

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

# **Controle de Força de Robôs Manipuladores Interagindo com Ambientes de Elasticidade Não Linear**

Dissertação submetida à Universidade Federal de Santa Catarina  
como requisito parcial à obtenção do grau de

**Mestre em Engenharia Elétrica**

por

**Marcos Fonseca Mendes**  
(Bacharel em Engenharia Elétrica - UFMG)

Florianópolis, 01 de março de 1999

# Controle de Força de Robôs Manipuladores Interagindo com Ambientes de Elasticidade Não Linear

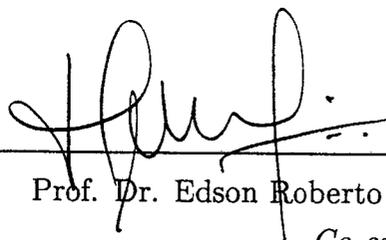
Eng. Marcos Fonseca Mendes

Esta dissertação foi julgada adequada para a obtenção do título de **Mestre em Engenharia** na especialidade **Engenharia Elétrica**, área de concentração **Controle, Automação e Informática Industrial**, e aprovada em sua forma final pelo curso de Pós-Graduação.

Florianópolis, 01 de março de 1999.



Prof. Ph.D. Werner Kraus Junior  
*Orientador*

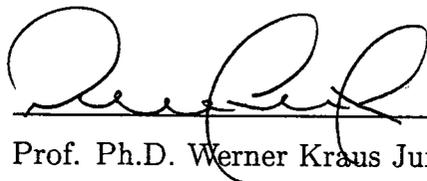


Prof. Dr. Edson Roberto De Pieri  
*Co-orientador*

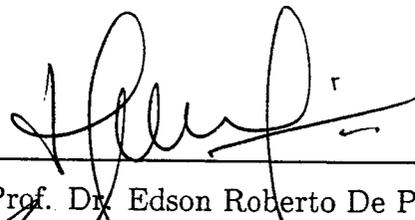


Prof. D.Sc. Ildemar Cassana Decker  
*Coordenador do curso de Pós-Graduação*

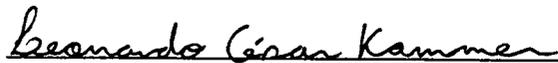
## Banca Examinadora



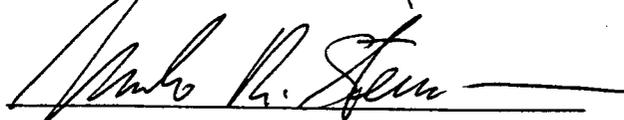
Prof. Ph.D. Werner Kraus Junior



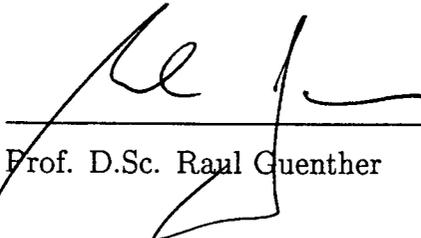
Prof. Dr. Edson Roberto De Pieri



Prof. Ph.D. Leonardo Cesar Kammer



Prof. Dr.-Ing. Marcelo Ricardo Stemmer



Prof. D.Sc. Raul Guenther

*Aos meus pais, Mendes (in memoriam) e Neréa.*

## *As Leis da Robótica*

*1ª: Um robô não pode ferir um ser humano ou, por omissão, permitir que um ser humano sofra algum mal.*

*2ª: Um robô deve obedecer as ordens que lhe sejam dadas por seres humanos, exceto nos casos em que tais ordens contrariem a Primeira Lei.*

*3ª: Um robô deve proteger sua própria existência, desde que tal proteção não entre em conflito com a Primeira e a Segunda Leis.*

*Lei Zero<sup>1</sup>: Um robô não pode causar mal à humanidade, nem permitir que ela própria o faça.*

**Isaac Asimov (1920-1992)**

---

<sup>1</sup> Criada posteriormente, por um robô que a intuiu no romance "Os Robôs e o Império".

# Agradecimentos

Primeiramente, agradeço a minha mãe, que mesmo distante sempre se fez tão perto, me confortando e alegrando.

Três pessoas ajudaram a plantar a semente de onde germinou este trabalho, a quem eu devo meus agradecimentos. Ao prof. Theldo Cruz Franqueira (PUC MG), por ter me incentivado a seguir o caminho da robótica, e aos profs. Luis Antonio Aguirre e Pedro Francisco Donoso Garcia (UFMG), por terem encorajado a minha aventura em Florianópolis.

Os meus professores da UFSC foram sem dúvida peças fundamentais neste trabalho. Agradeço aos meus orientadores, os profs. Werner Kraus Junior (*Take care with the SR man!*) e Edson Roberto De Pieri, por todo o trabalho de orientação e pela amizade. Agradeço também aos profs. da “fase de créditos”, especialmente aos profs. Antonio Augusto Rodrigues Coelho, Eugênio de Bona Castelan Neto e Raul Guenther. Além destes, agradeço aos membros da banca examinadora, pela participação, críticas e sugestões.

O meu muito obrigado aos colegas de luta do LCMI e dos outros laboratórios da Elétrica, principalmente os alunos: Alejandro, Amarilys, Augusto Loureiro, Carlos Montez, Carlos Rocha, Cristian, Eduardo (saydmin), Emerson, Fili, Ivana, Jorge, José de Oliveira, Julio Golin, Karina, Lau, Leandro Coelho, Luciano Rottawa, Lúcio Baretta, Micky, Nicole, Pierre, Rene (valeu a filmagem!), Ronei, Sandro, Simas e Sobral. Pelo convívio agradável no ambiente de trabalho e fora dele, pela amizade, pelas discussões acadêmicas, pelos churrascos extremamente necessários para o mestrado, pelos passeios e aventuras radicais e ecologicamente corretas, e pelas cervejas de sexta-feira. Agradecimentos especiais vão para o Carlos Venturo pela amizade e pelas discussões técnicas, e para o meu professor de espanhol, Cesar Torrico, *yo todavia voy a aprender a pronunciar la palabra rojo!*...

Agradeço aos membros da diretoria da APG gestão 97/98, Cláudia, Elpídio, Gustavo, Ivone, Jacob, Luis, Mauri, Maurício, Oscar e Rogério, e a todos os outros alunos da UFSC, que proporcionaram boas discussões políticas e acadêmicas, e lutam por uma Universidade pública, gratuita e de qualidade.

Algumas pessoas contribuíram para que a minha estadia em Floripa fosse a melhor possível, principalmente os amigos: Dimitri, Eduardo,

Fábio, Goiano, Kátia, Klaus, Pudim, Suzana e Tchola; as companheiras da Lingüística, Ina, Sandra Mara, Simone e Tatiana; os colegas do inglês; e os amigos do “Buteco do Cláudio”. Muito obrigado!

Agradeço à UFSC pelo suporte material, aos seus funcionários e demais professores que atuam indiretamente no curso de pós-graduação por fazerem a coisa funcionar, e ao CNPq pelo apoio financeiro.

E por último, mas não a menos importante, agradeço de coração à minha revisora ortográfica: Sandra Ghizoni Kafka. Obrigado pelos ótimos momentos e por toda a atenção, carinho, conselhos, bolos de chocolate e tudo mais que você me proporcionou. Te adoro!

## *Resumo*

*Este trabalho tem como objetivo principal o controle simultâneo de força e posição robusto de robôs manipuladores em contato com ambientes de elasticidade não linear. A finalidade do controlador é a automação do manuseio de peças flexíveis nos mais diversos tipos de linhas de montagem, proporcionando então, um avanço tecnológico na indústria de manufatura.*

*A estratégia usada é o controle híbrido de força e posição. Este esquema atende simultaneamente trajetórias de referências de força e de posição especificadas no espaço de tarefa. Para proporcionar robustez ao controlador, os sub-controladores são implementados utilizando controladores de modos deslizantes. Na apresentação da teoria de estrutura variável é implementado um controlador de posição no espaço de juntas num robô SCARA industrial. O controlador híbrido de força e posição projetado é simulado numericamente, apresentando resultados satisfatórios com relação à estabilidade e à robustez.*

**Palavras-chave:** *Robôs manipuladores; controle híbrido de força e posição; controle à estrutura variável*

## *Abstract*

*The main objective of this work is the robust simultaneous control of force and position of robot manipulators in contact with environments presenting nonlinear elasticity. The purpose of the controller is the automation of the handling of flexible parts in diverse types of assembly lines, thus providing a technological advance in the manufacturing industry.*

*The strategy used is the hybrid control of force and position. This scheme tracks simultaneously reference trajectories of force and position specified in task space. To provide robustness to the controller, the subcontrollers are implemented using sliding modes control. Along with the presentation of the theory of variable structure systems, a joint space position controller is implemented in an industrial SCARA robot. The hybrid controller of force and position is simulated numerically, presenting satisfactory results in relation to stability and robustness.*

**Key words:** *Robot manipulators; hybrid force and position control; variable structure control*

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Histórico . . . . .	1
1.2	Controle de Força . . . . .	2
1.2.1	Sensor de Força . . . . .	3
1.2.2	Controle Simultâneo de Força e Posição . . . . .	4
1.2.3	Aplicações Práticas do Controle Híbrido . . . . .	4
1.2.4	Revisão Bibliográfica . . . . .	5
1.3	Contribuições da Dissertação . . . . .	7
1.4	Estrutura da Dissertação . . . . .	8
<b>2</b>	<b>Robôs Manipuladores</b>	<b>9</b>
2.1	Introdução . . . . .	9
2.2	Cinemática . . . . .	10
2.2.1	Coordenadas Generalizadas . . . . .	10
2.2.2	Cinemática Direta . . . . .	11
2.2.3	Sistemas de Coordenadas - Espaços . . . . .	13
2.2.4	Matrizes Jacobianas . . . . .	14
2.3	Dinâmica . . . . .	16
2.3.1	Energia Cinética . . . . .	16
2.3.2	Energia Potencial . . . . .	18
2.3.3	Forças Generalizadas . . . . .	18
2.3.4	Formulação Lagrangeana . . . . .	20
2.3.5	Representação por Variáveis de Estado . . . . .	21
2.4	Exemplo: Manipulador 2DOF . . . . .	22
2.4.1	Cinemática Direta . . . . .	23
2.4.2	Equação Dinâmica . . . . .	23
2.4.3	Robô Manipulador SCARA . . . . .	24
2.5	Conclusões . . . . .	26

<b>3</b>	<b>Controle à Estrutura Variável</b>	<b>27</b>
3.1	Introdução . . . . .	27
3.2	Projeto . . . . .	28
3.2.1	Primeira Fase: Superfície de Deslizamento . . . . .	29
3.2.2	Segunda Fase: Lei de Controle . . . . .	32
3.3	<i>Chattering</i> . . . . .	37
3.4	Robustez . . . . .	40
3.5	Exemplo de Aplicação . . . . .	42
3.5.1	Simulação . . . . .	43
3.5.2	Implementação . . . . .	47
3.6	Conclusões . . . . .	50
<b>4</b>	<b>Controle Híbrido</b>	<b>52</b>
4.1	Introdução . . . . .	52
4.2	Espaço do Controle . . . . .	53
4.2.1	Controle Cartesiano . . . . .	53
4.2.2	Controle Cartesiano Usando a Jacobiana Transposta . . . . .	53
4.3	Fundamentos do Controlador Híbrido . . . . .	54
4.3.1	Movimentos Restritos . . . . .	55
4.3.2	Especificação de Tarefas . . . . .	57
4.4	Meios de Elasticidade Não Linear . . . . .	57
4.5	Projeto do Controlador . . . . .	58
4.5.1	Controle de Força . . . . .	59
4.5.2	Controle de Posição . . . . .	60
4.5.3	Lei de Controle . . . . .	60
4.6	Acoplamento / Estabilidade . . . . .	61
4.7	Exemplo de Manipulação de Materiais Flexíveis . . . . .	62
4.7.1	Descrição do Problema . . . . .	63
4.7.2	Problema de Controle . . . . .	64
4.8	Resultados de Simulação . . . . .	66
4.8.1	Parâmetros Nominais . . . . .	66
4.8.2	Variações Paramétricas da Carga . . . . .	68
4.8.3	Perturbações e Ruídos . . . . .	72
4.9	Conclusões . . . . .	74
<b>5</b>	<b>Conclusões e Perspectivas</b>	<b>75</b>
5.1	Conclusões . . . . .	75

5.2	Perspectivas . . . . .	77
<b>A</b>	<b>Parâmetros do Manipulador SCARA</b>	<b>78</b>
A.1	Dimensões e Massas . . . . .	78
A.2	Valores Limites . . . . .	78
A.3	Constantes de Tempo dos Motores . . . . .	79
A.3.1	Motor bls55 . . . . .	79
A.3.2	Motor bls72 . . . . .	79
<b>B</b>	<b>Códigos Fonte</b>	<b>80</b>
B.1	Módulo MathMen . . . . .	80
B.2	Módulo StateCtrl . . . . .	81

# Lista de Figuras

2.1	Robô Manipulador de Dois Graus de Liberdade . . . . .	22
2.2	Manipulador SCARA ( <i>Selective Compliance Assembly Robot Arm</i> ) . . . . .	25
3.1	Superfície de Deslizamento . . . . .	30
3.2	Trajectoria do Estado - Sistema de Primeira Ordem . . . . .	33
3.3	<i>Chattering</i> . . . . .	38
3.4	Camada Limite . . . . .	39
3.5	Função $\text{sat}(n/\epsilon)$ . . . . .	39
3.6	Posições das Juntas ( $\theta_1 = q1$ ; $\theta_2 = q2$ ) - Simulação . . . . .	46
3.7	Velocidades das Juntas ( $\omega_1 = w1$ ; $\omega_2 = w2$ ) - Simulação . . . . .	46
3.8	Proximidade das Superfícies ( $\sigma_1 = s1$ ; $\sigma_2 = s2$ ) - Simulação . . . . .	47
3.9	Sinais de Controle ( $u1$ ; $u2$ ) - Simulação . . . . .	47
3.10	Erros de Posição das Juntas ( $\theta_d - \theta$ ) - Simulação . . . . .	48
3.11	Erros de Velocidade das Juntas ( $\omega_d - \omega$ ) - Simulação . . . . .	48
3.12	Posições das Juntas ( $\theta_1 = q1$ ; $\theta_2 = q2$ ) - Implementação Real . . . . .	49
3.13	Velocidades das Juntas ( $\omega_1 = w1$ ; $\omega_2 = w2$ ) - Implementação Real . . . . .	49
3.14	Proximidade das Superfícies ( $\sigma_1 = s1$ ; $\sigma_2 = s2$ ) - Implementação Real . . . . .	49
3.15	Sinais de Controle ( $u1$ ; $u2$ ) - Implementação Real . . . . .	50
3.16	Erros de Posição das Juntas ( $\theta_d - \theta$ ) - Implementação Real . . . . .	50
3.17	Erros de Velocidade das Juntas ( $\omega_d - \omega$ ) - Implementação Real . . . . .	51
4.1	Controlador Cartesiano Usando a Jacobiana Transposta . . . . .	54
4.2	Controlador Híbrido de Força e Posição . . . . .	54
4.3	Inserção de um parafuso numa porca . . . . .	55
4.4	Controlador Híbrido de Força e Posição Detalhado . . . . .	61
4.5	Robô Manipulador 2DOF inserindo uma chapa flexível . . . . .	62
4.6	Mola Não Linear . . . . .	64
4.7	Componentes de Força x da Mola . . . . .	65
4.8	Componentes de Força y da Mola . . . . .	65
4.9	Proximidade das Superfícies ( $\sigma_p = s1$ ; $\sigma_f = s2$ ) . . . . .	67

4.10	Sinais de Controle ( $u_1 ; u_2$ ) . . . . .	67
4.11	Forças Externas ( $F_x = f_x ; F_y = f_y$ ) . . . . .	68
4.12	Posições no Espaço de Tarefa ( $\mathbf{x} ; \mathbf{y}$ ) . . . . .	68
4.13	Posição do Efetuador Final . . . . .	69
4.14	Sinais de Controle ( $u_1 ; u_2$ ) - Chapa Menor . . . . .	70
4.15	Forças Externas ( $F_x = f_x ; F_y = f_y$ ) - Chapa Menor . . . . .	70
4.16	Posições no Espaço de Tarefa ( $\mathbf{x} ; \mathbf{y}$ ) - Chapa Menor . . . . .	70
4.17	Sinais de Controle ( $u_1 ; u_2$ ) - Chapa Maior . . . . .	71
4.18	Forças Externas ( $F_x = f_x ; F_y = f_y$ ) - Chapa Maior . . . . .	71
4.19	Posições no Espaço de Tarefa ( $\mathbf{x} ; \mathbf{y}$ ) - Chapa Maior . . . . .	71
4.20	Perturbações ( $p_1 ; p_2$ ) . . . . .	72
4.21	Sinais de Controle ( $u_1 ; u_2$ ) - Com Perturbação e Ruído . . . . .	73
4.22	Forças Externas ( $F_x = f_x ; F_y = f_y$ ) - Com Perturbação e Ruído . . . . .	73
4.23	Posições no Espaço de Tarefa ( $\mathbf{x} ; \mathbf{y}$ ) - Com Perturbação e Ruído . . . . .	73

# Lista de Tabelas

4.1	Restrições Naturais . . . . .	56
4.2	Restrições Artificiais . . . . .	56
A.1	Dimensões e Massas do Manipulador SCARA . . . . .	78
A.2	Valores Limites do Manipulador SCARA . . . . .	78

# Capítulo 1

## Introdução

*“A imaginação é mais importante do que o conhecimento”*

**Albert Einstein (1879-1955)**

Peças flexíveis são usadas na indústria durante todo o processo de manufatura de bens duráveis e não duráveis. Os exemplos incluem a fabricação de carros, dobra de folhas de metal, inserção de tubos e fios flexíveis, colagem de faixas de borracha, etc. A importância destas aplicações industriais é a maior motivação para este trabalho.

Existem vários tópicos envolvendo a automação destes processos utilizando robôs manipuladores, entre eles o manuseio da carga, o planejamento e o controle da execução da tarefa, e o monitoramento da tarefa. Destes, o controle é o assunto deste trabalho. O objetivo principal é a solução do controle simultâneo de força e posição, através de um controlador híbrido, na manipulação de materiais flexíveis.

### 1.1 Histórico

Os robôs industriais tiveram sua origem em duas tecnologias anteriores: controle numérico para máquinas ferramenta, que consiste em ações baseadas em dados armazenados, e manipulação remota, que se trata da execução de tarefas à distância [2].

Os primeiros robôs foram concebidos como dispositivos de posicionamento em malha aberta [18]. Eles trabalhavam com pouca ou nenhuma realimentação do processo em que participavam.

A realidade hoje em dia é outra. A robótica vem se desenvolvendo bastante, utilizando os conhecimentos de vários campos diferentes, dentre eles: mecânica, estática e dinâmica, eletrônica, teoria de controle, sensores, visão, processamento de sinais, programação de computadores, inteligência artificial e manufatura.

Este crescente desenvolvimento é fruto da necessidade de execução cada vez melhor de tarefas cada vez mais complexas, o que exige portanto, tecnologias mais avançadas.

O controle de força de robôs manipuladores começou com o controle de manipuladores remotos e braços artificiais nas décadas de 50 e 60. No final da década de 60 e na de 70, os primeiros controladores com realimentação de força utilizando computadores foram testados e várias abordagens para a criação de estratégias surgiram.

Hoje em dia, o controle de força está atrasado em termos de sofisticação tanto na teoria quanto em aplicações na indústria [79]. A melhoria dos sistemas de controle está amparada pelo desenvolvimento de *hardware*, haja vista que os atuadores, sensores e a capacidade computacional atuais não estão limitando o progresso. É portanto necessário um esforço maior para identificar e resolver os problemas teóricos básicos que impedem o avanço. Dominando estes pontos, serão obtidos sistemas robóticos cada vez mais eficientes e confiáveis. Este trabalho visa contribuir nesta direção.

## 1.2 Controle de Força

As aplicações de robôs manipuladores podem ser divididas em duas grandes categorias do ponto de vista de controle:

- controle de posição;
- controle de força.

No *controle de posição* deseja-se controlar a posição<sup>1</sup> do efetuador final do manipulador. Nas tarefas desta categoria não ocorre o contato entre o manipulador e o meio, e normalmente deseja-se controlar apenas a trajetória (posição e algumas vezes velocidade) realizada pelo efetuador final. Algumas aplicações são pintura a spray, solda a ponto e transporte de peças. Note que nesta última tarefa existe o contato entre o manipulador e a peça manipulada, porém não há um controle da força exercida sobre a mesma; existe apenas um mecanismo para segurá-la.

No *controle de força* existe o contato entre manipulador e o meio, e daí a necessidade de se controlar a força<sup>2</sup> proveniente deste contato. As tarefas desta categoria são mais complexas, pois existem restrições de movimentos [5, 10] e a dinâmica do meio, que pode ser não linear [6, 36, 65] ou até mesmo desconhecida, deve ser considerada nas análises. Na indústria são encontradas muitas aplicações para esta categoria de

---

<sup>1</sup>Entenda-se aqui posição como posição e orientação.

<sup>2</sup>Entenda-se aqui força como força e torque.

controle como, por exemplo, montagem (encaixe de peças), perfuração, rebarbação, esmerilhamento, polimento, corte, escavação, operação de manivela, fresagem, explorações submarinas e espaciais, prótese de membros humanos, além de muitas outras aplicações de manufatura [2, 38, 39, 59].

O estudo de controle de força de robôs manipuladores ainda não está desenvolvido ao ponto de ser utilizado em aplicações industriais, como é o caso do controle de posição. Este último já atingiu uma maturidade considerável, ou seja, existem soluções bastante eficientes, apesar de ainda serem feitas muitas pesquisas nesta área. Por outro lado, o controle de força é uma área de concentração emergente e não apresenta a eficiência desejada em algumas aplicações ou nem mesmo se consegue executá-las com sucesso. Esta área necessita, portanto, de pesquisas mais profundas.

É utilizado neste trabalho o controle explícito de força. Esta estratégia envolve o comando direto e a medição dos valores de força, com o intuito de que a saída (força medida) siga a entrada (força de referência) o mais próximo possível. O controlador compara os sinais de referência e medido, os processa, e fornece um sinal de atuação para o robô manipulador.

Para medir a força de contato entre o robô e o meio, tanto em robôs manipuladores como em robôs móveis, é utilizado um *sensor de força* [13, 29, 30, 42, 83]. A seguir este sensor é apresentado.

### 1.2.1 Sensor de Força

As forças e torques que surgem na interação de um robô com o meio são medidos através da deformação ou deslocamento de um elemento elástico do sensor de força.

Em robôs manipuladores, normalmente o sensor de força é colocado no seu punho, entre a extremidade do último elo e o efetuador final. Sensores no punho são os mais amplamente utilizados em robótica [29].

A força de contato é o produto de um valor de rigidez pela deformação do contato. Esta deformação inclui a deformação do sensor de força, da estrutura do manipulador, e, possivelmente, do ambiente.

Para a construção dos sensores, o transdutor mais utilizado é o *strain gauge*<sup>3</sup> [19]. Sensores de força utilizando *strain gauges* possuem uma alta sensibilidade e boa precisão. Nestes sensores, a deformação causa a mudança da resistência elétrica do *strain gauge* montado sobre um elemento elástico.

---

<sup>3</sup>Neste trabalho é utilizado o termo *strain gauge* em inglês por falta de uma tradução consagrada na literatura em português.

Para uma completa informação sobre as forças de contato, o sensor deve fornecer os seis componentes mostrados na Equação (2.29): três de forças translacionais e três de torques. Porém, o controlador híbrido utiliza apenas os dados das direções controladas em força, ignorando os outros componentes.

O vetor fornecido pelo sensor é representado no seu sistema de coordenadas. Para ser utilizado na malha de controle, geralmente, deve ser feita uma transformação [29] para representá-lo no sistema de coordenadas da base do manipulador, como é o caso do controlador simultâneo de força e posição apresentado na próxima sub-seção.

### 1.2.2 Controle Simultâneo de Força e Posição

Nas tarefas onde existe o contato do robô manipulador com o ambiente, as operações do robô podem ser divididas em três fases [83]:

- movimento no espaço livre;
- contato físico;
- aplicação de uma força.

A fase de operação estudada neste trabalho é a terceira. Nestas operações raramente deseja-se apenas o controle puro de força. A grande maioria das tarefas envolvendo o contato entre o manipulador e o meio com uma certa força sendo aplicada também necessita do controle de posição.

Considere, por exemplo, um robô manipulador escrevendo num quadro negro [2]. O controle de posição puro é claramente não recomendado, uma vez que pequenos erros de posição na direção ortogonal ao quadro podem resultar em falta de contato com o quadro, ou, ao contrário, quebrar o giz. Enquanto o controle de posição se faz necessário ao longo do plano do quadro, algum tipo de controle de força é necessário na direção ortogonal ao quadro, para que um contato adequado do giz seja garantido.

Uma estrutura de controle capaz de desempenhar esta tarefa ilustrativa pode, em princípio, ser utilizada em outras aplicações industriais. A sub-seção a seguir discute algumas destas.

### 1.2.3 Aplicações Práticas do Controle Híbrido

Quando os robôs foram introduzidos, eles eram usados basicamente em tarefas antes realizadas por pessoas. Esta foi uma aplicação natural, uma vez que o novo dispositivo era capaz de realizar algumas tarefas mais rapidamente e de forma mais confiável que os humanos [38].

Hoje em dia a área de atuação dos robôs cresceu, e continua crescendo, significativamente. A indústria de manufatura apresenta várias tarefas onde o controle simultâneo de força e posição se faz necessário. Abaixo são descritos dois exemplos de aplicações industriais onde é interessante o controle híbrido de força e posição apresentado neste trabalho.

### **Esmerilhamento e Polimento**

Muitas vezes na manufatura é necessário obter-se uma superfície lisa, para uma boa aparência ou mesmo para questões de funcionamento do produto. Nestes casos é feito um esmerilhamento e/ou polimento. Várias indústrias têm utilizado robôs para realizar esmerilhamentos e polimentos.

As superfícies que se deseja esmerilhar ou polir apresentam irregularidades devido ao processo de produção ao qual elas foram submetidas anteriormente, como soldagem, perfuração ou fresagem, por exemplo. Portanto, um controle de força é necessário para que a ferramenta de esmerilhamento/polimento exerça uma pressão adequada na superfície trabalhada.

### **Montagem**

Em vários pontos de uma linha de montagem, peças devem ser encaixadas. As operações de montagem são extremamente tediosas devido a sua natureza repetitiva. Assim, este tipo de tarefa é um campo atrativo para o uso de robôs manipuladores.

Estas tarefas necessitam de um posicionamento bastante preciso. A habilidade de medir e controlar as forças de contato do manipulador pode ser uma alternativa para se conseguir esta precisão, pois pequenas variações de posição geram grandes forças de contato [29, 37, 58, 59, 78]. Uma realimentação de força melhora a capacidade de sensoriamento externo do robô permitindo que ele se adapte melhor a pequenos erros de posição das peças existentes durante a montagem, aumentando dessa forma a robustez do controlador.

Além disso, operações de montagem normalmente requerem que o robô exerça uma certa força para garantir o encaixe das peças, daí a necessidade de um controle de força.

#### **1.2.4 Revisão Bibliográfica**

As estratégias para o controle simultâneo de força e posição restringem-se basicamente a apenas três:

- controle de impedância;

- controle paralelo de força e posição;
- controle híbrido de força e posição.

No *controle de impedância* [2, 5, 29, 32] é especificado como o robô deve interagir com as restrições externas, ou seja, é definido o relacionamento dinâmico desejado entre o manipulador e os objetos no espaço de trabalho. A impedância é definida como a *função de transferência* entre a força externa agindo no manipulador e o seu deslocamento. A maior dificuldade de implementação desta estratégia é a necessidade do modelo dinâmico do meio.

No *controle paralelo de força e posição* [10], a estratégia mais recente das três, são controladas tanto a força imposta como a posição do efetuador final utilizando todas as variáveis medidas disponíveis (forças e posições), dando prioridade ao controle de força. Esta estratégia não é eficiente em situações onde a força desejada não é exatamente normal à superfície de restrição. Nestes casos ocorre um deslizamento do efetuador final ao longo da superfície de contato, conforme discutido em [11].

E por fim, no *controle híbrido de força e posição* [2, 43, 53], são controladas a posição e a força desejadas em dois sub-espacos mutuamente independentes. O presente trabalho estuda esta estratégia. O controle híbrido é talvez a estratégia mais amplamente utilizada para controlar simultaneamente força e posição de robôs manipuladores [11, 61].

## Principais Trabalhos

Vários trabalhos envolvendo o controle simultâneo de força e posição de robôs manipuladores já foram desenvolvidos. Nesta seção alguns dos mais relevantes encontrados na literatura são apresentados.

Raibert e Craig [53], utilizando a descrição da tarefa de Mason [47], apresentaram um dos primeiros trabalhos de controle simultâneo de força e posição, que serviu de inspiração para uma série de trabalhos de diversos autores [16, 22, 49, 60, 66]. Neste trabalho foram apresentados os fundamentos básicos do controle híbrido de força e posição. Eles utilizaram um mecanismo de seleção das direções controladas em força. O controlador é implementado utilizando uma lei de controle PID (proporcional mais integral mais derivativa) para a malha de posição e uma lei de controle PI (proporcional mais integral) com um termo de alimentação direta do sinal de força desejado para a malha de força [20, 80].

