

Universidade Federal de Santa Catarina
Programa de Pós-Graduação em Engenharia Elétrica

**Análise e Síntese de Controladores de
Força-Posição de Robôs Manipuladores:
Aspectos Teóricos e Experimentais**

Tese submetida à
Universidade Federal de Santa Catarina
como parte dos requisitos para a
obtenção do grau de **Doutor em Engenharia Elétrica**

Francisco Javier Triveño Vargas

Florianópolis, Maio de 2005

Análise e Síntese de Controladores de Força-Posição de Robôs Manipuladores: Aspectos Teóricos e Experimentais.

Francisco Javier Triveño Vargas

Esta tese foi julgada adequada para a obtenção do título de **Doutor em Engenharia Elétrica**, Área de concentração **Automação e Sistemas**, e aprovada em sua forma final pelo programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.

Prof. Edson Roberto De Pieri, Dr.

Orientador

Prof. Alexandre Trofino Neto, Dr.

Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora

Prof. Edson Roberto De Pieri, Dr.

Orientador

Prof. Eugênio de Bona Castelan, Dr.

Co-orientador

Prof. Luiz Góes, Dr. - ITA

Prof. Silas do Amaral, Dr. - FEJ/UFSC

Prof. Raul Guenther, Dr. - UFSC

Prof. Ubirajara Franco Moreno, Dr. - UFSC

*A Deus, pelo dom da vida,
que é a fonte da sabedoria.*

*A meus pais Francisco e Rosa,
pela sua integridade e sua nobreza.*

Agradecimentos

A meus pais, Francisco e Rosa, que ainda longe, deram-me seu amor, carinho, compreensão, fortaleza e todo seu apoio para terminar este trabalho.

A Amanda, minha Madeleine e minha Maïte por seu amor e por me dar suas horas para finalizar este trabalho.

A meus orientadores, os professores Edson e Eugênio, pela confiança e por todos os conhecimentos transmitidos, profissional e pessoalmente.

Aos professores Goes, Silas do Amaral, Guenther e Ubirajara pelas suas valiosas observações para o melhoramento do presente trabalho.

A meus colegas Rodrigo Sumar, Rodrigo Goitia e Bismark, pela sua amizade.

Ao senhor Jaime Salvatierra pela sua valiosa ajuda e compreensão.

Ao Marcelo e Wilson, pela sua ajuda constante e sincera.

Ao Carlos Marengo e meus colegas de trabalho, por ter permitido meu amadurecimento profissional numa das melhores empresas do Brasil.

A minha universidade na Bolívia por ter-me oferecido um formação de alto nível.

À UFSC, à CNPQ e à EMBRAER, pelo suporte material e financeiro.

Finalmente agradeço a todos aqueles que de uma ou outra forma fizeram parte deste trabalho.

Resumo da Tese apresentada à *UFSC* como parte dos requisitos necessários para obtenção do grau de Doutor em Engenharia Elétrica.

**Análise e Síntese de Controladores de Força-Posição de Robôs Manipuladores:
Aspectos Teóricos e Experimentais**

Francisco Javier Triveño Vargas

Maio/2005

Orientador: Edson Roberto De Pieri, Dr.

Área de Concentração: Controle, Automação e Informática Industrial.

Palavras-chave: Robótica industrial, controle na estrutura de dois graus de liberdade, controle a estrutura variável, controle nebuloso, compensação de atrito, controle de força.

Número de páginas: 175

Na presente tese é feito o desenvolvimento teórico de estratégias de controle de força e posição para robôs manipuladores contemplando aspectos de robustez e compensação de atrito nas juntas. Este desenvolvimento é feito a partir da modelagem de De Luca e Manes. As estratégias de controle correspondem ao controle de Dois Graus de Liberdade (*TDOF*), o Controle a Estrutura Variável (*VSC*) e o controle Nebuloso (*FLC*). A primeira estratégia corresponde a uma estratégia linear, onde é aplicada a técnica do *LQR* para a determinação dos ganhos do controlador. Paralelamente a partir do modelo linear utilizado no projeto dos controladores *TDOF* é feita a identificação dos parâmetros estáticos e dinâmicos para a compensação do atrito empregando a modelagem de *LuGre*. O controlador *VSC* é projetado baseado no modelo não linear do robô e considerando um sistema criticamente amortecido. O controlador *FLC* é projetado em forma similar ao controle *VSC* devido à disposição de suas regras para a posição e a velocidade. O aspecto de estabilidade na estratégia linear é abordado considerando a localização dos pólos e a função de sensibilidade que engloba os requisitos de margem de ganho e margem de fase. Já nas técnicas não lineares o tratamento de estabilidade é feito considerando funções de *Lyapunov* e sua derivada temporal. A análise e projeto teóricos são complementados por testes de simulação. Em seguida é realizada a implementação prática das leis de controle desenvolvidas. Diferentes tarefas são utilizadas tais como corte de material e seguimento de trajetórias de posição e força. Estas tarefas são realizadas em um robô de pesquisa, do tipo *SCARA* com quatro graus de liberdade, dotado de sensor de força e com arquitetura de controle aberta, permitindo a implementação de diferentes tipos de leis de controle.

Abstract of Thesis presented to UFSC as a partial fulfillment of the requirements for the degree of
Doctor in Electrical Engineering.

**Analysis and Synthesis of Force-Position Controllers of Manipulator Robots:
Theoretical and Experimental Aspects**

Francisco Javier Triveño Vargas

May/2005

Advisor: Edson Roberto de Pieri, Dr.

Area of Concentration: Control, Automation and Industrial Informatics

Keywords: Industrial robotics, two degree of freedom control, variable structure control, fuzzy logic control, friction compensation, force control.

Number of Pages: 175

In the present thesis is done the theoretical development of strategies for force and position control of manipulator robots considering aspects as robustness and friction compensation in the joints. This development is done considering the De Luca and Manes modelling. Control strategies correspond to Two Degree of Freedom controller (*TDOF*), Variable Structure Control (*VSC*) and Fuzzy Logic controller (*FLC*). First controller corresponds to linear strategy, where *LQR* technique is used to determine the optimal gains of controller. At the same time the linear model used in the project of *TDOF* controllers is used to identify static and dynamic parameters for friction compensation using the *LuGre* model. *VSC* controller is projected based on the non linear model of the robot and considering a critically damped system. *FLC* controller is projected in similar form as *VSC* due to rules disposition for position and speed error. The aspect of stability in the linear strategy is treated considering the poles location and sensitivity function that includes gain margin and phase margin requirements. Already in the not linear techniques the treatment of the stability is made considering *Lyapunov* functions and its derivative. The theoretical analysis and project are complemented by simulation tests. After that, practical implementation of developed control laws is done. Different tasks are used such as cut of material and tracking reference for force and position. These tasks were done in a research *SCARA* robot, with four degrees of freedom with force sensor and with open architecture of control. This architecture allows implementing different types of control laws.

Resumen de la tesis presentada a la UFSC como parte de los requisitos necesarios para obtener el grado de Doctor em Ingeniería Eléctrica.

Control de Fuerza-Posición de Robots Manipuladores: Proyecto de Controladores y Resultados Experimentales

Francisco Javier Triveño Vargas

Mayo/2005

Orientador: Edson Roberto De Pieri, Dr.

Area de concentración: Control y Automatización Industrial

Palabras claves: Robótica industrial, controladores de dos grados de libertad, Control a estructura variable, control difuso, compensación de fricción, control de fuerza.

Número de paginas: 175

En la presente tesis es presentado el desarrollo teórico de estrategias de control de fuerza e posición para robots manipuladores considerando aspectos de robustez e compensación de fricción en sus articulaciones. Este desarrollo es realizado a partir de modelo matemático de De Luca e Manes. Las estrategias de control corresponden al control de dos grados de libertad *TDOF*, el control a estructura variable *VSC* y el control difuso *FLC*. La primera estrategia corresponde a una estrategia lineal, donde es aplicada la técnica de *LQR* para el calculo de las ganancias del controlador. En forma paralela, el modelo lineal determinado para esta técnica, es utilizado para identificar los parámetros estáticos y dinámicos utilizados para compensación de fricción con el modelo de *LuGre*. El controlador *VSC* es proyectado utilizando el modelo no lineal del robot y considerando un sistema críticamente amortiguado. La técnica *FLC* es proyectada en forma similar al control *VSC* debido a la disposición de sus reglas para la posición y velocidad. Los aspectos de estabilidad en la estrategia lineal son abordados considerando la localización de los polos y la función de sensibilidad que engloba los requisitos de margen de ganancia y margen de fase. Ya en las técnicas no lineales el tratamiento de la estabilidad es realizada considerando funciones de *Lyapunov* y su derivada temporal. El análisis y proyecto teórico es complementado con testes de simulación. En seguida es realizada la implementación practica de las leyes de control desarrolladas. Diferentes tareas son utilizadas tales como el corte de material y el seguimiento de trayectorias de posición y fuerza. Estas tareas son realizadas en un robot de pesquisa del tipo *SCARA* con cuatro grados de libertad, el mismo tiene acoplado un sensor de fuerza y tiene la arquitectura de control abierta lo que permite la implementación de diferentes leyes de control.

Sumário

1	Introdução	1
1.1	O Problema de Controle	1
1.1.1	O problema do atrito	3
1.2	Revisão Bibliográfica	3
1.3	Objetivos e Contribuições da Tese	5
1.4	Organização e Conteúdo	6
2	Modelagem Matemática	7
2.1	Justificativa	7
2.2	Modelagem Cinemática e Dinâmica	9
2.2.1	Modelagem cinemática do manipulador	9
2.2.2	Modelagem cinemática do meio	11
2.2.3	Propriedades dos contatos cinemáticos e/ou dinâmicos	12
2.2.4	Modelo dinâmico do sistema manipulador-meio	15
2.2.5	Modelo dinâmico compacto	18
2.3	Controle do Sistema Robô-Meio	19
2.4	Controle do Robô Submetido a Restrições Puramente Cinemáticas	20
2.4.1	Robô rígido submetido a restrições cinemáticas	20
2.4.2	Robô rígido livre de restrições	21
2.5	Controle do Robô Submetido a Restrições Puramente Dinâmicas	22
2.5.1	Controle da força na direção dinâmica	22
2.5.2	Controle de movimento na direção dinâmica	23
2.6	Conclusões do Capítulo	24
3	Modelagem e Compensação do Atrito	25
3.1	Características do Atrito	25
3.2	Modelagem do Atrito	27
3.2.1	Modelos estáticos	27

3.3	Modelos Dinâmicos	29
3.3.1	Modelo de <i>Dahl</i>	30
3.3.2	Modelo de <i>Bliman/Sorine</i>	30
3.3.3	Modelo de <i>LuGre</i> (<i>Lund e Grenoble</i>)	31
3.4	Compensação do Atrito	32
3.5	Determinação dos Parâmetros do Modelo de <i>LuGre</i>	34
3.5.1	Determinação dos parâmetros estáticos	34
3.5.2	Determinação dos parâmetros dinâmicos	35
3.6	Conclusões do Capítulo	36
4	Estratégias de Controle	37
4.1	Controlador de Dois Graus de Liberdade TDOF	37
4.1.1	Estrutura de controle	38
4.1.2	Definições e propriedades	39
4.1.3	Objetivos de malha	40
4.1.4	Objetivos de resposta	42
4.1.5	Projeto de controle ótimo com erro nulo em regime	43
4.1.6	<i>Algoritmo LQR com Erro Nulo em Regime</i>	45
4.2	Controle a Estrutura Variável VSC	46
4.2.1	Descrição geral do sistema	47
4.2.2	Síntese de controle	48
4.2.3	Análise de robustez	50
4.2.4	Uso da camada limite	51
4.3	Controle Nebuloso <i>FLC</i>	53
4.3.1	Estrutura do <i>FLC</i>	53
4.4	Controladores <i>FLC</i> Baseados no <i>VSC</i>	56
4.4.1	Normalização do <i>FLC</i>	56
4.5	Conclusões do Capítulo	60
5	Bancada Experimental	61
5.1	Laboratório de Robótica	62
5.1.1	Robô industrial SCARA	62
5.1.2	Sensor de força	64
5.1.3	Ferramentas	65
5.2	O Sistema Operacional e a Linguagem de Programação <i>XO/2</i>	65
5.2.1	Escalonador de tempo real	65
5.2.2	Administrador de memória	66
5.2.3	Modularizado	66

5.3	Módulos de Controle	66
5.3.1	Módulo Robot	66
5.3.2	Módulo ScaraRobot	67
5.3.3	Módulo Jr3	67
5.3.4	Módulo JobPlanner	68
5.3.5	Módulo JobElements	68
5.3.6	Módulo PathPlanner	68
5.3.7	Módulo Main	68
5.4	Conclusões do Capítulo	68
6	Controle de Posição: Resultados Experimentais	70
6.1	Validação do Modelo Dinâmico	70
6.2	<i>TDOF</i> Aspectos de Projeto e Resultados	73
6.3	<i>VSC</i> Aspectos de Projeto e Resultados	80
6.4	<i>FLC</i> Aspectos de Projeto e Resultados	83
6.5	Identificação Experimental dos Parâmetros de <i>LuGre</i>	87
6.5.1	Identificação dos parâmetros estáticos	87
6.5.2	Identificação dos parâmetros dinâmicos	89
6.6	Implementação Experimental	90
6.6.1	Resultados experimentais	92
6.6.2	Análise dos resultados	97
6.7	Conclusões do Capítulo	98
7	Controle de Força-Posição: Resultados Experimentais	99
7.1	Apresentação do Problema de Força	99
7.2	Definição dos Experimentos	100
7.2.1	Obtenção das variáveis $\dot{s}_{K,m}$, $s_{K,m}$ e $\lambda_{R,m}$	101
7.3	O Problema da Força de Contato	102
7.3.1	Resultados experimentais	103
7.4	Implementação dos Controladores Desenvolvidos	106
7.4.1	Controle a estrutura variável	107
7.5	O Problema da Filtragem nas Medições de Força	108
7.5.1	Filtro exponencial de primeira ordem	108
7.5.2	Filtro exponencial de segunda ordem	109
7.5.3	Seleção da constante τ_f	109
7.6	Obtenção dos Resultados	110
7.6.1	Resultados experimentais da filtragem	111
7.6.2	Resultados experimentais usando controle <i>PI</i>	112

7.6.3	Resultados experimentais usando controle <i>VSC</i>	114
7.6.4	Resultados experimentais usando controle de força e posição	115
7.7	Conclusões do Capítulo	117
8	Conclusões e Perspectivas	118
8.1	Conclusões	118
8.2	Perspectivas	121
A	Modelagem Cinemática	122
A.1	Sistema de Coordenadas	122
A.2	Cinemática Direta	123
A.2.1	Jacobiano analítico	123
A.2.2	Derivada do Jacobiano analítico	124
A.2.3	Inversa do Jacobiano analítico	125
A.3	Cinemática Inversa	125
B	Controle a Estrutura Variável - Conceitos Básicos	127
B.1	Controle a Estrutura Variável	127
B.1.1	Definições	127
B.1.2	Projeto da Superfície de Deslizamento	129
B.1.3	Projeto do Controlador	130
C	Controle Nebuloso - Conceitos Básicos	132
C.1	Conjuntos Nebulosos	132
C.2	T-norma e T-conorma	133
C.3	Raciocínio Nebuloso	134
C.3.1	Regras nebulosas tipo Mamdani	135
C.4	Desnebulização	135
D	Controle Digital	137
D.1	Controle <i>PID</i> Digital	137
D.2	Controle <i>TDOF</i> Digital	138
E	Implementação de Controladores	140

Lista de Figuras

1.1	Componentes de um robô industrial	2
1.2	Controle híbrido	3
2.1	Tarefa de perfuração	7
2.2	Variáveis de junta e elos num robô de 5 graus de liberdade	10
2.3	Contato puramente cinemático	14
2.4	Contato cinemático e/ou dinâmico	14
2.5	Controle baseado na dinâmica inversa	20
3.1	Força de quebra	26
3.2	Retardo na variação da velocidade e a força de atrito	26
3.3	Deslocamento de pré-deslizamento	27
3.4	Atrito de Coulomb e atrito viscoso	28
3.5	Atrito de Coulomb, viscoso, estático e a velocidade de <i>Stribeck</i>	29
3.6	Observador de atrito baseado no erro de posição	32
3.7	Observador de atrito baseado no erro de velocidade	32
3.8	Observador de atrito baseado na velocidade desejada	33
3.9	Diagrama de controle para a obtenção dos parâmetros estáticos	34
4.1	Controlador de Dois Graus de Liberdade - <i>TDOF</i>	38
4.2	Diagrama de Bode de $ L(j\omega) $ para desempenho robusto	43
4.3	Modo deslizante: (a) Ideal (b) Com <i>chattering</i>	51
4.4	Camada limite	52
4.5	Funções <i>sgn</i> , <i>sat</i> e <i>tanh</i>	52
4.6	Estrutura básica de um controlador nebuloso	54
4.7	Características do controle <i>VSC</i>	56
4.8	Controlador nebuloso normalizado	57
4.9	Funções de pertinência	58
4.10	Superfícies dos controladores nebulosos	59

5.1	Malha para o controle de força e posição	61
5.2	Robô industrial <i>SCARA</i> de 4 graus de liberdade	62
5.3	Arquitetura aberta de controle	63
5.4	Sensor de força	64
5.5	Arquitetura do sensor de força	64
5.6	Ferramentas	65
5.7	Módulos de controle e procedimentos (atuais e implementados)	67
6.1	Estrutura de controle PD	71
6.2	Respostas simulada e experimental nas direções x e y	72
6.3	Respostas simulada e experimental nas direções z e θ	73
6.4	Pólos ótimos	74
6.5	Resposta ao degrau e esforço de controle	75
6.6	Função de sensibilidade, complementar de sensibilidade e ganho de malha	76
6.7	Trajectoria e velocidade desejadas eixo x	77
6.8	Trajectoria e velocidade desejadas eixo y	77
6.9	Trajectoria e velocidade desejadas eixo z	77
6.10	Trajectoria e velocidade desejadas eixo θ	78
6.11	Erro e esforço de controle no eixo x	78
6.12	Erro e esforço de controle no eixo y	79
6.13	Erro e esforço de controle no eixo z	79
6.14	Erro e esforço de controle no eixo θ	79
6.15	Requisitos de projeto para os controladores VSC e FLC	81
6.16	Erro e esforço de controle no eixo x	82
6.17	Erro e esforço de controle no eixo y	82
6.18	Erro e esforço de controle no eixo z	82
6.19	Erro e esforço de controle no eixo θ	83
6.20	Erro e esforço de controle no eixo x	85
6.21	Erro e esforço de controle no eixo y	85
6.22	Erro e esforço de controle no eixo z	85
6.23	Erro e esforço de controle no eixo θ	86
6.24	Perfil de velocidade e torque para a junta 0 (perfil médio de -0.01 [rad/s])	87
6.25	Perfil de velocidade e torque para a junta 0 (perfil médio de 0.02 [rad/s])	88
6.26	Perfil de velocidade e torque para a junta 0 (perfil médio de 0.03 [rad/s])	88
6.27	Perfil de velocidade e torque para a junta 0 (perfil médio de 0.1 [rad/s])	88
6.28	Atrito estático para as juntas 0 e 1	89
6.29	Início de movimento para a junta 0	90

6.30	Posição e velocidade na junta 0	91
6.31	Posição e velocidade na junta 1	91
6.32	Posição e velocidade na junta 2	92
6.33	Posição e velocidade na junta 3	92
6.34	Erro e torque estimado de atrito na junta 0 - TDOF	93
6.35	Erro e torque estimado de atrito na junta 1 - TDOF	93
6.36	Erro e torque estimado de atrito na junta 2 - TDOF	93
6.37	Erro e torque estimado de atrito na junta 3 - TDOF	94
6.38	Erro e torque estimado de atrito nas junta 0 - VSC	94
6.39	Erro e torque estimado de atrito nas junta 1 - VSC	95
6.40	Erro e torque estimado de atrito na junta 2 - VSC	95
6.41	Erro e torque estimado de atrito na junta 3 - VSC	95
6.42	Erro e torque estimado de atrito na junta 0 - FLC	96
6.43	Erro e torque estimado de atrito nas junta 1 - FLC	96
6.44	Erro e torque estimado de atrito na juntas 2 - FLC	96
6.45	Erro e torque estimado de atrito na junta 3 - FLC	97
7.1	Tarefa de corte	100
7.2	Posição e força no eixo x - madeira	104
7.3	Posição e força no eixo y - madeira	104
7.4	Posição e força no eixo z - madeira	104
7.5	Posição e força no eixo θ - madeira	105
7.6	Posição e força no eixo x - borracha	105
7.7	Posição e força no eixo y - borracha	106
7.8	Posição e força no eixo z - borracha	106
7.9	Posição e força no eixo θ - borracha	106
7.10	Diagramas de bode para os três filtros	110
7.11	Forças medidas nas direções x e y	111
7.12	Força e momento em torno de z e θ	112
7.13	Posição e força na direção x - controle PI	113
7.14	Posição e força na direção y - controle PI	113
7.15	Posição e força na z - controle PI	113
7.16	Posição angular e momento na direção θ - controle PI	114
7.17	Posição e força na direção x - controle VSC	115
7.18	Posição e força na direção y - controle VSC	115
7.19	Posição e força na direção z - controle VSC	115
7.20	Posição angular e momento em θ - controle VSC	116

7.21	Posição e torque - Controle VSC	116
7.22	Posição e torque - Controle VSC	117
7.23	Posição e torque - Controle VSC	117
A.1	Sistemas de coordenadas do robô <i>SCARA</i>	122
B.1	Superfícies de deslizamento	129
C.1	Formas típicas das funções de pertinência	133
C.2	Raciocínio nebuloso	136
C.3	Desnebulização pelo método da gravidade	136
D.1	Controlador <i>TDOF</i> digital	138

Lista de Tabelas

3.1	Valores de velocidade e torque	35
4.1	Base de regras	57
4.2	Conjuntos nebulosos	57
5.1	Limites do robô industrial <i>SCARA</i>	63
5.2	Limites do sensor de força	64
6.1	Parâmetros do robô industrial <i>SCARA</i>	71
6.2	Valores dos pólos ótimos	74
6.3	Pólos selecionados	75
6.4	Ganhos e polinômios obtidos	75
6.5	Ganhos dos controlador <i>TDOF</i> - caso digital	76
6.6	Parâmetros de projeto <i>VSC</i>	81
6.7	Parâmetros de projeto para e e \dot{e}	84
6.8	Parâmetros de projeto para τ	84
6.9	Erros para velocidade de $0.01[\frac{m}{s} \frac{rad}{s}]$	86
6.10	Erros para velocidade $0.2[\frac{m}{s} \frac{rad}{s}]$	86
6.11	Coefficientes de atrito	90
6.12	Indicadores do <i>TDOF</i>	97
6.13	Indicadores do <i>VSC</i>	97
6.14	Indicadores do <i>FLC</i>	97
7.1	Ganhos dos controladores <i>PI</i>	103
7.2	Parâmetros dos filtros exponenciais de primeira e segunda ordem	111
7.3	Ganhos dos controladores <i>PI</i>	112
7.4	Ganhos dos controladores <i>VSC</i>	114
7.5	Erros na direção z dos controladores <i>PID</i> e <i>VSC</i>	114
7.6	Ganhos dos controladores <i>VSC</i>	116

A.1	Parâmetros de <i>Denavit – Hartenberg</i>	123
-----	---	-----

Capítulo 1

Introdução

Grandes sonhos, grandes conquistas.

FJTV.

Os robôs industriais estão diretamente associados ao conceito atual de automação industrial. Esses dispositivos realizam os mais diversos tipos de tarefas, incorporando aspectos de sensoramento, controle e comunicação. Características tais como a redução de custos, incremento na produtividade, melhoria da qualidade e, principalmente, a eficiência na realização de tarefas perigosas e altamente repetitivas são algumas de suas qualidades.

A configuração e o controle dos robôs industriais é realizada por uma interface com o usuário, uma linguagem de programação e o controle propriamente dito. O sistema de controle lê as saídas fornecidas pelos diferentes sensores, provê as respectivas entradas (torques, momentos) e estabelece a comunicação entre o usuário e seus periféricos. Alguns desses componentes tais como o armário de controle, a unidade de energia e a interface de programação são ilustrados na figura 1.1.

1.1 O Problema de Controle

Pelas suas características, as tarefas executadas pelos robôs podem ser divididas em dois tipos: aquelas nas quais não existe interação com o meio ambiente (*controle de posição*), como a movimentação de materiais e soldagem, entre outras, e aquelas nas quais o robô interage com o meio ambiente (*controle de força*).

Várias pesquisas em *controle de posição* de robôs manipuladores têm sido feitas visando o planejamento de trajetórias para a prevenção de colisões, controle robusto, controle adaptativo e controle inteligente (Jaritz e Spong, 1996). Muitos métodos de controle têm sido propostos e testados na prática. Os mais comumente empregados são os chamados controladores clássicos *PD* e *PID* por apresentarem simplicidade em seu projeto (Barrientos *et al.*, 2000), (Ramirez *et al.*, 2000b). Tais controladores não são capazes, em geral, de garantir um desempenho adequado em termos de resposta,

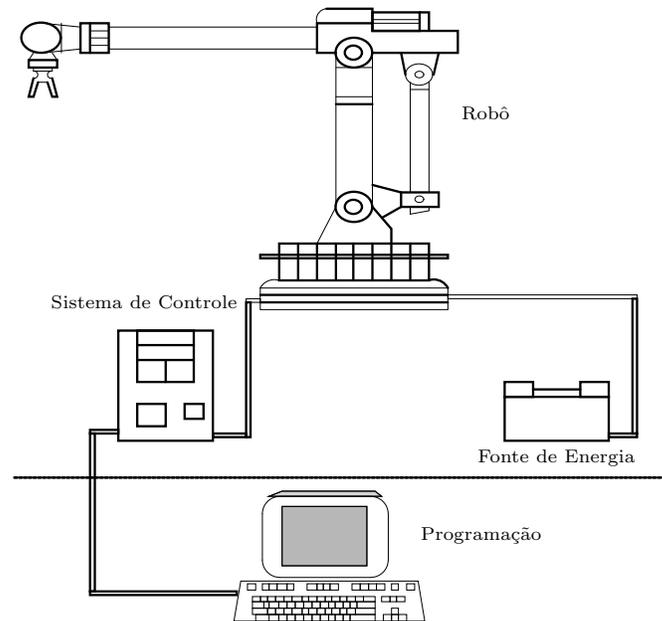


Figura 1.1: Componentes de um robô industrial

apresentando erros de seguimento nas trajetórias, devido à presença de perturbações, dinâmicas não modeladas, ruídos e atrito. Em vista disso, controladores avançados (lineares e não-lineares) têm sido propostos para suprir as deficiências destes controladores clássicos: controladores baseados na dinâmica inversa (Sciavicco e Siciliano, 2000), controladores na estrutura de Dois Graus de Liberdade (Wolovich, 1995), controladores a Estrutura Variável (Utkin, 1992) (Young *et al.*, 1999), controladores adaptativos (Villani *et al.*, 2000), controladores *nebulosos* (Sandri e Correia, 1999) e controladores baseados em redes neurais (Kiguchi e Fukuda, 1997), (Passold, 2003).

Já o *controle de força* (necessário quando ha interação entre o manipulador e o ambiente) é crucial para a execução bem sucedida de um número crescente de tarefas nas quais é necessário além de controlar a posição e a velocidade, controlar também a força resultante da interação entre o efetuador final e a superfície ou o objeto que está sendo manipulado. Exemplos típicos são as tarefas de polimento, corte, perfuração e montagem (inserção de peças), além de outras aplicações como a tele-operação, as tarefas de serviço e os experimentos na area medica (Sheridan, 1987), (Dario *et al.*, 1996).

Ao contrário daquilo que ocorre com o controle de posição, no controle de força, devido à dificuldade de dispor de robôs equipados com sensores de força e com arquitetura aberta de controle, a grande maioria dos métodos de controle propostos são teóricos e os principais testes são realizados em nível de simulação. Poucos são os trabalhos disponíveis (Villani *et al.*, 2000), (Stefano Chiaverini e Villani, 1999), que realizando experimentações, colocam em evidência problemas do tipo atrito e perturbações externas tais como ocorrem na prática. Pelas razões expostas, a implementação prática tanto do controle de força quanto do controle de posição se constitui na principal motivação desta Tese de Doutorado.

1.1.1 O problema do atrito

Independentemente do tipo de controlador utilizado ou da tarefa pertencer ao controle de força ou posição, a resposta do sistema é consideravelmente influenciada pela existência do atrito (Olsson e Aström, 1996). Essa característica é inerente aos vários tipos de robôs industriais e sua compensação torna-se necessária para melhorar a eficiência dos algoritmos de controle. O problema da compensação de atrito em robôs manipuladores tem sido largamente estudado (Årmstrong *et al.*, 1994), (Gomes, 1995), (Olsson *et al.*, 1997), (Ramirez, 2003). Nesta tese, será dada ênfase ao problema de compensação de atrito usando-se o modelo de *LuGre* (de Wit *et al.*, 1995) que tem sido bastante utilizado em diferentes aplicações por ser bastante completo, já que o mesmo inclui as características estáticas e dinâmicas do atrito.

1.2 Revisão Bibliográfica

As primeiras técnicas propostas para a análise e implementação do controle de força e posição foram o Controle *Híbrido* (Raibert e Craig, 1981) e o Controle de *Impedância* (Hogan, 1985).

O Controle *Híbrido* baseado no trabalho de Mason (1981) decompõe o espaço de trabalho (espaço cartesiano) em direções nas quais é feito o controle de posição e em direções nas quais é feito o controle de força. A implementação do controle de força e posição é feita definindo-se uma matriz S que representa as direções do controle de posição e de uma matriz S' que representa as direções do controle de força. O sistema de controle baseado neste método tem duas malhas de realimentação tal como ilustrado na figura 1.2.

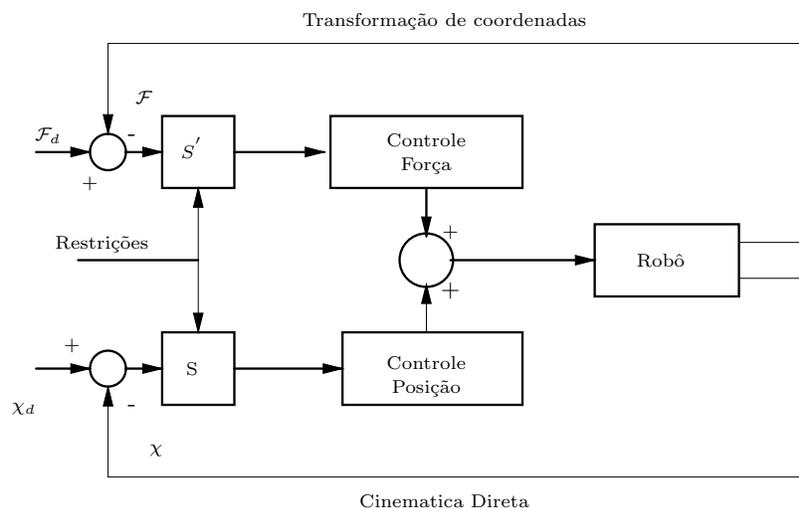


Figura 1.2: Controle híbrido

Já o Controle de *Impedância* permite controlar a força e a posição através do ajuste da impedância mecânica do efetador final e as forças externas geradas pelo contato com o meio ambiente. O Controle

de *Rigidez* (Salisbury, 1980) e o Controle de *Admitância* são casos especiais deste método (Whitney, 1987). Neste método a impedância mecânica do efetuador final (ganho de realimentação do erro de posição para a força) é grande nas direções nas quais se deseja o controle de posição e os ganhos são pequenos nas direções na quais deseja-se controlar a força.

Inicialmente ambos os métodos foram considerados em condições ideais, já que no caso do controle *híbrido* a decomposição é feita considerando superfícies de contato livres de atrito e assumindo que a geometria é completamente conhecida. Porém, na prática, isto não acontece já que a dinâmica do meio é menos conhecida que a dinâmica do robô, sendo necessário assumir erros na modelagem geométrica do meio, tal como, a imprecisão das restrições em termos da direção ou da orientação das superfícies tangentes ao plano de trabalho. Desta forma, os dois métodos têm de tratar imperfeições tais como o atrito nas juntas, a flexibilidade nas juntas e nos elos, o conhecimento impreciso do ambiente e dos parâmetros construtivos do robô, o ruído de medição e as perturbações externas (de Shutter *et al.*, 1997). Visando oferecer soluções a estes problemas, técnicas alternativas como o Controle *Paralelo* proposto por Chiavernini e Sciavicco (1993) e o Controle *Híbrido de Impedância* de Robert e Spong (1988) foram propostas.

McClamroch e Wang (1988) estabeleceram uma base teórica para o estudo dos casos onde existem restrições geométricas na interação do robô e o meio: modelaram matematicamente a dinâmica do robô sujeito a essas restrições e propuseram leis híbridas baseadas no torque computado para garantir condições de estabilidade. Yoshikawa (1987) apresentou uma formulação semelhante à de McClamroch e Wang, com a diferença de ter sido desenvolvida no espaço da tarefa.

Em muitas outras tarefas, o contato não é rígido, mas possui pelo menos alguma flexibilidade (própria do sensor), ou alguma dinâmica de primeira ou segunda ordem (Eppinger e Seering, 1986), (Eppinger e Seering, 1987). Por esse motivo, muitos autores representaram o ambiente de contato como uma mola linear, admitindo pequenas deformações (Eppinger e Seering, 1992), (Volpe e Khosla, 1994).

Ao mesmo tempo foram realizados vários estudos teóricos e alguns experimentais (Kazerooni, 1990), (Ferreti *et al.*, 1995), (Rocco, 1996), (Santibañez e Kelly, 1997), (Ramirez *et al.*, 2000b) envolvendo controladores do tipo *PID* para o controle de força em ambientes rígidos ou com alguma flexibilidade. Um estudo envolvendo a parte experimental sobre a estabilidade no controle de força para controladores clássicos, foi feito por Volpe e Khosla (1993). Da mesma forma, foram feitos estudos de estabilidade para sistemas de controle híbrido (Wen e Murphy, 1991), (Doulgeri, 1999), (Goldsmith, 1999).

A partir destes trabalhos, a compreensão do problema do movimento do robô submetido a restrições foi substancialmente melhorada. Em 1991 Luca e Manes (1991a,b) publicaram uma modelagem que generaliza o Controle *Híbrido* e o Controle de Impedância para diferentes tarefas, considerando restrições geométricas (cinemáticas) e dinâmicas ao mesmo tempo. Este modelo foi desenvolvido considerando um sistema de equações para o robô e outro sistema de equações para o ambiente onde é

executada a tarefa. Em 1993 Manes (1993a,b) faz a análise e síntese de controladores clássicos tipo *PID* para sistemas submetidos a restrições cinemáticas. Em (Luca e Manes, 1994) são empregados também controladores clássicos junto com a dinâmica inversa, adicionando agora restrições dinâmicas. Baseados nesta nova modelagem, Amaral *et al.* (1998) considerando restrições geométricas (Amaral *et al.*, 2000a) e dinâmicas (Amaral *et al.*, 2000b), propõem controladores robustos do tipo estrutura variável para compensar incertezas no modelo e perturbações externas. Li *et al.* (2000) empregam novamente a modelagem de De Luca e Manes para o projeto de controladores *PID* considerando incertezas. Vargas *et al.* (2001b) utilizam um controlador avançado na estrutura de dois graus de liberdade, para compensar incertezas no modelo e perturbações externas. Quase a totalidade dos trabalhos empregando a modelagem de De Luca e Manes trata somente aspectos teóricos e apresenta apenas resultados de simulação, não tendo sido feita a análise experimental desta modelagem. Da mesma forma, o problema do atrito nas juntas e menos ainda o atrito devido à interação com o meio não têm sido tratados de forma experimental. Já nos últimos anos, seguindo a linha experimental, trabalhos como o de Ramirez (2003) e Passold (2003) foram desenvolvidos para tratar o problema do controle de posição e velocidade de robôs manipuladores rígidos e com flexibilidades nas juntas constituindo-se em contribuições importantes. Pelas razões expostas a presente tese concentra-se principalmente na melhoria, sob o aspecto prático, de leis de controle para manipuladores atuando no espaço livre ou sob restrições.

1.3 Objetivos e Contribuições da Tese

Os objetivos e contribuições desta tese referem-se ao controle de força e posição de robôs manipuladores. O desenvolvimento teórico e o projeto de leis de controle lineares e não lineares tais como o controle de dois graus de liberdade, o controle a estrutura variável, o controle nebuloso, a compensação do atrito através do modelo e, principalmente, considerando a existência de poucas contribuições práticas na literatura, visa-se a implementação experimental destas leis de controle. Paralelamente, deseja-se obter procedimentos sistemáticos de projeto para sistemas complexos como é o caso do controle de força e posição de robôs manipuladores.

Os pontos acima mencionados são de grande importância para viabilizar a implementação destas estratégias em nível industrial.

Para atingir os objetivos desta tese, foram realizadas as seguintes etapas:

- O ponto de partida para o desenvolvimento do trabalho foi a modelagem de Luca e Manes (1994) e o trabalho realizado por Amaral (2000);
- Definição de tarefas e trajetórias para o controle de força e de posição ;
- Desenvolvimento de leis de controle de posição, testes de simulação e realização de experimentos no robô *SCARA*;

- Análise e identificação de um modelo para compensar o atrito no controle de posição;
- Desenvolvimento de técnicas de controle de força e testes em simulação;
- Aplicação experimental das técnicas de controle desenvolvidas;
- Análise do modelo de atrito para uso no controle de força.

1.4 Organização e Conteúdo

A presente tese é dividida em oito capítulos. No capítulo 2, é descrita de forma detalhada a modelagem de De Luca e Manes para robôs que interagem com meios cinemáticos e/ou dinâmicos. No capítulo 3, é detalhado o fenômeno do atrito e são apresentados diferentes modelos para a modelagem e a compensação de atrito. A ênfase é estabelecida no modelo de *LuGre*, que é empregado nesta tese. No capítulo 4, são apresentadas algumas das técnicas de controle a serem implementadas experimentalmente: a técnica de Dois Graus de Liberdade (*TDOF*), o Controle a Estrutura Variável (*VSC*) e o Controle Nebuloso (*FLC*). No capítulo 5, é apresentada em detalhe a bancada experimental da Universidade Federal de Santa Catarina utilizada ao longo deste trabalho. No capítulo 6, são apresentados os resultados experimentais para o controle de posição e/ou trajetória para cada uma das técnicas mencionadas considerando a compensação e a não compensação do atrito. No capítulo 7, são apresentados os resultados experimentais do controle de força/posição utilizando o controle *VSC*. Finalmente, no capítulo 8, são apresentadas algumas conclusões e sugeridos alguns temas de pesquisa visando sua aplicação experimental.

Capítulo 2

Modelagem Matemática

A análise e a síntese de técnicas de controle de robôs manipuladores exigem o conhecimento adequado de cada um dos componentes de sua estrutura mecânica e do ambiente no qual irá desenvolver as tarefas. Neste sentido, existe a necessidade da obtenção de modelos matemáticos, cuja complexidade seja capaz de representar diferentes situações e aplicações sem, no entanto, impossibilitar o projeto da lei de controle o seu uso para simular o comportamento do sistema em estudo.

Com este propósito, neste capítulo são abordados tanto a modelagem cinemática quanto a modelagem dinâmica para robôs manipuladores rígidos que interagem com o meio, submetidos a restrições cinemáticas e restrições dinâmicas. A modelagem utilizada é aquela proposta por De Luca e Manes (1994), a qual considera dois sistemas de coordenadas: o primeiro escrito em termos das coordenadas generalizadas que descrevem completamente a configuração do manipulador, e a segunda, composta pelas variáveis que descrevem o movimento admissível do efetuador final, do ponto de vista do meio.

2.1 Justificativa

Seja considerada a tarefa de perfuração ilustrada na figura 2.1, na qual é necessário o controle de posição da ferramenta ao longo da direção de movimento.

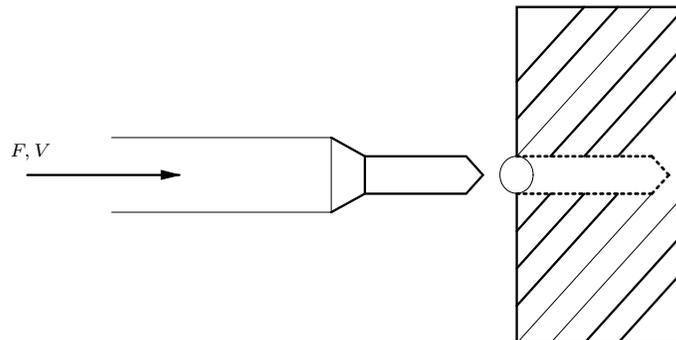


Figura 2.1: Tarefa de perfuração

Nesta tarefa, o meio é altamente resistivo e, em consequência disto, a velocidade de movimento dependerá das propriedades do meio (dureza, rigidez), das propriedades da ferramenta (força máxima) e da dinâmica do robô (limites dos atuadores, forças de atrito, etc.). Neste caso, o controle de força é feito na mesma direção do controle de posição.

A análise correspondente a esta tarefa é feita considerando algumas das possíveis estratégias de controle de força e posição mencionadas no capítulo 1 junto com o trabalho de (de Shutter *et al.*, 1997):

- **Somente controle de força:** Uma força de valor constante é controlada. A ferramenta se movimenta à medida que o material é removido, de forma que o controle de posição não é necessário. O valor de força desejado é determinado por seu valor máximo assim como pelo valor máximo de velocidade. O êxito da execução da tarefa requer o conhecimento da dinâmica do meio.
- **Somente controle de posição:** A posição ou a velocidade são controladas. Esta estratégia é adequada para meios altamente flexíveis onde forças elevadas não são verificadas. Em meios rígidos ou altamente resistivos, as propriedades do meio e da ferramenta devem ser conhecidas de forma exata (Sciavicco e Siciliano, 2000).
- **Controle de impedância:** Nesta estratégia proposta por (Hogan, 1985), para garantir um bom desempenho e o êxito da tarefa a ser realizada pelo robô, precisa-se também de um conhecimento exato da dinâmica do robô e do meio. Isso é necessário para estabelecer a impedância mecânica do efetuador final;
- **Controle paralelo:** Nesta estratégia, apresentada por Chiavernini e Sciavicco (1993), o controle de força precede o controle de posição, de forma que o erro de velocidade pode ser aceito para regular a força. Da mesma forma que as outras estratégias, precisa de um conhecimento exato da dinâmica do meio.

Essas estratégias de controle possuem em comum o fato de que para o êxito da execução da tarefa precisa-se do conhecimento adequado da dinâmica do meio assim como da geometria de contato. Esta característica está ligada ao fato de que o desenvolvimento de tarefas onde o manipulador interage com o meio, pode ou não implicar na troca de energia. Se o meio impõe restrições puramente cinemáticas (ou geométricas) sobre o movimento do efetuador final, somente um balanço estático de forças e torques acontece durante o contato (supondo que não existe atrito). Estes tipos de considerações na modelagem implicam a não transferência de energia (Yoshikawa, 1987), (McClamroch e Wang, 1988).

De outra forma, a troca de energia entre o robô e o meio é tratada considerando-se um modelo de impedância para a interação dinâmica (Hogan, 1985), (Eppinger e Seering, 1986), (Chiavernini *et al.*, 1994).

Como uma oposição às interações *completamente estáticas* ou *completamente dinâmicas*, encontram-se casos nos quais o robô, além de estar sujeito a restrições cinemáticas, também é submetido a forças dinâmicas no efetuador final; forças não compensadas nas quais existe troca de energia com o meio.

Como consequência das restrições dinâmicas, a modelagem clássica para a obtenção de estratégias de controle híbrido precisa ser redefinida para incluir estas restrições no ambiente de trabalho. Neste sentido, De Luca e Manes (1994) propõem um modelo que inclui ambos os tipos de restrições. Esta modelagem foi adotada para desenvolvimentos teóricos e de simulação por Amaral (2000). Da mesma forma, neste trabalho esta formulação será adotada, dando ênfase aos problemas experimentais: desenvolvimento de leis de controle, compensação de atrito, implementação prática, etc.

2.2 Modelagem Cinemática e Dinâmica

Para descrever completamente a interação do robô com o meio, representado por restrições cinemáticas e dinâmicas, é necessário, além das coordenadas de junta q , que caracterizam o robô, um outro conjunto de coordenadas complementar s . Este último caracteriza o meio, determina o tipo de contato e ainda permite descrever a posição do efetuador final x em dois referenciais diferentes: um associado ao robô e outro associado ao meio. As restrições cinemáticas correspondem a restrições de tipo geométrico (hipersuperfícies), que permitem o movimento livre do efetuador final do robô nas direções tangentes e o obstruem nas direções normais. Neste sentido, as restrições dinâmicas se caracterizam por apresentar um comportamento dinâmico em uma ou mais direções, sendo que nas outras, as restrições são cinemáticas.

O vetor de coordenadas s associadas ao meio será composto pelas componentes cinemáticas s_K , associadas ao contato e/ou movimento do meio, e pelas componentes dinâmicas s_D associadas às energias cinética e potencial do meio. Robôs sujeitos somente a restrições cinemáticas $s = s_K$, ou restrições puramente dinâmicas $s = s_D$, são casos especiais desta modelagem.

2.2.1 Modelagem cinemática do manipulador

Seja um robô manipulador de $(n+1)$ elos correspondentes a n juntas de translação ou rotação, constituído por uma cadeia cinemática aberta de corpos rígidos, parametrizado pelo vetor de coordenadas generalizadas $[q_1 \ q_2 \ \dots \ q_n]$ de variáveis de junta, conforme está ilustrado na figura 2.2.

Seja também o vetor $x = (p, o)$, de dimensão 6, onde p descreve a posição do efetuador final no espaço cartesiano \mathcal{R}^3 . Sua orientação $o = (\varphi, \vartheta, \psi)$ é dada por uma representação mínima, a dos ângulos de *Euler* por exemplo (de Wit *et al.*, 1996). Com isto, a cinemática do efetuador final em função das variáveis de junta é expressa por:

$$x = k(q) \tag{2.1}$$

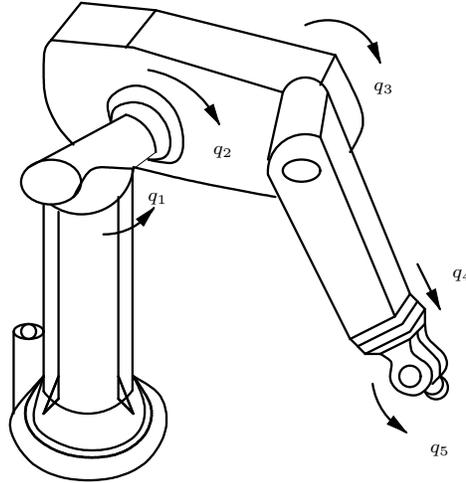


Figura 2.2: Variáveis de junta e elos num robô de 5 graus de liberdade

onde $k(q)$ é um vetor de funções não lineares, que mapeia o espaço de juntas no espaço cartesiano, cuja derivada em função do tempo é:

$$\dot{x} = \frac{\partial k(q)}{\partial q} \dot{q} = J_a(q) \dot{q} \quad (2.2)$$

onde $J_a(q)$ de dimensão $6 \times n$ é a matriz conhecida como *Jacobiano Analítico*.

O vetor de velocidades $v = (\dot{p}, \omega)$, composto pela velocidade linear \dot{p} e pela velocidade angular ω , está relacionado com \dot{x} , através de:

$$v = G(x) \dot{x} \quad (2.3)$$

onde a matriz $G(x)$ é definida como:

$$G(x) = \begin{bmatrix} I_{3 \times 3} & 0 \\ 0 & \hat{G}(\varphi, \vartheta, \psi) \end{bmatrix} \quad (2.4)$$

onde $\hat{G}(\varphi, \vartheta, \psi)$ é uma matriz de funções de dimensão (3×3) , que expressa o mapeamento entre a derivada do vetor de orientação $\frac{d\omega}{dt}$ e a velocidade angular ω (Luca e Manes, 1994), (Amaral, 2000).

Definindo

$$J(q) = G(k(q)) J_a(q) \quad (2.5)$$

o vetor de velocidades generalizadas pode ser escrito em função da velocidade \dot{q} das juntas como sendo:

$$v = \begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = J(q) \dot{q} \quad (2.6)$$

onde $J(q)$ de dimensão $6 \times n$ é o conhecido *Jacobiano Geométrico* ou jacobiano básico do manipulador (Luca e Manes, 1994).

Existem configurações dentro do espaço de trabalho do manipulador onde o Jacobiano é singular. Ao longo desta tese, supõe-se que o Jacobiano seja de posto completo (não singular), o que é conseguido planejando seu movimento, de tal forma que seja possível evitar as configurações singulares (Luca e Manes, 1994).

2.2.2 Modelagem cinemática do meio

Considere-se o meio como um sistema mecânico de $d \leq 6$ graus de liberdade, tal que pode ser descrito por um modelo dinâmico de segunda ordem, em termos do conjunto de variáveis de configuração $s_D \in \mathcal{R}^d$. Quando o efetuador final do robô tem um vínculo total com o meio, então o conjunto de posições depende simplesmente de s_D ; mas este conjunto de variáveis pode não ser suficiente para descrever algumas tarefas nas quais se tem contato entre o robô e o meio. Então, há a necessidade de k variáveis cinemáticas que sejam somadas às d variáveis já definidas, para descrever completamente o movimento do efetuador final visto do meio (Luca e Manes, 1994).

Com a inclusão destas variáveis puramente cinemáticas, identificadas por $s_K \in \mathcal{R}^k$, são definidas as direções em que é permitido o movimento, seja este caracterizado por alguma dinâmica imposta pelo meio (relacionada aos d graus de liberdade do meio), ou não (descrito pelas k variáveis cinemáticas), além de ficarem evidentes as direções em que o movimento é impedido.

A união destes vetores denotado por $s = (s_K, s_D) \in \mathcal{R}^e$, com $e = k + d \leq 6$, compõe um conjunto de parâmetros suficientes para descrever uma ampla variedade de tarefas, para as quais existe contato com o meio.

Considerando-se que o efetuador final está totalmente vinculado ao meio, isto é, que possui dinâmica em todas as direções e sentidos de rotação, então, $d = e = 6$ ($k = 0$). E, num contato que não possua dinâmica em qualquer direção, permitirá livre movimento ao efetuador final, em $k = e < 6$ direções, e lhe impõe $6 - k$ restrições, visto que o meio está submetido somente a restrições cinemáticas, isto é, $d = 0$.

Destas definições, a posição do efetuador final, vista do lado do meio pode também ser expressa como:

$$x = \Gamma(s) \tag{2.7}$$

onde $\Gamma(s)$ é um vetor de funções não lineares que relaciona s a x . Derivando a equação (2.7) com relação ao tempo:

$$\dot{x} = \frac{\partial \Gamma(s)}{\partial s} \dot{s} \tag{2.8}$$

Da mesma forma que a posição do efetuador final é obtida da equação (2.7), também é obtida da equação (2.1) do lado do robô. Decorre daí que a seguinte igualdade tem que ser satisfeita:

$$x = k(q) = \Gamma(s) \quad \Rightarrow \quad k(q) - \Gamma(s) = 0 \tag{2.9}$$

Além disso, substituindo a equação (2.8) na equação (2.3), é obtido o vetor das velocidades generalizadas v em função de s e sua derivada:

$$\begin{aligned} v &= \begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = \Upsilon(s)\dot{s} \\ v &= \begin{bmatrix} \Upsilon_K(s) & \Upsilon_D(s) \end{bmatrix} \begin{bmatrix} \dot{s}_K \\ \dot{s}_D \end{bmatrix} \end{aligned} \quad (2.10)$$

onde

$$\Upsilon(s) = G(\Gamma(s)) \frac{\partial \Gamma(s)}{\partial s} \quad (2.11)$$

desempenha um papel semelhante ao Jacobiano do manipulador e também é considerado de posto completo. Já as matrizes $\Upsilon_K(s) \in \mathcal{R}^{6 \times k}$ e $\Upsilon_D(s) \in \mathcal{R}^{6 \times d}$ relacionam o vetor de velocidades nas direções cinemáticas e dinâmicas ao vetor de velocidades generalizadas v .

Reescrevendo a equação (2.10) e igualando-a com a equação (2.6) é obtido:

$$v = \Upsilon_K(s)\dot{s}_K + \Upsilon_D(s)\dot{s}_D = J(q)\dot{q} \quad (2.12)$$

A equação (2.12) é válida se as matrizes $\Upsilon(s)$ e $J(q)$ têm posto completo.

2.2.3 Propriedades dos contatos cinemáticos e/ou dinâmicos

A partir dos trabalhos de (Luca e Manes, 1994) e (Amaral, 2000), as seguintes propriedades para contatos cinemáticos e/ou dinâmicos podem ser estabelecidas:

1. As forças de reação F_R não realizam trabalho nas direções admissíveis de movimento, sejam elas cinemáticas s_K ou dinâmicas s_D ; estas forças são expressas por:

$$F_R = Y_R(s)\lambda_R \quad (2.13)$$

onde $Y_R(s)$ é uma matriz de dimensão $(6 \times (6 - e))$, cujas linhas formam uma base para as direções de existência de F_R , e onde $\lambda_R \in \mathcal{R}^{6-e}$ é o vetor que parametriza estas forças;

2. As forças de contato admissíveis $F = F_R + F_A$ não realizam trabalho nas direções cinemáticas s_K ; as forças aplicadas nas direções dinâmicas são:

$$F_A = Y_A(s)\lambda_A \quad (2.14)$$

onde $Y_A(s)$ é uma matriz de dimensão $(6 \times d)$, cujas linhas formam uma base para as direções de existência de F_A , e onde $\lambda_A \in \mathcal{R}^d$ é o vetor que parametriza estas forças.

Destas considerações, a força total de contato entre o efetuador final e o meio, é escrita como:

$$\begin{aligned}
 F &= F_R + F_A \\
 &= Y_R(s)\lambda_R + Y_A(s)\lambda_A \\
 &= \begin{bmatrix} Y_R(s) & Y_A(s) \end{bmatrix} \begin{bmatrix} \lambda_R \\ \lambda_A \end{bmatrix} \\
 &= Y(s)\lambda
 \end{aligned} \tag{2.15}$$

A primeira propriedade pode ser expressa, numa forma alternativa, como: *o produto escalar entre as forças de reação e as velocidades nas direções admissíveis é nulo*, isto é,

$$v^T F_R = \begin{bmatrix} \dot{p}^T & \omega^T \end{bmatrix} \begin{bmatrix} f_R \\ m_R \end{bmatrix} = 0 \tag{2.16}$$

onde f_R são as forças de reação e m_R são os momentos de reação no efetuador final. Usando (2.10) e a definição da força de reação F_R , a equação (2.16) é reescrita como:

$$\dot{s}^T \Upsilon^T(s) Y_R(s) \lambda_R = 0 \tag{2.17}$$

que admite uma forma mais enxuta, visto que deve manter-se verdadeira para \dot{s} e λ_R quaisquer (nulos ou não), a saber:

$$\Upsilon^T(s) Y_R(s) = 0_{e \times (6-e)} \tag{2.18}$$

Esta identidade é a representação matemática da primeira propriedade. A matriz $Y_R(s)$ deve ser escolhida de tal modo a satisfazer a equação (2.18) e possuir posto igual a $(6 - e)$. Para contatos puramente cinemáticos, como o ilustrado na figura 2.3, a equação (2.18) é simplificada para:

$$\Upsilon_K^T(s) Y_R(s) = 0_{k \times (6-k)} \tag{2.19}$$

que reflete a condição de não-transferência de energia em contatos cinemáticos.

Da mesma forma, a segunda propriedade pode ser formulada da seguinte maneira: *o produto escalar entre as forças totais aplicadas sobre o ambiente $F = F_R + F_A$ e as velocidades nas direções cinemáticas é nulo*. Usando a equação (2.10) e usando as definições das forças F_R e F_A esta propriedade é expressa por:

$$\dot{s}_K^T \Upsilon_K^T(s) Y(s) \lambda = 0 \tag{2.20}$$

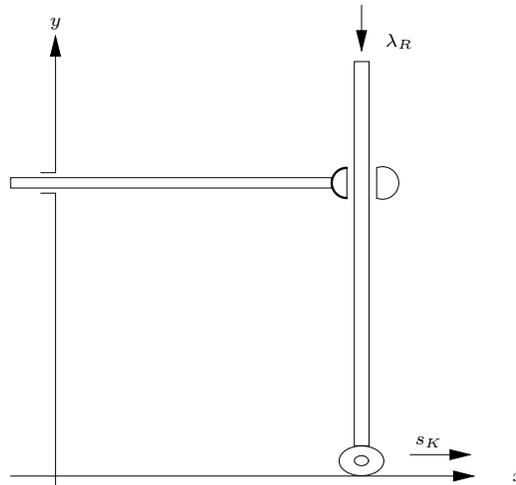


Figura 2.3: Contato puramente cinemático

Como esta identidade deve manter-se verdadeira para \dot{s}_K e λ quaisquer (nulos ou não), ela pode ser reduzida a uma forma mais compacta, isto é:

$$\Upsilon_K^T(s)Y(s) = 0_{k \times (6-k)} \quad (2.21)$$

A matriz $Y(s)$ possui duas partições $Y_R(s)$ e $Y_A(s)$. A primeira delas é obtida a partir da equação (2.19) e a segunda deve ser escolhida de tal forma a satisfazer a equação (2.21) com posto igual a d .

Para contatos dinâmicos (cinemáticos e/ou dinâmicos), como o ilustrado na figura 2.4, só há troca de energia entre o robô e o meio na direção dinâmica. De fato, o produto escalar entre a velocidade \dot{s}_D e a força na direção dinâmica F_A é diferente de zero, desde que \dot{s}_D e F_A sejam não nulos (Luca e Manes, 1994), (Amaral, 2000).

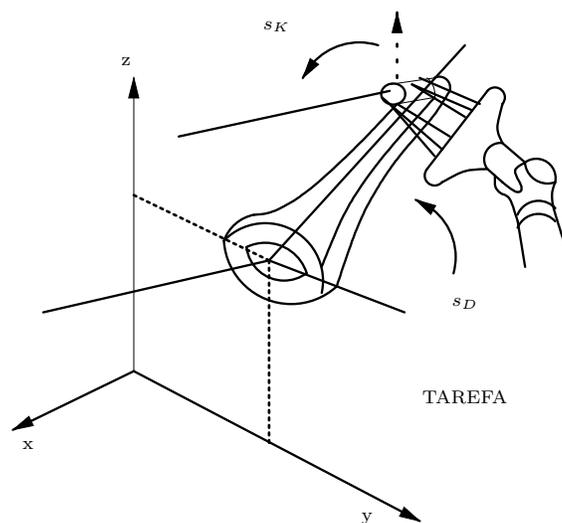


Figura 2.4: Contato cinemático e/ou dinâmico

2.2.4 Modelo dinâmico do sistema manipulador-meio

O modelo dinâmico do sistema *manipulador-meio* é obtido através da formulação de Lagrange (Sciavicco e Siciliano, 2000), empregando o conjunto de coordenadas generalizadas (componente dinâmica) s_D para o meio e as coordenadas de junta q para o robô.

As energias *cinética*, *potencial* e o *Lagrangeano* do manipulador são definidas respectivamente por:

$$\begin{aligned}\mathcal{T} &= \frac{1}{2}\dot{q}^T M(q)\dot{q} \\ \mathcal{U} &= \mathcal{U}(q) \\ \mathcal{L}(q, \dot{q}) &= \mathcal{T} - \mathcal{U}\end{aligned}\tag{2.22}$$

As energias *cinética e potencial do meio*, assim como o *Lagrangeano* correspondente ao meio, são dados por:

$$\begin{aligned}\mathcal{T}_E &= \frac{1}{2}\dot{s}_D^T M_E(s_D)\dot{s}_D \\ \mathcal{U}_E &= \mathcal{U}_D(s_D) \\ \mathcal{L}_E(s_D, \dot{s}_D) &= \mathcal{T}_E - \mathcal{U}_E\end{aligned}\tag{2.23}$$

A condição de contato robô-meio requer que a posição do efetuador final, dada em função dos dois conjuntos de parâmetros, coincida, ou seja, $x = k(q) = \Gamma(s)$.

Somando os Lagrangeanos do manipulador e do meio, na presença das restrições, obtém-se o seguinte *Lagrangeano Composto*:

$$\mathcal{L}_C(q, \dot{q}, s, \dot{s}) = \mathcal{L}(q, \dot{q}) + \mathcal{L}_E(s_D, \dot{s}_D) + \gamma^T [\Gamma(s) - k(q)]\tag{2.24}$$

onde $\gamma \in \mathcal{R}^6$ é o vetor de multiplicadores de Lagrange.

