

TESE DEFENDIDA POR Clivaldo de  
Oliveira E APROVADA PEL  
COMISSÃO JULGADORA EM 28/02/2003

João Maurício Rosário  
ORIENTADOR

**UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA MECÂNICA**

**Proposta de Implementação de um Sistema de  
Controle Didático baseado na modelagem  
Dinâmica de Robôs Industriais**

Autor: **Clivaldo de Oliveira**

Orientador: **João Maurício Rosário**

2003

UNICAMP  
BIBLIOTECA CENTRAL  
SEÇÃO CIRCULANTE

**UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA MECÂNICA  
DEPARTAMENTO DE PROJETO MECÂNICO**

**Proposta de Implementação de um Sistema de  
Controle Didático baseado na modelagem  
Dinâmica de Robôs Industriais**

Autor: **Clivaldo de Oliveira**

Orientador: **João Maurício Rosário**

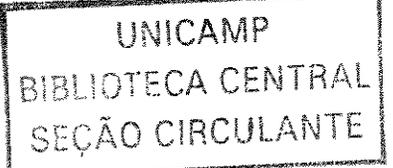
Curso: Engenharia Mecânica

Área de Concentração: Mecânica dos Sólidos

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, 2003

S.P. – Brasil



UNIDADE	BC
Nº CHAMADA	UNICAMP
	OL4p
V	EX
TOMBO BC	55943
PROC.	16/124103
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	R\$ 11,00
DATA	08/10/03
Nº CPD	

CM001B9217-5

Bibid 302014

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

OL4p

Oliveira, Clivaldo de

Proposta de implementação de um sistema de controle didático baseado na modelagem dinâmica de robôs industriais / Clivaldo de Oliveira. --Campinas, SP: [s.n.], 2003.

Orientador: João Maurício Rosário.

Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica.

1. Sistemas lineares invariantes no tempo. 2. Controladores PID. 3. Servomecanismos. 4. Controle automático. 5. Sistemas de controle por realimentação.  
Rosário, João Maurício. II. Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica. III. Título.

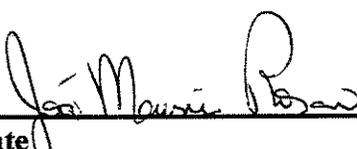
**UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA MECÂNICA  
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA  
DEPARTAMENTO DE PROJETO MECÂNICO**

**TESE DE DOUTORADO**

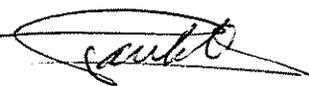
**Proposta de Implementação de um Sistema de  
Controle Didático baseado na modelagem  
Dinâmica de Robôs Industriais**

Autor: Clivaldo de Oliveira

Orientador: João Maurício Rosário

  
\_\_\_\_\_  
**Prof. Dr. João Maurício Rosário, Presidente**  
Universidade Estadual de Campinas

  
\_\_\_\_\_  
**Prof. Dr. José Manoel Balthazar**  
Universidade Estadual de Campinas

  
\_\_\_\_\_  
**Prof. Dr. Paulo Roberto Gardel Kurka**  
Universidade Estadual de Campinas

  
\_\_\_\_\_  
**Prof. Dr. João Carlos Mendes Carvalho**  
Universidade Federal de Uberlândia

  
\_\_\_\_\_  
**Prof. Dr. Vladimir Ivanovitch Dynnikov**  
Universidade Federal do Espírito Santo

UNICAMP  
BIBLIOTECA CENTRAL  
SEÇÃO CIRCULANTE

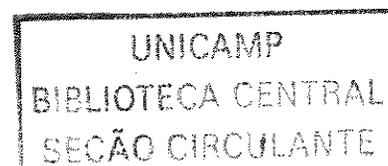
## **Dedicatória**

Aos meus pais.

## Agradecimentos

No desenvolvimento deste trabalho, contamos com a colaboração e apoio de muitas pessoas e instituições. Queremos aqui registrar nosso reconhecimento e gratidão.

- Ao meu orientador, Prof. Dr. João Maurício Rosário, pela eficiente orientação.
- Ao Professor Dr. Jocarly Patrocínio de Souza, da Universidade de Passo Fundo, pelos conhecimentos compartilhados e orientação nos trabalhos realizados no laboratório.
- À FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) pelo suporte financeiro.
- Aos amigos Dr. Cláudio Eduardo Aravechia de Sá e Sergio Adriani David, pelas proveitosas informações, sugestões e materiais fornecidos e, principalmente, pela amizade.
- À minha esposa, Vera Lúcia Freitas Marinho, pela paciência e compreensão demonstradas, bem como pela motivação oferecida em todas as fases deste trabalho.
- E finalmente a todos que, direta ou indiretamente, contribuíram para a conclusão deste projeto.



## **Resumo**

OLIVEIRA, Clivaldo de, Proposta de Implementação de Um Sistema de Controle Aplicado a Robôs Industriais Baseado na Modelagem Dinâmica, Campinas, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2003. 158 p. Tese Doutorado.

Este trabalho tem como objetivo a modelagem e simulação de sistemas dinâmicos, com ênfase no estudo e implementação de controladores de posição de juntas robóticas. Inicialmente é realizado o estudo dos elementos constituintes de uma junta robótica, tais como: motor de corrente contínua, inércia, redutores e acoplamento. A partir desse estudo é possível a definição da estratégia de controle a ser utilizada, incluindo o desenvolvimento de um gerador de trajetórias utilizado como referencia dentro de uma malha de controle envolvendo estes elementos. A validação final dos algoritmos de controle desenvolvidos é realizada através de uma bancada experimental descrita nesse trabalho.

### **Palavras Chave**

- Modelagem, controladores.

## **Abstract**

OLIVEIRA, Clivaldo de, *Mathematical Modelling and Control of nonlinear Dynamics System*, Campinas, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2003. 158 p. Tese Doutorado.

This work has as objective the modeling and simulation of dynamic systems, with emphasis in the study and controllers of position of joints robotics implementation. Initially the study of the constituent elements of a joint robotics is accomplished, such as: DC motor, inertia and reducers. To go on with this study will be necessary the definition of the control strategy, including the development of a generator of trajectories used inside as reference of a control involving these elements. The final validation of the developed control algorithms is accomplished through a supported experimental.

### Key Words

- Modeling, Controllers

## **Índice**

<b>Lista de Figuras</b>	xv
<b>Lista de Tabelas</b>	xxi
<b>Capítulo 1</b> Introdução	1
<b>Capítulo 2</b> Estrutura de controle de um Robô industrial	5
2.1 Utilização de Robôs Industriais	6
2.2 Programação de Robôs Industriais	8
2.2.1 Programação por aprendizagem	10
2.2.2 Memorização de tarefas	11
2.2.3 Programação através de linguagens	12
2.2.4 Níveis de programação	12
2.2.5 Vantagens da utilização de programação através de linguagens	13
2.3 Modelagem de Robôs Industriais	13
2.3.1 Descrição de um Robô Industrial	13
2.3.2 Sistemas de Referência	15
2.3.3 Transformação de coordenadas	16
2.4 Modelagem Cinemática	21
2.4.1 Modelo cinemático Direto	21
2.4.2 Modelo cinemático inverso	22
2.4.3 Existência da solução do modelo geométrico inverso	23

2.4.4 Matriz Jacobiana	23
2.4.5 Inversão da Matriz Jacobiana	24
2.5 – Estrutura de Controle de um robô Industrial baseado no Modelo Cinemático	24
2.6 Conclusão	26
<b>Capítulo 3 Modelagem Dinâmica de uma Junta Robótica</b>	<b>27</b>
3.1 Modelagem de Sistemas Mecânicos	27
3.1.1 Equação Elétrica	29
3.1.2 Equação de Acoplamento Eletro-mecânico	29
3.1.3 Equação Mecânica	29
3.2 Diagrama de Blocos	30
3.3 Transformadas de Laplace	31
3.4 Redutor	35
3.5 Dinâmica do Sistema	38
3.6 Conclusões	40
<b>Capítulo 4 Modelagem Dinâmica de um Robô Industrial</b>	<b>42</b>
4.1 Modelo Dinâmico de um Robô Manipulador - Caso Não Linear	42
4.2 Exemplos de Aplicação - Pêndulo Duplo - Dinâmica e Controle	46
4.3 Conclusões	50
<b>Capítulo 5 Projeto de um Controlador PID para juntas Robóticas a partir do Modelo Dinâmico do Sistema</b>	<b>51</b>
5.1 - Projeto de um Controlador de posição PID	52
5.2 - Determinação dos ganhos de um regulador PID	53
5.2.1 – Método de Ziegler-Nichols	53
5.2.2 – Método de Chien-Hrones-Reswick (CHR)	55
5.2.3 – Exemplos de validação - Cálculo dos ganhos do controlador PID	57
5.4 Conclusão	62
<b>Capítulo 6 Modelagem de um controlador não linear</b>	<b>63</b>

6.1	Revisão de Conceitos abordando Sistemas não lineares	63
6.2	Sistemas de controle não lineares	65
6.2.1	Abordagens utilizadas na análise e projeto de sistemas de controle não lineares	65
6.2.2	Não linearidades usuais mais freqüentemente encontradas em servo sistemas	66
6.3	Modelos de sistemas físicos	68
6.3.1	Comportamento de um sistema não linear	70
6.3.2	Métodos de estudo de um sistema não linear	71
6.3.3	Descrição matemática	71
6.4	Modelo dinâmico de um manipulador	72
6.4.1	Descrição Dinâmica de uma junta robótica	73
6.4.2	Função Descritiva	75
6.4.3	Modelo Dinâmico de uma Junta Robótica considerando aspectos não linearidades	77
6.4.4	Simulação Numérica do Sistema Proposto	80
6.4.5	Análise dos Resultados	84
6.5	Conclusão	85
<b>Capítulo 7</b> Proposição de Bancada Experimental para Validação e Testes de Junta Robóticas		87
7.1	Plataforma Experimental para implementação de um controlador de posição	87
7.1.1	Motor – Redutor – Inércia	88
7.1.2	Motor acoplado a uma Inércia Variável	89
7.2	Considerações sobre a Maquete Experimental utilizada	92
7.2.1	Estudo do Acoplamento Inercial	92
7.2.2	Implementação de Hardware Tratamento de Informações e Controle	93
7.2.3	Implementação de Sistema de Supervisão e Controle	93
7.2.4	Modelagem e Identificação de Parâmetros	94
7.3	Projeto e Implementação do Controlador de Posição	94

7.3.1 O Sistema de Desenvolvimento ALTERA™	94
7.3.2 Proposta de Desenvolvimento do Controlador de Posição	96
7.3.3 Descrição dos Sinais de Entrada/Saída da Interface Implementada	97
7.3.3.1 TRAJETÓRIA	97
7.3.3.2 ENCODER	97
7.3.3.3 CLOCK	98
7.3.4 Exemplo de Aplicação	98
7.4 Ambiente de Simulação	99
7.5 Protótipo experimental implementado	100
7.5.1 Módulo de Aquisição e Tratamento de Informações	100
7.5.2 Módulo de Controle de Posição	101
7.6 Conclusão	102
<b>Capítulo 8</b> Conclusões e perspectivas futuras	104
<b>Referências Bibliográficas</b>	106
<b>Anexo I</b> Descrição dos Elementos Constituintes da Plataforma Experimental	109
I 1 Bancada Experimental	109
I 2 Sistema de Acionamento - Motor cc	111
I 3 Sistema de Transmissão e Acoplamentos	112
I 4 Transdutores de Posição	112
I 5 Inércia Variável	112
I 6 Sistema de Aquisição e Controle	113
<b>Anexo II</b> Estrutura de Supervisão e Controle de um Manipulador Robótico	114
II 1 Estrutura do Sistema de Supervisão e Controle	114
II 2 Descrição Detalhada dos principais elementos	115
II 2.1 Sensor de Posição e final de curso das juntas	116
II 2.2 Programa de Geração de Trajetórias	116

II 3 Hardware de Acionamento e Controle das Juntas	116
II 3.1 Módulo de potência	116
II 3.2 Módulo de controle:	117
II 4 Software de Supervisão e Controle das Juntas	117
II 4.1 - Módulo de Aprendizagem:	117
II 4.2 - Supervisão e Controle de Movimentos	118
<b>Anexo III Implementação do Sistema de Supervisão e Controle</b>	119
III 1 Descrição Funcional do software implementado	119
III 1.1 Interface com o usuário	119
III 1.2 Programa de Aquisição de Informações do Encoder	119
III 1.3 ALTERA MAX+PLUS II <sup>TM</sup>	120
III 2 Descrição do software implementado em linguagem DELPHI <sup>TM</sup>	121
III 2.1 Descrição da Interface Principal	121
III 2.2 Descrição da Interface Principal	121
<b>Anexo IV Implementação de Hardware de Aquisição, Tratamento e Controle utilizando Lógica Reprogramável</b>	123
IV 1 Introdução	123
IV 2 Princípio de Funcionamento de um Encoder Incremental	124
IV 3 Implementação de Circuito de Tratamento de Informações do Encoder e aquisição de informações através de um PC	125
IV 3.1 Relação de Módulos Implementados	125
IV 3.2 Circuito Completo de Tratamento de Informações do <i>Encoder Incremental</i>	126
IV 4 Circuito de Tratamento de Informações do Encoder	126
IV 4.1 Módulo Freq-Sent: Circuito de Detecção de Frequência e Sentido de Rotação de um Motor	127
IV 4.2 Relação de Entradas e Saídas	127
IV 4.3 Relação de Módulos Implementados	128
IV 4.4 Módulo Codeur: Circuito de Discriminador de Sentido de	



Rotação de um Motor	128
IV 5 Módulo Chave: Circuito de Gerador de Frequências para Aquisição	134
IV 5.1 Relação de Entradas e Saídas	136
IV 5.2 Módulo Chave: Circuito de Chaveamento de Frequências	136
IV 5.3 Circuito de Chaveamento	137
IV 5.4 Circuito de Chaveamento	138
IV 6 Circuito de Aquisição com Micro-Computador	138
IV 6.1 Relação de Entradas/Saídas	138
IV 6.2 Circuito com Contadores Absolutos	140
<b>Anexo V Implementação de Circuito de Acionamento e Controle de um motor de Corrente</b>	
Contínua	141
V 1 Introdução	141
V 2 Revisão de Conceitos	141
V 2.1 Acionamento de Motores	141
V 2.2 Controlador através de PWM	143
V 3 Implementação do Sistema de Comunicação e Controle	143
V 3.1 Implementação do Controlador utilizando uma ponte H	143
V 3.2 Módulo Decpcm	145
V 3.2.1 Descrição dos Estados	146
V 3.2.2 Tabelas de Avaliação do Estado Dois	147
V 3.2.3 Programa em VHDL Implementado	148
V 3.2.4 Circuito montado para a Simulação do Módulo Decpcm	150
V 3.2.5 Comentários	150
V 3.3 Módulo Duty	151
V 3.3.1 Principio de Funcionamento	151
V 3.3.2 Descrição dos Estados	151
V 3.3.3 Tabelas de Avaliação na Transição Zero -> Um	152
V 3.3.4 Programa Computacional em VHDL Implementado	152
V 3.3.5 Circuito implementado para a Simulação do Módulo Duty	154
V 3.3.6 Comentários	154

V 4 Controle de velocidade e aceleração de um Motor de corrente contínua através de PWM	155
V5 Conclusões	156
<b>Anexo VI</b> Cálculo do momento de inércia	157

## Lista de Figuras

Figura 2.1: Espaço de Coordenadas de um Robô.	7
Figura 2.2: Vetores de posição e de orientação de um robô.	8
Figura 2.3: Programação por aprendizagem.	11
Figura 2.4: Ilustração de um robô industrial Manutec.	14
Figura 2.5: Sistema de Referência.	16
Figura 2.6: Transformação Direta de Coordenadas.	17
Figura 2.7: Modelo geométrico de posição do elemento terminal.	19
Figura 2.8: Geração de movimentos a partir do modelo cinemático.	20
Figura 2.9: O Modelo geométrico descreve a posição e orientação da garra em função das variáveis de juntas.	21
Figura 2.10: Utilização do modelo geométrico direto e da matriz Jacobiana inversa para determinar uma configuração $\theta^*$ correspondente a uma situação desejada $x^*$ .	22
Figura 2.13: Estrutura de controle de um robô baseada no Modelo Cinemático.	25
Figura 3.1: Modelo do sistema mecânico a ser estudado.	28
Figura 3.2: Representação Esquemática e Diagrama de blocos do Motor CC controlador pela corrente de armadura.	28

Figura 3.3: Diagrama de Blocos para a equação 3.4.	30
Figura 3.4: Diagrama de Blocos para a equação 3.5.	30
Figura 3.5: Diagrama de Blocos para a equação 3.6.	31
Figura 3.6: Diagrama de Blocos para a completa.	31
Figura 3.7: Diagrama de Blocos para a equação 3.9.	32
Figura 3.8: Diagrama de Blocos para a equação 3.10.	32
Figura 3.9: Diagrama de Blocos para a equação 3.13.	33
Figura 3.10: Diagrama de blocos para o modelo completo das funções de transferência.	33
Figura 3.11: Diagrama de blocos para a equação 3.22 na situação em que $T_r(s)=0$ .	34
Figura 3.12: Gráfico dos pólos.	35
Figura 3.13: Sistema de engrenagens.	36
Figura 3.14: (a) inércia+ sistema de engrenagens, (b) rigidez + sistema de engrenagens, (c) atrito viscoso + sistema de engrenagens.	37
Figura 3.15: Representação da carga no eixo do redutor.	38
Figura 3.16: Diagrama de blocos para a equação 3.33.	38
Figura 3.17: Diagrama de blocos para a equação do sistema.	40

Figura 4.1 Manipulador com três graus de liberdade.	44
Figura 4.2 Representação da carga no motor	45
Figura 4.3. Diagrama de blocos para as equações não lineares	46
Figura 4.4. Pêndulo Duplo	47
Figura 4.5. Dinâmica do Pêndulo Duplo	48
Figura 4.6. Pêndulo Duplo – Controle (os dois últimos gráficos representam o erro de trajetória)	49
Figura 4.7. Pêndulo Duplo - Controle com Massa Estimada	50
Figura 5.1: Esquema de um sistema para o cálculo do PID.	53
Figura 5.2 – Método de CHR - obtenção da resposta temporal a uma entrada degrau.	55
Figura 5.3: Curva resposta temporal utilizada no método CHR	56
Figura 5.4: Configuração Inercial 1 e parâmetros utilizados.	58
Figura 5.5: Resposta temporal a uma excitação degrau utilizando controlador PID (configuração1).	59
Figura 5.6: Configuração Inercial 2 e parâmetros utilizados.	59
Figura 5.7: Resposta temporal a uma excitação degrau utilizando controlador PID (configuração2).	60

Figura V 3: Princípio de funcionamento de um controlador através de PWM.	143
Figura V 4: Controlador	145
Figura V 5: Máquina de Estado implementada no módulo Decpcm	147
Figura V 6 – Circuito para a simulação do módulo Decpcm.	150
Figura V 7: Resultado da simulação do módulo Decpcm.	150
Figura V 8: Máquina de Estado do módulo Duty.	152
Figura V 9: Circuito para a simulação do módulo Duty.	154
Figura V 10: Simulação do módulo Decpcm.	154
Figura V 11: Exemplo 1: controle de um motor cc através de PWM	155
Figura V 12: Exemplo 2: controle de um motor cc através de PWM	156
Figura VI 1: Esquema utilizado na dedução do momento de inércia	157

## **Lista de tabelas**

Tabela 5.1: Ganhos do compensador PID.	54
Tabela 5.2: Escolha do tipo de Controlador utilizando o método de CHR.	57
Tabela 5.3: Cálculo dos Ganhos do Controlador utilizando o método de CHR.	57
Tabela 5.4: Configurações, parâmetros utilizados, ganhos do Controlador PID.	58

# Capítulo 1

## Introdução

Na realização de uma tarefa, o robô realiza um ciclo de operações seqüenciais, que pode ser alterado devido a fatores externos. A maioria dos robôs utilizados no meio industrial possui juntas rotativas, motivo que este trabalho será dirigido a esse tipo de robô, mas poderá ser facilmente estendido a outros tipos de configurações de robôs.

Para que a utilização de robôs na indústria seja viável, é necessário que a sua programação seja simples e de fácil modificação, permitindo ao usuário rapidez e flexibilidade.

Durante um programa de cooperação científica-tecnológica, envolvendo a Faculdade de Engenharia Mecânica da UNICAMP, CENPES-PETROBRÁS e Instituto Tecnológico de Geesthacht da Alemanha, na área de intervenções submarinas automatizadas utilizando programação off-line (realização de software para geração de trajetórias e comandos a partir do modelo geométrico do manipulador, ambiente de operação e obstáculos). Foram desenvolvidos trabalhos de pesquisa, inicialmente utilizando um robô industrial – MANUTEC r3 da SIEMENS, que apresentava grande precisão de posicionamento, mas poucos recursos de programação off-line, tendo em vista que a estrutura de controle existente por ser completamente fechada e, posteriormente, o manipulador robótico KRAFT-GRIPS, desenvolvido especialmente para realização de operações submarinas por aprendizagem, tipo Master-Slave (Mestre-Escravo), assim não necessitando de um controlador sofisticado (controlador atualmente existente é do tipo

P.D. - Proporcional - Derivativo), mas de um controlador com uma estrutura completamente aberta.

Durante a realização desses trabalhos de pesquisa, constatou-se que a intervenção de tarefas submarinas automatizadas exige a utilização de manipuladores robóticos mais precisos. Esses manipuladores são desenvolvidos para trabalhar em condições adversas e operações de tele-presença com a ajuda de um operador no mundo mestre-escravo (master-slave), operações que não exigem uma grande precisão do sistema de controle. Entretanto, intervenções submarinas em águas profundas, necessitam cada vez mais de tarefas completamente automatizadas, exigindo assim, o desenvolvimento de um sistema de controle mais apropriado para o manipulador.

Assim sendo, uma continuidade desse projeto de pesquisa, foi direcionada neste programa de doutoramento através da estruturação de um sistema de controle de um manipulador robótico, em que será implementado um novo sistema de controle de posição para a realização de tarefas completamente automatizadas a partir do conhecimento do ambiente de atuação e tipo de tarefa a ser realizada (Sá, 1996).

Com esta finalidade, aproveitaremos a arquitetura “aberta” do sistema de controle existente, para investigarmos a utilização de diferentes tipos de estrutura de controle.

A metodologia de trabalho consistiu num estudo inicial de modelagem dinâmica de um robô, estudando a partir de simulações numéricas, verificando o comportamento do sistema em presença de perturbações, usando primeiramente um controlador clássico do tipo PID (Proporcional, Integral e Derivativo) e, posteriormente, controladores não lineares utilizando componentes lógicos reprogramáveis.

A implementação final do trabalho será realizada através da implementação de uma bancada experimental representando uma junta robótica completa (acionamento, redutor, inércia, controlador), permitindo assim, a comparação com os resultados obtidos durante a simulação e o desenvolvimento de um controlador de posição.

A utilização de dispositivos robóticos vem aumentando a cada dia no meio industrial acarretando um aumento de produtividade e entrega de produtos de qualidade uniforme. Atualmente, muitos dos produtos manufaturados foram substituídos por produtos industrializados, levando em conta a utilização de máquinas com um propósito de realizar determinadas tarefas com maior precisão. Os robôs de última geração utilizados na indústria podem ser considerados rápidos e precisos, sendo que os algoritmos de controle existentes são perfeitamente aceitáveis dentro das exigências requeridas.

Entretanto, na maior parte dos robôs existentes no mercado, não foi desenvolvido para a realização de tarefas em meio adverso (usinas nucleares, submarinas, etc.), sendo que tradicionalmente, foram desenvolvidos sistemas tipo “master-slave” teleoperados pelo homem. Ao mesmo tempo, o alto custo de uma adaptação dos robôs tradicionais utilizados na indústria, comparados com o mercado, exigem o desenvolvimento de trabalhos de pesquisa no sentido de adaptarmos os manipuladores existentes para operações automatizadas.

O desenvolvimento de microprocessadores cada vez mais versáteis, rápidos e precisos tem viabilizado a utilização de técnicas avançadas de controle de movimentos de manipuladores. Em muitos casos, torna-se necessária à utilização de estruturas a microprocessadores em forma hierárquica para conseguir realizar um grande volume de cálculo que estas estratégias necessitam, permitindo um processamento paralelo a partir da geração das referências para cada junta.

Com o propósito de estudar o controle dessas juntas, neste trabalho de doutoramento é proposta a implementação de um sistema de supervisão e controle de um manipulador baseado na Modelagem Dinâmica. Com este objetivo utilizaremos, inicialmente, um controlador clássico do tipo PID (Proporcional, Integral e Derivativo), implementado a partir de simulações para verificar o comportamento do sistema em presença de perturbações, com um estudo posterior, em que esta estrutura de controle é substituída por controladores não-lineares utilizando componentes lógicos reprogramáveis.

Para esses objetivos, este trabalho de Doutorado foi subdividido em sete capítulos, descritos a seguir:

No capítulo 2 é realizado um estudo de controle de posição de um robô industrial com juntas rotacionais, baseado no modelo cinemático direto e inverso, sendo realizado uma breve análise dos métodos utilizados para a programação de robôs mostrando suas vantagens e desvantagens.

No capítulo 3 é abordada a estrutura de controle de uma junta robótica com ênfase na modelagem dinâmica, considerando a dinâmica completa da junta com inércia e acoplamento.

No capítulo 4 é visto a estrutura de controle de um robô industrial com ênfase na modelagem dinâmica, considerando a dinâmica das juntas robóticas acopladas e sistema de acionamento.

No capítulo 5 é apresentado o estudo dinâmico do Controlador e implementação de um controlador PID – Proporcional-Integral-Derivativo com parâmetros variáveis.

No capítulo 6 são apresentados os aspectos conceituais, abordando controladores não lineares, requisito básico para a proposta de implementação experimental utilizando componentes lógicos reprogramáveis.

No capítulo 7 é apresentada a infra-estrutura experimental (hardware dedicado e software), onde foi implementado no Laboratório de Automação Integrada e Robótica da UNICAMP duas bancadas, simulando uma junta robótica, que permitiram a validação e testes do sistema de aquisição, processamento e controle implementado.

Finalizando, o capítulo 8 apresenta uma síntese dos principais resultados obtidos no desenvolvimento da pesquisa, conclusões e perspectivas futuras.

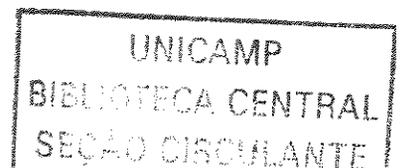
## Capítulo 2

### Estrutura de controle de um robô industrial

Um robô industrial pode ser definido como um sistema mecânico articulado que tem como objetivo principal executar operações pré-definidas. Isto é realizado através de um Supervisor de Controle que deverá especificar o que o manipulador deverá fazer para que o mesmo possa realizar as tarefas especificadas. Normalmente ele é constituído de seis graus de liberdade e o posicionamento de sua ferramenta de trabalho é especificado através do controle de modo apropriado de suas variáveis articulares.

Desta maneira, sua trajetória é definida através de um conjunto de ângulos associados ao movimento angular de cada grau de liberdade do robô, que utilizando um algoritmo de interpolação, servirá como sinal de referência para o controlador de posição de cada junta robótica que realizará uma comparação com os sinais provenientes dos transdutores de posição das juntas.

Diversas aplicações industriais exigem que o robô trabalhe de acordo com a posição e orientação do seu elemento terminal em relação ao sistema de coordenadas de trabalho, como por exemplo: um robô trabalhando em conjunto com uma máquina de comando numérico, numa célula automatizada com outros robôs, ou ainda quando o mesmo é dotado de um sistema de visão. Neste último caso, a interpretação das imagens se efetuará em relação ao sistema de coordenadas de trabalho (em duas ou três dimensões) e as informações extraídas das mesmas, serão transmitidas ao Sistema de Supervisão após tratamento apropriado.



O Supervisor de Controle é responsável pela geração dos sinais de referência individuais ao longo do tempo, para cada junta do robô. Através de uma malha de controle de posição independente para cada junta, nos quais os sinais são comparados com os valores atuais (obtidos através dos sensores de posição articulares), faz com que a configuração de um robô seja controlada a partir de um valor desejado, independente do movimento realizado e da carga transportada pelo robô.

Entretanto, 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 num erro que aumenta com a sua velocidade de operação. Conseqüentemente, a implementação de um controlador de posição para um robô industrial exige o conhecimento da precisão cinemática do movimento do manipulador.

Para estabelecermos estratégias de controle de posição de juntas robóticas eficientes e precisas (erro próximo de zero), o movimento do robô é descrito através de equações diferenciais considerando-se na sua arquitetura de construção, a massa dos diferentes elementos, as inércias e o tensor de inércia relacionados com a carga transportada, como também a modelagem completa de seu sistema de acionamento (motor-redutor) (Paul, 1981).

Neste capítulo serão abordados os aspectos conceituais que envolvem a modelagem e controle de robôs industriais, a partir de uma revisão dos principais conceitos matemáticos necessários para utilização de um supervisor de controle de um robô industrial.

## **2.1 Utilização de Robôs Industriais**

Os robôs industriais são equipamentos multifuncionais reprogramáveis com grande flexibilidade de operação. Atualmente, a programação de tarefas é realizada através de uma “caixa de aprendizagem” que é utilizada para conduzir o robô através das posições críticas do ciclo de operação. Este tipo de programação de tarefas possui alguns inconvenientes, como por exemplo: utilizam o robô no período de programação e não permitem um controle mais preciso sobre a trajetória da garra ou ferramenta.

Já a programação de tarefas off-line, não apresenta o inconveniente acima por ser realizada em computadores, necessitando apenas de um modelo matemático. Este modelo contém informações sobre a geometria do sistema (modelo cinemático).

Normalmente a programação de tarefas de robôs é realizada no espaço das juntas, não necessitando de um modelo geométrico e a trajetória angular de mesma natureza dos sinais provenientes do transdutor de posição servirá como sinal de referência para o controlador de cada junta robótica (figura 2.1). Entretanto, a realização de algumas tarefas em relação a um sistema de referência colocado na ferramenta (espaço cartesiano) exige o conhecimento completo do modelo geométrico e torna necessária a utilização de uma transformação de coordenadas, tendo em vista que o sinal de referência correspondente à trajetória, necessária para o controle das juntas, deverá ser angular.

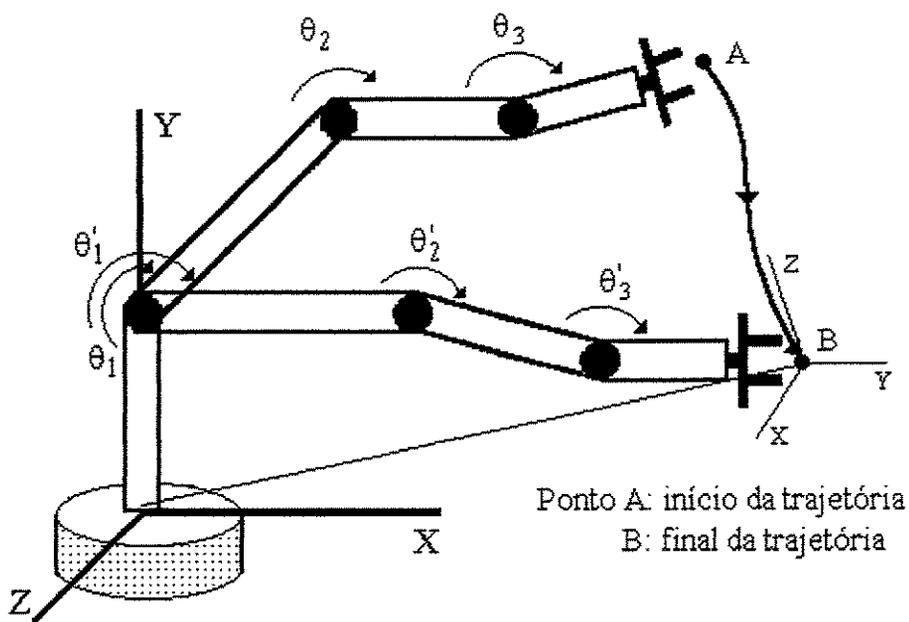


Figura 2.1: Espaço de Coordenadas de um Robô.

Num robô industrial, cada grau de liberdade está associado a um sistema de coordenada. Mais especificamente, o modelo geométrico é aquele que expressa a posição e orientação da garra em relação a um sistema de coordenadas solidário à base do robô, em função de suas coordenadas generalizadas (angulares no caso de juntas rotacionais).

Esta relação pode ser expressa matematicamente a partir de uma matriz que relaciona o sistema de coordenadas da base com o sistema de coordenadas do último elemento. Esta matriz é chamada de matriz de passagem homogênea do robô.

Como mencionado anteriormente, a tarefa de um robô é especificada em termos de coordenadas cartesianas,  $x$ . Essas consistem na posição, descrita por um vetor posição,  $p$ , e um vetor orientação, descritos por um vetor unitário de aproximação  $a$  e um vetor unitário de deslizamento  $s$ . Todos esses vetores são definidos em relação ao sistema de coordenadas da base. Por conveniência, um vetor unitário normal é definido como  $n = s \times a$ , onde  $\times$  denota o produto vetorial, figura 2.2.

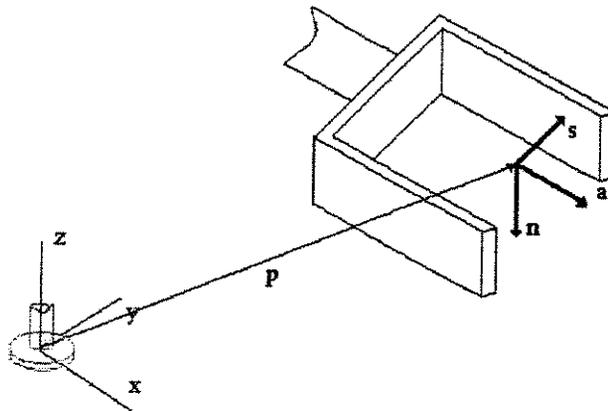


Figura 2.2: Vetores de posição e de orientação de um robô.

## 2.2 Programação de Robôs Industriais

Os métodos de programação mais freqüentemente utilizados em robôs industriais são:

- Aprendizagem ponto-a-ponto:

Este tipo de programação é caracterizado por uma coleção de pontos para o robô. Então construímos a série e guardamos na memória do controlador. Quando rodamos a série, o braço do robô vai se mover pelos vários pontos, de acordo com a ordem dos passos na série.

i) Movimento angular: neste método são gravados pontos de referência fornecidos pelos transdutores de posição localizados em cada junta e a partir desses

pontos são geradas trajetórias (angulares) através da utilização de algoritmos de interpolação. A simplicidade deste método não exige o conhecimento do modelo geométrico do robô;

*ii)* Movimento cartesiano: idêntico ao procedimento descrito anteriormente, mas a obtenção dos pontos de referência é realizada utilizando o modelo geométrico do manipulador;

- Programação off-line: através de um software para visualização gráfica do modelo geométrico de robôs, devem ser obtidos pontos de passagem correspondentes à trajetória do robô (expressos em coordenadas angulares). Esses pontos de passagem poderão ser obtidos do movimento angular de cada junta ou do modelo geométrico. A partir de um conjunto de pontos correspondentes à trajetória a ser realizada pelo robô, torna-se possível a implementação de algoritmos off-line para interpolação e filtragem, levando-se em consideração aspectos dinâmicos e testes de colisão.

- Programação on-line: a partir do conhecimento do modelo geométrico e das características da trajetória desejada (posição final, velocidade e forma do caminho), pode-se implementar algoritmos para modelagem cinemática inversa e controle de posição.

Os métodos descritos anteriormente estão associados com o tipo de aplicação requerida. Isto permitirá:

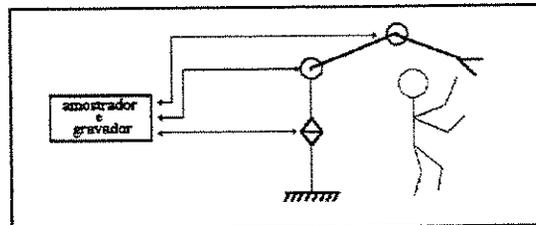
- a realização de tarefas mais precisas e complexas;
- a necessidade da utilização de posições determinadas analiticamente, ou informações provenientes de sensores de percepção externa;
- o uso de manipuladores em ambientes adversos à presença do homem;
- a necessidade do aumento do tempo útil de trabalho, pois durante a programação da

tarifa pelo método de aprendizagem, o manipulador é utilizado, diminuindo o seu tempo útil.

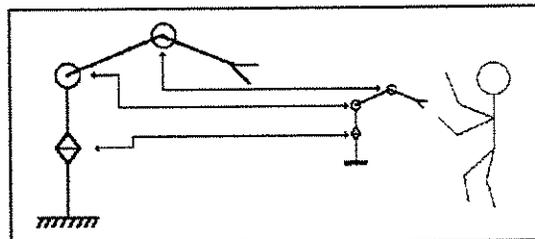
### **2.2.1 Programação por aprendizagem**

Programar visa o estabelecimento de uma seqüência de operações a serem executadas pelo robô. A programação das tarefas pode ser realizada através de uma programação por aprendizagem ou a partir de uma linguagem de programação de computadores. A programação por aprendizagem pode ser realizada pelos seguintes métodos:

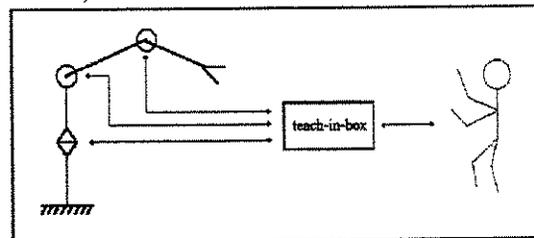
- **Aprendizagem direta:** nesse método, figura 2.3a, o operador guia fisicamente o robô por seu órgão terminal. Enquanto isto, os sensores de posição de cada junta do robô são utilizados para memorizar os pontos importantes da tarefa a ser executada. Este método é indicado para robôs de estrutura mecânica leve, de movimentos reversíveis e para a utilização em ambientes não “hostis”.
- **Aprendizagem por simulação física:** nesse método, figura 2.3b, o operador guia um simulador físico, que tem a geometria e os sensores idênticos ao robô original. Uma vez memorizada a tarefa, esta é transferida para o sistema de controle do robô. Este método é indicado para robôs de grande porte ou com estruturas mecânicas não reversíveis, bem como a programação em ambientes que exijam distância. Este método necessita de um modelo preciso e conhecido.
- **Aprendizagem por telecomando:** nesse método, figura 2.3c, um dispositivo de telecomando (teach pendant ou teach-in box) é utilizado para mover cada junta do robô isoladamente ou fornecer a posição e orientação da garra. Este método é adequado para qualquer tipo de robô e resulta mais barato que o método anterior, caso o telecomando seja feito a partir de um dispositivo de programação, que permita a atuação sobre cada ligação independentemente.



a) o operador guia o robô diretamente.



b) utilizando um simulador físico.



c) por telecomando.

Figura 2.3: Programação por aprendizagem.

### 2.2.2 Memorização de tarefas

A memorização de tarefas durante a fase de aprendizagem pode ser feita de duas maneiras:

- Programação ponto a ponto: são gravados apenas os pontos essenciais da trajetória do órgão terminal, de modo que o movimento entre dois pontos consecutivos gravados fica determinado pelos algoritmos de controle utilizados nos servo-mecanismos das diversas ligações. Esse método é adequado para aplicações que não requeiram um controle preciso da trajetória e da velocidade intermediários aos pontos essenciais da tarefa.

- Programação por caminhos contínuos: os pontos da trajetória, na aprendizagem, são amostrados com uma taxa elevada e armazenados na memória. Esses pontos podem ser amostrados no controle segundo uma taxa programada, conduzindo o robô, continuamente por inércia ou por “arrasto”, de maneira precisa e em uma velocidade que é função da taxa de amostragem.

### 2.2.3 Programação através de linguagens

A programação usando linguagens pode ser considerada como o processo pelo qual os programas são desenvolvidos sem a necessidade do uso do robô propriamente dito, mas através da utilização de uma linguagem de programação de computador. Com o atual avanço na tecnologia de desenvolvimento de hardware e software, a programação de robôs por linguagens, está cada vez mais próxima da realidade industrial. Esses avanços incluem uma maior sofisticação dos controladores, melhor precisão de posicionamento e incremento no número e tipos de sensores.

### 2.2.4 Níveis de programação

As linguagens de programação de robôs podem ser classificadas nos seguintes níveis:

- Nível de junta: as linguagens classificadas neste nível requerem a programação individual de cada junta do robô para que uma dada posição seja alcançada.
- Nível de manipulador: Nessa forma é necessário apenas fornecer a posição e orientação do órgão terminal e o sistema se encarrega de obter, através do modelo geométrico inverso do robô, as posições de cada junta.
- Nível de objeto: Nesse tipo são necessárias apenas as especificações relativas ao posicionamento de objetos no interior do volume de trabalho do robô. Deste modo, é necessária a existência de um modelo matemático que represente o ambiente de trabalho no qual o robô se encontra.

