

Nilton Alejandro Cuellar Loyola

**Interface Háptica de Cinco Graus
de Liberdade para Teleoperação de
Manipuladores Robóticos**

DISSERTAÇÃO DE MESTRADO

DEPARTAMENTO DE ENGENHARIA MECÂNICA
Programa de Pós-graduação em Engenharia
Mecânica

Rio de Janeiro
Maio de 2012

Nilton Alejandro Cuellar Loyola

**Interface Háptica de Cinco Graus de Liberdade
para Teleoperação de Manipuladores Robóticos**

Dissertação de Mestrado

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Mecânica da PUC-Rio como requisito parcial para obtenção do título de Mestre em Engenharia Mecânica

Orientador: Prof. Marco Antonio Meggiolaro

Rio de Janeiro
Maio de 2012

Nilton Alejandro Cuellar Loyola

**Interface Háptica de Cinco Graus de Liberdade
para Teleoperação de Manipuladores Robóticos**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Engenharia Mecânica do Centro Técnico Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Marco Antonio Meggiolaro

Orientador

Departamento de Engenharia Mecânica — PUC-Rio

Prof. Mauro Speranza Neto

Departamento de Engenharia Mecânica — PUC-Rio

Prof. Liu Hsu

Departamento de Engenharia Elétrica — COPPE/UFRJ

Prof. José Eugênio Leal

Coordenador Setorial do Centro Técnico Científico — PUC-Rio

Rio de Janeiro, 28 de Maio de 2012

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Nilton Alejandro Cuellar Loyola

Graduou-se em Engenharia Mecatrônica na *Universidad Nacional de Ingeniería - UNI* (Lima, Perú) em 2008. Trabalhou na área de desenvolvimento de projetos em robótica e automação industrial antes de começar mestrado em IGI-UNI. Suas áreas de interesse abrangem Robótica, Controle de Sistemas, Visão Computacional e Computação Gráfica

Ficha Catalográfica

Cuellar Loyola, Nilton Alejandro

Interface Háptica de Cinco Graus de Liberdade para Teleoperação de Manipuladores Robóticos / Nilton Alejandro Cuellar Loyola; orientador: Marco Antonio Meggiolaro. — 2012.

183 f.: il. (color.) ; 30 cm

Dissertação (mestrado) — Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Mecânica, 2012.

Inclui bibliografia

1. Engenharia Mecânica – Teses. 2. Robótica;. 3. Interface Háptica;. 4. Chai3D;. 5. Ambientes Virtuais;. 6. Controle não Linear;. 7. Modo Deslizantes..

I. Meggiolaro, Marco Antonio. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Mecânica. III. Título.

Aos meus pais, Alejandro e Ana
e meus irmãos, Rosa e Julio.

Agradecimentos

A meu orientador Marco Antonio Meggiolaro pela confiança, o suporte acadêmico e pela inestimável contribuição na minha formação.

À CAPES pelos auxílios concedidos e à PUC-Rio, pelo seu excelente corpo acadêmico que me deu as bases para este estudo, sem os quais este trabalho não poderia ter sido realizado.

A minha família pelo amor, pela educação, pelo apoio que próxima ou distante me incentivou nessa jornada.

A Pâmela, que conviveu, incentivou e inclusive trabalhou a meu lado, para que este desafio fosse superado.

Por fim, agradeço meus colegas da Pós-Graduação com os que passei bons momentos: Fernando, Niltshon, Juliana, Carmen, Ronald, Edward, Elder, Cesar, Ingrid, Juan, Marvin, Edwin, Mirko, Darwin e José.

Resumo

Cuellar Loyola, Nilton Alejandro; Meggiolaro, Marco Antonio. **Interface Háptica de Cinco Graus de Liberdade para Teleoperação de Manipuladores Robóticos**. Rio de Janeiro, 2012. 183p. Dissertação de Mestrado — Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

O sucesso de diversas tarefas de teleoperação depende muito da habilidade do operador e de sua capacidade de perceber o ambiente de trabalho. A realimentação visual em muitos casos não é suficiente, por exemplo quando a qualidade da imagem do ambiente de trabalho é baixa, quando ocorrem oclusões na visualização, ou quando a tarefa envolve forças de contato associadas a folgas pequenas visualmente imperceptíveis. Para compensar essas deficiências, os dispositivos hápticos surgem como uma alternativa à realimentação visual, ao interagir com o usuário através do tato, produzindo uma sensação de força.

Esta dissertação apresenta o desenvolvimento e modelagem de um sistema de interface háptica de cinco graus de liberdade para a teleoperação de robôs manipuladores, com foco naqueles que realizam trabalhos em ambientes perigosos ou hostis ao ser humano. A interface é desenvolvida a partir do acoplamento de dois dispositivos hápticos comerciais “Novint Falcon”, de três graus de liberdade cada. O sistema resultante do acoplamento é modelado como um manipulador paralelo, capaz de fornecer ao operador, realimentação de força 3D (em três direções) e realimentação de torque em duas direções.

Para demonstrar a eficiência do sistema háptico desenvolvido, um ambiente virtual é implementado com o auxílio de técnicas de computação gráfica e bibliotecas como OpenGL, ODE e Chai3D. Os modelos cinemático e dinâmico de um manipulador serial Schilling Titan IV, de seis graus de liberdade, são implementados no ambiente virtual, incluindo sua interação com objetos (virtuais) do ambiente de teleoperação. Controladores não lineares são implementados no manipulador serial virtual, incluindo controle de torque computado, robusto, e por modos deslizantes.

Palavras-chave

Robótica; Interface Háptica; Chai3D; Ambientes Virtuais; Controle não Linear; Modo Deslizantes.

Abstract

Cuellar Loyola, Nilton Alejandro; Meggiolaro, Marco Antonio (Advisor). **Five Degree-of-Freedom Haptic Interface for Teleoperation of Robotic Manipulators**. Rio de Janeiro, 2012. 183p. MSc. Dissertation — Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

The success of many teleoperation tasks depends heavily on the skills of the operator and his ability to perceive the work environment. Visual feedback, in many cases, is not sufficient e.g. when the image quality of the work environment is low, occlusions occur in the display, or when the task involves contact forces associated with visually unnoticeable small clearances. To compensate for these shortcomings, haptic devices emerge as an alternative to visual feedback, in which touch interaction with the user produces force-feedback.

This thesis presents the development and modeling of a haptic interface system of five degrees of freedom for the teleoperation of robot manipulators, focusing on those that work in hazardous or hostile environments for humans. The interface is developed from the coupling of two commercial haptic devices “Novint Falcon”, with three degrees of freedom each. The system resulting from the coupled devices is modeled as a parallel manipulator capable of providing the operator with 3D force feedback (in three dimensions) and torque feedback in two directions.

To demonstrate the effectiveness of the developed haptic system, a virtual environment is implemented with the aid of computer graphics techniques and libraries such as OpenGL, ODE and Chai3D. The kinematic and dynamic models of a serial manipulator Schilling Titan IV, with six degrees of freedom, are implemented in the virtual environment, including its interaction with virtual objects for the evaluation of typical teleoperation tasks. Nonlinear controllers are implemented in the virtual serial manipulator, including computed torque and sliding mode control.

Keywords

Robotics; Haptics Interfaces; Chai3D; Virtual Environment; Non-Linear Control; Sliding Mode.

Sumário

Sumário das notações	16
1 Introdução	18
1.1 Definição do Problema	18
1.2 Objetivos da Dissertação	22
1.3 Estrutura da Dissertação	23
2 Sistemas de Interface Háptica	25
2.1 Definições Prévias	25
2.1.1 Sensações Táteis e Cinestésicas	25
2.1.2 Percepção Humana e a Interfaces Hápticas	27
2.2 Realidade Virtual	28
2.2.1 Dispositivos de Entrada e Saída	30
2.2.2 Princípios de Projeto e Operação	31
2.3 Sistemas de Interface Háptica	36
2.4 Dispositivos Hápticos	37
2.4.1 Conceitos Básicos	38
2.5 Componentes dos Sistemas Hápticos	41
2.5.1 Sensores	41
2.5.2 Atuadores	42
2.5.3 Modelagem da Cena Háptica	42
2.5.4 Renderização Háptica	43
2.6 Principais Dispositivos Hápticos Comerciais	44
2.6.1 Phantom Omni	45
2.6.2 Delta.6	45
2.6.3 Sigma.7	46
2.7 Principais Aplicações com Dispositivos Hápticos	47
2.7.1 Histórico	47
2.7.2 Medicina	48
2.7.3 Educação	50
2.7.4 Militar e Aeroespacial	51
3 Modelagem do Sistema Escravo	55
3.1 Descrição do Sistema	55
3.2 Cinemática Direita	56
3.2.1 A Notação de Denavit-Hartenberg	57
3.2.2 Modelagem de Denavit-Hartenberg	59
3.3 Cinemática Inversa	60
3.3.1 Dificuldades dos Métodos Geométricos	65
3.3.2 Problema da escolha da Solução	66
3.3.3 Função Custo	66
3.4 Cinemática Diferencial	67
3.5 Dinâmica Inversa	68
3.6 Dinâmica Direta	69

4	Controle de Movimento	71
4.1	Controle no Espaço das Juntas	71
4.2	Controle de Posição	72
4.2.1	Controle PD + Compensação de Gravidade	77
4.3	Controle Seguimento de Trajetória	80
4.3.1	Controle de Torque Computado	80
4.4	Controle Robusto de Trajetória	85
4.4.1	Controle por Modos Deslizantes	85
5	Dispositivo Háptico de 5 Graus de Liberdade	102
5.1	Descrição do Falcon	102
5.2	Modelagem do Falcon	104
5.2.1	Geometria do Manipulador	105
5.2.2	Cinemática Inversa	106
5.2.3	Cinemática Direta	109
5.2.4	Jacobiano	110
5.3	Estática	114
5.4	Descrição do Hardware	115
5.5	Compensação de Gravidade	118
5.6	Controle de 5 Graus de Liberdade	120
6	Desenvolvimento do Simulador Virtual	126
6.1	Definições Prévias	126
6.1.1	Thread	126
6.1.2	Framework	126
6.1.3	API	127
6.1.4	Biblioteca	127
6.1.5	Motores Gráficos	127
6.1.6	OpenGL	128
6.1.7	ODE	128
6.2	CHAI3D	129
6.2.1	Efeitos Hápticos	131
6.2.2	Módulo ODE	131
6.2.3	Módulo GEL	132
6.2.4	Arquitetura da Interface Háptica	133
6.3	Desenvolvimento do Ambiente Virtual	135
6.3.1	Simulador do Robô Manipulador	137
6.3.2	Integração da Interface Háptica com o Ambiente Virtual	139
7	Resultados Experimentais	141
7.1	Teste da Interface Háptica	141
7.2	Controle do Robô Manipulador	149
8	Conclusões e Sugestões	156
8.1	Conclusões do Trabalho	156
8.2	Sugestões para Trabalhos Futuros	157
	Referências Bibliográficas	159
A	Apêndice A	166

A.1	Matriz Homogênea	166
A.2	Matriz Jacobiana	168
A.3	Algoritmo Walker-Orin	169
A.3.1	Método 1	170
A.3.2	Método 3	170
B	Apêndice B	176
B.1	Modelagem Dinâmica	176
B.1.1	Propriedade 1	176
B.1.2	Propriedade 2	176
B.1.3	Propriedade 3	176
B.1.4	Propriedade 4	177
B.1.5	Propriedade 5	177
C	Apêndice C	179
C.1	Instalação do CHAI3D em MVS2008	179

Lista de figuras

1.1	Diferentes tipos de Robôs teleoperados	19
1.2	Diferentes tipos de Robôs teleoperados [1].	21
2.1	Exemplo de seis procedimentos para explorar um objeto .	26
2.2	Um jogo simulador de corrido de carros.	28
2.3	Exemplo de algumas aplicações da Realidade Virtual.	29
2.4	Dispositivos Eletromagnéticos.	32
2.5	Dispositivos Mecânicos.	33
2.6	Dispositivos Acústicos.	34
2.7	Dispositivos Inerciais.	35
2.8	Dispositivos Ópticos.	37
2.9	Exemplo de um Sistema Háptico	38
2.10	Fluxo de informações na interação com um mouse convencional (esquerda) e um mouse háptico (direta)	40
2.11	Cena de uma Renderização Háptica	44
2.12	Dispositivo Háptico Phantom-Omni	45
2.13	Dispositivo Háptico Delta.6 Force Dimension	46
2.14	Dispositivo Háptico Sigma.7 Force Dimension	46
2.15	Simulador de automóveis desenvolvido pela empresa Toyota	48
2.16	Simulador Cirúrgico Sela[2]	49
2.17	Diagnóstico de Trombose [3]	50
2.18	Simulação da fixação da fratura do Quadril [4]	51
2.19	Sistemas Hápticos para a Educação	52
2.20	Simuladores para treinamento militar	52
2.21	Simulador de aviões militares	53
2.22	Simulador do veículo terrestre Stryker	54
3.1	Robot Schilling Titan IV e miniatura utilizado como "mestre" na teleoperação [5].	56
3.2	A notação Denavit-Hartenberg [6].	57
3.3	Posição e Orientação do efetuador terminal representado pela matriz \mathbf{T} [6].	59
3.4	Robô Schilling Titan IV e seu sistema de coordenadas.	60
3.5	Duas soluções possíveis para os ângulos de junta, dada uma mesma posição.	61
3.6	Método geométrico para calcular a cinemática inversa do robô Titan IV.	63
3.7	Eixos O_2 , O_3 e O_4 , sobre a forma cinemática inversa de dois graus de liberdade.	64
4.1	Esquema genérico do conceito de Controle no espaço da junta.	72
4.2	Robô Manipulador serial de 6 Graus de Liberdade.	72
4.3	Diagrama de Blocos: Controle PD.	73
4.4	Referência e resposta de posição: juntas q_1 , q_2 e q_3 .	74
4.5	Referência e resposta de posição: juntas q_4 , q_5 e q_6 .	75