Considerando-se que as forças de interação são internas ao sistema robô-meio, elas não aparecerão explicitamente no modelo. Portanto, somente as forças/torques τ , fornecidos pelos motores, e as forças de atrito, $-F_f(q, \dot{q})$ e $-F_{E_f}(s_D, \dot{s}_D)$, serão incluídas como forças não conservativas. Então, aplicando a equação de Lagrange à (2.24), resulta em:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}_C}{\partial \dot{q}}\right)^T - \left(\frac{\partial \mathcal{L}_C}{\partial q}\right)^T = \tau - F_f(q, \dot{q})\tag{2.25}$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}_C}{\partial \dot{s}_D}\right)^T - \left(\frac{\partial \mathcal{L}_C}{\partial s_D}\right)^T = -F_{E_f}(s_D, \dot{s}_D)\tag{2.26}$$

$$-\left(\frac{\partial \mathcal{L}_C}{\partial s_K}\right)^T = 0\tag{2.27}$$

$$-\left(\frac{\partial \mathcal{L}_C}{\partial \gamma}\right)^T = 0\tag{2.28}$$

que podem ser reescritas como:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right)^T - \left(\frac{\partial \mathcal{L}}{\partial q}\right)^T + \left(\frac{\partial k}{\partial q}\right)^T \gamma = \tau - F_f(q, \dot{q}) \quad (2.29)$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}_E}{\partial \dot{s}_D}\right)^T - \left(\frac{\partial \mathcal{L}_E}{\partial s_D}\right)^T - \left(\frac{\partial \Gamma}{\partial s_D}\right)^T \gamma = -F_{E_f}(s_D, \dot{s}_D) \quad (2.30)$$

$$\left(\frac{\partial \Gamma}{\partial s_K}\right)^T \gamma = 0 \quad (2.31)$$

$$\Gamma(s) - k(q) = 0 \quad (2.32)$$

A dinâmica completa de um robô, cujo efetuador final move-se livremente no espaço, é descrita totalmente pelos dois primeiros termos da equação (2.29). Desenvolvendo esses termos, obtém-se:

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right)^T - \left(\frac{\partial \mathcal{L}}{\partial q}\right)^T &= M(q)\ddot{q} + \dot{M}(q)\dot{q} - \frac{1}{2}\frac{\partial}{\partial q}(\dot{q}^T M(q)\dot{q}) + \frac{\partial}{\partial q}\mathcal{U}(q) \\ &= M(q)\ddot{q} + C(q, \dot{q}) + G(q) \end{aligned} \quad (2.33)$$

onde $C(q, \dot{q})$ corresponde ao vetor de torques centrífugos e de Coriolis e $G(q) = \frac{\partial}{\partial q}\mathcal{U}(q)$ corresponde ao vetor de torques gravitacionais. A k -ésima linha de (2.33) tem a seguinte forma:

$$\sum_j m_{kj}\ddot{q}_j + \sum_{i,j} c_{ijk}\dot{q}_i\dot{q}_j + g_k \quad (2.34)$$

onde c_{ijk} é dado por:

$$c_{ijk} = \frac{1}{2}\left(\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k}\right) \quad (2.35)$$

conhecido como símbolo de Christoffel (Sciavicco e Siciliano, 2000).

Um desenvolvimento semelhante é usado para os dois primeiros termos da equação (2.30); no entanto, além do termo gravitacional, o meio pode apresentar também um termo elástico (Manes, 1993a), (Amaral, 2000).

Usando argumentos de trabalho virtual¹, o multiplicador γ pode ser interpretado como uma força generalizada, realizando trabalho ao longo do deslocamento infinitesimal $dx(\dot{x})$; assim este trabalho é dado por:

$$dW = \gamma^T dx \quad (2.36)$$

Considerando-se a igualdade $v = G(x)\dot{x}$, e admitindo-se que v seja dada por: $\frac{dx_\omega}{dt}$, conclui-se que:

$$dx = G^{-1}dx_\omega \quad (2.37)$$

¹Relação das forças generalizadas aplicadas sobre o efetuador final e as forças aplicadas nas juntas (Sciavicco e Siciliano, 2000).

Com isso a equação (2.36) passa a ser escrita na forma:

$$dW = \gamma^T G^{-1} dx_\omega = (G^{-T} \gamma)^T dx_\omega \quad (2.38)$$

de onde se deduz que a força

$$F = G^{-T} \gamma \quad (2.39)$$

realiza trabalho ao longo de $dx_\omega(v)$.

Usando este resultado, o terceiro termo à esquerda da igualdade da equação (2.29) pode ser expresso em função de F como:

$$\begin{aligned} \left(\frac{\partial k}{\partial q}\right)^T \gamma &= J_a^T(q) \gamma \\ &= J^T(q) G^{-T} \gamma \\ &= J^T(q) F \end{aligned} \quad (2.40)$$

Além disso, sabendo que:

$$\begin{aligned} \Upsilon_D(s) &= G \left(\frac{\partial \Gamma(s)}{\partial s_D} \right) \\ \Upsilon_K(s) &= G \left(\frac{\partial \Gamma(s)}{\partial s_K} \right) \end{aligned} \quad (2.41)$$

o terceiro termo da equação (2.30) e a equação (2.31) são reescritos como:

$$\begin{aligned} \left(\frac{\partial \Gamma(s)}{\partial s_D}\right)^T \gamma &= \Upsilon_D^T(s) G^{-T} \gamma \\ &= \Upsilon_D^T F \end{aligned} \quad (2.42)$$

e

$$\begin{aligned} \left(\frac{\partial \Gamma(s)}{\partial s_K}\right)^T \gamma &= \Upsilon_K^T(s) G^{-T} \gamma \\ &= \Upsilon_K^T F \end{aligned} \quad (2.43)$$

Da última identidade e lembrando que $F = Y(s)\lambda$, a equação (2.31) transforma-se em:

$$\Upsilon_K^T(s) F = \Upsilon_K^T Y(s) \lambda = 0_{k \times 1} \quad (2.44)$$

que é uma variação da equação (2.21), pois deve ser verdadeira para qualquer λ .

Utilizando as equações (2.29) a (2.32), obtém-se o seguinte modelo dinâmico para o robô e o meio (Luca e Manes, 1991a):

1. n equações diferenciais de segunda ordem relativas ao robô:

$$M(q)\ddot{q} + n(q, \dot{q}) = \tau - J^T(q)F \quad (2.45)$$

onde

$$n(q, \dot{q}) = C(q, \dot{q}) + F_f(q, \dot{q}) + G(q) \quad (2.46)$$

incorpora os termos correspondentes às forças centrífugas e de Coriolis, o vetor das forças de atrito associadas ao robô e o vetor de forças gravitacionais;

2. d equações diferenciais de segunda ordem relativas ao meio:

$$M_E(s_D)\ddot{s}_D + n_E(s_D, \dot{s}_D) = \Upsilon_D^T F \quad (2.47)$$

onde

$$n_E(s_D, \dot{s}_D) = C_E(s_D, \dot{s}_D) + \left(\frac{\partial \mathcal{U}_E}{\partial s_D}\right)^T + F_{E_f}(s_D, \dot{s}_D) \quad (2.48)$$

sendo $\left(\frac{\partial \mathcal{U}_E}{\partial s_D}\right)^T$ a componente correspondente à gravidade e ou termos elásticos, C_E os termos de Coriolis e centrífugas e F_{E_f} corresponde ao vetor de forças de atrito associadas ao meio;

3. k equações algébricas, que estabelecem a não realização de trabalho das forças de contato ao longo das direções cinemáticas:

$$\Upsilon_K^T F = 0 \quad (2.49)$$

4. e finalmente, mais 6 equações algébricas, que estabelecem o acoplamento entre os dois subsistemas:

$$\Gamma(s) - k(q) = 0 \quad (2.50)$$

2.2.5 Modelo dinâmico compacto

Para propósitos de controle, uma forma mais apropriada e conveniente destas equações é detalhada. Derivando-se a equação (2.12), obtém-se:

$$\Upsilon_K(s)\ddot{s}_K + \dot{\Upsilon}_K(s, \dot{s})\dot{s}_K + \Upsilon_D(s)\ddot{s}_D + \dot{\Upsilon}_D(s, \dot{s})\dot{s}_D = J(q)\ddot{q} + \dot{J}(q, \dot{q})\dot{q} \quad (2.51)$$

Resolvendo as equações (2.45) e (2.47), para \ddot{q} e \ddot{s}_D e substituindo-as em (2.51) tem-se:

$$\Upsilon_K\ddot{s}_K + \dot{\Upsilon}_K\dot{s}_K + \Upsilon_D M_E^{-1} \Upsilon_D^T F - \Upsilon_D M_E^{-1} n_E + \dot{\Upsilon}_D \dot{s}_D = J M^{-1} \tau - J M^{-1} J^T F - J M^{-1} n + \dot{J} \dot{q} \quad (2.52)$$

Considerando que a força é dada por $F = Y_R \lambda_R + Y_A \lambda_A$ e sabendo que $\Upsilon_D^T Y_R = 0_{d \times (6-e)}$, o modelo do sistema *manipulador-meio* pode ser escrito da seguinte forma:

$$\begin{bmatrix} \lambda_A \\ \lambda_R \\ \ddot{s}_K \end{bmatrix} = Q^{-1}(m + JM^{-1}\tau) + d \quad (2.53)$$

onde

$$Q = Q(q, s) = \begin{bmatrix} (\Upsilon_D M_E^{-1} \Upsilon_D^T + JM^{-1} J^T) Y_A & JM^{-1} J^T Y_R & \Upsilon_K \end{bmatrix} \quad (2.54)$$

$$m = m(q, \dot{q}, s_K, \dot{s}_K) = -\dot{\Upsilon}_K \dot{s}_K - \dot{\Upsilon}_D \dot{s}_D + J\dot{q} + \Upsilon_D M_E^{-1} n_E - JM^{-1} n \quad (2.55)$$

e

$$d = \begin{bmatrix} d_A \\ d_R \\ d_K \end{bmatrix} \quad (2.56)$$

foi adicionado para levar em conta as perturbações externas ao sistema no efetuador final. A principal característica deste modelo é a explicitação das variáveis que se deseja controlar, isto é λ_A e λ_R , ligadas à força de contato e \ddot{s}_K relacionada ao movimento.

É importante observar que o sistema (2.53) tem solução única somente se a matriz Q for inversível, o que acontece se as matrizes J e Υ tiverem posto completo (Luca e Manes, 1991a).

2.3 Controle do Sistema Robô-Meio

Nesta seção, a lei de controle baseada na dinâmica inversa é aplicada ao sistema robô-meio considerando as restrições cinemáticas e dinâmicas. Com este propósito, na figura 2.5 é ilustrada esta estratégia de controle onde na malha interna é feita a compensação das não-linearidades e na malha externa podem ser aplicadas outras técnicas de controle lineares ou não.

A aplicação da malha interna de controle permite obter um sistema da forma:

$$\begin{bmatrix} \lambda_A \\ \lambda_R \\ \ddot{s}_K \end{bmatrix} = \begin{bmatrix} \lambda_{A,des} \\ \lambda_{R,des} \\ \lambda_{K,des} \end{bmatrix} \quad (2.57)$$

Observa-se que na equação (2.57) é necessário o projeto das técnicas de controle de força u_A , u_R e de movimento u_K . Em termos experimentais, deve-se expressar as componentes s_K , s_D , λ_A e λ_R em função das variáveis lidas pelos sensores do robô. Ao mesmo tempo, é necessário considerar que o sistema linearizado não é exato, pelo que os controladores a serem projetados devem ser capazes de compensar incertezas.

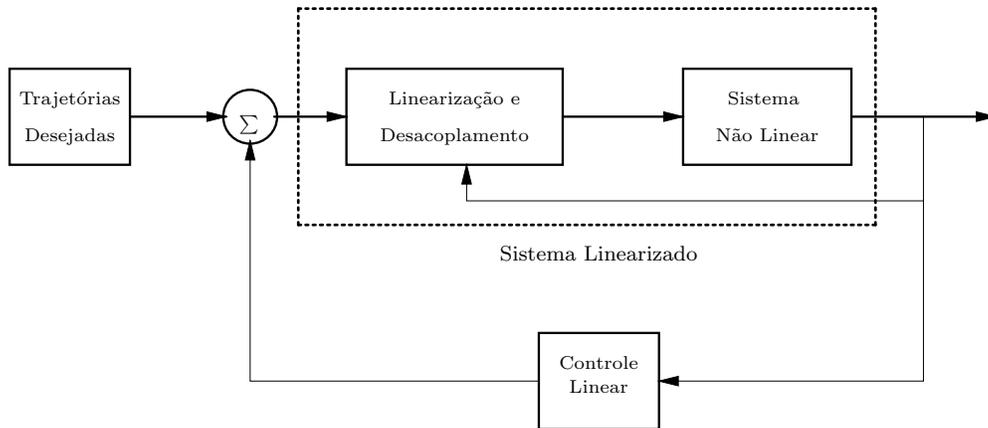


Figura 2.5: Controle baseado na dinâmica inversa

2.4 Controle do Robô Submetido a Restrições Puramente Cinemáticas

A modelagem de um robô rígido submetido somente a restrições cinemáticas é mais simples, já que não é considerada a dinâmica do meio, apenas admite o movimento livre do efetuador final em algumas direções e o restringe nas outras, contrabalançando a força aplicada com uma reação igual e oposta.

2.4.1 Robô rígido submetido a restrições cinemáticas

O modelo dinâmico de um robô rígido submetido a restrições puramente cinemáticas, que admite o movimento do efetuador final em $k < 6$ direções e o restringe nas restantes $6 - k$, consiste de:

- n equações diferenciais de segunda ordem relativas ao robô, idênticas às equações (2.45) e (2.46), porque o robô não sofreu qualquer alteração;
- k equações algébricas, que estabelecem a não realização de trabalho das forças $F_R = Y_R \lambda_R$ ao longo das direções cinemáticas:

$$\Upsilon_K^T Y_R = 0 \quad (2.58)$$

- e finalmente, mais 6 equações algébricas, idênticas às equações (2.50), que estabelecem o acoplamento entre os dois sub-sistemas.

Com essas considerações uma representação mais adequada para este sistema é:

$$\begin{bmatrix} \lambda_R \\ \ddot{s}_K \end{bmatrix} = Q^{-1}(m + JM^{-1}\tau) + d \quad (2.59)$$

onde:

$$Q = \begin{bmatrix} JM^{-1}J^TY_R & \Upsilon_K \end{bmatrix} \quad (2.60)$$

$$m = -\check{\Upsilon}_K\dot{s}_K + \dot{J}\dot{q} - JM^{-1}n \quad (2.61)$$

$$d = \begin{bmatrix} d_R \\ d_K \end{bmatrix} \quad (2.62)$$

Observa-se que a partir da equação (2.59) pode-se derivar com facilidade um controlador baseado na dinâmica inversa, onde as variáveis a serem controladas são λ_R e \check{s}_K . Descrevendo a tarefa a ser realizada através dos valores desejados da força de reação ($\lambda_{R,des}$) e do movimento nas direções cinemáticas ($\check{s}_{K,des}$),

$$\tau_{des} = \begin{bmatrix} \lambda_{R,des} \\ \check{s}_{K,des} \end{bmatrix} \quad (2.63)$$

tal que estes termos permitam o projeto de diferentes leis de controle.

Os torques de entrada que atendem a especificação (2.63) são imediatamente derivados de (2.59), ou seja:

$$\tau = (JM^{-1})^\# \left[Q \left(\begin{bmatrix} \lambda_{R,des} \\ \check{s}_{K,des} \end{bmatrix} - d \right) - m \right] \quad (2.64)$$

onde $(JM^{-1})^\#$ é qualquer pseudo inversa à direita de (JM^{-1}) , que pode ser por exemplo $MJ^\#$, sendo $J^\#$ a pseudo-inversa de J . Se J for quadrada e não singular, então $(JM^{-1})^\# = MJ^{-1}$.

Note-se que a equação (2.64) inclui as componentes correspondentes ao atrito e às perturbações externas. Ao contrário da nomenclatura empregada por Amaral (2000), esta representação não inclui incertezas na modelagem do robô, já que pelas características experimentais do trabalho as incertezas encontram-se implícitas no modelo do robô, o que não acontece no caso de somente considerar simulação.

2.4.2 Robô rígido livre de restrições

Se $k = 6$, o efetuador final, possui total liberdade de movimento dentro do espaço de trabalho do manipulador, as forças de reação, juntamente com Y_R e λ_R , não existem mais; conseqüentemente, a equação (2.58) não é mais necessária e o modelo dinâmico se resume a:

1. n equações diferenciais de segunda ordem relativas ao robô, idênticas às equações (2.45) e (2.46), neste caso, o termo J^TF representa os pesos dos corpos movimentados pelo manipulador, podendo ser incluído em $n(q, \dot{q})$, como mais uma parcela do torque gravitacional;
2. 6 equações algébricas, idênticas às equações (2.51), que estabelecem a equivalência entre as descrições da trajetória do efetuador final nos dois sistemas de coordenadas: a do robô e a do meio.

Não há mais a necessidade do controle de força, apenas a compensação do peso dos corpos que estão sendo manipulados. Portanto, este modelo simplificado descreve a dinâmica do robô quando ele realiza tarefas nas quais é suficiente o *controle de posição e/ou trajetória*.

Portanto, a equação (2.59) assume a seguinte forma:

$$M(q)\ddot{q} + n = \tau + d \quad (2.65)$$

que corresponde à equação dinâmica do manipulador, quando este não está submetido a restrições (Sciavicco e Siciliano, 2000), (Yoshikawa, 2000). A equação (2.65) também inclui os termos correspondentes ao atrito e às perturbações externas.

2.5 Controle do Robô Submetido a Restrições Puramente Dinâmicas

Nesta seção, a lei de controle baseada na dinâmica inversa é aplicada ao sistema robô-meio onde é adicionada a componente dinâmica.

2.5.1 Controle da força na direção dinâmica

Neste caso deseja-se controlar as seguintes variáveis:

- forças aplicadas nas direções dinâmicas, parametrizados pelo vetor λ_A , de dimensão d ;
- forças de reação nas direções estáticas, parametrizados pelo vetor λ_R , de dimensão $6 - k - d$;
- movimento nas direções cinemáticas, parametrizado pelo vetor \ddot{s}_K , de dimensão k .

Considerando o modelo obtido em (2.53), a tarefa a ser executada pode ser especificada em função das variáveis a serem controladas como $[\lambda_{A,des} \ \lambda_{R,des} \ \ddot{s}_{K,des}]$. Os torques de entrada que atenderão essa especificação são obtidos diretamente da expressão (2.53) como sendo:

$$\tau = (JM^{-1})^\# \left(Q \left(\begin{bmatrix} \lambda_{A,des} \\ \lambda_{R,des} \\ \ddot{s}_{K,des} \end{bmatrix} - d \right) - m \right) \quad (2.66)$$

Novamente as leis de controle da equação (2.66) são propostas considerando as variáveis $[\lambda_{A,des} \ \lambda_{R,des} \ \ddot{s}_{K,des}]$.

Dada a natureza experimental do trabalho, é possível a obtenção dos parâmetros do robô, mas não acontece o mesmo com o meio. Neste caso existe a necessidade de obter valores aproximados para os parâmetros do meio e incluir incertezas nestes termos. Para este propósito as seguintes equações foram utilizadas em (Amaral, 2000):

$$\begin{aligned}\tilde{M}_E &= \hat{M}_E - M_E \\ \tilde{n}_E &= \hat{n}_E - n_E\end{aligned}$$

onde os termos M_E, n_E e \hat{M}, \hat{n} correspondem aos valores nominais (aproximados) e aos valores estimados do meio respectivamente.

2.5.2 Controle de movimento na direção dinâmica

Ao invés da força (λ_A), pode-se preferir controlar o movimento (\ddot{s}_D) alterando-se o conjunto de variáveis a serem controladas para:

- movimento nas direções dinâmicas, parametrizado pelo vetor \ddot{s}_D , de dimensão d ;
- forças de reação nas direções cinemáticas, parametrizado pelo vetor λ_R , de dimensão $6 - k - d$;
- movimento nas direções cinemáticas, parametrizado pelo vetor \ddot{s}_K , de dimensão k ;

Com esta mudança, o modelo dinâmico do sistema robô-meio é descrito da seguinte forma:

$$\begin{bmatrix} \ddot{s}_D \\ \lambda_R \\ \ddot{s}_K \end{bmatrix} = Q_s^{-1}(m_s + JM^{-1}\tau) + d_s \quad (2.67)$$

onde:

$$Q_s = \begin{bmatrix} \Upsilon_D + JM^{-1}J^T Y_A (\Upsilon_D^T Y_A)^{-1} M_E & JM^{-1}J^T Y_R & \Upsilon_K \end{bmatrix} \quad (2.68)$$

$$m_s = -\dot{\Upsilon}_K \dot{s}_K - \dot{\Upsilon}_D \dot{s}_D - JM^{-1}Y^T Y_A (\Upsilon_D^T Y_A)^{-1} n_E + \dot{J}\dot{q} - JM^{-1}n \quad (2.69)$$

e

$$d_s = \begin{bmatrix} d_{sD} \\ d_{sR} \\ d_{sK} \end{bmatrix} \quad (2.70)$$

Descrevendo a tarefa a ser realizada através dos valores desejados de força ($\lambda_{R,des}$), de movimento nas direções dinâmicas ($\ddot{s}_{D,des}$) e cinemáticas ($\ddot{s}_{K,des}$):

$$\tau_{des} = \begin{bmatrix} \ddot{s}_{D,des} \\ \lambda_{R,des} \\ \ddot{s}_{K,des} \end{bmatrix} \quad (2.71)$$

os torques de entrada que atendem a esta especificação são imediatamente derivados de (2.67), de forma que:

$$\tau = (JM^{-1})^\# (Q_s(\tau_{des} - d_s) - m_s) \quad (2.72)$$

onde

$$Q_s = \begin{bmatrix} \Upsilon_D + JM^{-1}J^TY_A(\Upsilon_D^TY_A)^{-1}M_E & JM^{-1}J^TY_R & \Upsilon_K \end{bmatrix} \quad (2.73)$$

e

$$m_s = -\dot{\Upsilon}_K \dot{s}_K - \dot{\Upsilon}_D \dot{s}_D - JM^{-1}J^TY_A(\Upsilon_D^TY_A)^{-1}n_E + \dot{J}\dot{q} - JM^{-1}n \quad (2.74)$$

Da mesma forma que na seção anterior é necessário fazer uma aproximação para os termos M_E, n_E correspondentes ao meio.

2.6 Conclusões do Capítulo

Neste capítulo foi apresentado o modelo matemático de De Luca e Manes, para robôs submetidos tanto a restrições cinemáticas quanto dinâmicas ou, ainda para o caso livre de restrições. Observou-se que com esta modelagem é possível representar uma grande variedade de tarefas com diferentes tipos de restrições podendo, ainda, ampliar a formulação para contemplar meios elásticos (Amaral *et al.*, 2000b).

Verificou-se que esta nova modelagem é um caso geral dos dois tipos de restrições, podendo deles ser derivado o modelo clássico do robô para controle de força e/ou posição, assim como o modelo no qual somente são consideradas as componentes dinâmicas.

Foi feita a compensação do modelo não-linear do robô através da dinâmica inversa, possibilitando a proposta e aplicação de diferentes técnicas de controle, visando o desempenho, a compensação de atrito e a rejeição de perturbações.

Finalmente, do ponto de vista prático observa-se a necessidade de expressar as variáveis $s_K, s_D, \lambda_R, \lambda_A$ em termos das variáveis lidas pelos sensores do robô. Este aspecto será retomado nos capítulos seguintes.

Capítulo 3

Modelagem e Compensação do Atrito

O fenômeno do *atrito* está presente em todos os sistemas mecânicos, sejam estes sistemas de transmissão, cilindros pneumáticos ou hidráulicos, válvulas, volantes etc. Esse fenômeno aparece no contato físico entre duas superfícies.

O fenômeno do atrito é determinante no desempenho e na estabilidade de diferentes sistemas de controle. Sua característica altamente não-linear pode ocasionar erros em regime, ciclos limites, e movimentos indesejados como o adere-desliza (*stick-slip*) (Olsson e Aström, 1996) (Cao e Schwartz, 2000), tendo como imediata consequência a deterioração do desempenho do sistema que está sendo controlado.

Neste capítulo os modelos de atrito *estáticos* para médias e altas velocidades e dinâmicos para baixas velocidades são apresentados. O modelo de *LuGre* (de Wit *et al.*, 1995) é descrito em detalhe, são estabelecidas suas características para a determinação e identificação de seus parâmetros e paralelamente descritas as estruturas de controle empregadas para sua compensação.

3.1 Características do Atrito

O atrito é a força tangente de reação entre duas superfícies de contato. Fisicamente, esta força depende da geometria de contato, da topologia, do tamanho das superfícies, de suas propriedades, da velocidade relativa entre elas e da presença ou não de lubrificação. Estas e outras características como a deformação elástica ou plástica entre as superfícies em contato evidenciam a dificuldade para obter modelos matemáticos que possam descrever adequadamente este fenômeno (Olsson *et al.*, 1997).

O comportamento do *atrito* em condições nas quais supõem-se superfícies de contato limpas e velocidade constante foi amplamente abordado por diversos autores (Årmstrong *et al.*, 1994). Posteriormente, tendo-se evidenciado características dinâmicas, foram propostos e apresentados outros

modelos (de Wit *et al.*, 1995). Esses modelos, ao contrário dos estáticos, consideram diversos aspectos para a descrição do atrito conforme será visto a seguir:

1. A característica *estática* do *atrito*, que corresponde à força de rompimento necessária para o início do deslizamento seguido de movimento, como ilustrado na figura 3.1;

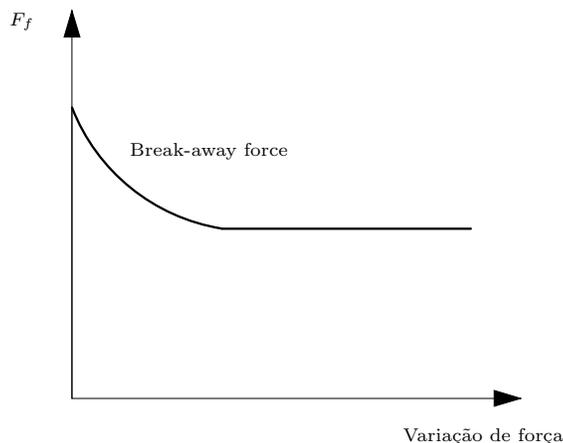


Figura 3.1: Força de quebra

2. Sendo considerados movimentos numa única direção, observou-se um retardo entre a mudança na velocidade e a correspondente mudança na força de atrito. Este fenômeno é ilustrado na figura 3.2;

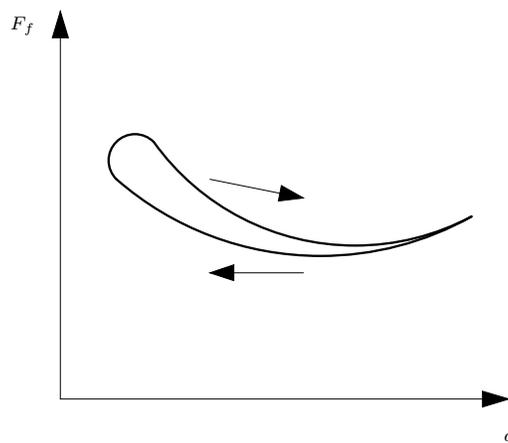


Figura 3.2: Retardo na variação da velocidade e a força de atrito

3. Outra característica dinâmica corresponde ao pré-deslizamento (*pre-sliding displacement*), descrito em (de Wit *et al.*, 1995), tal que se uma força menor à força de quebra (*break-away force*) é aplicada entre as superfícies em contato, pode-se ter deslocamentos na ordem de micrômetros. Este deslocamento é elástico com histerese, conforme ilustra a figura 3.3.

Visando o problema da compensação do atrito e levando em conta as características mencionadas,

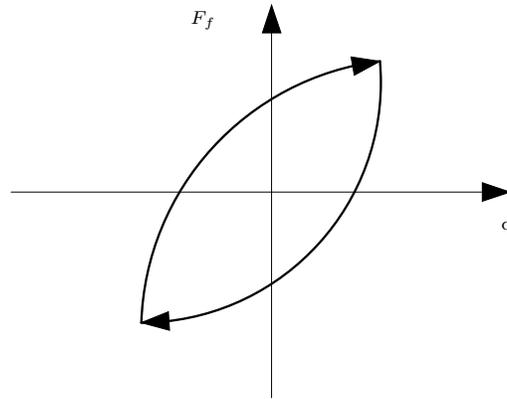


Figura 3.3: Deslocamento de pré-deslizamento

faz-se, na próxima seção a descrição matemática tanto dos modelos estáticos quanto dos dinâmicos.

3.2 Modelagem do Atrito

Um modelo de atrito é necessário para que possam ser considerados seus efeitos e para sua compensação nos sistemas de controle. Neste sentido, diferentes modelos de atrito foram publicados e empregados nas duas últimas décadas (de Wit *et al.*, 1991), (Årmstrong *et al.*, 1994). O estudo de alguns destes modelos é abordado a seguir.

3.2.1 Modelos estáticos

Os modelos estáticos são assim denominados por não incluírem equações diferenciais. Os modelos estáticos de atrito normalmente são representados pela relação entre a velocidade e a força de atrito propriamente dita. Desta forma, considerando-se que o *atrito* se opõe à direção do movimento e que sua magnitude varia em forma constante com a velocidade, uma possível formulação matemática é dada por:

$$F_f = F_c \operatorname{sgn}(\dot{q}) \quad (3.1)$$

onde a força F_c se opõe à direção do movimento e é proporcional à força normal F_N ($F_c = \mu F_N$). Este modelo corresponde ao atrito de *Coulomb*. Na figura 3.4(a), pode-se observar que a relação entre a força de atrito e a velocidade é constante, e que não é definida para $\dot{q} = 0$. Esta característica constitui-se numa desvantagem desta representação. Sua forma matricial é apresentada na equação (3.2).

$$F_c = \begin{bmatrix} f_{c_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & f_{c_n} \end{bmatrix} \quad (3.2)$$

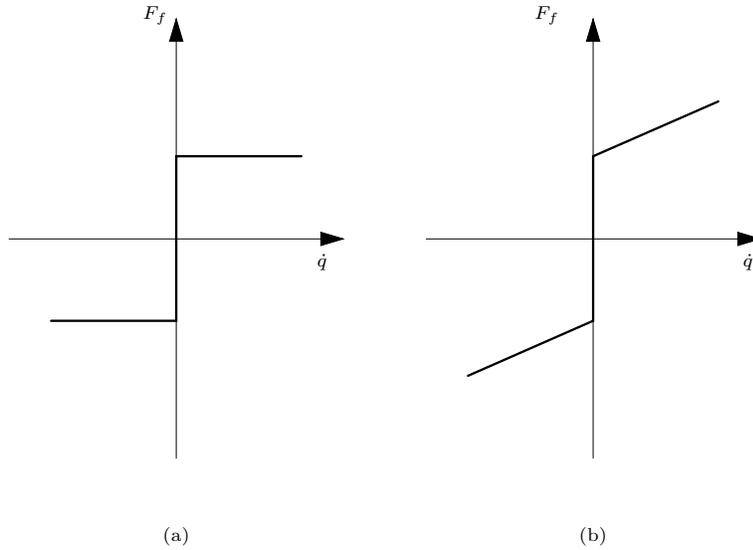


Figura 3.4: Atrito de Coulomb e atrito viscoso

onde os termos f_{c_i} representam os coeficientes de atrito associados aos diferentes eixos num sistema multi-variável. No caso dos robôs manipuladores, $f_{c_i}, i = 1, 2, 3, \dots, n$ representam o atrito em cada uma das juntas $[q_1, \dots, q_n]$.

Estudos de hidrodinâmica permitiram identificar uma forma de atrito atribuída à viscosidade nos sistemas com lubrificação (Olsson *et al.*, 1997). O termo *atrito viscoso* foi empregado para denominar esta componente que é dada pela seguinte equação:

$$F_f = F_v \dot{q} \quad (3.3)$$

onde F_v é um coeficiente associado às diferentes superfícies.

Na equação (3.3), pode-se observar a relação proporcional entre a força de atrito e a velocidade. A representação matricial da equação (3.3) para um sistema multi-variável é feita da mesma forma que na equação (3.2) (Jaritz e Spong, 1996). A combinação do atrito de *Coulomb* com o atrito *viscoso* é ilustrada na figura 3.4(b).

Posteriormente, especificou-se outra característica do atrito, sendo que, em um curto período de deslizamento, existe uma componente de magnitude maior à magnitude do atrito de *Coulomb*. Esta componente é chamada de *atrito estático* (Stiction) e caracteriza-se não apenas por ser dependente da velocidade mas também de uma força externa F_e , de tal forma que:

$$F_f = \begin{cases} F_e & \text{se } \dot{q} = 0 \text{ e } |F_e| < F_s \\ F_s \text{sgn}(F_e) & \text{se } \dot{q} = 0 \text{ e } |F_e| \geq F_s \end{cases} \quad (3.4)$$

onde F_s corresponde ao limite da força estática.

Na figura 3.5(a) são ilustradas estas três componentes e pode-se observar que as mesmas são

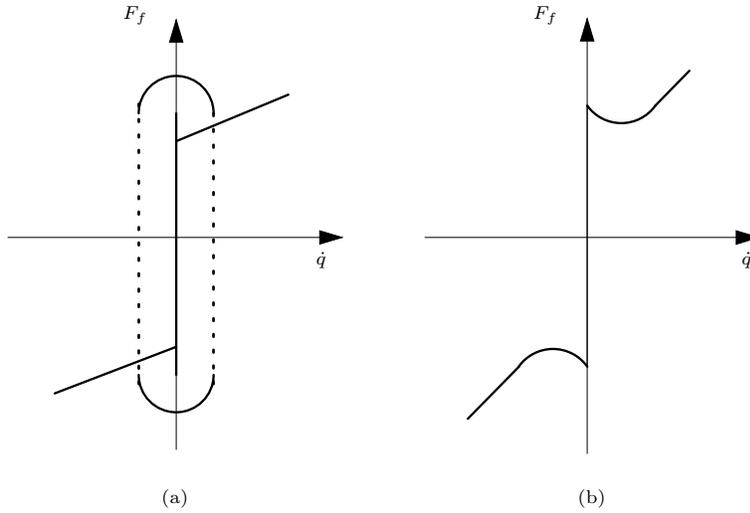


Figura 3.5: Atrito de Coulomb, viscoso, estático e a velocidade de *Stribeck*

constantes ou variam linearmente com a velocidade \dot{q} . A variação não-linear do atrito em baixas velocidades é chamado de efeito *Stribeck*, e corresponde a uma ampliação da área assinalada na figura 3.5(a) e ilustrada na figura 3.5(b).

Considerando-se todas as componentes mencionadas, uma descrição geral do modelo estático do atrito é (Olsson *et al.*, 1997):

$$F_f = \begin{cases} F(\dot{q}) & \text{se } \dot{q} \neq 0 \\ F_e & \text{se } \dot{q} = 0 \text{ e } |F_e| < F_s \\ F_s \text{sgn}(F_e) & \text{de outra forma} \end{cases} \quad (3.5)$$

onde $F(\dot{q})$ é uma função arbitrária cuja representação é dada pela equação a seguir:

$$F(\dot{q}) = (F_c + (F_s - F_c)e^{-\left(\frac{\dot{q}}{v_0}\right)^\delta}) \text{sgn}(\dot{q}) + F_v \dot{q} \quad (3.6)$$

onde v_0 corresponde à velocidade de *Stribeck* e a constante δ depende da geometria de contato.

A equação (3.6) é chamada de modelo *exponencial* e é bastante empregada para a compensação do atrito. Algumas aplicações podem ser encontradas em (de Wit *et al.*, 1991), (Ramirez *et al.*, 2000a), (Gandhi *et al.*, 2002), (Zhu e Pagilla, 2002), (Vargas *et al.*, 2004e).

3.3 Modelos Dinâmicos

Foi evidenciado que os modelos estáticos não descrevem as características dinâmicas do atrito. Para sua inclusão, os modelos dinâmicos incorporam um ou mais estados internos, representando de forma mais completa este fenômeno.

Características dinâmicas como o pré-deslizamento (pre-sliding displacement) e a força de quebra

foram considerados nos trabalhos publicados por *Dahl*, *Bliman/Sorine* e *LuGre* (Olsson *et al.*, 1997).

3.3.1 Modelo de *Dahl*

No modelo de *Dahl*, são considerados o atrito de *Coulomb*, e efeitos dinâmicos como o pré-deslizamento (Olsson *et al.*, 1997). Desta forma, a força de atrito é modelada em função do deslocamento q , através da seguinte equação:

$$\frac{dF_f}{dq} = \sigma \left(1 - \frac{F_f}{F_c} \operatorname{sgn}(\dot{q}) \right)^\alpha \quad (3.7)$$

onde σ corresponde ao parâmetro de rigidez, F_c é o atrito de *Coulomb*, α é denominado fator de forma e \dot{q} é a velocidade.

Derivando a equação (3.7) em relação ao tempo e considerando o fator $\alpha = 1$, tem-se que:

$$\frac{dF_f}{dt} = \sigma \left(\dot{q} - \frac{F_f |\dot{q}|}{F_c} \right) \quad (3.8)$$

Levando em conta a equação (3.8), pode-se re-escrever o modelo de *Dahl* na seguinte forma:

$$\begin{aligned} \frac{dz}{dt} &= \dot{q} - \sigma \frac{|\dot{q}|}{F_c} z \\ F_f &= \sigma z \end{aligned} \quad (3.9)$$

onde F_f corresponde à força de atrito e z é um estado interno não mensurável. Finalmente, é possível verificar que este modelo não considera os efeitos *Stribeck* e *Stiction*.

3.3.2 Modelo de *Bliman/Sorine*

Este modelo foi baseado nos estudos experimentais desenvolvidos por *Rabinowicz* (Olsson *et al.*, 1997). O modelo considera que a magnitude da força de atrito depende da função $\operatorname{sgn}(\dot{q})$ e da variável S , definida como:

$$S = \int_0^t |\dot{q}(\tau)| d\tau \quad (3.10)$$

Empregando-se a representação de estados, este modelo de atrito é descrito por:

$$\begin{aligned} \frac{dx_S}{dS} &= Ax_S + B\dot{q}_S \\ F_f &= Cx_S \end{aligned} \quad (3.11)$$

A equação (3.11) pode ser empregada com diferentes graus de complexidade, sendo que o modelo de primeira ordem é obtido utilizando os seguintes coeficientes:

$$A = \frac{-1}{\varepsilon_f}, \quad B = \frac{f_1}{\varepsilon_f}, \quad C = 1 \quad (3.12)$$

Usando estas definições o modelo será dado por:

$$\frac{dF_f}{dt} = \frac{dF_f}{dS} \frac{dS}{dt} = |\dot{q}| \frac{dF_f}{ds} = \frac{f_1}{\varepsilon_f} \left(\dot{q} - |\dot{q}| \frac{F_f}{f_1} \right) \quad (3.13)$$

que é igual ao modelo de *Dahl* se $f_1 = F_c$, $\sigma = \frac{f_1}{\varepsilon_f}$ e $\alpha = 1$. Tal modelo não inclui também o fenômeno de *Stiction*. Por outro lado, o modelo de segunda ordem inclui esta característica assim como o pré-deslizamento (Olsson *et al.*, 1997).

3.3.3 Modelo de *LuGre* (*Lund e Grenoble*)

O modelo de *LuGre* é uma extensão do modelo de *Dahl* e foi proposto por de Wit *et al.* (1995). Esta modelagem inclui as características estáticas e dinâmicas do atrito. Sendo assim, estima-se que o modelo de *LuGre* seja o mais completo.

O modelo de *LuGre* baseia-se numa análise microscópica do fenômeno do atrito, introduzindo-se uma variável não mensurável z (semelhante à equação (3.9)). Esta variável descreve a deformação média relativa (microscópica) das superfícies de contato que é descrita por:

$$\frac{dz}{dt} = \dot{q} - \sigma_0 \frac{|\dot{q}|}{g(\dot{q})} z \quad (3.14)$$

onde a função $g(\dot{q})$ é positiva e depende de fatores como a lubrificação, as propriedades do material e a temperatura. A força de atrito é descrita como:

$$F_f = \sigma_0 z + \sigma_1 \frac{dz}{dt} + \alpha_2 \dot{q} \quad (3.15)$$

onde os parâmetros σ_0 e σ_1 são termos associados à rigidez e ao amortecimento da variável interna z e α_2 é o coeficiente de atrito viscoso.

O modelo de *LuGre*, dado pelas equações (3.14) e (3.15), é caracterizado pela função $g(\dot{q})$ e pelos parâmetros σ_0 , σ_1 e α_2 . Para considerar o efeito de *Stribeck*, a função $g(\dot{q})$ é definida como:

$$g(\dot{q}) = \alpha_0 + \alpha_1 e^{-\left(\frac{\dot{q}}{v_0}\right)^2} \quad (3.16)$$

onde o termo $g(\dot{q})$ ($\infty > \alpha_0 + \alpha_1 \geq g(\dot{q}) \geq \alpha_0 > 0$) descreve as características do modelo em regime permanente e inclui o parâmetro v_0 da velocidade de *Stribeck*. O atrito estático é representado pelos parâmetros $(\alpha_0 + \alpha_1)$ e o atrito de *Coulomb*, pelo parâmetro α_0 . A equação (3.16) em regime permanente descreve o modelo *exponencial* apresentado na seção anterior.

Pode-se observar através das equações (3.14), (3.15) e (3.16) que o modelo de *LuGre* é caracterizado por 4 parâmetros estáticos (α_0 , α_1 , α_2 , v_0) e 2 parâmetros dinâmicos (σ_0 , σ_1).

3.4 Compensação do Atrito

Nesta seção, são apresentadas algumas estruturas de controle para a implementação experimental da compensação de atrito utilizando o modelo de *LuGre*.

Visando a compensação do atrito, em (de Wit *et al.*, 1995) foram propostas, inicialmente, duas estruturas de controle (observadores). O primeiro considera o erro de posição e o segundo o erro de velocidade. As duas estruturas são ilustradas nas figuras 3.6 e 3.7 respectivamente.

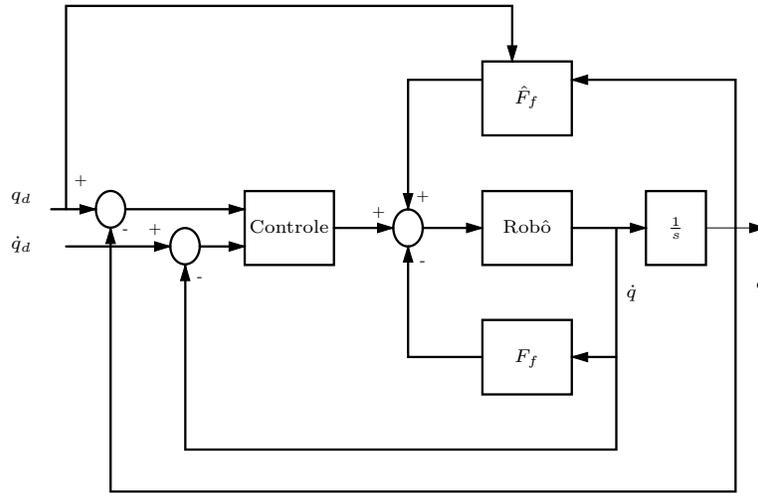


Figura 3.6: Observador de atrito baseado no erro de posição

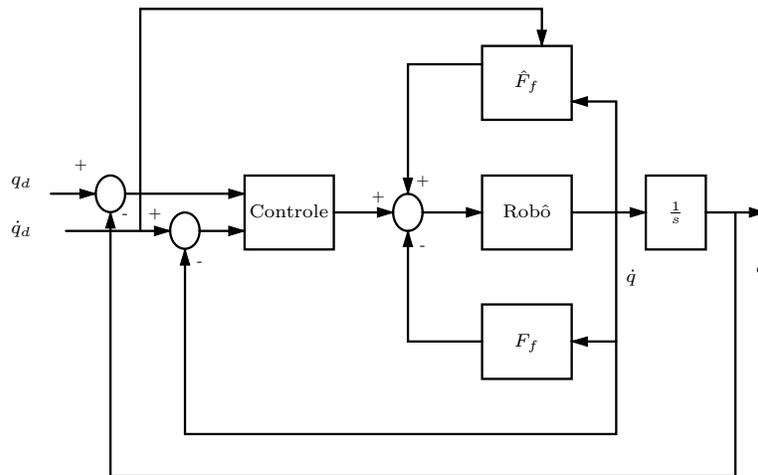


Figura 3.7: Observador de atrito baseado no erro de velocidade

O conjunto de equações para o observador baseado no erro de posição é dado por:

$$\begin{aligned}\frac{d\hat{z}}{dt} &= \dot{q} - \sigma_0 \frac{|\dot{q}|}{g(\dot{q})} \hat{z} - \kappa(q - q_d) \\ \hat{F}_f &= \sigma_0 \hat{z} + \sigma_1 \frac{d\hat{z}}{dt} + \alpha_2 \dot{q}\end{aligned}\quad (3.17)$$

onde a constante $\kappa > 0$ e $(q - q_d)$ corresponde ao erro de posição. Da mesma forma, a equação do observador baseado no erro de velocidade é dado por:

$$\begin{aligned}\frac{d\hat{z}}{dt} &= \dot{q} - \sigma_0 \frac{|\dot{q}|}{g(\dot{q})} \hat{z} - \kappa(\dot{q} - \dot{q}_d) \\ \hat{F}_f &= \sigma_0 \hat{z} + \sigma_1 \frac{d\hat{z}}{dt} + \alpha_2 \dot{q}\end{aligned}\quad (3.18)$$

onde $(\dot{q} - \dot{q}_d)$ corresponde ao erro de velocidade.

A estrutura apresentada na figura 3.8 ilustra uma modificação no modelo anterior, também proposta em (de Wit *et al.*, 1996).

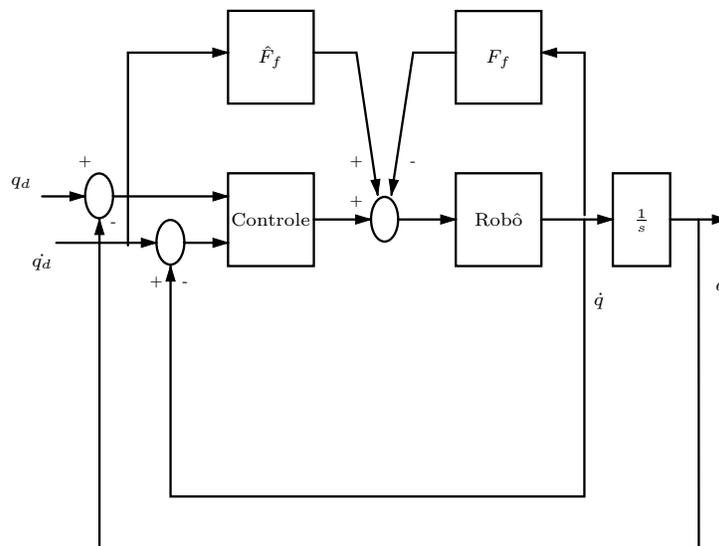


Figura 3.8: Observador de atrito baseado na velocidade desejada

Para esta estrutura a equação do observador é reescrita como:

$$\begin{aligned}\frac{d\hat{z}}{dt} &= \dot{q} - \sigma_0 \frac{|\dot{q}|}{g(\dot{q})} \hat{z} - \kappa \dot{q}_d \\ \hat{F}_f &= \sigma_0 \hat{z} + \sigma_1 \frac{d\hat{z}}{dt} + \alpha_2 \dot{q}\end{aligned}\quad (3.19)$$

Baseados nestas estruturas, Perondi (2002), Golin (2002) e Ramirez (2003) fizeram modificações considerando a soma dos erros de posição e velocidade. Nesta tese, foi empregada a estrutura apresentada na figura 3.7 que considera o erro de velocidade. Os detalhes da implementação serão feitos nos

próximos capítulos. As análises de estabilidade destas estruturas de controle podem ser encontradas em (de Wit *et al.*, 1995), (de Wit, 1998).

3.5 Determinação dos Parâmetros do Modelo de *LuGre*

A principal dificuldade na utilização do modelo de *LuGre* para a compensação do atrito relaciona-se à identificação dos parâmetros estáticos e dinâmicos. Este processo, assim como sua implementação em tempo real, introduzem um alto grau de complexidade em decorrência do modelo não-linear com uma variável interna não mensurável (z). Por isso, a abordagem normalmente considerada consiste em determinar primeiramente os coeficientes estáticos para, posteriormente, definir os coeficientes dinâmicos tal como sugerido em (P. Lischinsky e Morel, 1997).

3.5.1 Determinação dos parâmetros estáticos

Os parâmetros estáticos do modelo de *LuGre* α_0 , α_1 , α_2 e v_0 podem ser obtidos a partir do diagrama de atrito estático, considerando movimentos a velocidade constante. Em regime permanente, com as variáveis \dot{z} e \ddot{q} iguais a zero, a força de atrito é dada por:

$$F_{ss} = \left(\alpha_0 + \alpha_1 e^{-\left(\frac{\dot{q}}{v_0}\right)^2} \right) \text{sgn}(\dot{q}) + \alpha_2 \dot{q} \quad (3.20)$$

Dada a equação (3.20), é necessário estabelecer aproximações na modelagem de cada junta do robô para obter o respectivo torque de atrito. Na seqüência são feitos experimentos em malha fechada conforme ilustrado na figura 3.9.

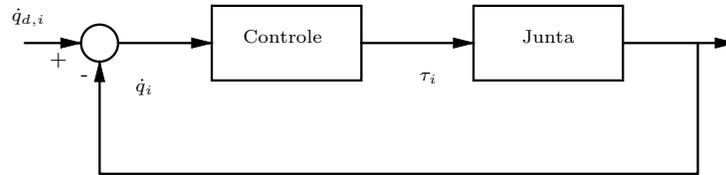


Figura 3.9: Diagrama de controle para a obtenção dos parâmetros estáticos

Para a realização destes experimentos são estabelecidas diferentes referências de velocidade. O torque de atrito é medido quando a velocidade torna-se constante. Dados experimentais de velocidade e torque para um conjunto de 20 amostras são apresentados na tabela 3.1.

Com dados similares aos apresentados na tabela 3.1 para diferentes velocidades (positivas e negativas), é possível obter a curva de atrito estático. Finalmente, empregando técnicas de otimização para a resolução da equação (3.20), são determinados os parâmetros estáticos do modelo de *LuGre* (P. Lischinsky e Morel, 1997), (Ramirez, 2003).

Uma técnica de otimização aplicada é a definida pela seguinte equação:

Tabela 3.1: Valores de velocidade e torque

<i>velocidade</i> [rad/s]	<i>torque</i> [Nm]
-0.0115	-4.4830
-0.0115	-5.4696
-0.0077	-5.8157
-0.0077	-5.7617
-0.0115	-6.6694
-0.0115	-6.0434
-0.0115	-6.5948
-0.0115	-6.1723
-0.0077	-6.0590
-0.0077	-5.5472
-0.0115	-5.5570
-0.0115	-5.7303
-0.0115	-5.9354
-0.0077	-5.6671
-0.0115	-5.2626
-0.0115	-4.8725
-0.0115	-5.3687
-0.0114	-5.3021
-0.0116	-5.7820
-0.0076	-5.6534
-0.0114	-5.9339

$$\min_X \sum_{i=1}^N (F(X, \dot{q}) - \tau)^2 \quad (3.21)$$

onde $X = \alpha_0, \alpha_1, \alpha_2, v_0$ e N corresponde ao número de amostras dos dados experimentais obtidos a velocidade constante. Uma outra forma de determinar estes parâmetros é estabelecer aproximações por inspeção (Golin, 2002), (Perondi, 2002).

3.5.2 Determinação dos parâmetros dinâmicos

A obtenção dos parâmetros dinâmicos não é possível através do uso de técnicas lineares, já que a variável interna z tem uma relação não-linear com os parâmetros que devem ser identificados. Além disto, a variável z não é uma variável que pode ser medida.

No entanto, estes parâmetros podem ser obtidos utilizando algumas aproximações. A proposta de P. Lischinsky e Morel (1997) foi utilizada neste trabalho. Para determinar σ_0 , são realizados experimentos com pequenos deslocamentos tal que \ddot{q} e \dot{z} podem ser considerados nulos. O parâmetro σ_1 pode ser obtido utilizando uma aproximação linear do sistema, fazendo $\dot{q} \approx 0$ e $z \approx 0$. Com estas considerações e supondo que a variável z é igual ao deslocamento angular q , os parâmetros são determinados a partir de:

$$\sigma_0 \approx \frac{\Delta\tau}{\Delta q} \quad (3.22)$$

$$\sigma_1 \approx 2\xi\sqrt{\sigma_0\bar{m}} - \alpha_2 \quad (3.23)$$

onde \bar{m} corresponde ao valor máximo do termo inercial do sistema em questão.

Finalmente, assim como nos parâmetros estáticos, técnicas de otimização podem ser empregadas (Ramirez, 2003). No capítulo 6, é detalhado o procedimento para a obtenção dos parâmetros estáticos e dinâmicos do modelo do atrito.

3.6 Conclusões do Capítulo

Neste capítulo foram apresentadas as principais características do fenômeno do atrito assim como alguns de seus efeitos nos sistemas de controle. Os modelos estáticos e os modelos dinâmicos foram detalhados, dando uma especial atenção ao modelo de *LuGre*, por considerar que este modelo é o mais completo. Foram apresentadas as principais estruturas de controle para a compensação deste fenômeno empregando a modelagem de *LuGre* e introduzidos os passos para a determinação dos parâmetros estáticos e dinâmicos neste modelo. Para a determinação dos parâmetros estáticos são considerados experimentos em malha fechada para diferentes velocidades. Nestes mesmos experimentos é utilizada a condição inicial de movimento para determinar os parâmetros dinâmicos, a partir de algumas aproximações. Do ponto de vista prático, é importante mencionar que esta modelagem apresenta uma dificuldade suplementar devido ao fato do estado interno z não estar disponível para ser medido.

Capítulo 4

Estratégias de Controle

O desenvolvimento e projeto de sistemas de controle requer estabilidade e um desempenho adequado do sistema em malha fechada, frente às perturbações, ruídos e incertezas (paramétricas e não paramétricas). Dentre os diferentes objetivos que podem ser definidos para o desenvolvimento de sistemas de controle, um que tem merecido grande atenção é o problema da robustez, principalmente nos trabalhos desenvolvidos a partir das décadas de 70 e 80.

Na década de 80, o controle na estrutura de **Dois Graus de Liberdade** $TDOF$ ¹ (Umeno e Hori, 1991) foi proposto por vários autores, visando associar ao controle clássico requisitos de robustez e de desempenho. Uma década antes faz seu aparecimento no ocidente a técnica de controle não-linear denominada de **Controle a Estrutura Variável** VSC ² (Utkin, 1977). Quase simultaneamente eram feitas as primeiras publicações sobre lógica nebulosa e sua aplicação nos sistemas de controle dando origem aos **Controladores Nebulosos** FLC ³ (Zadeh, 1965).

As três técnicas surgiram com o objetivo de tratar as demandas de robustez e de desempenho, em virtude de que muitos controladores foram desenvolvidos sem levar em conta estas características, ficando restritos ao controle de sistemas perfeitamente conhecidos ou com pequenas variações.

Visando tratar o problema de desempenho e de robustez, o presente capítulo é dedicado à aplicação destas técnicas ao controle de força e posição de robôs manipuladores.

4.1 Controlador de Dois Graus de Liberdade TDOF

A rejeição das perturbações externas e a robustez do sistema de controle às variações e incertezas paramétricas são importantes para garantir boas características de resposta e devem ser preocupações no projeto de controladores.

Dessas considerações, e do ponto de vista da teoria de controle, os controladores de *Dois Graus de*

¹Two Degree of Freedom

²Variable Structure Control

³Fuzzy Logic Controller

Liberdade (TDOF) permitem a integração de várias estratégias de controle robusto (Wolovich, 1995). Os trabalhos desenvolvidos em (Umeno e Hori, 1991), (Umeno *et al.*, 1993), (Tadeo *et al.*, 1998), (Bisso, 1999), (Vargas *et al.*, 2000) e (Vargas *et al.*, 2001b) são exemplos de aplicações desta técnica no controle de posição e força em robôs manipuladores.

Ao longo desta tese, a análise, o projeto e a implementação dos controladores de Dois Graus de Liberdade foram feitos empregando-se o índice LQR ⁴ que permite, de forma simultânea, a alocação ótima de pólos e a obtenção de erro nulo em regime $e_{ss}(t) = 0$. Ao mesmo tempo, são estabelecidas condições de estabilidade, atenuação de ruídos e a rejeição de perturbações.

4.1.1 Estrutura de controle

A estrutura do controle *TDOF* empregada para a análise e projeto é ilustrada na figura 4.1. A mesma permite a síntese do controlador considerando, de forma independente, os objetivos de malha e resposta (Wolovich, 1995).

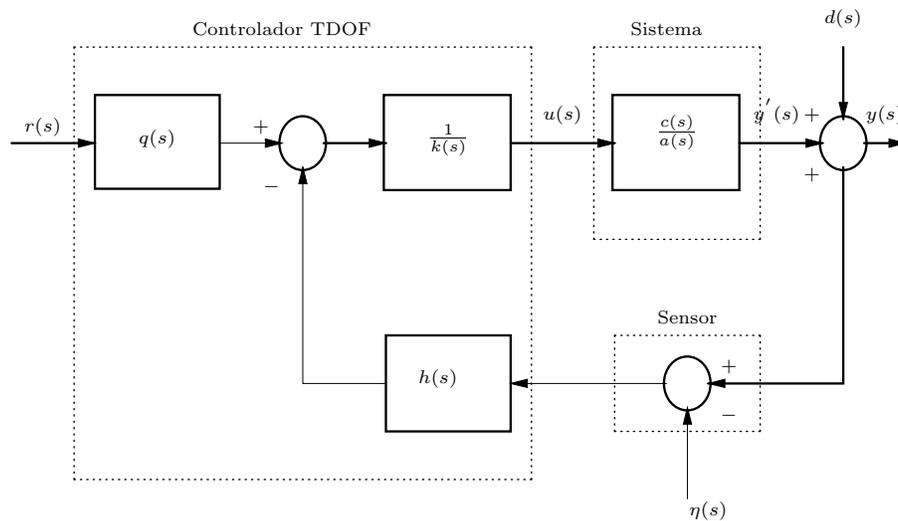


Figura 4.1: Controlador de Dois Graus de Liberdade - *TDOF*

Na estrutura de controle apresentada na figura 4.1, tem-se:

- a função de transferência racional e estritamente própria, que representa o comportamento dinâmico do sistema, é dada por:

$$G(s) = \frac{y'(s)}{u(s)} = \frac{c(s)}{a(s)} \quad (4.1)$$

sendo $a(s)$ e $c(s)$ polinômios coprimos (realização controlável e observável). O grau do polinômio

⁴Linear Quadratic Regulator

$a(s)$ é n e o grau do polinômio $c(s)$ é estritamente menor que n :

$$\text{grau}[c(s)] < n = \text{grau}[a(s)] \quad (4.2)$$

- as perturbações externas, com propriedades conhecidas, que atuam sobre o sistema podem ser representadas como um sinal $d(s)$, adicionado à sua saída, assim como o ruído $\eta(s)$ que afeta a saída medida.
- o controlador pode ser representado por um sistema linear invariante no tempo cujo comportamento dinâmico é descrito pela seguinte matriz de transferência:

$$u(s) = \frac{1}{k(s)} [q(s) \quad -h(s)] \begin{bmatrix} r(s) \\ y(s) - \eta(s) \end{bmatrix} \quad (4.3)$$

- a ordem do controlador é determinada pela ordem do polinômio $k(s)$; para evitar problemas de amplificação de ruído de alta frequência e saturação da entrada do sistema, supõe-se que:

$$\text{grau}[q(s) - h(s)] \leq \text{grau}[k(s)] \quad (4.4)$$

O projeto do controlador é constituído basicamente do cálculo dos polinômios $q(s)$, $h(s)$ e $k(s)$.

4.1.2 Definições e propriedades

Os pólos de malha fechada da função $\frac{y(s)}{r(s)}$ são dados pelas raízes da equação:

$$\delta(s) = a(s)k(s) + c(s)h(s) = 0 \quad (4.5)$$

Deduz-se de (4.2) e (4.4) que o grau do polinômio característico $\delta(s)$ é:

$$\text{grau}[\delta(s)] = \text{grau}[a(s)k(s)] = n + \text{grau}[k(s)] \quad (4.6)$$

A escolha dos polinômios $h(s)$, $q(s)$ e $k(s)$ deve garantir, além de estabilidade nominal em malha fechada, um comportamento temporal adequado e propriedades de robustez tais como a rejeição de perturbações e a atenuação do ruído.

Neste sentido, para a análise e o projeto deste controlador são definidas as seguintes funções de transferência (Wolovich, 1995).