- Nível de objetivo: Neste caso a tarefa não é realmente descrita, mas definida como, por exemplo: montar as peças A, B e C. Desta forma, é necessário, além do conhecimento do modelo do ambiente, um conjunto de dados relativos a uma determinada tarefa.

### **2.2.5 Vantagens da utilização de programação através de linguagens**

Entre as vantagens da utilização da programação através de linguagens é possível citar:

- Redução do tempo em que o robô está fora da linha de produção, pois ele pode continuar operando enquanto a sua próxima tarefa está sendo programada;
- O programador não necessita entrar em contato com o ambiente de trabalho;
- Integração com sistemas CAD-CAM permitindo uma maior integração entre as fases de projeto e de produção e, conseqüentemente, reduzindo o tempo do processo de produção;
- Simplificação da programação de tarefas com alto grau de complexidade. As linguagens de programação facilitam a elaboração de tarefas complexas pois, possuem formas analíticas de geração de trajetórias e análise de dados provenientes de sensores externos, além de várias estruturas de controle.
- Segurança na geração de trajetórias. As linguagens podem internamente gerar um modelo geométrico do ambiente de trabalho do robô e, através de algoritmos de detecção de colisão, produzir trajetórias seguras e simulá-las por software antes da execução final.

## **2.3 Modelagem de Robôs Industriais**

### **2.3.1 Descrição de um Robô Industrial**

Um manipulador mecânico consiste de *links*, conectados por juntas prismáticas ou rotacionais. Cada par junta-link constitui um grau de liberdade. Assim, para um manipulador com

n graus de liberdade, temos n pares *juntas-links*, onde o primeiro link é a base de sustentação do robô (sistema de coordenadas inerciais fixado) e no seu último link é incorporada a sua ferramenta de trabalho.

O conhecimento completo das variáveis articulares de um robô  $\theta_i$ , determina o posicionamento de sua ferramenta no sistema de coordenadas de trabalho. De um modo geral, os três primeiros graus de liberdade de um robô são responsáveis pelo posicionamento de sua ferramenta no espaço de tarefas e os restantes, pela sua orientação. A figura 2.4 mostra um robô industrial com 6 graus de liberdade.

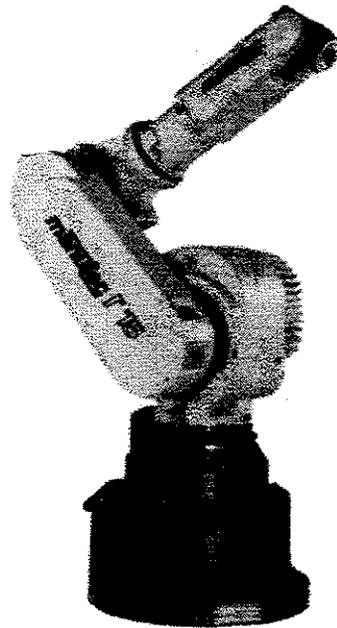


Figura 2.4: Ilustração de um robô industrial Manutec.

A maioria das aplicações industriais, a programação de tarefas de robôs, é realizada por aprendizagem, consistindo no movimento individual de cada junta. Assim sendo, a programação de trajetórias de um robô torna-se muito fácil, não necessitando de um conhecimento do modelo, sendo a fase de aprendizagem basicamente uma operação de armazenamento de uma seqüência de incrementos necessários para que o conjunto de variáveis articulares determine um posicionamento final  $X_i$ , especificado a partir de um perfil de trajetórias fornecido (robô controlado a partir do sistema de coordenadas de juntas).

Como um robô é controlado através de suas variáveis articulares, a realização do controle de posição em relação ao sistema de coordenadas cartesianas implicará no desenvolvimento de metodologias para transformação de coordenadas. A transformação de coordenadas articulares para cartesianas é normalmente realizada em tempo real, em que a partir do conjunto de variáveis articulares serão obtidas a posição e orientação da ferramenta.

### 2.3.2 Sistemas de Referência

Um Sistema Articular pode ser representado matematicamente através de  $n$  corpos móveis  $C_i$  ( $i = 1, 2, \dots, n$ ) e de um corpo  $C_0$  fixo, interligados por  $n$  articulações, formando uma estrutura em cadeia.

Para representar a situação relativa dos vários corpos da cadeia, é fixado a cada elemento  $C_i$  um referencial  $R_i$ . Podemos relacionar um determinado referencial  $R_{i+1}$  ( $O_{i+1}, x_{i+1}, y_{i+1}, z_{i+1}$ ) com o seu anterior  $R_i$  ( $O_i, x_i, y_i, z_i$ ), como também com o sistema de coordenadas da base (figura 2.5) através da equação 2.1, onde  $A_{i,i+1}$  representa as matrizes de transformação homogênea de rotação e  $L_i$  o vetor de translação de uma origem a outra, onde  $A_{i,i+1}$  é resultante do produto matricial global entre as diversas matrizes de transformações homogêneas relacionadas com rotações ou translações sucessivas das diferentes articulações (equação 2.2).

$$O_{i+1} = O_i + A_{i,i+1} * L_i \quad (2.1)$$

$$A_{i,i+1} = A_{1,2} \cdot A_{2,3} \cdot \dots \cdot A_{i,i} \quad (2.2)$$

onde

$$A_{i,i+1} = \begin{bmatrix} Nx_o & Sx_o & Ax_o \\ Ny_o & Sy_o & Ay_o \\ Nz_o & Sz_o & Az_o \end{bmatrix}$$

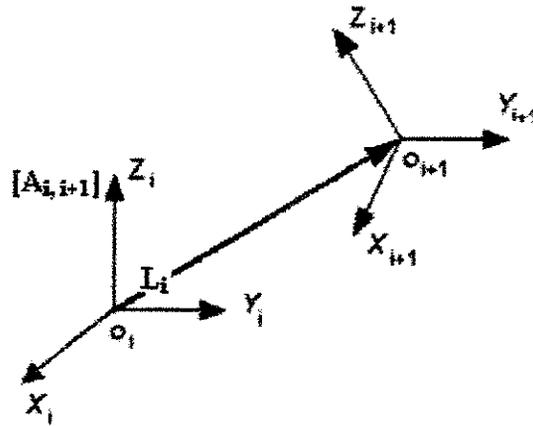


Figura 2.5: Sistema de Referência.

Qualquer rotação no espaço pode ser decomposta em um grupo de rotações elementares ao longo dos eixos X, Y e Z. A matriz de rotação elementar usada na equação de transformação é associada com a rotação elementar do referencial correspondente em relação ao seu anterior. Este procedimento matemático pode ser estendido para toda extensão do modelo. Assim sendo, a matriz de orientação pode ser obtida pela equação 2.2.

Conseqüentemente o posicionamento completo de um corpo rígido no espaço, poderá ser facilmente obtido através da equação 2.1 que fornece o seu vetor posição, sendo que a equação 2.2 representa a matriz de orientação associada, podendo ser expressa através de componentes angulares associadas às três direções de rotação correspondentes aos eixos de referência do sistema de coordenadas (por exemplo, *Roll, Pitch, Yaw* - RPY ou *quaternions*).

### 2.3.3 Transformação de coordenadas

Nas diversas aplicações industriais, um robô pode ser controlado e programado a partir do sistema de coordenadas associadas à sua ferramenta. É muito mais natural expressarmos o deslocamento absoluto do elemento terminal de um robô que considerarmos a variação de suas coordenadas articulares, embora a malha de controle de uma junta robótica seja estabelecida a partir da comparação de grandezas articulares, tornando-se necessária a realização de uma

transformação geométrica apropriada para o estabelecimento da correspondência entre as variáveis articulares  $\theta_i$  e as coordenadas absolutas do elemento terminal  $X_i$ .

A figura 2.6 apresenta um esquema descrevendo o problema de transformação direta de coordenadas para um robô com  $n$  graus de liberdade.

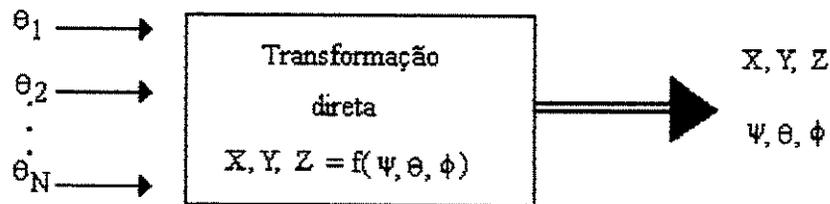


Figura 2.6: Transformação Direta de Coordenadas.

A operação que realiza a correspondência entre esses dois espaços é chamada de transformação de coordenadas. A transposição direta de coordenadas apresenta uma solução única, o mesmo não acontecendo com o problema inverso que, dependendo do tipo da cadeia cinemática, podem conduzir a soluções múltiplas.

A noção de modelo cinemático direto levanta o problema fundamental da robótica pois, os atuadores (motores, redutores, correias, cilindros hidráulicos, etc.) agem sobre as variáveis das juntas. Assim, é necessário que o modelo cinemático direto converta os valores destas variáveis nos valores de posição e orientação da garra no sistema de coordenadas do ambiente de trabalho do robô, para que este possa interagir com os outros objetos (operador, peças, máquinas, equipamentos, etc.).

Decorre disso, a idéia de localização do manipulador no espaço dos atuadores, no espaço das juntas e no espaço cartesiano, já que em cada um desses espaços pode-se, com um conjunto de variáveis distintas em significado e valor, definir completamente a localização do manipulador, ou de seu órgão terminal.

Portanto, o modelo cinemático direto mapeia a configuração do manipulador do espaço das juntas para o espaço cartesiano do sistema de coordenadas da base, ou seja, é uma transformação que leva de um vetor de posição das juntas (no exemplo planar, um vetor  $3 \times 1$ , composto por  $\theta_1$ ,  $\theta_2$  e  $\theta_3$ ) para um vetor posição e uma matriz de orientação da garra (no exemplo planar, pode-se pensar em um vetor  $3 \times 1$ , composto pelas coordenadas X e Y, e o ângulo de orientação final da garra).

Outra questão que surge é como podemos obter os valores das variáveis das juntas, necessárias para que a garra do robô esteja em uma localização desejada do espaço cartesiano da base, para que este possa realizar uma tarefa com sua garra, nesta posição.

Este mapeamento é feito pelo Modelo Cinemático Inverso, ou seja, uma transformação que leva uma localização no espaço cartesiano da base para valores das variáveis das juntas correspondentes.

Infelizmente, devido às não linearidades do modelo cinemático direto, não podemos simplesmente invertê-lo para obter o modelo cinemático inverso. Este é um problema mais difícil que deve ser solucionado caso a caso.

A figura 2.7 mostra o modelo geométrico que descreve a posição  $(x,y,z)^T$  e orientação (vetores  $\mathbf{n}$ ,  $\mathbf{s}$  e  $\mathbf{a}$ ) do elemento terminal em relação a um sistema de coordenadas solidário à base do robô.

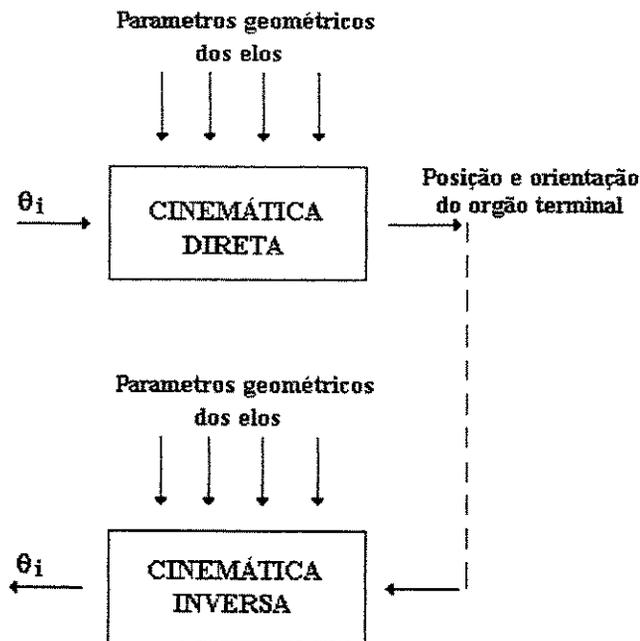


Figura 2.7: Modelo geométrico de posição e orientação do elemento terminal.

Embora a definição do modelo geométrico seja única, a maneira de obter a matriz de passagem homogênea do robô está associada ao sistema de referência utilizado. Para isso existem diversas maneiras e cada uma gera expressões diferentes que, embora equivalentes quantitativamente, podem diferir quanto ao número de operações aritméticas necessárias para fazer o cálculo numérico das mesmas. Tendo em vista a aplicação do modelo em sistemas de controle em tempo real, figura 2.8, é importante que se obtenha um modelo com o menor número possível de operações matemáticas.

Entretanto, a estratégia de controle nos robôs é exercida sobre seus atuadores, sendo que, o sistema de controle só segue referências ou movimentos desejados definidos neste espaço (espaço das juntas). Contudo, o operador define as tarefas ou movimentos de referência no espaço operacional. Vemos assim que os movimentos desejados e as leis de controle estão em espaços diferentes. A obtenção de referências correspondentes às tarefas definidas no espaço operacional é denominada coordenação de movimentos, como visto na figura 2.3. A coordenação é expressa matematicamente pela inversão do modelo geométrico. Existem dois métodos para a solução do problema da inversão do modelo geométrico :

- Métodos analíticos. Estes métodos não são gerais, isto é, a inversão analítica não é trivial e não há garantia de que seja possível fazê-la para um robô qualquer. Além disso, caso seja encontrada a solução ela pode apresentar soluções múltiplas (problema de redundâncias).
- Métodos numéricos. Estes métodos convergem para uma solução possível entre todas as existentes e com o atual desenvolvimento dos microcomputadores, a utilização destes métodos em tempo real é viável. Os dois métodos mais utilizados são: o método da linearização da matriz de passagem do robô e o método recursivo, que utiliza o cálculo do modelo geométrico e da matriz Jacobiana inversa.

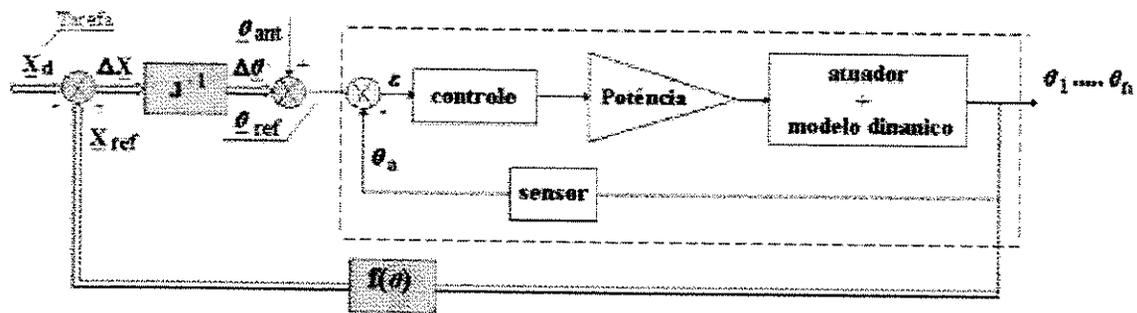


Figura 2.8: Geração de movimentos a partir do modelo cinemático.

Diferentemente das soluções analíticas, as soluções numéricas podem ser combinadas com estratégias de anti-colisão o que justifica a importância destes métodos. Entretanto, o uso de métodos numéricos pode ser um sucesso se, e somente se, um algoritmo eficiente for desenvolvido.

Diferentes algoritmos de geração de trajetórias utilizam o modelo cinemático inverso, obtido através de método numérico, e o estudo de métodos para a inversão da matriz Jacobiana (métodos de Gauss, de Greville e de Miss) que possam ser utilizados em tempo real e ao mesmo tempo possam garantir a convergência e estabilidade do sistema (Sá, 2000).

## 2.4 Modelagem Cinemática

O elemento terminal é acionado por atuadores ligados às juntas que necessitam de referências angulares. Deste modo torna-se necessária a determinação do conjunto de ângulos (espaço das juntas) correspondentes ao vetor  $\mathbf{x}$  (espaço cartesiano). Isto requer a inversão do modelo cinemático do robô (modelo cinemático inverso), que pode ser obtido analiticamente ou numericamente. A solução, caso exista, pode não ser única. Neste item, além da análise do modelo inverso, é também apresentada uma breve discussão sobre a matriz Jacobiana.

### 2.4.1 Modelo cinemático Direto

O modelo geométrico de um robô é definido como a função vetorial  $\mathbf{f}$ , que exprime um vetor  $\mathbf{x}$  (espaço cartesiano) em função do vetor  $\theta$  (coordenadas angulares):

$$\mathbf{x} = \mathbf{f}(\theta) \quad (2.3)$$

Como já visto, para se determinar o modelo geométrico é necessário apenas o cálculo da matriz de passagem homogênea que relaciona o deslocamento espacial do sistema de coordenadas da garra ao sistema de coordenadas de referência, figura 2.9.

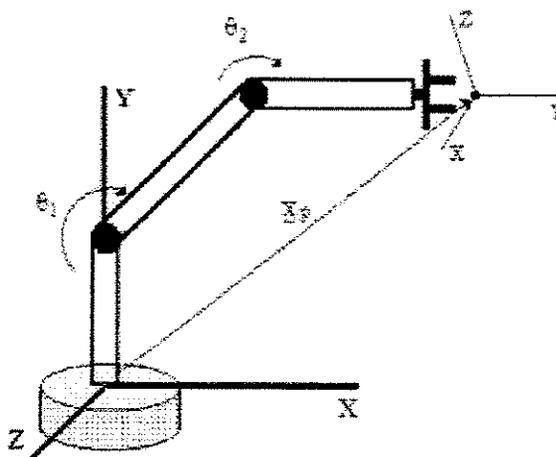


Figura 2.9: Modelo geométrico que descreve a posição e orientação da garra em função das variáveis de juntas.

O modelo geométrico é um modelo básico para a obtenção do modelo dinâmico, a implementação de estratégias de controle de trajetórias e a simulação cinemática.

#### 2.4.2 Modelo cinemático inverso

Como anteriormente dito, a coordenação de movimentos que consiste na obtenção de um movimento de referência (angular) para cada junta e, para um dado movimento de referência da garra (cartesiano), é expressa matematicamente pela inversão do modelo geométrico, isto é:

$$\theta = f(\theta)^{-1}(\mathbf{x}) \quad (2.4)$$

Em geral, a função  $f$  é não linear e composta de soma de produtos das coordenadas generalizadas das ligações de translação e de somas e produtos de senos e co-senos das coordenadas generalizadas das ligações de rotação. Por isso, a sua inversão é em geral não trivial.

Como  $f$  é não linear não se pode garantir a existência e/ou a unicidade de uma função inversa  $f^{-1}$ . No caso geral, só se pode determinar o número máximo de prováveis soluções.

Existem diversos métodos numéricos iterativos, entre eles o método recursivo, figura 2.10, que utiliza ao cálculo do modelo geométrico direto e da matriz Jacobiana inversa.

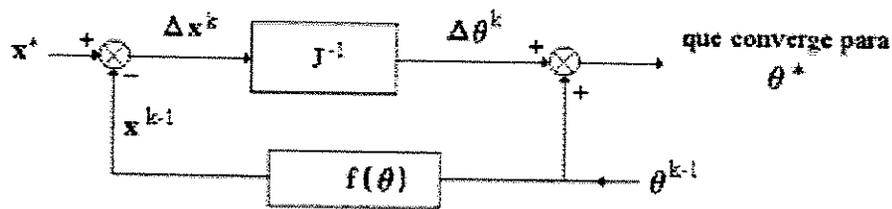


Figura 2.10: Utilização do modelo geométrico direto e da matriz Jacobiana inversa para determinar uma configuração  $\theta^*$  correspondente a uma situação desejada  $\mathbf{x}^*$ .

Neste trabalho optou-se pela utilização do método recursivo para a implementação de um software para a geração de trajetórias pois é de fácil implementação.

### 2.4.3 Existência da solução do modelo geométrico inverso

Para uma tarefa qualquer, definida no espaço operacional (cartesiano) de um robô, são necessários no mínimo seis graus de liberdade para a sua execução. Contudo se existirem obstáculos a se contornar, o número de graus de liberdade pode aumentar. Denotando-se  $m$  o número de graus de liberdade necessários à execução de uma tarefa, se o manipulador tem  $n$  graus de liberdade, pode-se encontrar, com relação às soluções do modelo inverso, três situações:

*i)* se  $n < m$  não existe solução;

*ii)* se  $n = m$  pode ocorrer dois casos.

- Existe um número finito de soluções fora das configurações singulares.
- Existe um número infinito de soluções nas configurações singulares.

*iii)* Se  $n > m$ , tem-se uma redundância e existe um número infinito de soluções. Para se obter uma solução única, introduz-se restrições no espaço operacional, ou usa-se um método que minimize determinado critério de desempenho.

Obviamente, se  $\underline{x}$  está fora do volume de trabalho do manipulador, não existe solução.

### 2.4.4 Matriz Jacobiana

A matriz Jacobiana, utilizada no método recursivo para o cálculo do modelo cinemático inverso é uma forma multidimensional da derivada e relaciona a velocidade no espaço de juntas à velocidade no espaço cartesiano isto é:

$$\Delta \mathbf{x}(t) = \mathbf{J} \cdot \Delta \theta(t) \quad (2.5)$$

Uma técnica para determinar o Jacobiano de um robô de seis graus de liberdade é fazendo uso das matrizes de transformações que definem a geometria do robô.

### **2.4.5 Inversão da Matriz Jacobiana**

A matriz Jacobiana inversa pode ser obtida através de dois métodos:

- Inversão simbólica: se deixarmos a matriz Jacobiana em sua forma original, forma simbólica, pode-se encontrar a inversa usando a álgebra. Mas a complexidade do Jacobiano torna este procedimento muito difícil ou até mesmo impossível.
- Inversão numérica: para cada instante, a configuração do robô define um conjunto de variáveis de juntas, deste modo, a matriz Jacobiana, em cada instante, é uma matriz numérica. A literatura em análise numérica apresenta diversas técnicas para a inversão de matrizes. Tais técnicas não devem apenas ser rápidas, mas também, retornar respostas precisas.

Pode-se notar que o modelo geométrico do robô é de grande importância pois, através dele pode-se calcular a matriz Jacobiana do robô e, além disso, ele é utilizado diretamente na malha de controle para a geração de uma trajetória ponto a ponto em tempo real.

No trabalho desenvolvido por Sá (2000) optou-se pela utilização do método recursivo para a geração de uma trajetória por ser de fácil implementação com um baixo número de operações aritméticas, reduzindo deste modo o tempo computacional gasto, embora o referido autor, apresente diferentes métodos para a inversão da matriz Jacobiana. São apresentados, em anexo, dois exemplos ilustrativos do procedimento descrito neste capítulo para obtenção do modelo cinemático.

### **2.5 Estrutura de Controle de um robô Industrial baseado no Modelo Cinemático**

Como mencionado anteriormente, em muitas aplicações a programação de tarefas de robôs é realizada no espaço das juntas e a trajetória é angular. Esses ângulos são de mesma natureza dos sinais provenientes do transdutor de posição, servindo como sinal de referência para o controlador de cada junta robótica, após interpolação.

Entretanto, na maioria das aplicações, a realização de tarefas está relacionada com o tipo de ferramenta utilizada, a partir de um sistema de coordenadas cartesianas fixo à base do robô, espaço cartesiano. Portanto, os movimentos desejados e as leis de controle estão em espaços diferentes, isto pode ser observado na figura 2.13.

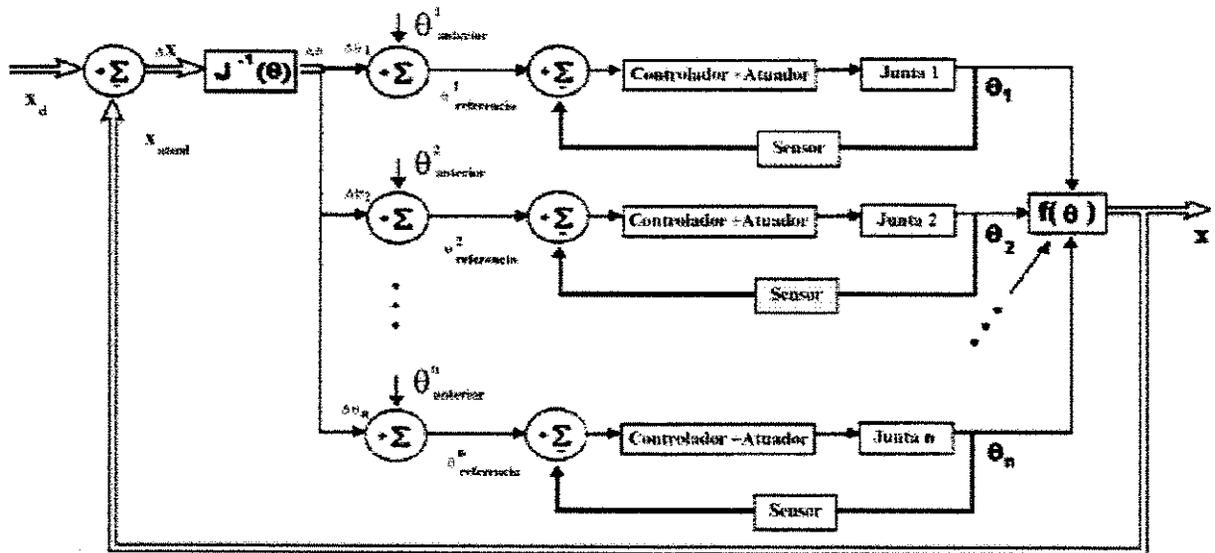


Figura 2.13: Estrutura de controle de um robô baseada no Modelo Cinemático.

Nesta figura pode-se observar que a partir da comparação de  $X_d$ , posição e orientação desejadas, e  $X_{atual}$ , referência, um sinal de erro é gerado e transposto em termos de erros de coordenadas articulares  $\Delta\theta_i$ ; as quais serão utilizadas como sinal de referência para o controle das juntas do robô.

A parte envolta por linhas pontilhadas representa o controle de trajetórias das juntas mostrando, deste modo, que o sistema suporta a concepção de trajetórias também no espaço das juntas.

## **2.6 Conclusão**

Nesse capítulo foi visto a estrutura de um robô industrial com ênfase no estudo da cinemática direta e inversa. Para o projeto e implementação do controlador de posição, torna-se importante o estudo dinâmico de uma junta robótica e a influência dos diferentes graus de liberdade de um robô. Posteriormente, nos capítulos (3, 4), apresentaremos um estudo completo da dinâmica e controle de uma junta robótica.

## **Capítulo 3**

### **Modelagem dinâmica de uma Junta Robótica**

Neste capítulo será abordado a modelagem e simulação de um sistema mecânico. Inicialmente estuda-se os elementos que compõem uma junta robótica, tais como: motor de corrente contínua, inércia, redutores e acoplamento. Estes estudos nos possibilita definir a trajetória a ser utilizada como referência dentro de uma malha de controle envolvendo estes elementos.

Posteriormente, são apresentadas as equações de um motor elétrico de corrente contínua, acrescentando a carga e acoplamentos. A partir das simulações do motor elétrico com seus acoplamentos, gera-se uma trajetória de referência para a malha de controle do sistema em questão.

#### **3.1 Modelagem de Sistemas Mecânicos**

Para uma modelagem a ser estudada consideremos um motor CC, um redutor e a dinâmica do sistema, figura 3.1. No estudo do motor CC será desenvolvido as equações referentes ao seu movimento. O redutor é utilizado para que, com um certo valor de torque se obtenha uma melhor performance do motor. A dinâmica do sistema consiste no estudo das forças que condicionam o movimento. Para acelerar um manipulador do seu estado inercial até uma velocidade constante e promover uma desaceleração, devem ser aplicados um conjunto de equações dinâmicas nas juntas dos atuadores (Craig, 1986).

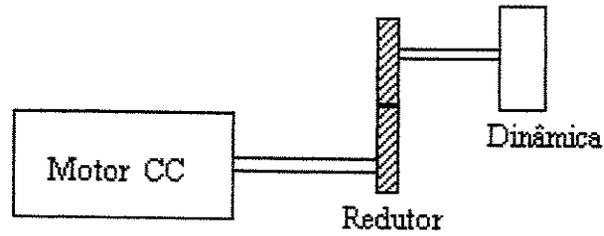


Figura 3.1: Modelo do sistema mecânico a ser estudado.

A representação de um motor CC (figura 3.2) controlador pela corrente da armadura é apresentado a seguir:

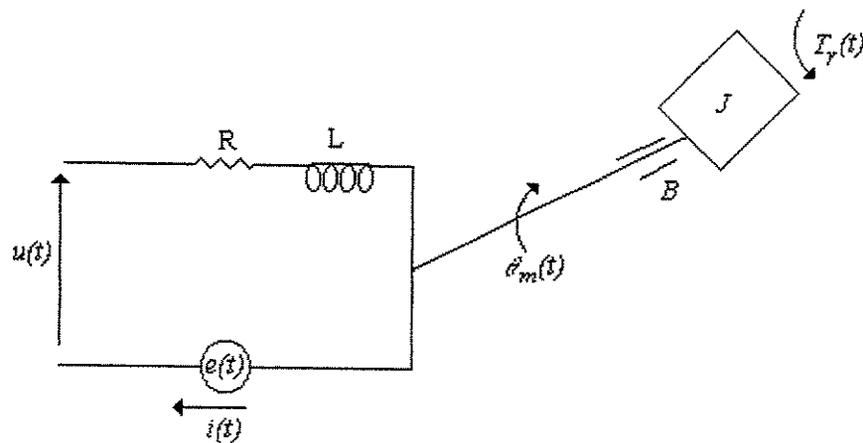


Figura 3.2: Representação Esquemática e Diagrama de blocos do Motor CC controlador pela corrente de armadura.

onde:

$i(t)$  - corrente (A);

$R$  - resistência induzida ( $\Omega$ );

$L$  - indutância (H);

$e(t)$  - força contra-eletromotriz (V);

$\theta_m$  - deslocamento angular (rad);

$u$  - tensão aplicada no circuito da armadura (V);

$J_m$  - momento de inércia do motor ( $\text{kg m}^2$ );  
 $K_e$  - constante da força contra-eletromotriz ( $\text{V/rad s}^{-1}$ );  
 $K_r$  - constante de torque ( $\text{Nm/A}$ );  
 $T_r$  - torque resistente (Nm)  
 $T_m$  - torque mecânico (Nm)  
 $B$  - fricção viscosa do motor ( $\text{Kg m/rad/s}$ )

O motor representado na figura 3.2, é regido pelas seguintes equações (Ogata, 1998):

### 3.1.1 Equação Elétrica

A velocidade de um motor CC controlado por armadura é controlada pela tensão de armadura  $u(t)$  que é suprida por um amplificador. A equação diferencial para o circuito de armadura é:

$$u(t) = Ri(t) + L \frac{di(t)}{dt} + e(t), \quad (3.1)$$

onde  $e(t) = K_e \Omega(t)$

Para um motor de corrente contínua (c.c.),  $K_e \approx K_r$

### 3.1.2 Equação de Acoplamento Eletro-mecânico

No motor CC controlado por armadura a corrente de campo é mantida constante. Para uma corrente de campo constante resulta um fluxo constante, o torque torna-se diretamente proporcional à corrente de armadura, de modo que:

$$T_m(t) = K_t i(t) \quad (3.2)$$

onde  $K_t$  é a constante de torque do motor.

### 3.1.3 Equação Mecânica

A corrente de armadura produz o torque que é aplicado à inércia e à fricção, portanto,

$$T_m(t) = J_m \frac{d\Omega(t)}{dt} + B\Omega(t) + T_r(t), \text{ onde } \Omega(t) = \dot{\theta} \quad (3.3)$$

### 3.2 Diagrama de Blocos

A seguir mostramos os diagramas de blocos associados as equações. O rearranjo das equações é feito para simplificar a organização do diagrama de blocos.

Para a equação 3.1, temos:

$$\frac{di(t)}{dt} = \frac{u(t)}{L} - \frac{Ri(t)}{L} - \frac{e(t)}{L} \quad (3.4)$$

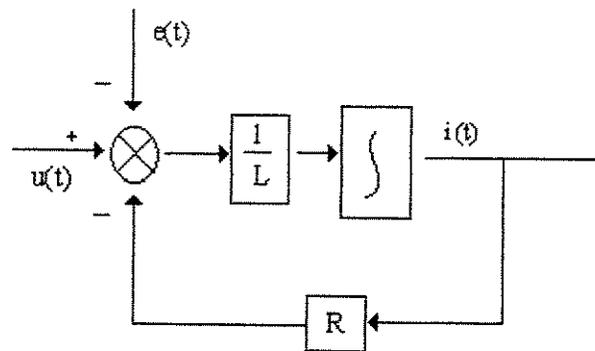


Figura 3.3: Diagrama de blocos para a equação 3.4.

Para a equação 3.2, temos:

$$T_m(t) = K_t i(t) \quad (3.5)$$

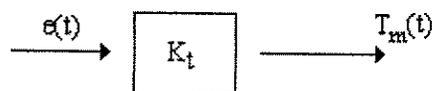


Figura 3.4: Diagrama de blocos para a equação 3.5

Para a equação 3.3, temos:

$$\frac{d\Omega(t)}{dt} = \frac{T_m(t)}{J_m} - \frac{B}{J_m}\Omega(t) - \frac{T_r(t)}{J_m} \quad (3.6)$$

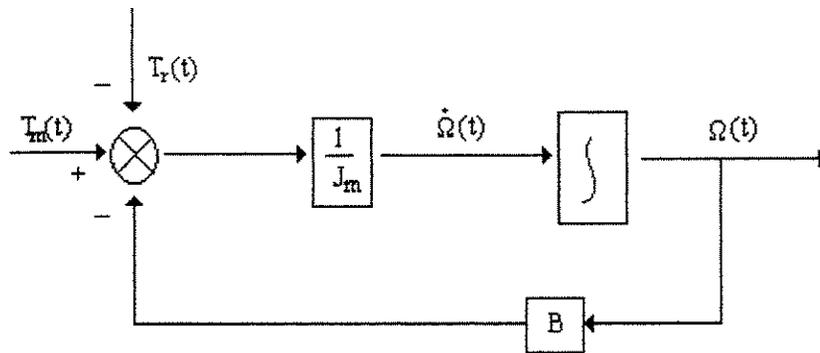


Figura 3.5: Diagrama de blocos para a equação 3.6

O diagrama de blocos para a equação completa, é mostrado a seguir:

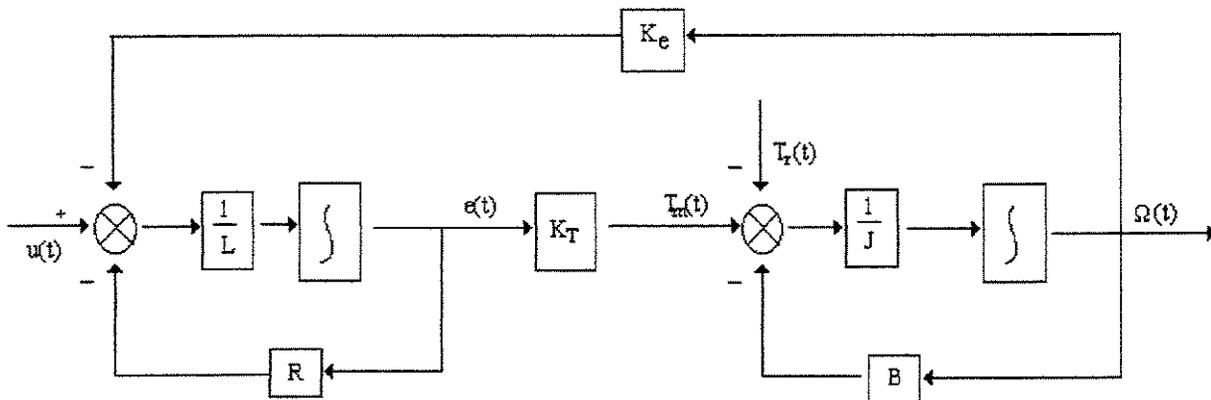


Figura 3.6: Diagrama de blocos para a equação completa.

### 3.3 Transformadas de Laplace

Utilizando as transformadas de Laplace chegamos as Funções de Transferência associada

UNICAMP  
BIBLIOTECA CENTRAL  
SEÇÃO CIRCULANTE

Para a equação 3.1, será:

$$U(s) = RI(s) + LsI(s) + E(s) \quad (3.7)$$

$$U(s) = (R + Ls)I(s) + E(s) \quad (3.8)$$

$$I(s) = \frac{1}{R + Ls}(U(s) - E(s)) = H_1(s)(U(s) - E(s)) \quad (3.9)$$

onde

$$H_1(s) = \frac{1}{R + Ls};$$

$$P_e = \frac{R}{L} \quad (\text{polo elétrico});$$

$$\tau_e = \frac{L}{R} \quad (\text{constante de tempo elétrica})$$

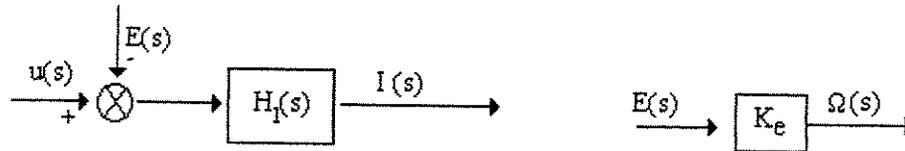


Figura 3.7: Diagramas de blocos para a equação 3.9.

Para a equação 3.2, temos:

$$T_m(s) = K_t I(s) \quad (3.10)$$

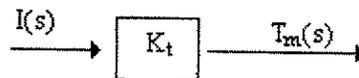


Figura 3.8: Diagrama de blocos para a equação 3.10.

Para a equação 3.3, temos:

$$T_m(s) = J_m s \Omega(s) + B \Omega(s) + T_r(s) \quad (3.11)$$

$$T_m(s) = \Omega(s)(J_m s + B) + T_r(s) \quad (3.12)$$

$$\Omega(s) = \frac{1}{J_m s + B} (T_m(s) - T_r(s)) = \Omega(s) = H_2(s)(T_m(s) - T_r(s)) \quad (3.13)$$

onde:

$$H_2(s) = \frac{1}{J_m s + B}$$

$$P_m = \frac{B}{J_m} \quad \text{polo mecânico}$$

$$\tau_m = \frac{J_m}{B} \quad \text{constante de tempo mecânica}$$

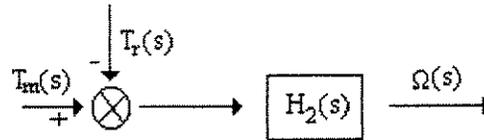


Figura 3.9: Diagrama de blocos para a equação 3.13.

Um diagrama de blocos para o modelo completo das funções de transferência é representado na figura 3.10.