Posteriormente, De Luca e Manes [16] redefiniram o controle híbrido de força e posição para tarefas que incluem restrições dinâmicas além das cinemáticas. Desta forma, a abordagem do controle híbrido tornou-se mais geral.

Wang e McClamroch [75] usaram o método direto de Lyapunov para desenvolver um controlador de força e posição para manipuladores restritos pelo contato com ambientes rígidos. Este método elimina as dificuldades devido às não linearidades da dinâmica do manipulador e o acoplamento entre a dinâmica do manipulador e as restrições impostas.

Chiaverini e Sciavicco [10] apresentaram uma abordagem conceitualmente diferente do controle simultâneo de força e posição, o controle paralelo. Ao contrário do controle híbrido, neste ambas as variáveis, força e posição, são utilizadas para o controle de uma mesma direção do espaço de tarefa sem qualquer mecanismo de seleção. A filosofia do controle paralelo é projetar a ação de controle de força prevalecendo sobre a ação de controle de posição. Quando erros ocorrem, é dada prioridade do controle de força sobre o de posição. Uma forma de se implementar este domínio é utilizar um controlador PI para a malha de força e um PD (proporcional mais derivativo) para a malha de posição.

Ferretti *et al.* [22, 23] propuseram uma estratégia por eles chamada de *controle triangular*<sup>4</sup>. O objetivo desta estrutura é exercer uma ação de controle de força que rejeite as perturbações devido ao movimento, sem interferir no sistema de controle de posição. Para o projeto da lei de controle de força é utilizado apenas o modelo cinemático diferencial (matriz Jacobiana) e não é feita nenhuma compensação de dinâmica baseada no modelo.

Do ponto de vista de estrutura de controle, o presente trabalho utilizou principalmente as idéias e conceitos apresentados nos trabalhos de Mason [47] e Raibert e Craig [53].

### 1.3 Contribuições da Dissertação

É focado neste trabalho um controlador simultâneo de força e posição. Uma fundamentação teórica completa e satisfatória para a análise e projeto deste tipo de controlador não está disponível ainda [66]. Este trabalho visa contribuir para o avanço neste sentido através do estudo de uma das abordagens deste controlador.

Um controlador híbrido de força e posição para robôs manipuladores em contato com meios cuja elasticidade é não linear, por exemplo, peças flexíveis como chapas metálicas e peças plásticas, é estudado e implementado em nível de simulação.

Parte do controlador projetado neste trabalho é implementado em um robô industrial SCARA, utilizando a linguagem de programação orientada a objetos em tempo real XOberon [50, 56].

---

<sup>4</sup>Traduzido do original “*triangular control*”.

Controladores deste tipo, com certeza proporcionarão grandes avanços tecnológicos nos mais diversos tipos de indústrias, tornando economicamente viável o aumento do uso de robôs manipuladores.

Deve também ser ressaltada a contribuição para o Laboratório de Robótica da UFSC, resultado das discussões e intercâmbio de informações entre os pesquisadores envolvidos com o robô SCARA, durante a elaboração desta dissertação. Esta cooperação favoreceu os primeiros avanços na utilização do novo equipamento do laboratório e facilitará o trabalho de pesquisadores que venham a utilizar o robô.

## 1.4 Estrutura da Dissertação

O presente trabalho é dividido em 5 capítulos.

O Capítulo 2 trata dos robôs manipuladores. São apresentados os elementos de um manipulador bem como as definições e conceitos necessários para o estudo deste tipo de planta. No final do capítulo é modelado o manipulador utilizado para simulações e implementações práticas deste trabalho.

O Capítulo 3 apresenta a teoria de controle à estrutura variável. Este capítulo apresenta todo o desenvolvimento necessário para o projeto deste tipo de controlador. Além disso, são abordadas as vantagens e desvantagens de se utilizar esta lei de controle. Por fim é apresentado um exemplo de projeto, com simulação e implementação em bancada.

O Capítulo 4 apresenta o produto deste trabalho: um controlador híbrido de força e posição. Neste capítulo são apresentados os fundamentos do controlador híbrido, passando por uma abordagem mais minuciosa da malha de controle de força. Utilizando o modelo do robô obtido no Capítulo 2 e a metodologia de projeto do controlador à estrutura variável, Capítulo 3, é feito o projeto de um controlador híbrido e são apresentados resultados de simulações numéricas.

Finalmente, o Capítulo 5 traz as conclusões gerais do trabalho e as perspectivas de continuidade de estudo nesta área.

# Capítulo 2

## Robôs Manipuladores

Neste capítulo é feita uma introdução aos *robôs manipuladores*, definindo os elementos que o compõe e descrevendo sua construção física. São apresentados também os conceitos necessários para o desenvolvimento dos capítulos posteriores.

É obtido um modelo que representa o comportamento dinâmico de um robô manipulador genérico, usando a formulação de Lagrange. Para isto, é necessário passar pelas questões cinemáticas, e definir os sistemas de coordenadas para a análise.

Por fim, é apresentado o modelo de um robô manipulador de dois graus de liberdade, o qual é utilizado para todas as simulações e implementações práticas deste trabalho.

### 2.1 Introdução

Para a análise e síntese de controladores clássicos é indispensável a modelagem do sistema físico (*planta*), ou seja, possuir um modelo matemático que represente bem o sistema real.

O modelo matemático de um robô manipulador consiste de equações diferenciais de movimento. Elas fornecem uma descrição do relacionamento entre os torques aplicados nas juntas (entrada) e o movimento dos elos (saída). Para obter este modelo duas formulações básicas podem ser utilizadas:

- Newton-Euler;
- Lagrange.

A primeira formulação é obtida pela interpretação direta da *segunda lei do movimento de Newton* [2, 4, 43, 67], que descreve um sistema dinâmico em função das forças e momentos que agem nos elos do robô manipulador, incluindo as forças de interação entre eles.

A *formulação Lagrangeana* descreve o sistema dinâmico utilizando as definições físicas de *trabalho* e *energia*. Este método é desenvolvido no sistema de coordenadas generalizadas e elimina automaticamente todas as forças que não realizam trabalho, como forças de restrição cinemática [22, 48, 82], por exemplo.

A vantagem da formulação Lagrangeana sobre a de Newton-Euler é que as equações dinâmicas obtidas são expressões em forma fechada compactas que fornecem as acelerações das juntas em função dos torques nelas aplicados, ou seja, o modelo desejado.

O modelo levantado neste capítulo não levará em conta a dinâmica dos atuadores (motores e caixas de redução), bem como as forças devido a atrito (de Coulomb e viscoso) [13, 28, 41, 43, 67, 68]. Além disso, o robô será considerado rígido, ou seja, sem flexibilidades nas juntas [5, 42, 43, 54, 68]. Com estas considerações o modelo obtido é mais simples e representa bem a planta para o propósito deste trabalho, uma vez que a velocidade do manipulador em tarefas com restrições geométricas é baixa.

## 2.2 Cinemática

Um robô manipulador, do ponto de vista mecânico, é basicamente uma série de corpos rígidos, chamados de *elos*, conectados numa estrutura cinemática. Os elos desta cadeia possuem *massa* e uma certa liberdade de movimento [2, 43, 67, 81].

A localização de cada elo é completamente descrita através da sua *posição* e *orientação*, em relação a um *referencial inercial* [4, 25]. Normalmente utiliza-se como referência a base do robô manipulador, por se tratar, na maioria dos casos, de uma parte estática do manipulador (fixa no chão).

### 2.2.1 Coordenadas Generalizadas

Para análise do manipulador, cada elo é numerado de 0 a  $n$ , começando do elo da base chegando até o elo mais distante. Os elos são interconectados através de uma *junta*, que pode ser *prismática* (movimentos lineares) ou *de revolução* (movimentos de rotação).

Cada junta do manipulador é acionada por um *atuador* (motor elétrico, cilindro hidráulico, etc.), que causa o movimento dos elos. O número de juntas de robôs industriais comercialmente disponíveis hoje em dia variam de três a sete.

Em geral a descrição da posição e orientação dos elos é feita utilizando a notação de *Denavit-Hartenberg* [17] por se tratar de uma forma eficiente. Este método utiliza representações matriciais e usa um número mínimo de parâmetros para descrever completamente os relacionamentos cinemáticos.

O deslocamento de cada junta é um ângulo de rotação  $\theta$  ou um deslocamento linear  $d$ , de acordo com o tipo da junta. Em geral o deslocamento da junta  $i$  é denotado por  $q_i$ , que é definido como:

- $q_i = d =$  um deslocamento linear, para juntas prismáticas;
- $q_i = \theta =$  um ângulo, para juntas de revolução.

Assim, o conjunto de *coordenadas generalizadas* [2, 5, 67, 46] de um robô manipulador com  $n$  juntas é um vetor do tipo:

$$q = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix}. \quad (2.1)$$

Note que este conjunto de coordenadas é não redundante, isto é, ele contém o número mínimo de elementos que descrevem completamente as configurações do robô manipulador. A formulação Lagrangeana para descrever a dinâmica do manipulador é desenvolvida utilizando estas coordenadas generalizadas.

O espaço onde é definido o vetor de posição das juntas  $q$  (2.1) é chamado de *espaço de juntas* [5, 13, 81]. A dimensão deste espaço ( $n$ ), que é igual ao número de juntas<sup>1</sup>, define o número de *graus de liberdade* [5, 13, 81] do manipulador.

Nos robôs industriais normalmente são medidas a posição e velocidade de cada uma de suas juntas. No caso de juntas de revolução, os robôs possuem em cada junta um *encoder óptico* para medir a sua posição e um *tacômetro* para medir a sua velocidade [19].

### 2.2.2 Cinemática Direta

Na extremidade livre do último elo da cadeia (elo número  $n$ ) está o *efetuador final*. Dependendo da aplicação, o efetuador final pode ser uma garra, uma pistola de solda, um eletro-ímã ou algum outro dispositivo. As tarefas executadas por um robô manipulador sempre utilizam o efetuador final e são compostas usando basicamente dois elementos fundamentais:

- movimentar o efetuador final seguindo uma trajetória desejada (carregando ou não uma carga);

---

<sup>1</sup>É considerado aqui que os manipuladores não possuem mecanismos paralelos.

- exercer uma força desejada sobre um objeto quando o efetuador final estiver em contato com ele.

Assim, é fácil perceber que a posição e orientação de maior interesse no manipulador é a do efetuador final, uma vez que as tarefas são executadas utilizando alguma ferramenta<sup>2</sup> fixa neste.

A posição e orientação do efetuador final podem ser expressas como:

$$p = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad (2.2)$$

onde  $p_x$ ,  $p_y$  e  $p_z$  são as coordenadas  $x$ ,  $y$  e  $z$ , respectivamente, com relação ao sistema de coordenadas da base, que representam uma posição no espaço, e:

$$\phi = \begin{bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{bmatrix} \quad (2.3)$$

onde  $\phi_x$ ,  $\phi_y$  e  $\phi_z$  são os três ângulos que representam uma orientação no espaço de forma única, como por exemplo, os *ângulos de Euler* [2, 5, 43, 81].

Uma forma mais compacta de representar a posição e orientação do efetuador final é através de um único vetor formado pelos dois anteriores<sup>3</sup>:

$$x = \begin{bmatrix} p \\ \phi \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}. \quad (2.4)$$

Este vetor pode ser determinado pela posição e orientação de cada elo individualmente através de *transformações homogêneas* [2, 6, 81] consecutivas. Como a posição e orientação de cada elo pode ser determinada através da sua coordenada generalizada  $q_i$  e da posição e orientação do elo anterior (elo  $i - 1$ ), o vetor  $x$  (2.4) pode ser obtido em função das coordenadas generalizadas (posição de todas as juntas):

$$x = \Lambda(q). \quad (2.5)$$

<sup>2</sup>A *garra* para segurar cargas é aqui considerada como sendo uma ferramenta.

<sup>3</sup>Note que a partir de um ponto de vista matemático rigoroso,  $\phi$  (2.3) não é um vetor, pois a *propriedade de comutatividade* não é verificada [2, 5, 81]. Ainda assim, para as operações de interesse,  $x$  comporta-se como vetor, justificando o emprego do nome.

A equação acima é chamada de equação *cinemática direta* do manipulador [2, 5, 13, 43, 67, 81].

### 2.2.3 Sistemas de Coordenadas - Espaços

O vetor de posição e orientação  $x$  (2.4) é definido no sistema de coordenadas cartesiano, no qual as tarefas são executadas. Por este motivo, este sistema é normalmente chamado de *espaço de tarefa*. Alguns autores preferem chamá-lo de *espaço cartesiano* [2, 5, 13, 43, 67, 81].

Note que um manipulador não consegue alcançar todos os pontos do espaço de tarefa. O conjunto dos pontos alcançáveis por um manipulador é o seu *espaço de trabalho* [2, 81].

No caso geral, são utilizadas três coordenadas para especificar uma posição e outras três para especificar uma orientação, como pode ser visto em (2.2) e (2.3). Portanto, a dimensão do espaço de tarefa é no máximo igual a seis ( $m \leq 6$ ).

Um número reduzido de coordenadas do espaço de tarefa pode ser utilizado de acordo com as características geométricas da tarefa. No caso de uma tarefa realizada no plano por exemplo, três coordenadas ( $m = 3$ ) são suficientes: duas para especificar a posição no plano e outra para especificar a orientação (dada por um ângulo).

Transformações entre os espaços de tarefa e de juntas são possíveis. A transformação do espaço de juntas para o espaço de tarefa é feito de forma trivial, utilizando a cinemática direta (2.5). Já a transformação inversa, denominada *cinemática inversa*, não é uma tarefa fácil, principalmente para robôs manipuladores com muitos graus de liberdade ( $n > 3$ ).

A transformação cinemática inversa é obtida analítica ou numericamente a partir da equação cinemática direta (2.5). Além disso, outras técnicas para o cálculo da cinemática inversa tem surgido. Chiacchio *et al.* propõem em [9] o uso de um sistema dinâmico para obter-se a cinemática inversa utilizando a cinemática direta (considerada bem conhecida) e uma inversão do Jacobiano do manipulador. Outro exemplo é o uso de *algoritmos genéticos* [62].

Uma das dificuldades para obter-se a cinemática inversa é a possível *redundância cinemática* [2, 67, 81]. Ela existe onde a dimensão do espaço de tarefa é menor que a do espaço de juntas ( $m < n$ ). Neste caso, uma mesma posição e orientação do efetuador final pode ser alcançada através de configurações diferentes das juntas do manipulador. Note que um mesmo manipulador pode ser redundante para uma determinada tarefa mas não para uma outra, pois o número de coordenadas no espaço de tarefa  $m$  é definido pela própria tarefa.

### 2.2.4 Matrizes Jacobianas

Considerando apenas movimentos infinitesimais do efetuador final, o seguinte relacionamento diferencial é obtido a partir da Equação (2.5):

$$dx = J_a(q) dq \quad (2.6)$$

onde  $J_a(q)$  é a matriz:

$$J_a(q) = \begin{bmatrix} \frac{\partial x_1}{\partial q_1} & \frac{\partial x_1}{\partial q_2} & \cdots & \frac{\partial x_1}{\partial q_n} \\ \frac{\partial x_2}{\partial q_1} & \frac{\partial x_2}{\partial q_2} & \cdots & \frac{\partial x_2}{\partial q_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial x_m}{\partial q_1} & \frac{\partial x_m}{\partial q_2} & \cdots & \frac{\partial x_m}{\partial q_n} \end{bmatrix}. \quad (2.7)$$

A matriz  $J_a(q)$  é chamada de *matriz Jacobiana analítica* e representa o relacionamento diferencial entre os deslocamentos das juntas e a localização do efetuador final. Esta matriz fornece a cinemática direta do manipulador em nível diferencial.

A cinemática inversa diferencial também pode ser obtida através da matriz Jacobiana analítica usando a Equação (2.6). Basta para isto calcular a inversa  $J_a^{-1}(q)$ , ou *pseudo-inversa*<sup>4</sup> [6]  $J_a^\#(q)$  no caso de  $J_a(q)$  não ser quadrada ( $n \neq m$ ):

$$dq = J_a^{-1}(q) dx. \quad (2.8)$$

Observe que como os elementos da matriz Jacobiana analítica (2.7) são dados em função das posições das juntas  $q$ , ela varia conforme a configuração do manipulador.

Configurações nas quais a matriz Jacobiana analítica perde o posto, ou seja, onde  $J_a(q)$  torna-se singular, o manipulador perde um ou mais graus de liberdade e são chamadas *configurações singulares* [2, 5, 67, 68]. Em muitas aplicações é importante planejar os movimentos do manipulador de forma que as configurações singulares sejam evitadas.

Além da relação (2.6), a matriz Jacobiana analítica é utilizada para se estabelecer um mapeamento entre as velocidades das juntas e do efetuador final no espaço de tarefa. Dividindo os dois lados da Equação (2.6) pelo diferencial de tempo  $dt$  é obtido:

$$\frac{dx}{dt} = \frac{J_a(q) dq}{dt}, \quad (2.9)$$

ou seja:

---

<sup>4</sup>A pseudo-inversa de uma matriz  $M$  é o produto matricial:  $M^\# = M^T (MM^T)^{-1}$ .

$$\dot{x} = J_a(q)\dot{q}. \quad (2.10)$$

O vetor de velocidades no espaço de tarefa  $v$  relaciona-se com  $\dot{x}$  (2.10) através de um mapeamento em função da orientação do efetuador final  $\phi$ :

$$v = T_a(\phi)\dot{x}. \quad (2.11)$$

A parte de velocidades lineares de  $v$  é a mesma de  $\dot{x}$ ; já para obter as velocidades angulares é necessário um mapeamento entre a primeira derivada temporal do vetor de orientação  $\phi$  (2.3) e a velocidade angular no espaço de tarefa  $\omega$  [5, 67, 81]:

$$\omega = T(\phi)\dot{\phi} \quad (2.12)$$

onde  $T(\phi)$  é uma matriz de transformação que depende do conjunto particular de parâmetros usados para representar a orientação do efetuador final.

Do ponto de vista de significado físico, o vetor  $\omega$  é superior a  $\dot{\phi}$  para expressar a velocidade. Os três elementos de  $\omega$  são componentes de velocidade angular, ortogonais, sobre os eixos  $x$ ,  $y$  e  $z$  de um sistema de coordenadas cartesiano. Por outro lado,  $\dot{\phi}$  geralmente representa componentes num sistema de coordenadas não ortogonal, cujos eixos variam conforme o valor de  $\phi$ .

Portanto, a matriz  $T_a(\phi)$  da Equação (2.11) tem a forma:

$$T_a(\phi) = \begin{bmatrix} I & 0 \\ 0 & T(\phi) \end{bmatrix}. \quad (2.13)$$

Por fim, para obter as velocidades  $v$  a partir das velocidades das juntas  $\dot{q}$  é utilizada a relação (2.10):

$$v = T_a(\phi)J_a(q)\dot{q}. \quad (2.14)$$

Definindo uma nova matriz, chamada de *matriz Jacobiana geométrica*<sup>5</sup> [5, 67, 81]:

$$J(q) = T_a(\phi)J_a(q), \quad (2.15)$$

a Equação (2.14) pode ser reescrita como:

---

<sup>5</sup>Esta é a matriz Jacobiana básica do manipulador, por isso normalmente chamada apenas de *matriz Jacobiana*. Esta será a nomenclatura utilizada neste trabalho

$$v = \begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = J(q)\dot{q}. \quad (2.16)$$

onde  $\dot{p}$  e  $\omega$  são vetores ( $3 \times 1$ ) de velocidades linear e angular no espaço de tarefa, respectivamente.

Na Equação (2.16) é possível ver que cada coluna  $i$  da matriz Jacobiana  $J(q)$  representa a velocidade no efetuador final gerada por um movimento da junta  $i$  com velocidade unitária e com todas as outras juntas paradas.

## 2.3 Dinâmica

Considere um sistema dinâmico, no caso um robô manipulador, cuja localização completa é dada por um conjunto de coordenadas generalizadas, por exemplo o vetor  $q$  (2.1). Sejam  $K$  e  $U$  as energias totais cinética e potencial, respectivamente, deste sistema dinâmico. O Lagrangiano deste sistema é definido da seguinte forma:

**Definição 2.3.1** [2, 25, 67] *O Lagrangiano  $\mathcal{L}$  de um sistema conservativo é igual à diferença entre a energia cinética  $K$  e a energia potencial  $U$ :*

$$\mathcal{L}(q_i, \dot{q}_i) = K - U. \quad (2.17)$$

Usando esta definição, as equações do movimento do sistema dinâmico são dadas por:

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = Q_i \quad (2.18)$$

onde  $Q_i$  é a força generalizada correspondente à coordenada generalizada  $q_i$ .

### 2.3.1 Energia Cinética

Como a energia é aditiva, a energia cinética total  $K$  armazenada no manipulador é dada por:

$$K = \sum_{i=1}^n K_i \quad (2.19)$$

onde  $K_i$  é a energia cinética do elo  $i$ .

Considere que o elo  $i$  possui uma velocidade linear (do centro de massa) dada pelo vetor  $\dot{p}_i$  ( $3 \times 1$ ) e velocidade angular dada pelo vetor  $\omega_i$  ( $3 \times 1$ ), ambas com relação

ao sistema de coordenadas da base. Estas velocidades podem ser obtidas utilizando a Equação (2.10) e considerando que o elo  $i$  é o efetuador final:

$$\begin{aligned}\dot{p}_i &= J_{L1}^{(i)}\dot{q}_1 + \dots + J_{Li}^{(i)}\dot{q}_i = J_L^{(i)}\dot{q}; \\ \omega_i &= J_{A1}^{(i)}\dot{q}_1 + \dots + J_{Ai}^{(i)}\dot{q}_i = J_A^{(i)}\dot{q}\end{aligned}\quad (2.20)$$

onde  $J_L^{(i)}$  e  $J_A^{(i)}$  são os  $i$ -ésimos vetores linha das matrizes Jacobianas ( $3 \times n$ )  $J_L$  e  $J_A$  para velocidades lineares e angulares, respectivamente.

As matrizes  $J_L$  e  $J_A$  são obtidas particionando a matriz Jacobiana completa (2.15):

$$J(q) = \begin{bmatrix} J_L \\ J_A \end{bmatrix} = \begin{bmatrix} J_{L1} & J_{L2} & \dots & J_{Ln} \\ J_{A1} & J_{A2} & \dots & J_{An} \end{bmatrix}. \quad (2.21)$$

A energia cinética armazenada em cada elo é dada pela soma dos termos de energias cinéticas translacional e rotacional do elo [20, 25]:

$$K_i = \frac{1}{2}\dot{p}_i^T m_i \dot{p}_i + \frac{1}{2}\omega_i^T I_i \omega_i \quad (2.22)$$

onde  $m_i$  e  $I_i$  são a massa e o *tensor de inércia* do centro de massa ( $3 \times 3$ ) expresso no sistema de coordenadas da base, respectivamente.

Substituindo as Equações (2.20) em (2.22) e depois em (2.19) é obtido:

$$K = \frac{1}{2} \sum_{i=1}^n \left( \dot{q}^T J_L^{(i)T} m_i J_L^{(i)} \dot{q} + \dot{q}^T J_A^{(i)T} I_i J_A^{(i)} \dot{q} \right). \quad (2.23)$$

Usando a matriz definida abaixo:

**Definição 2.3.2** *A matriz quadrada simétrica, baseada nos tensores de inércia individuais de cada elo:*

$$H(q) = \sum_{i=1}^n \left( J_L^{(i)T} m_i J_L^{(i)} + J_A^{(i)T} I_i J_A^{(i)} \right) \quad (2.24)$$

*contém todas propriedades de massa do manipulador e é chamada de “matriz tensor de inércia do manipulador”<sup>6</sup>. Seus elementos  $H_{ii}(q)$  representam as inércias efetivas e os elementos  $H_{ij}(q)$ , com  $i \neq j$ , as inércias de acoplamento.*

a Equação (2.23) pode ser reescrita de uma forma mais compacta:

---

<sup>6</sup>Esta matriz é geralmente chamada na literatura simplesmente de *matriz de inércia*, e esta será a nomenclatura utilizada neste trabalho.

$$K = \frac{1}{2} \dot{q}^T H(q) \dot{q}. \quad (2.25)$$

### 2.3.2 Energia Potencial

Num robô manipulador rígido, a energia potencial é devida unicamente à gravidade, enquanto que num robô flexível [42, 54] a energia potencial também inclui a *energia potencial elástica* [4].

Seja  $g$  o vetor ( $3 \times 1$ ) representando a aceleração da gravidade expresso no sistema de coordenadas da base:

$$g = \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} \quad (2.26)$$

onde  $g_x$ ,  $g_y$  e  $g_z$  são as componentes x, y e z da aceleração da gravidade, respectivamente, com relação ao sistema de coordenadas da base.

A energia potencial total armazenada no manipulador, considerado rígido, é então dada por:

$$U = \sum_{i=1}^n m_i g^T r_i \quad (2.27)$$

onde  $r_i$  é a posição do centro de massa do elo  $i$  no sistema de coordenadas da base.

Note que como as posições  $r_i$  são funções das juntas  $q$  (2.5), a energia potencial  $U$  (2.27) pode ser reescrita em função das coordenadas generalizadas.

### 2.3.3 Forças Generalizadas

Abaixo é dada a definição das *forças generalizadas* [2, 25, 67] não conservativas no contexto dos robôs manipuladores:

**Definição 2.3.3** *Forças generalizadas não conservativas são todas as forças e momentos que agem no manipulador com exceção das forças gravitacionais e inerciais.*

As forças generalizadas não conservativas são calculadas a partir dos torques<sup>7</sup>  $u$  de controle das juntas (entrada do sistema) e das forças e torques externos  $F_e$  agindo sobre o manipulador.

---

<sup>7</sup>A ação de controle aplicada nas juntas será chamada neste trabalho simplesmente de torques apesar do fato de que em juntas prismáticas podem ser aplicadas forças.

Será considerado que os atuadores do manipulador exercem torques apenas em juntas individuais, ou seja:

**Suposição 2.3.1** *Um manipulador de  $n$  graus de liberdade possui  $n$  atuadores, um para cada junta.*

Assim, o vetor de torques aplicados no manipulador pode ser escrito como:

$$u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \quad (2.28)$$

onde  $u_i$  é o torque exercido pelo atuador na junta  $i$ .

As forças e torques externos aplicados no efetuador final, no espaço de tarefa expressos no sistema de coordenadas da base, pode ser escrito como o vetor:

$$F_e = \begin{bmatrix} f_x \\ f_y \\ f_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \quad (2.29)$$

onde  $f_{(\cdot)}$  é a componente de força translacional agindo na direção  $(\cdot)$  e  $\tau_{(\cdot)}$  é o torque agindo no eixo  $(\cdot)$ .

Usando o princípio do trabalho virtual [2, 25, 67], o diferencial de trabalho  $\delta W$  pode ser calculado a partir dos torques  $u$  e das forças e torques  $F_e$ , em função de  $\delta q$ :

$$\delta W = (u + J^T(q)F_e)^T \delta q. \quad (2.30)$$

Igualando esta equação com a equação de trabalho virtual dado em função das forças generalizadas  $Q_i$ :

$$\delta W = Q^T \delta q, \quad (2.31)$$

as forças generalizadas não conservativas são obtidas como sendo:

$$Q = u + J^T(q)F_e. \quad (2.32)$$

### 2.3.4 Formulação Lagrangeana

Usando as energias cinética e potencial, e as forças generalizadas calculadas nas sub-seções anteriores, a equação de movimento de Lagrange pode ser obtida.

O primeiro termo da Equação (2.18) é calculado da seguinte forma:

$$\frac{d}{dt} \left( \frac{\partial K}{\partial \dot{q}_i} \right) = \frac{d}{dt} \left( \sum_{j=1}^n H_{ij} \dot{q}_j \right) = \sum_{j=1}^n H_{ij} \ddot{q}_j + \sum_{j=1}^n \frac{dH_{ij}}{dt} \dot{q}_j. \quad (2.33)$$

Como  $H_{ij}$  é função das coordenadas generalizadas  $q$ , a derivada temporal de  $H_{ij}$  é dada por:

$$\frac{dH_{ij}}{dt} = \sum_{k=1}^n \frac{\partial H_{ij}}{\partial q_k} \frac{dq_k}{dt} = \sum_{k=1}^n \frac{\partial H_{ij}}{\partial q_k} \dot{q}_k. \quad (2.34)$$

Parte do segundo termo da Equação (2.18) é dado por:

$$\frac{\partial K}{\partial q_i} = \frac{\partial}{\partial q_i} \left( \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n H_{jk} \dot{q}_j \dot{q}_k \right) = \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n \frac{\partial H_{jk}}{\partial q_i} \dot{q}_j \dot{q}_k \quad (2.35)$$

pois  $H_{jk}$  é função de  $q_i$ .

