

ESTE EXEMPLAR CORRESPONDE A REDAÇÃO FINAL DA  
TESE DEFENDIDA POR *Luís Gustavo de*  
*Mello Paracêncio*..... E APROVADA  
PELA COMISSÃO JULGADORA EM 31 / 07 / 2009  
*Helder Aníbal Hermini*  
ORIENTADOR

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

## **Proposta de Metodologias para Integração de Células de Manufatura**

Autor: Luís Gustavo de Mello Paracêncio  
Orientador: Prof. Dr. Helder Aníbal Hermini  
Co-orientador: Prof. Dr. João Maurício Rosário

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

# **Proposta de Metodologias para Integração de Células de Manufatura**

Autor: Luís Gustavo de Mello Paracêncio  
Orientador: Prof. Dr. Helder Aníbal Hermini  
Co-orientador: Prof. Dr. João Maurício Rosário

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

Tese de doutorado apresentada à comissão de Pós Graduação da Faculdade de Engenharia Mecânica, como requisito para a obtenção do título de Doutor em Engenharia Mecânica.

Campinas, 2009  
S.P. - Brasil

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

P21p Paracêncio, Luís Gustavo de Mello  
Proposta de metodologias para integração de células de  
manufatura / Luís Gustavo de Mello Paracêncio. --  
Campinas, SP: [s.n.], 2009.

Orientadores: Helder Anibal Hermini, João Maurício  
Rosario.

Tese de Doutorado - Universidade Estadual de  
Campinas, Faculdade de Engenharia Mecânica.

1. Sistemas flexíveis de fabricação. 2. Robótica. 3.  
Manipuladores (Mecanismo). 4. Células de fabricação. 5.  
Experiência. I. Hermini, Helder Anibal. II. Rosario, João  
Maurício. III. Universidade Estadual de Campinas.  
Faculdade de Engenharia Mecânica. IV. Título.

Título em Inglês: Proposal for a methodology for integration of cellular manufacturing

Palavras-chave em Inglês: Flexible Manufacturing System, Robotics, Manipulators  
(Mechanism), Manufacturing cells, Experience

Área de concentração: Mecânica dos Sólidos e Projeto Mecânico

Titulação: Doutor em Engenharia Mecânica

Banca examinadora: Auteliano Antunes dos Santos Junior, Antônio Batochhio, Humberto  
Ferasoli Filho, Renato Ventura Bayan Henriques

Data da defesa: 31/07/2009

Programa de Pós Graduação: Engenharia Mecânica

UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA MECÂNICA  
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA  
DEPARTAMENTO DE PROJETO MECÂNICO

TESE DE DOUTORADO

**Proposta de Metodologias para Integração  
de Células de Manufatura**

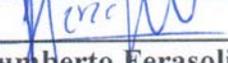
Autor: Luís Gustavo de Mello Paracêncio  
Orientador: Prof. Dr. Helder Aníbal Hermini  
Co-orientador: Prof. Dr. João Maurício Rosário

A Banca Examinadora composta pelos membros abaixo aprovou esta Tese:

  
\_\_\_\_\_  
Prof. Dr. João Maurício Rosário  
Universidade Estadual Paulista - UNICAMP/ DPM/ FEM

  
\_\_\_\_\_  
Prof. Dr. Auteliano Antunes dos Santos Junior  
Universidade Estadual Paulista - UNICAMP/ DPM/ FEM

  
\_\_\_\_\_  
Prof. Dr. Antônio Batocchio  
Universidade Estadual Paulista - UNICAMP/ DEF/ FEM

  
\_\_\_\_\_  
Prof. Dr. Humberto Ferasoli Filho  
Universidade de São Paulo - UNESP/ BAURU/ FC

  
\_\_\_\_\_  
Prof. Dr. Renato Ventura Bayan Henriques  
Universidade Federal do Rio Grande do Sul - UFRGS/ DEE

Campinas, 31 de julho de 2009.

## **Dedicatória**

Dedico este trabalho de pesquisa aos meus pais pelo amor, apoio e incentivo, sem os quais não teria chegado até aqui.

## **Agradecimentos**

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

A DEUS, pela vida e por me dar forças para realizar este trabalho.

Aos meus pais Luís Adolfo e Lucy, irmãos Carolina e Ricardo pelo incentivo em todos os momentos da minha vida.

Ao Professor Dr. Helder Anibal Hermini e Professor Dr. João Maurício Rosário pela qualidade da orientação fornecida e amizade construída ao longo destes anos.

A todos meus amigos do Laboratório de Robótica Integrada e Robótica, que com sugestões, auxílios, críticas e incentivos, contribuíram de maneira fundamental para a realização desta etapa. Em especial para Engenheiro Almiro Franco da Silveira Júnior, Álvaro Joffre Uribe Quevedo Msc.

A todos os colegas, professores e funcionários do Departamento de Projeto Mecânico e demais departamentos da UNICAMP, que ajudaram de forma direta e indireta na conclusão deste trabalho. Denise Aparecida Villela, Gastão Bosco Rodrigues, Isabela Rodrigues de Araujo.

A Coordenadoria de Aperfeiçoamento de Pessoal de Nível Superior - CAPES, pela bolsa de estudos concedida.

*Tudo posso naquele que me fortalece.*

Filipenses 4.13

## Resumo

PARACÊNCIO, Luís Gustavo de Mello, “Proposta de Metodologias para Integração de Células de Manufatura”. Campinas, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2009, 137 páginas. Tese de Doutorado.

Com o avanço tecnológico na área de engenharia mecatrônica, cada vez mais é necessário acompanhar e aperfeiçoar os estudos. Este trabalho pretende apresentar metodologias para integração de células de manufatura robotizadas com ênfase na modelagem e controle de um dispositivo robótico.

O modelo do dispositivo em estudo é composto de três graus de liberdade acionados hidraulicamente, o que permite o posicionamento de uma mesa, onde uma peça possa ser trabalhada por dois robôs industriais, constituindo assim um sistema colaborativo de manufatura. São apresentados estudos referentes à modelagem cinemática e dinâmica, e ao cálculo dos parâmetros do sistema de controle utilizando MatLab-Simulink™. Foi desenvolvida uma interface em linguagem LabVIEW™ para aquisição e tratamento de informações provenientes dos sensores das juntas e da implementação do sistema de supervisão e controle.

A partir da publicação de uma página de aplicação *HTML*, a célula colaborativa poderá ser disponibilizada na *WEB*, permitindo a criação de um laboratório virtual, direcionado à pesquisa científica e tecnológica e com possibilidade de se interligar com outros laboratórios de ensino e pesquisa. Isso permitirá, por exemplo, a realização de experiências e ensino à distância, executando tarefas complexas em tempo real.

### *Palavras-Chaves:*

- Sistema Colaborativos, Células Robóticas, Experiências à Distância.

## **Abstract**

PARACÊNCIO, Luís Gustavo de Mello, “Proposal for a Methodology for Integration of Cellular Manufacturing”. Campinas, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2009, 137 pages. These (Doctorate).

In recent years, with technological advances in mechatronics engineering it is necessary to monitor and improve the studies of these new trends. This research aims to present a methodology to integrate robotic manufacturing cells with emphasis on modeling and controlling of a robotic device.

The model of the device of this study consists of three degrees of freedom, driven hydraulically allowing the placement of a table where a piece can be worked by two industrial robots constituting a system of collaborative manufacturing. Studies are presented concerning kinematic and dynamic models and the calculation of parameters of the control system using MatLab-Simulink™. An interface was developed in LabVIEW™ language for acquisition and processing of the information from the sensors of the joints and the implementation of the system of supervision and control.

Since the publication of an HTML page, a cell collaborative application may be available on the collaborative WEB allowing the creation of a virtual laboratory directed to scientific and technological research and the possibility to connect with other laboratories for teaching and research. For instance, this will allow carrying out the implementation of distance learning experience and performing complex tasks in real time.

### *Key Words:*

- Collaborative systems, robotics workcell, e-learning.



2.4.1 Prototipagem Rápida em Mecânica .....	27
2.4.2 Prototipagem Rápida em Eletrônica .....	28
2.4.3 Prototipagem Rápida em Célula Flexível de Manufatura .....	30
2.4.4 Prototipagem Rápida em Instrumentação Virtual .....	31
2.4.5 Concepção de Sistemas de Supervisão e Controle baseados em Instrumentação Virtual .....	31
2.4.6 Concepção de Módulos de Supervisão e Controle baseados em Instrumentação Virtual .....	32
2.5 Sistemas Supervisórios e Colaborativos .....	34
2.5.1 Ferramentas para Supervisão - Supervisão e Controle de Células de Manufatura Integradas .....	35
2.5.2 Plataforma Colaborativa - Projeto Kyatera .....	37
2.5.2.1 Projeto Kyatera (Laboratório Virtual em Automação) .....	38
2.5.2.2 Arquitetura <i>WEB</i> Proposta para o Projeto Kyatera .....	39
2.6 Considerações finais .....	41

### **CAPÍTULO 3**

DISPOSITIVO MECATRÔNICO - PLATAFORMA ROBÓTICA DE 3 GRAUS DE LIBERDADE .....	42
3.1 Descrição dos Equipamentos da Plataforma Hidráulica .....	43
3.2 Descrição da Plataforma .....	45
3.3 Modelagem Matemática .....	46
3.3.1 Modelagem Cinemática .....	47
3.3.2 Cinemática Direta .....	48
3.3.3 Cinemática Inversa .....	50
3.4 Simulador Virtual .....	51
3.4.1 Módulo Geração de Trajetória .....	52
3.4.2 Módulo Acionamento .....	52
3.4.3 Módulo Cinemático .....	54
3.4.4 Interface Gráfica .....	55
3.5 Conclusão .....	61

## **CAPÍTULO 4**

### **VALIDAÇÃO EXPERIMENTAL: IMPLEMENTAÇÃO DE CÉLULA AUTOMATIZADA COLABORATIVA ..... 62**

4.1 Descrição da Célula Robótica em Estudo .....	63
4.2 Caso 1: Sistema Colaborativo em Automação .....	64
4.2.1 Ambiente de Validação .....	64
4.2.2 Ambientes Colaborativos para Ensino e Pesquisa baseados em <i>WEB</i> .....	66
4.2.3 Telas Implementadas em Ambiente LabVIEW™ .....	67
4.3 Caso 2: Implementação de Sistema de Supervisão e Controle da Plataforma 3 GDL .....	68
4.3.1 Descrição da Parte Operativa .....	70
4.3.2 Sistema de Supervisão e Controle .....	72
4.3.3 Descrição do Sistema Operativo .....	72
4.3.4 Interface de Controle .....	73
4.3.5 Telas Implementadas .....	74
4.3.5.1. Tela de Inicialização .....	74
4.3.5.2. Modo Calibração .....	75
4.3.6 Geração de Arquivos .....	78
4.3.6.1 Modo Aprendizado .....	78
4.3.6.2 Movimentação a partir de arquivo padrão EXCEL™ .....	80
4.3.6.3 Modo Controle .....	80
4.3.6.4 Tela de Controle (modo automático) .....	81
4.3.6.5 Tela de Status .....	82
4.3.6.6 Tela de Monitoramento .....	83
4.3.6.7 Tela de Ajuste de Parâmetros do Controlador de Posição PID .....	83
4.3.6.8 Modo de Segurança .....	84
4.3.6.9 Geração de Arquivo de Banco de Dados .....	86
4.4 Conclusão .....	86

## **CAPÍTULO 5**

### **CONCLUSÕES ..... 88**

<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>91</b>
<b>ANEXO A - GRAFCET FUNCIONAL .....</b>	<b>98</b>
<b>ANEXO B - CONCEITOS DOS SENSORES .....</b>	<b>110</b>
<b>ANEXO C - PROGRAMAÇÃO ESTRUTURADA UTILIZADA NOS ROBÔS INDUSTRIAIS .....</b>	<b>112</b>
<b>ANEXO D - INTRODUÇÃO AO LabVIEW .....</b>	<b>126</b>
<b>ANEXO E - ARTIGOS PUBLICADOS .....</b>	<b>137</b>

## Lista de Figuras

Figura 1.1: Integração de Célula de Manufatura Flexível.....	03
Figura 1.2: Laboratório de Automação e Manufatura (Boaretto, Sanzovo e Scandelari, 2004) .	04
Figura 1.3: Célula de Paletização (Pinotti e Amaral, 2006) .....	04
Figura 1.4: Braço Medidor ITG ROMER e Manipulador (Campos, 2006) .....	05
Figura 1.5: Manipulador Robótico - Ferramenta para Soldagem (Godinho e Pires, 2006) .....	05
Figura 1.6: Integração de Célula de Manufatura - LAR .....	07
Figura 2.1: Controle, Supervisão e Gerenciamento Virtual de um Dispositivo Mecatrônico ....	12
Figura 2.2: Exemplo de uma Célula de Trabalho .....	14
Figura 2.3: Programação <i>off-line</i> - Identificação de Colisões .....	17
Figura 2.4: Ferramentas Dedicadas .....	18
Figura 2.5: Utilização de Redes de Petri no Modelamento e Análise de Sistemas .....	25
Figura 2.6: Concepção de um Produto Desenvolvido em CAD (Sánchez, 2008) .....	28
Figura 2.7: <i>Hardware</i> In the Loop através do Mathworks® .....	30
Figura 2.8: Prototipagem de uma Célula Flexível de Manufatura (Freitas, 2004) .....	30
Figura 2.9: Integração de Diferentes tipos de <i>Hardware</i> em uma mesma Aplicação ( <i>National Instruments</i> ) .....	33
Figura 2.10: Integração em um mesmo <i>Hardware</i> de Diferentes Tipos de Aplicações ( <i>National Instruments</i> ) .....	33
Figura 2.11 Representação de um Sistema Supervisório .....	35
Figura 2.12 Sistema de Controle - Níveis .....	36
Figura 2.13: Arquitetura Proposta para Funcionamento do <i>WebLab</i> .....	39
Figura 2.14: Proposta de Implementação - Projeto Kyatera .....	40
Figura 3.1: Célula Flexível de Manufatura .....	43

Figura 3.2: Plataforma Hidráulica Robótica para Posicionamento .....	43
Figura 3.3: Reservatório Hidráulico .....	44
Figura 3.4: Célula Flexível de Manufatura .....	45
Figura 3.5: Mesa Robótica de Posicionamento .....	46
Figura 3.6: Estrutura de Controle .....	47
Figura 3.7: Plataforma 3GDL e Movimentos Relacionados .....	47
Figura 3.8: Cinemática Direta e Inversa de Robôs Manipuladores .....	48
Figura 3.9: Esquema da Mesa com 3 GDL .....	48
Figura 3.10: Cone de Rotação .....	50
Figura 3.11: Simulador de uma Mesa de 2 GDL usando Ambiente Simulink® .....	52
Figura 3.12: Sinal de Referência de Posição para Cada Junta .....	53
Figura 3.13: Malha de Controle de Posição de uma Junta .....	54
Figura 3.14: Função S-function - Matlab® Implementada nos Blocos do Simulink® .....	54
Figura 3.15: <i>Menu</i> do Simulador Robótico com 2 GDL .....	55
Figura 3.16: Haste colocada sobre a Mesa para Visualização dos Movimentos durante as Simulações .....	56
Figura 3.17: Sistema de Posicionamento .....	56
Figura 3.18: Perfil Velocidade das Juntas - Apenas Atuador 2 Acionado .....	57
Figura 3.19: Movimento Cinemático - Apenas a Junta 2 Acionada .....	57
Figura 3.20: Movimento Dinâmico - Apenas a Junta 2 Acionada .....	58
Figura 3.21: Sinal do Controle .....	58
Figura 3.22: Movimento do Centro da Mesa - Apenas o Atuador 2 Acionado .....	59
Figura 3.23: Vista dos Eixos X, Y e Z da Mesa em Movimento - Movimento Espacial da Haste .....	59
Figura 3.24: Posição Espacial da Haste - Acionamento somente do Atuador 2 .....	60
Figura 3.25: Erro do Atuador .....	60
Figura 3.26: Erro da Mesa .....	61
Figura 4.1: Célula Automatizada Cooperativa Virtual, a ser Implementada na FEM-UNICAMP ) .....	63
Figura 4.2: Plataforma de Posicionamento - LAR .....	64
Figura 4.3: Estruturação de Tarefas .....	65

Figura 4.4: Célula Integrada de Trabalho - LAR .....	66
Figura 4.5: Aspectos de um Ambiente de Ensino e Pesquisa Virtual .....	67
Figura 4.6: Proposta de pagina HTML - Painel de Controle Implementado em LabVIEW .....	67
Figura 4.7: Tela típica de Implementação em LabVIEW .....	68
Figura 4.8: Tela de Visualização de Câmera CCD <i>WEBCAM</i> com IP fixo .....	68
Figura 4.9: Célula Integrada de Manufatura - LAR .....	70
Figura 4.10: Esquema do Sistema Controlado .....	71
Figura 4.11: Interface de Comunicação <i>PCI-9112 16-CH 12-Bit 110 kS/s Multi-Function DAQ Card/ Low-Profile DAQ Card</i> da <i>NI Instrumentation</i> .....	73
Figura 4.12: Tela Principal do Programa .....	74
Figura 4.13: Telas do Programa de Calibração .....	75
Figura 4.14: Tela de Calibração da Haste .....	76
Figura 4.15: Mensagens Típicas do <i>Menu</i> de Calibração .....	77
Figura 4.16: Tela gerada após Validação da Calibração .....	77
Figura 4.17: Tela de operação- Geração de arquivos de trajetórias .....	79
Figura 4.18: Tela típica do Módulo de Controle .....	81
Figura 4.19: Tela de Controle (Modo Automático) .....	82
Figura 4.20: Telas Típicas de Status de Evolução da Trajetória .....	82
Figura 4.21: Tela de Monitoramento dos Sensores das Juntas (Tensão x Tempo) .....	83
Figura 4.22: Tela de Ajuste de Parâmetros do Controlador PID .....	84
Figura 4.23: Tela de Alarme de Funcionamento (Modo Segurança) .....	85
Figura 4.24: Banco de Dados (Arquivos de Movimentação) .....	86

## **Nomenclatura**

### ***Abreviações***

CAD - *Computer-Aided Design*

CAE - *Computer Aided Engineering*

CAM - *Computer Aided Manufacturing*

CIM - *Computer-Integrated Manufacturing*

CLP - Controladores Lógicos Programáveis

DPM - Departamento de Projeto Mecânico

FEM - Faculdade de Engenharia Mecânica

FMC - *Flexible Manufacturing System*

FMS - *Flexible Manufacturing System*

FPGA - *Field Programmable Gate Array*

GDL - Graus de Liberdade

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

HIL - *Hardware in the Loop*

HTML - *HyperText Markup Language*

IHM - Interface Homem Máquina

JIG's - Dispositivos de Fixação e Posicionamento

LabVIEW - *Laboratory Virtual Instrument Engineering Workbench*

LAR - Laboratório de Automação Integrada e Robótica

LAN - *Local Area Network*

LD - *Ladder Diagram*

MCFMS - *Multi Celd Flexible Manufacturing System*  
MMFMS - *Multi Machine Flexible Manufacturing System*  
NI - *National Instruments*  
PAC - *Programmable Automation Controller*  
PC - *Personal Computer*  
PCI - *Peripheral Component Interconnect*  
PO - *Parte Operativa*  
PRR - *Prismático Rotacional Rotacional*  
PXI - *PCI extensions for Instrumentation*  
RdP - *Redes de Petri*  
S.A. - *Sistemas Automatizados*  
SCM - *Sistema Controlado e Monitorado*  
SED - *Sistemas a Eventos Discretos*  
VI - *Virtual instruments*

# **CAPÍTULO 1**

## **INTRODUÇÃO**

Com rápido avanço das tecnologias e a crescente demanda por soluções que as acompanham, criou-se uma necessidade no mercado tecnológico de reduzir o tempo de implementação de um projeto, seja ele voltado a uma melhoria em um processo de produção ou na área acadêmica.

O desenvolvimento tecnológico permitiu uma melhoria nos processos de produção, com o aumento da quantidade produzida, como também, na qualidade do produto. A partir daí, surgiu uma importante área multidisciplinar com a integração de varias ciências, a qual se denominou Mecatrônica, que integra conhecimentos nas diferentes áreas de engenharia, matemática, computação, permitindo o desenvolvimento com maior rapidez de sistemas de uma estrutura mecânica, (como exemplo, a prototipagem rápida), composta por atuadores, sensores de posicionamento e um sistema de acionamento e controle, inserindo dentro desse contexto a robótica.

Nos dias atuais dispositivos robóticos podem ser encontrados nas mais diversas aplicações, desempenhando funções diversificadas, atuando em ambientes hostis, educacionais, ou simplesmente em trabalhos que requerem força e agilidade. Um robô industrial (manipulador) é composto de uma estrutura mecânica, com acoplamento dinâmico entre as articulações. Cada grau de liberdade é constituído por atuadores, sensores de posicionamento e pelo sistema de acionamento e controle. A robótica industrial possibilita que produtos sejam realizados de maneira mais rápida e eficiente do que no passado (Rosário, 2004).

Com o atual cenário tecnológico e o alto desenvolvimento de soluções cada vez mais integradas para concepção e validação de dispositivos mecatrônicos, a utilização de ferramentas dedicadas e a integração de células de manufatura para Instrumentação e Controle Industrial torna-se uma indispensável opção para agilizar e facilitar a implementação de diferentes arquiteturas, possibilitando assim a redução do tempo de finalização de um projeto e ou estudo científico, atendendo uma eficiência desejada.

O estudo de novas arquiteturas de supervisão e controle torna-se importante para que haja uma melhoria na integração de robôs industriais em células automatizadas de manufatura face às mudanças bruscas de parâmetros associados à posição, velocidade e aceleração durante a realização de uma determinada trajetória. A Automação Flexível possibilita que robôs sejam cada vez mais rápidos podendo interferir em um aumento do volume de trabalho.

Nos dias atuais, a indústria segue a tendência de flexibilizar os sistemas de manufatura a partir da utilização de ferramentas como os robôs industriais. Quando se fala no assunto, deve-se considerar, entre outros recursos, a precisão e o controle de velocidade de movimento nas atividades executadas, além da considerável capacidade de memória de programa. No entanto, tal tecnologia é de difícil alcance para a maioria das indústrias de pequeno e médio porte. As atividades realizadas por estas indústrias não permitem grandes investimentos tecnológicos, pois a aquisição de robôs acabaria encarecendo o processo produtivo, levando as empresas a problemas financeiros.

Assim, se torna necessário fabricar robôs com maior simplicidade nos recursos tecnológicos, isto é, robôs dedicados às atividades específicas em determinados pontos da linha de produção, os quais, por terem baixo custo, são mais acessíveis às pequenas e médias empresas. Ressalta-se ainda que a aquisição destes “robôs dedicados” favorece a inserção de automação dentro da linha produtiva, o que requer baixo custo de investimento. Como exemplo, a automação de guinchos pode ser uma das formas favoráveis à produção dos “robôs dedicados”.

Este capítulo apresenta um panorama geral do trabalho desenvolvido, sendo mostrado um estado da arte do conceito de Sistemas Colaborativos em Automação, com ênfase na área de

integração de sistemas, justificando o desenvolvimento do projeto para modelagem e controle de uma plataforma mecatrônica de três graus de liberdade, cujo seus objetivos gerais e específicos, a necessidade de capacitação nesta importante área e a forma como foi delimitada esta questão, além da estrutura geral dos capítulos da tese.

## 1.1 Motivação

Com o atual cenário tecnológico e o alto desenvolvimento de soluções cada vez mais integradas para concepção e validação de dispositivos mecatrônicos, a utilização de ferramentas dedicadas e integração de células de manufatura para Instrumentação e Controle Industrial torna-se indispensável opção para agilizar e facilitar a implementação de diferentes arquiteturas, possibilitando assim a redução do tempo de finalização de um projeto ou estudo científico, atendendo a eficiência desejada, conforme a figura 1.1.

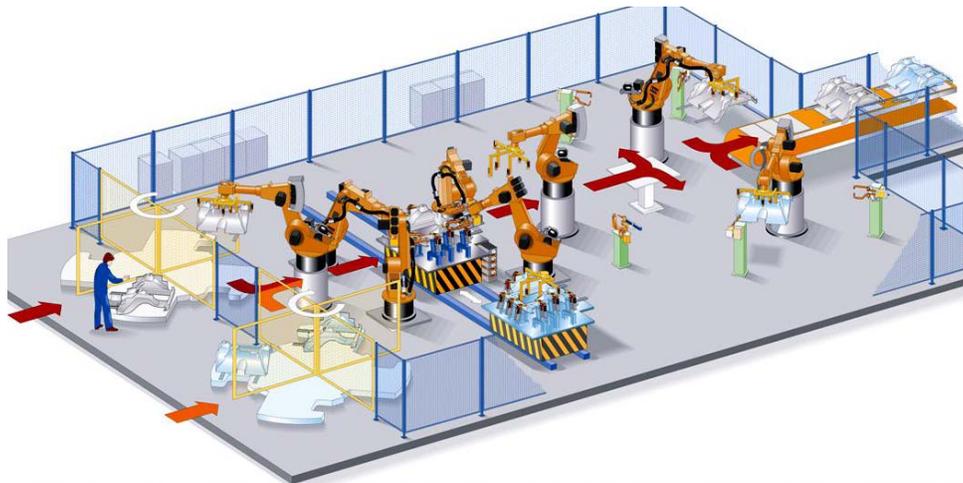


Figura 1.1: Integração de Célula de Manufatura Flexível

Diversos autores apontam a necessidade do desenvolvimento de metodologias adequadas para a concepção e implementação de projetos na área de integração de dispositivos automatizados. Diversos autores apresentam os estudos para a integração de uma célula flexível de manufatura na qual será de importante para a elaboração deste trabalho.

Boaretto, Sanzovo e Scandelari (2004) apresentam (figura 1.2) o laboratório de automação que constitui um sistema de manufatura de pequeno porte formado por uma célula de trabalho, cujo objetivo é confeccionar peças em tamanho reduzido com diversas aplicações industriais. Sua utilização está voltada para interesses didáticos, de pesquisa e de transferência de tecnologia para a indústria local e regional. No entanto, esse sistema possui características de sistemas industriais reais de alto desempenho, agregando tecnologias modernas de automação da manufatura, cujo objetivo é trabalhar os conceitos de modularidade, comunicação de dados, processamento em tempo real, otimização de processos, gerência da qualidade, gestão tecnológica e integração.



Figura 1.2: Laboratório de Automação e Manufatura (Boaretto, Sanzovo e Scandelari, 2004)

Na figura 1.3, Pinotti e Amaral (2006) desenvolveram no *software ABB RobotStudio™* uma célula virtual de paletização para a simulação e aprendizado, adquirindo conhecimentos mais aprofundados de células robotizadas e abrindo um caminho para o desenvolvimento de novos projetos, uma vez que se obtém uma real compreensão de uma simulação de robôs industriais.

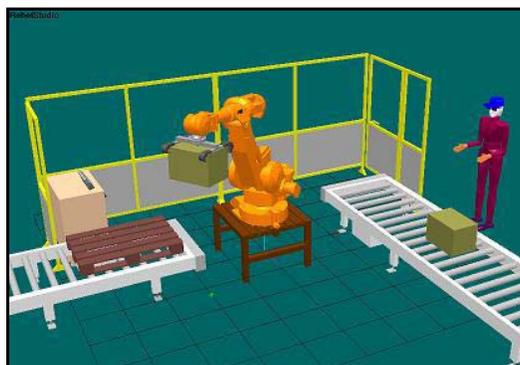


Figura 1.3: Célula de Paletização (Pinotti e Amaral, 2006)

Campos (2006) desenvolveu um protótipo simulador para modelagem e calibração de robôs industriais, para subsidiar, em futuros trabalhos, o planejamento de suas trajetórias programadas “*off-line*” (figura 1.4).



Figura 1.4: Braço Medidor ITG ROMER e Manipulador (Campos, 2006)

Godinho e Pires (2006) desenvolveram interfaces com plataformas Autodesk, para programação de robôs industriais, permitindo a diminuição do inoperante inerente à programação e reduzindo a dependência de técnicos extremamente especializados. As interfaces com plataformas CAD (como exemplo, AutoCAD, Inventor) foram desenvolvidas no Laboratório de Robótica Industrial do Departamento de Engenharia Mecânica da Universidade de Coimbra, para programação de células robotizadas, mostrado na figura 1.5.

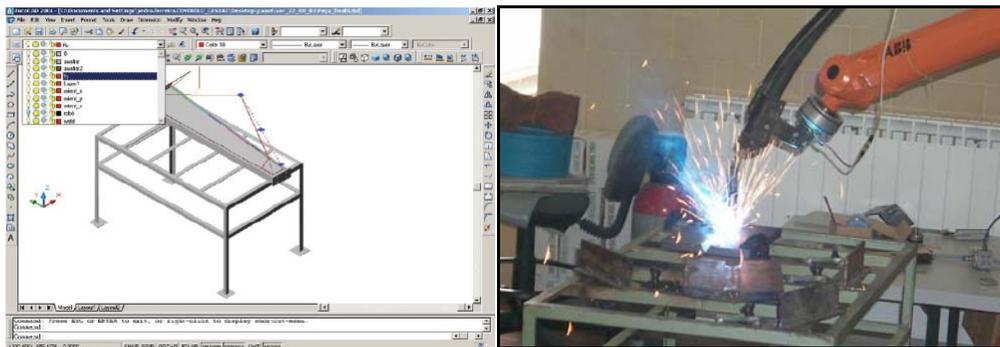


Figura 1.5: Manipulador Robótico - Ferramenta para Soldagem (Godinho e Pires, 2006)

Diante do apresentado, a motivação encontrada para o desenvolvimento deste trabalho de pesquisa deve-se ao fato da necessidade de vários procedimentos para a concepção e

implementação de projetos de células colaborativas em automação e a utilização de tecnologias adequadas para concepção e integração de sistemas robóticos em plantas de manufatura.

## 1.2 Objetivos

Este trabalho tem como objetivo a integração de uma célula de manufatura flexível com a implementação de dispositivos mecatrônicos e a comunicação com o sistema de supervisão e programação que permita uma flexibilidade de uso dos diversos tipos de ferramentas dedicadas. Com isso, o objetivo será dividido em duas partes apresentados:

### **Integração de Células Flexíveis de Manufatura**

- Apresentação de Metodologias para Integração de Células Flexíveis de Manufatura;
- Sistemas Colaborativos de Manufatura através da *WEB*;
- Integração de Dispositivos Mecatrônicos com Robôs Industriais;
- Sistema de Supervisão e Controle aberto utilizando programação estruturada e desenvolvimento de interface gráfica em linguagem LabVIEW<sup>TM</sup>; e
- Maior flexibilidade na integração com ferramentas dedicadas.

### **Modelagem, Acionamento, Controle e Integração de Dispositivos Mecatrônicos**

- Concepção, Modelagem (Cinemática e Dinâmica) e controle de dispositivos mecatrônicos utilizando Matlab-Simulink<sup>TM</sup>;
- Desenvolvimento de interface de acionamento e controle em linguagem LabVIEW<sup>TM</sup> para aquisição e tratamento de informações dos sensores e sistema de acionamento;
- Estudo de Caso: Dispositivo robótico, 3 GDL para posicionamento de uma mesa acionado hidraulicamente.
- Uma página de aplicação HTML, para que a célula colaborativa possa ser disponibilizada na *WEB*, permitindo a criação de um Laboratório ou Indústria Virtual, direcionado à interação entre diferentes empresas e/ou usuários. No caso de Ensino e Pesquisa, isso permitirá, por exemplo, a realização de experiências e ensino à distância, executando tarefas complexas em tempo real.

### 1.3 Estrutura do Trabalho

O desenvolvimento deste trabalho de doutorado foi realizado no Laboratório de Automação Integrada e Robótica (LAR) do DPM/ FEM/ UNICAMP, que vêm desenvolvendo, ao longo dos últimos anos, inúmeros projetos de parceria e pesquisa em cooperação com indústrias, com ênfase na área de implementação de dispositivos associados a uma metodologia para a organização de projetos. Na figura 1.6 mostra a estrutura física deste trabalho.

Para o desenvolvimento dos conhecimentos e habilidades que requerem o uso intensivo de laboratórios que consistem de equipamentos de grande porte, como robôs industriais, sistemas CAD/ CAE/ CAM, redes e protocolos de comunicação, células flexíveis de manufatura, são caros e nem sempre estão disponíveis em laboratórios. Experimentos interativos em plantas/ sistemas reais motivam o desenvolvimento de uma abordagem de resolução de problemas de Engenharia (Álvares e Ferreira, 2003).

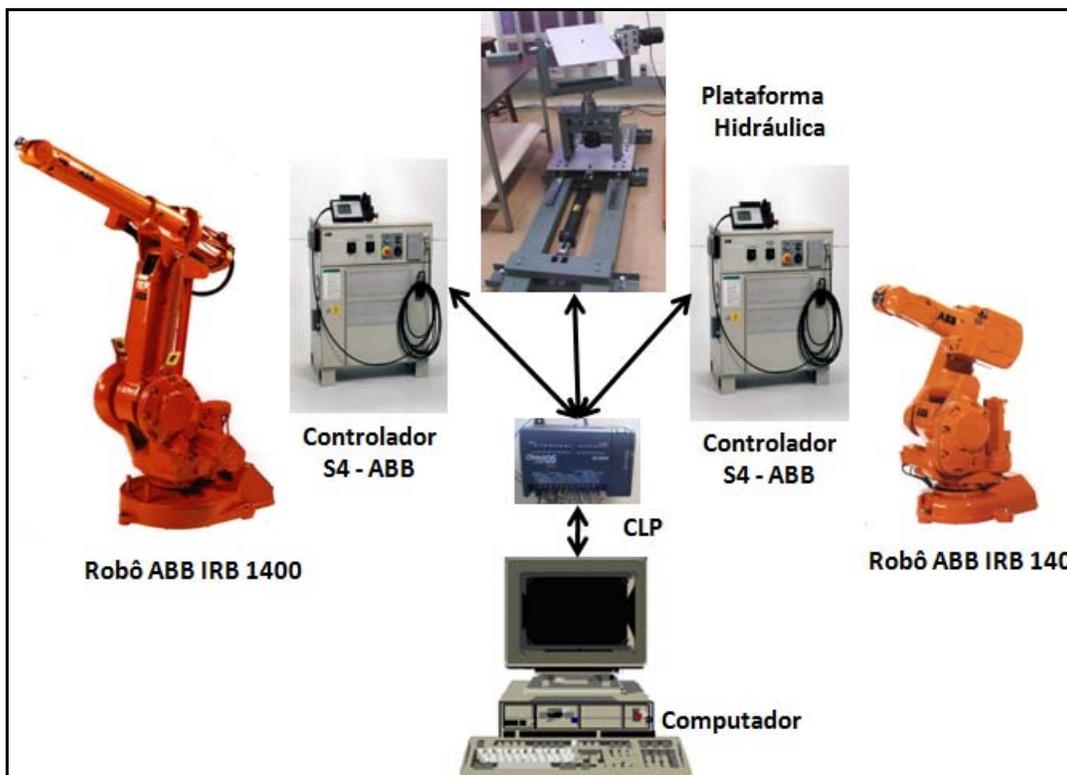


Figura 1.6: Integração de Célula de Manufatura - LAR

A partir da implantação de laboratórios controlados via Internet em um contexto de estação de programação, centro de treinamento, ensino à distância para o desenvolvimento de experimentos remotos na área de automação da manufatura e robótica, deve-se combinar necessariamente a teoria com a prática.