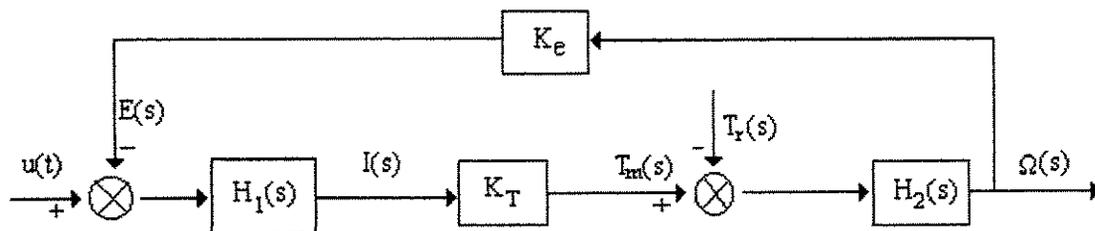


Figura 3.10: Diagrama de blocos para o modelo completo das funções de transferência

Pela manipulação algébrica das equações 3.9, 3.10 e 3.13, obtemos:

$$I(s) = H_1(s)(U(s) - E(s)) \quad (3.14)$$

$$T_m(s) = K_T I(s) \quad (3.15)$$

$$\Omega(s) = H_2(s)(T_m(s) - T_r(s)) \quad (3.16)$$

substituindo (3.14) em (3.15), temos:

$$T_m = K_t H_1(s)(U(s) - E(s)) \quad (3.17)$$

substituindo (3.17) em (3.16), temos:

$$\Omega(s) = H_2(s)[K_t H_1(s)(U(s) - E(s)) - T_r(s)] \quad (3.18)$$

$$\Omega(s) = (K_t H_1(s) H_2(s))(U(s) - E(s)) - H_2(s) T_r(s) \quad (3.19)$$

substituindo a equação  $E(s) = K_e \Omega(s)$  na equação 3.19, teremos:

$$\Omega(s) = (K_t H_1(s) H_2(s))(U(s) - K_e \Omega(s)) - H_2(s) T_r(s) \quad (3.20)$$

$$\Omega(s)[1 + K_t K_e H_1(s) H_2(s)] = K_t H_1(s) H_2(s) U(s) - H_2(s) T_r(s) \quad (3.21)$$

$$\Omega(s) = \frac{K_t H_1(s) H_2(s)}{1 + K_t K_e H_1(s) H_2(s)} U(s) - \frac{H_2(s)}{1 + K_t K_e H_1(s) H_2(s)} T_r(s) \quad (3.22)$$

$$\text{sendo: } H_3(s) = \frac{K_t H_1(s) H_2(s)}{1 + K_t K_e H_1(s) H_2(s)}$$

Se  $T_r(s)=0$  (sem perturbação)

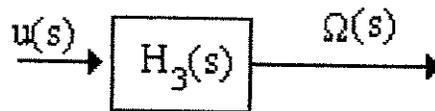


Figura 3.11: Diagrama de blocos para a equação 3.22 na situação em que  $T_r(s)=0$ .

onde:

$$H_1(s) = \frac{1}{R + LS}$$

$$H_2(s) = \frac{1}{J_m s + B}$$

A função de transferência será:

$$G(s) = \frac{\Omega(s)}{U(s)} = \frac{K_t}{LJ_m s^2 + (RJ_m + BL)s + BR + K_t K_e} \quad (3.23)$$

$$K_v = \frac{K_t}{BR + K_e K_t} \quad (\text{V/rad s}^{-1})$$

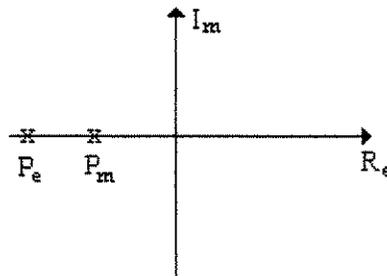


Figura 3.12. Gráfico dos pólos.

$$\tau_m = \frac{RJ_m}{BR + K_e K_t} \quad (\text{constante de tempo mecânica}).$$

### 3.4 Redutor

Considerando que uma junta robótica é constituída por motores, redutores e acoplamento, cabe aqui ressaltar os redutores de velocidades. Os sistemas de engrenagens transferem potência de um eixo para outro. As relações físicas que relaciona estes sistemas com outras grandezas, estão relacionadas por  $n$  que é a relação de engrenagens.

Dado um sistema de engrenagens, figura 3.13, as relações entre as grandezas, velocidade, deslocamento e torque são dados por:

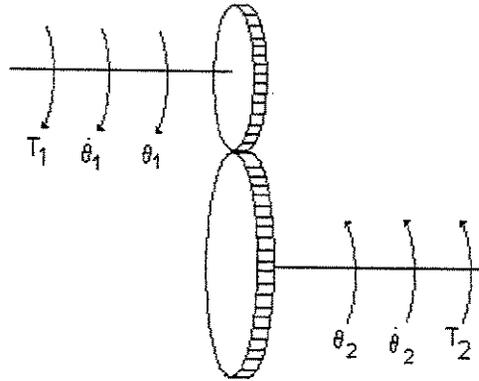


Figura 3.13: Sistema de engrenagens.

Seja,

$$\frac{n_2}{n_1} = n \quad (3.24)$$

a relação de engrenagens, então teremos as seguintes relações envolvendo as grandezas abaixo:

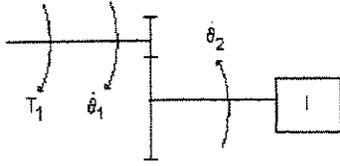
$$\frac{\theta_2}{\theta_1} = \frac{n_1}{n_2} = \frac{1}{n} \quad \text{relação de deslocamento} \quad (3.25)$$

$$\frac{\dot{\theta}_2}{\dot{\theta}_1} = \frac{n_1}{n_2} = \frac{1}{n} \quad \text{relação de velocidade} \quad (3.26)$$

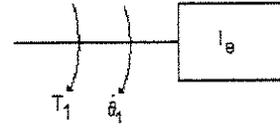
$$\frac{T_2}{T_1} = \frac{n_2}{n_1} = n \quad \text{relação de torque} \quad (3.27)$$

A seguir são apresentados, figura 3.14, os sistemas equivalente, relacionando as grandezas, inércia, rigidez e atrito viscoso acoplado por meio de sistemas de engrenagens, com suas respectivas equações equivalentes.

Sistema Original



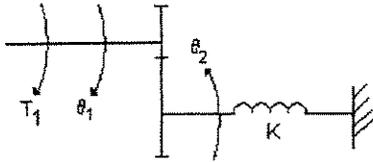
Sistema Equivalente



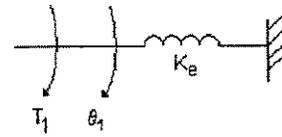
(a)

$$I_e = I \left( \frac{\dot{\theta}_2}{\dot{\theta}_1} \right)^2 = \frac{1}{n^2} I \quad (3.28)$$

Sistema Original



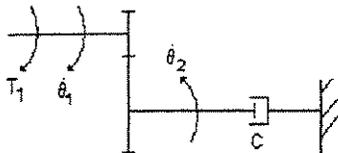
Sistema Equivalente



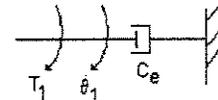
(b)

$$K_e = K \left( \frac{\theta_2}{\theta_1} \right)^2 = \frac{1}{n^2} K \quad (3.29)$$

Sistema Original



Sistema Equivalente



(c)

Figura 3.14: (a) inércia+ sistema de engrenagens, (b) rigidez + sistema de engrenagens, (c) atrito viscoso + sistema de engrenagens.

$$B_e = B \left( \frac{\dot{\theta}_2}{\dot{\theta}_1} \right)^2 = \frac{1}{n^2} B \quad (3.30)$$

Se considerarmos uma carga no eixo do redutor,

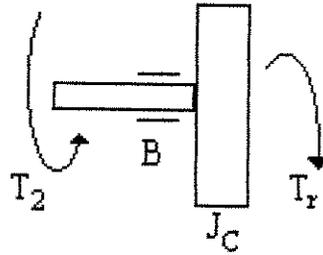


Figura 3.15: Representação da carga no eixo do redutor.

A equação que representará essa carga no redutor é dada por:

$$T_2 = J_c \ddot{\theta} + B_c \dot{\theta} + T_r \quad (3.31)$$

Aplicando a transformada de Laplace conseguimos as seguintes funções de transferência:

$$T_2(s) = (J_c s^2 + B_c s) \theta(s) + T_r(s) \quad (3.32)$$

$$\theta(s) = \frac{T_2(s) - T_r(s)}{J_c s^2 + B_c s} \quad (3.33)$$

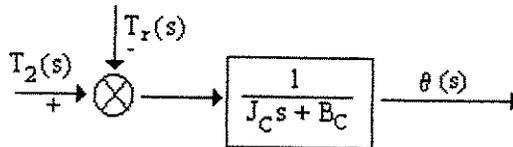


Figura 3.16: Diagrama de blocos para a equação 3.33.

### 3.5 Dinâmica do Sistema

Considerando agora o efeito de carga para:

O motor elétrico

$$(T_m(s) - T_r(s))H_2(s) = \Omega_{motor}(s) \quad (3.34)$$

O redutor e carga

$$(T_{carga}(s) - T_{pert.}(s))H_3(s) = \Omega_{carga}(s) \quad (3.35)$$

$$\Omega_{carga}(s) = \frac{1}{\eta} \Omega_{motor}(s) \quad \eta: \text{redutor} \quad (3.36)$$

$$T_{carga}(s) = \eta T_{motor}(s) \quad (3.37)$$

$$(\eta T_m(s) - T_{pert.}(s))H_3(s) = \frac{1}{\eta} \Omega_{motor}(s) \quad (3.38)$$

$$(\eta^2 T_m(s) - \eta T_{pert.}(s))H_3(s) = \Omega_{motor}(s) \quad (3.39)$$

Para melhor compreensão, consideraremos  $T_r(s), T_{pert.}(s) \approx 0$ , então:

$$T_m(s)H_2(s) = \Omega_{motor}(s) \quad (3.40)$$

$$\eta^2 T_m(s)H_3(s) = \Omega_{motor}(s) \quad (3.41)$$

Considerando fisicamente o problema,

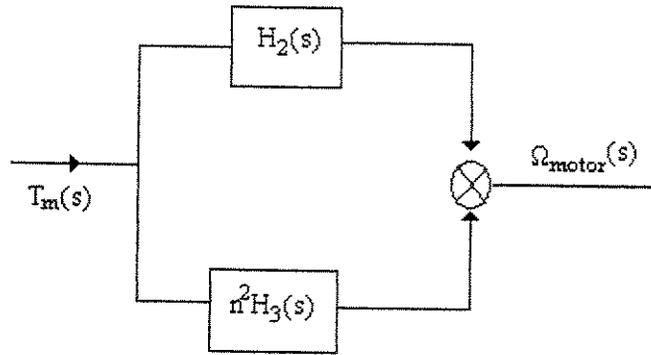


Figura 3.17: Diagrama de blocos para a equação do sistema.

Onde:

$$H_2(s) = \frac{1}{J_m(s) + B}$$

$$H_3(s) = \frac{1}{J_C(s) + B_C}$$

Com os resultados acima teremos:

$$\Omega_{motor}(s) = (H_2(s) + \eta^2 H_3(s))T_m(s) \quad (3.42)$$

ou ainda

$$\Omega_{motor}(s) = (J_m + \eta^2 J_C) + (B_m + \eta^2 B_C)T_m(s) \quad (3.43)$$

### 3. 6 Conclusão

Nesse capítulo foi abordado os elementos que compõem uma junta robótica, tais como: motor de corrente contínua, inércia, redutores e acoplamento. Este estudo nos possibilitou obter a modelagem dinâmica de uma junta robótica, necessário para o estudo do modelo dinâmico de um manipulador robótico.

No capítulo 4 o estudo será complementado considerando a dinâmica do robô, alterando os termos  $J$ ,  $B$  e  $T$ , mostrados nesse capítulo, agora com ênfase no desenvolvimento, simulação e implementação completa de um robô.

## **Capítulo 4**

### **Modelagem Dinâmica de um Robô Industrial**

Neste capítulo será complementado o estudo do modelo dinâmico de um manipulador robótico, com ênfase no desenvolvimento, simulação e implementação de um controlador de posição para uma junta robótica completa de um robô sujeita a perturbações causadas pelo acoplamento das outras juntas.

Além disso, apresentamos um estudo, para avaliar o comportamento do sistema, envolvendo a simulação do pêndulo duplo, cujos resultados são apresentados neste capítulo.

#### **4.1 Modelo Dinâmico de um Robô Manipulador - Caso Não Linear**

Os robôs industriais são servos mecanismos que utilizam tecnologia de alta precisão. A posição final do braço do robô é conseguida acionando os servos mecanismos que estão colocados em suas juntas. Em muitas aplicações não há um conhecimento prévio sobre qual é o melhor formato do braço a ser utilizado: a melhor combinação será encontrada com a forma do braço e o servo mecanismo a ser utilizado.

O robô é um sistema não linear devido a essas não linearidades dos acoplamentos e seus movimentos, a variação no momento de inércia e a dependência com a geometria.

Para se obter as equações dinâmicas de um robô manipulador com  $n$  graus de liberdade pode-se utilizar o sistema das equações de Euler-Lagrange de 2ª ordem:

$$\frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} = T_i, L = K - V \quad (4.1)$$

onde:

$K$  é a energia cinética

$V$  é a energia potencial

$T_i$  é a força generalizada

$q_i$  é a coordenada generalizada

As equações de movimento, para a  $i$ -ésima junta de um robô com  $n$  graus de liberdade, são da forma:

$$T_i = J_i \ddot{q}_i + B_i \dot{q}_i + T_{r,i} \quad (4.2)$$

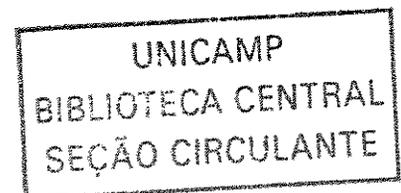
onde:

$J_i$  é o momento de inércia

$B_i$  é o atrito viscoso do motor

$T_{r,i}$  é o torque resistente

Tendo em vista que os dois graus de liberdade (2,3) responsáveis pelo posicionamento da garra, num dado ponto de um volume de trabalho, são os mais críticos do ponto de vista estrutural, acarretando problemas na implementação do controlador de posição das juntas, devido a seus “fortes” acoplamentos e não linearidades, a nossa análise será desenvolvida considerando esses 2 graus de liberdade, o caso mais crítico, podendo ser estendida futuramente aos outros graus de liberdade.



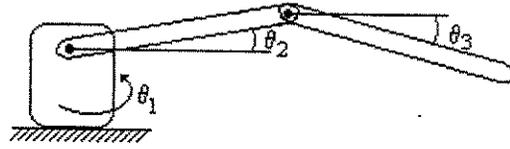


Figura 4.1: Manipulador com três graus de liberdade

Sendo assim, o modelo dinâmico de um robô industrial poderá ser simplificado para o modelo de duas juntas rotativas que servirá para posicionamento de uma ferramenta em relação a um painel de atuação, por exemplo, nesse caso, o sistema possui dois graus de liberdade ( $n=2$ ) e suas equações de movimento podem ser obtidas de forma análoga.

Com o objetivo de estudar o comportamento do sistema desde a situação mais simples (rígido e desacoplado), até a situação de maior complexidade (sistema acoplado), é proposta uma conveniente e sistemática parametrização dos termos nas equações de movimento, que para o manipulador rígido com dois graus de liberdade é mostrada a seguir (David, 1996):

$$J_2 \ddot{\theta}_2 + B_2 \dot{\theta}_2 + T_{r,2} = T_{m,2} \quad (4.3)$$

$$J_3 \ddot{\theta}_3 + B_3 \dot{\theta}_3 + T_{r,3} = T_{m,3} \quad (4.4)$$

onde os coeficientes são dados por:

$$J_2 = A_2 + K_1 + A_3 c \theta_3 \quad (4.5)$$

$$B_2 = -A_3 \dot{\theta}_3 s \theta_3 \quad (4.6)$$

$$T_{r,2} = (A_3 + \frac{1}{2} A_2 c \theta_3) \ddot{\theta}_3 + (-\frac{1}{2} A_2 \dot{\theta}_3^2 s \theta_3) + A_4 c \theta_2 + A_5 c (\theta_2 + \theta_3) \quad (4.7)$$

$$J_3 = A_3 \quad (4.8)$$

$$B_3 = 0 \quad (4.9)$$

$$T_{r,3} = (A_3 + \frac{1}{2}A_2c\theta_3)\ddot{\theta}_2 + \frac{1}{2}A_2\dot{\theta}_2^2s\theta_3 + A_5c(\theta_2 + \theta_3) \quad (4.10)$$

e

$$A_1 = \alpha_1 + \alpha_1' \quad A_4 = \alpha_4 + \alpha_4'$$

$$A_2 = \alpha_2 + \alpha_2' \quad A_5 = \alpha_5 + \alpha_5'$$

$$A_3 = \alpha_3 + \alpha_3'$$

$$K_1 = m_j L_1^2$$

com

$$\alpha_1 = \frac{1}{3}m_1L_1^2 + \frac{1}{3}m_2L_2^2 + m_2L_1^2 \quad \alpha_1' = m_pL_1^2 + m_pL_2^2 \quad (4.11)$$

$$\alpha_2 = m_2L_1L_2 \quad \alpha_2' = 2m_pL_1L_2 \quad (4.12)$$

$$\alpha_3 = \frac{1}{3}m_2L_2^2 \quad \alpha_3' = m_pL_2^2 \quad (4.13)$$

$$\alpha_4 = \frac{1}{2}m_1L_1g + m_2L_1g \quad \alpha_4' = m_pL_1g \quad (4.14)$$

$$\alpha_5 = \frac{1}{2}m_2L_2g \quad \alpha_5' = m_pL_2g \quad (4.15)$$

$m_1, m_2, L_1, L_2$  são parâmetros do sistema estudado

$m_p$  é a massa colocada no sistema final

$m_j$  é a massa da junta

A figura 4.2 mostra o modelo do sistema, neste caso não linear onde podemos observar o modelo das duas juntas.

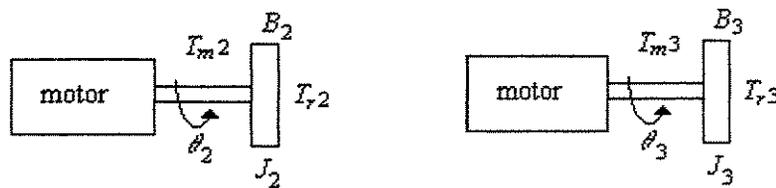


Figura 4.2: Representação da carga no motor

Um diagrama de blocos para o caso não linear é mostrado na figura 4.3.

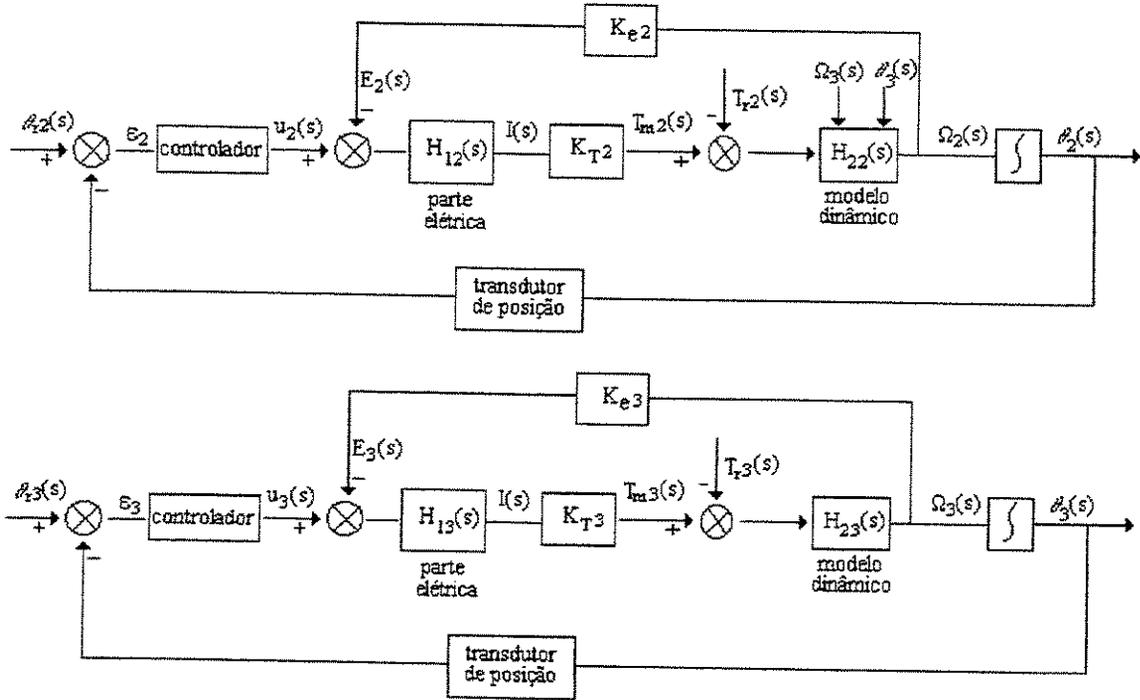


Figura 4.3: Diagrama de blocos para as equações não lineares

Os sinais de referência  $\theta_2$  e  $\theta_3$  serão gerados por um interpolador de trajetórias desenvolvido anteriormente no laboratório de Automação e Robótica da Unicamp (SÁ,1996). A partir da comparação destes sinais de referência com as posições angulares provenientes dos transdutores de posição de cada junta (encoder incremental) o controlador fará as devidas correções levando-se em conta o modelo dinâmico do robô em estudo.

Para exemplificar o estudo entre as duas juntas robóticas mais críticas do ponto de vista estrutural, será apresentado exemplos de aplicação sobre o pêndulo duplo.

#### 4.2 Exemplos de Aplicação - Pêndulo Duplo - Dinâmica e Controle

A modelagem dinâmica de um pêndulo duplo (Santos, 2001; Palm, 1986), figura 4.4, representa o modelo de dois graus de liberdade de um robô. As equações dinâmicas do pêndulo

duplo são descritas abaixo, já acrescida da dinâmica dos motores, representando cada grau de liberdade.

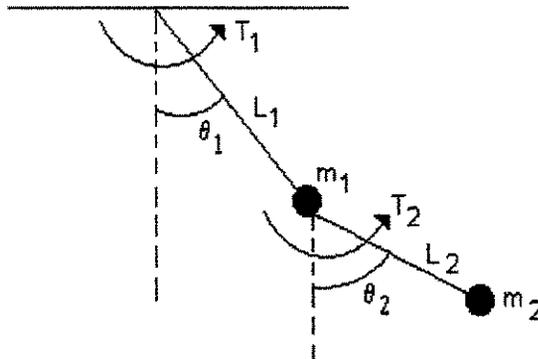
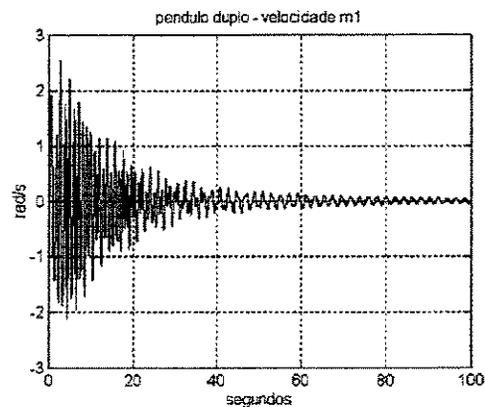
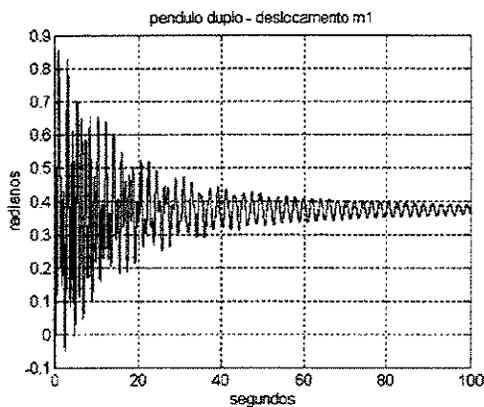


Figura 4.4: Pêndulo Duplo

$$T_1 = (J_1 + (m_1 + m_2)l_1^2)\ddot{\theta}_1 + B_1\dot{\theta}_1 + m_2l_1l_2\ddot{\theta}_2 \cos(\theta_2 - \theta_1) - m_2l_1l_2\dot{\theta}_2^2 \sin(\theta_2 - \theta_1) + (m_1 + m_2)l_1g \sin \theta_1 \quad (4.16)$$

$$T_2 = (J_2 + m_2l_2^2)\ddot{\theta}_2 + B_2\dot{\theta}_2 + m_2l_1l_2\ddot{\theta}_1 \cos(\theta_2 - \theta_1) + m_2l_1l_2\dot{\theta}_1^2 \sin(\theta_2 - \theta_1) + m_2l_2g \sin \theta_2 \quad (4.17)$$

As equações acima conduzem as seguintes curvas, as quais representam a dinâmica do pêndulo duplo, onde aplicamos uma excitação do tipo degrau (figura 4.5).



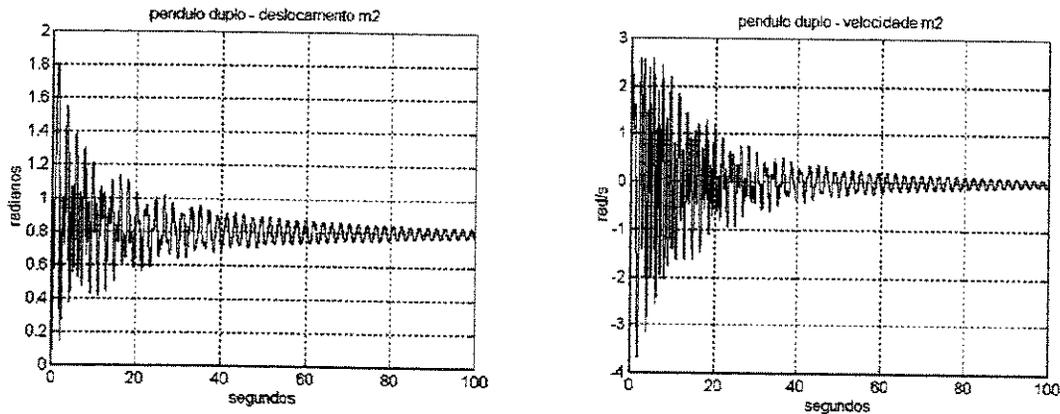
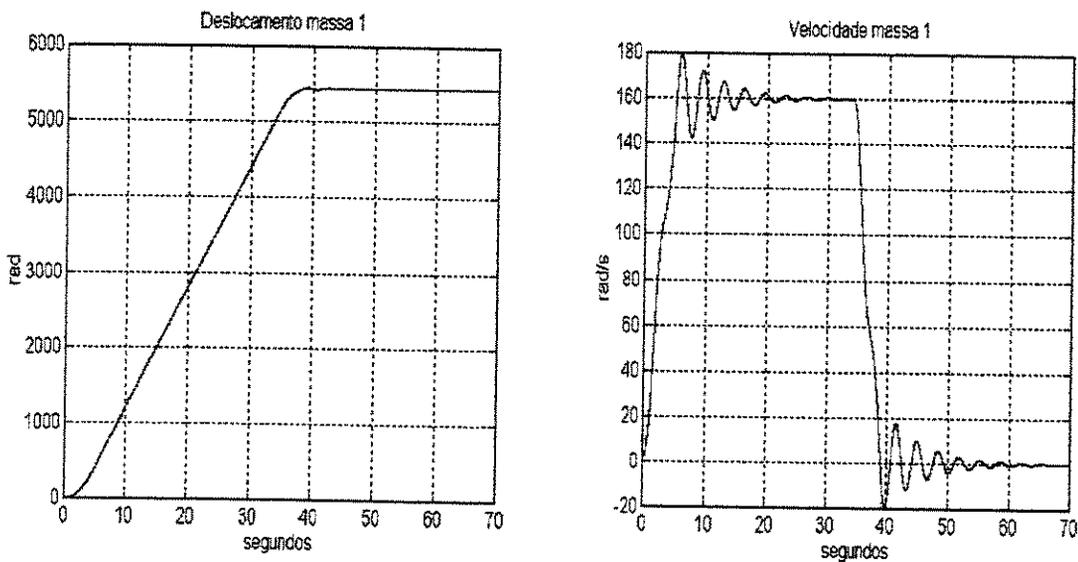


Figura 4.5: Dinâmica do Pêndulo Duplo

Uma vez estabelecidos os modelos dos sistemas de interesse, vamos avaliar o seu desempenho em malha fechada, utilizando um controlador do tipo PID e depois usando a técnica de compensação, figuras 4.6 e 4.7, implementados em MATLAB<sup>TM</sup>, utilizando diagrama de blocos, os quais representam as equações 4.16 e 4.17.



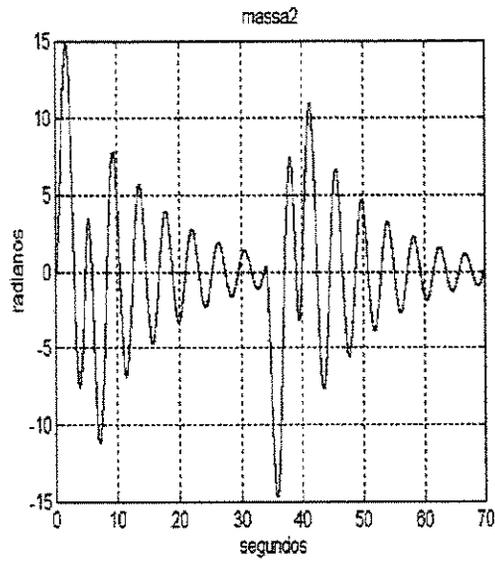
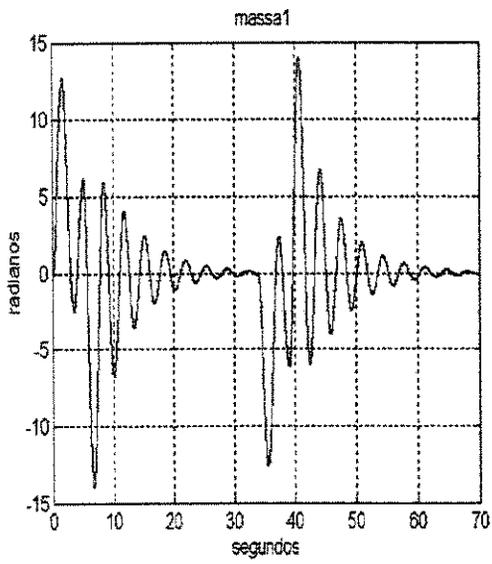
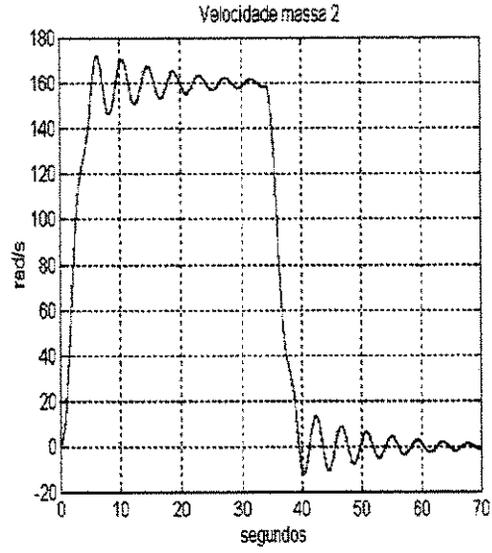
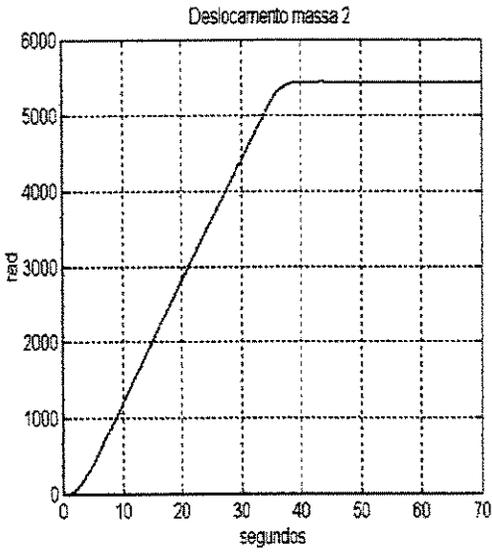


Figura 4.6: Pêndulo Duplo – Controle (os dois últimos gráficos representam o erro de trajetória)

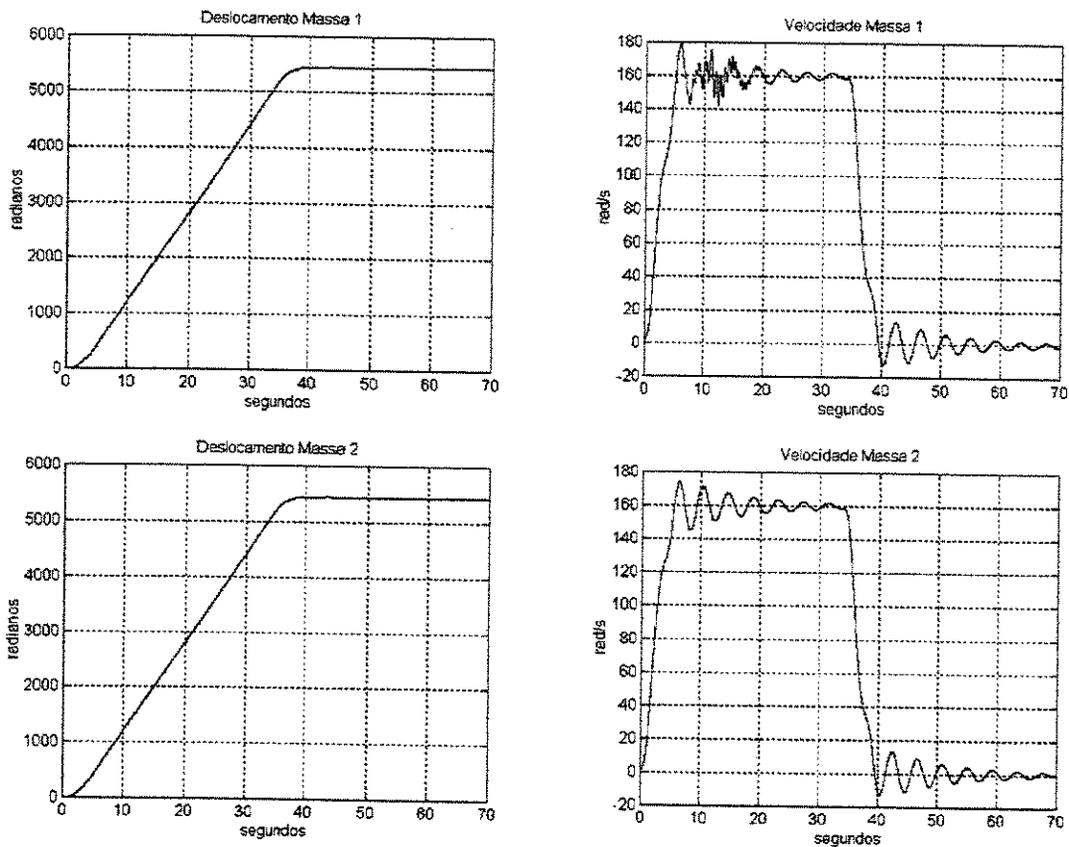


Figura 4.7: Pêndulo Duplo - Controle com Massa Estimada

### 4.3 Conclusão

O estudo apresentado nesse capítulo possibilitaram concluir a modelagem dinâmica do sistema, cujos resultados auxiliaram na confecção de um controlador de maior eficiência, apresentado no capítulo seguinte.

## **Capítulo 5**

### **Projeto de um Controlador PID para juntas Robóticas a partir do Modelo Dinâmico do Sistema**

O controlador é uma das peças fundamentais de um sistema completo de controle. A ele cabe a tarefa de controle global num estado/posição pré-determinado. O tipo de controlador mais utilizado é o PID (Proporcional/Integral/Derivativo). Este tipo de controlador possui um bom desempenho desde que o sistema a controlar seja conhecido, bem comportado (linear e invariante no tempo) e os parâmetros do controlador bem ajustados. A sua principal limitação, derivada do procedimento de ajuste, é ser sensível às diferentes condições de funcionamento. Nesta situação, a utilização de um controlador com ganhos variáveis é vantajoso.

Para podermos comparar o controlador proposto neste projeto de pesquisa (não linear utilizando lógica reprogramável), foi realizado durante esta etapa do trabalho um estudo inicial de um controlador PID com ganhos variáveis (é realizada uma tabela de ganhos em função de suas principais configurações inerciais). Isto se deve ao fato de que num manipulador robótico, durante a execução de uma determinada trajetória, apresentará uma grande variação de seus parâmetros dinâmicos (termos inerciais, coeficientes de atrito, efeitos gravitacionais), acarretando na mudança dos parâmetros do controlador projetado em função de sua configuração inercial.

Assim sendo, neste capítulo serão propostos métodos experimentais na determinação dos valores utilizados no controlador PID direcionados a uma junta robótica.

## 5.1 Projeto de um Controlador de posição PID

O presente trabalho tem como objetivo a modelagem de sistemas dinâmicos multivariáveis acoplados típicos da área de robótica, onde, através de simulações, via SIMULINK™, poderemos analisar o comportamento do sistema em questão.

O procedimento para cálculo dos ganhos do regulador PID, será realizado em cada junta, para diferentes configurações inerciais, considerando-se o efeito da inércia equivalente no eixo de rotação do motor (cálculo do raio de giração e massa equivalente). As constantes serão descritas como se segue:

$$J = J_m + mk^2$$

$$B$$

$$T_r = mgk \operatorname{sen}\theta$$

onde  $k$  é o raio de giração

Para o cálculo dos ganhos do regulador PID, o sistema será linearizado para as diferentes configurações inerciais que para pequenos ângulos, temos:

$$\operatorname{sen}\theta \approx \theta$$

Aplicando a transformada de Laplace:

$$\tau(s) = (Js^2 + Bs + T_r) \theta(s) \quad (5.1)$$

A função de transferência será:

$$T(s) = \frac{\theta(s)}{\tau(s)} = \frac{1}{Js^2 + Bs + T_r} \quad (5.2)$$

Através do critério de estabilidade de Routh-Hurwitz (Ogata,1998), podemos verificar a estabilidade do sistema e aplicando o método de Ziegler-Nichols (Ogata,1998) vamos encontrar

os valores utilizados no controlador PID. Estas simulações nos darão as bases necessárias para que possamos comparar os resultados obtidos com o do controlador proposto.

## 5.2 Determinação dos ganhos de um regulador PID

Após verificar a estabilidade do sistema, vamos calcular os ganhos do regulador PID. Neste trabalho, os ganhos do regulador PID para diferentes configurações inerciais serão calculados e validados a partir de simulação computacional, utilizando métodos baseados na otimização do comportamento da resposta do sistema, tradicionalmente utilizados em aplicações industriais.

### 5.2.1 Método de Ziegler-Nichols

A figura 5.1 mostra o diagrama de blocos de um sistema do qual se deseja calcular os ganhos do regulador PID, onde  $G_c(s)$  é equação do regulador PID e  $G_p(s)$  é a equação do sistema.

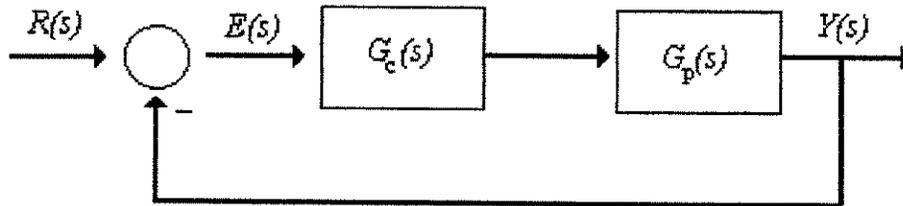


Figura 5.1: Esquema de um sistema para o cálculo do PID.

A equação de um controlador PID é dada por:

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \quad (5.3)$$

onde:

$K_p$  é o ganho proporcional

$K_i$  é o ganho integral

$K_d$  é o ganho derivativo

O método de Ziegler-Nichols para o cálculo dos ganhos do compensador PID é baseado no cálculo de energia mínima da resposta do sistema, consistindo na execução dos seguintes procedimentos:

1. Um compensador proporcional é aplicado:

$$G_c(s) = K_p$$

O ganho deverá ser ajustado até o sistema tornar-se marginalmente estável. Este valor de ganho é denominado  $K_{po}$ , e o período de oscilação  $T_o$ .

2. Assim, o ganho do compensador será definido por:

$$G_c(s) = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) \quad (5.4)$$

Onde teremos:

$$\text{Ganho do termo integrador: } K_i = \frac{K_p}{T_i} \quad (5.5)$$

$$\text{Ganho do termo Derivativo: } K_d = K_p T_d \quad (5.6)$$

Conseqüentemente, os valores utilizados no cálculo dos ganhos do controlador PID utilizando o método de Ziegler-Nichols são dados pela tabela 5.1

Tipo de Controlador	Ganhos do Controlador		
	$K_p$	$T_i$	$T_d$
<b>P</b>	$K_{po}$	$\infty$	0
<b>PI</b>	$0,5 K_{po}$	$T_i = 0,83 T_o$	0
<b>PID</b>	$0,5 K_{po}$	$T_i = 0,5 T_o$	$T_d = 0,125 T_o$

Tabela 5.1: Ganhos do compensador PID.

Como neste trabalho de pesquisa, o modelo dinâmico de um manipulador é conhecido, a metodologia de cálculo dos parâmetros do regulador será realizada utilizando a mesma metodologia apresentada anteriormente, mas com o ajuste do ganho  $K_p$  realizado através do critério de Routh-Hurwitz (a linha do polinômio  $s^2$  é forçada a zero, conseqüentemente o sistema é marginalmente estável).

### 5.2.2 Método de Chien-Hrones-Reswick (CHR)

Este método é bastante utilizado na indústria, no caso de não conhecimento do modelo dinâmico de um sistema. Esse método consiste no estudo da resposta do sistema em malha aberta submetido a uma entrada degrau (na prática um off-set) e a obtenção da respectiva curva de resposta temporal, como é mostrado na figura 5.2.

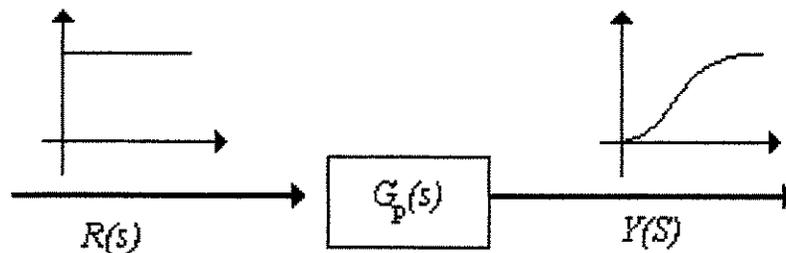


Figura 5.2: Método de CHR - obtenção da resposta temporal a uma entrada degrau.