A outra parte, o termo gravitacional, é obtido calculando a derivada parcial da energia potencial  $U$  com relação a  $q$ :

$$G_i = \frac{\partial U}{\partial q_i} = \sum_{j=1}^n m_j g^T \frac{\partial r_j}{\partial q_i}. \quad (2.36)$$

Como a derivada parcial de  $r_j$  com relação a  $q_i$  é igual ao elemento  $j$  do  $i$ -ésimo vetor coluna da matriz Jacobiana  $J_L$  (matriz Jacobiana ( $3 \times n$ ) para velocidades lineares), o termo gravitacional (2.36) pode ser reescrito como:

$$G_i = \sum_{j=1}^n m_j g^T J_{L_i}^{(j)}. \quad (2.37)$$

Substituindo os elementos calculados acima na Equação (2.18) é obtido:

$$\sum_{j=1}^n H_{ij} \ddot{q}_j + \sum_{j=1}^n \sum_{k=1}^n h_{ijk} \dot{q}_j \dot{q}_k + G_i = Q_i \quad (2.38)$$

onde:

$$h_{ijk} = \frac{\partial H_{ij}}{\partial q_k} - \frac{1}{2} \frac{\partial H_{jk}}{\partial q_i}. \quad (2.39)$$

Substituindo as forças generalizadas (2.32) em (2.38) e reescrevendo esta equação de uma forma especial, é obtida a equação dinâmica de um robô manipulador de  $n$

graus de liberdade:

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = u + J^T(q)F_e \quad (2.40)$$

onde  $H(q)$  é uma matriz ( $n \times n$ ),  $C(q, \dot{q})$  é uma matriz de ordem ( $n \times n$ ) e  $G(q)$  é um vetor ( $n \times 1$ ).  $u$  é o vetor ( $n \times 1$ ) de torques para o controle a ser calculado,  $J(q)$  é a matriz Jacobiana ( $m \times n$ ) e  $F_e$  é o vetor ( $m \times 1$ ) de forças/torques generalizados impostos ao manipulador pelo meio, no espaço de tarefa (expresso no sistema de coordenadas da base).

O primeiro termo desta equação,  $H(q)$ , representa os torques inerciais, incluindo os torques de interação entre os elos. O segundo termo,  $C(q, \dot{q})\dot{q}$ , representa as forças centrífugas e de Coriolis. O último termo,  $G(q)$ , representa os torques devido à forças gravitacionais [2, 5, 43, 67, 81].

### 2.3.5 Representação por Variáveis de Estado

Uma representação da planta por *variáveis de estado* [6, 7, 20] é útil para projeto de controladores, e esta será a representação utilizada no capítulo de síntese do controlador à estrutura variável (Capítulo 3).

Uma escolha natural das variáveis de estados é o conjunto de coordenadas generalizadas das juntas e as suas respectivas velocidades (primeira derivada temporal das posições):

$$z = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \quad (2.41)$$

Isolando os termos de acelerações nas juntas  $\ddot{q}$  da Equação (2.40):

$$\frac{d\dot{q}}{dt} = -H^{-1}(q) [C(q, \dot{q})\dot{q} + G(q)] + H^{-1}(q) [u + J^T(q)F_e] \quad (2.42)$$

é obtida a seguinte representação do sistema por variáveis de estados:

$$\dot{z} = \begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ -H^{-1}(q) [C(q, \dot{q})\dot{q} + G(q)] \end{bmatrix} + \begin{bmatrix} 0 \\ H^{-1}(q) \end{bmatrix} [u + J^T(q)F_e]. \quad (2.43)$$

Note que este sistema é não linear e acoplado [5, 36, 65, 72].

Outras representações por variáveis de estado são possíveis. Mesmo porque através de uma *transformação de estados* [6] outra representação do sistema acima é obtida.

Mas geralmente a representação (2.43) é a utilizada para estudos dos robôs manipuladores no espaço de juntas.

## 2.4 Exemplo: Manipulador 2DOF

Como exemplo ilustrativo, será analisado nesta seção o robô manipulador planar de 2 graus de liberdade (2DOF<sup>8</sup>), apresentado na Figura 2.1.

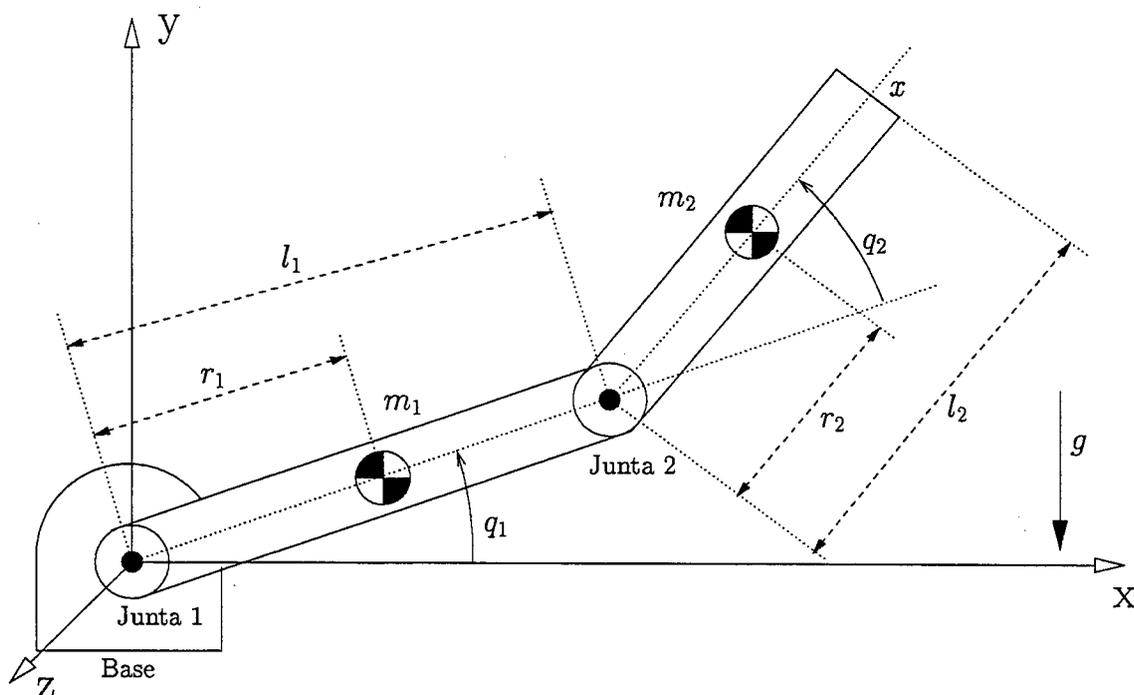


Figura 2.1: Robô Manipulador de Dois Graus de Liberdade

A seguinte notação é utilizada na Figura 2.1:

- $q_i$  - posição angular normal da junta  $i$ , em rad;
- $l_i$  - comprimento do elo  $i$ , em m;
- $r_i$  - distância entre a junta  $i$  e o centro de massa do elo  $i$ , em m;
- $m_i$  - massa do elo  $i$ , em Kg;
- $I_i$  - inércia do elo  $i$  com relação ao eixo que passa pelo centro de massa paralelamente ao eixo  $z$ , em  $\text{Kg m}^2$ .

<sup>8</sup>Este termo é bastante utilizado na literatura em inglês e significa: *2 Degrees Of Freedom*.

Conforme apresentado na Figura 2.1, o sistema de coordenadas cartesiano de referência tem a sua origem na junta 1 e o plano  $xy$  está paralelo com o manipulador.

### 2.4.1 Cinemática Direta

A cinemática direta é facilmente obtida usando as projeções dos elos nos eixos  $x$  e  $y$  através das funções trigonométricas seno e co-seno.

$$x = \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) \\ l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) \end{bmatrix}. \quad (2.44)$$

Usando a definição (2.7) a matriz Jacobiana (2.15) é obtida, calculando seus elementos através das derivadas parciais das Equações (2.44):

$$J(q) = \begin{bmatrix} -l_1 \sin(q_1) - l_2 \sin(q_1 + q_2) & -l_2 \sin(q_1 + q_2) \\ l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) & l_2 \cos(q_1 + q_2) \end{bmatrix}. \quad (2.45)$$

### 2.4.2 Equação Dinâmica

Nesta sub-seção é realizado o cálculo das matrizes da equação dinâmica (2.40) para um manipulador de dois graus de liberdade.

Para calcular a matriz de inércia  $H(q)$  é necessário obter as matrizes Jacobianas  $J_L$  e  $J_A$  (2.21). Elas são obtidas calculando as velocidades lineares e angulares dos elos com relação ao seu centro de massa.

As velocidades lineares dos elos são:

$$\dot{p}_1 = \begin{bmatrix} -r_1 \sin(q_1) & 0 \\ r_1 \cos(q_1) & 0 \end{bmatrix} \dot{q} \quad (2.46)$$

e

$$\dot{p}_2 = \begin{bmatrix} -l_1 \sin(q_1) - r_2 \sin(q_1 + q_2) & -r_2 \sin(q_1 + q_2) \\ l_1 \cos(q_1) + r_2 \cos(q_1 + q_2) & r_2 \cos(q_1 + q_2) \end{bmatrix} \dot{q}. \quad (2.47)$$

As velocidades angulares dos elos são:

$$\omega_1 = \dot{q}_1 = \begin{bmatrix} 1 & 0 \end{bmatrix} \dot{q} \quad (2.48)$$

e

$$\omega_2 = \dot{q}_1 + \dot{q}_2 = \begin{bmatrix} 1 & 1 \end{bmatrix} \dot{q}. \quad (2.49)$$

Substituindo estas velocidades na Equação (2.24), a matriz de inércia do manipulador  $H(q)$  é obtida:

$$\begin{bmatrix} m_1 r_1^2 + I_1 + m_2(l_1^2 + r_2^2 + 2l_1 r_2 \cos(q_2)) + I_2 & m_2 l_1 r_2 \cos(q_2) + m_2 r_2^2 + I_2 \\ m_2 l_1 r_2 \cos(q_2) + m_2 r_2^2 + I_2 & m_2 r_2^2 + I_2 \end{bmatrix}. \quad (2.50)$$

A matriz de forças centrífugas e de Coriolis  $C(q)$  é obtida substituindo os termos acima, elementos da matriz  $H(q)$ , na Equação (2.39):

$$C(q) = \begin{bmatrix} -m_2 l_1 r_2 \sin(q_2) \dot{q}_2 & -m_2 l_1 r_2 \sin(q_2) (\dot{q}_1 + \dot{q}_2) \\ m_2 l_1 r_2 \sin(q_2) \dot{q}_1 & 0 \end{bmatrix}. \quad (2.51)$$

O vetor de forças gravitacionais  $G(q)$  é calculado utilizando a Equação (2.37):

$$G(q) = \begin{bmatrix} m_1 r_1 \cos(q_1) + m_2 (r_2 \cos(q_1 + q_2) + l_1 \cos(q_1)) \\ m_2 r_2 \cos(q_1 + q_2) \end{bmatrix} g \quad (2.52)$$

onde  $g$  é o valor escalar da aceleração da gravidade.

E por fim, as forças generalizadas não conservativas  $Q$  são os torques aplicados nas juntas e as forças/torques externos aplicados no efetuador final, estes últimos transformados para o espaço de juntas:

$$Q = u + J^T(q) F_e = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + J^T(q) \begin{bmatrix} f_x \\ f_y \end{bmatrix}. \quad (2.53)$$

### 2.4.3 Robô Manipulador SCARA

O robô manipulador utilizado para a implementação dos controladores propostos neste trabalho é o SCARA (*Selective Compliance Assembly Robot Arm*) [13, 43], apresentado na Figura 2.2.

Este tipo de manipulador foi inicialmente projetado em 1979 no Japão. Ele é utilizado em aplicações que requerem movimentos rápidos e suaves.

O robô SCARA é extremamente útil em tarefas de montagens onde peças devem ser encaixadas, ou seja, onde há inserção de peças em orifícios como grampos e parafusos, por exemplo. Devido a sua construção física, ele é extremamente rígido na direção vertical e possui uma certa flexibilidade lateral, o que facilita o processo de inserção de peças.

Serão utilizados apenas os dois primeiros elos do manipulador. Os outros dois serão mantidos travados (freios acionados) e motores sem alimentação.

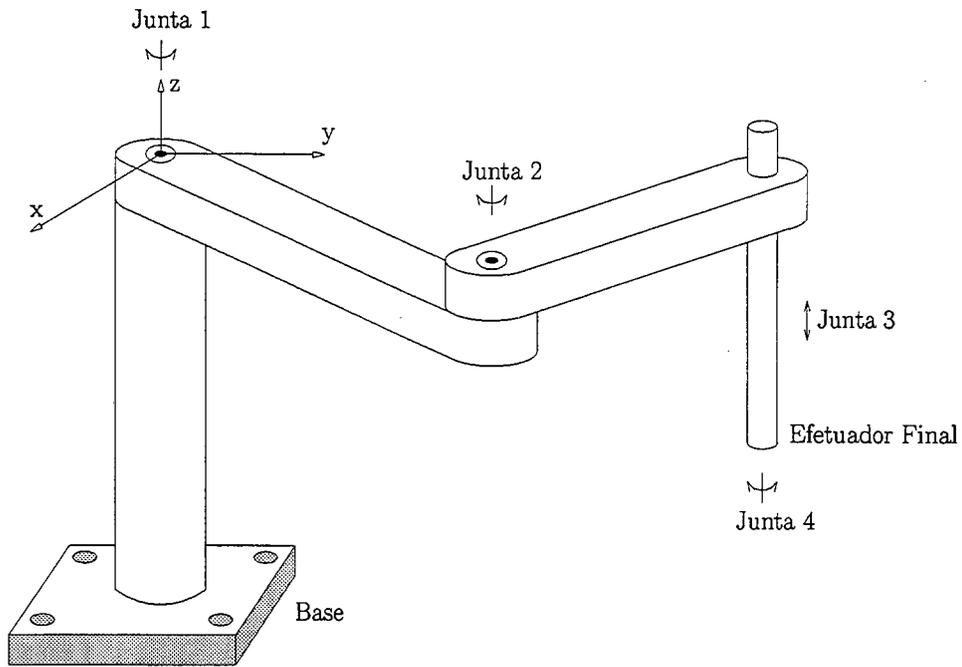


Figura 2.2: Manipulador SCARA (*Selective Compliance Assembly Robot Arm*)

Utilizando o sistema de coordenadas apresentado na Figura 2.2, as modelagens cinemática e dinâmica dos dois primeiros elos são as mesmas das Sub-seções 2.4.1 e 2.4.2, respectivamente. Apenas duas considerações para o modelo dinâmico devem ser feitas.

Primeiramente, os elos 3 e 4 são tratados como sendo uma carga fixa ao efetuador final do manipulador (considerado 2DOF) [40], de massa  $m_L$  e inércia com relação ao eixo que passa pelo *centro de massa* paralelamente ao eixo  $z$   $I_L$ .

Para modelar a carga do efetuador final, as matrizes  $H(q)$ ,  $C(q)$  e  $G(q)$  são alteradas de forma a contabilizar as forças devido a esta carga. Esta alteração consiste em somar as seguintes matrizes às matrizes originais  $H(q)$  (2.50),  $C(q)$  (2.51) e  $G(q)$  (2.52), respectivamente:

$$H_L(q) = \begin{bmatrix} I_L + m_L(l_1^2 + l_2^2 + 2l_1l_2 \cos(q_2)) & I_L + m_Ll_2^2 \\ I_L + m_Ll_2^2 & I_L + m_Ll_2^2 \end{bmatrix}, \quad (2.54)$$

$$C_L(q) = \begin{bmatrix} -m_Ll_1l_2 \sin(q_2)\dot{q}_2 & -m_Ll_1l_2 \sin(q_2)(\dot{q}_1 + \dot{q}_2) \\ m_Ll_1l_2 \sin(q_2)\dot{q}_1 & 0 \end{bmatrix} \quad (2.55)$$

e

$$G_L(q) = \begin{bmatrix} m_L(l_2 \cos(q_1 + q_2) + l_1 \cos(q_1)) \\ m_L l_2 \cos(q_1 + q_2) \end{bmatrix} g. \quad (2.56)$$

A segunda consideração a ser feita é a de que as forças gravitacionais não atuam nos elos 1 e 2 do manipulador SCARA. A força da gravidade está agindo na direção do eixo  $z$ , de forma que ela não exerce nenhuma força nas direções onde o manipulador pode se mover. Assim, o vetor de forças gravitacionais é identicamente igual a zero,  $G(q) = 0$ , para todas as configurações do manipulador.

Os parâmetros físicos de um robô manipulador SCARA industrial real, que é utilizado neste trabalho, são apresentados no Apêndice A.

## 2.5 Conclusões

Neste capítulo foram apresentadas a construção e as definições básicas de um robô manipulador, usando conceitos de dinâmica e estática.

A modelagem cinemática, que descreve as características geométricas, e a modelagem dinâmica, que descreve as características de movimento de um robô manipulador rígido genérico foram obtidas. Ao final do capítulo uma demonstração destas modelagens foi feita utilizando como exemplo um robô manipulador de dois graus de liberdade, que é o modelo utilizado nos capítulos posteriores.

# Capítulo 3

## Controle à Estrutura Variável

Neste capítulo é apresentado o *controlador à estrutura variável* com modos deslizantes. Este controlador é adequado para o controle de plantas não lineares, como é exemplificado através da sua aplicação em um manipulador.

Este tipo de controlador utiliza uma lei de controle não linear, que realiza chaveamentos em alta velocidade sobre uma superfície especificada no espaço de estados que envolva o ponto de equilíbrio do sistema.

Após a apresentação da metodologia de projeto do controlador, é simulado numericamente e implementado em bancada um controlador de posição no espaço de juntas para um robô manipulador industrial SCARA.

### 3.1 Introdução

A *teoria de estrutura variável* foi inicialmente proposta por autores soviéticos na década de 60, utilizando a base matemática do trabalho de Filippov [24]. Ela tem sido estudada por vários autores tanto em *tempo contínuo* [1, 3, 14, 21, 26, 41, 45, 64, 69] como em *tempo discreto* [8, 27, 31, 33, 35, 63] com o intuito de controlar, de forma robusta, diversos tipos de sistemas não lineares, entre eles o objeto deste trabalho: robôs manipuladores.

O controlador à estrutura variável proporciona uma forma efetiva e robusta de controlar plantas não lineares e tem suas raízes na teoria de controle do tipo bang-bang e a relés [14]. Ele pode ser projetado de forma a proporcionar boa rejeição a perturbações, bom rastreamento de trajetória, resposta rápida, e boa estabilidade [33].

O termo controle à estrutura variável surgiu porque a estrutura do controlador é intencionalmente mudada, de acordo com alguma regra pré-estabelecida, visando obter o comportamento da planta desejado (resposta desejada). Devido a esta mudança de

estrutura este controlador é do tipo não linear.

A regra que rege a alteração da estrutura do controlador é dada por funções das variáveis de estado do sistema criadas pelo projetista. Quando iguais a zero, estas funções definem uma superfície no *espaço de estados* [7, 20] chamada de *superfície de deslizamento* (alguns autores preferem chamá-la de *superfície de chaveamento*) [14, 64, 69], que governam o chaveamento do controlador.

A lei de controle visa manter o *estado do sistema* [7, 20] movendo-se nesta superfície até atingir o *ponto de equilíbrio* [6, 7], mesmo na presença de incertezas paramétricas e perturbações. Esta propriedade do estado do sistema permanecer “deslizando” nesta superfície é chamada de *modo deslizante* [14, 36, 69].

Em alguns casos é utilizado um *observador de estados* [20, 80] para se obter todas as variáveis de estado do sistema necessárias à implementação. Neste caso, uma dinâmica extra é incluída no sistema para se obter os estados não disponíveis.

Esta estratégia de controle apresenta duas grandes vantagens para aplicações em robótica:

- não é necessário o conhecimento exato do modelo da planta;
- a performance do sistema é insensível a perturbações externas limitadas que satisfazem certas condições. ✓

Estas propriedades são importantes devido à dificuldade do cálculo exato da dinâmica do manipulador e para rejeição de efeitos devido aos atritos de Coulomb e viscoso. Além disso, as forças devido à cargas acopladas ao efetuador final do manipulador, que nem sempre são conhecidas, podem ser consideradas como perturbações externas durante o projeto do controlador.

Por outro lado, na prática, o controle à estrutura variável apresenta o fenômeno conhecido como *chattering*, que é indesejável em sinais de controle. Uma solução para este problema é apresentada na Seção 3.3.

## 3.2 Projeto

Um sistema não linear de ordem  $n$ , considerando  $m$  entradas do sinal de controle, pode ser representado por um conjunto de equações diferenciais não lineares na seguinte forma:

$$\dot{x} = f(x, u, t) \quad (3.1)$$

onde  $x(t)$  é o vetor ( $n \times 1$ ) de variáveis de estados do sistema,  $u(t)$  é o vetor ( $m \times 1$ ) de entrada de controle do sistema e  $f(x, u, t)$  é um vetor ( $n \times 1$ ) de funções do vetor de estados e da entrada de controle, onde cada elemento é contínuo com primeira derivada com relação a  $x$  contínua e limitada.

A solução  $x(t)$  das Equações (3.1) corresponde a uma curva no espaço de estados quando o tempo  $t$  varia de zero a infinito. Esta curva é geralmente referida como sendo a *trajetória do estado* (alguns autores preferem chamá-la de *trajetória do sistema*) [6].

Para simplificar o problema de determinar o modo deslizante será considerada a seguinte suposição [70]:

**Suposição 3.2.1** *A entrada de controle aparece de forma linear na equação de estados.*

Portanto, no lugar do sistema geral (3.1) o sistema utilizado para análise neste capítulo é:

$$\dot{x} = f(x, t) + B(x, t)u(x, t) \quad (3.2)$$

onde  $B(x, t)$  é uma matriz ( $n \times m$ ) de elementos contínuos com a primeira derivada com relação a  $x$  contínua e limitada.

O projeto do controlador à estrutura variável é feito basicamente em duas fases: especificação da superfície de deslizamento e projeto da lei de controle chaveada, apresentadas nas sub-seções seguintes.

### 3.2.1 Primeira Fase: Superfície de Deslizamento

A primeira fase do projeto do controlador à estrutura variável é a escolha de um conjunto de  $m$  funções do vetor de estados do sistema:

$$\sigma_i(x, t) ; \quad i = 1, \dots, m. \quad (3.3)$$

Igualando este conjunto de funções a zero:

$$\sigma = \begin{bmatrix} \sigma_1(x, t) \\ \sigma_2(x, t) \\ \vdots \\ \sigma_m(x, t) \end{bmatrix} = 0, \quad (3.4)$$

a superfície de deslizamento é obtida. As funções (3.3) são projetadas de forma que o sistema original restrito à intercessão destas superfícies apresente o comportamento

dinâmico desejado. A Figura 3.1 apresenta um exemplo de superfície de deslizamento para duas funções do vetor de estados.

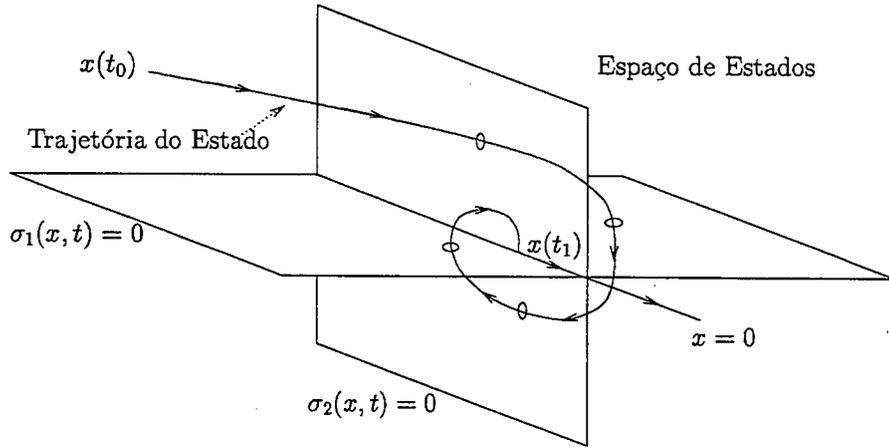


Figura 3.1: Superfície de Deslizamento

Apesar da possibilidade de se utilizar superfícies de deslizamento não lineares ( $\sigma_i(x, t)$  em (3.3) não lineares), as lineares são preferidas na literatura [3, 31, 70]. As superfícies não lineares não são projetadas de forma trivial. Já as lineares podem ser projetadas tratando o projeto como se fosse um problema de realimentação de estados [14].

### Forma Regular

Uma forma mais conveniente para o projeto da superfície de deslizamento é colocar a equação dinâmica do sistema (3.2) na forma regular apresentada abaixo:

$$\begin{aligned} \dot{z}_1 &= \bar{f}_1(z, t) \\ \dot{z}_2 &= \bar{f}_2(z, t) + \bar{B}_2(z, t)u(z, t) \end{aligned} \quad (3.5)$$

onde  $z_1$  e  $z_2$  são de dimensões  $((n - m) \times 1)$  e  $(m \times 1)$ , respectivamente.

Esta forma pode ser obtida através de uma transformação de estados linear:

$$z(t) = Tx(t) \quad (3.6)$$

onde  $T$  tem a propriedade:

$$TB(x, t) = TB(T^{-1}z, t) = \begin{bmatrix} 0 \\ \bar{B}_2(z, t) \end{bmatrix} \quad (3.7)$$

onde  $\bar{B}_2(z, t)$  é um mapeamento não singular ( $m \times m$ ) para todo  $z$  e  $t$ .

Em geral não existe uma transformação linear que satisfaça (3.7). Portanto, normalmente o cálculo da forma regular requer a seguinte transformação não linear [14]:

$$z(t) = T(x, t) = \begin{bmatrix} T_1(x, t) \\ T_2(x, t) \end{bmatrix} \quad (3.8)$$

onde:

- $T(x, t)$  é uma *transformação difeomórfica*, ou seja, um mapeamento inverso diferenciável contínuo  $\tilde{T}(z, t) = x$  existe, satisfazendo  $\tilde{T}(0, t) = 0$  para todo  $t$ ;
- $T_1(\cdot, \cdot) : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^{n-m}$  e  $T_2(\cdot, \cdot) : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^m$ ;
- $T(x, t)$  tem a propriedade:

$$\frac{\partial T}{\partial x} B(x, t) = \begin{bmatrix} \frac{\partial T_1}{\partial x} \\ \frac{\partial T_2}{\partial x} \end{bmatrix} B(\tilde{T}(z, t), t) = \begin{bmatrix} 0 \\ \bar{B}_2(z, t) \end{bmatrix}.$$

No caso de sistemas lineares uma transformação deste tipo sempre existe. Já no caso de sistemas não lineares, a existência desta transformação não é garantida [14].

Usando a transformação (3.8) é obtida a seguinte representação do sistema não linear (3.2) na forma regular:

$$\begin{aligned} \dot{z}_1 &= \frac{\partial T_1}{\partial x} f(\tilde{T}(z, t), t) + \frac{\partial T_1}{\partial t} \equiv \bar{f}_1(z, t) \\ \dot{z}_2 &= \frac{\partial T_2}{\partial x} f(\tilde{T}(z, t), t) + \frac{\partial T_2}{\partial t} + \frac{\partial T_2}{\partial x} B(\tilde{T}(z, t), t)u(z, t) \equiv \bar{f}_2(z, t) + \bar{B}_2(z, t)u(z, t). \end{aligned} \quad (3.9)$$

Em alguns casos todas as não linearidades do modelo da planta podem ser movidos para  $\bar{f}_2(z, t)$ , de forma que  $\bar{f}_1(z, t)$  pode ser representado linearmente:

$$\dot{z}_1 = \bar{f}_1(z, t) = \begin{bmatrix} A_{11} & A_{12} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \quad (3.10)$$

onde  $A_{11}$  e  $A_{12}$  são matrizes lineares.

Assim, podem ser utilizadas técnicas lineares para resolver o problema de projeto da superfície de deslizamento, como será visto adiante. Ele pode ser tratado, por exemplo,

como se fosse um problema de realimentação de estados [20].

### Escolha dos Ganhos da Superfície

Supondo que o sistema se encontre na forma regular (3.9), a superfície de deslizamento (3.4) pode ter a forma:

$$\sigma(z, t) = Kz + r(t) = \begin{bmatrix} K_1 & K_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + r(t) = 0 \quad (3.11)$$

onde  $K$  é uma matriz de ganhos escolhida de forma que  $K_2$  seja não singular, e  $r(t)$  é um sinal de referência a ser seguido.

A superfície de deslizamento (3.4) impõe  $m$  restrições à dinâmica da planta durante o modo deslizante. Portanto,  $m$  variáveis de estado podem ser eliminadas do modelo do sistema. Assim, o sub-conjunto de variáveis de estado  $z_2$  pode ser calculado a partir do outro sub-conjunto  $z_1$ :

$$z_2 = -K_2^{-1} (K_1 z_1(t) + r(t)). \quad (3.12)$$

Substituindo a equação acima na Equação (3.9), é obtido um sistema equivalente de ordem reduzida, cuja dinâmica representa o comportamento do sistema, a trajetória do estado, em modo deslizante:

$$\dot{z}_1 = \bar{f}_1(z_1, z_2, t) = \bar{f}_1(z_1, -K_2^{-1} (K_1 z_1(t) + r(t)), t). \quad (3.13)$$

A dinâmica deste sistema não linear pode ser alterada para se obter o comportamento desejado durante o deslizamento, através de uma seleção conveniente dos ganhos da matriz  $K$ .