4.6	Erros de posicionamento: juntas \tilde{q}_1 , \tilde{q}_2 e \tilde{q}_3 .	75
4.7	Erros de posicionamento: juntas \tilde{q}_4 , \tilde{q}_5 e \tilde{q}_6 .	76
4.8	Diagrama de Blocos: Controle PD com compensação de gravidade.	77
4.9	Referência e resposta de posição: juntas q_1 , q_2 e q_3 .	78
4.10	Referência e resposta de posição: juntas q_4 , q_5 e q_6 .	78
4.11	Erros de posicionamento: juntas \tilde{q}_1 , \tilde{q}_2 e \tilde{q}_3 .	79
4.12	Erros de posicionamento: juntas \tilde{q}_4 , \tilde{q}_5 e \tilde{q}_6 .	79
4.13	Diagrama de Blocos: Controle por Torque Computado.	80
4.14	Referência e resposta do seguimento de trajetória: juntas q_1 , q_2 e q_3 .	82
4.15	Referência e resposta do seguimento de trajetória: juntas q_4 , q_5 e q_6 .	83
4.16	Erros de posicionamento: juntas \tilde{q}_1 , \tilde{q}_2 e \tilde{q}_3 .	83
4.17	Erros de posicionamento: juntas \tilde{q}_4 , \tilde{q}_5 e \tilde{q}_6 .	84
4.18	Superfícies de Deslizamento.	87
4.19	Interpretação gráfica das equações (4-10) e (4-18) ($n = 2$).	89
4.20	Fenômeno de <i>Chattering</i> (ou Chaveamento).	92
4.21	(a)Camada Limite para um sistema de segunda ordem. (b)Controle de Interpolação na camada limite.	92
4.22	Camada Limite.	93
4.23	Referência e resposta do seguimento de trajetória: juntas q_1 , q_2 e q_3 .	97
4.24	Referência e resposta do seguimento de trajetória: juntas q_4 , q_5 e q_6 .	97
4.25	Erros de posicionamento: juntas \tilde{q}_1 , \tilde{q}_2 e \tilde{q}_3 .	98
4.26	Erros de posicionamento: juntas \tilde{q}_4 , \tilde{q}_5 e \tilde{q}_6 .	98
4.27	Gráfico dos torques produzidos: τ_1 , τ_2 e τ_3 .	99
4.28	Gráfico dos torques produzidos: τ_4 , τ_5 e τ_6 .	99
4.29	Referência e Resposta da Velocidade: \dot{q}_1 , \dot{q}_2 e \dot{q}_3 .	100
4.30	Referência e Resposta da Velocidade: \dot{q}_4 , \dot{q}_5 e \dot{q}_6 .	100
4.31	Superfícies Deslizantes do Sistema.	101
5.1	O Dispositivo Háptico Novint Falcon [®]	103
5.2	Membros do dispositivo háptico Novint Falcon [®]	103
5.3	Robô Manipulador Paralelo da Universidade de Maryland	104
5.4	Modelo CAD do Novint Falcon [®]	104
5.5	Representação esquemática do Novint Falcon [®]	105
5.6	Vista frontal e superior de um suporte do Novint Falcon [®]	107
5.7	Esquema do Dispositivo Háptico de 5 graus de liberdade	116
5.8	Falcons fixos, ainda desacoplados	116
5.9	Componentes mecânicos	117
5.10	Dispositivo Háptico de 5 graus de liberdade	117
5.11	Diagrama de Corpo Livre de um elo do Falcon	118
5.12	Forças de Gravidade em um elo do Falcon	119
5.13	Esquema de controle bilateral Posição - Posição	121
5.14	Esquema é o controle bilateral com sensor de força	121
5.15	Sistema de Coordenadas	122
5.16	Efetuator Terminal do dispositivo Háptico de 5 graus de liberdade	123
5.17	Efetuator Terminal do dispositivo Háptico de 5 graus de liberdade	124
6.1	Exemplo do módulo ODE em CHAI3D	132
6.2	Exemplo do módulo GEL em CHAI3D	132
6.3	Arquitetura da Interface Háptica no CHAI3D	133

6.4	Desenho da caixa fixa com orifício na parte central	135
6.5	Implementação da caixa fixa com orifício na parte central	135
6.6	Implementação dos vetores de força	136
6.7	Desenho do Robô Manipulador Schilling Titan IV	137
6.8	Implementação da Cinemática Inversa para o Robô Manipulador	138
6.9	Implementação do algoritmo de colisões	139
7.1	Objeto virtual para testes	142
7.2	Usuário 1: Sem Sensação de Força, mostrando as 5 tentativas de inserção (correspondente aos 5 picos)	142
7.3	Usuário 1: Com Sensação de Força, mostrando as 5 tentativas de inserção (correspondente aos 5 picos)	143
7.4	Usuário 2: Sem Sensação de Força, mostrando as 5 tentativas de inserção (correspondente aos 5 picos)	143
7.5	Usuário 2: Com Sensação de Força, mostrando as 5 tentativas de inserção (correspondente aos 5 picos)	144
7.6	Usuário 3: Sem Sensação de Força, mostrando as 5 tentativas de inserção (correspondente aos 5 picos)	144
7.7	Usuário 3: Com Sensação de Força, mostrando as 5 tentativas de inserção (correspondente aos 5 picos)	145
7.8	Usuário 4: Sem Sensação de Força, mostrando as 5 tentativas de inserção (correspondente aos 5 picos)	145
7.9	Usuário 4: Com Sensação de Força, mostrando as 5 tentativas de inserção (correspondente aos 5 picos)	146
7.10	Usuário 5: Sem Sensação de Força, mostrando as 5 tentativas de inserção (correspondente aos 5 picos)	146
7.11	Usuário 5: Com Sensação de Força, mostrando as 5 tentativas de inserção (correspondente aos 5 picos)	147
7.12	Gráficos dos testes feitos com 5 pessoas	148
7.13	Referência e resposta de posição: juntas q_1 , q_2 e q_3 .	149
7.14	Referência e resposta de posição: juntas q_1 , q_2 e q_3 . (PD)	150
7.15	Referência e resposta de posição: juntas q_4 , q_5 e q_6 . (PD)	150
7.16	Erros de posicionamento: juntas \tilde{q}_1 , \tilde{q}_2 e \tilde{q}_3 . (PD)	151
7.17	Erros de posicionamento: juntas \tilde{q}_4 , \tilde{q}_5 e \tilde{q}_6 . (PD)	151
7.18	Referência e resposta de posição: juntas q_1 , q_2 e q_3 . (Torque Computado)	152
7.19	Referência e resposta de posição: juntas q_4 , q_5 e q_6 . (Torque Computado)	152
7.20	Erros de posicionamento: juntas \tilde{q}_1 , \tilde{q}_2 e \tilde{q}_3 . (Torque Computado)	153
7.21	Erros de posicionamento: juntas \tilde{q}_4 , \tilde{q}_5 e \tilde{q}_6 . (Torque Computado)	153
7.22	Referência e resposta de posição: juntas q_1 , q_2 e q_3 . (Modos Deslizantes)	154
7.23	Referência e resposta de posição: juntas q_4 , q_5 e q_6 . (Modos Deslizantes)	154
7.24	Erros de posicionamento: juntas \tilde{q}_1 , \tilde{q}_2 e \tilde{q}_3 . (Modos Deslizantes)	155
7.25	Erros de posicionamento: juntas \tilde{q}_4 , \tilde{q}_5 e \tilde{q}_6 . (Modos Deslizantes)	155
A.1	Posição do centro de massa G com respeito à coordenada O	173
A.2	Centro de massa do sistema composto do elo j até o elo n	173

C.1	Microsoft Visual Studio 2008	179
C.2	Compilação dos exemplos demo do CHAI3D	180
C.3	Exemplo 01-devices do CHAI3D	180
C.4	Criando novo projeto no MVS2008	181
C.5	Enlace de arquivos cabeçalhos .h (Passos 2, 3 e 4)	182
C.6	Enlace de arquivos .lib (Passos 5 e 6)	182
C.7	Compilação de um projeto com Chai3D	183

Lista de tabelas

3.1	Parâmetros do robô Manipulador Schiling Titan IV	60
5.1	Valores da massa e comprimento dos Parâmetros	120

Sumário das notações

Lista de Símbolos

x, y, z	coordenadas do sistema cartesiano
i	i-ésimo componente
O	origem do sistema cartesiano
a, d	parâmetros D-H
\mathbf{X}	vetor posição.
\mathbf{R}	matriz de rotação
\mathbf{A}	matriz homogênea transformação
\mathbf{T}	matriz homogênea D-H
\mathbf{q}	vetor deslocamento das juntas
$\dot{\mathbf{q}}$	vetor velocidade das juntas
$\ddot{\mathbf{q}}$	vetor aceleração das juntas
p_x, p_y, p_z	posição do efetuador terminal do robô
L_i	comprimento dos elos do robô
n	numero de graus de liberdade do robô
dp	vetor diferencial da posição no espaço
$d\mathbf{x}_e$	vetor diferencial da posição tridimensional
$d\Phi_e$	vetor diferencial da orientação tridimensional
\mathbf{b}	vetor unitário na direção da junta
\mathbf{r}	vetor
\mathbf{J}_L	matriz Jacobiano linear
\mathbf{J}_A	matriz Jacobiano angular
\mathbf{v}_e	vetor da velocidade linear do efetuador terminal
\mathbf{w}_L	vetor da velocidade angular do efetuador terminal
\mathbf{f}	vetor força de acoplamento na junta
\mathbf{N}	vetor momento de acoplamento na junta
m	massa do elo robô
x_c, y_c, z_c	coordenadas do centro de massa
\mathbf{v}_c	vetor velocidade do centro de massa do elo robô
\mathbf{a}_c	vetor aceleração do centro de massa do elo robô
\mathbf{g}	vetor aceleração da gravidade
\mathbf{I}	matriz Momento de Inércia
$\bar{\mathbf{I}}$	matriz Tensor de Inércia
\mathbf{k}_{Fc}	coeficiente de atrito

\mathbf{k}_{Fv}	coeficiente de viscosidade
\mathbf{H}	matriz de inércia do robô
\mathbf{C}	matriz de forças coriolis e centrífuga do robô
\mathbf{G}	vetor da força gravitacionais do robô
\mathbf{E}	matriz de inércia do sistema composto
\mathbf{N}_j	vetor Momento do sistema composto
\mathbf{F}_j	vetor Força do sistema composto
M_j	massa total do sistema composto
c_o, g_o	algumas constantes
U	energia potencial
\mathbf{q}_d	vetor deslocamento das juntas desejada
$\dot{\mathbf{q}}_d$	vetor velocidade das juntas desejada
$\mathbf{K}_P, \mathbf{K}_D$	matriz diagonal definida positiva
V	função escalar Lyapunov
$\tilde{\mathbf{q}}$	vetor erro de deslocamento das juntas
$\tilde{\dot{\mathbf{q}}}$	vetor erro de velocidade das juntas
I	matriz identidade
u	entrada de controle
$f(x, t), g(x, t), d(x, t)$	funções não lineares
$s(x, t)$	superfície de deslizamento
u_{eq}	controle equivalente
K	ganho de controle
$\dot{\mathbf{q}}_r$	vetor velocidade da referência
θ	ângulo da junta
ω	velocidade da junta
τ	Torque na junta
ρ	densidade de massa
λ_{min}	valor proprio
λ	constante
ε	largura da camada limite
ϵ	constante
Λ	matriz simétrica definida positiva
η	constante Positiva
Φ	espessura da camada limite

1 Introdução

1.1 Definição do Problema

Na atualidade os robôs em geral são muito utilizados para substituir o homem em trabalhos repetitivos, ambientes hostis ou que ofereçam perigo. Em vários casos os robôs são até responsáveis pela realização de uma série de tarefas sem a presença humana, ou seja, há diversas situações nas quais a presença humana faz-se inadequada, ou mesmo impraticável, seja pelo risco inerente envolvendo a sua integridade física, pelo custo elevado, pelo desconhecimento das reais necessidades para a manutenção da vida, ou ainda por restrições de espaço para a execução apropriada de uma dada tarefa.

Dentro deste contexto surgiram os sistemas de teleoperação que como o próprio nome sugere, dizem respeito a um conjunto de dispositivos que permitem ser operados ou manipulados à distância. Geralmente, estes tipos de sistemas são utilizados em explorações oceanográficas ou geológicas em geral, aplicações militares e de resgate para a defesa civil, ocupações de plantas nucleares contaminadas, medicina e explorações espaciais, dentre outras. Podem-se alguns tipos de robô teleoperados visualizar na figura 1.1.

Existem vários tipos de sistemas de teleoperação, dentre os quais podem ser citados: os veículos operados remotamente ou ROV's [1], [2]; Sistemas como o Zeus [3]; [4] que são utilizados em cirurgias robóticas. E, robôs como o Sojourner, que fez parte da missão "Mars Pathfinder" ao planeta Marte [5], [6] e consistia em um teste, onde o Sojourner era controlado a partir de fotos e sinais enviados por ele mesmo para a Terra.