- *Resposta de saída:*

$$T(s) = \frac{y(s)}{r(s)} = \frac{c(s)q(s)}{\delta(s)} \quad (4.7)$$

- *Sensibilidade:*

$$S(s) = \frac{y(s)}{d(s)} = \frac{a(s)k(s)}{\delta(s)} \quad (4.8)$$

- *Complementar de sensibilidade:*

$$C(s) = \frac{y(s)}{\eta(s)} = \frac{c(s)h(s)}{\delta(s)} \quad (4.9)$$

Neste ponto, é importante observar que as funções de sensibilidade e complementar de sensibilidade comportam-se como um filtro passa alta e um filtro passa baixa, respectivamente, e verificam a seguinte igualdade:

$$C(s) = 1 - S(s) \quad (4.10)$$

- *Resposta do erro:*

$$\frac{e(s)}{r(s)} = \frac{\overbrace{a(s)k(s) + c(s)[h(s) - q(s)]}^{m_e(s)}}{\delta(s)} \quad (4.11)$$

que reflete o efeito da referência sobre o erro $e(t) = r(t) - y(t)$ e também permite estabelecer condições para a obtenção de erro nulo em regime,

$$\lim_{t \rightarrow \infty} e(t) = 0 \quad (4.12)$$

- O ganho de malha $L(s)$ é definido como o produto das funções de transferência ao redor da malha completa de realimentação (ganho líquido), isto é:

$$L(s) = \frac{c(s)h(s)}{a(s)k(s)} = \frac{\text{zeros em malha aberta}}{\text{pólos em malha aberta}} \quad (4.13)$$

O ganho de malha e a função de sensibilidade permitem uma avaliação adequada do projeto dos controladores *TDOF*. Ambas funções estão diretamente relacionadas pela seguinte expressão:

$$S(s) = \frac{1}{1 + L(s)} = \frac{a(s)k(s)}{\delta(s)} \quad (4.14)$$

Os objetivos de malha e resposta a serem considerados no projeto dos controladores são apresentados a seguir.

4.1.3 Objetivos de malha

O objetivos de malha mais importantes são a estabilidade robusta, a rejeição de perturbações e a atenuação de ruído. Isto é manter os pólos de malha fechada da função (4.7) no semiplano estável $\Re(s) < 0$ independentemente das incertezas presentes no sistema.

As incertezas no sistema podem ser produzidas pelas variações paramétricas, não-linearidades tais como a histerese, o atrito, as zonas mortas e as dinâmicas não-modeladas.

Estabilidade robusta em relação às variações paramétricas

Para assegurar a estabilidade robusta, os polinômios $h(s)$ e $k(s)$ devem, necessariamente, ser escolhidos para assegurar a estabilidade nominal de malha fechada, tal que as raízes de $\delta(s)$ estejam no semiplano esquerdo.

A eficiência destes polinômios pode ser medida a partir das margens de ganho (GM) e de fase (ΦM), geralmente definidas para sistemas estáveis em malha fechada caracterizados por um ganho de malha $L(s)$ estável. Uma outra forma de determinar a estabilidade robusta em relação a variações paramétricas pode ser avaliada a partir da função de sensibilidade $S(s)$ dada pela equação (4.8).

Um valor limite da magnitude da função $|S(j\omega)|$ pode ser usado como medida da estabilidade robusta para sistemas estáveis em malha fechada e sistemas instáveis em malha aberta, onde a margem de ganho GM e a margem de fase ΦM não são bem definidas. Para garantir um projeto nominal estável com $GM \geq 2$ e $\Phi M \geq 30^\circ$, que são valores nominais aceitáveis de margens de ganho e de fase, a função de sensibilidade deve verificar:

$$\bar{S} \leq 2 \approx 6 \text{ dB} \quad \text{ou} \quad \bar{S}^{-1} = \min_{\omega} |1 + L(j\omega)| \geq 0.5 \quad (4.15)$$

onde \bar{S} é definida como:

$$\|S\|_{\infty} = \max_{\omega} |S(j\omega)| = \bar{S} \geq 1 \quad (4.16)$$

Da equação (4.15) pode-se observar que a norma $|S(j\omega)|$ terá que ser menor que 2 para garantir uma boa robustez frente às variações paramétricas. Além disso, esta condição pode substituir os requisitos de GM e ΦM em qualquer tipo de sistema (Wolovich, 1995).

Estabilidade robusta em relação a dinâmicas não modeladas

Considera-se que a função de transferência $G(s)$ é uma boa aproximação do processo real $\tilde{G}(s)$ para baixas e médias frequências. Para as altas frequências, as dinâmicas não-modeladas causam uma diferença na resposta em frequência entre o modelo aproximado e o modelo real. A seleção dos polinômios $h(s)$ e $k(s)$ deve também garantir estabilidade robusta com respeito a estas dinâmicas.

Em Wolovich (1995) são definidas as funções de transferência real do sistema $\tilde{G}(s)$, a função de transferência de ganho de malha em altas frequências $\tilde{L}(j\omega)$ e uma função de ponderação $W_c(s)$ para caracterizar as dinâmicas não-modeladas.

$$\tilde{L}(j\omega) = \tilde{G}(j\omega)H(j\omega) = L(j\omega)[1 + \Delta(j\omega)W_c(j\omega)] \quad (4.17)$$

Da análise feita em Wolovich (1995), pode-se concluir que:

$$|L(j\omega)| \ll |W_c(j\omega)|^{-1} \ll 1 \quad (4.18)$$

o que garante estabilidade robusta com respeito às dinâmicas não modeladas.

Rejeição da perturbação e atenuação do ruído

As funções de sensibilidade, $S(s)$, e complementar de sensibilidade, $C(s)$, representam o efeito da perturbação $d(s)$ e do ruído $\eta(s)$ sobre a saída $y(s)$, respectivamente. Um dos objetivos de malha é minimizar o efeito desses sinais na saída.

Em Umeno *et al.* (1993) e Wolovich (1995), são definidas as funções de transferência $W_s(s)$ e $W_\eta(s)$ como as funções de ponderação da perturbação e do ruído, respectivamente, sendo obtidas as seguintes condições:

$$|L(j\omega)| > |W_s(j\omega)| \gg 1 \quad (4.19)$$

que garante a rejeição da perturbação, e

$$|L(j\omega)| \ll |W_\eta(j\omega)|^{-1} \ll 1 \quad (4.20)$$

que garante a atenuação de ruído.

Desempenho robusto de malha

O diagrama de *Bode* da figura 4.2 caracteriza um ganho de malha nominal que satisfaz as condições de rejeição da perturbação (4.19), estabilidade robusta frente a dinâmicas não modeladas (4.18) e atenuação do ruído (4.20):

Se a distância entre $|W_s(j\omega)|$ e $|W_c(j\omega)|^{-1}$ da figura 4.2 é pequena, pode ser impossível atingir todos os objetivos de malha. Neste caso, é necessário satisfazer os objetivos mais importantes e uma série de compromissos tais como estabilidade, erro em regime, atenuação de ruídos ou a rejeição de perturbações deverão ser considerados.

4.1.4 Objetivos de resposta

Um dos objetivos de resposta é garantir um adequado comportamento *servo-regulatório*. Este comportamento é desejado tanto em regime transitório quanto em regime permanente.

O comportamento desejável da saída implica não somente em uma resposta transitória adequada, como também numa boa regulação obtida através da minimização do erro entre a saída e a referência

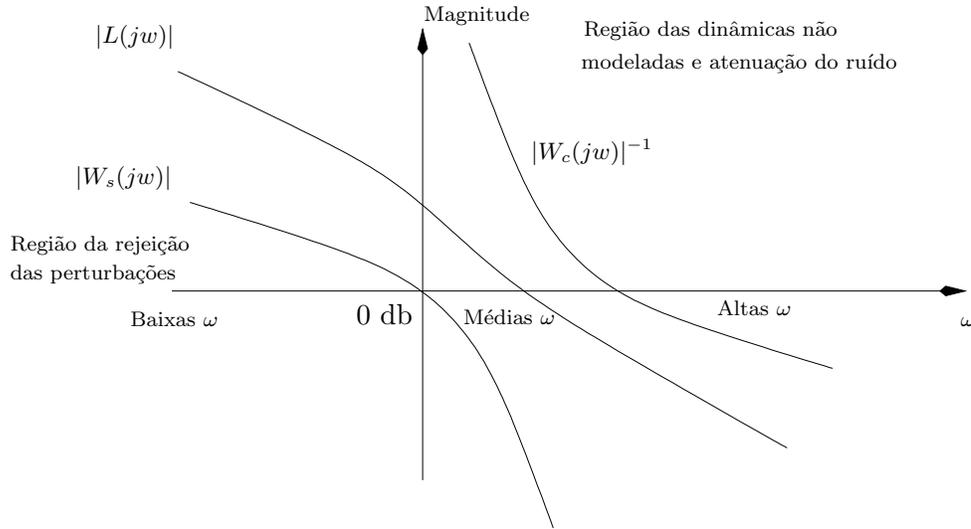


Figura 4.2: Diagrama de Bode de $|L(j\omega)|$ para desempenho robusto

(Wolovich, 1995). A referência é definida como:

$$r(s) = \mathcal{L}[r(t)] = \frac{m_r(s)}{p_r(s)} \quad (4.21)$$

onde os pólos de $p_r(s)$ encontram-se no semiplano esquerdo. Desta forma, a minimização do erro pode ser obtida a partir da função de transferência de resposta do erro apresentada na equação (4.11).

Substituindo a equação (4.21) na equação (4.11), pode-se mostrar que $e_{ss}(t) = 0$ é obtido se:

$$a(s)k(s) = \bar{m}_e(s)p_r(s) \quad (4.22)$$

$$h(s) - q(s) = \hat{h}(s)p_r(s) \quad (4.23)$$

para algum $\bar{m}_e(s) = \frac{m_e(s)}{p_r(s)}$ e $\hat{h}(s)$. A condição (4.22) é conhecida como o *princípio do modelo interno* (Wolovich, 1995). A equação (4.23) impõe restrições na escolha de $q(s)$, mas $k(s)$ pode ser sempre escolhida de tal forma a garantir a equação (4.22). As duas restrições são consideradas no projeto ótimo apresentado a seguir.

4.1.5 Projeto de controle ótimo com erro nulo em regime

Seja a seguinte função de transferência de resposta de saída:

$$T(s) = \frac{y(s)}{r(s)} = \frac{\alpha c(s)}{\hat{\delta}(s)} = \frac{\alpha c(s)\hat{q}(s)}{\hat{\delta}(s)\hat{q}(s)} \quad (4.24)$$

onde α é um escalar arbitrário e $\hat{\delta}(s)$ é um polinômio arbitrário estável e mônico com $\text{grau}[\hat{\delta}(s)] = \text{grau}[a(s)] = n$. As raízes de $\hat{\delta}(s)$ são os pólos desejados em malha fechada.

A determinação dos polinômios $\hat{\delta}(s)$ e $\hat{q}(s)$ pode ser feita empregando-se diferentes critérios. A

utilização do índice LQR , como descrito em Wolovich (1995), permite obter um desempenho de malha fechada adequado e erro nulo em regime ($e_{ss}(t) = 0$). Do ponto de vista temporal, a minimização do índice

$$J = \int_0^{\infty} \{\rho(y(t) - r(t))^2 + u(t)^2\} dt \quad (4.25)$$

permite ponderar o erro de seguimento $e(t)$ e o sinal de controle $u(t)$. Variando-se o valor de ρ , pode-se atingir a condição $u(t) \leq |M|$, assim como obter um comportamento em regime transitório e permanente aceitável (Wolovich, 1995), (Vargas *et al.*, 2000). Diferentes escolhas de $\hat{q}(s)$ permitem obter diferentes comportamentos de malha, para uma mesma resposta entrada-saída. A solução do problema de minimização do índice (4.25) no espaço de estado é a conhecida equação de *Ricatti*. Para abordagem no domínio da frequência, é empregada a fatorização espectral detalhada a seguir.

Fatorização espectral

Seja a função de transferência do sistema $G(s)$ racional, mínima e estritamente própria, definida como:

$$G(s) = \frac{c(s)}{a(s)} = \frac{c_m s^m + \dots c_1 s + c_o}{s^n + \dots a_1 s + a_o} \quad (4.26)$$

Então, para qualquer fator real de ponderação $\rho > 0$, o polinômio

$$\Delta(s) = a(s)a(-s) + \rho c(s)c(-s) \quad (4.27)$$

temos coeficientes dos termos com potência ímpar iguais a zero. Logo, se λ_j é uma raiz da equação (4.27), $-\lambda_j$ também é uma raiz desta equação.

Feitas essas as considerações, verifica-se que $\Delta(s)$ pode ser reescrito como:

$$\Delta(s) = \delta^{F^*}(s)\delta^{F^*}(-s) = [\Delta(s)]^+[\Delta(s)]^- \quad (4.28)$$

onde as n raízes de $[\Delta(s)]^-$ pertencem ao semiplano complexo esquerdo e as n raízes de $[\Delta(s)]^+$ pertencem ao semiplano direito, visto que as raízes de $\Delta(s)$ são simétricas com respeito ao eixo imaginário.

Sejam $\delta^{F^*}(s)$ os pólos ótimos LQR de malha fechada do sistema (4.24) dados pelas n raízes estáveis da fatorização espectral (4.27) (Wolovich, 1995), então:

$$\delta^{F^*} = |\Delta(s)|^+ = 0 \quad (4.29)$$

Da mesma maneira, uma outra fatorização espectral pode ser usada para determinar os pólos

ótimos de $\hat{q} = 0$. Estes pólos são dados por $n - 1$ raízes estáveis de $\delta^{H^*}(s)$, definidos a partir de:

$$\begin{aligned}\bar{\Delta}(s) &= a(s)a(-s) + \gamma c(s)c(-s) \\ &= [\bar{\Delta}(s)]^+ [\bar{\Delta}(s)]^- = \delta^{H^*}(s) \delta^{H^*}(-s)\end{aligned}\quad (4.30)$$

sendo γ um outro fator de ponderação. Os objetivos de malha e resposta são obtidos a partir dos pólos originados pela variação dos fatores de ponderação ρ e γ .

4.1.6 Algoritmo LQR com Erro Nulo em Regime

Este algoritmo permite estabelecer um compromisso entre os objetivos de malha e de resposta além de obter erro nulo em regime ($e_{ss} = 0$). Neste sentido são empregadas as equações (4.21), (4.22) e (4.23) definidas anteriormente. A seguir são detalhadas as etapas necessárias para o projeto do controlador.

- Define-se o polinômio $\bar{a}(s)$ como o mínimo denominador comum de $a(s)$ e $p_r(s)$, de forma que:

$$\bar{a}(s) = a(s)\bar{p}_r(s) = \tilde{a}(s)p_r(s) \quad (4.31)$$

onde seus graus são

$$\begin{aligned}\bar{n} &= \text{grau}[\bar{a}(s)] \geq n \\ \bar{r} &= \text{grau}[\bar{p}_r(s)] \leq r\end{aligned}\quad (4.32)$$

os menores possíveis.

- Assumindo que $c(s)$ e $\bar{a}(s)$ são coprimos, então pode-se resolver a seguinte fatorização espectral estendida:

$$\bar{a}(s)\bar{a}(-s) + \rho c(s)c(-s) = \bar{\delta}^{F^*}(s)\bar{\delta}^{F^*}(-s) \quad (4.33)$$

Variando o fator de ponderação ρ , são obtidos os $\bar{n} = n + \bar{r}$ pólos desejados de malha fechada $\bar{\delta}^{F^*}(s)$.

Uma outra forma de definir $\bar{\delta}^{F^*}(s)$ é fazendo

$$\bar{\delta}^{F^*}(s) = \delta^{F^*}(s)\delta_r(s) \quad (4.34)$$

tal que $\delta^{F^*}(s)$ é determinado pela fatorização espectral (4.29) e $\delta_r(s)$ é qualquer polinômio estável de ordem \bar{r} .

- Se o polinômio mônico $\hat{q}(s)$ é definido pelas $n - 1$ raízes de δ^{H^*} , determinadas por (4.30), então

a equação

$$a(s)\bar{p}_r(s)\bar{k}(s) + c(s)h(s) = \bar{\delta}^{F^*} \hat{q}(s) \quad (4.35)$$

tem uma solução única dada pelos polinômios $\bar{k}(s)$ de ordem $n - 1$ e $h(s)$ de ordem $\bar{n} - 1$, definidos por:

$$\bar{k}(s) = s^{n-1} + \bar{k}_{n-2}s^{n-2} + \dots + \bar{k}_1s + \bar{k}_0 \quad (4.36)$$

$$h(s) = h_{\bar{n}-1}s^{\bar{n}-1} + h_{\bar{n}-2}s^{\bar{n}-2} + \dots + h_1s + h_0 \quad (4.37)$$

- Uma vez determinado o polinômio $h(s)$, a equação Diofantina

$$\hat{q}(s)\bar{q}(s) + p_r(s)\hat{h}(s) = h(s) \quad (4.38)$$

pode ser resolvida para determinar os polinômios $\bar{q}(s)$ e $\hat{h}(s)$ de ordens \bar{r} e $\bar{n} - r - 1$.

- Os polinômios $k(s)$ e $q(s)$ são definidos pelas seguintes equações:

$$k(s) = \bar{p}_r(s)\bar{k}(s) \quad (4.39)$$

$$q(s) = \hat{q}(s)\bar{q}(s) \quad (4.40)$$

Essas equações junto com a equação (4.37) correspondem aos ganhos ótimos do controlador *TDOF* da figura 4.1.

- No projeto dos controladores, verifica-se uma resposta lenta quando empregada a igualdade $q(s) = \hat{q}(s)\bar{q}(s)$ da equação (4.40); neste caso, o polinômio $\hat{q}(s)$ pode ser substituído por $\tilde{q}(s)$, que é definido pelas $n - 1$ raízes de $\bar{\delta}^{F^*}(s)$, tal que:

$$q(s) = \tilde{q}(s)\bar{q}(s) \quad (4.41)$$

Além destes passos, e a fim de satisfazer requisitos de projeto como sobre-sinal e esforço de controle, duas modificações foram introduzidas neste procedimento, que estão detalhadas no capítulo 6.

4.2 Controle a Estrutura Variável VSC

Os sistemas de controle a estrutura variável *VSC* são compostos de uma lei de controle chaveada e de uma regra de decisão (Utkin, 1977), (Hung *et al.*, 1993). Dependendo do estado do sistema, a regra de decisão chamada de *função de chaveamento* determina qual lei de controle deve atuar. É uma técnica similar à metodologia dos ganhos escalonados (Aström e Wittenmark, 1995) onde a função de decisão é projetada para forçar o sistema a alcançar uma restrição, definida por uma superfície (fase de alcance)

e posteriormente permanecer na superfície pré-definida (fase de deslizamento). O comportamento dinâmico do sistema, quando é confinado na superfície é descrito como *modo deslizante*. O *VSC* apresenta boas características de robustez frente às perturbações externas e incertezas tornando seu uso bastante atrativo. A seguir é feita a descrição do sistema não-linear empregado para a síntese e o projeto deste controlador.

4.2.1 Descrição geral do sistema

De forma geral um sistema não-linear que inclui incertezas e perturbações externas pode ser representado como:

$$\dot{x}(t) = f(x, \delta, t) + B(x, \delta, t)u(x, t) + d(t) \quad (4.42)$$

$$f(x, \delta, t) = f_o(x, t) + \Delta f(x, \delta, t) \quad (4.43)$$

$$B(x, \delta, t) = B_o(x, t) + \Delta B(x, \delta, t) \quad (4.44)$$

onde:

- $x(t)$ é o vetor de estados do sistema $n \times 1$;
- $f(x, \delta, t)$ é o vetor de funções não lineares de dimensão $n \times 1$;
- $u(t, x)$ corresponde ao vetor de entradas de controle de dimensão $m \times 1$;
- δ é o vetor de parâmetros incertos;
- $d(t)$ corresponde ao vetor de perturbações externas de dimensão $n \times 1$;
- $B(x, \delta, t)$ é uma matriz de funções não-lineares de dimensão $n \times m$;
- $\Delta f(x, \delta, t)$ e $\Delta B(x, \delta, t)$ são os vetores que representam as incertezas paramétricas do sistema de dimensão $n \times 1$;
- $f_o(x, t)$ e $B_o(x, t)$ referem-se aos parâmetros nominais.

Assume-se que o sistema atende à condição de *Lipschitz*, ou seja, numa dada região de interesse, a solução do sistema existe e é única, mesmo na presença de perturbações (Slotine e Li, 1991). Desta forma, a matriz B deve satisfazer as seguintes condições (Gao e Hung, 1993):

$$\Delta f = B_o \tilde{f} \quad (4.45)$$

$$\Delta B = B_o \tilde{B} \quad (4.46)$$

$$d = B_o \tilde{d} \quad (4.47)$$

o que significa que Δf , ΔB e d têm que pertencer à imagem de B_o ; \tilde{f} e \tilde{B} são vetores que incorporam as incertezas paramétricas e \tilde{d} as perturbações externas.

4.2.2 Síntese de controle

Nesta seção é feita a síntese do *VSC* considerando a lei de *alcance* apresentada por Gao e Hung (1993), de forma a atingir as fases de alcance e deslizamento para o sistema dado pela equação (4.42).

De forma geral, a superfície de deslizamento de um sistema multivariável (σ_i) pode ser representada por uma equação linear ou, no caso mais geral, por equações não lineares (Utkin, 1992). Neste trabalho, optou-se por superfícies lineares que apresentam um projeto bem mais simples. Normalmente estas superfícies são funções das variáveis que se desejam controlar que, em geral, são representadas pelas saídas ou pelos estados do sistema.

Definindo o erro do sistema como:

$$e(t) = x(t) - x_d(t) \quad (4.48)$$

onde e corresponde ao erro do sistema e x_d ao valor desejado, as superfícies podem ser representadas por:

$$\sigma_i = k_{c_i}^T e = 0, \quad i = 1, m \quad (4.49)$$

Desta forma, na fase de deslizamento, os erros tenderão exponencialmente para zero de acordo com um padrão determinado pelo vetor $k_{c_i} \in \mathfrak{R}^m$.

A lei de *alcance* definida por Gao e Hung (1993) corresponde a uma equação diferencial que permite especificar a dinâmica das superfícies de deslizamento σ_i . A forma prática da lei de *alcance* é dada por:

$$\dot{\sigma}_i + \varrho_i \sigma_i = -\varpi_i \operatorname{sgn}(\sigma_i) \quad (4.50)$$

onde

$$\begin{aligned} \varrho_i &= \operatorname{diag}[\varrho_1, \dots, \varrho_m], \quad \varrho > 0 \\ \varpi_i &= \operatorname{diag}[\varpi_1, \dots, \varpi_m], \quad \varpi > 0 \end{aligned} \quad (4.51)$$

e sgn é definida como:

$$\operatorname{sgn}(\sigma) = \begin{cases} +1 & \text{se } \sigma > 0 \\ 0 & \text{se } \sigma = 0 \\ -1 & \text{se } \sigma < 0 \end{cases} \quad (4.52)$$

Resolvendo a equação (4.50) para a condição inicial $\sigma_0 = \sigma_i(x(0), 0)$, é possível demonstrar que esta equação admite as seguintes soluções:

$$\sigma_i = -\frac{\varpi_i}{\varrho_i} + \left(\frac{\varpi_i}{\varrho_i} + \sigma_o\right)e^{-\varrho_i t} \quad \text{se } \sigma_o > 0 \quad (4.53)$$

e

$$\sigma_i = \frac{\varpi_i}{\varrho_i} + \left(-\frac{\varpi_i}{\varrho_i} + \sigma_o\right)e^{-\varrho_i t} \text{ se } \sigma_o < 0 \quad (4.54)$$

Igualando-se a zero as equações (4.53) e (4.54), obtém-se a seguinte expressão para o tempo de alcance que satisfaz as duas situações:

$$t_a = \frac{1}{\varrho_i} \ln\left(1 + \frac{\varrho_i}{\varpi_i} |\sigma_o|\right) \quad (4.55)$$

e se $\varrho_i = 0$, o tempo de alcance t_a é dado por:

$$t_a = \frac{|\sigma_o|}{\varpi_i} \quad (4.56)$$

Estes resultados permitem observar que o *tempo de alcance* é finito e ajustável através dos parâmetros ϱ_i e ϖ_i , o que significa que a convergência não é assintótica e pode ser acelerada, aumentando-se estes valores (Amaral, 2000).

Derivando a equação (4.49) e igualando-a com a equação (4.50) obtém-se:

$$\begin{aligned} \dot{\sigma} &= \frac{\partial \sigma}{\partial x} \dot{x} + \frac{\partial \sigma}{\partial t} \\ &= \frac{\partial \sigma}{\partial x} (f + Bu + d) + \frac{\partial \sigma}{\partial t} \end{aligned} \quad (4.57)$$

$$= -\varpi \operatorname{sign}(\sigma) - \varrho \sigma \quad (4.58)$$

de onde deriva-se a seguinte lei de controle:

$$u = -\hat{B}_\sigma^{-1} \left(\hat{f}_\sigma + \frac{\partial \sigma}{\partial t} + \varpi \operatorname{sign}(\sigma) + \varrho \sigma \right) \quad (4.59)$$

onde

$$\hat{B}_\sigma = \frac{\partial \sigma}{\partial x} \hat{B} \quad (4.60)$$

$$\hat{f}_\sigma = \frac{\partial \sigma}{\partial x} \hat{f} \quad (4.61)$$

tal que \hat{B} e \hat{f} são estimativas de B e de f , respectivamente. Definindo

$$u^* = -\left(\frac{\partial \sigma}{\partial t} + \varpi \operatorname{sign}(\sigma) + \varrho \sigma\right) \quad (4.62)$$

e substituindo (4.59) em (4.57), resulta em:

$$\begin{aligned} \dot{\sigma} &= f_\sigma - B_\sigma \hat{B}_\sigma^{-1} (\hat{f}_\sigma - u^*) + d_\sigma + \frac{\partial \sigma}{\partial t} \\ &= -\varpi \operatorname{sign}(\sigma) - \varrho \sigma - \Omega \end{aligned} \quad (4.63)$$

onde $d_\sigma = \frac{\partial \sigma}{\partial x} d$ e

$$\Omega = (\hat{f}_\sigma - f_\sigma) + (\hat{B}_\sigma - B_\sigma)u + (0 - d_\sigma) \quad (4.64)$$

correspondem às incertezas e perturbações agindo sobre o sistema.

A lei de controle dada pela equação (4.59) é empregada ao longo desta tese.

4.2.3 Análise de robustez

Escolhendo-se a função de *Lyapunov* $V = \frac{1}{2}\sigma^T\sigma$, a superfície de deslizamento será atrativa desde que a lei de controle (4.59) faça que $\dot{V} = \sigma^T\dot{\sigma}$ seja negativa definida. Usando o resultado obtido na equação (4.63), a expressão para a derivada da função de *Lyapunov* é:

$$\begin{aligned} \dot{V} &= \sigma^T\dot{\sigma} \\ &= -\sigma^T\varpi\text{sgn}(\sigma) - \sigma^T\rho\sigma - \sigma^T\Omega \end{aligned} \quad (4.65)$$

Sendo $\sigma^T\rho\sigma \geq 0$, a condição para que $\dot{V} \leq 0$ pode ser expressa como:

$$-\sigma^T\varpi\text{sgn}(\sigma) < \sigma^T\Omega \quad (4.66)$$

que é satisfeita se as componentes diagonais de ϖ atendem a seguinte restrição:

$$\varpi_i > |\psi_i|, \quad \forall i \quad (4.67)$$

onde ψ representa o máximo efeito das perturbações paramétricas e não paramétricas.

Outro critério para estabelecer o menor valor de ϖ que torna σ atrativa pode ser derivado, se $\varpi = \varpi I$, e $\rho = \rho I$. Levando-se em conta que:

$$\|\sigma\| = \sqrt{\sigma^T\sigma} = \sqrt{2V} \quad (4.68)$$

e que:

$$\text{sgn}(\sigma) \geq \frac{\sigma}{\|\sigma\|} \quad (4.69)$$

chega-se às seguintes relações:

$$\begin{aligned} \sigma^T\varpi\text{sgn}(\sigma) &\geq \varpi\sigma^T\frac{\sigma}{\|\sigma\|} = \varpi\sqrt{2V} \\ \sigma^T\rho\sigma &= \rho\sigma^T\sigma = 2\rho V \\ \sigma^T\Omega &\leq \|\sigma\|\|\Omega\| = \pm\psi\sqrt{2V} \end{aligned} \quad (4.70)$$

É possível observar que o termo $\sqrt{2V}$ pode ser negativo; no entanto, na equação (4.65), o termo que lhe deu origem ($\sigma^T\varpi\text{sgn}(\sigma)$) é sempre positivo implicando que $\sqrt{2V} \geq 0$. A pior situação no que

se refere à atratividade acontece quando o termo $\sigma^T \Omega$ é negativo. Para esta situação, e considerando as relações (4.70), a equação (4.65) se transforma na seguinte desigualdade:

$$\dot{V} \leq -2\rho V - (\varpi - \psi)\sqrt{2V} \quad (4.71)$$

Da equação (4.71), pode-se concluir que para que \dot{V} seja negativa é suficiente que $\varpi > \psi$. Existem outras possibilidades para assegurar esta condição. Por exemplo, pode-se aumentar o valor de ρ tal que o valor de $(2\rho V)$ seja superior a $-(\varpi - \psi)\sqrt{2V}$, entretanto essa opção é mais complicada para ser implementada. Com esta restrição sobre o valor de ϖ , consegue-se garantir robustez frente às incertezas e perturbações.

4.2.4 Uso da camada limite

As limitações físicas dos atuadores tais como a histerese, impossibilitam o chaveamento extremamente rápido do *VSC*. Como consequência a trajetória de estados do sistema oscila numa vizinhança em torno da superfície de deslizamento, o que pode ocasionar a excitação de dinâmicas não modeladas de alta frequência que podem levar à instabilidade do sistema. Este fenômeno é conhecido como *chattering* e se constitui numa desvantagem no *VSC*. A figura 4.3 ilustra a comparação entre o modo deslizante ideal e o modo deslizante com a presença de *chattering*.

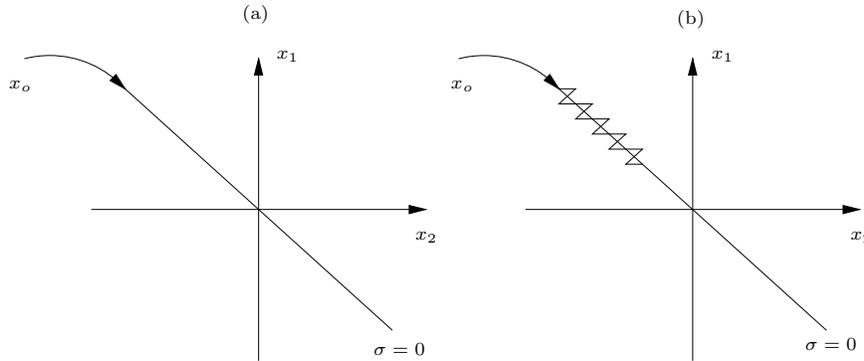


Figura 4.3: Modo deslizante: (a) Ideal (b) Com *chattering*

Para minimizar os efeitos deste problema prático, várias propostas têm sido feitas (Hung *et al.*, 1993), das quais a mais utilizada envolve o conceito de *camada limite* (Boundary Layer), que considera que o deslizamento não somente ocorre na superfície, mas ao longo de uma vizinhança em torno desta como ilustrado na figura 4.4.

A aplicação da camada limite visa obter um chaveamento menos freqüente. Com este objetivo, pode-se usar a função *sat* em lugar da função *sgn*. A função *sat* é definida como:

$$sat(\sigma) = \begin{cases} \frac{\sigma}{\phi} & \text{se } |\sigma| \leq \phi \\ sgn(\sigma) & \text{se } |\sigma| > \phi \end{cases} \quad (4.72)$$

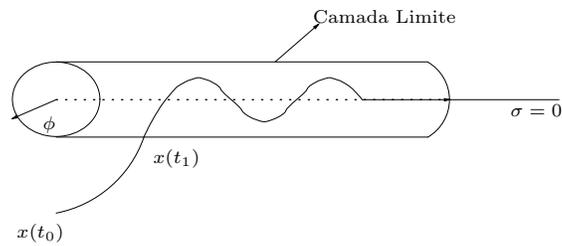
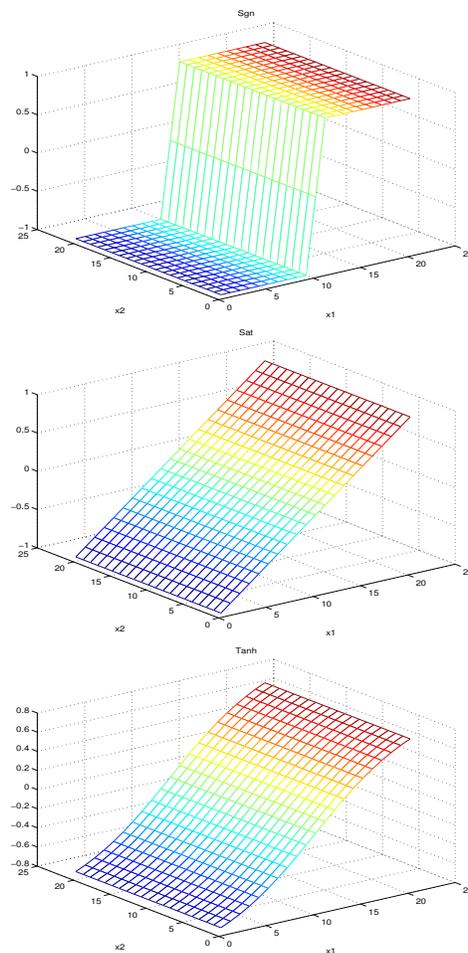


Figura 4.4: Camada limite

onde ϕ corresponde ao valor da *camada limite*. Outra forma de suavizar mais o controle, é utilizando a função tangente hiperbólica (Lewis *et al.*, 1993), (Glover e Munighan, 1997). Para obter este chaveamento neste trabalho foi implementada a função tanh.

$$\tanh\left(\frac{\sigma}{\phi}\right) \quad (4.73)$$

O perfil do chaveamento das três funções mencionadas é ilustrado na figura 4.5.

Figura 4.5: Funções *sgn*, *sat* e *tanh*

4.3 Controle Nebuloso *FLC*

Os primeiros estudos em teoria de conjuntos nebulosos foram iniciados em 1965 com a publicação dos trabalhos desenvolvidos por Zadeh (1965). Como resultado destes estudos, foi introduzido o conceito de lógica nebulosa, que na atualidade constitui-se numa ferramenta bastante eficiente para o desenvolvimento de controladores para sistemas complexos (Kiguchi e Fukuda, 1997), (Almeida, 2002). Ao contrário da lógica booleana, onde são conhecidos 2 estados, falso ou verdadeiro, na lógica nebulosa pode-se estabelecer um grau de verdade que varia entre 0 e 100%, o que permite obter estados parcialmente verdadeiros ou parcialmente falsos. A partir destes *níveis de verdade*, a lógica nebulosa diferencia-se da lógica digital de duas formas:

- Os *níveis de verdade* passam a ser classificados qualitativamente;
- Os elementos dos níveis apresentam diversos graus de pertinência (Zadeh, 1996).

Neste sentido, os sistemas baseados em lógica nebulosa possuem a característica de simular o raciocínio humano utilizando variáveis lingüísticas tais como: pequeno, médio, grande, muito pequeno, muito grande e outras, ao invés de variáveis numéricas (Sandri e Correia, 1999). Já no contexto de controle, se um operador humano é capaz de articular uma estratégia de controle como um conjunto de regras da forma **se** <condição> **então** <ação>, tem-se um algoritmo inteligente possível de ser armazenado num computador digital. O resultado é um controlador implementado por regras que possibilita sistemas menores e mais econômicos que os convencionais.

Mesmo que já exista um grande número de publicações sob o estado da arte, a necessidade de formas sistemáticas de projeto, de análise de estabilidade, robustez e, principalmente, questões relacionadas à implementação prática permitem afirmar que se trata de um assunto com varias questões práticas e teóricas ainda em aberto.

4.3.1 Estrutura do *FLC*

Ao contrário dos controladores convencionais, nos quais o algoritmo de controle é descrito de forma analítica, utilizando equações diferenciais ou algébricas, um *FLC*, na sua forma básica, utiliza regras do tipo **se** <condição> **então** <ação> junto com mecanismos de raciocínio na estrutura. A estrutura geral destes controladores é ilustrada na figura 4.6 (Lee, 1990).

Dependendo do tipo e das características requeridas para o projeto de controle, outras estruturas podem ser utilizadas, mas, em geral, essas estruturas são variações da figura 4.6

Procedimento de nebulização

A interface de nebulização é responsável pelo tratamento dos valores das variáveis de entrada que caracterizam o estado do sistema e pela normalização destas variáveis de forma que possam ser

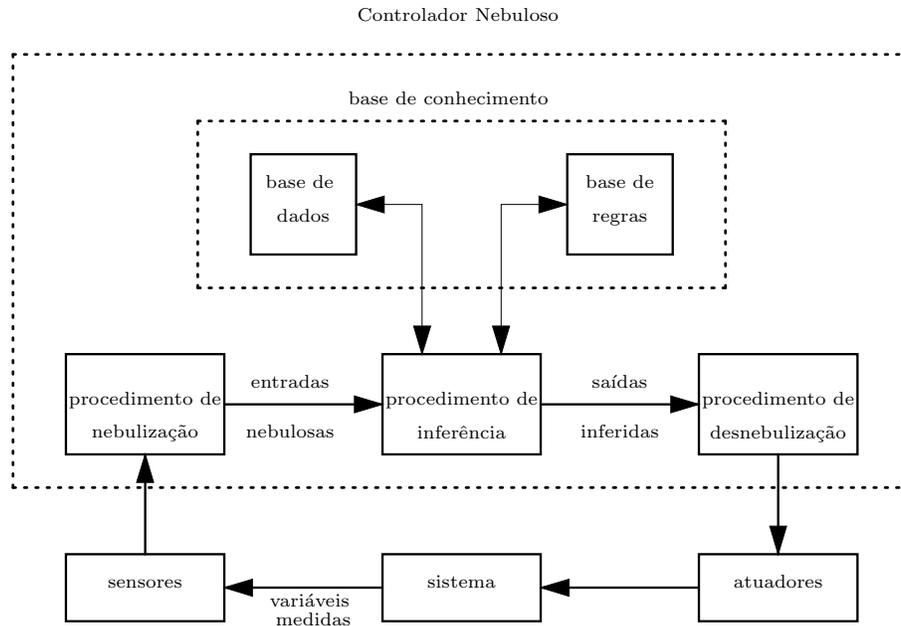


Figura 4.6: Estrutura básica de um controlador nebuloso

tratadas pelos conjuntos nebulosos do controlador. Estes valores são, então, desnebulizados, com a transformação da entrada *crisp* (valores numéricos transformados em instâncias linguísticas).

A base da nebulização é composta pelas funções de pertinência ($FP's$). Sua forma (triangulares, trapezoidais, etc), a sobreposição e os valores de pico influenciam diretamente como deve ser o projeto e qual o comportamento do sistema. Portanto, é importante estabelecer as propriedades das $FP's$ do conjunto, assim como dos conjuntos nebulosos que representam. Algumas destas propriedades são apresentadas no apêndice C e podem ser encontradas também em (Almeida, 2002).

Base de conhecimento

A base de conhecimento é composta por uma base de dados e uma base de regras, as mesmas que caracterizam a estratégia de controle. Na base de dados, ficam armazenadas as definições sobre a quantização e a normalização das variáveis a serem tratadas pelo controlador e as definições das funções de pertinência dos termos nebulosos. A base de regras é formada por estruturas do tipo:

Se < condição > **então** < ação >

Como por exemplo:

Se < Erro de Posição é Positivo Grande > **e** < Erro de Velocidade é Positivo Grande > **então** < Controle é Positivo Grande >.

As regras, juntamente com os dados de entrada, são processadas no procedimento de inferência resultando nas ações de controle de acordo com o estado do sistema.

Em um *FLC* é necessário que a base de regras seja completa para garantir que pelo menos uma regra seja disparada, para qualquer estado possível do sistema. Deseja-se também que a base de regras

seja consistente para evitar contradições nas ações do controlador.

Procedimento de inferência

O mecanismo de raciocínio nebuloso ou procedimento de inferência é o núcleo do controlador nebuloso. Este deve ser capaz de tomar decisões a partir das regras, que estabelecem as implicações nebulosas para o comportamento do sistema; assim como dos conceitos vindos da lógica nebulosa (Lee, 1990). O processo de inferência é composto pelas seguintes ações (Sandri e Correia, 1999):

1. verificação do grau de compatibilidade entre os fatos e as disposições nas premissas das regras;
2. determinação do grau de compatibilidade global ou de disparo (GD) da premissa de cada regra;
3. determinação do valor da conclusão, em função do grau de compatibilidade da regra com os dados e as ações de controle constantes na conclusão;
4. agregação dos valores obtidos como conclusão nas várias regras, obtendo-se uma ação de controle global.

Os controladores nebulosos encontrados na literatura são os modelos clássicos, compreendendo os de *Mamdani* e *Larsen* e os de interpolação de *Takagi – Sugeno* e *Tsukamoto*, (Lee, 1990) (Sandri e Correia, 1999). A diferença entre eles está associada à representação dos termos nas premissas, as ações de controle e os operadores empregados para a implementação do controlador. Nesta tese é empregado o mecanismo de *Mamdani*.

Os mecanismos *Mamdani* utilizam a composição *min-max*, *min* para a *T – norma* e *max* para a *T – conorma* (definições no apêndice C). No controle nebuloso é comum a utilização do produto para a *T – norma* e soma para a *T – conorma*. Uma comparação da capacidade de interpolação entre regras para as composições *min-max* e *produto-soma* esclarece algumas das vantagens da utilização da composição *produto-soma*.

Na implementação dos controladores nebulosos, geralmente os sistemas são compostos de duas entradas tais como o erro e a derivada do erro; portanto, a utilização das composições *min-max* ou *produto-soma* não geram resultados que divergem entre si (Almeida, 2002).

Procedimento de desnebulização

Nos controladores nebulosos do tipo clássico, a interface de desnebulização é utilizada para obter uma única ação de controle. O procedimento compreende a identificação do domínio das variáveis de saída num correspondente universo de discurso. Com a ação de controle nebulosa inferida, evolui-se para uma ação de controle não-nebulosa. O método de desnebulização mais empregado é o método do centro de gravidade (detalhes no apêndice C).

4.4 Controladores *FLC* Baseados no *VSC*

Tanto os *FLC*'s quanto o *VSC*'s têm sido desenvolvidos e aplicados ao controle de sistemas não-lineares durante as duas últimas décadas. Como foi apresentado na seção anterior, o *FLC* constitui-se numa técnica atraente, já que facilmente pode ser incorporado nas leis de controle. Ao mesmo tempo diferentes publicações têm mostrado a similaridade entre o *FLC* e o *VSC* (Glower e Munighan, 1997), (Yi e Chung, 1997). A combinação das duas técnicas tem despertado o interesse de pesquisadores, visando principalmente a análise de estabilidade, pelo fato da estabilidade do *FLC* ser difícil de provar, o que não acontece com o *VSC*. Outras vantagens do *VSC* são a propriedade de invariância e a habilidade de desacoplar sistemas de ordem elevada em sistemas de ordem reduzida. Estas características permitem uma redução na base de regras do *FLC*. Uma vantagem importante da combinação de ambas as técnicas é que é possível eliminar efetivamente o fenômeno do *chattering* que é uma desvantagem no *VSC*.

4.4.1 Normalização do *FLC*

Baseados na técnica *VSC*, os trabalhos publicados em (Yi e Chung, 1994), (Yi e Chung, 1997) e (Glower e Munighan, 1997) propõem controladores nebulosos que associam a superfície de deslizamento ilustrada na figura 4.7 à disposição da base de regras do *FLC* de formato *UNLP*⁵.

A base de regras e os conjuntos lingüísticos associados são apresentados nas tabelas 4.1 e 4.2.

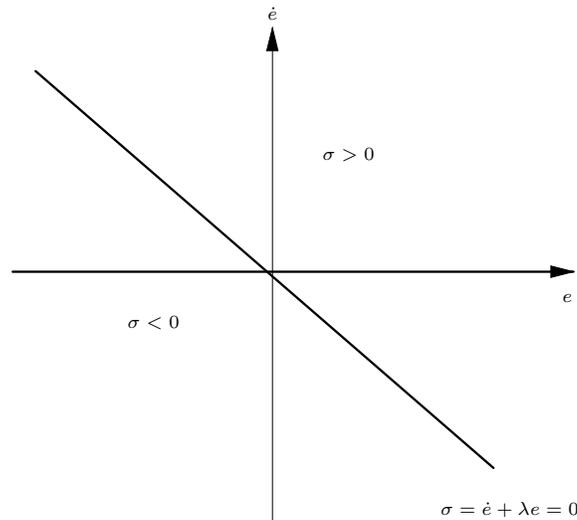


Figura 4.7: Características do controle *VSC*

Uma descrição mais precisa deste controlador é feita normalizando as entradas e saídas, tal que e_n , \dot{e}_n e $\tau_n \in [-1, 1]$ correspondem ao erro, à derivada do erro e à saída de controle como ilustrado na figura 4.8.

⁵Upper-diagonal-Negative, Lower-diagonal-Positive

Tabela 4.1: Base de regras

		e				
		NBE	NSE	ZRE	PSE	PBE
\dot{e}	PBDE	CZR	CNB	CNB	CNB	CNB
	PSDE	CPB	CZR	CNS	CNS	CNS
	ZRDE	CPB	CPS	CZR	CNS	CNB
	NSDE	CPB	CPS	CPS	CZR	CNB
	NBDE	CPB	CPB	CPB	CPB	CZR

Tabela 4.2: Conjuntos nebulosos

NBE, NBDE	- e, \dot{e} é negativo grande	CNB	- τ é negativo grande
NSE, NSDE	- e, \dot{e} é negativo pequeno	CNS	- τ é negativo pequeno
ZRE, ZRDE	- e, \dot{e} é zero	CZR	- τ é zero
PSE, PSDE	- e, \dot{e} é positivo pequeno	CPS	- τ é positivo pequeno
PBE, PBDE	- e, \dot{e} é positivo grande	CPB	- τ é positivo grande

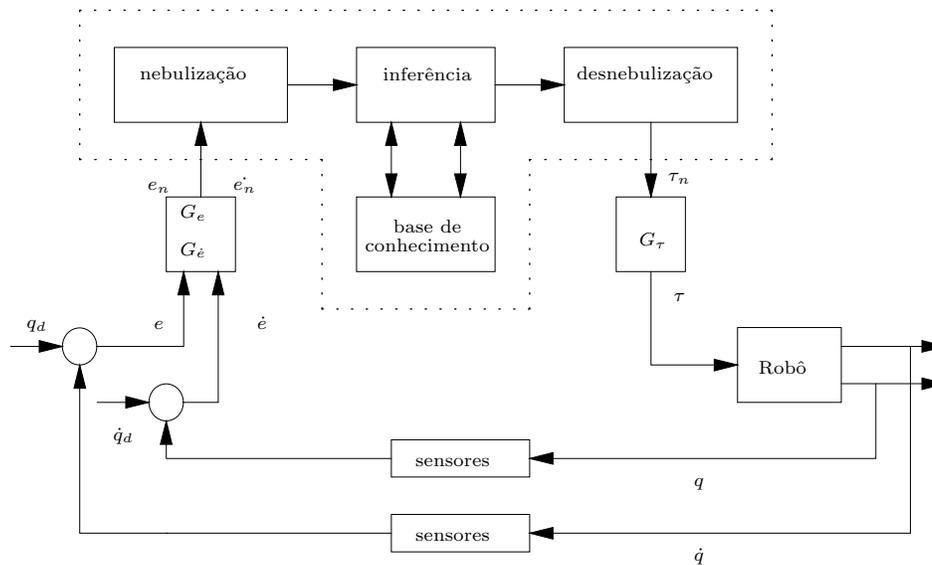


Figura 4.8: Controlador nebuloso normalizado

Os ganhos G_e , $G_{\dot{e}}$ e G_τ correspondem aos fatores de escala entre os valores reais lidos pelos sensores e os valores normalizados empregados no *FLC*, isto é,

$$e_n = G_e e \quad (4.74)$$

$$\dot{e}_n = G_{\dot{e}} \dot{e} \quad (4.75)$$

$$\tau = G_\tau \tau_n \quad (4.76)$$

Das funções de pertinência triangulares, ilustradas na figura 4.9, observa-se que a intersecção dos pares $\{NBE, PBDE\}$ e $\{PBE, NBDE\}$, correspondentes ao erro e e à derivada do erro \dot{e} , permitem

traçar uma linha diagonal cuja equação normalizada é:

$$\dot{e}_n + e_n = 0 \quad (4.77)$$

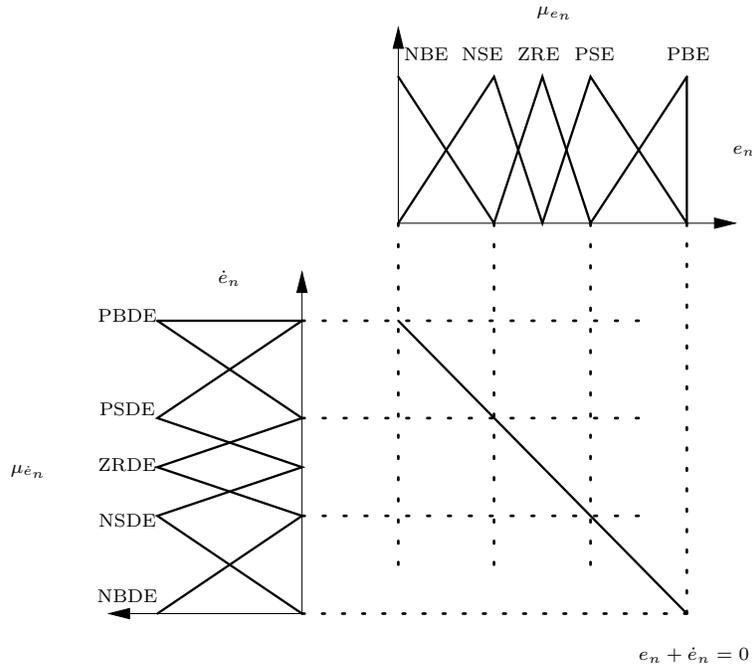


Figura 4.9: Funções de pertinência

A relação correspondente da equação (4.77) no sistema real é dada por:

$$\dot{e} + \lambda e = 0, \quad \lambda = \frac{G_e}{G_{\dot{e}}} > 0 \quad (4.78)$$

tal que a variável σ pode ser definida como:

$$\sigma = \dot{e} + \lambda e \quad (4.79)$$

Considerando a estrutura de regras *UNLP* do *FLC*, as seguintes propriedades podem ser observadas:

- a) $\tau(e, \dot{e})|_{\sigma=0} \approx 0$
 - b) $\tau(e, \dot{e})|_{\sigma>0} < 0$
 - c) $\tau(e, \dot{e})|_{\sigma<0} > 0$
- (4.80)

As propriedades apresentadas acima e a relação (4.79) permitem verificar a similaridade entre o

FLC e o *VSC*. O teorema 4.4.1 garante a existência da camada limite ϕ no *FLC*.

Teorema 4.4.1 *Existência de ϕ* : Para qualquer $k_c \in [0, G_\tau]$ existe uma constante positiva ϕ tal que a saída de controle τ do *FLC* seja:

$$\tau = -k_\beta \frac{\sigma}{|\sigma|} \quad \text{se } |\sigma| \geq \phi \quad (4.81)$$

onde $k_\beta > k_c$ e $\sigma = \dot{e} + \lambda e$, para $\lambda = \frac{G_e}{G_c} > 0$.

O teorema 4.4.1 pode ser provado considerando a figura 4.10, onde é ilustrada a linha perpendicular à superfície $\sigma = 0$ e as superfícies $\sigma = \phi_a^*$ e $\sigma = \phi_c^*$.

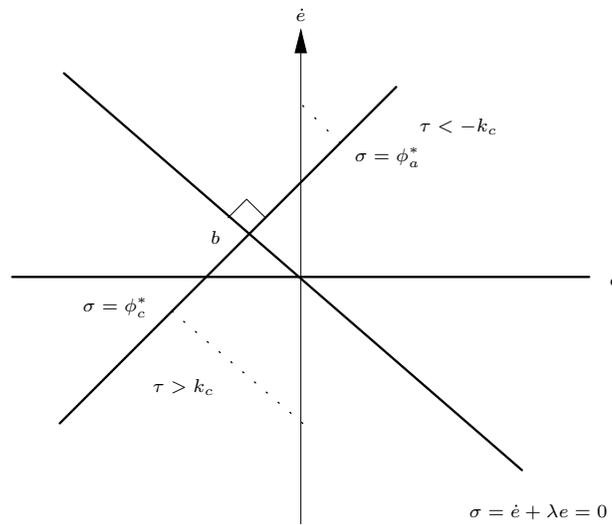


Figura 4.10: Superfícies dos controladores nebulosos

Com estas considerações, as seguintes equações de controle ao longo da superfície perpendicular são obtidas.

$$\tau(e, \dot{e})|_a = G_\tau|_{CNB} = -G_\tau \quad (4.82)$$

$$\tau(e, \dot{e})|_b \approx G_\tau|_{CZR} \approx 0 \quad (4.83)$$

$$\tau(e, \dot{e})|_c = G_\tau|_{CPB} = G_\tau \quad (4.84)$$

Para as superfícies $\sigma = \phi_a^*$ e $\sigma = \phi_c^*$ as saídas de controle correspondem a:

$$\tau(e, \dot{e}) < -k_c \quad \text{se } \sigma > \phi_a^* \quad (4.85)$$

$$\tau(e, \dot{e}) > k_c \quad \text{se } \sigma < \phi_c^* \quad (4.86)$$

tal que os termos ϕ e k_c são parâmetros de projeto determinados pelos valores das funções de per-

tinência e pelos fatores de escala como sendo:

$$k_c = G_\tau CPS \quad (4.87)$$

$$\phi = 2 \frac{PSE}{G_\epsilon} \quad (4.88)$$

Substituindo a equação (4.87) na equação (4.85) a seguinte lei de controle é definida:

$$\tau(e, \dot{e}) < G_\tau CNS \quad \text{se } \sigma > \phi \quad (4.89)$$

$$\tau(e, \dot{e}) > G_\tau CPS \quad \text{se } \sigma < -\phi \quad (4.90)$$

onde ϕ é definida pela equação (4.88). A lei de controle dada pelas equações (4.89) e (4.90) foi implementada experimentalmente neste trabalho. Informações adicionais podem ser obtidas em (Yi e Chung, 1997), (Yi e Chung, 1998) e (Glower e Munighan, 1997) onde este controlador é utilizado em trabalhos de simulação.

4.5 Conclusões do Capítulo

Neste capítulo, foram apresentadas técnicas robustas de controle, a técnica *TDOF*, a técnica *VSC* e o controle nebuloso *FLC* baseado no controle *VSC*. A primeira técnica corresponde a uma técnica linear e as outras são técnicas não-lineares. No *TDOF*, observou-se a necessidade de contar com um ou vários modelos lineares para sua implementação. Do ponto de vista de estabilidade, no *TDOF* a análise é feita no domínio da frequência. Para as técnicas não-lineares, a análise de estabilidade é feita na representação de estados e é baseada em funções de *Lyapunov*.

A principal motivação para o estudo e aplicação experimental destas técnicas advém das limitações que apresentam os controladores clássicos *PD* e *PID*, quando o sistema é submetido a incertezas, ruídos e perturbações.

Duas melhoras são evidentes nas técnicas *VSC* e *FLC*. A primeira é o uso e a implementação experimental da função *tanh* para obter um chaveamento mais suave, e a segunda é que o *FLC* elimina efetivamente o fenômeno do *chattering*.

Capítulo 5

Bancada Experimental

Para a implementação em tempo real do controle de força e posição é imprescindível que o robô manipulador, através de seus sensores, possa dispor de dados referentes às mudanças apresentadas no ambiente em que desenvolve sua tarefa. O robô industrial utilizado neste trabalho é um robô de configuração *SCARA*¹ de 4 graus de liberdade (*RRPR*). Este robô dispõe tanto de sensores de posição quanto de um sensor de força, necessários para a implementação do controle de força e posição. Em forma simplificada, a estrutura prática para a implementação do controle de força e posição é ilustrada na figura 5.1:

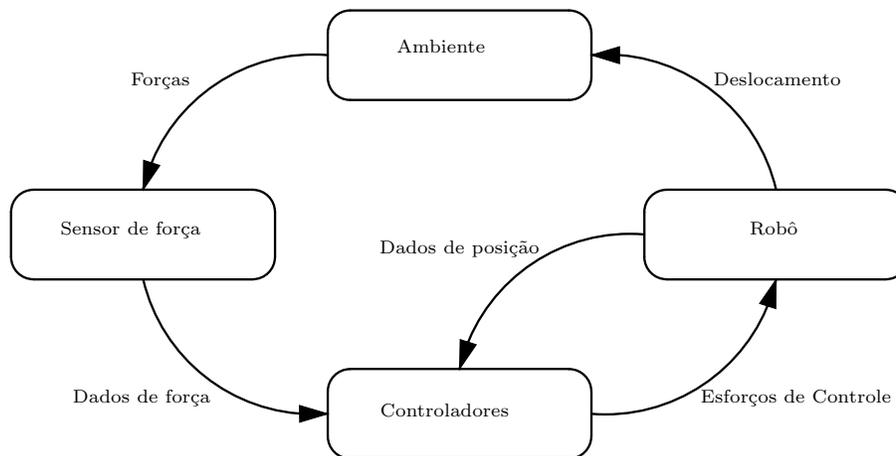


Figura 5.1: Malha para o controle de força e posição

Neste capítulo, é apresentada a bancada experimental disponível no Laboratório de Robótica da Universidade Federal de Santa Catarina, assim como as características do sistema operacional e linguagem de programação *XO/2*². Estes dois últimos são empregados para a programação e implementação em tempo real dos controladores propostos.

¹Selective Compliant Articulated Robot Arm

²Xoberon

5.1 Laboratório de Robótica

Todos os experimentos realizados neste trabalho foram executados no Laboratório de Robótica da Universidade Federal de Santa Catarina. Os equipamentos disponíveis neste laboratório são listados a seguir:

- O robô industrial de configuração *SCARA*;
- O sensor de força de marca *Jr3*³.
- O sistema operacional e a linguagem de programação em tempo real *XO/2*, executada em plataforma *Windows* que interage com a unidade de controle do robô.

Cada um destes componentes são descritos a seguir.

5.1.1 Robô industrial SCARA

O robô industrial de configuração *SCARA* (Inter)⁴, ilustrado na figura 5.2, foi construído pelo Instituto de Robótica da Universidade Técnica Federal de *Zurique (ETHZ)*, Suíça, e instalado no Laboratório de Robótica da Universidade Federal de Santa Catarina. A partir da base, as duas primeiras juntas são de rotação, girando em torno de seus eixos verticais e trabalhando, portanto, num plano horizontal. A terceira junta é de translação ou prismática, executando movimentos no sentido vertical. A última junta realiza movimentos rotacionais sobre o eixo *z*, o que permite definir sua orientação no plano horizontal.



Figura 5.2: Robô industrial *SCARA* de 4 graus de liberdade

Nas duas primeiras juntas do robô, a transmissão é feita através de *Harmonic Drives* com uma redução imposta de 1 : 100. Nas juntas 2 e 3 (Ao contrário da notação convencional, neste robô a

³Nome do fabricante

⁴Nome dado ao robô pelos seus fabricantes.

numeração das juntas é feita a partir do 0 até o 3, isto é devido ao software de controle), a redução é feita através de polias e correias dentadas, que movimentam um sistema *ball-screw-spline*, que produz movimentos lineares e de rotação simultaneamente (a junta 3 tem um redutor planetário com redução de 1 : 4.5). Os limites de operação correspondentes às posições, velocidades, acelerações e esforços de controle para cada uma das juntas são apresentados na tabela 5.1. Informações técnicas adicionais podem ser encontradas em (Weihmann, 1999) e (Hüppi e Gruener, 2001).

Tabela 5.1: Limites do robô industrial *SCARA*

Elo	q_{min} [rad]	q_{max} [rad]	\dot{q}_{min} [rad/s]	\dot{q}_{max} [rad/s]	\ddot{q}_{min} [rad/s ²]	\ddot{q}_{max} [rad/s ²]	τ_{min} [Nm]	τ_{max} [Nm]
0	-2,25	2,25	-3,0	3,0	-80,0	80,0	-333,0	333,0
1	-1,9	1,9	-3,0	3,0	-100,0	100,0	-157,0	157,0
2	-0,49 [m]	-0,21 [m]	-0,888 [m/s]	0,888 [m/s]	-12,0 [m/s ²]	12,0 [m/s ²]	-877,0	877,0
3	-2,4	2,4	-20,0	20,0	-500,0	500,0	-16,7	16,7

O acionamento das juntas é feito por quatro motores elétricos de corrente alternada, sendo que sensores de posição (*encoders*) acoplados às juntas do robô permitem obter as medições de posição. A precisão destas medições alcança a 4 [μ rad] nas juntas 0 e 1, 1 [μ m] na junta 2, e 24 [μ rad] na junta 3. As medições de velocidade são obtidas a partir de diferentes aproximações das medições de posição. O robô é equipado, ainda, por um sensor de força da marca *Jr3*, acoplado no efetuador final. Este sensor fornece as medidas de forças e momentos.

O robô é controlado por uma versão aberta do sistema operacional e linguagem de programação *XO/2*, baseado na arquitetura *VME* com uma placa central de processamento *Motorola 2604/2121*. A figura 5.3 ilustra a arquitetura aberta de controle do robô.