O trabalho foi então dividido nas seguintes etapas:

No Capítulo 2, é apresentada revisão bibliográfica, enfatizando os temas de sistemas colaborativos, programação *off-line*, integração de células de manufatura, apresentação de ferramentas de prototipagem rápida, modelagem e controle de sistemas automatizados e ferramentas para supervisão e controle de sistemas colaborativos.

No Capítulo 3, é apresentada a proposta de um dispositivo robótico com 3 GDL para integração com robôs industriais, enfatizando a modelagem dinâmica, a simulação, e o controle de posicionamento através da implementação de um simulador virtual.

O Capítulo 4 descreve a validação experimental deste trabalho, apresentando dois estudos de caso implementados no Laboratório de Automação Integrada e Robótica da UNICAMP:

- a) Integração de Sistemas Automatizados Colaborativos através da internet;
- b) Implementação de um sistema de supervisão e controle utilizando a ferramenta LabVIEW<sup>TM</sup> e modelagem proposta no capítulo 3.

Finalizando, no último capítulo deste trabalho, são apresentadas as conclusões e perspectivas futuras.

Nos anexos, apresentados conceitos básicos e complementos dos capítulos, que não se encontram no corpo da tese, mas que permitem uma melhor compreensão por parte do leitor.

## **CAPÍTULO 2**

### **REVISÃO BIBLIOGRÁFICA**

Neste capítulo serão apresentados os principais conceitos utilizados para modelagem de células automatizadas para a integração de sistemas colaborativos a partir da consulta de trabalhos publicados, contendo programação *off-line*, ferramentas de modelagem sequencial, ferramentas de supervisão e ferramentas dedicadas, de grande importância para a implementação experimental.

#### **2.1 Integração de Células de Manufatura Flexível**

Nos sistemas de manufatura flexíveis multi-máquinas MMFMS (*Multi Machine Flexible Manufacturing System*) e nos sistemas de manufatura flexíveis multi-células MCFMS (*Multi Cell Flexible Manufacturing System*) é comum termos mais de um robô cooperando em tarefas de manufatura e montagem. Suh e Shon (1989) utilizaram dois robôs PUMA 560<sup>TM</sup> para deslocar peças rígidas, grandes, longas e pesadas. A tarefa dos dois robôs foi movimentar o corpo rígido de um ponto a outro, ao longo de uma trajetória definida previamente dentro do espaço de trabalho dos robôs, e com limitações nas juntas. O robô principal realiza a operação de apreensão de um extremo do corpo da peça e o robô secundário realiza a apreensão do outro extremo do corpo da peça. Como estratégias de controle para o comando dos movimentos são geradas usando redundância cinemática.

Uchiyama e Yamashita (1991) identificaram os vetores que descrevem a tarefa cooperativa de dois robôs, apresentando as condições de força, posição e contato com equações lineares através de um algoritmo.

Stokic, Vukobratovic e Lekovic (1991), desenvolveram um *software* para simulação de FMC. Com este *software*, configuraram uma célula de manufatura flexível com dois robôs, três máquinas ferramentas, duas esteiras elétricas e vários tipos de sensores.

O sistema supervisor e de controle foram projetados utilizando Redes de Petri. Através da modelagem da cinemática dos robôs, foi definida a configuração da célula, e a partir do modelo dinâmico foram otimizadas as velocidades de operação e sincronização, e com as leis de controle dinâmico e compensações nas juntas conforme a precisão das trajetórias e tempos de ciclo (Marange, Gellot e Riera, 2006).

Com um objeto de grandes dimensões e de pesos superiores aos suportados pela garra de um robô, é necessário o desenvolvimento de novas ferramentas de prensão. Vigia *et al.* (1991), estudaram este caso de múltiplos braços de robôs cooperativos interagindo no mesmo espaço de trabalho. Encontraram que a interação entre o corpo rígido e os robôs se caracteriza por limitações unilaterais e configuram uma cadeia cinemática fechada.

Quando dois robôs movem o mesmo objeto cooperativamente, eles integram uma cadeia cinemática fechada; isto exige o controle de posição, velocidade e força.

Tarefas de robôs cooperativos são muito complexas, aumentando ainda mais esta complexidade quando as tarefas são definidas a partir da programação *off-line*. Chi-Keng (1991) reporta esta complexidade na programação *off-line* de robôs para três tipos de movimentos: simultâneo, coordenados e independentes.

Li-Chung e Ming (1994) desenvolveram uma aproximação sistemática para a formulação virtual de limitações no deslocamento e equações dinâmicas para cooperação entre robôs múltiplos, unificando um método para a dinâmica inversa e a capacidade de carga dinâmica.

Rehg (2000) apresenta critérios práticos e descreve a integração de robôs industriais em FMS, mostrando a necessidade de desenvolvimento de *hardware* e *software* dedicados para a implementação de células de trabalho automatizado orientadas aos CIM (Manufatura Integrada por Computador). A programação das FMC (Células Flexíveis de Manufatura) não é fácil e atualmente é uma área de pesquisa e desenvolvimento. Pires e Sá (2003) apresentam uma programação orientada aos objetos, que foi aplicada ao robô ABB IRB 1400 individualmente e depois o robô foi implementado na FMC para o desenvolvimento das trajetórias na indústria automotiva.

A comunicação do usuário com o robô acontece na forma remota, usando comandos especialmente criados para desenvolver as funções do robô. A FMC foi configurada com o robô ABB IRB 1400 no centro e três esteiras elétricas: o robô desenvolve alojamento de um painel da terceira esteira elétrica, logo pega uma segunda peça da segunda esteira elétrica e posiciona sobre o alojamento de outro painel da terceira esteira elétrica (Borangiu *et al.*, 2009).

### **2.1.1 Descrição da Célula de Manufatura Flexível - UNICAMP/ FEM/ DPM/ LAR**

O desenvolvimento deste trabalho de pesquisa envolve o Laboratório de Automação Integrada e Robótica (LAR) do DPM/ FEM/ UNICAMP, onde foram realizados ao longo dos últimos anos inúmeros projetos e pesquisa com ênfase na área de implementação de dispositivos mecatrônicos utilizando conceitos de instrumentação virtual baseada em LabVIEW<sup>TM</sup> associado a uma metodologia para a organização de projetos que pode ser resumida em três etapas de desenvolvimento:

- Elaboração e desenvolvimento da solução;
- Simulação e Prototipagem;
- Implementação e validação.

A partir da realização do projeto de pesquisa tornará possível o desenvolvimento de projetos na área de Mecatrônica, permitindo simulação da implementação de plantas industriais mecatrônicas, próximas a realidade do mercado. A figura 2.1 aponta uma visão sistêmica geral dada à área de Concepção de Sistemas Mecatrônicos, ênfase principal deste trabalho.

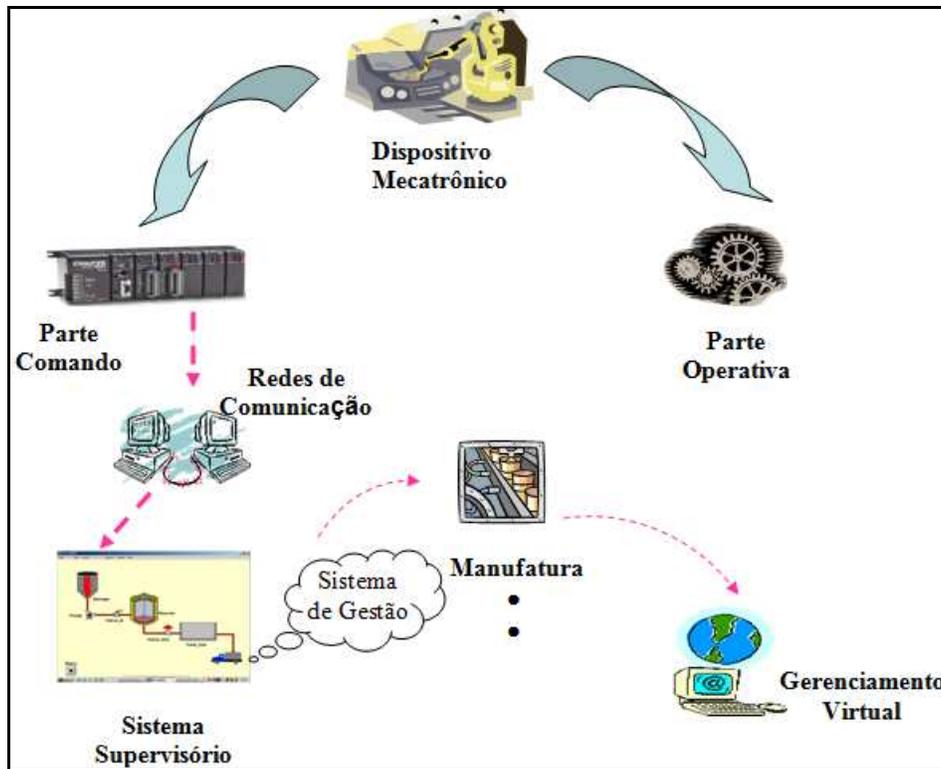


Figura 2.1: Controle, Supervisão e Gerenciamento Virtual de um Dispositivo Mecatrônico.

Desde os anos 80, as estruturas das plantas fabris vêm se modificando rapidamente, em busca de melhor produtividade e racionalização dos recursos investidos a fim de atender as exigências do mercado, a competição entre os fornecedores e a exigência dos clientes. Para tanto, a implementação de novos métodos de produção como as células flexíveis de manufatura e de linhas de produção automatizadas tornam-se necessárias para a obtenção melhorias (Rosário, 2004).

## 2.2 Dispositivos Robóticos

Automação e robótica são duas tecnologias intimamente relacionadas. Em um contexto industrial podemos definir a automação como uma tecnologia que se ocupa do uso de sistemas mecânicos, eletrônicos e da base de computadores na operação e controle da produção. Como exemplo, tem-se: máquinas de montagens mecanizadas, sistemas de controle de realimentação, máquinas operatrizes dotadas de comando numéricos e robôs (Groover, 1988).

A definição de um robô industrial dada pela Associação das Indústrias de Robótica (RIA) é: “Um robô industrial é um manipulador reprogramável, multifuncional, projetado para mover materiais, peças, ferramentas ou dispositivos especiais em movimentos variáveis programados para a realização de uma variedade de tarefas.”

A robótica é considerada hoje a mola mestra de uma nova mutação dos meios de produção, isto devido a sua versatilidade, em oposição à automação fixa. Os robôs, graças ao seu sistema lógico ou informático, podem ser reprogramados e utilizados em uma variedade de tarefas, sendo que a reprogramação não é o fator mais importante na versatilidade desejada e sim a adaptação às variações no seu ambiente de trabalho, mediante um sistema adequado de percepção e tratamento de informação.

### **2.2.1 Programação *off-line* de Robôs Industriais**

A programação *off-line* que será agora abordada é um método de ensino que vem ganhando novos impulsos com o desenvolvimento da tecnologia, com tendência de tornar-se popular no uso da programação de robôs, devido o aumento 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*, fazem da programação *off-line* uns dos métodos mais viáveis. Estes desenvolvimentos incluem sofisticções na unidade de controle do robô, a qual melhora as precisões de posicionamento, e adoção de tecnologias de sensores. Atualmente, há atividades consideráveis no método de programação *off-line* e estas técnicas estão sendo melhoradas nas indústrias de manufatura, o que proporciona uma forma efetiva de utilização desta prática (López, Romeo e Guerrero, 2009).

Na programação *on-line* o operador programador necessita pegar e colocar as peças produtos, os dispositivos de fixação e posicionamento de fixação e posicionamento, os robôs e, os

periféricos 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 terminal de programação manual “*Teach Pendant*”.

A mais significativa vantagem da programação *on-line* é que o robô tem o conteúdo programado em concordância com a atual posição dos equipamentos, dispositivos de fixação e posicionamento e peças. Contrariamente, 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-produtos, os dispositivos de fixação e posicionamento (*JIG's*), os robôs, os periféricos, 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 2.2, pode ser observado um exemplo de uma célula de trabalho, em ambiente virtual, em que os programas de robô são verificados dentro de simulações e qualquer erro pode ser corrigido (Freitas, 2004).

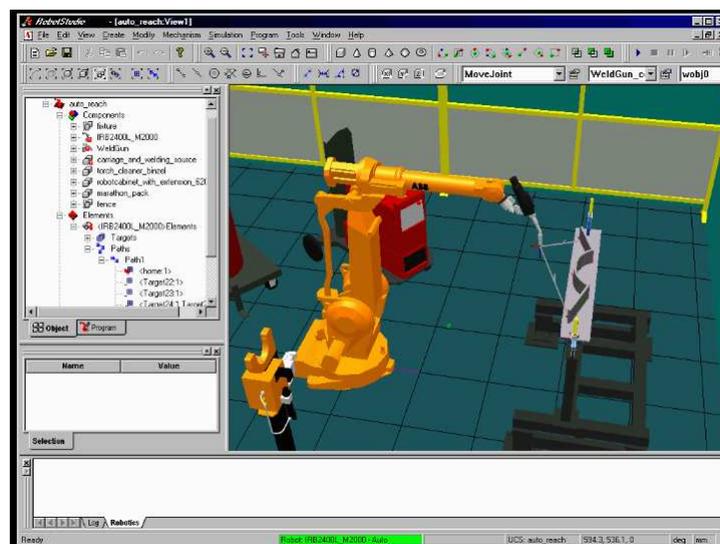


Figura 2.2: Exemplo de uma Célula de Trabalho

A programação *off-line* possui a 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) Verificação de programas e lógicas através da simulação com a visualização;
- c) Modelo bem documentado com a simulação, com o programa apropriado;
- d) Reutilização de dados CAD (Desenho Auxiliado por Computador) existentes, quando serão usados para simulação;
- e) Custo independente da produção. Durante a programação *off-line* a produção não para.

A 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. Outra vantagem é que não se ocupa equipamento de produção e dentro deste modo a produção pode continuar durante o processo de programação (Freitas, 2004).

As ferramentas avançadas de programação *off-line* contêm 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, a 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 lotes 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 estão crescendo e o aumento da complexidade da aplicação robótica, particularmente com

respeito a trabalho de montagem, soldagem com as vantagens da programação *off-line* é algo bem atrativo (Henriques, 2005).

Segundo Freitas (2004), 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. Todo o processo que envolve a robótica possui 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 podem ser resumida, por:

- 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 é feito 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 2.3.

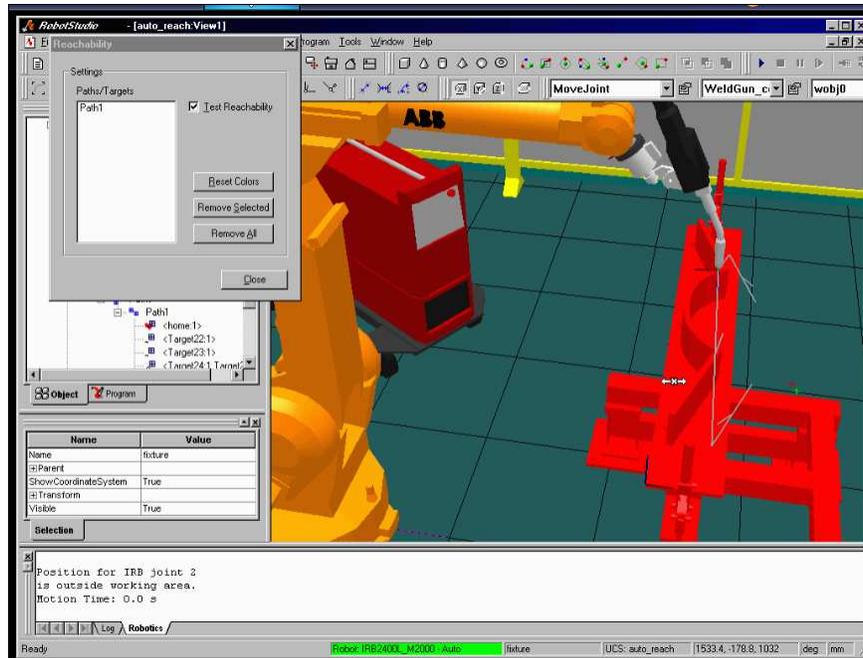


Figura 2.3: Programação *off-line* - Identificação de Colisões

É possível numerar 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.

## 2.2.2 Desenvolvimento de Ferramentas Dedicadas para Dispositivos Robóticos

A robótica está, nos últimos anos, em grande expansão, tanto no desenvolvimento de novas máquinas como na sua aplicação.

A seleção da ferramenta de uma base de dados depende da operação de manufatura do robô. Na figura 2.4 são apresentados diversos tipos de ferramentas e aplicações comumente utilizadas em células de manufatura.

Em especial, os robôs industriais vêm fazendo parte da automatização de células de manufatura, como pode ser observada nas aplicações industriais mais tradicionais: paletização, soldagem por arco contínuo, soldagem por pontos, usinagem, montagem, dobramento de chapas, cola em vidros de carros, corte por jato de água, corte a laser, polimento, desenhos, pintura por pulverização,



Figura 2.4: Ferramentas Dedicadas

### 2.3 Ferramentas de Modelagem Sequencial para Integração

Nesta subseção são apresentados conceitos teóricos e definições básicas referentes à Sistema Automatizados, desde ferramentas para Modelagem até a Integração de Sistemas

Automatizados, permitindo assim ao leitor uma melhor familiaridade com os conceitos de Instrumentação Virtual e Sistemas Embarcados.

### **2.3.1 Sistemas Automatizados**

Para compreender um sistema automatizado é necessário entender inicialmente o conceito de sistema. Um sistema é qualquer interação de elementos cujo funcionamento visa alcançar um objetivo comum e que evoluiu com o tempo. Portanto, segundo a definição citada, aquilo que pode ser definido como sistema em um contexto pode ser apenas um componente de outro sistema, originando o conceito de subsistema.

Pode-se definir ainda o sistema como um conjunto complexo de coisas diversas que relacionadas entre si, contribuem para determinado objetivo ou propósito.

Pode-se definir a Modelagem de um Sistema como uma representação de um objeto, sistema ou idéia em uma forma diferente ao elemento propriamente dito. Desta forma, o Modelo de um Sistema é um conjunto de informações sobre determinado sistema com o propósito de entender o mesmo, ou seja, um modelo é uma descrição do sistema real. Literalmente pode-se dizer que o modelo é a representação de alguma coisa, podendo ser definido também como a representação simplificada de um sistema com o propósito de estudar o mesmo.

Desta forma um modelo passa a ser uma réplica ou uma abstração com a característica essencial de um sistema ou processo. Através dos problemas que desobedecem a soluções diretas por causa do tamanho, complexidade ou estrutura, são frequentemente avaliados através de modelos de simulação, ou seja, o Modelo passa a ser uma representação simplificada de alguma parte da realidade de sistemas, podendo ser estes sistemas de diferentes tipos.

Existe a concepção de que os modelos devam ser construídos para resolver problemas específicos, podendo ter vários modelos de um sistema, cada um mais adequado à resolução de um problema particular do que o outro. Apesar de serem construídos a partir do problema a resolver, os modelos requerem uma estrutura organizada. Uma linguagem para simulação pode

fornecer uma destas estruturas, e sua compilação é o que vai traduzir a descrição do sistema em uma forma aceitável por um sistema de computação (Soares, 1992).

Ao desenvolver um modelo de simulação, o modelador precisa selecionar a estrutura conceitual na qual o modelo vai se apoiar para a descrição do sistema. A estrutura conterá o "enfoque" (ou visão), dentro do qual as relações funcionais entre os elementos do sistema são percebidas e descritas (Morcelli, 2007).

Estes modelos podem ser classificados como: físico (escala natural e reduzida) e matemático (numérico/ algorítmico), sendo que as principais etapas necessárias para a obtenção dos mesmos são a análise do sistema, através da identificação de entidades, atributos, etc., e a simplificação, através da desconsideração das entidades e atributos considerados irrelevantes.

### **2.3.2 Simulação de Sistemas Automatizados**

A simulação de um sistema pode ser definida como a capacidade de conduzir experimentos utilizando um modelo de sistema real de forma a compreender o comportamento deste e possibilitar a avaliação de estratégias para a operação do mesmo, ou seja, é a técnica de resolver problemas seguindo as variações ocorridas ao longo do tempo (Rosário, 2004).

Um importante fator para a larga utilização da simulação é a sua flexibilidade. A maioria das outras técnicas, tal como modelagem analítica, requer que os sistemas reais sejam transformados em um modelo idealizado em uma estrutura bastante específica. Quando esta idealização é possível sem um significativo compromisso da natureza do sistema considerado, a técnica analítica pode ser usada para obter uma solução (Morcelli, 2004).

Os objetivos da simulação podem ser classificados em quatro categorias (Adam e Dogramaci, 1979):

- Comparação de estratégias com respeito a um problema individual no mundo real: sendo que a maioria das aplicações de simulação inclui-se nesta categoria. Estas aplicações

iniciam com alguma análise estatística em relação a um problema específico e através da simulação procura-se encontrar uma solução para o problema.

- Desenvolvimento de afinidade funcional. Nesta situação, a utilização da simulação tem por objetivo obter critérios de relação entre variáveis que, posteriormente, com uma análise matemática poderá possivelmente chegar a uma ótima solução.
- Validação e avaliação de métodos analíticos recentemente desenvolvidos, ou seja, é a utilização da simulação como ferramenta ou fonte de idéias para novas técnicas analíticas, como a aplicação para validação de métodos matemáticos ou novas teorias.
- Propósitos educacionais ou de treinamento. A simulação permite colocar uma pessoa frente a situações difíceis e analisar seu comportamento e suas decisões sem que isso possa colocar em risco o andamento normal do sistema. Desta forma, a Simulação é um importante dispositivo de aprendizado.

Porém, a utilização da simulação possui limitações, dentre as quais pode-se destacar a dependência entre resultados e estímulos, o desenvolvimento de bons modelos podendo ser onerosos e a falta de precisão e/ ou qualidade da modelagem no fornecimento de valores das variáveis em todos os instantes de tempo (Rosário, 2004).

### **2.3.3 Elementos de Modelagem de um Sistema Automatizado**

A evolução tecnológica está levando à crescente complexidade dos sistemas automatizados, implicando a dificuldade por parte do usuário, na definição de uma maneira clara, concisa e não ambígua das especificações funcionais associadas a esses sistemas. Esta complexidade tende aumentar ainda mais, com a utilização de um número elevado de troca de informações entre os elementos constituintes deste sistema. Desta forma é necessário descrever o sistema através de uma linguagem de descrição adequada. Esta linguagem precisa ser do ponto de vista do homem, uma forma que expresse de modo natural a especificação do sistema e do ponto de vista do sistema, uma descrição simples que seja facilmente interpretada e executada (Aihara, 2001).

As linguagens verbais e textuais não são as mais indicadas para a modelagem dos S.A. (Sistemas Automatizados), pois pode levar a mais de uma interpretação, e até mesmo a

informações ambíguas. Portanto, em sistemas complexos, onde haja ações simultâneas e decisões com múltiplas possibilidades, deve-se evitar a utilização da linguagem verbal e textual. Sempre que possível e necessário, as descrições de sistemas automatizados devem ser representadas na forma gráfica, pois possuem uma facilidade maior em serem interpretadas e executadas, porém, existe a dificuldade de se encontrar uma forma que seja aceita e entendida por todos (Rosário, 2004).

A Parte Comando de cada processo necessita, dentre as diferentes tecnologias de comando disponíveis, de tecnologia adequada e a de melhor adaptação. Dentre as diferentes tecnologias existentes, podem-se citar os comandos pneumáticos, hidráulicos, eletro-mecânicos e os Controladores Lógicos Programáveis (CLP).

Desta forma, a Parte Comando é mantida informada sobre o estado dos subsistemas, através das informações fornecidas pela Parte Operativa. Esta pode ainda trocar informações com o exterior do sistema, de onde pode receber indicações, ordens (botões de comando, chaves, sensores, etc.) e fornecer sinalizações sonoras e/ ou luminosas (sinalizadores, buzinas, lâmpadas, etc.) (Aihara, 2005).

Portanto, para o desenvolvimento de um SA deve-se inicialmente descrevê-lo de modo a não ficar nenhuma dúvida sobre os objetivos a serem atingidos no projeto proposto, sem a preocupação com detalhes tecnológicos, quando então se devem descrever os elementos específicos do sistema automatizado (Aihara, 2005).

Para se tentar obter a melhor descrição dos Sistemas Automatizados deve-se desenvolver seu Caderno de Tarefas, cujo objetivo é a descrição de seu comportamento em função da evolução das etapas que descrevem o processo a ser automatizado. O mesmo deve apresentar uma descrição clara, precisa e sem ambigüidades, nem omissões, do papel das etapas constituintes do processo a ser automatizado (Araújo, 1997).

Estas considerações são muito importantes para a exploração do processo a automatizar, considerando-se suas repercussões sobre o aspecto econômico, que são frequentemente

esquecidas durante a elaboração do caderno de tarefas, pois são difíceis de serem expressas de maneira quantitativa.

O estudo de modelagem e simulação de sistemas discretos está associado às necessidades de se estabelecer limites e prever o comportamento destes sistemas. Nos S.A. este comportamento está intimamente ligado à utilidade destes sistemas, quer seja como extensão da capacidade humana de operação e transformação de objetos, como por exemplo, o sistema de manufatura quer seja como forma de substituir completamente o elemento humano em atividades perigosas, tais como a manutenção de centrais nucleares, as operações de inspeção, reparos, soldagem em grandes profundidades, ou mesmo a manipulação de carga em grandes alturas nas construções civis, ou ainda os sistemas de coleta de dados (Cardoso e Valette, 1997).

Entretanto, nestas atividades onde o forte é a extensão da capacidade humana, através da melhora da precisão de intervenção, repetibilidade e, confiabilidade, torna-se necessário que o processo de automatização represente um avanço, ao invés de uma simples modernização (Aihara, 2005).

Portanto, a maneira mais direta de representar formalmente um sistema controlável é através do mapeamento de seu comportamento, ou pelo menos dos seus principais processos, como autômatos finitos. Neste caso, todo o sistema, incluindo as suas partes, é considerada uma única representação chamada estado e a evolução dentre os estados, isto é, uma transição de um estado para outro, é uma condição, ou seja é o elemento de ligação entre estes estados. (Cassandras, 1993).

Esta transição pode ser representada através de um arco direcionado, sendo esta forma pictórica de representação da evolução dos estados, somente uma representação alternativa e talvez mais fácil de interpretar. Isto provavelmente por ter uma comunicação quase subliminar sobre a evolução do sistema e sobre o conceito de processo. Entretanto, a formalização mais acabada do sistema e seu funcionamento estão associados à definição do que se conhece como um autômato (Aihara, 2005).

Um autômato finito é definido como sendo um conjunto finito de estados, dos quais um é identificado como o estado inicial, e outro ou eventualmente nenhum, é considerado como o estado final, sendo este associado o final do processo (Ramadge e Wonham, 1989).

Para Peterson (1981), é comum nos sistemas encontrar componentes que apresentem atividades concorrentes ou paralelas e, neste sentido, as Redes de Petri é uma linguagem de modelagem que foi desenvolvida especificamente para modelar sistemas discretos que possuem componentes que interagem concorrentemente. Sistemas seqüenciais podem ser modelados através de GRAFCET, como segue no anexo A.

#### **2.3.4 Redes de Petri - Conceitos Básicos e Definições**

A metodologia de modelagem de Sistemas a Eventos Discretos (SED) utilizando Redes de Petri, (RdP) foi proposta em 1962, por Carl Petri, matemático alemão, que através de uma tese criou esse método de estudo para sistemas dinâmicos evento discreto, direcionado às Comunicações com Autômatos, originando posteriormente, duas linhas de desenvolvimento nas áreas de Ciências da Computação e em Engenharia de Sistemas padronizando as Redes de Petri (Huber, Jensen e Shapiro, 1990).

Portanto, as Redes de Petri são ferramentas gráficas e matemáticas de modelagem para descrição e/ ou especificação que podem ser aplicadas a diversos tipos de sistemas, apresentando um bom nível de abstração em comparação com outros modelos gráficos. Além disso, as Redes de Petri possibilitam a verificação do sistema especificado. Usando-se RdP, pode-se modelar sistemas paralelos, concorrentes, assíncronos e não-determinísticos (Melo e Sobreira, 2003).

A RdP é uma linguagem formal que permite a modelagem de sistemas dinâmicos discretos permitindo representar com facilidade todas as relações de causalidade entre processos em situações de: seqüencialidade, conflito, concorrência e sincronização (Matos e Santos, 2004), sendo que a sua aplicabilidade em diversas áreas transformou-o em tema alvo de investigação básica e aplicada, e sua utilização na Modelagem de Sistemas Automatizados apresenta algumas vantagens como:

- captura das relações de precedência e os vínculos estruturais dos sistemas reais;
- graficamente expressivas, permitindo a modelagem de conflitos e filas;
- possui fundamento matemático e prático;
- admite várias especializações (temporizadas, coloridas, estocásticas, de confiabilidade etc.).

Desta forma, o interesse por esta forma de linguagem tem aumentado nos últimos anos, pois permite modelar sistemas com níveis de complexidade maior, gerando modelos mais compactos e extensões que incorporam parâmetros temporais.

Portanto, podem-se definir as Redes de Petri (RdP's) por meio de conjuntos, funções e também por grafos, de maneira que suas propriedades possam ser obtidas pela teoria dos conjuntos e/ ou pela teoria dos grafos.

As redes de Petri permitem modelar sistemas constituídos por componentes que apresentem características de funcionamento concorrente e inter-atuantes. Segundo Peterson (1981) a sua utilização poderá ser realizada de modos diversos, dependendo do objetivo em vista.

Uma abordagem possível de utilização das RdP considera-as como uma ferramenta auxiliar de análise, sendo que nesta abordagem, outras técnicas e formalismos são utilizados para especificar o sistema. Se forem detectados problemas, procedem-se as alterações na especificação e o ciclo será repetido até que mais nenhum problema seja detectado. A figura 2.5 demonstra este processo (Rojas, 2004).

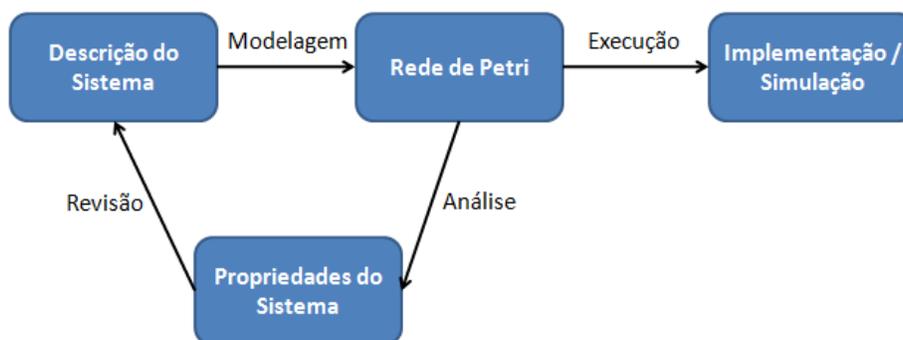


Figura 2.5: Utilização de Redes de Petri no Modelamento e Análise de Sistemas

## 2.4 Prototipagem de Sistemas de Controle - Dispositivos de Fixação e Posicionamento

Nos dias atuais, a evolução tecnológica está direcionando a uma gama de produtos na área de engenharia, acarretando em alterações drásticas na concepção de um projeto de um sistema mecatrônico no que diz respeito às evoluções do projeto e concepção mecânica, sistema de acionamento e de controle. Como exemplos, podemos citar a intensidade crescente de componentes eletromecânicos inteligentes, de máquinas automatizadas, de veículos com grande número de sensores e eletrônica embarcada e de dispositivos mecânicos de precisão.

Estes novos produtos mecatrônicos emergem da combinação apropriada dos sistemas mecânicos, da eletrônica e do processamento em tempo real dos sistemas de controle e tratamento de informações, através da implementação de diversificadas funções de controle embarcadas e integradas ao sistema (Rojas, 2004).

Um fator de importância industrial consiste na redução de tempo do desenvolvimento de um produto com a diminuição do número de etapas, gerando a produção de novos e diferenciados produtos. Isso é indispensável no mundo globalizado e particularmente necessário nas indústrias de alta tecnologia, como por exemplo, no que diz respeito à concepção de controladores cada vez mais rápidos e eficientes.

Muitos processos e produtos técnicos nas diferentes áreas da engenharia, principalmente na engenharia mecânica e elétrica, apresentam uma integração crescente dos sistemas mecânicos com processamento da eletrônica digital e de informação. O seu desenvolvimento envolve a busca de uma solução otimizada entre a estrutura mecânica básica, o sistema de sensoriamento e o elemento de atuação e controle, através do processamento automatizado de informações e controle global do sistema.

Tudo isso acarretará no desenvolvimento e utilização de ferramentas para o projeto simultâneo dos sistemas mecânicos e *hardware* de acionamento e controle, através da implementação do *software* e funções de controle embarcadas dentro de um ambiente de simulação, tendo como resultado e principal objetivo a eliminação da totalidade ou parte de

protótipos intermediários e a geração de um componente ou um sistema integrado de controle. Para tal propósito descrito, existe um conceito atual denominado prototipagem rápida de sistemas mecatrônicos (Rojas, 2004).

A prototipagem rápida é uma ferramenta que permite a construção de protótipos de uma maneira econômica e segura. Um conceito de prototipagem rápida utilizado anteriormente referia-se a construção de protótipos de peças mecânicas a partir de um projeto desenvolvido em CAD (*Computer Aided Design*) ou da implementação de componentes eletrônicos em FPGA's, a partir de um CAD de eletrônica. Mais recentemente, esse conceito é utilizado de forma mais ampla, envolvendo a concepção de todo um projeto de um sistema mecatrônico desde as fases de modelagem, simulação e arquitetura de controlador até a sua implementação final em *hardware* dedicado (Oliveira, 2008).

A prototipagem rápida tem como objetivo a geração automática do código equivalente ao controlador para testar os sistemas reais, resultando em diminuição nos custos de implementação de um controlador, principalmente se esse for desenvolvido para um projeto específico. Ou seja, a prototipagem rápida é uma ferramenta que possibilita a construção de protótipos de uma maneira econômica e segura, onde *hardware* pode ser implementado em um sistema embarcado (*embedded system*) a partir de componentes virtuais (Rosário, 2004).