Na Universidade Federal do Rio de Janeiro (UFRJ), o departamento de Pós-graduação e Pesquisa de Engenharia tem desenvolvido um sistema Automático em Robô Submarino, o BPM (Braço Passivo Mecânico), que é um sistema pioneiro. Acoplado a robôs submarinos tipo ROV (*Remotely Operated Vehicle*) orienta o robô, que por si só é incapaz de localizar-se no fundo do mar, e o mantém estacionado durante trabalhos de inspeção e reparos em equipamentos de petróleo offshore. Desde o início o sistema suscitou o interesse

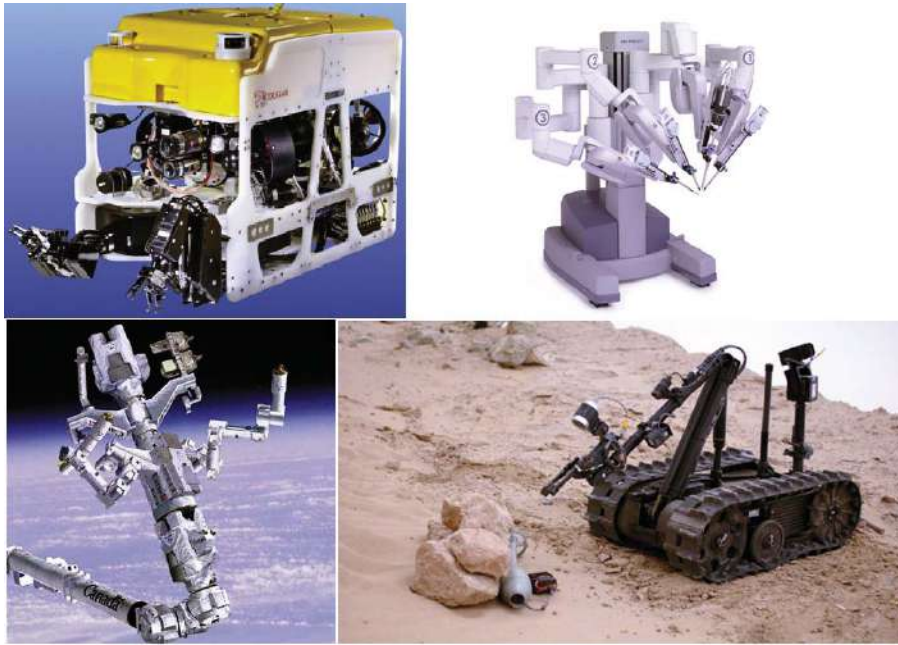


Figura 1.1: Diferentes tipos de Robôs teleoperados

do Centro de Pesquisa e Desenvolvimento da Petrobras (CENPES).

Um problema presente em todos os sistemas de teleoperação é a degradação da informação sensorial do ambiente remoto, devido a uma pobre realimentação visual. Este caso é comum em teleoperação, dado que em muitos casos se trabalha em ambientes com pouca visibilidade, um campo de visão limitado ou uma escala microscópica. Oyama[7] desenvolveu um sistema de teleoperação onde a realimentação visual se degradava com a fumaça. O operador controlava um braço mestre cinematicamente, equivalente ao robô escravo de 6 graus de liberdade.

Outro exemplo de teleoperação com uma pobre realimentação visual é a operação de robôs em ambientes submarinos. Inclusive os operadores mais experimentados podem se desorientar trabalhando em águas turbulentas que contêm uma rede complexa de dutos. Isto pode ser causa de acidentes ou perdas de equipamentos por más condições de uso. Em muitos casos, se utilizam simuladores para treinamento com técnicas de realidade aumentada.

Outra dificuldade associada à teleoperação é a presença de retardos nas comunicações devido às longas distâncias. Neste caso, ocorre tanto em aplicações espaciais como terrestres, caso por qualquer razão a comunicação for muito lenta. Em um esquema de teleoperação clássica, o usuário deve adotar uma estratégia que "ordena e espera", o qual conduz a tempos inaceitáveis em tarefas de longa duração.

Muitos destes dos trabalhos que tratam de sistemas com teleoperação de robôs fazem apenas referência à percepção de força isoladamente. Dessa forma, não levam em consideração outros parâmetros importantes envolvidos, como a percepção da textura, temperatura e, inclusive, a sensação de dor. Em vista disto, o termo háptico pode ser definido como a capacidade de avaliar parâmetros como a aspereza, e a força aplicada em um objeto real, ou seja, é a análise de toda a capacidade tátil do ser humano projetado em um equipamento remoto. Se o usuário puder utilizar seu próprio braço para movimentar o robô, e recebendo a realimentação de cada movimento, o manuseio do mesmo se tornará mais fácil e seguro, uma vez que o robô seguirá fielmente o movimento do operador, dando a este a sensação de estar presente no ambiente executando a tarefa. A fidelidade dos movimentos de um robô, em relação ao operador, em alguns casos é de extrema importância.

Pensando e identificando os problemas clássicos de teleoperação (ambientes com pobre realimentação visual, retardos de comunicação e falta de percepção), obtém-se uma considerável melhora quando utiliza-se a realidade virtual e sistemas hápticos integrados a eles.

Atualmente há várias pesquisas nos problemas clássicos de teleoperação, abordando e utilizando a tecnologia háptica. Na maioria dos laboratórios de pesquisa em robótica, existem projetos que tratam de resolver estes tipos de problemas. Citam-se a seguir alguns exemplos em que a colaboração da realidade virtual e a teleoperação estão obtendo importantes avanços.

Na Universidade de California, Blackmon e Stark[8] desenvolveram um modelo de controle para supervisão, utilizando um “*script* de seqüências de trabalho”. O sistema consistia numa estação gráfica de trabalho onde havia um simulador de ambiente remoto. Uma segunda estação de trabalho, colocada em um lugar remoto, controlava um robô RM501 da Mitsubishi de 5 graus de liberdade (figura 1.2). O modelo virtual do robô era controlado pelo operador com um par de *joysticks* de 2 graus de liberdade, onde o operador realizava a tarefa evitando potenciais colisões. Cada *script* da tarefa era realizada primeiro no simulador e logo enviada ao ambiente real. Estes experimentos comprovaram que o sistema oferecia melhor resultado que uma manipulação direta.

Outro trabalho importante foi desenvolvido na Drexel University, numa tese de Doutorado relacionada ao desenho de um robô para cirurgia por computador assistida, cuja utilidade principal foi diminuir traumas aos pacientes com doenças. Nesta pesquisa, restaura-se a capacidade de um retroalimentação de força para cirurgia assistida com robôs[9].

No Brasil, numerosos veículos submarinos foram desenvolvidos para a realização de tarefas submarinas. A aplicação de robôs teleoperados nestes

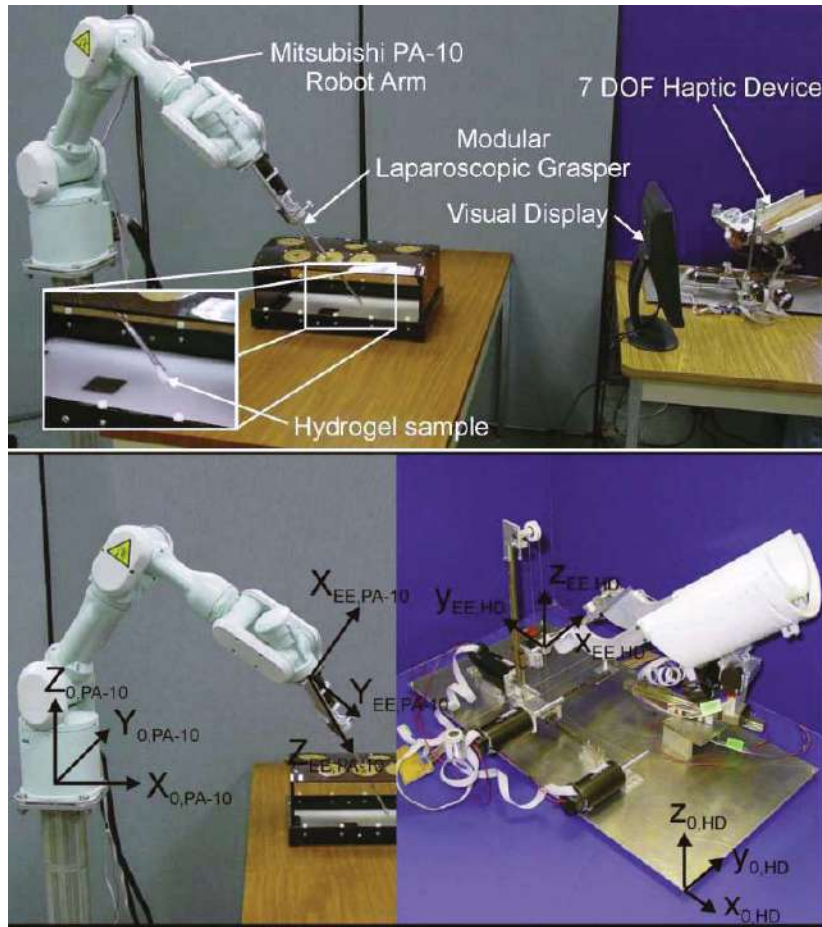


Figura 1.2: Diferentes tipos de Robôs teleoperados [8].

veículos deve-se aos riscos que tem para o humano a realização de trabalhos submarinos, devido às extremas condições de pressão e temperatura, correntes de água, etc. Mas são poucos os trabalhos que tentam incorporar a seu sistema de manipulação uma interface háptica para melhorar sua operação. Este foi o fator preponderante para criação deste projeto, um sistema háptico para teleoperar com realimentação de força, capaz de proporcionar ao usuário do sistema robótico em questão melhor capacidade de executar tarefas. O sistema desenvolvido neste trabalho atua através de um ambiente simulador virtual, onde um robô realizara tarefas ordenadas desde um dispositivo háptico de 5 graus de liberdade de baixo custo.

Os dispositivos hápticos comerciais, que possuem alta precisão e que têm uma funcionalidade em seis graus de liberdade, disponíveis para sua incorporação aos ambientes virtuais, têm o inconveniente de ter um custo relativamente elevado (Phantom Omni \$2400 USD, Phantom Desktop \$13,000 USD, Phantom Premium 1.5 \$27,500 USD), e isso dificulta o uso em maior

escala, sobre tudo em instituições educativas e de pesquisas. Embora existam dispositivos hápticos de baixo custo no mercado projetados para jogos, estes não têm o número de graus de liberdade requeridos para trabalhos com mais de 3 graus de liberdade.

Um das áreas de oportunidade identificadas neste trabalho de pesquisa é o projeto e adaptação de dispositivos hápticos de baixo custo, integrados a um ambiente virtual de três dimensões, para treinamento e teleoperação de robôs manipuladores. Por isso, é construído um dispositivo háptico de 5 graus de liberdade ligando dois dispositivos de 3 graus de liberdade (Novint Falcon[®]). Além disso, desenvolve-se um ambiente de simulação virtual, onde será teleoperado um modelo do robô manipulador Schilling Titan IV, de 6 graus de liberdade. O ambiente de simulação virtual será desenhado utilizando bibliotecas de OpenGL, ODE (Open Dynamic Engine) e CHAI3D, que são de código fonte aberto.

Para as interações das sensações hápticas ou realimentações de forças, é necessário que haja um sincronismo entre os dois sistemas (virtual-real). Com isto se irá proporcionar uma maior precisão e fidelidade nos movimentos a serem realizados pelo robô, bem como garantir uma maior segurança em um ambiente de trabalho com as condições descritas anteriormente (pois sua principal característica é permitir que o usuário que opera o sistema fique afastado fisicamente do local onde as tarefas serão realizadas).

Muitas aplicações práticas do trabalho em estudo, são em situações de ambientes hostis, que têm emprego em um número de operações, como inspeção de dutos de ar-condicionado e tubulações de alta pressão, na busca de sobreviventes em desabamentos ou tremores, em sistemas de segurança patrimonial, no desarme de bombas, na exploração espacial, etc.

1.2 Objetivos da Dissertação

O objetivo geral desta dissertação é estudar a importância dos Sistemas Hápticos e sua utilização, e implementá-lo em um sistema de tele-operação de robôs manipuladores. O sistema é caracterizado por proporcionar uma retroalimentação de força (força de reação), proveniente de um sistema robótico tipo escravo colocado dentro de um ambiente virtual, que também será desenvolvido e que deve permanecer sincronizado com o sistema robótico mestre. O sistema mestre é um dispositivo háptico, cuja movimentação envolve 5 graus de liberdade, para o usuário interagir em tempo real.

O trabalho se concentra em estudar três conceitos importantes:

- Comprovar mediante práticas reais a importância de utilizar sistemas hápticos.
- Desenvolver um simulador de ambiente virtual, capaz de controlar as trajetórias desejadas para um robô de 6 graus de liberdade.
- Aplicação de um dispositivo háptico de 5 graus de liberdade para dirigir o efetuador terminal do robô.

1.3

Estrutura da Dissertação

Esta dissertação de mestrado está organizada em oito capítulos, sendo o primeiro deles esta breve introdução, que procura dar a ambientação do trabalho realizado.

No capítulo 2 apresenta-se uma narrativa que aborda e mostra definições utilizadas neste trabalho. Além disso, descrevem-se conceitos fundamentais sobre Sistemas Hápticos, Realidade Virtual e tipos de dispositivos Hápticos que atualmente podem-se encontrar em numerosas pesquisas e empresas que os comercializam. Também é feita uma breve descrição de algumas aplicações reais de esta tecnologia.

No capítulo 3 apresentam-se os aspectos e as definições teóricas para estudar a modelagem matemática (cinemática e dinâmica) de um robô manipulador de 6 graus de liberdade (robô escravo). Além do estudo matemático para o manipulador proposto, descrevem-se brevemente dois algoritmos eficientes para estudar e calcular sua dinâmica computacionalmente. Esses algoritmos são ferramentas necessárias para sua implementação no simulador virtual, sobre o qual podem-se aplicar diferentes esquemas de controle.

No capítulo 4 é apresentado um breve estudo e a descrição dos diferentes tipos de algoritmos de controle aplicáveis ao robôs manipuladores. Esses algoritmos de controle foram implementados para o robô estudado no capítulo 3 e serão comparados nas simulações feitas com os diferentes tipos controladores propostos. Isso permitirá escolher o mais apropriado para implementá-lo no simulador virtual.

No capítulo 5 descreve-se o dispositivo háptico Novint Falcon[®]. Mostra-se um estudo de sua modelagem cinemática. Dois Falcons são usados para construir mecanicamente um dispositivo de 5 graus de liberdade (robô mestre). Também estudam-se algumas estratégias para compensar os efeitos gravitacionais devido à massa adicionada na construção do dispositivo, implementando no algoritmo de controle.