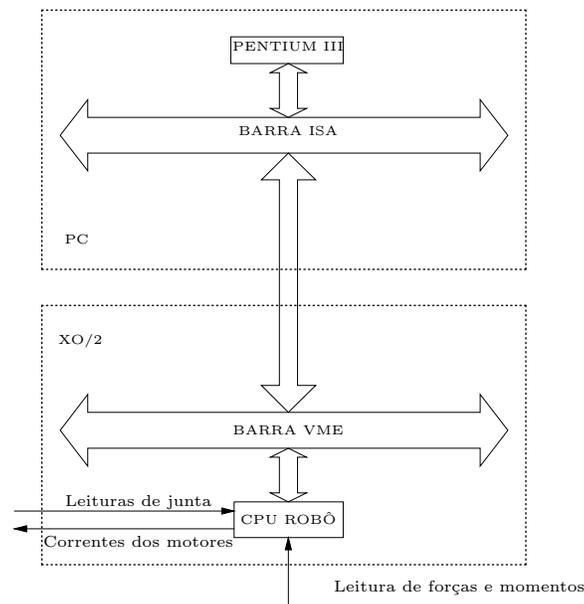


Figura 5.3: Arquitetura aberta de controle

5.1.2 Sensor de força

O sensor de força *Jr3* acoplado ao efetuador final do robô é ilustrado na figura 5.4.

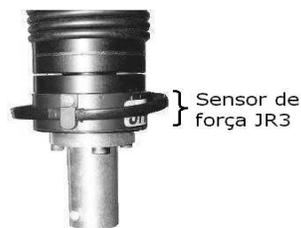


Figura 5.4: Sensor de força

Este sensor permite obter medições de forças e momentos, a partir da tensão nos extensômetros de resistência (*strain gauges*). Estes dados são transmitidos para a placa do sensor, que, por sua vez, é conectada à placa central da unidade de controle. A placa que acompanha o sensor de força é responsável por receber e processar os dados seriais transmitidos pelo sensor, bem como monitorar e ajustar, automaticamente, a alimentação elétrica do sensor. Ela contém vários processadores digitais de sinais (*DSP*). A composição do sensor de força assim como sua conexão com a unidade de controle é ilustrada na figura 5.5.

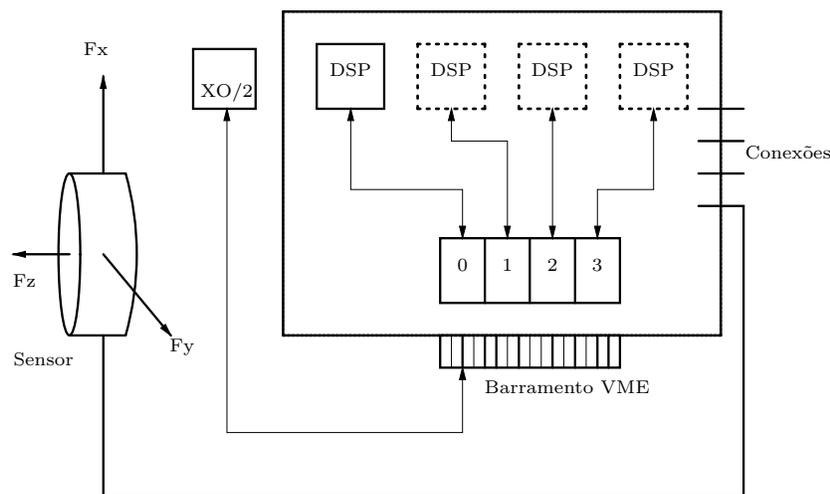


Figura 5.5: Arquitetura do sensor de força

Os limites do sensor de força são apresentados na tabela 5.2. Detalhes que consideram os aspectos construtivos dos sensores de força, podem ser encontrados em (Gorinevsky *et al.*, 1997) e, em forma específica, para o sensor *Jr3* em (Weihmann, 1999) e (Hüppi e Gruener, 2001).

<i>Eixo</i>	x	y	z
Força [N]	100	100	200
Momento [Nm]	6,3	6,3	6,3

Tabela 5.2: Limites do sensor de força

5.1.3 Ferramentas

Atualmente o robô conta com as ferramentas ilustradas na figura 5.6, sendo que as ferramentas de corte e perfuração foram acrescentadas durante o desenvolvimento desta tese. O objetivo destas ferramentas é permitir a realização de diferentes tipos de tarefas, tais como corte, perfuração, seguimento de contornos, etc. Ao mesmo tempo, permitem verificar o desempenho das diferentes leis de controle implementadas no robô.



Figura 5.6: Ferramentas

5.2 O Sistema Operacional e a Linguagem de Programação *XO/2*

A linguagem em tempo real e o sistema operacional *XO/2* foram desenvolvidos no Departamento de Robótica do Instituto Tecnológico Federal de Zurique (*ETHZ*), sendo seu principal propósito a rápida implementação de interfaces e algoritmos de controle em produtos mecatrônicos. O *XO/2* é baseado no sistema operacional *Oberon* e é escrito na linguagem orientada a objetos *Oberon – 2* (ambos desenvolvidos no Instituto de Computação de *Zurique*) Hüppi e Gruener (2001).

As principais características do *XO/2* são: o escalonador de tempo real, a administração da memória e o modularizado. Seus aspectos mais importantes são detalhados a seguir.

5.2.1 Escalonador de tempo real

O mecanismo que é responsável pela correta implementação das tarefas, em tempo real, e pela obtenção de resultados num tempo predefinido (*deadline*) é chamado de *escalonador* (*Scheduling*).

Neste sentido, uma aplicação em tempo real pode ser modelada como um conjunto de tarefas que trabalham cooperativamente. Considerando os requisitos de tempo, estas tarefas podem ser divididas em *críticas* e *não-críticas*. A primeira é aquela cujo tempo de execução deve ser, obrigatoriamente

atendido. Caso a tarefa não seja executada no *deadline*, o sistema gerará uma falha. As tarefas não-críticas são aquelas que não apresentam requisitos de tempo real como, por exemplo, as tarefas de manutenção (Brega, 1998).

O escalonamento de tarefas críticas é realizado priorizando a tarefa de menor *deadline*. As tarefas não-críticas podem ser executadas somente quando não há tarefas em tempo real sendo executadas ou em espera.

5.2.2 Administrador de memória

A administração da memória disponível durante a execução das tarefas pelo manipulador é feita através do chamado *coletor de lixo*. O coletor de lixo é uma tarefa não-crítica que tem prioridade variável, ou seja, quanto menor for a memória disponível, maior será a prioridade do coletor.

5.2.3 Modularizado

Os programas desenvolvidos em *XO/2* são escritos em unidades separadas de compilação chamadas *módulos*. Os módulos contêm um conjunto de funções e procedimentos, além das estruturas de dados locais e globais. Ao mesmo tempo, estes módulos contêm uma interface para permitir que, procedimentos sejam importados entre eles.

Durante a execução de tarefas, os módulos previamente compilados são transmitidos à unidade de controle (target) através da rede à qual estão conectados. Em alguns casos, é necessário o descarregamento de um módulo do target para ser atualizado. Isto é possível se o módulo em questão não está sendo importado por outro módulo. Do contrário, o *XO/2* permite o carregamento e descarregamento dinâmico dos módulos já compilados.

5.3 Módulos de Controle

Para uma melhor compreensão da linguagem de programação e do sistema operacional *XO/2* são apresentados a seguir os módulos de controle já implementados, assim como os que foram criados e implementados neste trabalho de doutorado. Os principais módulos de controle, os módulos adicionais, seus procedimentos e o relacionamento entre eles é ilustrado na figura 5.7.

5.3.1 Módulo Robot

Este módulo define o objeto *Robot* para que ele possa ser empregado como interface com os diferentes periféricos. São definidas variáveis globais que especificam o estado do robô, sendo estas variáveis, posições, velocidades, acelerações, forças e esforços de controle, tanto no espaço de juntas como no espaço cartesiano. Finalmente, são definidas as tarefas em tempo real utilizadas para controlar o robô (Hüppi e Gruener, 2001).

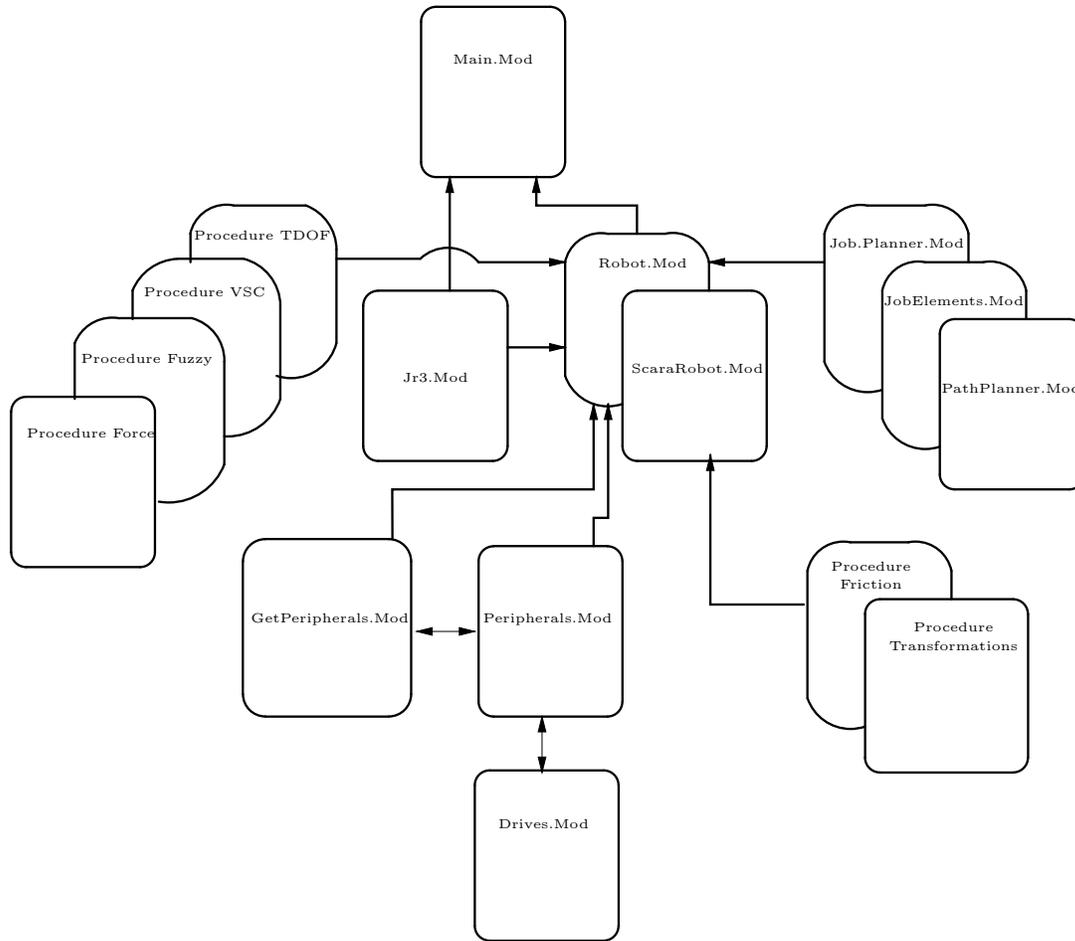


Figura 5.7: Módulos de controle e procedimentos (atuais e implementados)

5.3.2 Módulo ScaraRobot

Neste módulo são especificadas algumas características construtivas do robô. Ao mesmo tempo, são calculadas a cinemática direta, a cinemática inversa e a dinâmica inversa (compensação dinâmica). No cálculo da dinâmica inversa são considerados a rigidez, o amortecimento e o atrito de *Coulomb*. Inicialmente, esse cálculo não estava disponível e foi acrescentado durante o desenvolvimento desse trabalho como forma de permitir a inclusão de diferentes tipos de compensação usados nas leis de controle implementadas. Por outro lado, neste módulo foram implementados os procedimentos para o cálculo das matrizes de inércia, dos torques centrífugos e de Coriolis, dos jacobianos e suas derivadas, a transformação de coordenadas e a compensação de atrito de *LuGre*. A modelagem cinemática do robô é apresentada em detalhe no apêndice A.

5.3.3 Módulo Jr3

Este módulo foi implementado para permitir a comunicação entre o *XO/2* e a placa de controle do sensor de força. Os procedimentos implementados neste módulo permitem a leitura das forças e

momentos aplicados sobre o sensor, a inicialização do sensor, a eliminação do offset e outras.

5.3.4 Módulo JobPlanner

Este módulo gera e controla um conjunto de elementos *Job*, trocando a informação das variáveis de estados com o robô. É empregado para movimentar o robô, tanto no espaço cartesiano quanto no espaço de juntas, especifica pausas e finais nas trajetórias a serem executadas.

5.3.5 Módulo JobElements

Neste módulo, é feito o planejamento de trajetórias e outras tarefas como *esperar* ou *parar*. Cada uma das tarefas são armazenadas como *Jobs* de diferentes tipos, que interagem com outros. Cada um desses *Jobs* tem um procedimento de inicialização e outro de execução para cada instante de tempo.

5.3.6 Módulo PathPlanner

Neste módulo, são implementados os procedimentos para a geração de trajetórias, considerando as velocidades e acelerações máximas e mínimas, assim como as restrições nas derivadas da aceleração (Jerk) para uma determinada trajetória. As trajetórias atuais correspondem a trajetórias incrementais de terceira ordem.

5.3.7 Módulo Main

Este módulo implementa a interface principal entre os usuários e os outros módulos. Permite interagir com o *hardware* através de procedimentos em médio e baixo nível. Além disto, contém as instruções que permitem interagir com os procedimentos e objetos definidos para controlar o robô e seus periféricos.

Além destes módulos, outros complementares foram implementados visando a execução de operações matemáticas, a fuzzificação, a de-fuzzificação dos controladores nebulosos e a filtragem de sinais. Os procedimentos *ControlDOHdl*, *ControlVSHdl*, *ControlFuzzyHdl* e *ControlForcaHdl* foram criados no módulo *Robot.Mod* para a implementação destes controladores. Esses módulos são detalhados no apêndice E.

5.4 Conclusões do Capítulo

Neste capítulo, foram apresentados e descritos o robô industrial de configuração *SCARA*, o sensor de força, suas ferramentas e o conjunto de equipamentos adicionais empregados para a implementação experimental de algoritmos de controle de força e posição.

Foi apresentada a nova versão da linguagem de programação e sistema operacional *XO/2* para a implementação dos algoritmos de controle do robô e a definição dos tempos da execução das tarefas

em tempo real. Ao mesmo tempo, mostrou-se que este sistema operacional dispõe de uma memória mestra para o gerenciamento das tarefas e permite o carregamento e descarregamento dinâmico dos diferentes módulos implementados.

Finalmente, foram detalhados alguns de seus módulos básicos de controle e mencionados aqueles que foram implementados ao longo desta tese.

Capítulo 6

Controle de Posição: Resultados Experimentais

Neste capítulo, são apresentados e detalhados os projetos e a implementação experimental para o controle de posição e/ou trajetória dos controladores apresentados no capítulo 4. Nestas técnicas, é feita a compensação experimental do atrito utilizando o modelo de *LuGre* apresentado no capítulo 3.

A derivação do modelo linear para o projeto dos controladores *TDOF* foi feita considerando o modelo de De Luca e Manes sem restrições. Todos os controladores foram implementados no robô *SCARA* detalhado no capítulo 5.

Antes da implementação dos controladores *TDOF*, *VSC* e *FCL* é feita a validação do modelo dinâmico do robô para ter uma noção do grau de incertezas do mesmo. Alguns dos resultados apresentados neste capítulo foram publicados em (Vargas *et al.*, 2000), (Vargas *et al.*, 2002), (Vargas *et al.*, 2004c), (Vargas *et al.*, 2004b), (Vargas *et al.*, 2004e) e (Vargas *et al.*, 2004d).

Finalmente, este capítulo é composto pelos resultados da validação do modelo dinâmico, os projetos e resultados experimentais dos controladores *TDOF*, *VSC* e *FLC*, a determinação dos coeficientes de atrito e os resultados considerando a compensação do atrito.

6.1 Validação do Modelo Dinâmico

Para a implementação experimental dos controladores apresentados no capítulo 4, inicialmente é feita a validação do modelo dinâmico do robô *SCARA*.

A matriz de inércia $M(q)$ e o vetor de forças gravitacionais $G(q)$ do robô são dados pela equação

(6.1) e o vetor dos torques centrífugos e de Coriolis $C(q, \dot{q})\dot{q}$ é dado pela equação (6.2).

$$M(q) = \begin{bmatrix} m_{0,0} & m_{0,1} & 0 & I_3 \\ m_{1,0} & m_{1,1} & 0 & I_3 \\ 0 & 0 & m_2 + m_3 & 0 \\ I_3 & I_3 & 0 & I_3 \end{bmatrix} \quad G(q) = \begin{bmatrix} 0 \\ 0 \\ (m_2 + m_3)g \\ 0 \end{bmatrix} \quad (6.1)$$

$$C(q, \dot{q})\dot{q} = \begin{bmatrix} 2(m_1 l_0 l_{c_1} + (m_2 + m_3) l_0 l_1) \dot{q}_0 \dot{q}_1 \sin(q_1) + (m_1 l_0 l_{c_1} + (m_2 + m_3) l_0 l_1) \dot{q}_1^2 \sin(q_1) \\ (m_1 l_0 l_{c_1} + (m_2 + m_3) l_0 l_1) \dot{q}_0^2 \sin(q_1) \\ 0 \\ 0 \end{bmatrix} \quad (6.2)$$

As componentes da matriz $M(q)$ são dadas pela equação (6.3). Os parâmetros construtivos do robô são apresentados na tabela 6.1.

$$\begin{aligned} aux_1 &= I_0 + I_1 + I_2 + I_3 + (m_1 + m_2 + m_3)l_0^2 + m_1 l_{c_1}^2 + (m_2 + m_3)l_1^2 + m_0 l_{c_0}^2 \\ m_{0,0} &= aux_1 + 2(m_1 l_0 l_{c_1} + (m_2 + m_3) l_1 l_0) \cos(q_1) \\ m_{0,1} &= m_{1,0} = I_1 + I_2 + I_3 + (m_2 + m_3)l_1^2 + m_1 l_{c_1}^2 \\ &\quad + (m_1 l_0 l_{c_1} + (m_2 + m_3) l_1 l_0) \cos(q_1) \\ m_{1,1} &= I_1 + I_2 + I_3 + m_1 l_{c_1}^2 + (m_2 + m_3)l_1^2 \end{aligned} \quad (6.3)$$

Junta	I_i [Kgm^2]	m_i [Kg]	l_i [m]	l_{c_i} [m]
0	0.23	11.4	0.25	0.118
1	0.16	19.5	0.25	0.116
2	0.1	2	0.01	0.0
3	0.1	1.5	0.01	0.0

Tabela 6.1: Parâmetros do robô industrial *SCARA*

O controlador Proporcional Derivativo (*PD*) existente no robô foi empregado para a validação experimental do modelo. A estrutura deste controlador é ilustrada na figura 6.1.

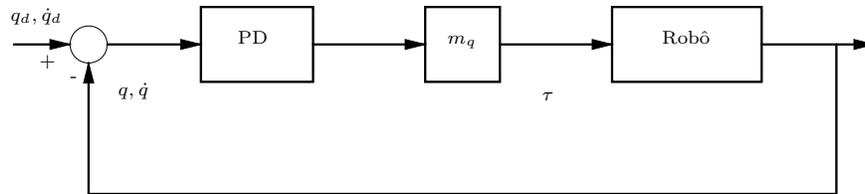


Figura 6.1: Estrutura de controle PD

A lei de controle no espaço de juntas é dada por:

$$\tau_i = (\omega_i^2(q_{d_i} - q_i) + 2\omega_i(\dot{q}_{d_i} - \dot{q}_i)) m_{q_i} \quad (6.4)$$

onde q_{d_i} , \dot{q}_{d_i} , q_i e \dot{q}_i correspondem aos valores de posição e velocidade desejados e reais em cada uma das juntas. Os ganhos ω_i e m_{q_i} foram determinados considerando um sistema criticamente amortecido (Hüppi e Gruener, 2001).

$$\omega_i = \begin{bmatrix} 70^2 & 110^2 & 300^2 & 120^2 \end{bmatrix} \quad (6.5)$$

$$m_{q_i} = \begin{bmatrix} 1.7 & 0.2 & 4.5 & 0.0059 \end{bmatrix} \quad (6.6)$$

As figuras 6.2 e 6.3 apresentam os resultados de simulação e experimentais para cada um dos eixos cartesianos x, y, z e a orientação θ do robô. A simulação foi feita empregando os dados experimentais obtidos para diferentes tipos de movimentos. Neste ponto é importante observar que se bem o controlador PD é projetado no espaço das juntas, os resultados apresentados no espaço cartesiano foram obtidos empregando a cinemática direta e inversa detalhada no apêndice A.

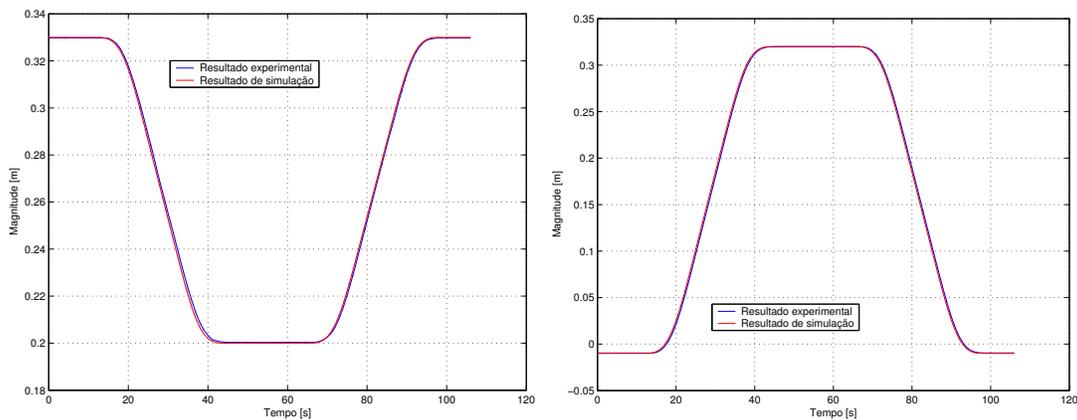


Figura 6.2: Respostas simulada e experimental nas direções x e y

Na figuras 6.2 e 6.3, observa-se que as direções x , y e z apresentam uma boa aproximação no que se refere aos resultados de simulação e experimentais. Isto não ocorre com o eixo de orientação que apresenta erros na modelagem e incertezas como pode ser verificado na figura 6.3.

Ao contrário do que ocorre quando são feitas simulações, na prática, as incertezas já se encontram implícitas no próprio robô e não é necessário estabelecer aproximações para que elas sejam consideradas. Este é o principal motivo para ter sido feita esta validação.

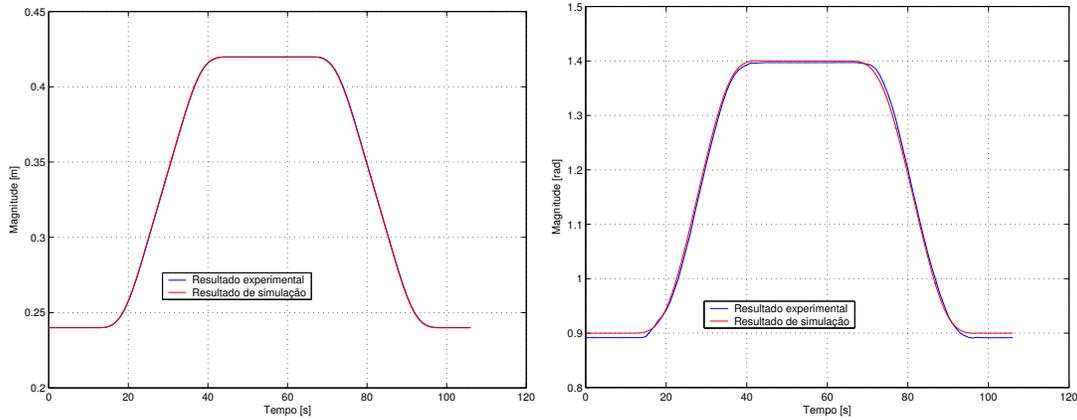


Figura 6.3: Respostas simulada e experimental nas direções z e θ

6.2 $TDOF$ Aspectos de Projeto e Resultados

Para o projeto do controlador $TDOF$, existe a necessidade de obter um ou vários modelos lineares do robô, a partir de seu modelo multivariável, não-linear e acoplado. Como apresentado em Vargas *et al.* (2000) e Vargas *et al.* (2001b), alguns destes modelos podem ser estabelecidos utilizando-se a lei da dinâmica inversa ou através de aproximações.

A partir da modelagem de De Luca e Manes (1994), a equação do robô executando movimentos não restritos em cada uma das juntas é dada por:

$$\sum_{j=1}^n m_{ij}(q)\ddot{q}_j + \sum_{j=1}^n \sum_{k=1}^n c_{ijk}(q)\dot{q}_k\dot{q}_j + g_i(q) + f_i(\dot{q}) = \tau_i \quad (6.7)$$

onde todos os termos são definidos em Sciavicco e Siciliano (2000).

Para a obtenção de um modelo linear, é necessário desacoplar a equação (6.7), o que é alcançado expressando-a somente em função da variável q_i , de suas derivadas e de coeficientes constantes. Isto pode ser obtido, limitando ou eliminando as variáveis q_j ($j \neq i$) e suas derivadas (Sciavicco e Siciliano, 2000). Neste sentido, as seguintes hipóteses são estabelecidas:

Hipótese 6.2.1 *São considerados robôs manipuladores que têm os coeficientes c_{iii} iguais a zero (isto é, os termos \dot{q}_i^2 da equação (6.7) não existem).*

Hipótese 6.2.2 *Considera-se que, quando a junta i está sendo posicionada, as outras juntas já estão na posição desejada. Isto é, se a posição desejada da junta j ($j \neq i$) é a posição fixa q_{d_j} , então:*

$$\dot{q}_j = 0 \quad ; \quad \ddot{q}_j = 0 \quad j = 1, \dots, n (j \neq i) \quad (6.8)$$

Substituindo-se $\dot{q}_j = \ddot{q}_j = 0$ na equação (6.7) e considerando-se a hipótese 6.2.1, obtém-se:

$$m_{ii}\ddot{q}_i + g_i + f_i = \tau_i \quad i = 1, \dots, n \quad (6.9)$$

A equação (6.9) é empregada para o projeto dos controladores $TDOF$, com $f_i = 0$ e considerando-se o valor máximo da diagonal da matriz de inércia $m_{ii} = \bar{m}_{ii}$. Por outro lado, esta equação também é empregada para fazer a identificação experimental dos parâmetros estáticos de atrito no modelo de *LuGre*, assumindo-se $g_i = 0$ e considerando-se a velocidade constante ($\ddot{q}_i = 0$) (Vargas *et al.*, 2004c).

Uma vez obtido o modelo linear, os pólos ótimos que se desejam alocar são determinados empregando o algoritmo LQR , conforme apresentado no capítulo 4. Estes pólos são obtidos variando-se os fatores de ponderação ρ e γ . Para a junta 2, os valores de ρ e γ são iguais a 10^{11} e 10, respectivamente. O polinômio \bar{p}_r corresponde a um derivador. Os pólos ótimos obtidos para esta junta são ilustrados na figura 6.4. Neste figura, observa-se que um dos pólos encontra-se no eixo real e os outros dois correspondem a pólos complexos conjugados. Os valores dos pólos são fornecidos na tabela 6.2.

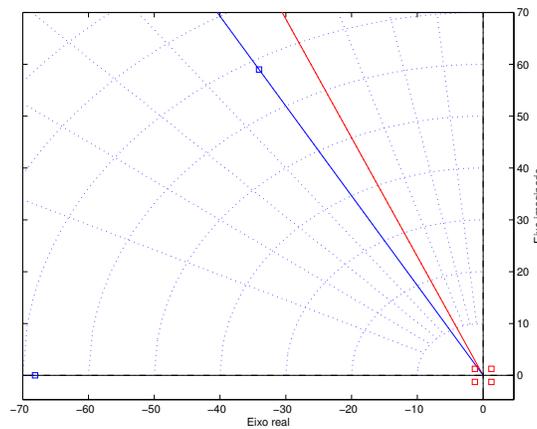


Figura 6.4: Pólos ótimos

δ^{F^*} para $\rho = 10^{11}$	δ^{H^*} para $\gamma = 10$
$-34.0646 \pm 59.0016j$	$-1.2574 \pm 1.2574j$
-68.1292	

Tabela 6.2: Valores dos pólos ótimos

Neste ponto, para cumprir os objetivos de projeto, duas modificações foram introduzidas no algoritmo LQR .

1. A existência de pólos complexos conjugados no conjunto de pólos ótimos e os requisitos do sistema permitem o uso destes pólos, ou somente da componente real, para a determinação dos ganhos $h(s)$, $q(s)$ e $k(s)$;
2. Os zeros da função de transferência $\frac{q(s)}{k(s)}$ podem ser modificados empregando-se a seguinte igualdade:

$$\tilde{q}^*(s) = s^2 + \Theta(\tilde{q}_1 s + \tilde{q}_2) \quad (6.10)$$

$$q(s) = \tilde{q}^*(s)\bar{q}(s) \quad (6.11)$$

tal que Θ varia entre 0.5 e 1.

Introduzidas estas modificações, dois casos são projetados e analisados para a junta 2. Os valores selecionados dos pólos e da constante Θ para ambos os casos são apresentados na tabela 6.3.

Tabela 6.3: Pólos selecionados

Caso	Pólos	Θ
a	$\Re(-34.0646 + 59.0016j)$ $\Re(-1.2574 + 1.2574j)$	1.0
b	$\Re(-34.0646 + 59.0016j)$ $\Re(-1.2574 + 1.2574j)$	0.965

Os ganhos obtidos para os dois casos são apresentados na tabela 6.4.

Tabela 6.4: Ganhos e polinômios obtidos

Caso	a	b
$h(s)$	$0.0945 \times 10^5 s^2 + 3.2790 \times 10^5 s + 3.9764 \times 10^5$	$0.0945 \times 10^5 s^2 + 3.2790 \times 10^5 s + 3.9764 \times 10^5$
$q(s)$	$0.0945 \times 10^5 s^2 + 3.3374 \times 10^5 s + 3.9764 \times 10^5$	$0.0945 \times 10^5 s^2 + 3.1825 \times 10^5 s + 3.9764 \times 10^5$
$k(s)$	$1.0000s^2 + 137.5158s + 0$	$1.0000s^2 + 137.5158s + 0$

Os resultados da análise obtidos via simulação correspondentemente à resposta ao degrau e o seguimento da referência (esforço de controle inicial) são ilustrados na figura 6.5.

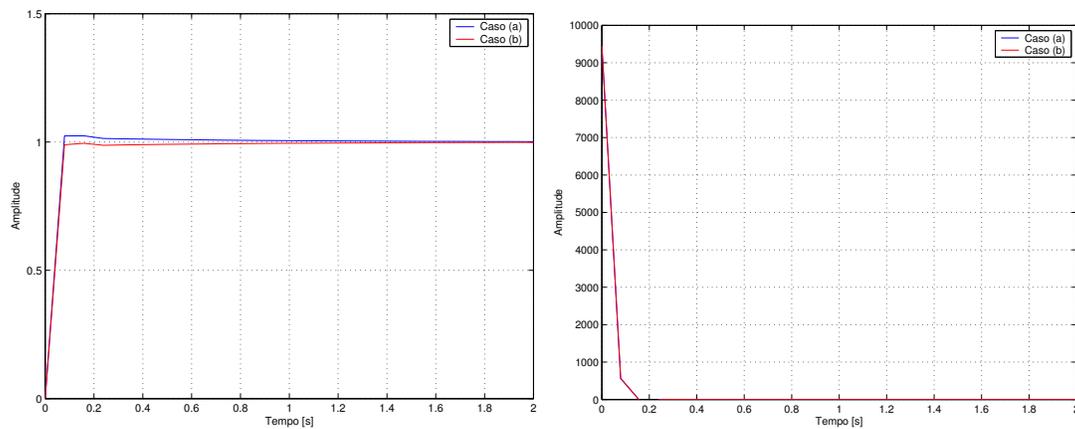


Figura 6.5: Resposta ao degrau e esforço de controle

A função de sensibilidade $S(s)$, complementar de sensibilidade $C(s)$ e o ganho de malha $L(s)$ são ilustrados na figura 6.6.

É sabido que em aplicações em robótica, na maioria dos casos, a existência de sobre-sinal é indesejada pois pode resultar em colisões e em danos para o manipulador ou para as peças que estão sendo manipuladas. Neste sentido, na resposta ao degrau ilustrada na figura 6.5, observa-se que o primeiro caso apresenta sobre-sinal, que é eliminado no segundo caso, ao mudar o valor da constante Θ . O valor que requer uma análise cuidadosa corresponde ao esforço de controle ilustrado na figura 6.5, já que o mesmo não pode ser elevado, porque poderia levar a danos físicos dos atuadores ou à

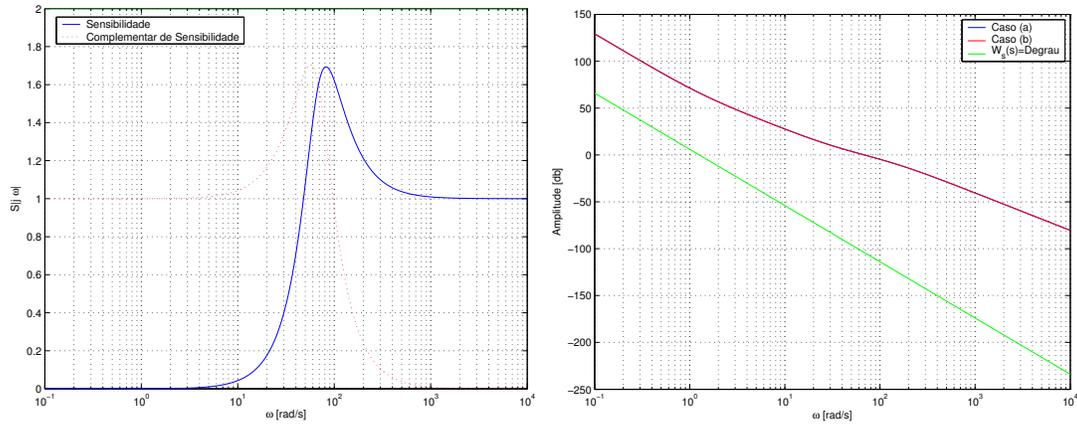


Figura 6.6: Função de sensibilidade, complementar de sensibilidade e ganho de malha

saturação. O esforço de controle nos dois casos é o mesmo. Na figura 6.6, observa-se que a magnitude da função de sensibilidade \bar{S} é menor que 2, o que garante estabilidade em relação às variações paramétricas e, junto com a complementar de sensibilidade $C(s)$, estabelece um compromisso entre as altas e baixas frequências. O ganho de malha se encontra mais afastado com respeito à função $W_s(s)$ das perturbações (Perturbações tipo degrau foram consideradas, sendo que estas podem corresponder a uma colisão com um obstáculo ou ao torque proveniente de uma peça mal posicionada). Desta forma, o controlador garante uma boa rejeição destas perturbações. Finalmente, verifica-se que a mudança introduzida melhora as características do sistema ao diminuir o sobre-sinal.

A implementação digital dos controladores *TDOF* foi feita através do método de aproximação trapezoidal tal como apresentado no apêndice D. Esta aproximação foi empregada no sentido de mapear todos os pólos alocados no semiplano esquerdo dentro do círculo unitário. Os equivalentes digitais dos casos analisados são apresentados na tabela 6.5.

Tabela 6.5: Ganhos dos controlador *TDOF* - caso digital

Caso	a	b
$h(z)$	$3.8474 \times 10^4 z^2 - 7.5635 \times 10^4 z + 3.7250 \times 10^4$	$3.8474 \times 10^4 z^2 - 7.5635 \times 10^4 z + 3.7163 \times 10^4$
$q(z)$	$3.8486 \times 10^4 z^2 - 7.5635 \times 10^4 z + 3.7151 \times 10^4$	$3.8464 \times 10^4 z^2 - 7.5635 \times 10^4 z + 3.7173 \times 10^4$
$k(z)$	$4.2750 z^2 - 8.0000 z + 3.7250$	$4.2750 z^2 - 8.0000 z + 3.7250$

Este controlador foi implementado a uma frequência de amostragem de 1000 [Hz]. Os módulos **RobotControl.Mod** e **ScaraRobot.Mod** foram modificados, sendo que os seguintes procedimentos foram criados:

- `PROCEDURE(robot:Robot)InitDO*():BOOLEAN`: Este procedimento inicializa os ganhos digitais do controlador e os valores das amostras passadas até $(n - 2)$;
- `PROCEDURE(h:ControlDOHdl)Run()`: Este procedimento implementa o controlador *TDOF*;

As trajetórias reais e desejadas assim como as velocidades para os eixos x e y são ilustradas nas

figuras 6.7 e 6.8. Da mesma forma, nas figuras 6.9 e 6.10, são ilustradas as trajetórias e velocidades para os eixos z e θ .

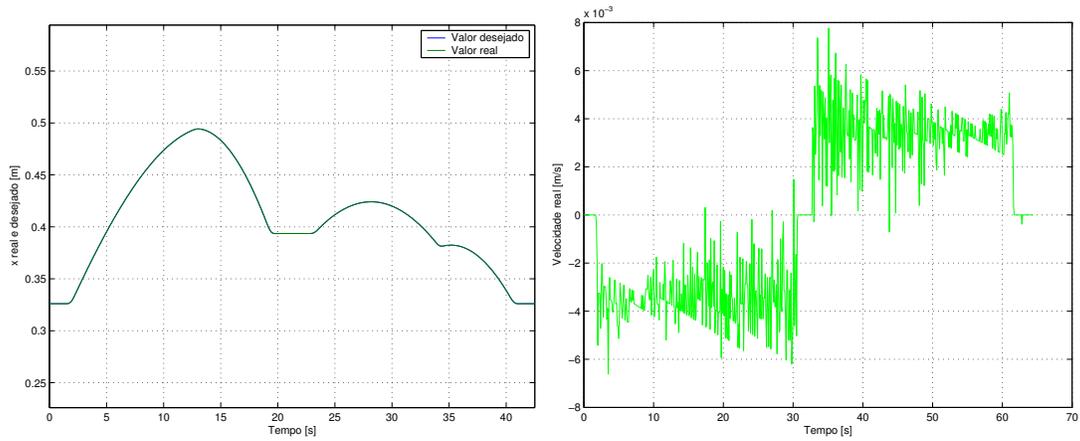


Figura 6.7: Trajetória e velocidade desejadas eixo x

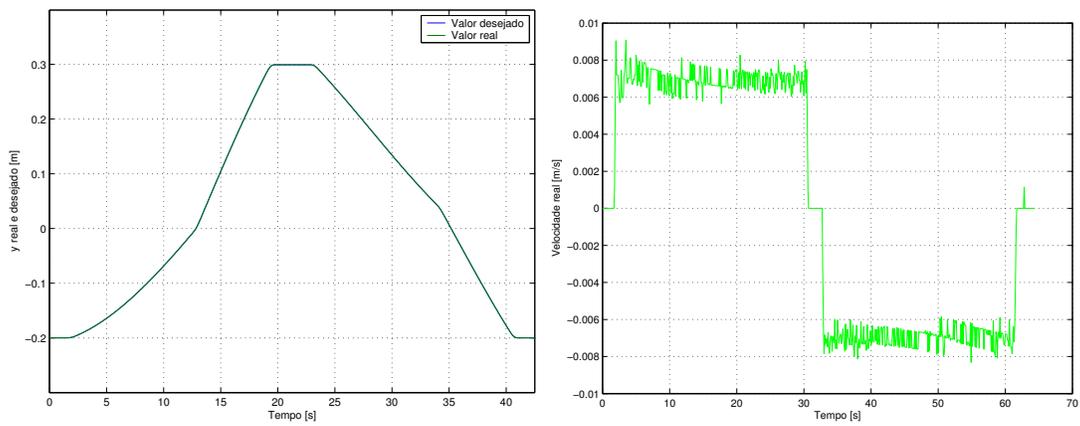


Figura 6.8: Trajetória e velocidade desejadas eixo y

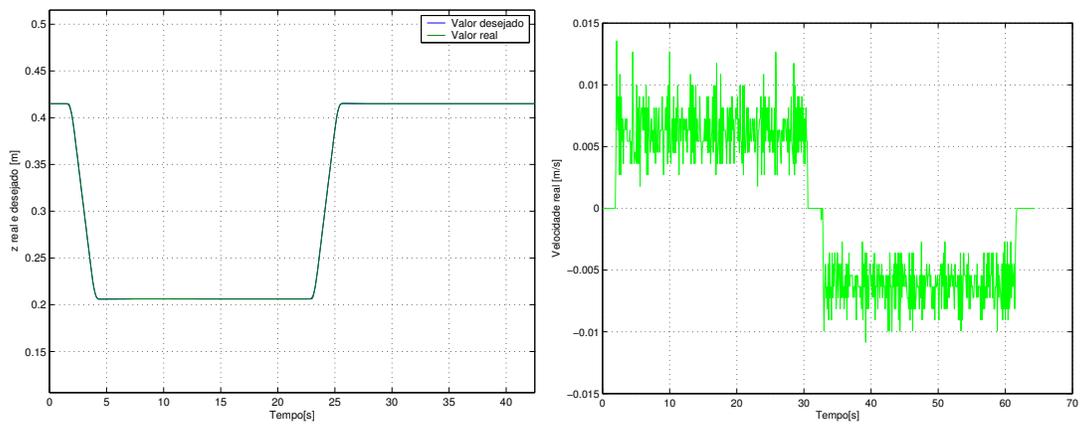


Figura 6.9: Trajetória e velocidade desejadas eixo z

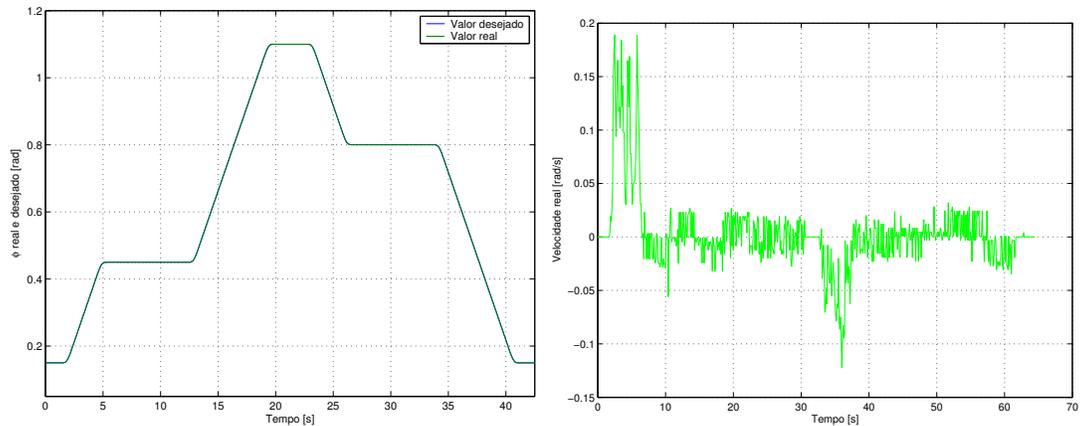


Figura 6.10: Trajetória e velocidade desejadas eixo θ

As quatro trajetórias e velocidades correspondem a polinômios incrementais de terceira ordem (Hüppi e Gruener, 2001). Observa-se que os valores de velocidade variam na ordem de $0.01 [m/s|rad/s]$ nos quatro eixos. Estas mesmas trajetórias e velocidades foram utilizadas para os controladores *VSC* e *FLC*.

Os erros de seguimento e os esforços de controle alcançados após os experimentos para os eixos x e y são ilustrados nas figuras 6.11 e 6.12. Destas figuras observa-se que os valores obtidos para os erros são da ordem de 10^{-4} nos dois eixos e os esforços de controle correspondem a $\pm 10 [Nm]$ e $\pm 6 [Nm]$ respectivamente. Estes valores permitem verificar erros reduzidos para velocidades baixas e uma ampla margem no esforço de controle quando comparados com os valores limite.

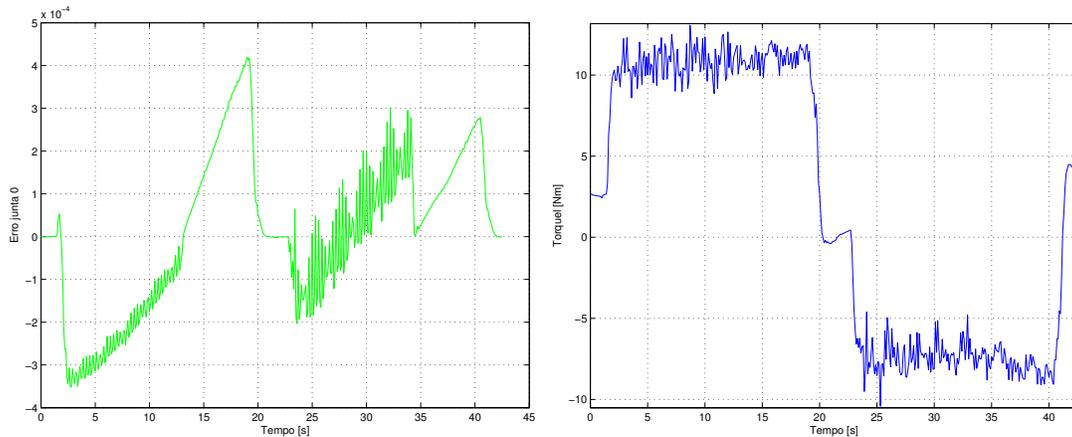
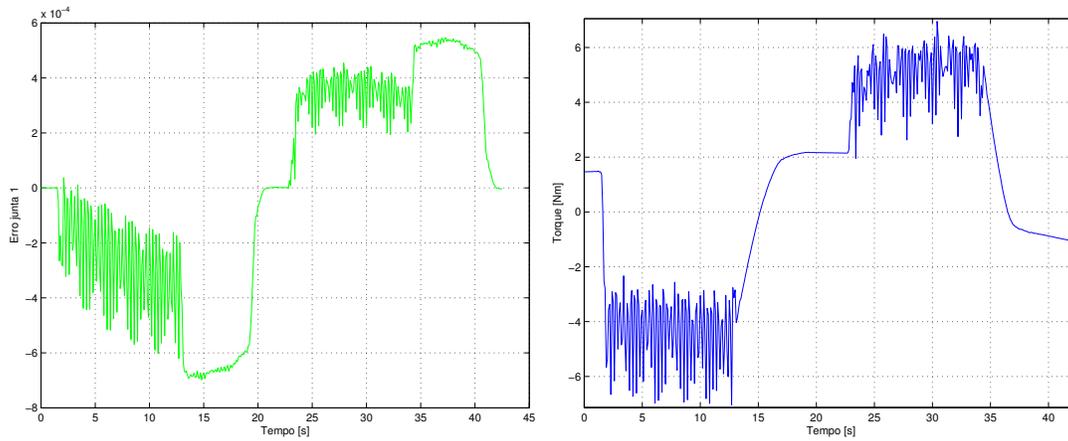
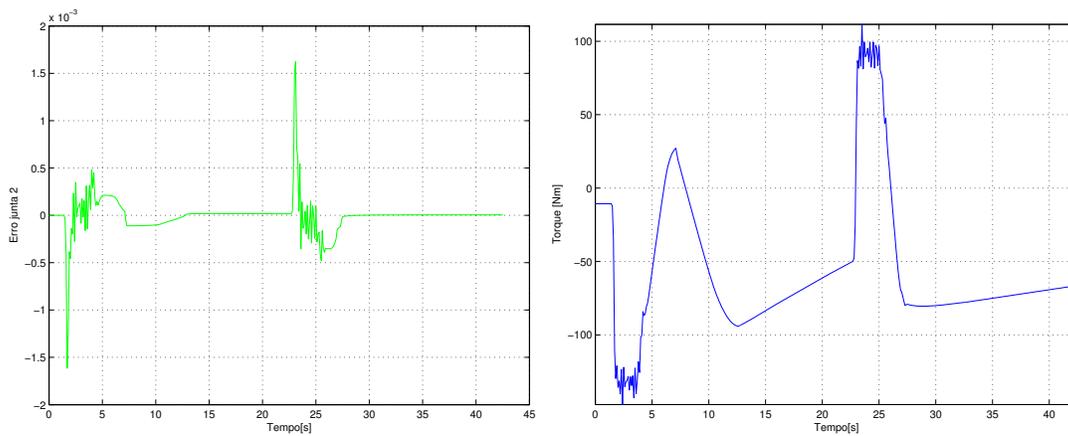
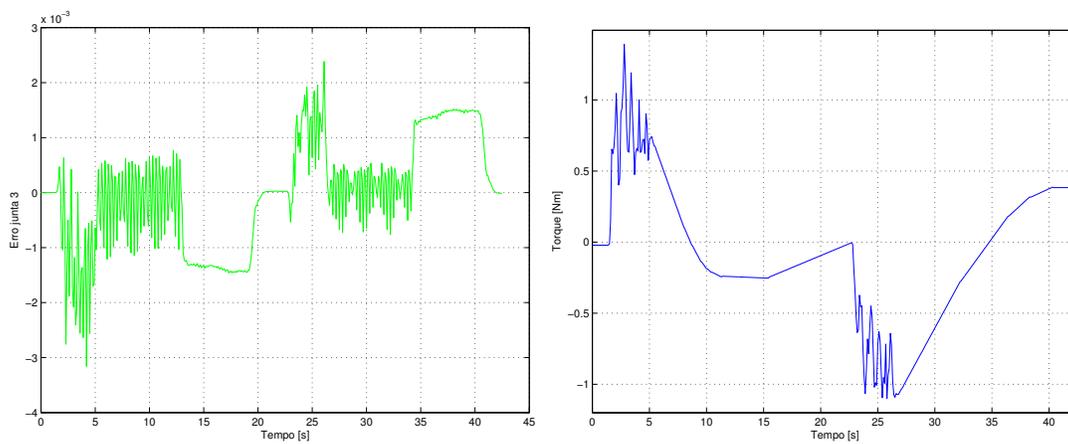


Figura 6.11: Erro e esforço de controle no eixo x

Os erros e esforços de controle para os eixos z e θ são apresentados nas figuras 6.13 e 6.14. Como nos outros eixos, os valores obtidos para os erros variam na ordem de 10^{-4} e 10^{-3} para os eixos z e θ . Os esforços de controle são de $\pm 130 [Nm]$ e $\pm 1 [Nm]$. Os esforços de controle também se encontram longe dos valores limite.

Figura 6.12: Erro e esforço de controle no eixo y Figura 6.13: Erro e esforço de controle no eixo z Figura 6.14: Erro e esforço de controle no eixo θ

6.3 VSC Aspectos de Projeto e Resultados

No projeto do controle a estrutura variável (*VSC* em inglês), desenvolvido neste trabalho, foram escolhidas superfícies de deslizamento lineares, escritas em função dos erros de posição e dos erros de velocidade. Estas superfícies são dadas por:

$$\sigma_i = c_i e_i + \dot{e}_i \quad (6.12)$$

onde c_i são os ganhos escolhidos para assegurar que σ_i seja uma superfície de deslizamento atrativa, e_i e \dot{e}_i são respectivamente os erros de posição e de velocidade. Estes erros são definidos como:

$$e_i = q_i - q_{d_i} \quad (6.13)$$

$$\dot{e}_i = \dot{q}_i - \dot{q}_{d_i} \quad (6.14)$$

A lei de controle implementada é:

$$\tau_i = - \left(\frac{\partial \sigma_i}{\partial t} + \varpi_i \text{sign}(\sigma_i) + \varrho_i \sigma_i \right) \quad (6.15)$$

onde a derivada $\frac{\partial \sigma_i}{\partial t}$ é dada por:

$$\frac{\partial \sigma_i}{\partial t} = \left[-\ddot{q}_{d_i} + c_{k_i} (\dot{q}_i - \dot{q}_{d_i}) \right] \quad (6.16)$$

O problema do *chattering* é solucionado empregando-se o conceito de camada limite, onde a função *sgn* é substituída pela função *tanh*:

$$\tau_i = - \left(\frac{\partial \sigma_i}{\partial t} + \varpi_i \tanh\left(\frac{\sigma_i}{\phi}\right) + \varrho_i \sigma_i \right) \quad (6.17)$$

Este controlador foi implementado a uma frequência de amostragem de 1000 [Hz]. Para a implementação foram modificados os módulos **RobotControl.Mod** e **MathLib.Mod**. Para a implementação do controlador e a execução dos experimentos, os seguintes procedimentos foram criados:

- `PROCEDURE(h:ControlVSChdl)Run()`: Este procedimento implementa os controladores em questão;
- `PROCEDURE tanh*(x:LONGREAL):LONGREAL`: Este procedimento implementa a função *tanh* para contornar o problema do *chattering* de forma a ter um chaveamento menos frequente do controlador;

Para propósitos de comparação com as outras duas estratégias, as trajetórias empregadas nos experimentos são idênticas a aquelas ilustradas nas figuras 6.7-6.13. A sintonia dos ganhos foi feita de

forma a obter um sistema criticamente amortecido $\xi = 1$ tal como ilustrado na figura 6.15. Esta figura corresponde aos resultados de simulação da primeira junta do robô, onde observa-se que a resposta ao degrau unitário não apresenta sobre-sinal e o esforço de controle, ao contrário do que ocorre com o controlador *TDOF*, é de 4 [Nm], valor muito abaixo do valor limite. Neste ponto, é importante comentar que as implementações ao longo deste trabalho foram feitas de forma a produzir uma resposta coerente com as aplicações em robótica, sem sobre-sinal e evitando a saturação dos atuadores.

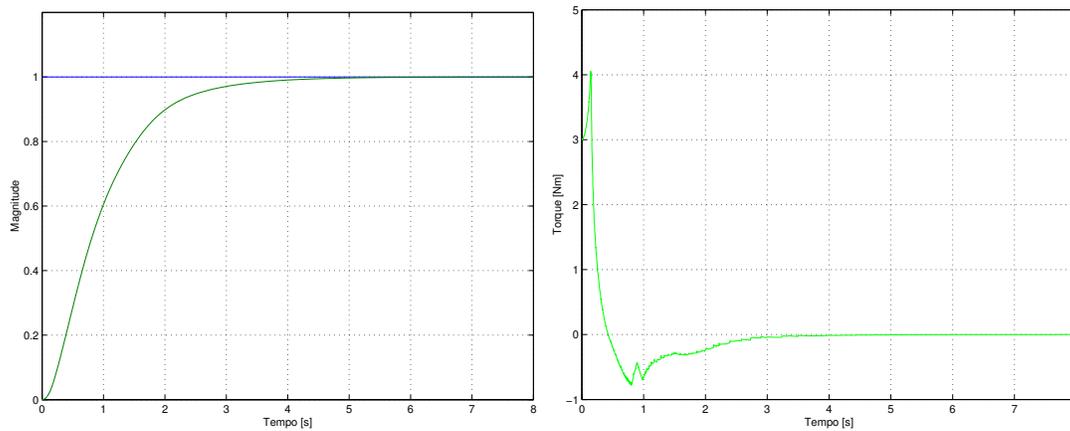


Figura 6.15: Requisitos de projeto para os controladores VSC e FLC

Para este controlador as medições são complementadas com a estimação da velocidade a partir das medições de posição, para cujo propósito é utilizada a seguinte expressão:

$$\dot{q}(k) = \frac{1}{h} \sum_{i=1}^{h-1} \dot{q}(k-i) + \frac{q(k) - q(k-1)}{\Delta T} \quad (6.18)$$

para $h = 5$ (h corresponde ao número de amostras passadas).

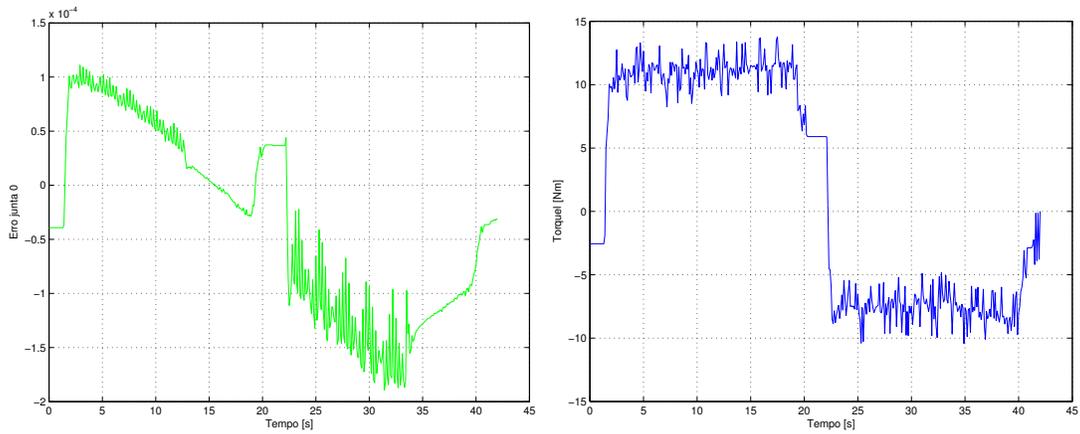
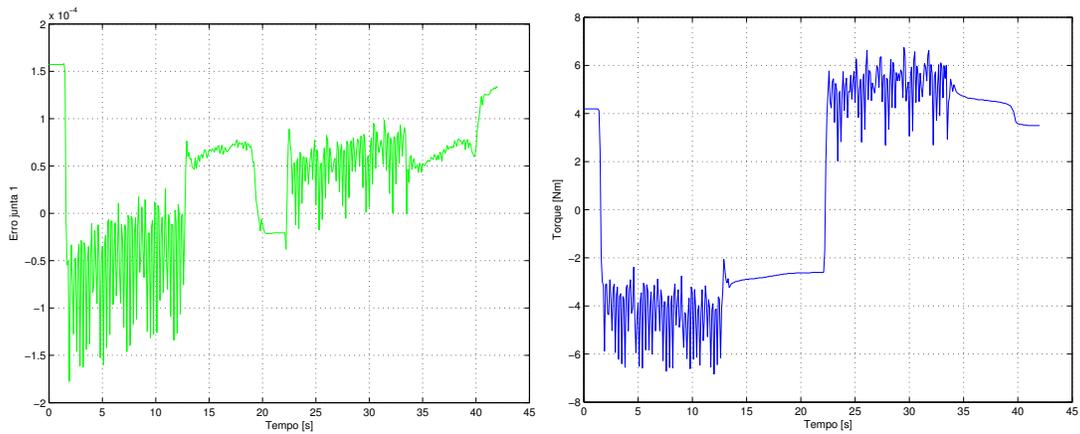
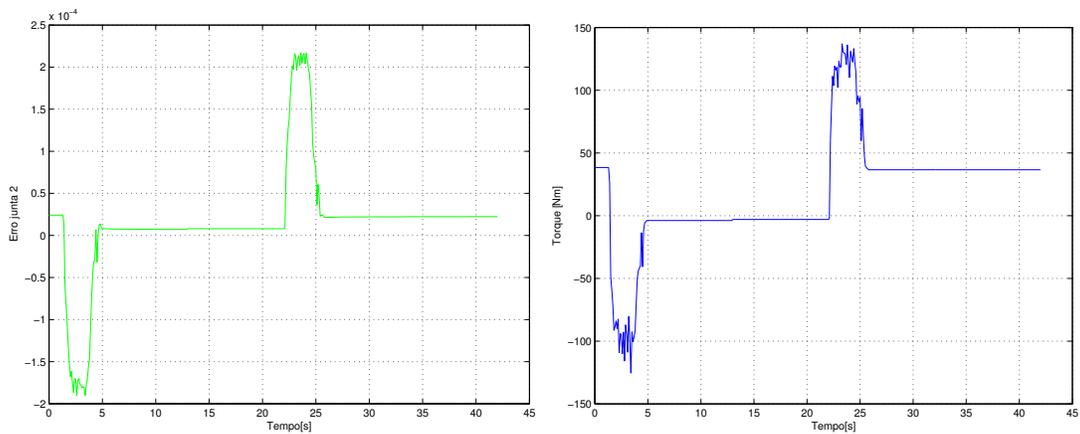
Os ganhos para as quatro juntas são apresentados na tabela 6.6.

Tabela 6.6: Parâmetros de projeto *VSC*

Junta	c_i	ρ_i	ϖ_i	ϕ_i
0	20	20	25	0.8
1	120	200	150	0.8
2	100	80	130	0.8
3	200	180	220	0.8

Os erros de seguimento e os esforços de controle obtidos após os experimentos para os eixos x e y são ilustrados nas figuras 6.16 e 6.17.

Destes resultados observa-se que a magnitude do erro é muito similar ao caso do controlador *TDOF* que é da ordem de 10^{-4} , enquanto os esforços de controle são de ± 10 [Nm] e ± 6 [Nm], respectivamente. Desta forma, verifica-se que os esforços de controle encontram-se longe da saturação, como desejado. Já os resultados dos erros de seguimento e os esforços de controle para os eixos z e ϕ

Figura 6.16: Erro e esforço de controle no eixo x Figura 6.17: Erro e esforço de controle no eixo y Figura 6.18: Erro e esforço de controle no eixo z

são apresentados nas figuras 6.18 e 6.19. Observa-se que a magnitude dos erros é da ordem de 10^{-4} e 10^{-3} para estes eixos. Os esforços de controle encontram-se na faixa de ± 130 [Nm] e ± 1 [Nm] respectivamente.