#### **2.4.1 Prototipagem Rápida em Mecânica**

O conceito de prototipagem rápida possui um vínculo muito forte relacionado à área mecânica devido à construção de peças mecânicas a partir de um projeto desenvolvido em CAD (*Computer Aided Design*). Na figura 2.6, tem um exemplo de aplicação de técnicas de prototipagem rápida na elaboração de projetos de dispositivos mecânicos.

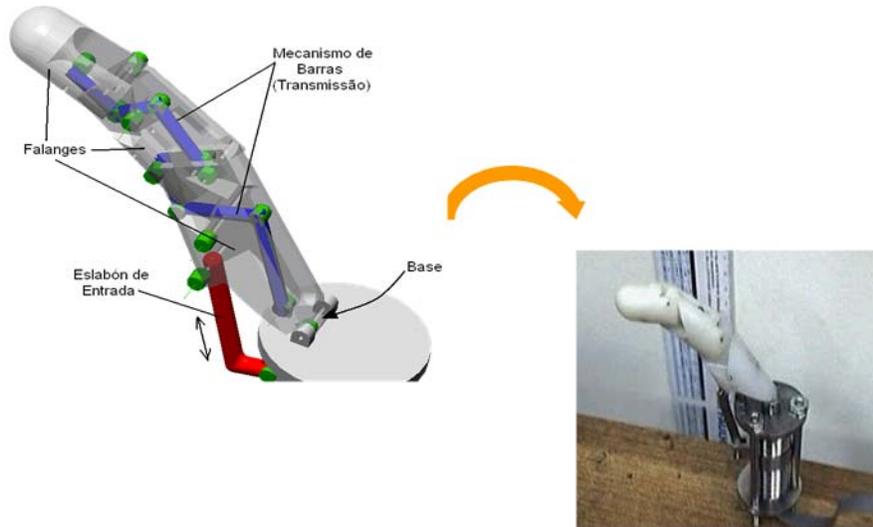


Figura 2.6: Concepção de um Produto Desenvolvido em CAD (Sánchez, 2008)

### 2.4.2 Prototipagem Rápida em Eletrônica

O conceito de prototipagem rápida em eletrônica refere-se inicialmente a implementação de componentes eletrônicos em FPGA's a partir de um CAD de eletrônica. Um exemplo de uso deste conceito é a aplicação de técnicas de *Hardware In the Loop* (HIL) (Sánchez, 2008).

Uma das principais vantagens dos sistemas de prototipagem rápida é que, após a simulação do modelo virtual do sistema, o sistema total ou partes desse modelo poderão ser facilmente trocados pelo *hardware* de controle real para validação e testes do modelo, de modo que o protótipo virtual se transformará em um produto muito próximo do real, simplificando etapas de concepção, validação e testes. Conseqüentemente, após a fase de simulação do sistema, que deverá incluir o modelo virtual do *hardware*, esta parte do modelo poderá ser trocada pelo *hardware* real, similar ao modelo real, fazendo que o *hardware* passa a ser integrado no modelo de simulação virtual e possibilitando o estudo do comportamento do novo componente com o resto do sistema, em particular o estudo do comportamento global do sistema (Sánchez, 2008).

Essa fase de testes é chamada de *Hardware In the Loop* (HIL), que deverá permitir a integração automática do *hardware* físico dentro do ambiente de simulação. As principais ferramentas necessárias para realização desse procedimento são as seguintes:

- Ferramentas de programação e *hardware* aberto para implementação do protótipo;
- Unidade de cálculo que permita assegurar a comunicação com o *hardware* e a área de trabalho, que deverá comportar portas de entradas e saídas e conversores analógico/ digital e digital/ analógico.
- Implementação de programa que permita a importação de modelos representando o sistema e que seja capaz de gerar as entradas e programável por blocos.

As aplicações do HIL são utilizadas para avaliar e validar os elementos desenvolvidos por um novo sistema. Elas consistem em testar esses elementos, antes de concretizar o sistema real, a partir unicamente da simulação do resto do sistema. Os componentes do *hardware* testados respondem aos sinais enviados pelo computador e simulam o resto do sistema, como se encontrassem dentro de um sistema real (Sánchez, 2008).

Atualmente existem várias empresas que oferecem soluções que realizam o *hardware* in loop (HIL), como por exemplo: dSPACE<sup>®</sup>, National Instruments<sup>®</sup> (NI), Altera<sup>®</sup> e Opal-RT<sup>®</sup> que apresentam produtos dedicados a prototipagem rápida.

A figura 2.7 apresenta esquematicamente um exemplo de implementação através do ambiente MathWorks<sup>®</sup> utilizando HIL. Neste exemplo observa-se que a partir do ambiente de simulação utilizado para validação e testes do modelo físico implementado, o *hardware* de controle poderá ser automaticamente gerado e incorporado dentro do simulador para validação e testes dentro desse mesmo ambiente de prototipagem. As principais etapas a serem implementadas no HIL são as seguintes:

- Desenvolvimento de um programa de simulação usando ambiente tal como o Matlab/ Simulink<sup>®</sup>;
- Configuração das entradas e saídas necessárias para o funcionamento do sistema simulado com o elemento *hardware*;
- Geração automática do código (com possibilidade de troca de parâmetros em tempo real e iniciação imediata da simulação em HIL);
- Visualização dos resultados em tempo real.

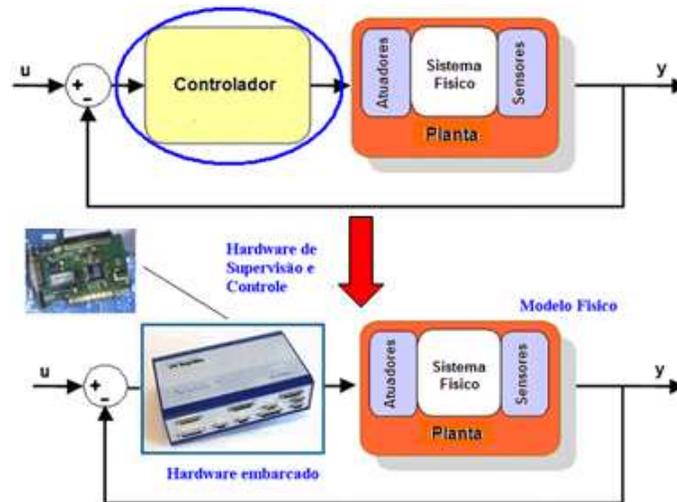


Figura 2.7: *Hardware* In the Loop através do Mathworks®

#### 2.4.4 Protipagem Rápida em Célula Flexível de Manufatura

O conceito de prototipagem rápida em células flexíveis de manufatura refere-se à integração de diferentes dispositivos em células automatizadas de manufatura (*Computer Integrated Manufacturing*). A figura 2.8 ilustra uma aplicação de prototipagem rápida em células flexíveis de manufatura.

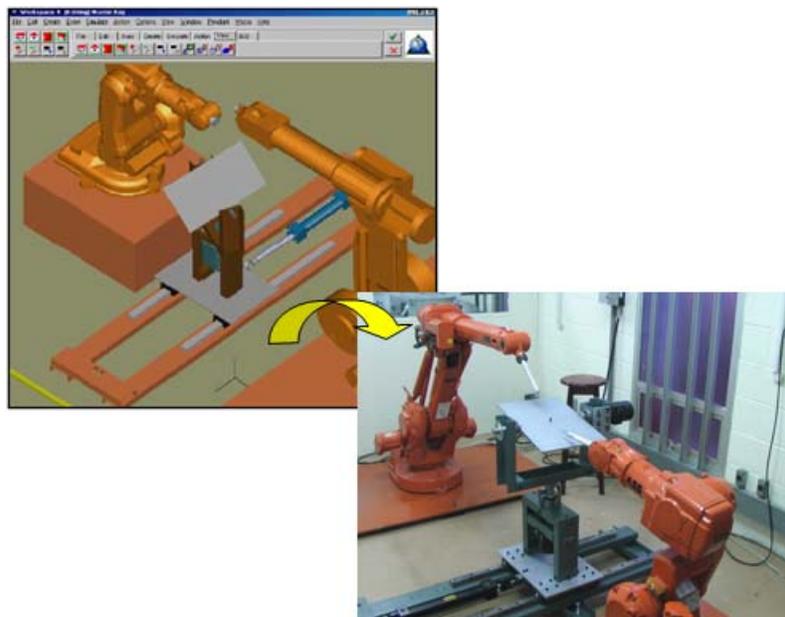


Figura 2.8: Prototipagem de uma Célula Flexível de Manufatura (Freitas, 2004)

### **2.4.5 Prototipagem Rápida em Instrumentação Virtual**

Pode definir prototipagem rápida em instrumentação virtual como a implementação de controle de sistemas mecatrônicos através de *software*, *hardware* e Controladores Programáveis para Automação (PAC) (Freitas, 2004).

Atualmente, conceito de Prototipagem Rápida envolve a concepção de todo o projeto de um sistema mecatrônico através de ferramentas colaborativas, desde as etapas de modelagem, simulação e arquitetura de controlador, até a sua implementação final em *hardware* dedicado. Ampliando esse conceito pode incluir a implementação em ambiente virtual do modelo do sistema (modelagem cinemática e dinâmica), simulação e *hardware* de supervisão e controle (Freitas, 2004).

### **2.4.6 Concepção de Sistemas de Supervisão e Controle baseados em Instrumentação Virtual**

O conceito de Instrumentação Virtual surgiu a alguns anos, mudando a forma como engenheiros e pesquisadores abordam problemas na área de Automação e Controle Industrial. Nos dias atuais, este conceito tem atingido níveis de aceitação no mercado bastante elevados, passando a ser usada em milhares de aplicações ao redor do mundo, como em indústrias automotivas, de bens de consumo e de óleo e gás (Aihara, 2005).

Instrumentos Virtuais são instrumentos definidos pelo usuário por meio do desenvolvimento de uma aplicação. Isso significa desenvolver um *software* baseado nos requerimentos do usuário e definir um *hardware* de propósito geral para medição e controle.

A Instrumentação Virtual combina as tecnologias comerciais mais recentes com *software* e uma vasta variedade de *hardware* para medição e controle. Assim, engenheiros e cientistas podem criar sistemas que atendam exatamente as necessidades do usuário. Além disso, conseguem reduzir tempo de desenvolvimento, projetar produtos de maior qualidade e baixar os custos de seus projetos. Isso pode ser notado, por exemplo, na indústria de telefonia móvel. Há pouco tempo, os aparelhos celulares possuíam áudio, agenda e enviavam mensagens. Já os

aparelhos de última geração, além dessas características, também possuem câmera, MP3 player, Bluetooth e conexão à Internet. Isso só é possível porque os aparelhos eletrônicos têm se tornado cada vez mais centrado em *software* (Oliveira, 2008).

Assim, engenheiros e pesquisadores devem adicionar novas funções ao aparelho sem substituir o *hardware*, resultando em melhores produtos, sem os custos de re-desenvolvimento do *hardware*, podendo acarretar uma interação com o usuário não prevista, sujeita a presença de erros. Nesse sentido, também se torna necessária a presença de ferramentas para a verificação constante das alterações feitas na aplicação. A única forma de atender a esses requisitos é pelo uso de arquiteturas de teste e controle também centradas em *software*.

A Instrumentação Virtual está muito associada ao *software*, tornando-se assim uma arquitetura capaz de atender características, adaptar-se a novas funcionalidades de um novo cenário de desenvolvimento, tornando mais rápido a desenvolvimento de produtos na medida em que provê um ambiente gráfico de programação (Oliveira, 2008).

#### **2.4.7 Concepção de Módulos de Supervisão e Controle baseados em Instrumentação Virtual**

Na concepção de Instrumentos Virtuais em Automação, podemos utilizar diferentes modos de integração, função de fatores como flexibilidade, custo, rapidez e arquitetura aberta em nível de *software* ou *hardware*. Um dos procedimentos é a Integração de diferentes tipos de dispositivos em uma mesma aplicação: onde não importando a escolha, um único programa poderá ser usado para qualquer das aplicações implementadas, sem a necessidade de se fazer qualquer mudança em seu código.

A figura 2.9 exemplifica a integração de três diferentes dispositivos em uma mesma aplicação (programa) usando Instrumentação Virtual através de um desktop e barramento PCI. Essa aplicação consiste um dispositivo para medir tensão DC e temperatura, e um sistema para distribuir a aplicação para um sistema PXI, que trabalhará em chão de fábrica.

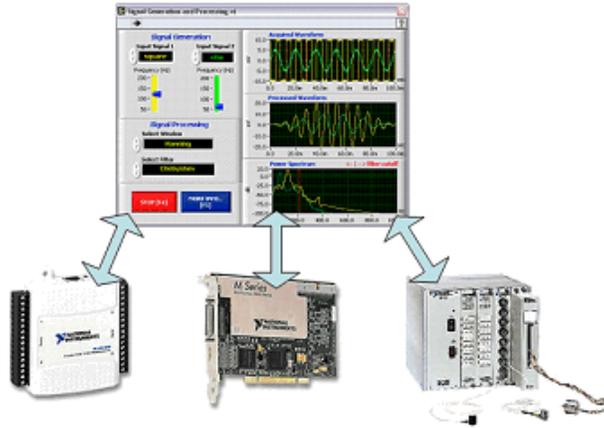


Figura 2.9: Integração de Diferentes tipos de *Hardware* em uma mesma Aplicação (*National Instruments*)

A figura 2.10 mostra a integração de diferentes aplicações em um mesmo dispositivo, através de um projeto em que um dispositivo de aquisição de dados e encoders de quadratura são utilizados para medir a posição de um motor e outro dispositivo para monitorar sua potência. Neste exemplo, embora ambos os projetos pudessem ser combinados em uma única aplicação utilizando um único dispositivo de aquisição de dados, são utilizados sistemas diferentes dentro de um mesmo projeto (programa).

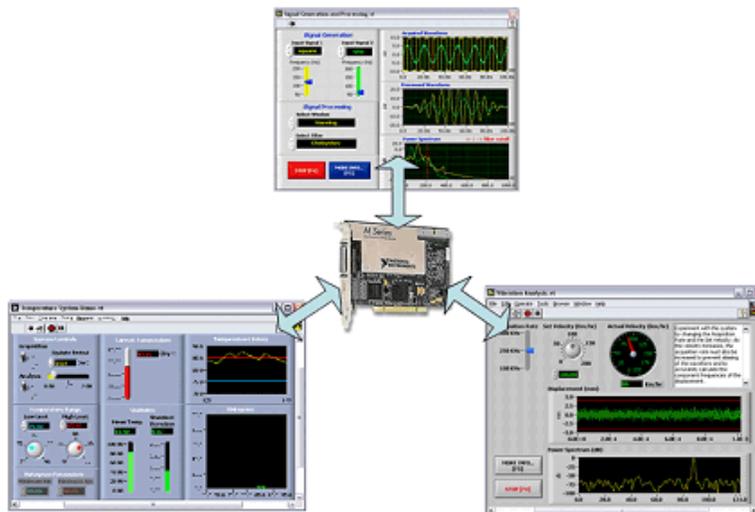


Figura 2.10: Integração em um mesmo *Hardware* de Diferentes Tipos de Aplicações (*National Instruments*)

Outro importante conceito da Instrumentação Virtual é a estratégia adotada para se adaptar à evolução de *hardware* e *software*. Essa estratégia é focada no uso dos altos investimentos de grandes companhias de tecnologia e visa oferecer *hardware* e *software* compatíveis com o mercado. Como a Instrumentação Virtual é baseada em *software*, uma tarefa de medição só é possível devido à digitalização. Assim, *hardware* de medição pode ser analisado sob dois aspectos: resolução (*bits* dos conversores AD) e frequência (taxa de amostragem).

## 2.5 Sistemas Supervisórios e Colaborativos

Um Sistema de Supervisão e Controle é responsável pelo monitoramento de variáveis de controle dos Sistemas Automatizados, como também pela integração entre estes sistemas e os sistemas hierarquicamente superiores responsáveis por um gerenciamento mais global como, por exemplo, nas indústrias os Sistemas de Gerenciamento da Produção (Aihara, Cosso, et al., 2001).

Chama-se um grupo ou rede de robôs de "cooperativos" quando suas ações são coordenadas de forma a realizar uma tarefa ou atingir um objetivo comum, ou seja, os robôs devem cooperar para resolver um problema específico, principalmente quando um só robô não é capaz de resolver o problema sozinho (Henriques, 2005).

Um exemplo da necessidade de robôs que atuem de forma cooperativa aparece na indústria quando se quer soldar uma peça com geometria complexa. Esta tarefa pode não ser muito complicada quando o soldador humano tem que executá-la, entretanto, se a peça tiver uma geometria complexa, for de grande dimensão ou de peso considerável recorre-se ao auxílio de um posicionador ou da robotização do processo.

Em função da geometria da peça, da disposição dos cordões, sua seqüência de soldagem, posicionamento da tocha e outros fatores inerentes ao processo tornam-se difícil a execução da tarefa com um único robô, justifica-se assim o uso de um segundo, ou até mais robôs, para a realização da tarefa de forma conjunta, ou seja, um cooperando com o outro, ou outros (Henriques, 2005).

## 2.5.1 Ferramentas para Supervisão - Supervisão e Controle de Células de Manufatura Integradas

Segundo Cosso (2002), atualmente pode-se definir um sistema supervisorio como uma interface homem máquina (IHM) amigável, com recursos tecnológicos capazes de controlar e/ ou supervisionar um sistema automatizado.

Portanto, um Sistema Supervisorio é também a rigor, um sistema de comunicação no sentido mais amplo da palavra, pois engloba a visualização de todo o processo, que aliado a um sistema de informação tem a finalidade de manter um banco de dados atualizado, fornecendo em tempo real o posicionamento do sistema e, acoplado a uma interface com o usuário, pode interagir em uma intervenção e/ ou controle, e ainda conectar a parte operacional dos processos com os sistemas mais altos em hierarquia de planejamento. A figura 2.11 mostra uma representação de um sistema supervisorio, segundo esta visão.

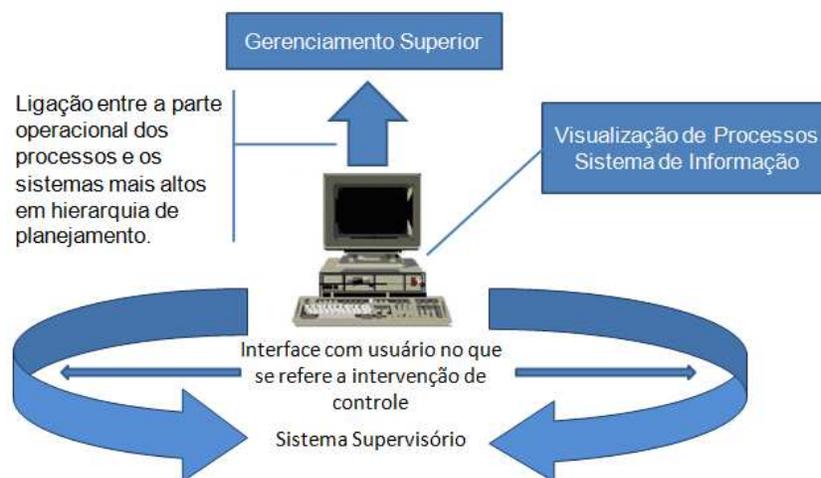


Figura 2.11 Representação de um Sistema Supervisorio

Segundo Silva et al. (1998), nos sistemas automatizados o controle pode ser organizado nos seguintes níveis: Planta, Controle Local, Supervisão, Ordenação e Planejamento. A figura 2.12 ilustra estes níveis, cuja forma piramidal referencia tanto a quantidade de recursos humanos envolvidos em cada nível, como também o número de controladores utilizados, ou seja, um controlador do nível supervisão controla diversos controladores do nível Controle Local.



Figura 2.12 Sistema de Controle - Níveis

Nos Sistemas Automatizados, a integração entre os postos de trabalho e a gestão de produção realizada a partir do Sistema de Supervisão, tem como uma das tarefas receber os dados do “chão de fábrica” e colocá-los a disposição dos níveis superiores de gerenciamento. Esta integração possibilita o acompanhamento em tempo real de variáveis e estados representativos das operações em curso no chão da fábrica (postos de trabalho), com a finalidade de tomada de decisões operacionais, otimização dos processos e criação de históricos. As aquisições dos dados a serem repassados ao Sistema Supervisório provenientes dos postos de trabalho, são realizados por CLP’s e dispositivos de controle com interfaces de aquisição de informações, ou seja, o Controle Local interage com os dispositivos físicos da Planta (Aihara, 2005).

O nível de Supervisão possui como características principais agir sobre o sistema de controle local de modo a executar as tarefas determinadas pelo nível de controle superior e supervisionar a evolução do processo na planta (Aihara, 2005).

Desta forma, o sistema de controle no nível supervisão é conhecido como sistema supervisório. Este sistema revela-se de crucial importância na estrutura de gestão de Sistemas, e, por isso, deixou de ser visto como mera ferramenta operacional ou de engenharia, e passou a ser visto como uma importante fonte de informação e controle (Oliveira, 2008).

Através da evolução tecnológica, os computadores passaram a assumir um papel de gestão na aquisição e tratamento de dados, permitindo a sua visualização em períodos curtos de tempo e ainda permitindo a geração de funções de controle complexas.

Dentro desta evolução tecnológica, a evolução dos sistemas de comunicação através das redes de comunicação, permite tal controle, uma vez que a mesma é utilizada como plataforma pelos sistemas supervisórios para a transferência de informações.

Atualmente os sistemas supervisórios estão sendo utilizados para automatizar a monitoração e o controle de Sistemas Automatizados, através do recolhimento de dados em ambientes complexos, podendo estes ambientes estarem eventualmente dispersos geograficamente, além de apresentar uma visualização de modo amigável para o operador, através de Interface Homem-Máquina (IHM) altamente sofisticada (Freitas, 2004).

Em um ambiente industrial cada vez mais complexo e competitivo, os fatores relacionados com a disponibilidade e segurança da informação assumem elevada relevância tornando-se necessária a garantia de que a informação estará disponível e segura, quando necessária, independentemente da localização geográfica. Portanto, implementações de mecanismos de acessibilidade, mecanismos de segurança e mecanismos de tolerância a falhas são importantes.

### **2.5.2 Plataforma Colaborativa - Projeto Kyatera**

O Projeto Kyatera está inserido no programa TIDIA que tem como principal objetivo estabelecer uma rede de fibras ópticas interligando laboratórios de pesquisa, desenvolvimento e demonstração de tecnologias pela Internet. Entre os objetivos deste projeto, está a implantação de laboratórios controlados através da Internet para o desenvolvimento de experimentos remotos na área de automação da manufatura e robótica ao *WebLab*.

### 2.5.2.1 Projeto Kyatera (Laboratório Virtual em Automação)

Os laboratórios virtuais mostram-se como uma solução para a parte prática do processo de aprendizagem, pois oferecem a possibilidade de interagir, criar e/ ou modificar novas informações e conteúdos em ensino e pesquisa de Engenharia na área de Automação Industrial. D'Abreu (2002) afirma que laboratórios de simulação de uma linha de produção propiciam uma vivência de ambiente de fábrica dentro de um laboratório da universidade.

Queiroz (1998) classifica estes laboratórios remotos segundo o nível de interação entre usuário e o ambiente de acordo com três tipos:

- *Nível hipermídia:* laboratórios que apresentam nível de interação entre o usuário e o ambiente remoto baixo, permitindo somente à captura de informações que se encontram distantes do usuário (textos, imagens ou vídeo sobre o conteúdo didático a ser ensinado).
- *Nível simulação:* laboratórios que apresentam um maior nível de interação com o usuário, fornecendo a capacidade de fazer uma simulação das experiências que seriam realizadas no laboratório real, em que o usuário envia dados para o experimento previamente implementado e recebe imagens, gráficos e/ ou áudio, de acordo com o processo. Na maioria das vezes o laboratório virtual de simulação está diretamente relacionado ao de hipermídia, pois simula a parte prática de um estudo teórico. A parte teórica é apresentada no formato de hipermídia e a simulação é considerada um complemento ao conteúdo didático.
- *Nível tele-presença real:* são laboratórios capazes de interagir com o ambiente remoto e realizar experimentos reais sobre o assunto a ser estudado, havendo a existência de um laboratório físico real que constitui o laboratório virtual por meio de uma camada de abstração de entrada e saída (E/S) e o meio de telecomunicação.

O Laboratório de Automação Integrada e Robótica da Faculdade de Engenharia Mecânica da UNICAMP é um dos participantes deste projeto e tem como principal objetivo a implementação de um laboratório virtual em Automação baseado na Plataforma LabVIEW, permitindo ao usuário a utilização colaborativa multi-usuários de plataformas didáticas disponibilizadas em rede cooperativa de alta velocidade nas instituições EESC-USP, UNESP e

UNICAMP, permitindo assim a capacitação na área através de experimentos remotos e/ ou virtuais (figura 2.13).

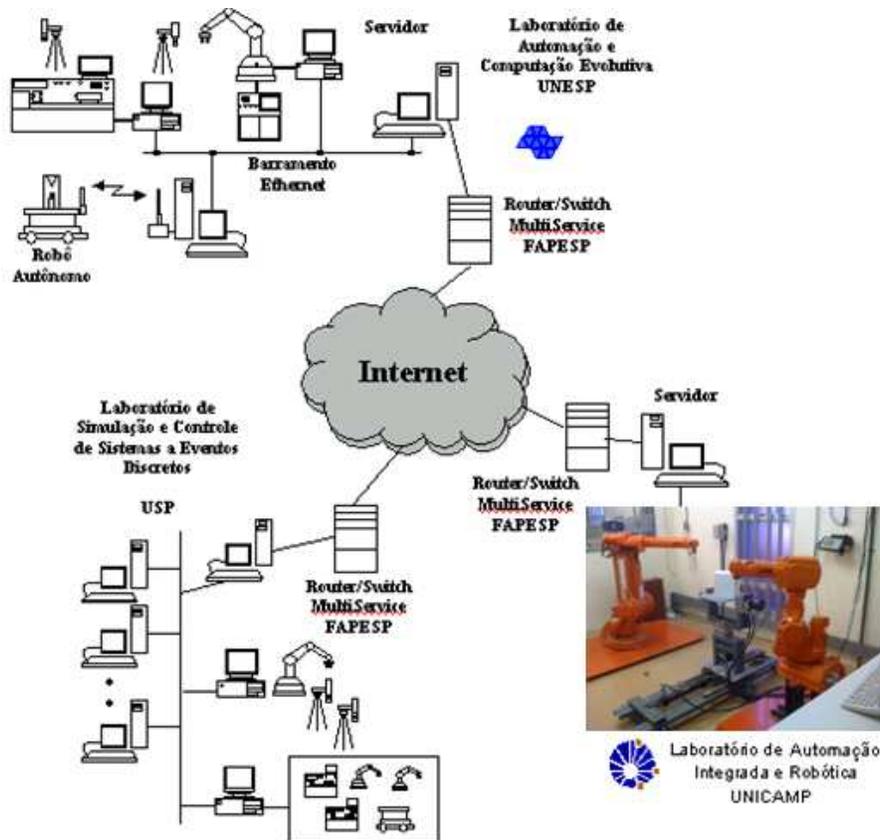


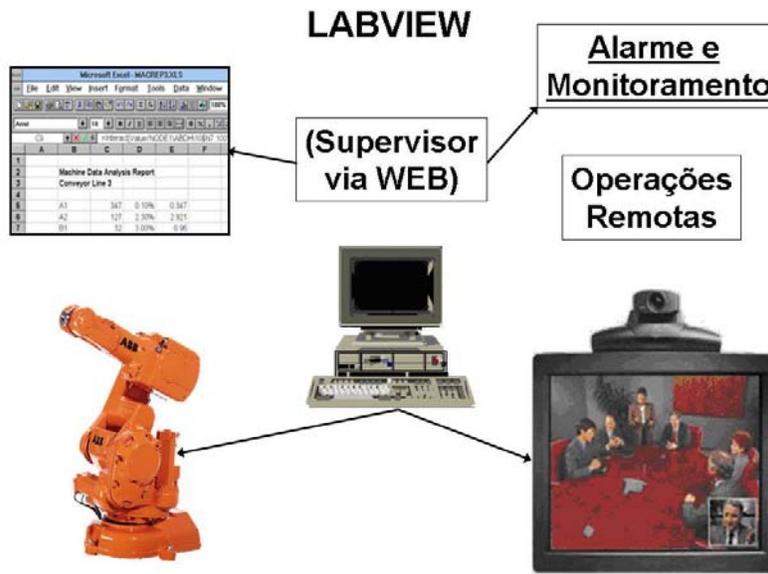
Figura 2.13: Arquitetura Proposta para Funcionamento do *WebLab*

### 2.5.2.2 Arquitetura WEB Proposta para o Projeto Kyatera

A figura 2.14a apresenta a rede cooperativa implementada a partir da utilização de um robô industrial através de sistema de supervisão remota controlada via *WEB*. Esta célula é composta de um robô industrial (ABB IRB 1400<sup>TM</sup>) e um CLP industrial (KOYO<sup>TM</sup>) que se comunica com o *software* LabVIEW<sup>TM</sup> através de configuração *OPC Server*. Ao mesmo tempo está sendo utilizado uma *WEBCAM* com IP fixo, permitindo assim sua fácil visualização via *WEB*.

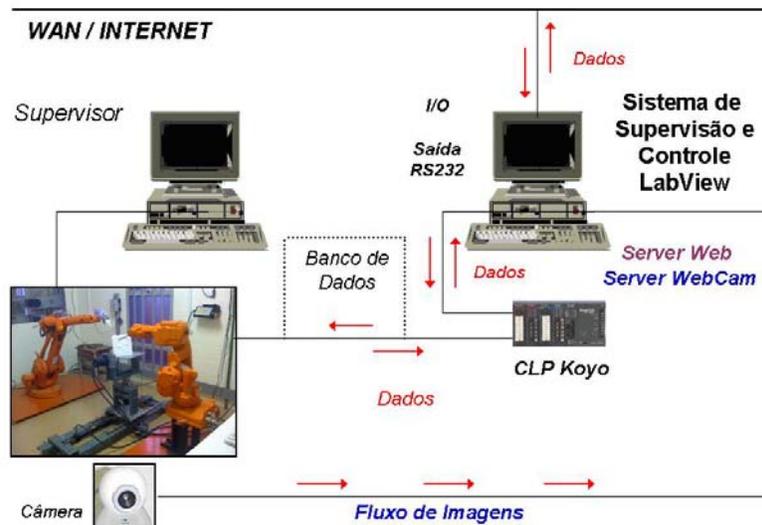
Uma das etapas principais desse trabalho de pesquisa foi a realização de uma série de testes envolvendo a integração do *software* LabVIEW com dispositivos externos (figura 2.14b), dentre eles:

- Visualização e controle da *webcam* com o LabVIEW, Comunicação de variáveis (VI) da CLP Koyo com o *software* de supervisão e controle implementado em LabVIEW, realizados através do aplicativo *OPC Server*, Publicação de VI do LabVIEW na rede do Laboratório e externa.



a) Proposta de Aplicação WEB

**Tele-operação a distância**



b) Arquitetura de Controle e Supervisão Remota

Figura 2.14: Proposta de Implementação - Projeto Kyatera

## **2.6 Considerações finais**

Neste capítulo foram revisados os principais conceitos utilizados para modelagem de células automatizadas para a integração de sistemas colaborativos a partir da consulta de trabalhos recentemente publicados, contendo programação *off-line*, ferramentas de modelagem sequencial, ferramentas de supervisão e ferramentas dedicadas.

O próximo capítulo apresenta o estudo realizado com a coordenação e integração de dois robôs industriais e uma mesa (dispositivo mecatrônico) com três graus de liberdade (robô PRR).

## **CAPÍTULO 3**

### **DISPOSITIVO MECATRÔNICO - PLATAFORMA ROBÓTICA DE 3 GRAUS DE LIBERDADE**

Frequentemente, a integração de robôs e dispositivos mecatrônicos dentro de Células Flexíveis de Manufatura envolve metodologias usando os conceitos de automação. O ambiente modelado por CAD pode ser associado com o desempenho do controle de robô incluindo equipamentos e mecanismos, modelagem matemática do robô (cinemática direta e inversa) e seus componentes acoplados, como também, a integração e coordenação dos movimentos do robô com outros dispositivos.

Uma aplicação particular pode ser vista na figura 3.1, que é baseada na coordenação e integração de dois robôs industriais e uma mesa (dispositivo mecatrônico) com três graus de liberdade (robô PRR) desenvolvida para auxiliar em trabalhos como: soldagem, usinagem. Esta mesa pode auxiliar em tarefas em que os manipuladores tradicionais têm dificuldade para alcançar algumas partes da peça a ser trabalhada. Para isso, a mesa é sincronizada com os manipuladores, permitindo a realização de tarefas complexas.

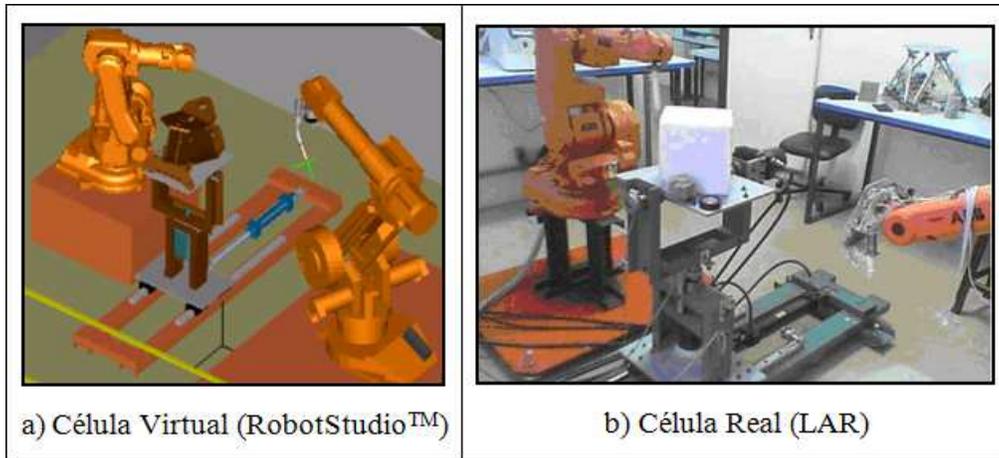


Figura 3.1: Célula Flexível de Manufatura

### 3.1 Descrição dos Equipamentos da Plataforma Hidráulica

Nesta subseção será apresentada a mesa de posicionamento (figura 3.2). A plataforma possui um atuador hidráulico linear e dois atuadores hidráulicos rotacionais. No entanto, para determinar a posição de cada junta foram adicionados potenciômetros.