No capítulo 6 são apresentados os conceitos básicos para integrar a interface háptica no ambiente virtual para interagir com o dispositivo háptico construído no capítulo 5. Descreve-se a utilização da biblioteca CHAI3D para visualizar e simular em tempo real os cálculos das forças de reação produzidos no ambiente virtual.

No capítulo 7 mostram-se os resultados experimentais com o dispositivo háptico construído. Além disso, apresenta-se a montagem do simulador virtual com um descritivo sobre o mesmo, além dos testes para o sistema, e finalmente discutindo os resultados obtidos nos ensaios realizados sobre a importância de sua utilização.

Finalmente, no capítulo 8 apresentam-se os comentários finais e a conclusão do trabalho, relatando os objetivos alcançados e sugerindo trabalhos que possam ser realizados tomando por base o projeto apresentado.

2

Sistemas de Interface Háptica

Neste capítulo, serão apresentados alguns conceitos básicos relacionados com Sistemas Hápticos e realidade virtual, bem como demonstradas algumas possíveis áreas de aplicação desta tecnologia. Além disso, serão discutidos alguns princípios de operação destes dispositivos hápticos, conjuntamente descrevendo também as vantagens, desvantagens e importância de sua aplicação.

2.1

Definições Prévias

A palavra háptica refere-se à capacidade de sentir um ambiente mecânico natural ou sintético através do tato, proveniente do grego "*háptō*" que significa próprio para tocar ou sensível ao tato. No ser humano, este sentido possui dois componentes independentes: o cutâneo e o cinestésico. O primeiro está ligado aos sensores localizados na superfície da pele, que são responsáveis por sensações tais como pressão, temperatura, vibração e dor. O segundo está ligado aos sensores localizados nos músculos, tendões e juntas que são responsáveis por sensações tais como movimento e força [10]. Ao tocar e mover um objeto qualquer, o indivíduo terá diversas sensações que serão convertidas pelo cérebro em informação cutânea e cinestésica [11]. Isto permite ao indivíduo inferir propriedades deste objeto, tais como textura (de acordo com a análise da informação cutânea) e características gerais, tais como peso e forma (de acordo com a análise da informação cinestésica). Um exemplo comum pode ser visto na figura 2.1

2.1.1

Sensações Táteis e Cinestésicas

As sensações táteis geralmente incluem pressão, textura, temperatura, maciez, umidade, além de sensações induzidas pela fricção como detecção de imperfeições nas superfícies dos objetos. Elas também incluem características dos objetos como forma e extremidades, além de sensações de vibração quando algum objeto oscila próximo à pele da pessoa. Ato simples, como segurar uma caneta, acarretam em várias dessas sensações. A qualidade e a textura



Figura 2.1: Exemplo de seis procedimentos para explorar um objeto .

do objeto são percebidas ao passar os dedos sobre a caneta, e a sua borda é encontrada através da exploração das extremidades (ou seja, identificação da forma). Segurar um telefone celular que vibra ao receber uma chamada causa sensações vibro-tácteis. A capacidade do ser humano de distinguir todas estas sensações provavelmente se deve a códigos neurais e a pequenos sensores (denominados mecano-receptores) presentes na pele conforme demonstraram Lamotte e Srinivasan[12].

Vários tipos de receptores foram encontrados para mediar as sensações descritas acima, na pele e nos tecidos subcutâneos; conseqüentemente, é costume designá-la como o órgão responsável pelo tato. De fato, existem centenas de receptores em cada centímetro quadrado dos mais de dois metros quadrados de área da pele de um adulto, que juntos formam o componente cutâneo do tato. As características biofísicas da pele, entretanto, mudam radicalmente de acordo com a parte do corpo que elas cobrem [13].

Devido à grande quantidade de sensores, o tato ocupa uma parte considerável do sistema nervoso central e periférico, como apontou Darian e Smith[14]. Já o componente cinestésico do tato refere-se à consciência do estado corporal, incluindo posição, velocidade e força, fornecidos pelos músculos através de uma grande variedade de receptores localizados na pele, juntas, ossos e tendões. Juntos, ambos os componentes do sentido do tato são essenciais para a manipulação e a locomoção.

2.1.2

Percepção Humana e as Interfaces Hápticas

Quando se assiste a um filme em alta resolução, não se percebem uma série de imagens estáticas que são apresentadas em sequência, muito menos detecta-se uma série de pixels coloridos juntos na tela. Em vez disso, é percebida uma cena visual que é bastante parecida com as experiências visuais experimentadas no cotidiano. Isto é possível porque a sensibilidade temporal da visão humana não é suficiente para detectar a mudança rápida de quadros no cinema, nem tem resolução suficiente para enxergar pixels individuais, caso esteja a uma distância normal da tela. Estes são exemplos de como a arquitetura e os limites do sistema sensorial humano podem ser explorados para construir sistemas engenhosos que possam oferecer experiências realistas e complexas. Exemplos destes sistemas incluem monitores de computador, televisões, gravadores de áudio, simuladores de voo e, obviamente, dispositivos hápticos.

O sentido do tato, entretanto, difere bastante da visão no sentido de que ele necessita de taxas de atualização muito maiores do que aquelas necessárias para assistir a um vídeo. Assim, em geral é muito difícil produzir uma interação háptica realista. Felizmente, mesmo quando um dispositivo háptico imperfeito é usado, o usuário se adapta rapidamente à sua atuação, ignora as suas imperfeições, e naturalmente associa os estímulos mecânicos criados pelo dispositivo às suas experiências do dia-a-dia, como perceber a textura de uma superfície e a forma dos objetos.

Além disso, quando as interfaces hápticas são combinadas com representações gráficas, o usuário prontamente associa o estímulo tátil ao objeto mostrado na tela. Isto acontece mesmo quando o que é visto e o dispositivo que gera feedback háptico estão localizados em locais bastante diferentes do espaço. Por exemplo, um jogador de jogos de corrida, que opta por diversas opções de câmera dentro do jogo, não deixa de associar o volante que vibra em suas mãos com o volante do veículo virtual que está sendo conduzido, muito embora ele possivelmente não o esteja vendo. Este cenário é ilustrado na figura 2.2.

Entretanto, se as imperfeições do dispositivo háptico são muito intrusivas, a sensação de realismo háptico é perdida. Um efeito parecido é o que ocorre quando um projetor de cinema diminui a taxa de atualização de quadros e passa a mostrar apenas um por segundo: o filme se transforma em uma série de fotografias. Assim, a qualidade da experiência háptica ilusória é uma função da combinação do sistema perceptivo do usuário e das qualidades técnicas próprias de cada interface, como resolução e adequação dos sinais sendo gerados.



Figura 2.2: Um jogo simulador de corrido de carros.

2.2 Realidade Virtual

Ackerman[15] afirma, em seu livro *A Natural History of the Senses*, que 70% dos receptores do sentido humano encontram-se nos olhos, tornando-os os grandes “monopolistas dos sentidos”. A maioria das informações recebidas pelo ser humano tem a forma de imagens visuais, as quais são interpretadas por um “computador” extremamente eficiente. Os computadores digitais, por sua vez, interpretam informações fornecidas por algum dispositivo de entrada de dados, como um teclado, por exemplo.

Realidade Virtual (RV) é o uso da tecnologia para criar a ilusão no usuário de que ele está em outra realidade, um novo meio de “estar” e “tocar” em informações ou seja, é um lugar onde os homens e os computadores fazem contato de forma natural. Refere-se a uma experiência imersiva e interativa baseada em imagens gráficas em 3D (três dimensões) geradas em tempo real por computador.

Uma definição um pouco mais refinada de realidade virtual é a seguinte: “realidade virtual é uma forma das pessoas visualizarem, manipularem e interagirem com computadores e dados extremamente complexos” [16]. Agrupando algumas outras definições de realidade virtual [17][18][19], pode-se dizer que a realidade virtual é uma técnica avançada de interface, onde o usuário pode re-

alizer imersão, navegação e interação em um ambiente sintético tridimensional gerado por computador, utilizando canais multi-sensoriais, que podem ser a forma mais avançada de interface do usuário de computador até agora disponível.

A interface com realidade virtual envolve um controle tridimensional altamente interativo de processos computacionais. O usuário entra no espaço virtual das aplicações e visualiza, manipula e explora os dados da aplicação em tempo real, usando seus sentidos, particularmente os movimentos naturais tridimensionais do corpo. A grande vantagem desse tipo de interface é que o conhecimento intuitivo do usuário a respeito do mundo físico pode ser transferido para manipular o mundo virtual. Para suportar esse tipo de interação, o usuário utiliza dispositivos não convencionais como capacete de visualização e controle, luva, e outros. Estes dispositivos dão ao usuário a impressão de que a aplicação está funcionando no ambiente tridimensional real, permitindo a exploração do ambiente e a manipulação natural dos objetos com o uso das mãos, por exemplo para apontar, pegar, e realizar outras ações. A figura 2.3 apresenta algumas aplicações utilizando a realidade virtual.

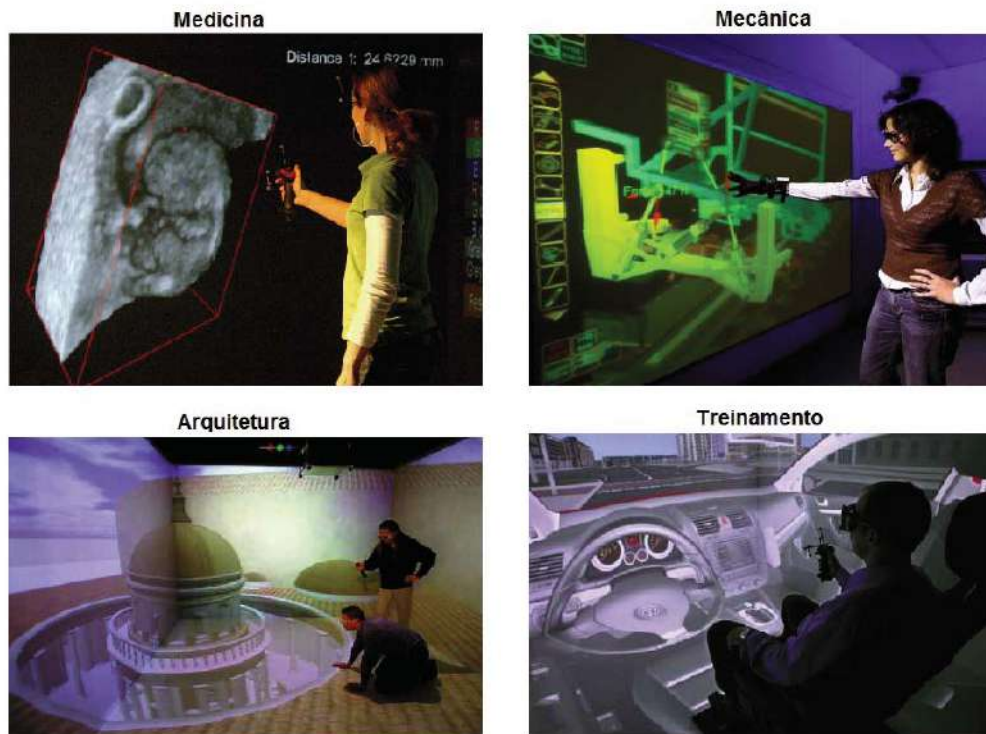


Figura 2.3: Exemplo de algumas aplicações da Realidade Virtual.

2.2.1

Dispositivos de Entrada e Saída

Entendemos a Realidade Virtual (RV) como uma ciência que engloba conhecimento de diversas áreas, como computação, eletrônica, robótica e cognição, dentre outras, visando oferecer sistemas computacionais que integram características de imersão e interatividade para simular ambientes reais. A utilização de dispositivos de interação e imersão utilizados em sistemas de RV está diretamente associada aos aspectos cognitivos do ser humano e pode explorar estímulos visuais, auditivos, táteis, motores e olfativos [20]. Esses dispositivos dividem-se em dois grupos: entrada e saída.

Os dispositivos de entrada são responsáveis pelo envio de informações de interação ou movimentação do usuário no sistema. Essas informações são utilizadas para o processamento de uma nova cena gráfica ou outra informação resultante da ação do usuário. Nesse grupo encontram-se: os sensores de trajetória, responsáveis por detectar a posição do objeto ou parte do corpo ao qual estão acoplados; os dispositivos de interação, como as luvas de dados (utilizadas no reconhecimento dos movimentos da mão), as bolas isométricas para movimentação com 6DOF (seis graus de liberdade); e os sensores biológicos, que usam reconhecimento de voz ou sinais elétricos musculares [21].

Os dispositivos de saída visam o estímulo dos sentidos do usuário, enviando para ele as respostas do sistema. Nesse grupo encontram-se os dispositivos visuais, auditivos, físicos e de locomoção. sendo que uma grande porção do cérebro é dedicada ao processamento das informações visuais [22]. Os dispositivos visuais e o tipo de imagem gerada pelo sistema de RV têm forte influência no nível de imersão do usuários. Os dispositivos visuais utilizados em RV podem ser monoscópicos ou estereoscópicos, permitindo ou não visualização das imagens apresentadas em três dimensões. No caso dos dispositivos estereoscópicos, cada exibição de imagem de uma cena é processada duas vezes. A distância existente entre os olhos humanos é utilizada para esse processamento, que resulta em duas imagens ligeiramente diferentes. Quando visualizadas individualmente por cada olho, essas imagens são reunidas no cérebro e resultam em uma visualização tridimensional. Exemplos de dispositivos visuais são os vídeo-capacetes, os *head-coupled displays* (visores montados sobre um braço mecânico com sensores de movimentação), os monitores convencionais ou auto-estereoscópicos, e os sistemas de projeção. No caso dos sistemas de projeção, estes são compostos por óculos polarizadores, obturadores ou filtros coloridos que permitem a visualização estereoscópica em monitores ou telas [21].