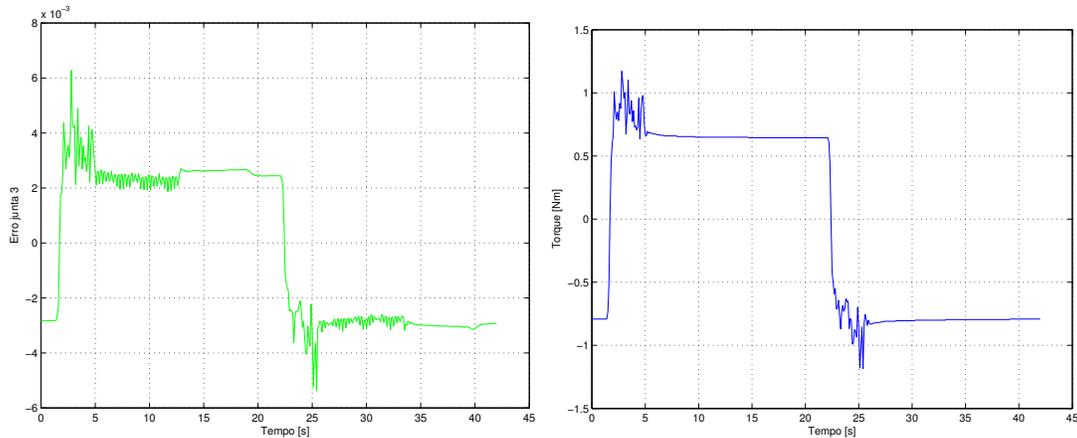


Figura 6.19: Erro e esforço de controle no eixo θ

6.4 FLC Aspectos de Projeto e Resultados

Da mesma forma que os outros dois controladores, este também foi implementado a uma frequência de 1000 [Hz]. Para sua implementação, os seguintes módulos e procedimentos foram criados:

- O modulo MathFuzzy.Mod foi criado para implementar os procedimentos de pertinência, max, fuzzificação e de-fuzzificação;
- PROCEDURE Pertinência* (erro: LONGREAL; X, A: ARRAY OF LONGREAL; PDiscretos: LONGINT; VAR pertinencia: LONGREAL): Este procedimento é usado para calcular o valor pertinente com respeito aos valores de e e \dot{e} da entrada baseado no universo de discurso X e o conjunto de pertinência A ;
- PROCEDURE Max*(a,b : LONGREAL) : LONGREAL: Este procedimento retorna o maior valor entre dois valores comparados;
- PROCEDURE Inferência* (w : LONGREAL; C : ARRAY OF LONGREAL; PDiscretos:LONGINT; VAR O :ARRAY OF LONGREAL): Este procedimento é usado para realizar o calculo de inferência baseado em A_o , B_o e C . O resultado é armazenado em O ;
- PROCEDURE DeFuzzyCG*(Y,B :ARRAY OF LONGREAL;PDiscretos:LONGINT; VAR Y_o : LONGREAL): Este procedimento é usado para desnebulizar o universo Y baseado no conjunto Fuzzy B , usando o teorema do centro da gravidade (*Centróide*). O resultado é armazenado em Y_o ;

- PROCEDURE(h:ControlFuzzyHdl)Run(): Este procedimento implementa o controlador *FLC* no módulo Robot.Mod;
- As funções de inferência utilizadas correspondem a funções triangulares;
- O procedimento de inferência empregado é o do *max*;
- A desnebulização é feita empregando o método do centro de gravidade;
- As características de projeto correspondem às mesmas utilizadas para o controle *VSC*, ou seja, obter um sistema criticamente amortecido;
- A velocidade é obtida empregando a mesma aproximação que no caso do controle *VSC*.

Os parâmetros de projeto dos *FLC*'s de cada uma das juntas do robô são apresentados nas tabelas 6.7 e 6.8. Os primeiros resultados experimentais deste controlador foram publicados em (Vargas *et al.*, 2002).

Tabela 6.7: Parâmetros de projeto para e e \dot{e}

Junta	G_e	$G_{\dot{e}}$	$NB_{E,DE}$	$NS_{E,DE}$	$ZR_{E,DE}$	$PS_{E,DE}$	$PB_{E,DE}$
0	40	6	-1.0	-0.5	0.0	0.5	1.0
1	90	5	-1.0	-0.5	0.0	0.5	1.0
2	280	4	-1.0	-0.5	0.0	0.5	1.0
3	320	3	-1.0	-0.5	0.0	0.5	1.0

Tabela 6.8: Parâmetros de projeto para τ

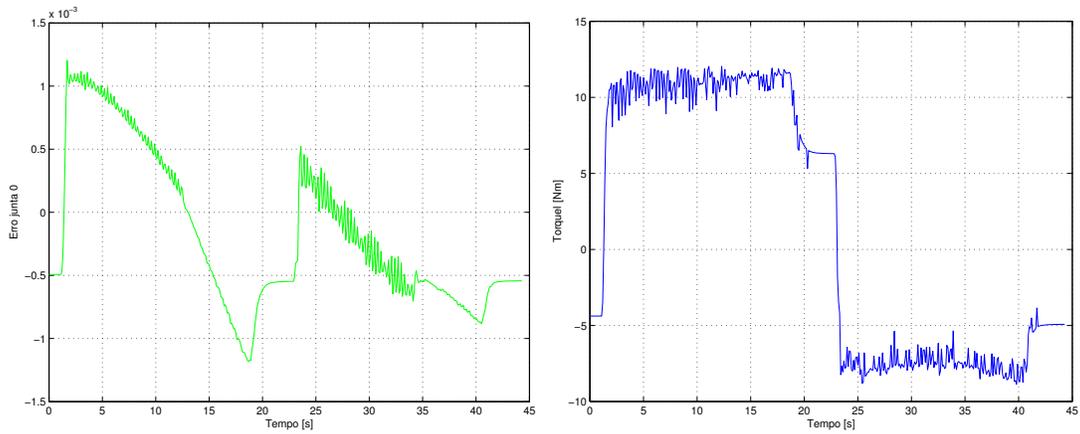
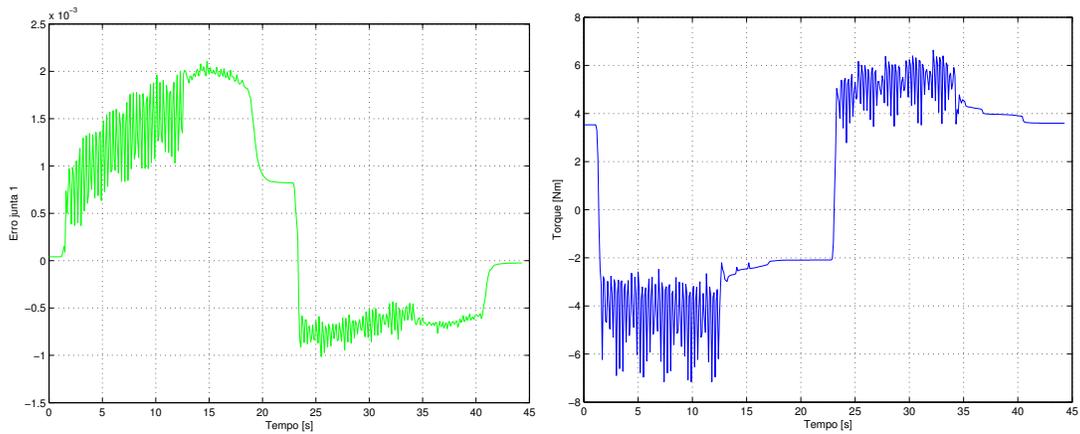
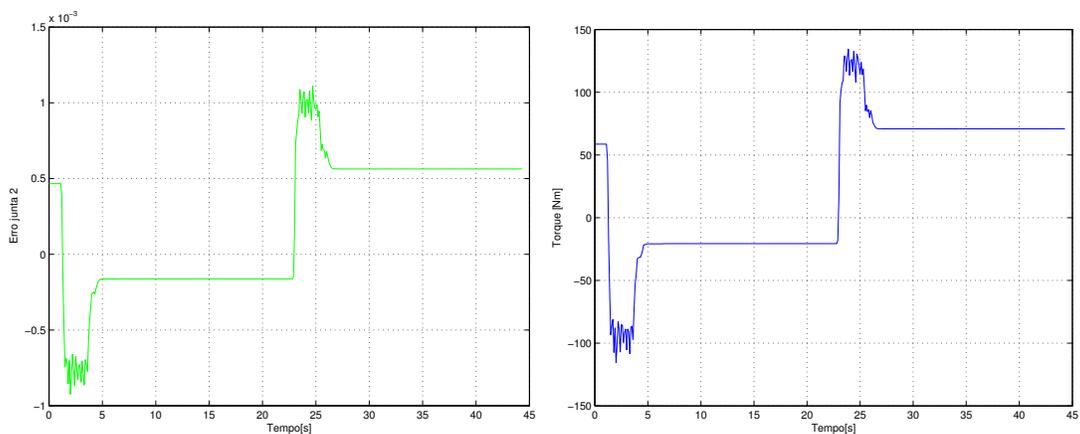
Junta	G_τ	CNB	CNS	CZR	CPS	CPB
0	30	-1	-0.6	0.0	0.6	1.0
1	65	-1	-0.6	0.0	0.6	1.0
2	90	-1	-0.6	0.0	0.6	1.0
3	360	-1	-0.6	0.0	0.6	1.0

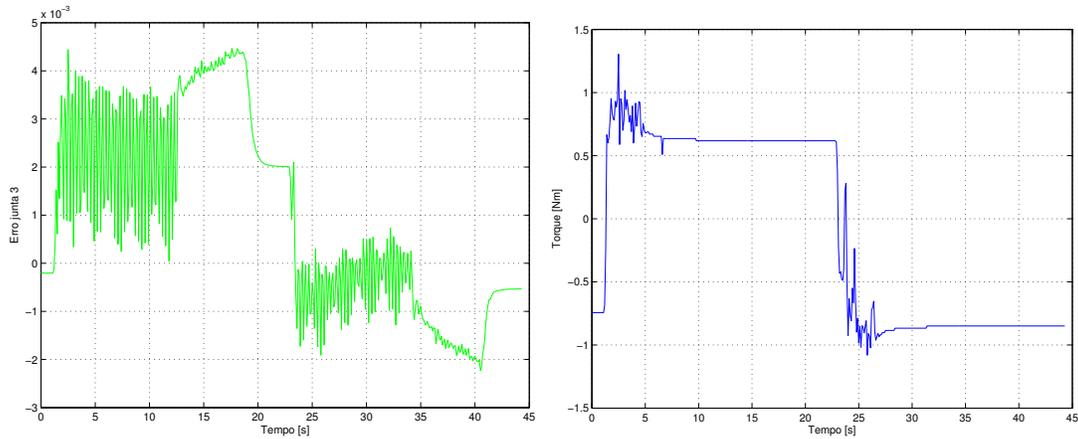
Os erros de seguimento e os esforços de controle do *FLC* para os eixos x e y são apresentados nas figuras 6.20 e 6.21. Destes resultados, observa-se que se bem a magnitude dos esforços de controle são similares às técnicas anteriores a magnitude dos erros é da ordem de 10^{-3} . Obviamente ainda que a magnitude do erro seja maior, esta técnica apresenta a vantagem de eliminar em forma efetiva o fenômeno do *chattering*.

Os resultados obtidos para os eixos z e θ são apresentados nas figuras 6.22 e 6.23. Da mesma forma que os eixos x e y a magnitude do erro é da ordem de 10^{-3} e a magnitude dos esforços de controle similar às técnicas *TDOF* e *VSC*.

Por termos evidenciado resultados similares entre as três estratégias implementadas, as mesmas foram avaliadas considerando-se o índice do erro absoluto no espaço cartesiano dado por:

$$|e_{CE}| = \sqrt{\sum_{j=1}^n (x_d(j) - x(j))^2} \quad (6.19)$$

Figura 6.20: Erro e esforço de controle no eixo x Figura 6.21: Erro e esforço de controle no eixo y Figura 6.22: Erro e esforço de controle no eixo z

Figura 6.23: Erro e esforço de controle no eixo θ

As tabelas 6.9 e 6.10 apresentam os índices obtidos para os experimentos feitos a baixas velocidades (resultados apresentados) e velocidades médias (cujos resultados podem ser verificados em Vargas *et al.* (2004b)).

Tabela 6.9: Erros para velocidade de $0.01 \left[\frac{m}{s} \mid \frac{rad}{s} \right]$

Eixo	<i>FLC</i>	<i>TDOF</i>	<i>VSC</i>
x	0.0132	0.0038	0.0019
y	0.0234	0.0084	0.0015
z	0.0103	0.0040	0.0013
θ	0.0473	0.0207	0.0574

Tabela 6.10: Erros para velocidade $0.2 \left[\frac{m}{s} \mid \frac{rad}{s} \right]$

Eixo	<i>FLC</i>	<i>TDOF</i>	<i>VSC</i>
x	0.0122	0.0076	0.0010
y	0.0191	0.0186	0.0011
z	0.0086	0.0044	0.0007
θ	0.0525	0.0328	0.0301

Dos resultados obtidos, observa-se que os controladores *VSC* e *TDOF* apresentam um melhor índice que o *FCL*, tanto para baixas velocidades quanto para velocidades médias. Em termos de esforço de controle as características entre os três controladores são similares conforme foi verificado. Na técnica *VSC*, o problema de *chattering* foi tratado empregando a camada limite e implementando a função *tanh*. Com esta implementação, obteve-se um esforço de controle mais suave. Já no *FLC* verificou-se que, analogamente ao *VSC* não apresenta mais o problema do *chattering* constituindo-se na principal vantagem deste controlador.

Durante a realização da parte experimental, notou-se a existência de erros em regime e de dificuldades em reduzir a magnitude desses erros apenas variando os ganhos utilizados. Em vista disso e dado o problema de atrito existente nos manipuladores, o caminho natural foi a incorporação da

compensação do atrito nas leis de controle conforme é mostrado a seguir.

6.5 Identificação Experimental dos Parâmetros de *LuGre*

Como foi mencionado no capítulo 3, a identificação experimental dos parâmetros de *LuGre* é feita em duas etapas, uma para a determinação dos coeficientes estáticos e outra para os dinâmicos.

6.5.1 Identificação dos parâmetros estáticos

Assumindo deslocamentos a velocidade constante ($\ddot{q}_i = 0$) e a compensação exata da gravidade no robô, a equação (6.9) pode ser reescrita como:

$$f_i = \tau_i; \quad i = 1, \dots, n \quad (6.20)$$

Da mesma forma, a equação do atrito de *LuGre* para velocidade constante é:

$$f_i = (\alpha_{0_i} + \alpha_{1_i} e^{-\left(\frac{\dot{q}_i}{v_{0_i}}\right)^2}) \operatorname{sgn}(\dot{q}_i) + \alpha_{2_i} \dot{q}_i; \quad i = 1, \dots, n \quad (6.21)$$

Igualando a equação (6.20) com a equação (6.21) para cada uma das juntas, obtém-se:

$$f_i = \tau_i = (\alpha_{0_i} + \alpha_{1_i} e^{-\left(\frac{\dot{q}_i}{v_{0_i}}\right)^2}) \operatorname{sgn}(\dot{q}_i) + \alpha_{2_i} \dot{q}_i; \quad i = 1, \dots, n \quad (6.22)$$

Uma vez obtida a equação (6.22) foi feito um conjunto de experimentos em malha fechada para cada uma das juntas do robô. O controlador empregado foi o *TDOF*. Os perfis de velocidade de valores de 0.01 [rad/s] e 0.02 [rad/s] junto com os torques de controle obtidos para a primeira junta são ilustrados nas figuras 6.24 e 6.25.

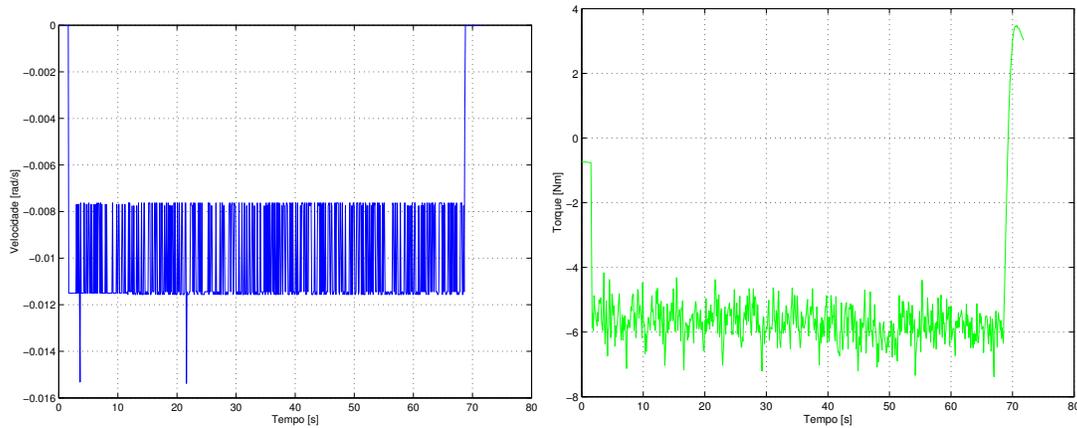


Figura 6.24: Perfil de velocidade e torque para a junta 0 (perfil médio de -0.01 [rad/s])

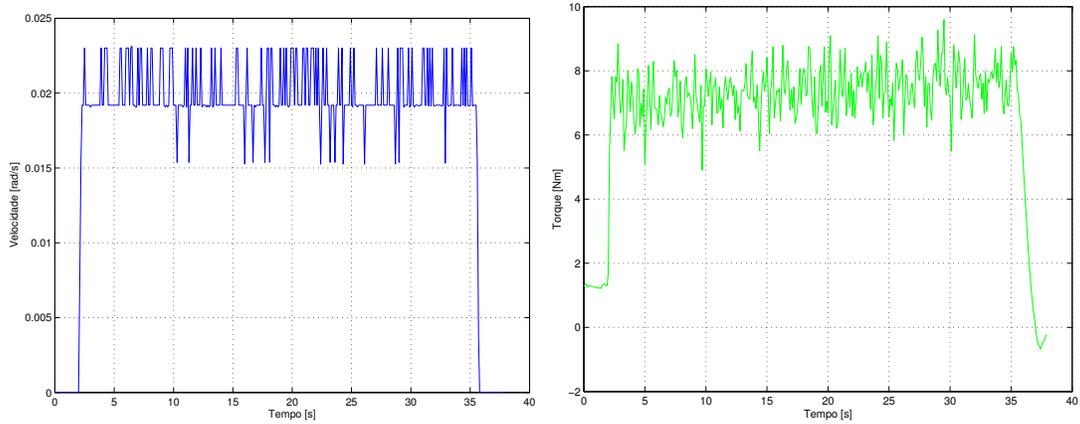


Figura 6.25: Perfil de velocidade e torque para a junta 0 (perfil médio de 0.02 [rad/s])

Da mesma forma os perfis de velocidade de valores de 0.03 [rad/s] e 0.1 [rad/s] junto com os esforços de controle correspondentes são ilustrados nas figuras 6.26 e 6.27.

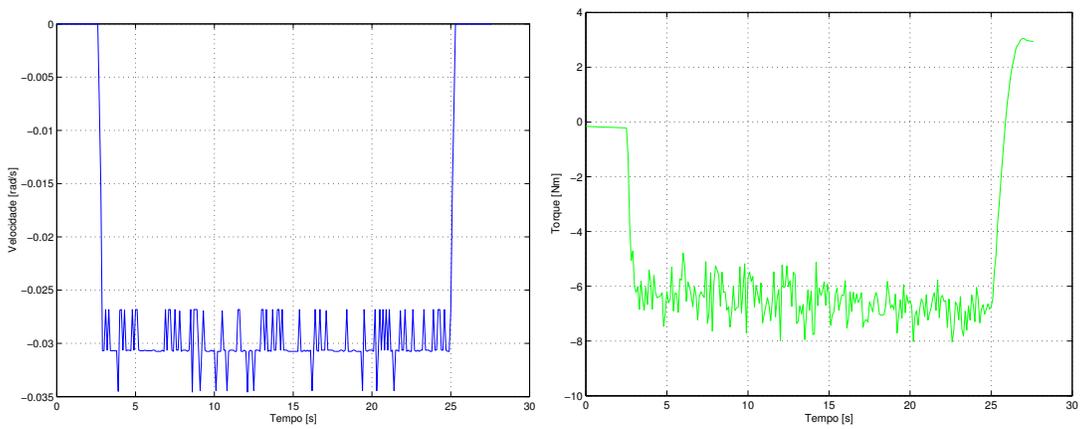


Figura 6.26: Perfil de velocidade e torque para a junta 0 (perfil médio de 0.03 [rad/s])

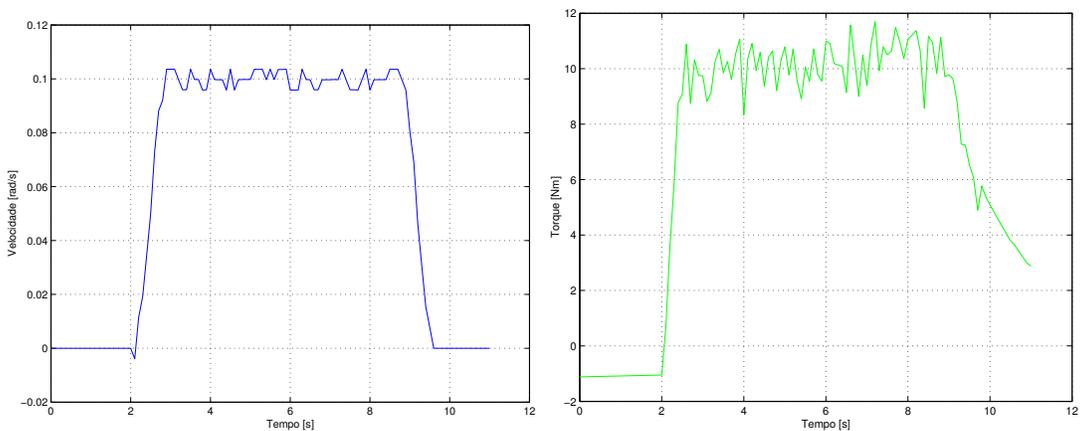


Figura 6.27: Perfil de velocidade e torque para a junta 0 (perfil médio de 0.1 [rad/s])

Para a construção das curvas de atrito estático, foram utilizados os valores médios de cada um dos perfis de velocidade e torque lidos. Desta forma, as curvas de atrito estático correspondentes às duas primeiras juntas são ilustradas na figura 6.28. É importante observar que estas curvas têm o mesmo perfil obtido em (Gandhi *et al.*, 2002), onde a transmissão é feita utilizando Harmonic Drives.

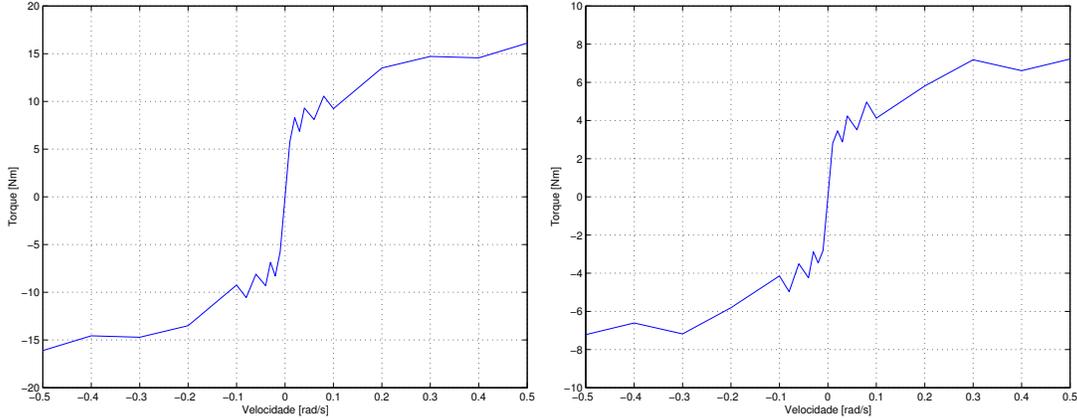


Figura 6.28: Atrito estático para as juntas 0 e 1

Finalmente, os coeficientes estáticos de atrito foram obtidos empregando a rotina de otimização *lsqcurvefit* do *MATLAB*[®], onde a função a minimizar é:

$$\min_X \sum_{i=1}^N (F(X, \dot{q}) - \tau)^2 \quad (6.23)$$

tal que $X = \alpha_{0_i}, \alpha_{1_i}, \alpha_{2_i}, v_{0_i}$ e N corresponde ao número de amostras dos dados experimentais obtidos à velocidade constante.

6.5.2 Identificação dos parâmetros dinâmicos

A identificação dos parâmetros dinâmicos σ_{0_i} e σ_{1_i} foi feita empregando os dados obtidos para a menor velocidade em cada uma das juntas de forma a considerar a variação da posição e do torque entre a velocidade zero e a subsequente amostra diferente de zero. Com estes dados, o parâmetro σ_{0_i} foi aproximado usando a proposta de P. Lischinsky e Morel (1997):

$$\sigma_{0_i} \simeq \frac{\Delta \tau_i}{\Delta q_i} \quad (6.24)$$

onde $\Delta \tau_i = \tau_i(k) - \tau_i(k-1)$ e $\Delta q_i = q_i(k) - q_i(k-1)$.

Para determinar o coeficiente σ_{1_i} é considerado o período de *Stiction* nas equações de atrito tal que $\dot{q}_i \simeq 0$ e $z_i \simeq 0$; então, para um sistema criticamente amortecido $\xi = 1$, este parâmetro é dado por (P. Lischinsky e Morel, 1997):

$$\sigma_{1_i} = 2\sqrt{\sigma_{0_i} - \bar{m}_{ii}} - \alpha_{2_i} \quad (6.25)$$

O cálculo dos parâmetros σ_{0_i} e σ_{1_i} através das equações (6.24) e (6.25) permite o uso do modelo de *LuGre* para a compensação de atrito. Na figura 6.29, observa-se o início do deslocamento e o torque para a junta 0.

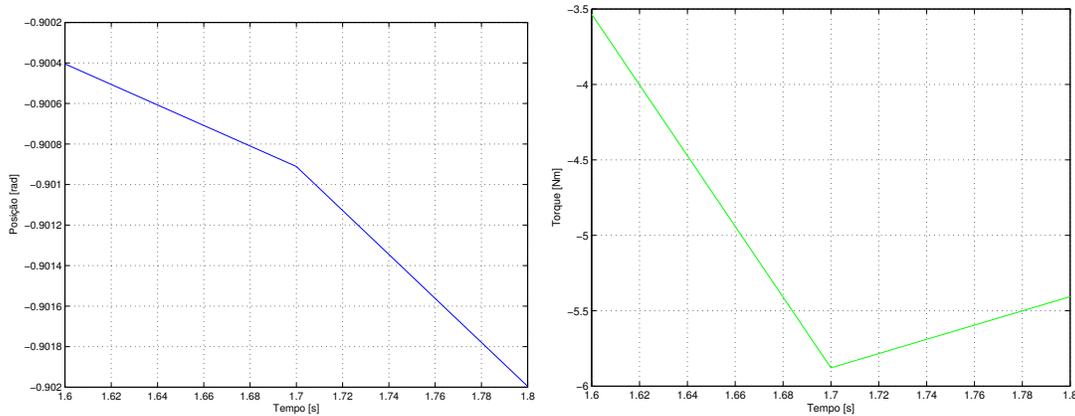


Figura 6.29: Início de movimento para a junta 0

Neste ponto, observou-se que, se os valores iniciais dos parâmetros dinâmicos obtidos são elevados, é necessário diminuir-los até obter um valor que não apresente problemas de vibração no robô.

Efetuosos os experimentos para as quatro juntas do robô, os coeficientes apresentados na tabela 6.11 foram determinados.

Tabela 6.11: Coeficientes de atrito

Junta	α_0	α_1	ν_0	α_2	σ_0	σ_1	κ_i
0	3.0525	3.0693	0.03	3.1827	20.53	15.18	0.001
1	2.44	1.28	0.03	1.88	1.13	1.4429	0.001
2	56.16	60.41	0.18	110.06	11.1	13.99	0.001
3	0.69	0.24	0.08	1.0277	1.09	1.19	0.001

É importante mencionar que existem outras formas de identificar os parâmetros dinâmicos, algumas destas formas podem ser encontradas em (Olsson *et al.*, 1997) e (Ramirez, 2003).

6.6 Implementação Experimental

Da mesma forma que os três controladores apresentados, a compensação de atrito foi implementada a uma frequência de amostragem de 1000 [Hz]. A integração da equação do observador para a variável z foi feita utilizando a aproximação trapezoidal de forma a garantir a estabilidade na implementação. Os módulos modificados foram **RobotControl.Mod** e **ScaraRobot.Mod**. Os novos procedimentos criados são:

- PROCEDURE (robot: Robot) InitFriction*(): BOOLEAN: Este procedimento inicializa os parâmetros do modelo de *LuGre* e desativa a compensação de atrito;

- PROCEDURE (robot: Robot) SetLuGreFrictParam*(VAR joint: LONGINT; sigma0, sigma1, alpha0, alpha1, alpha2, stribeck, kObs: LONGREAL): Este procedimento inicializa os valores dos parâmetros no modelo de *LuGre*;
- PROCEDURE(robot:Robot)EnableLuGre*(): Este procedimento ativa a compensação de atrito, uma vez que os parâmetros sejam inicializados;
- PROCEDURE(robot:Robot)DisableLuGre*(): Após os experimentos ou durante eles, este procedimento desabilita a compensação de atrito;
- PROCEDURE(robot.ScaraRobot)LuGreComp*(VAR $q, \dot{q}, \tilde{\tau}_{Frict}$: ARRAY OF LONGREAL): Este procedimento lê a velocidade real, a velocidade desejada, importa os parâmetros de *LuGre* e calcula o torque observado de atrito para cada uma das juntas;
- Os perfis das trajetórias e as velocidades empregadas nos experimentos para as duas primeiras juntas são apresentados nas figuras 6.30 e 6.31;

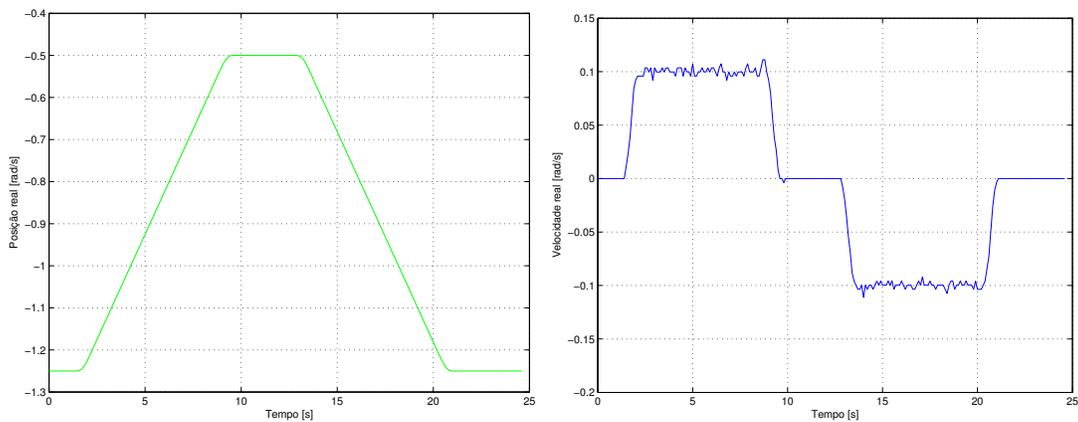


Figura 6.30: Posição e velocidade na junta 0

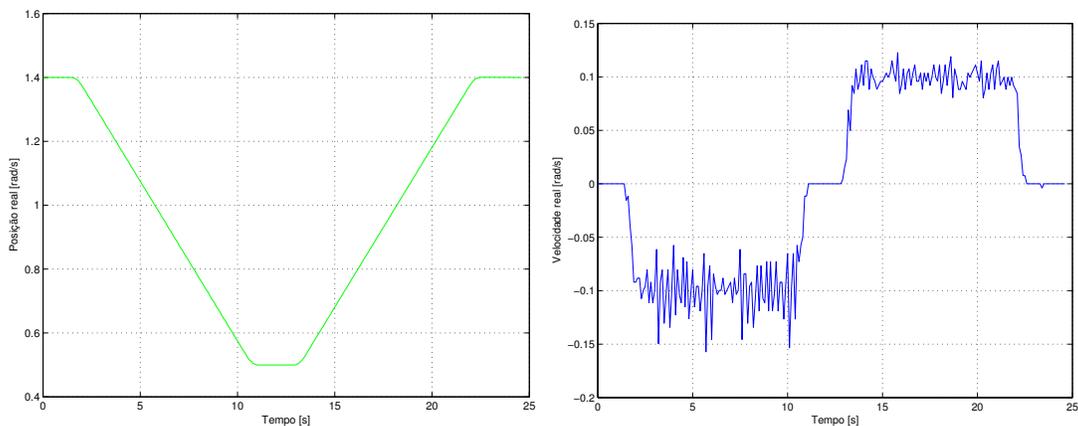


Figura 6.31: Posição e velocidade na junta 1

- Da mesma forma os perfis das trajetórias e velocidade para as duas últimas juntas são ilustrados nas figuras 6.32 e 6.33.

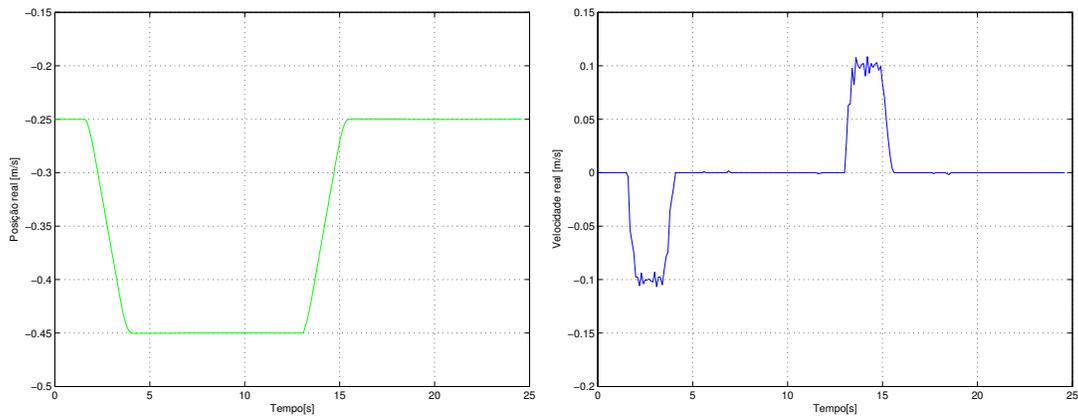


Figura 6.32: Posição e velocidade na junta 2

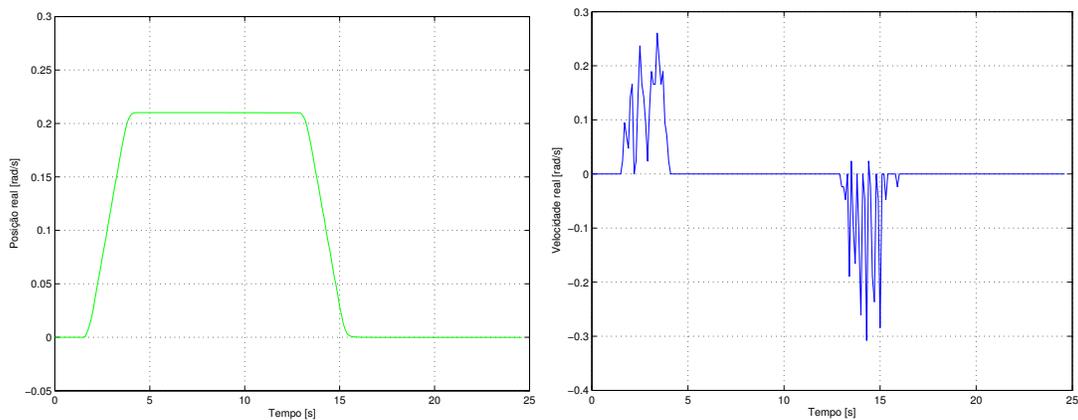


Figura 6.33: Posição e velocidade na junta 3

Na continuação são apresentados os resultados do controle de trajetória com compensação de atrito para os três controladores considerando baixas velocidades. Alguns destes resultados foram publicados em (Vargas *et al.*, 2004c), (Vargas *et al.*, 2004d) e (Vargas *et al.*, 2004e).

6.6.1 Resultados experimentais

Os resultados obtidos para as duas primeiras juntas dos experimentos feitos com e sem compensação de atrito com o controlador *TDOF* são ilustrados nas figuras 6.34 e 6.35. Nestes resultados observa-se a diminuição na magnitude do erro de seguimento quando é feita a compensação do atrito. Ao mesmo tempo verifica-se que a magnitude do torque de atrito estimado nas duas juntas é de ± 1 [Nm] e ± 0.3 [Nm] respectivamente.

Os resultados das duas últimas juntas são apresentados nas figuras 6.36 e 6.37. Nestes resultados

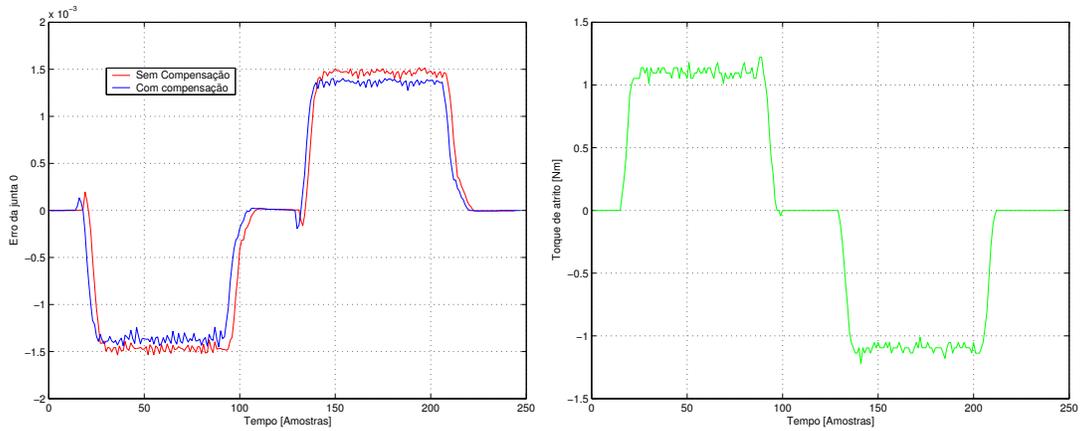


Figura 6.34: Erro e torque estimado de atrito na junta 0 - TDOF

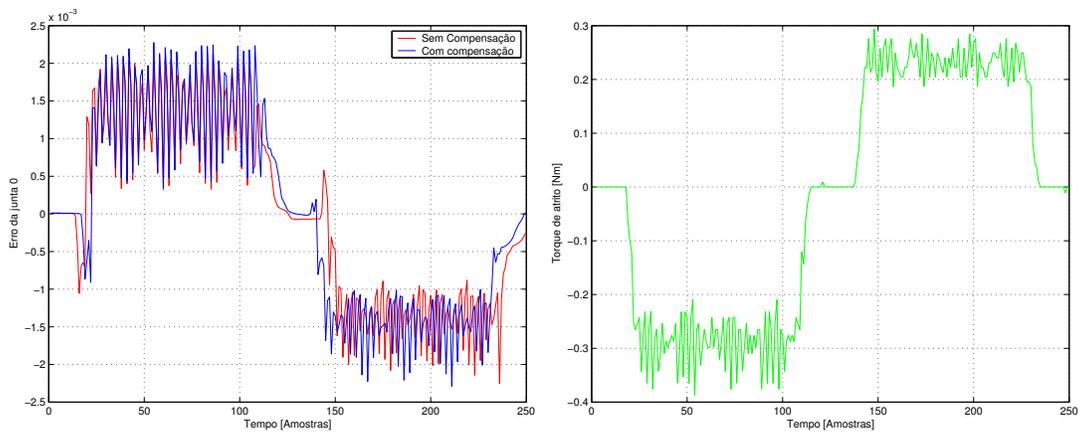


Figura 6.35: Erro e torque estimado de atrito na junta 1 - TDOF

verifica-se uma pequena diminuição dos erros de seguimento, entretanto os torques de atrito estimado são de $\pm 10[Nm]$ e $\pm 6 [Nm]$.

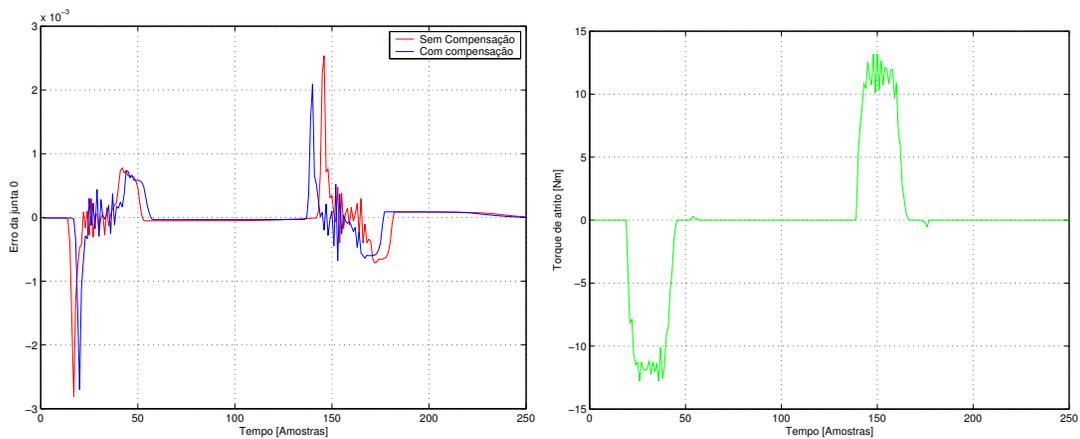


Figura 6.36: Erro e torque estimado de atrito na junta 2 - TDOF

Nas figuras 6.38 e 6.39 são apresentados os resultados experimentais obtidos com o controlador

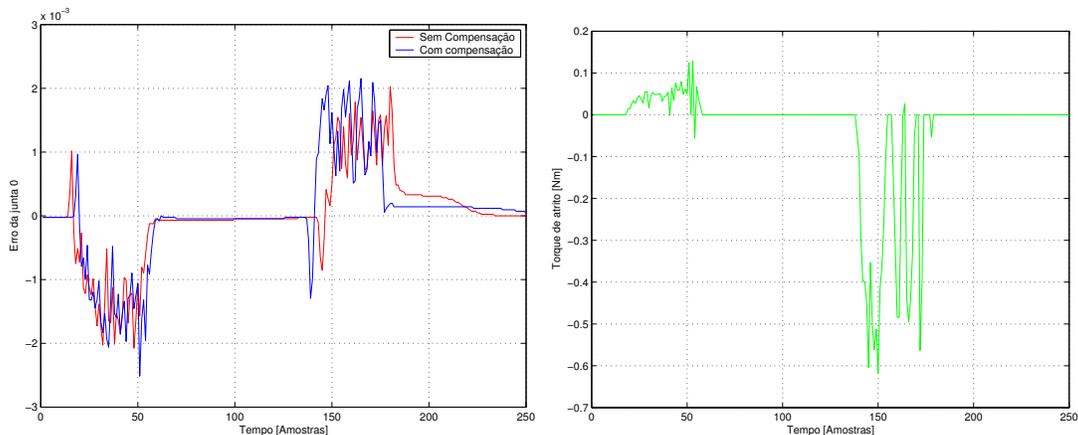


Figura 6.37: Erro e torque estimado de atrito na junta 3 - TDOF

VSC para as duas primeiras juntas do robô. Nestes resultados observa-se a diminuição do erro de seguimento quando feita a compensação do atrito. Logo os valores do torque estimado são similares aos obtidos com o controlador *TDOF*.

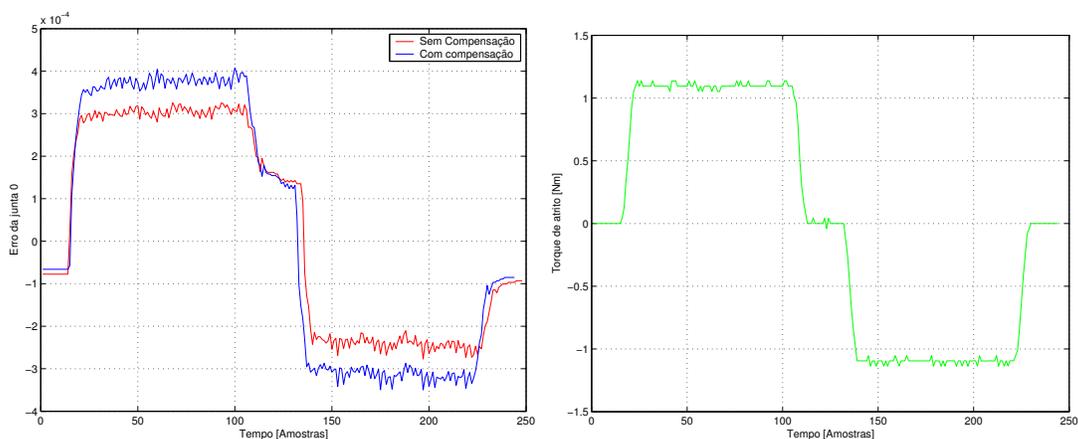


Figura 6.38: Erro e torque estimado de atrito nas junta 0 - VSC

Os resultados do controlador *VSC* para as duas últimas juntas são ilustrados nas figuras 6.40 e 6.41. Observa-se também uma pequena redução no erro de seguimento quando efetuada a compensação do atrito, já os valores do torque estimado são menores que no caso do controlador *TDOF*.

As figuras 6.42 e 6.43 apresentam os resultados obtidos para as duas primeiras juntas empregando o controlador *FLC*. Nestes resultados observa-se também um erro menor quando feita a compensação. Os torques de atrito estimado são similares às outras duas estratégias.

Finalmente os resultados para as duas últimas juntas empregando o controlador *FLC* são ilustrados nas figuras 6.44 e 6.45. Nestes resultados verifica-se também uma pequena diminuição do erro quando feita a compensação e uma diminuição no valor dos torques estimados quando comparados com as outras estratégias.

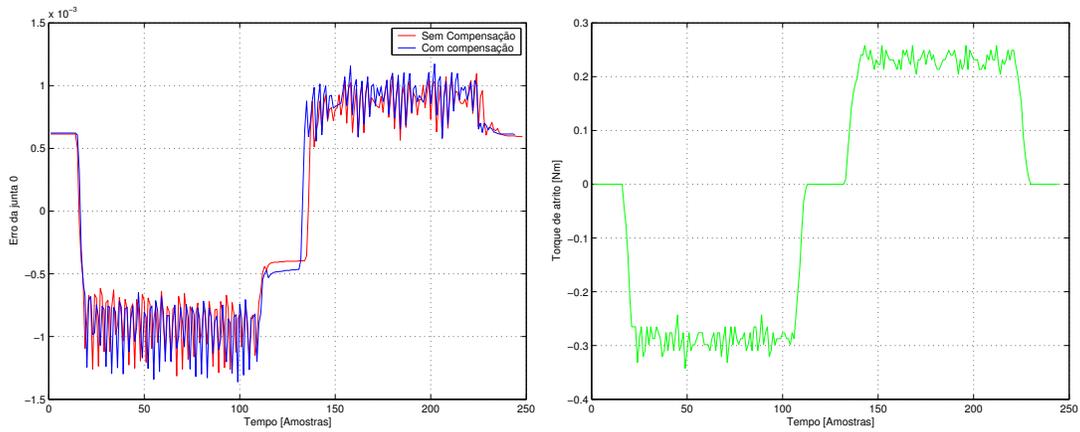


Figura 6.39: Erro e torque estimado de atrito nas junta 1 - VSC

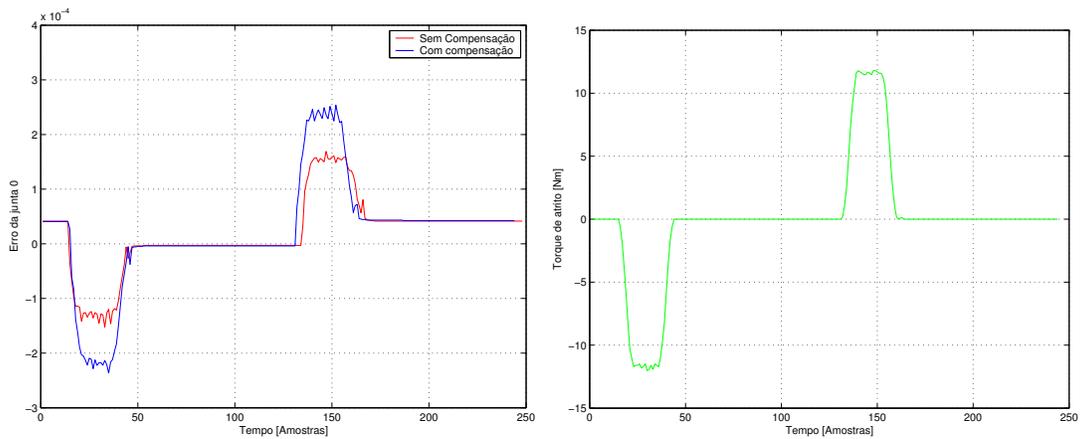


Figura 6.40: Erro e torque estimado de atrito na junta 2 - VSC

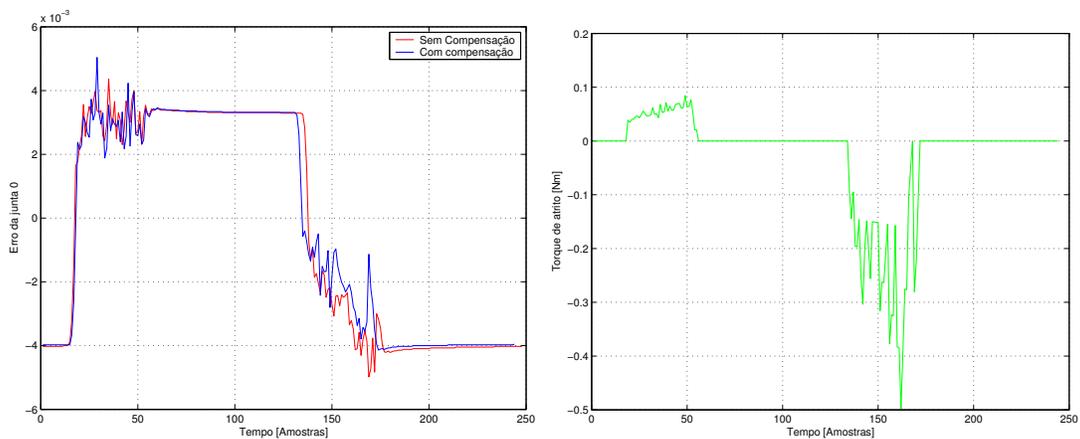


Figura 6.41: Erro e torque estimado de atrito na junta 3 - VSC

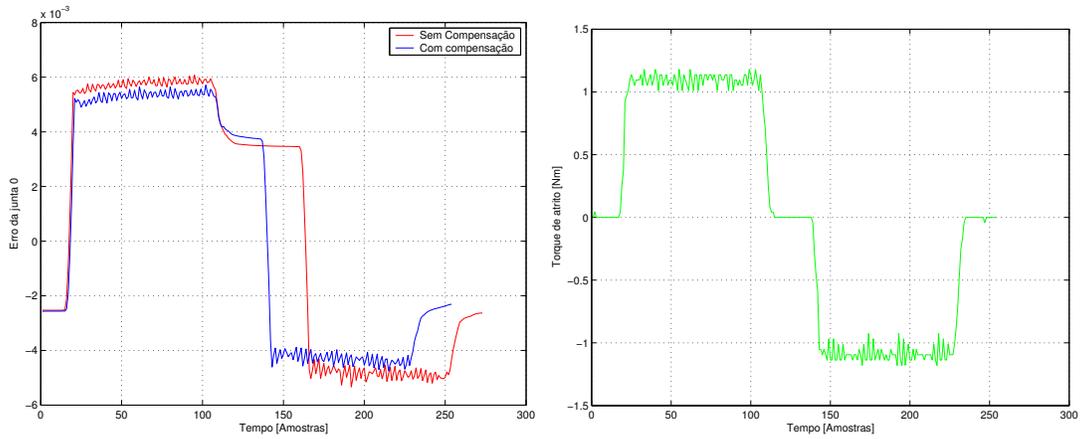


Figura 6.42: Erro e torque estimado de atrito na junta 0 - FLC

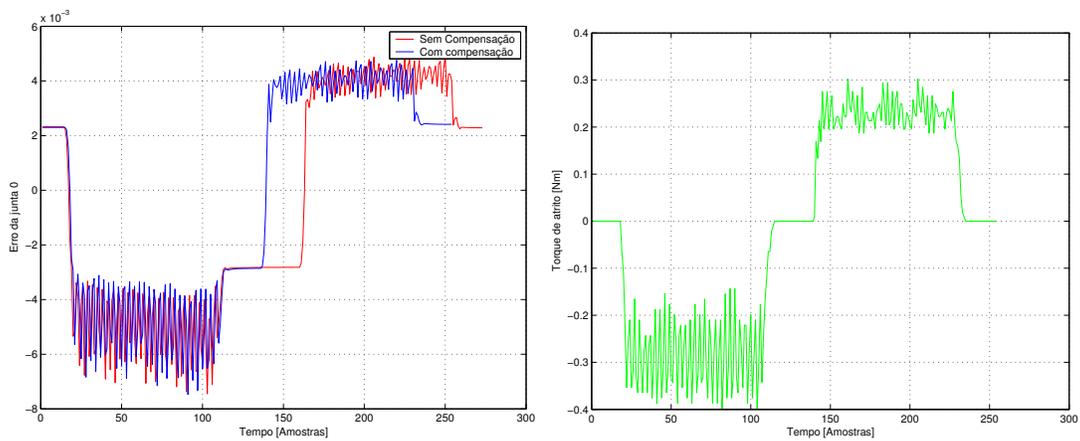


Figura 6.43: Erro e torque estimado de atrito nas junta 1 - FLC

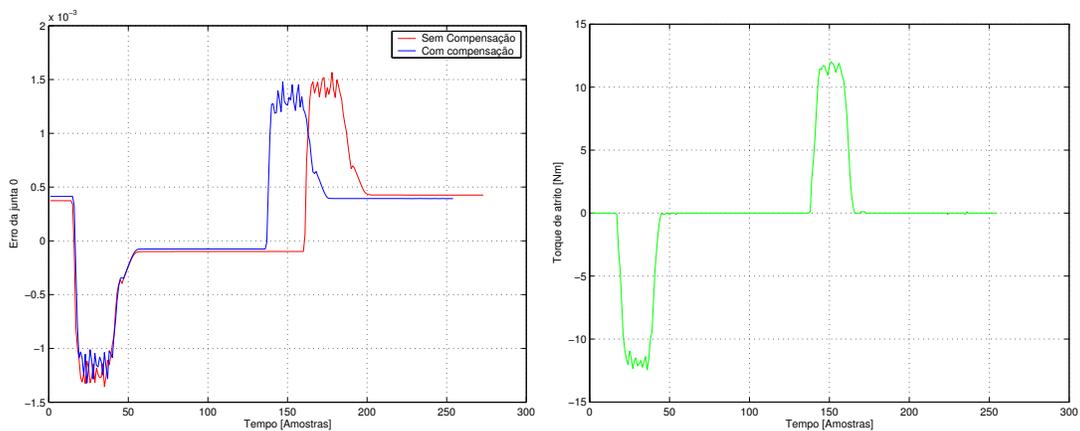


Figura 6.44: Erro e torque estimado de atrito na juntas 2 - FLC

Novamente para propósitos de comparação, uma avaliação quantitativa para as três estratégias é feita a seguir.

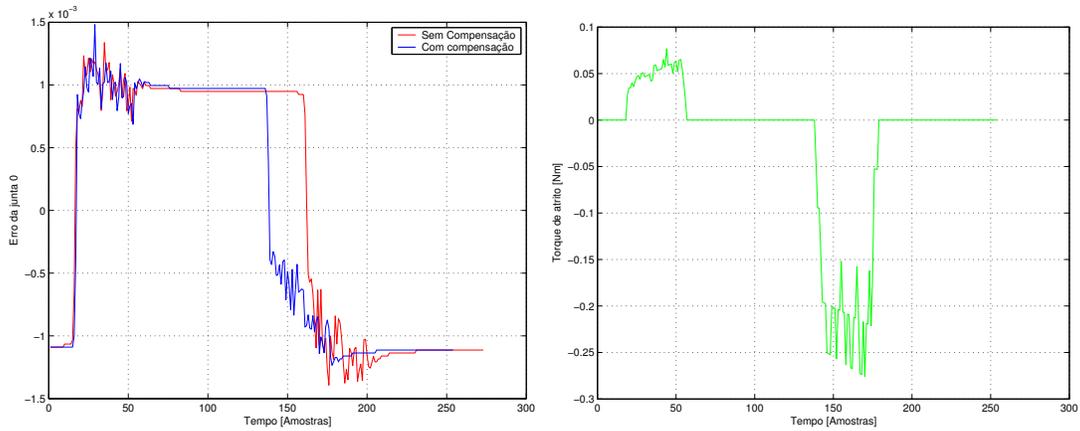


Figura 6.45: Erro e torque estimado de atrito na junta 3 - FLC

6.6.2 Análise dos resultados

Para fazer a análise quantitativa dos resultados experimentais obtidos com os três controladores com e sem compensação do atrito, foi empregado o seguinte índice:

$$|e_{JE}| = \sqrt{\sum_{j=1}^n (q_d(j) - q(j))^2} \quad (6.26)$$

Os índices obtidos para cada uma das técnicas são apresentados nas tabelas 6.12, 6.13 e 6.14.

Tabela 6.12: Indicadores do TDOF

Junta	Com compensação	Sem compensação
0	2.6333E-4	4.2817E-4
1	0.0015	0.0025
2	6.1049E-4	5.7703E-4
3	0.002	0.0021

Tabela 6.13: Indicadores do VSC

Junta	Com compensação	Sem compensação
0	0.0062	0.0067
1	0.0071	0.0069
2	0.0046	0.0046
3	0.0593	0.0844

Tabela 6.14: Indicadores do FLC

Junta	Com compensação	Sem compensação
0	0.1023	0.1801
1	0.0699	0.1611
2	0.0331	0.0351
3	0.0148	0.0144

Dos resultados obtidos nas tabelas 6.12, 6.13 e 6.14 observa-se uma diminuição do erro de segui-

mento nas duas primeiras juntas quando efetuada a compensação do atrito, verifica-se também que para junta de translação a compensação do atrito é bastante pequena, o que ser atribuído a que é necessária uma melhor aproximação dos parâmetros dinâmicos ou que o atrito nesta junta é muito reduzido, as mesmas características observadas na terceira junta podem ser verificadas na última junta.

6.7 Conclusões do Capítulo

Neste capítulo, foram apresentados os resultados experimentais dos controladores de posição e/ou trajetória dos controladores *TDOF*, *VSC* e *FLC* apresentados no capítulo 4. A bancada utilizada corresponde ao robô de configuração *SCARA* do laboratório de robótica da Universidade Federal de Santa Catarina.

Nos experimentos, utilizaram-se trajetórias semi-trapezoidais de terceira ordem. Os três controladores foram implementados de forma de obter um sistema criticamente amortecido e erro nulo em regime. No projeto do controlador *TDOF* foi feita a alocação ótima de pólos empregando a técnica *LQR*. Neste procedimento, foram feitas algumas modificações a fim de obter um sistema criticamente amortecido.

O controle *VSC* implementado foi o controle proposto em Amaral (2000). A principal mudança introduzida nesta técnica corresponde ao uso da função *tanh* na camada limite, com o objetivo de obter uma resposta mais suave e compará-la com o *FLC*.

O *FLC* implementado corresponde a uma técnica análoga à *VSC*, de forma a eliminar o *chattering*. As mudanças introduzidas correspondem ao uso de funções de pertinência triangulares na saída e a variação das escalas nestas funções.

Para a implementação experimental do atrito foi empregado o modelo de *LuGre* e um observador de velocidade. Durante a identificação dos parâmetros dinâmicos observou-se que valores elevados destes parâmetros podem excitar as dinâmicas não modeladas do robô resultando, entre outros efeitos, em vibrações nas juntas do robô e oscilações no efetuador final.

O índice do erro absoluto foi empregado para fazer a avaliação das três técnicas, verificando-se um desempenho similar entre elas. Este índice apresenta uma sensível melhora quando é incorporada nas leis de controle a compensação de atrito.

Capítulo 7

Controle de Força-Posição: Resultados Experimentais

Neste capítulo, são apresentados e detalhados os projetos e a implementação do controle de força e posição para os controladores *VSC* e *PID* apresentados no capítulo 4 e no apêndice D, respectivamente. A especificação de uma tarefa de corte é feita considerando a modelagem de De Luca e Manes para as restrições cinemáticas apresentadas no capítulo 2. Os controladores foram implementados no robô *SCARA* de configuração industrial detalhado no capítulo 5.

