (Rosário, 1998), representado graficamente por um conjunto de:

- a) *Etapas*, às quais são associadas ações;
- b) *Transições*, às quais estão associadas receptividades, (informações que permitem a sua transposição);
- c) *Ligações orientadas*, ligando as etapas às transições e as transições as etapas.

As regras de evolução são representadas pela alternância entre etapas e transição e vice – versa. Cabe ressaltar que nenhuma ação é realizada ou nenhuma receptividade é avaliada em um espaço de tempo nulo, isto é estas operações não são instantâneas.

O comportamento do Grafcet é baseado em cinco regras de evolução, as quais são: a inicialização fixa as etapas ativas no início do funcionamento; uma transição pode ser validada ou não validada; a transposição de uma etapa provoca a ativação de todas as etapas imediatamente a seguir e a desativação de todas as etapas imediatamente precedente; várias transições simultâneas transponíveis são simultaneamente transpostas; se no decurso do funcionamento, uma mesma etapa deve ser desativada e ativada simultaneamente, ela permanece ativa.

O Grafcet Funcional auxilia a visualização lógica do processo, possibilitando a implicação tecnológica através do Grafcet Tecnológico.

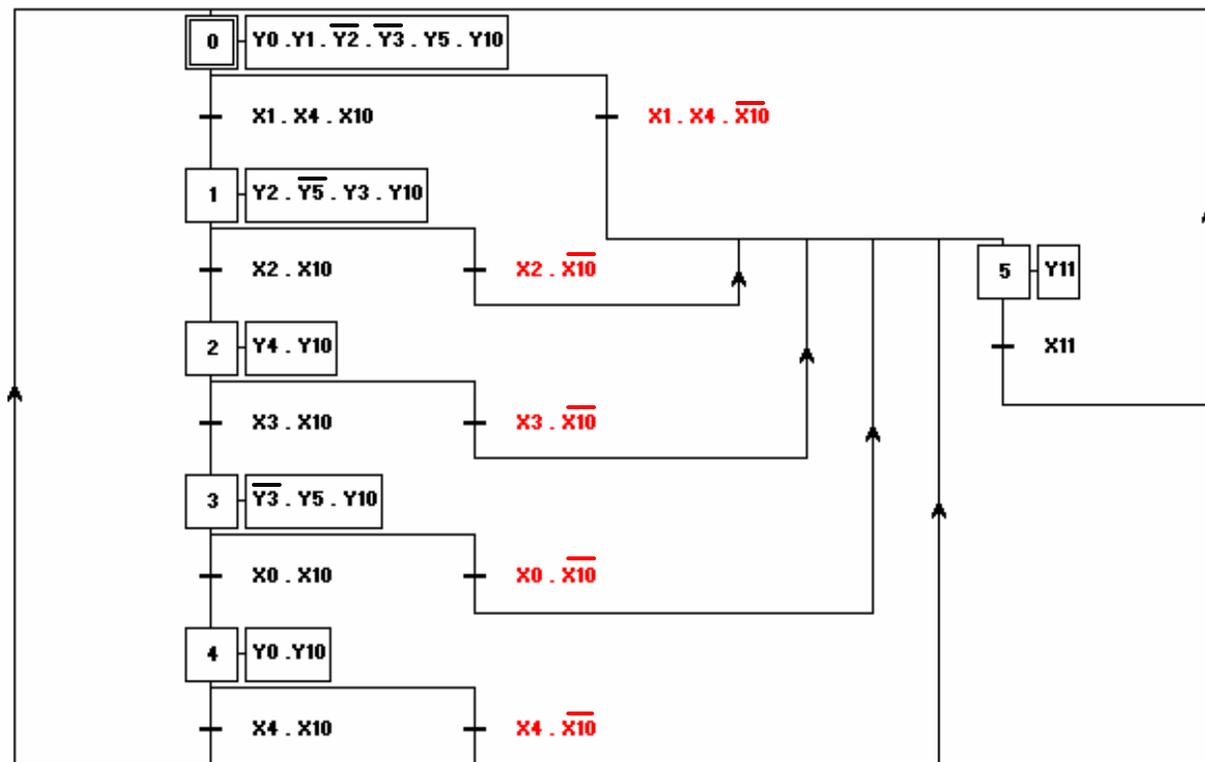


Figura 2.8 Grafcet Funcional

A lista seguinte mostra as variáveis introduzidas no Grafcet Funcional, com o respectivo significado.

SAIDAS: (*Etapas*)

Y0 = Movimentação do robô para posição inicial

Y1 = Acionamento da esteira de transporte da peça

Y2 = Movimentação do robô até a esteira

Y3 = Fechamento da garra

Y4 = Movimentação do robô até o “pallet”

Y5 = Abertura da garra

Y10 = Acionamento do sistema de segurança

Y11 = Parada de Emergência

ENTRADAS: (*Transições*)

X0 = Robô na posição inicial

X1 = Peça na esteira

X2 = Garra fechada

X3 = Robô na posição “pallet”

X4 = Garra aberta

X10 = Sistema de segurança normalizado

X11 = Reset manual do sist. de segurança



2.3.6 Programação de robôs industriais

Normalmente pode-se programar robôs industriais de modo on-line e off-line. A programação on-line esta associada à condição em que o robô está sendo manipulado, conduzido dentro de um determinado processo industrial, sendo baseado na aprendizagem de tarefas em ambientes industriais, ou seja, a movimentação das juntas até as posições requeridas. A programação off-line é implementada através de ambiente de programação estruturada ou visualização gráfica.

2.4 Programação on-line de robôs industriais

2.4.1 Método de aprendizagem (Teaching)

O painel de acionamento e controle¹ “Teach Pendant” é utilizado para movimentar as juntas de um robô através de seqüências de movimentos pré-definidas, de modo a interagir com o processo através de sua ferramenta terminal, como, por exemplo, alimentando com o material de adição, arame, no processo de soldagem “GMAW” ou em um processo de manuseio, abrindo ou fechando uma garra.

O “Teach Pendant”, consiste de um painel de teclas ou chaves de controle através da qual as juntas do robô podem ser movimentadas. O controle do robô deve estar ajustado no modo de aprendizagem. As informações dos movimentos e outros dados necessários são gravados pelo controlador do robô, guiando o robô através do caminho desejado durante o processo de aprendizagem.

¹ O painel de acionamento e controle é a interface homem-máquina, que possui diversos nomes na literatura, tais como “Teaching Box”, “Teaching Pendant”, etc... Não é usual a tradução deste termo para a língua portuguesa, ficando um uso à nomenclatura de origem de seu equipamento que é adotado de maneira própria por cada um dos fabricantes de robôs no mercado, por exemplo, a Motoman, utiliza o nome “programming pendant” (PP); Kuka”, Kuka painel control “(KPC); Reis, Main programming control (MPC); etc...”.

Nos pontos específicos no caminho de movimento o operador pode também posicionar ou seqüenciar (periféricos), relacionados com o envelope de trabalho do robô. Editar programa é usado para somar dados suplementares ao programa de controle de movimento para operação automática do robô ou a equipamento (periférico) associado à produção.

Por outro lado à edição de programas ensinados promove um significado de correção ou modificação de um programa de controle existente para mudar um ponto incorreto ou compensar uma mudança na tarefa a ser executada. Durante o processo de ensinamento o operador pode desejar retornar diversos segmentos do programa para uma verificação visual de movimento ou operação. Ensino de características de retomada pode incluir “forward”, adiante e “backward” atrás, operações de passos simples e operações de seleção de retomada de velocidades de movimentos.

A escolha de aproximação no aprendizado de programação é um pouco dependente do controle de algoritmo utilizado para movimentação do robô através de um caminho desejado.

A programação do robô implementada através do controle “POINT TO POINT” (PTP) movimentada a partir de uma posição até a próxima sem considerar o caminho a executar pelo manipulador. Geralmente, cada eixo movimenta-se na sua máxima razão ou limitada até que ele alcance a posição desejada. Apesar de que todos os eixos iniciarão os movimentos simultaneamente, eles não necessariamente completarão seus movimentos ao mesmo tempo.

Controle “CONTROLLED PATH” (CP) concerne o controle das coordenadas de todas as juntas de movimento para alcançar um caminho desejado entre dois pontos programados. Neste método de controle, cada eixo move suavemente e proporcionalmente para fornecer um movimento de caminho controlado, previsível.

2.4.2 Método Walk through

Este método utiliza usualmente os meios de um braço ensinado, onde o programador posiciona manualmente o braço do robô através de vários movimentos, através do contato físico, deslocando e controlando o braço do robô através de posições desejadas dentro do envelope de trabalho do robô. Durante esse tempo o controle do robô irá varrer e armazenar valores de coordenadas de posição, dentro de uma base de tempo fixada. Gravadas as posições e outras informações funcionais, dentro de uma memória, para rodar em uma forma “PLAYBACK” (executar a repetição das posições gravadas e suas informações funcionais), como foi ensinado durante a produção. A velocidade pode ser controlada de forma independente. O controle de algoritmo utilizado para mover o robô no método “Walk through” através do caminho desejado é o controle “CONTINUOUS PATH”, CP, que concerne a retomada de pontos espaçados próximos que foram gravados quando o robô foi guiado ao longo de um caminho desejado. A posição de cada eixo foi gravada pela unidade de controle dentro de uma base de tempo constante pela varredura dos ‘transdutores’, nos eixos durante o movimento do robô. O algoritmo tenta retomar até duplicar aquele movimento.

2.5 Programação off-line de robôs industriais

A programação off-line será agora abordada, é um método de ensino que vem ganhando novos impulsos com o desenvolvimento da tecnologia, bem como uma tendência suscetível de tornar-se popular no uso da programação de robôs, com o aumento de fato da flexibilidade e habilidade de se arranjar para novas produções, com uma variedade ilimitada de cenários e movimentos. Programação off-line deve ser considerada como o processo através do qual programas de robôs são desenvolvidos, parcialmente ou completamente, sem a necessidade do uso do próprio robô. Os dados de coordenadas dos pontos são gerados, como também os dados de função e o ciclo lógico.

Desenvolvimentos na tecnologia de robô, tanto de hardware quanto de software, estão a fazer da programação off-line um método mais viável. Estes desenvolvimentos incluem grandes sofisticções na unidade de controle do robô, a qual melhora as precisões de posicionamento, e adoção de tecnologias de sensores. Há atualmente, atividades consideráveis no método de programação off-line e estas técnicas estão sendo melhoradas nas indústrias de manufatura, que proporciona uma forma efetiva de utilização desta prática.

É interessante para um entendimento inicial, comparar os métodos de programação, assim é possível uma visão mais clara das alternativas de ensinamento.

Na programação on-line o operador programador necessita pegar e colocar as peças-produtos, os dispositivos (JIG's) de fixação e posicionamento, os robôs, os periféricos, etc... ao redor da produção e a envolve na célula de trabalho. O robô e todo o sistema (célula de trabalho) são integrados e programados com o auxílio de um "Teach Pendant".

A programação on-line permite uma fácil acessibilidade, nos componentes da célula de trabalho, ao realizar as gravações de posições e orientações. Os fatores que inibi uma maior capacidade de programação são:

- a) Movimento lento do robô quando programado;
- b) Uma lógica de programação e cálculos são pesados para o programa;
- c) Suspensão da produção na execução da programação;
- d) Custo equivalente ao valor de produção, mal documentado;

A mais significativa vantagem da programação on-line é que o robô é programado em concordância com a atual posição dos equipamentos, dispositivos (Jig's) e peças. Contrariamente, a mais significativa desvantagem é que os equipamentos e periféricos tornam-se indisponíveis para a produção, já que são utilizados no processo de aprendizagem.

Na programação off-line, o operador programador insere virtualmente, posiciona e orienta, as peças-productos, os dispositivos (JIG's) de fixação e posicionamento, os robôs, os periféricos, etc... dentro de um ambiente gráfico computadorizado, com auxílio de software de modelagem gráfica e são usados modelos da célula de trabalho com robô, peças e ambientes. Os programas de robôs podem em muitos casos, serem criados pela reutilização de dados de CAD (Desenhos Auxiliados por Computador), existentes de forma que a programação seja rápida e efetiva. Na figura 2.9, pode ser observado um exemplo de uma célula de trabalho, em ambiente virtual, os programas de robô são verificados dentro de simulações e qualquer erro é corrigido.

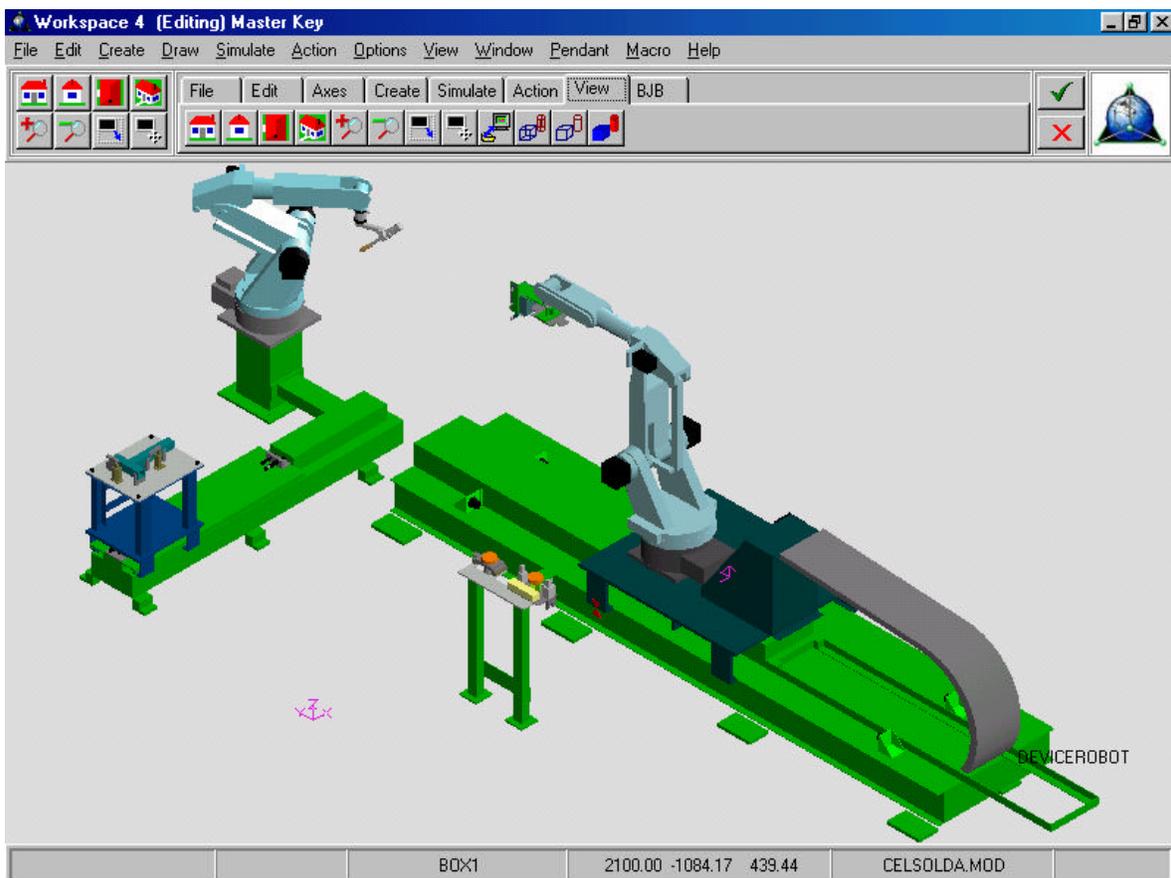


Figura 2.9 Exemplo de uma célula de trabalho.

A programação off-line possui uma grande necessidade de investimento em um sistema de programação off-line e conseqüentemente, necessita de mão de obra mais qualificada, o que torna uma desvantagem em relação à programação on-line. Os fatores que propiciam uma maior capacidade de programação no processo off-line são:

- a) Programação efetiva da lógica de programação e cálculos com as facilidades do “estado da arte” de depuração;
- b) As localizações são construídas de acordo com os modelos e isto pode significar que os programadores terão uma precisão apurada de programas;
- c) Verificação de programas e lógicas através da simulação com a visualização;
- d) Modelo bem documentado com a simulação, com o programa apropriado;
- e) Reutilização de dados CAD (Desenho Auxiliado por Computador) existentes, quando serão usados para simulação;
- f) Custo independente da produção. Durante a programação off-line a produção não para;
- g) Os processos suportam diferentes ferramentas, como por exemplo, seleção de parâmetros de soldagem.

A grande vantagem da programação off-line é que não é necessário ter disponível os equipamentos para realizar a programação. O investimento em equipamento será feito depois de realizado o programa e o estudo de processo, por exemplo. De outra forma, a vantagem é que não se ocupa equipamento de produção e dentro deste modo à produção pode continuar durante o processo de programação. Sem dúvida a maioria dos robôs hoje, estão sendo programados on-line isto é devido principalmente ao fato de que a programação off-line tem tido uma forte razão à demanda e à necessidade de usuários especialistas.

Avançadas ferramentas de programação off-line contem facilidades de depurar e isto assiste efetivamente a programação. A ferramenta de programação admite utilização de ferramentas de suportes para o processo de programação, por exemplo, otimização do processo de soldagem, montagem, pintura etc..., Muitas aplicações de robô envolvem processos de produção em massa, como por exemplo, soldagem em linhas automotivas, onde o tempo de reprogramação exigido é ausente ou mínimo. Contudo, para a aplicação robótica ser possível no campo de pequenos e

médios btes de produção, onde o tempo de programação possa ser volumoso, um sistema de programação off-line torna-se essencial (Yong, 1999).

Existe um enquadramento sensível de forma contundente do mercado brasileiro neste contexto de pequenos e médios lotes de produção. De forma discreta as utilizações de robôs vêm crescendo e o aumento da complexidade da aplicação robótica, particularmente com respeito a trabalho de montagem, soldagem vêm fazendo das vantagens da programação off-line atrativa. Uma outra vertente significativa, no mercado nacional é a utilização da simulação, que é uma consequência da programação off-line como uma ferramenta a mais, para orientar um cliente em potencial da necessidade de uma determinada solução automatizada robotizada, culminando para uma tomada de decisão em cima de fatos virtuais muito próximos do real. A realidade do mercado nacional, ainda é muito tímida. Todos os processo que envolvem robótica possuem uma expectativa muito alta, o que agrega um custo também elevado. Nos mercados europeus, asiáticos e norte americano o processo de programação off-line e simulação é significativo em virtude da relação custo benefício. A programação off-line e simulação atuam de forma expressiva e pode ser resumida, como se apresenta:

- a) Redução do ciclo de paradas do robô;
- b) Remoção do operador programador de meios potencialmente perigosos, no momento em que a maioria do desenvolvimento dos programas são feitos longe do robô, reduz-se o tempo durante o qual o operador programador está na área de risco, junto ao equipamento;
- c) Um único sistema de programar. O sistema pode ser usado para programar uma variedade de robôs sem a necessidade de se conhecer a sintaxe da linguagem de cada unidade de controle de robô;
- d) Integração com sistemas existentes de CAD/CAM;
- e) Simplificação de atividades complexas;
- f) Verificação dos programas, softwares de programação e simulação podem ser usados para demonstrar tarefas livres de colisão antes da geração do programa do robô, conforme a figura 2.10.

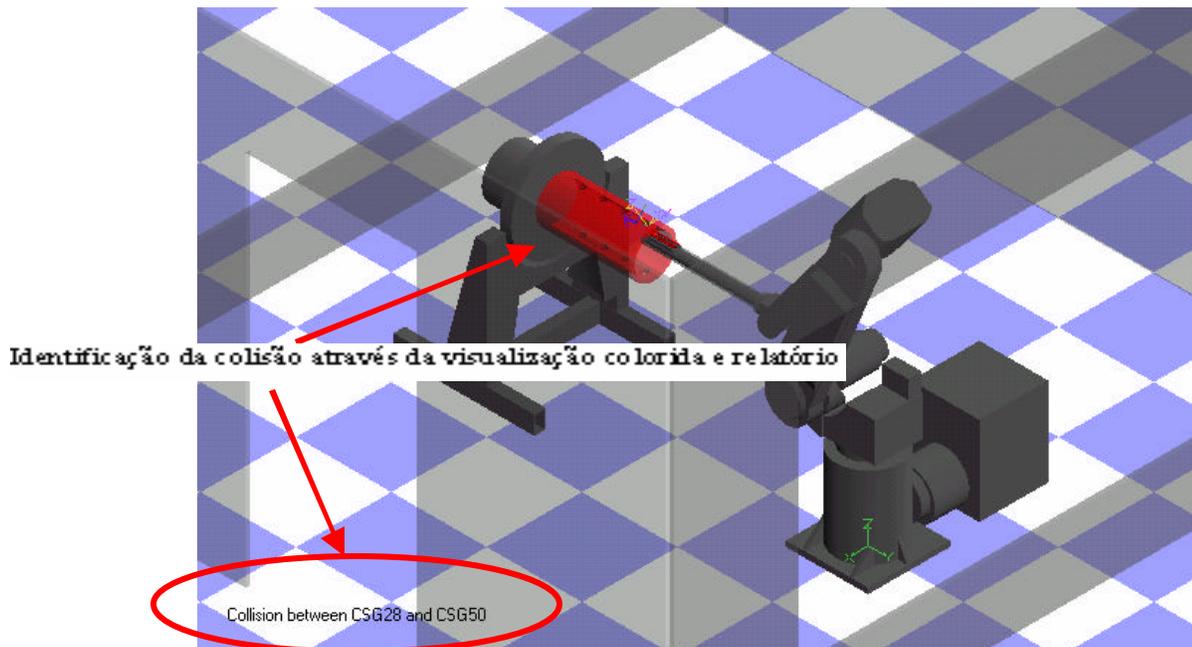


Figura 2.10 Programação off-line – Identificação de colisões.

É possível enumerar outras possibilidades de utilização da ferramenta de programação off-line e simulação, além das descritas, tais como:

- a) Planejamento da produção;
- b) Projeto e otimização de “layout”;
- c) Projeto de produto e otimização;
- d) Coleta e supervisão de dados;
- e) Construção de modelos de robôs e Educação.

De uma forma mais abrangente, é possível alternar os métodos de programação em on-line e off-line, conforme a melhor utilização de suas vantagens otimizando as técnicas de programação conforme necessidade e facilidade de desenvolvimento.

Essa variante de programação é denominada de “Programação Híbrida”, (Trotman, 1988), um programa de robô consiste principalmente de duas partes:

- a) Localização (posição e orientação);
- b) Lógica de programação (estrutura de controle, comunicação, cálculos).

O programa lógico pode ser desenvolvido off-line com a efetiva facilidade de depuração e simulação. As principais partes de comandos de movimentos podem ser criadas off-line pela reutilização dos dados de CAD (Desenhos Auxiliados por Computador) e pela interação do programador. Comandos de movimentos para localização e colocação de peças dentro das células robotizadas podem ser programadas on-line se for necessário. Dentro desta ótica, ambos os métodos podem ser vantajosos e utilizados, propicia uma melhora de flexibilidade na produção. Após a verificação do programa do robô na simulação o programa é transmitido² para a unidade de controle do robô.

A correta combinação de programação on-line e programação off-line levam principalmente a redução de custo dentro dos ajustes de produção.

Diante das incertezas no mercado nacional, tal tecnologia pode ser ajustada conforme o grau de domínio que cada usuário dispõe. Embora nos dias atuais, esta ferramenta de programação no mercado seja totalmente tímida e porque não dizer desconhecida pela maioria dos usuários, cria-se uma barreira muito grande, em primeiro lugar pelo investimento na capacitação do profissional para desenvolver a potencialidade de trabalho no software, investimento de aquisição de ferramenta e em seguida pela credibilidade de uso na integração dos processos.

² processo de transmissão do programa do computador para a unidade de controle do robô é chamado de “download” e quando transmitido da unidade de controle do robô para um computador “upload”.

2.6 Conclusão

Neste capítulo foi apresentada uma revisão bibliográfica direcionada a Integração de Robôs em Ambientes Industriais, sendo abordado uma revisão dos conceitos de sistemas automatizados e seus atributos, passa pela concepção de sistemas automatizados, modelagem de sistemas a eventos discretos e conceitos de estruturação, enfatiza as diferentes possibilidades de uso mais flexível de um robô dentro de uma célula automatizada, aspectos relacionados à programação de robôs industriais e redes de comunicação em ambientes industriais, envolvendo a utilização e programação de robôs de dispositivos automatizados constituintes de células de manufatura a distância.

No próximo capítulo serão apresentados aspectos relacionados à Programação Off-Line de Robôs, métodos de programação, e conceitos básicos de modelagem cinemática direta e inversa, enfatizando conceitos de estruturação e simulação gráfica.

Capítulo 3

Programação Off-line de Robôs

Neste capítulo são abordados aspectos concernentes à programação off-line de robôs, através da apresentação dos requisitos básicos necessários para a programação estruturada e simulação, conceitos básicos de modelagem cinemática, enfatizando o problema de orientação da ferramenta, e sistemas de coordenadas e procedimentos de inicialização e calibração da ferramenta terminal, direcionando o enfoque para a concepção de sistemas automatizados dentro da engenharia, baseada em modelos virtuais, propiciando análises e decisões mais próximas do real.

3.1 A utilização de programação off-line de robôs

O sistema de programação off-line advém de um processo que requer o conhecimento exato dos parâmetros geométricos e cinemáticos associados aos modelos geométrico e cinemático direto e inverso de robôs industriais.

Normalmente, alguns, fabricantes de robôs industriais possuem os seus sistemas específicos de programação off-line, otimizando os modelos desenvolvidos para os equipamentos, de forma que propicie a sua utilização em sistemas baseados em plataforma “PC”. A representação gráfica

propicia uma interação mais agradável e real no ambiente virtual, alinhado com as atuais capacidades de processamento (*há mais variedade relativamente grande de possíveis configurações nos micro-computadores - PC's que acabam gerando variações significativas em alguns estudos de casos e aplicações*), que torna o processo muito realístico.

Somando a possibilidade de controle dos robôs virtuais, têm-se os recursos de CAD, para a modelagem do ambiente virtual que criam a representação gráfica de peças e dispositivos que, possibilita, por exemplo, a interação completa de robôs dentro da célula de manufatura robotizada. O somatório dessas características permitem executar um estudo de planejamento de layout, planejamento de movimentos do robô, equipamentos, dispositivos, considerando possibilidades de segurança e detecção de colisões, etc...

Uma das possibilidades de utilização de ferramentas de simulação e programação off-line é na concepção e especificação de novas células de manufatura robotizada, onde não haja nenhum equipamento disponível ou dispositivo desenvolvidos. Nesse caso, um estudo que necessite a integração de vários robôs e equipamentos independentes do fabricante, podem ser utilizados e analisados. A partir desta concepção se tem o primeiro problema, ao dispor de um pacote de software de simulação e programação off-line “proprietário”, ou seja, de um determinado fabricante de robô, ficará limitado aos modelos de robôs disponíveis pelo fabricante.

Conseqüentemente, a solução é que se disponha de um pacote de software de simulação “aberto”, “universal”, onde se possa fazer uso de qualquer modelo de robô de diferentes fabricantes. Dos pacotes de software de simulação e programação off-line “universal” disponível comercialmente, existem alguns de uso consagrados nos mercados norte americano, canadense, europeu e asiático. Algumas vantagens que podem ser citadas no uso destes pacotes estão na possibilidade de criar modelos geométrico, modelos cinemático direto e inverso baseado na convenção Denavit-Hartenberg (“D-H”), que será abordada na próxima seção. Esses parâmetros cinemáticos de “D-H” podem ser implementados nos dispositivos a serem modelados, associados ao robô em estudo (comprimentos, distância entre “links”, variações angulares do deslocamento

das juntas de rotação, velocidade, aceleração, etc...), geralmente disponíveis nos catálogos dos fabricantes de robôs.