Outros dispositivos de saída são: os auditivos, sistemas de som 3D responsáveis pela exibição dos sons; as plataformas móveis, que interferem no equilíbrio do corpo e são utilizadas em simuladores de vôo e jogos eletrônicos; e os dispositivos físicos. Neste último caso, os dispositivos estimulam as sensações físicas por meio do retorno de forças de sensações de toque e de temperatura [23].

Assim, os Sistemas de Realidade Virtual (SRV) permitem que usuários visualizem, interajam e movimentem-se em três dimensões em um ambiente gerado em tempo real pelo computador [24]. Diferente dos sistemas computacionais convencionais, nos SRVs as informações recebidas pelo computador podem ser movimentos capturados do usuário por meio de dispositivos de leitura tridimensional associados a alguma parte do seu corpo ou outros sensores de movimento; o processamento das informações é feito em tempo-real e as respostas fornecidas ao usuário podem ocorrer por meio de dispositivos físicos, imagens tridimensionais ou sons [21].

O grau de imersão e a interatividade influenciam o nível de realismo de um SRV e são determinados pela qualidade dos gráficos apresentados, pelo desempenho dos dispositivos que apresentam esses gráficos, e pelo desempenho do sistema computacional no processamento dos gráficos e processos relacionados aos dispositivos de rastreamento de posição [24]. Outras tecnologias importantes para a RV estão relacionadas à sintetização de som, apresentação de reações por meio de dispositivos físicos e desenvolvimento de dispositivos específicos de interação e de técnicas de interação [24].

2.2.2

Princípios de Projeto e Operação

Uma das consequências do advento da RV foi a necessidade de redefinir o paradigma de interface homem-computador. O sistema tradicional mouse-teclado foi substituído por dispositivos que permitiram maior imersão do usuário no ambiente virtual e o manuseio de todas as potencialidades dessa nova tecnologia. O modo como os participantes interagem com o sistema de RV influencia enormemente suas experiências no ambiente virtual, facilitando seu uso, aumentando a sensação de imersão e ampliando a variedade de ações que se pode tomar dentro do ambiente virtual. Um importante dispositivo de interação é o rastreador de posição que pode ser utilizado para acompanhar a posição, do corpo e os movimentos do usuário, assim como a posição de outro objeto sendo por ele utilizado.

Há várias técnicas para criar sistemas hápticos, e uma variedade de dispositivos de rastreamento, cada um utilizado uma tecnologia diferente, entre

eles, os eletromagnéticos, mecânicos, acústicos, inerciais e ópticos. Ao analisar as tecnologias utilizadas pelos rastreadores, três fatores devem ser levados em consideração: precisão e velocidade de resposta do sensor; interferência do meio; restrições (fios, conexões mecânicas, etc.). Alguns desses dispositivos serão abordados nas seções seguintes.

Eletromagnéticos: Algumas características com relação aos dispositivos eletromagnéticos são (figura 2.4):

- Princípio de funcionamento: os rastreadores eletromagnéticos utilizam campos magnéticos para medir posição e orientação. O sistema é composto por transmissor e receptor em forma de bobina. Um sensor unidimensional para estimar a posição no eixo Z, por exemplo, é composto por uma única bobina transmissora orientada na direção Z. Quando uma corrente é aplicada à bobina, um campo magnético é gerado. No receptor, o campo induz uma tensão elétrica máxima proporcional à intensidade do campo magnético medido em uma bobina orientada na mesma direção do campo. A voltagem induzida fornece a distância do transmissor ao receptor, assim como a diferença de alinhamento entre os eixos.
- Precisão/Velocidade: esses sistemas são bastante precisos, cerca de 1 a 2 mm para posição e 0.1° para orientação. A velocidade de captura de dados é de 100 a 200 medidas/segundo.
- Interferência do meio: a presença de metais e o próprio tubo de raios catódicos do monitor podem causar interferência eletromagnética.
- Restrições: pequeno espaço de utilização devido ao alcance do campo magnético gerado. O receptor deve estar cerca 1-3 metros do transmissor.



Figura 2.4: Dispositivos Eletromagnéticos.

Mecânicos: Algumas características com relação aos dispositivos mecânicos são (figura 2.5):

- Princípio de funcionamento: os rastreadores mecânicos medem ângulos e distância entre juntas. Dada uma posição conhecida, todas as outras podem ser determinadas pela relação entre as juntas. Os rastreadores podem estar no chão ou anexadas ao corpo do usuário, usualmente na forma de um exoesqueleto. As rotações e as distâncias podem ser medidas por engrenagens, potenciômetros ou sensores de dobra.
- Precisão/Velocidade: por serem mecânicos, possuem alta precisão (0.1° de rotação). A latência média é de 200 ms.
- Interferência do meio: não sofrem interferência do meio.
- Restrições: a própria arquitetura do rastreador pode restringir o movimento do usuário caso o mesmo seja preso ao chão ou possua muitas juntas.



Figura 2.5: Dispositivos Mecânicos.

Acústicos: Algumas características com relação aos dispositivos acústicos são (figura 2.6):

- Princípio de funcionamento: rastreadores acústicos utilizam, tipicamente, ondas sonoras ultra-sônicas para medir distância. Os métodos mais usados são o cálculo do tempo de voo e a coerência de fase. Em ambos, o objetivo é converter tempo em distância. Um único par transmissor/receptor

fornece a distância do objeto em relação a um ponto fixo. O resultado é uma esfera em cuja superfície o objeto está localizado. A adição de um segundo receptor restringe a região a um círculo, e um terceiro receptor restringe a dois pontos, sendo um deles geralmente descartado. Portanto, para estimar a posição são necessários um transmissor e três receptores, ou um receptor e três transmissores. Para estimar posição e orientação, são necessários três transmissores e três receptores.

- Precisão/Velocidade: existe um atraso inerente à espera do sinal. Esse atraso é intensificado devido à relativamente baixa velocidade de propagação do som.
- Interferência do meio: as propriedades do som limitam esse método. O desempenho é degradado na presença de um ambiente ruidoso ou devido a geração de ecos. O som deve percorrer um caminho sem obstrução entre os altos-falantes e os microfones.
- Restrições: a configuração do sistema não é cara, pois o equipamento necessário é composto de microfones, alto-falantes e um computador. Devido às restrições de interferência, a distância média entre receptor e transmissor são alguns metros, contudo, sistemas mais precisos podem cobrir áreas de até 40x30m.



Figura 2.6: Dispositivos Acústicos.

Inerciais: Algumas características com relação aos dispositivos inerciais são (figura 2.7):

- Princípio de funcionamento: utilizam magnetômetros passivos, acelerômetros e girômetros. Os magnetômetros passivos medem o campo magnético do ambiente (geralmente da Terra) e fornecem medidas angulares. Os girômetros fornecem medidas angulares mais precisas e os acelerômetros fornecem medidas lineares. Todo são baseados na segunda lei do movimento de Newton, sendo assim, o sistema deve integrar a leitura para obter a velocidade e a posição.
- Interferência do meio: Não existe interferência, pois o sistema é "auto-contido", não havendo necessidade de um ponto externo para obtenção de dados.
- Restrições: Não existe limitação física para o espaço de trabalho, sendo o mesmo limitado somente pela conexão entre o dispositivo e o computador.

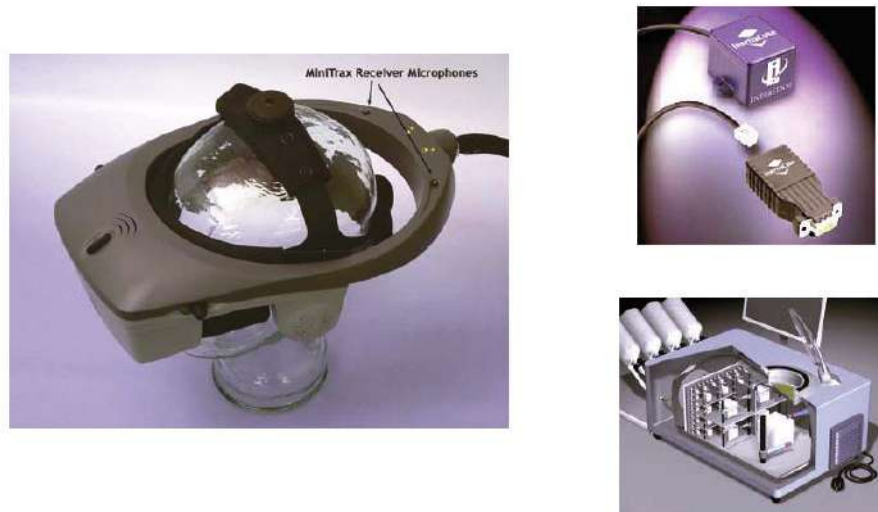


Figura 2.7: Dispositivos Inerciais.

Ópticos: Algumas características com relação aos dispositivos ópticos são (figura 2.8):

- Princípio de funcionamento: baseado na análise da projeção bidimensional de uma imagem ou na determinação dos ângulos de feixes da

varredura para calcular a posição e orientação de um dado objeto. Os sensores ópticos são geralmente câmeras (por exemplo, Charge-Coupled Device (CCD), um detector 4Q ou um diodo de efeito lateral). Um CCD é um conjunto de detectores recebendo imagens no plano da câmera. Um detector 4Q é um componente plano capaz de gerar sinais especificando o centro do feixe de luz que incide em sua superfície. Um diodo de efeito lateral é um componente que gera um sinal proporcional à posição da luz chegando em um eixo. Quando o sensor utilizado é uma câmera, técnicas de visão computacional devem ser utilizadas para determinar a posição do objeto. Se somente uma câmera for utilizada, é possível determinar um segmento de reta que passa pelo objeto detectado e pelo centro de projeção da câmera. Usando mais de uma câmera, podem-se determinar a posição e orientação do objeto.

- **Precisão/Velocidade:** a velocidade de captura depende muito do sensor empregado. Uma câmera padrão National Television Systems Committee (NTSC) consegue capturar imagens a taxas de 30 quadros por segundo, limitando a amostragem, enquanto câmeras digitais podem capturar a taxas de 200 a 1000 quadros por segundo. A precisão dos dados depende das técnicas de visão computacional empregadas: calibração de câmera, extração de informação da imagem e utilização de filtro para evitar tremidos.
- **Interferência do meio:** o laser e outros emissores podem refletir em objetos próximos, atrapalhando a medição.
- **Restrições:** a câmera deve estar sempre enxergando o objeto sendo rastreado e o emissor de luz não pode estar obstruído. Uma solução com três ou quatro câmeras oferece redundância que permite que uma ou duas sejam bloqueadas antes do sistema deixar de funcionar.

2.3

Sistemas de Interface Háptica

Sistemas hápticos oferecem interação com o ambiente virtual através de dispositivos e programas que permitem ao usuário sentir fisicamente as reações e os movimentos realizados no ambiente virtual. Os dispositivos fornecem sensação tátil e de força. A sensação tátil está ligada ao contato com o objeto, com sua textura e temperatura, enquanto a sensação de força é relacionada ao senso de posição e movimentação junto com as forças associadas ao movimento durante a interação com um objeto [25]. Os programas, por sua vez, são utilizados para controlar esses equipamentos, processando as informações



Figura 2.8: Dispositivos Ópticos.

recebidas do dispositivo e enviando para este a reação ao movimento feito no mundo virtual. Essa reação depende do movimento realizado, da posição e da intensidade com a qual a força foi aplicada sobre a superfície ou o objeto. Essas informações são essenciais para o cálculo da força de reação a ser enviada para o dispositivo e sentida pelo usuário.

Desta forma, com o uso de sistemas hápticos, é possível movimentar ou deformar algum objeto virtual - dependendo do material de que ele é feito - sentindo sua textura, peso, ou a força resultante deste movimento [26]. Para isso, o programa deve calcular precisamente as propriedades de força que o dispositivo deve retornar para o usuário. Se a força aplicada for excessiva, o usuário se cansará rapidamente e, se for muito fraca, as sensações não parecerão verdadeiras. O programador deve encontrar uma forma de calibrar e equilibrar essas forças, tornando a interface o mais realista possível [27]. Na figura 2.9 mostra-se um exemplo de sistemas hápticos.

2.4 Dispositivos Hápticos

Antes da disseminação do uso de computadores nos locais de trabalho, praticamente todas as atividades humanas envolviam o uso de habilidades motoras, como escrever e desenhar gráficos. As interfaces de computador,

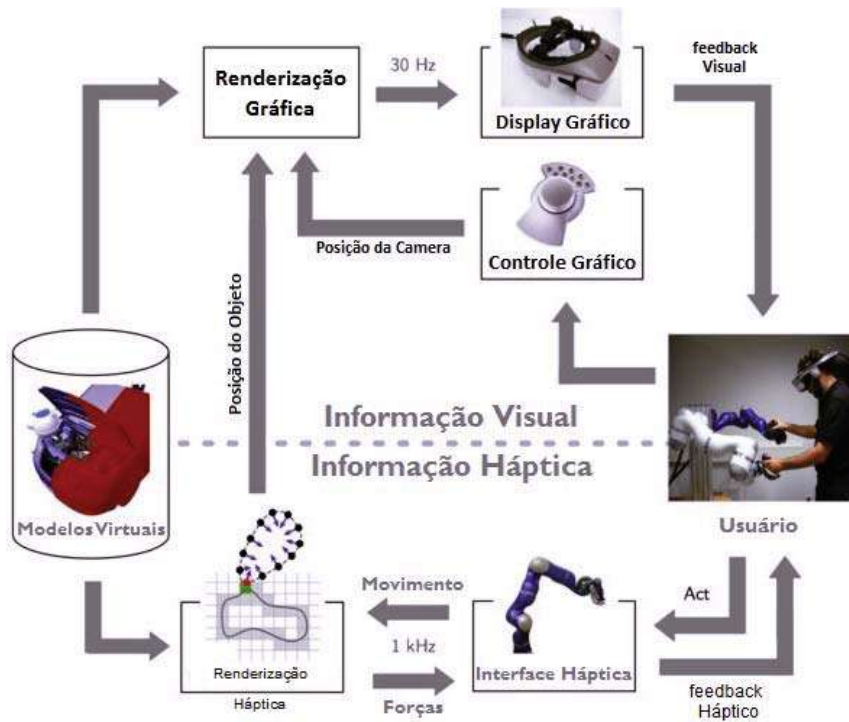


Figura 2.9: Exemplo de um Sistema Háptico

entretanto, não fizeram uso adequado destas capacidades fundamentais dos seres humanos. Com exceção de alguns dispositivos de entrada como o mouse ou joysticks, a interação entre o usuário e o computador depende de habilidades similares às necessárias para se usar uma máquina de escrever. Os dispositivos hápticos podem ser vistos como uma alternativa para sanar estas limitações.