Alguns resultados obtidos considerando o controle de força e posição foram publicados em (Vargas *et al.*, 2001b), (Vargas *et al.*, 2001a) e (Vargas *et al.*, 2004a).

7.1 Apresentação do Problema de Força

As primeiras publicações considerando o problema do controle de força, empregando a modelagem de De Luca e Manes (1994) foram apresentadas por Manes em 1993 (Manes, 1993a), (Manes, 1993b), considerando inicialmente somente aspectos de simulação para controladores tipo *PID*. Posteriormente, Amaral (2000) propôs controladores robustos do tipo estrutura variável para compensar incertezas no modelo e perturbações externas, também considerando somente aspectos de simulação. Posteriormente Li *et al.* (2000) empregaram novamente a modelagem de De Luca e Manes e propuseram também controladores *PID* em nível de simulação. Em Vargas *et al.* (2001b), o problema de controle de força foi realizado usando a estrutura de dois graus de liberdade e os resultados obtidos de simulação. Recentemente, seguindo a linha de pesquisa experimental, os trabalhos de Ramirez (2003) e Passold (2003) foram desenvolvidos com resultados em controle de posição para robôs apresentando flexibilidades nas juntas e para o controle de força e posição de robôs rígidos usando técnicas de redes neurais, respectivamente. Neste capítulo, serão apresentados os resultados do controle de força para um robô rígido do tipo *SCARA*, usando controladores *PID* e *VSC* (Vargas *et al.*, 2004a).

7.2 Definição dos Experimentos

Os experimentos desenvolvidos para a obtenção das medições de força e de seu controle propriamente dito partem da definição de uma tarefa de corte, na qual o efetuador final percorre uma distância sobre os eixos x ou y (sendo que uma das duas direções permanece fixa), enquanto uma força no eixo z é aplicada. Pode-se considerar também o momento na junta de orientação θ do robô ou mantê-la fixa. A tarefa em questão é esquematizada na figura 7.1:

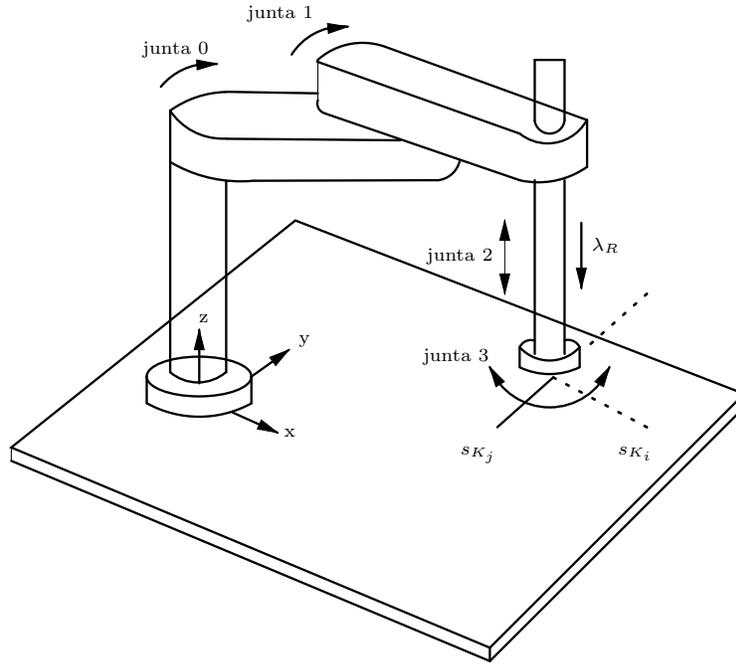


Figura 7.1: Tarefa de corte

Utilizando a modelagem de De Luca e Manes (1994), a posição do efetuador final é expressa como:

$$r = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_o + s_{K_i} \\ y_o + s_{K_j} \\ z_o \end{bmatrix} = \Gamma(s_K) \quad (7.1)$$

Derivando a equação (7.1), a velocidade é dada por:

$$v = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{s}_{K_i} \\ \dot{s}_{K_j} \end{bmatrix} \quad (7.2)$$

A partir da equação (7.2), é definido o vetor Υ_K . A força de reação, definida pela equação (2.13) para restrições geométricas como a tarefa definida, pode ser parametrizada como $F_R = Y_R \lambda_R$, onde a matriz Y_R deve ser arbitrária, satisfazendo a condição de ortogonalidade, isto é, $\Upsilon_K^T Y_R = 0$, isto é:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} Y_{R11} \\ Y_{R21} \\ Y_{R31} \end{bmatrix} = 0 \quad (7.3)$$

Para que a igualdade (7.3) seja cumprida, o valor de Y_R pode ser igual a:

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (7.4)$$

Desta forma a força de reação é determinada por:

$$F_R = Y_R \lambda_R = \begin{bmatrix} 0 \\ 0 \\ \lambda_R \end{bmatrix} \quad (7.5)$$

7.2.1 Obtenção das variáveis $\dot{s}_{K,m}$, $s_{K,m}$ e $\lambda_{R,m}$

Nas equações (7.1)-(7.5) pode-se observar que para a implementação do modelo de De Luca e Manes (1994), é necessária a obtenção das variáveis λ_R , $s_{K_i,j}$ e $\dot{s}_{K_i,j}$ em termos das variáveis lidas pelos sensores do robô (força, posição e velocidade). Para este fim em (Manes, 1992), (Manes, 1993a) e (Manes, 1993b) foram propostas as seguintes equações:

$$\dot{s}_{K,m} = \Upsilon_K^\dagger v_m \Rightarrow \Upsilon_K^\dagger = (\Upsilon_K^T W_\alpha \Upsilon_K)^{-1} \Upsilon_K^T W_\alpha \quad (7.6)$$

$$\lambda_{R,m} = Y_R^\dagger F_m \Rightarrow Y_R^\dagger = (Y_R^T W_\alpha^{-1} Y_R)^{-1} Y_R^T W_\alpha^{-1} \quad (7.7)$$

onde o termo m refere-se à variável medida e W_α é dada por:

$$W_\alpha = \begin{bmatrix} I_3 & 0_3 \\ 0_3 & \alpha^2 I_3 \end{bmatrix} \quad (7.8)$$

A equação (7.8) foi proposta de forma a evitar os problemas observados no controle híbrido clássico (Raibert e Craig, 1981) publicados por Duffy (1990). Maiores detalhes podem ser obtidos em (Manes, 1992).

Na tarefa definida, considerando os três primeiros graus de liberdade, as equações (7.6) e (7.7) podem ser expressas como:

$$\Upsilon_K^\dagger = \left[\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \right]^{-1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7.9)$$

$$Y_R^\dagger = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7.10)$$

Finalmente é obtido:

$$\begin{bmatrix} \dot{s}_{K_{i,m}} \\ \dot{s}_{K_{j,m}} \end{bmatrix} = \begin{bmatrix} \dot{x}_m \\ \dot{y}_m \end{bmatrix} \quad (7.11)$$

$$\begin{bmatrix} \lambda_{R,m} \end{bmatrix} = \begin{bmatrix} f_{z,m} \end{bmatrix} \quad (7.12)$$

Das equações (7.11) e (7.12), observa-se que os valores medidos para os parâmetros $\dot{s}_{K_{i,m}}$, $\dot{s}_{K_{j,m}}$ e λ_R para a tarefa definida correspondem aos valores cartesianos de velocidade nas direções de x e y e a força medida na direção z . Para obter os valores medidos de $s_{K_{i,m}}$ e $s_{K_{j,m}}$ assume-se um erro pequeno de forma que estes valores sejam iguais à integral de suas velocidades (Manes, 1992) ou seja, valores próximos às posições cartesianas em x e y . Também é necessário lembrar que no robô as posições obtidas através dos *encoders* correspondem às posições no espaço de juntas. Para obter as posições e velocidades cartesianas, é necessário o cálculo da cinemática direta do robô (Hüppi e Gruener, 2001).

7.3 O Problema da Força de Contato

Para a análise do problema da força de contato, dois tipos de superfícies são utilizadas:

1. Superfície de madeira;
2. Superfície de borracha;

Partindo da tarefa definida na seção anterior, somente a aplicação de uma força no eixo z é considerada, permitindo ao mesmo tempo o movimento desta junta até estabelecer contato com a superfície considerada. As outras direções permanecem fixas. Utilizando o controle pela dinâmica inversa a seguinte equação é obtida:

$$\tau = (JM^{-1})^\# \left[Q \left(\begin{bmatrix} \lambda_{R,des} \\ \ddot{s}_{K_{i,des}} \\ \ddot{s}_{K_{j,des}} \end{bmatrix} - d \right) - m \right] \quad (7.13)$$

onde:

$$Q = \begin{bmatrix} JM^{-1}J^TY_R & \Upsilon_K \end{bmatrix} \quad (7.14)$$

$$m = -\dot{\Upsilon}_K \dot{s}_K + \dot{J}\dot{q} - JM^{-1}n \quad (7.15)$$

$$d = \begin{bmatrix} d_R \\ d_{K_i} \\ d_{K_j} \end{bmatrix} \quad (7.16)$$

Sabendo-se que a posição do robô não é uma configuração singular, a seguinte igualdade pode ser utilizada $(JM^{-1})^\# = MJ^{-1}$. Ao mesmo tempo os torques de controle nas juntas 0, 1 e 3 são considerados nulos e a força de atrito na junta 2 é considerada inicialmente zero também.

Executando operações e considerando as simplificações, a seguinte equação é obtida:

$$\tau = J^T \lambda_{R,des} \quad (7.17)$$

Considerando a igualdade (7.12), o controle implementado é:

$$\lambda_{R,des} = \lambda_{R,d} - K_p \left(e + \frac{1}{T_i} \int_0^t e \, dt + T_d \dot{e} \right) \quad (7.18)$$

onde o erro e é dado pela diferença entre o valor desejado de força $\lambda_{R,d}$ e o valor medido ou real $\lambda_{R,m}$.

A equação (7.18) corresponde ao controle *PI* apresentado no apêndice D (esta técnica é utilizada por diferentes autores a fim de anular o erro em regime (Wilfinger *et al.*, 1994), (Yoshikawa, 2000)).

7.3.1 Resultados experimentais

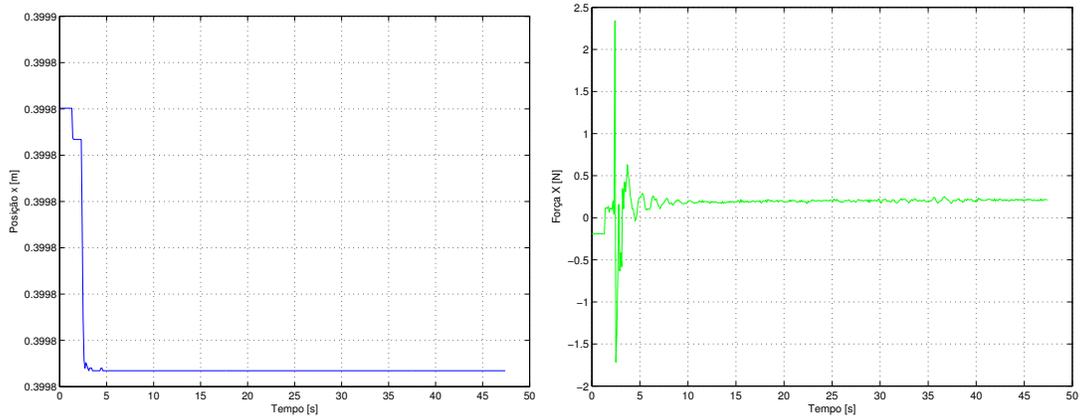
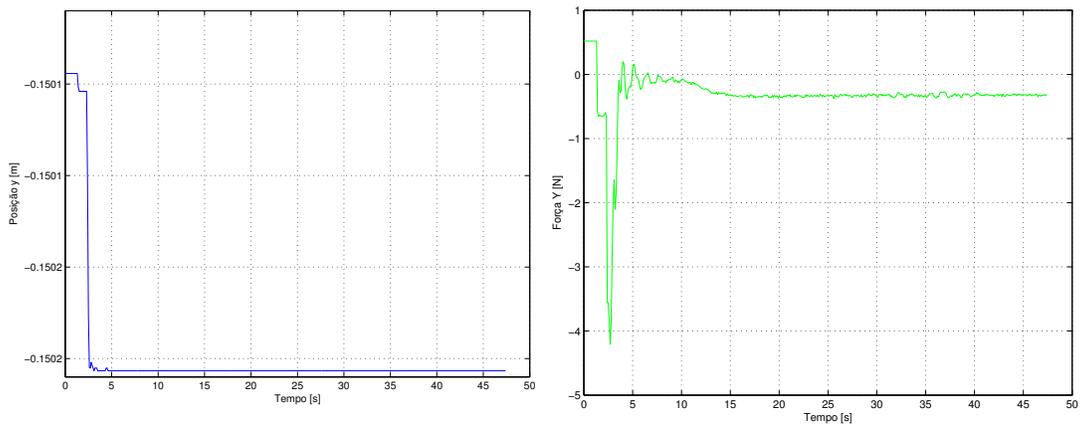
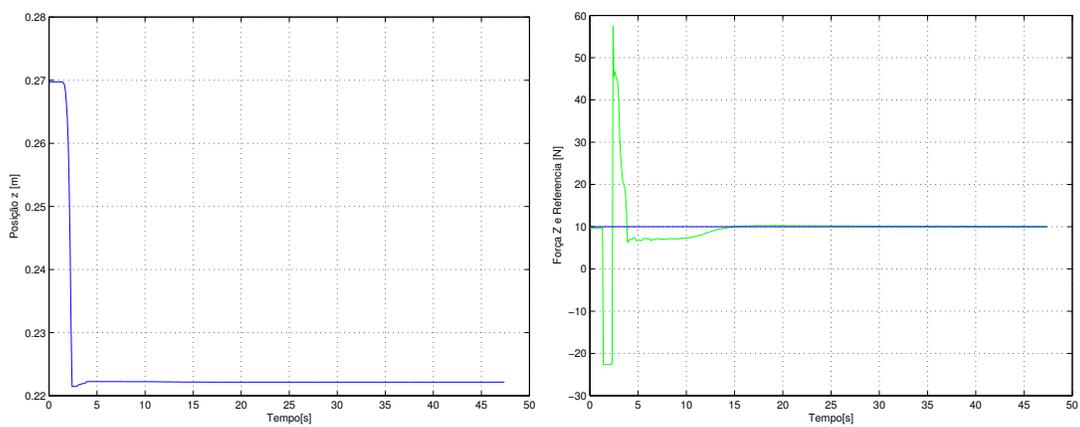
O controlador dado pela equação (7.18) foi implementado com uma frequência de amostragem de 1000 [Hz] (Alguns resultados experimentais para o controle de posição são encontrados em (Vargas *et al.*, 2004e)). Os ganhos deste controlador para as três posições cartesianas e a orientação do robô são apresentados na tabela 7.1. A força desejada no eixo z corresponde ao valor de 10 [N]

Tabela 7.1: Ganhos dos controladores *PI*

<i>Eixo</i>	K_p	T_i	T_d
x	0.0	1.0	0.0
y	0.0	1.0	0.0
z	1.0	0.2	0.0
θ	0.0	1.0	0.0

Nas figuras 7.2 e 7.3, são apresentados os resultados experimentais obtidos para a força de impacto utilizando a superfície de madeira.

Destes resultados, observa-se que as posições nas direções x e y permanecem constantes antes e após o impacto. Já as forças nestas direções apresentam uma variação no momento do impacto. Os

Figura 7.2: Posição e força no eixo x - madeiraFigura 7.3: Posição e força no eixo y - madeiraFigura 7.4: Posição e força no eixo z - madeira

resultados para as direções z e θ são apresentados nas figuras 7.4 e 7.5.

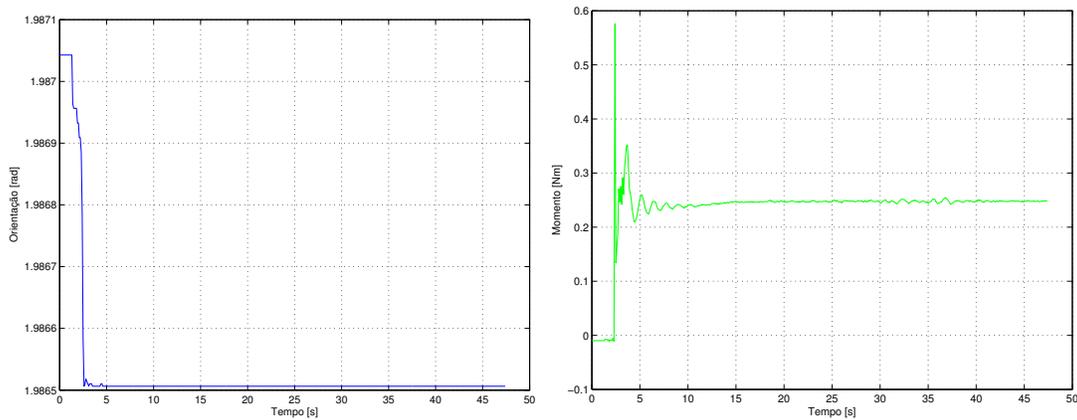


Figura 7.5: Posição e força no eixo θ - madeira

Na figura 7.4 observa-se a variação da posição de $0.27 [cm]$ a $0.23 [cm]$ na direção z . Nesta mesma direção, no momento do impacto, a força medida alcança $60 [N]$. Pouco depois do impacto, a força desejada de $10 [N]$ é alcançada.

As figuras 7.6 e 7.7 ilustram os resultados obtidos para a superfície de borracha nas direções x e y . Da mesma forma que no caso da superfície de madeira, observa-se que estas direções permanecem constantes e as forças têm uma variação no momento do impacto.

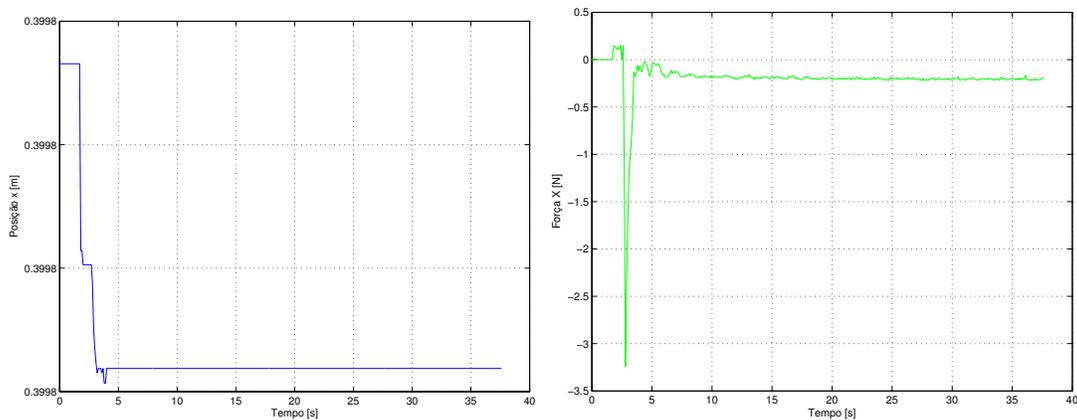
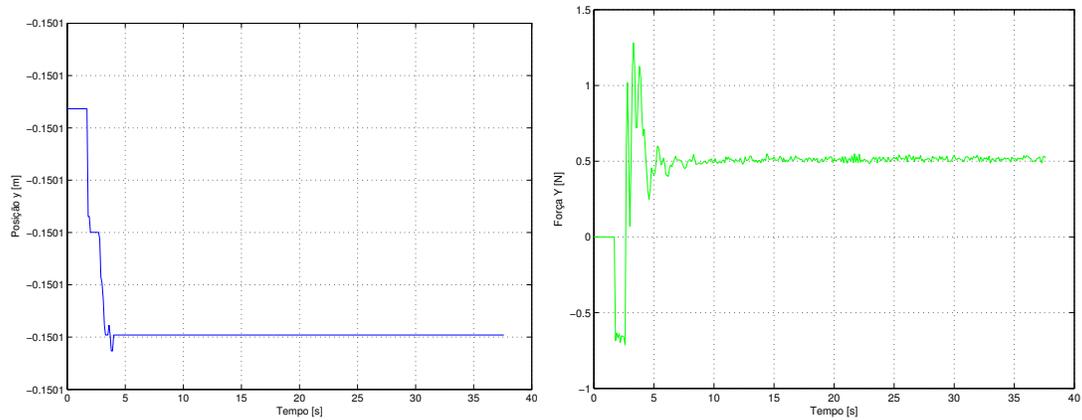
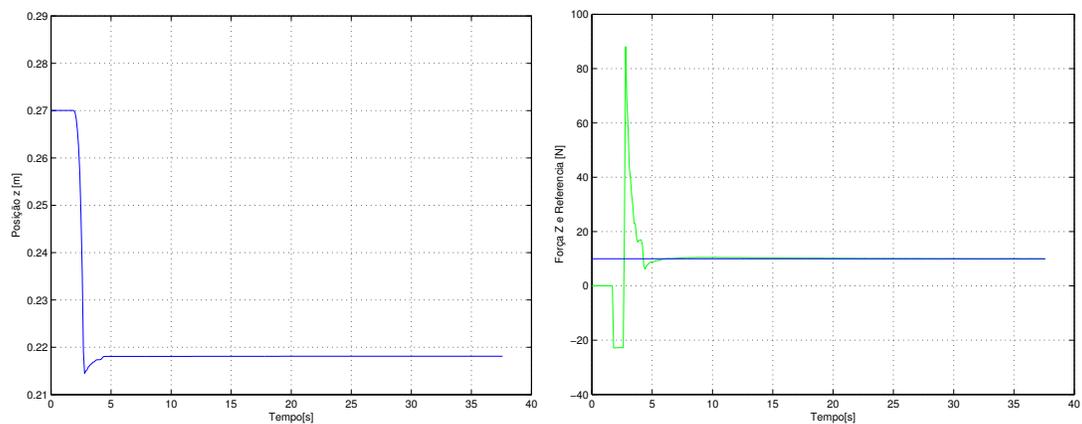
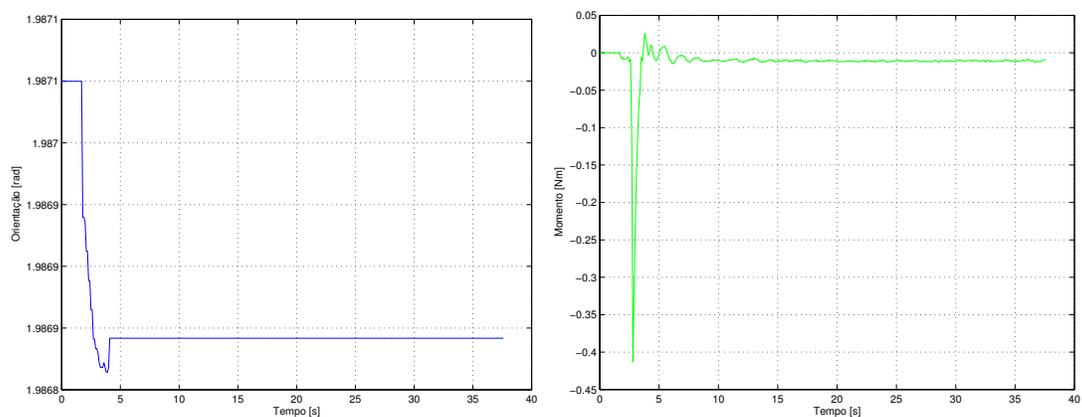


Figura 7.6: Posição e força no eixo x - borracha

As figuras 7.8 e 7.9 apresentam os resultados para as direções z e θ . Novamente é observada a variação na posição na direção z . Neste caso como a superfície tem um valor menor de rigidez, no momento do impacto, a força medida alcança $80 [N]$. Finalmente a força desejada de $10 [N]$ é alcançada após o impacto.

Estes resultados permitem mostrar que valores elevados nas forças medidas podem levar o sistema à instabilidade e danificar o robô e o meio no qual está sendo realizada a tarefa. Para evitar este problema, é melhor iniciar os experimentos com o contato já estabelecido.

Figura 7.7: Posição e força no eixo y - borrachaFigura 7.8: Posição e força no eixo z - borrachaFigura 7.9: Posição e força no eixo θ - borracha

7.4 Implementação dos Controladores Desenvolvidos

Para o controle de força, inicialmente foi considerada a estratégia PI implementada para a análise da força de impacto. A partir desta estratégia, é implementado o controle VSC para tarefas submetidas

a restrições cinemáticas proposto por Amaral (2000). Novamente, a lei da dinâmica inversa é dada por:

$$\tau = (JM^{-1})^\# \left[Q \left(\begin{bmatrix} \lambda_{R,des} \\ \ddot{s}_{K_i,des} \\ \ddot{s}_{K_j,des} \end{bmatrix} - d \right) - m \right] \quad (7.19)$$

onde

$$\begin{aligned} \lambda_{R,des} &= \lambda_{R,d} - K_p \left((\lambda_R - \lambda_{R,d}) + \frac{1}{T_i} \int_0^t (\lambda_R - \lambda_{R,d}) dt \right) \\ \ddot{s}_{K_i,des} &= \ddot{s}_{K_i,d} - K_{v_i} (\dot{s}_{K_i,m} - \dot{s}_{K_i,d}) - K_{p_i} (s_{K_i,m} - s_{K_i,d}) \\ \ddot{s}_{K_j,des} &= \ddot{s}_{K_j,d} - K_{v_j} (\dot{s}_{K_j,m} - \dot{s}_{K_j,d}) - K_{p_j} (s_{K_j,m} - s_{K_j,d}) \end{aligned} \quad (7.20)$$

Para a realização dos experimentos, assume-se que o robô não se encontra numa configuração singular e que o contato com a superfície já tenha sido estabelecido.

7.4.1 Controle a estrutura variável

Para a implementação experimental desta estratégia, baseados no trabalho de Amaral (2000), para a tarefa definida, foram definidos três grupos de superfícies de chaveamento, uma para a força exercida no eixo z e as outras para o movimento nas direções x e y . Estas superfícies são:

$$\sigma = \begin{bmatrix} \sigma_r \\ \sigma_{s_{K_i}} \\ \sigma_{s_{K_j}} \end{bmatrix} = \begin{bmatrix} \frac{1}{T_i} \int_0^t (\lambda_R - \lambda_{R,d}) dt \\ (\dot{s}_{K_i,m} - \dot{s}_{K_i,d}) - c_{k_i} (s_{K_i,m} - s_{K_i,d}) \\ (\dot{s}_{K_j,m} - \dot{s}_{K_j,d}) - c_{k_j} (s_{K_j,m} - s_{K_j,d}) \end{bmatrix} \quad (7.21)$$

juntamente com suas derivadas em relação ao tempo:

$$\frac{\partial \sigma}{\partial t} = \begin{bmatrix} -\lambda_{R,d} \\ -\dot{s}_{K_i,d} + c_{k_i} (\dot{s}_{K_i,m} - \dot{s}_{K_i,d}) \\ -\dot{s}_{K_j,d} + c_{k_j} (\dot{s}_{K_j,m} - \dot{s}_{K_j,d}) \end{bmatrix} \quad (7.22)$$

Considerado a lei de controle definida pela equação (4.59) chega-se à seguinte lei de controle:

$$\tau = -MJ^{-1}Q \left(Q^{-1}m + \frac{\partial \sigma}{\partial t} + \varpi \operatorname{sgn}(\sigma) + \varrho \sigma \right) \quad (7.23)$$

substituindo a função sgn pela função tangente hiperbólica \tanh para obter um chaveamento mais suave, é obtido:

$$\tau = -MJ^{-1}Q \left(Q^{-1}m + \frac{\partial \sigma}{\partial t} + \varpi \tanh(\sigma) + \varrho \sigma \right) \quad (7.24)$$

Executando operações na equação (7.24) pode-se identificar claramente as componentes para o

controle de força e posição. Também pode ser verificado que a superfície de chaveamento de força é baseada numa componente integral (Wilfinger *et al.*, 1994).

7.5 O Problema da Filtragem nas Medições de Força

Nas medições feitas para o controle da força de impacto, verificou-se que o sinal de força é contaminado por ruídos. É, portanto, importante que as medições de força sejam filtradas antes de serem utilizadas no projeto das leis de controle. A não filtragem poderia implicar no fato de que os ruídos possam vir a ser amplificados de tal forma a comprometer o desempenho do sistema, interferindo na qualidade da resposta e, em alguns casos, na estabilidade. Pelas razões expostas antes da implementação dos controladores de força, foram implementados diferentes filtros e analisados seus efeitos no sinal da força medida.

7.5.1 Filtro exponencial de primeira ordem

Chamado também de filtro passa baixas, é empregado para atenuar ruídos de alta frequência. Seu funcionamento pode ser descrito através de sua equação diferencial ou de sua função de transferência como sendo:

$$\tau_f \frac{d\bar{x}(t)}{dt} + \bar{x}(t) = x(t) \quad (7.25)$$

onde $x(t)$ é o valor medido, $\bar{x}(t)$ é o valor filtrado e τ_f é a constante de tempo do filtro. Observa-se que o ganho estático deste filtro corresponde ao valor unitário ¹.

Para a implementação do filtro exponencial dado pela equação (7.25), é necessário que este seja digitalizado a uma determinada frequência de amostragem. Utilizando a aproximação *backward* (Aström e Wittenmark, 1997) na equação (7.25) é obtido:

$$\frac{d\bar{x}}{dt} \approx \frac{\bar{x}(k) - \bar{x}(k-1)}{T} \quad (7.26)$$

Considerando a aproximação (7.26), a equação (7.25) transforma-se em:

$$\tau_f \frac{\bar{x}(k) - \bar{x}(k-1)}{T} + \bar{x}(k) = x(k) \quad (7.27)$$

Re-arranjando a equação (7.27), obtém-se:

$$\bar{x}(k) = \left(\frac{\tau_f}{\tau_f + T} \right) \bar{x}(k-1) + \left(\frac{T}{\tau_f + T} \right) x(k) \quad (7.28)$$

Igualando

$$\left(\frac{\tau_f}{\tau_f + T} \right) = \alpha \Rightarrow (1 - \alpha) = \left(\frac{T}{\tau_f + T} \right) \quad (7.29)$$

¹Este filtro também é chamado de filtro *RC* pelo fato de que pode ser construído a partir de um circuito *RC*

Então:

$$\bar{x}(k) = \alpha\bar{x}(k-1) + (1-\alpha)x(k) \quad (7.30)$$

Segundo a equação (7.30), o valor filtrado é igual à soma ponderada do valor filtrado no instante anterior $\bar{x}(k-1)$ e do valor medido no tempo atual $x(k)$.

7.5.2 Filtro exponencial de segunda ordem

Este filtro corresponde a dois filtros de primeira ordem em série. Seu funcionamento pode ser descrito através da seguinte função de transferência:

$$\frac{\bar{x}}{x} = \left(\frac{1}{\tau_f s + 1} \right)^2 \quad (7.31)$$

onde seus termos são aqueles especificados no filtro de primeira ordem. A principal desvantagem deste filtro é que introduz um atraso considerável no sinal medido, ainda que ele incremente a banda de rejeição para as altas frequências.

7.5.3 Seleção da constante τ_f

A constante de tempo τ_f para os filtros de primeira e segunda ordem pode ser calculada fazendo uma analogia entre o filtro exponencial e o exponencial ponderado (Aström e Hägglund, 1995). O filtro exponencial ponderado é baseado numa determinada quantidade de amostras, tal que a constante α é dada por:

$$\alpha = \left(\frac{n}{n+1} \right) \quad (7.32)$$

onde n é o número de amostras consideradas para o cálculo do filtro. Igualando as equações (7.29) e (7.32) verifica-se a seguinte igualdade:

$$\tau_f = nT \quad (7.33)$$

onde T corresponde ao tempo de amostragem. Com as especificações dadas pelas equações (7.32) e (7.33), foram projetados três filtros: dois de primeira ordem e um de segunda. Ao primeiro filtro foi atribuído um valor de $\alpha = 0.55$, ao segundo filtro, o valor de $\alpha = 0.65$ e o terceiro filtro consiste de dois filtros de primeira ordem em série, onde o valor de α é igual ao do primeiro filtro. A frequência de amostragem utilizada é de 1000 [Hz].

A figura 7.10 apresenta as respostas dos três filtros projetados. Nesta figura, observa-se a redução da banda passante na medida em que a ordem do filtro ou a constante de tempo τ_f são incrementados. Em outras palavras, quanto maior o valor de $\alpha \in [0 \ 1]$ ou maior a ordem do filtro, uma melhor atenuação de altas frequências é obtida, mas ao mesmo tempo um atraso maior no sistema é

introduzido.

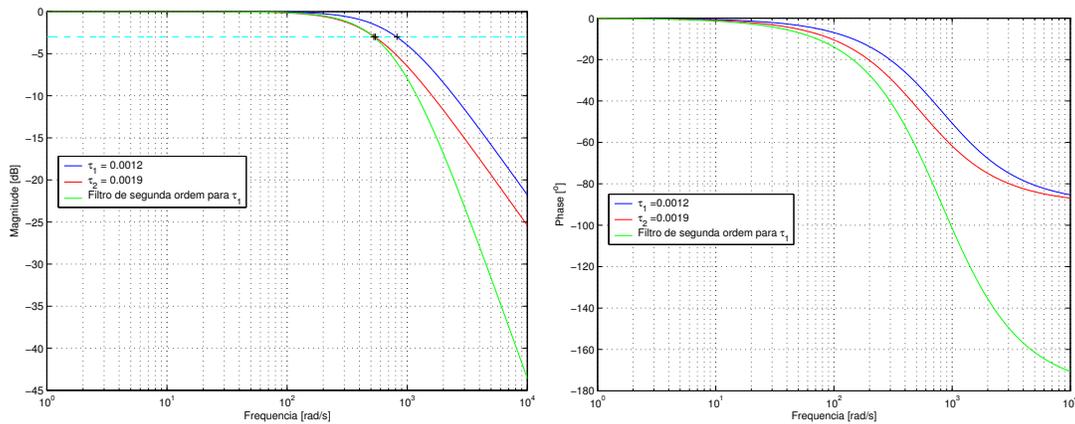


Figura 7.10: Diagramas de bode para os três filtros

7.6 Obtenção dos Resultados

Para a implementação experimental do controle de força e posição, foi criado o módulo *Filter.Mod* e modificados os módulos *Robot.Mod* e *ScaraRobot.Mod*. A implementação foi feita considerando uma frequência de amostragem de 1000 [Hz]. Estes módulos e procedimentos são descritos a seguir.

- PROCEDURE ForceFilter*(force: ARRAY OF LONGREAL; VAR forceFiltered : ARRAY OF LONGREAL): Este procedimento recebe os sinais do sensor e o valor da constante α do filtro, entrega os sinais filtrados para serem empregadas no algoritmo de controle.
- PROCEDURE (robot: ScaraRobot) ForceSensToTCP*(VAR forceSens, forceTCP: ARRAY OF LONGREAL): Este procedimento transforma as variáveis do sistema de referência do sensor para o da ferramenta;
- PROCEDURE (robot: ScaraRobot) ForceTCPToBase*(VAR q, forceTCP, forceBase: ARRAY OF LONGREAL): Este procedimento transforma os sinais lidos do sensor de força do sistema da ferramenta ao sistema cartesiano;
- PROCEDURE (robot: ScaraRobot) ForceToTau*(VAR q, dx, force, tau: ARRAY OF LONGREAL): Este procedimento transforma os sinais de torque no espaço cartesiano (operacional) para o espaço das juntas;
- PROCEDURE (robot: ScaraRobot) MatrizIner*(q, qd, inercia, lc, masa: ARRAY OF LONGREAL; VAR InerMatrix: ARRAY OF ARRAY OF LONGREAL): Este procedimento implementa a matriz de inércia;

- PROCEDURE (robot: ScaraRobot) Coriolis*(VAR q, qd, masa, lc, Cori: ARRAY OF LONGREAL): Este procedimento implementa a matriz de forças centrífugas e de Coriolis;
- PROCEDURE (h:ControlForçaHdl) Run: Neste procedimento é feita a filtragem das medições de força e momentos e a transformação de coordenadas a partir das coordenadas do sensor. É feita também a chamada das matrizes de forças centrífugas e de coriolis, inércia e os jacobianos transposto e inverso.

Além destes módulos implementados para o controle de força e para a filtragem, estão também definidos e implementados os módulos para a compensação do atrito desenvolvidos no capítulo 6, e dependendo do tipo e da execução de uma tarefa, podem ser utilizados se necessário.

7.6.1 Resultados experimentais da filtragem

As frequências de corte e os valores de α para os filtros projetados são apresentados na tabela 7.2.

Tabela 7.2: Parâmetros dos filtros exponenciais de primeira e segunda ordem

Filtro	Frequência de corte [Hz]	α
1	130.8383	0.55
2	86.4082	0.65
3	84.0510	0.55

A fim de não introduzir um atraso significativo no sinal medido nos experimentos a serem feitos, somente o primeiro filtro projetado foi implementado. Nas figuras 7.11 e 7.12 são apresentados os resultados dos dados lidos com e sem filtro.

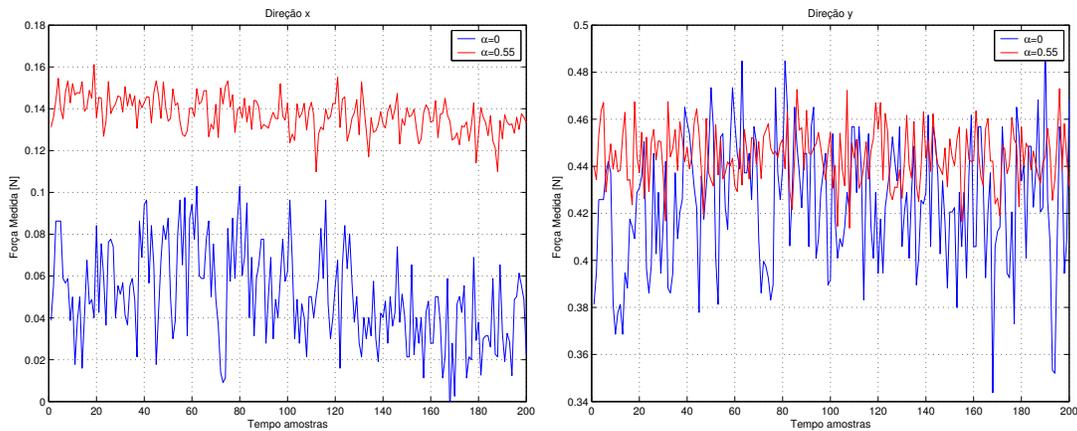


Figura 7.11: Forças medidas nas direções x e y

Destes resultados, observa-se que as medições nas três direções cartesianas e a orientação sem filtro apresentam sinais muito mais ruidosos do que no caso dos sinais com filtro. A magnitude na direção x sem filtro varia de 0.02 [N] a 0.1 [N]; já a sinal filtrada varia entre 0.12 [N] a 0.16 [N] pelo que

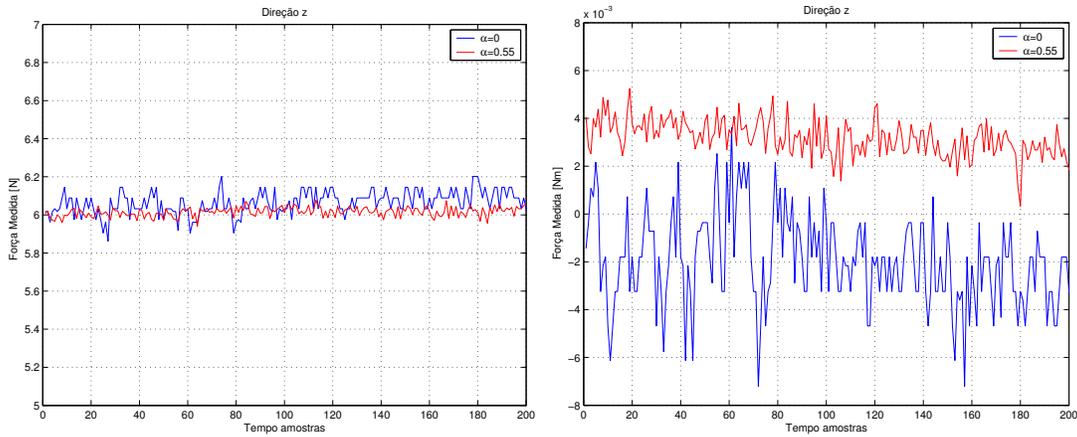


Figura 7.12: Força e momento em torno de z e θ

verifica-se um sinal mais limpo quando empregado o filtro. Esta mesma característica é verificada também nas outras direções.

7.6.2 Resultados experimentais usando controle PI

Considerando o contato estabelecido, e empregando a equação (7.20), somente os controladores de força são ativados. Neste caso, independentemente dos ganhos, os torques dos controladores de posição são nulos. A compensação de atrito, que é significativa nas duas primeiras juntas, não tem o efeito esperado. Os ganhos utilizados nesta estratégia são apresentados na tabela 7.3.

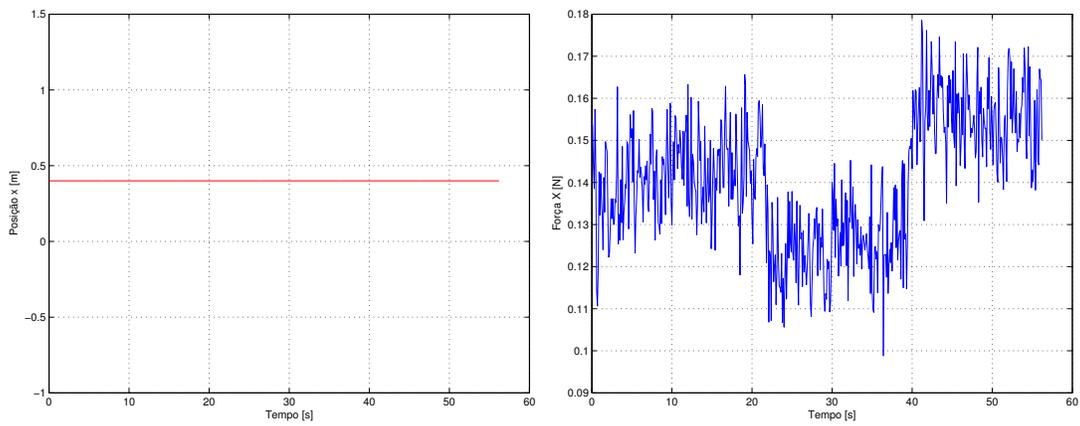
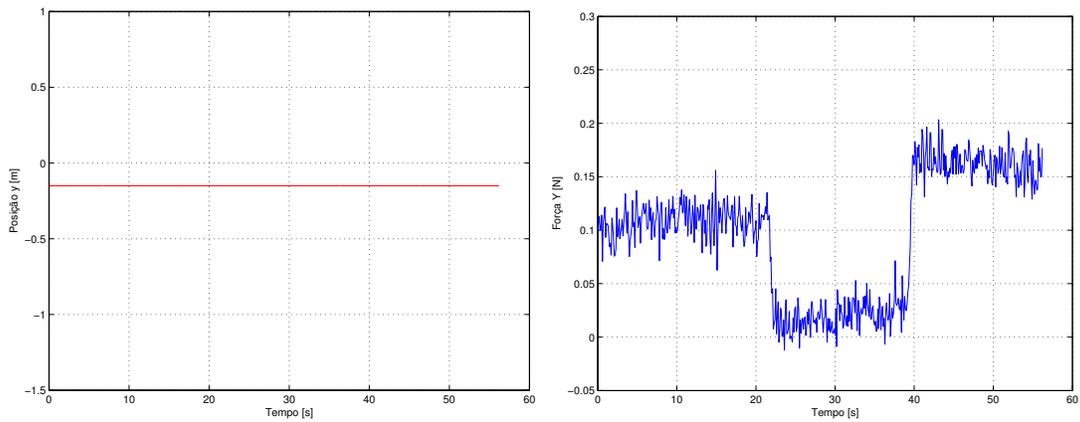
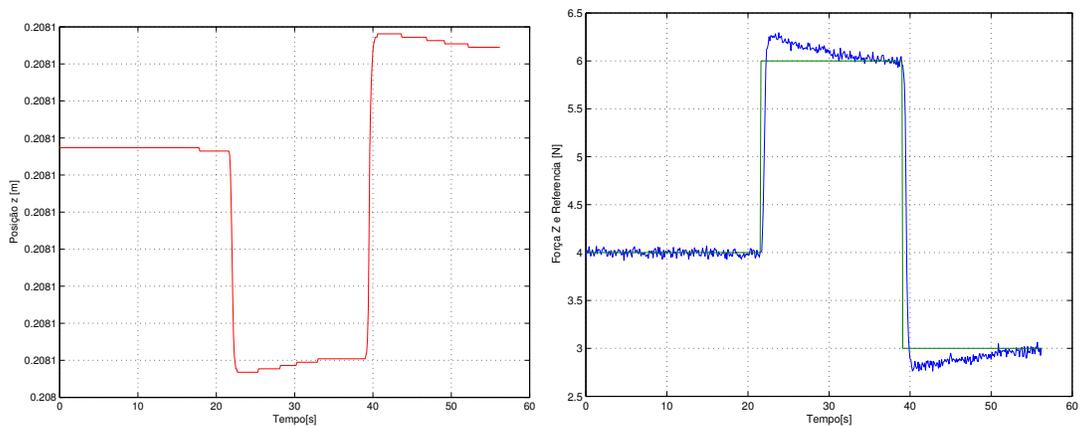
Tabela 7.3: Ganhos dos controladores PI

$Eixo$	K_p	T_i	T_d
x	0.0	1.0	0.0
y	0.0	1.0	0.0
z	0.65	0.025	0.0
θ	0.0	1.0	0.0

Estes ganhos foram determinados de forma a obter o menor sobre-sinal no seguimento da referência de força. Durante a determinação dos ganhos, constatou-se que valores elevados transmitem um valor muito alto de torque na terceira junta, ocasionando a saturação do atuador.

Os resultados experimentais obtidos para as duas primeiras direções cartesianas são ilustrados nas figuras 7.13 e 7.14. Destes resultados, observa-se que a posições x e y permanecem constantes; entretanto, os valores de força nestas direções têm uma pequena variação cujo perfil corresponde à força lida na direção z .

Já os resultados para a direção z e θ são apresentados nas figuras 7.15 e 7.16. Nestes resultados observa-se que a força medida na direção z consegue seguir os valores de referência estabelecidas apresentando uma pequena sobre-sinal na variação positiva e negativa da referência. Novamente as direções z e θ permanecem constantes.

Figura 7.13: Posição e força na direção x - controle PIFigura 7.14: Posição e força na direção y - controle PIFigura 7.15: Posição e força na z - controle PI

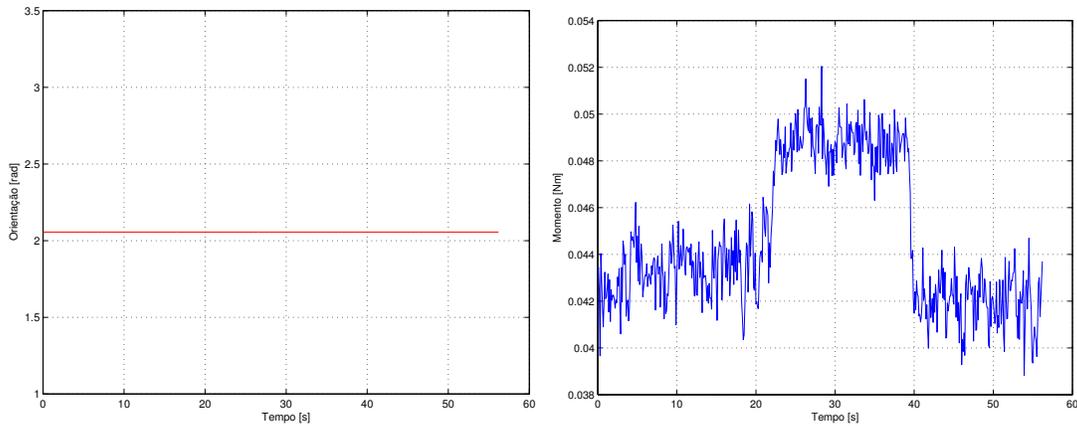


Figura 7.16: Posição angular e momento na direção θ - controle PI

7.6.3 Resultados experimentais usando controle *VSC*

A implementação do controle *PI* apresentou bons resultados visto que a literatura de controle de força indica a necessidade de uma ação integral para tratar o problema (Volpe e Khosla, 1993), (Amaral, 2000). Como no caso anterior, supõe-se que as duas primeiras juntas e a junta de orientação têm seus torques de controle nulos. Os ganhos do *VSC* obtidos a partir dos ganhos integrais são apresentados na tabela 7.4.

Tabela 7.4: Ganhos dos controladores *VSC*

<i>Eixo</i>	ϖ	T_i	ρ	ϕ
x	0.0	1.0	0.0	0.8
y	0.0	1.0	0.0	0.8
z	6.0	0.1	1.0	0.8
θ	0.0	1.0	0.0	0.8

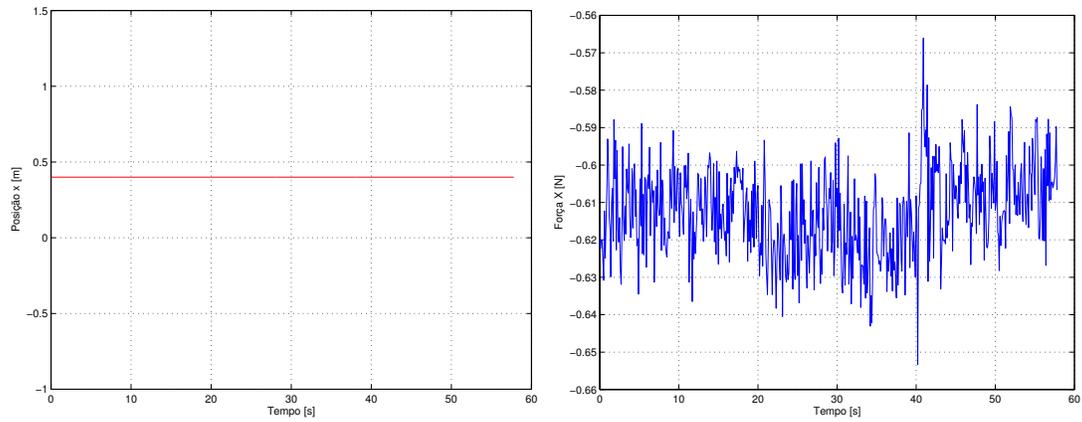
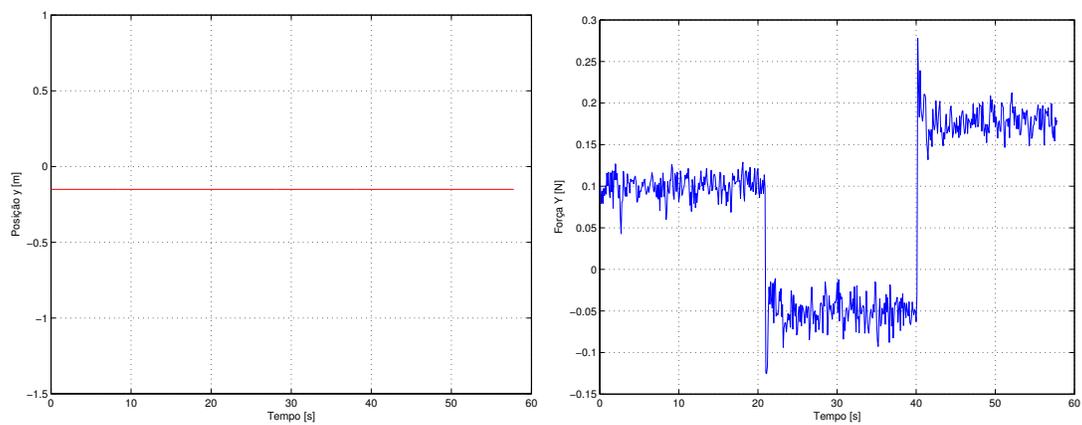
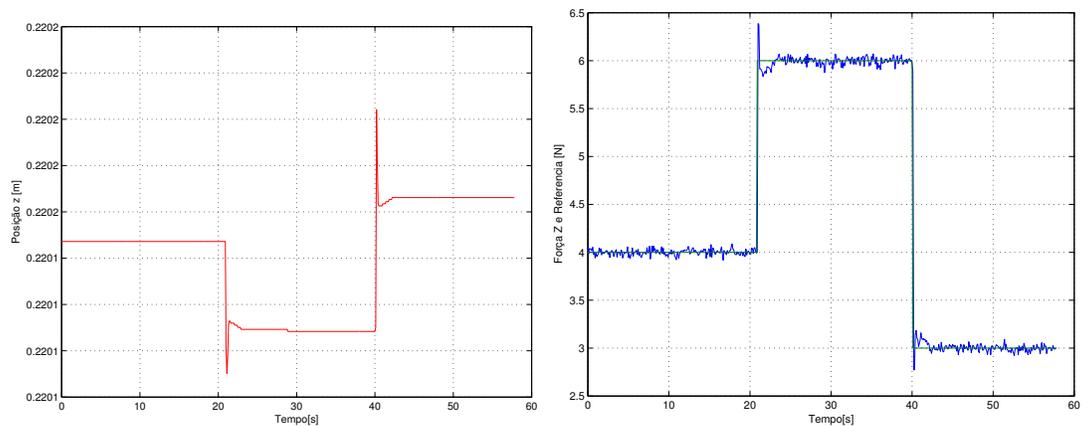
Os resultados experimentais obtidos para as duas primeiras direções são ilustrados nas figuras 7.17 e 7.18. Nestes figuras, observa-se o controle nulo nas direções x e y . A medição de força na direção x tem um perfil constante o que não acontece com a direção y que tem o mesmo perfil da força desejada na direção z .

Os resultados para as direções z e θ são ilustrados nas figuras 7.19 e 7.20. Destes resultados, observa-se que a força medida na direção z consegue seguir os valores de referência estabelecidos e, ao contrário do que acontece com o controlador *PI* nesta direção, não se tem a presença de sobre-sinal.

Por observação, verifica-se que o erro obtido com o controlador *VSC* na direção z é menor que no caso do controlador *PID*. Finalmente, a tabela 7.5 apresenta a média do somatório dos erros em Nm (equação (6.26)) das duas estratégias.

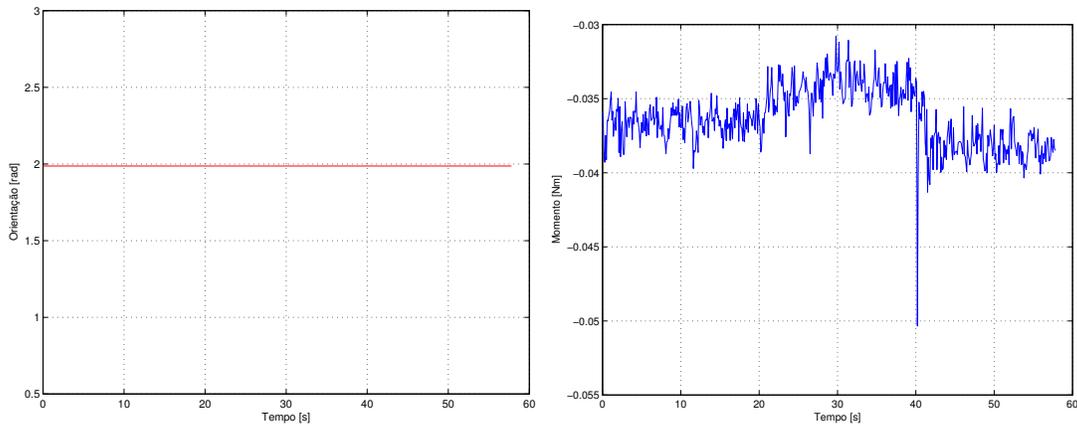
Tabela 7.5: Erros na direção z dos controladores *PID* e *VSC*

<i>Eixo</i>	<i>PI</i>	<i>VSC</i>
z	7.73	0.56

Figura 7.17: Posição e força na direção x - controle VSCFigura 7.18: Posição e força na direção y - controle VSCFigura 7.19: Posição e força na direção z - controle VSC

7.6.4 Resultados experimentais usando controle de força e posição

A seguir é feito o experimento para a tarefa de corte, sendo y a direção selecionada e a direção no eixo x permanece fixa. O valor de referência de força no eixo z corresponde a um valor constante e igual

Figura 7.20: Posição angular e momento em θ - controle VSC

a 5 [N], a velocidade para a execução desta tarefa foi de 0.01 [m/s]. Foi também feita a compensação do atrito conforme descrito nos experimentos de controle de posição. Os valores dos ganhos utilizados são apresentados na tabela 7.6. É importante observar que estes ganhos foram determinados baseados nos experimentos anteriores, de forma a obter o melhor seguimento das referências, tanto para a força quanto para a posição.

Tabela 7.6: Ganhos dos controladores VSC

<i>Eixo</i>	ϖ	T_i	ϱ	ϕ	c_k
x	8.0	0.0	5.0	0.8	2.0
y	10.0	0.0	7.0	0.8	3.0
z	6.0	0.1	2.0	0.8	0.0

Os resultados obtidos para as duas primeiras direções são ilustrados nas figuras 7.21 e 7.22. Destes resultados, observa-se que a direção x permanece fixa em 0.34 [m]. Já a direção y percorre uma distância entre 0.02 [m] e -0.12 [m].

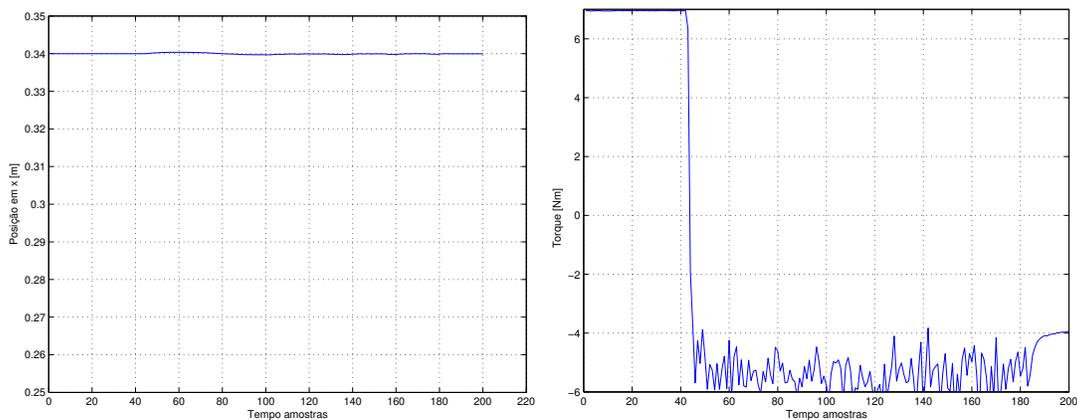


Figura 7.21: Posição e torque - Controle VSC

O resultado para o eixo z é apresentado na figura 7.23. Nesta direção observa-se que o valor medido

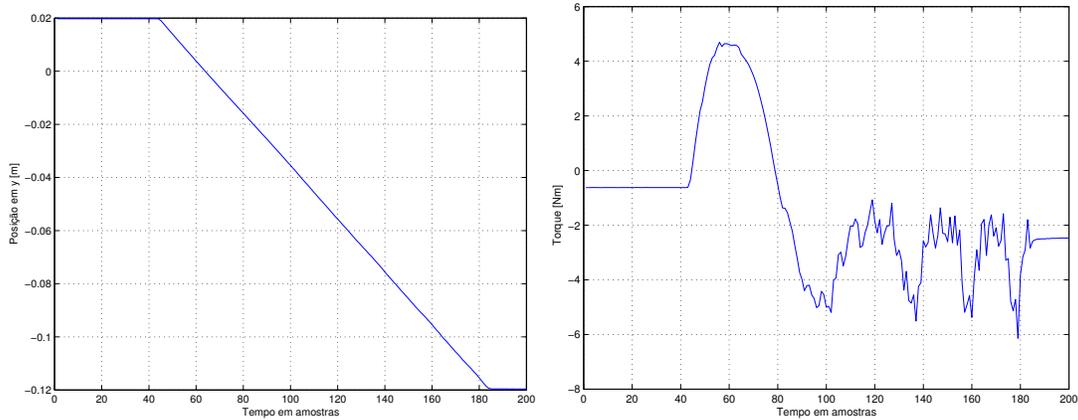


Figura 7.22: Posição e torque - Controle VSC

de força consegue seguir o valor de referência especificado. Os ganhos não foram mais incrementados por problemas de saturação nos atuadores, problemas de vibração, oscilações na direção z e, em alguns casos, perda de contato brusca na realização da tarefa.