### 3.2.2 Segunda Fase: Lei de Controle

A segunda fase do projeto do controlador consiste na determinação da lei de controle chaveada que faz o estado do sistema ser atraído para a superfície de deslizamento  $\sigma(x, t) = 0$ , definida na sub-seção anterior, e, uma vez atingida esta superfície, lá o mantenha até que o ponto de equilíbrio do sistema seja alcançado. Para que isto ocorra, o sinal de controle deve ser projetado de forma que os vetores tangentes à trajetória do estado apontem para a superfície [14, 69].

Uma lei de controle que atende a requisição acima é a lei chaveada [14, 69, 71]:

$$u_i = -u_{0i} \operatorname{sgn}(\sigma_i(x, t)) = \begin{cases} -u_{0i}, & \text{se } \sigma_i(x, t) > 0 \\ 0, & \text{se } \sigma_i(x, t) = 0 \\ +u_{0i}, & \text{se } \sigma_i(x, t) < 0 \end{cases} \quad (3.14)$$

onde  $u_{0i}$  é o módulo do sinal de controle máximo para a entrada  $i$ .

Para entender o funcionamento deste controlador considere o sistema de primeira ordem apresentado abaixo:

$$\dot{x} = u(x, t), \quad (3.15)$$

com a seguinte superfície de deslizamento:

$$\sigma(x, t) = x = 0. \quad (3.16)$$

Se o estado do sistema estiver acima da superfície de deslizamento ( $\sigma(x, t) > 0$ ), um sinal de controle negativo  $-u_0$  o levará até ela ( $\sigma(x, t) = 0$ ). Se o estado ultrapassar a superfície ( $\sigma(x, t) < 0$ ), o sinal de controle é chaveado para um valor positivo  $+u_0$  que o levará novamente de volta a ela. Desta forma, o controle faz com que os vetores tangentes à trajetória do sistema apontem para a superfície de deslizamento, levando e mantendo o estado sobre a mesma, conforme apresentado na Figura 3.2.

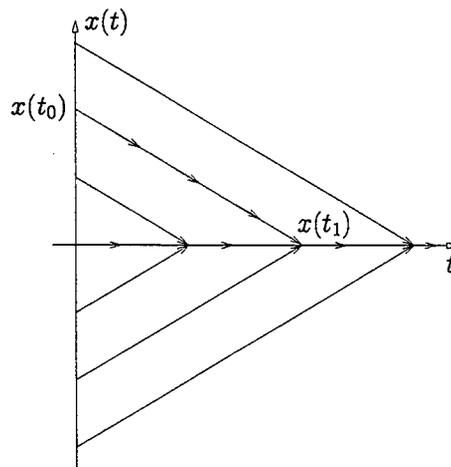


Figura 3.2: Trajetória do Estado - Sistema de Primeira Ordem

Observe que independentemente dos valores das variáveis de estado, bem como dos ganhos  $K$  da Equação (3.11), a ação de controle  $u(x, t)$  é limitada em  $\pm u_0$ .

### Controle Equivalente

Utkin [69, 70, 71] propôs o *método do controle equivalente* para obter o comportamento do sistema quando o mesmo estiver em modo deslizante. Este método consiste em excitar o sistema com um *controle equivalente*  $u_{eq}(x, t)$ , de forma que a trajetória do estado do sistema siga a superfície de deslizamento sempre que o estado inicial  $x(t_0)$  estiver sobre a superfície de deslizamento, ou uma trajetória paralela à superfície quando o estado inicial estiver fora dela. Note que, este último caso nunca ocorre, uma vez que o controle chaveado (3.14) garante que o estado do sistema alcance a superfície de deslizamento a partir de algum instante de tempo finito.

Considerando a existência do modo deslizante a partir de um tempo  $t_1$  implica que, para todo tempo  $t \geq t_1$ ,  $\sigma(x, t) = 0$ , e portanto  $\dot{\sigma}(x, t) = 0$ . Usando este fato, pode ser obtido o controle equivalente  $u_{eq}(x, t)$  derivando a função  $\sigma(x, t)$  em relação ao tempo (utilizando a *regra da cadeia*) e utilizando a Equação (3.2):

$$\dot{\sigma}(x, t) = \frac{\partial \sigma}{\partial t} + \frac{\partial \sigma}{\partial x} \dot{x} = \frac{\partial \sigma}{\partial t} + \frac{\partial \sigma}{\partial x} f(x, t) + \frac{\partial \sigma}{\partial x} B(x, t) u(x, t) = 0. \quad (3.17)$$

Usando a suposição:

**Suposição 3.2.2** O produto matricial  $\frac{\partial \sigma}{\partial x} B(x, t)$  é não singular para todo  $x$  e  $t$ .

o controle  $u(x, t)$ , da Equação (3.17), pode ser isolado obtendo então o controle equivalente  $u_{eq}(x, t)$ :

$$u_{eq}(x, t) = - \left[ \frac{\partial \sigma}{\partial x} B(x, t) \right]^{-1} \left( \frac{\partial \sigma}{\partial t} + \frac{\partial \sigma}{\partial x} f(x, t) \right). \quad (3.18)$$

Aplicando este controle  $u_{eq}(x, t)$  no sistema (3.2) tem-se  $\sigma(x(t_1), t_1) = 0$  e  $\dot{\sigma}(x(t_1), t_1) = 0$ ; então, para todo  $t \geq t_1$  a dinâmica do sistema estará restrita à superfície de deslizamento. Assim, é obtida a *equação dinâmica do sistema equivalente* [14, 70] sobre a superfície de deslizamento:

$$\dot{x}(t) = \left[ I - B(x, t) \left[ \frac{\partial \sigma}{\partial x} B(x, t) \right]^{-1} \frac{\partial \sigma}{\partial x} \right] f(x, t) - B(x, t) \left[ \frac{\partial \sigma}{\partial x} B(x, t) \right]^{-1} \frac{\partial \sigma}{\partial t}. \quad (3.19)$$

### Ganhos do Controlador - Estabilidade

Para encontrar os ganhos do controlador deve ser verificada a estabilidade para a superfície de deslizamento. Isto pode ser feito baseado no teorema abaixo, utilizando uma *função de Lyapunov* [36, 65, 72] adequada.

**Teorema 3.2.1** *Se para um sistema dinâmico  $\dot{x} = f(x, u, t)$  pode ser proposta uma função escalar do vetor de estados  $V(x)$ , cuja a primeira derivada temporal seja contínua, e que obedeça às seguintes condições para qualquer valor de  $x$ :*

- $V(x)$  é definida positiva;
- $\dot{V}(x)$  é definida negativa;
- $V(x) \rightarrow \infty$  quando  $\|x\| \rightarrow \infty$ .

então a origem do sistema é um ponto de equilíbrio globalmente assintoticamente estável.

A função de Lyapunov aqui utilizada é obtida através de uma matriz definida positiva cuja definição é dada abaixo:

**Definição 3.2.1** *Uma matriz quadrada  $M$ ,  $M = [M_{ij}]$ , é definida positiva se para qualquer vetor  $v \neq 0$ :*

$$v^T M v > 0$$

Em outras palavras, uma matriz  $M$  é definida positiva se a função quadrática  $v^T M v$  é uma função definida positiva. O teorema de Sylvester [6] mostra que, assumindo que  $M$  é simétrica, uma condição necessária e suficiente para que  $M$  seja definida positiva é que os seus menores principais:

$$M_{11}, (M_{11}M_{22} - M_{21}M_{12}), \dots, \text{ e } \det(M)$$

sejam todos estritamente positivos.

Considere a função de Lyapunov na forma quadrática:

$$V(x, \sigma, t) = \frac{1}{2} \sigma^T(x, t) M \sigma(x, t) \quad (3.20)$$

onde  $M$  é uma matriz definida positiva simétrica.

Calculando a derivada temporal da função (3.20) é obtido:

$$\dot{V}(x, \sigma, t) = \frac{1}{2} \dot{\sigma}^T(x, t) M \sigma(x, t) + \frac{1}{2} \sigma^T(x, t) M \dot{\sigma}(x, t) = \sigma^T(x, t) M \dot{\sigma}(x, t). \quad (3.21)$$

Usando a derivada temporal de  $\sigma(x, t)$  apresentada em (3.17) é obtido:

$$\dot{V}(x, \sigma, t) = \sigma^T(x, t)M \frac{\partial \sigma}{\partial t} + \sigma^T(x, t)M \frac{\partial \sigma}{\partial x} f(x, t) + \sigma^T(x, t)M \frac{\partial \sigma}{\partial x} B(x, t)u(x, t). \quad (3.22)$$

Utilizando uma lei de controle que englobe o controle equivalente  $u_{eq}(x, t)$  (3.18):

$$u(x, t) = u_{eq}(x, t) + u_n(x, t) \quad (3.23)$$

a Equação (3.22) se torna:

$$\dot{V}(x, \sigma, t) = \sigma^T(x, t)M \frac{\partial \sigma}{\partial x} B(x, t)u_n(x, t). \quad (3.24)$$

Para facilidade de análise da função acima, lembrando que o produto matricial  $\frac{\partial \sigma}{\partial x} B(x, t)$  é considerado não singular para todo  $x$  e  $t$  (Suposição 3.2.2), é conveniente definir o controle  $u_n(x, t)$  como sendo:

$$u_n(x, t) = \left[ \frac{\partial \sigma}{\partial x} B(x, t) \right]^{-1} \bar{u}_n(x, t). \quad (3.25)$$

Usando o controle  $u_n(x, t)$  acima e substituindo a matriz  $M$  por uma matriz identidade de dimensão  $m$  (observe que não há perda de generalidade e que obviamente  $I_m > 0$ ), a Equação (3.24) pode ser reescrita como:

$$\dot{V}(x, \sigma, t) = \sigma^T(x, t)\bar{u}_n(x, t). \quad (3.26)$$

Portanto, pelo Teorema 3.2.1, para garantir a estabilidade basta escolher o sinal de controle  $\bar{u}_n(x, t)$ , de forma que a função (3.26) seja negativa. Se  $\bar{u}_n(x, t)$  for escolhido em função do sinal de  $\sigma(x, t)$  é sempre possível satisfazer esta restrição. A forma mais simples de se fazer isto é escolher  $\bar{u}_n$  como sendo uma função do tipo:

$$\bar{u}_n = -K \text{sgn}(\sigma(x, t)) \quad (3.27)$$

onde  $K$  é uma matriz diagonal de ordem  $m$  de ganhos positivos e

$$\text{sgn}(\sigma(x, t)) = \begin{bmatrix} \text{sgn}(\sigma_1(x, t)) \\ \text{sgn}(\sigma_2(x, t)) \\ \vdots \\ \text{sgn}(\sigma_m(x, t)) \end{bmatrix}$$

Desta forma, a primeira derivada temporal da função de Lyapunov é:

$$\dot{V}(x, \sigma, t) = -\sigma^T(x, t)K \operatorname{sgn}(\sigma(x, t)) = -K\|\sigma(x, t)\| < 0. \quad (3.28)$$

Assim, utilizando o controlador à estrutura variável a estabilidade assintótica é garantida. Além disso, a superfície de deslizamento é atingida num tempo finito.

Note que a análise aqui apresentada não considerou variações paramétricas da planta nem perturbações externas. Quando estas são consideradas, para garantir a estabilidade não é suficiente que os ganhos  $K$  da Equação (3.28) sejam estritamente positivos. Neste caso, é necessário que os ganhos  $K$  sejam “suficientemente altos”, como é apresentado na Seção 3.4.

### Região de Atração

Além da estabilidade, é uma questão de interesse determinar o quão longe da superfície de deslizamento a trajetória do sistema pode ainda convergir para ela quando o tempo  $t$  tende a infinito. Disto surge a definição de *região de atração*, também chamada de *região de estabilidade assintótica* ou *domínio de atração* [65, 72].

A região de atração, no contexto dos modos deslizantes, é o maior sub-conjunto do espaço de estados a partir do qual o deslizamento é alcançado [14, 69]. Um modo deslizante é *globalmente alcançável* se o domínio de atração é todo o espaço de estados do sistema.

Encontrar uma região de atração exata, analiticamente, pode ser uma tarefa difícil ou até mesmo impossível [36]. Entretanto, o segundo método de Lyapunov [36, 65] pode ser usado para estimar a região de atração, ou seja, encontrar o maior sub-conjunto do espaço de estados que contenha a região de atração. Neste sub-conjunto a primeira derivada temporal de uma função generalizada de Lyapunov  $V(x, \sigma, t)$  é negativa.

Portanto, no caso aqui apresentado, considerando a derivada temporal da função de Lyapunov (3.28) o modo deslizante é globalmente alcançável. Desta forma, qualquer que sejam as condições iniciais a superfície de deslizamento é sempre atingida pelo estado do sistema.

### 3.3 Chattering

Um *modo deslizante ideal* [69] existirá apenas quando a trajetória do estado estiver confinada à superfície de deslizamento, ou seja, quando  $\sigma(x(t), t)$  for identicamente igual a zero para todo instante  $t \geq t_1$  para algum  $t_1$ . Entretanto, em casos reais, as limitações físicas dos atuadores, como por exemplo *atraso*, *zona morta* e *histerese*

[20, 34, 41, 43, 80], impossibilitam que um chaveamento extremamente rápido seja feito, de forma que um modo deslizante ideal não é alcançado.

Devido a esta limitação, surge um ruído de alta frequência na ação de controle quando o estado do sistema encontra-se próximo da superfície de deslizamento. Este fenômeno é conhecido como *chattering*<sup>1</sup> [1, 14, 21, 31, 69].

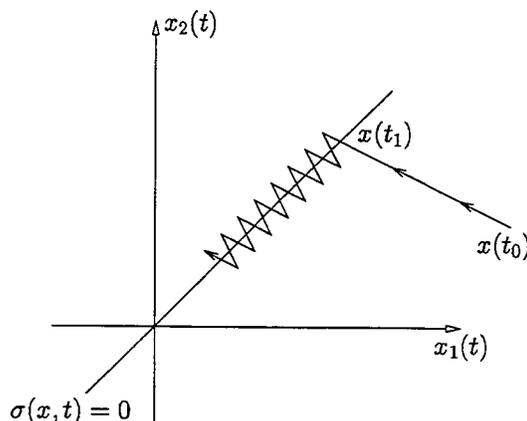


Figura 3.3: *Chattering*

O *chattering* faz com que o estado do sistema oscile em torno da superfície de deslizamento, conforme apresentado na Figura 3.3, e apresenta dois inconvenientes em sistemas de controle:

- as dinâmicas de alta frequência não modeladas podem ser excitadas, resultando em instabilidades não previstas;
- a vida útil dos atuadores é reduzida ou eles podem até mesmo ser danificados.

Para solucionar este problema, pode ser utilizada uma lei de controle contínua que aproxime o funcionamento da lei descontínua numa região próxima da superfície de chaveamento, obtendo assim um rastreamento de referência com uma exatidão controlada. O nível de *chattering* no sinal de controle e a exatidão do seguimento de referência se relacionam de forma direta, existindo portanto, um compromisso entre eles.

Utilizando o conceito de *camada limite* [14, 65, 69], é considerando que o deslizamento ocorre ao longo de uma vizinhança, de tamanho igual a  $2\epsilon$ , da superfície de deslizamento definida abaixo:

<sup>1</sup>Neste trabalho é utilizado o termo *chattering* em inglês por falta de uma tradução consagrada na literatura em português. Esta palavra é utilizada para se referir ao ruído gerado por um elemento chaveador.

$$\{ x \mid \|\sigma(x, t)\| \leq \epsilon \} \quad (3.29)$$

onde  $\epsilon > 0$ .

A Figura 3.4 mostra o comportamento do estado do sistema quando é utilizada uma camada limite.

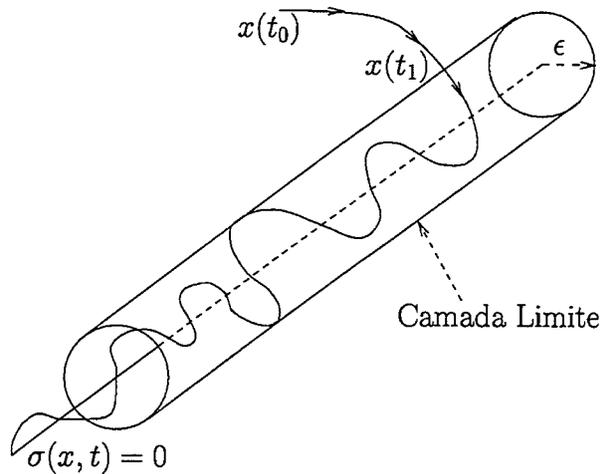


Figura 3.4: Camada Limite

Para implementar a camada limite a função  $\text{sgn}(\cdot)$  da Equação (3.14) é alterada para uma função do tipo:

$$\text{sat}(n/\epsilon) = \begin{cases} \text{sgn}(n), & \text{se } |n| \geq \epsilon \\ \frac{n}{\epsilon}, & \text{se } |n| < \epsilon \end{cases} \quad (3.30)$$

Desta forma uma comutação infinitamente rápida (3.14) é substituída por uma comutação mais suave (3.30) apresentada na Figura 3.5.

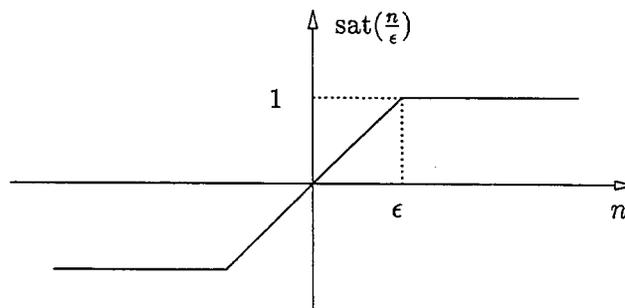


Figura 3.5: Função  $\text{sat}(n/\epsilon)$

Com esta modificação na lei de controle, a trajetória do estado do sistema confinada na camada limite é apenas uma aproximação da dinâmica desejada sobre a superfície de deslizamento. Assim, a estabilidade assintótica não é mais garantida, mas apenas a estabilidade numa região de largura  $\epsilon$  a partir do ponto de equilíbrio. Observe que quanto menor o valor de  $\epsilon$ , mais próximo do deslizamento ideal será a trajetória do sistema.

Outras abordagens para a eliminação do *chattering* são encontradas na literatura de teoria de estruturas variáveis. Uma delas consiste no uso de um *filtro passa-baixas* [20, 80] na entrada da planta. Entretanto, esta solução apresenta dificuldades teóricas com relação à prova da estabilidade do sistema [1]. Hamerlain *et al.* [31] propõe o uso de um controlador PID (proporcional mais integral mais derivativo) na camada limite. Além disso, outras funções lineares são utilizadas no lugar da função  $\text{sgn}(\cdot)$  da Equação (3.14), como por exemplo a proposta por Ambrosino *et al.* em [1].

Neste trabalho é utilizada a função (3.30) por ser a mais utilizada na literatura e por apresentar bons resultados de eliminação do *chattering*.

### 3.4 Robustez

Em sistemas reais de controle são encontradas várias incertezas devido a erros de modelagem, dinâmicas desprezadas, dinâmicas de alta frequência não modeladas, etc. Além disso, em geral uma planta trabalha em diversos pontos de operação (várias referências), que podem apresentar características físicas diferentes. Portanto, um controlador deve ser capaz de apresentar um bom resultado mesmo na presença destas incertezas e variações paramétricas, ou seja, ele deve ser *robusto* [20, 55, 80].

A primeira coisa a ser observada, do ponto de vista da robustez, é que o comportamento dinâmico do sistema controlado por estrutura variável não depende dos parâmetros da planta. A dinâmica do sistema em modo deslizante depende apenas da superfície de deslizamento (3.4) (variáveis de estado do sistema). Portanto, o sistema é insensível a qualquer variação paramétrica ou perturbação limitada que satisfaçam a chamada “condição de *matching*”<sup>2</sup>, em particular, as perturbações que são aditivas ao sinal de controle [3, 14, 41, 71].

Para analisar a influência de *variações paramétricas* e *perturbações* no sistema, a Equação (3.2) é alterada para:

---

<sup>2</sup>Neste trabalho é utilizado o termo *matching* em inglês por falta de uma tradução consagrada na literatura em português.

$$\dot{x} = [f(x, t) + \Delta f(x, i(t), t)] + [B(x, t) + \Delta B(x, i(t), t)]u(x, t) + p(t) \quad (3.31)$$

onde  $i(t)$  é o vetor de funções representando incertezas paramétricas,  $\Delta f$  e  $\Delta B$  representam o efeito acumulativo de todas as incertezas da planta e  $p(t)$  denota uma perturbação externa determinística [6].

Dois suposições são feitas para o estudo da robustez deste sistema:

**Suposição 3.4.1** *Todas as incertezas e perturbações externas satisfazem a condição de matching, o que significa que  $\Delta f$ ,  $\Delta B$  e  $p(t)$  pertencem à imagem da matriz  $B(x, t)$  para todo  $x$  e  $t$ .*

Desta forma todas as incertezas podem ser agrupadas num único vetor de funções:

$$\xi(x, i, p, u, t), \quad (3.32)$$

e a Equação (3.31) pode ser reescrita como:

$$\dot{x} = f(x, t) + B(x, t)u(x, t) + B(x, t)\xi(x, i, p, u, t). \quad (3.33)$$

**Suposição 3.4.2** *Existe uma função positiva contínua limitada  $\rho(x, t)$  que satisfaz:*

$$\|\xi(x, i, p, u, t)\| \leq \rho(x, t). \quad (3.34)$$

Isto significa dizer que as incertezas paramétricas e perturbações externas são superiormente limitadas.

Usando o controle  $u(x, t)$  da Equação (3.23) para controlar o sistema com incertezas é necessário escolher um sinal de controle  $u_n(x, t)$  de forma que:

$$\dot{V}(x, \sigma, t) = \sigma^T(x, t)M \frac{\partial \sigma}{\partial x} B(x, t)u_n(x, t) + [B(x, t)\xi(x, i, p, u, t)] < 0. \quad (3.35)$$

Para satisfazer esta desigualdade, continua sendo possível utilizar a lei de controle (3.27), utilizando a Equação (3.25), com o cuidado de se escolher os ganhos da matriz  $K$  suficientemente altos.

Alternativamente, pode ser utilizada a estrutura de controle proposta por DeCarlo *et al.* [14, 15]:

$$u_n(x, t) = \begin{cases} -\frac{B^T \left[ \frac{\partial \sigma}{\partial x} \right]^T \sigma}{\left\| B^T \left[ \frac{\partial \sigma}{\partial x} \right]^T \sigma \right\|} [\rho(x, t) + \alpha(x, t)], & \text{se } \sigma(x, t) \neq 0 \\ 0, & \text{caso contrário} \end{cases} \quad (3.36)$$

onde  $\alpha(x, t)$  deve ser determinado.

Escolhendo novamente  $M = I_m$ , segue que, para  $\sigma(x, t) \neq 0$ :

$$\begin{aligned} \dot{V}(x, \sigma, t) &= -\sigma^T \frac{\frac{\partial \sigma}{\partial x} B B^T \left[ \frac{\partial \sigma}{\partial x} \right]^T \sigma}{\left\| B^T \left[ \frac{\partial \sigma}{\partial x} \right]^T \sigma \right\|} [\rho(x, t) + \alpha(x, t)] \\ &\quad + \sigma^T \frac{\partial \sigma}{\partial x} B \xi(x, i, p, u, t) \leq - \left\| B^T \left[ \frac{\partial \sigma}{\partial x} \right]^T \sigma \right\| \alpha(x, t) \end{aligned} \quad (3.37)$$

Escolhendo  $\alpha(x, t) > 0$ , está garantida a estabilidade da trajetória do sistema para a superfície de deslizamento utilizando a lei de controle (3.36).

Portanto, utilizando qualquer das duas leis de controle aqui apresentadas, todas as variações paramétricas e perturbações de (3.31) são atenuadas, provando então a robustez do controlador.

### 3.5 Exemplo de Aplicação

Deseja-se neste exemplo de aplicação o controle de posição (e velocidade) das juntas de um robô manipulador. Para isto, a lei de controle projetada considera as juntas do manipulador independentemente, ou seja, sem acoplamento entre elas. Assim, cada junta é controlada como um sistema com uma entrada e uma saída, e o acoplamento é tratado como se fosse uma perturbação externa.

Considere o robô manipulador de  $n$  graus de liberdade apresentado no Capítulo 2, sem contato com o ambiente. Como não existe nenhuma força externa atuando sobre ele ( $F_e = 0$ ), a equação dinâmica (2.40) pode ser reescrita como:

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = u. \quad (3.38)$$

O vetor de estados deste sistema é apresentado na Equação (2.41). Para facilitar a visualização do desenvolvimento do controlador é feita a seguinte mudança de nomenclatura das variáveis de estado:

$$\begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \theta \\ \omega \end{bmatrix} \quad (3.39)$$

onde está claro que  $\theta$  e  $\omega$  são as posições e velocidades das juntas, respectivamente.

Considerando que o termo de forças gravitacionais  $G(q)$  pode ser calculado de forma exata em tempo real, de forma que ele pode ser eliminado da Equação (3.38), o sistema pode ser expresso em equações de estado na forma regular:

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & [-H^{-1}(q)C(q, \dot{q})] \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ H^{-1}(q) \end{bmatrix} u, \quad (3.40)$$

ou na forma da Equação (3.5):

$$\dot{q}_1 = f_1(q, \dot{q}, t) = \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \quad (3.41)$$

$$\dot{q}_2 = f_2(q, \dot{q}, t) + B_2(q, \dot{q}, t)u = \begin{bmatrix} 0 & [-H^{-1}(q)C(q, \dot{q})] \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} + \begin{bmatrix} H^{-1}(q) \end{bmatrix} u$$

Como é desejado controlar as posições  $\theta$  e velocidades  $\omega$  das juntas, uma escolha natural da superfície de deslizamento é:

$$\sigma(\theta, \omega, t) = \begin{bmatrix} K_1 & K_2 \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + r(t) \quad (3.42)$$

onde  $K_1$  e  $K_2$  são matrizes de ganhos positivos e  $r(t)$  é um sinal de referência a ser regulado ou seguido.

Utilizando a lei de controle (3.14), selecionando os ganhos  $K_1$  e  $K_2$  e fornecendo um sinal de referência  $r(t)$  adequado, é obtido o comportamento desejado do sistema (regulação de referência ou seguimento de trajetória).

### 3.5.1 Simulação

Como exemplo de simulação numérica será considerado um robô manipulador de dois graus de liberdade num plano horizontal, modelado na Seção 2.4. Este manipulador pode ser os elos 1 e 2 de um manipulador SCARA. Neste caso não é necessário a compensação de gravidade, uma vez que esta não atua sobre os dois primeiros elos do manipulador [13, 43].

A superfície de deslizamento (3.42) pode ser expressa como:

$$\sigma(\theta, \omega, t) = \begin{bmatrix} k_{11} & 0 & k_{21} & 0 \\ 0 & k_{12} & 0 & k_{22} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \omega_1 \\ \omega_2 \end{bmatrix} + \begin{bmatrix} r_1(t) \\ r_2(t) \end{bmatrix} = \begin{bmatrix} k_{11}\theta_1 + k_{21}\omega_1 + r_1(t) \\ k_{12}\theta_2 + k_{22}\omega_2 + r_2(t) \end{bmatrix} \quad (3.43)$$

onde  $\theta_i$  e  $\omega_i$  são a posição e velocidade da junta  $i$ , respectivamente, e  $k_{ij}$  são ganhos estritamente positivos.

Assim, cada elemento do vetor  $\sigma(\theta, \omega, t)$  é:

$$\sigma_i(\theta_i, \omega_i, t) = k_{1i} \theta_i + k_{2i} \omega_i + r_i(t). \quad (3.44)$$

Considerando que a posição desejada da junta  $i$ ,  $\theta_{di}$  continuamente diferenciável e limitada, é solução da Equação (3.44), o sinal de referência  $r_i(t)$  é dado por:

$$r_i(t) = -k_{1i} \dot{\theta}_{di} - k_{2i} \omega_{di} \quad (3.45)$$

onde  $\omega_{di}$  é a primeira derivada temporal da posição desejada  $\theta_{di}$ .

Substituindo a referência  $r_i(t)$  (3.45) na Equação (3.44) é obtido:

$$\sigma_i(\theta_i, \omega_i, t) = k_{1i} (\theta_i - \theta_{di}) + k_{2i} (\omega_i - \omega_{di}). \quad (3.46)$$

Definindo os vetores de erros negativos de posição e velocidade das juntas como:

$$\tilde{\theta} = \theta - \theta_d \quad (3.47)$$

e:

$$\tilde{\omega} = \omega - \omega_d, \quad (3.48)$$

a Equação (3.46) pode ser reescrita como:

$$\sigma_i(\theta_i, \omega_i, t) = k_{1i} \tilde{\theta}_i + k_{2i} \tilde{\omega}_i. \quad (3.49)$$

Se a ação de controle aplicada à planta conseguir manter a trajetória do sistema na superfície de deslizamento a partir de algum tempo  $t_1$ , então a equação acima se iguala a zero:

$$\sigma_i(\theta_i, \omega_i, t) = k_{1i} \tilde{\theta}_i + k_{2i} \tilde{\omega}_i = 0. \quad (3.50)$$

Isolando  $\tilde{\omega}_i$ :

$$\tilde{\omega}_i = \dot{\tilde{\theta}}_i = -\frac{k_{1i}}{k_{2i}} \tilde{\theta}_i. \quad (3.51)$$

Portanto, a partir do tempo  $t_1$ , os erros de posição e velocidade vão convergir exponencialmente para zero.