Com eles é possível simular diferentes efeitos como: inércia, rigidez, colisão, textura, atrito, gravidade e força de reação. Também conhecidos como dispositivos de force-feedback (retorno de força), esses equipamentos recebem as características do movimento realizado pelo usuário (deslocamento, direção e aceleração) e enviam um resposta háptica a esse movimento.

2.4.1 Conceitos Básicos

Ao contrário das interfaces tradicionais, as interfaces hápticas geram sinais mecânicos que estimulam os componentes cutâneos e cinestésicos discutidos anteriormente. Além disso, elas dão aos usuários a capacidade de agir sobre o ambiente. Dessa forma, é possível definir os dispositivos hápticos como sendo interfaces homem-computador que associam gestos ao toque e à cinestesia, com o intuito de prover um meio de comunicação mais natural entre

homens e máquinas.

Uma característica fundamental desses dispositivos é a programabilidade. Este conceito refere-se à capacidade que os dispositivos hápticos têm de modificar suas propriedades mecânicas e físicas através de comandos provenientes do computador. Com isso, cria-se a possibilidade de uma troca bidirecional de energia (e conseqüentemente de informação) entre o usuário e o sistema. Uma maneira simples de entender melhor essa característica é comparar um mouse convencional com um mouse equipado com tecnologia háptica, como o WingMan da Logitech[28]. Com um mouse típico, o fluxo de informações tem apenas uma direção, do usuário para o computador. Dessa forma, o usuário praticamente não recebe informações sobre os seus movimentos, embora a inércia e a fricção do dispositivo com a superfície o auxiliem a executar os movimentos necessários. Os seus botões, ao contrário, são consideravelmente mais ricos do ponto de vista da interação, pois apresentam certa resistência e produzem um som característico para indicar que ocorreu uma mudança de estado. De qualquer forma, os botões não são programáveis.

O mouse háptico, por outro lado, pode dar ao usuário um feedback programável baseado no tato, permitindo uma interação mais rápida e mais intuitiva com a máquina. A Figura 2.10 ilustra ambos os cenários: do lado esquerdo, tem-se o fluxo de informações no caso do uso de um mouse convencional. Do lado direito, o uso do mouse háptico. Todos os objetos, naturais ou artificiais, podem ser inanimados ou animados. Os inanimados (ou inertes) podem apenas dissipar energia mecânica, enquanto os animados podem fornecer algum tipo de energia. Da mesma forma, podem existir dois tipos de dispositivos hápticos, convencionalmente denominados passivos ou ativos, apesar de ambos compartilharem o fato de serem programáveis.

Dispositivos passivos são geralmente projetados para terem uma dissipação programável, como função da posição ou do tempo. Pertencem a esta categoria dispositivos que possuem velocidade programável. Outra categoria de dispositivos passivos inclui aqueles capazes de modificar seu comportamento elástico, ou seja, modificar sua rigidez.

Dispositivos ativos são aqueles em que a troca de energia entre o usuário e a máquina é um fruto exclusivo da forma em que o retorno é fornecido. Assim, pode-se dividi-los em duas categorias. Nos dispositivos ativos isotônicos, os atuadores no dispositivo funcionam como uma fonte de força, e a posição é então medida. Nos isométricos, a posição é fornecida e a seguir a força é medida. Assim, percebe-se que, nos primeiros, a força que o dispositivo aplica não muda com a posição, enquanto que nos últimos a sua posição não muda com a força aplicada pelo usuário.

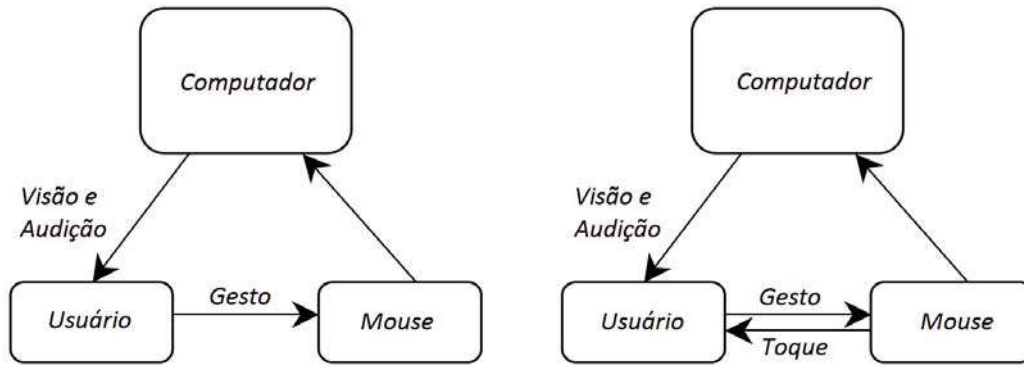


Figura 2.10: Fluxo de informações na interação com um mouse convencional (esquerda) e um mouse háptico (direita)

Normalmente, deseja-se que os dispositivos hápticos ativos sejam usados em ambientes artificiais, de tal forma que estes ambientes sejam passivos, como na simulação de uma cirurgia. O sucesso deste tipo de aplicação representa, porém, um desafio considerável [29]. Entretanto, a habilidade para criar uma situação ativa temporária pode ser bastante útil para aumentar o fluxo de informação entre a máquina e o usuário. Por exemplo, simular o comportamento de um volante em um jogo requer que a interação seja ativa, uma vez que dispositivos passivos não podem criar simulações. Finalmente, deve-se ter em mente que existe a possibilidade de interações instáveis entre ambientes passivos (rolar um tronco de madeira num ambiente virtual, por exemplo), caso o usuário não possua a energia necessária para executar a tarefa até o fim.

Em suma, independentemente da abordagem usada em um projeto, a bidirecionalidade é a característica que mais facilmente distingue os dispositivos hápticos, quando comparados com outras interfaces. Esta observação explica em parte a razão pela qual eles criam uma sensação de imediatismo no usuário, que foi bastante explorada nas primeiras aplicações desta tecnologia. Um dispositivo háptico precisa ser projetado para receber estímulos da mão humana (ou do pé, ou de qualquer outra parte do corpo) ao mesmo tempo em que gera estímulos para ela. A recepção de estímulos tem sido extensivamente explorada e vários tipos de dispositivos existem atualmente, como dispositivos apontadores, botões e joysticks. A geração de estímulos, no entanto, é consideravelmente mais difícil de ser alcançada de forma satisfatória.

2.5

Componentes dos Sistemas Hápticos

Uma interface háptica completa normalmente inclui um ou mais transdutores eletromecânicos (sensores ou atuadores) em contato com o usuário, de modo a aplicar sinais mecânicos a determinadas áreas do seu corpo e para medir outros sinais igualmente mecânicos nestas mesmas áreas.

Outra parte importante das interfaces hápticas completas é o sistema computacional que manipula os transdutores. A função deste sistema computacional é fornecer capacidades de “renderização” háptica, que são similares aos processos de renderização de imagens em sistemas gráficos. A primeira, entretanto, impacta na troca de informação bidirecional entre o dispositivo e o usuário [30]. A tarefa computacional neste tipo de renderização é gerar sinais que são relevantes a uma determinada aplicação. Várias abordagens existem para criar tal feedback. Por exemplo, um modelo pode ser usado para representar um ambiente, e suas equações solucionadas pelo computador, para encontrar as forças a serem aplicadas como uma função da disposição dos objetos. O modelo pode ser desenvolvido a partir de princípios básicos do cotidiano, ou parametrizado para representar apenas alguns aspectos desejados [31]. As características do tato humano permitem, em alguns casos, o uso de modelos físicos simplificados para gerar objetos virtuais que ofereçam estímulos táteis que competem em realismo com objetos verdadeiros, como foi demonstrado em [32].

2.5.1

Sensores

Em princípio, ao pensar em percepção háptica, quase de imediato nos vem à mente o termo sensor. Realmente vários trabalhos buscam o desenvolvimento destes dispositivos de forma que eles atendam à necessidade de cada aplicação. Bons trabalhos podem ser encontrados na literatura, dentre os quais podem ser citados: Rosa[33], Voyles[34], Bicchi[35], Mascaro[36], Fasse[37].

Em Rosa[33], foi proposto o desenvolvimento de um dispositivo denominado SCROLLIC, que consiste em duas garras paralelas com aderência, que têm como objetivo fornecer múltiplos contatos e permitir a manipulação de objetos de formas variadas, desde que suas dimensões estejam dentro do espaço de trabalho da garra. Em Voyles[34], foi desenvolvido um sensor tátil modular e um sistema de atuador para observar demonstrações de tarefas de contato. O sistema consiste em três partes: um sensor tátil intrínseco para medir força líquida/torque, um sensor tátil extrínseco para medir distribuições de contato, e um atuador tátil para exibir distribuições táteis. Os componentes

modernos são o sensor extrínseco e o atuador tátil que é "simétrico por dentro e por fora" para um ou outro e emprega um gel eletroreológico para atuação. A maior vantagem de usar o gel eletroreológico é que ele pode ser controlado eletricamente; o que é conveniente, pois requer pequena carga embora as tensões sejam muito altas, e ele possa ser feito de uma forma bem compacta. Mascaeo[36] trabalhou no desenvolvimento de um sensor para detectar força e toque, quando o operador tem a intenção de pressionar botões ou interruptores. Este sensor permite que dispositivos físicos normais como interruptores presos em paredes sejam substituídos por "dispositivos virtuais", que contêm circuitos elétricos e partes mecânicas, mas que são apenas desenhos. O desenvolvimento de dispositivos relacionados à simulação, como transmissão de sensações em ambientes virtuais, também é muito comum já que tais trabalhos precisam medir o quão real seja o ambiente simulado. Devido a este fato, Fasse[37] em seu trabalho propôs um modelo teórico com o objetivo de medir quantitativamente dispositivos de percepção háptica em ambientes virtuais.

2.5.2

Atuadores

São elementos que atuam sobre as grandezas físicas do processo, respeitando comandos, como motores, válvulas e aquecedores elétricos. A interface háptica é composta por um ou mais transdutores eletromecânicos (sensores ou atuadores) diretamente ligados ao usuário pois, através destes componentes conectados a determinadas partes do corpo, é possível captar e medir os sinais mecânicos. É composta também por um sistema computacional que efetiva a renderização háptica, que tem como finalidade a troca de informação bidirecional nos dispositivos hápticos e os objetos virtuais [38].

2.5.3

Modelagem da Cena Háptica

A modelagem da cena háptica em certos aspectos é semelhante a de uma cena gráfica (visual). A cena gráfica é formada basicamente pelo volume de visualização gráfico dentro do mundo virtual e pelos objetos virtuais dentro deste volume. De forma análoga, a cena háptica também define um volume dentro do mundo virtual, conhecido como volume háptico, e posiciona objetos dentro deste volume, sendo um dos objetos desta cena o próprio dispositivo háptico. Na cena háptica são irrelevantes as características visuais dos objetos, tais como cor e iluminação. Neste caso, os objetos devem apresentar propriedades materiais como aspereza, maciez e elasticidade, que permitam identificá-los a partir do toque. Assim, sem o auxílio da visualização, pode-se

identificar quando está em contato com um determinado objeto, sentir sua textura, variações superficiais e, até mesmo, temperatura [39].

Visualmente, o ser humano é capaz de perceber de forma suave e contínua as movimentações e alterações de um objeto pela tela do computador a uma frequência de 30Hz~60Hz. Para detectar as características físicas de um objeto, paralela e sincronizada em relação à visualização gráfica, as rotinas de cálculo da cena háptica necessitam ser executadas a uma frequência de 1000Hz [40], modificando a posição do dispositivo no ambiente virtual e enviando o sinal de retorno de força ou tátil ao usuário. Cada instante deste ciclo de cálculos na cena háptica é chamado de renderização háptica (figura 2.11).

Embora elas possam parecer drasticamente diferentes, todas têm dois elementos importantes em comum: um software para determinar as forças que resultam quando a identidade virtual do usuário interage com um objeto, e um dispositivo por meio do qual essas forças podem ser aplicadas ao usuário. O processo atual usado pelo software para realizar seus cálculos é chamado de renderização háptica. Um método comum de renderização usa modelos poliédricos para representar objetos no mundo virtual. Esses modelos 3D podem retratar uma variedade de formas e calcular o dado do toque ao avaliar como as linhas de força interagem com as várias faces do objeto. Tais objetos 3D podem ser feitos para parecerem sólidos e podem ter textura.

2.5.4 Renderização Háptica

Renderização háptica pode ser definida como o processo pelo qual as rotinas de controle háptico calculam as modificações na cena háptica, atualizando-a e enviando estas modificações ao usuário em tempo real [40].

O processo de renderização háptica pode ser descrito da seguinte forma: o usuário manipula o dispositivo háptico cuja representação visual pode ser vista na cena gráfica, o qual denominará de cursor ou ponto de interação. À medida que o usuário move o ponto de interação ao longo da cena, sua posição e orientação são detectadas pelos codificadores do dispositivo; quando há uma colisão com um objeto virtual, esta deve ser detectada imediatamente, passando a atuar o algoritmos de cálculo de retorno de força; quando o ponto de interação colidir com objetos, os algoritmos calculam a força de reação baseado na profundidade de penetração. Os vetores de força podem então ser modificados de forma a representar o tipo de superfície que se deseja renderizar [39]. De modo simplificado, as principais etapas da renderização háptica são:

1. Localização do ponto de interação no ambiente virtual.

2. Detecção das colisões entre o ponto de interação e a geometria dos objetos virtuais.
3. Cálculo do vetor de força de reação (baseado nas propriedades materiais definidas do ambiente virtual).
4. Envio do vetor de força ao dispositivo háptico.
5. Retorna passo 1.