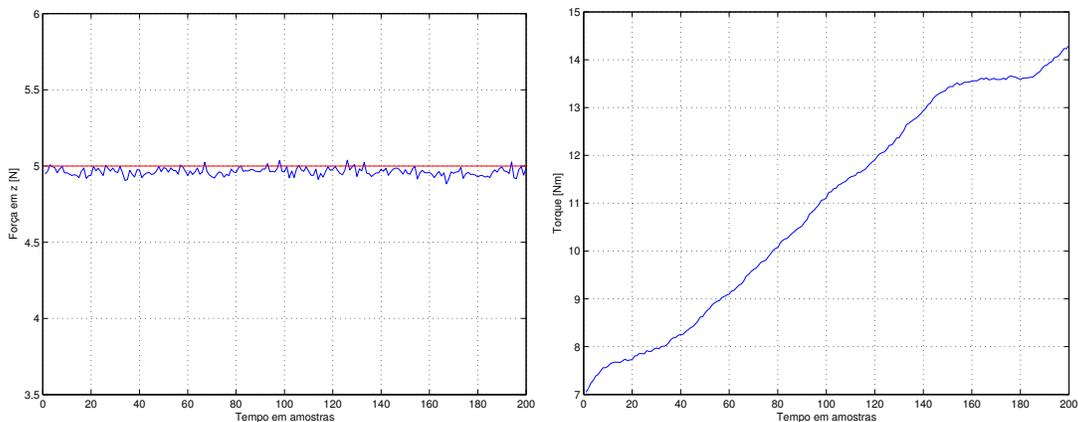


Figura 7.23: Posição e torque - Controle VSC

7.7 Conclusões do Capítulo

Neste capítulo, foram apresentados os resultados da implementação dos controladores de força e posição empregando a modelagem de De Luca e Manes. Foi estabelecida a respectiva transformação das variáveis s_{K_i} , s_{K_j} e λ_R em função das variáveis lidas pelos sensores. Duas estratégias de controle foram implementadas: o controle PI e o controle VSC . O controle VSC apresentou um melhor desempenho quando comparado com o controle PID . Foi verificado também que com o uso desta abordagem, é possível a hibridização de controladores no sentido de usar uma técnica para o controle de posição e uma outra para o controle de força. Finalmente, a implementação foi feita por etapas de forma a adquirir uma melhor compreensão do problema no que respeita aos aspectos práticos.

Capítulo 8

Conclusões e Perspectivas

A revisão bibliográfica apresentada no início deste trabalho evidencia uma certa lacuna nas publicações referentes ao controle de força de robôs manipuladores. Nota-se um grande número de publicações nas décadas de 80 e início dos anos 90. Boa parte das publicações mostra um bom desenvolvimento de estratégias de controle em termos teóricos. Aparentemente, a lacuna existente diz respeito às dificuldades para a realização de experimentos que auxiliem a passagem do campo teórico para o experimental, abrindo, em seguida, perspectivas para as aplicações industriais. Foi dentro deste contexto que o presente trabalho foi desenvolvido. Partindo inicialmente da modelagem, passando pela simulação para chegar em seguida aos problemas práticos.

Para a confirmação dos trabalhos desenvolvidos nesta tese, este capítulo apresenta as conclusões gerais e algumas perspectivas.

8.1 Conclusões

Inicialmente, foi apresentada a modelagem de De Luca e Manes para o controle de força e posição de robôs manipuladores. Esta modelagem inclui restrições cinemáticas e restrições dinâmicas quando o robô interage com o meio. A representação destas restrições é feita incluindo novas variáveis para o robô e para o meio. Estas variáveis merecem especial atenção quando deseja-se implementar leis de controle, já que as mesmas devem ser expressas em termos das variáveis lidas pelos sensores do robô.

Em seguida, foi considerado o problema do atrito e seus diferentes modelos. Deu-se ênfase na apresentação e análise do modelo de *LuGre*, que foi o empregado para compensar o atrito presente nas juntas do robô, estabelecendo uma forma sistemática para a obtenção dos parâmetros estáticos e dinâmicos do modelo.

O capítulo quatro foi dedicado às estratégias de controle empregadas para o projeto de leis de controle de força e posição. As estratégias de controle são o Controle de Dois Graus de Liberdade (TDOF), o Controle a Estrutura Variável (VSC) e o Controle Nebuloso ou Fuzzy (FLC). A primeira

delas consiste em uma estratégia de controle linear baseada num modelo linear para seu projeto. Uma vez determinado um modelo linear estabelecendo algumas aproximações no modelo não linear do robô, estas mesmas aproximações, posteriormente, são utilizadas para determinar os parâmetros estáticos e dinâmicos do atrito. Os ganhos do controlador TDOF são obtidos empregando a técnica do *LQR* de forma a obter os pólos ótimos que se desejam alocar e gerar os ganhos correspondentes. A segunda técnica corresponde à estratégia do controle a estrutura variável. Para o projeto de leis de controle empregando esta técnica, podem ser utilizados modelos lineares ou não. Foi considerado um modelo não linear para o projeto das leis de controle. Para obter um chaveamento mais suave, a função *sgn* é substituída pela função *tanh* e empregada na camada limite. A terceira estratégia corresponde aos controladores nebulosos. O controlador apresentado é baseado na técnica de controle a estrutura variável. A disposição de suas regras permite estabelecer uma similaridade com o controle a estrutura variável, ao mesmo tempo em que permite provar a estabilidade utilizando a teoria de *Lyapunov*. Nesta estratégia o problema de *chattering* é eliminado devido à forma com que é construída a lei de controle.

No quinto capítulo, foram apresentados o robô industrial de configuração *SCARA* (Inter), seus componentes e ferramentas. Detalharam-se o sistema operacional e a linguagem de programação *XO/2*. Finalmente, foram apresentados os módulos de controle do robô e especificados os módulos implementados neste trabalho.

No sexto capítulo, foram apresentados os resultados experimentais dos controladores de posição e/ou trajetória dos controladores *TDOF*, *VSC* e *FLC*. Para a realização dos experimentos, utilizaram-se trajetórias semi-trapezoidais de terceira ordem. Os três controladores foram projetados considerando um sistema criticamente amortecido. No projeto do controlador *TDOF*, foram feitas modificações no algoritmo *LQR* a fim de obter um sistema criticamente amortecido.

No caso do controle *VSC*, foi implementada uma lei de controle proposta por Amaral (2000) incluindo uma função do tipo *tanh* para evitar o problema do *chattering* e a utilização inadequada dos atuadores.

O controlador nebuloso *FLC* implementado corresponde a uma técnica análoga ao *VSC*. Este controlador de fato elimina o problema do *chattering* sendo esta sua principal vantagem. As funções de pertinência das entradas e saída são funções triangulares e sua implementação prática é relativamente simples, podendo ser comparada à implementação dos controladores clássicos ou de estrutura variável.

Foi feita também a compensação do atrito presente nas juntas do robô. Para este propósito, foi utilizado o modelo de *LuGre* e um observador de atrito. Para a identificação dos parâmetros estáticos do atrito foram empregados experimentos realizados a velocidade constante. Os parâmetros dinâmicos foram determinados utilizando os dados obtidos no início do movimento. Nos testes experimentais, observou-se que valores elevados nos parâmetros dinâmicos produzem vibrações no robô devido às dinâmicas de alta frequência não modeladas. Também observou-se que a principal dificuldade presente no modelo de *LuGre* refere-se ao estado não mensurável. O índice do erro absoluto foi utilizado para

fazer a comparação entre as três técnicas, verificando-se que os controladores *TDOF* e o controle *VSC* apresentam melhor desempenho que o controlador nebuloso. Este índice, conforme era de se esperar diminui quando é feita a compensação do atrito. Isso demonstra a necessidade da compensação do atrito, sendo essa compensação mais necessária segundo a precisão da tarefa e também a velocidade de manipulação.

Os resultados referentes ao controle de força foram apresentados no sétimo capítulo, sendo definida uma tarefa de corte usando a modelagem de De Luca e Manes para os três primeiros graus de liberdade do robô. Nesta etapa, o ponto mais importante foi a obtenção das variáveis da modelagem de De Luca e Manes a partir dos dados lidos pelos sensores do robô. A partir da tarefa definida, o problema do controle de impacto é analisado e apresentados os resultados experimentais para superfícies de diferente rigidez. Para o controle de força e posição, os controles *PID* e o *VSC* são projetados empregando a modelagem de De Luca e Manes. Nas medições de força verificou-se a presença de ruídos, razão pela qual foi necessário o projeto de filtros para obter uma sinal de força não contaminado.

No que diz a os objetivos propostos na tese, os mesmos foram alcançados podendo-se ressaltar que as principais contribuições dizem respeito aos resultados experimentais obtidos. Estes resultados permitiram a publicação dos seguintes trabalhos:

- *Controle de posição de um robô tipo SCARA usando controladores de dois graus de liberdade*, Congresso Brasileiro de Automática, Florianópolis, SC., Brasil, setembro 2000.
- *Estudo experimental de um controlador a Estrutura Variável e Compensação de atrito em um Robô Industrial*, Congresso Nacional de Engenharia Mecânica, Natal, RN, Brasil, setembro 2000.
- *Force and Position Control of Constrained Industrial Robots With Two Degrees of Freedom Controllers*, IX International Diname, SC, Brasil, março 2001.
- *Estratégia Nebulosa para o Controle de Força e Posição de Robôs Manipuladores*, Reunión de Trabajo en Procesamiento de la Información y Control, Santa Fe, Argentina, setembro 2001.
- *Avaliação experimental de uma estratégia nebulosa para o controle de posição de um robô industrial*, XIV Congresso Brasileiro de Automática, setembro, 2002.
- *Identificação e Compensação de Atrito Num Robô Industrial SCARA*, XV Congresso Brasileiro de Automática, Gramado, RS, Brasil, setembro 2004.
- *Experimental Implementation of Robust and Intelligent Controllers in Trajectory Control of Industrial SCARA Robot*, 18th IFIP World Computer Congress, Toulouse, France, 2004.
- *Experimental Evaluation of PD, PID and VSC Controllers with Friction Compensation in an Industrial Robot*, Induscon, Joinville, 2004.

- *Identification and Friction Compensation for an Industrial Robot Using Two Degrees of Freedom Controllers*, IEEE Eighth International Conference on Control, Automation, Robotics and Vision, China, 2004.
- *About Practical Implementation of Friction Compensation and Force-Position Control Using Variable Structure Control*, Practical Engineering Control Journal, Artigo submetido, dezembro de 2004.

8.2 Perspectivas

Baseados nos resultados obtidos e nas dificuldades encontradas, são sugeridos os seguintes temas de pesquisa:

- A ausência de sensores de velocidade requer a utilização de diferentes aproximações para a obtenção desta variável a partir das medições de posição. Este ponto é um incentivo para o desenvolvimento e a implementação de observadores para a obtenção da velocidade e da aceleração;
- Os diferentes modelos de atrito já implementados poderiam ser complementados com outros existentes na literatura. Uma comparação para verificar o desempenho experimental desses modelos pode ajudar na melhoria dos controladores e auxiliar também na difícil tarefa de escolher o modelo de atrito mais adequado a uma determinada tarefa;
- Baseados nesta primeira implementação de controladores nebulosos empregando a linguagem $XO/2$, é possível a hibridização desta técnica com outras, tais como o controle PID e, desta forma, incluir a componente integral nesta técnica;
- O grau de amadurecimento obtido para o controle de posição e/ou trajetória permite a implementação experimental de outras técnicas tais como a de back-stepping, de controle baseado na passividade e diversos controladores não lineares;
- A partir dos resultados até agora obtidos empregando a modelagem de De Luca e Manes, é possível completar esta abordagem para tarefas submetidas a restrições dinâmicas e flexíveis, e principalmente buscar tarefas de complexidade industrial e estudar seus modelos;
- Um outro objeto de estudo consiste na avaliação de técnicas para a obtenção de diferentes modelos lineares que tornem mais eficiente o uso dos controladores $TDOF$.

Apêndice A

Modelagem Cinemática

A.1 Sistema de Coordenadas

Os sistemas de coordenadas adotados no robô industrial de configuração *SCARA* são ilustrados na figura A.1. Os mesmos são empregados para a obtenção da cinemática direta, da cinemática inversa, e dos jacobianos.

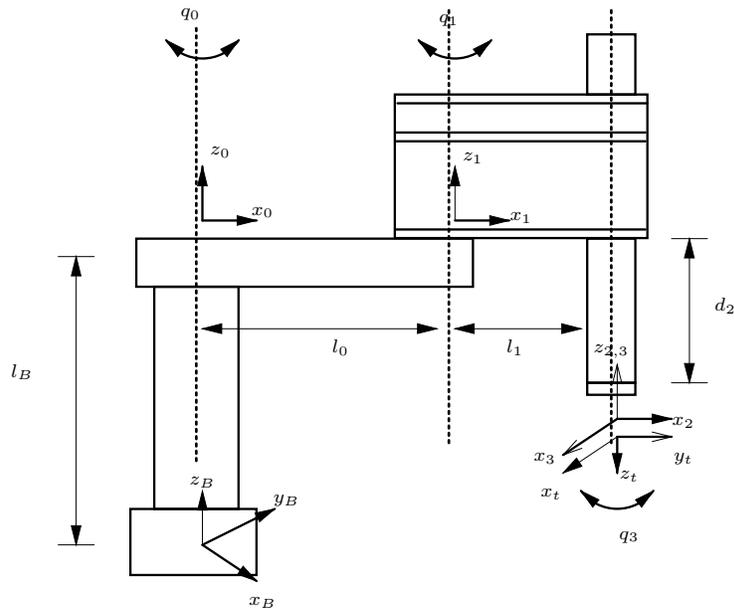


Figura A.1: Sistemas de coordenadas do robô *SCARA*

Pode-se observar, que no conjunto de coordenadas, considera-se a posição de uma ferramenta genérica definida pelas coordenadas x_t , y_t e z_t . Os parâmetros de *Denavit – Hartenberg* para este robô são apresentados na tabela A.1.

Tabela A.1: Parâmetros de *Denavit – Hartenberg*

Junta	α_i	l_i	$offset_i$	θ_i
0	0	l_0	l_b	q_0
1	0	l_1	0	q_1
2	0	0	0	q_2
3	0	0	0	q_3

A.2 Cinemática Direta

A posição do efetuador final no sistema de coordenadas cartesianas pode ser determinada a partir das transformações de *Denavit – Hartenberg* ou através de relações e igualdades trigonométricas. Neste sentido, a cinemática direta do robô *SCARA* é dada pela equação (A.1), onde está incluído o sistema de coordenadas da ferramenta (Refere-se às juntas de rotação e cilíndricas como sendo q_i tal como apresentado na tabela A.1).

$$X = \begin{bmatrix} x \\ y \\ z \\ \phi \end{bmatrix} = \begin{bmatrix} l_0 c_0 + l_1 c_{01} + x_t c_{013} - y_t s_{013} \\ l_0 s_0 + l_1 s_{01} + x_t s_{013} + y_t c_{013} \\ l_b + q_2 + z_t + k q_3 \\ q_0 + q_1 + q_3 \end{bmatrix} \quad (\text{A.1})$$

onde a constante k reflete o acoplamento mecânico entre as duas últimas juntas do robô e c_0 , s_0 , c_{01} , s_{01} , c_{013} e s_{013} são definidas como:

$$\begin{aligned} c_0 &= \cos q_0 \\ s_0 &= \sin q_0 \\ c_{01} &= \cos(q_0 + q_1) \\ s_{01} &= \sin(q_0 + q_1) \\ c_{013} &= \cos(q_0 + q_1 + q_3) \\ s_{013} &= \sin(q_0 + q_1 + q_3) \end{aligned}$$

A.2.1 Jacobiano analítico

O jacobiano analítico $J(q)$ é definido como sendo (Sciavicco e Siciliano, 2000):

$$\dot{X} = \frac{\partial K(q)}{\partial q} \dot{q} + \frac{\partial K_{TCP}(q)}{\partial q} \dot{q} = J(q) + J_{TCP} \quad (\text{A.2})$$

onde é incluída a componente devida à modelagem da ferramenta. Derivando a equação (A.1), obtém-se:

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -l_0 s_0 \dot{q}_0 - l_1 s_{01}(\dot{q}_0 + \dot{q}_1) - dx_t s_{013}(\dot{q}_0 + \dot{q}_1 + \dot{q}_3) - dy_t c_{013}(\dot{q}_0 + \dot{q}_1 + \dot{q}_3) \\ l_0 c_0 \dot{q}_0 + l_1 c_{01}(\dot{q}_0 + \dot{q}_1) + dx_t c_{013}(\dot{q}_0 + \dot{q}_1 + \dot{q}_3) - dy_t s_{013}(\dot{q}_0 + \dot{q}_1 + \dot{q}_3) \\ \dot{q}_2 + k\dot{q}_3 \\ \dot{q}_0 + \dot{q}_1 + \dot{q}_3 \end{bmatrix} \quad (\text{A.3})$$

Em forma matricial, sem considerar as coordenadas da ferramenta:

$$J(q) = \begin{bmatrix} -l_0 s_0 - l_1 s_{01} & -l_1 s_{01} & 0 & 0 \\ l_0 c_0 + l_1 c_{01} & l_1 c_{01} & 0 & 0 \\ 0 & 0 & 1 & k \\ 1 & 1 & 0 & 1 \end{bmatrix} \quad (\text{A.4})$$

A inclusão da ferramenta é dada por:

$$J_{TCP} = \begin{bmatrix} k_1 & k_1 & 0 & k_1 \\ k_2 & k_2 & 0 & k_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.5})$$

onde dx_t e dy_t correspondem às derivadas das coordenadas da ferramenta e:

$$k_1 = -dx_t s_{013} - dy_t c_{013} \quad (\text{A.6})$$

$$k_2 = dx_t c_{013} - dy_t s_{013} \quad (\text{A.7})$$

A.2.2 Derivada do Jacobiano analítico

A definição da derivada do Jacobiano é dado por:

$$\ddot{X} = J(q) \ddot{q} + \dot{J}(q, \dot{q}) \dot{q} + J_{TCP} \ddot{q} + \dot{J}_{TCP} \dot{q} \quad (\text{A.8})$$

onde $\dot{J}(q, \dot{q})$ é igual a:

$$\dot{J}(q, \dot{q}) = \begin{bmatrix} -l_0 c_0 \dot{q}_0 - l_1 c_{01}(\dot{q}_0 + \dot{q}_1) & -l_1 c_{01}(\dot{q}_0 + \dot{q}_1) & 0 & 0 \\ -l_0 s_0 \dot{q}_0 - l_1 s_{01}(\dot{q}_0 + \dot{q}_1) & -l_1 s_{01}(\dot{q}_0 + \dot{q}_1) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.9})$$

Novamente considerando a ferramenta obtém-se:

$$\dot{J}_{TCP} = \begin{bmatrix} k_3 & k_3 & 0 & k_3 \\ k_4 & k_4 & 0 & k_4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.10})$$

onde:

$$k_3 = -dx_t c_{013} + dy_t s_{013}$$

$$k_4 = -dx_t s_{013} - dy_t c_{013} \quad (\text{A.11})$$

$$(\text{A.12})$$

A.2.3 Inversa do Jacobiano analítico

A inversa do jacobiano é definida como:

$$\dot{q} = J^{-1}(q)\dot{x} + J_{TCP}^\dagger \dot{x} \quad (\text{A.13})$$

$$J^{-1}(q) = \begin{bmatrix} \frac{c_{01}}{l_0 s_1} & \frac{s_{01}}{l_0 s_1} & 0 & 0 \\ -\frac{l_0 c_0 + l_1 c_{01}}{l_0 l_1 s_1} & -\frac{l_0 s_0 + l_1 s_{01}}{l_0 l_1 s_1} & 0 & 0 \\ -\frac{k c_0}{l_1 s_1} & -\frac{k s_0}{l_1 s_1} & 1 & -k \\ \frac{c_0}{l_1 s_1} & \frac{s_0}{l_1 s_1} & 0 & 1 \end{bmatrix} \quad (\text{A.14})$$

$$J_{TCP}^\dagger = \begin{bmatrix} 0 & 0 & 0 & -\frac{k_1 c_{01} + k_2 s_{01}}{l_0 s_1} \\ 0 & 0 & 0 & \frac{k_1 l_0 s_0 + k_2 (l_0 s_0 + l_1 s_{01})}{l_0 l_1 s_1} \\ 0 & 0 & 0 & k \frac{k_1 c_0 + k_2 s_0}{l_1 s_1} \\ 0 & 0 & 0 & -\frac{k_1 c_0 + k_2 s_0}{l_1 s_1} \end{bmatrix} \quad (\text{A.15})$$

A.3 Cinemática Inversa

A cinemática inversa permite o mapeamento das variáveis do robô dadas no sistema cartesiano para sistema de referencia de juntas, a mesma é determinada pela seguinte expressão:

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \text{atan2}(x_2, y_2) - \text{atan2}(l_1 s_1, l_0 + l_1 c_1) \\ \pm \text{acos}(c_1) \\ z - l_b - z_t - k q_3 \\ \phi - q_0 - q_1 \end{bmatrix} \quad (\text{A.16})$$

onde:

$$\begin{aligned}x_2 &= x - x_t \cos \phi + y_t \sin \phi \\y_2 &= y - x_t \sin \phi - y_t \cos \phi \\c_1 &= \frac{x_2^2 + y_2^2 - l_0^2 - l_1^2}{2l_0l_1} \\s_1 &= \sin(\pm \arccos(c_1))\end{aligned}\tag{A.17}$$

As equações apresentadas neste apêndice permitem a passagem do espaço de juntas para o espaço operacional e do espaço operacional para o espaço de juntas do robô. São apresentadas também as transformações do sistema de coordenadas do sensor para o espaço operacional, visando a implementação do controle de força (Weihmann, 1999), (Golin, 2002).

Apêndice B

Controle a Estrutura Variável - Conceitos Básicos

Neste apêndice, são introduzidos os principais conceitos do controle a estrutura variável. Descrições mais detalhadas e um resumo do estado da arte podem ser encontradas em Utkin (1977), Bailey e Arapostathis (1987), DeCarlo *et al.* (1988) Young *et al.* (1999), Utkin (1992).

B.1 Controle a Estrutura Variável

O objetivo deste tipo de controle é conseguir a redução da ordem das equações diferenciais que descrevem o sistema, dividindo o projeto em duas etapas:

1. A escolha da superfície de deslizamento.
2. A determinação da lei de controle chaveada.

B.1.1 Definições

O controle a Estrutura Variável utiliza leis chaveadas de controle extremamente rápidas, capazes de dirigir a trajetória do sistema no espaço de estados para hipersuperfícies definidas em projeto, conhecidas como *superfícies de deslizamento*, que, uma vez atingidas, devem confinar a trajetória do sistema que desliza sobre ela até o equilíbrio. Como consequência, o comportamento do sistema entra em regime de deslizamento ou *modo deslizante*. Estas superfícies são funções escalares do vetor de estados que determinam o modo como o chaveamento da lei de controle deve ser executado.

Modelo do sistema

A equação não-linear geral dos sistemas a serem controlados com a estratégia de estrutura variável é representada por:

$$\dot{x} = f(x, u, t), \quad x \in \mathcal{R}^n, \quad u \in \mathcal{R}^m \quad (\text{B.1})$$

Os sistemas não-lineares, representados em (B.1), serão considerados dentro de uma classe especial desta equação, na qual o vetor de estado x é não-linear e a lei de controle u é afim ou linear no controle (os robôs manipuladores possuem esta propriedade):

$$\dot{x}(t) = f(x, t) + B(x, t)u(x, t) \quad (\text{B.2})$$

onde $x(t) \in \mathcal{R}^n$, $u \in \mathcal{R}^m$, $f(x, t) \in \mathcal{R}^n$ e $B(x, t) \in \mathcal{R}^{n \times m}$, sendo que as i componentes de $f(x, t)$ e de $B(x, t)$ são contínuas com suas derivadas limitadas e contínuas com respeito ao vetor de estado x .

A equação de estado (B.2) é sujeita a uma lei de controle descontínua do tipo:

$$u_i = \begin{cases} u_i^+(x, t) & \text{quando } \sigma_i(x, t) > 0 \\ u_i^-(x, t) & \text{quando } \sigma_i(x, t) < 0 \end{cases} \quad i = 1, 2, \dots, m \quad (\text{B.3})$$

onde $\sigma_i(x, t) = 0$ é a i -ésima superfície de deslizamento, associada com a superfície de dimensão m da forma:

$$\sigma(x, t) = [\sigma_1(x, t), \dots, \sigma_m(x, t)]^T = 0 \quad (\text{B.4})$$

Assim, $\sigma(x, t) = 0$ é determinada pela intersecção das m superfícies de deslizamento de $\sigma_i(x, t) = 0$. Estas superfícies de deslizamento são projetadas de modo que a resposta do sistema restrito a $\sigma(x, t) = 0$ tenha um comportamento desejado, incluindo particularmente a estabilidade ou seguimento de uma trajetória (Hung *et al.*, 1993).

O conjunto de funções $\sigma_i(x, t) = 0$, no espaço de estados define uma hipersuperfície para a qual a trajetória do sistema deve ser conduzida. Se num instante t_0 , o estado do sistema se encontra acima da superfície $\sigma(x, t) = 0$, a lei de controle $u^+(x, t)$ o conduz ao encontro de $\sigma(x, t) = 0$. Ao passar pela superfície, o controle é chaveado para $u^-(x, t)$, que traz novamente o sistema de volta a $\sigma(x, t) = 0$.

Superfícies de deslizamento

Superfícies da forma (B.4) não lineares podem ser empregadas, mas o projeto delas é bastante complexo. Além disso, as superfícies lineares são mais utilizadas por responderem de forma adequada para vários tipos de sistemas, incluindo os robôs manipuladores.

Definição B.1 : A função $\sigma(x, t)$ linear é descrita na forma:

$$\sigma(x, t) = Cx(t) = 0 \quad (\text{B.5})$$

onde C é uma matriz de dimensão $\mathcal{R}^{m \times n}$ com coeficientes reais.

Modos deslizantes

Um aspecto muito importante na estratégia a Estrutura Variável é garantir a existência do *modo deslizante* Utkin (1977), Hung *et al.* (1993), Gao e Hung (1993).

Definição B.2 : O modo deslizante existe se, na vizinhança da superfície de deslizamento $\sigma(x, t) = 0$, o vetor velocidade da trajetória de estados \dot{x} está sempre direcionado sobre a superfície de deslizamento.

Considerando a definição B.2, se o modo deslizante existe sobre $\sigma(x, t) = 0$, então $\sigma(x, t)$ é chamada *superfície de deslizamento*. No caso de hipersuperfícies, isto é, $m > 1$, o modo deslizante deverá existir na intersecção das superfícies $\sigma_i = 0$, conforme está ilustrado na figura B.1, para $m = 2$.

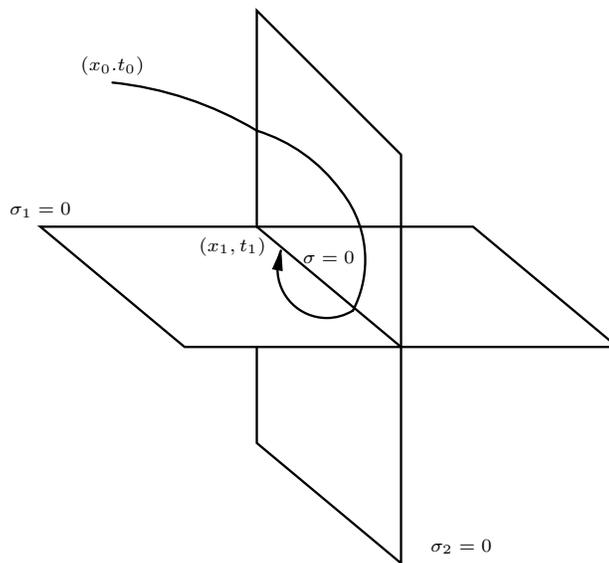


Figura B.1: Superfícies de deslizamento

B.1.2 Projeto da Superfície de Deslizamento

Em Utkin (1977) e diversos outros autores, propõe-se o *método de controle equivalente*¹, para obter o comportamento do sistema quando este estiver em modo deslizante ou restrito a $\sigma(x, t) = 0$.

Suponha-se que, no instante t_0 , o estado inicial do sistema descrito em (B.2) intercepta a superfície $\sigma(x, t) = 0$, e que o modo deslizante existe para $t \geq t_0$. A existência do modo deslizante implica que:

$$\sigma(x, t) = 0 \Rightarrow \dot{\sigma}(x, t) = 0 \quad \forall t \geq t_0 \quad (\text{B.6})$$

¹Técnica formal usada para determinar as equações ideais de modo deslizante (Utkin, 1977).

Empregando a regra da cadeia e derivando-se $\sigma(x, t) = 0$, obtém-se:

$$\dot{\sigma}(x, t) = \frac{\partial \sigma(x, t)}{\partial t} + \frac{\partial \sigma(x, t)}{\partial x} \dot{x} = 0 \quad (\text{B.7})$$

Substituindo a equação de estados (B.2) em (B.7), obtém-se:

$$\dot{\sigma}(x, t) = \frac{\partial \sigma(x, t)}{\partial t} + \frac{\partial \sigma(x, t)}{\partial x} [f(x, t) + B(x, t)u(x, t)] = 0 \quad (\text{B.8})$$

Assumindo que o produto $\frac{\partial \sigma(x, t)}{\partial x} B(x, t)$ é não singular e, portanto, possui inversa, pode-se isolar o termo $u(x, t)$ como:

$$u_{eq}(x, t) = - \left[\frac{\partial \sigma(x, t)}{\partial x} B(x, t) \right]^{-1} \left[\frac{\partial \sigma(x, t)}{\partial t} + \frac{\partial \sigma(x, t)}{\partial x} f(x, t) \right] \quad (\text{B.9})$$

A equação (B.9) é conhecida como **lei de controle equivalente** (Utkin, 1977): representa a entrada que, ao excitar o sistema, conduz a trajetória de estados do sistema x ao longo de $\sigma(x, t) = 0$.

B.1.3 Projeto do Controlador

Na subseção B.1.2, foi considerado que a superfície já havia sido atingida pela trajetória de estados do sistema e que o controle equivalente expresso pela equação (B.9) é a parcela de controle responsável em guiar a dinâmica restrita ao longo do deslizamento. É necessário, pois, determinar uma parcela de controle, u_N , que leve o sistema ao encontro da superfície $\sigma(x, t) = 0$ para que o deslizamento ocorra. Para isso, define-se o controle como sendo composto pela soma de duas componentes:

$$u = u_{eq} + u_N \quad (\text{B.10})$$

O termo $u_N(t)$ pode ser determinado através da aplicação do método direto de Lyapunov (Utkin, 1992), (DeCarlo *et al.*, 1988), de forma a definir-se a seguinte função de Lyapunov em termos da superfície de deslizamento:

$$V(\sigma, t) = \sigma^T W \sigma \quad (\text{B.11})$$

onde W é uma matriz constante simétrica definida positiva. Portanto, se a função $V(\sigma, t)$ é uma forma quadrática, ela é positiva para $\sigma \neq 0$. Sua derivada em função do tempo é:

$$\dot{V}(\sigma, t) = \dot{\sigma}^T W \sigma + \sigma^T W \dot{\sigma} = 2\sigma^T W \dot{\sigma} \quad (\text{B.12})$$

Substituindo a derivada de σ , dada pela equação (B.8) em (B.12), obtém-se:

$$\dot{V}(\sigma, t) = 2\sigma^T W \left[\frac{\partial \sigma}{\partial t} + \frac{\partial \sigma}{\partial x} [f(x, t) + B(x, t)u(x, t)] \right] \quad (\text{B.13})$$

Pode-se agora empregar o teorema de Lyapunov para a estabilidade global de sistemas não lineares conforme proposto por Bailey e Arapostathis (1987), Santibañez e Kelly (1997).

Teorema B.1 : *Se um sistema dinâmico $\dot{x} = f(x, u, t)$, com uma função escalar do vetor de estados $V(x)$, cuja derivada de primeira ordem 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 é um ponto de equilíbrio globalmente assintoticamente estável (Slotine e Li, 1991).

A aplicação do teorema B.1, nas equações (B.11) e (B.13), se faz no sentido de que a superfície de deslizamento seja assintoticamente estável a partir de qualquer condição inicial do espaço de estados, se $\dot{V}(x, t) \leq 0$.

Essa é, de fato, a situação esperada para garantir o deslizamento, visto que o comportamento dinâmico do sistema, após alcançada a superfície, é dada pela equação (B.9). É necessário, portanto, garantir que a derivada temporal de $V(\sigma, t)$ expressa em (B.13) seja definida negativa.

$$\dot{V}(\sigma, t) = 2\sigma^T W \left[\frac{\partial \sigma}{\partial t} + \frac{\partial \sigma}{\partial x} [f(x, t) + B(x, t) [u_{eq}(x, t) + u_N(x, t)]] \right] \quad (\text{B.14})$$

Substituindo (B.9) em (B.14), tem-se:

$$\dot{V}(x, t) = 2\sigma^T W \frac{\partial \sigma}{\partial x} B(x, t) u_N(x, t) \quad (\text{B.15})$$

Sendo que o termo $\frac{\partial \sigma(x, t)}{\partial x} B(x, t)$ foi assumido não singular, pode-se definir:

$$u_N = \left[\frac{\partial \sigma}{\partial x} B(x, t) \right]^{-1} u_N^*(x, t) \quad (\text{B.16})$$

Ainda para simplificar, pode-se escolher $W = I$, já que a restrição em W é ser definida positiva, e sua ponderação pode ser compensada na lei de controle. Então, a equação (B.15) toma a forma:

$$\dot{V}(\sigma, t) = 2\sigma^T u_N^* \quad (\text{B.17})$$

Para que esta função seja definida negativa, e, pelo teorema B.1, se garanta a estabilidade assintótica da superfície de deslizamento, podem ser definidas diferentes leis de controle u_N^* (Slotine e Li, 1991).

Apêndice C

Controle Nebuloso - Conceitos Básicos

Neste apêndice, são apresentados os conceitos básicos da lógica nebulosa utilizados no projeto do controlador nebuloso *FLC*. Detalhes podem ser obtidos em (Zadeh, 1965), (Zadeh, 1996), (Yi e Chung, 1994), (Yi e Chung, 1997) e (Yi e Chung, 1998).

C.1 Conjuntos Nebulosos

Na teoria de conjuntos nebulosos, conjuntos basicamente são denominados como conjunto *crisp*. Se um conjunto *crisp* C pertence ao conjunto dos números reais \mathcal{R} , é verdadeira a preposição de que um elemento $x \in \mathcal{R}$ implica que $x \in C$ ou $x \notin C$. Já nos conjuntos nebulosos, esta preposição não necessariamente é verdadeira, ou seja, um elemento $x \in \mathcal{R}$ pode pertencer parcialmente ao conjunto nebuloso N . Esta propriedade caracteriza o conceito de informação vaga ou imprecisa. Os conjuntos nebulosos são caracterizados por funções de pertinência que assumem valores contínuos entre *falso* (0) ou *verdadeiro* (1).

Definição C.1 Uma função de pertinência μ_N de um conjunto nebuloso N é uma função da forma

$$\mu_N : \mathcal{R} \rightarrow [0, 1] \tag{C.1}$$

a qual determina o grau de pertinência $\mu_N(x)$ de um elemento x num conjunto nebuloso N .

Um conjunto nebuloso é completamente definido por

$$F = \{(x, \mu_N(x)) | x \in \mathcal{R}, \mu_N(x) : \mathcal{R} \rightarrow [0, 1]\} \tag{C.2}$$

As funções de pertinência μ_N podem ter formas triangular, trapezoidal, gaussiana ou do tipo singleton, conforme ilustrado na figura C.1.

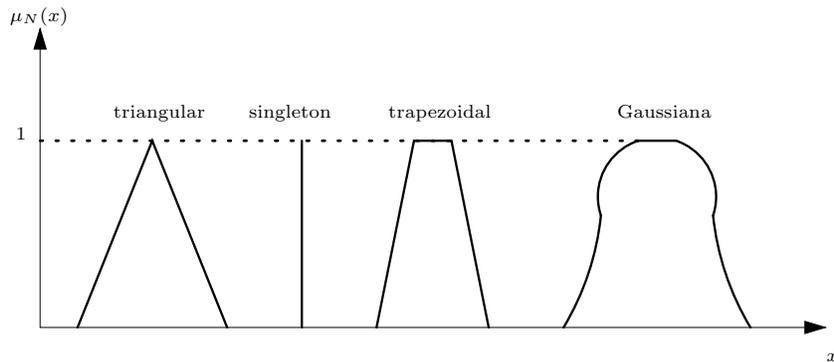


Figura C.1: Formas típicas das funções de pertinência

Um conjunto nebuloso denominado *singleton* apresenta uma forma especial para a função de pertinência. Um *singleton* é um conjunto que possui apenas um elemento x_0 com a função de pertinência dada por:

$$\mu_F = \begin{cases} 1, & \text{se } x = x_0 \\ 0, & \text{se } x \neq x_0 \end{cases} \quad (\text{C.3})$$

Singletons são bastante utilizados como conseqüentes de regras nebulosas.

Através das funções de pertinência avalia-se o grau de verdade de uma preposição nebulosa.

Definição C.2 Uma preposição nebulosa é uma expressão do tipo " x é N ", onde N é um conjunto nebuloso com função de pertinência μ_N . A função de pertinência é a medida do grau de verdade da preposição nebulosa.

Sistemas nebulosos são uma coleção de regras {se}...{então} formadas por preposições relacionadas através de conectivos lógicos do tipo {e} e {ou}.

C.2 T-norma e T-conorma

Conjuntos nebulosos são geralmente relacionados através de uma função *interseção* ou *união*, representadas através de normas do tipo *T-norma*, *T-conorma* (ou *S-norma*). A função de pertinência do conjunto nebuloso resultante é composta de funções de pertinência dos conjuntos originais através de *T-norma* ou *T-conorma*. As normas são também utilizadas nos mecanismos de raciocínio nebuloso.

Sejam a , b e c os valores de verdade de uma preposição nebulosa.

Definição C.3 *Seja $T : [0, 1] \times [0, 1] \rightarrow [0, 1]$, T é uma T -norma se e somente se para todo $a, b, c \in [0, 1]$:*

$$T(a, b) = T(b, a) \quad (\text{C.4})$$

$$T(a, b) \leq T(a, c), \text{ se } b < c \quad (\text{C.5})$$

$$T(a, T(b, c)) = T(T(a, b), c) \quad (\text{C.6})$$

$$T(a, 1) = a \quad (\text{C.7})$$

$$(\text{C.8})$$

onde a, b e c são valores de verdade das funções de pertinência.

T -norma é utilizada para representar a intersecção entre dois conjuntos nebulosos ou um conectivo lógico **e** entre duas preposições nebulosas. Zadeh (1965) propôs a função $T(a, b) = \min(a, b)$ para a T -norma, atualmente o produto $T(a, b) = ab$ é mas utilizado.

Definição C.4 *Seja $S : [0, 1] \times [0, 1] \rightarrow [0, 1]$, S é uma S -norma se e somente se para todo $a, b, c \in [0, 1]$:*

$$S(a, b) = S(b, a) \quad (\text{C.9})$$

$$S(a, b) \leq S(a, c), \text{ se } b < c \quad (\text{C.10})$$

$$S(a, S(b, c)) = S(S(a, b), c) \quad (\text{C.11})$$

$$S(a, 0) = a \quad (\text{C.12})$$

$$(\text{C.13})$$

T -conorma é utilizada para representar a união entre dois conjuntos nebulosos ou um conectivo lógico **ou** entre duas preposições nebulosas. Zadeh (1965) propôs a função $T(a, b) = \max(a, b)$ para a T -conorma; outra possibilidade bastante utilizada é $S(a, b) = a + b - ab$. Outros tipos de normas também podem ser encontrados em Almeida (2002).

C.3 Raciocínio Nebuloso

No raciocínio nebuloso, a decisão é baseada na informação de entrada e no conhecimento armazenado na base de regras. As regras são formadas por preposições nebulosas nos antecedentes e conseqüentes das mesmas. O método de raciocínio nebuloso, por sua vez, é classificado com base na forma do conseqüente das regras.

C.3.1 Regras nebulosas tipo Mamdani

Se em ambos antecedentes e conseqüentes, as regras são combinadas utilizando-se conectivos nebulosos, estas são do tipo Mamdani e são expressas da seguinte forma:

$$\text{se } x_1 \text{ é } X_1^i \text{ e } x_2 \text{ é } X_2^i \text{ e } \dots \text{ e } x_{nx} \text{ é } X_{nx}^i \text{ então } z \text{ é } Z^i$$

onde X_j^i é o conjunto nebuloso da j -ésima entrada (nx é o número de entradas) e Z^i é o conjunto nebuloso da saída, ambos relacionados através da i -ésima regra.

Na execução do raciocínio nebuloso aplicando as regras do tipo Mamdani, inicialmente o valor verdade dos antecedentes são avaliados conforme o conectivo utilizado. Para as regras do tipo Mamdani em que o conectivo **e** é utilizado, o valor verdade é determinado por:

$$\psi_i = T\left(\mu_{X_1^i}(x_1), \dots, \mu_{X_{nx}^i}(x_{nx})\right) \quad (\text{C.14})$$

O conjunto nebuloso de saída é obtido através da avaliação com T -norma dos conseqüentes das regras nebulosas. O raciocínio nebuloso produz um conjunto nebuloso de saída do tipo:

$$\mu_{Z^i} = T(\psi_i, \mu_{Z^i}(z)) \quad (\forall z \in \mathcal{R}) \quad (\text{C.15})$$

Os conseqüentes de todas as regras são combinados juntos (agregados) através da T -conorma como segue:

$$\mu_{Z'}(z) = S(\mu_{Z^1}(z), \dots, \mu_{Z^m}(z)) \quad (\forall z \in \mathcal{R}) \quad (\text{C.16})$$

onde m é o número de regras.

O raciocínio nebuloso do tipo Mamdani é ilustrado na figura C.2 para o caso de duas entradas e duas regras do tipo:

$$\begin{aligned} \text{se } x_1 \text{ é } X_1^1 \text{ e } x_2 \text{ é } X_2^1 \text{ então } z \text{ é } Z^1 \\ \text{se } x_1 \text{ é } X_1^2 \text{ e } x_2 \text{ é } X_2^2 \text{ então } z \text{ é } Z^2 \end{aligned}$$

utilizando-se a função *min* para o conectivo T-norma e a função *max* para o conectivo T-conorma.

C.4 Desnebulização

Regras do tipo Mamdani produzem uma saída nebulosa. Na prática, um valor "Crisp" é necessário como saída. Portanto, um procedimento de desnebulização é necessário.

Assumindo que o conseqüente do raciocínio nebuloso é uma função de pertinência $\mu(z)$, o valor de saída do conseqüente nebuloso pode ser obtido através de diferentes métodos (Almeida, 2002). O

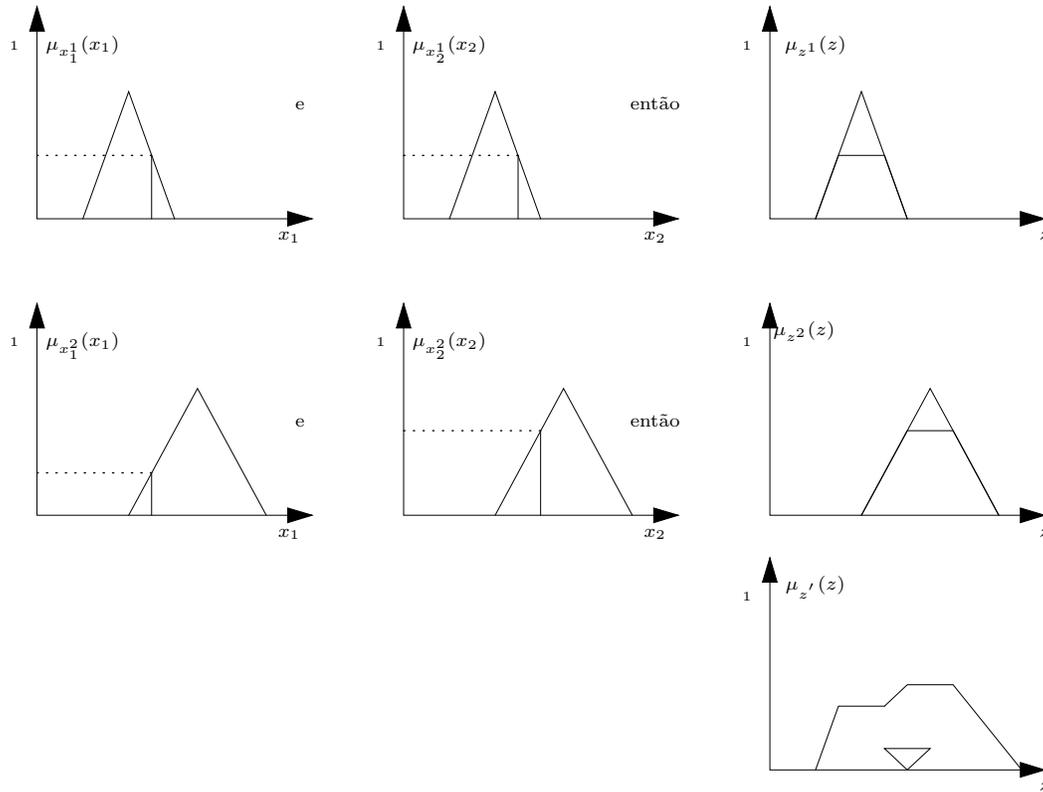


Figura C.2: Raciocínio nebuloso

método mais comum emprega a técnica do centro de gravidade.

$$z = \frac{\sum_{i=1}^n \mu(z_i) z_i}{\sum_{i=1}^m \mu(z_i)} \tag{C.17}$$

onde z_i são pontos da função de pertinência de saída $\mu(z)$ e n o número de pontos (C.3).

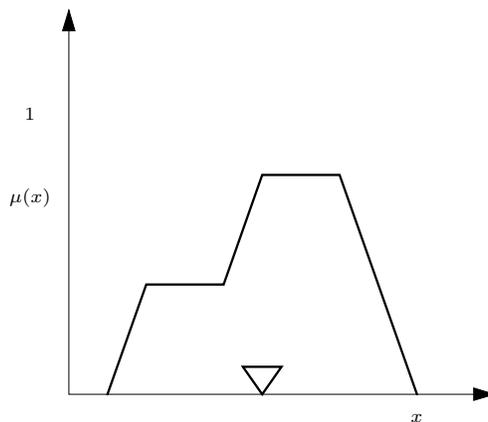


Figura C.3: Desnebulização pelo método da gravidade

Outras definições podem ser encontradas em (Sandri e Correia, 1999) e (Almeida, 2002).

Apêndice D

Controle Digital

Neste apêndice são apresentados os controladores digitais implementados em forma experimental, a partir de seus equivalentes analógicos.

D.1 Controle *PID* Digital

Na atualidade, muitos problemas práticos de controle são resolvidos empregando controladores *PID* (Aström e Wittenmark, 1997), (Lewis e Stevens, 1992). A versão analógica deste controlador é descrita pela seguinte equação:

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \dot{e}(t) \right) \quad (\text{D.1})$$

onde K_p é o ganho proporcional, T_i é a constante de integração e T_d corresponde ao tempo derivativo.

Diferentes versões digitais da equação (D.1) podem ser obtidas, sendo as mais empregadas o *algoritmo de posição* e o *algoritmo de velocidade*. Destes algoritmos o primeiro precisa o tratamento especial da saturação devido à ação integral. Essa característica não está presente no *algoritmo de velocidade*.

O *algoritmo de velocidade* pode ser obtido derivando a equação (D.1) de forma que:

$$\dot{u}(t) = K_p \left(\dot{e}(t) + \frac{1}{T_i} e(t) + T_d \ddot{e}(t) \right) \quad (\text{D.2})$$

Empregando a aproximação *backward* (Aström e Wittenmark, 1997) na equação (D.2) é obtida:

$$\ddot{e}(k) = \frac{e(k) - 2e(k-1) + e(k-2)}{T^2} \quad (\text{D.3})$$

Desta forma, a versão digital da equação (D.1) é dada por:

$$u(k) = u(k-1) + K_1 e(k) + K_2 e(k-1) + K_3 e(k-2) \quad (D.4)$$

onde K_1 , K_2 e K_3 são iguais a:

$$\begin{aligned} K_1 &= K_p \left(1 + \frac{T}{T_i} + \frac{T_d}{T}\right) \\ K_2 &= -K_p \left(1 + 2\frac{T_d}{T}\right) \\ K_3 &= K_p \frac{T_d}{T} \end{aligned} \quad (D.5)$$

onde T é o tempo de amostragem. A equação (D.4) foi implementada experimentalmente no robô.

D.2 Controle *TDOF* Digital

A figura D.1 ilustra o equivalente digital do controlador *TDOF* onde as funções de transferência analógicas foram transformadas em seu equivalente digital empregando a aproximação *trapezoidal* conforme Aström e Wittenmark (1997).

$$s = \frac{2(z-1)}{T(z+1)} \quad (D.6)$$

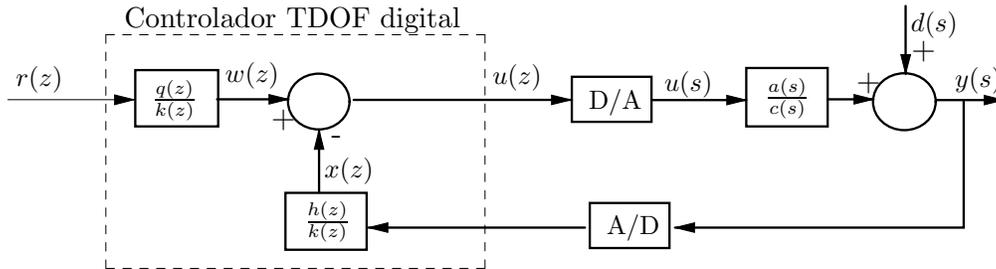


Figura D.1: Controlador *TDOF* digital

Desta forma, a $FT \frac{h(s)}{k(s)}$ do controlador *TDOF* é dada por:

$$\frac{h(s)}{k(s)} = \frac{h_1 s^2 + h_2 s + h_3}{k_1 s^2 + k_2 s + k_3} \quad (D.7)$$

Empregando a aproximação *trapezoidal* é obtido:

$$\frac{h(z)}{k(z)} = \frac{\overbrace{(4h_1 + 2Th_2 + T^2h_3)}^{h_{1d}} + \overbrace{(-8h_1 + 2T^2h_3)}^{h_{2d}} z^{-1} + \overbrace{(4h_1 - 2Th_2 + T^2h_3)}^{h_{3d}} z^{-2}}{(4k_1 + 2Tk_2 + T^2k_3) + (-8k_1 + 2T^2k_3)z^{-1} + (4k_1 - 2Tk_2 + T^2k_3)z^{-2}} \quad (D.8)$$

O mesmo pode ser feito para a $FT \frac{q(s)}{k(s)}$. O sinal de controle $u(k)$ é implementado através do equivalente digital do controle $TDOF$ (D.8) utilizando as seguintes equações a diferenças:

$$\begin{aligned}w(k) &= \frac{1}{k_{1d}}[q_{1d}r(k) + q_{2d}r(k-1) + q_{3d}r(k-2) - k_{2d}w(k-1) - k_{3d}w(k-2)] \\x(k) &= \frac{1}{k_{1d}}[h_{1d}y(k) + h_{2d}y(k-1) + h_{3d}y(k-2) - k_{2d}x(k-1) - k_{3d}x(k-2)] \\u(k) &= w(k) - x(k)\end{aligned}\tag{D.9}$$

O controlador $TDOF$ implementado experimentalmente corresponde ao ilustrado na figura D.1 representado pelas equações (D.8) e (D.9).

Apêndice E

Implementação de Controladores

Neste apêndice são incluídos os procedimentos em tempo real implementados ao longo deste trabalho.

```
PROCEDURE (h: ControlVSHdl) Run(); VAR
i: LONGINT;
g1, g2, q: LONGREAL;
done: BOOLEAN;
vel: ARRAY 4 OF ARRAY 4 OF LONGREAL;
cons: ARRAY 4 OF LONGREAL;
BEGIN
done:= JR3.ReadForce(h.robot.force^);

FOR i:=0 TO 3 DO
vel[i,0]:=0.0;
vel[i,1]:=0.0;
vel[i,2]:=0.0;
vel[i,3]:=0.0;
END;

FOR i := 0 TO h.robot.nq-1 DO
q := h.robot.counter[i].Read();
h.robot.qdact[i] := (1/5)*(vel[i,0]+vel[i,1]+vel[i,2]+vel[i,3])+(q-h.robot.qact[i])/samplingTimeVS;
(*h.robot.qdact[i] := (q-h.robot.qact[i])/samplingTimeVS; *)
vel[i,3] := vel[i,2];
vel[i,2] := vel[i,1];
vel[i,1] := vel[i,0];
vel[i,0] := h.robot.qdact[i];

h.robot.qact[i] := q;
h.robot.qe[i] := h.robot.q[i]-h.robot.qact[i];
h.robot.sigma[i] := h.robot.k[i]*(h.robot.qact[i]-h.robot.q[i])+(h.robot.qdact[i]-h.robot.qd[i]);
delta[i,0] := delta[i,1]+h.robot.epsilon[i]*(((sigma[i,0]-sigma[i,1])/samplingTimeVS)+ h.robot.c[i]*MathLib.tanh(cons[i],sigma[i,1]));
(*h.robot.qdVS[i] := -1*h.robot.c[i]*MathLib.tanh((h.robot.sigma[i]/h.robot.epsilon[i]));*)
END;

FOR i := 0 TO h.robot.nq-1 DO
h.robot.tauVS[i] := h.robot.m[i]*h.robot.qddVS[i];
IF (h.robot.securityState.no = noErr) THEN
IF h.robot.lugre THEN
h.robot.analogOut[i].Write(SHORT(h.robot.tauVS[i]+h.robot.tauFF[i]+h.robot.tauFrict[i]));
(* Total torque to the actuator, if no error, is FB + FF + Friction *)
ELSIF h.robot.lugre THEN
h.robot.analogOut[i].Write(SHORT(h.robot.tauVS[i]+h.robot.tauFF[i]));
(* Total torque to the actuator *)
END;
ELSE
h.robot.qd[i] := 0.0;
h.robot.qdd[i] := 0.0;
h.robot.qact[i] := h.robot.q[i];
h.robot.qdact[i] := 0.0;
h.robot.qobs[i] := h.robot.q[i];
h.robot.qdobs[i] := 0.0;
h.robot.qe[i] := 0.0;
h.robot.qddVS[i] := 0.0;
h.robot.tauVS[i] := 0.0;
h.robot.tauFF[i] := 0.0;
h.robot.tauFrict[i] := 0.0;
h.robot.analogOut[i].Write(0.0);
END;
END;

(* Update actual cartesian position *)
h.robot.DirKinematics(h.robot.qact^, h.robot.dx^, h.robot.xact^);

(* Update actual operational ("cartesian") velocity of the tool (TCP system) - in base coordinates *)
h.robot.OperationalSpeed(h.robot.qact^, h.robot.qdact^, h.robot.dx^, h.robot.xdact^);
END Run;
```

```

PROCEDURE (h: ControlDOHd1) Run(); VAR
  g1, g2: LONGREAL;
  i, j: LONGINT;
  q: LONGREAL;
  done : BOOLEAN;
  vel: ARRAY 4 OF ARRAY 4 OF LONGREAL;

BEGIN

  (* Update force sensor measurements *)
  done:= JR3.ReadForce(h.robot.force`);

  FOR i:=0 TO 3 DO
    vel[i,0]:=0.0;
    vel[i,1]:=0.0;
    vel[i,2]:=0.0;
    vel[i,3]:=0.0;
  END;

  FOR i := 0 TO h.robot.nq-1 DO
    q := h.robot.counter[i].Read();
    h.robot.qdact[i] := (1/5)*(vel[i,0]+vel[i,1]+vel[i,2]+vel[i,3])+(q-h.robot.qact[i])/samplingTimeD0;
    (* h.robot.qdact[i] := (q-h.robot.qact[i])/samplingTimeD0; *)
    vel[i,3] := vel[i,2];
    vel[i,2] := vel[i,1];
    vel[i,1] := vel[i,0];
    vel[i,0] := h.robot.qdact[i];

    h.robot.qact[i] := q;
    h.robot.qe[i] := h.robot.q[i]-h.robot.qact[i];

    (* Two Degree Controller*)

    h.robot.R[i,3]:=h.robot.R[i,2]; h.robot.R[i,2]:=h.robot.R[i,1]; h.robot.R[i,1]:=h.robot.R[i,0];
    h.robot.Y[i,3]:=h.robot.Y[i,2]; h.robot.Y[i,2]:=h.robot.Y[i,1]; h.robot.Y[i,1]:=h.robot.Y[i,0];

    h.robot.R[i,0]:=h.robot.q[i];
    h.robot.Y[i,0]:=h.robot.qact[i];

    h.robot.W[i,3]:=h.robot.W[i,2]; h.robot.W[i,2]:=h.robot.W[i,1]; h.robot.W[i,1]:=h.robot.W[i,0];
    h.robot.X[i,3]:=h.robot.X[i,2]; h.robot.X[i,2]:=h.robot.X[i,1]; h.robot.X[i,1]:=h.robot.X[i,0];

    h.robot.a1[i]:=h.robot.Q[i,0]*h.robot.R[i,0];
    h.robot.a2[i]:=h.robot.Q[i,1]*h.robot.R[i,1];
    h.robot.a3[i]:=h.robot.Q[i,2]*h.robot.R[i,2];
    h.robot.a5[i]:=h.robot.K[i,1]*h.robot.W[i,1];
    h.robot.a6[i]:=h.robot.K[i,2]*h.robot.W[i,2];
    h.robot.W[i,0]:=(h.robot.a1[i]+h.robot.a2[i]+h.robot.a3[i]-h.robot.a5[i]-h.robot.a6[i])/h.robot.K[i,0];

    h.robot.b1[i]:= h.robot.H[i,0]*h.robot.Y[i,0];
    h.robot.b2[i]:= h.robot.H[i,1]*h.robot.Y[i,1];
    h.robot.b3[i]:= h.robot.H[i,2]*h.robot.Y[i,2];
    h.robot.b5[i]:= h.robot.K[i,1]*h.robot.X[i,1];
    h.robot.b6[i]:= h.robot.K[i,2]*h.robot.X[i,2];
    h.robot.X[i,0]:=(h.robot.b1[i]+h.robot.b2[i]+h.robot.b3[i]-h.robot.b5[i]-h.robot.b6[i])/h.robot.K[i,0];

    (* Control law *)
    h.robot.qddD0[i]:=h.robot.W[i,0]-h.robot.X[i,0];
  END;

  (* Corresponding control torque Two Degree of Freedom Control*)

  FOR i := 0 TO h.robot.nq-1 DO
    h.robot.tauD0[i] := h.robot.InerMatrix[i,i]*h.robot.qddD0[i];
    IF (h.robot.securityState.no = noErr) THEN
      IF h.robot.lugre THEN
        h.robot.analogOut[i].Write(SHORT(h.robot.tauD0[i]+h.robot.gravidade[i]+h.robot.Cori[i]+h.robot.tauFF[i]+h.robot.tauFrict[i]));
        (* Total torque to the actuator, if no error, is FB + FF + Friction *)
      ELSEIF "h.robot.lugre THEN
        h.robot.analogOut[i].Write(SHORT(h.robot.tauD0[i]+h.robot.tauFF[i]+h.robot.gravidade[i]+h.robot.Cori[i]));
      END;
    ELSE
      h.robot.qd[i] := 0.0;
      h.robot.qdd[i] := 0.0;
      h.robot.qact[i] := h.robot.q[i];
      h.robot.qdact[i] := 0.0;
      h.robot.qobs[i] := h.robot.q[i];
      h.robot.qdobs[i] := 0.0;
      h.robot.qe[i] := 0.0;
      h.robot.qddD0[i] := 0.0;
      h.robot.tauD0[i] := 0.0;
      h.robot.tauFF[i] := 0.0;
      h.robot.tauFrict[i] := 0.0;
      h.robot.tauFrictObs[i]:=0.0;
      h.robot.analogOut[i].Write(0.0);
    END;
  END;

  (* Update actual cartesian position *)
  h.robot.DirKinematics(h.robot.qact`, h.robot.dx`, h.robot.xact`);

  (* Update actual operational ("cartesian") velocity of the tool (TCP system) - in base coordinates *)
  h.robot.OperationalSpeed(h.robot.qact`, h.robot.qdact`, h.robot.dx`, h.robot.xdact`);

END Run;

PROCEDURE (h: ControlFuzzyHd1) Run(); VAR
  i, PDiscretos: LONGINT;
  g1, g2, q: LONGREAL;
  done : BOOLEAN;
  (*Ge,Gde,Gc: ARRAY 4 OF LONGREAL; *)
  Se,Sde,escde,esce,Sc,escc: LONGREAL;
  ut,vel: ARRAY 4 OF ARRAY 4 OF LONGREAL;

  (***** Variaveis do Universo *****)
  E, DE,C: ARRAY 7 OF LONGREAL;

  (***** Resultados de Inferencias Intermediarias *****)