3.2 Convenção Denavit-Hartenberg ‘D-H’

A notação ‘D-H’, foi implementada por Denavit, para descrever as Transformações Homogêneas, que compõem as posições e orientações de sistemas de referência de coordenadas no espaço, por conseguinte as ‘linkagens’ que possibilitam a relação entre objetos e é usada para o estudo da manipulação dos robôs, através de uma representação que descreve os manipuladores (Paul, 1981). A figura 3.1 representa a cinemática espacial de um robô 6R,

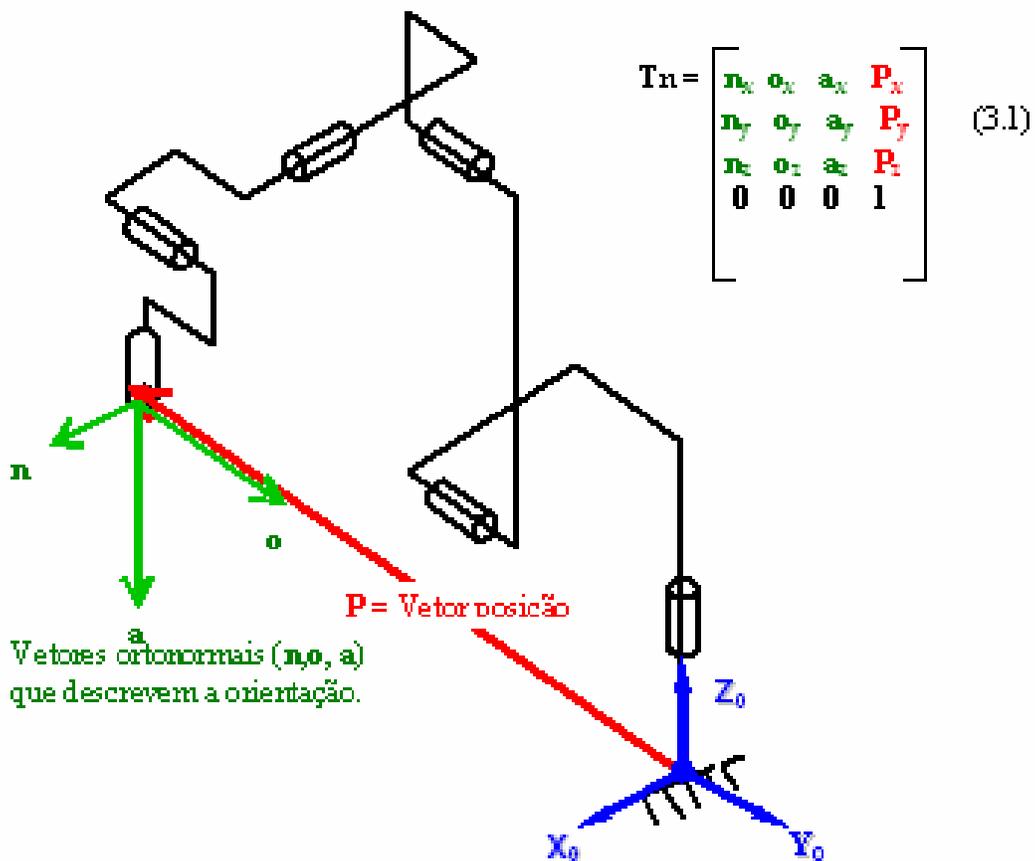


Figura 3.1 Representação da cinemática espacial de um robô 6R

A relação é expressa matematicamente pela equação (3.1), que relaciona o sistema de coordenada solidária à base do robô com um sistema de coordenadas associadas ao seu elemento terminal. Essa matriz é chamada de *matriz de passagem homogênea* (Rosário, 2002), historicamente foi chamada como matriz “T”, sendo obtida a partir do produto das matrizes de transformação homogênea, que historicamente é chamada de matriz “A” (Denavit,1955), $A_{n, n-1}$, que relaciona o sistema de coordenada de um elemento n com o sistema n – 1.

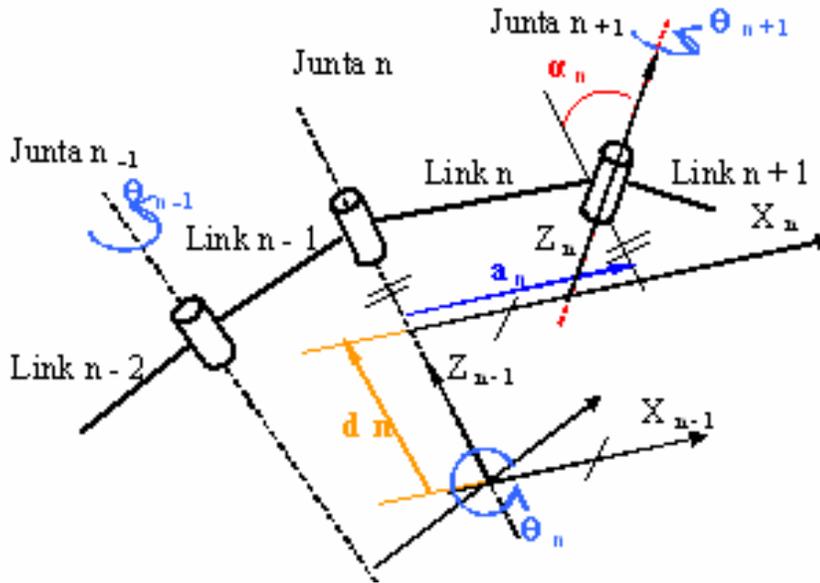


Figura 3.2 Parâmetros a , a , d , T .

Para especificar a matriz “A”, é necessário reconhecer os sistemas de coordenadas, de todos os “Link”s de acordo com um procedimento, conforme ilustrado na figura 3.2, descritos em (Paul, 1981), (Fu, 1987), (Craig, 1989), (Ferreira, 1991). Assim, quatro parâmetros (a_n , a_n , d_n , T_n), podem ser associados com cada “Link” do manipulador.

É possível estabelecer a relação entre sistemas de referências sucessivos $n-1$, n pela seguinte relação entre rotações e translações, conforme ilustra a figura 3.3:

- a) Rotação sobre Z_{n-1} , um ângulo T_n ;
- b) Translação ao longo de Z_{n-1} , uma distância d_n ;
- c) Translação ao longo de $X_{n-1} = X_n$, um comprimento a_n ;

d) Rotação sobre X_n , o ângulo de rotação a_n ;

Isto pode ser expresso como o produto de quatro transformações homogêneas relativas ao sistema de coordenada do “Link” n até o sistema de coordenada do “Link” $n - 1$. Conforme a equação 3.2.

$$A_n = \text{Rot}(Z, \theta) \text{Trans}(0, 0, d) \text{Trans}(a, 0, 0) \text{Rot}(x, \alpha) \quad (3.2)$$

$$\begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

$$A_n = \begin{bmatrix} \cos \theta & -\sin \theta \cos \alpha & \sin \theta \sin \alpha & a \cos \theta \\ \sin \theta & \cos \theta \cos \alpha & -\cos \theta \sin \alpha & a \sin \theta \\ 0 & \sin \alpha & \cos \alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

A base do manipulador é o “Link” 0 e não é considerado um dos seis “Link”s. O “Link” 1 é conectado até o “Link” base pela junta 1. não existe junta no final do “Link” final. O significado dos “Link”s é que eles mantêm um relacionamento fixo entre as juntas no final de cada um dos “Link”s. Qualquer “Link” pode ser caracterizado por duas dimensões: a distancia normal a_n e o ângulo a_n entre o eixo no plano perpendicular até a_n

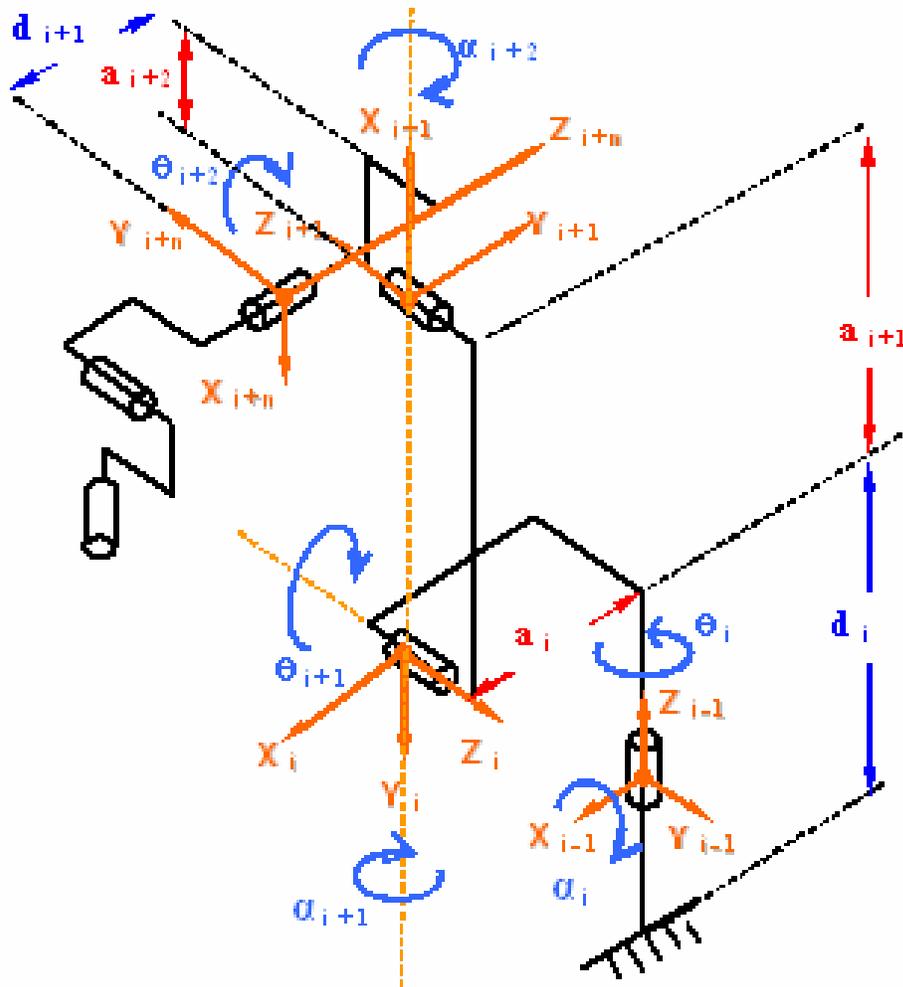


Figura 3.3 Esquema de procedimento da referência dos “Link”s

A notação de “D-H” é uma ferramenta utilizada para sistematizar a descrição cinemática de sistemas mecânicos articulados com n graus de liberdade (Denavit, 1955).

Na figura 3.3 podemos visualizar cinco “Link”s conectados por juntas. Um eixo de uma junta estabelece a conexão de dois “Link”s. Esses eixos de juntas devem ter duas normais conectadas a eles, uma para cada “Link”. A posição relativa desses dois “Link”s conectados (“Link” $i - 1$ e “Link” i) é dada por d_i , que é a distância medida ao longo do eixo da junta entre

suas normais. O ângulo de junta T_i entre as normais é medido em um plano normal ao eixo da junta. Assim, d_i e T_i podem ser chamados.

3.3 Conceitos básicos

Atualmente, estamos vivendo uma situação que requer uma quebra imediata dos paradigmas na engenharia tradicional, como país que se caracteriza de forma global em usuário de tecnologia, salvo algumas áreas de excelência. Nos processos industriais de fabricação, nas indústrias montadoras e mais ainda, nos fabricantes de autopeças responsáveis pelos fornecimentos às montadoras esse contexto é crítico.

Com a necessidade de atender a demanda industrial mundial que forneça qualidade, produtividade e preço, é preciso popularizar as tecnologias de automação, não só como usuários, mas como desenvolvedores e isso significam em todos os estágios do processo, da formação de profissionais, concepção (projeto) à fabricação.

A simulação propicia esta quebra de paradigma, onde a visualização não é tradicional no mundo da engenharia. Os problemas que são comuns em algumas situações no desenvolvimento de projetos são muito mais fáceis de serem identificadas pela visualização do que pela solução analítica. Ao mesmo tempo, isto representa uma redução significativa de custos na aquisição de equipamentos necessários para a montagem de células automatizadas.

Os processos industriais estão sofrendo uma revolução tecnológica, uma nova era da engenharia é um desafio que devemos encarar com seriedade, para que não se transforme numa questão de sobrevivência.

Muitos dos trabalhos encontrados relatam a pesquisa e desenvolvimento robótico. Alguns destes trabalhos distribuídos com controle de robôs industriais, o qual possibilitam automatização

de sistemas robóticos e melhoria de seus alcances de aplicações nos processos de manufaturas (Craig, 1989), (Groover, 1987), (Fu & Gonzáles, 1998), (Paul, 1981) dentre outros.

Para a modelagem, integração, simulação e programação off-line, detecção de colisões e análise de “lay out”, atualmente as companhias industriais estão usando estações de trabalho (Workstation) de última geração que possui capacidade de processamento superior aos computadores pessoais (PC), possibilitando um maior grau de elaboração e fidelidade, modelo gráfico mais detalhado a condições reais. Existem alguns pacotes de softwares de simulação disponíveis comercialmente com diferentes capacidades e preços.

Algumas pesquisas foram realizadas em sistemas de simulação de eventos discretos para planejamento e propósitos de projetos. A motivação para se usar simulação é que ela pode oferecer captura e descrição de interações complexas com um Sistema Flexível de Manufatura - “FMS” em particular, onde métodos analíticos, provavelmente falham. Simulação é utilizada, novamente para “insights” em sistemas de produção (Djuric, 1999).

Uma nova tendência da gestão de manufatura vem ganhando força nos ambientes fabril, como exemplo dessa tendência, é a solução da Tecnomatix Technologies Ltd. Tecnomatix eMPower, para o Gerenciamento no Processo da Manufatura (MPM) e Gerenciamento do Ciclo de Vida do Produto (PLM), possibilitando o projeto, simulação e execução de processos, providenciando um controle e visualização em tempo real através das operações no piso fabril.

Neste caso une-se logisticamente o projeto, processo e produção, possibilitando assim, virtualmente integrar via software os processos antes analisados de forma individual e ou separados. Essa concepção de gerenciamento reforça a utilização da simulação e programação off-line como uma filosofia de trabalho. Alguns pacotes disponíveis no mercado são consagrados no uso de simulação robótica, conforme mostrado na tabela 3.1.

Fabricantes	Software	Endereço
Tecnomatix Technologies ltd	RobCad®	http://www.tecnomatix.com/
Dessault Delmia	Igrip®	http://www.delmia.com/
Flow software Technologies	Workspace®	http://www.Workspace5.com/
Easy-Rob	Easy-Rob®	http://www.easy-rob.com/

Tabela 3.1 Exemplos de softwares comerciais disponíveis no mercado para simulação e programação off-line de robôs.

Existe também software de simulação comercializado pelos principais fabricantes de Robôs industriais, conforme mostra a tabela 3.2.

Fabricantes	Software	Endereço
ABB	RobotStudio®	http://www.abb.com
Fanuc	ROBOGUIDE SimPRO™	http://www.fanurobotics.com/Products/Software.asp
Kuka	KUKA Sim®	http://www.kuka.be/kukasim/main/portuguese.htm
Motoman	MotoSim®	http://www.motoman.com/products/software/simulation. htm
Reis	ProSim®	http://www.reis.com/

Tabela 3.2 Exemplos de softwares de simulação e programação off-line desenvolvidos por fabricantes de robôs industriais.

3.4 Modelagem cinemática do robô

O problema cinemático inverso refere-se à modelagem de um robô articulado, enquanto o problema geométrico, o de computar a posição e orientação do ponto central da ferramenta (TCP) do robô. Especificamente, fornecer ajustes angulares das juntas, é computar a posição e orientação do sistema da ferramenta em relação ao sistema de coordenada da base. Explicitar esta concepção é mudar a representação da posição e orientação do robô a partir de uma junta no espaço dentro de uma descrição espacial cartesiana.

Esta condição é clara no momento de se realizar um simples movimento, do robô no espaço, realizando uma trajetória de um ponto conhecido a outro, é claro que garantindo a posição e orientação do TCP. Os cálculos das juntas que formam a cadeia cinemática podem atender diversas condições e se comportarem de formas diferentes, garantindo o mesmo trajeto do TCP, este comportamento se difere do controle da cinemática direta em função das *não-linearidades* que surgem das várias possibilidades de solução de trajeto, e não são sempre possíveis de se obter. Outras condições podem ainda surgir, as equações da existência de uma solução e de múltiplas soluções.

Diferente das equações lineares, não há algoritmos gerais que podem ser empregados para resolver um ajuste de equação não linear. A maior parte dos manipuladores robóticos industriais possuem seis graus de liberdade de rotação, onde os três últimos eixos intersectam-se. Para tais robôs uma forma fechada de solução do modelo cinemático direto é disponível ou facilmente implementada. Entretanto, para o modelo cinemático inverso, algoritmos numéricos são normalmente empregados, que são de rápida solução, permitindo ainda a solução de redundâncias onde não são possíveis soluções de formas fechadas.

Sistemas de simulação estão sendo disponibilizados com a possibilidade de gerar mudanças para suportar diferentes tipos de robôs, geralmente com diferentes graus de liberdade. Por isso tais sistemas devem providenciar um método genérico para resolução do problema cinemático

direto e inverso, desde que, os robôs sejam projetados pelos sistemas de simulação e suas estruturas cinemáticas não sejam conhecidas de forma profunda pelo usuário.

O conhecimento desses métodos e ferramentas de programação off-line é de fundamental importância aos usuários de robô facilitando e permitindo a obtenção de posições e orientações precisas, melhorando assim, a fidelidade do modelo.

Atualmente percebe-se uma necessidade latente, principalmente em relação às montadoras automobilísticas da utilização de soluções por intermédio de simulação e programação off-line e visualização da posição e orientação da ferramenta terminal de um robô, em função de diversos ganhos que podem ser agregados nos processos e a crescente utilização de robôs industriais.

3.4.1 Modelagem cinemática direta

Ao explicitar a condição de cinemática direta, serão feitas as condições da transformação homogênea de um sistema robô com n juntas, definindo a posição e orientação do TCP, conforme referenciado na figura 3.1:

$$T(q) = \begin{bmatrix} \bar{n}(q) & \bar{o}(q) & \bar{a}(q) & \bar{p}(q) \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^0_1D(q_1) \cdot {}^1_2D(q_2) \cdot \dots \cdot {}^{n-2}_{n-1}D(q_{n-1}) \cdot {}^{n-1}_nD(q_n) \quad (3.5)$$

Os vetores \mathbf{n} , \mathbf{o} , \mathbf{a} , são ortonormais e descrevem a orientação do TCP no sistema de referência da base (mundo). O vetor \mathbf{p} descreve a translação do TCP do sistema de referência da base a sua posição atual. O reconhecimento da coordenada do robô em relação ao seu referencial zero possui uma solução única. \mathbf{T} descreve uma transformação homogênea 4×4 e é calculado como o produto de ${}^{i-1}_iD(q_i)$, conforme a equação 3.5.

Ele descreve uma transformação homogênea para cada link i até o seu anterior link $i-1$ e são somente dependente do valor da junta q_i deste link. São chamados parâmetros de Denavit

Hartenberg (“D-H”), usados para descrever uma transformação de coordenadas de um referencial associado a uma determinada junta até a consecutiva junta. Estas transformações são chamadas matrizes de transformação de coordenadas.

3.4.2 Modelagem cinemática inversa

Para resolver a cinemática inversa à equação

$$\begin{bmatrix} \bar{n} & \bar{o} & \bar{a} & \bar{p} \\ 0 & 0 & 0 & Z \end{bmatrix} = T(\mathbf{q}) \quad (3.6)$$

tem que ser resolvido por \mathbf{q} : dando $T(\mathbf{q})$ como 16 valores numéricos (quatro dos quais são triviais),

Resolvendo a equação (3.5) para os n ângulos da junta q_1, \dots, q_n entre as 9 equações surgidas a partir da matriz de rotação partindo de T , somente três equações são independentes. Essas somas com 3 equações a partir da posição do vetor parte de T dão 6 equações com n desconhecidos. Estas são equações *não – lineares*, equações transcendentais que podem ser de difícil solução. Métodos diferenciais podem ser usados dentro de tais casos: o jacobiano de um robô especifica um reconhecimento a partir das velocidades internas das juntas espaciais até a velocidade no espaço cartesiano.

$$dx = J(\mathbf{q}) d\mathbf{q} \quad (3.7)$$

A natureza deste reconhecimento muda com os valores das juntas de vários robôs. Certos pontos, chamados de singularidades, este reconhecimento não é irreversível. A idéia básica para resolver a cinemática inversa é até interativamente determinar o erro cartesiano entre posição cartesiana desejada e a atual (incluindo a orientação) pelo significado de cinemática direta e até interpretar este erro como uma velocidade cartesiana.

A equação (3.7) é resolvida pela dq pela pseudo-inversão da matriz jacobiana para determinar a correspondência da velocidade das juntas. Isto é repetido de forma que a posição e orientação cartesiana sejam bastante precisa (Anton, 2000).

O passo crucial é o cálculo da pseudo-inversa da matriz jacobiana. Os métodos de solução do problema da inversão do modelo geométrico são geralmente analíticos e numéricos iterativos. Os analíticos conduzem à obtenção de todas as soluções, estes métodos não são triviais e, além disso, não há garantia de que seja possível fazê-la. Normalmente utiliza-se em robôs que possuem um grande número de parâmetros de “D-H” nulos.

Os métodos numéricos iterativos convergem para uma solução possível, são de caráter geral e, com o atual desenvolvimento dos microcomputadores, a utilização destes métodos em tempo real é viável (Romano et al., 2002). Isto pode ser implementado, por exemplo, através do método numérico recursivo que usa o modelo cinemático inverso e a matriz jacobiana, utilizando o método de eliminação de Gauss (Dorf et al., 1972).

3.4.3 Modelagem dinâmica

Os robôs representam um complicado sistema dinâmico, uma das possíveis soluções para a modelagem dinâmica, vêm pela utilização de equações dinâmicas obtidas da mecânica Lagrangiana (Paul, 1981), que é definida pela diferença entre a energia cinética e potencial do sistema, conforme equação 3.8.

$$L = K - P \quad (3.8)$$

Onde, $K = \frac{1}{2} m_i v_i^2$ e $P = - m_i g d_i \cos T$

A energia cinemática e potencial podem ser expressa em qualquer sistema conveniente de coordenada que simplificará o problema. A equação dinâmica, nos termos das coordenadas usadas para expressar a energia cinemática e potencial, são obtidas como,

$$F_i = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} \quad (3.9)$$

Onde q_i são as coordenadas na qual a energia cinemática e potencial são expressas, \dot{q}_i é a velocidade correspondente e F_i a força correspondente ou torque, dependendo sobre quando q_i é uma coordenada linear ou angular. Estas forças, torques e coordenadas são chamadas como força, torque e coordenadas generalizadas.

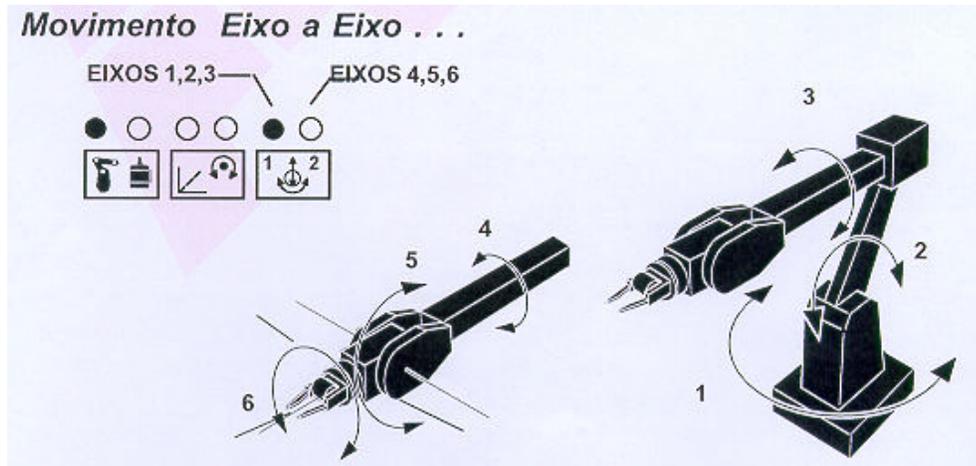
As equações dinâmicas relatam forças e torques para posições, velocidades e acelerações elas são usualmente utilizadas para obter a equação de movimento do robô. Isto é, ao fornecer forças e torques como entrada, essas equações especificam resultados de movimentação do sistema. Em alguns casos, os movimentos são fornecidos e deseja-se conhecer as forças e torques que devem ser aplicadas dentro de uma ordem para controlá-los (Paul, 1981). As equações dinâmicas fornecem a obtenção em cada link de movimento alguns coeficientes, um é conhecido como inércia na junta, outro a inércia de acoplamento. Outro termo que será obtido é a ação da força centrípeta, devido à velocidade nas juntas. Uma outra combinação a ser obtida é conhecida como a ação da força de coriolis e por fim um termo representa a força da gravidade nas juntas. Neste trabalho não iremos ater a simulação dinâmica e sim a geração de movimentos que serão enviados ao controle pelo robô.

3.5 A programação de robôs via teach - pendant

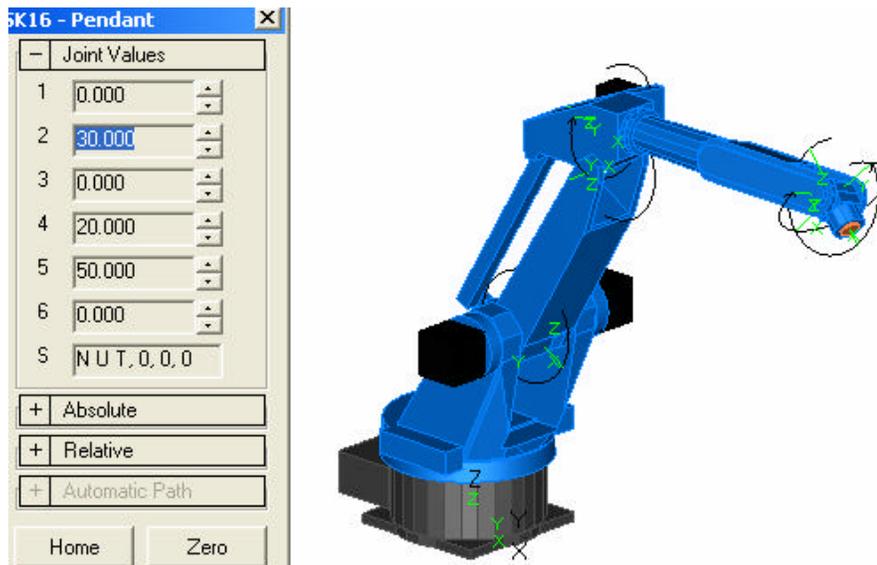
O método de programação mais frequentemente utilizado em robôs industrial é a aprendizagem ponto-a-ponto onde as juntas de um robô são movimentadas individualmente até as posições desejadas, através de um processo de aprendizagem de tarefas. Normalmente, esta programação pode ser realizada através da movimentação angular das juntas, movimentação cartesiana da ferramenta e reorientação da ferramenta, conforme é apresentado a seguir.

3.5.1 Movimentação angular das juntas

Neste método são gravados pontos de referência fornecidos pelos transdutores de posição localizados em cada junta e a partir desses pontos são geradas trajetórias (angulares) através da utilização de algoritmos de interpolação e filtragem. A simplicidade deste método não exige o conhecimento do modelo geométrico do robô (figura 3.4).



a) Através do Teach-in box de programação do robô.



b) Através de software de Programação off-line.

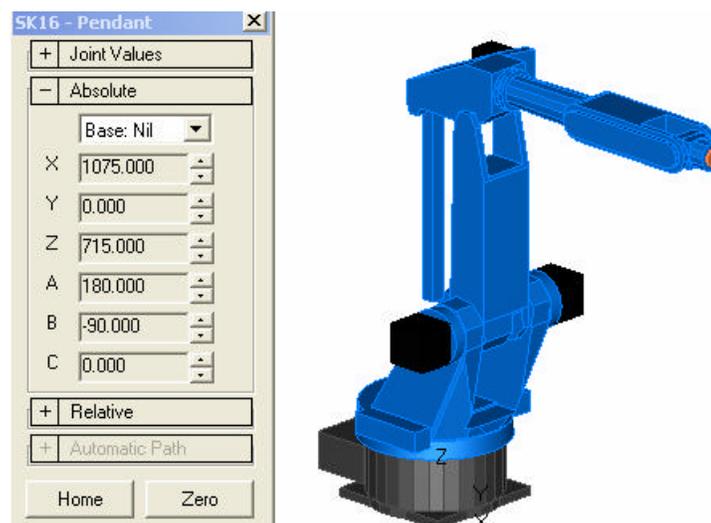
Figura 3.4 Movimento angular das juntas.