A velocidade de convergência do estado do sistema para a superfície de deslizamento depende dos valores dos ganhos  $k_{ij}$  da Equação (3.51). Quanto maior o ganho  $k_{1i}$  (e menor o ganho  $k_{2i}$ ), mais rápida é a convergência dos erros da junta  $i$  para zero. Entretanto, uma convergência mais rápida requer um esforço de controle maior.

Para simulação numérica é utilizado o software Matlab<sup>©</sup> versão 4.2. Os parâmetros do robô manipulador utilizado na simulação são os do manipulador SCARA do Apêndice A.

Os elos 3 e 4 do robô manipulador são considerados como sendo uma carga fixa ao efetuador final, conforme proposto na Sub-seção 2.4.3. Portanto, a massa e a inércia da carga são:

$$l_c = m_3 + m_4 = 3,5 \text{ Kg};$$

$$I_c = I_3 + I_4 = 0,20 \text{ Kg m}^2.$$

O período de amostragem [34, 52, 76] do controlador é 1,0 ms. Os parâmetros do controlador utilizados são:

$$k_{11} = 10,0 ; k_{21} = 1,0 ; k_{12} = 10,0 ; k_{22} = 1,0 ;$$

$$u_{01} = 70,0 \text{ N m} ; u_{02} = 50,0 \text{ N m};$$

$$\epsilon_1 = 0,05 ; \epsilon_2 = 0,15.$$

Estes valores foram obtidos empiricamente (tentativa e erro) analisando o módulo do sinal de controle e os erros de segmento de referência. Recentemente, a literatura tem apresentado metodologias baseadas em controle adaptativo, nebuloso, redes neurais e algoritmos genéticos aplicados à sintonia de controladores clássicos e avançados [12, 44, 51, 74, 76] que podem ser utilizados aqui.

As posições iniciais  $\theta_i$  e finais  $\theta_f$  das juntas são:

$$\theta_i = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{ rad} \quad \theta_f = \begin{bmatrix} 2,0 \\ 1,5 \end{bmatrix} \text{ rad}$$

As trajetórias de posição entre os pontos iniciais e finais são obtidas utilizando um algoritmo que gera um polinômio de interpolação cúbica [43], implementado no robô SCARA. Cada trajetória de posição é uma *função linear misturada com parábolas*<sup>3</sup>. Ela possui três partes: as partes inicial e final são parabólicas, e a parte central é linear.

As curvas de posição são obtidas com aceleração de  $5,0 \text{ rad/s}^2$  e velocidade máxima de  $3,0 \text{ rad/s}$  para ambas as juntas. As velocidades alcançadas são suficientes para que as interações dinâmicas entre os elos sejam significativas, permitindo testar a robustez do controlador.

As Figuras 3.6 e 3.7 apresentam, em linhas pontilhadas, os perfis de posições e velocidades desejadas, respectivamente. Além disso, elas apresentam os resultados obtidos na simulação, em linhas contínuas.

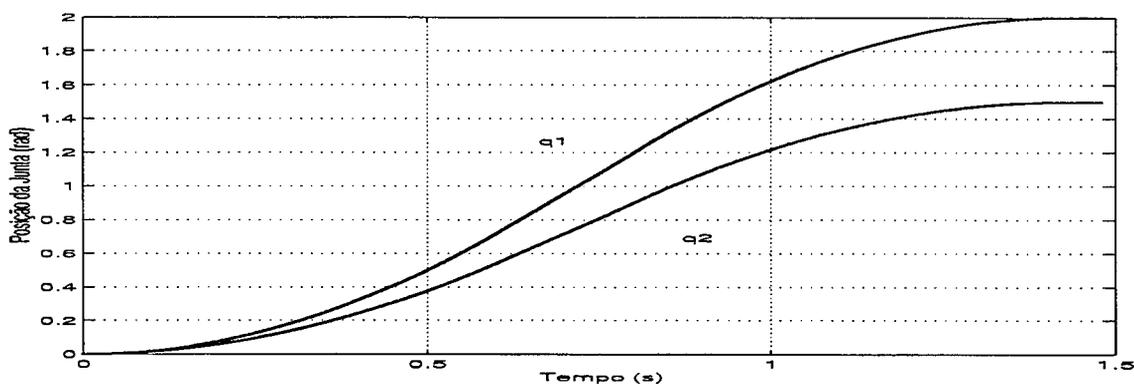


Figura 3.6: Posições das Juntas ( $\theta_1 = q_1$  ;  $\theta_2 = q_2$ ) - Simulação

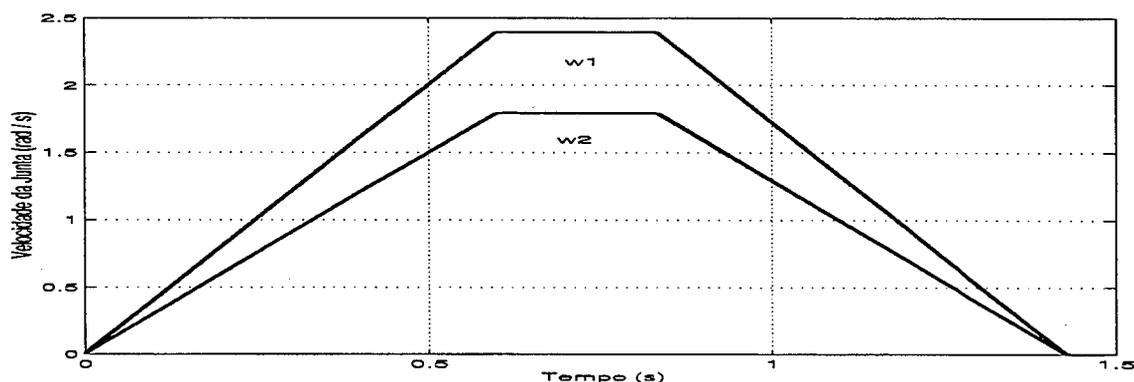


Figura 3.7: Velocidades das Juntas ( $\omega_1 = w_1$  ;  $\omega_2 = w_2$ ) - Simulação

A Figura 3.8 apresenta a proximidade das superfícies de deslizamento, ou seja, as funções  $\sigma_1(x, t)$  e  $\sigma_2(x, t)$  na camada limite.

<sup>3</sup>Termo original em inglês: *linear function with parabolic blends*.

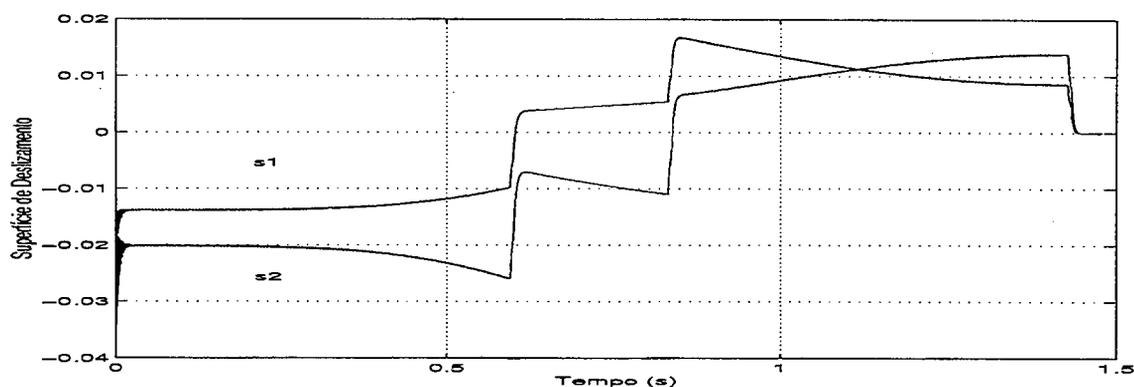


Figura 3.8: Proximidade das Superfícies ( $\sigma_1 = s_1$  ;  $\sigma_2 = s_2$ ) - Simulação

A Figura 3.9 apresenta o controle aplicado no robô manipulador. Observe que ele apresenta *chattering* significativo apenas no começo da simulação, e se mantém entre os níveis máximos dados pelos parâmetros do controlador, que são bem abaixo dos limites do robô utilizado para implementação prática (Apêndice A).

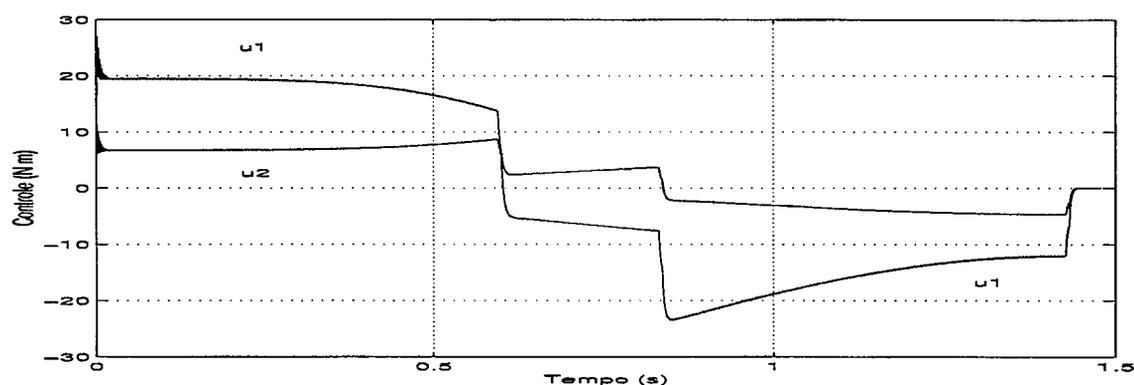


Figura 3.9: Sinais de Controle ( $u_1$  ;  $u_2$ ) - Simulação

As Figuras 3.10 e 3.11 apresentam os erros de posições e velocidades, respectivamente. Os erros máximos de posição estão em torno de  $2,0 \times 10^{-3}$  rad e os de velocidade em torno de  $2,0 \times 10^{-2}$  rad/s, o que representa um bom resultado.

### 3.5.2 Implementação

O mesmo controlador utilizado na simulação numérica foi implementado em Xoberon [56, 50] para controlar um robô SCARA industrial. O código fonte do algoritmo de controle é apresentado no Apêndice B.

Na experimentação prática são utilizados basicamente os mesmos parâmetros do controlador, inclusive o período de amostragem (1,0 ms), e as mesmas referências usados na simulação. O único ajuste necessário é o aumento das camadas limite para:

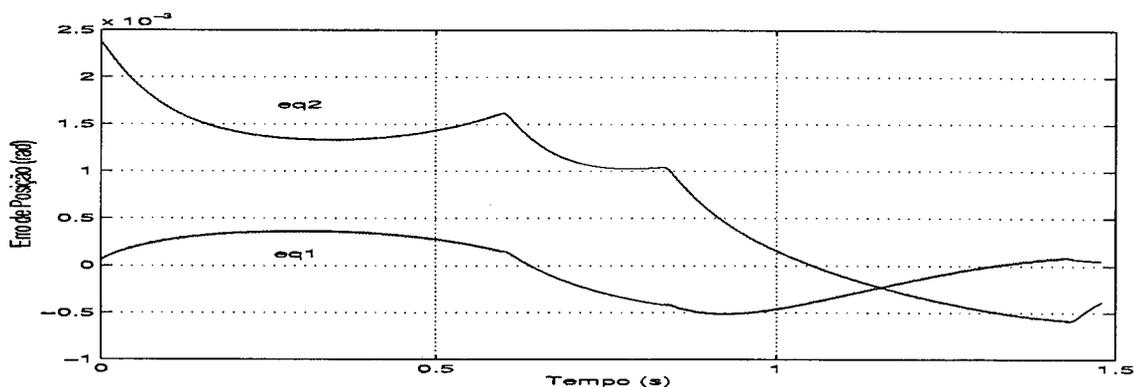


Figura 3.10: Erros de Posição das Juntas ( $\theta_d - \theta$ ) - Simulação

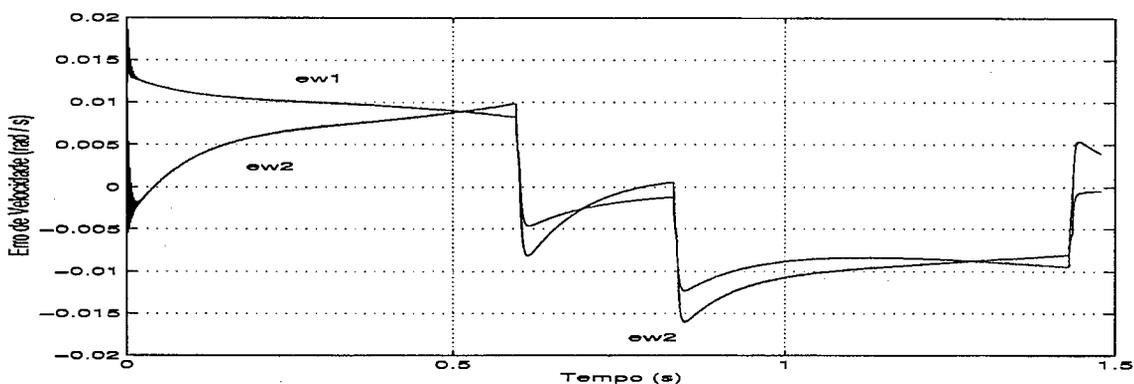


Figura 3.11: Erros de Velocidade das Juntas ( $\omega_d - \omega$ ) - Simulação

$$\epsilon_1 = 0,10 \quad ; \quad \epsilon_2 = 0,30$$

uma vez que, com o valor utilizado na simulação, o robô apresenta *chattering*. Isto se deve às limitações físicas dos atuadores, que não foram consideradas na simulação numérica.

Os parâmetros físicos disponíveis do manipulador (Apêndice A) utilizados na simulação não são exatos. Além disso, não são simulados os atritos, as flexibilidades e a dinâmica dos atuadores. Assim, como já era esperado, os resultados na prática são diferentes da simulação. Os resultados obtidos na implementação em bancada são mostrados nas figuras abaixo.

As Figuras 3.12 e 3.13 apresentam os perfis de posições e velocidades, respectivamente. As referências são mostradas em linhas pontilhadas e os valores medidos em linhas contínuas.

A Figura 3.14 apresenta a proximidade das funções  $\sigma_1$  e  $\sigma_2$  da superfície de deslizamento.

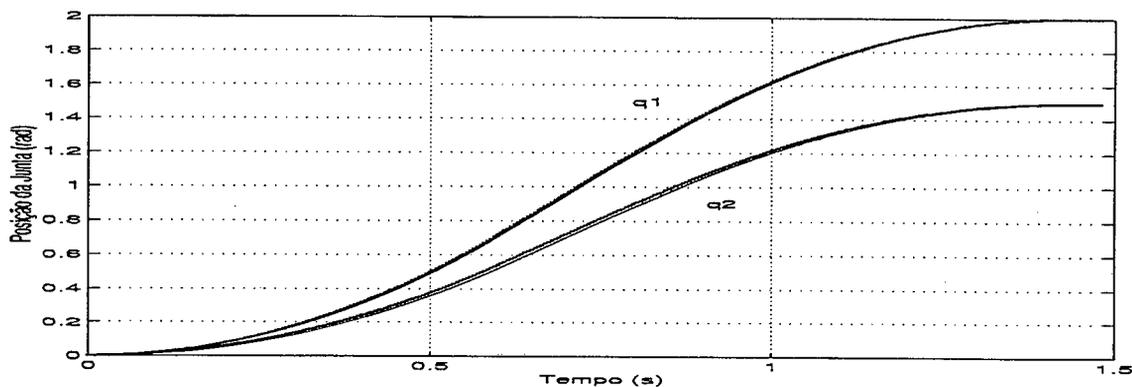


Figura 3.12: Posições das Juntas ( $\theta_1 = q_1$  ;  $\theta_2 = q_2$ ) - Implementação Real

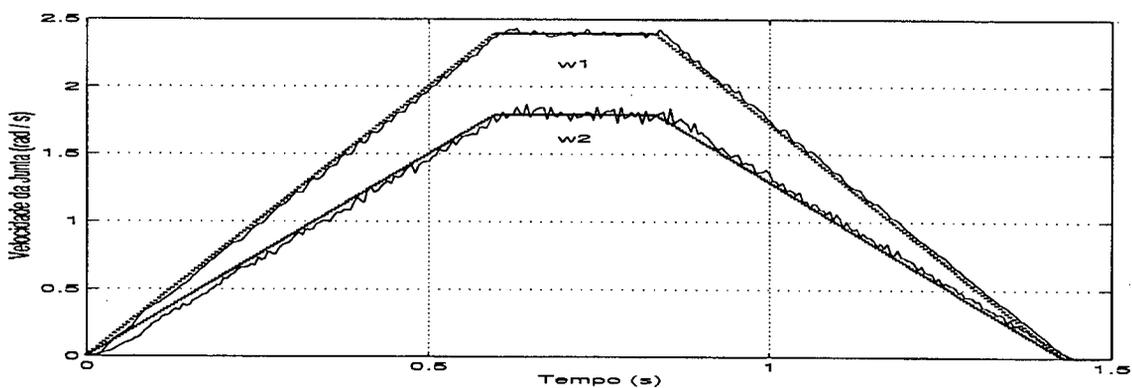


Figura 3.13: Velocidades das Juntas ( $\omega_1 = w_1$  ;  $\omega_2 = w_2$ ) - Implementação Real

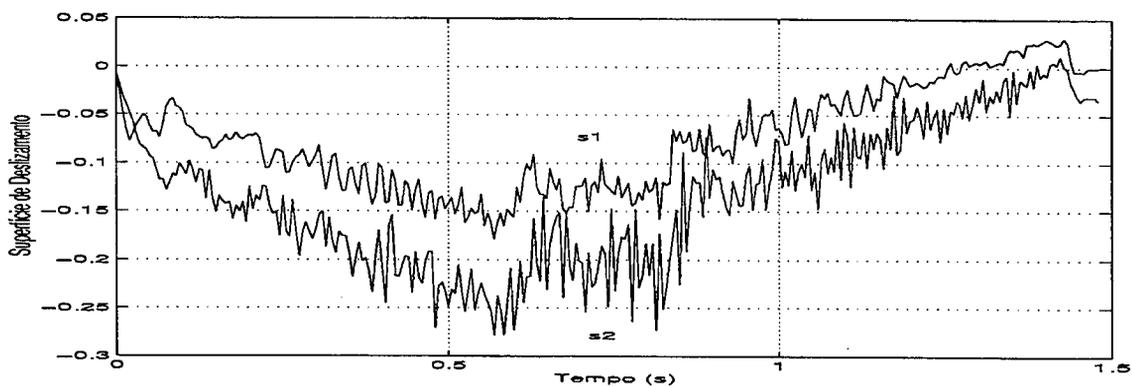


Figura 3.14: Proximidade das Superfícies ( $\sigma_1 = s_1$  ;  $\sigma_2 = s_2$ ) - Implementação Real

Os sinais de controle são mostrados na Figura 3.15.

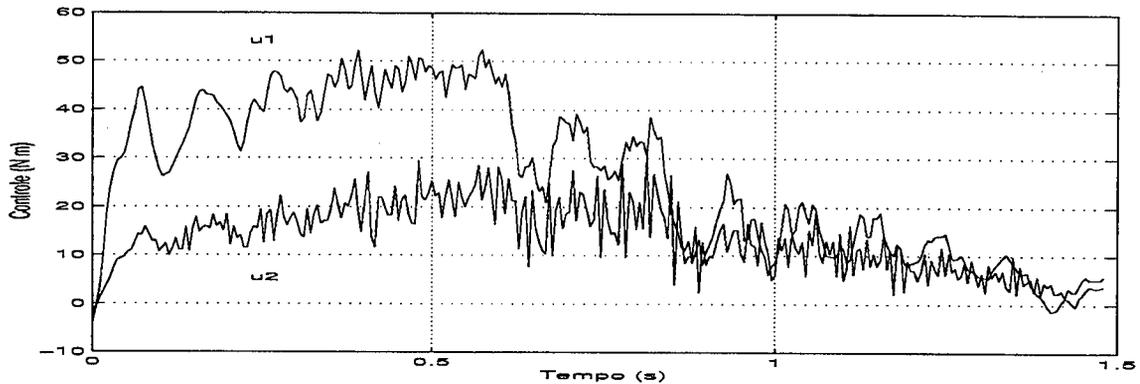


Figura 3.15: Sinais de Controle ( $u_1$  ;  $u_2$ ) - Implementação Real

Os erros de posição, mostrados na Figura 3.16, estão na faixa de  $2,0 \times 10^{-2}$  rad e os de velocidade, mostrados na Figura 3.17, estão em torno de  $1,0 \times 10^{-1}$  rad/s.

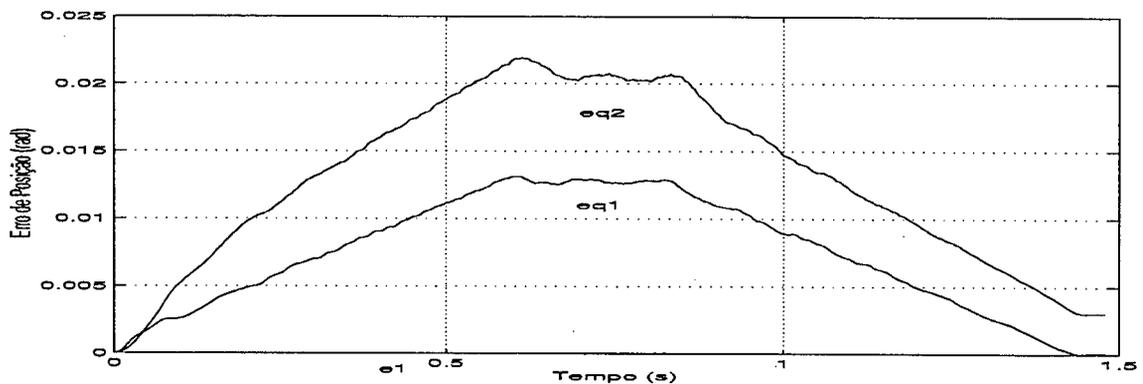


Figura 3.16: Erros de Posição das Juntas ( $\theta_d - \theta$ ) - Implementação Real

Apesar destes erros serem maiores do que os da simulação numérica, devido principalmente ao fato da camada limite utilizada ser maior que a da simulação, eles podem ser considerados satisfatórios. Além disso, os erros em regime permanente são bem próximos de zero.

### 3.6 Conclusões

Neste capítulo foi apresentada a teoria de estrutura variável com modos deslizantes, com aplicação em um robô manipulador de  $n$  graus de liberdade. Foram apresentadas simulações e resultados práticos da utilização deste controlador para controlar a posição e velocidade dos elos 1 e 2 de um robô do tipo SCARA.

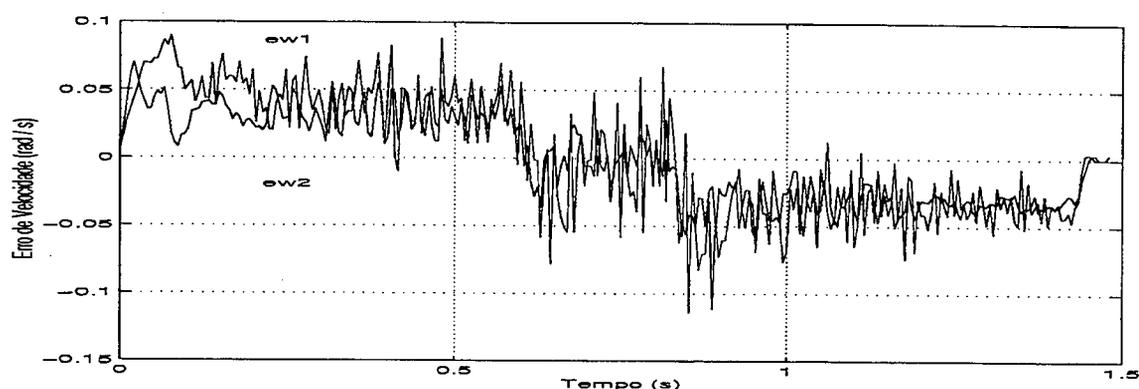


Figura 3.17: Erros de Velocidade das Juntas ( $\omega_d - \omega$ ) - Implementação Real

Foi demonstrada a robustez deste tipo de controlador, bem como suas vantagens para aplicações em robôs manipuladores, uma vez que nem sempre se dispõe de modelos matemáticos que os representem com uma exatidão adequada, e devido as suas características físicas (atritos, cargas variáveis, etc.).

A sintonia dos controladores, tanto na simulação como na implementação prática, foi feita de forma empírica, o que provavelmente não proporciona o melhor resultado possível (tempo de resposta, sobre-sinal, etc.). Para reparar esta deficiência, podem ser utilizadas técnicas de *sintonia automática* [12, 44, 51, 74, 76] para se obter valores ótimos dos parâmetros do controlador.

A técnica de controle aqui apresentada será utilizada no Capítulo 4, para realizar o controle híbrido de força e posição de um robô manipulador, que é o foco principal deste trabalho.

# Capítulo 4

## Controle Híbrido

Neste capítulo é abordado o controle simultâneo de força e posição através de um controlador híbrido baseado nos trabalhos de Mason [47] e Raibert e Craig [53].

É apresentada a estrutura do controlador híbrido, abordando desde o mapeamento do sinal de controle entre os espaços de tarefa e de juntas, até a especificação da tarefa sujeita a restrições do ambiente. Uma atenção maior é dada ao controle de força em ambientes de elasticidade não linear.

Antes do projeto do controlador (através de um exemplo de aplicação), são apresentados alguns pontos relacionados à estabilidade do sistema. Para isto, é discutido o acoplamento entre os sub-controladores (controlador de força e controlador de posição), que são implementados utilizando a teoria de estrutura variável.

### 4.1 Introdução

Para utilizar robôs manipuladores em diversas tarefas encontradas na indústria, onde existe interação do manipulador com o ambiente, um controle puro de posição não é adequado. Nestes casos, é indicado o uso de um controlador simultâneo de força e posição, como o controlador híbrido.

A filosofia de funcionamento do controlador híbrido é a divisão do controle simultâneo de posição do efetuador final e de força de contato em dois sub-problemas separados e desacoplados. Este controlador é obtido utilizando dois *sub-controladores*<sup>1</sup> distintos, um de força e outro de posição. As direções controladas em força e posição são definidas no espaço cartesiano, o que é mais natural para a especificação de tarefas.

O projeto de um controlador envolve, além da definição da sua estrutura, a seleção

---

<sup>1</sup>Neste trabalho, é utilizado o termo *sub-controlador* para se referir separadamente a cada um dos controladores (controlador de força e controlador de posição) do esquema de controle híbrido.

dos ganhos para satisfazer as exigências das tarefas. O controlador deste trabalho deve ser suficientemente geral e robusto, para lidar com diversas tarefas que envolvam a manipulação de materiais flexíveis.

Em robótica é importante considerar a robustez com relação a quatro principais perturbações: distúrbios externos limitados, flexibilidades, dinâmica dos atuadores e atritos [51]. Para garantir robustez ao sistema, os sub-controladores são projetados utilizando a teoria de estrutura variável, apresentada no Capítulo 3.

## 4.2 Espaço do Controle

Muito freqüentemente é desejado que o efetuator final siga linhas retas ou outras formas de caminho descritas no espaço de tarefa [13]. A especificação da grande maioria das tarefas neste espaço é muito mais fácil e intuitiva [18], devido a sua ortogonalidade (sistema cartesiano).

Assim, para utilizar o *controle no espaço de juntas* [13, 40] deve ser feita uma conversão dos caminhos especificados no espaço de tarefa para o espaço de juntas, o que é computacionalmente dispendioso.

### 4.2.1 Controle Cartesiano

Uma opção alternativa ao controle no espaço de juntas é o *controle cartesiano*, também conhecido como *controle no espaço de tarefa* [13, 40]. Nesta abordagem, a posição do manipulador (posição das juntas) é transformada para o espaço de tarefa através da função cinemática direta (2.5) que normalmente é bem conhecida. Esta posição é então comparada com a referência, gerando valores de erros para o controlador no espaço cartesiano.

Da mesma forma, para o controle de força, o processamento dos erros de força é feito no espaço de tarefa, utilizando valores de força descritos no sistema cartesiano de coordenadas da base (espaço de tarefa).

### 4.2.2 Controle Cartesiano Usando a Jacobiana Transposta

O controlador implementado neste trabalho utiliza o esquema do *controle cartesiano usando a Jacobiana transposta* [13]. Neste esquema o vetor de forças/torques de controle  $u_x$  no espaço de tarefa é mapeado através da matriz Jacobiana transposta,  $J^T(q)$ , para o torque de controle das juntas  $u$  equivalente. Esta transformação se baseia no princípio do trabalho virtual (Sub-seção 2.3.3). Um controlador de posição utilizando

esta estrutura é apresentado na Figura 4.1.