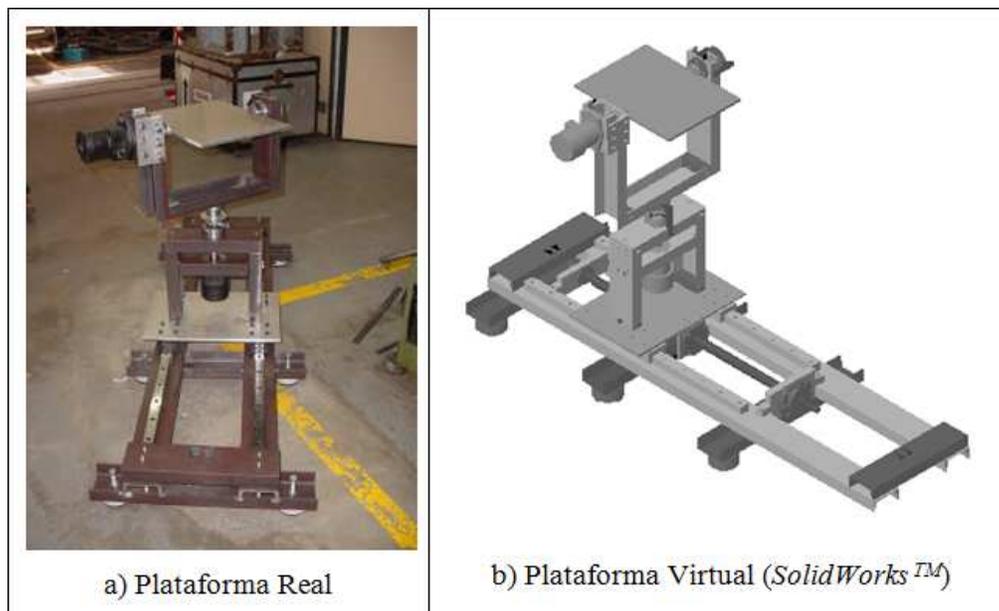


Figura 3.2: Plataforma Hidráulica Robótica para Posicionamento

Os encoders incrementais foram acoplados nas duas articulações rotacionais da plataforma robótica para processar a transdução da posição angular da junta à qual o transdutor foi acoplado. No anexo B mostra alguns conceitos dos sensores utilizados.

Ainda na parte construtiva da plataforma hidráulica, o sistema contém: reservatório hidráulico onde há o condicionamento do óleo, filtragem, bomba hidráulica, eletro-válvulas, mangueiras, conexões, manômetro, conforme mostrado na figura 3.3.

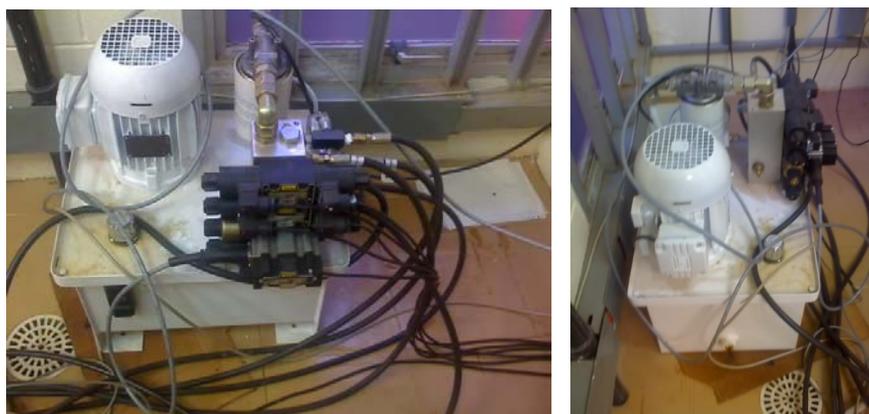


Figura 3.3: Reservatório Hidráulico

Em muitas aplicações industriais, a tarefa do robô é programada por aprendizagem, sem a necessidade de um modelo geométrico. Desse modo, sua trajetória é definida através de um conjunto de ângulos associados ao movimento angular de cada grau de liberdade do robô. Após a interpolação, estes ângulos atuarão como sinais de referência para o controlador de posição de cada junta que compara os sinais que provem dos sensores de posição de cada junta (David e Rosário, 1998). Isso faz com que a configuração da mesa seja controlada a partir de um valor desejado, independente do movimento desejado e da carga transportada pela mesma.

No entanto, os valores das variáveis articulares utilizados como sinal de referência na malha de controle de posição das juntas quando comparados com os valores das juntas podem traduzir em um erro, que aumenta com a sua velocidade de operação. Assim, a concepção de um controlador de posição para qualquer tipo de robô industrial exige o conhecimento da precisão cinemática de seu movimento.

### 3.2 Descrição da Plataforma

Freqüentemente, a integração de robôs e dispositivos mecatrônicos dentro de Células Flexíveis de Manufatura envolve metodologias usando os conceitos de integração de sistemas automatizados. O ambiente modelado por CAD (*Computer Aided Design*) pode ser associado com o desempenho do controle de robô incluindo equipamentos e mecanismos, modelagem matemática do robô (cinemática direta e inversa) e seus componentes acoplados, como também, a integração e coordenação dos movimentos do robô com outros dispositivos.

Uma aplicação particular pode ser vista na figura 3.4, que é baseada na coordenação e integração de dois robôs industriais e uma mesa de posicionamento (plataforma hidráulica). Esse dispositivo apresenta três graus de liberdade (robô PRR) e foi desenvolvido para auxiliar os robôs manipuladores em trabalhos genéricos de dispositivos mecânicos (trajetórias complexas) (Henriques, 2005). Esta mesa de posicionamento pode auxiliar em tarefas nas quais os manipuladores tradicionais têm dificuldade em alcançar algumas partes da peça a ser trabalhada.



Figura 3.4: Célula Flexível de Manufatura

Para melhor compreensão do problema, aqui é apresentada a mesa de posicionamento em estudo (figura 3.5). Essa mesa possui três atuadores hidráulicos.

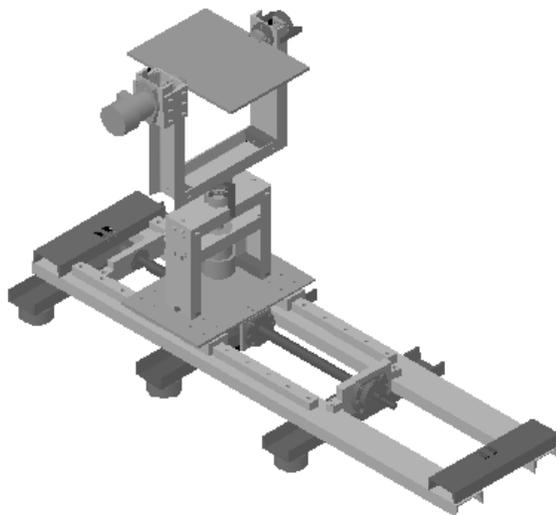


Figura 3.5: Mesa Robótica de Posicionamento

### 3.3 Modelagem Matemática

As tarefas realizadas pelos robôs baseiam-se no movimento independente de cada grau de liberdade, coordenados a partir de um planejador de trajetórias baseado no seu modelo cinemático. Na maioria dos casos, a programação das tarefas é realizada através do *modo aprendizagem*, onde ocorre a armazenagem da seqüência de movimentos independentes de cada junta, até alcançar a posição final desejada. A partir do conhecimento destas posições articulares, é facilmente implementado um gerador de referências (perfil de velocidades) baseado nas características cinemáticas das juntas.

Para realização de tarefas a nível sistema de coordenadas da mesa de posicionamento (sistema cartesiano) e para geração dos sinais de referência para o controlador de posição de cada junta robótica do sistema mecatrônico em estudo, torna-se necessário o estabelecimento de um modelo matemático baseado na cinemática direta e inversa do sistema. Portanto, o controle de um robô necessita de procedimentos para transformar as coordenadas articulares em cartesianas, quando se deseja realizar o controle através de um referencial cartesiano, conforme observado na figura 3.6. Na figura 3.7 mostra a estrutura dos movimentos relacionados a cada grau de liberdade.

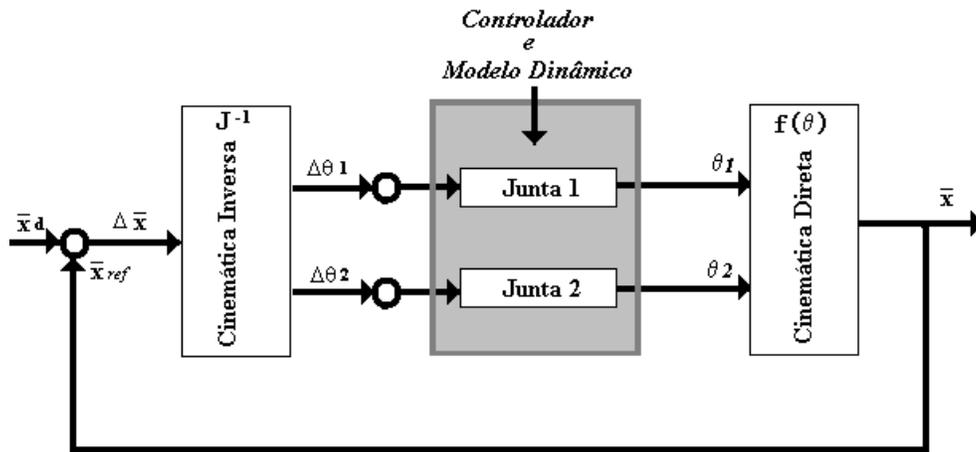


Figura 3.6: Estrutura de Controle

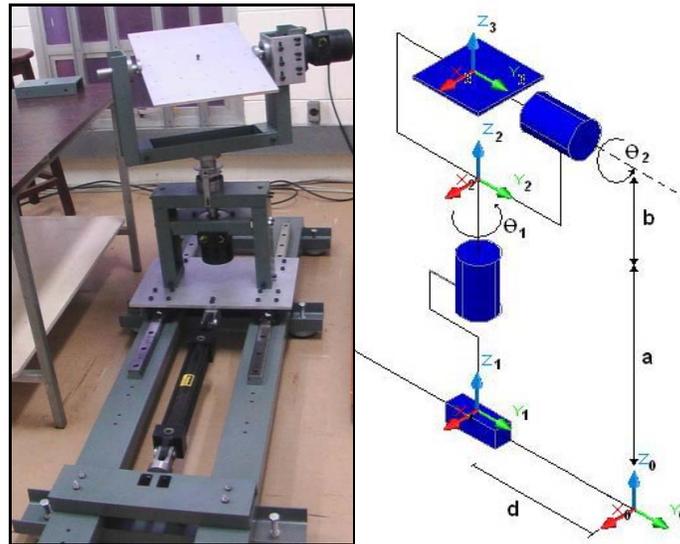


Figura 3.7: Plataforma 3DGL e Movimentos Relacionados.

### 3.3.1 Modelagem Cinemática

Dada uma configuração de ângulos das juntas de um robô, o cálculo da posição e orientação da extremidade do manipulador é chamado de cinemática direta, e é sempre possível obter a solução do problema cinemático direto. Em contrapartida, o problema cinemático inverso é de solução um pouco mais complexa, dependendo das características do robô. Além disso, múltiplas soluções do problema e singularidades podem ocorrer. A figura 3.8 ilustra a relação entre as duas cinemáticas.



**i) Matrizes de Rotação**

$$T_{\theta_1} = \begin{pmatrix} \cos \theta_1 & -\text{sen} \theta_1 & 0 \\ \text{sen} \theta_1 & \cos \theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \text{ devido à rotação em torno do eixo } z \text{ (base).}$$

$$T_{\theta_2} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_2 & \text{sen} \theta_2 \\ 0 & \text{sen} \theta_2 & \cos \theta_2 \end{pmatrix}, \text{ devido à rotação em torno do eixo } x \text{ (rotação da mesa).}$$

**ii) Cálculo dos Vetores Locais**

$$O_0 O_1 = \begin{Bmatrix} 0 \\ 0 \\ R \end{Bmatrix} \quad O_1 O_2 = \begin{Bmatrix} 0 \\ 0 \\ L_h \end{Bmatrix}$$

$$O_1 = O_0 + T_{\theta_1} * O_{11}$$

$$O_2 = O_1 + T_{\theta_1} * T_{\theta_2} * O_{12}$$

$$\text{Logo, } O_2 = O_0 + O_1 + \begin{pmatrix} -S\theta_1 * S\theta_2 * L_h \\ C\theta_1 * S\theta_2 * L_h \\ C\theta_2 * L_h \end{pmatrix}$$

$O_0$  = posicionamento da mesa em relação ao referencial inercial

$O_1$  = posicionamento da haste em relação ao centro da mesa (x, y)

Fator de Conversão do Encoder = X graus/ pulsos

Posição em graus = No de pulsos \* Fator de Conversão do *Encoder*

$$\theta_1 = \theta_{1init} + \theta_{1lido}$$

$$\theta_2 = \theta_{2init} + \theta_{2lido}$$

A partir da definição do ângulo fixo de posicionamento ou de um perfil de deslocamento

da mesa ( $\theta_2$ ), é realizada a movimentação de uma haste fixa na mesa ( $L_h$ ) a partir do movimento de rotação da base da mesa ( $\theta_1$ ) descrevendo um cone de revolução ( $\theta_2$  diferente de  $0$ ), ou um cilindro de revolução ( $\theta_2$  igual a  $0$ , no caso da haste estar posicionada a uma distância  $d$  do centro da mesa).

### 3.3.3 Cinemática Inversa

Considerando que a rotação da haste sobre a mesa (figura 3.10), esta realizará um cone no espaço. Para a obtenção do modelo cinemático inverso consideraremos como parâmetros de entrada o raio do cone de revolução da haste (cilindro, no caso de mudança do centro) de revolução ( $x$ ), posição atual da mesa ( $\theta_{1init}$ ), valor e sentido de deslocamento da haste em graus ( $\theta_{1deslocl}$ ) e velocidade de rotação da base da mesa. No caso da descrição de um cone espacial, o programa calculará automaticamente, o ângulo de inclinação desejado para mesa ( $a = \theta_2$ ), permitindo a rotação da base da mesa a partir da posição inicial até a posição final desejada ( $\theta_{1init}$  e  $\theta_{1final}$ ), ou seja:

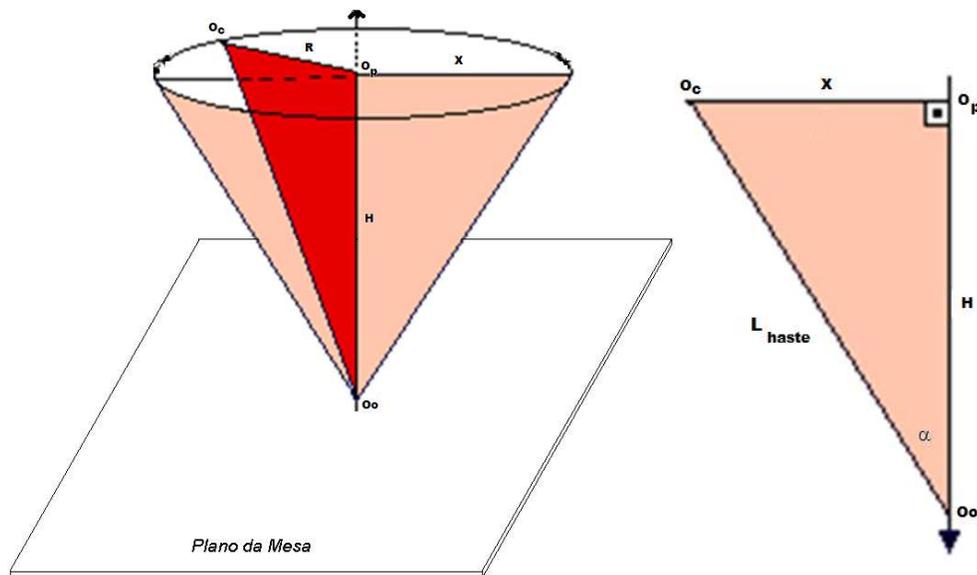


Figura 3.10: Cone de Rotação

#### *i - Parâmetros*

X: Raio do círculo de Revolução

$L_h$ : Comprimento da haste fixa a base da mesa

$\theta_{1desloc}$ : Deslocamento desejado da base da mesa em graus

$\theta_{1init}$ : posição inicial

## *ii - Equações do Modelo*

### **• Caso 1: Cone Espacial com Haste no Centro de Rotação da Mesa:**

$$\theta_2 = \text{ATAN2} \left( \frac{X}{\sqrt{L_h^2 - X^2}} \right)$$

$$\theta_{1final} = \theta_{1init} + \theta_{1desloc}$$

### **• Caso 2: Cone Espacial com Haste Deslocada do Centro de Rotação da Mesa:**

$X$ : Raio do círculo de Revolução (distância da haste até o centro de rotação da mesa)

$X_o$ : Distância da haste ao centro de revolução da mesa

$L_h$ : Comprimento da haste fixa a base da mesa

$\theta_{1desloc}$ : Deslocamento desejado da base da mesa em graus

$$\theta_2 = \text{ATAN2} \left( \frac{X}{\sqrt{L_h^2 - X^2}} \right)$$

$$\theta_{1final} = \theta_{1init} + \theta_{1desloc}$$

### **• Caso 3: Cilindro de Revolução com Haste Deslocada do Centro de Rotação da Mesa**

$X$ : Raio do círculo de Revolução (distância da haste até o centro de rotação da mesa)

$L_h$ : Comprimento da haste fixa a base da mesa

$\theta_{1desloc}$ : Deslocamento desejado da base da mesa em graus

$$\theta_2 = 0$$

$$\theta_{1final} = \theta_{1init} + \theta_{1desloc}$$

## **3.4 Simulador Virtual**

O sistema de controle, que consiste em malha de controle em cascata para cada eixo, foi construído em blocos do Simulink<sup>®</sup>. O conjunto de malhas de controle de posição, velocidade e torque podem fazer parte do modelo do sistema de acionamento e controle de uma junta robótica.

O controle de posição do manipulador pode ser implementado através de realimentação para cada junta isolada, requerendo o modelo de cada junta. Por fim, todas as juntas devem ser coordenadas como foi apresentada na figura 3.6, para isso o modelo dinâmico da estrutura deve ser definido.

No problema em estudo, foi implementada apenas a malha de controle de posição acoplada ao modelo completo de uma junta robótica, utilizando arquitetura aberta, de modo à facilmente serem implementados diferentes estratégias de controle, para posterior simulação, análise e comparação de desempenhos.

Outros elementos do sistema mecatrônico (incluindo possíveis cargas externas) são representados por modelos não-lineares, um para cada motor. O simulador também inclui um módulo de geração de trajetórias, proporcionando às juntas com as trajetórias dos eixos como sinais de referência para o controlador. E ainda, uma interface gráfica é disponibilizada, mostrando resultados dos movimentos obtidos através de trajetórias simples (figura 3.11).

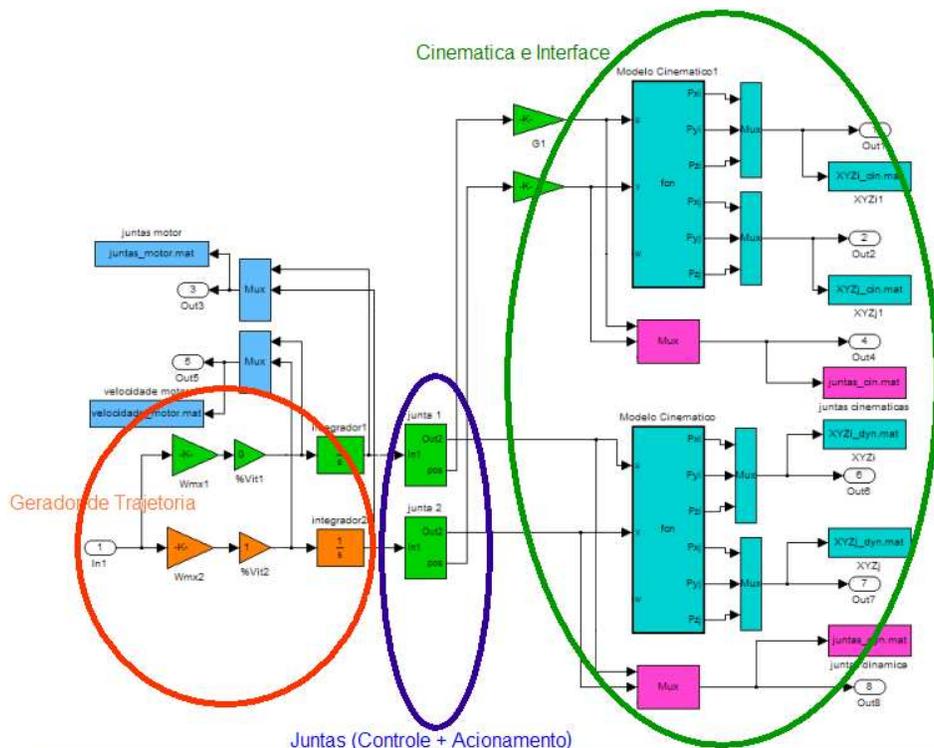


Figura 3.11: Simulador de uma Mesa de 2 GDL usando Ambiente Simulink®

### 3.4.1 Módulo de Geração de Trajetórias

O módulo de geração de trajetórias foi implementado em Matlab<sup>TM</sup> com o objetivo de geração de sinais de referência para cada junta (perfil de velocidade), baseado nas características cinemáticas das mesmas. Um integrador permite a obtenção da referência de posição de cada junta. A figura 3.12 mostra um perfil de trajetória a ser seguida por cada junta, com a aceleração, velocidade constante e frenagem em um intervalo de 10 segundos. Os tempos de aceleração e frenagem escolhidos foram baseados nas características dinâmicas do sistema em estudo (constante de tempo mecânica). O módulo de geração de trajetórias pode ser implementado através da cinemática inversa a partir do modelo cinemático da estrutura a ser controlada utilizando o Jacobiano inverso, conforme foi apresentado anteriormente.

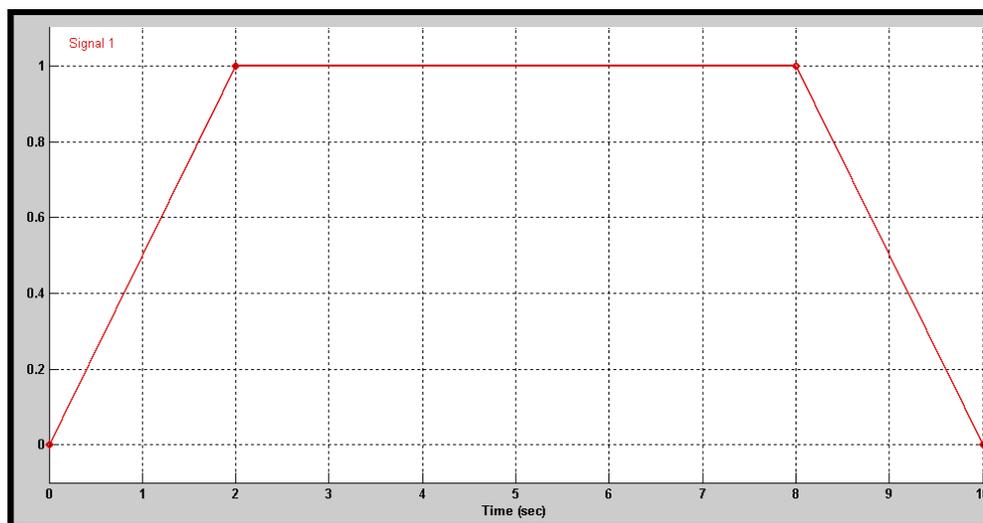


Figura 3.12: Sinal de Referência de Posição para cada Junta.

### 3.4.2 Módulo Acionamento

O módulo de acionamento (figura 3.13) apresenta as componentes relativas ao modelo elétrico e mecânico do sistema de acionamento, e ainda o controlador de posição.

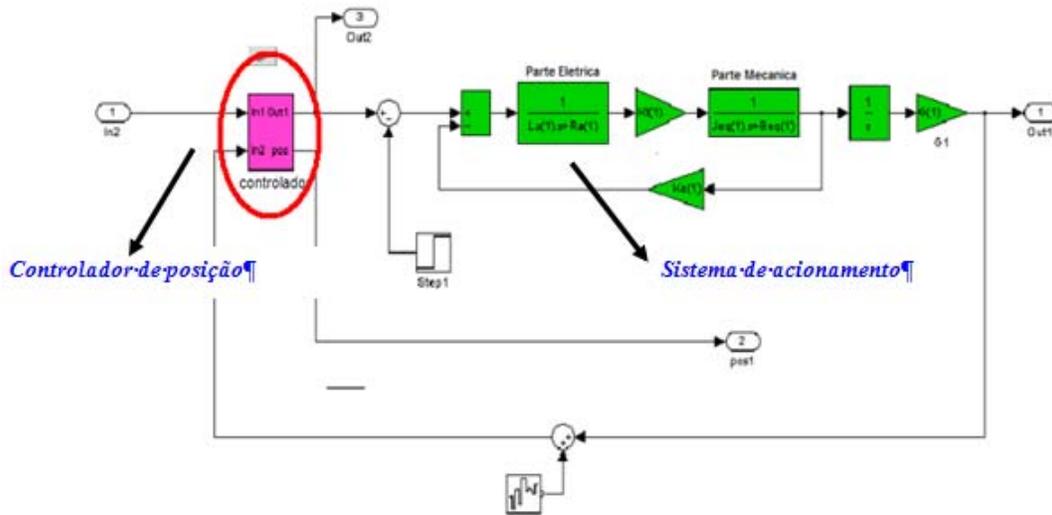


Figura 3.13: Malha de Controle de Posição de uma Junta.

### 3.4.3 Módulo Cinemático

O modelo cinemático do sistema mecatrônico em estudo foi implementado utilizando funções *S-function*, dentro do ambiente Matlab<sup>®</sup> integrados nos blocos do Simulink<sup>®</sup>, conforme a figura 3.14.

```

1 function [Pxi,Pyi,Pzi,Pxj,Pyj,Pzj] = fcn(v,w)
2 % Função modelo cinemático do Robo
3 % Visualização espacial da mesa (2 rotações e 1 translação)
4
5
6 wo = 30; % angulo de orientação da mesa
7 Tet1= v; % rotação 1
8 v=30;
9 Tet2= w + wo; % rotação 2
10
11 % Modelo Cinemático do robo
12 Cos_Tet1 = cos (Tet1); % rotacao em Z
13 Sin_Tet1 = sin (Tet1);
14 Cos_Tet2 = cos (Tet2); % rotacao em Y
15 Sin_Tet2 = sin (Tet2);
16
17 % dimensões da haste na mesa (origem do referencial em relação ao centro da mesa)
18 Xi = 100; % mm
19 Yi = 100; % mm
20 Lh = 100; % mm
21
22 % vetor de posição
23 Pxi = Xi - Lh * (Sin_Tet1)*(Sin_Tet2);
24 Pyi = Yi + Lh * (Cos_Tet1)*(Sin_Tet2);
25 Pzi = (Cos_Tet2)* Lh;
26
27 Pxj = Xi - Lh* (Sin_Tet1)*(Sin_Tet2);
28 Pyj = Yi + Lh* (Cos_Tet1)*(Sin_Tet2);
29 Pzj = (Cos_Tet2)* Lh;
30

```

Figura 3.14: Função S-function - Matlab<sup>®</sup> Implementada nos Blocos do Simulink<sup>®</sup>

### 3.4.4 Interface Gráfica

O simulador implementado em ambiente Simulink<sup>®</sup> permite a visualização temporal das saídas e entradas do sistema em estudo. Para melhor compreender e analisar o comportamento espacial do sistema torna-se imprescindível a implementação de um simulador gráfico de movimentos espaciais que será descrito a seguir.

Assim, após as simulações realizadas no domínio do tempo através do simulador implementado em Simulink<sup>®</sup>, são obtidos arquivos de dados temporais correspondentes as variáveis de estudo (posição angular e cartesiana, velocidade, corrente, sinal de controle), que após tratamento conveniente, torna-se possível verificar importantes resultados disponíveis no *menu* da figura 3.15.

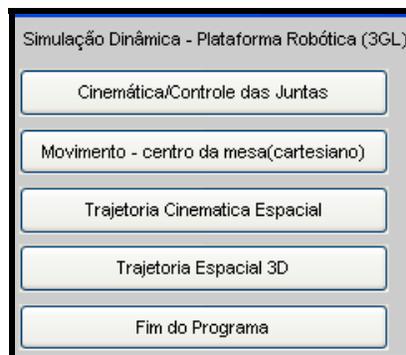


Figura 3.15: *Menu* do Simulador Robótico com 2 GDL

Para possibilitar a visualização dos movimentos em relação às coordenadas cartesianas das juntas, foi inserida uma haste de  $100\text{ mm}$  de comprimento (entre os pontos  $A$  e  $B$  da figura 3.16) centralizada na base da mesa e inclinada de um ângulo  $30^\circ$  em relação à mesma. Para efeito de melhor visualizar o comportamento do sistema em estudo, as simulações apresentadas neste capítulo mostram a haste girando em torno do eixo  $Z$ , ou seja, movimentando apenas pelo atuador 2. Como a haste encontra-se no centro da mesa, posição  $(0, 0, 0)$ , a figura projetada será um cone. Deslocando-se a haste para outra posição  $(P_x, P_y, 0)$  e fazer movimentar a junta a figura será um cilindro.

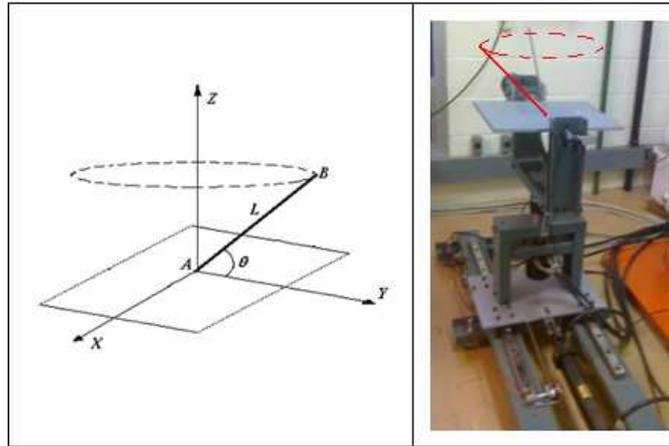


Figura 3.16: Haste colocada sobre a Mesa para Visualização dos Movimentos durante as Simulações

Na figura 3.17, apresenta o sistema de posicionamento das juntas rotacionais onde:

- na junta 1: coloca-se os ângulos para movimentação e apresenta o gráfico de velocidade e posição;
- na junta 2: a) Modulo 1: coloca-se o valor do ângulo de inclinação da mesa e automaticamente é calculado o raio da haste.  
b) Modulo 2: coloca-se o valor do raio da haste e automaticamente é calculada a inclinação da mesa.
- tem-se ainda o campo para inserir a velocidade máxima e o tempo.

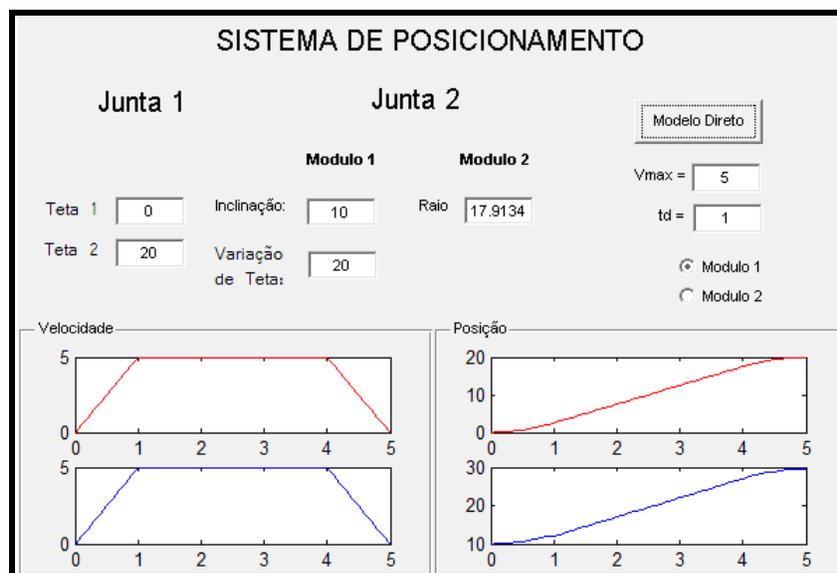


Figura 3.17: Sistema de Posicionamento

Conseqüentemente, serão apresentados alguns resultados obtidos para estas condições de simulação:

*i - Velocidade do Motor:*

Pode-se verificar o movimento de cada junta separadamente ou as três juntas conjuntamente, sendo que a velocidade máxima de cada junta é escolhida como parâmetro de entrada no simulador, como mostra a figura 3.18.

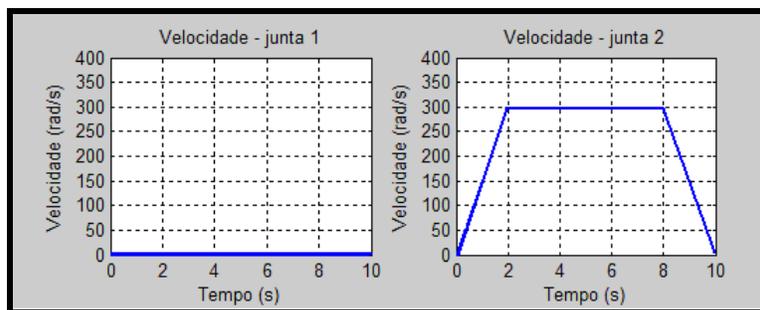


Figura 3.18: Perfil Velocidade das Juntas - Apenas Atuador 2 Acionado

*ii - Deslocamento do Motor:*

Nesse caso nas figuras 3.19 e 3.20, tem-se o movimento considerando o deslocamento cinemático e dinâmico das juntas.

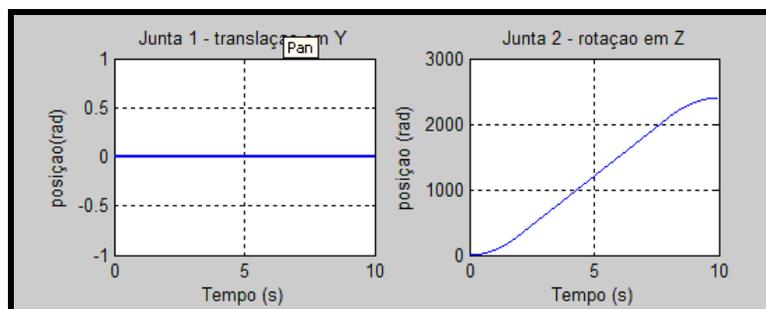


Figura 3.19: Movimento Cinemático - Apenas a Junta 2 Acionada

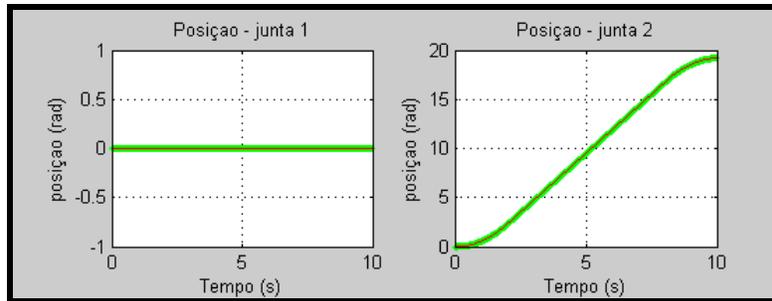


Figura 3.20: Movimento Dinâmico - Apenas a Junta 2 Acionada

*iii - Sinal de Controle*

O sinal de controle pode ser visualizado conforme a figura 3.21.