3.5.2 Movimento na direção cartesiana

Idêntico ao procedimento descrito anteriormente, mas a obtenção dos pontos de referência é realizada utilizando o modelo cinemático do manipulador, possibilitando assim a operação do robô seguindo as direções X, Y e Z (figura 3.5).



a) Através do Teach-in box de programação do robô.

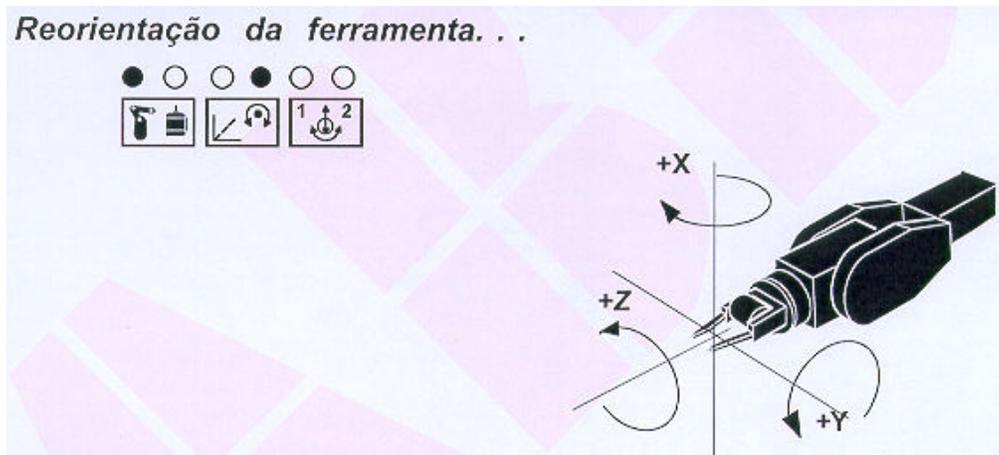


(b) Através de software de Programação off-line.

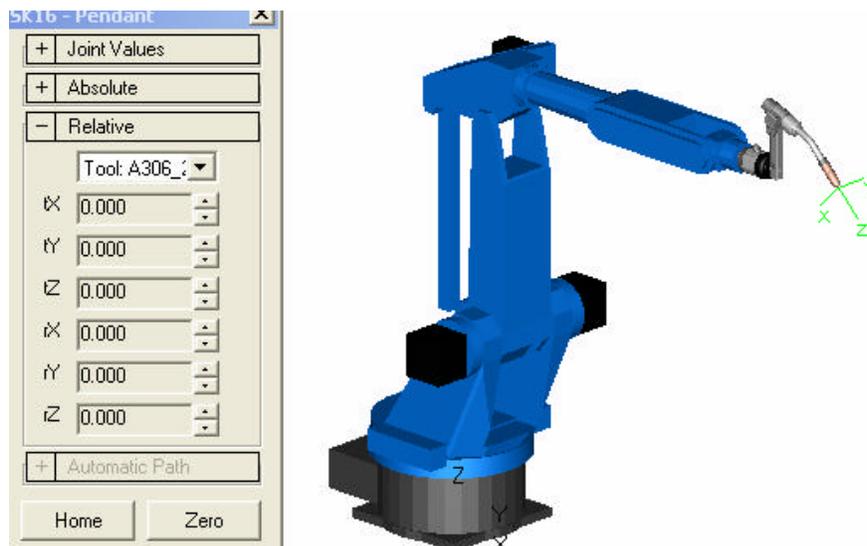
Figura 3.5 Movimento em Coordenadas Retangulares.

3.5.3 Movimento de reorientação da ferramenta

Idêntico ao procedimento descrito anteriormente, mas a obtenção dos pontos de referência é realizada utilizando a orientação da ferramenta em torno da rotação dos eixos X, Y, Z, correspondendo as três rotações Roll, Pitch, Yaw (figura 3.6). A ferramenta terminal de um robô é designada de TCP (Tool Center Point) ou ponto central da ferramenta.



a) Através do Teach-in box de programação do robô.



b) Através de software de programação off-line.

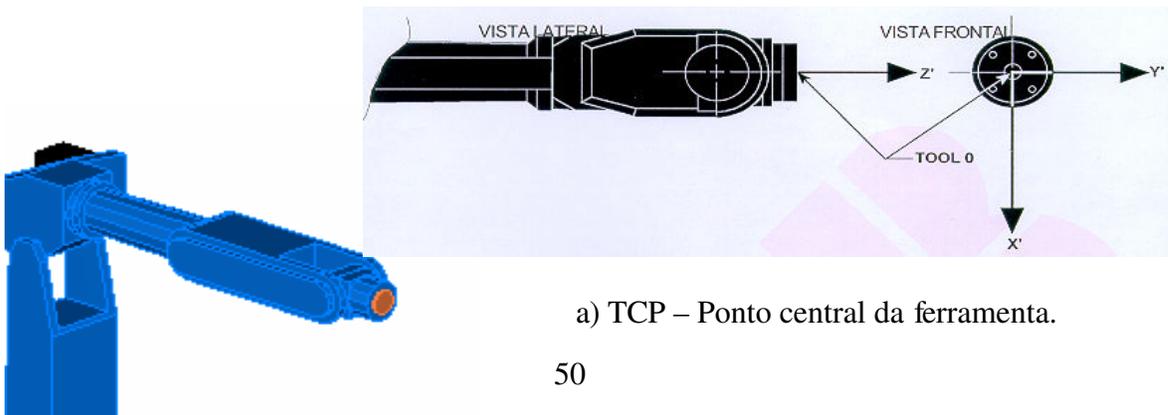
Figura 3.6 Re-orientação da ferramenta.

3.6 Calibração da ferramenta

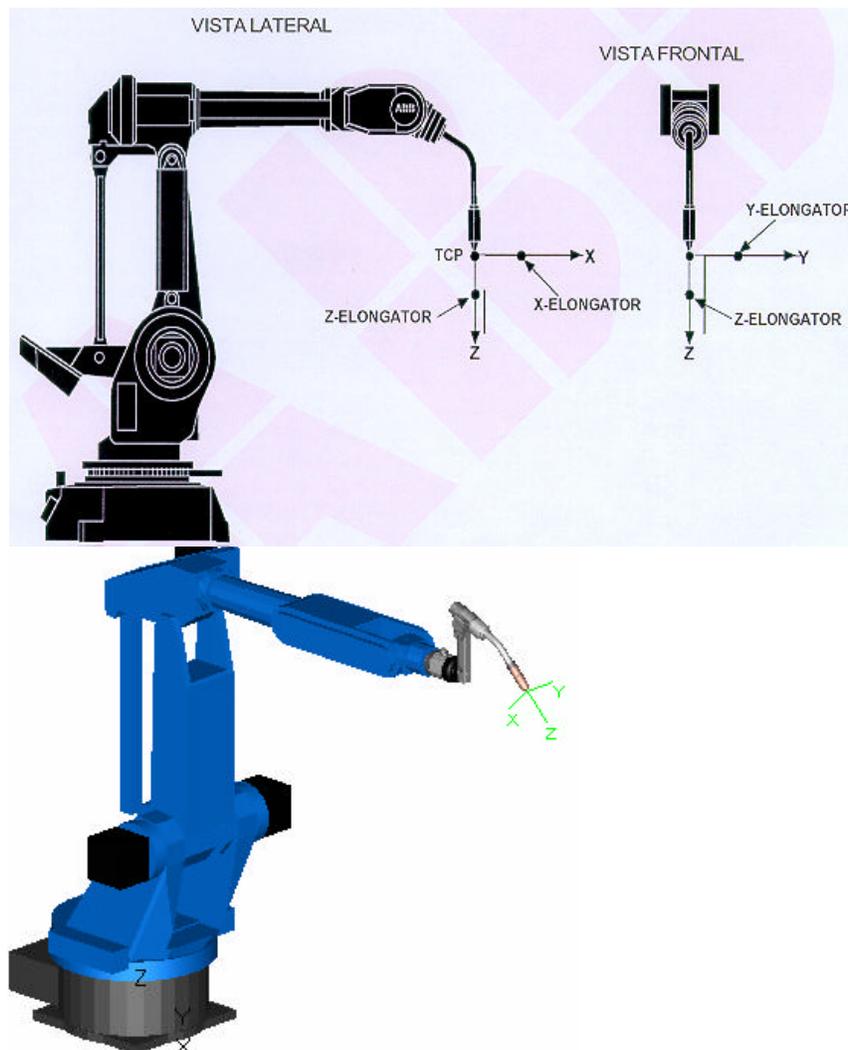
Os modos descritos anteriormente para movimentação dos robôs são associados com a melhor forma de aprendizagem, função principalmente de uma boa especificação funcional. Isto permitirá a realização de tarefas mais precisas e complexas sem a necessidade da utilização de posições determinadas analiticamente, ou informações provenientes de sensores de percepção externos; embora implique numa necessidade do aumento do tempo útil de trabalho, tendo em vista, que durante a programação das tarefas pelo método de aprendizagem, o manipulador obrigatoriamente é utilizado.

Existe a necessidade de realizar alguns procedimentos ao iniciar o trabalho com um robô industrial, dentre eles devemos considerar, para um perfeito funcionamento do sistema e melhor performance do robô, dois procedimentos básicos:

- i. **Sincronização das juntas:** os contadores das informações provenientes dos encoders (ou resolvers no caso de interrupção de energia ou desconexão do robô) deverão ser zerados, com o robô posicionado nas marcas de calibração;
- ii. **Calibração de sua ferramenta terminal TCP:** nesta etapa deve inicialmente informar ao sistema de controle do robô (painel de comando), as características principais da ferramenta (massa, dimensão, etc.), para que o sistema de controle possa realizar todo o procedimento de calibração da ferramenta terminal do robô (figura 3.7).



a) TCP – Ponto central da ferramenta.



b) Ferramenta Terminal.

Figura 3.7 Procedimento de calibração da ferramenta terminal.

3.7 Linguagem de programação de robôs

Com o atual avanço na tecnologia de desenvolvimento de hardware e software, a programação de robôs através de linguagens, está cada vez mais próxima da realidade industrial. Estes avanços incluem uma maior sofisticação dos controladores, melhor precisão de posicionamento e incremento no número e tipos de sensores.

As programações quando utilizam linguagens têm como principal objetivo à implementação de programas sem a necessidade do uso do robô propriamente dita, mas através de um ambiente computacional de programação num computador.

As linguagens de programação de robôs são normalmente interativas, permitindo a implementação de programas estruturados e fácil interatividade com o usuário. Podem ser classificadas em níveis diferentes:

- i. **Nível de junta:** as linguagens classificadas neste nível requerem a programação individual de cada junta do robô para que uma dada posição seja alcançada;
- ii. **Nível de manipulador:** neste nível é necessário, apenas fornecer a posição e a orientação da ferramenta terminal e o sistema se encarrega de obter, através do modelo geométrico inverso do robô, as posições de cada junta;
- iii. **Nível de objeto:** neste nível são necessários apenas às especificações relativas ao posicionamento de objetos no interior do volume de trabalho do robô. Deste modo, é necessária a existência de um modelo matemático que represente o ambiente de trabalho no qual o robô se encontra;
- iv. **Nível de objetivo:** neste nível a tarefa não é realmente descrita, mas definida como, por exemplo, montar as peças A, B e C. Neste caso é necessário, além do conhecimento do modelo do ambiente, um conjunto de dados relativos a uma determinada tarefa.

3.8 Procedimentos básicos para implementação e execução de programas

Para melhorar a performance de um programa e diminuir o seu tempo de implementação e validação, antes de iniciar a implementação de um programa num robô industrial, o usuário deverá tomar alguns procedimentos básicos como:

- a) Conhecer o processo a ser automatizado;
- b) Conhecer e ou definir as variáveis de controle;
- c) Saber e ou implementar a seqüência de lógica do processo;

- d) Visar segurança, definir nomes, rotinas, I/O;
- e) Verificar o sincronismo das juntas, certificar-se do controle eficaz da unidade de controle em relação ao robô;
- f) Criar, definir e ativar o TCP da ferramenta a ser utilizada no processo.

As linguagens de programação variam de acordo com o fabricante de robô. Exemplificar a linguagem própria de robôs industriais disponíveis no mercado ajuda no entendimento para uma estruturação de programa. Segue no Anexo I - Principais Comandos e Exemplos de Programação de Robôs Utilizando Linguagem Textual, alguns exemplos de robôs disponíveis no mercado nacional.

3.9 Vantagens da programação off-line

Uma das principais vantagens da programação off-line é a não necessidade de ter disponíveis os equipamentos para desenvolvimento do programa, necessitando numa fase inicial, apenas do desenho da planta e especificações funcionais e tecnológicas dos elementos da célula e posteriormente da geração de movimentos do manipulador e posteriormente com a lógica de programação e cálculos.

Isto possibilitará num planejamento de tarefas, ambientes e ferramentas, não necessitando a parada dos equipamentos durante a implantação de novos programas em células existentes, e no caso de implementação de novas células. Toda a programação e estudo de viabilidade poderão ser realizados sem investimento nestes equipamentos a partir da alteração de documentação inicialmente estabelecida, considerando inclusive tempos de execução, permitindo assim a orientação de clientes para a tomada de decisões em cima de uma solução robotizada.

O processo de programação off-line permite inclusive uma abordagem mais apurada na estruturação dos programas, sendo extremamente necessária uma participação efetiva do programador na estruturação da lógica a ser definida.

3.10 Programação através de visualização gráfica

A utilização de técnicas de Desenho Auxiliado por Computador (CAD), tem encontrado extensivo uso na recolocação e melhora de processos de desenhos de engenharia, desenhos de arquitetura, e muitas outras aplicações. Contudo, um processo de engenharia envolvendo partes em movimento somente pode ser totalmente entendido, através do processo de simulação.

Técnicas iniciais foram envolvidas utilizando um Desenho Auxiliado por Computador (CAD) de uma parte mecânica e criando uma simulação animada de movimentos que as máquinas de usinagem “CNC” devem seguir para criar a peça a partir de uma linha de blocos. Também melhorando a visualização de processos, um arquivo pode ser criado contendo as instruções requeridas para as máquinas CNC. O arquivo pode então ser executado para criar a peça.

Essa extensão do CAD para o CAM aparentemente é desejável para robôs industriais. Contudo, a cinemática envolvida nos movimentos dos robôs é consideravelmente mais complexa do que o relacionado com as máquinas de usinagem CNC, e a relação entre as **varreduras das curvas** pela ferramenta de final de braço do robô e as variáveis das juntas não é uma cinemática direta. Adicionalmente, para uma simulação de robô ser de uso geral, deve ser capaz de simular uma grande variedade de tipos de robôs e configurações.

3.11 Exemplo de aplicação

Um exemplo simulado graficamente e programado off-line, conforme figura 3.8, possibilita um trabalho de estruturação da linguagem trabalhada e uma explanação de um ambiente completo de programação, (figura 3.9), de um robô industrial que será utilizado nos próximos capítulos.

Para exemplificar a diferenciação de um programa off-line e um programa off-line estruturado será utilizado como exemplo uma situação em que um robô execute um processo de soldagem em uma peça de trabalho que tem como característica quatro posições simétricas a serem soldadas, ilustrado na figura 3.8.



Figura 3.8 Exemplo de uma célula de trabalho com solda robotizada

A linguagem a ser implementada neste exemplo é a KAREL 2 , linguagem de algumas unidades de controle dos robôs industrial FANUC. O robô utilizado na simulação é um modelo SK 16 do fabricante Yaskawa/Motoman, que reforça a idéia da flexibilidade na programação off-line, onde a linguagem independe do robô a ser simulado.

Na figura 3.9 pode ser observada a representação de um programa escrito, com a sintaxe da linguagem KAREL 2;

```
PROGRAM  
-- ! LANGUAGE KAREL 2  
-- ! MEMORY 8192  
-- ! ROBOT SK16  
-- TEACHPOINT DECLARATIONS  
VAR  
  TP1 : POSITION  
  TP2 : POSITION  
  TP3 : POSITION  
  TP4 : POSITION  
  TP5 : POSITION  
  TP6 : POSITION  
BEGIN  
  $UTOOL=POS(-11.5265,2.7334,307.2149,0,45,-160,")  
  $USEMAXACCEL=TRUE  
  %INCLUDE TEACH POINTS#  
  ...  
END PROGRAM
```

Figura 3.9 Ilustrando um programa escrito em linguagem Karel 2 pertencente a alguns controladores FANUC.

Nos quatro cordões de solda a serem realizados no processo os trajetos são os mesmos, serão representados pelos pontos ensinados TP1; TP2; TP3; TP4; TP5 e TP6, pontos que foram ensinados no robô.

A figura 3.10 ilustra as coordenadas de posição e orientação dos pontos ensinados, virtualmente pelo robô.

```
-- WORKSPACE teachpoint file
TP1 = POS(847.2979,226.6498,-171.5,152.7637,41.6411,161.1183,'LUFB')
TP2 = POS(800,0,-263.5,-148.5703,17.462,169.8249,'LUFB')
TP3 = POS(1000,10,-263,-148.5703,17.462,169.8249,'LUFB')
TP4 = POS(1199.95,10,-263,-148.5703,17.462,169.8249,'LUFB')
TP5 = POS(1200.1093,10,-103.3819,-171.2326,22.7317,168.6665,'LUFB')
TP6 = POS(914.8239,-84.3288,18.3028,152.7637,41.6411,161.1183,'LUFB')
```

Figura 3.10 Arquivo dos seis pontos ensinados (TP), que possui as coordenadas de posição e orientação.

É necessário rotacionar o dispositivo em uma posição desejada e também controlada, na implementação real é necessário trabalhar com sinais de I/O, para enviar a necessidade de entendimento da mudança de posição no momento desejado.

Apesar de ter sido realizada uma programação off-line, não houve uma preocupação em estruturar o programa, quando se utilizam os recursos de estruturação que a linguagem dispõe. O processo de estruturação pode ser realizado textualmente, já que os pontos e orientações necessários ao processo, já foram definidos através da programação ponto a ponto virtual, ao basear-se nos modelos gráficos.

A linguagem, utilizada, dispõe de comandos que permitem de forma distintas o mesmo resultado, já que o processo programado de soldagem utiliza o mesmo bloco de movimentos do robô, alterando somente a posição de soldagem através do dispositivo de posicionamento, uma instrução de “looping“, atende de forma satisfatória a repetição do movimento do robô em um numero conhecido de quatro repetições. Ao implementar as mudanças propostas existirão

algumas variantes de programação interessantes de serem analisadas, conforme descrito no Anexo II - Aplicação de Programação “Off-Line”, com Estruturação.

São exemplificados e ilustrados um “ambiente completo de programação”, nas figuras 3.11 e 3.12 respectivamente, célula robotizada para soldagem “GMAW”, com dois robôs e um dispositivo para alimentação e saída de uma peça, após o processo de soldagem, com a possibilidade de troca de ferramenta no final de braço do robô de manuseio.

Este exemplo permite a utilização do trabalho em cooperação¹, a unidade de controle do robô controla de forma independente 13 graus de liberdade no processo de soldagem, com possibilidade de extensão de controle de até 21 graus de liberdade. Com diversas abordagens de programação e comunicação entre robôs e dispositivo.

A complexidade da integração entre robôs e dispositivo, enriquece o estudo e a possibilidade de realizar mudanças em todos os níveis de integração bem com no planejamento de novas tarefas. É possível ainda objetivar novas possibilidades no trato da programação, ao desenvolver situações e arranjos limitado ao equipamento e layout preconizado no ambiente disponível.

A arquitetura definida na célula é proprietária, “fechada”, definida pelo integrador e fabricante, sendo assim, é possível alterar a lógica de trabalho ao implementar novas tarefas, o que torna o sistema mais flexível.

¹ cooperação, processo de trabalho entre robôs de forma que possibilite a sincronia de movimentos em tarefas simultâneas ou não. Na cooperação, a unidade de controle, controla todos os robôs de forma multiplexada, através de algoritmos implementados nos modelos cinemáticos, possibilitando inclusive controlar eixos externos, dispensando a necessidade de integração de comunicação por porta serial via I/O's.

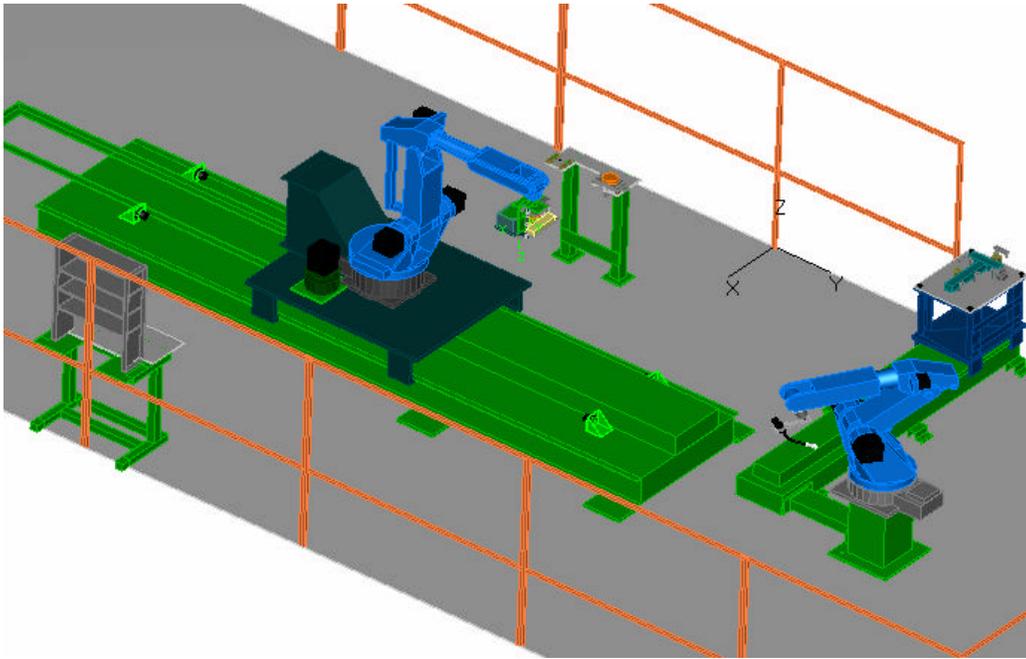


Figura 3.11 Célula de Manufatura Automatizada Robotizada

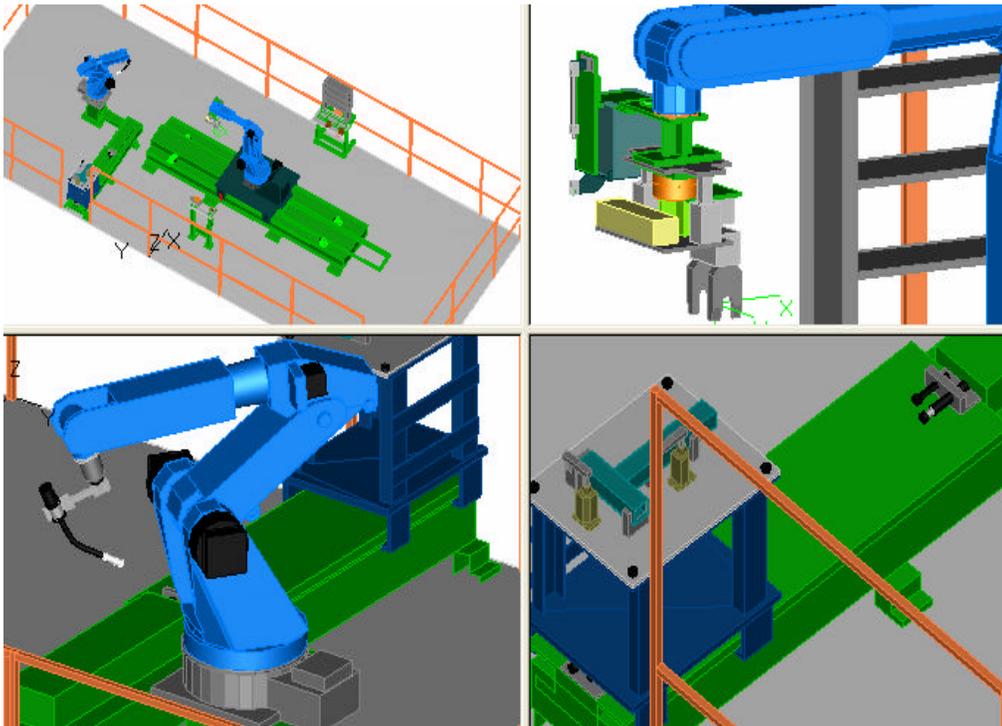


Figura 3.12 Detalhes da Célula de Manufatura Automatizada Robotizada

3.12 Conclusão

Neste capítulo foram apresentados conceitos básicos de Programação Off-line de Robôs, através da apresentação dos requisitos básicos necessários para a programação estruturada e simulação, conceitos básicos de modelagem cinemática, enfatizando o problema de orientação da ferramenta, e sistemas de coordenadas e procedimentos de inicialização e calibração da ferramenta terminal. Direcionamento do enfoque para a concepção de sistemas automatizados dentro da engenharia, baseada em modelos virtuais, propiciando análises e decisões mais próximas do real. Neste capítulo foi também apresentado um ambiente completo de programação off-line de um robô industrial que será utilizado nos próximos capítulos.

Um dos questionamentos em relação aos pacotes de softwares de simulação e programação off-line está na questão da confiabilidade do processo, na perfeita simulação e programação off-line, ou seja: - “Uma boa programação e simulação garantem a execução correta do robô real?” Esta pergunta deve ser respondida de várias maneiras, função principalmente do grau de precisão e exatidão do processo.

Existem distorções de trajetórias, ou seja, de comportamento de movimento quando um robô desloca-se de um ponto conhecido geometricamente a outro. Os pontos programados são conhecidos, porque possuem uma referência inicial e precisam possuir a mesma referência entre modelo virtual e modelo real. Dessa forma geometricamente não haverá distorção, porém em relação à trajetória, o comportamento de movimento da ferramenta de final de braço do robô, pode haver um erro de desvio de centímetros em relação da posição e acima de 30 % do tempo de ciclo, conforme citado por (Anton, S. et al, 2003), através de comparação realizada entre uma trajetória programada através da unidade de controle de um robô (controle real) e uma trajetória programada através de um software de simulação e programação off-line (controle simulado).

Este erro de desvio é facilmente compreendido, já que o modelo cinemático inverso da unidade de controle do robô e os algoritmos de controle da trajetória são métodos numéricos

proprietários e os modelos utilizados pertencem ao fabricante do software “universal”. O que é visualizado na tela do computador, graficamente, não exprime a trajetória real de robô industrial, lembrando que a posição dos pontos ensinados são os mesmos. Isto, se o modelo gráfico está calibrado de acordo com o modelo real. Normalmente este tipo de erro de desvio já é minimizado e acontece em torno de milímetros e abaixo de 3% para o tempo de ciclo.

Para conseguir essa redução de erro de desvio de trajeto na programação off-line é necessário utilizar um modelo específico, usando o modelo do software “proprietário” ou usando uma interface com modelo conhecido como RRS (Realistic Robot Simulation – versão atual 1.3 – dezembro de 2002) interface esta desenvolvida, pela necessidade dos usuários de robótica com a finalidade de reduzir os desvios de imprecisão em relação às posições e ao tempo de ciclo.

O desenvolvimento dessa interface deu-se em parceria, com algumas companhias do setor automobilístico, tais como Audi, BMW, Ford, Mercedes-Benz, Opel, Peugeot, Renault, Volvo e VW. Esse grupo foi gerenciado pelo instituto de pesquisa Fraunhofer-IPK, Berlim, instituição acadêmica de pesquisa que desenvolveu a integração e ficou responsável pelos testes e melhorias. Os fabricantes de robôs e alguns fornecedores de controles tais como ABB, COMAU, Fanuc, Kuka, Siemens, Renault Automation e VW, e fabricantes de softwares de simulação e programação off-line, tais como Dassault Systèmes, Deneb, Silma e Tecnomatix implementaram uma cooperação usando essa nova interface.

Dentro dos robôs, essas interfaces com os módulos de simulação começaram a ser implementados e comercializados pelos principais fabricantes de robôs industriais a partir de 1994. Esses módulos são específicos para cada modelo e marca de robô e podem ser adquiridos separadamente pelos usuários de processos de simulação e programação off-line, permitindo assim a utilização dos pacotes “abertos / universais”, utilizando o mesmo modelo cinemático inverso dos fabricantes e seus algoritmos de controle, minimizado os erros de trajetória.