Conseqüentemente, os ganhos do controlador PID poderão ser facilmente obtidos a partir da curva resposta temporal do sistema submetido a uma entrada degrau (figura 5.3).

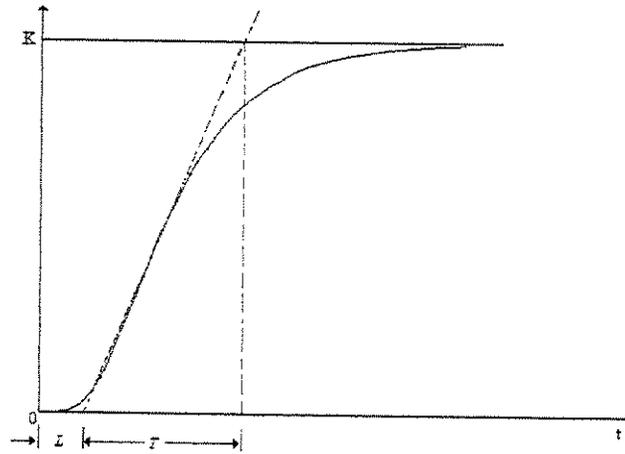


Figura 5.3: Curva resposta temporal utilizada no método CHR

Os seguintes procedimentos deverão ser utilizados:

1. Com o auxílio da curva resposta teremos os valores de  $T$  e  $L$ , obtidos a partir do cruzamento da reta tangente com a interseção no eixo do tempo, conforme mostra a figura 5.3.
2. Os valores para a escolha dos ganhos do controlador PID resultam do cálculo da razão  $R$ , obtida através da relação 5.7, onde a escolha do melhor controlador é obtido através da tabela 5.2.

$$R = \frac{T}{L} \tag{5.7}$$

Tipo de controlador	R
P	$R > 10$
PI	$7,5 < R < 10$
PID	$3 < R < 7,5$

Tabela 5.2: Escolha do tipo de Controlador utilizando o método de CHR.

3. A partir da escolha do controlado, a tabela 5.3 deverá ser utilizada para o cálculo dos ganhos dos diferentes tipos de controlador, onde  $K_p = G_p(0)$ .

Tipo de Controlador	Ganhos do Controlador		
	$K_p$	$T_i$	$T_d$
P	$\frac{T}{L}$	$\infty$	0
PI	$0,9 \frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2 \frac{T}{L}$	2L	0,5L

Tabela 5.3: Cálculo dos Ganhos do Controlador utilizando o método de CHR.

Este método foi aplicado para cálculo dos ganhos do controlador nos exemplos de validação.

### 5.2.3 Exemplos de validação - Cálculo dos ganhos do controlador PID

Apresentaremos a seguir o resultado do cálculo dos ganhos do controlador PID para algumas configurações inerciais do robô. Para efeito de análise e estudo inicial, utilizaremos a hipótese em que o eixo motriz está posicionada nestas configurações, não havendo variações da mesma no tempo. Conseqüentemente, os ganhos do controlador PID obtidos serão constantes.

A massa do sistema será considerada em seus centros de massa dos braços do robô para podermos obter a curva característica e calcular o valor das constantes utilizadas no PID. O cálculo para o momento de inércia do sistema é apresentado no anexo VI deste trabalho. A tabela 5.4 apresenta as configurações e valores dos parâmetros inerciais utilizados.

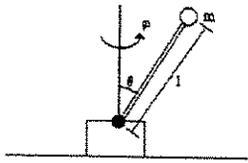
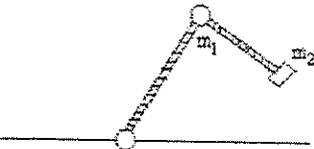
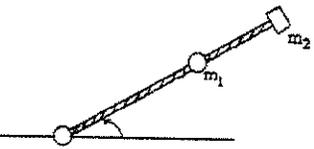
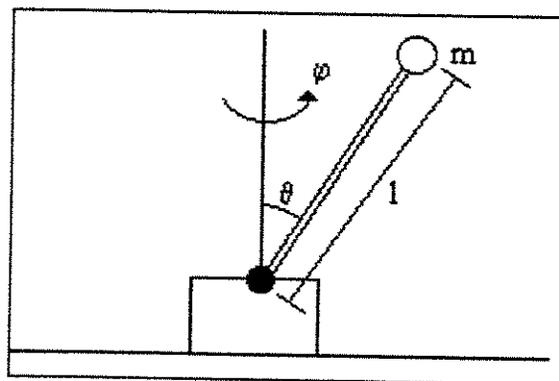
No	Configuração	Massa M (Kg)	Comprimento L (m)	Ganhos do Controlador (PID)
1		0,2	L = 0,2	<b>Kp = 1.2</b> <b>Ki = 0.6</b> <b>Kd = 0.3</b>
2		0,2	L1 = 0,2 L2 = 0,2	<b>Kp = 4.8</b> <b>Ki = 0.125</b> <b>Kd = 0.03</b>
3		0,2	L1 = 0,2 L2 = 0,2	<b>Kp = 6.6</b> <b>Ki = 0.1</b> <b>Kd = 0.025</b>

Tabela 5.4: Configurações, parâmetros utilizados, ganhos do Controlador PID.

### Configuração 1:



**m = 0,2 Kg**  
**l = 0,2 m**  
**g = 9,8 m/s<sup>2</sup>**

Figura 5.4: Configuração Inercial 1 e parâmetros utilizados.

Os valores utilizados nesta simulação e a configuração utilizada (nº 1, um grau de liberdade, com raio de giração  $l$ ) foram apresentados na figura 5.4. A partir da utilização do procedimento descrito anteriormente, os ganhos do controlador PID obtidos foram respectivamente:  $K_p = 1.2$ ,  $K_i = 0.6$  e  $K_d = 0.3$ .

A figura 5.5 apresenta os resultados obtidos a partir da simulação deste modelo utilizando o SIMULINK™, com um controlador de posição PID.

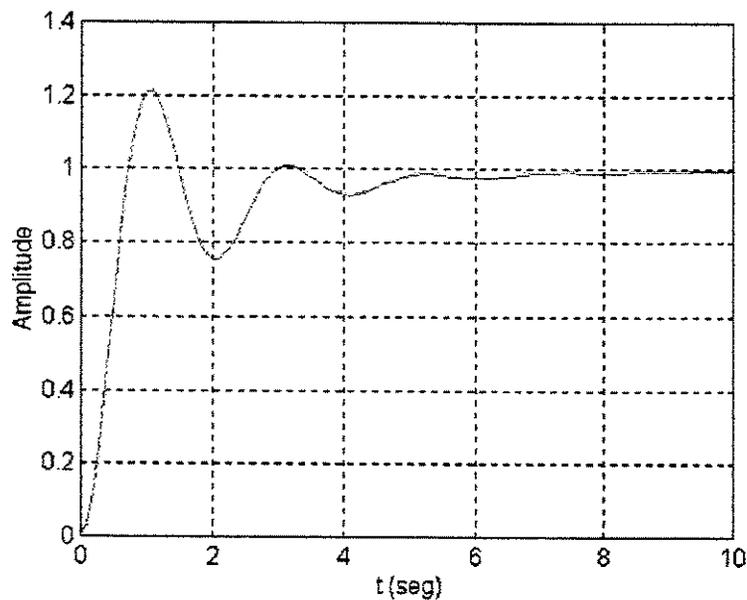
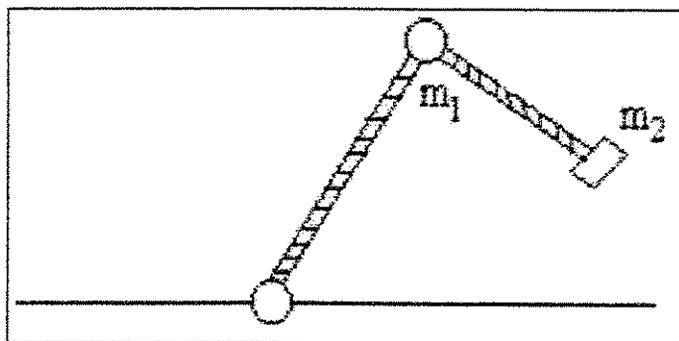


Figura 5.5: Resposta temporal a uma excitação degrau utilizando controlador PID (configuração 1).

**Configuração 2:**



$m = 0,2 \text{ Kg}$   
 $l = 0,2 \text{ m}$   
 $g = 9,8 \text{ m/s}^2$

Figura 5.6: Configuração Inercial 2 e parâmetros utilizados.

Os valores utilizados nesta simulação e a configuração utilizada ( $n^{\circ}$  2, dois graus de liberdade) foram apresentados na figura 5.6. A partir da utilização do procedimento descrito anteriormente, os ganhos do controlador PID obtidos foram respectivamente:  $K_p = 4.8$ ,  $K_i = 0.125$  e  $K_d = 0.03$ .

A figura 5.7 apresenta os resultados obtidos a partir da simulação deste modelo utilizando o SIMULINK<sup>TM</sup>, com um controlador de posição PID.

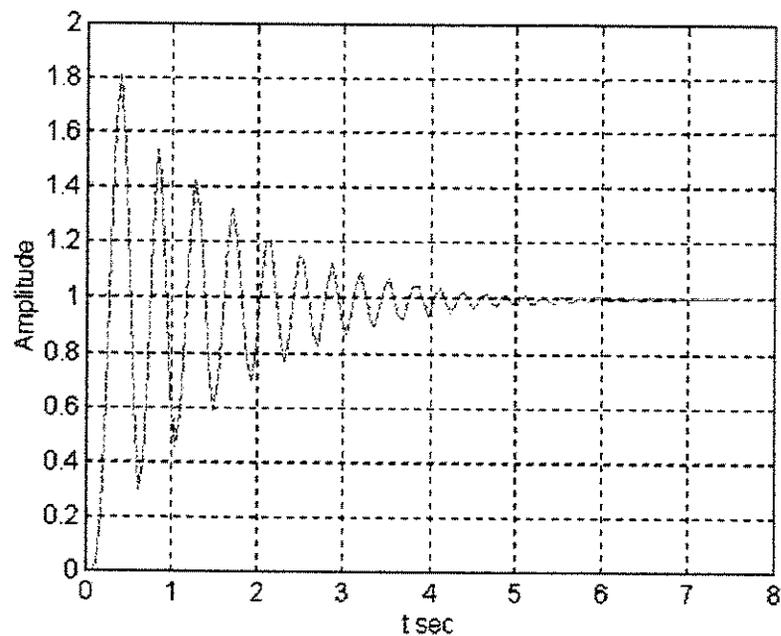
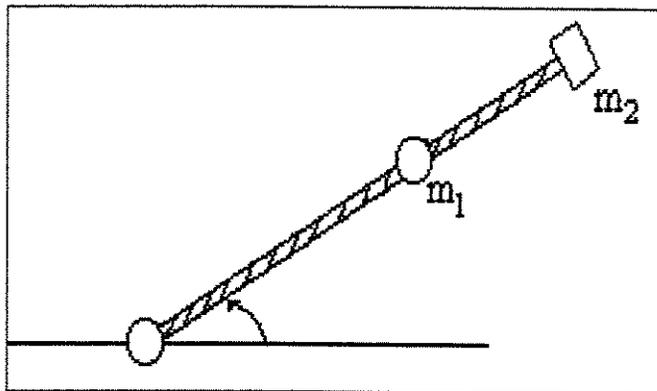


Figura 5.7: Resposta temporal a uma excitação degrau utilizando controlador PID (configuração2).

### Configuração 3:



$$\begin{aligned} m &= 0,2 \text{ Kg} \\ l &= 0,2 \text{ m} \\ g &= 9,8 \text{ m/s}^2 \end{aligned}$$

Figura 5.8: Configuração Inercial 3 e parâmetros utilizados.

Os valores utilizados nesta simulação e a configuração utilizada (nº 3, dois graus de liberdade com segundo braço totalmente esticado) foram apresentados na figura 5.8. A partir da utilização do procedimento descrito anteriormente, os ganhos do controlador PID obtidos foram respectivamente:  $K_p = 6.6$ ,  $K_i = 0.1$  e  $K_d = 0.025$ .

A figura 5.9 apresenta os resultados obtidos a partir da simulação deste modelo utilizando o SIMULINK™, com um controlador de posição PID.

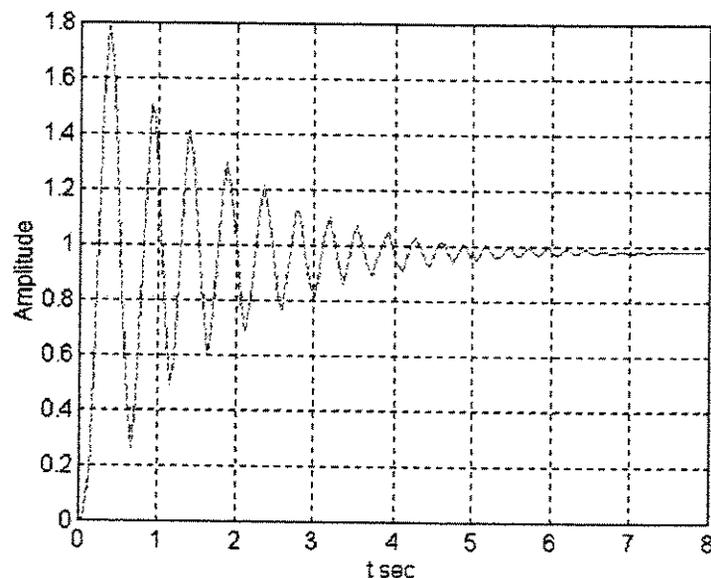


Figura 5.9: Configuração Inercial 3 e parâmetros utilizados.

### **5.3 Conclusão**

Neste capítulo foi realizado um estudo inicial de um controlador PID com ganhos variáveis devido as suas diferentes configurações inerciais.

A implementação do controlador de posição será realizada na próxima etapa deste trabalho de pesquisa, utilizando o software de lógica reprogramável ALTERA™, com o objetivo principal na obtenção de um controlador de posição que considere as não-linearidades do sistema em estudo.

## Capítulo 6

### Modelagem de um Controlador não linear

Neste capítulo são apresentados aspectos conceituais, abordando controladores não lineares, requisito para a proposta de implementação experimental de controladores não lineares em juntas robóticas, utilizando componente lógicos reprogramáveis EPLD, apresentado no próximo capítulo.

#### 6.1 Revisão de Conceitos abordando Sistemas não lineares

Sabe-se que muitas das relações entre quantidades físicas não são lineares, embora, freqüentemente sejam aproximadas por equações lineares, principalmente pela simplicidade matemática. Essa simplificação pode ser satisfatória desde que as soluções resultantes estejam de acordo com os resultados experimentais. Uma das características mais importantes de sistemas não lineares é a dependência do comportamento da resposta do sistema em relação à amplitude e ao tipo de entrada. Por exemplo, um sistema não linear pode se comportar de forma completamente diferente em resposta a entradas de diferentes amplitudes.

A ciência de controle de automação em uma grande parte está interessada na identificação, análise e no projeto dos sistemas dinâmicos. Embora o critério de projeto inclua alguns aspectos relacionados com o controle do sistema e que o projeto satisfatório não possa ser planejado sem uma análise anterior, permanece o fato básico em que a engenharia de controle emprega um considerável tempo nessas análises.

A ferramenta matemática da engenharia de controle é a teoria de sistemas dinâmicos, no qual esses também são empregados em um controle de processos químicos ou em um sistema de controle biológico. A teoria se fundamenta em um sistema linear e invariante no tempo, podendo ser contínuo ou discreto.

Um sistema linear é aquele cuja representação obedece ao princípio da superposição. Este estabelece que se uma entrada  $r_1(t)$  produz a saída  $c_1(t)$  e uma entrada  $r_2(t)$  produz uma saída  $c_2(t)$ , então para todo  $a, b, r_1(t), r_2(t)$  a resposta para  $ar_1(t) + br_2(t)$  será  $ac_1(t) + bc_2(t)$ . Além disso o sistema é definido como tempo invariante se a entrada  $r_1(t+\tau)$  produz a saída  $c_1(t+\tau)$  para todos os valores de  $\tau$  e funções de entrada  $r_1(t)$ . Dessa forma, nesta classe de sistemas, a origem do tempo não é consequência. Em princípio, a resposta para qualquer entrada pode ser solucionada a partir da resposta de qualquer outra entrada conhecida que tenha energia diferente de zero para todas as frequências. O elemento de entrada de uso mais comum como função base é o impulso unitário, a resposta para tal é o impulso resposta ou função peso.

Infelizmente, nenhum sistema físico é pertencente à classe de sistemas lineares invariantes no tempo. Além disso é verdade, se não por outras razões, que há sempre alguns limites para uma faixa (taxa) de variáveis físicas e que as componentes mudam com o tempo. Este estado não tem um significado claro em sistema de tempo linear invariante, mesmo havendo significado, há limitações e em algumas situações é essencial haver teorias que considere os efeitos da não linearidade e/ou dependência do tempo.

Um sistema não linear pode ser definido como aquele no qual os princípios da superposição não são aplicados. Dessa forma, não há significado pelo qual a resposta para uma entrada particular possa ser encontrada se a resposta para alguma outra entrada é conhecida. Como resultado desse fundamento, existe a dificuldade com sistemas não lineares de não ser possível uma aproximação uniforme para suas análises.

Os métodos analíticos são utilizados para encontrar soluções de um tipo específico de entrada. Isto porque com a teoria de funções descritivas podemos encontrar respostas para muitos destes problemas, particularmente para sistema com mais de uma entrada, no qual a aproximação

é importante. A este respeito os métodos de aproximação são utilizados por não haver uma descrição teórica adequada. Não há um método geral para se lidar com todos os sistemas não lineares, mas há muitas maneiras para se propor à resolução de um sistema não linear.

## **6.2 Sistemas de controle não lineares**

Diferentes tipos de não linearidades podem ser encontradas em sistemas de controles reais, podendo ser divididos em duas classes, dependendo de serem inerentes ao sistema ou inseridos propositadamente a este.

As não linearidades inerentes são inevitáveis em sistemas de controle. Exemplos de tais não linearidades são:

1. saturação
2. zona morta
3. histereses
4. folga
5. atrito estático, atrito de Coulomb e outros atritos não lineares
6. mola não linear

De uma forma geral, a presença de tais não linearidades no sistema de controle afeta prejudicialmente seu desempenho.

Não linearidades essenciais ou inseridas são resultantes de uma intenção deliberada de engenheiro, visando a simplificação do projeto, dependendo das condições exigidas.

### **6.2.1 Abordagens utilizadas na análise e projeto de sistemas de controle não lineares**

Uma maneira de se analisar e projetar um grupo particular de sistemas de controle não linear, em que o grau de não linearidade é pequeno, é usar técnicas de linearização equivalente e resolver o problema linearizado obtido.

O método da função descritiva seria um destes, sendo permitido estudar a estabilidade de muitos sistemas de controle não lineares simples do ponto de vista do domínio da frequência.

Este método nos fornece informação sobre a estabilidade para um sistema de qualquer ordem, porém não fornece informação exata sobre as características de resposta temporal. Em muitos casos particulares, estamos interessados principalmente na estabilidade de sistemas de controle não lineares e soluções analíticas de equações diferenciais não lineares.

Outras formas de analisar e projetar sistemas de controle não linear, que podem ser altamente não lineares, incluem a atual solução das equações diferenciais não lineares ou simplificações destas por meio de técnicas de plano de fase.

O método do plano de fase fornece informação tanto sobre estabilidade quanto sobre o comportamento e resposta temporal, mas é limitado a sistemas de primeira e segunda ordem.

### **6.2.2 Não linearidades usuais mais freqüentemente encontradas em servo sistemas**

A característica da amplitude da saída versus entrada em regime estático de um sistema perfeitamente linear é retilíneo para todos os valores de amplitude de entrada de zero ao infinito (figura 6.1 a), isto exprime a propriedade da homogeneidade e expressa a equação.

$$s(t) = Ket(t) \quad \text{em regime estático permanente}$$

onde K é o ganho estático do bloco.

Na realidade, a característica de amplitude de um sistema real, jamais é perfeitamente retilínea, sempre tende a uma característica gráfica curva, Figura 6.1.

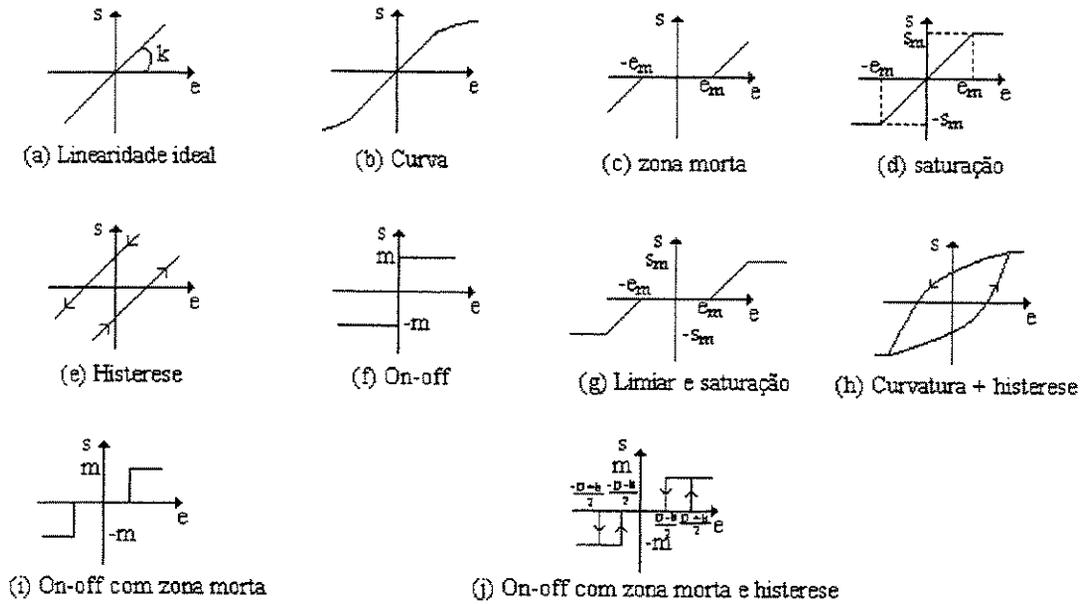


Figura 6.1: Não linearidades usuais

Além disso, todo sistema físico mesmo sendo linear, possui um limiar de sensibilidade ou zona morta e uma saturação, figuras 6.1 c,d.

De outra parte, os fenômenos hereditários do tipo histerese fornece curva com pode ser vista na figura 6.1 e.

Em fim, os sistemas que trabalham com on-off possuem uma característica representada na figura 6.1 f, e é definida pela equação:

$$s(t) = M \text{ sinal } e(T)$$

onde função sinal é igual a +1 para  $e > 0$  e a -1 para  $e < 0$  (idealização de duas posições reais).

Nós temos cinco tipos fundamentais de não linearidade representadas nas figuras de 6.1 b a 6.1 f.

Estes cinco tipos de não linearidades não se apresentam esquematizadas comodamente, podendo elas associar-se de diversas maneiras, com inúmeras possibilidades de combinações, o que permite representar todos os tipos de não linearidades possíveis em um sistema físico, figuras 6.1g a 6.1j.

### **6.3 Modelos de sistemas físicos**

A implementação de um esquema de controle envolve várias passos que podem ser apresentados como se segue:

1. estudo preliminar e especificação da performance
2. modelagem
3. análise
4. simulação
5. projeto e otimização
6. realizações físicas

Os passos acima, necessariamente, não seriam feitos na seqüência em que foram citadas e certamente, no processo de teste, pode ser usado posteriormente e isso pode levar a uma repetibilidade dos processos usados acima.

O estágio de modelagem, que é a formulação matemática a qual relaciona as variáveis do sistema, é extremamente importante desde que os resultados para grandes estágios, exceto aqueles das relações físicas, são obtidos usando a modelagem de equações. Não há pontos de aplicação que utilizem sofisticadas técnicas de otimização em sistemas que sejam “mal” modelados. Os modelos para os elementos de um sistema podem ser obtidos de leis físicas básicas controlando seus comportamentos ou por medidas experimentais.

A forma de um determinado modelo para um comportamento particular vai depender de como foi obtido e pode também estar bem próximo de como foi utilizado em uma análise particular. O projetista deverá sempre recorrer às hipóteses feitas durante a determinação do modelo e estar ciente da mudança das variáveis, o que é suficiente para se obter uma boa

precisão. Muitas técnicas analíticas têm sido extensamente usadas nos últimos anos as quais utilizam modelos complexos. Muitos destes modelos analíticos utilizam diagramas de bloco.

Um junta robótica pode ser representada através do modelo elementar do movimento de um pêndulo simples, apresentado na figura 6.2, que é descrito por uma equação diferencial de segunda ordem

$$\ddot{x} + (g/l)\text{sen } x = 0 \quad (6.1)$$

onde  $l$  é o comprimento do pêndulo e  $g$  é a aceleração da gravidade. Se aproximarmos  $\text{sen } x$  por  $x$ , para pequenas oscilações de  $x$ , a equação diferencial fica reduzida a uma equação familiar

$$\ddot{x} + w^2 x = 0 \quad (6.2)$$

onde  $w^2 = g/l$ .

A equação 6.2 descreve um movimento harmônico simples e ela aparece em várias situações, tendo como solução

$$x(t) = A\text{sen}(wt + \phi) \quad (6.3)$$

onde  $A$  e  $\phi$  são constantes a serem determinadas pelas condições iniciais. A equação diferencial 6.2 descrevendo um movimento harmônico simples é um exemplo de equação diferencial linear. A equação diferencial linear possui a propriedade da superposição, isto é, se  $x_1$  e  $x_2$  são soluções da equação diferencial, então  $ax_1 + bx_2$  é também uma solução para a equação diferencial, onde  $a$  e  $b$  são constantes arbitrárias.

Um exemplo de uma equação não linear é a equação 6.1, que não satisfaz a propriedade da superposição. A equação diferencial 6.2 aproxima o comportamento do pêndulo para pequenas oscilações de  $x$ . Por exemplo, o período de oscilação do modelo do pêndulo simples apresentado na figura 6.2 depende das condições iniciais, embora o período de oscilação para o movimento harmônico simples é independente das condições iniciais.

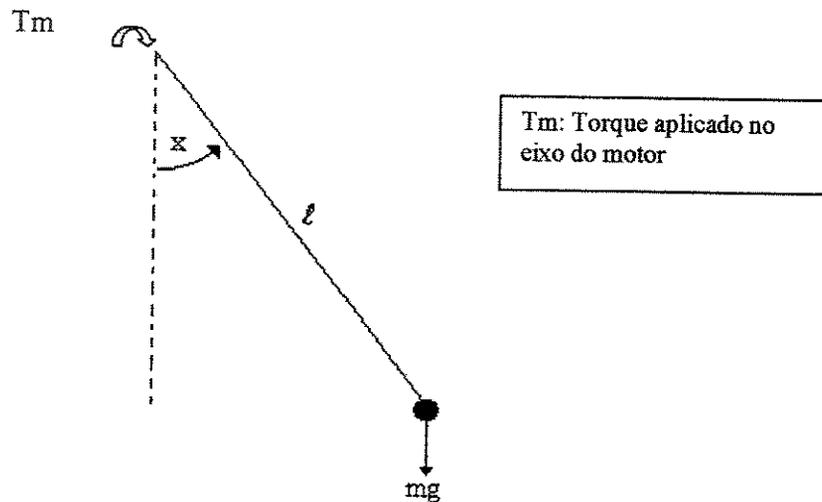


Figura 6.2: Movimento de um pêndulo simples

### 6.3.1 Comportamento de um sistema não linear

Já que um sistema não linear não obedece a lei da superposição, análises e medidas têm sido obtidas por várias maneiras e magnitude a fim de descobrir as várias facetas do comportamento. A engenharia de controle deve, entretanto, considerar todas as várias possibilidades e parâmetros ao qual o sistema está submetido. Idéias qualitativas que ocorrerá em um sistema não linear para um tipo de entrada, dado sua performance em relação à outra, existirá embora sua convicção em tais comparações pode somente ser obtidas de considerações experimentais de vários aspectos do comportamento não linear que serão associados com diferentes tipos de entrada.

Algumas das propriedades distintas de sistemas não lineares são mencionadas aqui, começando com aquelas de sistemas que não recebem dados externos. Como um sistema é usualmente referido como um sistema autônomo e, matematicamente podendo ser descrito por uma ou um conjunto de equações diferenciais. Primeiramente, a resposta do sistema será dependente das condições iniciais e não será uma combinação linear de exponenciais ou exponenciais dependente do tempo como no caso de um sistema linear. A resposta da configuração pode ser diferentemente marcada. Em alguns casos, isto pode convergir a um ponto

de equilíbrio e em outros casos pode não convergir, isto é o sistema será instável. Múltiplos pontos de equilíbrio podem existir, tanto que quando um estado fixo é alcançado ele pode não ser o mesmo para todas as condições iniciais.

### **6.3.2 Métodos de estudo de um sistema não linear**

O desempenho de um sistema não linear pode ser estudado por hardware, simulação e/ou análise. Rigorosamente falando, os dois casos consideram a performance do modelo do sistema. Testes e medidas do sistema atual, que certamente providenciaram as informações mais reais sobre como será seu comportamento, é o estado final do projeto. Construindo um sistema e testando-o sem análises ou simulação pode ser perigoso e dispendioso. Além disso se não há uma performance como esperado não haverá uma idéia de como testá-lo podendo ser calculado para compensação. Técnicas adequadas de hardware podem, algumas vezes, ser facilmente aplicadas nos estágios iniciais do projeto se os modelos de escala forem usados, como por exemplo navios e aeronaves.

Modernos computadores são usados para análises de modelos complexos. Entretanto, para simular sem ter feito algumas análises preliminares pode levar a um completo arraso do projeto. Cada simulação em um computador pode auxiliar na solução de um dado parâmetro de entrada. As aproximações teóricas são necessárias para indicar a magnitude dos dados de entrada o qual revelará todas as possibilidades do comportamento do sistema, os efeitos dos distúrbios, os diferentes valores de parâmetros e mudanças na configuração do sistema. Já que essas análises e o projeto de um sistema não linear envolve aproximações, as simulações em computadores são essenciais para um completo estudo.

### **6.3.3 Descrição matemática**

O comportamento dinâmico temporal de um sistema robótico é descrito pelas equações dinâmicas que são constituídas por expressões matemáticas. Elas expressam a forma de como a energia externa aplicada ao servo mecanismo é trocada entre seus armazenadores. As respostas do sistema, as forças nele aplicada descrevem seu comportamento dinâmico que, obviamente,

está em função de sua estrutura, a qual é definida pelas constantes de massa, inércia, rigidez, comprimento e outros parâmetros.

Um atuador usado em estruturas pode ser rotacional ou linear e pode incluir transmissões, caixas de engrenagens e articulações. Os mecanismos de transmissão podem também ser modelados no sistema dinâmico adicionando-se variáveis de estado apropriadas para o atuador. Sendo assim, para simplificar o trabalho, a dinâmica da transmissão não é incluída nela e o acionamento é tratado como um corpo rígido com inércia.

As equações do movimento são obtidas das leis da mecânica de Newton quanto de Euler-Lagrange. Apesar dos princípios elementares serem exatamente os mesmos, os diferentes formalismos tratam o estudo do manipulador de formas particulares, cada um possibilitando a elaboração de uma equação de movimento particular, com características peculiares.

Utilizaremos o método de Euler-Lagrange para o estudo da dinâmica do sistema, sendo que este possibilita a dedução de um conjunto de equações não lineares de segunda ordem acopladas que, desprezando-se o efeito do controlador e atuador representam matricialmente o sistema. Além disso, este enfoque possibilita encontrar matrizes que representam os termos envolvidos no movimento do processo como termos inerciais, centrípetos e de Coriolis causados pela velocidade das juntas e os termos devido à ação da gravidade.

#### **6.4 Modelo dinâmico de um manipulador**

Podemos considerar um braço de um robô como um pêndulo em equilíbrio instável. O método clássico para estabilizá-lo consiste na aplicação ao sistema de uma ação proporcional ao desvio em relação ao valor da posição.

O pêndulo tem uma faixa de tolerância de erro em torno da posição de equilíbrio, a qual é denominada zona morta. Quando o erro ultrapassa este limiar, o acionador desenvolve uma ação de torque contrária para compensar o desvio.

Devido ao caráter no ponto de equilíbrio ser instável, é importante notar que ao ser implementada uma ação de controle por relê, este tende a permanecer em regime oscilatório constante na região do ponto final.

Na figura 6.3 representamos o modelo físico, sendo importante salientar que quanto menor for a largura da zona morta, menor será a amplitude da oscilação final, podendo, conforme for, até desaparecer.

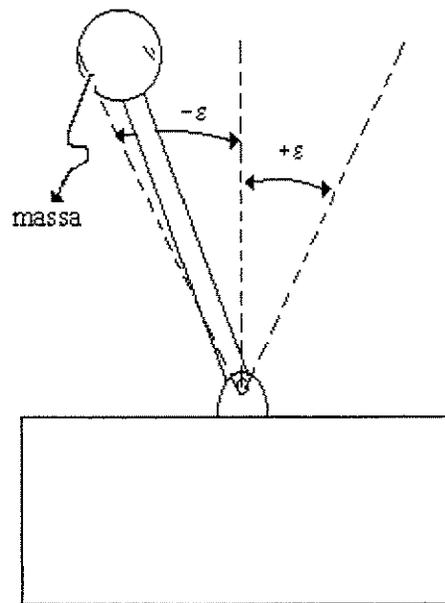


Figura 6.3: Modelo físico de um Manipulador

#### 6.4.1 Descrição Dinâmica de uma junta robótica

A figura 6.4 representa a estrutura de uma cadeia cinemática de uma articulação de um robô, considerando aspectos dinâmicos.

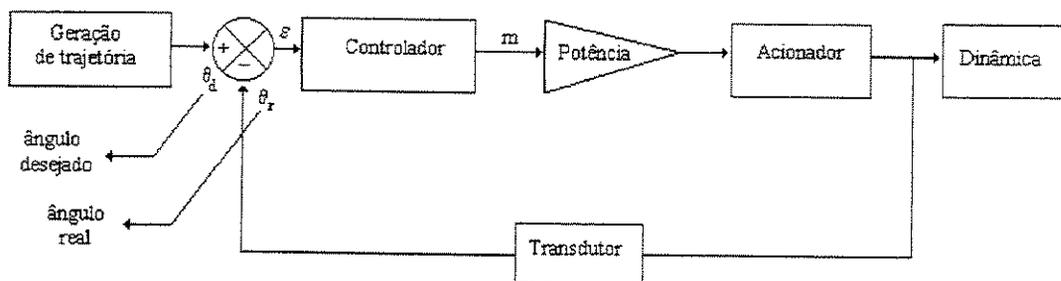


Figura 6.4: Diagrama básico de controle de um manipulador

Um sinal de referência é comparado com o sinal que realimenta o sistema na caixa soma, decodificado por um transdutor de posição o qual traz a informação do posicionamento atual do servo mecanismo, e é comparado com o sinal consignado pelo gerenciador, fornecendo o sinal do erro. Esse sinal resultante, pode ser tratado ao passar pelo bloco controlador, o qual vai gerar a ação corretora, com o propósito de reduzir o grau de desvio do desejado com o real.

O sinal que sai do bloco controlador, geralmente tem intensidade de corrente elétrica muito pequena, sendo que antes de ir ao bloco acionador, é amplificado e dimensionado de tal forma que fique compatível com o tipo de sistema de acionamento utilizado.

O bloco acionador, no caso do manipulador tratado neste trabalho, é constituído de um motor elétrico de corrente contínua acoplado a uma caixa de redução, alterando o torque e a velocidade angular fornecido pelo motor.

Na redução é acoplada a parte de carregamento do manipulador, a qual é constituída pelo braço e o conjunto ferramenta terminal e a massa presa a esta.

A carga é constituída fundamentalmente pelas inércias em movimento. Nota-se que a carga é variável: em efeito, as inércias em movimento dependem da posição do robô a cada instante.

### 6.4.2 Função Descritiva

A função descritiva mais usualmente estudada é aquela para o qual se tem uma entrada senoidal. Esta função tem como objetivo a representação linear de um sistema não linear sujeitos a uma entrada senoidal e seu uso na análise de um sistema não linear é então baseado que a entrada dos elementos não lineares são senoidais.

Na análise por função descritiva, suponhamos que apenas a componente harmônica fundamental da saída é significativa.

O método é um ensaio da generalização da noção da função descritiva para o caso de sistemas não lineares.

Para um sistema linear de entrada  $\varepsilon$  e saída  $s$  definimos a função de transferência a partir do sinal senoidal.

$$\varepsilon = \varepsilon_0 \text{ sen } \omega t \quad (6.4)$$

pela relação da amplitude  $A = s_0/\varepsilon_0$  e da fase  $\phi$  da resposta

$$s = s_0 \text{ sen } (\omega T + \phi) \quad (6.5)$$

graças à propriedade de linearidade;  $A$  e  $\phi$  são funções da frequência da entrada, porém não dependem mais de suas amplitudes  $\varepsilon_0$ : isto permite definir uma função de transferência que está em função somente de  $\omega$ .

Ao estendermos o processo a um sistema não linear, somos conduzidos a estudar a saída devido a uma entrada harmônica.

$$x = s_1 \text{ sen } \omega t \quad (6.6)$$

Esta saída  $s(t)$  será em geral periódica porém, não senoidal. Se por uma convenção qualquer, definirmos uma função senoidal

$$w(t) = w_1 \text{ sen } (\omega t + \psi) \quad (6.7)$$

que qualificamos (arbitrariamente) equivalente a função  $s(t)$ , podemos definir para o sistema uma função de transferência equivalente ou generalizada

$$\text{módulo} = (w_1/x_1), \quad \text{argumento} = \psi. \quad (6.8)$$

Como função senoidal equivalente, fixamos geralmente a primeira harmônica de  $s(t)$

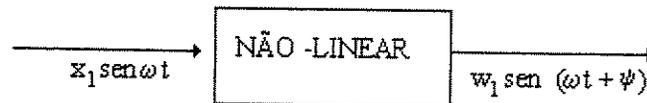


Figura 6.5 - Entrada no bloco NL da primeira hamônica de  $s(t)$

A diferença como no caso linear é dupla:

- 1) Ele é um elemento arbitrário na escolha da senóide equivalente da saída.
- 2) A função de transferência obtida não depende somente de  $\omega$ , mas sim também da amplitude da entrada  $x_1$ , pois não há a proporcionalidade dos efeitos e causas como no caso linear:

$$w_1/x_1 = B(x_1, \omega) \quad \psi = \psi(x_1, \omega) \quad (6.9)$$

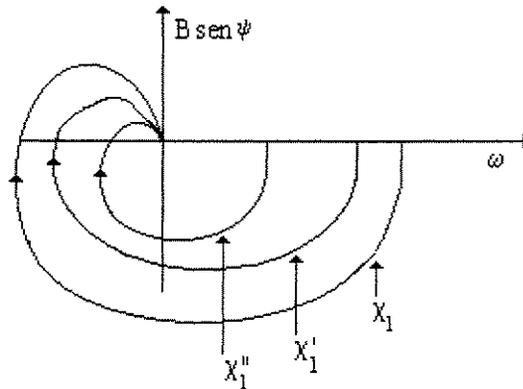


Figura 6.6: Família de linhas de transferência, em função da amplitude de entrada

Geometricamente, efetuamos para um sistema não linear, não mais uma linha de transferência graduada em frequência, porém uma família de linhas de transferência, em função da amplitude de entrada.

A validade prática da aproximação de equivalência harmônica repousa sobre a filtragem de alta frequência que realiza o estágio de posse de todo o servo-mecanismo.

Com esse estudo podemos verificar os formalismos que muitas vezes não são vistos nos equacionamentos dos manipuladores.

#### 6.4.3 Modelo Dinâmico de uma Junta Robótica considerando aspectos não lineares

O comportamento de um sistema robótico é descrito pelas equações dinâmicas as quais são complexas e não lineares (Rosário,1998) e esse comportamento dinâmico depende das características do sistema.

Usaremos as equações de Euler-Lagrange para estudar a dinâmica do sistema, as quais conduzem a um conjunto de equações dinâmicas não-lineares que podem ser representadas na forma matricial.