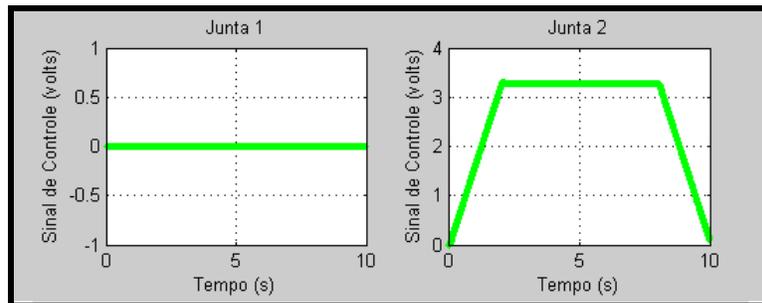


Figura 3.21: Sinal do Controle

*iv - Movimento - Centro da Mesa*

O movimento do centro da mesa diz respeito à haste posicionada sobre a mesma. A idéia é mostrar através da visualização espacial da posição apresentada pela haste em cada instante de tempo (intervalo de 10 segundos), mostrado na figura 3.22.

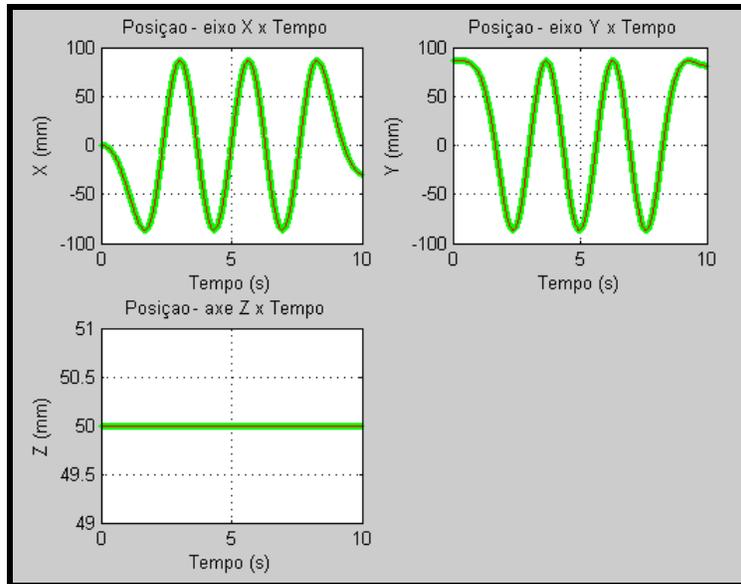


Figura 3.22: Movimento do Centro da Mesa - Apenas o Atuador 2 Acionado

*v - Trajetória Cinemática Espacial*

A figura 3.23 representa o movimento espacial da mesa a partir da haste relacionada aos eixos x, y e z.

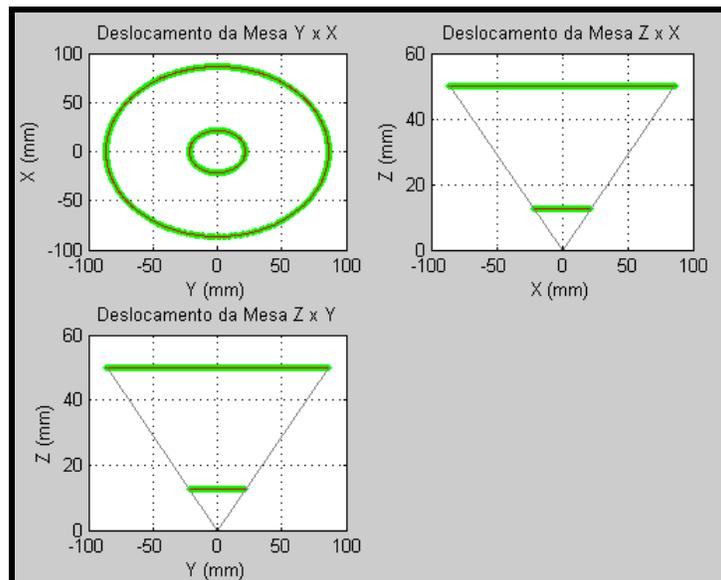


Figura 3.23: Vista dos Eixos X, Y e Z da Mesa em Movimento - Movimento Espacial da Haste

### vi - Trajetória Espacial 3D

A figura 3.24 mostra o movimento espacial da haste, permitindo verificar possíveis erros e perturbações de deslocamento no espaço cartesiano.

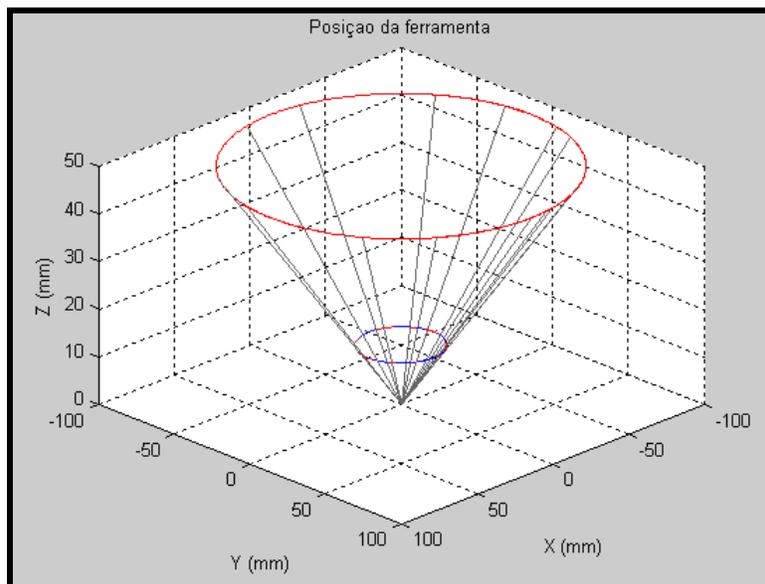


Figura 3.24: Posição Espacial da Haste - Acionamento somente do Atuador 2

### vii - Estudo do Erro:

Os erros apresentados nesse item são com relação à posição da mesa e erro do atuador, como são mostrados nas figuras 3.25 e 3.26.

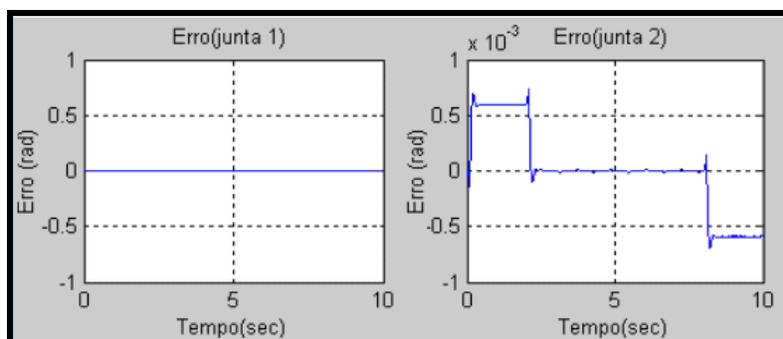


Figura 3.25: Erro do Atuador

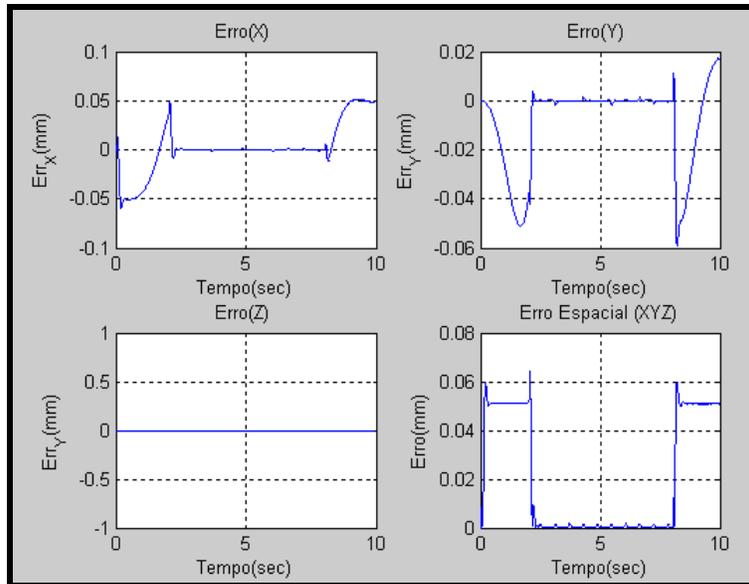


Figura 3.26: Erro da Mesa

O simulador foi idealizado em arquitetura modular e aberta em ambiente Matlab/Simulink<sup>®</sup>, facilitando a criação e modificação das bibliotecas nele utilizadas, possibilita também, a implementação de novos módulos que pode ser feita de maneira simples, isso permite uma evolução contínua do simulador e extensão de sua utilização para outros tipos de sistemas mecatrônicos.

### 3.5 Conclusão

Este capítulo apresentou as principais características do dispositivo robótico (plataforma hidráulica) e sua modelagem matemática para que fosse implementado dentro de um simulador virtual desenvolvido em ambiente Matlab/ Simulink<sup>®</sup>, com arquitetura modular e aberta.

A concepção do simulador considerou apenas as duas juntas rotacionais da plataforma (base superior), mostrando algumas simulações possíveis. O simulador desenvolvido pode facilmente ser acrescido com outros módulos. No próximo capítulo é mostrada a validação experimental: a integração da plataforma hidráulica com robôs manipuladores.

## **CAPÍTULO 4**

### **VALIDAÇÃO EXPERIMENTAL: IMPLEMENTAÇÃO DE CÉLULA AUTOMATIZADA COLABORATIVA**

Com a utilização das metodologias vistas anteriormente 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 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 E/S. 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.

Para validação deste trabalho de pesquisa foram implementados dois estudos:

- a) Plataforma *WEB* colaborativa em Automação: integração de dois robôs industriais junto com dispositivo robótico de 3 GDL para realização de tarefas colaborativas automatizadas através de *WEB*;

- b) Implementação de Sistema de Supervisão e Controle de um dispositivo robótico com 3 GDL, com ênfase na modelagem cinemática e controle de posicionamento.

#### 4.1 Descrição da Célula Robótica em Estudo

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 4.1. Assim foi implementada uma célula integrada de usinagem, constituída de dois robôs ABB: o IRB1400 com capacidade de carga de 5 Kgf; IRB140 com capacidade de carga de 5 Kgf e um dispositivo mecatrônico com 3 GDL, conforme figura 4.1.

Para a execução da programação *off-line*, consideraram-se os 3GDL do dispositivo mecatrônico como eixos externos dos robôs ABB IRB 140 e 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.

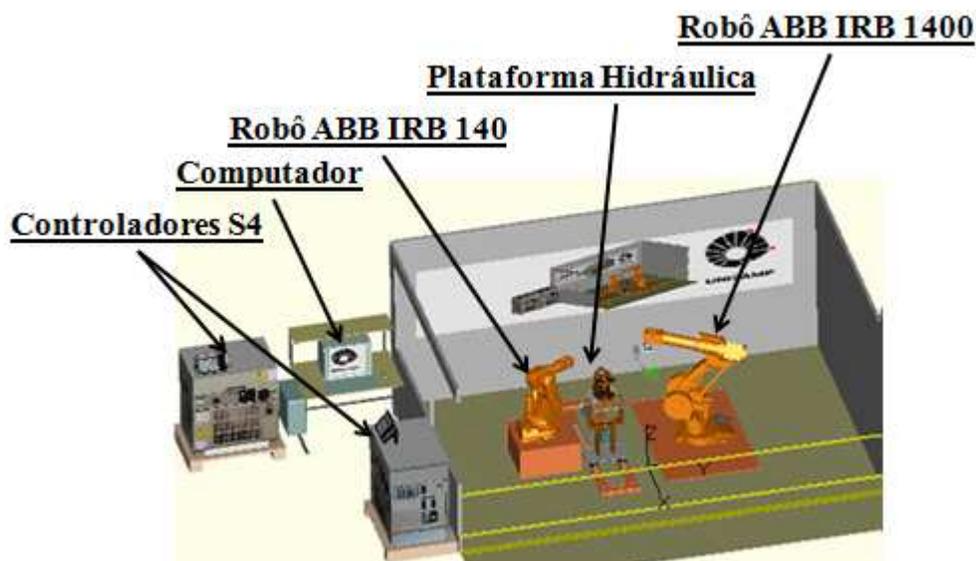


Figura 4.1 Célula Automatizada Cooperativa Virtual, a ser Implementada na FEM-UNICAMP

Para validação da plataforma robótica usada para a operação virtual foi implementado um sistema de supervisão e controle desenvolvido em ambiente LabVIEW<sup>TM</sup>, representando uma tarefa de execução de trabalho genérico complexo nas cinco faces de um cubo utilizando a

plataforma hidráulica 3 GDL descrita no capítulo anterior trabalhando cooperativamente com dois robôs industriais da ABB (IRB 140 e IRB 1400) e o estudo de acionamento e controle da plataforma 3 GDL com interface de acionamento e controle implementada em ambiente LabVIEW™, a partir da movimentação de uma haste disposta na mesa.

A figura 4.2a apresenta uma peça na mesa para ser trabalhada cooperativamente pelos dois robôs industriais, onde a peça será posicionada pelo dispositivo mecatrônico para em seguida ser trabalhada pelos robôs industriais (caso 1). Já na figura 4.2b apresenta-se uma haste colocada sob a mesa validação do modelo cinemático, interface de acionamento e controle do dispositivo mecatrônico descrito no capítulo anterior.

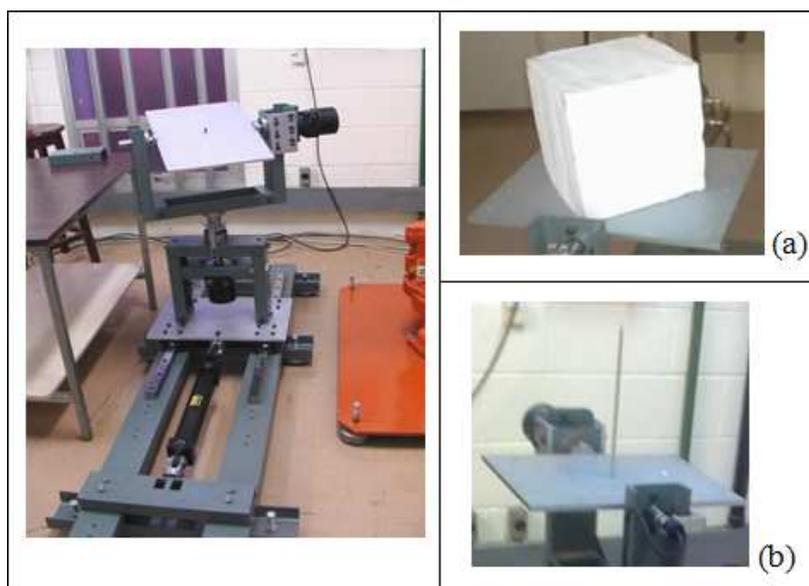


Figura 4.2: Plataforma de Posicionamento - LAR

## 4.2 Caso 1: Sistema Colaborativo em Automação

### 4.2.1 Ambiente de Validação

Uma infra-estrutura complexa para a realização de trabalhos experimentais na área de robótica nem sempre é possível de ser implementada em consequência dos altos custos envolvidos. A procura de soluções adequadas é constante, com o emprego de modelos

matemáticos capazes de representarem parte de um sistema real sem perder sua generalidade, para validar experimentalmente novas estruturas de controle.

Dentro desse propósito, foi implementado um ambiente estruturado de trabalho cooperativo envolvendo um trabalho de integração entre os dois robôs ABB e a plataforma 3 GDL de posicionamento, apresentando uma estrutura modular, hierarquizada e aberta, pode ser facilmente utilizado para integração de diferentes dispositivos mecatrônicos. A figura 4.3 apresenta a estruturação da tarefa de execução de trabalho genérico, utilizando uma mesa indexada e dois robôs industriais da ABB (IRB 140 e IRB 1400). A figura 4.4 apresenta alguns testes de validação. No anexo A segue a programação dos robôs industriais.

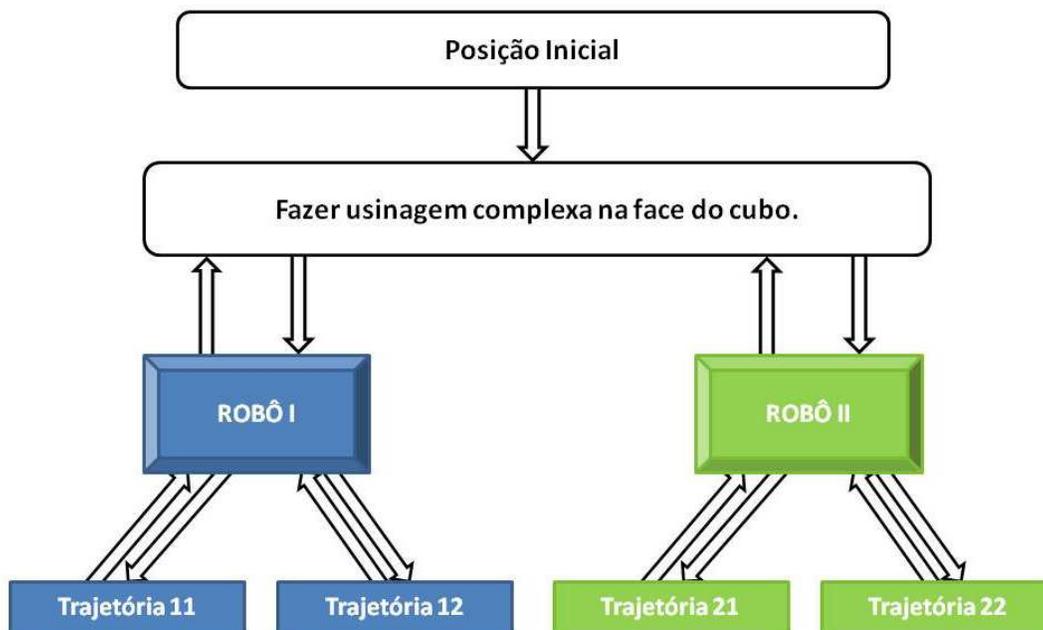


Figura 4.3: Estruturação de Tarefas

A interface de visualização no seguinte endereço [http:// 143.106.9.151](http://143.106.9.151). A figura 4.4 apresenta imagens operativas das tarefas cooperativas executadas.

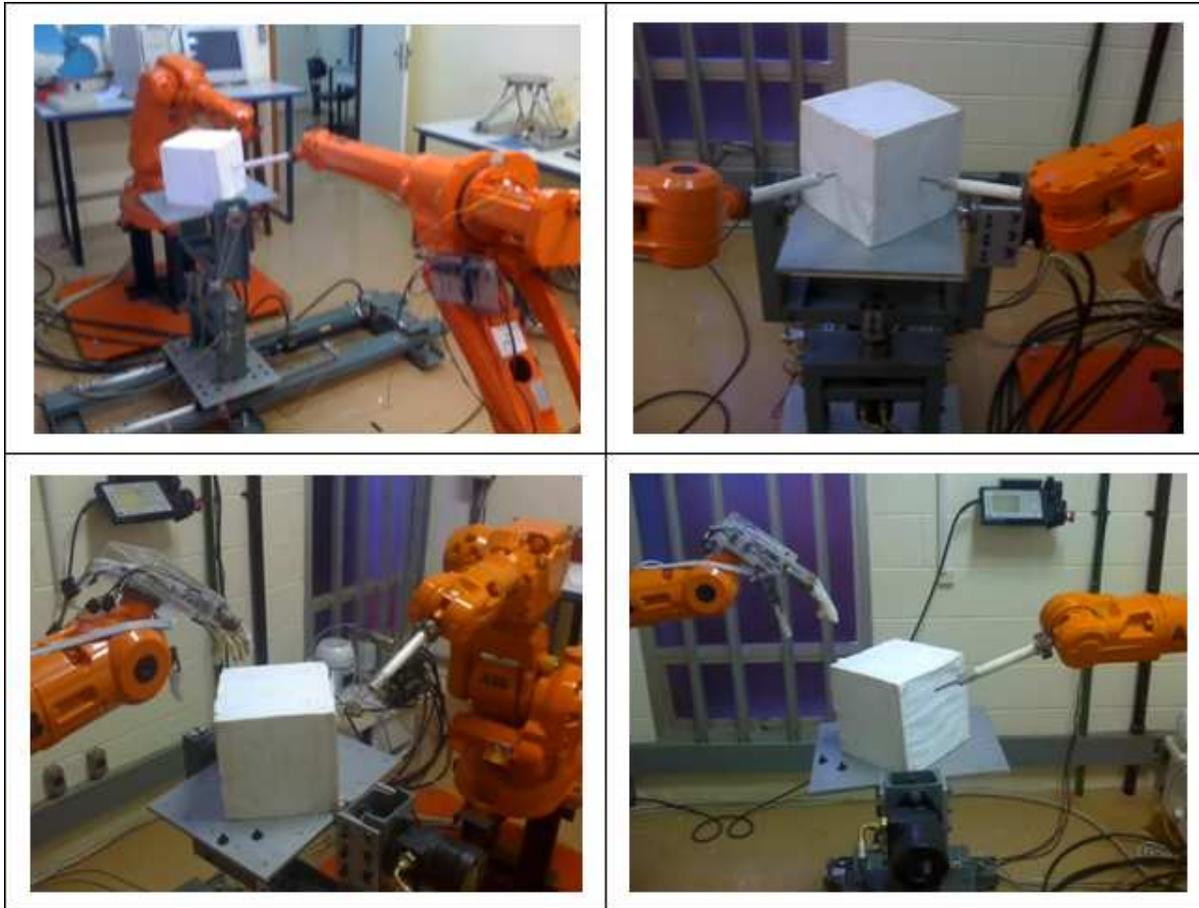


Figura 4.4: Célula Integrada de Trabalho - LAR

#### 4.2.2 Ambientes Colaborativos para Ensino e Pesquisa baseados em *WEB*

Através da evolução das Tecnologias da Informação e das Telecomunicações, a Internet passa a revolucionar a ciência, a economia e a sociedade. Desta forma, a utilização de ambientes educacionais com recursos computacionais permite um maior acesso a novos conhecimentos de maneira mais rápida, acarretando impactos sociais e novas áreas de pesquisa e desenvolvimento.

A utilização de ambientes ou plataformas de ensino deve conter informações que possibilitem que o aprendiz evolua conforme seu ritmo e flexibilidade (Traylot, Heer e Fiez, 2003). Para tanto, estes ambientes devem promover a integração de conhecimentos, inovação e experiências para a resolução de pequenos problemas, além de motivar e melhorar a visualização da continuidade do aprendizado. Visando tais aspectos a utilização de laboratórios virtuais e a

Internet passam a ser aliados no processo de aprendizagem (a figura 4.5 ilustra as relações apresentadas).

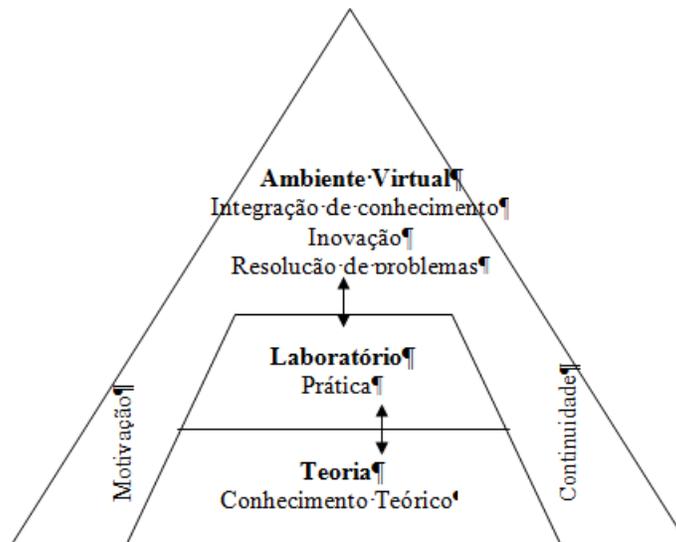


Figura 4.5: Aspectos de um Ambiente de Ensino e Pesquisa Virtual

### 4.2.3 Telas Implementadas em Ambiente LabVIEW™

As figuras 4.6, 4.7 e 4.8 apresentam as telas implementadas em LabVIEW das tarefas cooperativas executadas.



Figura 4.6: Proposta de pagina HTML - Painel de Controle Implementado em LabVIEW

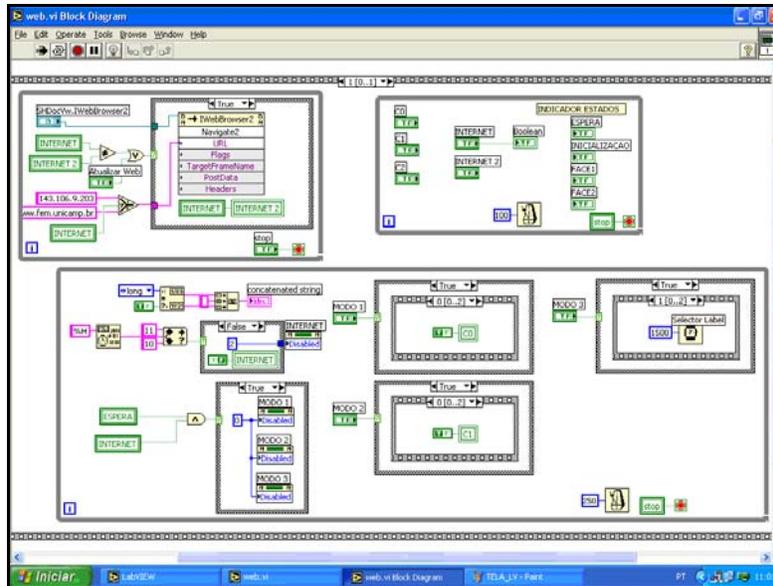


Figura 4.7: Tela típica de Implementação em LabVIEW

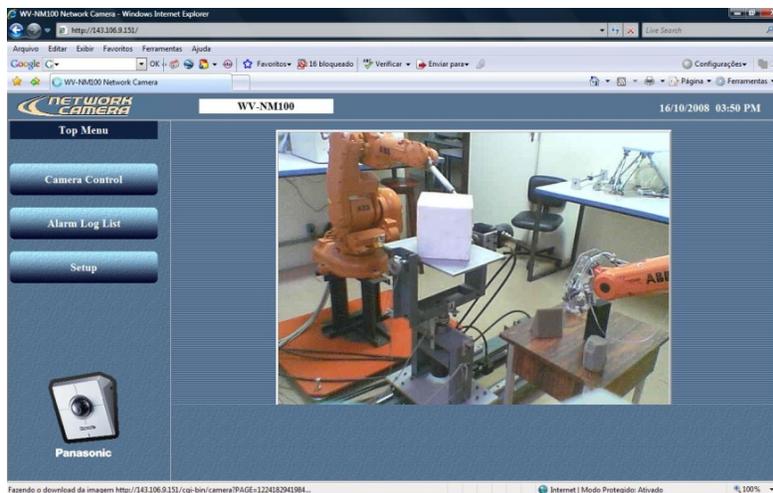


Figura 4.8: Tela de Visualização de Câmera CCD WEBCAM com IP fixo

### 4.3 Caso 2: Implementação de Sistema de Supervisão e Controle da Plataforma 3 GDL

Dentro do propósito de sistema de supervisão e controle, a implementação de um simulador através de blocos estruturados, apresentando uma estrutura modular, hierarquizada e aberta, pode ser facilmente utilizado para as simulações de diferentes dispositivos mecatrônicos.

A figura 4.9 apresenta esquematicamente o simulador implementado. Este simulador é constituído dos seguintes módulos:

- Geração de sinais de referências de trajetórias das juntas;
- Modelo cinemático direto e inverso;
- Acionamento e controle;
- Interface gráfica de modo a permitir a visualização dos resultados dos movimentos obtidos através de trajetórias de referências.

A partir da subsecção 4.3.5 serão apresentadas às telas de inicialização, posicionamento, orientação, calibração e controle.

No Laboratório de Automação Integrada e Robótica da Faculdade de Engenharia Mecânica da UNICAMP, foi implementada uma plataforma de posicionamento com três graus de liberdade (robô PRR) para trabalhar cooperativamente com dois robôs industriais (IRB 140 e IRB 1400 da ABB<sup>TM</sup>) com o objetivo de permitir o trabalho cooperativo de robôs manipuladores convencionais em operações de usinagem e soldagem de dispositivos mecânicos complexos que necessitam de mais graus de liberdade para realizarem trajetórias complexas (figura 4.9).

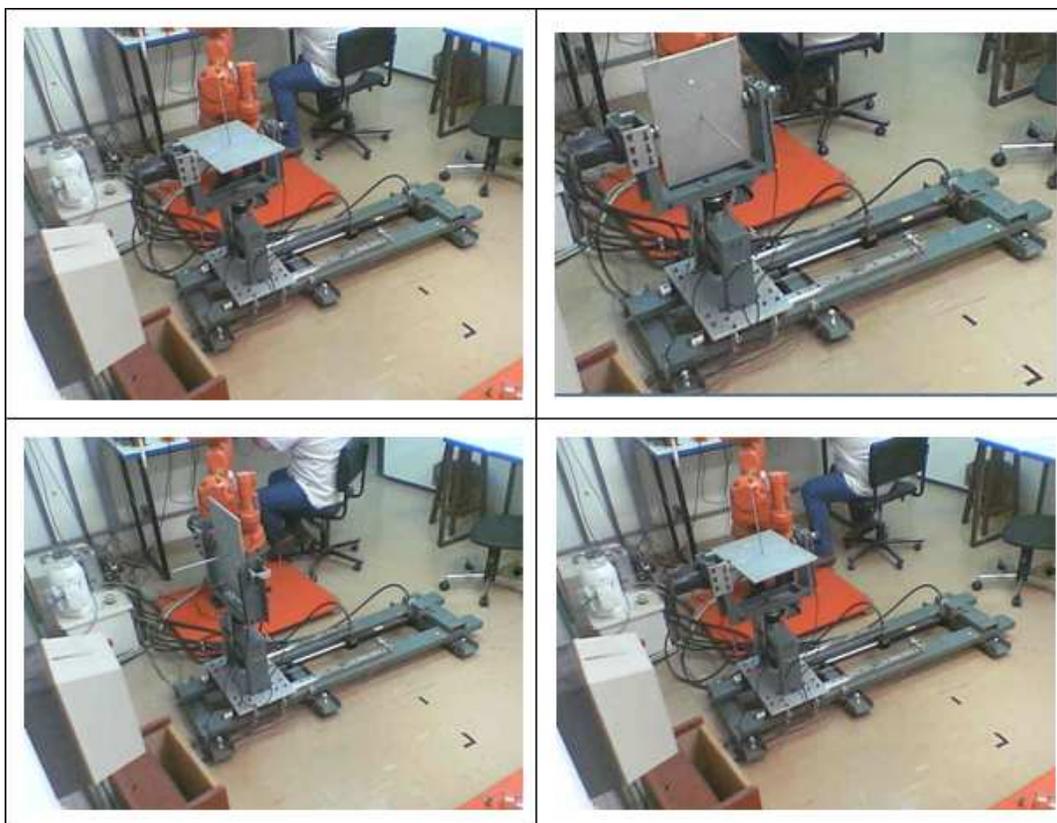


Figura 4.9: Célula Integrada de Manufatura - LAR

As próximas subseções terão como objetivo apresentar uma descrição do *software* implementado em LabVIEW™ para acionamento e controle dos três atuadores hidráulicos de uma Plataforma de Movimentação capaz de posicionamento de uma base e orientação de uma mesa com dois graus de liberdade.

#### 4.3.1 Descrição da Parte Operativa

Esta plataforma é constituída de atuadores hidráulicos com sensores de posicionamento. O primeiro grau de liberdade possui um cilindro de posicionamento linear para movimentação da base em uma determinada direção (movimento translacional) e sensores indutivos de final de curso e posicionamento central da mesa. Dois graus de liberdade são responsáveis pela orientação espacial (robô RR), através da movimentação angular de motores hidráulicos rotativos (movimento de rotação) constituídos de sensores de rotação (*encoders* incrementais). Na base da

mesa esta prevista a colocação de uma haste de calibração, cujas dimensões (comprimento) serão disponibilizado como parâmetro variável da tela de inicialização.

O *software* LabVIEW™ pode ser usado para controle e supervisão. O controle das juntas robóticas é realizado através de uma interface D/A que comanda os motores das juntas. Um programa de supervisão e controle, residente em um computador é responsável pelo gerenciamento e controle das informações provenientes de sensores e atuadores do sistema.

Uma interface de visualização foi implementada em ambiente Windows, neste ambiente foram desenvolvidas telas de supervisão que coletam as informações dos sensores, dos programas de tratamento matemático (modelagem cinemática da mesa), as informações sobre o sistema (velocidade, parâmetros do controlador, números de pontos da trajetória, etc.), da inicialização e da calibração automatizada.

A interface de aquisição de dados utilizada neste trabalho foi a PCI 9112 da *ADLINK Technology*. Esta interface recebe as informações provenientes dos *encoders* e do potenciômetro para determinação da posição atual das juntas constituintes da mesa. Estas informações são comparadas com os valores de referência, logo após esta interface calcula através de um algoritmo de controle enviam informações para a saída do controlador (posição, velocidade) aos acionadores de cada grau de liberdade da mesa (figura 4.10).

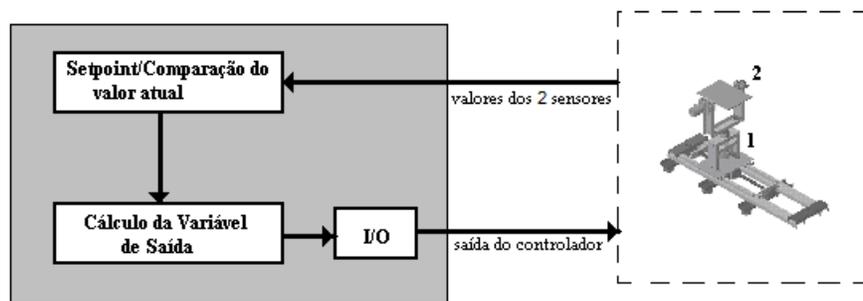


Figura 4.10: Esquema do Sistema Controlado

Todos os valores de referência são calculados em tempo real através de um banco de dados, mesmo a cinemática inversa com as altas não linearidades. Outros tipos de sensores podem ser

incluídos na bancada ou simplesmente através de uma VI (*Virtual Interface*) do *software* LabVIEW.