Entretanto, a trajetória exata de um robô dependerá também do hardware de controle do robô, não sendo possível nem necessário que a programação off-line contemple o modelo

dinâmico, e suas reações e características do controlador utilizado, cabendo a esses pacotes computacionais apenas a preocupação com a evolução das trajetórias e movimentos associados, considerando que o controlador deva assegurar a precisão e repetibilidade das mesmas.

Capítulo 4

Programação Avançada Direcionada a Aplicações Industriais

Neste capítulo são apresentados através de exemplos práticos aspectos relativos à validação e testes dos conceitos apresentados nos capítulos anteriores, sendo enfatizados dois exemplos de aplicações: uma Célula Automatizada implementada na Escola SENAI e Faculdade SENAI de Tecnologia Mecatrônica – SP, “Armando de Arruda Pereira”, e um modelo de uma Célula Virtual a ser implementada no Laboratório de Automação Integrada e Robótica da Faculdade de Engenharia Mecânica da UNICAMP.

Através desses dois exemplos, foram enfatizados aspectos concernentes aos processos de programação avançada de robôs, onde existe a possibilidade real de ganho no processo, enfocando uma utilização racional nos processos de programação e que venha a agregar de forma significativa a concepção de programação.

Neste capítulo ainda é apresentado um estudo relativo à calibração de robôs, a partir de obtenção de parâmetros cinemáticos durante a simulação, onde a partir desse estudo de casos será possível a realização de uma análise da aplicabilidade enfocando a concepção e planejamento no uso da simulação e programação Off-Line, dificuldades e facilidades, ganhos reais diretos e indiretos.

4.1 Descrição de Software de Programação Off-Line de Robôs

A metodologia utilizada para desenvolvimento de uma programação avançada é à utilização dos conceitos apresentados anteriormente, e do uso de um software comercial “universal”, disponível no mercado, Workspace® 5.04, pertencente ao grupo Flow Software Technologies. O software é considerado “midle-end”¹, sua performace é satisfatória, justificando o seu custo / benefício em relação a outros softwares “high-end”, envolvendo os fatores básicos de programação, métodos de programação, integração de dispositivos automatizados e redes de comunicação - indústria virtual.

Foi possível a utilização do software Workspace®, constatando a tecnologia que implementa as abordagens cinemáticas com suas possibilidades e variantes, a utilização de métodos numéricos iterativos, possibilitando uma convergência possível de solução, através do método recursivo, utilizando os cálculos dos modelos cinemático direto e inverso através da resolução da matriz jacobiana.

Para realização de integração de dispositivos automatizados exige um alto nível de estruturação em automação utilizando programação Off-Line, utilizou-se uma estratégia de discretização das possíveis atividades em processos de concepção, implementação e projeto propiciando a utilização da ferramenta como uma metodologia no auxílio da engenharia.

¹ “Midle end” o conceito que será atribuído a esta expressão, é em função da restrição na característica atribuída à interação do processo, tal como ajuste de parâmetros de soldagem, como a amperagem, tensão e etc... Não é possível ser ajustado, através do software, de forma a ser implementado no processo real, é dada uma característica meramente funcional para enriquecer a visualização na simulação. O conceito atribuído a expressão “High End”, permite de forma real a interação dos ajustes dos parâmetros do processo, no software, que elimina ou diminui relativamente a necessidade de ajustes no processo real.

Para validação de resultados, são apresentadas duas soluções distintas, a primeira que considera um sistema de célula automatizada robotizada existente na Escola SENAI e Faculdade SENAI de Tecnologia Mecatrônica – SP, “Armando de Arruda Pereira”, onde a integração e arquitetura foram desenvolvidas pelo fabricante e integrador. A segunda, com a utilização da metodologia para implementação virtual de um processo no Laboratório de Automação Integrada e Robótica da FEM-UNICAMP.

4.2 Integração de Sistemas Automatizados

A maior parte de células automatizadas robotizadas existentes no mercado nacional se caracterizam como sistemas proprietários “fechado”, que necessite do desenvolvimento de metodologias direcionadas a concepção de sistemas de integração “abertos” entre os robôs e dispositivos periféricos o qual constituirá um novo sistema de célula robotizada.

Uma das características de um sistema proprietário “fechado” é a dificuldade de utilização do seu sistema pelo usuário, é permitido que se faça uso das tarefas projetadas e disponíveis.

Ao fazer uso de uma metodologia, para conceber e implantar novos projetos com a concepção de programação Off-Line de robôs torna-se necessário, o desenvolver de ferramentas dedicadas e dispositivos periféricos, para serem integradas no manipulador robótico, tais como o estudo da implementação dos mecanismos, onde se modelaram os dispositivos periféricos, tratados através do software Workspace®, onde as movimentações do mecanismo foram implementadas através da cinemática direta de um sistema com quatro juntas, como mostra a figura 4.1.

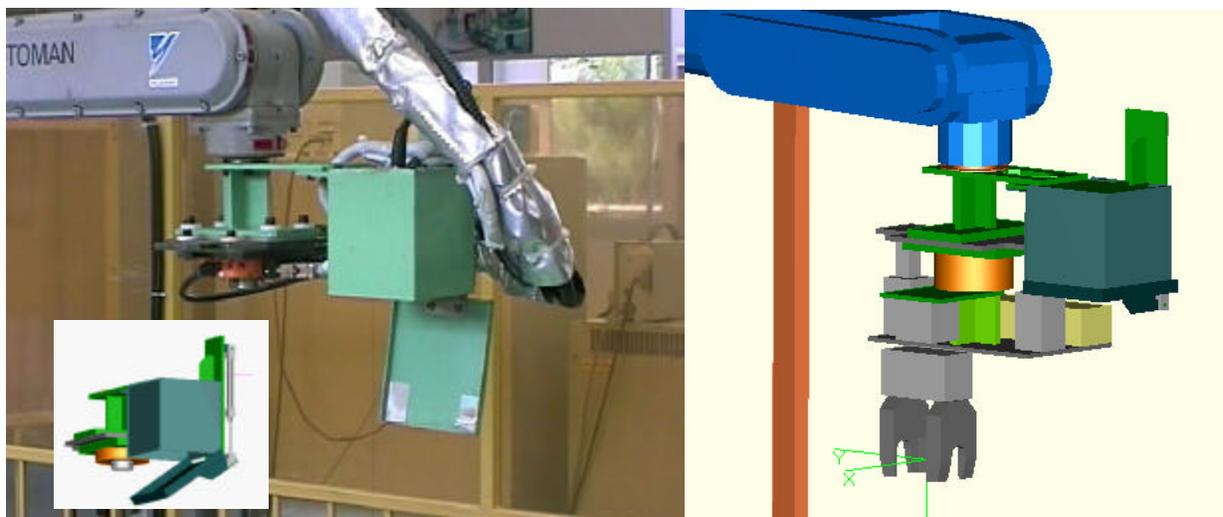


Figura 4.1 Dispositivo periférico, a considerar com quatro juntas.

Outro estudo possível foi à utilização da ferramenta para análise de dispositivo, fazendo a análise cinemática de operação do dispositivo periférico, para troca automática da ferramenta (ATC – Automatic tool change) de final de braço do robô, conforme ilustrada na figura 4.2, dispositivo para troca automática.

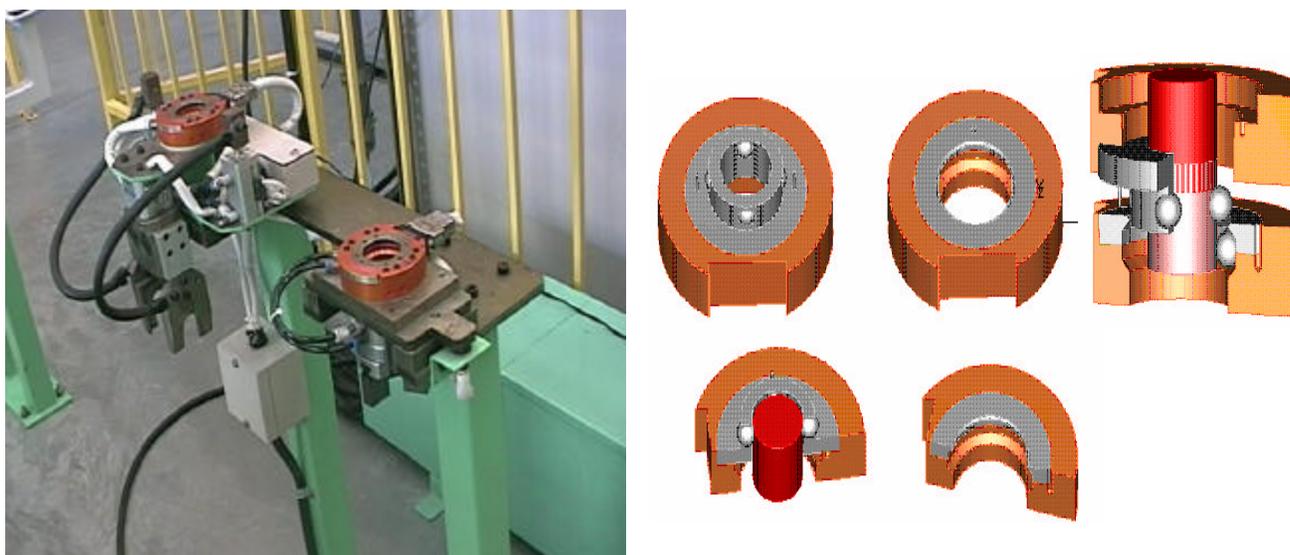


Figura 4.2 Dispositivo para troca automática da ferramenta (ATC).

4.3 Exemplo de implementação Utilizando Entradas-Saídas Digitais

Através da utilização da porta serial de comunicação I/O, é instituído um tratamento seqüencial das informações, que estabelece um protocolo de comunicação seqüencial entre o robô e dispositivos periféricos integrados na célula automatizada, implementado através do software Workspace®, foi possível a inclusão de metodologias de integração entre robô e dispositivo periférico, conforme é mostrado nos exemplos apresentado nas figuras 4.3 e 4.4, respectivamente.

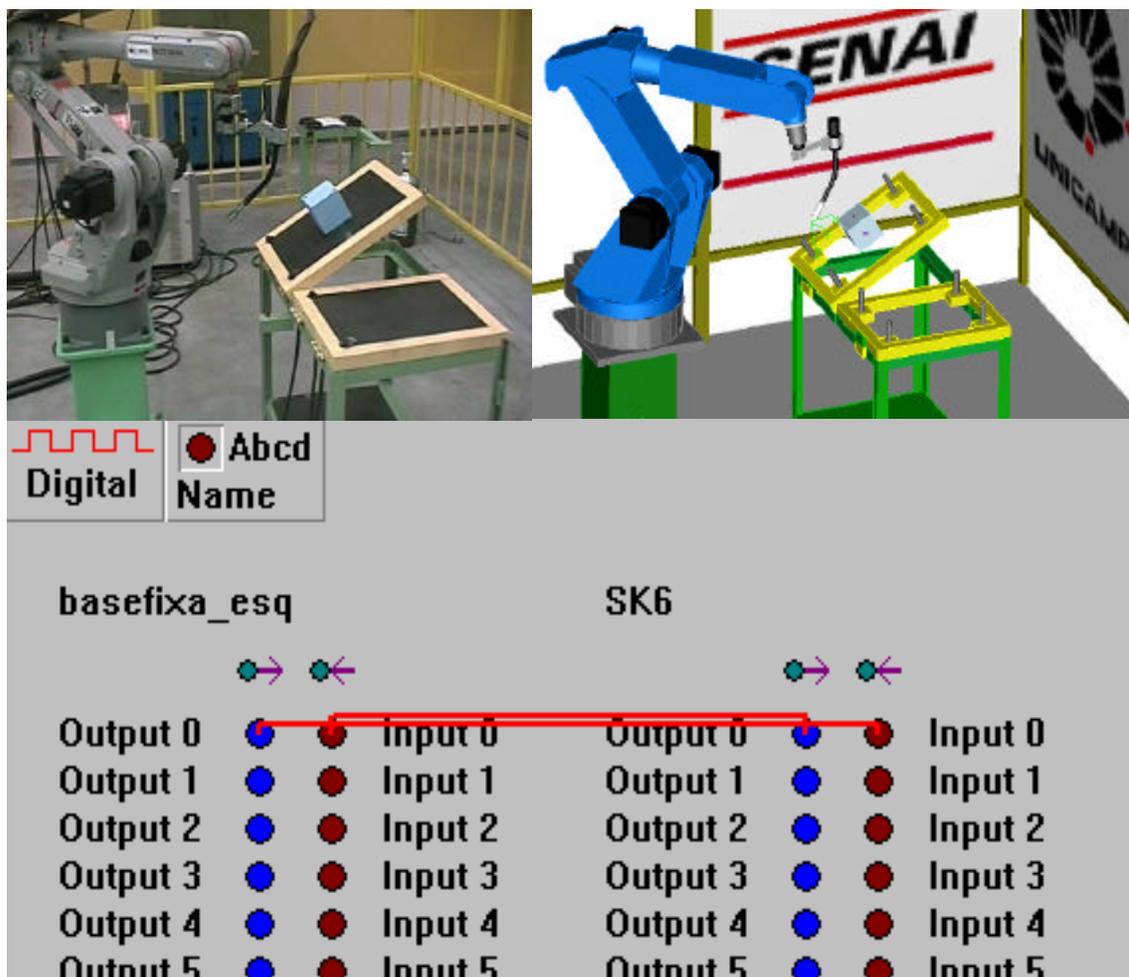


Figura 4.3 Exemplo de Célula, Integração Real -Virtual.
Integração de Robôs e dispositivos periféricos através de I/Os.

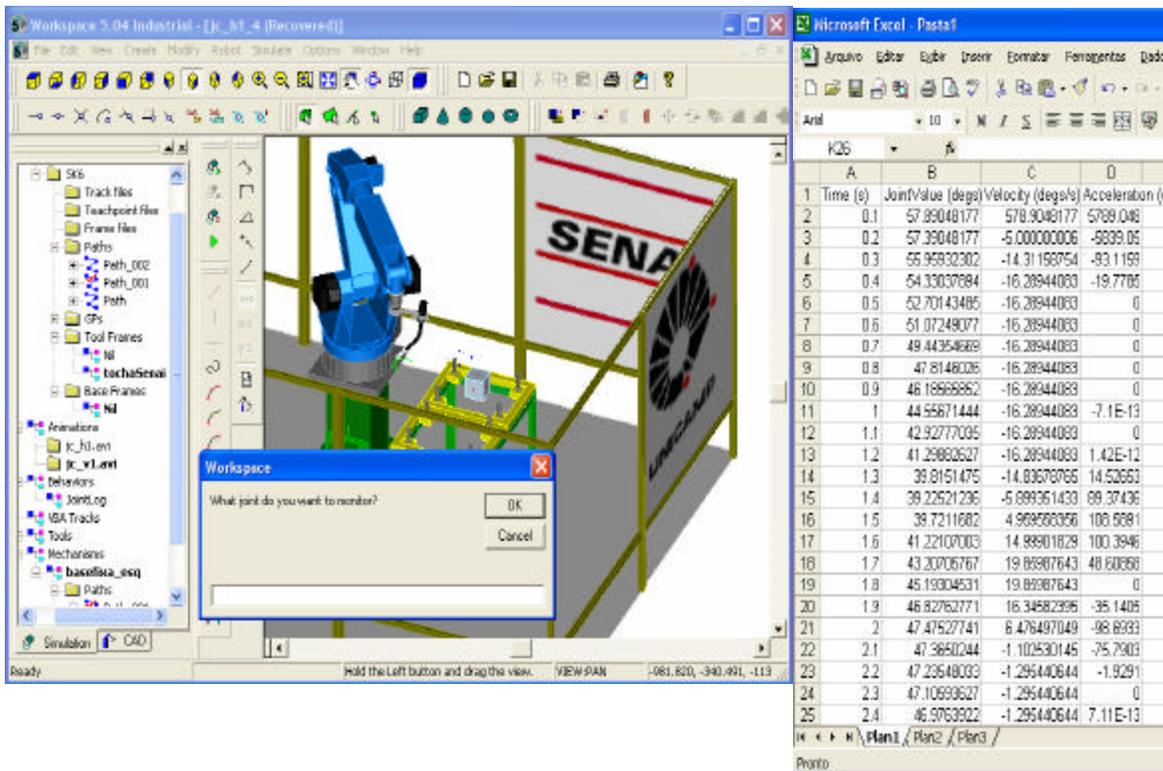


Figura 4.4 Integração de robôs e dispositivos periféricos através de I/Os. Tratamento das informações.

4.4 Integração de Robôs Industriais

A partir dos estudos apresentados anteriormente, torna-se possível à realização de uma análise da aplicabilidade que enfoque a concepção e planejamento no uso da simulação e programação Off-Line através de estudo de casos, o qual permite, uma análise crítica do problema, com ênfase na sua estruturação e torna possível à integração de dispositivos mecâtrônicos, a utilizar, de forma objetiva e racional os recursos disponíveis na sintaxe de programação de robôs, deixando ao operador poucas operações de aprendizagem.

O fato de possuir um ferramental para simular e programar um novo processo, não garante uma programação econômica, que atenua a gravação de pontos que caracterizam o processo, é preciso principalmente planejar o processo, definindo a melhor estratégia de aprendizagem,

garantindo menos pontos de aprendizagem, e conseqüentemente um menor tempo de programação do operador no processo de aprendizado de pontos.

Esta ferramenta possibilita uma visualização do ambiente no qual, está inserido o processo, podendo ser previsto uma maior eficiência de alguns movimentos que comprometam a tarefa. A estruturação do programa é realizada textualmente, reprogramando a sintaxe da linguagem no qual se deseja trabalhar, nesta fase é preciso conhecer os níveis de possibilidades que uma sintaxe disponibiliza somado a rotina de movimentos que se deseja implementar. Este conjunto de concepções leva um programa à condição de estruturado.

Para validação dos resultados apresentados anteriormente dois exemplos de aplicações foram implementados, que permitiram dar ênfase a aspectos concernentes aos processos de programação avançados, onde existe a possibilidade real de ganho no processo, e uma utilização racional nos processos de programação e que venha a agregar de forma significativa na concepção e integração de dispositivos mecatrônicos.

4.4.1 – Célula Integrada de Manufatura implementada no SENAI

Inicialmente foi implementada uma Célula Automatizada Robotizada na Escola SENAI e Faculdade SENAI de Tecnologia Mecatrônica – SP “Armando de Arruda Pereira”, conforme mostra a figura 4.5.

Esta célula é constituída de dois robôs industriais do fabricante Motoman / Yaskawa Modelos Sk6 e Sk16. A principal característica da unidade de controle desta célula é a concepção de uma única unidade de controle para os dois robôs, unidade está que pode controlar simultaneamente 21 GL “graus de liberdade” (deslocamentos dos eixos de forma dependente ou independente).



Figura 4.5 Célula Automatizada Robotizada, implementada na Escola SENAI e Faculdade SENAI de Tecnologia Mecatrônica – SP. “Armando de Arruda Pereira”.

Cada robô possui uma ferramenta de final de braço, o robô SK6 possui a capacidade de trabalhar com 6 Kgf e utiliza uma ferramenta de final de braço para processo de soldagem “GMAW”, uma tocha de soldagem. O segundo robô SK16 possui uma capacidade de trabalho de 16 Kgf contém um sistema de troca automática da ferramenta, “ATC - Automatic tool change”, que possibilita a utilização de duas ferramentas de final de braço.

É possível a utilização de duas garras, uma para atuar no processo de soldagem, a primeira segura e posiciona a peça a ser soldada e a segunda, também uma garra, porém com outra configuração, para manusear peças que utiliza o sensoriamento de um sistema de visão.

Existe um dispositivo periférico que tem como função o posicionar e orientar das peças. No início do processo, na primeira operação de união, o dispositivo executa um travamento nas peças, possibilita o trabalho de solda do robô SK6. Seqüencialmente, após a primeira operação

de solda, o dispositivo desloca-se, posiciona-se e destrava a peça para com o auxílio do segundo robô, SK16, pegá-la e posicioná-la na seqüência da operação de soldagem.

A célula automatizada e robotizada têm como característica a integração com um “CLP”, que gerencia os programas dos dois robôs e o sensoriamento dos periféricos. Essa concepção permite rodar dois programas, um em relação à unidade de controle do robô e outro do próprio “CLP” que gerencia a célula, utilizando uma lógica seqüencial de I/O.

Na figura 4.6, conforme segue, está a célula automatizada robotizada, modelada de forma virtual no software Workspace ®

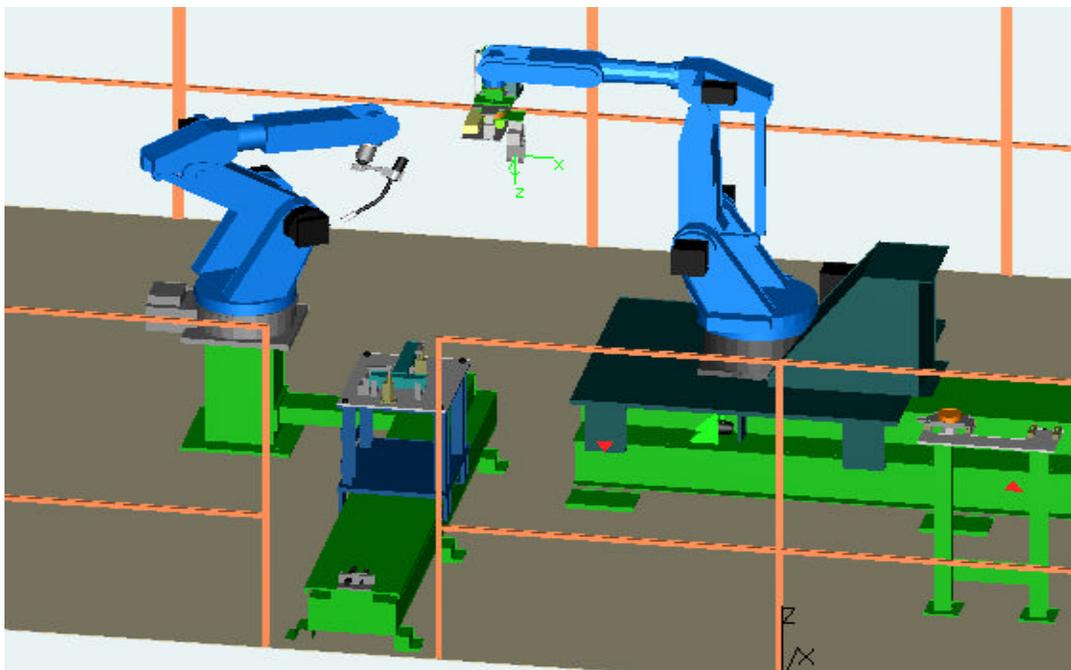


Figura 4.6 Célula Automatizada Robotizada, Virtual (Workspace®).

4.4.2 – Célula Integrada de Soldagem implementada na UNICAMP

Com a utilização desta metodologia torna-se possível fazer uma implementação virtual que atenda de forma satisfatória uma situação real em relação a uma implementação virtual de um processo. A concepção de uma programação avançada a priori é atenuada no momento em que se busca uma integração com uma cooperação entre robôs e periféricos, visto que o processo de trabalho entre os processamentos da unidade de controle acontece de forma multiplexada, permitindo um entendimento e execução gerenciada no mesmo instante de tempo.

Como uma alternativa para a solução ao integrar e implementar uma programação executa-se as instruções programadas por uma simples lógica de comunicação via porta serial I/O. Essa solução, não atende de forma geral a alguns tipos de processos, principalmente em atividades que se exigem respostas rápidas em relação aos movimentos dos braços robóticos, bem como na sincronia de movimentos.

No Laboratório de Automação Integrada e Robótica da FEM-UNICAMP foi possível constatar esta condição na programação virtual do sistema cooperativo, conforme ilustrado na figura 4.7 e detalhes nas figuras 4.8 e 4.9. Assim foi implementada uma célula integrada de soldagem, constituída de dois robôs ABB, o IRB1400 com capacidade de carga de 14 Kgf, utilizando como ferramenta de final de braço uma tocha de soldagem. IRB140 com capacidade de carga de 4 Kgf e um dispositivo mecatrônico com 3 GL, conforme figura 4.7.

Para a execução da programação Off-Line, consideraram-se os 3GL do dispositivo mecatrônico como eixos externos do robô IRB 1400. Esta possibilidade é viável no robô real, possibilitando os ajustes e integração de eixos externos, como dispositivos, a serem controlados pela unidade de controle do robô.

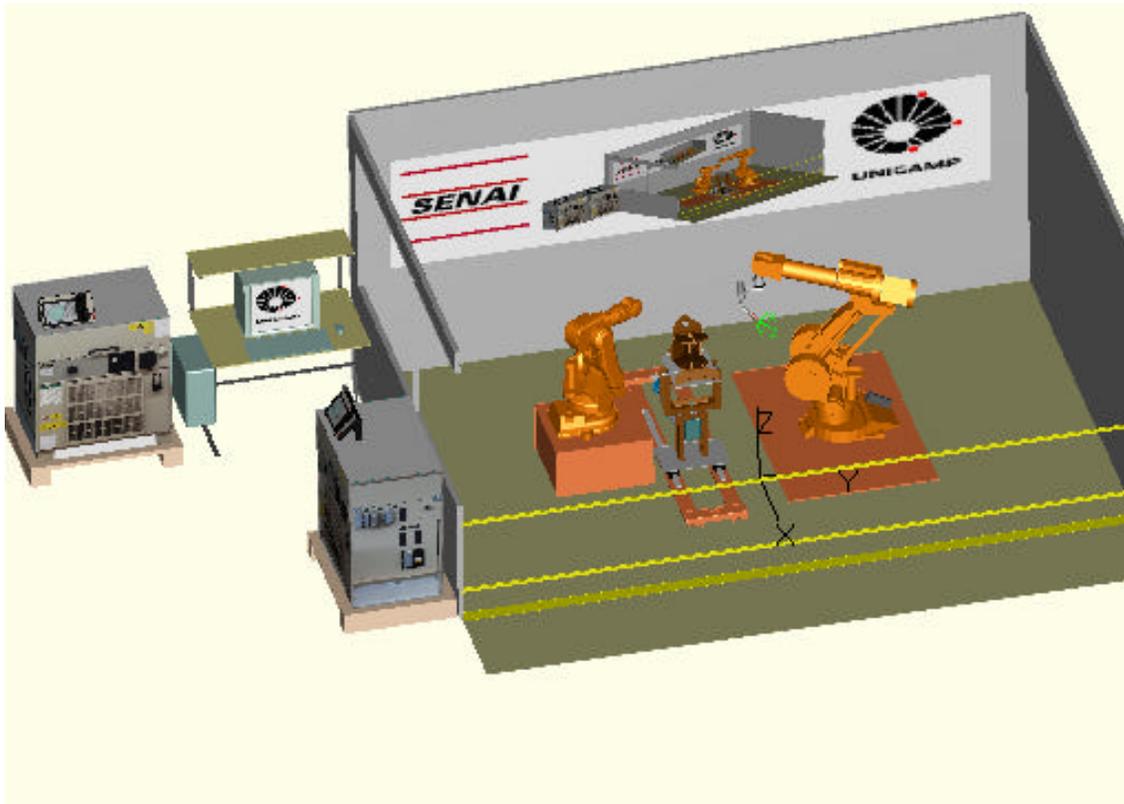


Figura 4.7 Célula Automatizada Robotizada Virtual, a ser implementada na FEM-UNICAMP.