```

```

R : ARRAY 7 OF LONGREAL;

(***** Variaveis de Pertinencia *****)
NGE,NPE,ZE,PPE,PGE: ARRAY 7 OF LONGREAL;
NGDE,NPDE,ZDE,PPDE,PGDE : ARRAY 7 OF LONGREAL ;
NGC,NPC,ZC,PPC,PGC: ARRAY 7 OF LONGREAL ;

(***** Variavel de Pertinencia Zero *****)
Zero: ARRAY 7 OF LONGREAL;

(***** Pertinencias para o Erro e a Derivada do Erro *****)
ENG,ENP,EZE,EPP,EPG: LONGREAL;
DENG,DENP,DEZDE,DEPP,DEPG: LONGREAL;

(***** Resultados acumulados de Inferencias *****)
O: ARRAY 7 OF LONGREAL ;

(***** Pertinencias para o controle C *****)
CNG,CNP,CZC,CPP,CPG : LONGREAL;

(***** Controle Crisp *****)
Co: ARRAY 4 OF LONGREAL;

BEGIN
(* Update force sensor measurements *)
done:= JR3.ReadForce(h.robot.force`);

(*
Ge[0]:=3.0; Ge[1]:=3.0; Ge[2]:=3.0; Ge[3]:=3.0;
Gde[0]:=3.0; Gde[1]:=3.0; Gde[2]:=3.0; Gde[3]:=3.0;
Gc[0]:=2.0; Gc[1]:=10.0; Gc[2]:=5.0; Gc[3]:=5.0;
*)

(* Gc[0]:=300.0; Gc[1]:=150.0; Gc[2]:=800.0; Gc[3]:=15.0; *)

Sde:=1.0; Se:=1.0;
(*esc_de:=Sde/3.0 ; esc_e:=Se/3.0;*)
Sc:=1.0;
(*esc_c:=Sc/3.0;*)

FOR i:=0 TO 3 DO
vel[i,0]:=0.0;
vel[i,1]:=0.0;
vel[i,2]:=0.0;
vel[i,3]:=0.0;
ut[i,0]:=0.0;
ut[i,1]:=0.0;
ut[i,2]:=0.0;
ut[i,3]:=0.0;
END;

(***** Definicao dos conjuntos utilizados na Logica Fuzzy *****)

(** Definicao dos conjuntos utilizados na Logica Fuzzy **)

(** Conjunto do Erro (E) **)
E[0]:=-Se; E[1]:=-0.5;E[2]:=0.0 ; E[3]:=0.5; E[4]:=Se;
(*E[6]:=se-esc_e; E[7]:=Se;*)

NGE[0]:=1.0; NGE[1]:=0.0; NGE[2]:=0.0; NGE[3]:=0.0; NGE[4]:=0.0; (*NGE[6]:=0;NGE[7]:=0;*)
NPE[0]:=0.0; NPE[1]:=1.0; NPE[2]:=0.0; NPE[3]:=0.0; NPE[4]:=0.0; (*NPE[6]:=0;NPE[7]:=0;*)
ZE[0]:=0.0; ZE[1]:=0.0; ZE[2]:=1.0; ZE[3]:=0.0; ZE[4]:=0.0; (*ZE[6]:=0;ZE[7]:=0;*)
PPE[0]:=0.0; PPE[1]:=0.0; PPE[2]:=0.0; PPE[3]:=1.0; PPE[4]:=0.0; (*PPE[6]:=0;PPE[7]:=0;*)
PGE[0]:=0.0; PGE[1]:=0.0; PGE[2]:=0.0; PGE[3]:=0.0; PGE[4]:=1.0; (*PGE[6]:=0;PGE[7]:=1;*)

(** Conjunto da Derivada do Erro (E) **)
DE[0]:=-Sde; DE[1]:=-0.5; DE[2]:=0.0 ; DE[3]:=0.5; DE[4]:=Sde;
(*DE[6]:=se-2*esc_de; DE[7]:=Sde;*)

NGDE[0]:=1.0; NGDE[1]:=0.0; NGDE[2]:=0.0; NGDE[3]:=0.0; NGDE[4]:=0.0; (*NGDE[6]:=0;NGDE[7]:=0;*)
NPDE[0]:=0.0; NPDE[1]:=1.0; NPDE[2]:=0.0; NPDE[3]:=0.0; NPDE[4]:=0.0; (*NPDE[6]:=0;NPDE[7]:=0;*)
ZDE[0]:=0.0; ZDE[1]:=0.0; ZDE[2]:=1.0; ZDE[3]:=0.0; ZDE[4]:=0.0; (*ZDE[6]:=0;ZDE[7]:=0;*)
PPDE[0]:=0.0; PPDE[1]:=0.0; PPDE[2]:=0.0; PPDE[3]:=1.0; PPDE[4]:=0.0; (*PPDE[6]:=0;PPDE[7]:=0;*)
PGDE[0]:=0.0; PGDE[1]:=0.0; PGDE[2]:=0.0; PGDE[3]:=0.0; PGDE[4]:=1.0; (*PGDE[6]:=0.7;PGDE[7]:=1;*)

(** Conjunto do Controle (C) **)
C[0]:=-Sc; C[1]:=-0.6; C[2]:=0.0; C[3]:=0.6; C[4]:=Sc;
(*C[6]:=(Sc-esc_c); C[7]:=Sc;*)

NGC[0]:=1.0; NGC[1]:=0.0; NGC[2]:=0.0; NGC[3]:=0.0; NGC[4]:=0.0; (*NGC[6]:=0;NGC[7]:=0;*)
NPC[0]:=0.0; NPC[1]:=1.0; NPC[2]:=0.0; NPC[3]:=0.0; NPC[4]:=0.0; (*NPC[6]:=0;NPC[7]:=0;*)
ZC[0]:=0.0; ZC[1]:=0.0; ZC[2]:=1.0; ZC[3]:=0.0; ZC[4]:=0.0; (*ZC[6]:=0;ZC[7]:=0;*)
PPC[0]:=0.0; PPC[1]:=0.0; PPC[2]:=0.0; PPC[3]:=1.0; PPC[4]:=0.0; (*PPC[6]:=0;PPC[7]:=0;*)
PGC[0]:=0.0; PGC[1]:=0.0; PGC[2]:=0.0; PGC[3]:=0.0; PGC[4]:=1.0; (*PPC[6]:=0.7;PGC[7]:=1;*)

(**** Função de Pertinência Zero *****)
Zero[0]:=0.0; Zero[1]:=0.0; Zero[2]:=0.0; Zero[3]:=0.0; Zero[4]:=0.0;
(*Zero[6]:=0;Zero[7]:=0;*)

(* Algoritmo de Controle*)

FOR i := 0 TO h.robot.nq-1 DO
q := h.robot.counter[i].Read();
h.robot.qdact[i] := (1/5)*(vel[i,0]+vel[i,1]+vel[i,2]+vel[i,3])+(q-h.robot.qact[i])/samplingTimeFuzzy;
(*h.robot.qdact[i] := (q-h.robot.qact[i])/samplingTimeFuzzy; *)
vel[i,3] := vel[i,2];
vel[i,2] := vel[i,1];
vel[i,1] := vel[i,0];
vel[i,0] := h.robot.qdact[i];

h.robot.qact[i] := q;
h.robot.qe[i] := h.robot.q[i]-h.robot.qact[i];
h.robot.qde[i] := h.robot.qd[i]-h.robot.qdact[i];
h.robot.qe[i] := h.robot.Ge[i]*h.robot.qe[i];
h.robot.qde[i] := h.robot.Gde[i]*h.robot.qde[i];
PDiscretos:=4;

```

```

MathFuzzy.Pertinencia(h.robot.qe[i],E,NGE,PDiscretos, ENG);
MathFuzzy.Pertinencia(h.robot.qe[i],E,NPE,PDiscretos, ENP);
MathFuzzy.Pertinencia(h.robot.qe[i],E,ZE,PDiscretos, EZE);
MathFuzzy.Pertinencia(h.robot.qe[i],E,PPE,PDiscretos, EPP);
MathFuzzy.Pertinencia(h.robot.qe[i],E,PGE,PDiscretos,EPG);

PDiscretos:=4;
MathFuzzy.Pertinencia(h.robot.qde[i],DE,NGDE,PDiscretos, DENG);
MathFuzzy.Pertinencia(h.robot.qde[i],DE,NPDE,PDiscretos, DENP);
MathFuzzy.Pertinencia(h.robot.qde[i],DE,ZDE,PDiscretos, DEZDE);
MathFuzzy.Pertinencia(h.robot.qde[i],DE,PPDE,PDiscretos, DEPP);
MathFuzzy.Pertinencia(h.robot.qde[i],DE,PGDE,PDiscretos,DEPG);

CNG := 0.0;
CNP := 0.0;
CZC := 0.0;
CPP := 0.0;
CPG := 0.0;

(**** AVALIACAO DAS REGRAS *****)

(* {1} {IF (E = PG) AND (DE=NG) THEN C = ZERO} *)
CZC := MathFuzzy.Max(EPG*DENG,CZC);

(* {2} {IF (E = PP) AND (DE=NG) THEN C = PG} *)
CPG := MathFuzzy.Max(EPP*DENG, CPG);

(* {3} {IF (E = ZE) AND (DE=NG) THEN C = PG} *)
CPG := MathFuzzy.Max(EZE*DENG, CPG);

(* {4} {IF (E = NP) AND (DE=NG) THEN C = PG} *)
CPG := MathFuzzy.Max(ENP*DENG, CPG);

(* {5} {IF (E = NG) AND (DE=NG) THEN C = PG} *)
CPG := MathFuzzy.Max(ENG*DENG, CPG);

(* {6} {IF (E = PG) AND (DE=NP) THEN C = NG} *)
CNG := MathFuzzy.Max(DENP*EPG, CNG);

(* {7} {IF (E = PP) AND (DE=NP) THEN C = ZERO} *)
CZC := MathFuzzy.Max(EPP*DENP, CZC);

(* {8} {IF (E = ZE) AND (DE=NP) THEN C = PP} *)
CPP := MathFuzzy.Max(EZE*DENP, CPP);

(* {9} {IF (E = NP) AND (DE=NP) THEN C = PP} *)
CPP := MathFuzzy.Max(ENP*DENP, CPP);

(* {10} {IF (E = NG) AND (DE=NP) THEN C = PG} *)
CPG := MathFuzzy.Max(ENG*DENP, CPG);

(* {11} {IF (E = PG) AND (DE=ZDE) THEN C = NG} *)
CNG := MathFuzzy.Max(EPG*DEZDE, CNG);

(* {12} {IF (E = PP) AND (DE = ZDE) THEN C = NP} *)
CNP := MathFuzzy.Max(EPP*DEZDE, CNP);

(* {13} {IF (E = ZE) AND (DE = ZE) THEN C = ZERO} *)
CZC:= MathFuzzy.Max(EZE*DEZDE, CZC);

(* {14} {IF (E = NP) AND (DE = ZE) THEN C = PP} *)
CPP := MathFuzzy.Max(ENP*DEZDE, CPP);

(* {15} {IF (E = NG) AND (DE = ZE) THEN C = PG} *)
CPG := MathFuzzy.Max(ENG*DEZDE, CPG);

(* {16} {IF (E = PG) AND (DE = PP) THEN C = NG} *)
CNG := MathFuzzy.Max(EPG*DEPP, CNG);

(* {17} {IF (E = PP) AND (DE = PP) THEN C = NP} *)
CNP := MathFuzzy.Max(DEPP*EPP, CNP);

(* {18} {IF (E = ZE) AND (DE = PP) THEN C = NP} *)
CNP := MathFuzzy.Max(DEPP*EZE, CNP);

(* {19} {IF (E = NP) AND (DE = PP) THEN C =ZERO} *)
CZC := MathFuzzy.Max(ENP*DEPP, CZC);

(* {20} {IF (E = NG) AND (DE = PP) THEN C = PG} *)
CPG := MathFuzzy.Max(ENG*DEPP, CPG);

(* {21} {IF (E = PG) AND (DE = PG) THEN C = NG} *)
CNG := MathFuzzy.Max(EPG*DEPG, CNG);

(* {22} {IF (E = PP) AND (DE = PG) THEN C = NG} *)
CNG := MathFuzzy.Max(EPP*DEPG, CNG);

(* {23} {IF (E = ZE) AND (DE = PG) THEN C = NG} *)
CNG := MathFuzzy.Max(EZE*DEPG, CNG);

(* {24} {IF (E = NP) AND (DE = PG) THEN C = NG} *)
CNG := MathFuzzy.Max(ENP*DEPG, CNG);

(* {25} {IF (E = NG) AND (DE = PG) THEN C = ZC} *)
CZC := MathFuzzy.Max(ENG*DEPG, CZC);

PDiscretos:=4;

MathFuzzy.Inferencia(CNG,NGC,PDiscretos,R);
MathFuzzy.Maximo(R,Zero,PDiscretos,0);

MathFuzzy.Inferencia(CNP,NPC,PDiscretos,R);
MathFuzzy.Maximo(R,0,PDiscretos,0);

MathFuzzy.Inferencia(CZC,ZC,PDiscretos,R);
MathFuzzy.Maximo(R,0,PDiscretos,0);

```

```

MathFuzzy.Inferencia(CPP,PPC,PDiscretos,R);
MathFuzzy.Maximo(R,0,PDiscretos,0);

MathFuzzy.Inferencia(CPG,PGC,PDiscretos,R);
MathFuzzy.Maximo(R,0,PDiscretos,0);

MathFuzzy.DeFuzzyCG(G,0,PDiscretos,Co[i]);

Co[i]:= Co[i]-2.1*ut[i,0]+1.47*ut[i,1]-0.343*ut[i,2];

ut[i,2]:=ut[i,1];
ut[i,1]:=ut[i,0];
ut[i,0]:=Co[i];
h.robot.qddFuzzy[i]:=h.robot.Gc[i]*Co[i];
END;

(* Corresponding control torque Variable Structure and FeedForward
Controller*) FOR i := 0 TO h.robot.nq-1 DO
  h.robot.TauFuzzy[i] := h.robot.m[i]*h.robot.qddFuzzy[i];
  IF (h.robot.securityState.no = noErr) THEN
    IF h.robot.lugre THEN
      h.robot.analogOut[i].Write(SHORT(h.robot.TauFuzzy[i]+h.robot.tauFF[i]+h.robot.tauFrict[i]));
      (* Total torque to the actuator, if no error, is FB + Friction*)
    ELSIF ~h.robot.lugre THEN
      h.robot.analogOut[i].Write(SHORT(h.robot.TauFuzzy[i]+h.robot.tauFF[i]));
      (* Total torque to the actuator *)
    END;
  ELSE
    h.robot.qd[i] := 0.0;
    h.robot.qdd[i] := 0.0;
    h.robot.qact[i] := h.robot.q[i];
    h.robot.qdact[i] := 0.0;
    h.robot.qobs[i] := h.robot.q[i];
    h.robot.qdobs[i] := 0.0;
    h.robot.qe[i] := 0.0;
    h.robot.qdVS[i] := 0.0;
    h.robot.qddFuzzy[i]:=0.0;
    h.robot.tauVS[i] := 0.0;
    h.robot.TauFuzzy[i] := 0.0;
    h.robot.tauFF[i] := 0.0;
    h.robot.tauFrict[i] := 0.0;
    h.robot.tauFrictObs[i]:=0.0;
    h.robot.analogOut[i].Write(0.0);
  END;
END;

(* Update actual cartesian position *)
h.robot.DirKinematics(h.robot.qact, h.robot.dx, h.robot.xact);

(* Update actual operational ("cartesian") velocity of the tool (TCP system) - in base coordinates *)
h.robot.OperationalSpeed(h.robot.qact, h.robot.qdact, h.robot.dx, h.robot.xdact);

END Run;

PROCEDURE (h: ControlForcaHdl) Run(); VAR
  g1, g2: LONGREAL;
  i, PDiscretos: LONGINT;
  q, fx2, fy2, forceMod: LONGREAL;
  ni, tau : ARRAY 2 OF LONGREAL;
  forceSensor, forceFiltered, pos: ARRAY 4 OF LONGREAL;
  vel: ARRAY 4 OF ARRAY 4 OF LONGREAL;
  done : BOOLEAN;
  Se,Sde,escde,esce,Sc,esc: LONGREAL;
  ut: ARRAY 4 OF ARRAY 4 OF LONGREAL;

  (***** Variaveis do Universo *****)
  E, DE,C: ARRAY 7 OF LONGREAL;

  (***** Resultados de Inferencias Intermediarias *****)
  R : ARRAY 7 OF LONGREAL;

  (***** Variaveis de Pertinencia *****)
  NGE,NPE,ZE,PPE,PGE: ARRAY 7 OF LONGREAL;
  NGDE,NPDE,ZDE,PPDE,PGDE : ARRAY 7 OF LONGREAL;
  NGC,NPC,ZC,PPC,PGC: ARRAY 7 OF LONGREAL ;

  (***** Variavel de Pertinencia Zero *****)
  Zero: ARRAY 7 OF LONGREAL;

  (***** Pertinencias para o Erro e a Derivada do Erro
  *****)
  ENG,ENP,EZE,EPP,EPG: LONGREAL;
  DENG,DENP,DEZDE,DEPP,DEPG: LONGREAL;
  (***** Resultados acumulados de Inferencias *****)
  O: ARRAY 7 OF LONGREAL ;

  (***** Pertinencias para o controle C *****)
  CNG,CNP,CZC,CPP,CPG : LONGREAL;

  (***** Controle Crisp *****)
  Co: ARRAY 4 OF LONGREAL;

BEGIN

  FOR i:=0 TO 3 DO
    vel[i,0]:=0.0;
    vel[i,1]:=0.0;
    vel[i,2]:=0.0;
    vel[i,3]:=0.0;
    ut[i,0]:=0.0;
    ut[i,1]:=0.0;
    ut[i,2]:=0.0;
    ut[i,3]:=0.0;
  END;

  Sde:=1.0; Se:=1.0;
  (*esc_de:=Sde/3.0 ; esc_e:=Se/3.0;*)
  Sc:=1.0;

```

```

(*esc_c:=Sc/3.0;*)
(***** Definição dos conjuntos utilizados na Logica Fuzzy
*****)

(** Definição dos conjuntos utilizados na Logica Fuzzy **)

(** Conjunto do Erro (E) **)
E[0]:=-Se; E[1]:=-0.5;E[2]:=0.0 ; E[3]:=0.5; E[4]:=Se;
(*E[6]:=se-esc_e; E[7]:=Se;*)

NGE[0]:=1.0; NGE[1]:=0.0; NGE[2]:=0.0; NGE[3]:=0.0; NGE[4]:=0.0; (*NGE[6]:=0;NGE[7]:=0;*)
NPE[0]:=0.0; NPE[1]:=1.0; NPE[2]:=0.0; NPE[3]:=0.0; NPE[4]:=0.0; (*NPE[6]:=0;NPE[7]:=0;*)
ZE[0]:=0.0; ZE[1]:=0.0; ZE[2]:=1.0; ZE[3]:=0.0; ZE[4]:=0.0; (*ZE[6]:=0;ZE[7]:=0;*)
PPE[0]:=0.0; PPE[1]:=0.0; PPE[2]:=0.0; PPE[3]:=1.0; PPE[4]:=0.0; (*PPE[6]:=0;PPE[7]:=0;*)
PGE[0]:=0.0; PGE[1]:=0.0; PGE[2]:=0.0; PGE[3]:=0.0; PGE[4]:=1.0; (*PGE[6]:=0;PGE[7]:=1;*)

(** Conjunto da Derivada do Erro (E) **)
DE[0]:=-Sde; DE[1]:=-0.5; DE[2]:=0.0 ; DE[3]:=0.5; DE[4]:=Sde;
(*DE[6]:=se-2*esc_de; DE[7]:=Sde;*)

NGDE[0]:=1.0; NGDE[1]:=0.0; NGDE[2]:=0.0; NGDE[3]:=0.0; NGDE[4]:=0.0; (*NGDE[6]:=0;NGDE[7]:=0;*)
NPDE[0]:=0.0; NPDE[1]:=1.0; NPDE[2]:=0.0; NPDE[3]:=0.0; NPDE[4]:=0.0; (*NPDE[6]:=0;NPDE[7]:=0;*)
ZDE[0]:=0.0; ZDE[1]:=0.0; ZDE[2]:=1.0; ZDE[3]:=0.0; ZDE[4]:=0.0; (*ZDE[6]:=0;ZDE[7]:=0;*)
PPDE[0]:=0.0; PPDE[1]:=0.0; PPDE[2]:=0.0; PPDE[3]:=1.0; PPDE[4]:=0.0; (*PPDE[6]:=0;PPDE[7]:=0;*)
PGDE[0]:=0.0; PGDE[1]:=0.0; PGDE[2]:=0.0; PGDE[3]:=0.0; PGDE[4]:=1.0; (*PGDE[6]:=0.7;PGDE[7]:=1;*)

(** Conjunto do Controle (C) **)
C[0]:=-Sc; C[1]:=-0.6; C[2]:=0.0; C[3]:=0.6; C[4]:=Sc;
(*C[6]:=(Sc-esc_c); C[7]:=Sc;*)

NGC[0]:=1.0; NGC[1]:=0.0; NGC[2]:=0.0; NGC[3]:=0.0; NGC[4]:=0.0; (*NGC[6]:=0;NGC[7]:=0;*)
NPC[0]:=0.0; NPC[1]:=1.0; NPC[2]:=0.0; NPC[3]:=0.0; NPC[4]:=0.0; (*NPC[6]:=0;NPC[7]:=0;*)
ZC[0]:=0.0; ZC[1]:=0.0; ZC[2]:=1.0; ZC[3]:=0.0; ZC[4]:=0.0; (*ZC[6]:=0;ZC[7]:=0;*)
PPC[0]:=0.0; PPC[1]:=0.0; PPC[2]:=0.0; PPC[3]:=1.0; PPC[4]:=0.0; (*PPC[6]:=0;PPC[7]:=0;*)
PGC[0]:=0.0; PGC[1]:=0.0; PGC[2]:=0.0; PGC[3]:=0.0; PGC[4]:=1.0; (*PPC[6]:=0.7;PGC[7]:=1;*)

(**** Função de Pertinência Zero ****)
Zero[0]:=0.0; Zero[1]:=0.0; Zero[2]:=0.0; Zero[3]:=0.0; Zero[4]:=0.0;
(*Zero[6]:=0;Zero[7]:=0;*)

(* Updates joint positions, velocities, observer and position error *)
FOR i := 0 TO h.robot.nq-1 DO
  q := h.robot.counter[i].Read();
  h.robot.qdact[i] := (1/5)*(vel[i,0]+vel[i,1]+vel[i,2]+vel[i,3])+(q-h.robot.qact[i])/samplingTimeForca;
  (*h.robot.qdact[i] := (q-h.robot.qact[i])/samplingTimeForca; *)
  vel[i,3] := vel[i,2];
  vel[i,2] := vel[i,1];
  vel[i,1] := vel[i,0];
  vel[i,0] := h.robot.qdact[i];

  h.robot.qact[i] := q;
  h.robot.q[i] := q;
  h.robot.qe[i] := h.robot.q[i]-h.robot.qact[i];
  (*pos[i] := h.robot.omega[i]*h.robot.omega[i]*(h.robot.q[i]-h.robot.qact[i])+2.0*h.robot.omega[i]*(h.robot.qd[i]-h.robot.qdact[i]);*)
END;

done:= JR3.ReadForce(h.robot.force^);

FOR i:= 0 TO h.robot.nq-1 DO
  forceSensor[i]:=h.robot.force[i];
END;

(* TO DO: Filter the forces read by the sensor *)
Filters.ForceFilter(forceSensor, forceFiltered);
(* h.robot.forceFiltered := forceFiltered; *)

(* Forces in TCP coordinates; also equivalent to change the sensor's coordinate system to right handed *)
(* h.robot.ForceSensToTCP(forceSensor, h.robot.forceTCP^); *)
h.robot.ForceSensToTCP(forceFiltered, h.robot.forceTCP^);

(* Forces in base coordinate system *)
h.robot.ForceTCPToBase(h.robot.qact^, h.robot.forceTCP^, h.robot.forceBase^);

FOR i:= 0 TO h.robot.nq-1 DO
  h.robot.K1[i] := h.robot.Kp[i]+(h.robot.Kp[i]*samplingTimeForca)/h.robot.Ti[i]+(h.robot.Kp[i]*h.robot.Td[i])/samplingTimeForca;
  h.robot.K2[i] := -h.robot.Kp[i]-2*h.robot.Kp[i]*h.robot.Td[i]/samplingTimeForca;
  (*h.robot.K3[i] := (h.robot.Kp[i]*h.robot.Td[i])/samplingTimeForca; *)
END;

(* PI and VSC Control*)

FOR i := 0 TO h.robot.nq-1 DO
  h.robot.e[i] := h.robot.forcaRef[i]-h.robot.forceBase[i]; (* Force Error *)
  h.robot.aux1[i] := h.robot.u[i]+h.robot.K1[i]*h.robot.e[i];
  h.robot.aux2[i] := h.robot.K2[i]*h.robot.e[i];
  h.robot.u[i] := h.robot.aux1[i]+h.robot.aux2[i];

  h.robot.tauCartFB[i] := h.robot.forcaRef[i]-h.robot.u[i];

  (*h.robot.tauCartFB[i] := h.robot.forcaRef[i]-h.robot.cfor[i]*MathLib.tanh(h.robot.u[i]/h.robot.epsilon[i])-h.robot.kfor[i]*h.robot.u[i];*)

  h.robot.e1[i] := h.robot.e[i];
  h.robot.u1[i] := h.robot.u[i];
END;

(* Fuzzy Control*)
(*)

FOR i := 0 TO h.robot.nq-1 DO
  h.robot.e1[i] := h.robot.forcaRef[i]-h.robot.forceBase[i]; (* Force Error *)
  h.robot.e2[i] := h.robot.forceBase[i]-h.robot.e1[i]/samplingTimeForca;
  h.robot.e1[i] := h.robot.forceBase[i];

  PDiscretos:=4;

  MathFuzzy.Pertinencia(h.robot.e1[i],E,NGE,PDiscretos, ENG);
  MathFuzzy.Pertinencia(h.robot.e1[i],E,NPE,PDiscretos, ENP);
  MathFuzzy.Pertinencia(h.robot.e1[i],E,ZE,PDiscretos, EZE);
  MathFuzzy.Pertinencia(h.robot.e1[i],E,PPE,PDiscretos, EPP);

```

```

MathFuzzy.Pertinencia(h.robot.e[i],E,PGE,PDiscretos,EPG);

PDiscretos:=4;

MathFuzzy.Pertinencia(h.robot.e2[i],DE,NGDE,PDiscretos, DENG);
MathFuzzy.Pertinencia(h.robot.e2[i],DE,NPDE,PDiscretos, DENP);
MathFuzzy.Pertinencia(h.robot.e2[i],DE,ZDE,PDiscretos, DEZDE);
MathFuzzy.Pertinencia(h.robot.e2[i],DE,PPDE,PDiscretos, DEPP);
MathFuzzy.Pertinencia(h.robot.e2[i],DE,PGDE,PDiscretos,DEPG);

CNG := 0.0;
CNP := 0.0;
CZC := 0.0;
CPP := 0.0;
CPG := 0.0;

(**** AVALIACAO DAS REGRAS *****)

(* {1} {IF (E = PG) AND (DE=NG) THEN C = ZERO} *)
CZC := MathFuzzy.Max(EPG*DENG,CZC);

(* {2} {IF (E = PP) AND (DE=NG) THEN C = PG} *)
CPG := MathFuzzy.Max( EPP*DENG, CPG);

(* {3} {IF (E = ZE) AND (DE=NG) THEN C = PG} *)
CPG := MathFuzzy.Max( EZE*DENG, CPG);

(* {4} {IF (E = NP) AND (DE=NG) THEN C = PG} *)
CPG := MathFuzzy.Max( ENP*DENG, CPG);

(* {5} {IF (E = NG) AND (DE=NG) THEN C = PG} *)
CPG := MathFuzzy.Max( ENG*DENG, CPG);

(* {6} {IF (E = PG) AND (DE=NP) THEN C = NG} *)
CNG := MathFuzzy.Max( DENP*EPG, CNG);

(* {7} {IF (E = PP) AND (DE=NP) THEN C = ZERO} *)
CZC := MathFuzzy.Max( EPP*DENP, CZC);

(* {8} {IF (E = ZE) AND (DE=NP) THEN C = PP} *)
CPP := MathFuzzy.Max( EZE*DENP, CPP);

(* {9} {IF (E = NP) AND (DE=NP) THEN C = PP} *)
CPP := MathFuzzy.Max( ENP*DENP, CPP);

(* {10} {IF (E = NG) AND (DE=NP) THEN C = PG} *)
CPG := MathFuzzy.Max( ENG*DENP, CPG);

(* {11} {IF (E = PG) AND (DE=ZDE) THEN C = NG} *)
CNG := MathFuzzy.Max( EPG*DEZDE, CNG);

(* {12} {IF (E = PP) AND (DE = ZDE) THEN C = NP} *)
CNP := MathFuzzy.Max( EPP*DEZDE, CNP);

(* {13} {IF (E = ZE) AND (DE = ZE) THEN C = ZERO} *)
CZC := MathFuzzy.Max( EZE*DEZDE, CZC);

(* {14} {IF (E = NP) AND (DE = ZE) THEN C = PP} *)
CPP := MathFuzzy.Max( ENP*DEZDE, CPP);

(* {15} {IF (E = NG) AND (DE = ZE) THEN C = PG} *)
CPG := MathFuzzy.Max( ENG*DEZDE, CPG);

(* {16} {IF (E = PG) AND (DE = PP) THEN C = NG} *)
CNG := MathFuzzy.Max( EPG*DEPP, CNG);

(* {17} {IF (E = PP) AND (DE = PP) THEN C = NP} *)
CNP := MathFuzzy.Max( DEPP*EPP, CNP);

(* {18} {IF (E = ZE) AND (DE = PP) THEN C = NP} *)
CNP := MathFuzzy.Max( DEPP*EZE, CNP);

(* {19} {IF (E = NP) AND (DE = PP) THEN C =ZERO} *)
CZC := MathFuzzy.Max( ENP*DEPP, CZC);

(* {20} {IF (E = NG) AND (DE = PP) THEN C = PG} *)
CPG := MathFuzzy.Max( ENG*DEPP, CPG);

(* {21} {IF (E = PG) AND (DE = PG) THEN C = NG} *)
CNG := MathFuzzy.Max( EPG*DEPG, CNG);

(* {22} {IF (E = PP) AND (DE = PG) THEN C = NG} *)
CNG := MathFuzzy.Max( EPP*DEPG, CNG);

(* {23} {IF (E = ZE) AND (DE = PG) THEN C = NG} *)
CNG := MathFuzzy.Max( EZE*DEPG, CNG);

(* {24} {IF (E = NP) AND (DE = PG) THEN C = NG} *)
CNG := MathFuzzy.Max( ENP*DEPG, CNG);

(* {25} {IF (E = NG) AND (DE = PG) THEN C = ZC} *)
CZC := MathFuzzy.Max( ENG*DEPG, CZC);

PDiscretos:=4;

MathFuzzy.Inferencia(CNG,NGC,PDiscretos,R);
MathFuzzy.Maximo(R,Zero,PDiscretos,0);

MathFuzzy.Inferencia(CNP,NPC,PDiscretos,R);
MathFuzzy.Maximo(R,0,PDiscretos,0);

MathFuzzy.Inferencia(CZC,ZC,PDiscretos,R);
MathFuzzy.Maximo(R,0,PDiscretos,0);

MathFuzzy.Inferencia(CPP,PPC,PDiscretos,R);
MathFuzzy.Maximo(R,0,PDiscretos,0);

MathFuzzy.Inferencia(CPG,PGC,PDiscretos,R);
MathFuzzy.Maximo(R,0,PDiscretos,0);

```

```

MathFuzzy.DeFuzzyCG(G,0,PDiscretos,Co[i]);

Co[i]:= Co[i]-2.1*ut[i,0]+1.47*ut[i,1]-0.343*ut[i,2];

ut[i,2]:=ut[i,1];
ut[i,1]:=ut[i,0];
ut[i,0]:=Co[i];
h.robot.tauCartFB[i]:=h.robot.Gc[i]*Co[i];

END;
*)

(* Control torque in joint space *)
h.robot.ForceToTau(h.robot.qact`, h.robot.dx`, h.robot.tauCartFB`, h.robot.tauFB`);

(* Total torque is the sum of feedback and feedforward control torques written to the actuators if no errors are detected *)
FOR i := 0 TO h.robot.nq-1 DO
  IF (h.robot.securityState.no = noErr) THEN
    IF h.robot.lugre THEN
      h.robot.analogOut[i].Write(SHORT(h.robot.tauFB[i]+h.robot.tauFF[i]+h.robot.tauFrict[i])); (* Total Torque+ Frict *)
    ELSEIF `h.robot.lugre THEN
      h.robot.analogOut[i].Write(SHORT(h.robot.tauFB[i]+h.robot.tauFF[i]));
    END;
  ELSE
    h.robot.qd[i] := 0.0;
    h.robot.qdd[i] := 0.0;
    h.robot.qact[i] := h.robot.q[i];
    h.robot.qdact[i] := 0.0;
    h.robot.qobs[i] := h.robot.q[i];
    h.robot.qdobs[i] := 0.0;
    h.robot.qe[i] := 0.0;
    h.robot.qddFB[i] := 0.0;
    h.robot.tauFB[i] := 0.0;
    h.robot.tauFF[i] := 0.0;
    h.robot.tauFrict[i] := 0.0;
    h.robot.analogOut[i].Write(0.0);
  END;
END;

(* Update actual cartesian position *)
h.robot.DirKinematics(h.robot.qact`, h.robot.dx`, h.robot.xact`);

(* Update actual operational ("cartesian") velocity of the tool (TCP system) - in base coordinates *)
h.robot.OperationalSpeed(h.robot.qact`, h.robot.qdact`, h.robot.dx`, h.robot.xdact`);

END Run;

PROCEDURE (h: ControlFFHdl) Run(); BEGIN
  h.robot.InvDynamics(h.robot.x`, h.robot.xd`, h.robot.xdd`, h.robot.dx`, h.robot.dxd`, h.robot.dxdd`, h.robot.q`, h.robot.qd`, h.robot.qdd`, h.robot.p`, h.robot.tauFF`);
  h.robot.MatrixIner(h.robot.qact`, h.robot.qdact`, h.robot.inercia`, h.robot.lc`, h.robot.masa`, h.robot.InerMatrix`);
  h.robot.Coriolis(h.robot.qact`, h.robot.qdact`, h.robot.masa`, h.robot.lc`, h.robot.Cori`);
  h.robot.LuGreComp(h.robot.qdact`,h.robot.tauFrict`);
END Run;

PROCEDURE (robot: ScaraRobot) LuGreComp*(VAR qd, tauFrict: ARRAY
OF LONGREAL); VAR
  g, z, g1, z1,qd1, aux: ARRAY 4 OF LONGREAL;
  alpha0, alpha1, alpha2, stribeck, sigma0, sigma1: ARRAY 4 OF LONGREAL;
  i: LONGINT;
  sampling: LONGREAL;
BEGIN
  sampling := Robot.samplingTimeFF;

  FOR i:=0 TO robot.nq-1 DO
    g[i] := robot.g1[i];
    z1[i] := robot.z1[i];
    qd1[i] := robot.qd1[i];

    alpha0[i]:= robot.alpha0[i];
    alpha1[i]:= robot.alpha1[i];
    alpha2[i]:= robot.alpha2[i];
    stribeck[i]:= robot.stribeck[i];
    sigma0[i]:= robot.sigma0[i];
    sigma1[i]:= robot.sigma1[i];
  END;

  FOR i:=0 TO robot.nq-1 DO
    g[i] := alpha0[i] + alpha1[i]*MathLib.exp(-MathLib.Sqr(qd[i]/stribeck[i]));
    aux[i] := qd[i]-sigma0[i]*(ABS(qd[i])/g[i])*z[i];
    z[i] := z1[i]+(sampling/2)*(qd1[i]-sigma0[i]*(ABS(qd1[i])/g1[i])*z1[i]+aux[i]);
    tauFrict[i] := sigma0[i]*z[i]+(sigma1[i]/Robot.samplingTimeFF)*(z[i]-z1[i])+alpha2[i]*qd[i];

    g1[i] := g[i];
    z1[i] := z[i];
    qd1[i] := qd[i];
  END;
END LuGreComp;

PROCEDURE MathFuzzy.
IMPORT
  MathL, MathLib;

(*
TYPE
universo = ARRAY 40 OF LONGREAL;
Procedure Pertinencia (Erro : real; var X : universo; var A : universo) : real;
Procedure Maximo ( a : universo; b : universo; var c : universo);
Function Max ( a , b : real) : real;
Procedure Inferencia ( w : real; C : universo; var O : universo);
Procedure DeFuzzyCG ( Y : universo; B : universo; var Yo : real);

```

```

Procedure DeFuzzyMM ( Y : universo; B : universo; var Yo : real);

*)

VAR
  vers*: ARRAY 32 OF CHAR;

(*****
(* A funcao Pertinencia e usada para calcular o valor
pertinente *) (* com respeito a um valor de erro de entrada
baseado no universo *) (* de discurso X e o conjunto de
pertinencia A. *) (* Variaveis da
funcao: *) (*
- Erro : valor do erro lido do processo (Set-point - VP); *)
(* - X : Universo da variavel erro (0 a 5V); *)
*) (* - A : Conjunto Fuzzy (Conjunto Fuzzy p/ a var.
linguistica); *)
(*****

PROCEDURE Pertinencia* (erro: LONGREAL; X, A: ARRAY OF LONGREAL;
PDiscretos: LONGINT; VAR pertinencia: LONGREAL);

VAR
  i : LONGINT;
  achou : BOOLEAN;

BEGIN
  achou := FALSE;
  IF (erro < X[0]) THEN
    pertinencia:= A[0];
    achou:= TRUE;
  END;
  IF (erro > X[PDiscretos]) THEN
    pertinencia:= A[PDiscretos];
    achou:= TRUE;
  END;
  IF ^achou THEN
    i:=0;
    REPEAT
      IF (erro >= X[i]) & (erro <= X[i+1]) THEN
        pertinencia:= (erro-X[i]) * ((A[i+1]-A[i])/(X[i+1]-X[i])) + A[i];
        i:=PDiscretos-1;
      END;
      i:= i + 1;
    UNTIL i=PDiscretos;
  END;
END Pertinencia;

(*****
(* Esta funcao retorna o maior valor entre dois valores
comparados *)
(*****

PROCEDURE Max*(a,b : LONGREAL) : LONGREAL; VAR max: LONGREAL;
BEGIN
  IF a > b THEN
    max:= a;
  ELSE
    max:= b;
  END;
  RETURN max;
END Max;

(*****
(* Este Procedimento compara dois valores, um do array A[ ], e
outro *) (* do array B[ ] e o maior deles e armazenador em
outro array c[ ]. *)
(*****

PROCEDURE Maximo* (A,B: ARRAY OF LONGREAL; PDiscretos:LONGINT; VAR
C : ARRAY OF LONGREAL);
VAR
  i : LONGINT;
BEGIN
  FOR i:=0 TO PDiscretos DO
    C[i]:= Max(A[i],B[i]);
  END;
END Maximo;

(*****
(* Este procedimento e usado para realizar o calculo de
inferencia *) (* baseado em Ao e Bo e C[ ]. O resultado e
armazenado em O[ ] *)
(*****

PROCEDURE Inferencia* (w : LONGREAL; C : ARRAY OF LONGREAL;
PDiscretos:LONGINT; VAR O :ARRAY OF LONGREAL);
VAR
  i : LONGINT;

BEGIN
  FOR i:=0 TO PDiscretos DO
    O[i]:= w*C[i];
  END;
END Inferencia;

(*****
(* Este procedimento e usado para de-fuzzyficar o universo Y[ ]
baseado *) (* no conjunto Fuzzy B[ ], usando o teorema do centro
da gravidade *) (* (Centroide). O resultado e armazenado em
Yo. *)
(*****

```

```

PROCEDURE DeFuzzyCG*( Y,B :ARRAY OF LONGREAL;PDiscretos:LONGINT;
VAR Yo : LONGREAL);
VAR
  yK,K,Soma1,Soma2 : LONGREAL;
  i : LONGINT;
BEGIN
  yK := 0.0;
  K := 0.0;
  FOR i:=0 TO PDiscretos-1 DO
    Soma1:= ((Y[i] + Y[i+1])/2)*((B[i] + B[i+1])/2);
    Soma2:= (B[i] + B[i+1]) / 2;
    yK:= yK + Soma1;
    K := K + Soma2;
  END;
  IF (K # 0.0) THEN
    Yo:= yK/K;
  END;
END DeFuzzyCG;

(*****
(* Este procedimento e usado para de-fuzzyficar o universo Y [ ]
baseado *) (* no conjunto Fuzzy B [ ], usando a tecnica da media
dos maximos. *) (* O resultado e armazenado em Yo.
*)
(*****

PROCEDURE DeFuzzyMM* (Y,B : ARRAY OF LONGREAL;PDiscretos:LONGINT;
VAR Yo : LONGREAL);

VAR
  cont,a,c : LONGINT;
  maximo,soma : LONGREAL;
  int : ARRAY 10 OF LONGREAL;

BEGIN
  maximo := B[0];
  int[0] := Y[0];
  cont:=0;
  FOR a:=1 TO PDiscretos DO
    IF B[a] = maximo THEN
      cont:=cont +1;
      int[cont]:=Y[a];
    END;
    IF B[a] > maximo THEN
      maximo:=B[a];
      int[0]:= Y[a];
      cont:=0;
    END;
  END;
  soma:=0;
  FOR c:=1 TO cont DO
    soma:=soma + int[c];
  END;
  Yo:= soma/cont;
END DeFuzzyMM;

(*****

BEGIN
  vers :="02/03/2002";
END MathFuzzy.

PROCEDURE tanh* (x: LONGREAL): LONGREAL; (* tanh(x) is the
hyperbolic tangent of x. All arguments are legal. *)
CONST
  P0=0.16134119023996228053D+4; P1=-0.99225929672236083313D+2; P2=-0.96437492777225469787D+0;
  Q0= 0.484023570719886868686D+4; Q1= 0.22337720718962312926D+4; Q2= 0.11274474380534949335D+3;
  ln3over2=0.54930614433405484570D0;
  BIG=19.06154747D0; (* (ln(2)+(t+1)*ln(B))/2 where t=mantissa bits, B=base *)
VAR f, t: LONGREAL;
BEGIN f:=ABS(x);
IF f>BIG THEN t:=ONE
ELSEIF f>ln3over2 THEN t:=-ONE-TWO/(MathL.exp(TWO*f)+ONE)
ELSEIF f<Limit THEN t:=f
ELSE (* approximation from "Software Manual for the Elementary Functions" *)
  t:=f*f; t:=t*(((P2+t*P1)*t+P0)/(((t+Q2)*t+Q1)+t*Q0)); t:=f+f*t
END;
IF x<ZERO THEN RETURN -t ELSE RETURN t END
END tanh;

```

Referências Bibliográficas

- Almeida, O. D. M. (2002). *Controle PID Auto-Ajustável, Inteligente e Preditivo*. Tese de Doutorado, Programa de Pós-Graduação em Engenharia Elétrica (PPGEEL), Universidade Federal de Santa Catarina, Brasil.
- Amaral, S. D. (2000). *Controle a Estrutura Variável de Robôs Manipuladores Interagindo com Ambientes Passivos*. Tese de Doutorado, Programa de Pós-Graduação em Engenharia Elétrica (PPGEEL), Universidade Federal de Santa Catarina, Brasil.
- Amaral, S. D., de Pieri, E. R., e Guenther, R. (1998). Controle a estrutura variável de robôs manipuladores em ambientes cinemáticos. *Congresso Brasileiro de Automática*, 1(12):1113–1118.
- Amaral, S. D., de Pieri, E. R., e Guenther, R. (2000a). Controle a estrutura variável de robôs manipuladores em ambientes cinemáticos. *Revista Controle e Automação*, 11(2):117–127.
- Amaral, S. D., de Pieri, E. R., e Guenther, R. (2000b). Controle a estrutura variável de robôs manipuladores em ambientes cinemáticos e/ou dinâmicos. *Congresso Brasileiro de Automática*, 1(13):348–352.
- Årmstrong, B., Dupont, P., e de Wit, C. C. (1994). A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica*, 30(7):1083–1138.
- Aström, K. J. e Häggglund, T. (1995). *Pid Controllers, Theory, Design and Tuning*. Instrument Society of America.
- Aström, K. J. e Wittenmark, B. (1995). *Adaptive Control*. Addison-Wesley Publishing.
- Aström, K. J. e Wittenmark, B. (1997). *Computer Controlled Systems*. Prentice Hall.
- Bailey, E. e Arapostathis, A. (1987). Simple sliding mode control scheme applied to robot manipulators. *International Journal Control*, 45(4):1197–1209.
- Barrientos, A., Peñin, L. F., Balaguer, C., e Aracil, R. (2000). *Fundamentos de Robótica*. MacGraw-Hill.

- Bisso, C. J. V. (1999). *Controle de Posição de Robôs Manipuladores Rígidos e com Transmissões Flexíveis Com Controladores de Dois Graus de Liberdade*. Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Elétrica (PPGEEL), Universidade Federal de Santa Catarina, Brasil.
- Brega, R. (1998). *A Real-Time Operating System Designed for Predictability and Run-Time Safety*. Technical Report, Institute of Robotics, Swiss Federal Institute of Technology, Switzerland.
- Cao, L. e Schwartz, H. M. (2000). Stick-slip friction compensation for pid position control. *Proceedings of the Control Conference, Chicago, Illinois, June 2000*, 1(1):1078–1082.
- Chiavernini, S. e Sciavicco, L. (1993). The parallel approach to force/position control of robotic manipulators. *IEEE Transactions on Robotics and Automation*, 9(4):361–373.
- Chiavernini, S., Siciliano, B., e Villani, L. (1994). Force/position regulation of compliant robot manipulators. *IEEE Transactions on Automatic Control*, 39(3):647–652.
- Dario, P., Guglielmini, E., e Carroza, M. C. (1996). Robotics for medical applications. *IEEE Robotics and Automation Magazine*, 1(1):45–56.
- de Shutter, J., , e Spong, M. W. (1997). Force control: A bird's eye view. *IEEE CSS/RAS International Workshop on Control Problems in Robotics and Automation Future Directions, San Diego, CA, December 1997*, 1(1):1–14.
- de Wit, C. C. (1998). Comments on "a new model for control of systems with friction". *IEEE Transactions on Automatic Control*, 43(8):1189–1190.
- de Wit, C. C., Noël, P., Aubin, A., Brogliato, B., e Drevet, P. (1991). Adaptive friction compensation in robot manipulators: Low-velocities. *The International Journal of Robotics Research*, 10(3):189–199.
- de Wit, C. C., Olsson, H., Aström, K. J., e Lischinsky, P. (1995). A new model for control of systems with friction. *IEEE Transactions on Automatic Control*, 40(3):419–425.
- de Wit, C. C., Siciliano, B., e Bastin, G. (1996). *Theory of Robot Control*. Springer-Verlag Company.
- DeCarlo, R. A., Sak, S. H., e Matthews, G. P. (1988). Variable structure control of nonlinear multi-variable systems: A tutorial. *Proceeding of the IEEE*, 76(3):212–232.
- Doulgeri, Z. (1999). Conditions for kinematic stability of position/force control for robots. *International Journal of Robotic Research*, 18(2):259–261.
- Duffy, J. (1990). The fallacy of modern hybrid control theory that is based on "orthogonal complements" of twist and wrench spaces. *Journal of Robotic Systems*, 7(2):139–144.

- Eppinger, S. D. e Seering, W. P. (1986). On dynamic models of robot force control. *Proceedings for the IEEE International Conference on Robotics and Automation*, 1(1):29–34.
- Eppinger, S. D. e Seering, W. P. (1987). Understanding bandwidth limitations. *IEEE International Conference on Robotics and Automation*, 1(1):904–909.
- Eppinger, S. D. e Seering, W. P. (1992). Three dynamics problems in robot force control. *IEEE Transactions on Robotics and Automation*, 8(6):751–758.
- Ferreti, G., Magnani, G., e Rocco, P. (1995). On the stability of integral force control in case of contact with stiff surfaces. *Journal of Dynamics Systems, Measurement and Control*, 117(1):547–553.
- Gandhi, P. S., Ghorbel, F. H., e Dabney, J. (2002). Modeling, identification, and compensation of friction in harmonic drives. *Proceedings of IEEE Conference on Decision and Control*, 1(1):160–166.
- Gao, W. e Hung, J. C. (1993). Variable structure control of nonlinear systems: a new approach. *IEEE Transactions on Industrial Electronics*, 40(1):45–55.
- Glower, J. S. e Munighan, J. (1997). Designing fuzzy controllers from a variable structures standpoint. *IEEE Transactions on Fuzzy Systems*, 5(1):138–144.
- Goldsmith, P. B. (1999). Global asymptotic stability of hybrid position/force control applied to compliant unilateral constraints. *Mechanism and Machine Theory*, 34(1):1009–1021.
- Golin, J. F. (2002). *Controle de Força e Compensação de Atrito de um Robô Industrial*. Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Mecânica (PPGEM), Universidade Federal de Santa Catarina, Brasil.
- Gomes, S. C. P. (1995). Modelagem de atritos internos às articulações de robôs manipuladores. *Congreso Brasileiro de Engenharia Mecânica*, 1b-02(5):1–4.
- Gorinevsky, D. M., Alexander, M., e Schneider, A. (1997). *Force Control of Robotics Systems*. CRC Press New York.
- Hogan, N. (1985). Impedance control: An approach to manipulation part i, ii, iii. *Journal of Dynamics Systems, Measurement and Control*, 107(1):1–24.
- Hüppi, R. e Gruener, G. (2001). *Software Documentation for the SCARA Robot Inter*. Technical Report, Institute of Robotics, Swiss Federal Institute of Technology, Switzerland.
- Hung, J. Y., Gao, W., e Hung, J. C. (1993). Variable structure control: A survey. *IEEE Transactions on Industrial Electronics*, 40(1):2–22.
- Jaritz, A. e Spong, M. W. (1996). An experimental comparison of robust control algorithms on a direct drive manipulator. *IEEE Transactions on Control Systems Technology*, 4(6):627–640.

- Kazerooni, H. (1990). Contact instability of the direct drive robot when constrained by a rigid environment. *IEEE Transactions on Automatic Control*, 35(6):710–714.
- Kiguchi, K. e Fukuda, T. (1997). Intelligent position/force controller for industrial robot manipulators-application on fuzzy neural networks. *IEEE Transactions on Industrial Electronics*, 44:753–761.
- Lee, C. C. (1990). Fuzzy logic in control systems: fuzzy logic controller - part i, ii. *IEEE Transactions on Systems, Man and Cybernetics*, 20:404–435.
- Lewis, F. L., Abdallah, e Dawson, D. M. (1993). *Control of Robot Manipulators*. Macmillan Publishing Company.
- Lewis, F. L. e Stevens, B. L. (1992). *Aircraft Control and Simulation*. John Wiley and Sons.
- Li, Y., Ho, Y. K., e Chua, C. S. (2000). Model based pid control of constrained robot in a dynamic environment with uncertainty. *Proceedings of the 2000 IEEE International Conference on Control Applications*, 1(1):74–79.
- Luca, A. D. e Manes, C. (1991a). Hybrid force-position control for robots in contact with dynamic environment. *3rd. IFAC Symposium on Robot Control, Vienna, Austria*, 1(1):1–10.
- Luca, A. D. e Manes, C. (1991b). On the modeling of robots in contact with a dynamic environment. *5th International Conference on Robotics, Pisa, Italy*, 1(1):568–574.
- Luca, A. D. e Manes, C. (1994). Modeling of robots in contact with dynamic environment. *IEEE Transactions on Robotics and Automation*, 10(4):542–548.
- Manes, C. (1992). Recovering model consistence for force and velocity measures in robot hybrid control. *Proceedings of IEEE, International Conference on Robotics an Automation, Nice, France*, 1(1):1276–1281.
- Manes, C. (1993a). Constrained robot motion in presence of stiff and compliant contact. *Proceeding of IEEE, International Conference on Robotics an Automation*, 1(1):1–8.
- Manes, C. (1993b). Stable force-position control for robots during constrained motion. *International Symposium on Intelligent Robotics, Bangalore, India*, 1(1):1–10.
- Mason, M. T. (1981). Compliance and control for computer controlled manipulators. *IEEE Transactions on Systems, man and cybernetics*, 1(6):418–432.
- McClamroch, N. H. e Wang, D. (1988). Feedback stabilization and tracking of constrained robots. *IEEE Transactions on Automatic Control*, 33(5):419–426.
- Olsson, H. e Aström, K. (1996). Friction generated limit cycles. *Proceeding of IEEE International Conference on Control Applications*, 1(1):798–803.

- Olsson, H., Aström, K. J., de Wit, C. C., Gäfvert, M., e Lischinsky, P. (1997). *Friction Models and Friction Compensation*. Departament of Automatic Control, Lund Institute of Technology.
- P. Lischinsky, C. C. d. W. e Morel, G. (1997). Friction compensation of schilling hydraulic robot. *Proceeding of IEEE International Conference on Control Applications*, 1(1):294–299.
- Passold, F. (2003). *Controle Inteligente de Máquinas: Aplicação no Controle de Posição e Força de Um Robô Manipulador*. Tese de Doutorado, Programa de Pós-Graduação em Engenharia Elétrica (PPGEEL), Universidade Federal de Santa Catarina, Brasil.
- Perondi, E. A. (2002). *Controle Não Linear em Cascata de um Servoposicionador Pneumático com Compensação de Atrito*. Tese de Doutorado, Programa de Pós-Graduação em Engenharia Elétrica (PPGEM), Universidade Federal de Santa Catarina, Brasil.
- Raibert, M. H. e Craig, J. J. (1981). Hybrid position-force control of manipulators. *Journal of Dynamics Systems, Measurement and Control*, 102(5):126–133.
- Ramirez, A. G., Vargas, F. J. T., e Pieri, E. R. D. (2000a). Estratégias de controle a estrutura variável e compensação de atrito aplicadas em um robô industrial. *Congreso Brasileiro de Engenharia Mecânica*, 1(1):1–12.
- Ramirez, A. R. G. (2003). *Controle de Posição de Robôs Manipuladores com Transmissões Flexíveis Considerando a Compensação de Atrito*. Tese de Doutorado, Programa de Pós-Graduação em Engenharia Elétrica (PPGEEL), Universidade Federal de Santa Catarina, Brasil.
- Ramirez, J. A., Cervantes, I., e Kelly, R. (2000b). Pid regulation of robot manipulators: Stability and performance. *Systems and Control Letters*, 41(1):73–83.
- Robert, J. e Spong, M. W. (1988). Hybrid impedance control of robotic manipulators. *IEEE Journal of Robotics and Automation*, 4(5):549–556.
- Rocco, P. (1996). Stability of pid control for industrial robots arms. *IEEE Transactions on Robotics and Automation*, 12(4):606–614.
- Salisbury, J. K. (1980). Active stiffness control of a manipulator in cartesian coordinates. *Proc. 19th IEEE Conf. on Decision and Control*, 1(1):95–100.
- Sandri, S. e Correia, C. (1999). *Lógica Nebulosa, Tutorial*. V Escola de Redes Neurais, INPE, São Jose dos Campos, SP, Brasil.
- Santibañez, V. e Kelly, R. (1997). Strict lyapunov functions for control of robot manipulators. *Automatica*, 33(4):675–682.
- Sciavicco, L. e Siciliano, B. (2000). *Modeling and Control of Manipulators*. The Springer-Verlag London.

- Sheridan, T. B. (1987). Telerobotics. *Automatica*, 25(4):487–507.
- Slotine, J. J. E. e Li, W. (1991). *Applied Nonlinear Control*. Prentice Hall.
- Stefano Chiaverini, B. S. e Villani, L. (1999). A survey of robot interaction control schemes with experimental comparison. *IEEE Transactions on Mechatronics*, 4(3):273–285.
- Tadeo, F., Vega, P., e Holohan, A. (1998). Independent joint control using two degree of freedom control structures. *Proceedings of IEEE International Conference on Control Applications, Trieste Italy*, 1(2):552–556.
- Umeno, T. e Hori, Y. (1991). Robust speed control of dc servomotors using modern two degrees of freedom controller design. *IEEE Transactions on Industrial Electronics*, 38(5):363–368.
- Umeno, T., Kaneko, T., e Hori, Y. (1993). Robust servosystem design with two degrees of freedom and its application to novel motion control of robot manipulators. *IEEE Transactions on Industrial Electronics*, 40(5):473–485.
- Utkin, V. I. (1977). Variable structure systems with sliding modes. *IEEE Transactions on Automatic Control*, 22(2):212–222.
- Utkin, V. I. (1992). *Sliding Modes in Control and Optimization*. Springer Verlag Company.
- Vargas, F. J. T., Pieri, E. R. D., e Castelan, E. B. (2000). Controle de posição de um robô tipo scara utilizando controladores de dois graus de liberdade. *Congresso Brasileiro de Automática*, 1(13):699–704.
- Vargas, F. J. T., Pieri, E. R. D., e Castelan, E. B. (2001a). Estratégia nebulosa para o controle de força e posição de robôs manipuladores. *Reunión de Trabajo en Procesamiento de la Información y Control, Santa Fe, Argentina*.
- Vargas, F. J. T., Pieri, E. R. D., e Castelan, E. B. (2004a). About practical implementation of friction compensation and force-position control using variable structure control. Submitted to *Practical Engineering Control Journal*.
- Vargas, F. J. T., Pieri, E. R. D., e Castelan, E. B. (2004b). Experimental implementation of robust and intelligent controllers in trajectory control of industrial scara robot. *18th IFIP World Computer Congress, Toulouse, France CD Room*, 11(5):415–228.
- Vargas, F. J. T., Pieri, E. R. D., e Castelan, E. B. (2004c). Identificação e compensação de atrito num robô industrial scara. *XV Congresso Brasileiro de Automática CD Room*, 1(1):49–54.
- Vargas, F. J. T., Pieri, E. R. D., e Castelan, E. B. (2004d). Identification and friction compensation for an industrial robot using two degrees of freedom controllers. *Eighth International Conference on Control, Automation, Robotics and Vision*, 1(1):1–12.

- Vargas, F. J. T., Pieri, E. R. D., Castelan, E. B., e Amaral, S. D. (2001b). Force and position control of constrained industrial robots with two degrees of freedom controller. *Proceedings of the IX DINAME*, 1(9):201–206.
- Vargas, F. J. T., Sumar, R. R., e Mejia, R. G. (2004e). Experimental evaluation of pd, pid and vsc controllers with friction compensation in an industrial robot. *Induscon CD Room*, 1(1):1–12.
- Vargas, F. J. T., Sumar, R. R., e Santos, C. H. (2002). Avaliação experimental de uma estratégia nebulosa para o controle de posição de um robô industrial. *XIV Congresso Brasileiro de Automática CD Room*, 1(1):49–54.
- Villani, L., Natale, C., Siciliano, B., e de Wit, C. (2000). An experimental study of adaptive force/position control algorithms for an industrial robot. *IEEE Transactions on Control Systems Technology*, 8(5):777–786.
- Volpe, R. e Khosla, P. (1993). A theoretical and experimental investigation of explicit force control strategies for manipulators. *IEEE Transactions on Automatic Control*, 38(11):1634–1649.
- Volpe, R. e Khosla, P. (1994). Analysis and experimental verification of a fourth order plant model for manipulator force control. *Journal of IEEE Robotics and Automation*, 1(2):4–13.
- Weihmann, L. (1999). *Descrição, Instalação, Programação e Funcionamento de um Robô Manipulador Tipo SCARA*. Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Mecânica (PPGEM), Universidade Federal de Santa Catarina, Brasil.
- Wen, J. T. e Murphy, S. (1991). Stability analysis of position and force control for robot arms. *IEEE Transactions on Automatic Control*, 36(3):365–371.
- Whitney, D. E. (1987). Historical perspective and state of the art in robot force control. *Int. J. Robotics Research*, 6(1):3–14.
- Wilfinger, L. S., Wen, J. T., e Murphy, S. (1994). Integral force control with robustness enhancement. *IEEE Control Systems Magazine*, 1(1):31–40.
- Wolovich, W. A. (1995). *Automatic Control Systems: Basic Analysis and Design*. Saunders College Publishing.
- Yi, S. Y. e Chung, M. (1994). Stability analysis of a fuzzy logic controller for an uncertain dynamic system. Em *Proceedings of IEEE International Fuzzy Systems*, páginas 1028–1034, Orlando (USA).
- Yi, S. Y. e Chung, M. (1997). A robust fuzzy logic controller for robot manipulators with uncertainties. *IEEE Transactions on Systems Man, and Cybernetics - Part B*, 27:706–713.
- Yi, S. Y. e Chung, M. (1998). Robustness of fuzzy logic control for an uncertain dynamic system. *IEEE Transactions on Fuzzy Systems*, 6:216–225.

- Yoshikawa, T. (1987). Dynamic hybrid position-force control of robot manipulators-description of hand constraints and calculation of joint driving force. *IEEE Journal of Robotics and Automation*, RA-3(5):386–392.
- Yoshikawa, T. (2000). Force control of robot manipulators. *Proceedings International Conference on Robotics and Automation*, 1(1):220–226.
- Young, D., Utkin, V. I., e Özgüner, . (1999). A control engineer's guide to sliding mode control. *IEEE Transactions on Control Systems Technology*, 7(3):328–342.
- Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 08:338–353.
- Zadeh, L. A. (1996). Fuzzy logic = computing with words. *IEEE Transactions on Fuzzy Systems*, 04:103–111.
- Zhu, Y. e Pagilla, P. R. (2002). Static and dynamic friction compensation in trajectory tracking control of robots. *Proceedings of IEEE International Conference on Robotics and Automation*, 1(1):2644–2649.