### 4.3.2 Sistema de Supervisão e Controle

O sistema de supervisão e controle foi implementado em linguagem LabVIEW™, muito utilizado industrialmente, em projetos de dispositivos relacionados as áreas de medição e controle podem utilizá-lo para criar aplicações personalizadas que rodam em plataformas NI (*National Instruments*) com E/S reconfiguráveis baseadas em FPGA's. Juntos, o LabVIEW™ FPGA e o *hardware* NI para E/S reconfiguráveis permitem a criação de uma plataforma flexível para o desenvolvimento de sistemas sofisticados que antes somente eram possíveis com *hardware* projetado de forma dedicada.

### 4.3.3 Descrição do Sistema Operativo

Para implementação do Sistema de Supervisão e Controle da Plataforma Hidráulica foi utilizado a interface de comunicação *PCI-9112 16-CH 12-Bit 110 kS/s Multi-Function DAQ Card/ Low-Profile DAQ Card* da *National Instrumentation™* (figura 4.13), constituída de um módulo com oito entradas analógicas (A/D) para leitura de dois *encoders* incrementais das juntas de movimentação rotativa, um módulo de oito saídas analógicas (D/A) para acionamento dos cilindros rotativos (2) e avanço e recuo do cilindro de movimentação linear, um módulo de oito entradas e oito saídas digitais para leitura dos sensores de posicionamento do cilindro linear e sensores de final de curso, e acionamento de chaves (por exemplo, chaves de início e final de operação) e acionamento.

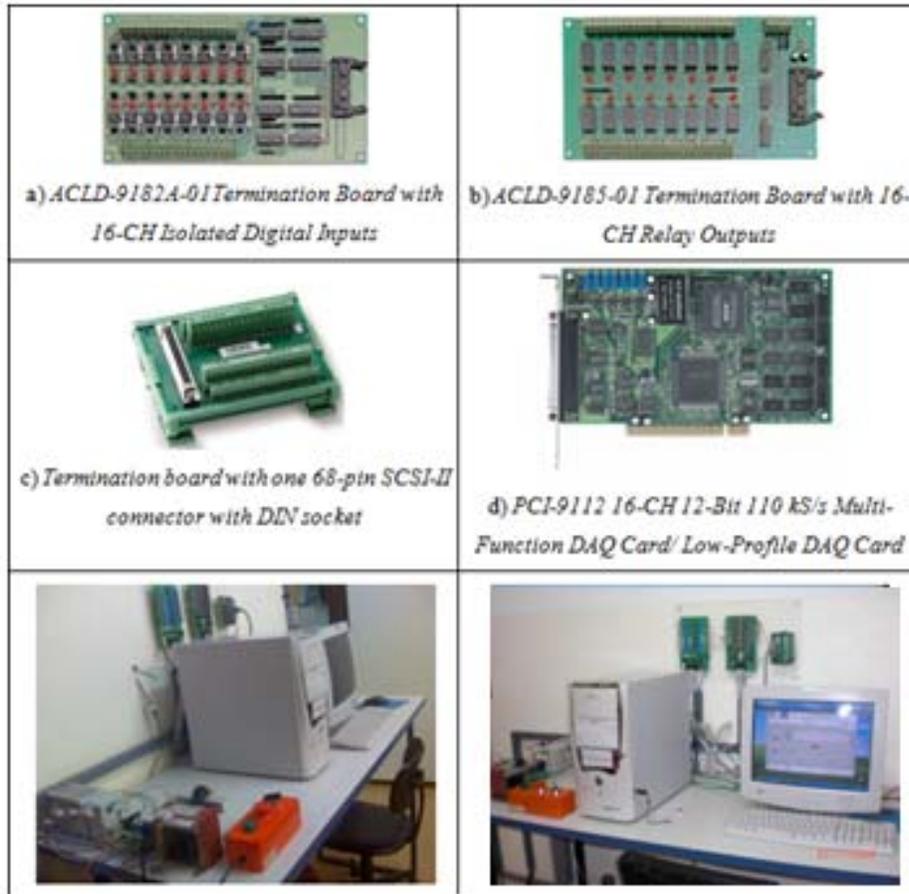


Figura 4.11: Interface de Comunicação *PCI-9112 16-CH 12-Bit 110 kS/s Multi-Function DAQ Card/ Low-Profile DAQ Card* da NI Instrumentation

#### 4.3.4 Interface de Controle

A interface com o usuário foi desenvolvido utilizando o *software* LabVIEW™, compatível com esta interface, possibilitando o monitoramento e controle de informações, dois modos de funcionamento: aprendizagem e leitura do arquivo de dados, permitindo assim, o aprendizado e gravação de trajetórias obtidas a partir da modelagem cinemática da plataforma e trajetórias em relação a ferramenta terminal dos robôs industriais utilizados, armazenamento de informações, leitura dos sensores de posição (*encoders* incrementais) através de interface A/D e acionamento do atuador linear hidráulicos para posicionamento da plataforma (avanço e recuo) e atuadores hidráulicos rotativos através da interface de saída digital, após tratamento do regulador PID de cada junta, interface de entrada digital responsável pela aquisição das informações provenientes dos sensores externos (por exemplo, chaves de início e final de operação).

Assim, todo o gerenciamento dessas informações é realizado através de um programa computacional de supervisão e controle residente em um computador PC, onde foi implementada uma interface de visualização, a partir do desenvolvimento de telas de monitoramento das informações dos sensores (final de curso, segurança, chaves lógicas, etc.), e programas de tratamento matemático (modelagem cinemática do dispositivo), informações sobre o status do sistema (porcentagem de velocidade, parâmetros do regulador PID, número de pontos da trajetória, etc.), inicialização e calibração automatizada.

#### 4.3.5 Telas Implementadas

##### 4.3.5.1. Tela de Inicialização

A tela de inicialização do programa permite que o aplicativo computacional funcione em modo manual (controlado através de botões dedicados (calibração e controle) no painel de comando) ou modo computador, através de *menus* acionados pelo *mouse* (figura 4.12). Quatro procedimentos são disponibilizados ao usuário: Calibração, Aprendizado (Manual), Gerar Arquivo e Controle.

No momento inicial de execução do programa, automaticamente o sistema é conduzido ao modo de calibração da plataforma, que realiza o procedimento de leitura e armazenamento das informações provenientes dos sensores de posição de cada um dos atuadores (*encoders*).



Figura 4.12: Tela Principal do Programa

### 4.3.5.2. Modo Calibração

Este modo permite efetuar a calibração da plataforma, devendo ser utilizado sempre na inicialização do sistema, e em caso de checagem de posicionamento pelo usuário. A tela principal é constituída de dois botões localizados no lado esquerdo da tela: efetuar calibração e posição inicial, um display responsável pelas informações referentes aos sensores de posição de cada uma das juntas (potenciômetro e *encoders*), indicando a posição linear (movimento de translação) e angular (referente aos dois movimentos de rotação da mesa). A figura 4.13 mostra essa tela.



a) Calibração de Posição do Atuador Linear



b) Calibração de Posição do Atuador 2



c) Calibração de Posição do Atuador 3  
 Figura 4.13: Telas do Programa de Calibração

Outros indicadores luminosos mostrarão ao usuário a posição zero do *encoders* (movimento rotativo) e sensores de final de curso no caso de movimento linear da base da plataforma. Os parâmetros de calibração dos sensores e dimensão da haste de calibração disposta em um ponto referenciado ( $x, y$ ) da mesa (neste caso, por *default*,  $x=0, y=0$  e  $L_{haste} = 0,20m$ ) serão introduzidos pelo usuário e não poderão ser mais modificados após a fase de calibração. Dois botões localizados no lado inferiores da tela, permitem gravar estas posições e sair desta tela, retornando ao *menu* principal, apresentado anteriormente.



Figura 4.14: Tela de Calibração da Haste

Cada passo do procedimento de calibração é realizado através de telas interativas com o usuário, de modo a permitir a movimentação dos acionadores, monitoramento dos sensores de posicionamento e rotina de inicialização (posição zero). A figura 4.15 apresenta mensagens típicas do *menu* calibração.

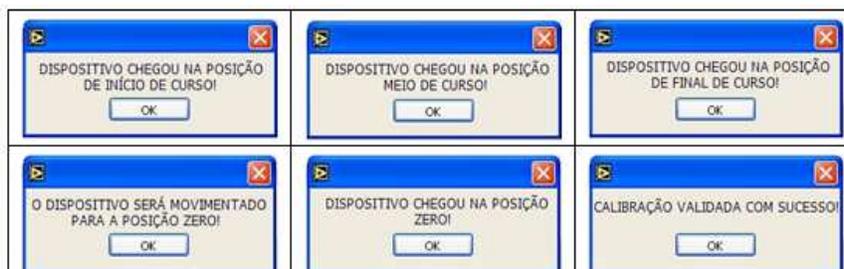


Figura 4.15: Mensagens Típicas do *Menu* de Calibração

Ao concluir todo o processo de calibração para cada grau de liberdade e validar os mesmos, automaticamente gera uma tela conforme a figura 4.16, com:

- os limites de posicionamento de cada junta;
- o valor da posição atual de cada junta;
- botão de acionamento/ desligamento;
- botões para o acionamento do atuador linear de avanço e recuo para o posicionamento;
- botões para o acionamento dos atuadores rotativos no sentido horário e anti-horário para a orientação.



Figura 4.16: Tela gerada após Validação da Calibração

## 4.3.6 Geração de Arquivos

### 4.3.6.1 Modo Aprendizado

Após inicialização da plataforma (posicionamento do atuador linear até movimentação da mesa até sensor de posição central, e calibração dos sensores de posicionamento angular para orientação da mesa, operações realizadas no *menu* anteriormente descrito), o usuário poderá entrar no modo aprendizado. Nesta tela o usuário poderá realizar movimentos no modo junta (rotações dos dois graus de liberdade da mesa) ou no modo cartesiano em relação ao ponto final da haste de calibração (neste modo o programa utiliza o modelo cinemático da mesa, considerando as dimensões e posicionamento da haste em relação ao centro da mesa rotativa). Este modo permite efetuar a geração de arquivos de movimentos de rotação da mesa a partir do movimento de aprendizagem.

No modo angular (figura 4.17a), o usuário realiza as movimentações de cada junta angular armazenando, para posterior interpolação e geração de movimentos no modo automático. A posição do ponto terminal da haste em coordenadas cartesianas é afixada na tela. No modo cartesiano (ou de posição) mostrado na figura 4.17b, o usuário movimenta a mesa realizando movimentos cartesianos em relação ao ponto terminal da ferramenta utilizada (no caso a haste de calibração), sendo mostradas na tela as respectivas posições angulares correspondentes ao movimento.

#### **Modo Inclinação:**

Junta 1: rotação (graus)

Junta 2: ângulo de inclinação (graus)

- Variação de Deslocamento (graus)
- Mostra o valor do Raio para esta inclinação

#### **Modo Raio:**

Junta 1: rotação (graus)

Junta 2: Raio (mm)

- Variação de Deslocamento (graus)
- Mostrar o valor da inclinação (grau)




**DISPOSITIVO SIMULAÇÃO**

quarta-feira, 22 de julho de 2009 16:02:59

**GERAR ARQUIVO**

RAIO 	JUNTA 1 ROT. 1 12	JUNTA 2 INCLINAÇÃO 8	PONTOS 5	CALCULAR 	
INCLINAÇÃO 	Rotação em <b>GRAUS</b> e RAIO em mm				
REGISTRAR 					
	ROT. 1	ROT. 2	X	Y	Z
	1	1	74,5	14,0	38,3
	1	1	74,5	14,0	38,3
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0
	1	1	0,0	0,0	0,0

#### 4.3.6.2 Movimentação a partir de arquivo padrão EXCEL™

A partir de arquivo gerado em EXCEL™, representando a movimentação das juntas, foi implementado um módulo utilizando o *software* LabVIEW™ de modo a permitir:

i) a partir de modelo cinemático apresentado no capítulo 3, realizar a conversão desses movimentos em sinais de referência (expressos em volts) a serem comparados com os sensores de posição (potenciômetro de precisão) a serem enviados para um controlador de posição PID, que fornecerá sinais de comando aos atuadores hidráulicos correspondentes aos dois graus de liberdade de rotação do dispositivo.

ii) Implementação de *Software* de Supervisão e Controle e *Hardware* de Acionamento e Controle em ambiente LabVIEW™ que permitirá a geração automática de movimentos da juntas.

#### 4.3.6.3 Modo Controle

O modo controle permite a leitura de um arquivo de movimentos gerado no *menu* de geração de arquivos apresentado anteriormente, ou a partir da leitura de um arquivo típico de forma de onda gerado em EXCEL™ apresentado anteriormente, que através de um fator de escala permitirá a redução do número de pontos na leitura do arquivo. A figura 4.18 apresenta exemplos de telas típicas deste módulo.



Figura 4.18: Tela típica do Módulo de Controle

#### 4.3.6.4 Tela de Controle (modo automático)

A tela de controle em modo automático (figura 4.19) é constituída de módulos básicos de controle, status do programa, visualização gráfica dos sensores de posição (*encoders* e sensores indutivos de posicionamento da plataforma) e ajuste de parâmetros do controlador de posição (PID).

Ao mesmo tempo a tela é constituída de um botão de habilitação do programa de geração de movimentos para o acionamento das válvulas hidráulicas e interrupção (com sinalização em *display*).



Figura 4.19: Tela de Controle (Modo Automático)

#### 4.3.6.5 Tela de Status

A tela de status mostra os diferentes passos constituintes da trajetória, sendo indicado o número do passo e o valor de referência atual a ser enviado ao controlador para posterior comparação com o *encoder* e a porcentagem de evolução da trajetória (figura 4.20).



Figura 4.20: Telas Típicas de Status de Evolução da Trajetória

#### 4.3.6.6 Tela de Monitoramento

A tela de monitoramento permite o monitoramento no tempo dos valores relativos aos sensores de posição das juntas (*encoders*), conforme mostra a figura 4.21.

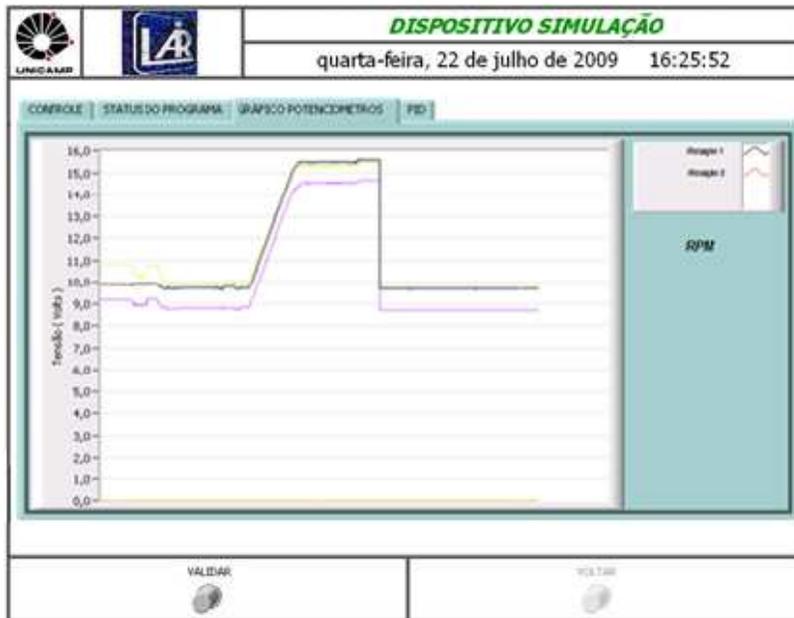


Figura 4.21: Tela de Monitoramento dos Sensores das Juntas (Tensão x Tempo)

#### 4.3.6.7 Tela de Ajuste de Parâmetros do Controlador de Posição PID

Esta tela permite o ajuste de parâmetros do controlador PID (Proporcional, Integral e Derivativo) e escolha da porcentagem de erro máximo (figura 4.22).



Figura 4.22: Tela de Ajuste de Parâmetros do Controlador PID

#### 4.3.6.8 Modo de Segurança

Na implementação do programa final em LabVIEW™, foram contemplados aspectos relacionados à segurança de utilização do dispositivo. Os aspectos considerados foram: estabelecimento de limitação via *software* dos limites de final de curso de cada junta rotativa (informação fornecida durante a fase de calibração), e final de curso da base (informação fornecida pelos sensores indutivos dispostos na base). Essa proteção consiste basicamente na interrupção do programa (desliga o sistema de acionamento) e mensagem de alarme na tela ao usuário, conforme mostra a figura 4.23.



a) Limites de Segurança do Atuador Rotativo 1



b) Limites de Segurança do Atuador Rotativo 2

Figura 4.23: Tela de Alarme de Funcionamento (Modo Segurança)

### 4.3.6.9 Geração de Arquivo de Banco de Dados

Após carregamento dos arquivos, um banco de dados, constituído de informações de funcionamento do arquivo é automaticamente gerado (figura 4.24).



Figura 4.24: Banco de Dados (Arquivos de Movimentação)

## 4.4 Conclusão

Como uma solução para integrar e implementar uma programação executa-se as instruções programadas por uma simples lógica de comunicação. 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.

Para validação deste trabalho de pesquisa foram implementados dois estudos:

- a) Plataforma *WEB* colaborativa em Automação: integração de dois robôs industriais junto com dispositivo robótico de 3 GDL para realização de tarefas colaborativas automatizadas através de *WEB*;
- b) Implementação de Sistema de Supervisão e Controle de um dispositivo robótico com 3 GDL, com ênfase na modelagem cinemática e controle de posicionamento.

## CAPÍTULO 5

### CONCLUSOES

A integração de dispositivos mecatrônicos em células de manufatura é cada vez mais utilizada na área de automação, necessitando rapidez e redução de custos de implementação.

Esta tese apresentou um panorama geral do trabalho desenvolvido, sendo mostrado um estado da arte do conceito de Sistemas Colaborativos em Automação, com ênfase na área de integração de sistemas, justificando o desenvolvimento desse projeto de pesquisa, seus objetivos, a necessidade de capacitação nesta importante área e a forma como foi delimitada esta questão, além da estrutura geral dos capítulos da tese.

Neste trabalho de doutorado foram apresentadas as principais ferramentas necessárias para a integração e utilização desses conceitos. Os resultados são apresentados a partir da validação experimental de um dispositivo robótico com 3 GDL e dois robôs industriais constituindo uma célula colaborativa em automação, e poderão ser generalizados para outras aplicações.

No Capítulo 2, foi apresentada a revisão bibliográfica enfatizando os temas de sistemas colaborativos, programação *off-line*, integração de células de manufatura, apresentação de ferramentas prototipagem rápida, modelagem e controle de sistemas automatizados e ferramentas para supervisão e controle de sistemas colaborativos.

No Capítulo 3, foi apresentada a proposta de um dispositivo robótico com 3 GDL para integração de robôs industriais, enfatizando a modelagem dinâmica, simulação, e controle de posicionamento através da implementação de um simulador virtual.

O Capítulo 4 descreveu a validação experimental deste trabalho, apresentando dois estudos de caso implementados obtidos no Laboratório de Automação Integrada e Robótica da UNICAMP:

- a) Integração de Sistemas Automatizados Colaborativos através da *WEB*;
- b) Implementação de um sistema de supervisão e controle utilizando a ferramenta LabVIEW™, e modelagem proposta no capítulo 3.

Podemos concluir, com este trabalho, que a validação experimental atingiu os objetivos inicialmente propostos com a integração de uma célula de manufatura robotizada, desenvolvendo o modelo do dispositivo robótico, que permitiu o posicionamento da plataforma hidráulica, onde uma peça foi trabalhada por dois robôs industriais, constituindo assim um sistema colaborativo de manufatura. Além disso, apresentamos a modelagem cinemática e dinâmica, o cálculo dos parâmetros do sistema de controle utilizando MatLab-Simulink™ e assim desenvolvido a interface em LabVIEW™ para supervisão e controle da célula robotizada. Com tudo isso, foi possível demonstrar a aplicação da célula colaborativa como um laboratório virtual através da *WEB*.

Os resultados apresentados parcialmente em cada capítulo e validados experimentalmente no capítulo 4 mostram que o trabalho é complexo e de importância na área de mecatrônica, e poderá ser estendido através da continuidade e aprimoramento das atividades de pesquisa desenvolvidas durante o projeto, onde algumas implementações se tornam importantes para trabalhos futuros:

- 1) Utilização de servo-válvulas hidráulicas para melhorar o controle das juntas hidráulicas e conseqüentemente precisão de posicionamento;
- 2) Extensão da modelagem de um motor de corrente contínua para um motor hidráulico;
- 3) Incluir barramentos para o sincronismo da mesa hidráulica com os robôs industriais;

- 4) Aprimoramento do controle do sistema através de implementação de controladores de junta baseados em *FPGA*;
- 5) Preparação de cursos na área de automação e controle utilizando todos os recursos desenvolvidos neste trabalho, permitindo o conhecimento multidisciplinar reunindo informações de diferentes áreas.
- 6) Implementação de outros recursos e inclusão de novos conhecimentos, aproveitando uma das características deste trabalho que é o desenvolvimento de plataforma aberta com capacidade de incorporação de técnicas e tecnologias.

## **REFERÊNCIAS BIBLIOGRÁFICAS**

Adam, N. R. e Dogramaci A.: “Current Issues in Computer Simulation”, New York: Academic, 292 p., 1979.

Aihara, C. K.: “Uma Abordagem Interativa para o Problema de Capacitação e Pesquisa em Automação”, Campinas. Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2005, 104p. Tese (Doutorado).

Aihara, C. K., Cosso, S. G., Saramago, M. A. P. e Rosário, J. M.: “Desenvolvimento de Aplicativos para Monitoramento de Variáveis de Controle de Processos Industriais”, In: Aplicon, EEUSP São Carlos, Julho 2001.

Álvares, J. A. e Ferreira, J. C. E.: “Metodologia para Implantação de Laboratórios Remotos Via Internet na Área de Automação da Manufatura”, 2º COBEF, Uberlândia-SP, 2003.

Boaretto, N., Sanzovo, N. e Scandelari, L.: “Implementação de um Laboratório de Automação de Manufatura”, CEFET-PR, Cobenge 2004, Paraná, 2004.

Borangi, T., Gilbert, P., Ivanescu, N. and Rosu, A.: “An Implementing Framework for Holonic Manufacturing Control with Multiple Robot Stations”, Engineering Applications of Artificial Intelligence, Vol. 22, Issues 4-5, p. 505-521, 2009.

- Chi-Keng, T.: "Multiple robot coordination and Programming". Proceeding of the 1991 IEEE International Conference on Robotics and Automation. Sacramento, Califórnia. pp. 978-985, 1991.
- Campos, B. A. N.: "Simulação para Modelagem e Calibração de Robôs Industriais", Faculdade de Tecnologia, Universidade de Brasília, 2006, 153p., Tese (Mestrado)
- Cardoso, J. e Valette, R.: "Redes de Petri", Florianópolis: Editora da UFSC, 212 p., 1997.
- Cassandras, C. G.: "Discrete Event Systems: Modeling and Performance Analysis". Burr Ridge: Irwin, 1993, 790p.
- Cosso, S. G., Integração de Ferramentas de Automação direcionadas à Aplicações de Telerobótica - Implementação de um Sistema de Supervisão e Controle num Sistema Teleoperado, Campinas, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2002. 134p., Tese (Mestrado).
- David, S. and Rosário J. M.: "Modeling.: "Simulation and Control of Flexible Robots", CONTROLO'98, pp. 532-539, Coimbra, Portugal, 1998.
- Ferreira, E. P.: "Robótica Básica Modelagem de Robôs", R. Vieira Gráfica e Editora Ltda. Versão Preliminar Publicada para a V Escola Brasileiro-Argentina de Informática, Rio de Janeiro, 1991.
- Freitas, J. C. A.: "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. Dissertação (Mestrado).
- Godinho, T e Pires, J. N.: "Autodesck na Programação de Robôs Industriais", In e-newsletter Autodesck, Campus Ideas, nº. 3, Laboratório de Robótica Industrial, Universidade de Coimbra, 2006.

Groover, M., et al. Robótica: Tecnologia e Programação. McGraw-Hill, São Paulo, 1988.

Henriques, R., V., B.:” Desenvolvimento de Metodologia para Planejamento de Trajetórias de Soldagem Robotizada”, Escola de Engenharia da UFMG, Belo Horizonte, Tese de Doutorado, 2005.

Huber, P., Jensen K. and Shapiro, R. M.; "Hierarchies in Coloured Petri Nets"; In: G. Rozenberg (ed.); Advances in Petri Nets 1990; Lecture Notes in Computer Science, vol. 483, Springer, 1990, pp. 313-341 e "High-level Petri Nets - Theory and Application"; K. Jensen, G. Rozenberg (Eds.), Springer-Verlag, pp. 215-243,1991.

Li-Chung, T. and Ming, J. K.:” Dynamic load - Carrying Capacity and Inverse Dynamics of Multi Cooperating Robots Manipulators”. IEE Trans. On Robotics and Automation, Vol. 10, nº. 1, pp. 71-77. 1994.

López, G., Romeo, A. and Guerrero, J.:”Simulation Tools for Active Learning in Robot Control and Programming”, 20th EAEEIE Annual Conference: Innovation in Education for Electrical and Information Engineering, Valencia. Spain, 2009.

Marange, P., Gellot, F. and Riera, B.: “Requirement and use for Remote Teaching of Discrete Events Systems”, 7th IFAC Symposium on Advances in Control Education (ACE'06), Madrid : Espagne 2006.

Matos, M. V. e Santos, W. R.: “Redes de Petri”, Disponível em: [http://www.unigran.br/biblioteca/producaointelectual/rede\\_de\\_petri.pdf](http://www.unigran.br/biblioteca/producaointelectual/rede_de_petri.pdf) . Acesso em 2008.

Morcelli, J. C. M.: “Simulador Seqüencial de sistemas de Filas”, Disponível em: <http://www.inf.pucpcaldas.br/~morselli/> , Acessado em Agosto de 2007.

Oliveira, E.: “Prototipagem Rápida em Sistemas Mecatrônicos Baseada em Instrumentação Virtual””, Campinas. Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2008, 271p., Tese (Mestrado).

Peterson, J. L.: “Petri Net Theory and the Modeling of Systems”. Prentice-Hall, Inc., 1981.

Pinotti, A. J. e Amaral, S.: “Simulação de Célula Automatizada para Aplicação Industrial - Paletização”, CRICTE2006 - XXI Congresso de Iniciação Científica e Tecnologia, Joinville-SC, 2006.

Pires, J. N.: “Desafios da Robótica Industrial”, Departamento de Engenharia Mecânica, Universidade de Coimbra, 2000.

Pires, J. M. e Sá, J. M. G.: “Objeted - Oriented and Distributed Approach for Programming Robotics Manufacturing Cells”. Robotics and Computer Integrated Manufacturing. Vol. 16, pp. 29-42, 2003.

Queiroz, L.R.: “Um Laboratório Virtual de robótica e visão computacional”, Dissertação de Mestrado, UNICAMP, Instituto de Computação, Campinas, 1998.

Ramadge, P. J. and Wonham, W. M.: “The Control of Discrete Event Systems”. Proceedings of the IEEE, V. 77, nº. 1, 1989.

Rehg, J.: “Introduction to Robotics in CIMs”. Prentice Hall, 4/e, 440p. 2000.

Rojas, J. H. C.: “Metodologia de Modelagem, Simulação e Programação *off-line* de Robôs e Mecanismos Mecatrônicos Integrados e Direcionados a Células de Manufaturas Flexíveis”, Campinas. Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2004, 168p. Tese (Doutorado).

Rosário, J. M., Oliveira C. and Sá C.E.A.: “Proposal Methodology for the Modeling and Control of Manipulators”, International Journal of the Brazilian Society of Mechanical Engineering, Vol. XXIV - nº. 3, Julho 2002.

Rosário, J. M.: “Princípios de Mecatrônica”, Pearson Prentice Hall do Brasil, 360p., 2004, ISBN: 85-7605-010-2.

Sánchez, O. F. A: “Desenvolvimento de Sistema de Preensão para utilização em Dispositivos Robóticos Campinas. Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2008, 178p. Tese (Doutorado).

Silva, E. N. et al.: “Novas tecnologias para educação no Amazonas”, In: Congresso da Rede Iberoamericana de informática educativa Laboratório de Tecnologias Cognitivas, 1998, Brasília. Disponível em: <http://www.c5.cl/ieinvestiga/actas/ribie98/205.html>. Acesso em 2005.

Soares, L. F. G.: “Modelagem e Simulação Discreta”, São Paulo, IME-USP, 1992, 254p.

Suh, I. H. and Shon, K. G.: "Coordination of Dual Robot Arms Using Kinematic Redundancy," IEEE Journal of Robotics and Automation, Vol. 5, Nº. 2, pp.236-241, 1989.

Traylot, R. L., Heer, D. and Fiez T.S.: “Using an Integrated Platform for Learning<sup>TM</sup> to Reinvent Engineering Education”, In: IEEE Transactions on Education, Vol. 46, nº. 4, p. 409-419, November 2003.

Uchiyama, M. and Yamashita, T.: “Adaptive Load Sharing for Hybrid Controlled Two Cooperative Manipulators”. Proceedings of the 1991 IEEE International Conference on Robotics and Automation, Sacramento, (1991), 986-991.

Vigia, K. et al.: “Control of Contact Conditions for Manipulations with Multiple Robotic Systems”. Proceeding of the 1991 IEEE International Conference on Robotics and Automation. pp. 170-175. Sacramento, Califórnia.

Yong, Y.F. and 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.

### **Bibliografia Consultada**

ABB Robotics: “ABB Flexible Automation”, Product On-Line Manual IRB1400, 3HAC 2914-1, 1998.

Araujo, E. S.: “Modelagem e Descrição da Parte Comando de um Sistema Automatizado de Produção utilizando o GRAFCET - Aplicado a uma Plataforma Industrial em Automação”. Campinas: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 1997, 91 p., Tese (Mestrado).

Brams G. W.: “Réseaux de Petri: “Théorie et pratique”, tomos 1 et 2, Masson Editions, 1983.

Boucher, P., Dumur D.: “La Commande Prédicative, Éditions Technip”, Paris, 1996.

Craig, J. J.: “Introduction to Robotics: Mechanics and Control”, Second edition, Addison-Wesley Publishing Company, 1989.

D’Abreu, J. V. V.: “Integração de Dispositivos Mecatrônicos para Ensino-Aprendizagem de Conceitos na Área de Automação”, Campinas, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2002. Tese (Doutorado).

Denavit, J. and Hartenberg R. S.: “A Kinematics Notation for Lower-Pair mechanism Based on Matrices”, ASME J. Appl. Mech, pp. 215-221, June 1955.

Jensen, K.: “An Introduction to the Practical Use of Coloured Petri Nets”, In: Lecture Notes on Computer Science, n°. 1492, p. 237-292, 1998. Disponível em: [http://www.daimi.au.dk/~kjensen/papers\\_books/rec\\_papers\\_books.html](http://www.daimi.au.dk/~kjensen/papers_books/rec_papers_books.html) Acessado em fevereiro de 2006.

Kimie, C. A.: “Uma Abordagem Interativa para o Problema de Capacitação e Pesquisa em Automação”, Campinas. Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2005, 141p. Tese (Doutorado).

Melo, J. J. L. e Sobreira P.L.: “Petri Net”. Disponível em: <http://www.jameson.hpg.ig.com.br/introd.html>, Acessado em agosto de 2007.

Murata, T.: “Petri Nets: Properties, Analysis and Applications”, Proceedings of the IEEE, Vol. 77, n°. 4, pp. 541-580, 1989.

National Instruments: LabVIEW - User Manual, July 2000.

Ogata, K.: “Modern Control Engineering”, Prentice Hall, Third Edition, 1997.

Spong, M. W. and Vidyasagar M.: “Robot Dynamics and Control”, John Wiley & Sons, New York, 1989.

Stokiv, D. M., Vukobratovic, M. K. And Lekovic, D. B.: “Simulaion of Robots in Flexible Manufacturing Cell”. Robotics and Computer Integrated Manufacturing, Vol. 8, Nº. 1, pp. 1-8, 1997.

Thomazini, D. e Albuquerque, P. U. B.: “Sensores Industriais – Fundamentos e Aplicações”, São Paulo: Érica, 2005.

## **ANEXO A - GRAFCET FUNCIONAL**

### **A.1 Primeira Tela feita em LabVIEW para a Integração da Célula Cooperativa**

#### **A.1.1 Funcionamento do sistema supervisorio**

O software supervisorio mostrado acima foi desenvolvido em linguagem LabVIEW™.

O mesmo tem a função de coordenar as atividades de um sistema de Robôs que trabalham cooperativamente.

#### **A.1.2 Descrição do funcionamento das chaves e botões**

Os botões/ chaves *LIGA* e *DESLIGA*, tem a função de energizar e desenergizar o sistema.

Os botões em amarelo na parte superior do sistema supervisorio, são utilizados para posicionar os robôs em movimentos conjuntos ou coordenados, para início dos trabalhos.

Os botões em vermelho, contidos nos lados, esquerdo e direito do supervisorio, tem a função de posicionamento e alinhamento dos Robôs individualmente, para que o sistema possa ser inicializado corretamente.

Os botões em Laranja na parte inferior do sistema supervisorio, são utilizados para posicionar a mesa, para início dos trabalhos.

Após o alinhamento inicial da Mesa e dos Robôs, o sistema está em condições de executar os trabalhos coordenados e em conjunto por parte da mesa e por parte de ambos os robôs.

Para iniciar os trabalhos coordenados basta Pressionar o botão *INICIA* em amarelo, contido na parte inferior esquerda da tela.

Para encerrar os trabalhos temos o botão *STOP*, que para as atividades do sistema coordenado. Caso o Operador necessite reiniciar os trabalhos basta apertar o botão *START* que o sistema reinicia as atividades de onde parou.

### **A.1.3 Protocolo de comunicação:**

O meio físico (hardware) utilizado para a comunicação entre o sistema supervisor e os *CLP's* do Robôs será uma comunicação serial RS485.

#### **A.1.3.1 Sistema Mestre e Escravos**

Onde o Mestre é o Computador *PC*, plataforma Windows *XP*, rodando Software LabVIEW™ como sistema Supervisor, e os escravos são os *CLP's* dos ROBÔS.

Este protocolo foi escolhido, em virtude da existência de mais de um escravo (mais de um escravo não é suportado pela comunicação RS232).