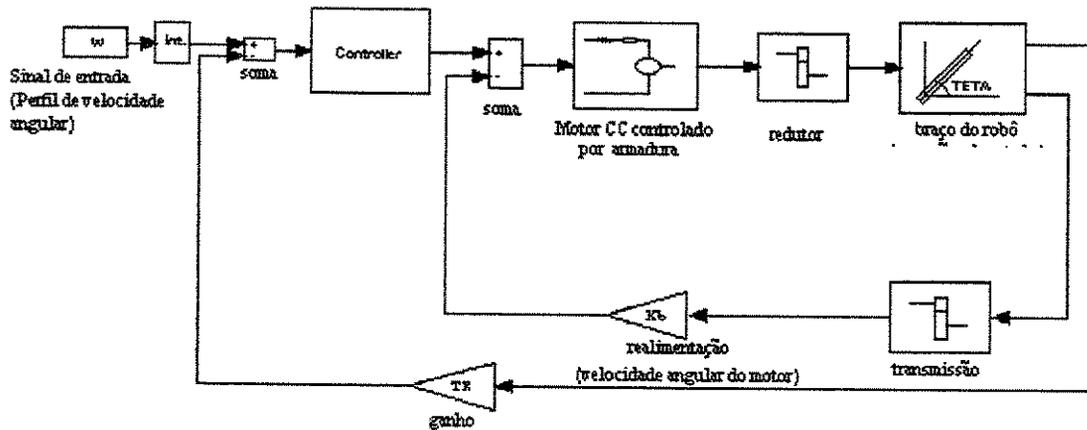


Figura 6.7: Estrutura Dinâmica de uma junta robótica considerando não linearidades.

O modelo dinâmico de um robô pode ser representado no esquema da figura 6.7. Com referência a esta figura, apresentaremos a descrição dos elementos como se segue:

**Bloco de geração de trajetória (W).** Este bloco é responsável pela geração da trajetória. Geralmente, o perfil de velocidade da trajetória é trapezoidal.

**Bloco soma.** O sinal de referência é comparado com o sinal de entrada e codificado por um transdutor de posição. O sinal de entrada trás a informação da posição instantânea do servomecanismo para o bloco soma e então um sinal de erro é gerado.

**Bloco do controlador.** O sinal de erro entra nesse bloco e uma ação de correção é gerada.

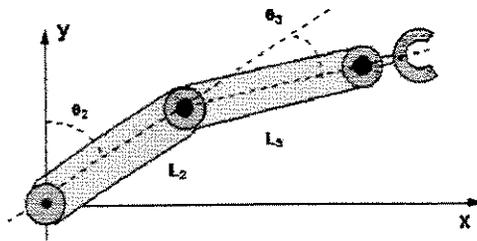
**Bloco do amplificador.** O sinal de saída do bloco do controlador tem uma pequena intensidade elétrica e é necessário amplificar e dimensioná-lo para que o mesmo tome-se compatível com o tipo de sistema utilizado.

**Bloco do atuador.** Este elemento é constituído por um motor CC controlado por armadura.

**Bloco de redução.** Uma caixa de engrenagens é colocada entre o atuador e o sistema a ser movimentado (braço do robô).

**Bloco dinâmico.** As equações dinâmicas de segunda ordem de um manipulador de dois graus de liberdade são obtidas usando as equações de Euler-Lagrange. O manipulador é considerado rígido ignorando possíveis torção ou flexão na estrutura.

Um esquema representativo de um manipulador de 2 braços é mostrado na figura 6.8.



**L<sub>2</sub>: Comprimento do braço 2**  
**L<sub>3</sub>: Comprimento do braço 3**  
**θ<sub>2</sub>: Deslocamento angular do braço 2**  
**θ<sub>3</sub>: Deslocamento angular do braço 3**

Figure 6.8: Esquema representativo de um manipulador de 2 braços.

As equações para essa articulação são:

$$\tau_2 = J_{22}\ddot{\theta}_2 + J_{23}\ddot{\theta}_3 + N_2(\theta, \dot{\theta}) + G_2(\theta) \quad (1)$$

para o braço 2, e

$$\tau_3 = J_{32}\ddot{\theta}_2 + J_{33}\ddot{\theta}_3 + N_3(\theta, \dot{\theta}) + G_3(\theta) \quad (2)$$

para o braço 3,

onde:

$$J_{22} = m_3 L_2^2 + \frac{m_3 L_3^2}{3} + \frac{m_2 L_2^2}{3} + m_3 L_2 L_3 \cos \theta_3$$

$$J_{23} = \frac{m_3 L_3^2}{3} + \frac{m_3 L_2 L_3}{2} \cos \theta_3$$

$$J_{33} = \frac{m_3 L_3^2}{3}$$

$$J_{32} = J_{23}$$

$$N_2(\theta, \dot{\theta}) = -m_3 L_2 L_3 \sin \theta_3 \left( \dot{\theta}_2 \dot{\theta}_3 + \frac{\dot{\theta}_3^2}{2} \right)$$

$$N_3(\theta, \dot{\theta}) = m_3 L_2 L_3 \sin \theta \frac{\dot{\theta}_2^2}{2}$$

$$G_2(\theta) = -\frac{1}{2} m_2 L_2 g + m_3 L_2 g \sin \theta_2 - \frac{1}{2} m_3 L_3 g \sin(\theta_2 + \theta_3)$$

$$G_3(\theta) = -\frac{1}{2}m_3L_3g \sin(\theta_2 + \theta_3)$$

Utilizando essas equações é possível simular os efeitos dos controladores linear e não-linear baseando no modelo geral descrito na figura 6.7.

#### 6.4.4 Simulação Numérica do Sistema Proposto

Utilizando as equações dinâmicas para o manipulador obtidas através do modelo matemático descrito anteriormente, são feitas simulações (Rosário,1998) utilizando controladores linear e não-linear com o propósito do controle de posição.

O modelo foi implementado de acordo com a figura 6.7 e simulado em SIMULINK (MATLAB™).

Baseada nessa simulações a eficiência do controlador será verificada. A possível eliminação de um indesejável efeito não-linear para o sistema, (por exemplo, limite de oscilação) usando controles não-lineares baseado em circuitos lógicos, será analisado.

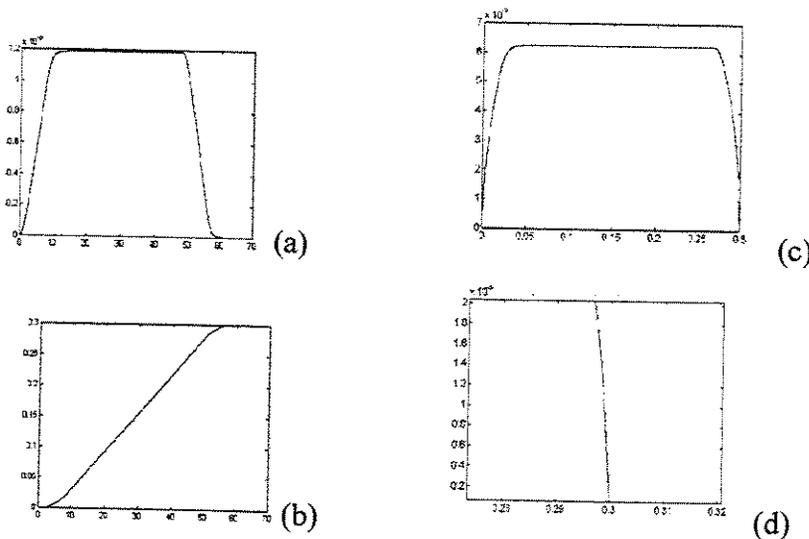


Figura 6.9: Controle linear - Proporcional (P): P=10

Para a ação de controle proporcional, os seguintes aspectos são observados:

(a)-Erro (rad×s). O erro tem oscilação dependendo do comportamento durante o regime transitório com a amplitude diminuindo na região das coordenadas desejadas.

(b)-Deslocamento (rad). o sistema permanece oscilando em amplitude constante.

(c)-Plano de fase (rad×rad/s). Um comportamento oscilatório é verificado na região transitória, principalmente na região de velocidade transitória.

(d)-Plano de fase (rad×rad/s). A amplitude de oscilação na resposta do servo sistema controlado pela ação proporcional tende a aumentar já que esta é uma função de sinal de erro obtida pela comparação entre o deslocamento desejado e o real.

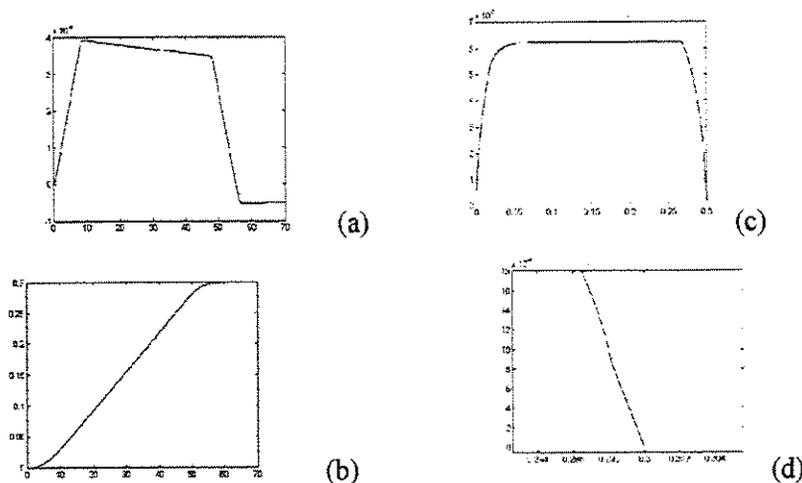


Figura 6.10: Controle linear – Proporcional e derivativo (PD): P=10, D= 10

Para a ação de controle proporcional e derivativo, os seguintes aspectos são observados:

(a)-Erro (rad×s). O erro tem oscilação mantida para um estágio inicial do perfil de trajetória, estabilizando no final de cada estágio, obtendo um erro nulo no final transitório.

(b)-Deslocamento (rad×s). O sistema tem oscilação mantida para o final do deslocamento, ocorrendo a redução do comprimento de onda, aumentando a frequência e subsequente estabilização depois de algum tempo.

(c)-Plano de fase (rad×rad/s). Devido ao fato de ser uma ação corretiva preditiva, a estabilização durante o processo transitório é observado, exceto algumas alterações no perfil da velocidade.

(d)-Plano de fase (rad×rad/s). é verificado a antecipação da estabilização no regime transitório comparado com o sistema controlado com a ação proporcional. Esta ação de antecipação sobre o erro atual gera uma ação de correção antecipada, tendendo ao aumento da estabilidade num curto espaço de tempo.

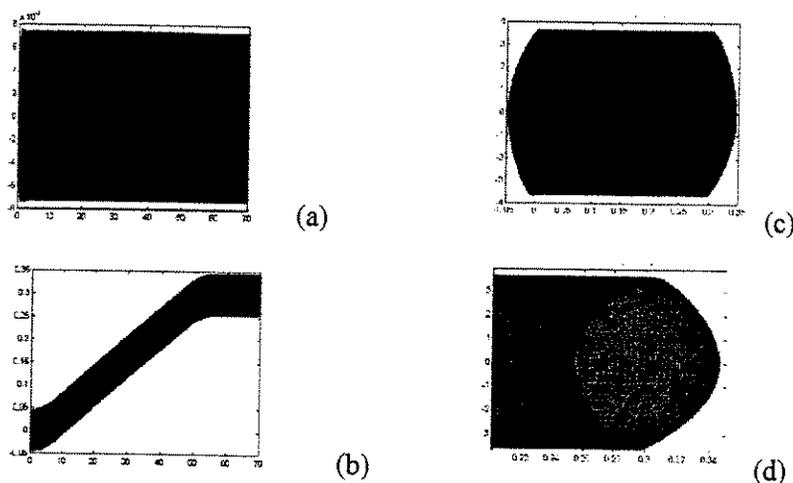


Figura 6.11: Controle não-linear – controle On-Off (Relay):  $g_e=30v$

Para a ação de controle não-linear On-Off, os seguintes aspectos são observados:

(a)-Erro (rad×s). O erro tem comportamento ondulatorio durante todo o processo.

(b)-Deslocamento (rad). o sistema tem oscilação permanente devido a amplitude não variável devido ao fenômeno de oscilação perpétua.

(c)-Plano de fase (rad×rad/s). A curva apresentado na figura confirma que a implementação On-Off gerando um regime fixo e permanente.

(d)-Plano de fase (rad×rad/s). O plano de fase amplificado apresenta com detalhes o fenômeno da oscilação perpétua.

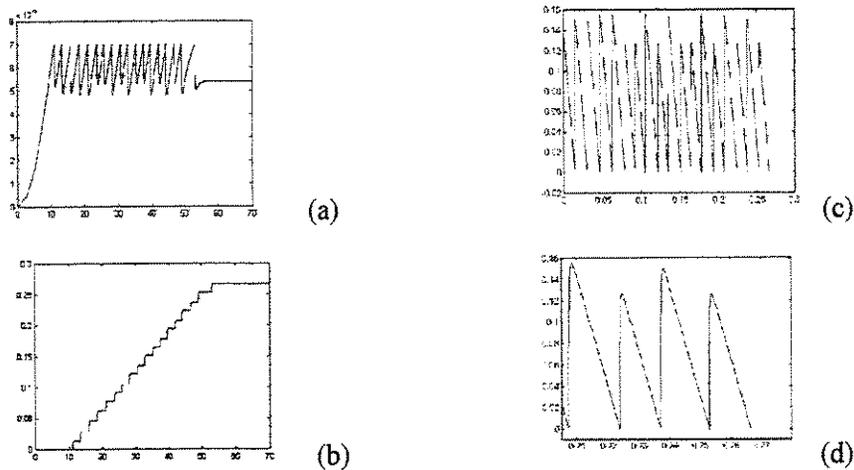


Figura 6.12: Controle não-linear Off (Relay)+Zona-Morta:  $g_e = 30v + z.m. = 0.007$

Para a ação de controle não-linear On-Off+Zona-Morta, os seguintes aspectos são observados:

(a)-Erro (rad×s). O erro tem comportamento oscilatório mantido durante todo o processo e o fenômeno de oscilação perpétuo dentro de um ponto tende a desaparecer. A posição posterior tem uma imprecisão devido a razão direta com a largura da zona-morta.

(b)-Deslocamento (rad). O sistema para de vibrar, ocorrendo estabilização. Com a implementação da zona-morta ocorre a eliminação dos efeitos prejudiciais.

(c)- Plano de fase (rad×rad/s). A característica de um sistema controlado pela ação on-off +zona morta é reduzido em comparação com o controle on-off, onde a estabilização é proporcional a largura da banda de implementação da zona-morta.

(d)- Plano de fase (rad×rad/s). A amplitude do plano de fase apresentado na estabilização do sistema, mostrando que a incorporação da zona-morta no controle on-off elimina a oscilação perpétua.

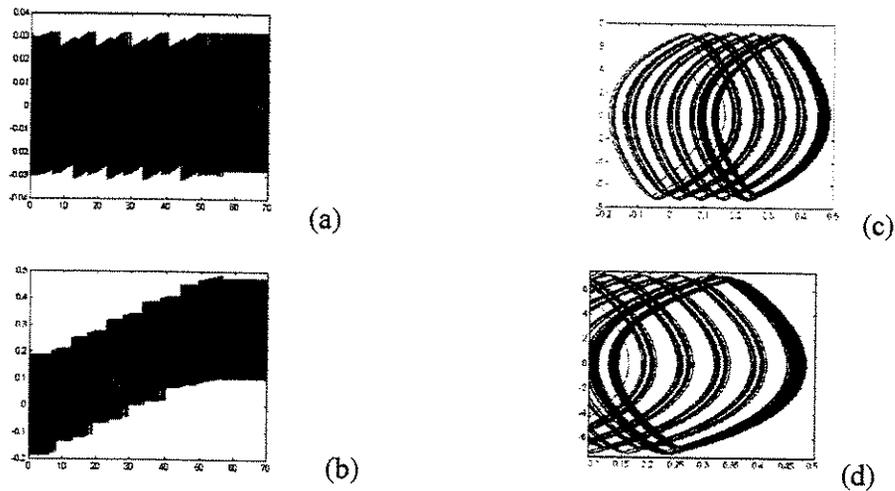


Figura 6.13: Controle não-linear On-Off (Relay)+ Histerese:  $g_e = 30v + B = 0.009$

Para a ação de controle não-linear On-Off+Histerese, os seguintes aspectos são observados:

(a)-Erro (rad×s). O erro tem oscilação forçada durante todo o processo, assumindo uma banda de erro na amplitude máxima com relação ao caso ideal on-off.

(b)-Deslocamento (rad). O sistema tem oscilação permanente de amplitude invariada, entretanto, a amplitude da resposta é maior que a mesma do caso on-off.

(c)- Plano de fase (rad×rad/s). A implementação da histeres no controle on-off leva ao aumento da amplitude de oscilação.

(d)- Plano de fase (rad×rad/s). O espaçamento maior entre os círculos indica a redução da frequência de oscilação devido ao efeito do aumento da faixa de vibração.

#### 6.4.5 Análise dos Resultados

Com as ações de controles linear e não-linear implementadas no modelo, obtivemos a confirmação dos resultados teóricos de ações de correção não-linear descrita pelo método da função descritiva (Gille, Pélegrin, 1988), em uma simulação numérica.

Foi possível verificar a ação dos efeitos dos controles linear e não-linear sendo verificado características peculiar de cada um. O controle do servo sistema feito por on-off (figuras 6.11 (a) e (b)), mostra o fenômeno de oscilação limite no final do regime transitório, o que possibilitou as mudanças pela inserção da histerese (figuras 6.13 (a) e (b)). Este último efeito aumenta a amplitude e diminui a frequência de oscilação em relação ao valor médio, o qual pode reduzir ou eliminar as características de oscilação do processo com a variação da zona-morta (figuras 6.12 (a) e(b)), quando é implementado ao servo sistema.

Analisando os planos de fase, pode-se verificar que o controle linear com ação proporcional gera um comportamento oscilatório, e essas oscilações são aumentadas na medida que o erro aumenta (figuras 6.9 (c) e (d)). No caso do controle PD, pode-se verificar um comportamento sub amortecido com convergência a um foco estável no final do processo (Figuras 6.10 (c) e (d). De acordo com a ação derivativa e proporcional, ocorrendo a antecipação da estabilização.

Nos casos não lineares ocorre o ciclo limite (figura 6.11 (a) e (b)). Adicionando a zona morta no controlador pode-se verificar a redução desse fenômeno (figuras 6.12 (c) e (d)).

Inserindo a histeresi no on-off pode-se verificar um maior espaçamento entre os ciclos indicando a redução da frequência de vibração.

Comparando a ação dos controles linear e não linear, pode-se fazer uma analogia entre os dois tipos de controladores.

A ação on-off tem um comportamento equivalente a ação proporcional (figuras 6.9 (a) e (b)); a ação zona morta tem um efeito equivalente a ação derivativa (figuras 6.11 (a) e (b)) e a histerese tem um efeito equivalente a ação integral (figuras 6.10 (a) e (b)).

## **6.5 Conclusão**

Neste capítulo foram apresentados aspectos conceituais, abordando controladores não lineares, e a implementação computacional com simulação do comportamento do modelo de uma junta robótica considerando as suas não linearidades.

No próximo capítulo será proposto uma plataforma experimental para validação de uma junta robótica com ênfase em controladores não lineares, utilizando componente lógicos reprogramáveis EPLD.

## Capítulo 7

### **Proposição de Bancada Experimental para Validação e Testes de Junta Robóticas**

Nos últimos anos, os trabalhos desenvolvidos dentro do Laboratório de Automação Integrada e Robótica (LAR) da Unicamp, têm apresentado um forte componente relativo a validação experimental e que tem proporcionado um aumento a sua infra-estrutura, possibilitando assim o desenvolvimento de novos trabalhos de caráter experimental.

Trabalhos experimentais na área de robótica, geralmente vem acompanhados de montagens complexas que nem sempre são possíveis de serem implementadas, principalmente devido ao alto custo envolvido. Daí, a procura de soluções adequadas e ao mesmo tempo em que possam expressar, sem perdas de generalidades, um dado modelo de um sistema. Essa é uma busca constante, com o intuito da validação experimental, o que tem funcionado como um elemento motivador.

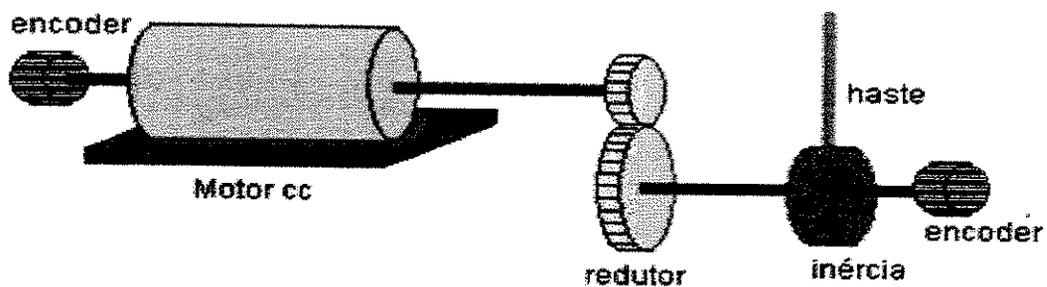
Dentro deste contexto apresentamos duas bancadas para validação e teste de juntas robóticas, destacando suas principais características. Ao mesmo tempo foi desenvolvida uma infra-estrutura (hardware dedicado e software) para aquisição, processamento e controle, além de outros programas dedicados, desenvolvidos para este fim.

#### **7.1 Plataforma Experimental para implementação de um controlador de posição**

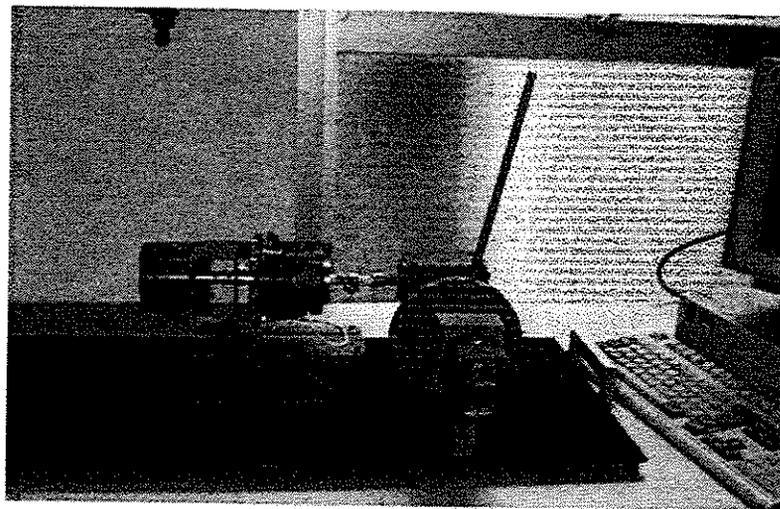
Durante o desenvolvimento desse trabalho de pesquisa foram avaliadas duas bancadas experimentais, descritas a seguir.

### 7.1.1 Motor – Redutor – Inércia

A primeira plataforma implementada no Laboratório de Automação Integrada e Robótica da UNICAMP é constituída de um motor elétrico de corrente contínua acoplado a um disco de inércia com uma haste presa a este disco, de maneira a simular um grau de liberdade de um robô, conforme mostra a figura 7.1.



a) Representação Esquemática da Plataforma Experimental



b) Plataforma Implementada.

Figura 7.1: Bancada experimental para simulação de junta robótica sujeita a perturbações.

O acoplamento entre o disco de inércia e o motor é feito por intermédio de uma caixa de redução por engrenagens, e para medida de posição/velocidade são utilizados dois *encoders* de

posição, um no eixo do motor e o outro no eixo da carga. Esta plataforma experimental possui as seguintes características:

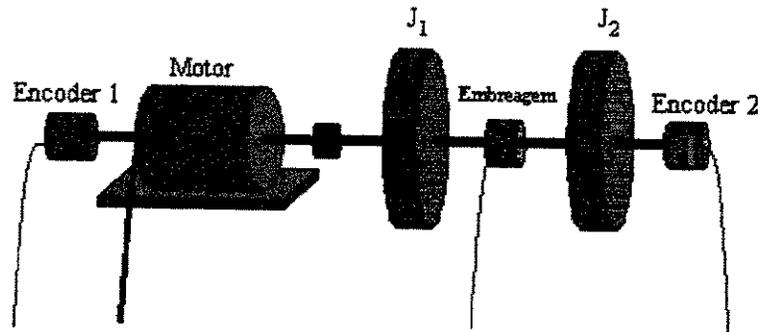
- i) Simples Construção Mecânica realizada a partir de componentes comerciais de baixo custo.
- ii) Possibilidade de simular um dos graus de liberdade de um robô sem acoplamento dinâmico entre as juntas, permitindo a realização experimental de um número de experimentos de grande interesse tanto do ponto de vista científico quanto acadêmico.
- iii) Necessidade de realizar o trabalho de pesquisa na área de identificação dos parâmetros deste sistema, tendo em vista o desconhecimento dos parâmetros desses componentes.
- iv) Necessidade de realizar ajustes ainda necessários no sistema, tais como alinhamentos entre motor, redutor, sensor e carga.
- v) O redutor, apresenta folgas e estas podem comprometer o desempenho de uma estratégia de controle.

### **7.1.2 Motor acoplado a uma Inércia Variável**

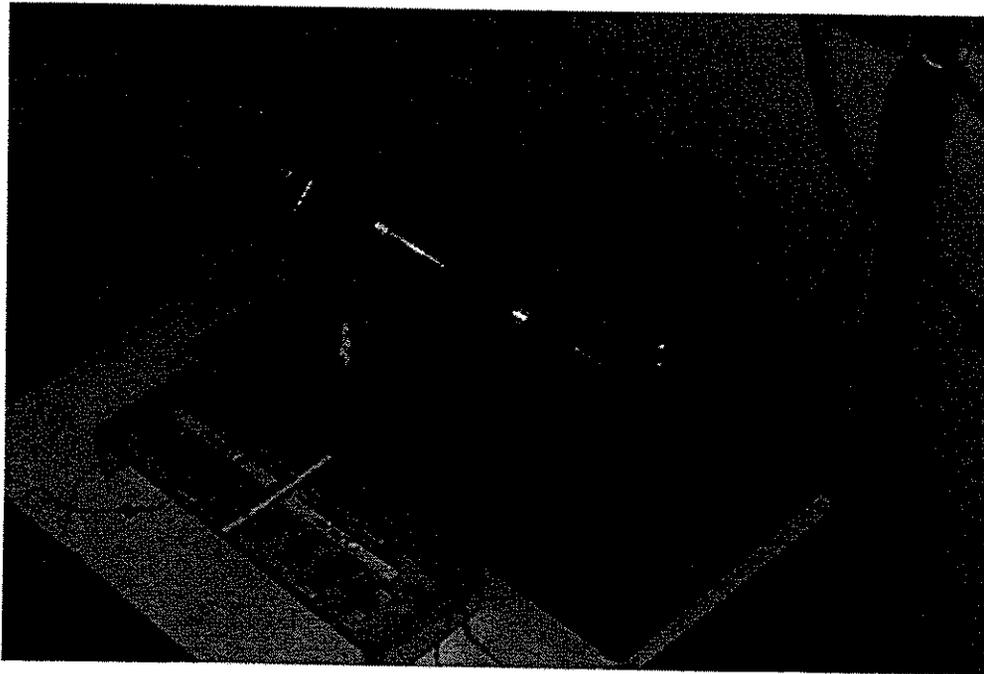
No desenvolvimento da segunda bancada experimental, procurou-se evitar questões relativas a identificação de parâmetros, utilizando componentes cujos parâmetros fossem conhecidos.

Estes problemas acarretariam em dificuldades adicionais na validação experimental do projeto de pesquisa, tais como estudo e desenvolvimento de métodos e procedimentos experimentais de identificação de parâmetros do sistema experimental, fugindo totalmente do escopo inicial da pesquisa, que era o projeto e implementação de um controlador de posição baseado no Modelo Dinâmico da junta robótica. Assim, optamos pela implementação de uma nova bancada experimental, onde todos os parâmetros fossem conhecidos, introduzindo somente efeitos de mudanças inerciais de uma junta robótica ao longo de uma trajetória, efeito este mais preponderante em aplicações industriais.

A bancada experimental apresentada na figura 7.2, foi implementada com o propósito de simular um grau de liberdade de um robô, com a possibilidade de introduzir efeitos de mudanças inerciais ao longo de uma trajetória. Desta forma, estaríamos avaliando o desempenho de uma dada técnica de controle, sujeita a perturbações (acoplamento de outras juntas).



a) Esquema Representativo



b) Plataforma Experimental proposta

Figura 7.2: Plataforma Experimental Proposta Motor – Inércia Variável.

Esta bancada experimental é constituída de um motor de corrente contínua acoplado a dois discos de inércia e sensores incrementais de posição. O primeiro disco de inércia solidário foi acoplado ao eixo do motor, enquanto que o acoplamento do segundo disco de inércia é realizado

através de uma embreagem eletromagnética, que é utilizada para acoplar ou não uma nova carga ao motor, possibilitando assim, variarmos a carga inercial no sistema (efeito similar a uma mudança de configuração de um robô numa determinada trajetória).

Para validação experimental deste trabalho, foi utilizada esta bancada experimental que permitiu a aquisição de dados precisos e a variação da carga é feita através de uma embreagem eletromagnética. Esta embreagem é acionada através de um sinal de comando lógico e uma vez acionada, a mesma acopla a segunda carga ao sistema.

A figura 7.3 apresenta uma representação esquemática da Plataforma Experimental (Operativa e Comando), e no Anexo I são apresentados uma descrição detalhada dos principais elementos constituintes dessa bancada experimental.

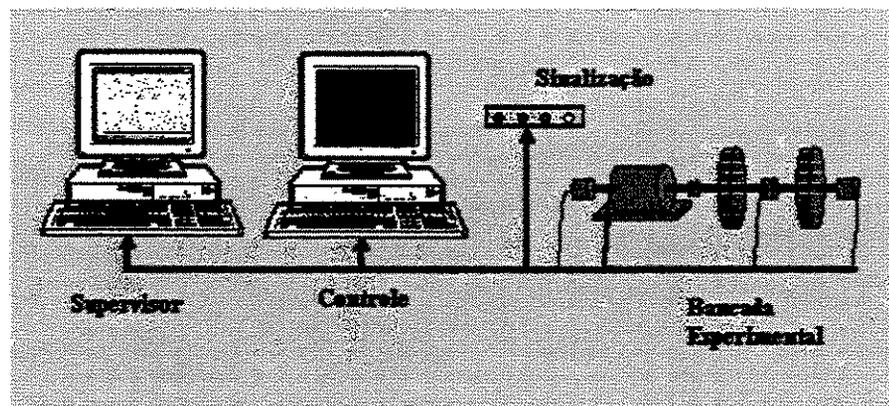


Figura 7.3: Sistema de Supervisão e Controle.

Através dessa variação da carga temos uma perturbação ao sistema, o que nos permitirá verificar o comportamento do controlador devido a essa perturbação. A velocidade inicial desta carga induz uma perturbação sobre o sistema. Após um certo tempo a embreagem não desliza mais e a velocidade da segunda inércia é então igual a do motor. Podemos considerar este torque como sendo uma perturbação relativa a mudança de parâmetros do sistema.

Para identificarmos precisamente o comportamento dinâmico da junta no eixo do motor e da carga, e influência do acoplamento e folgas no sistema, serão utilizados dois encoders incrementais, um no eixo da carga (após o redutor) e outro no eixo do motor.

## 7.2 Considerações sobre a Maquete Experimental utilizada

### 7.2.1 Estudo do Acoplamento Inercial

Considerando nula a elasticidade do sistema, fisicamente o acoplamento impõe a conservação da quantidade de movimento de todo o sistema, com perda de energia cinética, ligada ao deslizamento inicial da embreagem. Entretanto, no momento em que a embreagem é acionada, não há modificação da inércia do motor, não produzindo assim perturbações sobre o sistema e nem mudança de velocidade ou perda de energia.

O torque fornecido pela embreagem deve ser suficientemente elevado para evitar o deslizamento durante o acoplamento, mesmo assim, esta ação introduz uma não linearidade sobre o sistema. Para evitar este deslizamento é necessário verificar:

$$T_{emb} \geq T_{mot} \frac{J_2}{J_{sistema}} \quad 7.1$$

onde:

$T_{emb}$  – torque aplicado pela embreagem

$T_{mot}$  – torque no eixo do motor;

$J_2$  – inércia da segunda carga;

$J_{sistema}$  – inércia do sistema (motor + carga 1).

Em resumo, a bancada proposta, aproxima muito a um grau de liberdade de um robô, com uma pequena diferença: pelo fato da segunda carga entrar no sistema quase de forma instantânea. No caso de um robô, essa variação é feita de forma contínua no tempo. A embreagem é acionada através de comando elétrico e, uma vez acionada, a mesma acopla a segunda carga ao sistema. Com essa variação da carga temos uma perturbação ao sistema, o que nos permitiram verificar o comportamento do controlador.

Sob estes aspectos, esta bancada se apresenta, de uma forma geral, como uma estrutura interessante para validação de diferentes algoritmos de controle.

### 7.2.2 Implementação de Hardware Tratamento de Informações e Controle

Para tratamento das informações provenientes dos sensores de posição das juntas e implementação do controlador não linear e controlador clássico PID foram utilizados componentes lógicos reprogramáveis (EPLD's) no ambiente ALTERA™. Esse ambiente possibilitou o desenvolvimento de algoritmos de controle usando bibliotecas próprias e até mesmo o desenvolvimento de programas e inclui-los dentro da malha de controle do sistema.

A principal vantagem dessa implementação foi o fato de implementarmos a malha de controle independente do micro-computador utilizado como sistema de Supervisão e Controle. Isso nos forneceu uma total liberdade de aplicação já que não iremos depender por exemplo do *clock* do micro-computador para execução do programa.

### 7.2.3 Implementação de Sistema de Supervisão e Controle

Para validação experimental do experimento proposto foi realizado inicialmente o processo de identificação dos parâmetros reais do sistema, através da implementação de um sistema de aquisição e controle. Isto tornou possível a sistematização dos procedimentos experimentais, possibilitando avaliarmos quantitativamente a nova estrutura de controle proposta. A figura 7.4 apresenta uma representação esquemática da infra-estrutura experimental proposta.

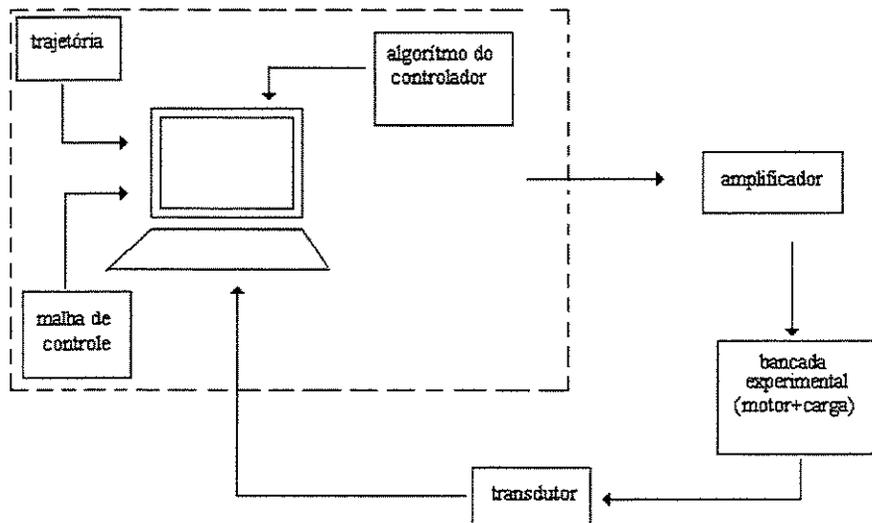


Figura 7.4: Esquema representativo do processo de aquisição e controle proposto.

O sistema é constituído de dois micro-computadores, um responsável pela aquisição das informações provenientes dos transdutores de posição da bancada experimental (após tratamento em ALTERA™); e o outro responsável pela geração de sinais de referência (ou trajetórias) para o controlador. Isto permite uma análise de erro do comportamento da trajetória de referência depois de comparada com a posição final da carga medida em relação ao eixo do motor e em relação à carga.

A geração de uma trajetória é realizada através de um programa computacional para o envio de arquivos de pontos (coordenadas de juntas) para o controlador de posição de uma determinada junta em hardware dedicado desenvolvido em lógica reprogramável utilizando o sistema ALTERA™.

#### **7.2.4 Modelagem e Identificação de Parâmetros**

Durante o desenvolvimento desse trabalho foram realizados estudos complementares para a obtenção dos parâmetros da Plataforma Experimental utilizada, entre eles podemos referenciar:

- Obtenção dos parâmetros do motor (inércia do redutor, indutância).
- Obtenção dos parâmetros do redutor e da carga inercial (momento de inércia, atrito).
- Calculo dos ganhos do controlador PID (método de CHR).

### **7.3 Projeto e Implementação do Controlador de Posição**

#### **7.3.1 O Sistema de Desenvolvimento ALTERA™**

Os componentes PLD (Programmable Logic Device) vêm se mostrando nos últimos anos, como uma valiosa ferramenta para validação de projetos de *hardware* (Lima, 2001). A sua velocidade, capacidade de integração, flexibilidade e facilidade de programação apresentam uma solução atrativa também no projeto de controladores.

Uma das principais propostas dessa tese de doutoramento refere-se ao desenvolvimento de um controlador de posição para uma junta robótica utilizando um PLD. Uma das vantagens desse desenvolvimento foi o baixo custo de implementação e não necessidade de utilização de conversores Digitais-Analógicos (DA) e Analógicos-Digitais (AD).

Para a implementação em PLD foi utilizado um circuito integrado EPM7128SLC84-7 da FAMILIA MAX7000. O desenvolvimento do projeto foi realizado através do sistema de desenvolvimento Altera MAX+PLUS II (\*), com uma interface de programação gráfica, onde as principais razões de escolha desse sistema foram:

- A utilização deste sistema foi realizada através de um programa de incentivo educacional (Altera University Program), que permite a utilização deste ambiente de desenvolvimento com custo relativamente baixo.
- Possibilidade de execução do projeto em linguagem gráfica (esquemáticos), VHDL (VHSIC Hardware Description Language) e AHDL (Altera Hardware Description Language).
- Possibilidade de utilização de variados tipos de encapsulamentos para o projeto da interface em placa de circuito impresso: PLCC, PGA, PQFP, RQFP e TQFP.
- Opção variada de famílias PLD, permitindo a escolha do dispositivo que melhor se adapte ao projeto.
- Possibilidade de criação de bibliotecas dedicadas, e ambiente estruturado de programação.
- O sistema de desenvolvimento pode ser executado em plataformas do tipo IBM-PC e estações de trabalho, permitindo a simulação, análise de diagramas de tempo e programação direta do dispositivo.

### 7.3.2 Proposta de Desenvolvimento do Controlador de Posição

Esta interface tem como principal objetivo, o controle de um motor de corrente contínua acionado por um “*driver*” de potência linear ou PWM. O controlador programável é capaz de processar diretamente os sinais digitais oriundos de um *encoder* acoplado ao eixo do motor e de um sinal digital de controle, correspondente a trajetória de uma junta robótica.

Na figura 7.5 é mostrada uma representação esquemática da estrutura do sistema de controle implementado neste trabalho.

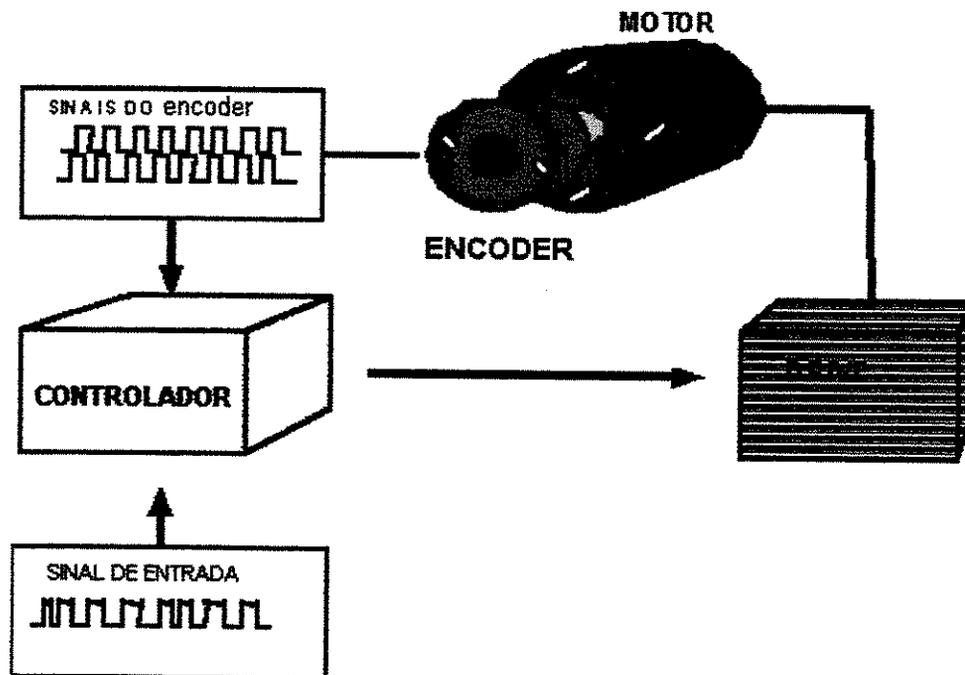


Figura 7.5: Representação de blocos do sistema proposto.

Na implementação experimental desse projeto foram utilizados dois procedimentos para geração do sinal de saída do controlador:

- i) Geração de uma palavra binária de 16 bits que será enviada a um conversor DA, que por sua vez gera uma tensão de controle para o módulo de potência.