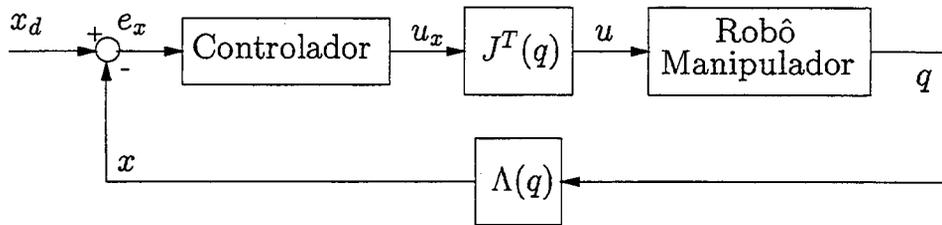


Figura 4.1: Controlador Cartesiano Usando a Jacobiana Transposta

Este esquema de controle surgiu intuitivamente. Ele pode ser visto como se fosse o mapeamento de uma força no espaço de tarefa para o espaço de juntas, onde esta força aplicada no efetuador final do manipulador tem o intuito de levá-lo na direção em que os erros cartesianos serão reduzidos. Apesar desta estrutura de controle funcionar bem, a estabilidade do sistema não pode ser garantida [13].

### 4.3 Fundamentos do Controlador Híbrido

O controlador híbrido de força e posição, aqui apresentado, utiliza o esquema original proposto por Raibert e Craig [53], mostrado na Figura 4.2. Neste controlador cada grau de liberdade no espaço de tarefa é controlado por uma única malha de controle de força ou posição, de acordo com a natureza da tarefa.

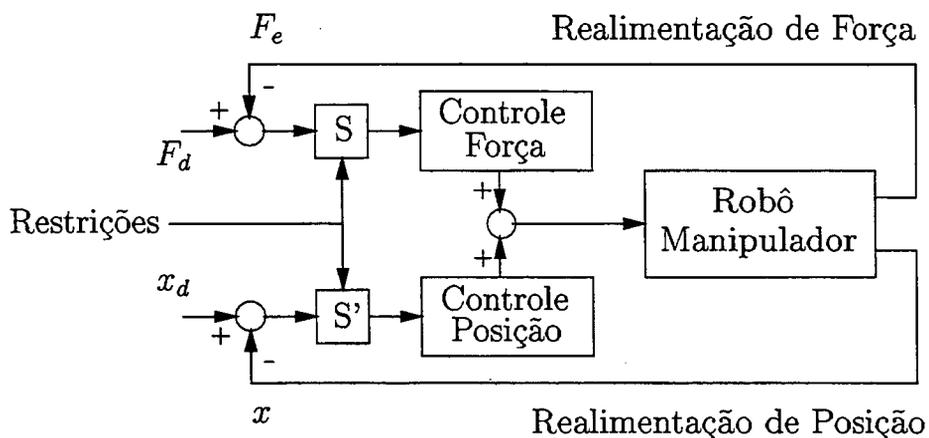


Figura 4.2: Controlador Híbrido de Força e Posição

A matriz diagonal  $S$ , definida pelas restrições, é utilizada para projetar o erro de

força do robô no sub-espaço controlado em força, enquanto a matriz  $S'$ , complemento ortogonal de  $S$ , projeta o erro de posição no sub-espaço controlado em posição [60].

### 4.3.1 Movimentos Restritos

Como a aplicação de uma força exige o contato do manipulador com o meio, o manipulador não pode se mover livremente em todas as direções. Portanto, este tipo de movimento é comumente chamado de *movimento restrito* [5, 10, 30, 47]. Um exemplo clássico são as tarefas de montagem [58, 59], onde algumas peças devem ser movimentadas em contato com outras.

A natureza geométrica e mecânica do meio com o qual o robô manipulador interage cria um conjunto de restrições de força e/ou movimentos chamadas de *naturais* [2, 13, 47, 53]. Estas restrições não estão relacionadas aos movimentos e forças de contato desejados, elas dependem apenas da situação de contato. De acordo com as trajetórias desejadas especificadas em força e posição, são criadas novas restrições, estas chamadas *artificiais* [2, 13, 47, 53].

Como exemplo, considere o parafuso sendo inserido numa porca, mostrado na Figura 4.3. Por suposição o passo do parafuso é de  $\rho$  m e a taxa angular de inserção desejada é constante:  $\alpha$  rad/s. Além disso, é considerado que não existe atrito entre o parafuso e a porca, e a ferramenta utilizada pelo robô não se desloca lateralmente quando encaixada no parafuso.

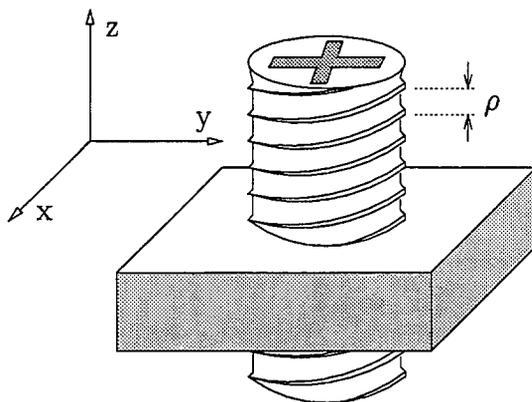


Figura 4.3: Inserção de um parafuso numa porca .

As restrições naturais que o orifício impõe ao parafuso, e as restrições artificiais definidas pela trajetória desejada são apresentadas nas Tabelas 4.1 e 4.2, respectivamente. Note que quando uma restrição natural de posição existe numa determinada

direção no espaço de tarefa, uma restrição artificial de força deve ser especificada para esta direção, e vice-versa.

Tipo da Restrição	Valor
velocidade linear x	$\dot{p}_x = 0$ m/s
velocidade linear y	$\dot{p}_y = 0$ m/s
velocidade angular x	$\omega_x = 0$ rad/s
velocidade angular y	$\omega_y = 0$ rad/s
força z	$f_z = 0$ N
torque z	$\tau_z = 0$ N m

Tabela 4.1: Restrições Naturais

Tipo da Restrição	Valor
velocidade linear z	$\dot{p}_z = \rho\alpha$ m/s
velocidade angular z	$\omega_z = \alpha$ rad/s
força x	$f_x = 0$ N
força y	$f_y = 0$ N
torque x	$\tau_x = 0$ N m
torque y	$\tau_y = 0$ N m

Tabela 4.2: Restrições Artificiais

Tanto as restrições naturais como as artificiais estão relacionadas a uma *superfície no espaço de tarefa*, mas existe uma grande diferença entre as duas. No caso das restrições naturais, as restrições de força ocorrem no sentido tangente a esta superfície e as de posição no sentido normal a esta superfície. Já no caso das restrições artificiais ocorre exatamente o contrário.

Considere novamente a Figura 4.3, tomando o plano *yz* como uma superfície. As restrições naturais relacionadas a esta superfície são:  $\dot{p}_x = 0$  m/s (normal) e  $f_z = 0$  N (tangente). Já as artificiais são:  $\dot{p}_z = \rho\alpha$  m/s (tangente) e  $f_x = 0$  N (normal).

A idéia básica do controle híbrido é que o controlador deve se preocupar unicamente com as restrições artificiais, ignorando as naturais. Isto é motivado assumindo que não existe necessidade de controlar as variáveis sujeitas a restrições do ambiente, pois estas já estão “controladas”.

As restrições particionam os graus de liberdade do manipulador no espaço de tarefa em dois conjuntos ortogonais que devem ser controlados de acordo com o tipo da restrição: força ou posição. Os torques gerados separadamente em cada malha de controle devem trabalhar de forma cooperativa para controlar cada junta do manipulador, satisfazendo tanto as restrições de força como as de posição.

### 4.3.2 Especificação de Tarefas

Antes que o projeto do controlador possa ser feito, é necessário saber quais são os requerimentos da tarefa sob uma perspectiva de controle, ou seja, quais são as *variáveis controladas* [20] e quais são as referências para cada variável [39].

A estratégia de controle híbrido de força e posição não permite que sejam simultaneamente impostas referências arbitrárias de força e posição ao longo de qualquer grau de liberdade [10, 43]. As especificações devem ser compatíveis com a tarefa em execução, ou seja, nas direções onde existe uma restrição geométrica não é possível definir uma trajetória de posição, e onde não existe tal restrição não é possível impor uma trajetória de força.

A especificação dos espaços controlados em força e posição é extremamente importante para o bom funcionamento do controlador. Por exemplo, se um contato ocorrer numa direção na qual não era esperado, o manipulador e/ou o meio podem ser danificados, uma vez que não existe um controle de força nesta direção (direção controlada em posição).

Desta forma, se uma tarefa envolve várias fases para ser completamente executada, várias especificações de naturezas de controle por direção devem ser feitas, uma para cada fase da tarefa. Esta mudança de estrutura do controlador é um dos inconvenientes deste esquema de controle, pois a continuidade da lei de controle só pode ser garantida numa tarefa perfeitamente estruturada, onde o planejamento possa ser realizado com uma grande precisão.

## 4.4 Meios de Elasticidade Não Linear

As tarefas nas quais será utilizado o controlador híbrido aqui projetado, situações onde existe o contato do manipulador com meios de elasticidade não linear, possuem dois problemas principais do ponto de vista de controle:

- a elasticidade do contato é não linear, ou seja, a rigidez do meio varia de acordo com a deformação imposta;
- existem forças, criadas pela deformação do meio, que atuam nas direções controladas em posição.

O primeiro problema está associado ao controle de força e exige que o mesmo seja robusto de forma que o sistema seja estável para os diversos valores de rigidez das peças. O segundo está associado ao controle de posição, pois estas forças atuam

como perturbações no sistema, exigindo então, que o sub-controlador de posição seja insensível a elas.

Além dos problemas supra citados, os controladores clássicos assumem o conhecimento do modelo da planta [20], mas este modelo não é exato devido a mudanças de parâmetros, dinâmicas não modeladas, distúrbios externos, ruído de sensores, etc. Portanto, estes controladores não apresentam a performance desejada nas aplicações estudadas neste trabalho.

Visando solucionar estes problemas acima apresentados, os sub-controladores aqui implementados baseiam-se na teoria de estrutura variável apresentada no Capítulo 3.

## 4.5 Projeto do Controlador

Geralmente no projeto de controladores híbridos as interações do manipulador com o ambiente são consideradas completamente estáticas (restrições geométricas) ou puramente dinâmicas (impedância). Entretanto, para uma análise mais geral é interessante supor que as interações são uma combinação das duas acima [16], como é considerado neste trabalho.

Deseja-se controlar um robô manipulador rígido com  $n$  graus de liberdade. Para especificar quais graus de liberdade no espaço de tarefa são controlados em força e quais são controlados em posição, é utilizada uma *matriz de seleção*  $S$  [53, 60]:

$$S = \begin{bmatrix} s_1 & & & 0 \\ & s_2 & & \\ & & \ddots & \\ 0 & & & s_m \end{bmatrix} \quad (4.1)$$

onde  $s_i = 1$  indica um grau de liberdade controlado em força,  $s_i = 0$  controlado em posição e  $m$  é o número de graus de liberdade no espaço de tarefa.

Para implementação é também utilizado o *complemento ortogonal* da matriz diagonal  $S$ :

$$S' = I_m - S \quad (4.2)$$

onde  $I_m$  é uma matriz identidade de ordem  $m$ .

Uma vez estipuladas as variáveis a serem controladas no espaço de tarefa, definindo assim a matriz  $S$ , é feito o projeto dos sub-controladores de modos deslizantes, utilizando a teoria de estrutura variável.

### 4.5.1 Controle de Força

Existem basicamente dois tipos de controle de força:

- controle baseado em laço interno de posição<sup>2</sup>;
- controle explícito de força.

Os controladores baseados em laço interno de posição trabalham em cascata. Como o próprio nome indica, eles possuem um laço de controle de força externo que fornece um comando de posição (referência de posição) para um controlador de posição interno, que pode ser um controlador de posição qualquer.

As técnicas de *controle explícito de força* utilizam algum tipo de controle como o PID, bem como outros tipos de controladores simples ou mais complexos para criar uma malha de controle baseada em força [10, 29, 42, 73, 78, 79]. A lei de controle utiliza os erros de força, diferença entre os valores de referência e medido, para gerar o sinal de controle. Esta é a estratégia utilizada neste trabalho.

#### Superfície de Deslizamento

Para o controle explícito de força utilizando modos deslizantes, é definida a seguinte função do vetor de estados do sistema:

$$\sigma_f = K_f S(F_e - F_d) + Sv \quad (4.3)$$

onde  $F_e$  e  $F_d$  são os vetores de forças medido e desejado no espaço de tarefa, respectivamente;  $v$  é o vetor de velocidades atuais no espaço de tarefa [cf. Equação (2.16)] e  $K_f$  uma matriz diagonal de ganhos positivos.

A realimentação da velocidade  $v$  é utilizada em (4.3) para aumentar o *amortecimento do sistema*. Isto é necessário porque o amortecimento naturalmente existente (como força contra-eleto-motriz num motor elétrico ou atrito viscoso num cilindro hidráulico) pode ser insuficiente, resultando num sistema muito oscilatório.

Note que devido às características das tarefas focadas por este trabalho é desejado aqui apenas uma regulação de força ( $F_e = F_d = Cte_f$ ) e não um seguimento de trajetória de força, ainda que este seguimento seja possível utilizando a técnica de controle à estrutura variável. Neste caso, as derivadas temporais dos vetores de força, desejado ( $\dot{F}_d$ ) e atual ( $\dot{F}_e$ ), devem ser utilizadas na função (4.3).

---

<sup>2</sup>Também chamado de *controle baseado em laço externo de força e controle implícito de força* [29, 79].

Existem várias aplicações onde um robô manipulador deve seguir uma superfície aplicando uma força constante. Por outro lado, aplicações onde a força de contato deve seguir alguma função do tempo arbitrária são raras [13]. Além disso, a medição de forças de contato é ruidosa [18, 29, 73], o que torna desaconselhável a derivação numérica da força medida.

### 4.5.2 Controle de Posição

Para o controle de posição, agora visando um rastreamento de trajetória, é definida a seguinte função do vetor de estados do sistema:

$$\sigma_p = K_p S'(x - x_d) + S'(v - v_d) \quad (4.4)$$

onde  $x$  e  $x_d$  são os vetores de posições no espaço de tarefa atual e desejado, respectivamente,  $v$  e  $v_d$  são os vetores de velocidades no espaço de tarefa atual e desejado, respectivamente e  $K_p$  uma matriz diagonal de ganhos positivos.

As posições  $x$  e velocidades  $v$  no espaço de tarefa podem ser facilmente obtidas a partir das posições e velocidades das juntas utilizando a função cinemática direta (2.5) e a Equação (2.16), que envolve a matriz Jacobiana  $J(q)$  (2.15), respectivamente.

### 4.5.3 Lei de Controle

A partir das duas funções vetoriais (4.3) e (4.4), é obtido o seguinte conjunto de  $m$  equações  $\sigma = 0$ :

$$\sigma = \sigma_f + \sigma_p = \begin{bmatrix} \sigma_{f1} + \sigma_{p1} \\ \sigma_{f2} + \sigma_{p2} \\ \vdots \\ \sigma_{fm} + \sigma_{pm} \end{bmatrix} = 0, \quad (4.5)$$

que define uma superfície no espaço de estados para a qual o estado do sistema deve ser levado e que o conduzirá até o ponto de equilíbrio.

Para manter o estado do sistema deslizando nesta superfície, deve ser aplicado um esforço de controle chaveado do tipo apresentado na Equação (3.14).

Para reduzir o *chattering* devido ao controle chaveado, é utilizada uma camada limite (3.29). Note que desta forma a estabilidade assintótica não é mais garantida, mas apenas a estabilidade local, conforme indicado na Seção 3.3.

Uma vez encontrada a ação de controle no espaço de tarefa, resta apenas transformá-la para o espaço de juntas e aplicá-la no robô manipulador, conforme apresentado no

esquema da Figura 4.1. Esta transformação baseia-se no princípio do trabalho virtual, apresentado na Sub-seção 2.3.3, e é feita utilizando a matriz Jacobiana transposta:

$$u = J^T(q)u_x \quad (4.6)$$

onde  $u$  é o torque de controle no espaço de juntas e  $u_x$  é a força/torque equivalente no efetuador final, no espaço de tarefa.

## 4.6 Acoplamento / Estabilidade

A Figura 4.4 apresenta o esquema detalhado do controlador híbrido de força e posição projetado.

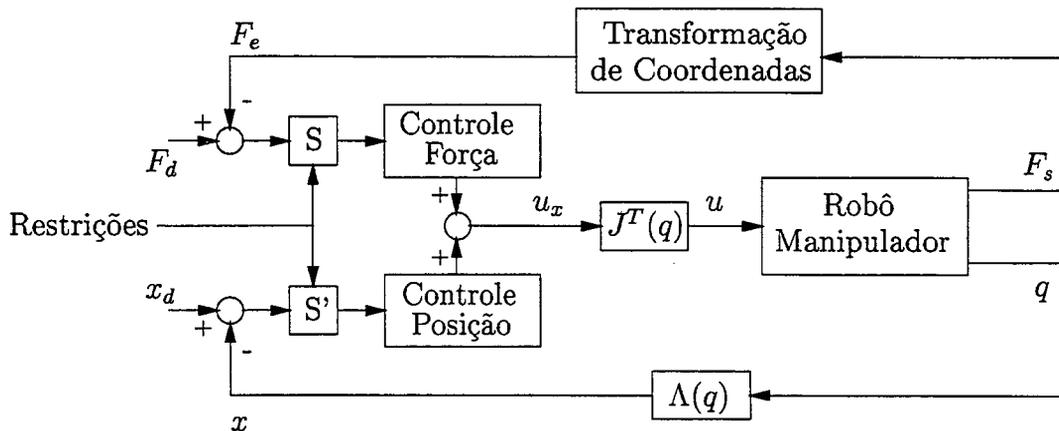


Figura 4.4: Controlador Híbrido de Força e Posição Detalhado

Os sub-controladores de força e posição podem interagir entre si direta ou indiretamente. As *interações diretas* ocorrem tanto quando o sub-controlador de posição afeta o movimento restrito, quanto quando o sub-controlador de força afeta o movimento livre. Já as *interações indiretas* representam a influência mútua dos sub-controladores através das forças inerciais.

As interações diretas podem ser eliminadas através dos sub-controladores de força e posição, mas as interações indiretas são inerentes ao sistema. Geralmente, elas não podem ser eliminadas simplesmente através do projeto dos sub-controladores [66].

A estabilidade dos sub-controladores trabalhando independentemente é apresentada na Sub-seção 3.2.2. O passo inicial para a análise da estabilidade do sistema completo é o estudo do acoplamento entre os sub-controladores acima descrito, o que não se trata de uma tarefa fácil.

Em [77] é abordada uma análise de estabilidade de algoritmos que particionam o controle simultâneo de força e posição em dois sub-espacos. Entretanto, não é apresentado um tratamento rigoroso da estabilidade em malha fechada do controle híbrido de força e posição quando o ambiente é flexível. Experimentos com robôs têm mostrado que uma malha fechada pode ser desestabilizada quando uma restrição dura (ambiente infinitamente rígido) é imposta ao efetuador final [48].

Uma das dificuldades na análise da estabilidade consiste no fato de que um modelamento do contato entre o manipulador e o ambiente [30] é usualmente difícil de ser obtido de forma analítica, devido à complexidade dos fenômenos físicos envolvidos durante a interação [10, 11].

A estabilidade é analisada neste trabalho do ponto de vista empírico, ou seja, através de várias simulações em computadores e de algumas experiências práticas. Os resultados mostraram uma boa estabilidade do sistema, quebrada apenas no uso de ganhos muito grandes e camadas limite muito pequenas.

## 4.7 Exemplo de Manipulação de Materiais Flexíveis

Como exemplo de manipulação de materiais flexíveis, considere a tarefa de encaixar uma chapa metálica utilizando o robô manipulador planar de dois graus de liberdade, apresentado na Figura 4.5. O estudo do problema a seguir discute os aspectos de projeto envolvidos.

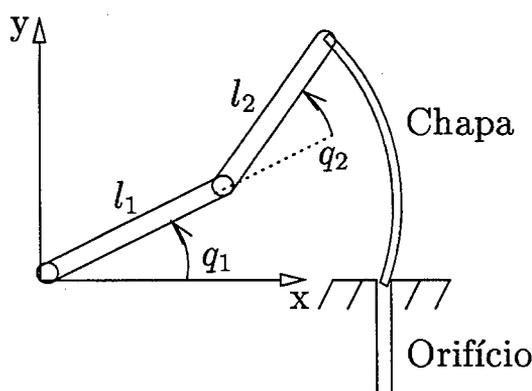


Figura 4.5: Robô Manipulador 2DOF inserindo uma chapa flexível

### 4.7.1 Descrição do Problema

O problema de encaixe apresentado na Figura 4.5, consiste em fazer com que o manipulador segure uma das extremidades da chapa metálica, que está numa posição qualquer do espaço de trabalho, e realizar o movimento para o encaixe sem permitir que a outra extremidade, que já está colocada na borda do orifício, solte-se.

A solução comum deste problema é apresentada em duas etapas: primeiramente, procura-se alinhar o efetuador final com a orientação do orifício sem permitir a perda do ponto de contato da chapa com o orifício, e então realiza-se o encaixe propriamente dito. É apresentado aqui o estudo da primeira etapa. Para facilitar a exemplificação é considerado que não existe momento na direção  $z$  no ponto de contato do efetuador final com a chapa [39].

A modelagem do sistema pode conter uma série de incertezas, como por exemplo, os comprimentos e massas dos elos do manipulador e da carga (chapa), os atritos das juntas do manipulador e da chapa com o orifício, etc. Desta forma, a utilização de controladores clássicos é limitada devido à pequena robustez com relação a variações paramétricas proporcionada ao sistema. Além de apresentar erros em regime permanente eles podem ter um comportamento instável. Por outro lado, o controlador à estrutura variável é bastante indicado para este tipo de sistema por garantir robustez na execução da tarefa.

Se o alinhamento for feito utilizando o controle puro de posição, não é garantida a robustez com relação às incertezas acima citadas. A utilização do controle de força, através de uma estrutura de controle híbrido de força e posição, facilita o problema de se conseguir o alinhamento desejado do efetuador final.

Para simulação numérica na Seção 4.8 a chapa metálica é modelada como sendo uma mola não linear de elasticidade variável, apresentada na Figura 4.6, cuja força é dada por:

$$F_e = \frac{\delta}{l_a} (K_{1m}(1 + |\delta_x|)) (1 - e^{-K_{2m}(l_m - l_a)}) \quad (4.7)$$

onde  $l_m$  é o comprimento da mola livre,  $K_{1m}$  e  $K_{2m}$  são parâmetros de elasticidade da mola.  $\delta$  é o vetor dado por:

$$\delta = \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} = [x - x_m] \quad (4.8)$$

onde  $x_m$  é a posição da extremidade fixa da mola. E por fim,  $l_a$  é o comprimento atual da mola que é dado por:

$$l_a = \sqrt{\delta_x^2 + \delta_y^2}. \quad (4.9)$$

Esta mola representa bem a chapa real, visto que através de dados experimentais [39] é verificado que a chapa apresenta uma resposta de rigidez exponencial, além desta ser variável de acordo com a deformação da chapa (função da posição  $x$ ).

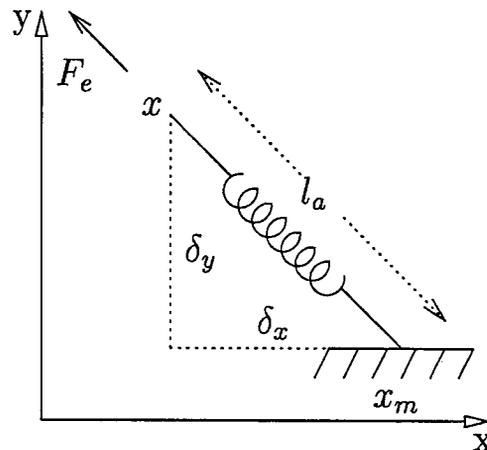


Figura 4.6: Mola Não Linear

Os valores dos parâmetros da mola utilizados para simulação na Seção 4.8 são:

$$x_m = \begin{bmatrix} 0,3 \\ 0,0 \end{bmatrix} \text{ m} ; \quad l_m = 0,3 \text{ m} ; \quad K_{1m} = 100 \text{ N m} ; \quad K_{2m} = 2,0 \text{ m}^{-1} \quad (4.10)$$

Com estes parâmetros, do ponto de vista da tarefa, a mola apresenta as características mostradas nas Figuras 4.7 e 4.8. Cada curva destes gráficos é obtida mantendo a posição  $y$  fixa (mostrada ao lado dos gráficos) enquanto a posição  $x$  é variada.

Uma vez descrito o problema, é necessária uma análise do ponto de vista de controle. Esta análise é abordada a seguir.

#### 4.7.2 Problema de Controle

O alinhamento, discutido na sub-seção anterior, deve ser feito levando o efetuator final para a posição de encaixe (direção do orifício) sem perder o contato da chapa com o orifício. Portanto, a direção  $y$  é controlada em força ( $f_{yd} = Cte_f$ ) para não perder o contato, e a direção  $x$  controlada em posição, para alcançar a posição desejada, que é

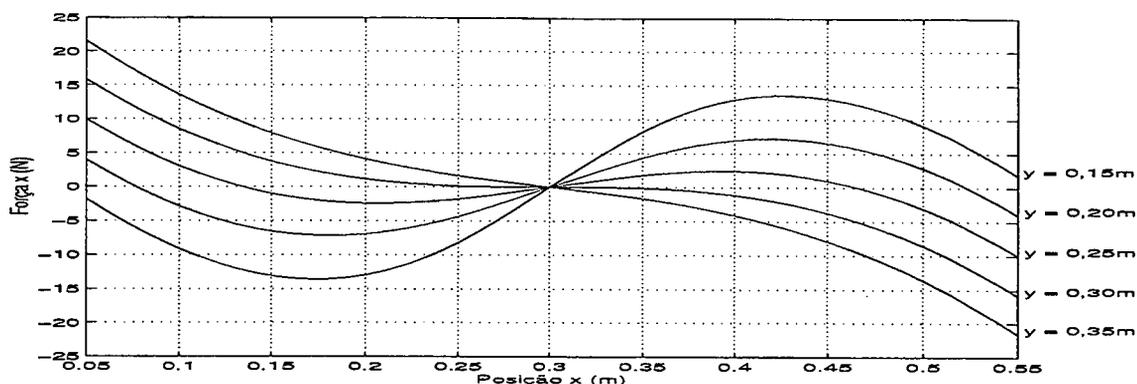


Figura 4.7: Componentes de Força x da Mola

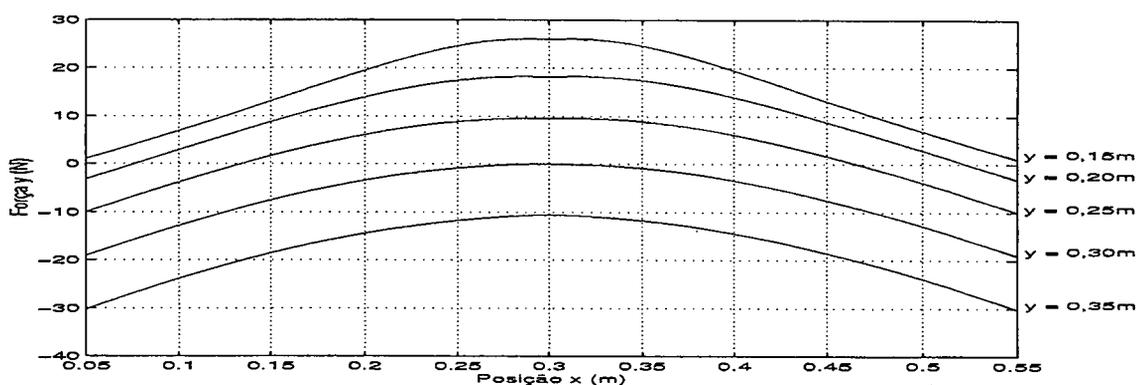


Figura 4.8: Componentes de Força y da Mola

atingida utilizando uma velocidade constante ( $\dot{p}_{xd} = Cte_v$ ). Desta forma, a matriz de seleção  $S$  (4.1) é:

$$S = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}. \quad (4.11)$$

Após o alinhamento passa-se para a segunda etapa da tarefa, a introdução da chapa no orifício. Esta etapa não é simulada neste trabalho. A transição da primeira etapa para a segunda é detectada através da posição  $p_x$  do efetuador final (posição desejada). Além disso, esta transição é sentida pela monitoração da velocidade na direção  $y$ . Quando a chapa estiver alinhada com o orifício, ocorre uma variação brusca da velocidade linear na direção  $y$ , pois a força imposta pelo manipulador faz com que a chapa comece a entrar no orifício. Uma vez detectada a transição para a segunda etapa da tarefa, uma nova matriz de seleção  $S$  deve ser utilizada, de acordo com as novas restrições artificiais.

As mudanças das restrições naturais são sempre detectadas observando a variável de força ou posição que não está sendo controlada. No exemplo acima, a velocidade

na direção  $y$  deve ser monitorada, já que na primeira etapa da tarefa esta direção é controlada em força.

O local do orifício da Figura 4.5 é  $(0,3 ; 0,0)$  m, o qual é definido pelo parâmetro  $x_m$  (4.10) da mola. Assim, a posição desejada do efetuador final é o ponto onde a sua componente  $x = 0,3$  m.