Para a Transferência das Palavras de Controle mencionadas abaixo, será utilizado um software servidor rodando no *PC*, sendo este, um *OPC SERVER*.

Podendo ser este *OPC SERVER* das marcas:

- *Matricon*;
- *RS LINX*;
- *Aromat*, etc.

Como o Servidor *OPC SERVER* transfere e recebe as palavras de comando, entre o mestre e os escravos automaticamente, não se torna necessário a utilização ou Implementação de recursos de comunicação para testes de consistência dos dados, como por exemplo:

- *Header's*;
- Algoritmos para testes de *Check-sum*;
- Polinômios de consistência;
- etc.

Todo este trabalho é feito pelo *OPC SERVER*, fazendo com que esses problemas de comunicação se tornem transparentes para o projetista.

### **A.1.3.2 Conjunto das palavras de comando:**

Cada palavra de Comando é composta de 1 *byte*, conforme segue:

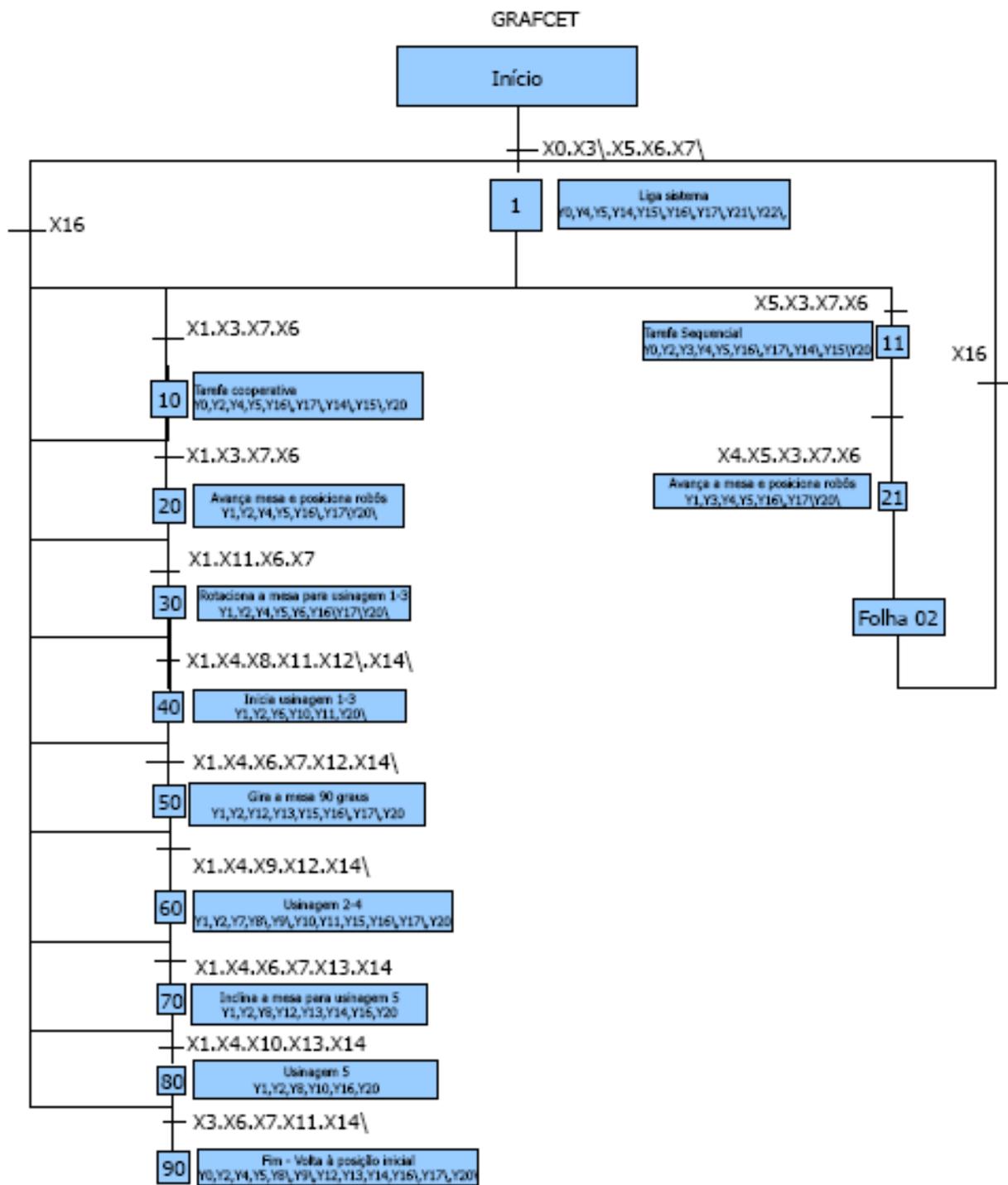
Número do *Byte* Comando:

- BYTE 0.....Início
- BYTE 1.....Mesa
- BYTE 2.....Movimento de Translação da Mesa
- BYTE 3.....Rotação R1 Mesa (Face 1 e 3)
- BYTE 4.....Rotação R1 Mesa (Face 2 e 4)
- BYTE 5.....Rotação R2 Mesa (inclinação) - Robô 1
- BYTE 6.....Rotação R2 Mesa (inclinação) - Robô 2
- BYTE 7.....Init\_Rob
- BYTE 8.....Posição Inicial - Robô (R1) - Traj11
- BYTE 9.....Posição Inicial - Robô (R2) - Traj21
- BYTE 10.....Posicionamento Cooperativo Robôs (R1 e R2)
- BYTE 11.....Task1\_Rob
- BYTE 12.....Traj12: Trajetória Robô 1 - Face 1 (2)
- BYTE 13.....Traj22: Trajetória Robô 2 - Face 3 (4)

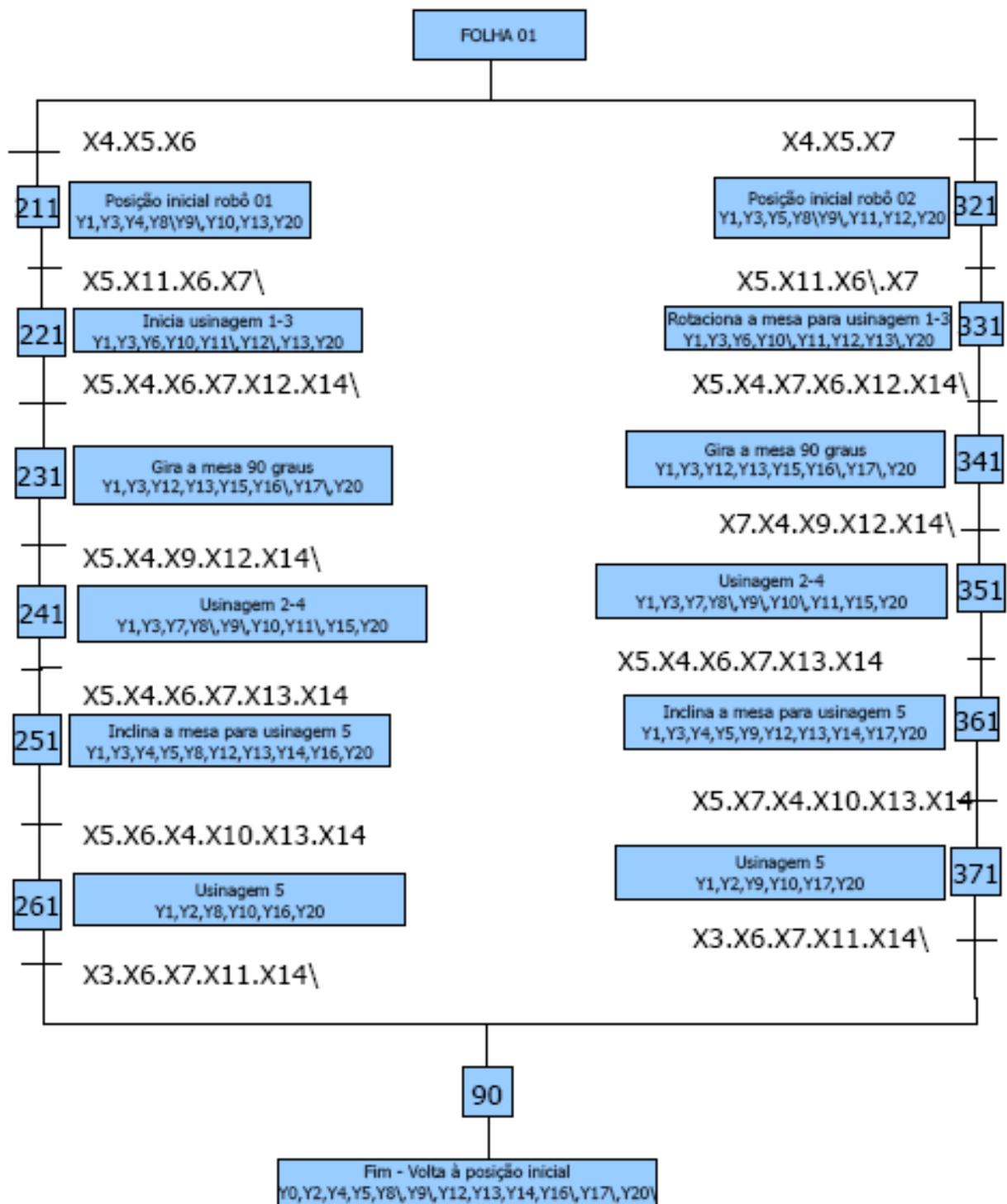
BYTE 14.....Tarefa Cooperativa Robôs (R1 e R2)  
BYTE 15.....Task2\_Rob  
BYTE 16.....Traj13: Trajetória Robô 1 - Mesa Inclinada  
BYTE 17.....Traj23: Trajetória Robô 2 - Mesa Inclinada  
BYTE 18.....Tarefa Seqüencial Robôs (R1 e R2)

Foram utilizados *Bytes* para comunicação para comandos que podem ser de apenas *Bits*, porque a maioria dos Servidores *OPC SERVER*, utiliza *Bytes* que são utilizados como *Bits* no Sistema supervisorio.

Essas palavras de comando são trocadas entre os *CLP's*, os robôs e o sistema supervisorio, elas são trocadas de forma contínua, gerenciadas pelo sistema *OPC SERVER*. Na subseção A.1.4 mostra a relação das entradas e saídas do sistema e a lógica de comando supervisor.



a) Início



b) Fim

Figura A.1: Estrutura GRAFCET

#### A.1.4 GRAFCET - Relação das entradas e saídas do sistema

Nº.	VARIÁVEIS	START/END SUB-ROTINA - MESA
1	S0	Início do Programa
2	SF	Final do Programa
3	S10	Task2 (inclinação da mesa)
4	S11	Movimento de Translação da Mesa
5	S12	Rotação R1 Mesa (Face 1 - 3)
6	S13	Rotação R1 Mesa (Face 2 - 4)
7	S14	Rotação R2 Mesa (inclinação) – Robô 1
8	S15	Rotação R2 Mesa (inclinação) – Robô 2
9	S16	Traj13: Trajetória Robô 1 – Mesa Inclinada
10	S17	Traj23: Trajetória Robô 2 – Mesa Inclinada

Nº.	VARIÁVEIS	MODO MOVIMENTO DOS ROBÔS - DESCRIPTION
1	S21	Pos_Inic_ – Robô (R1) - Traj11
2	S22	Pos_Inic_ – Robô (R2) - Traj21
3	S23	Posicionamento Cooperativo Robôs (R1 e R2)
9	S31	Tarefa Seqüencial Robô (R1)
9	S31	Tarefa Seqüencial Robô (R2)
7	S12	Traj13: Trajetória Robô 1 – Mesa Inclinada
8	S13	Traj23: Trajetória Robô 2 – Mesa Inclinada
	S24	Task_1_Usinagem de superfície
	S25	Traj12: Trajetória Robô 1 – Face 1 (2)
	S26	Traj22: Trajetória Robô 2 – Face 3 (4)
	S27	Tarefa Cooperativa Robôs DE USINAGEM (R1 e R2)

##### A.1.4.1 Lógica de comando do supervisor associados aos subprogramas da CLP

Nº.	AUX.	LÓGICAS VARIÁVEIS
1	C1	Lógica de modo Mesa e Robô
2	C2	Lógica de modo Mesa e Robô
3	C3	Supervisor
4	C4	Supervisor
5	C5	Supervisor

6	C6		Supervisor
7	C7		Switch ON/OFF
8	OUT		Funções das saídas
9	S5		Task2 (inclinação da mesa)
10	S51		Traj13: Trajetória Robô 1 – Mesa Inclinação
11	S52		Traj23: Trajetória Robô 2 – Mesa Inclinação
12	S53		Tarefa Seqüencial Robôs (R1 e R2)
	<b>Elemento</b>		
01	X0	L_D	Liga Desliga da operação
02	X1	Coop	

#### A.1.4.2 Estruturação Lógica

##### a) Início do Programa – Seleção Modo Movimenta Mesa ou Robô

C1	C2	VALOR	S	LÓGICA DE CONTROLE DE MOV. MESA OU ROBÔ
0	0	0	S0	Menu de Início de Programa
0	1	1	S1	Modo Movimenta Mesa
1	0	2	S2	Modo Movimenta Robô
1	1	3	X	VAZIO

##### b) Modo Mesa: Movimentação da Mesa

C1	C2	C3	C4	C5	VLR.	S	LÓGICA DE CONTROLE
0	1	0	0	0	0	S1X	MOV. DE MESA
0	1	1	1	1	7	S11	TRANSLAÇÃO DA MESA
0	1	0	0	1	4	S12	ROTAÇÃO MESA (1-3)
0	1	0	1	0	2	S13	ROTAÇÃO DA MESA (1-4)
0	1	1	0	1	5	S14	INCLINAÇÃO DA MESA PARA ROBÔ 01
0	1	1	1	0	3	S15	INCLINAÇÃO DA MESA PARA ROBÔ 02
0	0	X	X	X	X	S0	MENU DE INÍCIO DE PROGRAMA

**c) Modo Robô: Movimentação do Robô**

C1	C02	C3	C4	C5	C6	VLR.	S	LÓGICA DE CONTROLE
1	0	0	0	X	X	0	S2	MODO – Movimenta Robô
1	0	0	1	X	X	2	S3	INIALIZAÇÃO DO ROBÔ
1	0	0	1	0	0	2	S31	Posição Inicial – Robô (R1) - Traj1
1	0	0	1	1	1	1	S32	Posição Inicial – Robô (R2) - Traj2
1	0	0	1	1	1	3	S33	Posicionamento Cooperativo Robôs (R1 e R2)
1	0	1	0	X	X	1	S4	Task1 (Faces)
1	0	1	0	0	0	2	S41	Posição Inicial – Robô (R1) - Traj1
1	0	1	0	1	1	1	S42	Posição Inicial – Robô (R2) - Traj2
1	0	1	0	1	1	3	S43	Posicionamento Cooperativo Robôs (R1 e R2)
1	0	1	1	X	X	3	S5	Task2 (Mesa Inclinada)
1	0	1	1	0	0	2	S51	Traj13: Trajetória Robô 1 – Mesa Inclinada
1	0	1	1	1	1	1	S52	Traj23: Trajetória Robô 2 – Mesa Inclinada
1	0	1	1	1	1	3	S53	Tarefa Sequencial Robôs (R1 e R2)
0	0	X	X	X	X	X	S0	Menu de Inicio de Programa

**d) Entradas do CLP e Entradas (sensores)**

EL.	Nick_name	DESCRIÇÃO
X0	LD	Liga Desliga
X1	COOP	Modo cooperação
X2	ST	Start
X3	CIL_AV	Mesa recuada
X4	CIL_REC	Mesa avançada
X5	SEQ	Modo sequencial
X6	PINI_R1	Posição inicial robô 1
X7	PINI_R2	Posição inicial robô 2
X8	F1_3	Usinagem 1-3
X9	F2_4	Usinagem 2-4
X10	F5	Usinagem 5
X11	ROTM1_3	Rotação mesa 1-3
X12	ROTM1_4	Rotação mesa 2-4

X13	ROTM5	Rotação mesa 5
x14	INCL_MESA	Inclinação da mesa
x15	STOP	Soft Stop
X16	EMER	Emergência

#### e) Saídas do CLP

EL.	Nick_name	DESCRIÇÃO
y0	L_M_R	Led de mesa recuada
y1	L_M_A	Lede de mesa avançada
y2	L_COOP	Led tarefa coop.
y3	L_SEQ	Led tarefe seq.
Y4	L_POS_I_R1	Led pos. Inic. Robô 01
Y5	L_POS_I_R2	Led pos. Inic. Robô 02
Y6	L_1-3	Led trajetória face 1-3
Y7	L1-4	Led trajetória face 2-4
Y8	L_I_R1	Led trajetória mesa inclinada Robô 01
Y9	L_I_R2	Led trajetória mesa inclinada Robô 02
Y10	L_RB_OP_01	Led robô 01 em operação
Y11	L_RB_OP_02	Led robô 02 em operação
Y12	L_RB_DE_01	Led robô 01 em descanso
Y13	L_RB_DE_02	Led Robô 02 em descanso
Y14	L_R_M_1-3	Rot. Mesa F 1-3
Y15	L_R_M_2-4	Rot. Mesa F 2-4
Y16	L_I_M_R1	Inclinação mesa R1
Y17	L-I_M_R2	Inclinação mesa R2
Y18		NOP
Y19		NOP
Y20	CIL_AV_REC	Acionamento do cilindro de recuo avanço da mesa
Y21	MR	Motor rotação
Y22	MI	Motor Inclinação

### f) Entradas do robô

X4	X5	X6	X7	VLR.	COMANDO
1	0	0	0	16	Modo Robô
1	1	1	0	112	Posição Inicial do Robô R1
1	1	0	1	196	Trajectoria Robô 1 – Face 1 (2)
1	1	1	1	240	Trajectoria Robô 1 – Mesa Inclinada
0	0	1	0	64	Posição Inicial do Robô R2
0	0	0	1	128	Trajectoria Robô 2 – Face 3 (4)
0	0	1	1	192	Trajectoria Robô 2 – Mesa Inclinada

### g) Saídas do robô

Y3	Y4	Y5	-VLR	COMANDO
1	0	0	8	Modo Robô
1	1	1	56	Posição inicial do Robô 01
1	1	0	24	Traj. Robô 01 para usinar face 1-3
1	0	1	40	Mesa inclinada para robô 01
0	1	1	48	Posição inicial do Robô 02
0	1	0	16	Traj. Robô 02 para usinar face 2-4
0	0	1	32	Mesa inclinada para robô 02

### h) Lógica de Entradas do CLP

X0	X1	X2	X3	VLR	COMANDO
1	0	0	0	1	Operação Mesa
1	1	0	0	3	Pos. Inic. Mesa (na translação)
1	0	1	0	5	Rot. Mesa (1-3)
1	0	0	1	9	Rot. Mesa (2-4)
1	1	1	0	7	Inclinação da mesa para robô 01
1	1	0	1	11	Inclinação da mesa para robô 02

**i) Lógica de entrada dos Robôs**

<b>X4</b>	<b>X5</b>	<b>X6</b>	<b>X7</b>	<b>VLR</b>	<b>COMANDO</b>
1	0	0	0	16	Modo Robô
1	1	1	0	112	Posição inicial do Robô 01
1	1	0	1	196	Traj. Robô 01 para usinar face 1-3
1	1	1	1	240	Mesa inclunada para robô 01
0	0	1	0	64	Posição inicial do Robô 02
0	0	0	1	128	Traj. Robô 02 para usinar face 2-4
0	0	1	1	192	Mesa inclunada para robô 02

## ANEXO B - CONCEITOS DOS SENSORES

Quanto ao princípio de funcionamento, o *encoder* incremental é um transdutor que converte um movimento angular ou linear em uma série de pulsos digitais elétricos. Esses pulsos gerados podem ser usados para determinar velocidade, taxa de aceleração, distância, rotação, posição ou direção como mostra a figura B.1 e B.2.

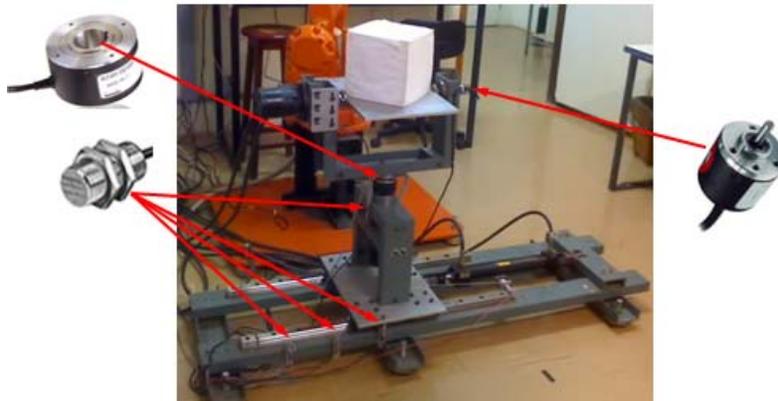


Figura B.1: Encoders Incrementais

O *encoder* incremental fornece normalmente dois pulsos quadrados defasados em  $90^\circ$ , que são chamados usualmente de canal A e canal B. A leitura de apenas um canal fornece somente a velocidade, enquanto que a leitura dos dois canais fornece também o sentido do movimento.



Figura B.2: Encoder de Movimento Angular

Outro sinal chamado de *Z* ou zero também está disponível e ele dá a posição absoluta “zero” do *encoder*. Este sinal é um pulso quadrado em que a fase e a largura são as mesmas do canal *A*.

Os sensores indutivos mostrado na figura B.2 são equipamentos eletrônicos capazes de detectar a aproximação de peças, componentes, elementos de máquinas, etc., em substituição as tradicionais chaves fim de curso. A detecção ocorre sem que haja o contato físico entre o acionador e o sensor, aumentando a vida útil do sensor por não possuir peças móveis sujeitas a desgastes mecânicos.

Os sensores Indutivos são sensores de proximidade, ou seja, geram um sinal de saída quando um objeto metálico (aço, alumínio, cobre, latão, etc.) entra na sua área de detecção, vindo de qualquer direção, sem que seja necessário o contato físico. A plataforma hidráulica contém quatro sensores indutivos onde três sensores são utilizados para o posicionamento (início, meio e fim de curso do primeiro grau de liberdade do atuado linear) e um sensor indutivo para ser usado como limite de curso do atuador rotacional (segundo grau de liberdade).



```
PERS tooldata pincel:=[TRUE,[[7.86115,- 30.1179,124.792],[1,0,0,0]],[0.2,[85,0,65],[1,0,0,0],0.01,0.01,0.01]];
PERS tooldata Bic:=[TRUE,[[0.500839,-0.574904,226.276],[1,0,0,0]],[0.25,[85,0,65],[1,0,0,0],0.01,0.01,0.01]];
!
```

---

```
PROC pOS_INIT_R1()
```

```
  MoveAbsJ [[-120,0,0,0,90,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]\NoEOffs,v600,z50,Bic\WObj:=sgual;
ENDPROC
```

---

```
PROC R1FACE1()
```

```
!
```

```
! Circulo
```

```
MoveAbsJ [[0,0,0,0,90,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]\NoEOffs,v200,z50,Bic\WObj:=sgual;
```

```
MoveL p1_1,v200,z50,Bic\WObj:=sgual;
```

```
MoveL Offs(p1,0,0,20),v200,z0,Bic\WObj:=sgual;
```

```
MoveL p1,v200,z0,Bic\WObj:=sgual;
```

```
MoveC Offs(p1,75,75,0),Offs(p1,150,0,0),v200,z0,Bic\WObj:=sgual;
```

```
MoveC Offs(p1,75,-75,0),Offs(p1,0,0,0),v200,z0,Bic\WObj:=sgual;
```

```
MoveL Offs(p1,0,0,20),v200,z0,Bic\WObj:=sgual;
```

```
!
```

```
! Triangulo
```

```
MoveJ Offs(p1,10.0481,-37.5,20),v200,z0,Bic\WObj:=sgual;
```

```
MoveL Offs(p1,10.0481,-37.5,0),v200,z0,Bic\WObj:=sgual;
```

```
MoveL Offs(p1,139.952,-37.5,0),v200,z0,Bic\WObj:=sgual;
```

```
MoveL Offs(p1,126.989,-15.0478,0),v200,z0,Bic\WObj:=sgual;
```

```
MoveL Offs(p1,23.0109,-15.0478,0),v200,z0,Bic\WObj:=sgual;
```

```
MoveL Offs(p1,36.0289,7.5,0),v200,z0,Bic\WObj:=sgual;
```

```
MoveL Offs(p1,113.971,7.5,0),v200,z0,Bic\WObj:=sgual;
```

```
MoveL Offs(p1,100.981,30,0),v200,z0,Bic\WObj:=sgual;
```

```
MoveL Offs(p1,49.0192,30,0),v200,z0,Bic\WObj:=sgual;
```

```
MoveL Offs(p1,62.0096,52.5,0),v200,z0,Bic\WObj:=sgual;
```

```
MoveL Offs(p1,87.9904,52.5,0),v200,z0,Bic\WObj:=sgual;
```

```
MoveL Offs(p1,75,75,0),v200,z0,Bic\WObj:=sgual;
```

```
MoveL Offs(p1,62.0096,52.5,0),v200,z0,Bic\WObj:=sgual;
```

```
!
```

```
! Retas Quebradas do Triangulo
```

```
MoveL Offs(p1,62.0096,52.5,20),v200,z0,Bic\WObj:=sgual;
```

```
MoveL Offs(p1,87.9904,52.5,20),v200,z0,Bic\WObj:=sgual;
```

```
!
```

```
MoveL Offs(p1,87.9904,52.5,0),v200,z0,Bic\WObj:=sgual;
```

```

MoveL Offs(p1,100.981,30,0),v200,z0,Bic\WObj:=sgua1;
!
MoveL Offs(p1,100.981,30,20),v200,z0,Bic\WObj:=sgua1;
MoveL Offs(p1,49.0192,30,20),v200,z0,Bic\WObj:=sgua1;
!
MoveL Offs(p1,49.0192,30,0),v200,z0,Bic\WObj:=sgua1;
MoveL Offs(p1,36.0289,7.5,0),v200,z0,Bic\WObj:=sgua1;
!
MoveL Offs(p1,36.0289,7.5,20),v200,z0,Bic\WObj:=sgua1;
MoveL Offs(p1,113.971,7.5,20),v200,z0,Bic\WObj:=sgua1;
!
MoveL Offs(p1,113.971,7.5,0),v200,z0,Bic\WObj:=sgua1;
MoveL Offs(p1,126.989,-15.0478,0),v200,z0,Bic\WObj:=sgua1;
!
MoveL Offs(p1,126.989,-15.0478,20),v200,z0,Bic\WObj:=sgua1;
MoveL Offs(p1,23.0109,-15.0478,20),v200,z0,Bic\WObj:=sgua1;
MoveL Offs(p1,23.0109,-15.0478,0),v200,z0,Bic\WObj:=sgua1;
MoveL Offs(p1,10.0481,-37.5,0),v200,z0,Bic\WObj:=sgua1;
!
MoveL Offs(p1,10.0481,-37.5,20),v200,z0,Bic\WObj:=sgua1;
MoveL p1_1,v200,z50,Bic\WObj:=sgua1;
MoveAbsJ [[0,0,0,0,90,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]\NoEOffs,v200,z50,Bic\WObj:=sgua1;
ENDPROC

```

---

```

PROC R1FACE2()
!
! Circulo
MoveAbsJ [[0,0,0,0,90,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]\NoEOffs,v200,z50,Bic\WObj:=sgua2;
! MoveAbsJ [[0,0,0,-20,15,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]\NoEOffs,v200,z50,
Bic\WObj:=sgua2;
MoveJ p2_1,v200,z50,Bic\WObj:=sgua2;
MoveL Offs(p2,0,0,20),v200,z0,Bic\WObj:=sgua2;
MoveL p2,v200,z0,Bic\WObj:=sgua2;
MoveC Offs(p2,75,75,0),Offs(p2,150,0,0),v200,z0,Bic\WObj:=sgua2;
MoveC Offs(p2,75,-75,0),Offs(p2,0,0,0),v200,z0,Bic\WObj:=sgua2;
MoveL Offs(p2,0,0,20),v200,z0,Bic\WObj:=sgua2;
!
! Triangulo

```

MoveJ Offs(p2,10.0481,-37.5,20),v200,z0,Bic\WObj:=sgua2;  
 MoveL Offs(p2,10.0481,-37.5,0),v200,z0,Bic\WObj:=sgua2;  
 MoveL Offs(p2,139.952,-37.5,0),v200,z0,Bic\WObj:=sgua2;  
 MoveL Offs(p2,126.989,-15.0478,0),v200,z0,Bic\WObj:=sgua2;  
 MoveL Offs(p2,23.0109,-15.0478,0),v200,z0,Bic\WObj:=sgua2;  
 MoveL Offs(p2,36.0289,7.5,0),v200,z0,Bic\WObj:=sgua2;  
 MoveL Offs(p2,113.971,7.5,0),v200,z0,Bic\WObj:=sgua2;  
 MoveL Offs(p2,100.981,30,0),v200,z0,Bic\WObj:=sgua2;  
 MoveL Offs(p2,49.0192,30,0),v200,z0,Bic\WObj:=sgua2;  
 MoveL Offs(p2,62.0096,52.5,0),v200,z0,Bic\WObj:=sgua2;  
 MoveL Offs(p2,87.9904,52.5,0),v200,z0,Bic\WObj:=sgua2;  
 MoveL Offs(p2,75,75,0),v200,z0,Bic\WObj:=sgua2;  
 MoveL Offs(p2,62.0096,52.5,0),v200,z0,Bic\WObj:=sgua2;  
 !  
 ! Retas Quebradas do Triangulo  
 MoveL Offs(p2,62.0096,52.5,20),v200,z0,Bic\WObj:=sgua2;  
 MoveL Offs(p2,87.9904,52.5,20),v200,z0,Bic\WObj:=sgua2;  
 !  
 MoveL Offs(p2,87.9904,52.5,0),v200,z0,Bic\WObj:=sgua2;  
 MoveL Offs(p2,100.981,30,0),v200,z0,Bic\WObj:=sgua2;  
 !  
 MoveL Offs(p2,100.981,30,20),v200,z0,Bic\WObj:=sgua2;  
 MoveL Offs(p2,49.0192,30,20),v200,z0,Bic\WObj:=sgua2;  
 !  
 MoveL Offs(p2,49.0192,30,0),v200,z0,Bic\WObj:=sgua2;  
 MoveL Offs(p2,36.0289,7.5,0),v200,z0,Bic\WObj:=sgua2;  
 !  
 MoveL Offs(p2,36.0289,7.5,20),v200,z0,Bic\WObj:=sgua2;  
 MoveL Offs(p2,113.971,7.5,20),v200,z0,Bic\WObj:=sgua2;  
 !  
 MoveL Offs(p2,113.971,7.5,0),v200,z0,Bic\WObj:=sgua2;  
 MoveL Offs(p2,126.989,-15.0478,0),v200,z0,Bic\WObj:=sgua2;  
 !  
 MoveL Offs(p2,126.989,-15.0478,20),v200,z0,Bic\WObj:=sgua2;  
 MoveL Offs(p2,23.0109,-15.0478,20),v200,z0,Bic\WObj:=sgua2;  
 MoveL Offs(p2,23.0109,-15.0478,0),v200,z0,Bic\WObj:=sgua2;  
 MoveL Offs(p2,10.0481,-37.5,0),v200,z0,Bic\WObj:=sgua2;  
 !

```

MoveL Offs(p2,10.0481,-37.5,20),v200,z0,Bic\WObj:=sgua2;
MoveJ p2_1,v200,z50,Bic\WObj:=sgua2;
MoveAbsJ [[0,0,0,0,90,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]\NoEOffs,v200,z50,Bic\WObj:=sgua2;
!
ENDPROC
!


---


PROC R1_MES_INC()
!
! Circulo
MoveAbsJ [[0,0,0,0,90,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]\NoEOffs,v200,z50,Bic\WObj:=sgua5;
! MoveAbsJ [[0,0,0,0,45,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]\NoEOffs,v200,z50,Bic\WObj:=sgua5;
MoveJ p5_1,v400,z50,Bic\WObj:=sgua5;
MoveL Offs(p5,0,0,20),v200,z0,Bic\WObj:=sgua5;
MoveL p5,v200,z0,Bic\WObj:=sgua5;
MoveC Offs(p5,75,75,0),Offs(p5,150,0,0),v200,z0,Bic\WObj:=sgua5;
MoveC Offs(p5,75,-75,0),Offs(p5,0,0,0),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,0,0,20),v200,z0,Bic\WObj:=sgua5;
!
! Triangulo
MoveJ Offs(p5,10.0481,-37.5,20),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,10.0481,-37.5,0),v200,z0,Bic\WObj:=sgua5;
! MoveL Offs(p5,10.0481,-37.5,20),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,139.952,-37.5,0),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,126.989,-15.0478,0),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,23.0109,-15.0478,0),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,36.0289,7.5,0),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,113.971,7.5,0),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,100.981,30,0),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,49.0192,30,0),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,62.0096,52.5,0),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,87.9904,52.5,0),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,75,75,0),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,62.0096,52.5,0),v200,z0,Bic\WObj:=sgua5;
!
! Retas Quebradas do Triangulo
MoveL Offs(p5,62.0096,52.5,20),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,87.9904,52.5,20),v200,z0,Bic\WObj:=sgua5;
!

```

```

MoveL Offs(p5,87.9904,52.5,0),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,100.981,30,0),v200,z0,Bic\WObj:=sgua5;
!
MoveL Offs(p5,100.981,30,20),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,49.0192,30,20),v200,z0,Bic\WObj:=sgua5;
!
MoveL Offs(p5,49.0192,30,0),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,36.0289,7.5,0),v200,z0,Bic\WObj:=sgua5;
!
MoveL Offs(p5,36.0289,7.5,20),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,113.971,7.5,20),v200,z0,Bic\WObj:=sgua5;
!
MoveL Offs(p5,113.971,7.5,0),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,126.989,-15.0478,0),v200,z0,Bic\WObj:=sgua5;
!
MoveL Offs(p5,126.989,-15.0478,20),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,23.0109,-15.0478,20),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,23.0109,-15.0478,0),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,10.0481,-37.5,0),v200,z0,Bic\WObj:=sgua5;
!
MoveL Offs(p5,10.0481,-37.5,20),v200,z0,Bic\WObj:=sgua5;
MoveJ p5_1,v400,z50,Bic\WObj:=sgua5;
MoveAbsJ [[0,0,0,90,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]\NoEOffs,v200,z50,Bic\ WObj:=sgua5;
!

```

```

ENDPROC

```

```

!

```

---

```

PROC main()

```

```

!

```

```

! INICIALIZAÇÃO DO ROBO

```

```

!

```

```

! Posicao Inicial Robo R1

```

```

!

```

```

! IF DI10_3=1 AND DI10_4=1 AND DI10_5=1 THEN

```

```

!