- ii) Geração de um sinal digital para o módulo de potência com controle por largura de pulso (PWM). Neste caso, nenhum conversor AD ou DA é necessário na malha de controle.

### **7.3.3 Descrição dos Sinais de Entrada/Saída da Interface Implementada**

#### **7.3.3.1 TRAJETÓRIA**

*Trajectoria* é um sinal digital (trem de pulsos) referente à entrada de comando para o sistema de acionamento. Ele apresenta as seguintes características:

- Este sinal tem associado ao seu período a velocidade com que a trajetória é executada.
- O número de períodos (ou de pulsos) está associado ao deslocamento executado nesta trajetória.
- São definidos os limites práticos para os sinais gerados para a trajetória:
  - Valor máximo em frequência: 1,2 kHz (0,7843 ms).
  - Valor mínimo em frequência maior que zero: 5 Hz (200 ms).

#### **7.3.3.2 ENCODER**

*Encoder* é um sinal digital (trem de pulsos) proveniente de um transdutor de posição acoplado no eixo do motor (encoder incremental). Ele apresenta as seguintes características:

- Este sinal tem associado ao seu período a rotação do motor. O número de períodos (ou de pulsos) está associado ao deslocamento angular do eixo do motor.
- São definidos os limites práticos para os sinais provenientes deste transdutor:

- Valor máximo em frequência: 1,2 kHz (0,7843 ms).
- Valor mínimo em frequência maior que zero: 5 Hz (200 ms).

### 7.3.3.3 CLOCK

*Clock* é um sinal digital responsável pela taxa de incremento dos contadores internos à PLD. Considerando os limites estabelecidos para os sinais de entrada, bem como a resolução de 16 bits é suficiente um CLOCK de frequência de 50 kHz.

A saída do sistema é convertida para um sinal de controle de nível 0-5V através de um conversor D/A. Este sinal é enviado à interface de controle do motor.

### 7.3.4 Exemplo de Aplicação

Para exemplificar o comportamento do sistema são apresentados na figura 7.6 exemplos típicos de sinais de entrada em duas situações para a detecção de erro.

Na figura 7.6a, o motor está sendo acelerado, com o sinal ENCODER apresentando um período maior que o sinal TRAJETÓRIA. Já na figura 7.6b, o motor está desacelerando com o sinal ENCODER apresentando um período menor que o sinal TRAJETÓRIA.

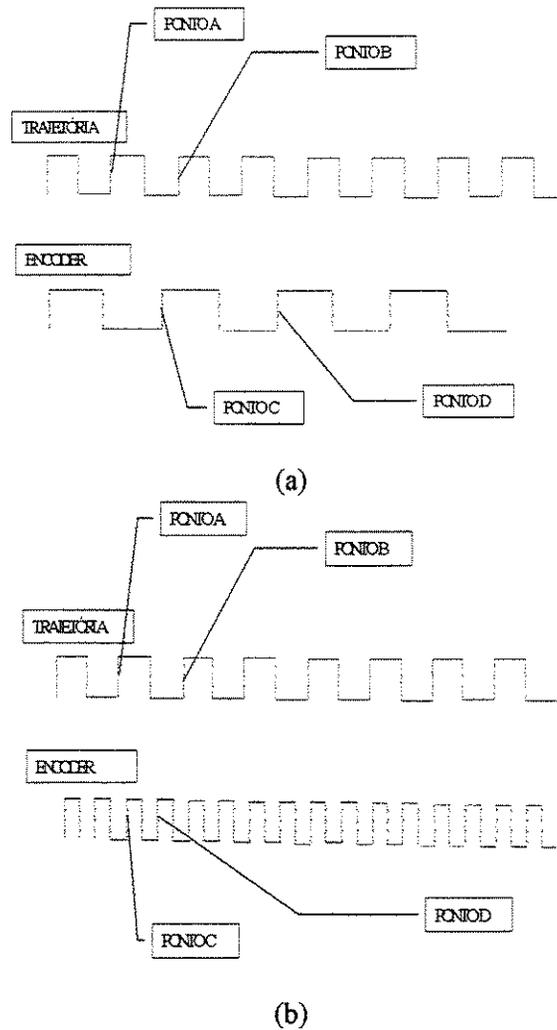


Figura 7.6: Exemplos típicos de sinais entradas no controlador de posição proposto.

#### 7.4 Ambiente de Simulação

O ambiente de simulação do sistema de desenvolvimento Altera permite o projeto dos sinais de entrada e conseqüente visualização dos sinais de saída simulados. Esta simulação é bastante sofisticada, permitindo que sejam considerados tempos de *setup* e *hold*, *delay* e *glitch*. Na figura 7.7 é apresentado um diagrama de tempos extraído de uma tela de simulação gráfica, onde o contador apresenta resultados convergindo para zero a medida que os períodos dos sinais comparados se aproximam.

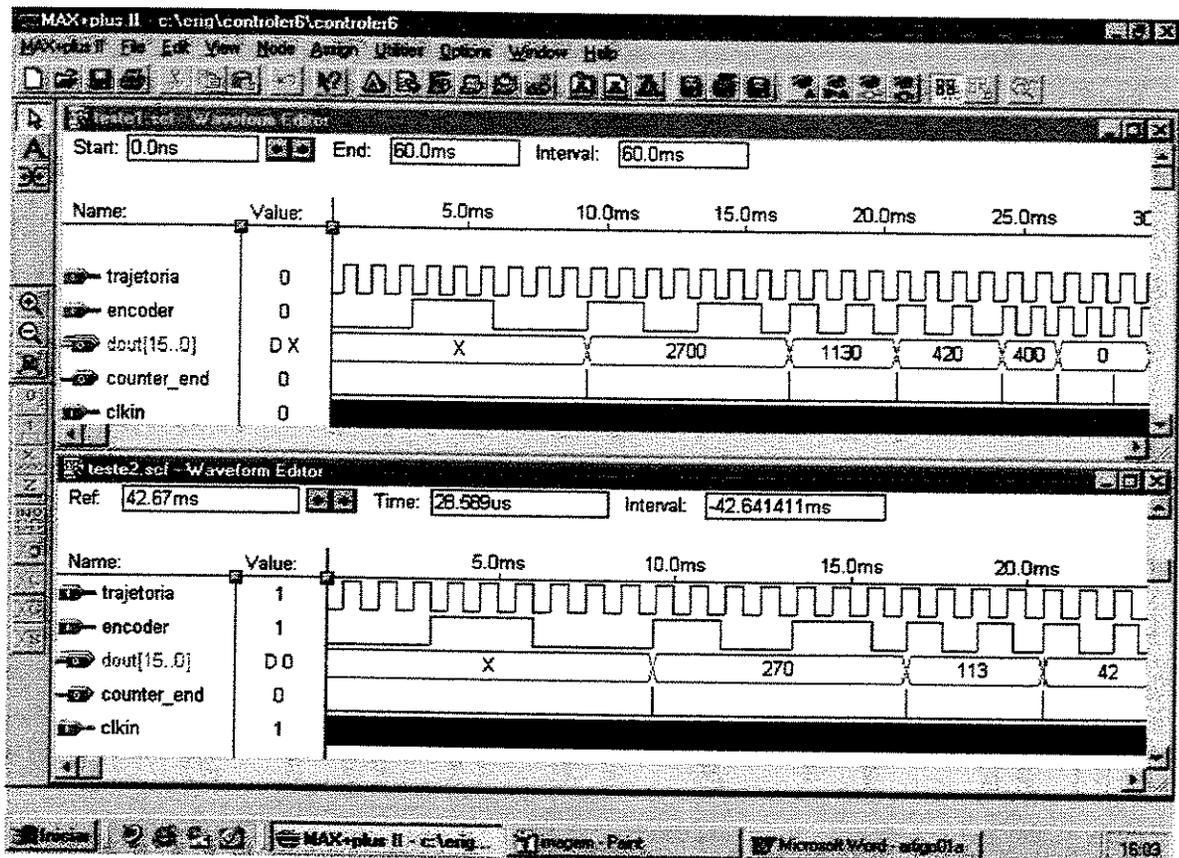


Figura 7.7: Diagrama de tempos enfatizando a saída do contador de erro em função dos sinais de TRAJETÓRIA e ENCODER

## 7.5 Protótipo experimental implementado

Para a implementação do controlador de posição da junta robótica descrita neste capítulo foi desenvolvido no Laboratório de Automação Integrada e Robótica da Faculdade de Engenharia Mecânica da UNICAMP um protótipo experimental para implementação dos circuitos lógicos reprogramáveis ALTERA™ (figura 7.8). A utilização desses componentes permitiu a alta capacidade de integração de recursos graças a sua arquitetura aberta e estruturada de programação, a partir da implementação de dois módulos.

### 7.5.1 Módulo de Aquisição e Tratamento de Informações

O módulo de Aquisição de sinais e tratamento de informações do sensor de posição: a partir do tratamento dos sinais gerados pelo *encoder* foi realizado um circuito lógico utilizando lógica

reprogramável, que permitiu a obtenção do sentido de rotação e frequência. A partir de um circuito de contagem/decontagem essas informações foram enviadas a uma interface I/O de um PC, permitindo assim um tratamento posterior dessas informações (análise quantitativa dos resultados).

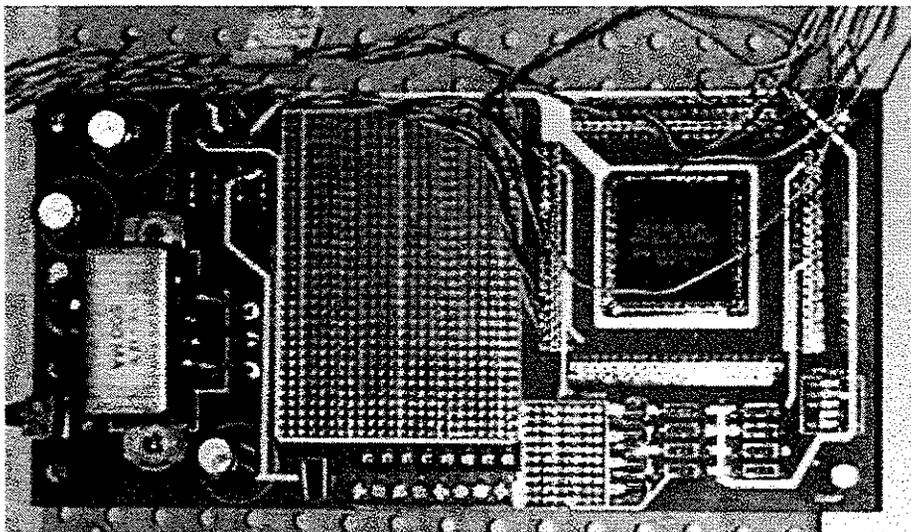
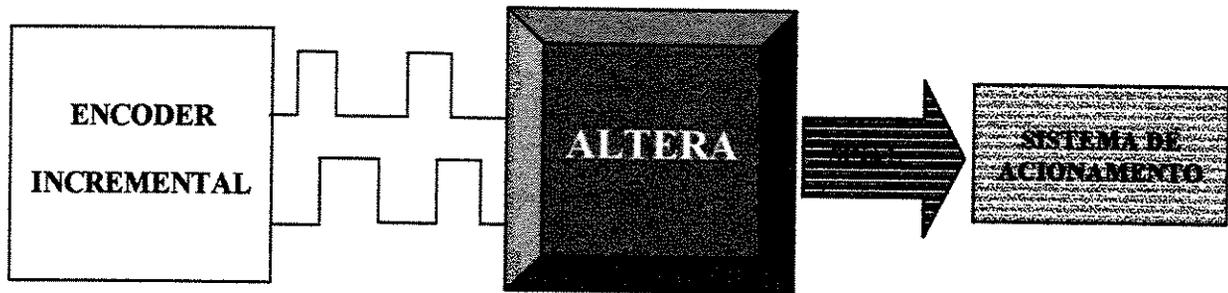


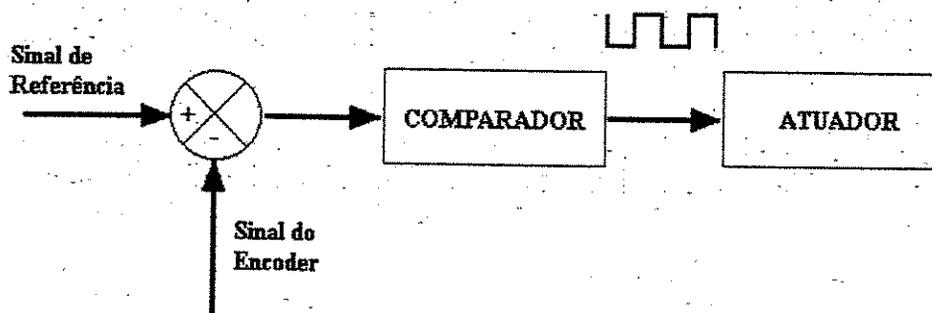
Figura 7.8: Interface desenvolvida em Lógica Reprogramável.

### 7.5.2 Módulo de Controle de Posição

O Módulo de controle de posição foi implementado em lógica reprogramável em um controle de posição proporcional que permitisse a comparação de sinais digitais de referencia provenientes de um PC, com a posição angular da junta medida através de um *encoder* incremental (modulo de tratamento das informações descrito anteriormente) e em função do erro de posição da junta, um sinal sob a forma de pulsos era enviado ao amplificador de potência do sistema de acionamento do motor (figura 7.9).



a) Diagrama esquemático do Circuito de Acionamento e Controle



b) Lógica de Controle programada em ALTERA.

Figura 7.9: Controle de Posição Proporcional de junta

## 7.6 – Conclusão

Neste capítulo foram apresentadas duas bancadas implementadas no LAIR-UNICAMP para validação e teste de juntas robóticas, destacando suas principais características. Ao mesmo tempo foi apresentada uma infra-estrutura (hardware dedicado e software) para aquisição, processamento e controle, além de outros programas dedicados desenvolvidos para este fim.

Foram apresentados os detalhes relativos à implementação experimental no Laboratório de Automação Integrada e Robótica da UNICAMP, utilizando o Sistema de Desenvolvimento ALTERA™, foram colocados em anexos:

- Anexo I: Descrição detalhada dos elementos constituinte da Plataforma Experimental.
- Anexo II: Descrição da Estrutura do Sistema de Supervisão e Controle de um Manipulador Robótico.
- Anexo III: Descrição detalhada da implementação detalhado do Sistema de Supervisão e Controle.
- Anexo IV: Descrição detalhada do Sistema de Aquisição, Tratamento e Controle de Informação provenientes do encoder utilizando circuitos lógicos reprogramáveis.
- Anexo V: Implementação detalhada do circuito de Acionamento e Controle de um motor de corrente contínua.
- Anexo VI: Cálculo do momento de inércia.

Finalizando, os procedimentos apresentados nesse capítulo permitiram a simulação de uma junta robótica, a validação e testes do sistema de aquisição, processamento e controle implementado, fundamentais para a conclusão dessa tese de doutoramento apresentada no capítulo seguinte.

## Capítulo 8

### Conclusões e perspectivas futuras

Este programa de doutoramento desenvolvido no Laboratório de Automação Integrada e Robótica apresenta uma proposta de implementação de um sistema de controle didático aplicado a robôs industriais baseado na modelagem dinâmica.

O projeto de pesquisa teve como objetivo a modelagem e simulação de sistemas dinâmicos com ênfase no estudo e implementação de controladores de posição de juntas robóticas. Para isso foi realizado a estruturação de um sistema de controle de um manipulador robótico, implementando um novo sistema de controle de posição de arquitetura “aberta” para a realização de tarefas complementares automatizadas a partir do conhecimento do ambiente de atuação e tipo de tarefa a ser realizada, e assim investigar a utilização de diferentes tipos de estrutura de controle.

O método de trabalho consistiu num estudo inicial de modelagem dinâmica de um robô, analisando à partir de simulações numéricas o comportamento do sistema em presença de perturbações, usando um controlador clássico PID e controladores não-lineares.

Para atingir esse propósito foi desenvolvido um controlador, utilizando duas bancadas experimentais, para a realização de testes visando simular um grau de liberdade de um robô, possibilitando introduzir efeitos de mudanças inerciais ao longo de uma trajetória e, dessa forma, avaliar o desempenho de uma dada técnica de controle sujeita as perturbações do sistema. Esta etapa do trabalho permitiu a aquisição de dados precisos e a variação da carga. Assim foi possível

analisar a perturbação do sistema e verificar o comportamento do controlador diante dessas perturbações.

No tratamento das informações provenientes dos sensores de posição das juntas e implementação do controlador não linear e controlador clássico PID, foram utilizados componentes lógicos reprogramáveis (EPLD's) no ambiente ALTERA™, o que favoreceu no desenvolvimento de algoritmos de controle usando bibliotecas próprias. Bem como no desenvolvimento de programas, sendo possível implementar equações, e inclui-los dentro da malha de controle do sistema. Isso contribuiu para implementar a malha de controle independente do micro-computador, utilizando como sistema de Supervisão e Controle, tornando possível a sistematização dos procedimentos experimentais e assim avaliar quantitativamente a nova estrutura de controle proposta.

Além dessas contribuições, também podemos citar o desenvolvimento de um controlador de posição para uma junta robótica utilizando um PLD, o qual destacamos a vantagem do seu baixo custo de implementação e a não necessidade de utilização de conversores digitais-analógicos (DA) e analógicos-digitais (AD).

Dessa forma, a proposta de implementação de um sistema de controle de um manipulador foi realizada tendo como base na modelagem dinâmica. Para isso foi utilizado um controlador clássico do tipo PID, implementado a partir de simulações que verificaram o comportamento do sistema em presença de perturbações, atingindo assim aos objetivos inicialmente propostos dessa tese de doutoramento.

Esse trabalho proporcionou a implementação de um sistema de controle didático baseado na modelagem dinâmica, que poderá ser utilizado em outras aplicações, como por exemplo a implementação do controlador em estruturas de dinâmica variável e em estruturas flexíveis.

## Referências Bibliográficas

- **Craig, J. J.** [1986]. "Introduction to Robotics Mechanics & Control" . Addison - Wesley Publishing Company, Inc.
- **David, S. A.** [1996]. "Modelagem, simulação e controle de robôs flexíveis". Tese de Mestrado, Faculdade de Engenharia Mecânica, Unicamp.
- **Eletro-Craft Corp.** [1972]. "DC Motors, Speed Controls, Servo Systems".
- **Fu, K. S., et al** [1987]. "Robotics : Control, Sensing, Vision and Intelligence". McGraw - Hill, Inc.
- **Gille, J.C., P.DECANE, M.PÉLEGRIN,** [1988]. "Systems Asservis Non Linears". Dunod, 5a Edition, France.
- **Hermi, H. A.** [1996]. "Uma Contribuição ao Estudo de Não-Linearidade na Modelagem e Controle de Manipuladores". Tese de Mestrado, Faculdade de Engenharia Mecânica, Unicamp.
- **Hervella, C.** [1995]. "Projeto e Desenvolvimento de um Controlador Lógico Programável Flexível para Controle de Manipuladores e Robôs Industriais". Tese de Mestrado, Faculdade de Engenharia Mecânica, Unicamp.
- **Kuo, B. C.** [1995]. "Automatic Control Systems". Prentice-Hall, Inc.

- **Lima; C.R.E., Rosário; J. M., Dumur, D. [2001].** “Open Architecture Design for Mobile Robotics”, Canela, RS, IAD’2001, 2nd IFAC Workshop on Intelligent Assembly and Disassembly, 5-7 November 2001, 6 p., publicado nos Proceedings of the 2nd IFAC – 2nd IFAC Workshop on Intelligent Assembly and Disassembly (International Federation on Automatic Control).
- **Machado, J. A. T. [1988],** “Gestão dos Recursos Estruturais no Controlo de Robots Manipuladores”. Tese de Doutoramento, Universidade do Porto.
- **Ogata, K. [1998].** “Engenharia de Controle Moderno”. Prentice-Hall do Brasil Ltda.
- **Oliveira, C.; Sá, C. E. A.; Rosário, J. M. [2002].** “Proposal of methodology for the modeling and control of manipulator”. International Journal of Brazilian Society of Mechanical Sciences, Vol XXIV, n 3.
- **Oliveira, C.; Sá, C. E. A.; Rosário, J. M. [2000].** “Using a Self Turn PID Controller to Anthropomorphic Protoses’s Articulations”. Medical Physics and Biomedical Engineering Rev., vol. 27, Number 6, Chicago.
- **Oliveira, C.; Sá, C. E. A.; Rosário, J. M. [2000].** “Implementation of Numerical Algorithms for the Solution of Kinematic Problem and Dynamical Control of Manipulator”. Nonlinear Dynamics, Chaos, Control and their Applications to Engineering Sciences - vol3- Chapter 5.
- **Oliveira, C.; Sá, C. E. A.; Rosário, J. M. [1999].** “Numerical Algorithms for the Solution of Kinematic Problem and Dynamical Control of Manipulator”. XV COBEM - Congresso Brasileiro de Engenharia Mecânica, Águas de Lindóia, Novembro.
- **Oliveira, C.; Sá, C. E. A.; Rosário, J. M. [1998].** “Modelagem Matemática e Simulação de Sistemas Dinâmicos não Lineares – Uma Aplicação a Robôs Manipuladores”. V CEM-NNE - UFPB.

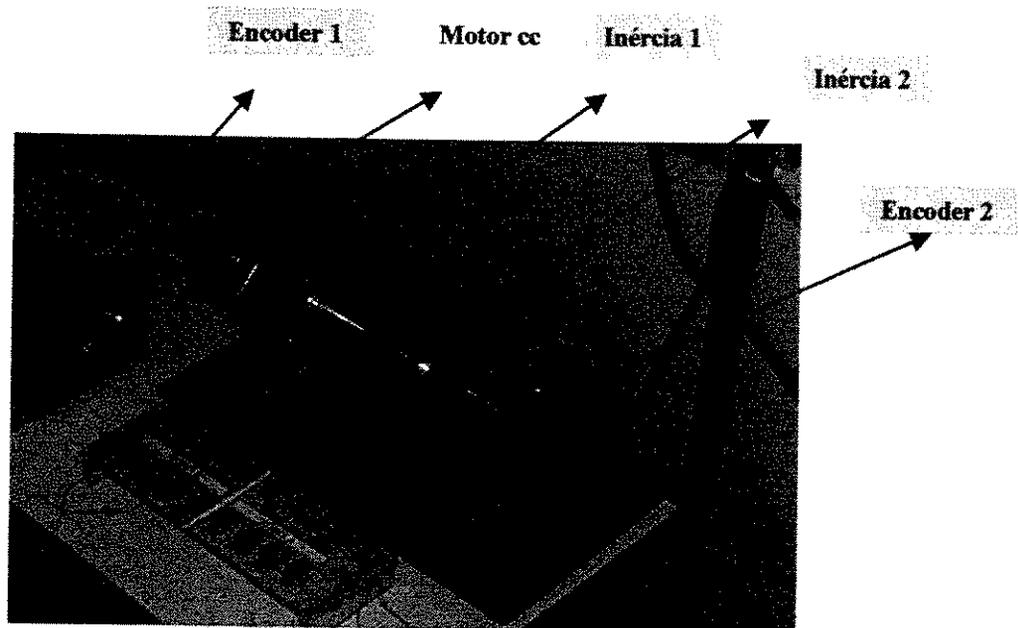
- **Oliveira, C.;** Sá, C. E. A.; Rosário, J. M. [1998]. “Implementation of Nonlinear Control in Robots Manipulators”. Controlo’98 – Special Session on Robotics and Automation. Universidade de Coimbra, Portugal.
- **Palm, W. J.** [1986]. “Control Systems Engineering”, John Wiley & Sons.
- **Paul, R. P.** [1981]. “Robot Manipulators : Mathematics, Programming and Control” . The Mit Press.
- **Phillips, C. L., et al** [1997]. “Sistemas de Controle e Realimentação”. Makron Books do Brasil Editora Ltda.
- **Romano, V. F. (editor)** [2002]. “Robótica industrial. Aplicação na industria de manufatura e de processos”. Edgard Blücher Ltda.
- **Rosário, J.M., Hermini, H. A., Sousa, J.P.** [1998]. “Application of Intentional Non-linearities in Dynamic Systems Control”. COCIM, Anais do 8o Congresso Chileno de Ingenieria Mecanica”, Outubro 1998, Conception, Chile, pg. 463-467.
- **Sá, C. E. A. de** [1996]. “Implementação de Métodos Numéricos para a Resolução do Problema Cinemático Inverso de Robôs com Ênfase em Controle de Posição”. Tese de Mestrado, Faculdade de Engenharia Mecânica, Unicamp.
- **Sá, C. E. A. de** [2000]. “Desenvolvimento e Implementação de um Programa Computacional para a Supervisão e Controle de Manipuladores Robóticos”. Tese de Doutorado, Faculdade de Engenharia Mecânica, Unicamp.
- **Santos, I. F.** [2001]. “Dinâmica de sistemas mecânicos – modelagem, simulação, visualização, verificação”. Makron Books Ltda.
- **Spong, M. W.** [1989]. “Robot Dynamics and Control” . John Wiley & Sons, Inc.

## **ANEXO I**

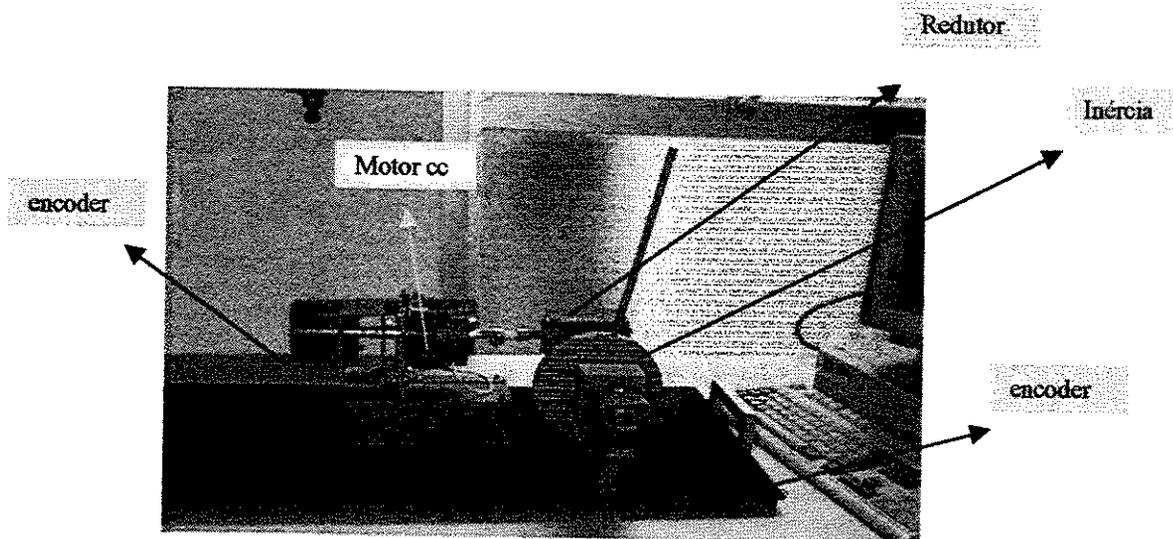
### **Descrição dos Elementos Constituintes da Plataforma Experimental**

#### **I 1 - Bancada Experimental**

Neste trabalho foram implementadas duas bancadas experimentais, onde a primeira é constituída de um motor de corrente contínua, dois discos de inércia acoplados através de uma embreagem eletromagnética e sensores incrementais de posição, enquanto na segunda foi implementada um sistema completo de transmissão de uma junta robótica (motor + redutor + carga). A figura I 1 apresenta essas duas bancadas e os principais elementos mecatrônicos utilizados.



a) Bancada Motor + Inércia Variável.



b) Bancada Motor + Redutor + Carga

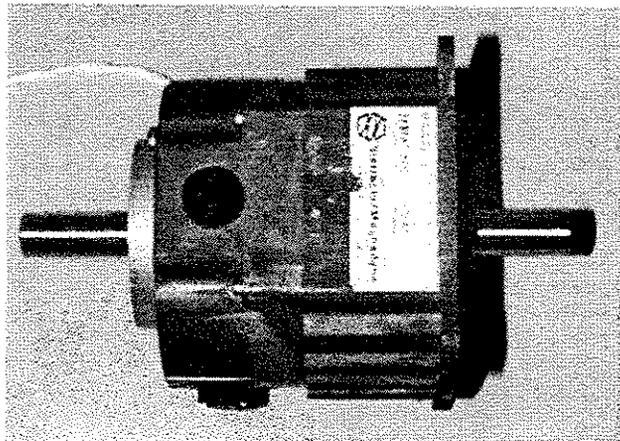
Figura I 1. Sistema de Acionamento utilizado.

A primeiro disco de inércia esta disposta no eixo do motor, enquanto que o acoplamento do segundo disco de inércia é realizado através de uma embreagem eletromagnética, que é

utilizada para acoplar ou não uma nova carga ao motor, possibilitando assim, variarmos a carga inercial no sistema (efeito similar a uma mudança de configuração de um robô).

## I 2 - Sistema de Acionamento - Motor cc

A figura I 2 apresenta as principais características do motor elétrico de corrente contínua utilizado na Plataforma Experimental motor + inércia variável, descrito no capítulo 6 desta tese.



a) Motor de corrente contínua

Modelo: EC M-1030
Velocidade máxima (segurança): 8 Krpm
Velocidade nominal com carga: 4 Krpm
Coefficiente de Atrito viscoso máximo: 2.5 Oz-in
Frequência Máxima: 4 KHz
Constante de tempo elétrica: 0.11 ms
Constante de tempo mecânica: 2.3 ms
Coefficiente de amortecimento: 1 Oz-in/Krpm

b) Principais Características

Figura I 2. Motor cc e características.

UNICAMP  
BIBLIOTECA CENTRAL  
SEÇÃO CIRCULANTE

### **I 3 - Sistema de Transmissão e Acoplamentos**

Para aproximar ainda mais o modelo físico ao modelo utilizado na simulação um sistema de transmissão mecânica através de um redutor de engrenagens foi utilizado uma vez que isso é muito comum em sistemas robóticos. A relação de engrenagens é de 1:100, lubrificada para redução do coeficiente de atrito.

Nas simulações realizadas tratamos o sistema sob dois aspectos: primeiro considerando o acoplamento entre o motor e a carga rígido e segundo considerando-o não rígido. Na bancada experimental para simular este acoplamento não rígido, utilizamos uma conexão flexível.

### **I 4 - Transdutores de Posição**

Os transdutores utilizados para medida de posição e velocidade são do tipo incremental (encoders), onde um deles se encontra acoplado diretamente no eixo do motor e outro no eixo da carga. Como os sinais oriundos dos transdutores são pulsos digitais, foi necessário desenvolver uma interface (hardware) e um programa computacional (software) para tratamento dessas informações (pulsos digitais relativos a um deslocamento realizado).

### **I 5 - Inércia Variável**

A bancada experimental simula um grau de liberdade de um robô, no entanto, para tornar nosso sistema mais próximo a um sistema robótico com carga inercial variável, utilizamos uma haste rígida presa ao disco de inércia principal com a possibilidade de acoplar diferentes massas em sua extremidade para simular o efeito desejável. Nessa mesma estrutura, usando uma haste flexível, abre-se um grande espaço para análise e controle de estruturas flexíveis, que é tema de um outro trabalho de pesquisa em andamento.

## I 6 - Sistema de Aquisição e Controle

Para aquisição de sinais e controle do sistema foi implementado programa computacional com os algoritmos de controle usando bibliotecas dedicadas (figura I 3). Parte desse experimento foi implementado em lógica reprogramável, utilizando o sistema de desenvolvimento ALTERA™. A vantagem de se ter esse experimento em ALTERA™ é que teremos uma malha de controle, executada em tempo real independente do *clock* do micro-computador.

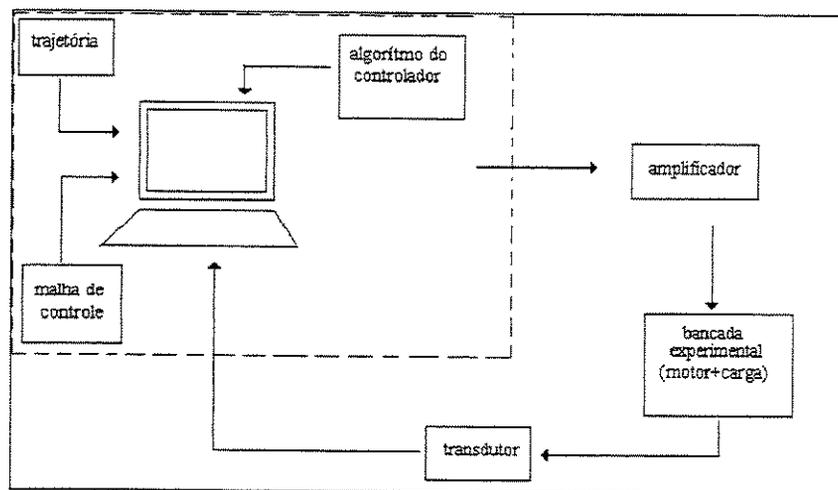


Figura I 3. Sistema de aquisição e controle.

## **ANEXO II**

### **Estrutura de Supervisão e Controle de um Manipulador Robótico**

#### **II 1.1 Estrutura do Sistema de Supervisão e Controle**

A estrutura do Sistema Completo de Supervisão e Controle de um Robô Industrial é apresentada na figura II 1. Ela é constituída basicamente dos seguintes elementos:

- Interface de Acionamento de cada junta, implementada em lógica reprogramável (Ai)
- Conjunto Motor + Redutor de cada junta (Mi)
- Interface de acionamento da ferramenta terminal (garra): G
- Transdutor de posição de cada junta (encoder incremental): Ei
- Entradas digitais para comunicação com outros dispositivos: I0 .. I7
- Saídas digitais para comunicação com outros dispositivos: O0 .. O7:

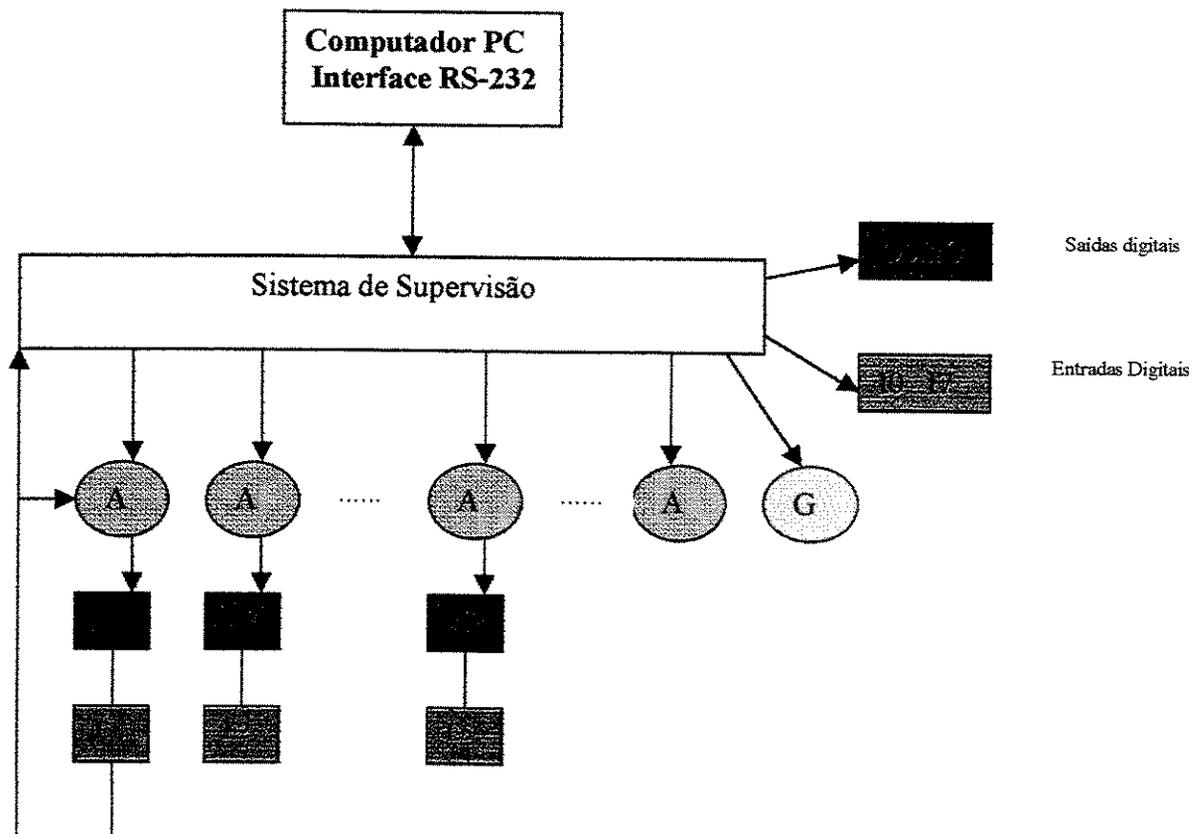


Figura II 1: Modelo de Interface de Supervisão e Controle de um Robô Industrial

## II 2 Descrição Detalhada dos principais elementos

### II 2.1 Sensor de Posição e final de curso das juntas

A medida de posição realizada através de encoder incremental (disposto no eixo de cada um dos motores antes dos redutores) com duas pistas (medida de posição e sentido de rotação) e eventualmente uma pista destinada a inicialização.

Em todas os graus de liberdade do robô são colocados sensores de final de curso: todos os cinco graus de liberdade serão constituídos de limitadores mecânicos de final de curso, sensores

de final de curso (dispostos antes dos limitadores), e proteção via software (antes dos sensores de final de curso).

## **II 2.2 Programa de Geração de Trajetórias**

A estrutura de geração de trajetórias será arborescente permitindo a realização de uma tarefa completa, sendo o software desenvolvido a partir de macros, contendo as seguintes informações:

- Posição desejada em incrementos;
- Velocidade máxima;
- Aceleração máxima;
- Informações I/O;
- Abertura/fechamento da garra;
- Temporização.

## **II 3 Hardware de Acionamento e Controle das Juntas**

### **II 3.1 Módulo de potência**

Os motores de corrente contínua deverão ser controlados por PWM (transistores de potência com foto-acopladores óticos), obtidos através de um trem de pulsos gerado em ALTERA, após a comparação entre a posição real (*encoder*) e faixas de posição pré-calculadas em programa dedicado implementado num microprocessador (função de um perfil trapezoidal de velocidade).

### **II 3.2 Módulo de controle:**

O controle das juntas será desenvolvido utilizando lógica ALTERA, que consistirá na comparação da posição da junta (contagem das informações provenientes dos *encoders* de cada junta) com informações do microprocessador pré-armazenadas em um Registro, e em função do resultado será enviado um trem de pulsos (PWM), função de um perfil trapezoidal de velocidades (aceleração, velocidade constante, frenagem e parada). Um sinal de erro, após interrupção de funcionamento, poderá ser enviado ao microprocessador.

A operação de abertura e fechamento da garra de prensão será realizada num módulo de acionamento e controle independente (utilizando relês), executado sempre no final de uma determinada trajetória.

Analogamente as informações I/O serão enviadas ou recebidas sempre no final da trajetória.

## **II 4 Software de Supervisão e Controle das Juntas**

A supervisão de movimentos será realizada num microprocessador constituído de dois módulos básicos:

### **II 4.1 Módulo de Aprendizagem:**

No módulo de aprendizagem será possível a realização de operações de movimento individuais de cada junta, com possibilidade de monitoramento das informações provenientes dos *encoders* de posição de cada junta, entradas I/O, abertura/fechamento da garra de prensão, e eventualmente informações sobre o status do sistema (velocidade, aceleração, numero de pontos da trajetória, etc.);

Este programa será desenvolvido num computador, com comunicação serial com um microprocessador que deverá fornecer as informações para o sistema;

O controle das juntas será realizado através do teclado (*ou mouse*), podendo as mesmas ser operadas em duas velocidades (baixa e alta);

Nesta etapa será possível a criação de trajetórias e tarefas (macros) num editor ASCII que poderá ser lido ou trabalhado num editor ou processador de texto comercial.

#### **II 4.2 Supervisão e Controle de Movimentos**

Neste módulo as macros realizadas na etapa anterior serão transferidas ao microprocessador (via interface serial RS232), e este se encarregará de transferir essas informações ao circuito lógico ALTERA (via barramento), responsável pelo controle em tempo real das juntas (independentes).

No final de cada movimento o conjunto de informações de posições angulares das juntas e o status do sistema deverá ser transferido ao software de supervisão do computador através da interface serial.

## **ANEXO III**

### **IMPLEMENTAÇÃO DO SISTEMA DE SUPERVISÃO E CONTROLE**

#### **III 1 – Descrição Funcional do software implementado**

##### **III 1.1 – Interface com o usuário**

O software desenvolvido consiste em uma interface gráfica que simula um painel de controle para a bancada experimental, onde o usuário pode controlar o funcionamento da junta, através dos seguintes botões:

- a) Liga/desliga
- b) Movimentação do motor em sentido horário
- c) Movimentação do motor em sentido anti-horário
- d) Acoplamento da carga

##### **III 1.2 – Programa de Aquisição de Informações do Encoder**

Este programa é capaz de montar um banco de dados com as informações coletadas no movimento da junta, a partir da aquisição e tratamento das informações dos sinais dos *encoders*, obtendo assim, diretamente informações de posição das juntas e de velocidade; conseguido com a divisão dos dados de posição em intervalos de tempo. Os resultados podem ser obtidos tanto em valores numéricos quanto em gráficos.