Como mencionado anteriormente, esses passos da renderização háptica precisam ser executados a uma frequência de 1000Hz, permitindo uma percepção mais natural das características dos objetos pelo usuário. Durante o passo 3, o modo como estas forças são computadas pode produzir diferentes efeitos sobre o objeto virtual. Pode-se, por exemplo, renderizar superfícies de objetos rígidos, flexíveis, ásperos e etc.

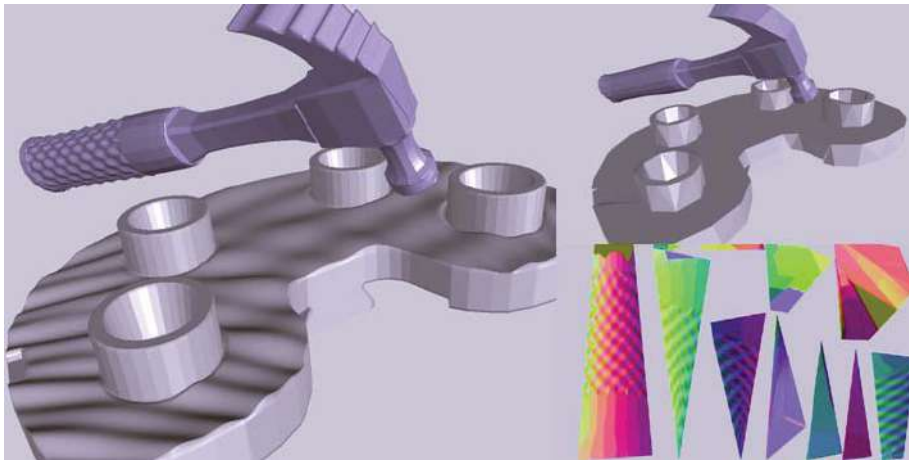


Figura 2.11: Cena de uma Renderização Háptica

2.6

Principais Dispositivos Hápticos Comerciais

O trabalho de transferir imagens hápticas para o usuário é do dispositivo de interface. Em muitos aspectos, o dispositivo de interface é análogo ao mouse, exceto que o mouse é um dispositivo passivo que não pode comunicar dado háptico sintetizado para o usuário. Apresentam-se a seguir alguns sistemas hápticos específicos para entender como esses dispositivos funcionam.

2.6.1 Phantom Omni

Desenvolvido pela SensAble Technologies, o Phantom Omni é compacto e portátil. Sua interface principal de manipulação e force-feedback consiste de um braço mecânico terminado em uma peça semelhante a uma caneta, como pode ser visto na figura 2.12. Sua estrutura adequa-se a simulações de procedimentos médicos nos quais se utilizam ferramentas como seringas, agulhas, bisturis dentre outros instrumentos similares [41].

As informações captadas são os movimentos de translação e rotação nos eixos x , y e z , por isso ele é considerado equipamento com seis graus de liberdade. Além disso, este tipo de dispositivo oferece um mecanismo de force-feedback nos eixos x , y e z permitindo, desta forma, uma interação entre o usuário da simulação e a aplicação [42].

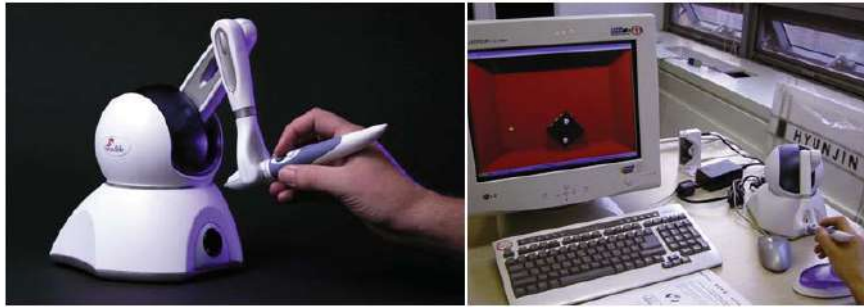


Figura 2.12: Dispositivo Háptico Phantom-Omni

2.6.2 Delta.6

Desenvolvido pela empresa Force Dimension, o Delta.6 é uma interface de alto desempenho com manipulação force-feedback, baseado em um manipulador paralelo delta. Ele oferece seis graus de liberdade compreendidos em translação e rotação, e além disso oferece uma área de trabalho maior do que os dispositivos comuns. Graças à sua configuração paralela, o Delta.6 pode transmitir grandes forças contínuas e torques em qualquer lugar de sua área de trabalho.

Sua rigidez mecânica, combinada com seu controlador em tempo real por USB 2.0, permite uma representação elevada de forças de contato a uma taxa de 4KHz. Assim, para proporcionar um maior grau de transparência háptica, há compensação de gravidade, mantida no espaço de translação e rotação, acoplado com componentes passivos e atuadores. É uma solução versátil para aplicações que exigem grandes forças e grande espaço de trabalho, mas o custo

é elevado em comparação aos dispositivos descritos anteriormente. A figura 2.13 mostra este dispositivo.



Figura 2.13: Dispositivo Háptico Delta.6 Force Dimension

2.6.3 Sigma.7

O Sigma.7 é a interface háptica mestre mais avançado desenvolvido pela empresa Force Dimension. O Sigma.7 introduz sete graus de liberdade, e além disso possui uma capacidade de alta precisão háptica para agarre. Calibrado para uma excelente compensação de gravidade, a força e torque no efetuador terminal oferecem um desempenho extraordinário tátil, permitindo a interação instintiva com aplicações táteis complexas.

Projetado para a indústria aeroespacial avançada e indústrias médicas, permite um elaborado controle para robôs hábeis. Seu desenho ergonômico, disponível na configuração esquerda e direita, pode ser integrado em uma configuração de estação de trabalho dual para operação bimanual. A figura 2.14 mostra este dispositivo.



Figura 2.14: Dispositivo Háptico Sigma.7 Force Dimension

2.7

Principais Aplicações com Dispositivos Hápticos

Diversos projetos têm comprovado que a utilização de dispositivos hápticos em aplicações na saúde, para a educação, e treinamento militar, melhora o nível de percepção e aprendizado efetivo, ampliando a sensação de imersão sentida pelo usuário.

2.7.1

Histórico

Os desenvolvimentos que deram origem aos sistemas de realidade virtual começaram nos Estados Unidos com a construção de simuladores de voo após a Segunda Guerra mundial. A indústria do entretenimento também teve um papel bastante importante no surgimento da realidade virtual, com o simulador Sensorama em 1958: uma espécie de cabine onde filmes 3D eram projetados e combinados com som estéreo, vibrações mecânicas, aromas e ar movimentado por ventiladores, fornecendo ao usuário/participante uma viagem multisensorial [43].

Em 1963, Ivan Sutherland publicou sua tese de doutorado e descreveu como os computadores poderiam ser usados para exibir gráficos interativos. Mais tarde ele apresentou o primeiro vídeo-capacete que permitia ao seu usuário a visualização de um gráfico, representado em estrutura de arame, através de pequenos monitores de tubo de raios catódicos posicionado diretamente diante dos olhos [44]. A partir de então, as pesquisas se intensificaram e em 1982 os simuladores de voo da força aérea americana já utilizavam vídeo-capacete com integração de áudio e vídeo, e permitiam movimentos com seis graus de liberdade [45].

A primeira luva de dados, dispositivo capaz de capturar os movimentos da mão e inclinação dos dedos do usuário e transmiti-los ao computador, surgiu comercialmente em 1985. Em 1986, a NASA possuía um ambiente virtual que permitia aos usuários ordenar comandos de voz, escutar fala sintetizada e som 3D, além de manipular objetos diretamente com os movimentos das mãos utilizando luva de dados machado[21].

Diversas pesquisas, como as mencionadas acima, levaram à conscientização de que produtos de realidade virtual podiam ser comercializáveis, o que ocasionou a criação de empresas de equipamentos e sistemas para realidade virtual, dando início a diversos programas de pesquisa nessa área no mundo inteiro. Em 1989, a empresa Autodesk Inc. apresentou o primeiro sistema de realidade virtual baseado em um computador pessoal [45].

Com o desenvolvimento tecnológico dos últimos anos, a realidade virtual está sendo utilizada para os mais diversos fins na diversas áreas da ciência, Na última década, aplicações médicas utilizando realidade virtual passaram a ser desenvolvidas, tornando essa área comercialmente e clinicamente importante em termos de tecnologia aplicada à medicina [46].

Na figura 2.15, pode-se observar um dos últimos projetos desenvolvidos pela empresa Toyota, tratando-se de um simulador de treinamento para dirigir automóveis, sendo o sistema mais avançado até o momento. Este projeto tem um automóvel de tamanho real localizado na frente de uma tela de vídeo em 360 graus. As sensações ao dirigir dentro do simulador são de: aceleração, velocidade, mudança de marcha, forças centrífugas nas curvas, e outras manobras que estão presentes quando dirigimos um automóvel real. Além disso, o simulador completa com som 3D, promovendo uma experiência totalmente realista.



Figura 2.15: Simulador de automóveis desenvolvido pela empresa Toyota

2.7.2 Medicina

O uso de interfaces hápticas na criação de simuladores voltados para o treinamento de profissionais da área de saúde tem recebido atenção especial por parte dos pesquisadores da área. A principal motivação é o fato de que simulações não colocam a vida de pacientes em risco, além de serem executadas em um ambiente controlado, sendo possível criar situações específicas de

acordo com o objetivo do treinamento. Além disso, a tecnologia pode melhorar consideravelmente a experiência dos cirurgiões que operam ou diagnosticam pacientes à distância.

Sela[47] apresentou um simulador cirúrgico (figura 2.16) que permite aos médicos praticarem e aprimorarem seus conhecimentos em um ambiente virtual antes de entrar em uma sala de cirurgia. Foi utilizado um algoritmo batizado de FEM-DFFD, de baixa complexidade computacional, que permite execução em tempo real, podendo manter uma precisão razoável. O resultado obtido nesse projeto mostrou que o sistema 4D DFFD pode ser usado para simular em tempo real incisão em um modelo 3D usando um dispositivo háptico.

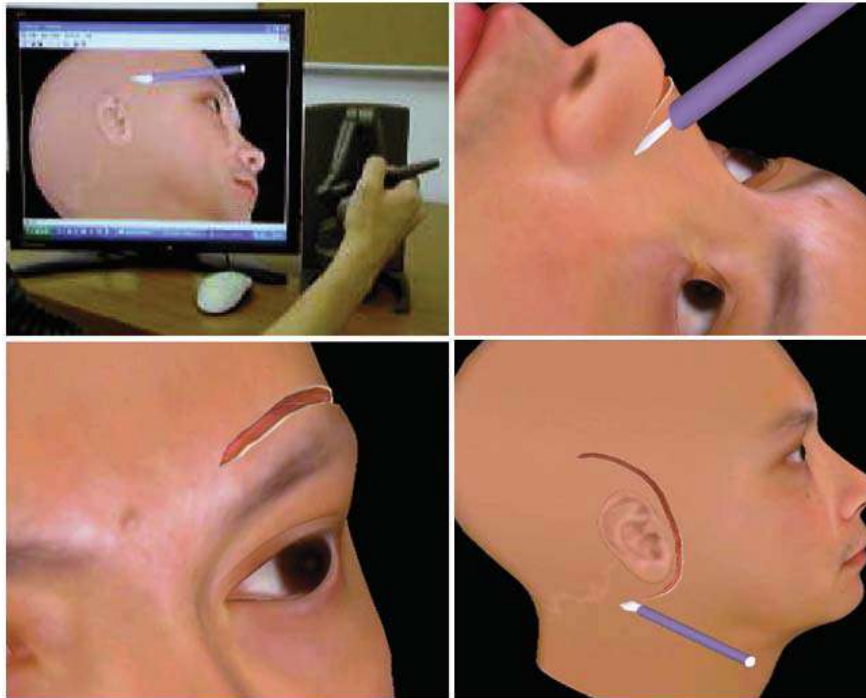


Figura 2.16: Simulador Cirúrgico Sela[47]

D'Aulignac[48] desenvolveu uma tese sobre um simulador que permite a estudantes da área médica praticarem diagnóstico de trombose em um ambiente virtual (figura 2.17). Foi utilizada a ferramenta Echography, um modelo de membro inferior, e um dispositivo háptico para realizar os testes. O resultado obtido nesse projeto mostrou que é possível utilizar um simulador ecográfico juntamente com um braço robótico para diagnosticar a trombose.