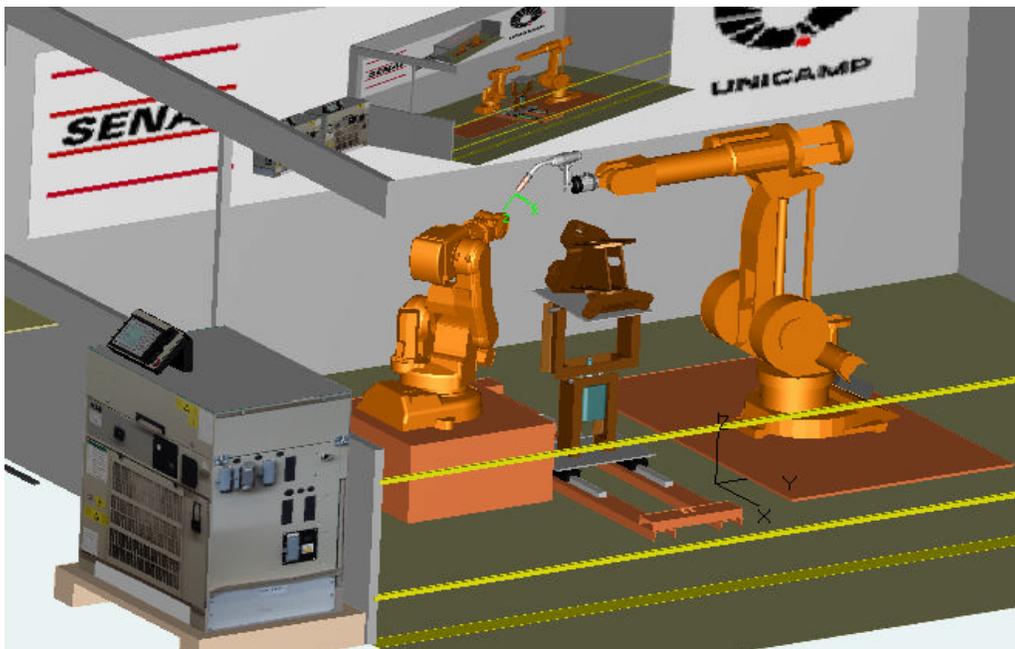
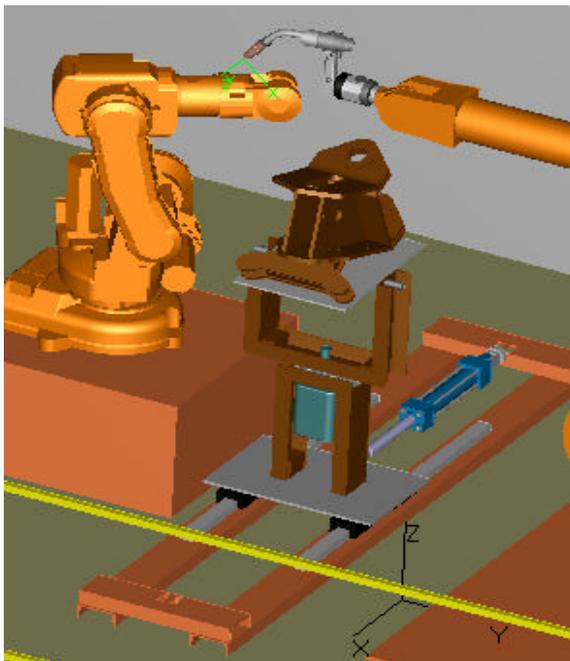


Figura 4.8: Célula Automatizada Cooperativa Virtual a ser implementada na UNICAMP.



(a) Virtual



(b) Real

Figura 4.9: Detalhe do Dispositivo Mecatrônico a ser implementada na FEM-UNICAMP.

Estes dois exemplos apresentados mostram o alto grau de estruturação na programação necessário para sincronização utilizando entradas e saídas dos robôs e dispositivos de integração utilizados dentro da célula de trabalho implementada.

4.5 Exemplo de implementação utilizando Cooperação Robótica

Um outro exemplo de cooperação robótica é apresentado a seguir (figura 4.10), onde uma operação de soldagem é executada através de dois robôs que utiliza um modelo cinemático redundante, que torna possível a eliminação de sincronização I/O descrito anteriormente, e é possível a geração de movimentos contínuos, com a consideração do objetivo final da soldagem expressa em relação à ferramenta terminal dos robôs.

Assim, um robô fará a operação de apreensão, com posição e orientação do elemento de soldagem e o segundo robô desenvolverá o processo de soldagem “GMAW”. Essa integração permitirá uma maior flexibilidade de movimentação que facilitará a posição e a orientação satisfatória no processo de soldagem.

A concepção desta célula automatizada foi implementada na Escola SENAI, Faculdade SENAI de Tecnologia Mecatrônica “Armando de Arruda Pereira”, e possui uma característica que possibilita o controle dos braços mecânicos devido à existência de dois modelos cinemático e dinâmico, contempla a facilidade de executar trabalhos cooperativos². Essa característica é fornecida pelo fabricante dos robôs que disponibilizou uma única unidade de controle para os dois braços mecânicos (com a diferenciação de um modelo cinemático embarcado).

² Eixos adicionais são eletricamente interfaceado no controle do robô. O software de movimento cooperativo permite integração matemática dos eixos adicionais com os seis eixos do robô. Os resultados da combinações elétrica mecânica e software permitem ao robô, manipular peças e ou mover-se ao longo de um caminho de forma controlada. A interconexão com o robô fornece um envelope de trabalho expandido e uma maior capacidade de manipulação de peças.

O software de simulação e programação Off-Line, não contempla esta possibilidade, pelo menos de forma direta de uso, é preciso configurar de forma não usual para atingir sucesso na implementação e neste nível surgem desafios, pouco trivial, no momento de implementar a programação para execução das atividades tarefas da célula automatizada.

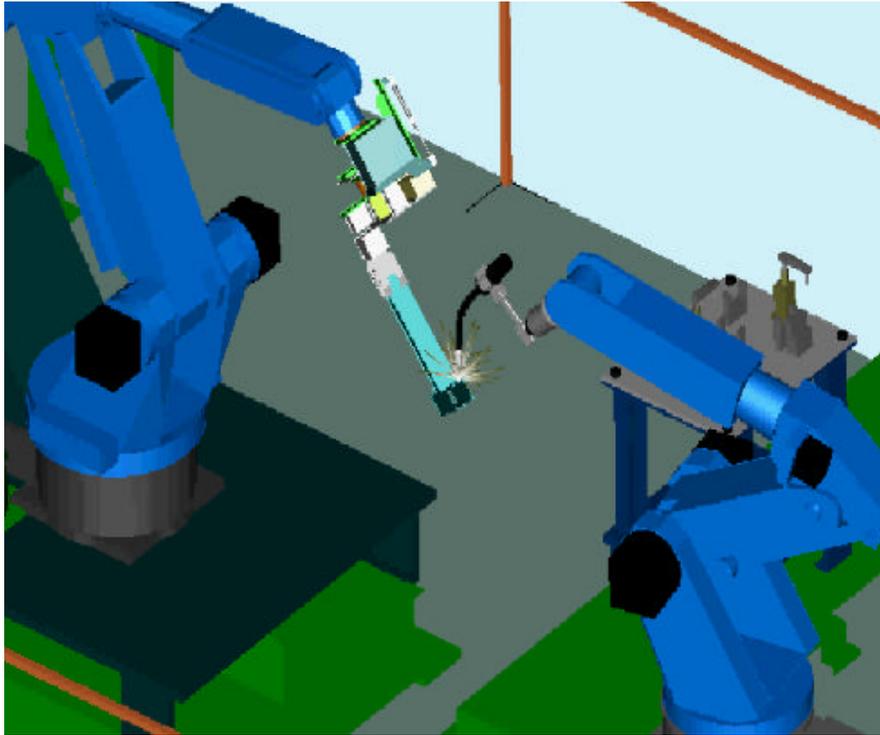


Figura 4.10 Soldagem em Cooperação.

Através dessa solução utilizou-se a concepção de uma programação avançada para contornar a falta de sincronismo entre os sinais na comunicação.

A complexidade da implementação da lógica de sincronismo e de seqüência em uma programação Off-Line implementada na célula automatizada robotizada existente na Escola SENAI e Faculdade SENAI de Tecnologia Mecatrônica “Armando de Arruda Pereira”, será abordada e demonstrada.

Na execução da validação, uma das maiores dificuldades esbarrou-se no tratamento dado na condição de trabalho de cooperação dos robôs SK6 & Sk16 e periféricos, visto que na condição real a unidade de controle do modelo compreende a possibilidade de 21 GL, e a célula

automatizada robotizada dispõem de 13 GL, respectivamente, 6 GL do robô SK6, 6 GL do robô SK16 e 1 GL da base móvel. Na figura 4.11 está delineada a lógica real da execução do programa

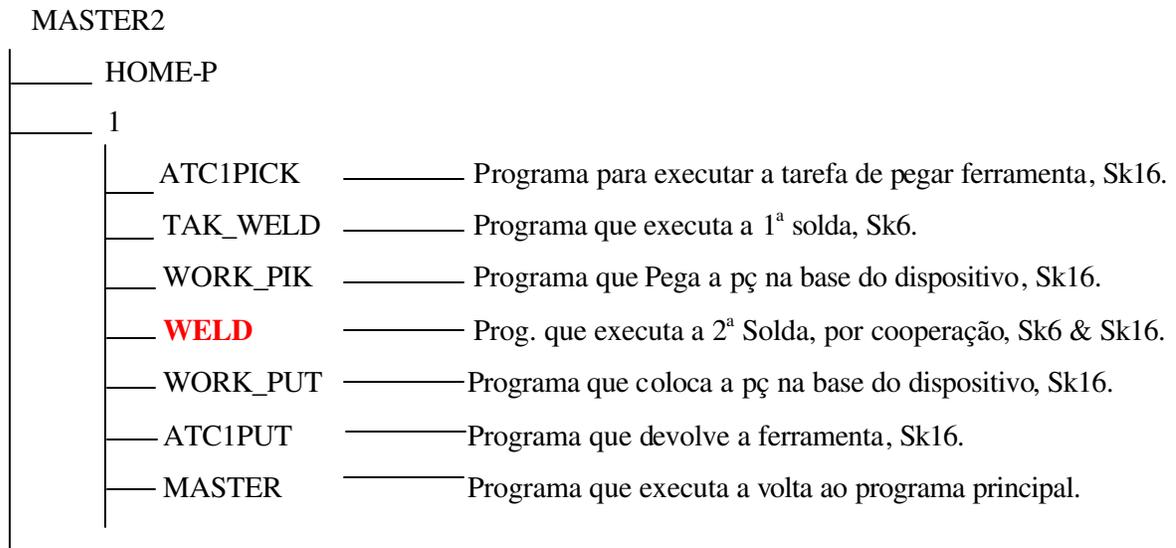


Figura 4.11 Lógica Real da execução do Programa

Nesta lógica é implementado um gerenciamento via CLP que trabalha na condição de “master” / “slave”, na realidade rodam dois programas de forma simultânea, um do CLP que é o “master”, e outro da unidade de controle do robô, “slave”. Utilizam-se sensores nos periféricos que informarão as condições necessárias para a seqüência de execução entre os programas.

No programa WELD, é utilizado o controle por cooperação entre os robôs, que representa uma complexidade para programação Off-Line, onde o robô SK16 posiciona e orienta a peça na soldagem e o robô SK6 faz a soldagem com o ajuste das orientações.

Para possibilitar a programação off-line, foi implementada uma descrição lógica que utilizou o “Grafcet Funcional”, esta ferramenta possibilita a descrição das tarefas em forma de “ações e transições” de modo seqüencial, utilizando combinações de lógicas booleanas. É preciso observar algumas regras de execução.

As ações são entendidas como saídas (Y) e as transições como entradas (X), para haver uma seqüência de execução “ação” (saída / Y) é obrigatório à passagem pela “transição” (entrada / X). O “Grafcet” pode ser desenvolvido de duas maneiras uma de forma funcional, onde não é necessária, a preocupação de uma solução e implementação tecnológica entre as entradas e saídas (valores numéricos e analógicos), e a segunda implementação, de forma tecnológica, neste caso se leva em consideração à tecnologia a ser utilizada (tipos de sensores, equipamentos, etc...).

As entradas e saídas são descritas conforme segue:

SAIDAS: (BD = base dispositivo; BM = base móvel).

Y0 = SK6 posição inicial

Y1 = SK16 posição inicial

Y2 = BD posição inicial

Y3 = BM posição inicial

Y4 = grampos da BD abertos

Y5 = SK6 posicionado 1ª / 2ª solda

Y6 = SK16 pegar ferramenta

Y7 = SK16 / BM posicionado para pegar peça na BD

Y8 = SK6 inicia a 1ª solda

Y9 = SK16 pegar peça na BD

Y10 = SK16 / BM desloca-se para posição da 2ª solda

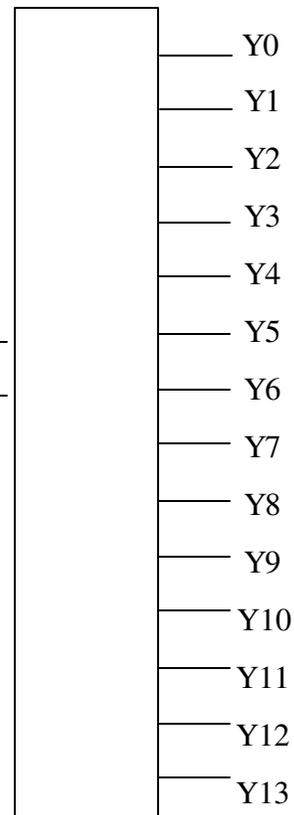
Y11 = SK6 & SK16 iniciam processo de soldagem por cooperação

Y12 = SK16 / BM desloca-se e coloca peça na BD

Y13 = SK16 devolver ferramenta

X1 _____

X2 _____



ENTRADAS:

X1 = Acionamento da botoeira

X2 = Sensoriamento e Gerenciamento CLP

Na figura 4.12, é representado a descrição lógica através do Grafcet Funcional.

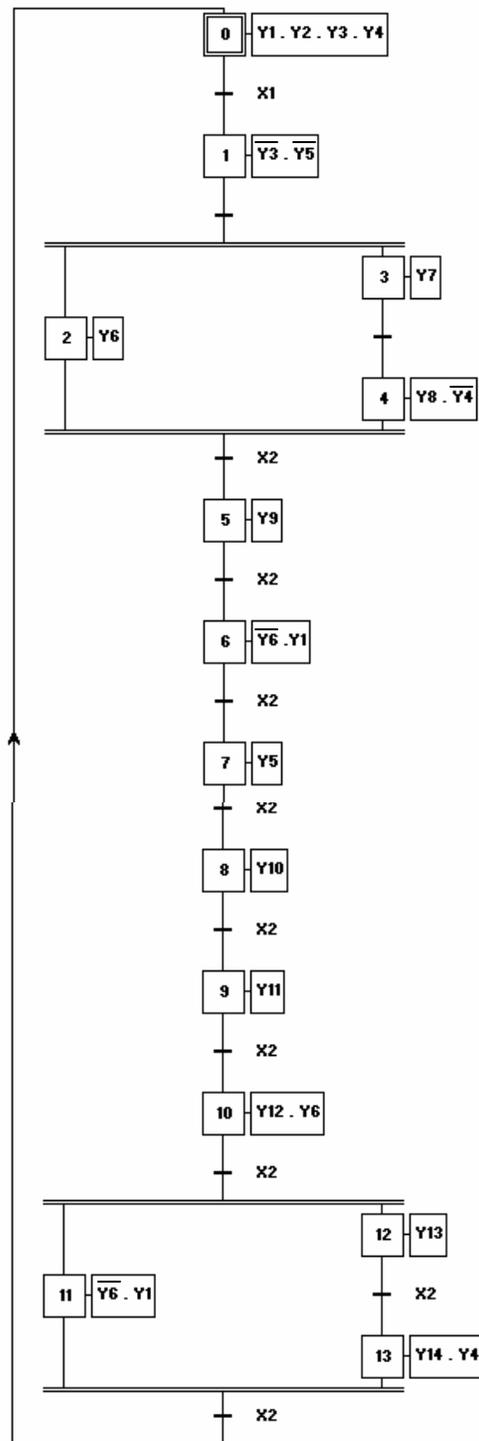


Figura 4.12 Descrição lógica através do Grafset Funcional

A dificuldade para executar a tarefa com sincronismo é elevada, a solução utilizada para contornar a complexidade, foi modelar virtualmente, individualmente os robôs. O robô SK6 teve definido a base do dispositivo com os dois grampos, como eixos externos controlado pela unidade de controle do robô e o SK16, a base móvel, e as duas partes móveis da garra.

Na figura 4.13, está representado o modelo e a situação proposta entre os robôs e os periféricos. No painel de conexão virtual, na visualização da lógica das ligações entre os equipamentos, percebe-se que a complexidade da ligação virtual dos fios nos seus respectivos endereços aumenta de forma proporcional ao número de periféricos, inviabilizando a definição do modelo.

Dessa forma foram propostos dois controles distintos e integrados, conforme ilustrado na figura 4.14 e 4.15 e visualizado no painel de controle virtual, figura 4.14a. Conseqüentemente não houve a necessidade de integrar através de sinais de I/O, os periféricos, diminuindo a dificuldade de implementar as ligações entre os robôs e periféricos. O trabalho de sincronismo é contornado e implementado por uma lógica seqüencial, levando em consideração a possibilidade de utilizar as 16 entradas e 16 saídas disponíveis.

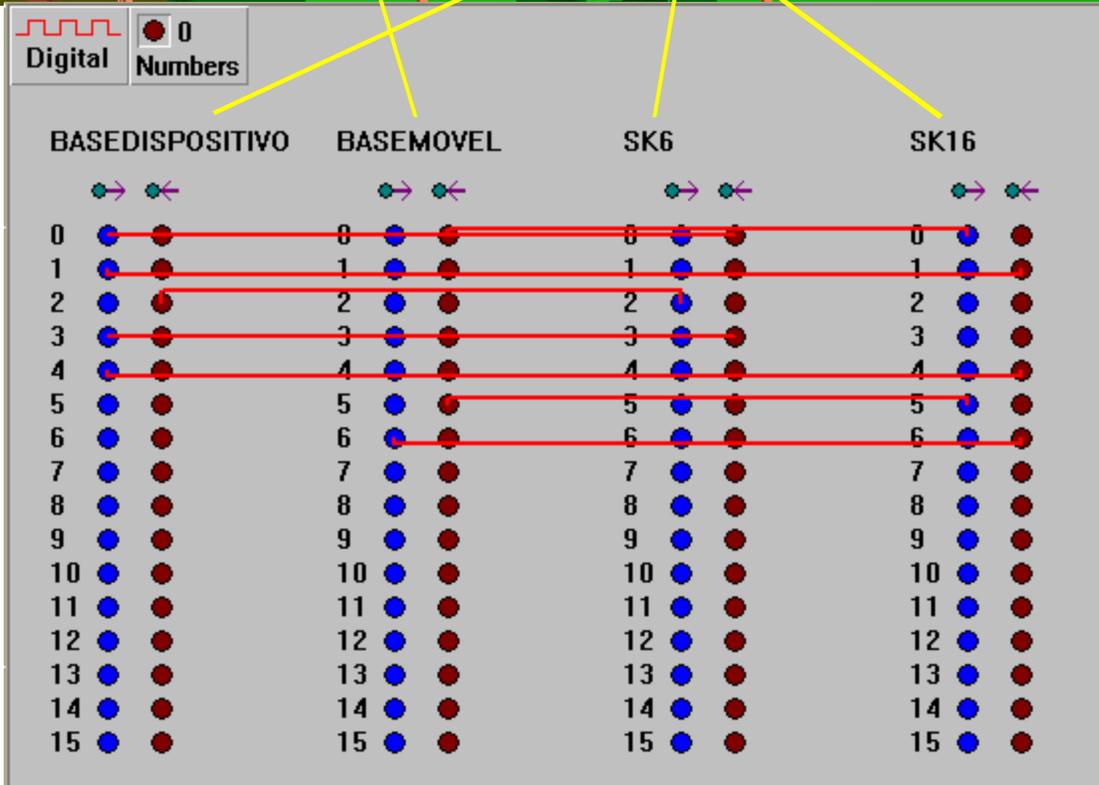
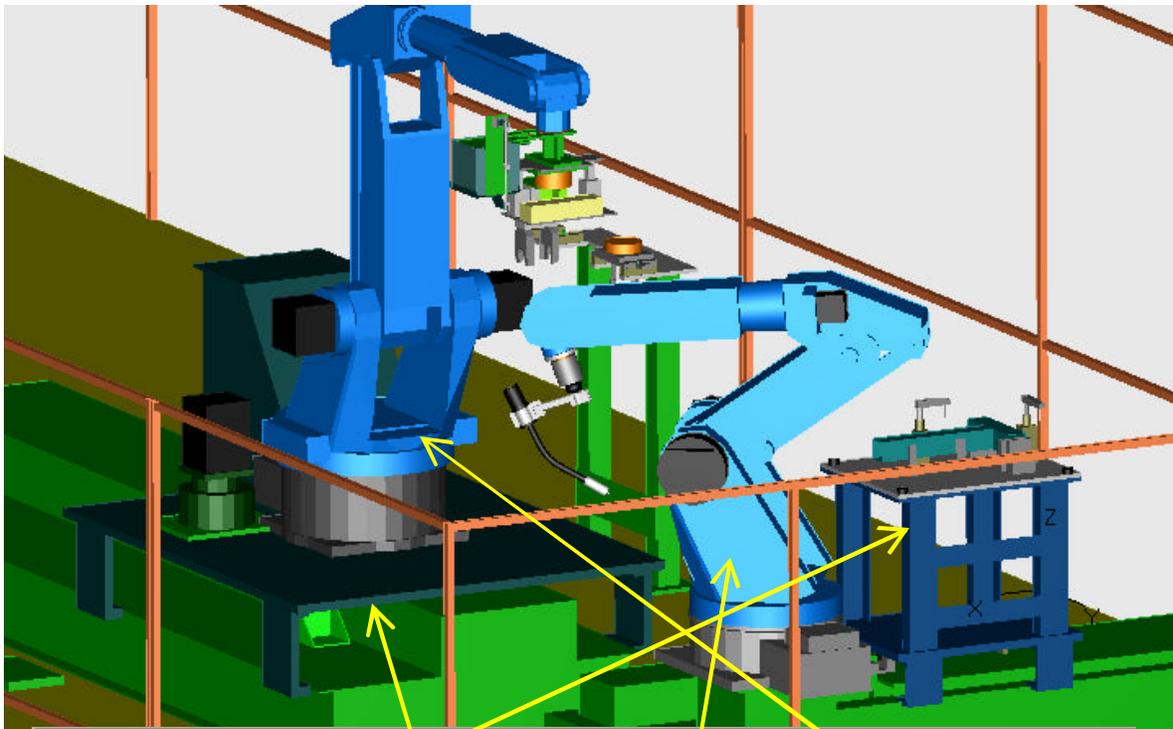
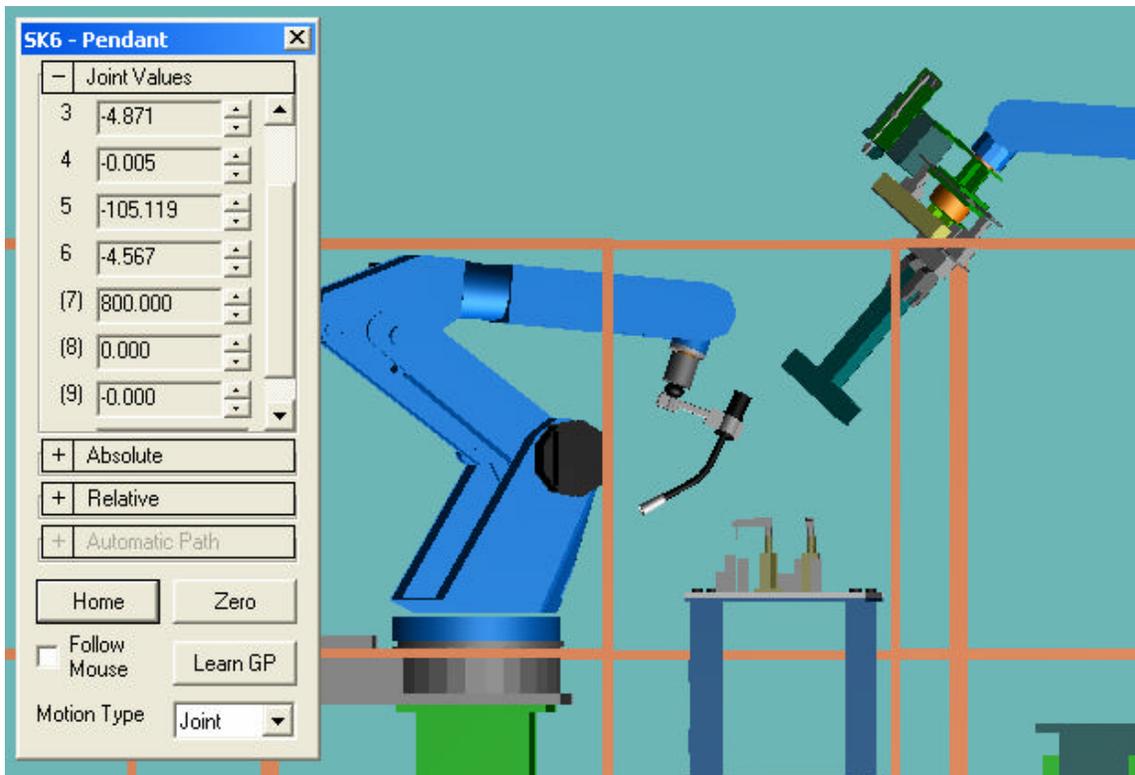


Figura 4.13 Modelo e situação proposta entre Robôs e Periféricos.



(7) Eixo externo, deslocamento da base do dispositivo; (8) 2º eixo externo e (9) 3º eixo externo, respectivamente os dois grampos de fixação do dispositivo. Controlados a través do pendante do robô SK6.

Figura 4.14 1º Controle distinto e integrado.

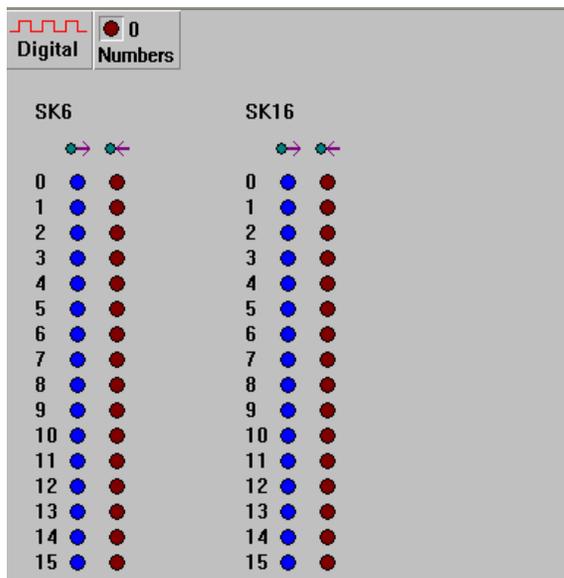
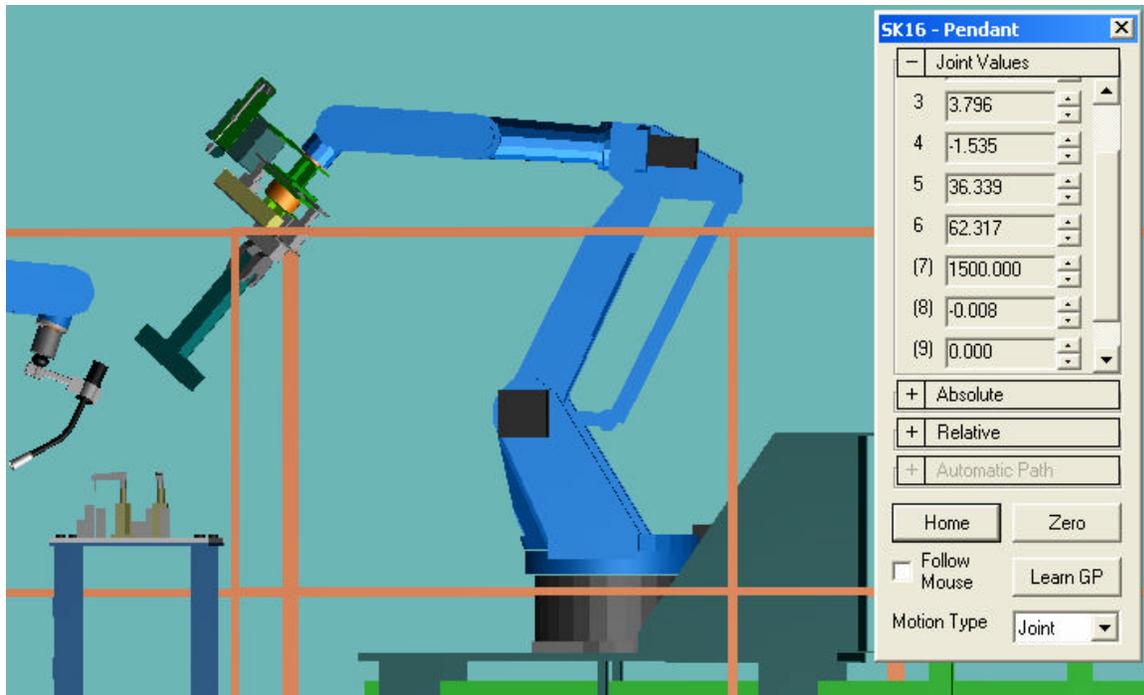


Figura 4.14a Painel de Controle Virtual.