Ao mesmo tempo foram confeccionadas interfaces para comunicação entre sistema e computador através de interface paralela (DB25) com foto-acopladores (por medida de segurança), criando infra-estrutura, que será usada em outras atividades do curso de Engenharia de Controle e Automação. Além disso, foi feito o *download* de *drivers* de DELPHI™ da página na Internet da empresa Advantech (<http://www.advantech.com>).

### III 1.3 – ALTERA MAX+PLUS II™

A aquisição e monitoramento de dados na junta robótica é realizada através de circuitos lógicos reprogramáveis ALTERA™. Esse sistema foi desenvolvido em conjunto com um projeto de controle do robô didático ROBIX, pois os interesses eram semelhantes (tratar os dados provenientes de uma máquina).

O programa é responsável pela montagem de um banco de dados com as informações coletadas por sensores no *encoder* durante o movimento da junta robótica.

Esses dados podem ser mostrados em valores numéricos de posição em forma de telas com gráficos (forma preferida, pois, pode-se verificar o histórico do movimento das máquinas). Essa é a parte que está sendo compartilhada com o ROBIX. No caso deste projeto, é interessante também apresentar a velocidade da junta; isso pode ser feito de forma simples através de divisão da variação da distância por intervalos de tempo ( $ds / dt$ ).

## III 2 – Descrição do software implementado em linguagem DELPHI™

### III 2.1 – Descrição da Interface Principal

A interface principal é composta pelas seguintes partes:

- **Caixa de Botões de Controle:** simula a bancada experimental com botões para Ligar / Desligar, Girar em Sentido Horário / Girar em Sentido Anti-Horário e Acoplar / Desacoplar a Carga.
- **Caixa de Controle Automático:** pode inserir, remover e alterar um controle em um dado instante; possui também uma caixa onde são mostrados os comandos a serem executados (não foi muito desenvolvida devido a outras prioridades).
- **Caixa de Status da junta:** mostra todos as informações da junta robótica (dados), inclusive informações de Entrada e Saída, informações sobre posição e velocidade.

### III 2.2 – Descrição da Interface Principal

Os principais procedimentos e funções serão descritos a seguir:

- **function Pos(var s1, s2: String);** devolve a posição em que a string que representa o tempo de início de algum comando (e/ou o fim dele) começa. Esta função poderá ser usada posteriormente no “Comando Automático” (como já foi iniciado).
- **procedure ButtonSairClick (Sender: TObject);** é responsável para sair do programa ao clicar no botão sair.

- **procedure SpeedButtonLiga1Click (Sender: TObject)** e **procedure SpeedButtonLiga0Click (Sender: TObject)**: respectivamente liga (ao clicar no **BOTÃO 1**) e desliga (ao clicar no **BOTÃO 0**) o motor.
- **procedure SpeedButtonSentHorClick (Sender: TObject)**: faz o motor girar em sentido horário.
- **procedure SpeedButtonSentAntHorClick (Sender: TObject)**: é responsável para fazer o motor girar em sentido anti-horário.
- **procedure SpeedButtonAcopla1Click (Sender: TObject)** e **procedure SpeedButtonAcopla0Click(Sender: TObject)**: fazem a junta acoplar e desacoplar a carga ao clicar nos **BOTÕES 1 e 0**.
- **procedure TimerRelogioTimer (Sender: TObject)**: Foi criado um relógio interno, útil para fazer um controle do sistema em tempos definidos pelo usuário (por exemplo: liga o motor em sentido horário às 13:00:00 e desliga às 17:00:00) e que também seria usado no “Controle Automático” com o intuito de perceber o início e / ou o fim de um comando com o procedimento.
- **procedure ButtonInserirClick (Sender: TObject)**, **procedure ButtonRemoverClick (Sender: TObject)** e **procedure ButtonAlterarClick (Sender: TObject)**: fazem, em ordem, inserção, remoção e alteração de um dado comando em um certo momento (seriam usados no “Controle Automático”).
- **procedure TFormRobo.TimerTimer (Sender: TObject)**: criado para fazer a comunicação entre a máquina e a junta robótica via porta paralela.

## ANEXO IV

### Implementação de Hardware de Aquisição, Tratamento e Controle utilizando Lógica Reprogramável

#### **IV 1. Introdução**

Trabalhos experimentais na área de robótica necessitam, geralmente, de uma infraestrutura complexa, nem sempre possível de ser implementada, tendo em vista o alto custo envolvido. A fim de minimizar os custos, então, trabalha-se com softwares que facilitam bastante o trabalho. No nosso caso foi utilizado o MAX+plus II Manager da ALTERA.

O emprego de componentes lógicos reprogramáveis ALTERA permite a otimização e redução do número de elementos a utilizar-se, comparativamente o emprego de lógica discreta e aumento da confiabilidade do sistema. Assim, o sistema pode ser estruturado de modo a permitir uma flexibilidade de utilização.

Dentro desse contexto, utilizaremos a lógica reprogramável para controlar uma junta robótica, com uma infra-estrutura computacional para aquisição e tratamento de informações dos sensores utilizados (encoders).

Assim, foi implementada uma malha de controle de posicionamento de uma junta robótica. Uma central de processamento envia o comando da coordenada que o elemento mecânico acoplado ao eixo deve assumir. O circuito de controle de posição recebe a informação e

a compara com a posição atual do eixo. A diferença entre as duas informações é processada e gera um sinal de comando ao motor, acionando-o em um determinado sentido de rotação e passos que o motor executa.

Neste anexo são apresentadas as representações gráficas das interfaces implementadas neste trabalho de doutoramento.

## **IV 2 – Princípio de Funcionamento de um Encoder Incremental**

Os encoders (codificadores) são aparelhos que determinam a posição através de um disco ou trilho marcado.

Eles podem ser classificados como relativos, nos quais a posição é demarcada por contagem de pulsos transmitidos, acumulados ao longo do tempo, e absolutos, onde há um código digital gravado no disco ou trilho, lido por um conjunto de sensores ópticos (fonte de luz e sensor). Os códigos adotados são os de Gray, nos quais de um número para o seguinte só muda um bit, o que facilita a identificação e correção de erros.

A demarcação do disco ou trilho é feita através de furo ou ranhuras, ou por pintura num disco plástico transparente, que podem ser feitos através de técnicas foto-litográficas, permitindo grande precisão e dimensões micrométricas.

A fonte de luz é geralmente o LED, e o sensor um fotodiodo ou fototransistor. Estes sensores são muito precisos e práticos em sistemas digitais (encoder absoluto), e usam-se em robôs, máquinas-ferramenta, CNC e outros.

### IV 3 – Implementação de Circuito de Tratamento de Informações do Encoder e aquisição de informações através de um PC

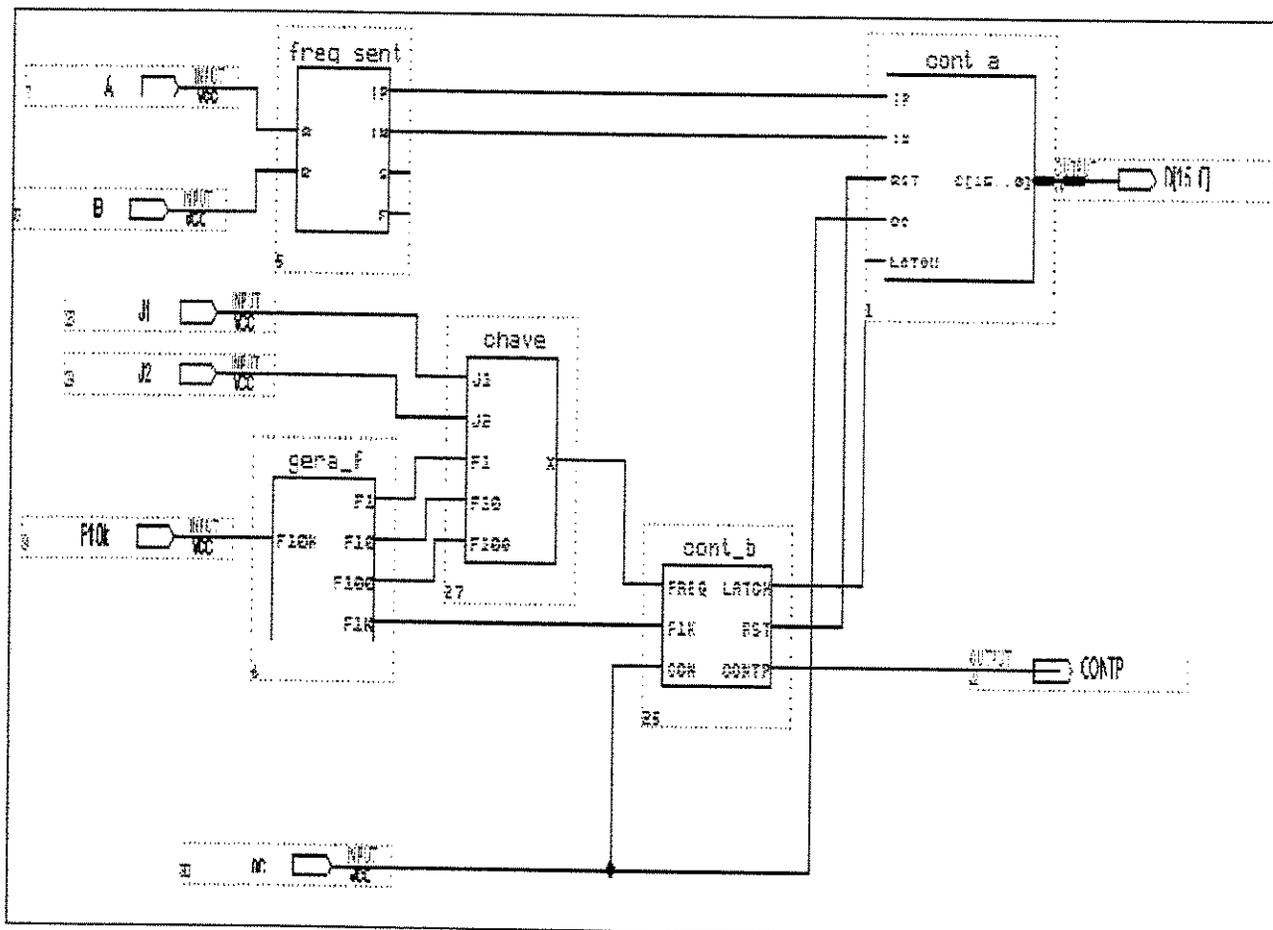
Nesta fase foi implementado um circuito completo para determinação da posição, velocidade e aceleração de um motor de uma junta robótica, a partir das informações fornecidas pelo *encoder* incremental, (2 pistas defasadas de 90°), que se apresentam defasadas de 90 graus uma da outra.

Dessa forma foram implementados em lógica reprogramável utilizando o software MAX-PLUS II da ALTERA, circuitos lógicos para detecção da frequência e sentido de rotação, um circuito de contagem e um módulo de comunicação com a interface paralela de um PC para transmissão de informações (sentido, posição). A seguir são apresentados o circuito completo implementado e relação de módulos implementados.

#### IV 3.1 – Relação de Módulos Implementados

Módulo	Função	Item
Freq_Sent	Circuito Discriminador de Frequência e Sentido a partir das informações do Encoder Incremental	D4
Gera_f	Circuito Gerador de Frequências para o sistema de Aquisição através do Computador	D5
Chave	Chave de Seleção	
Cont_a	Circuito de Informação de Posição para PC (8 bits)	D6
Cont_b	Circuito de Comunicação com PC	D7

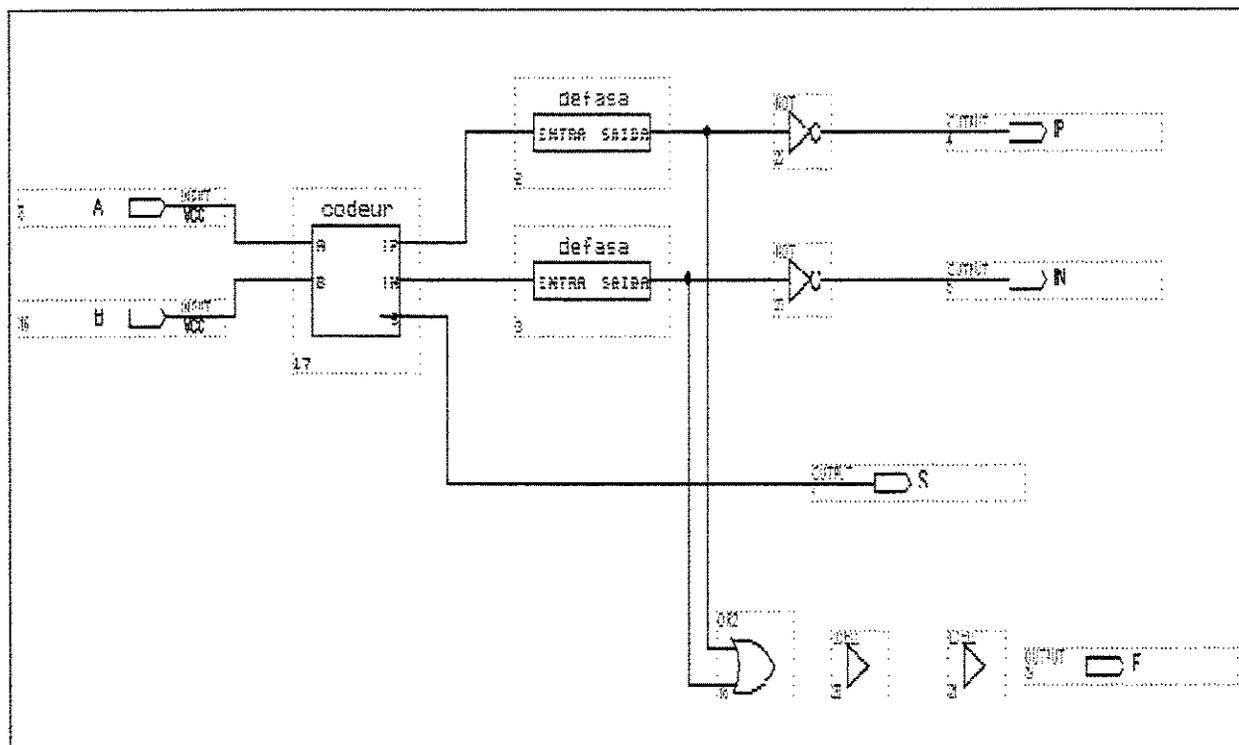
### IV 3.2 – Circuito Completo de Tratamento de Informações do *Encoder Incremental*



### IV 4 – Circuito de Tratamento de Informações do Encoder

Este módulo foi implementado com o objetivo de tratamento das informações (sinais A, B) de um encoder incremental colocado no eixo do motor. A relação de módulos implementados, entradas e saídas utilizadas na elaboração dos circuitos lógicos são apresentados a seguir.

#### IV 4.1 – Módulo Freq-Sent: Circuito de Detecção de Frequência e Sentido de Rotação de um Motor



#### IV 4.2 – Relação de Entradas e Saídas

ENTRADAS	Descrição
A	Pista A do encoder de posição da junta
B	Pista B do encoder de posição da junta

SAÍDAS	Descrição
IP	Impulsão positiva (junta se movimentando em sentido horário)
IN	Impulsão negativa (junta se movimentando em sentido anti-horário)
F	Frequência (posição angular)
S	Sentido (sentido de rotação da junta)

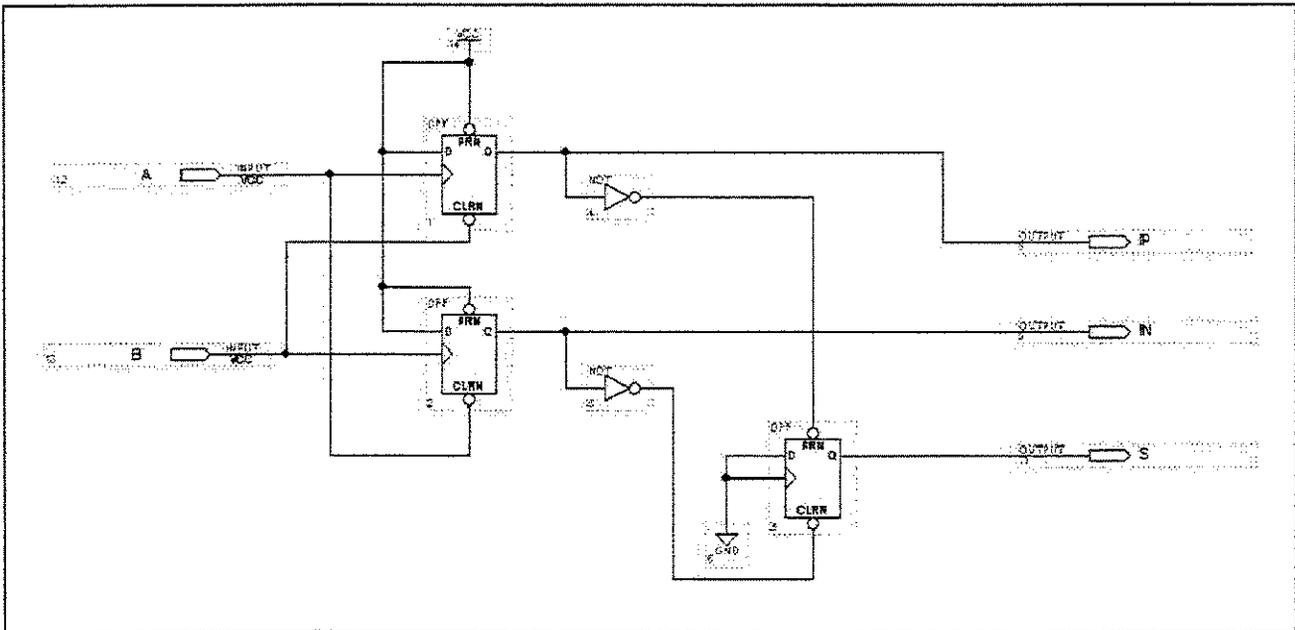
#### IV 4.3 – Relação de Módulos Implementados

<b>Módulo</b>	<b>Função</b>	<b>Item</b>
codeur	Circuito Discriminador de Frequência e Sentido a partir das informações do Encoder Incremental	D3.4
defasa	Módulo de Defasagem do Sinal (utilizado no detector de sentido do encoder)	D.3.5

#### IV 4.4 – Módulo Codeur: Circuito de Discriminador de Sentido de Rotação de um Motor

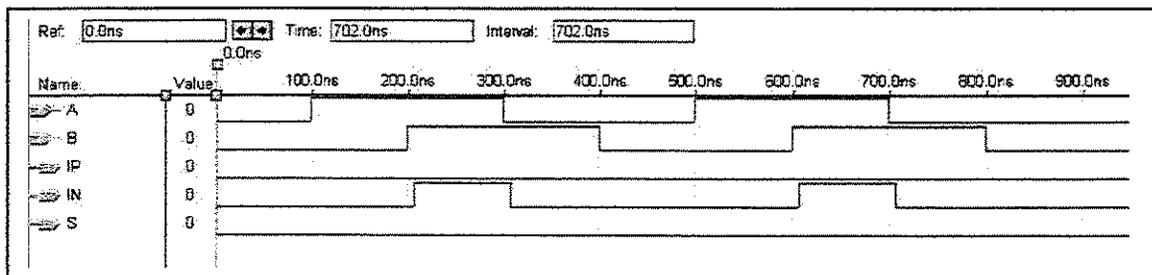
Sabendo desta defasagem montamos um circuito lógico que irá avaliar ambos os sinais de cada um dos fotosensores do encoder gerando assim dados suficientes para a realização das comparações e cálculos necessários à solução do problema.

Utilizamos para isso o software. Considerando que o Encoder (de duas pistas) nos fornece duas saídas, uma defasada de 90° graus em relação à outra, desenvolvemos um bloco chamado CODEUR, o qual irá interpretar os sinais enviados pelo encoder.

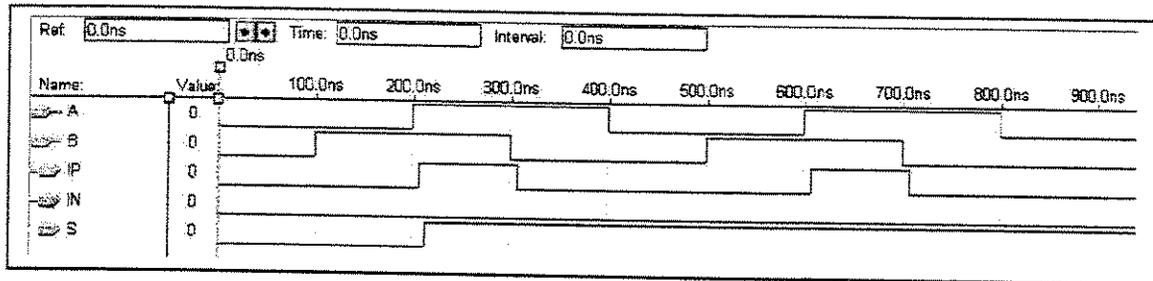


Este circuito irá analisar os dados e nos fornecer as seguintes informações: Sentido (S), onda de saída quando temos o movimento de A->B (IN), onda de saída quando temos o movimento de B->A (IP). É importante notar que IP e IN são equivalentes, mudando apenas pelo sentido que estamos considerando. A seguir temos a simulação das suas possíveis situações:

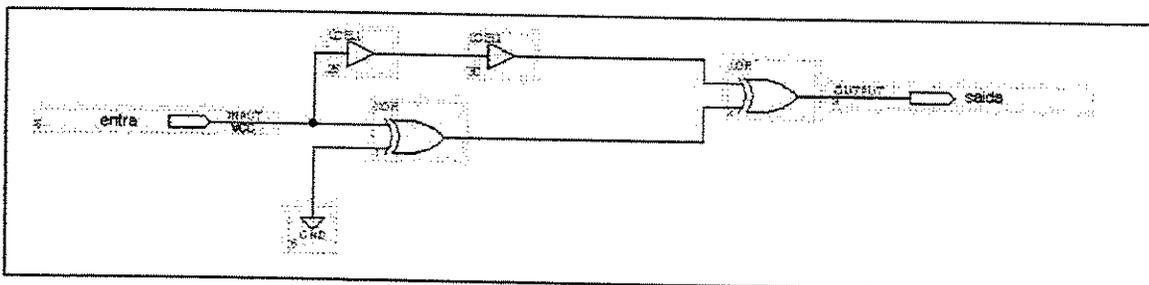
- **Simulação da situação A->B**



- **Simulação da situação B->A.**

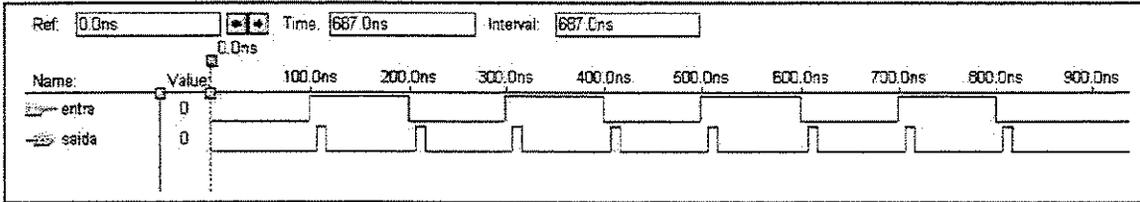


Nota-se que nesta ultima simulação, a partir de 200 ns a saída S passa de zero para 1, indicando que o sentido do movimento é B->A, podemos notar que na primeira simulação, a saída S permanece em 0 todo o tempo indicando o sentido de A->B. Esta avaliação de sentido pode ser também percebida se olharmos para qual saída (IP ou IN) está ativa, ou seja, qual está demonstrando pulsos ao longo do tempo. Para A->B temos que a saída ativa é IN, enquanto que a IP fica ativa quando temos sentido de B->A



Considerando a situação em que um dos sensores começa inicialmente já ativado, foi necessário desenvolver um bloco chamado DEFASA, o qual tem por função criar um curto pulso em cada sinal de subida ou descida do sinal de entrada. Isso faz com que o sinal de saída seja atrasado em relação o sinal de entrada. A seguir são apresentados resultados de simulação para diferentes situações.

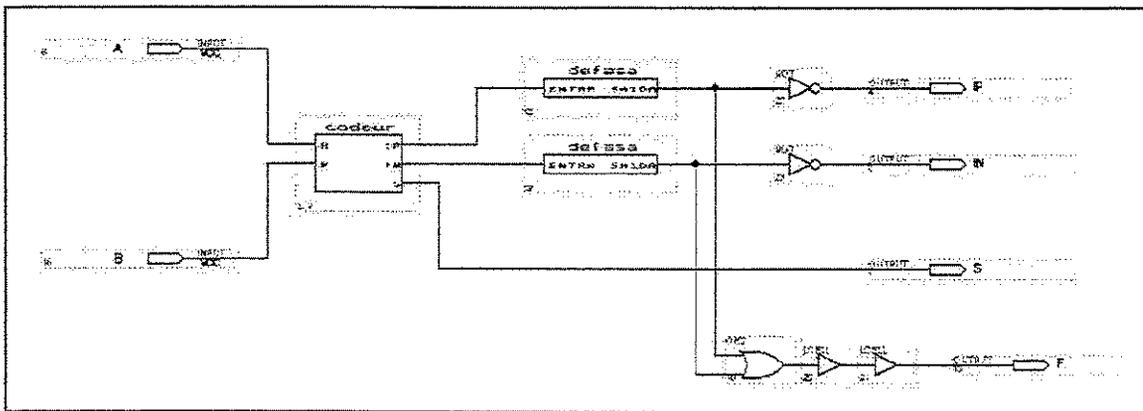
- Simulação da situação em que o sinal de entrada se encontra em nível baixo inicialmente.



A seguir está o circuit

o solução total do nosso problema. O funcionamento é simples, a partir das entradas A e B que são os geradas a partir dos dados dos leitores de feixe de luz do encoder teremos o tratamento destas entradas para que tenhamos prontamente o sentido do movimento, especificado pela saída S e ainda obtenhamos duas firmas de onda, uma delas em IP ou IN e a outra, inversa a estas em F.

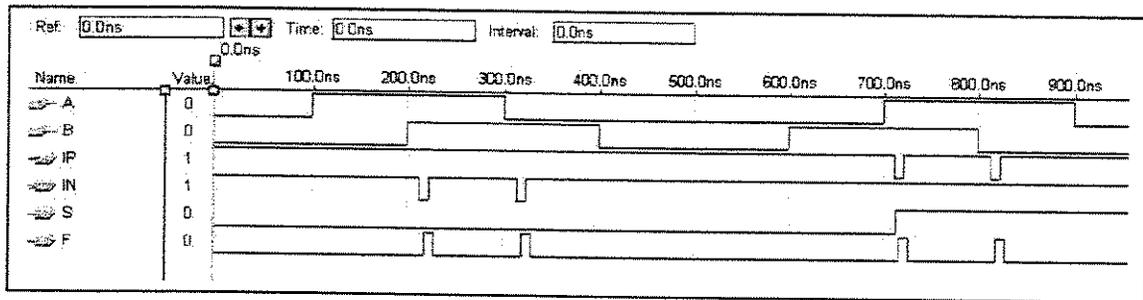
A partir da obtenção desta forma de onda, precisaremos avalia-la para obter as outras respostas para nosso problema, que são a velocidade e a aceleração do movimento. Para isso utilizaremos além de um clock de frequência conhecida como referência, um código AHDL para comparar as formas de onda, determinando a velocidade, e por cálculos também realizados pelo



bloco AHDL obteremos a aceleração.

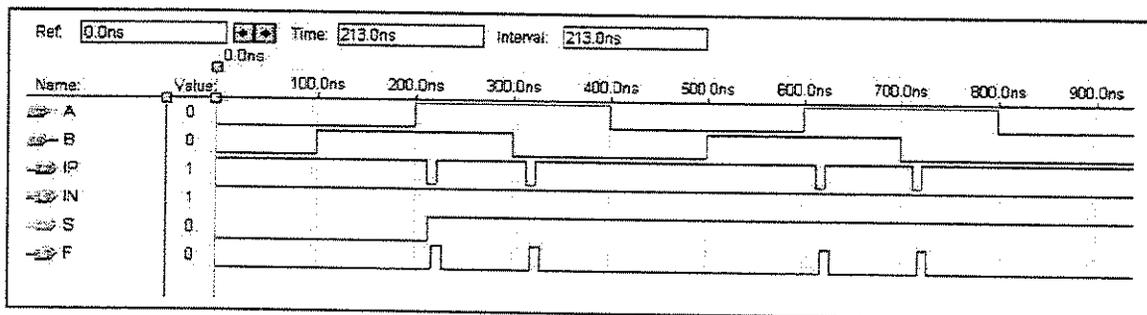
E as possíveis simulações:

- **Mudança de direção A -> B**



- **Mudança de direção B -> A**

- **Mudança de direção Mista: A -> B e B -> A**



Abaixo, mostraremos o código AHDL que irá avaliar as formas de onda geradas pelo encoder conseguindo solucionar o problema de se determinar a velocidade e aceleração do movimento (lembrando que já determinamos o sentido deste movimento). Para conseguir solucionar este problema devemos criar primordialmente um clock que, tendo frequência conhecida, poderá servir de base para o cálculo da velocidade e aceleração (ficarão em função da frequência do *clock* de referência).

No circuito implementado adotamos o seguinte procedimento: Contagem do número de pulsos em um determinado tempo do clock de referência e também da nossa saída IN ou IP (que serão entradas no novo circuito gerado em AHDL), dependendo do sentido. Note que poderíamos ter simplesmente utilizado a solução gráfica para isso, uma vez que a primeira parte do código

AHDL poderia ser eliminada simplesmente entrando com a saída F do circuito antigo ao invés de entrar com IP e IN e S. Outras soluções poderiam ser utilizadas em AHDL.

Após termos contado o número de pulsos num certo espaço de tempo para ambas as entradas IQQ e CLK (ver o AHDL abaixo) fica fácil calcular a velocidade e a aceleração, da maneira exposta abaixo.

```

SUBDESIGN relfinal
( IP, IN, S, iqq, clk1, ua, I      : INPUT;      %as entradas são as saídas do circuito que verific. o sentido%
  RespSerial[7..0]                : OUTPUT;      %saída serial para a leitura da velocidade e acel.%
)
VARIABLE
(   count[7..0], countp[7..0], countp2[7..0]    : DFF;
)

BEGIN

  IF S=="0" THEN
    iqq = IN;
  ELSE IF S=="1" THEN
    iqq = IP;
  END IF;
  END IF;

  count[].clk1 = clk1;
  countp[].iqq = iqq;
  I==0;
  FOR I < 10 GENERATE
    IF clk1 THEN
      count[] = count[] + 1;
    END IF;
    IF iqq THEN
      countp[] = countp[] + 1;
    END IF;
  END FOR

```

```
END GENERATE;
```

```
FOR I < 20 GENERATE  
  IF iqq THEN  
    countp2[] = countp2[] + 1;  
  END IF;
```

```
END GENERATE;
```

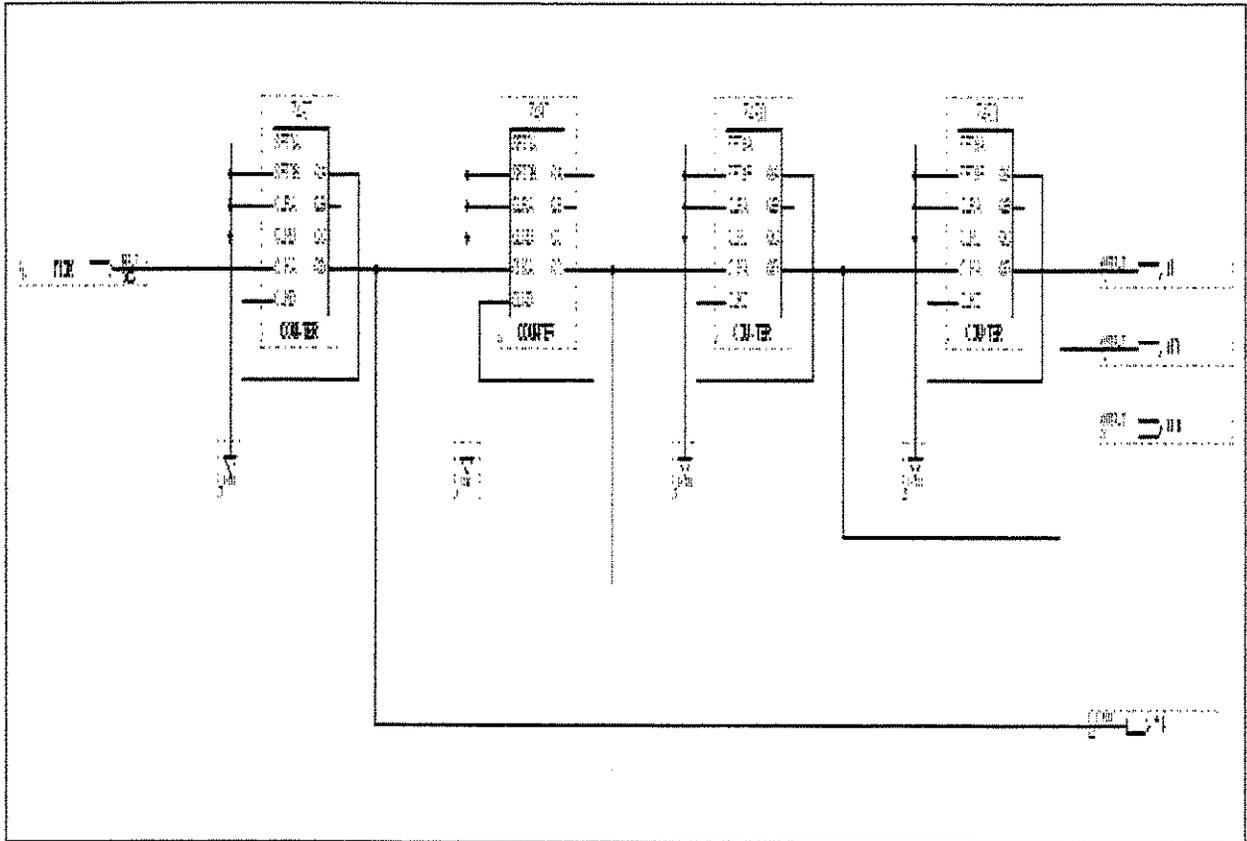
```
IF ua == 1 THEN  
  RespSerial[] = (countp[] + countp2[])/(count[]*2);  
ELSE  
  RespSerial[] = (countp2[] - countp[])/count[];  
END IF;
```

```
END;
```

#### IV 5 – Módulo Chave: Circuito de Gerador de Frequências para Aquisição

ENTRADAS	Descrição
F10K	Sinal de frequência externo

SAIDAS	Descrição
F1	Sinal de frequência de 1 Hz
F10	Sinal de frequência de 10 Hz
F100	Sinal de frequência de 100 Hz
F1K	Sinal de frequência de 1K Hz



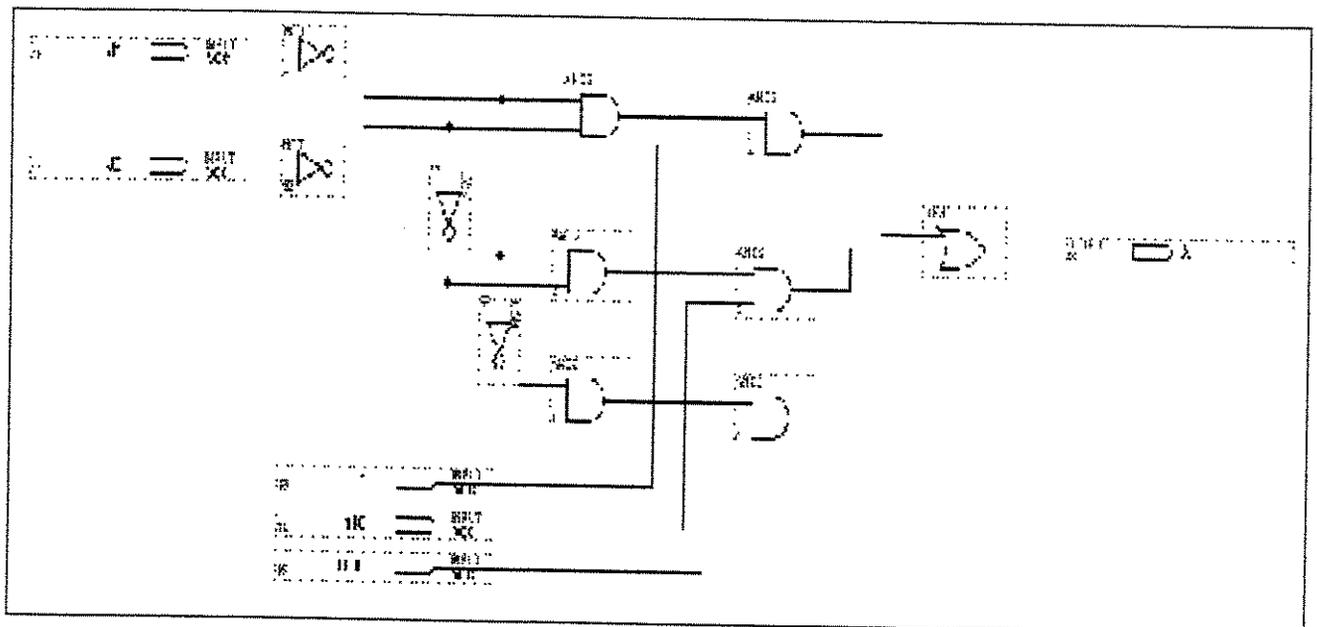
UNICAMP  
 BIBLIOTECA CENTRAL  
 SEÇÃO CIRCULANTE

#### IV 5.1 – Relação de Entradas e Saídas

ENTRADAS	Descrição
J1	Sinal lógico proveniente do PC
J2	Sinal lógico proveniente do PC
F1	Sinal de frequência de 1 Hz
F10	Sinal de frequência de 10 Hz
F100	Sinal de frequência de 100 Hz

SAÍDAS	Descrição
Freq	Sinal de frequência utilizado para aquisição pelo hardware implementado

#### IV 5.2 – Módulo Chave: Circuito de Chaveamento de Frequências

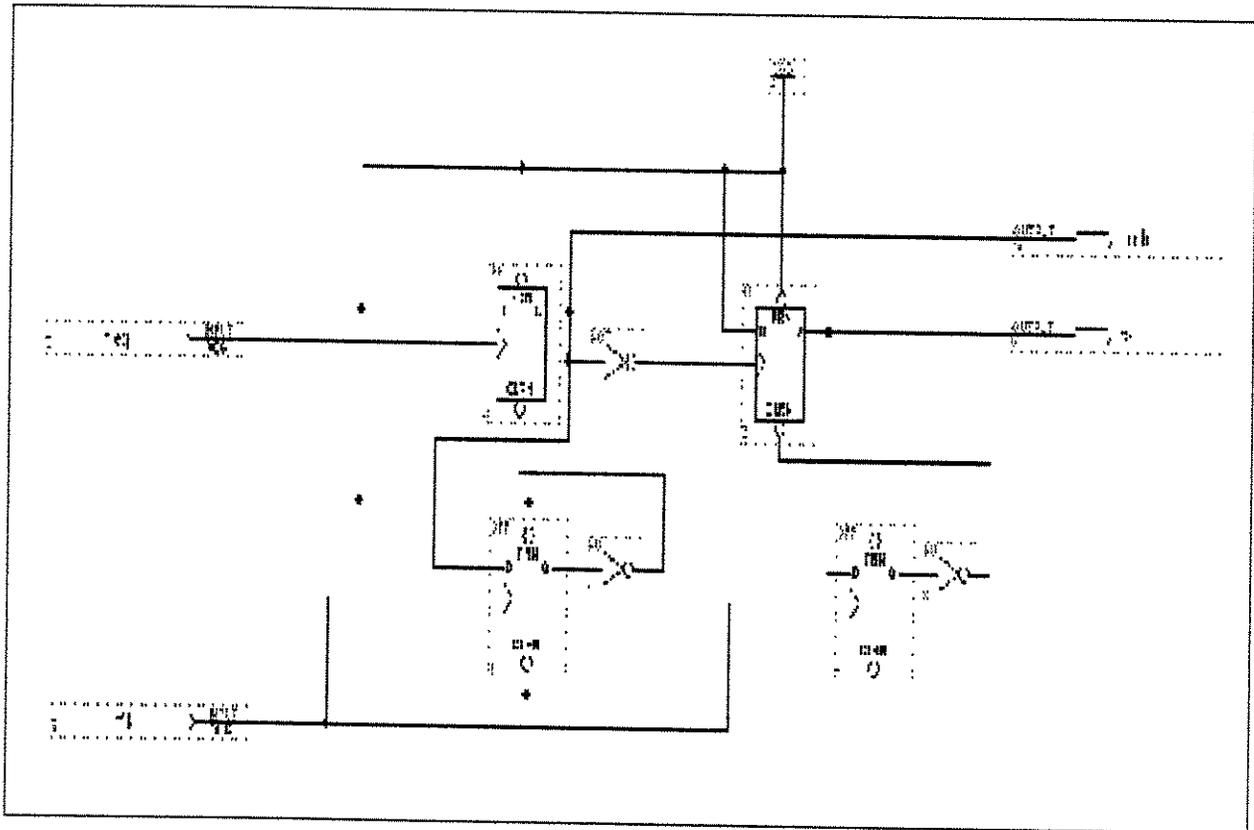


#### IV 5.3 – Circuito de Chaveamento

ENTRADAS	Descrição
freq	Sinal de frequência chaveado
F1K	Sinal de frequência de 1 KHz

SAÍDAS	Descrição
LATCH	Sinal de armazenamento de informações pelo contador Cont_a2
RST	Sinal de reset de informações pelo contador Cont_a2

#### IV 5.4 – Circuito de Chaveamento



#### IV 6 – Circuito de Aquisição com Micro-Computador

##### IV 6.1 – Relação de Entradas/Saídas

ENTRADAS	Descrição
IP	Impulsão positiva (junta se movimentando em sentido horário)
IN	Impulsão negativa (junta se movimentando em sentido anti-horário)
LATCH	Sinal de armazenamento de informações
RST	Sinal de reset de informações
OC1	Sinal de controle de aquisição proveniente do PC

OC2	Sinal de controle de aquisição proveniente do PC
-----	--

SAÍDAS	Descrição
D[7..0]	Sinal de saída para a interface paralela do PC (inf. do encoder de pos.)