Por suposição, a posição alvo deve ser alcançada com velocidade constante na direção  $x$  de  $0,1$  m/s, e a força de referência na direção  $y$  de  $1,0$  N. A próxima seção apresenta varias simulações para teste do controlador.

## 4.8 Resultados de Simulação

Para as simulações numéricas do problema de manipulação de materiais flexíveis apresentado na seção anterior, é utilizado o modelo dinâmico do manipulador SCARA, obtido na Sub-seção 2.4.3.

Para uma melhor aproximação da realidade, é incluída na Equação (2.40) uma dinâmica de primeira ordem com ganho unitário e constante de tempo igual a  $5,0$  ms, representando o atuador<sup>3</sup>.

O Apêndice A contém os parâmetros do manipulador utilizado nas simulações. Os parâmetros da mola são os apresentados em (4.10). As condições iniciais em todas as simulações são:

$$x = \begin{bmatrix} 0,1 \\ 0,2 \end{bmatrix} \text{ m} \Leftrightarrow q = \begin{bmatrix} 0 \\ 2,2143 \end{bmatrix} \text{ rad} \quad \dot{q} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \frac{\text{rad}}{\text{s}}$$

Para verificar o funcionamento do controlador, primeiro é aplicado um *degrau* [20] para uma posição próxima ao orifício ( $x = 0,2$  m) e, após a estabilização (que se dá aproximadamente em  $1,0$  s), é feito o movimento para a posição de encaixe.

### 4.8.1 Parâmetros Nominais

Aqui é apresentada uma simulação sem variações paramétricas da carga manipulada, sem perturbações externas nem ruídos nos sinais medidos. Esta simulação serve para verificar o comportamento do controlador quando os parâmetros são bem conhecidos, e para a sintonia do controlador.

Os parâmetros dos sub-controladores foram obtidos empiricamente. Eles são apresentados abaixo:

<sup>3</sup>O Apêndice A contém as constantes de tempo dos motores do robô SCARA real.

$$K_f = 0,1 ; K_p = 8,0$$

$$u_{x0f} = 70 \text{ N m} ; u_{x0p} = 100 \text{ N m} ; \epsilon = 0,01 \quad (4.12)$$

As superfícies de deslizamento de força e posição, Figura 4.9, são rapidamente atingidas, tanto no começo da simulação (as condições iniciais não estavam sobre a superfície de deslizamento), bem como no instante de tempo igual a 1,0 s, onde a trajetória de posição apresenta uma descontinuidade de velocidade. O que comprova que as superfícies de deslizamento são alcançadas num tempo finito.

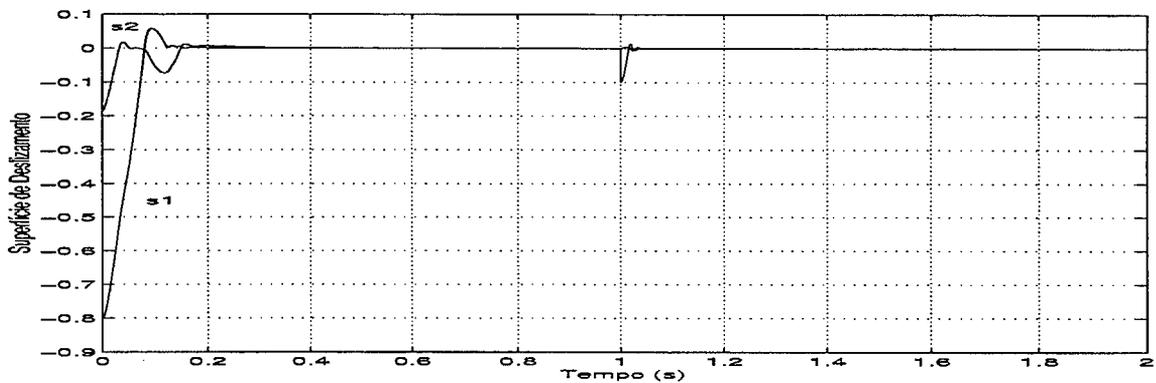


Figura 4.9: Proximidade das Superfícies ( $\sigma_p = s1$  ;  $\sigma_f = s2$ )

O esforço de controle, Figura 4.10, não apresentou *chattering* e manteve-se em níveis práticos aceitáveis (veja o Apêndice A), não apresentando picos excessivos.

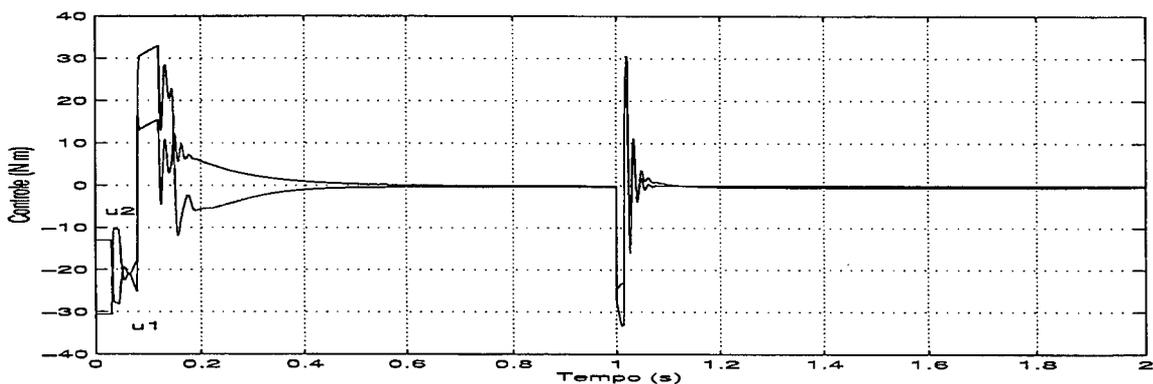


Figura 4.10: Sinais de Controle ( $u1$  ;  $u2$ )

As Figuras 4.11 e 4.12 apresentam os valores de forças e posições, respectivamente. As referências são atingidas de forma praticamente assintótica em torno de 0,7 s. A referência de posição foi seguida até o fim da simulação, já a força apresentou um

erro na mudança da referência de velocidade, porém ela tende assintoticamente para a referência (1, 0 N). Observando estas duas figuras verifica-se o comportamento não linear e variante das forças medidas.

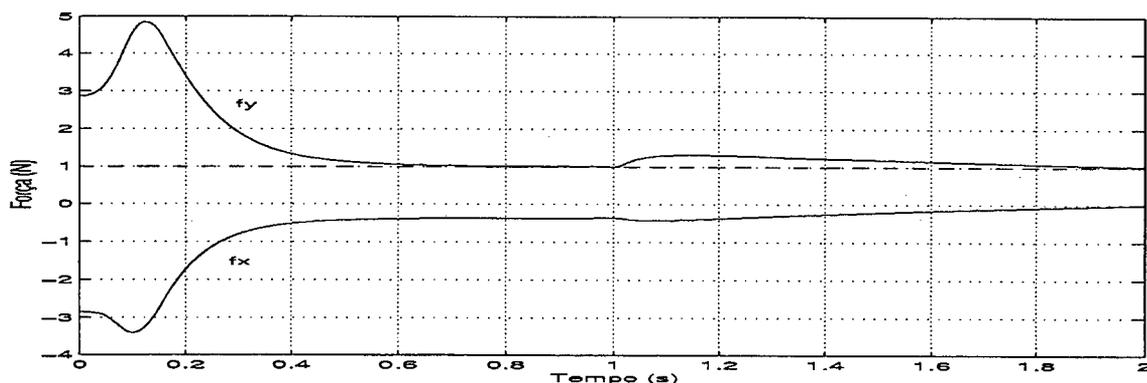


Figura 4.11: Forças Externas ( $F_x = f_x$  ;  $F_y = f_y$ )

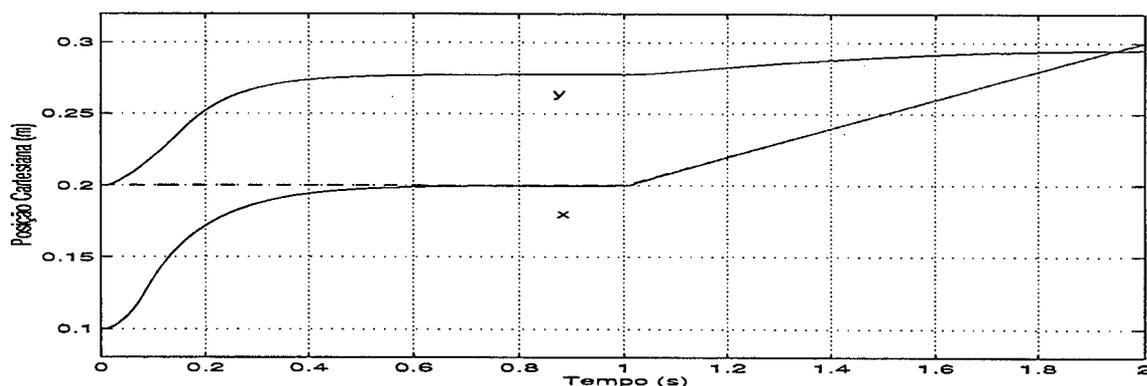


Figura 4.12: Posições no Espaço de Tarefa ( $x$  ;  $y$ )

A Figura 4.13 mostra o movimento do efetuador final, visto de cima do plano do manipulador. Nesta figura pode-se perceber mais claramente a trajetória da posição  $y$  para a regulação da força.

#### 4.8.2 Variações Paramétricas da Carga

Para testar a robustez do controlador com relação a variações paramétricas são feitas simulações com alteração do comprimento da chapa metálica. Estas alterações simulam variações paramétricas reais existentes em linhas de montagem, como erros de posição inicial das peças a serem encaixadas por exemplo.

As variações aqui introduzidas são de ordem maior do que as encontradas na prática, o comprimento da chapa é alterada para  $\pm 10\%$  do seu valor nominal. As alterações são desta grandeza para simular situações extremas de erros.

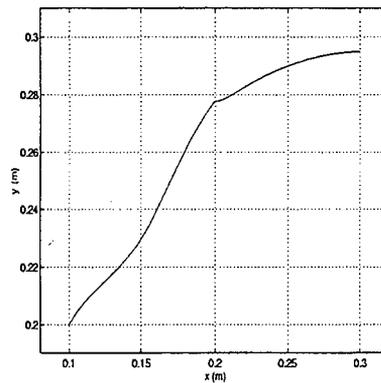


Figura 4.13: Posição do Efetuador Final

O controlador não é alterado para as novas situações, ele continua usando os parâmetros apresentados em (4.12).

### Comprimento Menor

Nesta simulação o comprimento da chapa metálica é menor que o seu valor nominal. O novo tamanho é:

$$l_m = 0,30 \text{ m} - 10\% = 0,27 \text{ m}$$

A Figura 4.14 apresenta o sinal de controle. Nesta figura fica mais clara a característica chaveada do controle.

A Figura 4.15 mostra as componentes da força externa. Devido à nova configuração, a componente de força inicial na direção  $y$  é negativa. Esta componente apresenta um grande *sobre-sinal* [20] devido ao controle de posição, pois no intervalo de tempo onde o sobre-sinal ocorre a posição  $x$  desejada ainda não foi atingida.

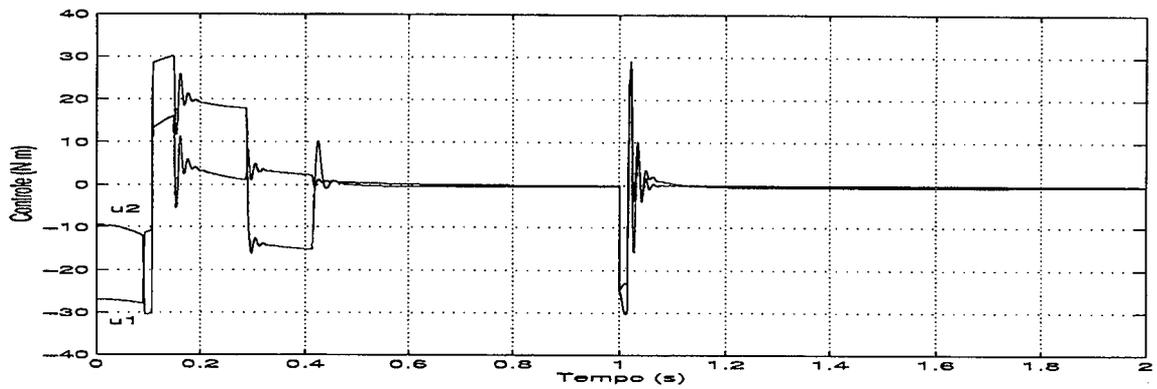
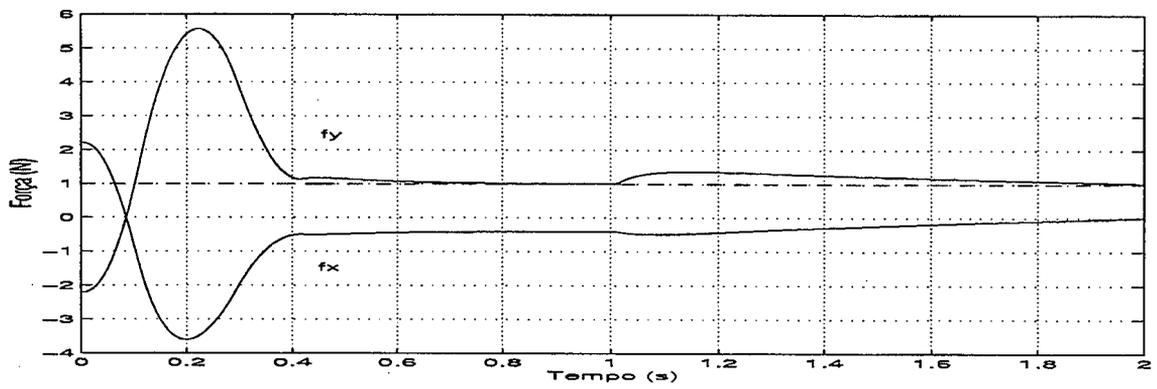
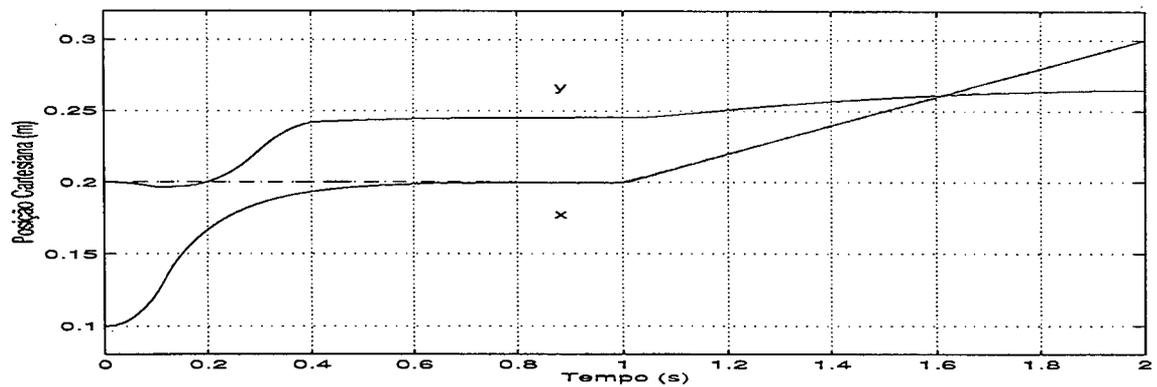
A Figura 4.16 mostra as posições  $x$  e  $y$ . A referência de posição continua sendo atingida em aproximadamente 0,7 s.

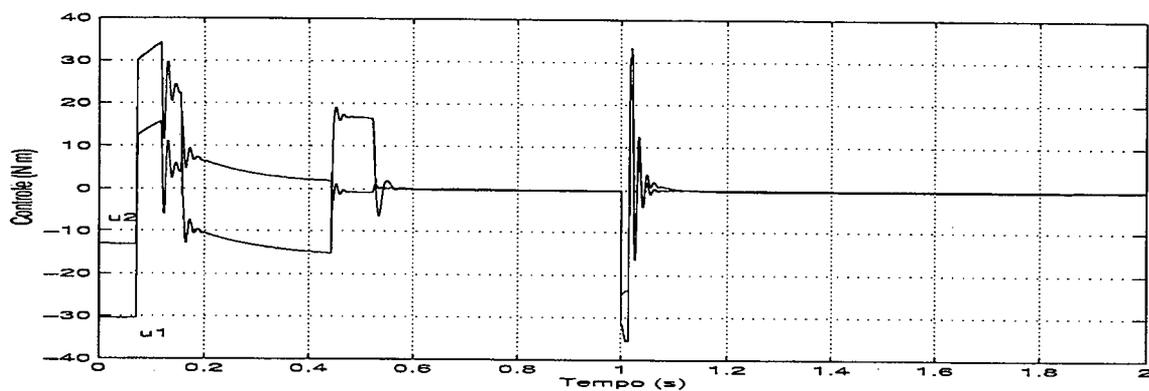
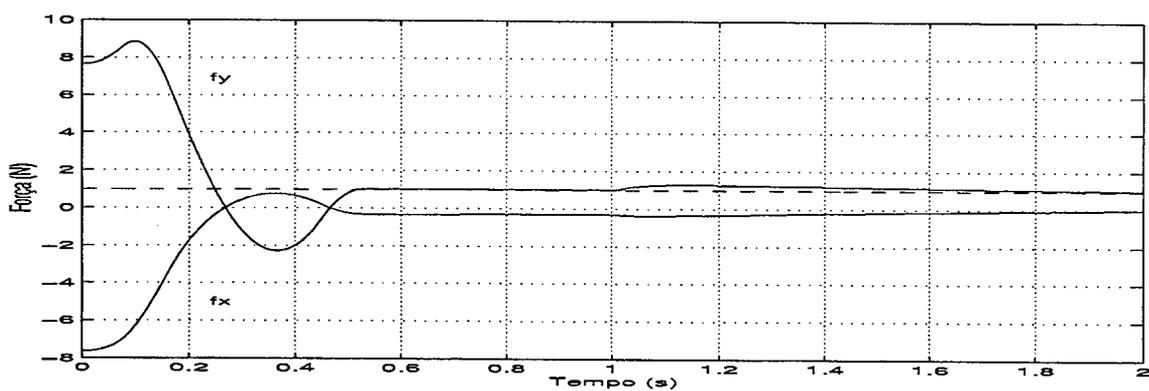
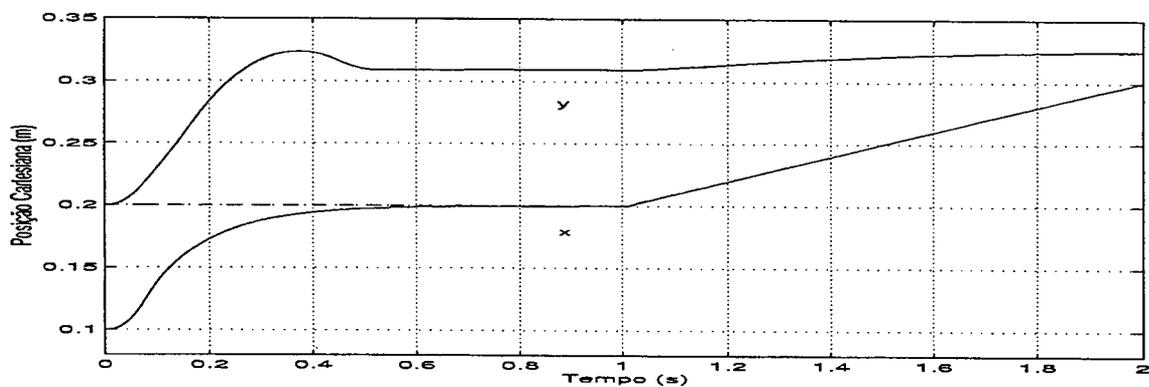
### Comprimento Maior

Nesta simulação o comprimento da chapa metálica é maior que o seu valor nominal. O novo tamanho é:

$$l_m = 0,30 \text{ m} + 10\% = 0,33 \text{ m}$$

O sinal de controle, Figura 4.17, continua bem comportado com relação à valores máximos e *chattering*.

Figura 4.14: Sinais de Controle ( $u_1$  ;  $u_2$ ) - Chapa MenorFigura 4.15: Forças Externas ( $F_x = f_x$  ;  $F_y = f_y$ ) - Chapa MenorFigura 4.16: Posições no Espaço de Tarefa ( $x$  ;  $y$ ) - Chapa Menor

Figura 4.17: Sinais de Controle ( $u_1$  ;  $u_2$ ) - Chapa MaiorFigura 4.18: Forças Externas ( $F_x = f_x$  ;  $F_y = f_y$ ) - Chapa MaiorFigura 4.19: Posições no Espaço de Tarefa ( $x$  ;  $y$ ) - Chapa Maior

Como pode ser visto na Figura 4.18, novamente a força controlada (componente  $y$ ) apresenta sobre-sinal, desta vez menor e negativo.

Esta simulação apresenta as maiores componentes de força. Mesmo assim, o controle de posição, Figura 4.19, continua com praticamente o mesmo comportamento. O controlador trata a componente de força  $x$  como uma perturbação externa, e a despreza.

### 4.8.3 Perturbações e Ruídos

Para testar a robustez do controlador com relação a perturbações externas e a ruídos do sensor de força, são feitas novas simulações. Assim como no estudo das variações paramétricas da carga, o controlador continua usando os mesmos parâmetros sintonizados na Sub-seção 4.8.1 para o valor nominal da chapa metálica.

A perturbação adicionada ao sinal de controle é composta de degraus. Os degraus têm duração de 0,2 s, e são de amplitude aleatória com o valor máximo em torno de 10% do valor máximo do esforço de controle. A perturbação aplicada é apresentada na Figura 4.20.

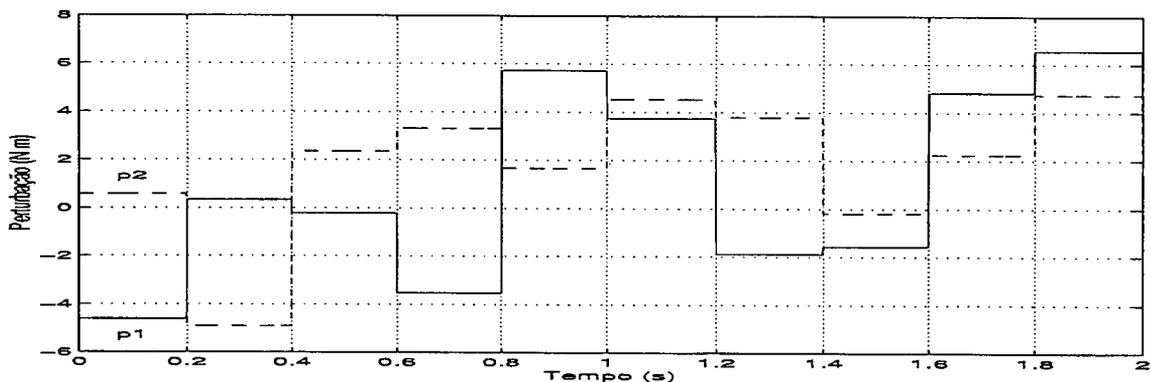
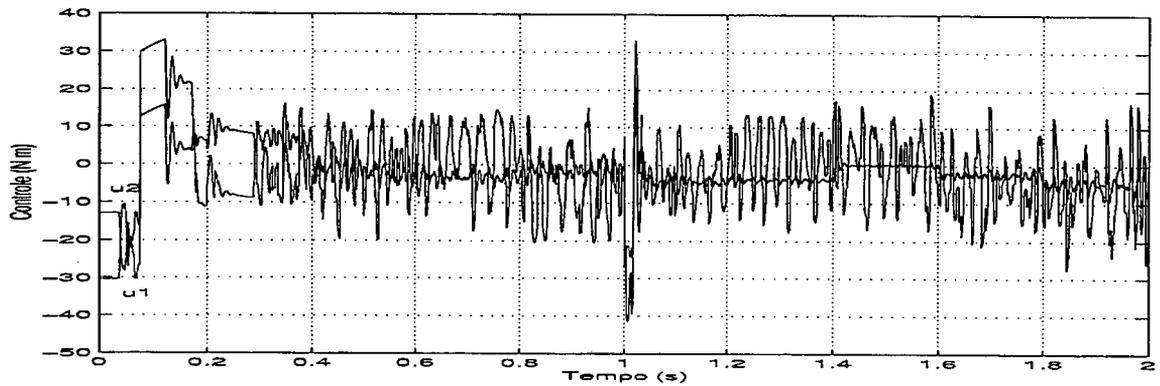
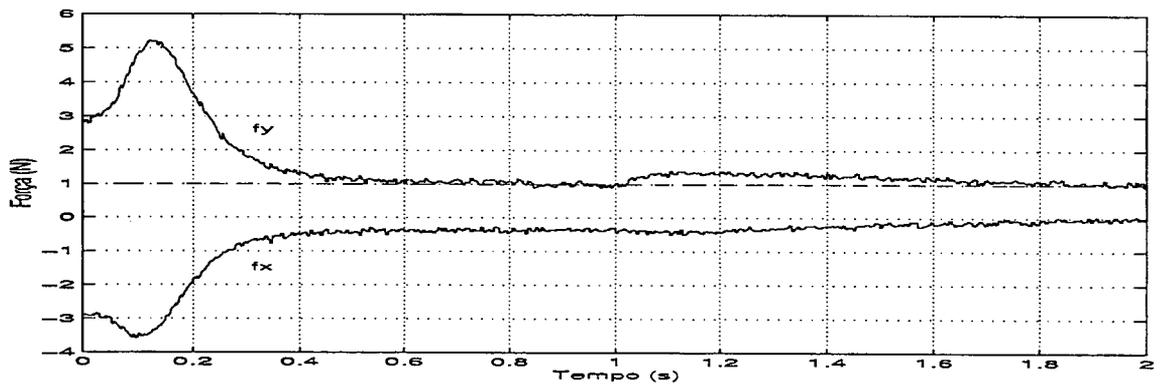
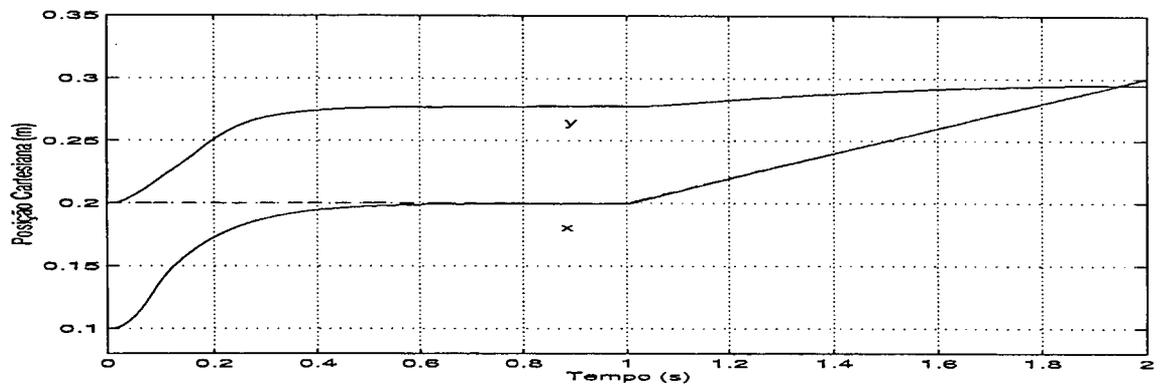


Figura 4.20: Perturbações (p1 ; p2)

O distúrbio adicionado ao sinal de força, simulando o ruído do sensor, é um *ruído branco* [19, 76]. Este ruído foi gerado com período de amostragem de 5,0 ms e amplitude máxima de 0,1 N. Na Figura 4.22 o ruído pode ser visto, já adicionado ao sinal de força medido.

Devido ao ruído, o sinal de controle, Figura 4.21, possui uma frequência de chaveamento maior que nas simulações anteriores. Além disso, para seguir as referências um esforço de controle maior é necessário.

Tanto o controle de força, Figura 4.22, como o controle de posição, Figura 4.23, continuam estáveis, mostrando a robustez do controlador com relação à perturbações e ruídos.

Figura 4.21: Sinais de Controle ( $u_1$  ;  $u_2$ ) - Com Perturbação e RuídoFigura 4.22: Forças Externas ( $F_x = f_x$  ;  $F_y = f_y$ ) - Com Perturbação e RuídoFigura 4.23: Posições no Espaço de Tarefa ( $x$  ;  $y$ ) - Com Perturbação e Ruído

## 4.9 Conclusões

O controlador proposto está diretamente relacionado às restrições impostas pela tarefa que se deseja executar, utilizando os princípios de projeto de um controlador híbrido clássico de força e posição para robôs manipuladores.

Tanto o sub-controlador de força como o de posição utilizam uma lei de controle à estrutura variável. Desta forma, foi obtido um controlador robusto sem a necessidade do conhecimento exato do modelo do manipulador nem do ambiente. O desempenho das variáveis controladas manteve-se dentro de um patamar bastante satisfatório apesar de grandes variações paramétricas da carga e da aplicação de perturbações e ruídos no sistema.

Este controlador não se restringe apenas à tarefa aqui exemplificada; ele pode ser utilizado em diversas tarefas de manufatura, observando que as considerações de especificação da tarefa utilizadas no projeto são de grande importância.