(7) Eixo externo, deslocamento da base móvel; (8) 2º eixo externo e (9) 3º eixo externo, respectivamente os dois grampos da garra. Controlados a través do pendante do robô SK16.

Figura 4.15 2º Controle distinto e integrado.

4.5.1 Modelagem Off-Line da Célula Automatizada

O software Workspace® permite simular o comportamento de número variado de robôs industriais e de inserir desenhos relativos a modelagem em três dimensões do ambiente completo de trabalho do robô.

Inicialmente, será realizada a modelagem do ambiente de trabalho dos robôs constituintes da célula de trabalho dos robôs, ou seja, o modelo geométrico relativo ao espaço de trabalho onde serão implantados os robôs e a integração de diferentes dispositivos constituintes deste espaço de trabalho.

A partir da realização da modelagem desses diferentes elementos, torna-se conveniente verificar o espaço de trabalho dos robôs, ou seja, o espaço acessível para atuação da ferramenta terminal do robô, para verificar a correspondência entre as zonas de trabalho necessárias e determinar as eventuais regiões de colisões com os diferentes componentes da célula. Com essa finalidade, basta selecionar a opção envelope para visualizar o espaço de trabalho e realizar uma primeira verificação, como mostra a figura 4.16.

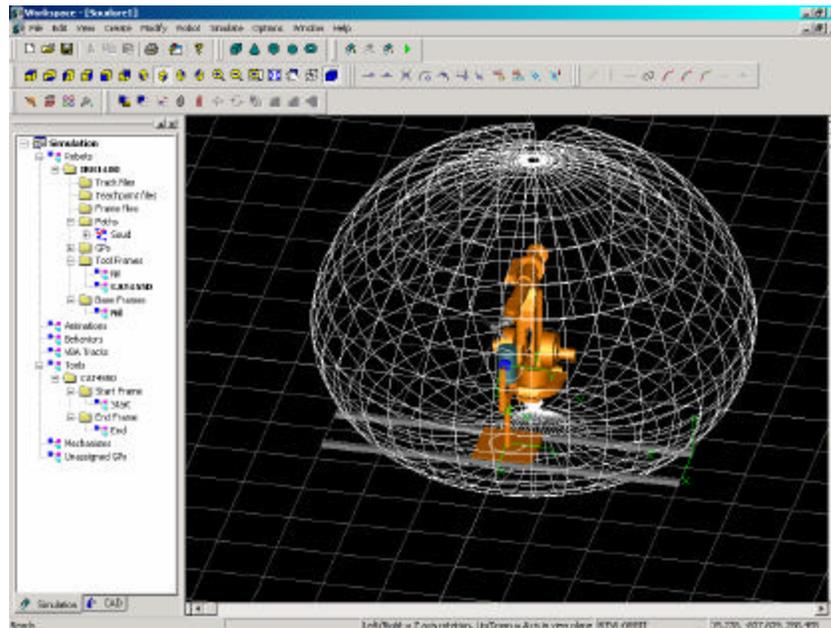


Figura 4.16: Espaço de trabalho do robô dentro da célula.

4.5.2 Simulação e Programação Off-Line

Definido o ambiente de atuação e características geométricas da célula, a próxima fase de um projeto consiste na simulação e definição das ações dos atuadores. Isto pode ser realizado a partir da definição de trajetórias dos robôs através da programação de diferentes pontos de passagem (GP's) e definição posterior da velocidade, aceleração, tipo de trajetória desejada entre dois pontos (interpolação linear, trajetória livre, etc...).

O software a ser implementado deverá prever igualmente a sincronização entre os diferentes elementos constituintes da célula automatizada. Ele deverá conter uma tabela contendo uma lista das entradas / saídas (numéricas e analógicas) de cada dispositivo automatizado, para conexão entre eles e troca de informações digitais para assegurar a sincronização entre os robôs e elementos automatizada constituintes da célula através dessa tabela. A implementação proposta nesse trabalho foi realizada através do “Grafcet”, onde cada ponto de passagem está associado a uma ação, seja pela mudança de estado da porta, ou pela espera de um valor específico sobre a porta utilizada. Por exemplo, se o robô numero um têm que atender uma ordem para efetuar a operação de soldagem, necessita que a plataforma carregue a peça para a operação e quando finalizar, efetuar a operação de descarga, obedecendo a estrutura proposta na tabela 4.1.

Plataforma	Robô
GP1 (posição de carregamento)	
GP2 (posição de soldagem) => Coloca valor porta 0 para 1	
	GP1 (posição de repouso) => Aguarda valor de 1 para porta 0
	GP2 (posição de soldagem)
	GP1 (posição de repouso) => Coloca valor porta 1 para 1
GP3 (posição de descarregamento)	

Tabela 4.1 Estrutura de troca de informações I/O.

Após especificação das diferentes ações referentes aos pontos de passagem as entradas / saídas dos diferentes dispositivos automatizados no painel de conexão virtual, conforme mostra a figura 4.17.

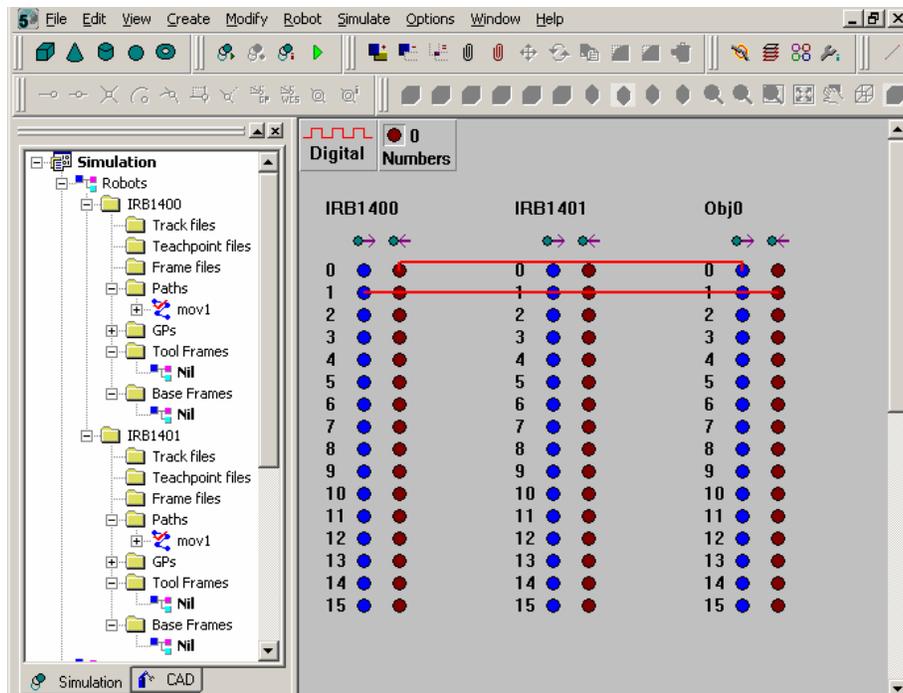


Figura 4.17 Painel de conexão virtual entre robôs e dispositivos automatizados da célula.

Uma vez terminada esta operação, as trajetórias do robô e da plataforma deverão ser sincronizadas, sendo necessário associar outras ações a alguns pontos de passagem de forma que ao colocar em funcionamento a tocha de soldagem durante a operação de soldagem. Isto pode ser realizado através de uma temporização para uma operação de soldagem ponto a ponto ou contínua entre diversos pontos para obtenção de um cordão de solda.

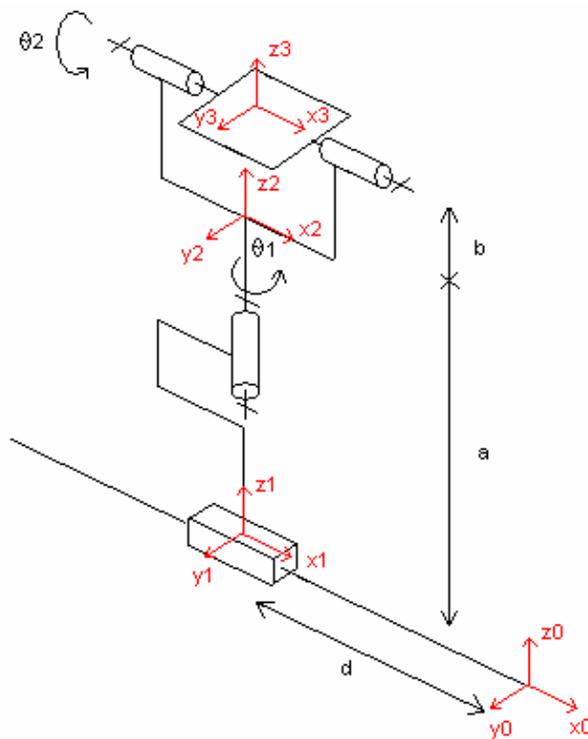
A programação Off-Line com o software Workspace® permite obtermos as linhas de código correspondente à simulação de grande parte das linguagens de programação de robôs industriais (neste caso, a linguagem RAPID para o robô ABB). Este código poderá ser diretamente transferido a unidade controle dos robôs através de uma conexão por rede ou disquete.

Os modelo cinemático direto e inverso são necessários para implementação do sistema de supervisão e controle da plataforma em ambiente Labview®. Através do modelo cinemático é fornecido o posicionamento do dispositivo em função das coordenadas articulares e o modelo

cinemático inverso fornece as coordenadas articulares em função da posição orientação final do dispositivo.

4.5.3 Modelagem Cinemática Direta

Para determinar o modelo geométrico, torna-se necessário à implementação do mesmo através da convenção de Denavit-Hartenberg (“D-H”), a partir de coeficientes contidos em tabela correspondente a posição relativa dos diferentes graus de liberdade do dispositivo (figura 4.18).



J	α_j	d_j	θ_j	r_j
1	0	D	0	0
2	0	0	θ_1	A
3	θ_2	0	0	B

Figura 4.18 Modelo cinemático do Dispositivo – Parâmetros de Denavit Hartenberg.

A partir desses parâmetros, é obtida a matriz de passagem apresentada a seguir :

$$\begin{bmatrix} \cos\theta_j & -\sin\theta_j & 0 & dj \\ \cos\alpha_j * \sin\theta_j & \cos\alpha_j * \cos\theta_j & -\sin\alpha_j & -rj * \sin\alpha_j \\ \sin\alpha_j * \sin\theta_j & \sin\alpha_j * \cos\theta_j & \cos\alpha_j & rj * \cos\alpha_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

obtendo as matrizes de passagem descritas a seguir:

$${}^0T_1 = \begin{bmatrix} 1 & 0 & 0 & d \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

$${}^1T_2 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

$${}^2T_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_2 & -\sin\theta_2 & 0 \\ 0 & \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.4)$$

$${}^0T_3 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 \cos\theta_2 & \sin\theta_1 \sin\theta_2 & d \\ \sin\theta_1 & \cos\theta_1 \cos\theta_2 & -\cos\theta_1 \sin\theta_2 & 0 \\ 0 & \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

A partir da matriz de orientação 0T_3 associada as coordenadas articulares, pode-se obter a posição e orientação final a partir da tabela .

$${}^0T_3 = \begin{bmatrix} s & n & a & x \\ s & n & a & y \\ s & n & a & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.6)$$

onde s, n e a correspondem a três vetores para a orientação de x, y e z referente as coordenadas de referência da origem.

4.6 Identificação de Parâmetros e posicionamento em relação ao referencial Zero do Robô

A integração de robôs no ambiente de trabalho necessitam do posicionamento real do mesmo em relação às tarefas a serem realizadas através da programação e elaboração de trajetórias pré-definidas em robôs, independente da operação.

Este processo pode ser otimizado através da utilização de softwares industriais para programação Off-Line, que permitem simular com relativa precisão todo o trabalho a ser realizado por um robô, desde a sua instalação, otimização de tarefas e carregamento do programa no robô.

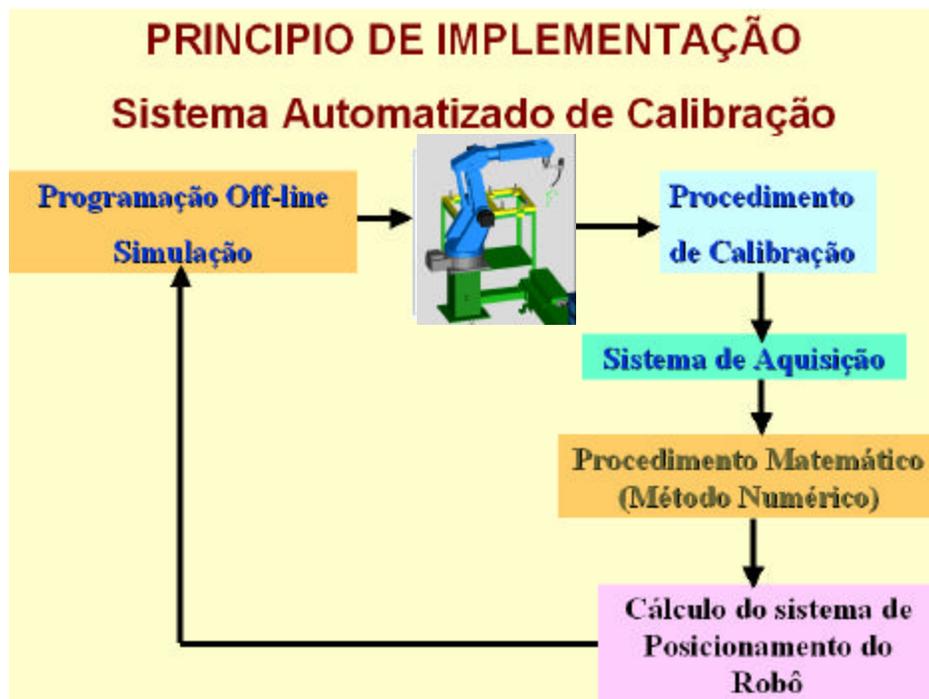


Figura 4.19: Sistemática proposta para calibração.

A qualidade da metodologia utilizada para programação é função do posicionamento real do robô em relação ao referencial zero. A execução de tarefas pré-programadas deverá considerar a identificação exata dos sistemas de referências externos ao robô, através de procedimentos de calibração automáticos, que a partir de uma série de medidas simples pré-definidas será possível à determinação do sistema real de posicionamento do robô em relação ao referencial zero e em relação ao sistema externo de medição. A figura 4.14 mostra as diferentes etapas descritas anteriormente que serão estudadas e avaliadas para realizar o objetivo preconizado neste projeto de pesquisa.

4.7 Conclusões

Neste capítulo foram apresentados através de exemplos práticos aspectos concernentes aos processos de programação avançada de robôs, onde existe a possibilidade real de ganho no processo, enfocando uma utilização racional nos processos de programação e que venha a agregar de forma significativa a concepção de programação, enfatizando também um estudo relativo à calibração de robôs, a partir de obtenção de parâmetros cinemáticos durante a simulação, onde a partir desse estudo de casos será possível a realização de uma análise da aplicabilidade enfocando a concepção e planejamento no uso da simulação e programação Off-Line de robôs.

Capítulo 5

Conclusões Finais e Perspectivas Futuras

Esta dissertação de mestrado teve como objetivo a descrição de ferramentas e metodologias direcionadas a concepção, integração, planejamento, e operação de células automatizadas utilizando robôs industriais.

Dentre as características exploradas nesse trabalho foram considerados os seguintes aspectos:

- i. Estruturação de programas de integração: que requerem a utilização de conceitos de automação, tais como as ferramentas de descrições que permitem de forma funcional e tecnológica seqüenciar uma lógica de execução de uma tarefa (Fluxograma seqüencial - SFC e Rede de Petri - RdP), permitindo a implementação de programas off-line estruturados e posterior carregamento nos softwares de simulação a criar possibilidade da transferência para dispositivos robóticos (carregamento).
- ii. Concepção de sistemática de calibração do sistema robótico, envolvendo robô, ferramentas de final de braço e processos. O tratamento a ser utilizado para identificação da posição de referencia Zero, utilizando sensores externos de medida, auxiliados por procedimentos

matemático envolvendo vetor posição inicial e matriz de orientação a ser inserido automaticamente no programa off-line para carregamento posterior no dispositivo robótico.

Para que haja uma possibilidade real direcionada a implementação nas empresas de pequeno e médio porte, da metodologia de um Projeto de Células Robotizadas Industriais, utilizando conceitos de programação off-line, é preciso propiciar um maior conhecimento na utilização dessa metodologia, mostrar a necessidade de um novo perfil de profissional para atuar no processo, através de um estabelecimento de proposta de formação profissional.

Atualmente muitas empresas de pequeno e médio porte (PME-PMI), vêm adaptando-se às tecnologias através de investimentos tímidos e precários, por falta de informação ou uma relação custo / benefício desfavorável de investimento em equipamento ou em profissionais qualificados.

Conforme observado por (Nitsche e Romeiro, 2002) todo o processo de fabricação é integrado através da utilização de ferramentas de CAD / CAE / CAM. Realidade muito distante as pequenas e medias empresas, quando muito está iniciando o processo de via sistema CAD em duas dimensões e desconhecem o potencial competitivo da geração de modelos geométricos tridimensionais, além disso, o alto custo de implementação destes sistemas torna muitas vezes impossível a utilização consistente destes recursos. (Nitsche e Romeiro, 2003) denominou de “Estado das Práticas” um estudo para comparação na utilização de ferramentas CAD / CAE / CAM entre pequenas e medias empresas, do setor automotivo.

Doravante, utilizaremos emprestado o termo “Estado das Práticas”, na filosofia de um Projeto de Células Robotizadas Industriais, utilizando conceitos de programação Off-Line, para fomentar uma popularização no seu uso. Pode tornar-se pouco motivador em função de vários fatores, tal como investimento em ferramentas e principalmente na obtenção da mão-de-obra qualificada. Uma vertente possível para minimizar as dificuldades iniciais de qualquer novo empreendimento tecnológico pode ser através de uma cooperativa de tecnologia, a qual possibilite uma distribuição de investimentos e recursos, visto que o investimento em

equipamentos e formação de mão-de-obra qualificada torna-se inviável para maioria das pequenas e médias empresas.

Existem algumas iniciativas por parte do governo para suprir estas carências e limitações, como por exemplo, a concepção da Rede de Cooperação e Pesquisa, através do sub grupo de Automação e Robótica (RECOPE-MANET)(<<http://www.manet.org.br/manet/home/index.jsp>>).

Essas ações não foram suficientes, é necessária uma profunda mudança na formação profissional somada aos apelos tecnológicos para a utilização de novas perspectivas de maneira consistente.

Para enquadrar o “Estado das Práticas”, devemos perfilar um novo profissional de engenharia que consiga conceber aplicações rearranjando as ferramentas tecnológicas que estão fragmentadas em diversas vertentes, exigindo de forma concreta a integração dos conceitos de automação e simulação com as tecnologias de informação.

5.1 O ensino da programação off-line.

Ao discutir a formação em Robótica, duas vertentes precisa ser distintamente analisada uma relacionada, à formação no nível técnico médio e outro em relação ao nível técnico superior. As atribuições relacionadas são ligeiramente distintas. O enfoque desejável para o profissional que atua no desenvolvimento e planejamento de novas aplicações enfatiza situações de concepção em engenharia e para o profissional inserido dentro de um contexto industrial, torna-se grande a importância, o ensinamento de técnicas de aprendizado de tarefas a partir de conceitos básicos de sistemas de coordenadas e modelagem cinemática básica, se preocupando com o posicionamento e orientação da ferramenta terminal e conceitos de programação estruturada direcionada a integração de sistemas automatizados.

O emprego destas ferramentas representa uma nova filosofia de trabalho na concepção, projeto e programação de uma célula robotizada, devendo ser trabalhada em todos os aspectos apresentados e o profissional com nível técnico médio e o profissional com nível técnico superior terão suas atribuições em níveis diferentes de aplicação.

5.1.1 Visualização gráfica.

Em termo de visualização gráfica, este profissional deverá gerar um modelo gráfico e analisar a disposição de lay-out. Definir a estratégia de lógica na programação faz parte de uma das partes dos processos e a exigência na concepção, projeto e programação serão implementados e gerados conforme a formação atribuída e a necessidade de exigência de cada usuário.

O processo de geração de uma simulação independe do software, da linguagem de programação que se utilize e deseje, o processo pode atender a uma simples demanda em que o foco principal é a visualização do processo e não será levada em consideração a lógica implementada ou muito menos o grau de estruturação que se desenvolveu para uma determinada situação.

Em muitos casos não é necessário um software que gere a simulação pelo processo de programação, um simples software que possibilite tratar partes gráficas, no plano, ou seja, em duas dimensões 2D, resolve de forma satisfatória o problema em estudo, não exigindo ainda a necessidade de um profissional que tenha conhecimento sobre o processo robótico.

Neste contexto a dicotomia de capacitação está no que pode ser executado, levando em consideração justamente à lógica de implementação e da estruturação, viabilizando a transferência do programa para a unidade de controle do robô.

5.1.2 O profissional de nível médio.

Atualmente um profissional de nível médio com formação em robótica tem necessidade de conhecimentos de estruturação e conhecimento espacial para um planejamento customizado de uma determinada tarefa, onde a abordagem principal recaia na palavra redução de custos, que traduz na otimização de tempos de implementação de programas e sistemática de aprendizagem de pontos importantes. Assim dois aspectos deverão ser considerados:

- i. Devemos observar que quanto menor tempo um usuário dispensar na aprendizagem de pontos de uma determinada trajetória (posicionamento e orientação da ferramenta terminal em relação à tarefa pré-determinada, melhor a relação programação\ produção).
- ii. Outro enfoque recai na implementação de programas estruturados utilizando uma forma textual mais abrangente, implementado sub-rotinas e funções. Essa possibilidade advém de uma prática no ato de programar e ou reprogramar permitindo programas inteligentes e de fácil modificação em novas situações e implementações, podendo facilmente ser modificados e direcionados a outras aplicações, tornando a utilização do robô mais flexível.

5.1.3 O profissional de nível técnico superior

Por outro lado, o profissional de nível técnico superior deverá ter um conhecimento analítico e base matemática relativa à representação do comportamento do problema através da modelagem geométrica, cinemática e dinâmica além de conhecimento de conceitos e formalismos utilizados para integração de sistemas automatizados.

Esta formação permitirá o desenvolvimento de novos equipamentos e sua integração com a unidade de controle de um robô. Tal enfoque exige um alto nível de conhecimento e uma forma especial na interação de tecnologias.

Estes predicados tornam de forma evidente uma formação mais especializada possível, desta forma a dicotomia entre formação é evidente e o cuidado em fazê-lo é fundamental para uma formação condizente com as atribuições que se espera dos profissionais envolvidos com essa tecnologia, onde:

- i. O profissional envolvido com programação e reprogramação não precisa ter grandes conhecimentos em modelagem de robôs ou muito menos a implicação que as fazem nos processos de programação, simplesmente necessita conhecer os princípios básicos de controle do robô e os comandos necessários para a implementação de trajetórias seguras durante a fase de aprendizagem. Deverá ser enfocada na formação, são os aspectos relacionados à integração de equipamentos envolvendo os diversos fabricantes de robôs e periféricos, onde os conceitos básicos são os mesmos, diversificando na forma de apresentação dos comandos, segundo a sua sintaxe e lógica de programação.

Esses conhecimentos serão de extrema importância na concepção, programação e implantação de células robóticas flexíveis, que exige elevado nível de conceitos e de estruturação de programas, permitindo a implementação de especificações tecnológicas de maneira rápida e eficiente, contemplando ainda as condições de segurança necessárias no projeto final. Dentro desse contexto, os sistemas CAD têm uma consequência diferente, que de certa forma são subutilizados, impedindo uma utilização correta e ampla, que podem agregar valor no processo ou projeto no qual são utilizados, segundo (Nitsche, 2003).

Nas duas situações de formação abordadas neste contexto, o objetivo é dar condições para um crescimento gradual na utilização das tecnologias envolvidas, que corretamente utilizados proporcionarão um resultado efetivo no ambiente fabril e na implementação de novos processos que envolvam a tecnologia robótica.

5.1.4 A integração de sistemas automatizados.

A integração de sistemas tem uma abrangência maior de utilização, onde também são necessários conhecimentos de protocolos de comunicação, o que tem recebido uma abordagem mais aberta possível, em que a interconexão de sistemas e equipamentos, vem sendo mais flexíveis nas suas utilizações.

A formação é direcionada a um conhecimento de programação que não só defina aspectos relativos à movimentação de um robô, mas a comunicação entre diferentes dispositivos através de suas linguagens.

Uma tendência atual direciona a facilidade na integração e programação que só será alcançada com a formação mais capacitada dos profissionais relacionados à área em questão. Essa formação precisa ser mais explorada permitindo uma disseminação, diminuindo a distância entre os profissionais capacitados que buscaram dominar a tecnologia, através de formações complementares e os novos profissionais a serem formados, propiciando uma relação mais favorável na utilização da robótica.

A figura 5.1 apresenta um fluxograma proposto relativo à formação profissional de nível técnico superior e o profissional de nível médio. Essa visão é destinada a demonstrar a diferença de atitude e comportamento entre os níveis de formação e o grau de profundidade a ser trabalhado na formação desses profissionais que evidencia a atuação e atributo no campo de robótica.

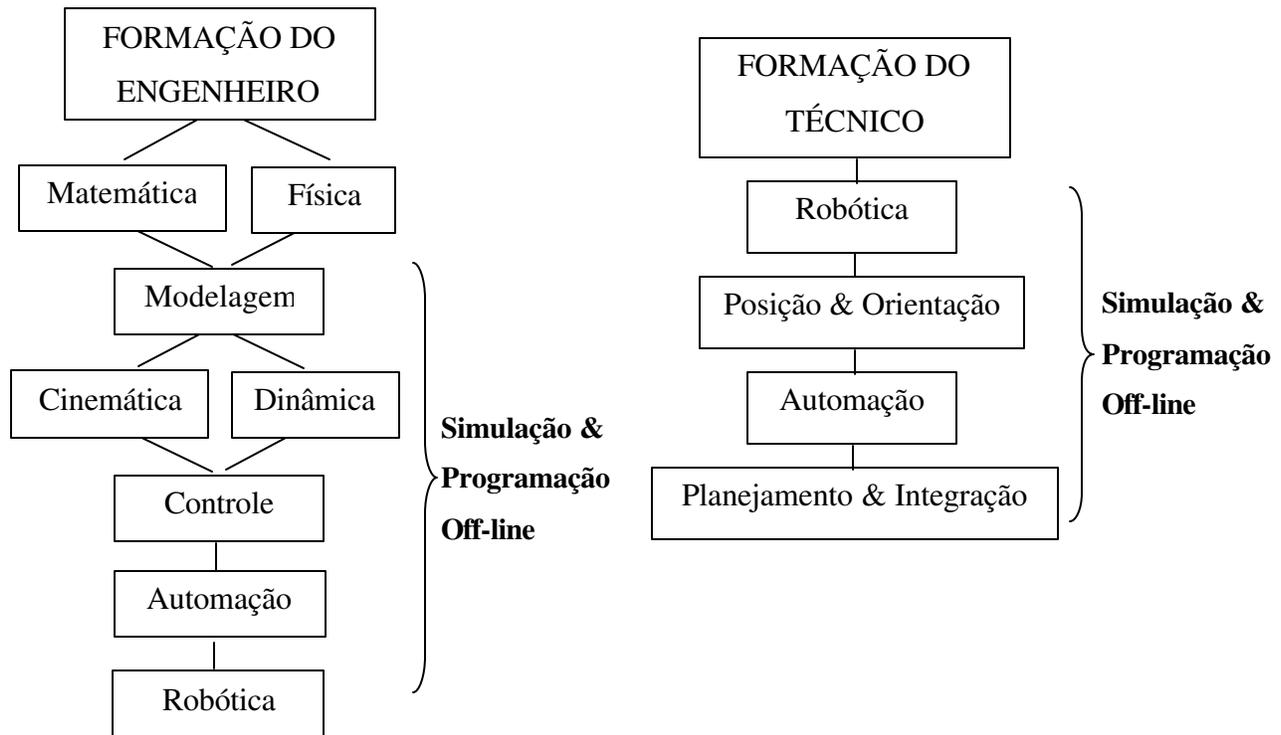


Figura 5.1 Proposta de formação de profissional em Mecatrônica.