## **ANEXO V**

### **Implementação de Circuito de Acionamento e Controle de um motor de Corrente Contínua**

#### **V 1 - Introdução**

O objetivo deste Anexo é apresentar a implementação do circuito de acionamento e controle de um motor de corrente contínua. Assim, iremos apresentar uma breve revisão sobre o acionamento dos motores e controlador PWM. Em seguida analisaremos os principais módulos do controle através da interpretação dos códigos e da realização de simulações utilizando os recursos do software MAXPLUS™

#### **V 2 – Revisão de Conceitos**

##### **V 2.1 – Acionamento de Motores**

Os motores são controlados a partir da interface paralela de um PC, que transmite sinais para uma interface dedicada desenvolvida da partir de circuitos lógicos reprogramáveis, onde um módulo de controle comanda a interface de potência. A figura V1 apresenta um esquema da malha de controle utilizada.

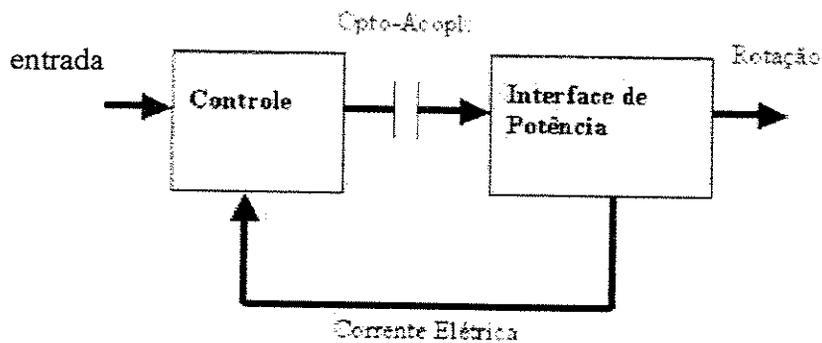


Figura V 1: Sistema de Acionamento e Controle.

Os motores trabalham nos dois sentidos de rotação e são responsáveis pela movimentação do grau de liberdade do robô. O acionamento dos motores da juntas é realizado de forma independente.

Cada motor é acionado por um conjunto de chaves que está montado em arranjo “Ponte H”. Esse arranjo permite o acionamento do motor nos dois sentidos de rotação com a utilização de apenas uma fonte de alimentação DC. A figura V 2 ilustra o arranjo das chaves:

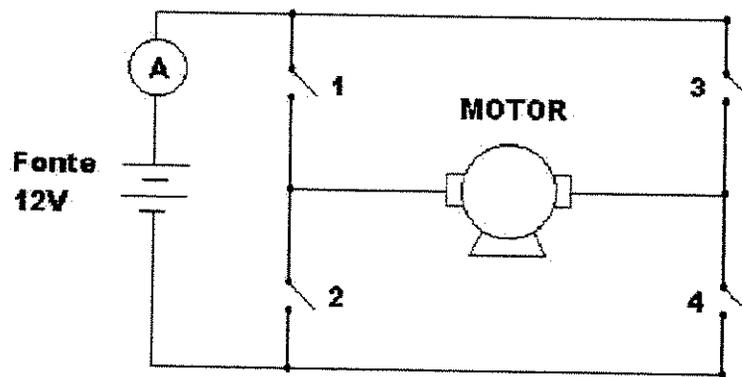


Figura V 2: Arranjo das chaves em “Ponte H”.

Através da Figura V 2, observamos que as chaves são acionadas aos pares. Assim as chaves 1 e 4 produzirão um sentido de rotação, enquanto que as chaves 2 e 3 produzirão o outro sentido.

## V 2.2 - Controlador através de PWM

Um controlador através de PWM (Pulse Width Modulation), ou MLP - Modulação por Largura de Pulso, é uma forma de controle de tensão por recorte onde os transistores de potência são ligados ou bloqueados de modo a obter na saída o valor de tensão desejada.

Este sinal de saída será uma forma de onda quadrada em que o “sinal alto” representa a tensão oferecida ao motor para sua rotação; ou seja, quando necessária uma alta rotação, aumenta-se a largura do “sinal alto” e diminui-se a do “sinal baixo” e vice-versa.

A partir da forma de onda desejada (velocidade ou aceleração), um circuito a compara com a onda real do DC motor e emite uma onda quadrada referente ao erro. Este erro é então utilizado para compensar o estado real, aproximando-o do ideal.

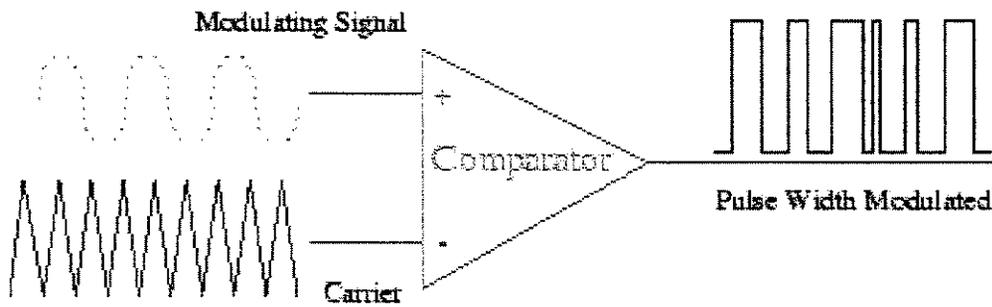


Figura V 3: Princípio de funcionamento de um controlador através de PWM.

## V 3 – Implementação do Sistema de Comunicação e Controle

### V 3.1 – Implementação do Controlador utilizando uma ponte H

O controlador deve gerar sinais capazes de controlar a Ponte H e dessa maneira fornecer a potência adequada ao motor. Foram utilizados opto-acopladores com objetivo de separar as correntes de controle e de potência. Dessa forma, o controle aciona o opto-acoplador e este polariza a chave de potência.

Nessa aplicação, os dois motores são tratados de forma simétrica, ou seja, o controle utilizado possui a mesma arquitetura e a mesma lógica para cada um dos dois motores. Por esse motivo apresentaremos apenas o acionamento de um motor.

Considerando um motor, temos que a entrada do controlador será o sinal de um dos canais do rádio, e as saídas serão dois sinais de comando, um sinal de giro no sentido horário e um sinal de rotação no sentido anti-horário.

Além do sentido de giro esse sinal também trás a informação do tempo em que chave permanecerá polarizada, quanto maior for o tempo em que as chaves estiverem fechadas, maior será a quantidade de corrente suprida ao motor e em conseqüência, maior será a sua potência. A figura V 4 mostra a arquitetura do controle utilizado.

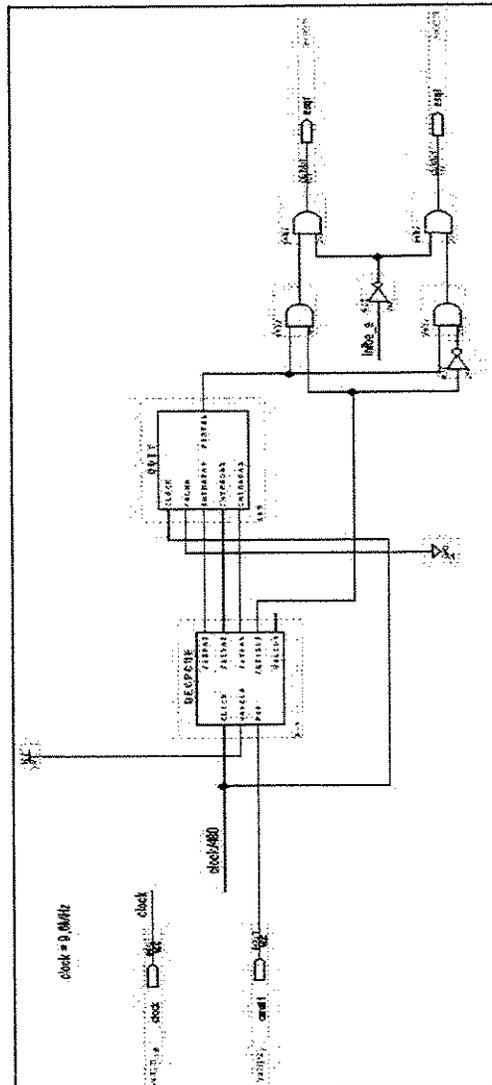


Figura V 4: Controlador

Na seqüência será feita a descrição dos módulos Decpcm e Duty, que são os módulos responsáveis pela interpretação e tratamento dos sinais de comunicação entre o PC e o Hardware de Controle.

### V 3.2 - Módulo Decpcm

A função desse módulo é converter um sinal PCM em uma palavra binária. São utilizados três bits para definir a palavra binária. De acordo com a largura do sinal PCM uma diferente

palavra será gerada. Desta forma, é possíveis a associação de uma determinada palavra com a largura do pulso, e podemos através das palavras definir níveis intermediários de operação.

A estratégia adotada foi a seguinte: Utiliza-se um sinal o sinal de *clock* de alta frequência para sincronizar os eventos e servir de referência. É utilizado um contador para armazenar o número de períodos de *clock* que cabem dentro do sinal PCM, dessa forma é obtida a largura do sinal PCM. Para associar a largura do pulso com a palavra binária, foi criada uma tabela. A tabela associa cada valor possível do contador a uma palavra binária.

#### **Entradas:**

***Clock***: sinal de *clock* para sincronização.

***Janela***: janela de ruído.

***Pcm***: sinal pcm do rádio.

#### **Saídas:**

***Saida0, Saida1, Saida2***: são os três bits da palavra binária.

***Sentido***: bit que determina o sentido.

***Valida***: sinal de valor válido na saída.

A implementação foi realizada em VHDL através da criação de uma máquina de 4 estados. A figura V 5 ilustra a máquina de estado implementada.

#### **V 3.2.1 - Descrição dos Estados**

***Zero***: o estado zero está aguardando o início de uma nova contagem.

***Um***: o estado um permanece contando até que ocorra um evento de fim de contagem.

***Dois***: o estado dois caracteriza o final de uma contagem, terminada a contagem, no estado dois será avaliado o valor do contador e em função disso serão determinadas as saídas 0, 1 e 2 e o sentido.

**Três:** o estado três caracteriza a ocorrência de um erro, a partir dele o contador será zerado e o estado retornara para Um para aguardar o início de uma nova contagem.

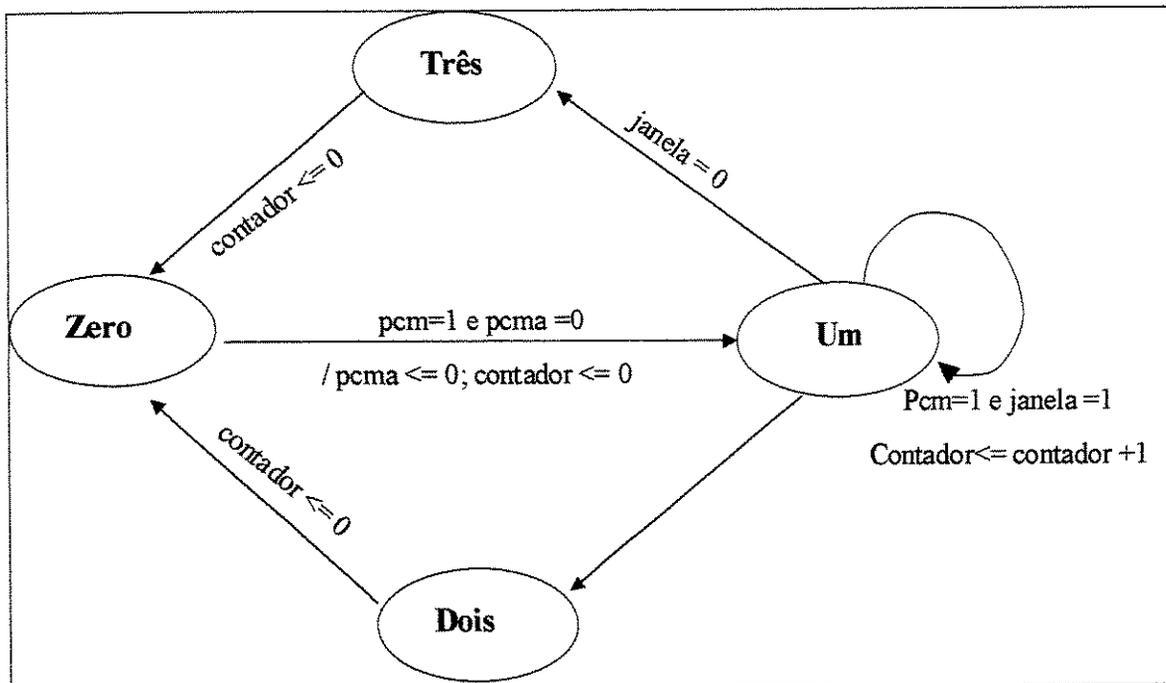


Figura V 5: Máquina de Estado implementada no módulo Decpcm

### V 3.2.2 - Tabelas de Avaliação do Estado Dois

Contador	Sentido
> 30	1
=< 30	0

Contador	Saída 0	Saída 1	Saída 2	Comentário
0<cont<15 ou cont>45	0	0	0	Fora da região de trabalho
15=<cont<26 ou 35<cont=<45	1	1	1	Velocidade Total
26=<cont<28 ou 32<cont=<35	1	0	1	Velocidade Parcial
cont>=28 Ou cont=<32	0	0	0	Zona Morta

### V 3.2.3 – Programa em VHDL Implementado

-- UNICAMP - LABORATORIO DE AUTOMAÇÃO E ROBÓTICA  
-- Projeto Gerra de Robos - setembro de 2001

-- Módulo: decpcm  
-- Descrição: converte um sinal PCM (canal de Radio Futaba) em palavra binária  
-- São usados quatro bits para definir níveis intermediários entre máximo e mínimo  
-- Arquivo: decpcm.VHD  
-- EPLD:  
-- Autor: Carlos Raimundo Erig Lima  
-- Data de criação: 05 de setembro de 2001  
-- Histórico:  
-- modificado em 13 de novembro para ajuste de tabela

```
ENTITY decpcm IS
    PORT(
        clock      : IN      BIT;    -- Sinal de clock para sincronização
        janela     : IN      BIT;    -- Janela de ruído de 5ms
        pcm        : IN      BIT;    -- Sinal pcm do canal de radio
        saida0     : OUT     BIT;    -- bit de seleção
        saida1     : OUT     BIT;    -- bit de seleção
        saida2     : OUT     BIT;    -- bit de seleção
        sentido    : OUT     BIT;    -- bit de seleção
        valida     : OUT     BIT);   -- pulso de valor valido na saída
END decpcm;
```

ARCHITECTURE decpcm OF decpcm IS

-- SINAIS INTERNOS

```
TYPE ESTADOS IS (zero, um, dois, tres);
SIGNAL estado : ESTADOS;
SIGNAL contador : INTEGER RANGE 0 TO 128;
SIGNAL pcma, j2a:BIT;
```

BEGIN

```
PROCESS (clock,janela, pcm)
BEGIN
    IF ((clock'EVENT) AND (clock='1')) THEN
```

CASE estado IS

```
    -- o estado zero espera o inicio de uma nova contagem
    WHEN zero =>
        -- detecta subida de pcm
        IF pcm = '1' AND pcma = '0' THEN -- bit 0 = 1
            pcma <= '1';
            contador <= 0;
            estado <= um;
        END IF;
```

```
    -- o estado um está contado a espera de um evento de fim de contagem
    -- , ou seja, pcm vai para zero ou janela vai para zero
    WHEN um =>
        IF pcm = '1' AND janela = '1' THEN contador <= contador +1;
```

```

END IF;
IF pcm = '0' THEN
pcma <= '0';
estado <= dois;
END IF;
IF janela = '0' THEN
estado <= tres;
END IF;

-- o estado dois caracteriza o fim de contagem de um valor válido
WHEN dois =>
-- determina sentido
IF contador > 30 THEN sentido <= '1';
ELSE sentido <= '0';
END IF;
-- faixa menor que 1ms (aproximadamente) é desconsiderada
IF (contador >= 0 AND contador <15) OR (contador >45) THEN
saida0 <= '0';
saida1 <= '0';
saida2 <= '0';
END IF;
IF (contador >= 15 AND contador <26) OR (contador >35 AND contador <=45) THEN
saida0 <= '1';
saida1 <= '1';
saida2 <= '1';
END IF;
IF (contador >= 26 AND contador <28) OR (contador >32 AND contador <=35) THEN
saida0 <= '1';
saida1 <= '0';
saida2 <= '1';
END IF;

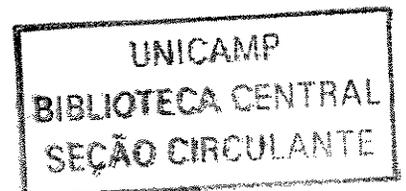
IF (contador >= 28 AND contador <=32) THEN
saida0 <= '0';
saida1 <= '0';
saida2 <= '0';
END IF;

contador <= 0;
estado <= zero;

-- o estado tres caracteriza a ocorrencia de erro na valor (ruído)
WHEN tres =>
contador <= 0;
estado <= zero;

END CASE;
-- j1a <= janela1;
END IF;
END PROCESS;
valida <= '1' WHEN (estado = dois) ELSE '0';
END decpcme;

```



### V 3.2.4 - Circuito montado para a Simulação do Módulo Decpcm

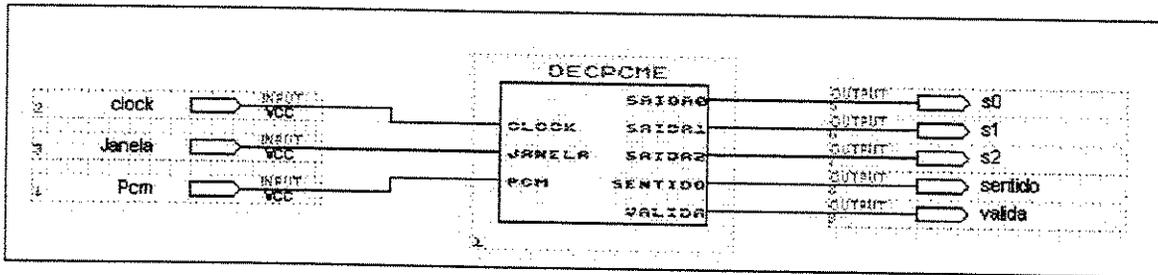


Figura V 6 – Circuito para a simulação do módulo Decpcm.

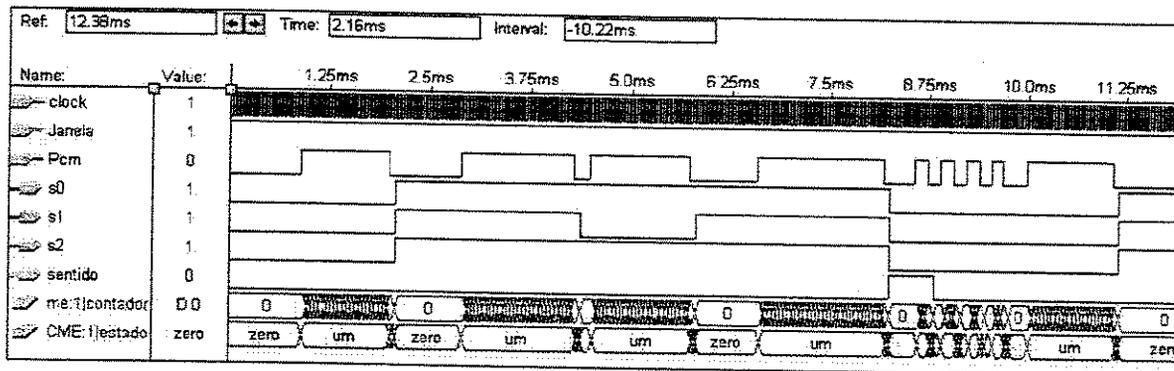


Figura V 7: Resultado da simulação do módulo Decpcm.

### V 3.2.4 – Comentários

Através da simulação podemos observar o funcionamento do circuito, no trecho selecionado vemos as saídas passando pelas palavras “000”, “111”, “101”, “000” e “111”.

Podemos também observar os estados percorridos pela máquina de estado.

Note que a tabela ainda pode ser refinada e com três bits podemos definir até oito palavras diferentes, ou seja, oito níveis diferentes de potência variando desde a mais baixa até a mais alta.

## V 3.3 - Módulo Duty

### V 3.3.1 – Princípio de Funcionamento

A função desse modulo é de uma palavra binária gerar um perfil de trabalho na saída do modulo, esse perfil de trabalho será o sinal que comandará a polarização das chaves de potência. São lidos os três bits gerados pelo modulo decpcm e em função desses bits é gerado um perfil de trabalho

#### Entradas:

**Clock:** sinal de *clock* para sincronização.

**Falha:** sinal de inibição.

**Entrada0, Entrada1, Entrada2:** palavra binária.

#### Saídas:

**Perfil:** pulso com o perfil selecionado.

A implementação foi realizada em VHDL através da criação de uma máquina de 2 estados. A Figura E.8 ilustra a máquina de estado implementada.

### V 3.3.2 - Descrição dos Estados

**Zero:** o perfil está em nível 1, o contador é incrementado até atingir uma condição para a transição para o estado um.

**Um:** o perfil permanece em nível 0, o contador é incrementado até atingir um valor de transição para o estado zero.

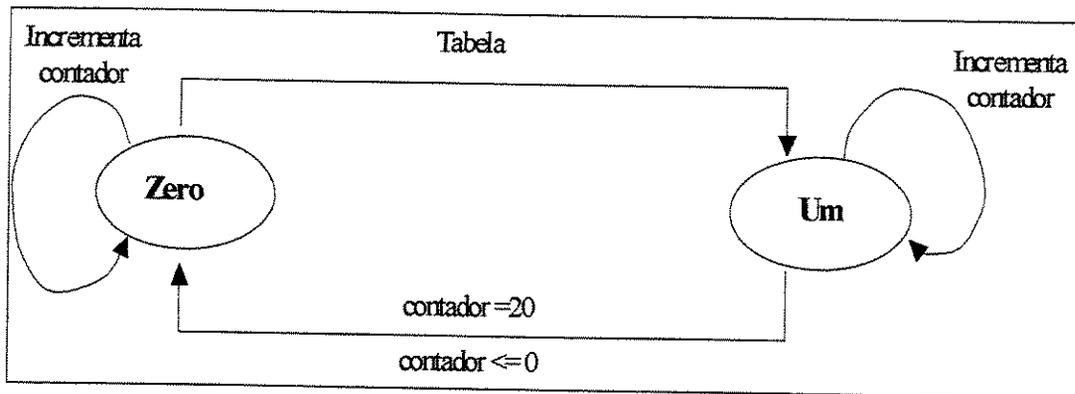


Figura V 8: Máquina de Estado do módulo Duty.

### V 3.3.3 -Tabelas de Avaliação na Transição Zero -> Um

Contador	Entrada 0	Entrada 1	Entrada 2	Comentário
X	0	0	0	Estado vai para Um
10	1	0	1	Estado vai para Um
19	1	0	1	Estado vai para Um

### V 3.3.4 – Programa Computacional em VHDL Implementado

– UNICAMP - LABORATORIO DE AUTOMAÇÃO E ROBÓTICA  
 – Projeto Gerra de Robos - setembro de 2001

- Módulo: duty
- Descrição: gera perfis de trabalho. São possíveis 16 perfis entre
- com diferentes ciclos de trabalhos. Todos os perfis são de 2kHz
- Arquivo: duty.VHD
- EPLD:
- Autor: Carlos Raimundo Erig Lima
- Data de criação: 05 de setembro de 2001
- Histórico:

```

ENTITY duty IS
  PORT(
    clock          : IN      BIT; -- Sinal de clock para sincronização
    falha          : IN      BIT; -- sinal de inibição
    entrada0       : IN      BIT; -- bit de seleção
    entrada1       : IN      BIT; -- bit de seleção
    entrada2       : IN      BIT; -- bit de seleção
  );

```

```

        perfil          : OUT BIT);          -- pulso com o perfil desejado
END duty;

ARCHITECTURE duty OF duty IS
        -- SINAIS INTERNOS
    TYPE ESTADOS IS (zero, um);
    SIGNAL estado : ESTADOS;
    SIGNAL contador : INTEGER RANGE 0 TO 32;

BEGIN
    PROCESS (clock, entrada0, entrada1, entrada2, falha)
    BEGIN
        IF ((clock'EVENT) AND (clock='1')) THEN

CASE estado IS
            -- NÍVEL 1
            WHEN zero =>
                contador <= contador +1;
                IF entrada0 = '0' AND entrada1 = '0' AND entrada2 = '0' THEN
                    estado <= um;
                END IF;
                IF entrada0 = '1' AND entrada1 = '0' AND entrada2 = '0' AND contador = 2 THEN
                    estado <= um;
                END IF;
                IF entrada0 = '0' AND entrada1 = '1' AND entrada2 = '0' AND contador = 4 THEN
                    estado <= um;
                END IF;
                IF entrada0 = '1' AND entrada1 = '1' AND entrada2 = '0' AND contador = 6 THEN
                    estado <= um;
                END IF;
                IF entrada0 = '0' AND entrada1 = '0' AND entrada2 = '1' AND contador = 8 THEN
                    estado <= um;
                END IF;
                IF entrada0 = '1' AND entrada1 = '0' AND entrada2 = '1' AND contador = 10 THEN
                    estado <= um;
                END IF;
                IF entrada0 = '0' AND entrada1 = '0' AND entrada2 = '1' AND contador = 16 THEN
                    estado <= um;
                END IF;
                IF entrada0 = '1' AND entrada1 = '0' AND entrada2 = '1' AND contador = 19 THEN
                    estado <= um;
                END IF;

            -- NÍVEL 0
            WHEN um =>
                contador <= contador +1;
                IF contador = 20 THEN
                    contador <= 0;
                    estado <= zero;
                END IF;

END CASE;
        END IF;
    END PROCESS;

```

```

perfil <= '1' WHEN (estado = zero AND falha = '0' AND (entrada0 /= '0' OR entrada1 /= '0' OR entrada2 /= '0'))
OR (entrada0 = '1' AND entrada1 = '1' AND entrada2 = '1') ELSE '0';
END duty;

```

### V 3.3.5 – Circuito implementado para a Simulação do Módulo Duty

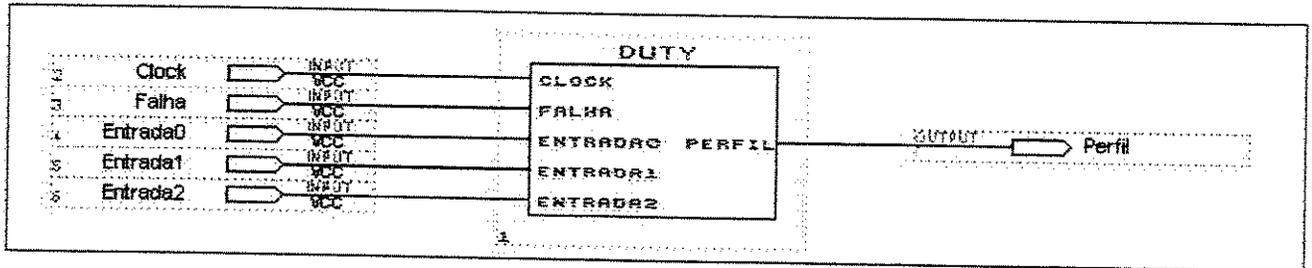


Figura V 9: Circuito para a simulação do módulo Duty.

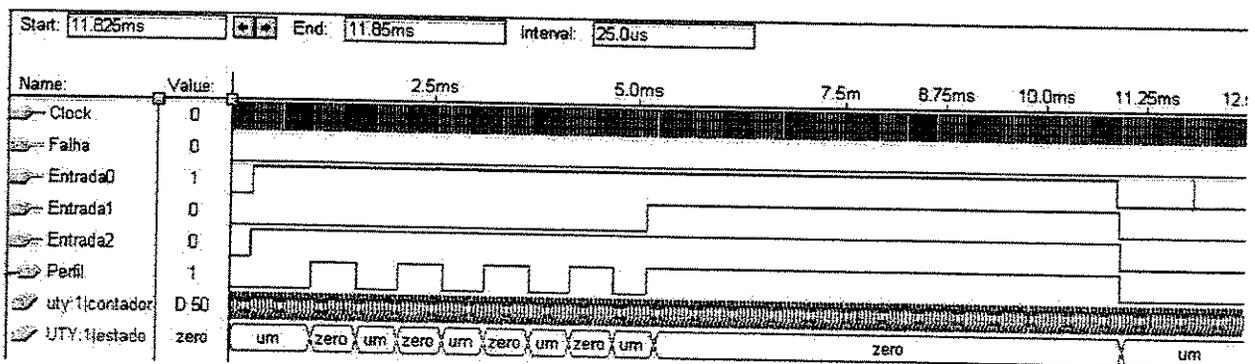


Figura V 10: Simulação do módulo Decpcm.

### V 3.3.6 - Comentários

Através da simulação podemos observar o funcionamento do módulo. Foram simuladas as entradas “000”, “101”, “111” e “000”, respectivamente.

Observando o perfil podemos notar que para “000” o nível de saída é 0, para ”101” o perfil é pulsado, e para “111” o perfil é nível 1. Dessa forma a palavra “111” corresponde à potência plena enquanto que a palavra ”101” corresponde a um nível intermediário de potência. Podemos também observar a transição na máquina de estado.

#### V 4 – Controle de velocidade e aceleração de um Motor de corrente contínua através de PWM

Para o controle de um motor cc através do PWM, utilizamos um circuito que gera duas saídas. A primeira saída fornece o sentido de rotação do motor, sendo que adotamos “1” para horário e “0” para anti-horário. A segunda saída é a frequência do giro do motor.

Pensando em exemplificar o uso deste circuito, geramos uma forma de onda simulando a rotação real do motor e uma segunda simulando a onda desejada. Assim obtivemos os seguintes resultados.

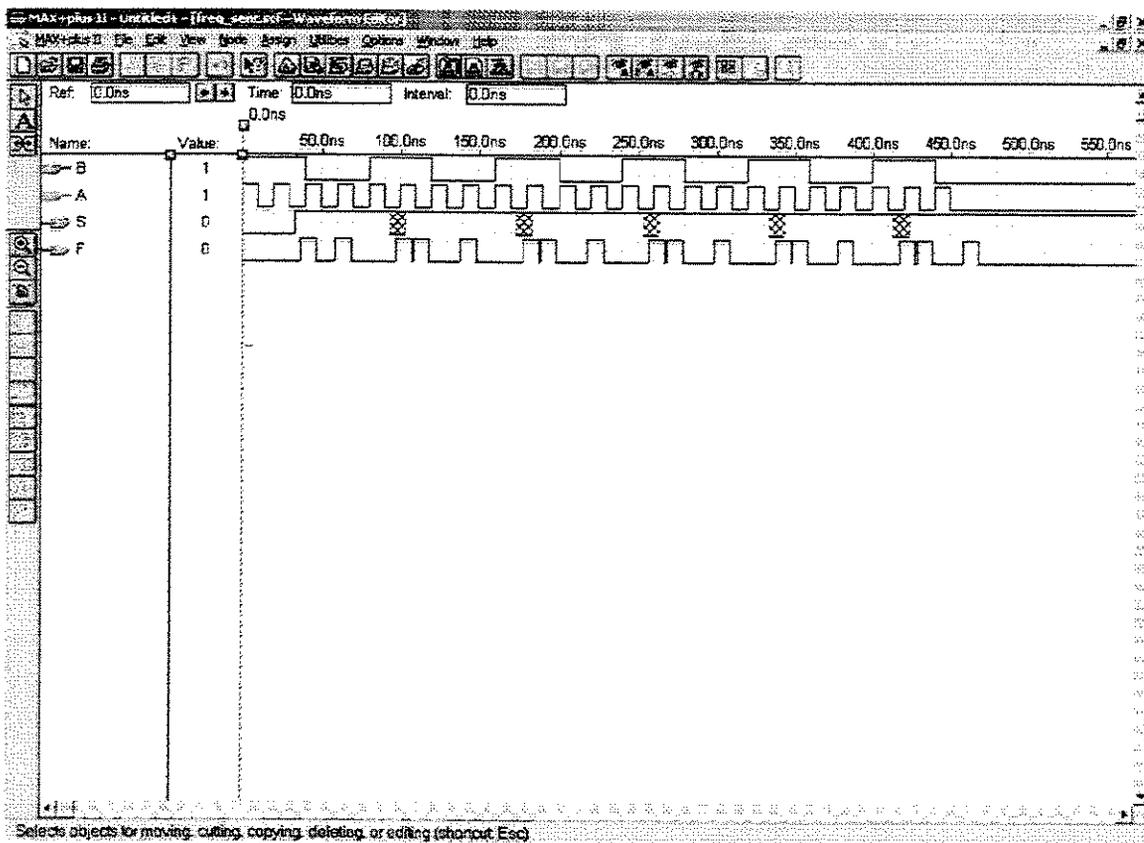


Figura V 11: Exemplo 1: controle de um motor cc através de PWM

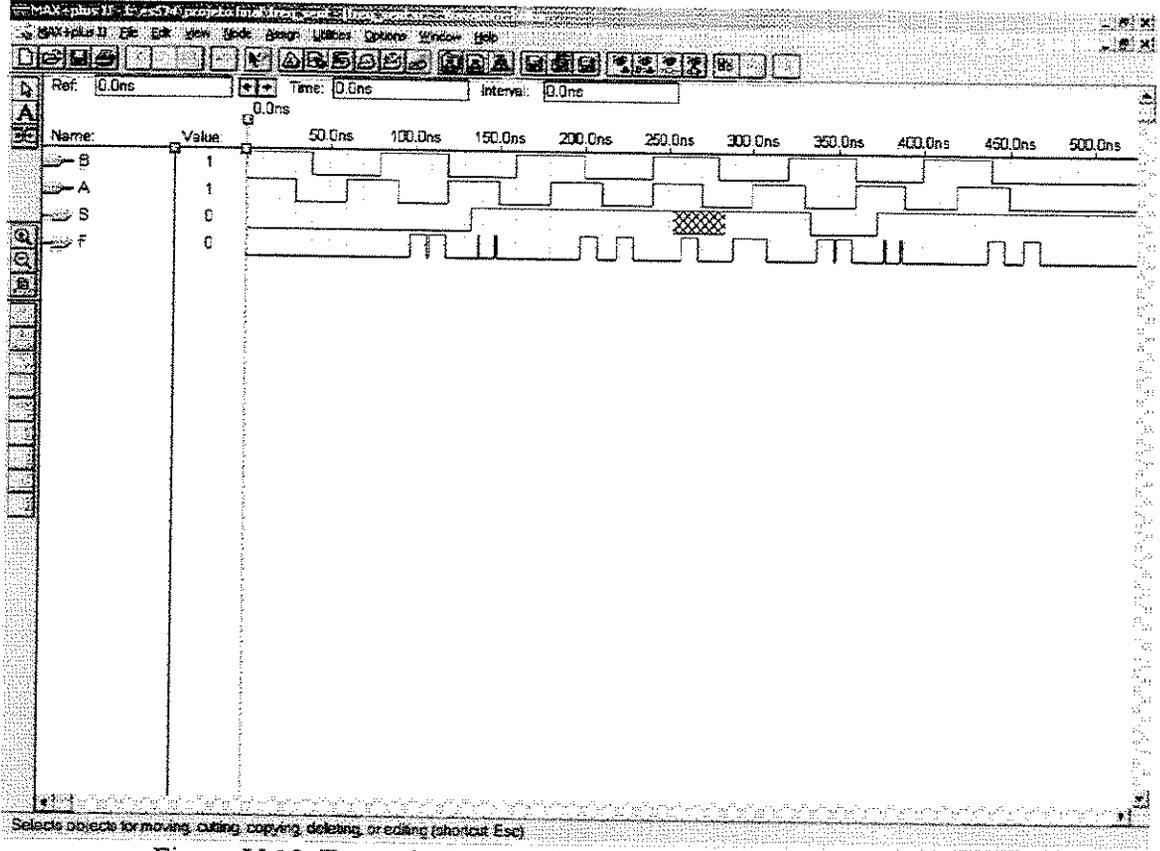


Figura V 12: Exemplo 2: controle de um motor cc através de PWM

## V 5 - Conclusões

Através da simulação dos módulos foi possível compreender melhor o funcionamento do controlador e visualizar a função de seus módulos principais.

A implementação realizada permite, se necessário, a introdução de novos níveis intermediários de potência, através do refino dos intervalos das tabelas do módulo Decpcm.

## ANEXO VI

### CÁLCULO DO MOMENTO DE INÉRCIA

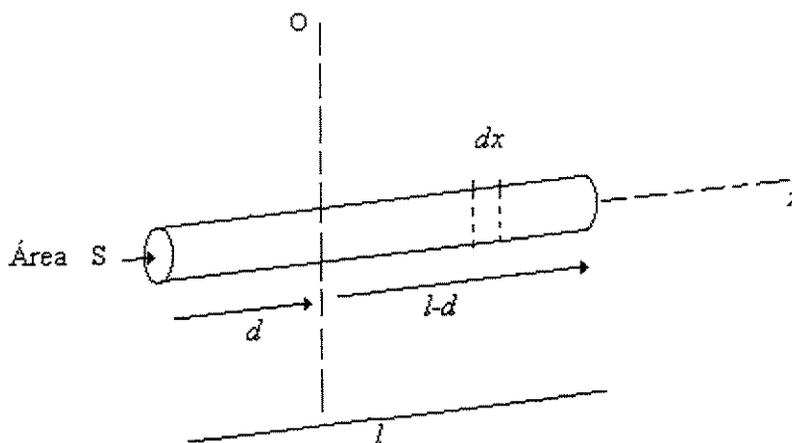


Figura VI.1 Esquema utilizado na dedução do momento de inércia

O momento de inércia é definido por:

$$I_0 = \int r^2 dm = \int x^2 dm$$

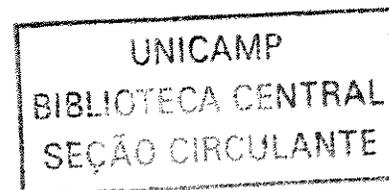
sendo

$$\rho = \frac{dm}{dV}$$

teremos,

$$dm = \rho s dx$$

Multiplicando e dividindo por  $l$ , temos:



$$dm = \rho \frac{sl}{l} dx$$

sendo  $sl=V$

$$dm = \rho \frac{V}{l} dx$$

e  $\rho V=m$

$$dm = \frac{m}{l} dx.$$

Assim,

$$I_0 = \int x^2 \frac{m}{l} dx$$

Colocando os limites de integração,

$$I_0 = \frac{m}{l} \int_{-d}^{l-d} x^2 dx$$

teremos:

$$I_0 = \frac{m}{l} \frac{1}{3} x^3 \Big|_{-d}^{l-d}$$

$$I_0 = \frac{1}{3} \frac{m}{l} [(l-d)^3 - (-d)^3]$$

$I_0 = \frac{1}{3} ml^2 - mld + md^2$ , onde  $I_0$  é o momento de inércia para a barra.

Teorema de Steiner:

$$I_A = I_0 + mh^2$$

onde:

$h$  é a distância do ponto de referência A ao centro de massa do corpo em estudo.