Foi apresentada uma análise da estabilidade dos sub-controladores, de força e de posição, separadamente no Capítulo 3. Estudos sobre a interação entre eles ainda são necessários para a análise da estabilidade do sistema completo. Apesar disto, as simulações numéricas sugerem resultados promissores para uma implementação prática do controlador híbrido de força e posição aqui apresentado.

# Capítulo 5

## Conclusões e Perspectivas

Neste capítulo são apresentadas as conclusões gerais sobre o trabalho e as perspectivas futuras.

### 5.1 Conclusões

Este trabalho teve como objetivo principal o estudo do controle simultâneo de força e posição robusto para robôs manipuladores interagindo com ambientes de elasticidade não linear.

Para o desenvolvimento do controlador foi necessário o conhecimento do modelo matemático do manipulador. Este modelo foi obtido, de forma detalhada e exemplificada, utilizando a formulação Lagrangeana.

A estratégia de controle utilizada para o controle simultâneo de força e posição foi o controlador híbrido. Esta é a estratégia mais utilizada atualmente para este tipo de controle. Neste esquema são informadas as posições e forças desejadas explicitamente no espaço cartesiano, o que facilita a especificação de tarefas.

Para garantir a robustez do controlador, necessária devido às incertezas inerentes às tarefas a serem executadas e às incertezas da própria planta a ser controlada, foi utilizada a teoria de controle à estrutura variável. Os sub-controladores, de força e de posição, do controlador híbrido foram implementados utilizando controladores de modos deslizantes.

Utilizando também a teoria de controle à estrutura variável, foi projetado um controlador de posição (e velocidade) para um manipulador planar de dois graus de liberdade no espaço de juntas. Este controlador foi simulado numericamente, e posteriormente implementado em bancada num robô industrial SCARA.

Os resultados da implementação real proporcionaram uma idéia das diferenças com

relação à simulação utilizando a dinâmica do robô modelada. Neste caso, onde as velocidades envolvidas são altas para existir forças dinâmicas de interação entre os elos, as dinâmicas não modeladas, como as flexibilidades por exemplo, tornam-se mais proeminentes, mostrando que o modelo do robô não é muito adequado para este tipo de estudo. Entretanto, para o estudo do controlador híbrido de força e posição este modelo proporciona uma boa representação do sistema real, uma vez que quando este controlador está sendo utilizado as velocidades do manipulador são baixas por se tratar de movimentos geometricamente restritos.

Através de simulações foi verificado que o controlador híbrido de força e posição apresenta um funcionamento satisfatório. Nas simulações foram impostas variações na carga manipulada, aplicadas perturbações no sinal de controle e incluídos sinais de ruído no sensor de força para testar a robustez e estabilidade do controlador. Estes testes simulam situações reais de manipulação de peças flexíveis, onde nem tudo é conhecido com grande exatidão. Em todas estas situações, o desempenho permanece praticamente inalterado, comprovando a propriedade de robustez do controlador utilizado.

A idéia inicial deste trabalho era a implementação do controlador num robô real, o que infelizmente não foi possível devido à algumas limitações. Sem dúvida a maior limitação para a não implementação do controlador foi o tempo disponível.

O robô SCARA usado para as implementações era uma novidade para o Laboratório de Robótica, de forma que até se conseguir um domínio do seu ambiente de desenvolvimento, da sua linguagem de programação (XOberon) e do seu funcionamento básico levou algum tempo. Os avanços neste sentido foram parcialmente documentados e discutidos entre os pesquisadores envolvidos em trabalhos relacionados com o robô; assim trabalhos futuros de implementação neste robô serão um pouco mais fáceis.

O sistema utilizado pelo robô é bastante singular. Ele utiliza uma linguagem de programação similar ao Pascal, com algumas particularidades. A interface homem/máquina não é semelhante ao Windows<sup>©</sup>, ao MSDOS<sup>©</sup> nem a sistemas operacionais do tipo Unix<sup>©</sup>. Esta interface utiliza um mouse de três botões que podem ser usados individualmente ou em conjunto para realizar diferentes funções, como abrir um arquivo texto ou executar um método de algum objeto.

Apesar da falta da implementação do controlador híbrido de força e posição num robô real, as simulações mostraram-se bastante satisfatórias. Além disso, a implementação do controlador de posição utilizando modos deslizantes no SCARA foi um passo importante para a comparação das simulações numéricas com a implementação real. A semelhança entre os resultados teóricos e práticos do controlador de posição encorajam a implementação do controlador híbrido num robô industrial. Assim, certamente o controlador projetado neste trabalho poderá ser utilizado em tarefas reais obtendo

resultados eficientes de produção.

## 5.2 Perspectivas

Uma das questões mais importantes, que foi abordada na Seção 4.6, trata-se da análise da estabilidade do sistema completo. Um estudo do acoplamento entre os sub-controladores é necessário para que a robustez do controlador seja garantida, possibilitando então, que o mesmo seja utilizado em aplicações reais.

Outra questão importante é a implementação do controlador aqui projetado num robô industrial, como o SCARA do Laboratório de Robótica da UFSC, por exemplo. Esta implementação não precisa, necessariamente, ser posterior à análise de estabilidade acima citada. A implementação do controlador provavelmente irá impor algumas alterações de ordem prática no algoritmo de controle, o que indicará pontos do projeto a serem estudados de forma mais aprofundada.

Como foi apresentado na Seção 4.7, a sintonia do controlador foi feita de forma empírica. O estudo de algoritmos de sintonia para o controlador aqui projetado proporcionaria uma melhora significativa no desempenho do controlador. Assim, uma sugestão de estudo são as sintonias automática e adaptativa do controlador híbrido.

Este trabalho abordou a teoria do controle híbrido de força e posição em nível de controle. O estudo da especificação de tarefas é extremamente importante para a utilização deste controlador em aplicações reais. A literatura atual não apresenta um estudo satisfatório da especificação de tarefas. Desta forma, o estudo do planejamento e especificação de tarefas carece de uma atenção maior, além de ser necessário para o uso de controladores híbridos de força e posição em robôs industriais.

# Apêndice A

## Parâmetros do Manipulador SCARA

Parâmetros do robô manipulador industrial SCARA (*Selective Compliance Assembly Robot Arm*) do Laboratório de Robótica da Universidade Federal de Santa Catarina.

### A.1 Dimensões e Massas

Elo $i$	Comprimento $l_i$ (m)	Centro de Massa $r_i$ (m)	Massa $m_i$ (Kg)	Inércia $I_i$ (Kg m <sup>2</sup> )
1	0,25	0,118	11,4	0,23
2	0,25	0,116	19,5	0,16
3	0,0	0,0	2,0	0,10
4	0,0	0,0	1,5	0,10

Tabela A.1: Dimensões e Massas do Manipulador SCARA

### A.2 Valores Limites

Elo $i$	Posição Máxima $q_{max}$	Posição Mínima $q_{min}$	Velocidade Máxima $\dot{q}_{max}$	Torque Máximo $\tau_{max}$ (N m)
1	2,25 rad	-2,25 rad	3,00 rad/s	333,0
2	1,90 rad	-1,90 rad	3,00 rad/s	157,0
3	0,40 m	0,14 m	0,88 m/s	6,975
4	2,50 rad	-2,50 rad	20,0 rad/s	16,74

Tabela A.2: Valores Limites do Manipulador SCARA

## A.3 Constantes de Tempo dos Motores

### A.3.1 Motor bls55

As juntas 2, 3 e 4 utilizam este motor. As suas constantes de tempo são:

- mecânica: 1,7 ms
- elétrica: 1,4 ms

### A.3.2 Motor bls72

A junta 1 utiliza este motor. As suas constantes de tempo são:

- mecânica: 2,1 ms
- elétrica: 1,1 ms

# Apêndice B

## Códigos Fonte

### B.1 Módulo MathMen

Abaixo são apresentados alguns procedimentos matemáticos utilizados na lei de controle.

```
PROCEDURE abs* (num: REAL): REAL;
VAR
  saida: REAL;
BEGIN
  IF num >= 0.0 THEN
    saida := num
  ELSE
    saida := -num;
  END;
  RETURN saida;
END abs;
```

```
PROCEDURE sign* (num: REAL): REAL;
VAR
  saida: REAL;
BEGIN
  IF num = 0.0 THEN
    saida := 0.0
  ELSE
    IF num > 0.0 THEN
      saida := +1.0
    ELSE
      saida := -1.0;
    END;
  END;
  RETURN saida;
END sign;
```

```
PROCEDURE signum* (epsilon, num: REAL): REAL;
VAR
  saida: REAL;
BEGIN
  IF abs(num) < epsilon THEN
    saida := num / epsilon
  ELSE
    saida := sign(num);
  END;
  RETURN saida;
END signum ;
```

## B.2 Módulo StateCtrl

Abaixo são apresentados a definição do objeto *StateCtrl* e alguns *métodos* [50, 57] modificados para se utilizar a nova lei de modos deslizantes projetada. Observe que os procedimentos não estão completos, apenas as suas partes modificadas estão aqui.

```
MODULE StateCtrl;

TYPE
  StateCtrlPara* = POINTER TO RECORD
    kPos*, kVel*, acc*, acc2, deltaS, deltaS2, limspeed,
    systemdelay*, systemdelayDiv2, i*, iLimit*, inertia*,
    ineDivclock*, integ, clock*: REAL;
    (* Atributos incluídos no objeto *)
    typeCtrl*: LONGINT;
    kSM*, u0SM*, epsilon*: REAL;
  END;

  StateCtrl* = POINTER TO StateCtrlDesc;

  StateCtrlDesc* = RECORD (Base.ObjDesc)
    stateCtrlPara*: StateCtrlPara;
    rs, rp: RF.RampType;
    rampTime*: REAL;
    bsFilter*: FilterBSn0.Filter;
  END;
```

```
PROCEDURE (c: StateCtrl) Assign* (o: Base.Object; name: ARRAY OF CHAR):
    BOOLEAN;
VAR
    value: Base.Value;
BEGIN
    IF (o=NIL) THEN RETURN FALSE
        IF name = 'kPos' THEN
            c.stateCtrlPara.kPos := SHORT(value.x);
            :
        ELSIF name = 'kSM'
            THEN c.stateCtrlPara.kSM := SHORT(value.x);
        ELSIF name = 'u0SM'
            THEN c.stateCtrlPara.u0SM := SHORT(value.x);
        ELSIF name = 'epsilon'
            THEN c.stateCtrlPara.epsilon := SHORT(value.x);
            :
        ELSE
            RETURN c.Assign^(o, name);
        END;
    ELSIF o IS FilterBSn0.Filter THEN
        IF name = 'bsFilter' THEN
            c.bsFilter:=o(FilterBSn0.Filter)
        END
    ELSE
        RETURN c.Assign^(o, name)
    END;
RETURN TRUE;
END Assign;
```

```
PROCEDURE AttrMsg (obj: StateCtrl; VAR M: O.AttrMsg);
BEGIN
  IF M.id = O.get THEN
    IF M.name = "Gen" THEN
      M.class:=O.String; M.s:='StateCtrl.NewStateCtrl'; M.res:=0
      :
    ELSIF M.name = "kSM" THEN
      M.class:=O.Real; M.x:=obj.stateCtrlPara.kSM; M.res:=0
    ELSIF M.name = "u0SM" THEN
      M.class:=O.Real; M.x:=obj.stateCtrlPara.u0SM; M.res:=0
    ELSIF M.name = "epsilon" THEN
      M.class:=O.Real; M.x:=obj.stateCtrlPara.epsilon; M.res:=0
      :
    ELSIF M.name= 'bsFilter' THEN
      M.class:= O.String;
      IF obj.bsFilter#NIL THEN
        O.GetName(obj.bsFilter.lib.dict, obj.bsFilter.ref, M .s);
        M.res:= 0
      ELSE
        M.s:= 'not configurated'; M.res:= 0
      END;
    ELSE
      Base.Handler(obj, M)
    END
  ELSIF M.id = O.set THEN
    Base.Handler(obj, M)
  ELSIF M.id = O.enum THEN
    :
    M.Enum("kSM"); M.Enum("u0SM"); M.Enum("epsilon");
    :
    Base.Handler(obj, M)
  END;
END AttrMsg;
```

```
(* Novo procedimento criado para mudança do tipo (lei) de controle. *)
```

```
PROCEDURE (c: StateCtrl) ChangeTC* (typeCtrl: LONGINT);  
BEGIN  
  IF (typeCtrl = Configura.TCpd) OR (typeCtrl = Configura.TCsm) THEN  
    XOK.NotInterruptible;  
    c.stateCtrlPara.typeCtrl := typeCtrl;  
    XOK.Interruptible;  
  END;  
END ChangeTC;
```

```
(* Nova lei de controle. *)
```

```
PROCEDURE AlgoSM (p: StateCtrlPara; istpos, sollpos, istspeed,  
  sollspeed: REAL): REAL;  
VAR  
  forceout, s: REAL;  
BEGIN  
  s := ( p.kSM * ( istpos - sollpos ) ) + ( istspeed - sollspeed );  
  forceout := -p.uOSM * MathMen.signum(p.epsilon,s);  
  RETURN forceout;  
END AlgoSM;
```

```
(* Novo algoritmo principal de controle. *)  
  
PROCEDURE (c: StateCtrl) Algo* (istpos, sollpos, istspeed,  
    sollspeed: REAL): REAL;  
VAR  
    forceout: REAL;  
    p: StateCtrlPara;  
BEGIN  
    p := c.stateCtrlPara;  
    IF Configura.usarampa THEN  
        sollspeed := RF.RampFilter(sollspeed, c.rs);  
        sollpos := RF.RampFilter(sollpos, c.rp);  
    END;  
    CASE p.typeCtrl OF  
        Configura.TCpd:  
            forceout := AlgoPD(p,istpos, sollpos, istspeed, sollspeed);  
        |Configura.TCsm:  
            forceout := AlgoSM(p,istpos, sollpos, istspeed, sollspeed);  
    END;  
    IF c.bsFilter#NIL THEN  
        forceout := c.bsFilter.Algo(forceout)  
    END;  
    RETURN forceout;  
END Algo;
```

```
PROCEDURE NewStateCtrl*;
  VAR c: StateCtrl; p: StateCtrlPara;
BEGIN
  NEW(c); c.handle :=Handler; NEW(p);
      :
  c.stateCtrlPara.typeCtrl := Configura.TCpd ;  (* PD original *)
  c.stateCtrlPara.kSM := 0.0;
  c.stateCtrlPara.u0SM := 0.0;
  c.stateCtrlPara.epsilon := 0.0;
      :
  O.NewObj :=c;
END NewStateCtrl;
```

Outros procedimentos foram criados e modificados. Eles não fazem parte deste apêndice porque não estão diretamente relacionados com o controlador, são procedimentos de definições globais, configuração, plotagem, etc.

# Referências Bibliográficas

- [1] G. Ambrosino, G. Celetano, and F. Garofalo. Variable structure model reference adaptative control systems. *International Journal of Control*, 39(6):1339–1349, 1984.
- [2] H. Asada and J-J. E. Slotine. *Robot Analysis and Control*. John Wiley & Sons, Inc., New York - USA, 1986.
- [3] E. Bailey and A. Arapostathis. Simple sliding mode control scheme applied to robot manipulators. *International Journal of Control*, 45(4):1197–1209, 1987.
- [4] T. D. Burton. *Introduction to Dynamics System Analysis*. McGraw-Hill, New York - USA, 1994.
- [5] C. Canudas, B. Siciliano, and G. Bastin. *Theory of Robot Control*. Springer-Verlag, New York - USA, 1996.
- [6] C. T. Chen. *Linear System Theory and Design*. Holt, Rinehart and Winston, New York - USA, 1984.
- [7] C. T. Chen. *Analog and Digital Control System Design: Transfer-Function, State Space and Algebraic Methods*. Saunders College Publishing, New York - USA, 1993.
- [8] T.-L. Chern, C.-W. Chuang, and R.-L. Jiang. Design of discrete integral variable structure control systems and application to a brushless dc motor control. *Automatica*, 32(5):773–779, 1996.
- [9] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano. Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy. *The International Journal of Robotics Research*, 4(10):410–425, August 1991.

- [10] S. Chiaverini and L. Sciavicco. The parallel approach to force/position control of robotics manipulators. *IEEE Transactions on Robotics and Automation*, 9(4):361–373, August 1993.
- [11] S. Chiaverini, B. Siciliano, and L. Villani. Force/position regulation of compliant robot manipulators. *IEEE Transactions on Automatic Control*, 39(3):647–652, March 1994.
- [12] L. S. Coelho and A. A. R. Coelho. Projeto e sintonia evolutiva de controladores pd e pid com aplicação em um manipulador robótico. No *Congresso Brasileiro de Automática 98 - CBA98*, volume I, páginas 381–386, Uberlândia - Brasil, Setembro 1998.
- [13] J. J. Craig. *Introduction to Robotics: Mechanics & Control*. Addison-Wesley Publishing Company, USA, 1986.
- [14] R. A. DeCarlo, S. H. Zak, and G. P. Matthews. Variable structure control of nonlinear multivariable systems: A tutorial. *Proceedings of the IEEE*, 76(3):212–232, March 1988.
- [15] R. A. DeCarlo, S. H. Zak, and S. V. Drakunov. *Variable Structure, Sliding-Mode Controller Design*. *The Control Handbook*, pages 941–951. CRC Press, Inc., New York - USA, January 1996.
- [16] A. DeLuca and C. Manes. Modeling of robots in contact with a dynamic environment. *IEEE Transactions on Robotics and Automation*, 10(4):542–548, August 1994.
- [17] J. Denavit and R. S. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *ASME Journal of Applied Mechanics*, 22:215–221, June 1955.
- [18] J. DeSchutter and H. Bruyninckx. *Force Control of Robot Manipulators*. *The Control Handbook*, pages 1351–1358. CRC Press, Inc., New York - USA, 1996.
- [19] E. O. Doebelin. *Measurement Systems: Application and Design*. McGraw-Hill, New York - USA, 1990.
- [20] R. C. Dorf and R. H. Bishop. *Modern Control Systems*. Addison-Wesley Publishing Company, New York - USA, 1995.
- [21] R. El-Khazali and R. DeCarlo. Output feedback variable structure control design. *Automatica*, 31(6):805–816, 1995.

- [22] G. Ferretti, C. Maffezzoni, and P. Rocco. Decoupling force and motion control in industrial robots. *Control Engineering Practice*, 1(6):1019–1027, 1993.
- [23] G. Ferretti, G. Magnani, and P. Rocco. Toward the implementation of hybrid position/force control in industrial robots. *IEEE Transactions on Robotics and Automation*, 13(6):838–845, December 1993.
- [24] A. F. Filippov. Differential equations with discontinuous right hand sides. *Am. Math. Soc. Transl.*, 42:199–231, 1964.
- [25] G. R. Fowles. *Analytical Mechanics*. Saunders College Publishing, New York - USA, 1998.
- [26] L.-C. Fu and T.-L. Liao. Globally stable robust tracking of nonlinear systems using variable structure control and with an application to a robotic manipulator. *IEEE Transactions on Automatic Control*, 35(12):1345–1350, December 1990.
- [27] W. Gao, Y. Wang, and A. Homaifa. Discrete-time variable structure control systems. *IEEE Transactions on Industrial Electronics*, 42(2):117–122, April 1995.
- [28] S. C. P. Gomes. Modelagem de atritos internos às articulações de robôs-manipuladores. No *COBEM-CIDIM/95*, Rio de Janeiro - Brasil, 1995.
- [29] D. M. Gorinevsky, A. M. Formalsky, and A. Yu. Schneider. *Force Control of Robotic Systems*. McGraw-Hill, New York - USA, 1997.
- [30] R. Guenther and E. R. DePieri. A simulação de robôs em contato com o meio. No *Workshop de Robótica Inteligente*, páginas 195–206, Brasília - Brasil, Agosto 1997.
- [31] M. Hamerlain, M. Belhocine, and K. Bouyoucef. Sliding mode control for a robot scara. In *The 4<sup>th</sup> IFAC Symposium on Advances in Control Education*, pages 153–157, Istanbul - Turkey, July 1997.
- [32] N. Hogan. Impedance control: An approach to manipulation; part i - theory; part ii - implementation; part iii - applications. *ASME - Journal of Dynamic Systems, Measurement, and Control*, 107(1):1–24, March 1985.
- [33] H. N. Iordanou and B. W. Surgenor. Experimental evaluation of the robustness of discrete sliding mode control versus linear quadratic control. *IEEE Transactions on Control Systems Technology*, 5(2):254–260, March 1997.

- [34] R. G. Jacquot. *Modern Digital Control Systems*. Marcel Dekker Inc., New York - USA, 1995.
- [35] O. Kaynak and A. Denker. Discrete-time sliding mode control in the presence of system uncertainty. *International Journal of Control*, 57(5):1177–1189, 1993.
- [36] H. K. Khalil. *Nonlinear System*. Prentice Hall, New York - USA, 2nd edition, 1996.
- [37] J-S. Kim, K. Suzuki, A. Konno, and M. Uchiyama. Force control of constrained flexible manipulators. In *International Conference on Robotics and Automation*, pages 635–641, Minneapolis - Minnesota, April 1996.
- [38] R. D. Klafter, T. A. Chmielewski, and M. Negin. *Robotic Engineering: An Integrated Approach*. Prentice Hall, New York - USA, 1989.
- [39] W. Kraus Jr. *The Robotic Manipulation of Flexible Materials: A Hybrid Position/Force Approach*. PhD thesis, Australian National University, Canberra - Australia, March 1997.
- [40] C-Y. Kuo and S-P. Wang. Robust position control of robotic manipulator in cartesian coordinates. *IEEE Transactions on Robotics and Automation*, 7(5):653–659, October 1991.
- [41] C. M. Kwan. Hybrid force/position control for manipulators with motor dynamics using a sliding-adaptative approach. *IEEE Transactions on Automatic Control*, 40(5):963–968, May 1995.
- [42] C. C. S. Leal. Estabilidade no controle de forças em robôs manipuladores. Dissertação de mestrado, Pós-Graduação em Engenharia Elétrica - Universidade Federal de Santa Catarina, Florianópolis - Brasil, Fevereiro 1998.
- [43] F. L. Lewis, C. T. Abdallah, and D. M. Dawson. *Control of Robot Manipulators*. Macmillan Publishing Company, New York - USA, 1993.
- [44] Y. Li, K. C. Ng, D. J. Murray-Smith, G. J. Gray, and K. Sharman. Genetic algorithm automated approach to the design of sliding mode control systems. *International Journal of Control*, 63(4):721–739, 1996.
- [45] L. G. C. O. Lopes. Controle a estrutura variável aplicado a um atuador hidráulico. Dissertação de mestrado, Pós-Graduação em Engenharia Elétrica - Universidade Federal de Santa Catarina, Florianópolis - Brasil, Julho 1997.

- [46] J. Y. S. Luh. Conventional controller design for industrial robots - a tutorial. *IEEE Transactions on Systems, Man, and Cybernetics*, 13(3):298-316, May 1983.
- [47] M. T. Mason. Compliance and force control for computer controlled manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-11(6):418-432, June 1981.
- [48] N. H. McClamroch and D. Wang. Feedback stabilization and tracking of constrained robots. *IEEE Transactions on Automatic Control*, 33(5):419-426, May 1988.
- [49] M. F. Mendes, W. Kraus Jr., and E. R. DePieri. Automated manufacturing with flexible parts: A control perspective. In *5<sup>th</sup> IFAC Workshop on Intelligent Manufacturing Systems - IMS'98*, Gramado - Brazil, November 1998.
- [50] H. Mossenbock. *Object-Oriented Programming in Oberon-2*. Springer-Verlag, Berlin Heidelberg - Germany, 1993.
- [51] R. Ortega and M. W. Spong. Adaptive motion control of rigid robots: a tutorial. *Automatica*, 25(6):877-888, 1989.
- [52] C. L. Phillips and T. H. Nagle. *Digital Control System Analysis and Design*. Prentice Hall, New Jersey - USA, 3rd edition, 1995.
- [53] M. H. Raibert and J. J. Craig. Hybrid position/force control of manipulators. *ASME - Journal of Dynamic Systems, Measurement, and Control*, 102:126-133, June 1981.
- [54] A. R. G. Ramírez. Controle de posição de robôs manipuladores com transmissões flexíveis. Dissertação de mestrado, Pós-Graduação em Engenharia Elétrica - Universidade Federal de Santa Catarina, Florianópolis - Brasil, Março 1998.
- [55] R. Reginatto and E. R. DePieri. Some nonlinear robust control design approaches. No *Congresso Brasileiro de Automática 98 - CBA98*, volume IV, páginas 1443-1450, Uberlândia - Brasil, Setembro 1998.
- [56] M. Reiser. *The Oberon System: user guide and programmer's manual*. Addison-Wesley Publishing Company, New York - USA, 1991.
- [57] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen. *Object-Oriented Modeling and Design*. Prentice Hall, New Jersey - USA, 1991.

- [58] J. M. Schimmels and M. A. Peshkin. Admittance matrix design for force-guided assembly. *IEEE Transactions on Robotics and Automation*, 8(2):213–227, April 1992.
- [59] J. M. Schimmels and M. A. Peshkin. Force-assembly with friction. *IEEE Transactions on Robotics and Automation*, 10(4):465–479, August 1994.
- [60] J. M. Selig and P. R. McAree. A simple approach to invariant hybrid control. In *International Conference on Robotics and Automation*, pages 2238–2245, Minneapolis - Minnesota, April 1996.
- [61] B. Siciliano and L. Villani. A passivity-based approach to force regulation and motion control of robot manipulators. *Automatica*, 32(3):443–447, 1996.
- [62] C. H. Silveira, L. S. Coelho, and M. F. M. Campos. The use of genetic algorithms for the evaluation of inverse kinematics of manipulators. No 1<sup>o</sup> Congresso Brasileiro de Redes Neurais, Itajubá - Brasil, 1994.
- [63] H. Sira-Ramirez. Non-linear discrete variable structure systems in quasi-sliding mode. *International Journal of Control*, 54(5):1171–1187, 1991.
- [64] J. J. Slotine and S. S. Sastry. Tracking control of non-linear systems using sliding surfaces, with application to robot manipulators. *International Journal of Control*, 38(2):465–492, 1983.
- [65] J.-J. E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, New Jersey - USA, 1991.
- [66] P. Song and A. Goldenberg. Fundamental principles of design of position and force controller for robot manipulators. In *International Conference on Robotics and Automation*, pages 2246–2251, Minneapolis - Minnesota, April 1996.
- [67] M. W. Spong and M. Vidyasagar. *Robot Dynamics and Control*. John Wiley & Sons, Inc., New York - USA, 1989.
- [68] M. W. Spong, J. DeSchutter, H. Bruyninckx, and J. T.-Y. wen. *Control of Robots and Manipulators. The Control Handbook*, pages 1339–1351. CRC Press, Inc., New York - USA, 1996.
- [69] V. I. Utkin. Variable structure systems with sliding modes. *IEEE Transactions on Automatic Control*, AC-22(2):212–222, April 1977.

- [70] V. I. Utkin. Variable structure systems: Present and future. *Automation and Remote Control*, 44(9):1105–1120, September 1983.
- [71] V. I. Utkin. *Sliding Modes in Control Optimization*. Springer-Verlag, New York - USA, 1992.
- [72] M. Vidyasagar. *Nonlinear Systems Analysis*. Prentice Hall, New Jersey - USA, 2nd edition, 1993.
- [73] R. Volpe and P. Khosla. A theoretical and experimental investigation of explicit force control strategies for manipulators. *IEEE Transactions on Automatic Control*, 38(11):1634–1650, November 1993.
- [74] M. Vukobratovic and N. Kircanski. An approach to adaptative control of robotic manipulators. *Automatica*, 21(6):639–647, 1985.
- [75] D. Wang and N. H. McClamroch. Position and force control for constrained manipulator motion: Lyapunov's direct method. *IEEE Transactions on Robotics and Automation*, 9(3):308–313, June 1993.
- [76] P. E. Wellstead and M. B. Zarrop. *Self-Tuning Systems: Control and Signal Processing*. John Wiley & Sons, Inc., Chichester - England, 1991.
- [77] J. T. Wen and S. Murphy. Stability analysis of position and force control for robot arms. *IEEE Transactions on Automatic Control*, 36(3):365–371, March 1991.
- [78] D. E. Whitney. Force feedback control of manipulator fine motions. *Journal of Dynamic Systems, Measurement, and Control*, pages 91–97, June 1977.
- [79] D. E. Whitney. Historical perspective and state of the art in robot force control. *The International Journal of Robotics Research*, 6(1):3–14, Spring 1987.
- [80] W. A. Wolovich. *Automatic Control Systems: Basic Analysis and Design*. Saunders College Publishing, New York - USA, 1994.
- [81] T. Yoshikawa. *Foundations of Robotics: Analysis and Control*. The MIT Press, Cambridge, Massachusetts, 1990.
- [82] R. R. Y. Zhen and A. A. Goldenberg. Asymptotic robust constrained robot motion control. In *International Conference on Robotics and Automation*, pages 3428–3433, Minneapolis - Minnesota, April 1996.

- 
- [83] Y. F. Zheng and Y. Fan. Robot force sensor interacting with environments. *IEEE Transactions on Robotics and Automation*, 7(1):156–164, February 1991.