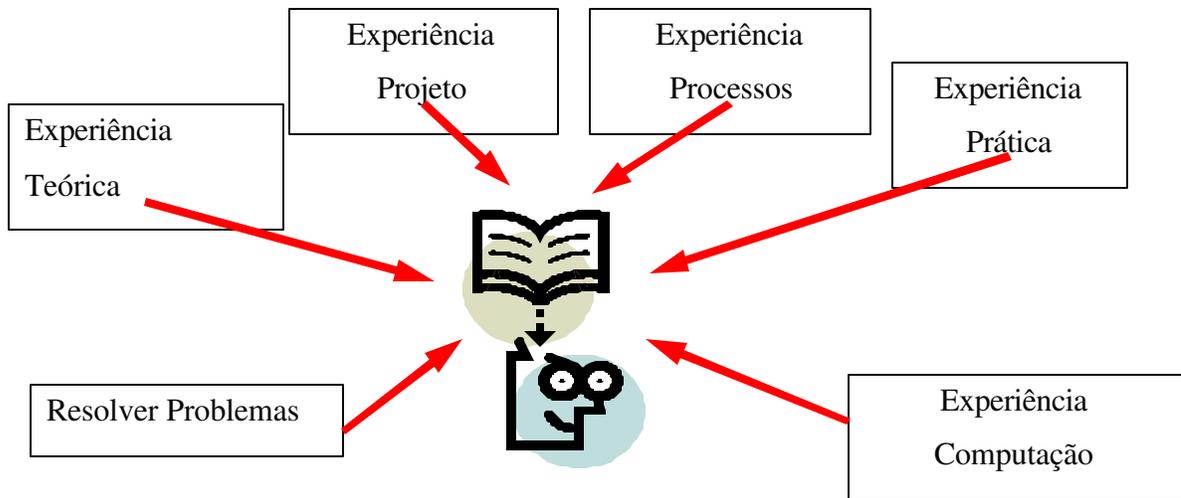
5.2 – Exigências do profissional na área de robótica

Dentre as exigências necessárias na formação dos profissionais de Robótica, podemos contemplar aspectos relacionados à modelagem e projeto de controle de equipamentos robóticos e de sua integração com outros equipamentos disponíveis. Nesta condição existe uma lacuna muito grande entre os profissionais em questão, e se não forem tomados os cuidados necessários pode ocasionar uma grande discrepância na formação dos profissionais na área, onde a relação entre as ferramentas matemáticas abordadas podem acabar levando a um descrédito do processo de ensino. Ferramentas matemáticas são suficientemente abstratas por serem desenvolvidas de forma genérica, possibilitando também uma abordagem no campo robótico e o grau de maturidade no aprendizado pode levar a uma falha grave de utilização, despendendo tempo e recursos.

O técnico de nível médio, está ansioso pelo aprender no uso do equipamento e suas possibilidades de integração e a proposta de uso do robô, dispensa qualquer abordagem matemática, tornando neste caso uma condição secundária. O uso de robô para o ensino ponto a ponto, onde se procede com as questões de posição e orientação é de fácil entendimento e relativamente baixa complexidade em fazê-la.

Uma das questões que impossibilita a disseminação da tecnologia no ambiente fabril, já mencionado anteriormente na revisão bibliográfica é exatamente ter um profissional capacitado para atender de forma ampla a implementação entre tecnologia e o ambiente fabril, neste caso a média e pequena empresa precisa desprender um investimento significativo em função do próprio valor do equipamento robô com seus periféricos, bem como na capacitação de profissionais para tornar exequível a tecnologia. A disponibilidade de um departamento que concentre este tipo de profissional normalmente é a Engenharia de Métodos & Processos e com a escassez econômica que as empresas de pequeno e médio porte dispõe, é utópico imaginar que tais empresas disponham da Engenharia de Métodos & Processos e dos engenheiros ou dos técnicos que venham a suprir as suas necessidades.

Esse contexto é reforçado através de uma nova formação de profissionais que atendam de forma natural essa nova tendência tecnológica que já é uma realidade em nossas plantas fabris e que vem sofrendo desacertos em virtude de uma qualificação que não fornece a competência adequada para a resolução e novas propostas de integração utilizando as concepções abordadas.



Adaptação da figura Do Engenheiro de Simulação
“Ideal”, em (De Vin, Jägstam, 2001) por (Freitas 2004)

Figura 5.2 Enfoque de profissional de engenharia do século XXI

O perfil de saída de um novo profissional de engenharia do século XXI, simplificado na figura 5.2, força a conceber uma formação em robótica, audaciosa que atenda inclusive as necessidades das empresas pequenas e de médio porte, levando em consideração as precariedades de recursos comumente encontradas.

Referências Bibliográficas

Acar, M. Mechatronics Challenge for the Higher Education World. *IEEE Transactions on Components, Packing, and Manufacturing Technology*, vol. 20, n. 1, p.14-20, 1997.

Anton, S., et al. *A framework for Realistic Robot Simulation and Visualisation*.

Disponível em: <<http://www.easy-rob.com/data/Frame-RRS.pdf>>. Acesso em: 29 mar. 2003.

Asfahl, C. Ray. Building Blocks of Automation. In: *Robots and Manufacturing Automation*. 2nd ed. New York: John Willey & Sons, 1992. p. 23-53.

Ashley, S. Getting a Hold on Mechatronics. *Mechanical Engineering, ASME*, Mai. p. 60-63, 1997.

Brumson, B. Emerging Trends in Robotic Integration. *Robotics online e-news*. Michigan, Nov.2001. Disponível em: <<http://www.roboticonline.com/public/articles/archivedetails.cfm?id=577>> Acesso em:19 jan. 2004.

Brumson, B. Robots for small business: A Growing Trend. *Robotics online e-news*. Michigan, Out.2003.

Disponível em: <<http://www.roboticsonline.com/public/articles/archivedetails.cfm?id=1231>>

Acesso em: 19 jan. 2004.

Camelot, Ropsim. *Robot Off-line Programming and Simulation system for industrial robots*. 2003. Disponível em: < <http://www.camelot.dk/english/offvson.htm> > Acesso em:26 jun. 2003.

Campbell, J.E. Robot Motion Control: How Special is it? *Robotics online e-news*. Michigan, Set.2003.

Disponível em: < <http://www.roboticsonline.com/public/articles/archivedetails.cfm?id=1174> >

Acesso em:19 jan. 2004.

Craig, J.J. *Introduction to robotics: mechanics and control*. 2nd ed. New York: Addison Wesley, 1989. 450 p.

De Vin, Leo J., Jägstam, Mats. Why we need to offer a modeling and simulation engineering curriculum. *Proceedings of the 2001 Winter Simulation Conference*, p. 1599-1604, 2001.

Disponível em: <<http://www.informs-cs.org/wsc01papers/219.pdf>.> Acesso em: 12 Fev. 2003.

Denavit, J., Hartenberg, R.S. A Kinematic Notation for Low-Pair Mechanisms Based on Matrices. *ASME Journal of Applied Mechanics*, p 215-221, 1955.

Djuric, A. *Economical Industrial Workcell Modeling Simulation and Layout Design*. Windsor, Ontario, Canada: Faculty of Graduate Studies and Research Industrial and Manufacturing System Engineering Program, University of Windsor, 1999. 164 p. Tese (Mestrado)

Easy-Rob,*Easy-rob*,The Company for 3D Robot Simulation.

Disponível em: <<http://www.easy-rob.com>> Acesso em:03 Set. 2003.

EF-Robotertechnik GmbH, Cosimir® Professional. *Manual do usuário*. 2002.

Ferreira, E. P. Robótica Básica. In: *V Escola Brasileiro-Argentina de Informática*, 1991. Rio de Janeiro, 1991.

Fiedler, P. J., Schilb, C. J. Open Architecture Robot Controllers and Workcell Integration. *Robotics online e-news*. Technical Papers, Michigan. Disponível em: http://www.roboticonline.com/public/articles/Open_Architecture_Genesis.PDF Acesso em: 17 set. 2003.

Fiedler, P. J., Dehof, M.K. Workcell Communications: Connectivity, Man-Machine interfaces, and multi system management . *Robotics online e-news*. Technical Papers, Michigan. Disponível em:

http://www.roboticonline.com/public/articles/Workcell_Communications_ANS98_Genesis.PDF > Acesso em: 17 Set. 2003.

Flow Software Technologies Ltd. *Workspace5, Robotic simulation and off-line programming software*. Disponível em: < <http://www.workspace5.com> > Acesso em: 03 Set. 2003.

Flow Software Technologies Ltd. *Workspace 5.04*: manual do usuário. Maio. 2003.

Fu, K.S., Gonzáles,R.C., Lee,C.S.G. *Robotics: Control, Sensing, Vision, and Intelligence*. Singapore: McGraw-Hill International, 1987. 580 p.

Gong, W. *Automation Robot Path Generation for Manufacturing on Sculptured Surfaces*. Windsor, Ontario, Canada: Faculty of Graduate Studies and Research Industrial and Manufacturing System Engineering Program, University of Windsor, 1998, 155 p. Tese (Mestrado).

Groover, Mikell P. Automation. In: Dorf, Richard C., Nof, Shimon Y. *International Encyclopedia of Robotics: applications and automation*. New York: John Willey & Sons, 1988. v.1. p 136-151.

Hardin, W. Robotic Integrators take their Specialties Seriously. *Robotics online e-news*. Michigan, Abr.2002.

Disponível em: <<http://www.roboticonline.com/public/articles/archivedetails.cfm?id=723>>

Acesso em:17 set. 2003.

Hewit, J.R., King, T.G. Mechatronics Design for Product Enhancement. *IEE & ASME Transactions on Mechatronics*, v. 1, n. 2, p. 111-119, 1996.

Nagle, B. Your First Robot. In: *Robotics Industry Directory - 2002-03 Edition*. Robotic Industries Association, 2003. p 8.

Nitsche, A.T., Romeiro Filho, E. Aplicação de Tecnologias de CAD/CAE/CAM em Desenvolvimento de Produtos. In: *Encontro Nacional de Engenharia de Produção*, ENEGEP, XVII, 2002. Curitiba. PR.

Nitsche, A.T., Romeiro Filho, E., Gonçalves, J. B., Gomes, D.T. Uso das Ferramentas de CAD/CAE/CAM no desenvolvimento de produtos: o estado das práticas em pequenas e médias empresas. In: 2ºCOBEF, 2003.

Paul, R.P. *Robot Manipulators, Programming and Control*. Cambridge: MIT Press,1981. 279 p.

Robot Simulation Ltd. *Workspace 4.01 Manual do usuário*. Local: 1999.

Rosário, J.M. *Curso de Especialização modalidade extensão universitária em Automação Industrial, Automação Industrial – Robótica – Diagrama Funcional para Sistema Automatizados Discretos*. Campinas: Faculdade de Engenharia Mecânica, 1998.

Rosário, J.M. *Projeto Pedagógico do Curso de Graduação de Engenharia de Controle e Automação da UNICAMP, Faculdade de Engenharia Mecânica, Coordenação de Mecatrônica*. Campinas: UNICAMP, 2000.

Rosário, J.M. Modelagem e Controle de Robôs. In: Romano, V.F. (Ed). *Robótica Industrial: aplicação na Indústria de Manufatura e Processos*. São Paulo: Edgard Blucher, 2002. p 20-46.

Roos, E., Behrens, A., Anton, S. *RDS – Realistic Dynamic Simulation of Robots*. Disponível em: < <http://www.easy-rob.com/data/rds.pdf>>. Acesso em: 29 mar. 2003.

Roos, E., Behrens, A. Off-line Programming of industrial robots: adaptation of simulated user programs to the real environment. *Computers in industry*, v. 33, n.1, p. 139-150, 1997.

Sá, C. E. *Implementação de Métodos Numéricos para a Resolução do Problema Cinemático Inverso de Manipuladores Robóticos com Ênfase em Controle de Posição*. Campinas: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 1996, 114p. Tese (Mestrado).

Sá, C. E. *Desenvolvimento e Implementação de um Programa Computacional para a Supervisão e Controle de Manipuladores Robóticos*. Campinas: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2000, 255p. Tese (Doutorado).

Salminen, V: Ten Years of Mechatronics Research and Industrial Applications in Finland. *IEEE - ASME Transactions on Mechatronics*, vol. 1, n. 2, p. 103-105, 1996.

Trostman, E., Conrad, F., Trostman, S., Nielsen L.F. Robot Off-line programming and simulation as a true CIME subsystem. *IEEE International Conference on Robotics and Automation*. 1992.

Vollmann, K. Realistic Robot Simulation: Multiple Instantiating of Robot Controller Software. In: *IEEE ICIT'02, International Conference On Industrial Technology, 2002* . p. 1194-1198, 2002.

Yong, Y.F., Bonney, M.C. Off-line Programming. In: Nof, Shimon.Y. (Ed). *Handbook of Industrial Robotics*, 2nd ed. New York: John Wiley & Sons, 1999. p. 353-371.

Anexo I – Principais Comandos e Exemplos de Programação de Robôs Utilizando Linguagem Textual

Iniciaremos baseado na linguagem de programação RAPID™, disponível nos robôs industriais fabricados pela ABB™.

Os comandos e estrutura de programação utilizada normalmente são baseados na linguagem computacional PASCAL, tendo forte ênfase à programação estruturada. Assim as funções e procedimentos podem ter argumentos e as variáveis podem ser globais ou locais (a uma função ou procedimento), com possibilidades de implementação de recursões e módulos. A figura AI.1 apresenta a lista de instruções da linguagem de programação RAPID™ do robô ABB™.

<u>Instruções Básicas</u>		<u>IO's</u>	<u>Comunicação</u>	
1. MoveJ	11. WaitUntil	1. InvertDo	11. Close	
2. MoveL	12. WaitTime	2. PulseDo	12. ErrWrite	
3. MoveC	13. Wait Di	3. Reset	13. Open	
4. Test	14. Set	4. Set	14. TpErase	
5. ProcCall	15. Reset	5. SetAO	15. TpReadFK	
6. Return	16. Break	6. SetDO	16. TpReadNum	
7. For	17. Exit	7. SetGO	17. TpWrite	
8. While	18. Stop		18. Write	
9. Compact If	19. Label			
10. If	20. Comment			
<u>Parâmetros</u>		<u>Erros e Sistema</u>	<u>Interrupções</u>	<u>Funções Matemáticas</u>
1. AccSet	11. PdispSet	. Exit	1. Connect	1. :=
2. ConfJ	12. ingArea	. Raise	2. IDelete	2. Add
3. ConfL	13. SoftAct	. Retry	3. IDisable	3. Clear
4. EoffsOff	14. oftDeact	. Return	4. IEnable	4. Decr
5. EoffsOn	15. VelSet	. ClkReset	5. ISignalDI	5. Incr
6. EoffsSet	16. ActUntil	. ClkStart	6. ISleep	
7. GripLoad	17. eactUntil	. ClkStop	7. ITimer	
8. LimConfL	18. SearchC	. Test	8. IWatch	
9. PdispOff	19. SearchL			
10. PdispOn	20. SpotL			

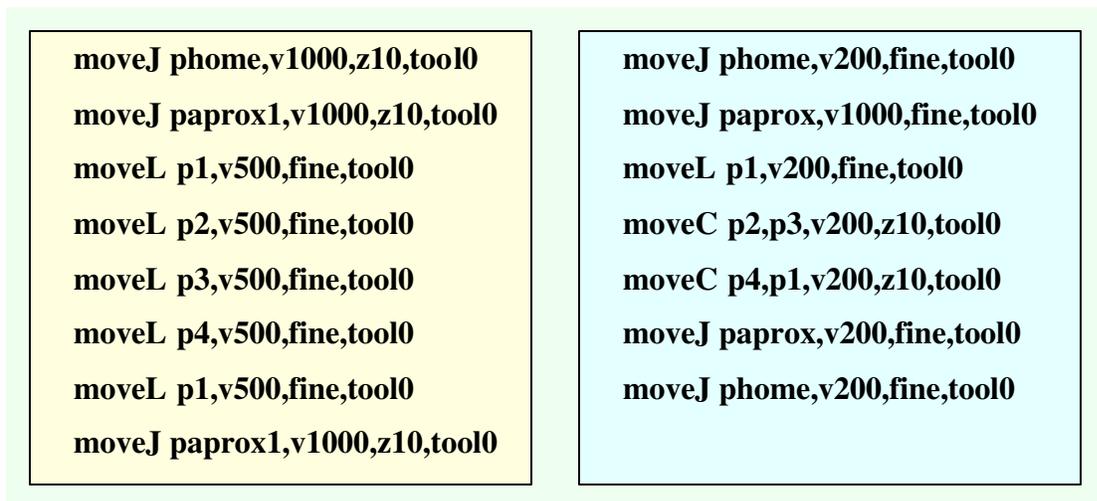
Figura AI.1: Instruções da linguagem de programação RAPID™ do robô ABB™.

A maioria dos robôs industriais disponíveis no mercado vem adotando linguagens com características semelhantes, com pequenas mudanças de sintaxe e forma de estruturação, de modo que os exemplos apresentados neste tópico poderão ser facilmente traduzidos para qualquer robô industrial que apresente configurações cinemáticas semelhantes ao robô utilizado - IRB 140 da ABB™ (robô 6R).

A programação de um robô industrial utiliza uma linguagem de programação direta, como por exemplo, a instrução dada pela linha de programação: **MoveL phome,v500,z5,tool0** significa que o robô está se movendo para a posição PHOME, a uma velocidade de 500 mm/s, com uma zona de precisão de 5mm, e como TCP, TOOL0.

Para a implementação de programas iniciais pelo usuário, sem conhecimento aprofundado em linguagens de programação estruturada, na implementação dos primeiros programas A figura AI.2 apresenta dois exemplos de programas implementados num Robô ABB 140TM, correspondentes à execução pela ferramenta de final de braço, de um quadrado e de um círculo. O usuário poderá implementar tarefas de um robô utilizando três tipos de comandos básicos para movimentação de um robô.

- i. **MoveJ:** é o movimento que o robô faz saindo de um ponto e chegando ao ponto de destino movendo todas as juntas necessárias e de qualquer forma para chegar, neste movimento obtém-se as maiores velocidades;
- ii. **MoveL** é o movimento utilizado para fazer trajetórias retas;
- iii. **MoveC** é o movimento utilizado para fazer trajetórias circulares.

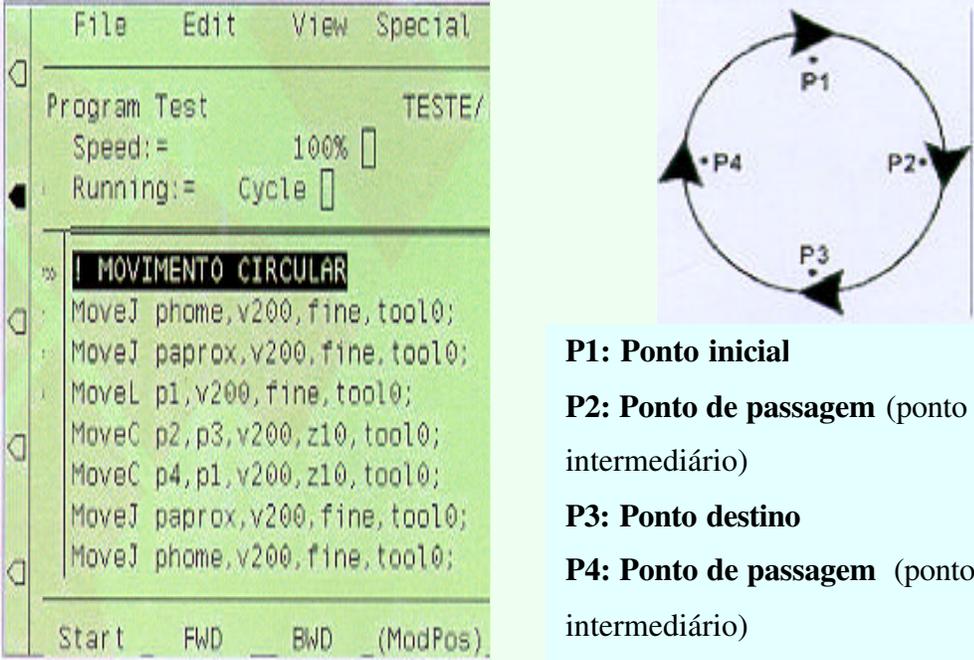


a) desenho de um quadrado

b) desenho de um círculo

Figura AI.2 Exemplos de programação.

Pode ser observado que a instrução *moveL* (movimento linear) o robô funciona com reorientação da ferramenta de trabalho na direção dos três eixos cartesianos (X,Y,Z) e sempre movimenta os eixos necessários para fazer uma trajetória reta, e na instrução *moveC* (movimento circular), o robô funciona com reorientação da ferramenta de trabalho em torno da rotação dos eixos X, Y, Z. Na figura AI.3 é apresentado um exemplo de implementação deste programa no robô (desenho de um círculo).



The image shows a screenshot of a robot programming interface on the left and a diagram of a circle on the right. The interface displays a program named 'Program Test' with the following code:

```
File Edit View Special
Program Test TESTE/
Speed:= 100%
Running:= Cycle
I MOVIMENTO CIRCULAR
MoveJ phome,v200,fine,tool0;
MoveJ paprox,v200,fine,tool0;
MoveL p1,v200,fine,tool0;
MoveC p2,p3,v200,z10,tool0;
MoveC p4,p1,v200,z10,tool0;
MoveJ paprox,v200,fine,tool0;
MoveJ phome,v200,fine,tool0;
Start FWD BWD (ModPos)
```

The diagram on the right shows a circle with four points labeled P1, P2, P3, and P4. Arrows indicate a clockwise path starting from P1, passing through P2, P3, and P4, and returning to P1.

P1: Ponto inicial
P2: Ponto de passagem (ponto intermediário)
P3: Ponto destino
P4: Ponto de passagem (ponto intermediário)
P1: Ponto destino (ponto final)

Figura AI.3 Exemplo de implementação do programa (desenho de um círculo).

Para ilustrar os comandos básicos apresentados anteriormente (moveJ, moveL, moveC), na figura AI.4, é mostrado a implementação de um programa completo para um desenho para o Robô ABB™.

```
moveJ phome,v1000,z10,tool0
moveJ paprox1,v1000,z10,tool0
moveL p1,v500,fine,tool0
moveL p2,v500,fine,tool0
moveL p3,v500,fine,tool0
moveL p4,v500,fine,tool0
moveL p1,v500,fine,tool0
moveJ paprox1,v1000,z10,tool0
moveJ paprox2,v1000,z10,tool0
moveL p5,v500,fine,tool0
moveL p6,v500,fine,tool0
moveL p7,v500,fine,tool0
moveL p8,v500,fine,tool0
moveL p5,v500,fine,tool0
moveJ paprox2,v1000,z10,tool0
moveJ paprox3,v1000,z10,tool0
moveL p9,v500,fine,tool0
moveC p10,p11,v500,z10,tool0
moveC p12,p9,v500,z10,tool0
moveJ paprox3,v1000,z10,tool0
moveJ phome,v1000,z10,tool0
```

Figura AI.4 Exemplo de Programação – Robô BB™.

AI.1 Programação textual baseada na estruturação

Para programação estruturada de um robô, o usuário deverá ter conhecimento das estruturas de programação e instruções básicas apresentadas a seguir:

- i. **MAIN ROTINE** : Todo o programa deverá ter uma rotina principal;
- ii. **IF – THEN**: Rotinas para tomada de decisões em função de condições lógicas.

Exemplo de Sintaxe de utilização:

IF condição **THEN**

Instrução, Programa, Condição Lógica, etc

ELSE

Instrução, Programa, Condição Lógica, etc.

ENDIF;

- iii. **FOR-ENDFOR**: *É usada* para repetir um dado número de vezes, uma ou diversas instruções.

Exemplo de Sintaxe de utilização:

FOR variável (**I**) **FROM** valor inicial **TO** valor final [**STEP** valor do passo] **DO**

Programa, Instrução, Condição Lógica, etc.

ENDFOR;

Observação:

O valor de “I” só pode ser lido por instruções dentro do comando **FOR**

O argumento **STEP** determina a quantidade de incremento ou decremento.

- iv. **TPREADNUM**: Rotinas de decisão utilizadas na leitura de um número no teach pendant, carregamento num registrador, para posterior utilização, como no caso de utilização junto com o comando *FOR*, para executar um número de vezes uma rotina ou sub rotina. A figura AI.5 apresenta um exemplo de utilização desta instrução.

```
PROC main()
  TPreadNum reg2,"Quantidade: ";
  FOR i FROM 1 TO reg2 DO
    MoveL phome,v400,z10,Bic;
    MoveL pprox1,v400,z10,Bic;
    .
    .
    .
    .
    MoveL p20,v400,fine,Bic;
  ENDFOR
ENDPROC
ENDMODULE
```

Figura AI.5 Exemplo de utilização da instrução TPREADNUM

- v. **TPWRITE**: comando utilizado para colocar uma mensagem, no início da operação com o robô.
- vi. **WAITTIME**: comando utilizado para dar um tempo (dado em segundos) antes ou depois de executar uma determinada instrução ou uma sub rotina.

vii. **TPERASE**: comando utilizado para limpar a tela do “teach-pendant”.

A figura AI.6 apresenta um exemplo de programação, utilizando as instruções TPWRITE, WAITTIME, TPERASE, para a execução de desenhos, correspondente a um retângulo, losango, círculo, pela ferramenta de trabalho.

```
PROC main()
  TPWrite "Oi Operador...";
  WaitTime 2;
  TPErase;
  TPReadFK reg1,"Escolha o desenho:","Quadrado","Losango","Circulo","Home",stEmpty;
  IF reg1=1 THEN
    quadrado;
  ELSE
    IF reg1=2 THEN
      losango;
    ELSE
      IF reg1=3 THEN
        circulo;
      ELSE
        MoveJ phome,v400,z50,Bic;
      ENDIF
    ENDIF
  ENDIF
ENDPROC
ENDMODULE
```

Figura AI.6 Exemplo de programação

Por outro lado, na figura AI.7, esse programa é apresentado sob a forma de uma rotina principal contendo três sub-rotinas: retângulo, losângulo, círculo.

```
PROC main()
  TPWrite "Oi Operador...";
  WaitTime 2;
  TPErase;
  TPWrite "O tempo foi de:\Num:=ClkRead(clock1);
  TPReadFK reg1,"Escolha o desenho:", "Quadrado", "Losango", "C írculo", "Home",stEmpty;
  ClkReset clock1;
  ClkStart clock1;
  IF reg1=1 THEN
    quadrado;
  ELSE
    IF reg1=2 THEN
      losango;
    ELSE
      IF reg1=3 THEN
        círculo;
      ELSE
        MoveJ phome,v400,z50,Bic;
      ENDIF
    ENDIF
  ENDIF
  ClkStop clock1;
ENDPROC
ENDMODULE
```

Figura AI.7 Exemplo de programação - Sub-rotinas

Anexo II – Exemplo de Aplicação de Programação “Off-Line”

A linguagem Karel 2, linguagem de programação do software Workspace® 4.01, permite a utilização das funções, FOR... ENDFOR, REPEAT...UNTIL, WHILE...ENDWHILE, estes comandos são encontrados em diversas linguagens de programação e permitem de forma distintas resultados semelhantes.

A sintaxe é detalhada a seguir:

- i. FOR... ENDFOR
- ii. REPEAT...UNTIL
- iii. WHILE...ENDWHILE

O processo em análise é um programa de soldagem e utiliza o mesmo bloco de movimentos do robô, que altera somente a posição de soldagem através do dispositivo de posicionamento, descritos nos programas das figuras AII.1, AII.2, AII.3, AII.4. Uma instrução de “looping“, atende de forma satisfatória a repetição do movimento do robô em um número conhecido de quatro repetições.

Os caracteres, -- ! representam instruções exclusiva do software de simulação, quando transferido para uma unidade de controle do robô, é entendido como um simples comentário. Dessa forma o software podem utilizar, funções de controle, comandos, etc... Exclusivos de sua sintaxe.

```
WITH $MOTYPE=LINEAR
  MOVE TO TP1
WITH $MOTYPE=LINEAR
  MOVE TO TP2
-- ! ARCWELDON 250,25
WITH $MOTYPE=LINEAR
  MOVE TO TP3
WITH $MOTYPE=LINEAR
  MOVE TO TP4
WITH $MOTYPE=LINEAR
  MOVE TO TP5
  ! ARCWELDOFF
WITH $MOTYPE=LINEAR
  MOVE TO TP6
```

Bloco de deslocamento e execução da 1ª soldagem, executando a aproximação, soldagem e afastamento da primeira divisória.