```

```

  pOS_INIT_R1;

```

```

  ! Set DO10_4;

```

```

  ! Set DO10_5;

```

```

! Set DO10_6;
! WaitTime 2;
! Reset DO10_4;
! Reset DO10_5;
! Reset DO10_6;
! ENDIF
!
! Trajetoria Robo 1 - Face 1(2)
! IF DI10_4=1 AND DI10_5=0 AND DI10_3=1 THEN
!
! R1FACE1;
! R1FACE2;
!
! Set DO10_4;
! Set DO10_5;
! Set DO10_7;
! WaitTime 2;
! Reset DO10_4;
! Reset DO10_5;
! Reset DO10_7;
! ENDIF
!
! Trajetoria Robo 1 - Mesa Inclorada
! IF DI10_5=1 AND DI10_3=1 AND DI10_4=0 THEN
!
! R1_MES_INC;
! pOS_INIT_R1;
!Set DO10_4;
!Set DO10_5;
!Set DO10_6;
!Set DO10_7;
!WaitTime 2;
!Reset DO10_4;
!Reset DO10_5;
!Reset DO10_6;
!Reset DO10_7;
!ENDIF
ENDPROC

```

ENDMODULE

## C.2 Programa - Robô Industrial ABB IRB 1400

%%%

VERSION:1

LANGUAGE:ENGLISH

%%%

MODULE aut1400

PERS wobjdata sgua3:=[FALSE,TRUE,"",[[1383.42,183.711,811.144],[0.649735,0.652177,-0.28073,-0.271478]],[[1.06305,1.72323,-0.33319],[0.999997,0.002284,2.4E-05,8E-06]]];

PERS wobjdata sgua4:=[FALSE,TRUE,"",[[1529.4,317.814,815.56],[0.255224,0.262435,-0.667034,-0.648888]],[[0,0,0],[1,0,0,0]]];

PERS wobjdata sgua5:=[FALSE,TRUE,"",[[1275.91,183.04,846.869],[0.845215,0.136804,-0.364511,-0.3661]],[[0,0,0],[1,0,0,0]]];

CONST robtarget p5\_1:=[[-13.45,77.89,137.49],[0.013321,0.390585,-0.92037,-0.013705],[0,-1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget p5:=[[19.1,98.98,20.1],[0.013349,0.390597,-0.920364,-0.013697],[0,-1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget p4\_1:=[[116.92,110.47,199.54],[0.306494,0.783191,-0.494542,-0.219323],[0,-1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget p4:=[[60.72,105.56,21.71],[0.306492,0.783199,-0.494532,-0.219322],[0,-1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget p3:=[[28.04,102.06,21.56],[0.156073,-0.373344,0.879487,-0.250515],[0,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS tooldata pincel:=[TRUE,[[7.86115,-30.1179,124.792],[1,0,0,0]],[0.2,[85,0,65],[1,0,0,0],0.01,0.01,0.01]];

PERS tooldata Bic:=[TRUE,[[0.500839,-0.574904,226.276],[1,0,0,0]],[0.25,[85,0,65],[1,0,0,0],0.01,0.01,0.01]];

!

---

PROC pOS\_INIT\_R2()

MoveAbsJ [[120,0,0,0,90,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]\NoEOffs,v500,z100,

Bic\WObj:=sgua3;

ENDPROC

!

---

PROC R2FACE3()

!

! Circulo

MoveAbsJ [[0,0,0,0,90,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]\NoEOffs,v400,z50,Bic\WObj:=sgua3;  
 ! MoveJ p3,v400,z50,Bic\WObj:=sgua3;  
 MoveL Offs(p3,0,0,20),v200,z0,Bic\WObj:=sgua3;  
 MoveL p3,v200,z0,Bic\WObj:=sgua3;  
 MoveC Offs(p3,75,75,0),Offs(p3,150,0,0),v200,z0,Bic\WObj:=sgua3;  
 MoveC Offs(p3,75,-75,0),Offs(p3,0,0,0),v200,z0,Bic\WObj:=sgua3;  
 MoveL Offs(p3,0,0,20),v200,z0,Bic\WObj:=sgua3;  
 !  
 ! Triangulo  
 MoveJ Offs(p3,10.0481,-37.5,20),v200,z0,Bic\WObj:=sgua3;  
 MoveL Offs(p3,10.0481,-37.5,0),v200,z0,Bic\WObj:=sgua3;  
 ! MoveL Offs(p3,10.0481,-37.5,20),v200,z0,Bic\WObj:=sgua3;  
 MoveL Offs(p3,139.952,-37.5,0),v200,z0,Bic\WObj:=sgua3;  
 MoveL Offs(p3,126.989,-15.0478,0),v200,z0,Bic\WObj:=sgua3;  
 MoveL Offs(p3,23.0109,-15.0478,0),v200,z0,Bic\WObj:=sgua3;  
 MoveL Offs(p3,36.0289,7.5,0),v200,z0,Bic\WObj:=sgua3;  
 MoveL Offs(p3,113.971,7.5,0),v200,z0,Bic\WObj:=sgua3;  
 MoveL Offs(p3,100.981,30,0),v200,z0,Bic\WObj:=sgua3;  
 MoveL Offs(p3,49.0192,30,0),v200,z0,Bic\WObj:=sgua3;  
 MoveL Offs(p3,62.0096,52.5,0),v200,z0,Bic\WObj:=sgua3;  
 MoveL Offs(p3,87.9904,52.5,0),v200,z0,Bic\WObj:=sgua3;  
 MoveL Offs(p3,75,75,0),v200,z0,Bic\WObj:=sgua3;  
 MoveL Offs(p3,62.0096,52.5,0),v200,z0,Bic\WObj:=sgua3;  
 !  
 ! Retas Quebradas do Triangulo  
 MoveL Offs(p3,62.0096,52.5,20),v200,z0,Bic\WObj:=sgua3;  
 MoveL Offs(p3,87.9904,52.5,20),v200,z0,Bic\WObj:=sgua3;  
 !  
 MoveL Offs(p3,87.9904,52.5,0),v200,z0,Bic\WObj:=sgua3;  
 MoveL Offs(p3,100.981,30,0),v200,z0,Bic\WObj:=sgua3;  
 !  
 MoveL Offs(p3,100.981,30,20),v200,z0,Bic\WObj:=sgua3;  
 MoveL Offs(p3,49.0192,30,20),v200,z0,Bic\WObj:=sgua3;  
 !  
 MoveL Offs(p3,49.0192,30,0),v200,z0,Bic\WObj:=sgua3;  
 MoveL Offs(p3,36.0289,7.5,0),v200,z0,Bic\WObj:=sgua3;  
 MoveL Offs(p3,36.0289,7.5,20),v200,z0,Bic\WObj:=sgua3;  
 MoveL Offs(p3,113.971,7.5,20),v200,z0,Bic\WObj:=sgua3;

```

!
MoveL Offs(p3,113.971,7.5,0),v200,z0,Bic\WObj:=sgua3;
MoveL Offs(p3,126.989,-15.0478,0),v200,z0,Bic\WObj:=sgua3;
!
MoveL Offs(p3,126.989,-15.0478,20),v200,z0,Bic\WObj:=sgua3;
MoveL Offs(p3,23.0109,-15.0478,20),v200,z0,Bic\WObj:=sgua3;
MoveL Offs(p3,23.0109,-15.0478,0),v200,z0,Bic\WObj:=sgua3;
MoveL Offs(p3,10.0481,-37.5,0),v200,z0,Bic\WObj:=sgua3;
!
MoveL Offs(p3,10.0481,-37.5,20),v200,z0,Bic\WObj:=sgua3;
ENDPROC

```

---

```

PROC R2FACE4()

```

```

!
! Circulo
MoveAbsJ [[0,0,0,0,90,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]\NoEOffs,v400,z50,Bic\WObj:=sgua4;
MoveAbsJ [[0,0,0,-20,15,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]\NoEOffs,v200,z50,Bic\WObj:=sgua4;
MoveJ p4_1,v200,z50,Bic\WObj:=sgua4;
MoveL Offs(p4,0,0,20),v200,z0,Bic\WObj:=sgua4;
MoveL p4,v200,z0,Bic\WObj:=sgua4;
MoveC Offs(p4,75,75,0),Offs(p4,150,0,0),v200,z0,Bic\WObj:=sgua4;
MoveC Offs(p4,75,-75,0),Offs(p4,0,0,0),v200,z0,Bic\WObj:=sgua4;
MoveL Offs(p4,0,0,20),v200,z0,Bic\WObj:=sgua4;
!
! Triangulo
MoveJ Offs(p4,10.0481,-37.5,20),v200,z0,Bic\WObj:=sgua4;
MoveL Offs(p4,10.0481,-37.5,0),v200,z0,Bic\WObj:=sgua4;
! MoveL Offs(p4,10.0481,-37.5,20),v200,z0,Bic\WObj:=sgua4;
MoveL Offs(p4,139.952,-37.5,0),v200,z0,Bic\WObj:=sgua4;
MoveL Offs(p4,126.989,-15.0478,0),v200,z0,Bic\WObj:=sgua4;
MoveL Offs(p4,23.0109,-15.0478,0),v200,z0,Bic\WObj:=sgua4;
MoveL Offs(p4,36.0289,7.5,0),v200,z0,Bic\WObj:=sgua4;
MoveL Offs(p4,113.971,7.5,0),v200,z0,Bic\WObj:=sgua4;
MoveL Offs(p4,100.981,30,0),v200,z0,Bic\WObj:=sgua4;
MoveL Offs(p4,49.0192,30,0),v200,z0,Bic\WObj:=sgua4;
MoveL Offs(p4,62.0096,52.5,0),v200,z0,Bic\WObj:=sgua4;
MoveL Offs(p4,87.9904,52.5,0),v200,z0,Bic\WObj:=sgua4;
MoveL Offs(p4,75,75,0),v200,z0,Bic\WObj:=sgua4;

```

```

MoveL Offs(p4,62.0096,52.5,0),v200,z0,Bic\WObj:=sgua4;
!
! Retas Quebradas do Triangulo
MoveL Offs(p4,62.0096,52.5,20),v200,z0,Bic\WObj:=sgua4;
MoveL Offs(p4,87.9904,52.5,20),v200,z0,Bic\WObj:=sgua4;
!
MoveL Offs(p4,87.9904,52.5,0),v200,z0,Bic\WObj:=sgua4;
MoveL Offs(p4,100.981,30,0),v200,z0,Bic\WObj:=sgua4;
!
MoveL Offs(p4,100.981,30,20),v200,z0,Bic\WObj:=sgua4;
MoveL Offs(p4,49.0192,30,20),v200,z0,Bic\WObj:=sgua4;
!
MoveL Offs(p4,49.0192,30,0),v200,z0,Bic\WObj:=sgua4;
MoveL Offs(p4,36.0289,7.5,0),v200,z0,Bic\WObj:=sgua4;
!
MoveL Offs(p4,36.0289,7.5,20),v200,z0,Bic\WObj:=sgua4;
MoveL Offs(p4,113.971,7.5,20),v200,z0,Bic\WObj:=sgua4;
!
MoveL Offs(p4,113.971,7.5,0),v200,z0,Bic\WObj:=sgua4;
MoveL Offs(p4,126.989,-15.0478,0),v200,z0,Bic\WObj:=sgua4;
!
MoveL Offs(p4,126.989,-15.0478,20),v200,z0,Bic\WObj:=sgua4;
MoveL Offs(p4,23.0109,-15.0478,20),v200,z0,Bic\WObj:=sgua4;
MoveL Offs(p4,23.0109,-15.0478,0),v200,z0,Bic\WObj:=sgua4;
MoveL Offs(p4,10.0481,-37.5,0),v200,z0,Bic\WObj:=sgua4;
!
MoveL Offs(p4,10.0481,-37.5,20),v200,z0,Bic\WObj:=sgua4;
MoveJ p4_1,v200,z50,Bic\WObj:=sgua4;
MoveAbsJ [[0,0,0,0,90,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]\NoEOffs,v400,z50,Bic\WObj:=sgua4;
!
ENDPROC
!


---


PROC R2_MES_INC()
!
! Circulo
MoveAbsJ [[0,0,0,0,90,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]\NoEOffs,v400,z50,Bic\WObj:=sgua5;
MoveAbsJ [[0,0,0,0,45,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]\NoEOffs,v400,z50,Bic\WObj:=sgua5;
MoveJ p5_1,v400,z50,Bic\WObj:=sgua5;

```

MoveL Offs(p5,0,0,20),v200,z0,Bic\WObj:=sgua5;  
 MoveL p5,v200,z0,Bic\WObj:=sgua5;  
 MoveC Offs(p5,75,75,0),Offs(p5,150,0,0),v200,z0,Bic\WObj:=sgua5;  
 MoveC Offs(p5,75,-75,0),Offs(p5,0,0,0),v200,z0,Bic\WObj:=sgua5;  
 MoveL Offs(p5,0,0,20),v200,z0,Bic\WObj:=sgua5;  
 !  
 ! Triangulo  
 MoveJ Offs(p5,10.0481,-37.5,20),v200,z0,Bic\WObj:=sgua5;  
 MoveL Offs(p5,10.0481,-37.5,0),v200,z0,Bic\WObj:=sgua5;  
 ! MoveL Offs(p5,10.0481,-37.5,20),v200,z0,Bic\WObj:=sgua5;  
 MoveL Offs(p5,139.952,-37.5,0),v200,z0,Bic\WObj:=sgua5;  
 MoveL Offs(p5,126.989,-15.0478,0),v200,z0,Bic\WObj:=sgua5;  
 MoveL Offs(p5,23.0109,-15.0478,0),v200,z0,Bic\WObj:=sgua5;  
 MoveL Offs(p5,36.0289,7.5,0),v200,z0,Bic\WObj:=sgua5;  
 MoveL Offs(p5,113.971,7.5,0),v200,z0,Bic\WObj:=sgua5;  
 MoveL Offs(p5,100.981,30,0),v200,z0,Bic\WObj:=sgua5;  
 MoveL Offs(p5,49.0192,30,0),v200,z0,Bic\WObj:=sgua5;  
 MoveL Offs(p5,62.0096,52.5,0),v200,z0,Bic\WObj:=sgua5;  
 MoveL Offs(p5,87.9904,52.5,0),v200,z0,Bic\WObj:=sgua5;  
 MoveL Offs(p5,75,75,0),v200,z0,Bic\WObj:=sgua5;  
 MoveL Offs(p5,62.0096,52.5,0),v200,z0,Bic\WObj:=sgua5;  
 !  
 ! Retas Quebradas do Triangulo  
 MoveL Offs(p5,62.0096,52.5,20),v200,z0,Bic\WObj:=sgua5;  
 MoveL Offs(p5,87.9904,52.5,20),v200,z0,Bic\WObj:=sgua5;  
 !  
 MoveL Offs(p5,87.9904,52.5,0),v200,z0,Bic\WObj:=sgua5;  
 MoveL Offs(p5,100.981,30,0),v200,z0,Bic\WObj:=sgua5;  
 !  
 MoveL Offs(p5,100.981,30,20),v200,z0,Bic\WObj:=sgua5;  
 MoveL Offs(p5,49.0192,30,20),v200,z0,Bic\WObj:=sgua5;  
 !  
 MoveL Offs(p5,49.0192,30,0),v200,z0,Bic\WObj:=sgua5;  
 MoveL Offs(p5,36.0289,7.5,0),v200,z0,Bic\WObj:=sgua5;  
 !  
 MoveL Offs(p5,36.0289,7.5,20),v200,z0,Bic\WObj:=sgua5;  
 MoveL Offs(p5,113.971,7.5,20),v200,z0,Bic\WObj:=sgua5;  
 !

```

MoveL Offs(p5,113.971,7.5,0),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,126.989,-15.0478,0),v200,z0,Bic\WObj:=sgua5;
!
MoveL Offs(p5,126.989,-15.0478,20),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,23.0109,-15.0478,20),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,23.0109,-15.0478,0),v200,z0,Bic\WObj:=sgua5;
MoveL Offs(p5,10.0481,-37.5,0),v200,z0,Bic\WObj:=sgua5;
!
MoveL Offs(p5,10.0481,-37.5,20),v200,z0,Bic\WObj:=sgua5;
MoveJ p5_1,v400,z50,Bic\WObj:=sgua5;
MoveAbsJ [[0,0,0,90,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]\NoEOffs,v400,z50,Bic\WObj:=sgua5;
!
ENDPROC

```

---

```

!
PROC main()
!
!   OBS.: WObj:=sgua3 e 4 são as duas faces do quadrado;
!       WObj:=sgua5 é a face superior quando inclina o quadrado
!
! INICIALIZAÇÃO DO ROBO
!
! Posicao Inicial Robo R2
!
! IF DI10_3=0 AND DI10_4=1 AND DI10_5=1 THEN
!
pOS_INIT_R2;
!
!Set DO10_6;
!reg1 :=0;
!WaitTime 2;
!Reset DO10_6;
!   ENDIF
!
! Trajetoria Robo 2 - Face 3 e 4
! IF DI10_4=1 AND DI10_5=0 AND DI10_3=0 THEN
!
R2FACE3;

```

```
R2FACE4;
!
!Set DO10_7;
!reg1 :=0;
!WaitTime 2;
!Reset DO10_7;
!  ENDIF
!
! Trajetoria Robo 2 - Mesa Inclinata
!  IF DI10_5=1 AND DI10_3=0 AND DI10_4=0 THEN
!
! R2_MES_INC;
!
! Set DO10_6;
! Set DO10_7;
!! reg1 :=0;
! WaitTime 2;
! Reset DO10_6;
! Reset DO10_7;
!  ENDIF
ENDPROC
ENDMODULE
```

## **ANEXO D - INTRODUÇÃO AO LabVIEW**

Para melhor entendimento do procedimento de programação pelo software *LabVIEW* utilizado no capítulo 4, será apresentada nas seções abaixo a descrição da linguagem utilizada.

### **D.1 Introdução ao *LabVIEW***

O *LabVIEW* (*Laboratory Virtual Instrument Engineering Workbench*) utiliza uma linguagem de programação chamada G. Essa linguagem possui diretivas, como PASCAL e C, porém ao invés de utilizar comandos na forma de texto para gerar as linhas de código, usa uma linguagem de programação gráfica, ou seja, o programa é feito na forma de um diagrama de blocos.

Por utilizar uma estrutura de programação orientada pelo fluxo de dados e hierárquica, o *LabVIEW* torna simples a implementação de sistemas complexos que englobam aquisição e manipulação de dados, ou ainda o controle de equipamentos através do computador. Além disso, o *LabVIEW* inclui diversas bibliotecas compostas por componentes, contendo funções para aplicações específicas (algoritmos de análise estatística, processamento e geração de sinais, etc.) (LabView<sup>®</sup>, 1996).

#### **D.2.1 Instrumentos Virtuais**

Qualquer programa feito em *LabVIEW* é chamado de instrumento virtual (*VI - Virtual Instrument*) já que sua aparência e operação assemelham-se às de instrumentos reais. Um *VI*, assim como um programa usual, é composto por um conjunto de instruções que fazem a

manipulação e fluxo dos dados, e por uma interface com o usuário, é possível encontrar as entradas e saídas necessárias. Basicamente, pode-se identificar em um *VI* duas partes que o compõem:

- **Diagrama de blocos** - é a estrutura do programa propriamente dita que contém o código fonte construído de forma gráfica;

- **Painel frontal** - constitui a interface com o usuário, apresentando de forma visual todos os controles, gráficos e indicadores que formam uma tela, que simula o painel físico de um instrumento. Este pode ser formado por botões, *leds*, *knobs* e indicadores que permitem a interação através do *mouse* ou do teclado do computador.

Pode-se ainda encapsular um *VI* inteiro (isto é, diagrama de blocos + painel frontal) em um módulo re-utilizável dentro de um outro *VI*. Esse módulo encapsulado constituirá um *subVI*. Assim sendo, podemos incluir um terceiro ingrediente componente de um *VI* genérico:

- **Ícone / conector** - define as entradas e saídas do *VI* acessíveis à conexão quando este é utilizado como um *subVI* dentro de um outro *VI* (é análogo às definições de entrada e saída para se usar uma rotina como função dentro de um programa numa linguagem convencional). O ícone é a definição da aparência gráfica que se deseja que este *VI* tenha no diagrama de blocos quando usado como um *subVI*.

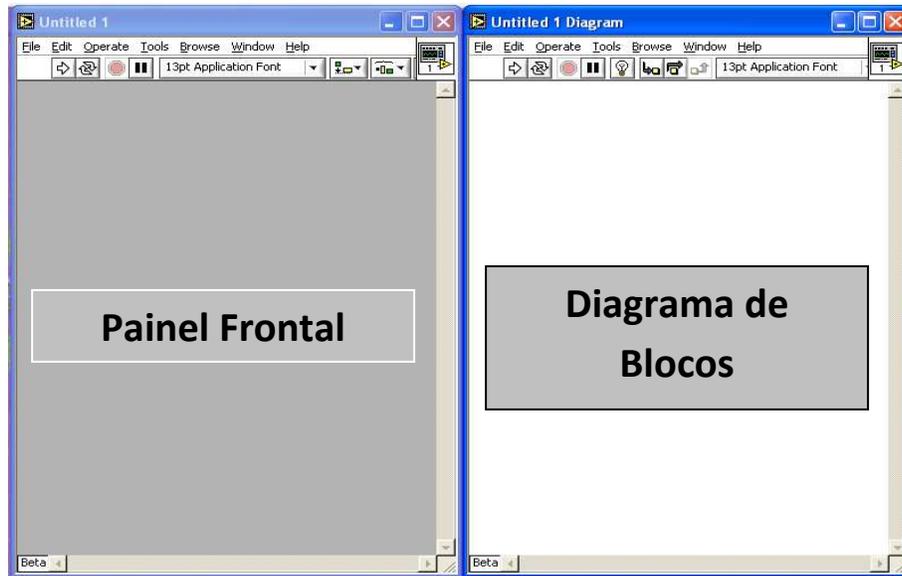


Figura D.1: Painel Frontal e Diagrama de Blocos

Na parte superior do painel frontal há uma barra de ferramentas como descrita a seguir:



onde,



Roda o *VI*, uma vez.



Roda o *VI*, continuamente, isto é, quando chega ao final, volta novamente ao início e recomeça.



Aborta a execução do *VI*.



Pausa / Continua a execução do *VI*.



Ícone / Conector.

## D.2.2 Paleta de Ferramentas

Para editar, operar ou construir um VI são usadas várias ferramentas. Desta forma, o conhecimento e compreensão dessas ferramentas e sua utilização são essenciais para usar o *LabVIEW*.



Figura D.2: Paleta de Ferramentas

Descrição das ferramentas principais:



Permite a operação dos controles e botões no painel frontal (*Operation Tool*).



Posiciona, formata e seleciona objetos (*Position Tool*).



Edita textos e cria rótulos (*Text Tool*).



Conecta objetos no diagrama de blocos (*Wiring Tool*).

## D.2.3 Paleta de Função e Controle

As paletas de função e controle contém sub-paletas de objetos que podem ser utilizadas para criar um VI. A paleta de função serve para montar o diagrama de bloco enquanto a paleta de controle para indicar indicadores e controles.

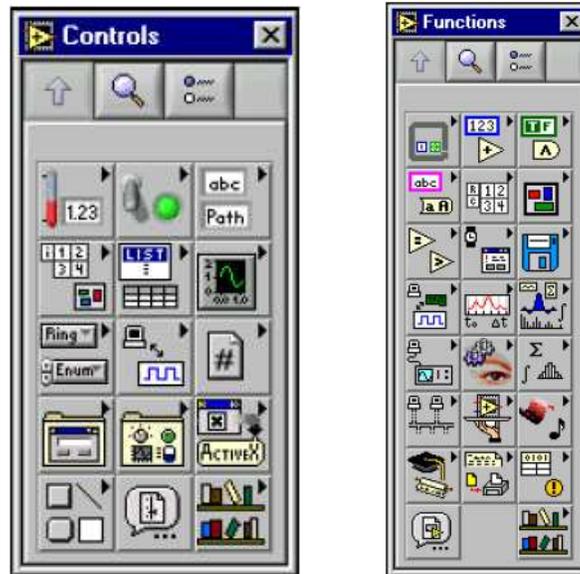


Figura D.3: Paleta de controle e função

## D.2.4 Controles e Indicadores

No painel frontal as entradas e saídas do VI são representadas respectivamente por controles e indicadores, que visualmente podem ser apresentados seja como *knobs*, botões, indicadores digitais, *leds*, ou ainda sob vários outros formatos e tipos.

## D.2.5 Estruturas e Gráficos

Assim como em outras linguagens nos programas contém laços, no *LabVIEW* há determinadas estruturas que representam graficamente laços, como o *for* e o *while* no diagrama de blocos. Além disso, é possível usar estruturas de decisão, como o *case* ou ainda estruturas específicas do *LabVIEW*, como o *sequence* e a *formula node*.

Quando se faz necessário que os dados gerados por um VI sejam apresentados através de um gráfico, não é preciso dar comandos especiais ou fazer uma rotina só para isso, como em linguagens convencionais. No *LabVIEW* já existe uma função implementada apenas para gerar um gráfico, que pode ser formatado de acordo com o padrão que se deseja obter.

## D.2.6 Componentes utilizados na Programação

### D.2.6.1 Ligações

Transfere os dados entre objetos do diagrama de bloco através de ligações. São semelhantes às variáveis em linguagens de programação baseadas em texto. Cada ligação tem uma única fonte de dados, mas podem ser ligadas a vários VI's e funções que fazem a leitura dos dados. As ligações são de cores, estilos e espessuras diferentes, dependendo de seus tipos de dados. A tabela D.1 apresenta os tipos mais comuns de ligações.

Tipo de Ligação	Escalar	Arranjo 1D	Arranjo 2D	Cor
Numérico				Laranja (ponto flutuante), Azul (inteiro)
Booleano				Verde
String				Rosa

Tabela D.1: Tipos de ligação

### D.2.6.2 Bloco - *Wait Until Next ms Multiple*

Esta função sincroniza atividades. É possível chamar esta função em um laço para controlar a taxa da execução. Esta função faz chamadas assíncronas do sistema, mas os seus nós funcionam sincronizadamente. Conseqüentemente, a execução não termina até que o tempo especificado decorra.

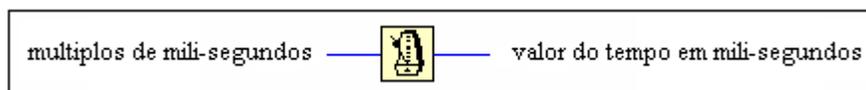


Figura D.4: Bloco de Sincronismo

 **múltiplos de milissegundos** - entrada que especifica quantos milissegundos decorrem quando os funcionamentos do VI.

 **valor do tempo em milissegundo** - retorna o valor do temporizador em milissegundo.

### D.2.6.3 Bloco - Mean

Computa a média ( $\mu$ ) dos valores na seqüência de entrada  $X$ .

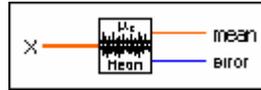


Figura D.5: Bloco de Média

  $X$  - seqüências de entrada.

 *média* - média dos valores na seqüência de entrada  $X$ .

 *erro* - retorna todo o erro ou circunstância de advertência do VI.

### Detalhe da Média

Este VI usa a seguinte fórmula:

$$\mu = \frac{1}{n} \sum_{i=0}^{n-1} x_i \quad (D.1)$$

onde  $n$  é o número dos elementos em  $X$ .

### D.2.6.4 Bloco *Bundle*

Utilizada para montar um conjunto dos elementos individuais. Pode ser utilizada para mudar os valores de elementos individuais em um conjunto já existente sem ter que especificar novos valores para todos os elementos. A placa do conector indica os tipos de dados.

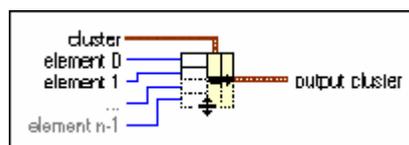


Figura D.6: Bloco para agrupamento

 **cluster** - conjunto cujo valor queira mudar. O número de terminais da entrada deve combinar com o número dos elementos no conjunto da entrada.

 **element 1..n** - aceita qualquer tipo de dados.

 **output cluster** - resultado do conjunto.

### D.2.6.5 Estrutura de Sequência

Consiste em um ou mais sub-diagramas, ou molduras, que executam seqüencialmente. Um clique a direita da estrutura é utilizado para adicionar e suprimir molduras, ou para criar local de seqüência com objetivo de passar dados entre as molduras.

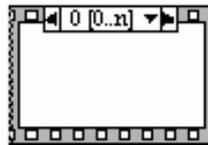


Figura D.7: Estrutura Sequencial

### D.2.6.6 Bloco Divisor

Computa o quociente das entradas.

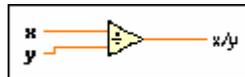


Figura D.8: Bloco divisor

 **x** - podem ser um número, uma disposição ou um conjunto escalar dos números, disposição dos conjuntos dos números, e assim por diante.

 **y** - podem ser um número, uma disposição ou um conjunto escalar dos números, disposição dos conjuntos dos números, e assim por diante.

  $x/y$  - número de dupla precisão como ponto flutuante se  $x$  e  $y$  forem inteiros. No general, o tipo da saída é a representação larga das entradas, se as entradas não forem inteiros ou se suas representações diferirem.

### D.2.6.7 Bloco de Entrada Analógica para Aquisição da Forma de Onda

Adquire um número especificado das amostras em uma taxa especificada de um único canal da entrada e retorna os dados adquiridos. A entrada analógica adquire uma forma de onda o VI executa uma medida programada de uma forma de onda (leituras múltiplas da tensão em uma taxa especificada) em um único canal da entrada análoga. Se ocorrer um erro, uma caixa de diálogo aparece, dando lhe a opção para parar o VI ou para continuar.

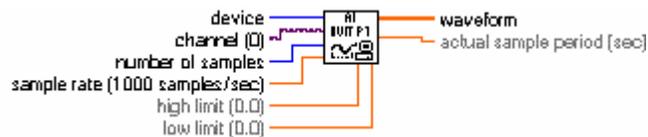


Figura D.9: Bloco de Aquisição de Gráfico

 **dispositivo** - número do dispositivo atribuído à configuração.

 **canal** - identifica o canal da entrada análoga a medir.

 **número de amostras** - número de amostras do canal que o VI adquirem antes que a aquisição termine. Para este parâmetro - 1000 amostras.

 **taxa de amostras** - número pedido das amostras por segundo onde o VI adquire do canal especificado. Para este parâmetro - taxa de 1000 amostras /segundo.

 **limite elevado** - nível previsto mais elevado dos sinais que queira medir. A entrada é  $\theta$ . Se incorporar  $\theta$ , o sistema usa os limites incorporados ao máximo, ou se estiver usando nome no canal, seleciona os limites para configuração do canal. Expressa o valor limite nas unidades de leituras. Este valor computa o ganho.

 **limite baixo** - nível previsto mais baixo dos sinais que queira medir. A entrada é  $0$ . Se incorporar  $0$ , o sistema usa os limites incorporados ao máximo, ou se você estiver usando nome no canal, seleciona os limites para configuração do canal. Expressa o valor de limite nas unidades de leituras. Este valor computa o ganho.

 **forma de onda** - contém dados de entrada análoga escalados.

 **período real da amostra** - intervalo real entre amostras, que é o inverso da taxa real da amostra. O período real da amostra pode diferir da taxa pedida da amostra, dependendo das potencialidades de sua unidade física.

#### D.2.6.8 Bloco *For Loop*

Executa seus tempos do sub-diagrama  $n$ , onde  $n$  é o valor ligado ao terminal da contagem ( $N$ ). O terminal da iteração ( $i$ ) fornece a contagem da iteração do laço atual, que varia de  $0$  a  $n-1$ .



Figura D.10: Bloco *For Loop*

#### D.2.6.9 Bloco Derivativo

Executa uma diferenciação discreta do sinal de amostra  $X$ .

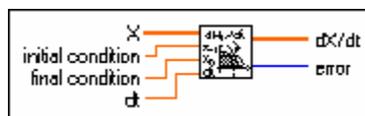


Figura D.11: Bloco derivativo

  $X$  – é o sinal de amostra.

 *condição inicial* - descrito pela equação abaixo nesta seção.

 *condição final* - descrito pela equação abaixo nesta seção.

 **dt** - é o intervalo de amostragem e deve ser maior que zero.

 **dX/dt** - é a derivada do sinal de entrada.

 *erro* - returns any error or warning from the VI.

### Detalhes da Derivação $x(t)$

A diferenciação  $f(t)$  de uma função  $F(t)$  é definida como:

$$f(t) = \frac{d}{dt} F(t). \quad (\text{D.2})$$

Permitir que  $Y$  represente a amostra de saída da seqüência  $\mathbf{d/dt X}$ . A execução discreta é dada por:

$$y_i = \frac{1}{2dt} (x_{i+1} - x_{i-1}). \quad (\text{D.3})$$

para  $i = 0, 1, 2, \dots, n-1$ , quando  $n$  é o número de amostra em  $x(t)$ ,

$x_{-1}$ : é especificado pela condição inicial quando  $i = 0$ , e  $x_n$  é especificado pela condição final quando  $i = n-1$ .

A condição inicial e a condição final minimizam o erro nos limites.

## **ANEXO E - ARTIGOS PUBLICADOS**

1. Paracêncio, L. G. M., Rosário, J. M., Hermini, H. A., Sanchez, O. F. A., “Collaborative Mechatronical Platform for Integrated Robotics Environment”, International Workshop ROBOCONTROL’08, 2008, Bauru - SP.
2. Sanchez, O. F. A., Rosário, J. M., Uribe, A. J., Paracêncio, L. G. M.,: “Estratégia de Controle para Mão Mecânica Baseado em Informação de Forma: Experiências com Mão Mecatrônica MUC-1”, International Workshop ROBOCONTROL’08, 2008, Bauru - SP.
3. Filippo, T. R. M., Hermini, H. A., Balthazar, J. M., Paracêncio, L. G. M.,: “On some Methodologies for the Implementation of Active Orthosis of Inferior, Applicable Limbs in Servant-assistance of Physiotherapy Processes of Evaluation and Whitewashing”, 5th Brazilian Conference on Dynamics, Control and their Applications - DINCON, 2006, Guaratingueta - SP.
4. Paracêncio, L. G. M., Hermini, H. A., Balthazar, J. M.,: “Projeto e Montagem Experimental de um Manipulador Robótico Não-linear de Dois Graus de Liberdade”, IV Congresso Nacional de Engenharia Mecânica - CONEM, 2006, Guaratinguetá - SP.
5. Paracêncio, L. G. M., Hermini, H. A., Balthazar, J. M.,: “Sobre a Montagem e Resultados Experimentais de um Manipulador Robótico Não-linear e Não-holonomo de Dois Graus de Liberdade”, 5th Brazilian Conference on Dynamics, Control and their Applications - DINCON, 2006, Guaratinguetá - SP.