Figura AII.1 Bloco de deslocamento e a 1ª soldagem

A instrução --! TURNOBJ ‘MESA’, demonstrada na figura AII.2, exclusiva do software de simulação, possibilita a rotação do dispositivo de posicionamento “MESA”, para cada uma das posições desejada, o que leva a completar e executar o processo de cada uma das partes discretas.

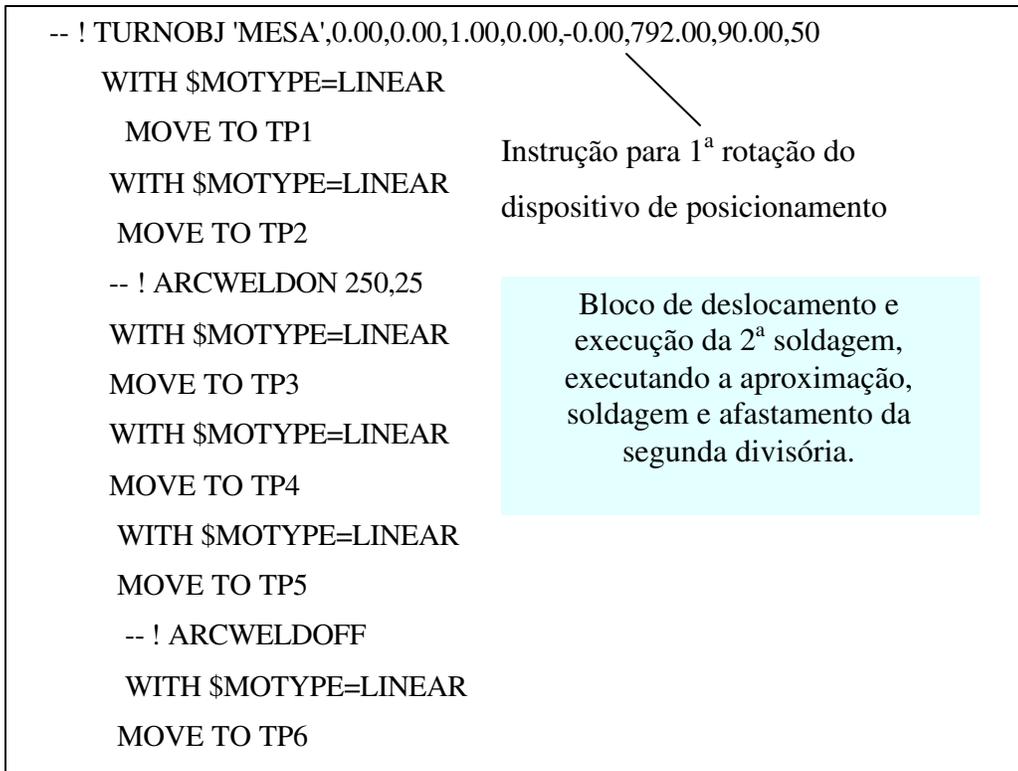


Figura AII.2 Bloco de deslocamento e a 2ª soldagem

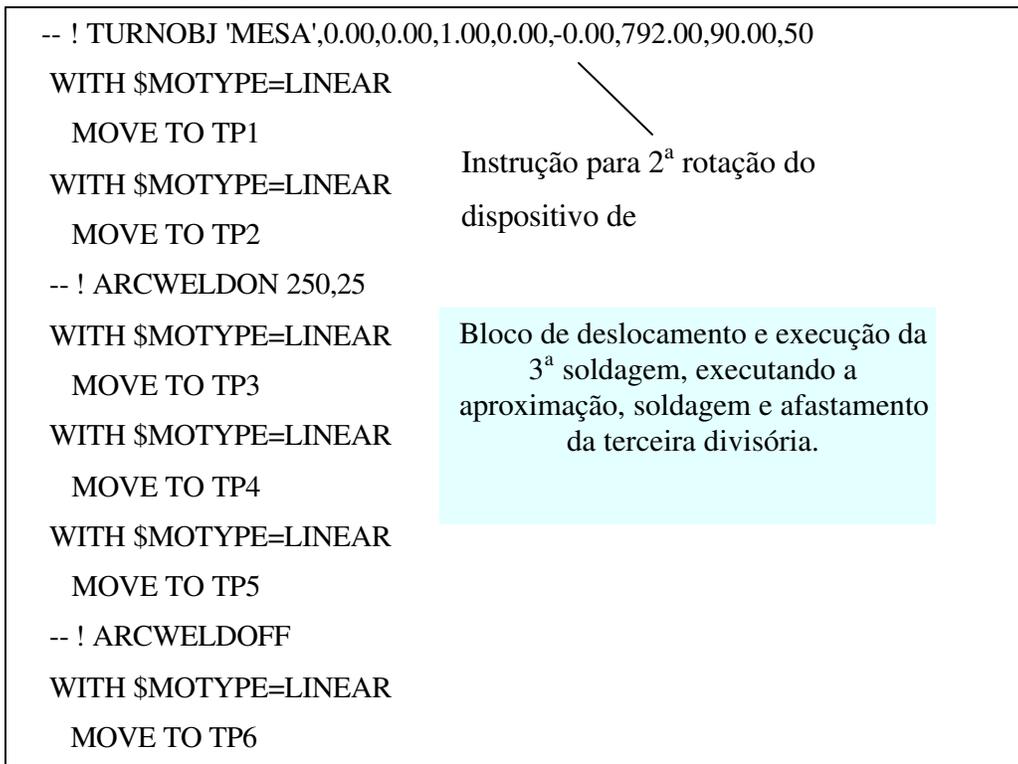


Figura AII.3 Bloco de deslocamento e a 3ª soldagem

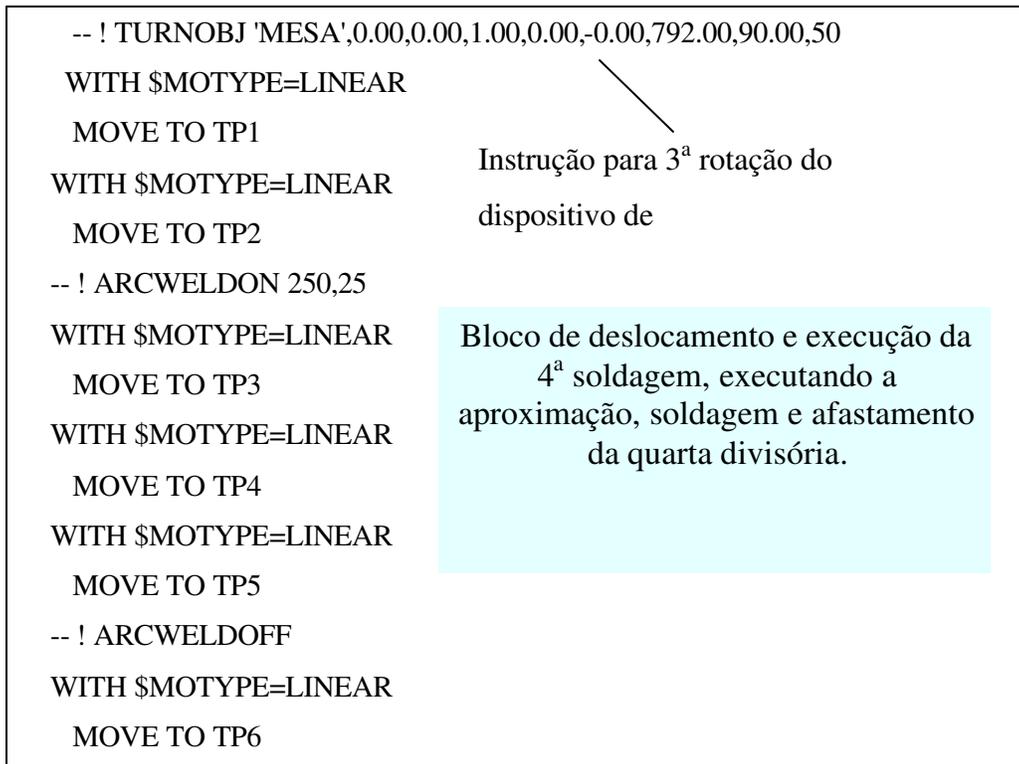


Figura AII.4 Bloco de deslocamento e a 4ª soldagem

Nas figuras AII.1, AII.2, AII.3, AII.4, as funções ARCWELDON e ARCWELDOFF ligam e desligam as ações de soldagem é interpretado com um comentário em qualquer unidade de controle de robôs.

As mudanças propostas são demonstradas, conforme seguem, nas figuras AII.5, AII.6 e AII.7.

```

VAR
  TP1 : POSITION
  TP2 : POSITION
  TP3 : POSITION
  TP4 : POSITION
  TP5 : POSITION
  TP6 : POSITION
VAR
  I: INTEGER      Declaração de variável
BEGIN
  I=0             Condição inicial de valor na variável
  $UTOOL=POS(-11.5265,2.7334,307.2149,0,45,-160,")
  $USEMAXACCEL=TRUE
  %INCLUDE TP#
  FOR I=0 TO 3 DO  Condição para execução do programa
  WITH $MOTYPE=JOINT
  MOVE TO TP1
  WITH $MOTYPE=LINEAR
  MOVE TO TP2
  -- ! ARCWELDON 300,3
  WITH $MOTYPE=LINEAR
  MOVE TO TP3
  WITH $MOTYPE=LINEAR
  MOVE TO TP4
  WITH $MOTYPE=LINEAR
  MOVE TO TP5
  -- ! ARCWELDOFF
  WITH $MOTYPE=LINEAR
  MOVE TO TP6
  -- ! TURNOBJ 'MESA',0.00,0.00,1.00,0.00,0.00,792.00,90.00,30
  I = I+1        Lógica para implementar a contagem para execução
ENDFOR

```

Figura AII.5 Utilização da Instrução For... EndFor

VAR

TP1 : POSITION

TP2 : POSITION

TP3 : POSITION

TP4 : POSITION

TP5 : POSITION

TP6 : POSITION

VAR

I: BOOLEAN Declaração de variável utilizada para a contagem

BEGIN

\$UTOOL=POS(-11.5265,2.7334,307.2149,0,45,-160,")

\$USEMAXACCEL=TRUE

%INCLUDE TP#

REPEAT Instrução a ser executada

WITH \$MOTYPE=JOINT

MOVE TO TP1

WITH \$MOTYPE=LINEAR

MOVE TO TP2

-- ! ARCWELDON 300,30

WITH \$MOTYPE=LINEAR

MOVE TO TP3

WITH \$MOTYPE=LINEAR

MOVE TO TP4

WITH \$MOTYPE=LINEAR

MOVE TO TP5

-- ! ARCWELDOFF

WITH \$MOTYPE=LINEAR

MOVE TO TP6

-- ! TURNOBJ 'CSG3',0.00,0.00,1.00,0.00,0.00,792.00,90.00,30

UNTIL FALSE Condição para execução

Figura AII.6 Utilização da Instrução Repeat... Until.

```

VAR
  TP1 : POSITION
  TP2 : POSITION
  TP3 : POSITION
  TP4 : POSITION
  TP5 : POSITION
  TP6 : POSITION
VAR
I : INTEGER Declaração de variável utilizada para a contagem
BEGIN
  $UTOOL=POS(-11.5265,2.7334,307.2149,0,45,-160,")
  $USEMAXACCEL=TRUE
  %INCLUDE TP#
  WITH $MOTYPE=JOINT
  MOVE TO TP1
WHILE I<4 DO Condição a ser executada, em função do valor
                                     armazenado na variável.
  WITH $MOTYPE=LINEAR
  MOVE TO TP2
  -- ! ARCWELDON 300,30
  WITH $MOTYPE=LINEAR
  MOVE TO TP3
  WITH $MOTYPE=LINEAR
  MOVE TO TP4
  WITH $MOTYPE=LINEAR
  MOVE TO TP5
  -- ! ARCWELDOFF
  WITH $MOTYPE=LINEAR
  MOVE TO TP6
  -- ! TURNOBJ 'CSG3',0.00,0.00,1.00,0.00,0.00,792.00,90.00,30
I = I+1           Variável utilizada como contador

```

Figura AII.7 Utilização da Instrução While... EndWhile

Neste exemplo de estruturação, é possível identificar o ganho na programação. O tamanho do programa e o tempo de programação é reduzido, ao analisar o tempo de programação textual e permite o mesmo resultado de trabalho dos equipamentos da célula robotizada.

Anexo III – Proposta de Formação Profissional em Robótica Industrial utilizando a Programação Off-Line.

Baseado na metodologia apresentada nos capítulos anteriores, este anexo dará ênfase numa proposta de formação profissional em Robótica, responsável pelo projeto de células robotizadas industriais.

A partir da utilização de conceitos de programação off-line, será possível a viabilização e exequibilidade de implementação dessa ferramenta e procedimentos em empresas de pequeno e médio porte, constituindo assim uma nova área de atuação profissional no ramo da Engenharia do século XXI.

AIII.1 A formação profissional

A partir de meados da década de 80, países como Austrália, Japão, Coréia do Sul, além de alguns países europeus, iniciaram a criação de cursos de graduação e pós-graduação voltados ao ensino multidisciplinar de Mecatrônica (Acar, 1997).

Nos Estados Unidos não foram criados cursos específicos de Engenharia Mecatrônica, porém foram introduzidas, nos currículos da maioria dos cursos de graduação em Engenharia,

disciplinas que apresentam o conceito de Mecatrônica (Ashley, 1997). Na grande maioria dos cursos de Engenharia Mecânica foram realizadas modificações e adaptações, adicionando disciplinas que abordam a integração de Mecânica, Eletrônica e Computação, para o desenvolvimento de componentes e máquinas.

Na Finlândia foi introduzido em 1987 um programa especial de pesquisa em Mecatrônica com a participação de quatro universidades tecnológicas. Este programa contou com um orçamento de 6,5 milhões de dólares até 1990, e a participação de aproximadamente 80 indústrias atuando em setores estratégicos (máquinas para fabricação de papel, telefonia móvel, máquinas florestais, robôs especiais) (Salminen, 1996). Este programa teve como principal objetivo difundir os conceitos de Mecatrônica nas indústrias e em 1995 um novo programa foi introduzido, com horizonte de quatro anos e um orçamento de 20 milhões de dólares, envolvendo universidades, centros de pesquisa e indústrias, com novos temas direcionados para a área de Mecatrônica.

Na Inglaterra, a comunidade envolvida com Mecatrônica só recebeu aceitação oficial em 1990 com a criação de um Fórum de Mecatrônica apoiado pelo IEE (Institute of Electrical Engineers) e o MechE (Institute of Mechanical Engineers) (Hewit, 1996).

AIII.2 A formação no Brasil

No Brasil, os primeiros cursos de formação profissional em Mecatrônica começaram a surgir no final da década de 80, sendo denominado pelo MEC como curso de Engenharia de Controle e Automação, sendo implementadas em diversas universidades, faculdades de tecnologia e cursos técnicos (Rosário, 2000).

As atividades de um Engenheiro de Controle e Automação (Mecatrônica) incluem a análise dos processos; projeto e dimensionamento, configuração, avaliação, segurança e manutenção dos sistemas de controle e automação; bem como os sistemas produtivos e das informações.

Atuará na interface entre o sistema produtivo e o sistema gerencial de empresas. A formação multidisciplinar nas áreas de mecânica, eletrônica, instrumentação industrial, informática, controle e gestão da produção permitirão ao profissional elaborar estudos e projetos, bem como participar da direção e fiscalização de atividades relacionadas com o controle de processos e a automação de sistemas industriais.

A característica predominante do Engenheiro de Controle e Automação é a sua formação generalista exigindo um esforço notável tanto do corpo docente quanto do corpo discente, mas que acabará em suprir uma lacuna que realmente existe no mercado, possibilitando uma realização profissional bastante interessante.

AIII.3 Alguns conceitos básicos de Mecatrônica

Atualmente a Mecatrônica é entendida como uma filosofia relacionada à aplicação combinada de conhecimentos de áreas tradicionais como a Engenharia Mecânica, Eletrônica e Computação de forma integrada e concorrente, conforme mostra a figura AIII.1 e a figura AIII.1a. Uma combinação para ser concorrente deve extrair o que há de mais adequado em cada uma das áreas, de tal forma que o resultado final é mais do que a simples soma de tais especialidades, mas sim uma sinergia entre elas.

O conceito de Mecatrônica representa a combinação adequada de materiais (resistência dos materiais, comportamento térmico, etc.), mecanismos (cinemática, dinâmica), sensores, atuadores, eletrônica e processamento digital (controle, processamento de sinais, simulação, projeto assistido por computador), possibilitando as seguintes características:

a) Desenvolvimento do projeto

- i. Simplificação do sistema mecânico;
- ii. Redução de tempo e de custo de desenvolvimento;

- iii. Facilidade de se introduzir modificações ou novas capacidades;
- iv. Flexibilidade para receber futuras modificações ou novas funcionalidades.

b) Desenvolvimento do produto

- i. Flexibilidade de operação: programabilidade;
- ii. Inteligência: capacidade para sensoriamento e processamento das informações, permitindo assim, a adaptação a diferentes condições de operação;
- iii. Automonitoramento e prevenção ativa de acidentes;
- iv. Autodiagnóstico em caso de falhas;
- v. Redução do custo de manutenção e consumo de energia;
- vi. Elevado grau de precisão e confiabilidade.

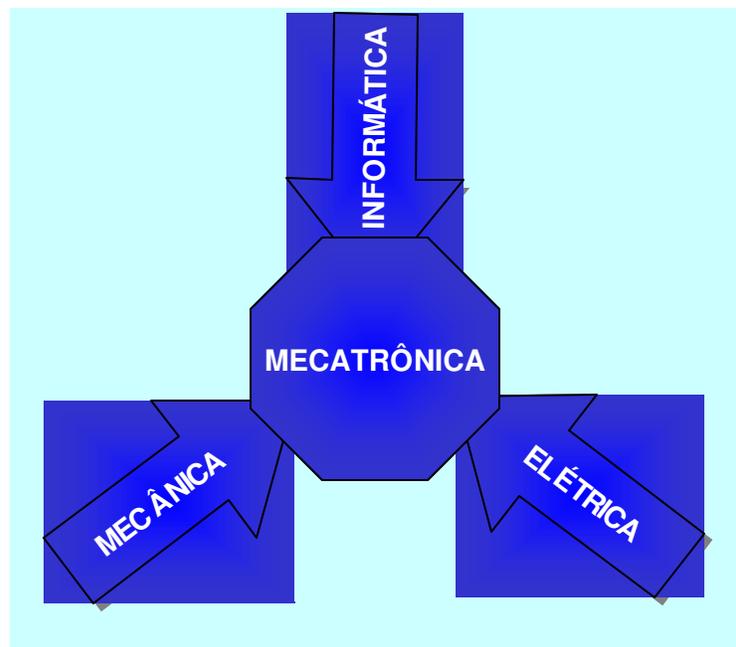


Figura AIII. 1 Definição de Mecatrônica

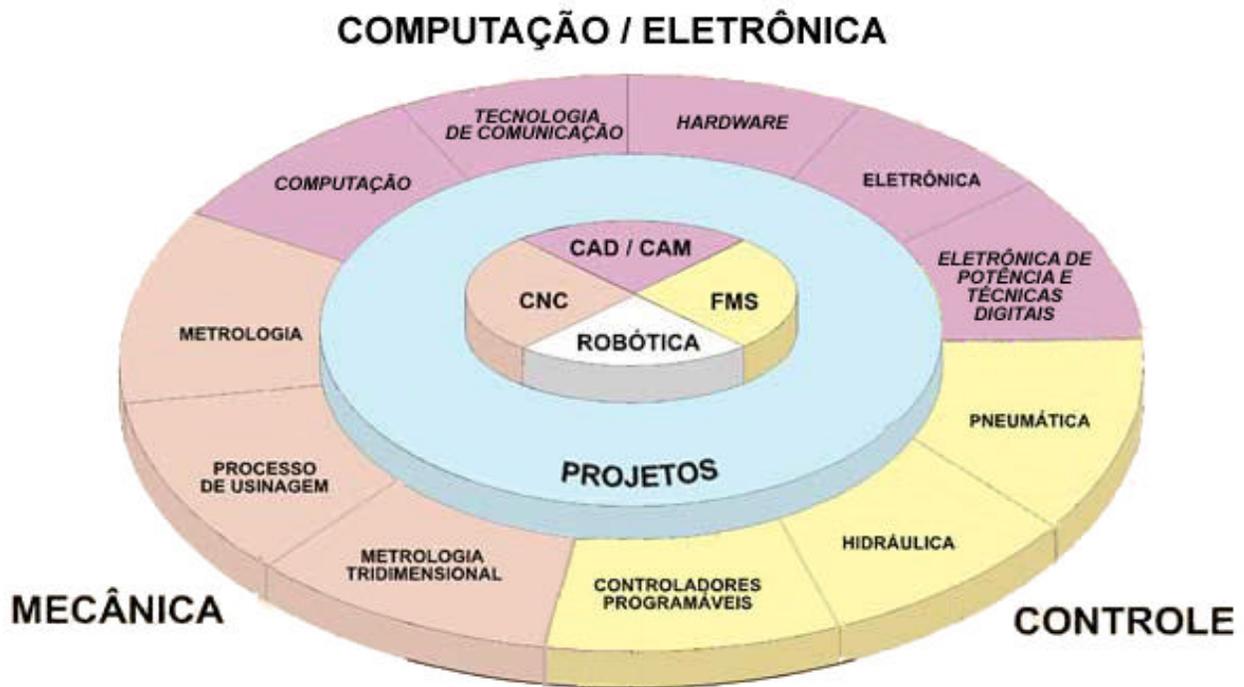


Figura AIII. 1a Definição de Mecatrônica nas Áreas de Automação.

É possível citar alguns exemplos de como esses resultados são possíveis, dentro da área de Automação Industrial. Sistemas tais como máquinas ferramentas e máquinas de manufatura em geral eram compostas por mecanismos para sincronização de movimentos e normalmente acionados por um só atuador (em geral, um motor elétrico). A grande complexidade dos mecanismos exigia precisão elevada para diminuir folgas e dispositivos de lubrificação para reduzir atritos. Essas máquinas sofreram um grande desenvolvimento, com a introdução do Controle Numérico Computadorizado (CNC), possibilitando a obtenção de peças com formas tridimensionais complexas. Os Controladores Lógicos Programáveis (CLP) possibilitaram grandes modificações na indústria com a automação de processos, melhorando o desempenho e a qualidade dos produtos.

A utilização de mecanismos flexíveis tem se tornado uma realidade e possibilita a eliminação de juntas articuladas, por exemplo, estruturas flexíveis podem ser controladas, através de sensores e atuadores montados ao longo dessas estruturas, passando a apresentar

comportamentos desejados, como por exemplo, maior rigidez e eliminação de modos de vibração.

As aplicações de computação em Engenharia Mecânica evoluíram a partir do início da década de 80 com a evolução vertiginosa do poder de processamento dos computadores, acompanhado por um imenso declínio de preços.

Antes disso, programas para análise estrutural, térmica ou fluída eram rodados em computadores tipo *main frame* com entrada de dados em cartões perfurados e saídas em forma de listagens. Atualmente esses programas de análise oferecem excelentes interfaces gráficas para o usuário, tanto relacionadas à entrada de dados como apresentação dos resultados. Hoje, modelos matemáticos sofisticados e cada vez mais complexos podem ser simulados mesmo em computadores pessoais.

AIII.4 Níveis de automação

Para alguns, Mecatrônica é o conceito de engenharia integrada que utiliza CAD e CAM para gerar um produto complexo como, por exemplo, um robô. Um engenheiro de produção, por outro lado, pode entender a Mecatrônica como sendo a implementação de um Sistema Flexível de Manufatura (FMS). Um engenheiro, ao projetar uma câmara de vídeo, pode entendê-la como a utilização de eletrônica numa aplicação Mecânica. Já um engenheiro químico pode entender a Mecatrônica como o controle de um processo químico utilizando sensores e atuadores, controlados por um processador digital. Provavelmente todos estão corretos, pois a Mecatrônica está presente em diferentes níveis, tais como:

- i. *Componente* (por exemplo, circuitos integrados, sensores, atuadores, mecanismos);
- ii. *Máquina* (máquinas de usinagem, medição, inspeção, movimentação, embalagem);
- iii. *Sistema* (FMS – Sistemas Flexíveis de Manufatura, FAS – Sistema de Automação de Fabrica, CIM – Sistemas Integrados de Manufatura).

A atuação profissional nos diferentes níveis está relacionada com o grau de compreensão exigido para os fenômenos físicos envolvidos: quanto mais próximo maior deve ser o domínio sobre eles. O nível de componente exige o maior grau de domínio, enquanto que o nível de sistema requer o menor. Desta forma, conforme nos distanciamos do nível físico, diminui-se a complexidade física envolvida devido ao aumento do nível de abstração. Por outro lado aumenta também as complexidades lógicas do sistema, exigindo maior poder de processamento para lidar com uma maior quantidade de informação (figura AIII.2).

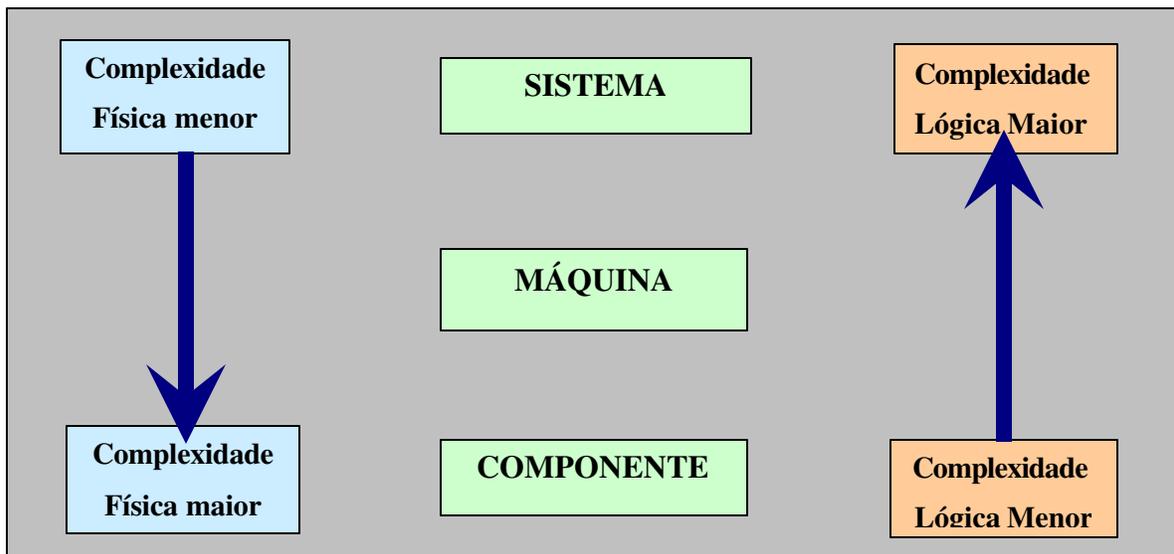


Figura AIII.2 Complexidade de um Sistema Mecatrônico.

No caso de um sensor de temperatura, por exemplo, precisamos ter conhecimento dos fenômenos físicos que podem ser utilizados para realizar a medida (variação de resistência, dilatação térmica, junção termopar, etc), as vantagens e desvantagens de cada um, as condições em que a medida deverá ser feita (tempo de resposta, faixa de temperatura, precisão e condições ambientais adversas), e a eletrônica necessária para condicionar o sinal e permitir a sua leitura.

No caso extremo do projeto de um sensor desse tipo, a informação desejada é o valor real de uma temperatura, e seu processamento envolve o tratamento para um sinal elétrico. Num outro extremo, um Sistema de Automação de Fábrica (FAS) deve lidar com informações bastante

abstratas, tais como adequação de estoques, capacidade produtiva das máquinas, previsões de demanda, escalas de manutenção, possibilidade de falhas, limites de consumo de energia, etc.

A geração de um planejamento otimizado de produção (o que produzir, quando e como) e o posterior controle da produção (com correções ocorrendo ao longo do trabalho) exigem o conhecimento preciso e instantâneo de todas estas variáveis e de muitas outras mais, além de envolver algoritmos sofisticados para tomada de decisões.