A tese de Blyth[49] relata o desenvolvimento de um simulador cirúrgico e um componente de avaliação. O simulador permitiu a simulação da fixação de implante fratura do quadril (figura 2.18). Foi utilizado um quadril virtual utilizando imagens radiográficas para orientar a redução da fratura e colocação

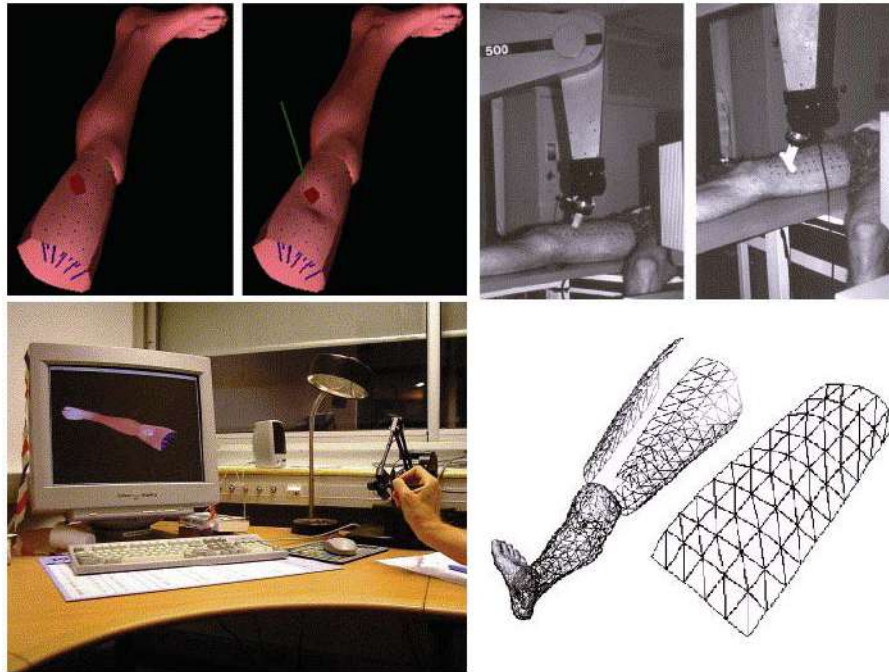


Figura 2.17: Diagnóstico de Trombose [48]

do implante. Os testes foram feitos com dez participantes do estudo. Segundo os autores, os resultados mostraram que o simulador atenderia o problema proposto e a maioria das pessoas afirmaram que o simulador forneceu uma vis ao realista do ambiente operacional, e que a visão tridimensional fornecida era necessária.

Entretanto, apesar das promissoras potencialidades que este tipo de aplicação parece oferecer, existem poucos testes sistemáticos para simuladores como os descritos acima.

2.7.3 Educação

Os sistemas hápticos conjuntamente com a realidade virtual (RV) e a realidade aumentada (RA) contribuem de maneira significativa na área da educação como processo de exploração, descoberta, observação e construção de uma nova visão do conhecimento, oferecendo ao aprendiz a oportunidade de melhor compreensão do objeto de estudo. Essas tecnologias, portanto, têm potencial de colaborar no processo cognitivo do aprendiz, proporcionando não apenas a teoria, mas também a experimentação prática do conteúdo em questão [50]. A introdução da RA na matemática, por exemplo, pode eliminar uma das principais dificuldades do aluno: visualizar um problema complexo de geometria. Ela fornece a professores e estudantes um método intuitivo e colaborativo

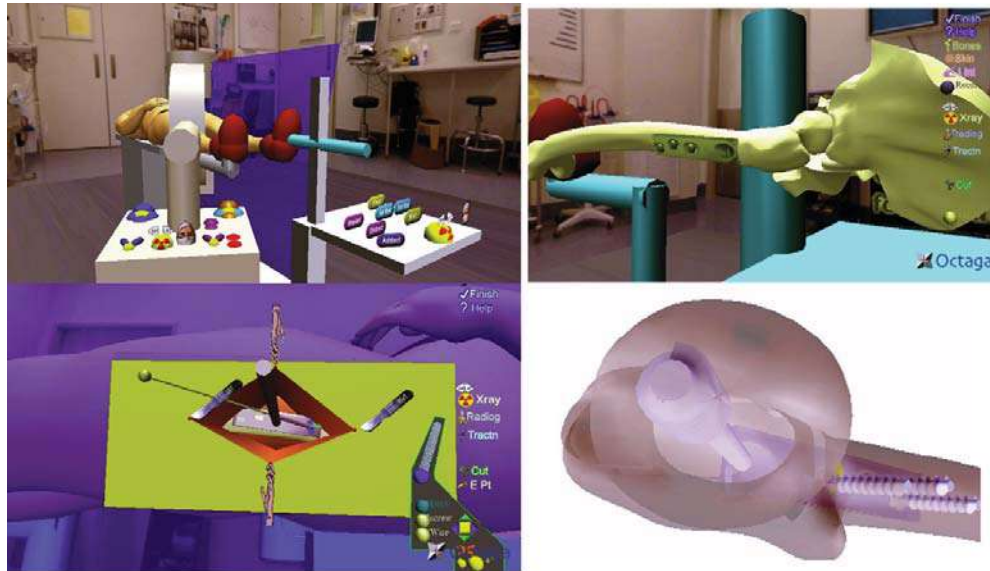


Figura 2.18: Simulação da fixação da fratura do Quadril [49]

de aprender. Os alunos vêem os objetos tridimensionais, sem precisar imaginá-los ou desenhá-los numa folha de papel. A tendência da educação é fomentar o entendimento da aprendizagem como processo individual e complexo, e sustentar-se na visão global do mundo, fruto das vivências do indivíduo, inter-relacionada com o desenvolvimento de capacidades ligadas à observação, análise, planejamento, decisão, aplicação e avaliação. Para isso, os sistemas hápticos pode revelar-se como recurso potente, uma vez que propiciam a visualização, interação e resposta em tempo real. Na figura 2.19 são mostrados alguns exemplos.

2.7.4 Militar e Aeroespacial

Dispositivos hápticos também têm sido utilizados em simulações de treinamento de militares e de astronautas. Existem várias circunstâncias no contexto militar nas quais eles podem agir como fontes de informação alternativas. Ou seja, há situações em que o toque poderia fornecer informação que por alguma razão não esteja disponível ou não seja precisa quando proveniente da visão ou da audição. Em alguns casos, combatentes têm sua visão bloqueada ou podem não poder desviar a visão de algum ponto específico para consultar outra fonte de informação. Além disso, as condições no campo de batalha podem exigir que as comunicações sejam inadiáveis. Em cenários como os descritos acima, a tecnologia háptica poderia ser usada como



Figura 2.19: Sistemas Hápticos para a Educação

modalidade de comunicação alternativa ao som ou imagens, em situações em que informações simples como perigo iminente ou ordens para deslocamento precisem ser transmitidas. A figura 2.20 mostra diferente tipos de treinamento militar utilizando estes sistemas hápticos.



Figura 2.20: Simuladores para treinamento militar

Provavelmente, o simulador militar mais conhecido é o de voo. A Força Aérea, o Exército e a Marinha usam simuladores de voo para treinarem seus pilotos. As missões de treinamento podem incluir: como voar em batalha, como

se reestabelecer em caso de emergência, ou como coordenar a sustentação no ar com operações terrestres. Alguns simuladores de voo incluem um módulo completamente fechado, enquanto outros simplesmente têm um conjunto de monitores de computador organizados de modo a cobrir o campo de visão do piloto. Alguns exemplos destes simuladores são mostrados na figura 2.21.



Figura 2.21: Simulador de aviões militares

Os simuladores de RV de veículos terrestres são uma parte importante da estratégia do exército. Na verdade, os simuladores são uma peça fundamental do FCS (Futuro Sistema de Combate). O FCS consiste em um sistema de comando de batalha via rede, além de veículos avançados e plataformas de defesas. Os cientistas da computação criaram os simuladores de FCS para serem usados juntos em uma rede, facilitando as complexas missões de treinamento que envolvem vários participantes desempenhando vários papéis.

Os simuladores podem ser muito caros. O simulador do Stryker mostrado na figura 2.22, o mais moderno, custa cerca de R\$ 1.720.000 por unidade. Entretanto, quando se compara o custo desse equipamento ao de um veículo real de verdade (que, dependendo do modelo, pode chegar a milhões de reais), e se tem em mente que o soldado por trás dos controles estará seguro, é fácil justificar o custo.

Hoje, muitas instalações de treinamento usam os simuladores para que os soldados se acostumem com táticas de combate urbano. Os campos de ba-



Figura 2.22: Simulador do veículo terrestre Stryker

talha atuais são muito diferentes dos que existiam antigamente, com soldados se aventurando em cidades em vez de estabelecerem linhas de batalha tradicionais. Os simuladores dão ao exército a chance de ensinar aos soldados navegarem e trabalharem efetivamente dentro de cenários urbanos sem ter que construir um ambiente artificial físico.

3

Modelagem do Sistema Escravo

Este capítulo tem como foco o desenvolvimento do modelo matemático do robô manipulador Titan IV, baseado em um completo modelagem cinemático e dinâmico representado por equações que expressam restrições ou que limitam a possibilidade de movimento dos corpos que constituem o sistema. Este manipulador, muito usado em intervenções submarinas, foi escolhido para representar o sistema escravo a ser controlado no ambiente virtual desenvolvido neste trabalho.

A modelagem cinemática de um robô manipulador é o estudo da posição e da velocidade do seu efetuador e dos seus elos. Quando se menciona posição, está se referindo tanto à posição propriamente dita, como à orientação, e quando se fala em velocidade, considera-se tanto a velocidade linear como angular. Podem-se distinguir dois tipos de cinemática, a cinemática direta e a inversa. A modelagem dinâmica (modelagem matemática) de um robô manipulador desempenha um papel preponderante na simulação do movimento, na análise da estrutura do manipulador e no projeto dos algoritmos de controle. Ele fornece uma descrição da relação entre as forças generalizadas (forças e torques) aplicadas nas juntas e o movimento do manipulador [51].

3.1

Descrição do Sistema

O robô Schilling Titan IV é amplamente conceituado como um dos sistemas manipuladores remotos servo-hidráulicos [52]. Desde 1987, os sistemas TITAN foram o padrão da indústria para sistemas de manipuladores hábeis usados em aplicações submarinas, e são extensamente usados em ROVs (veículos de operação remota) de manutenção ultra pesados. O Titan IV tem a destreza e precisão necessárias para desempenhar movimentos suaves para tarefas complexas. Quando esta capacidade é combinada com o alcance do sistema de manipulação (1,922m, ou 75,7 pol.), capacidade de carga (122kg ou 270lb em extensão completa), classificação de profundidade (disponível até 7.000m), e grande envelope operacional, o Titan IV oferece um desempenho sem igual em uma ampla gama de aplicações submarinas. O robô está configurado com an-

gulação de ombro, do cotovelo, rotação do braço inferior, angulação do punho, rotação de garra e abertura e fechamento de garra. A movimentação é dirigida por válvulas hidráulicas, atuadores rotatórios e motores hidráulicos, possuindo seis graus de liberdade. Sua operação é feita por meio da manipulação de uma réplica miniatura do manipulador, visualizada na figura 3.1.



Figura 3.1: Robot Schilling Titan IV e miniatura utilizado como "mestre" na teleoperação [52].

3.2 Cinemática Direita

O problema de cinemática direta de robôs manipuladores consiste em determinar a localização do efetador ou garra (e eventualmente de cada elo do manipulador) a partir do valor atual das variáveis das juntas (ângulos ou deslocamentos), o qual pode ser resolvido para qualquer robô manipulador serial utilizando transformações que relacionam a localização de um elo em relação ao anterior. A solução da cinemática direta é única e pode ser obtida tanto na forma analítica quanto na forma numérica através de um procedimento sistemático.

3.2.1 A Notação de Denavit-Hartenberg

A relação cinemática entre um par de elos adjacentes envolvido em uma cadeia cinemática aberta, é descrito pela notação Denavit-Hartenberg que introduz um método sistemático para descrever a relação cinemática. O método é baseado em uma matriz 4×4 que representa a posição e orientação de um corpo rígido no espaço.

A figura 3.2 representa um par de elos adjacentes, elo_{i-1}, elo_i e suas juntas associadas $i - 1, i, i + 1$. A posição e orientação relativa entre os dois elos é descrita pelas transformações de translação e de rotação entre os dois sistemas de coordenadas fixos a estes elos. O primeiro passo para definir os

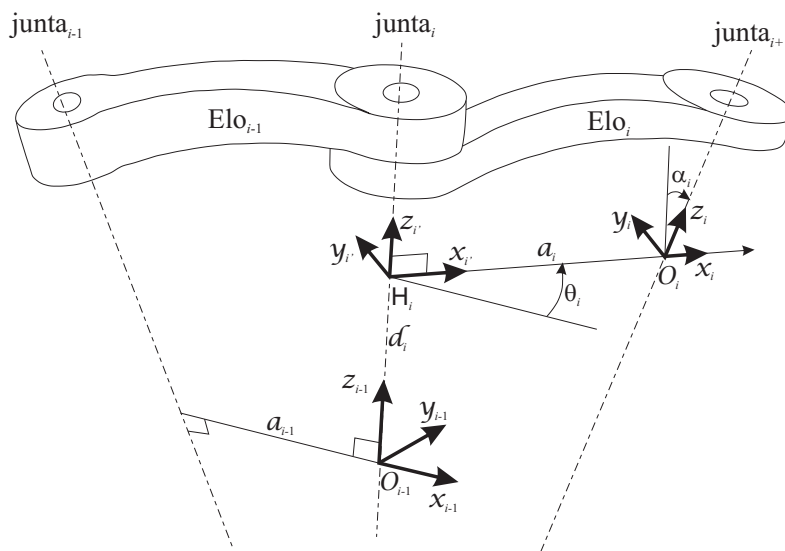


Figura 3.2: A notação Denavit-Hartenberg [51].

sistemas de coordenadas de um robô é localizar os eixos z ao longo dos eixos das articulações, de forma que o eixo z_{i-1} é o eixo da junta i . Seja a reta $H_i O_i$ a normal comum aos eixos das juntas i e $i + 1$ (eixos z_{i-1} e z_i). A origem do sistema O_i é localizada na intercessão do eixo da articulação $i + 1$ (eixo z_i) e a normal comum entre os eixos z_{i-1} e z_i . O eixo x_i é direcionado ao longo da extensão desta normal comum, na direção de z_{i-1} para z_i . Finalmente, o eixo y_i é escolhido de forma que o sistema resultante $O_i - x_i y_i z_i$ seja um sistema de coordenadas que segue a regra da mão direita.

A posição relativa entre dois sistemas de coordenadas consecutivos, sistemas $O_{i-1} - x_{i-1} y_{i-1} z_{i-1}$ e $O_i - x_i y_i z_i$ é completamente determinada pelas posições relativas entre os eixos x_{i-1} e x_i , e entre os eixos z_i e z_{i-1} , que são definidas pelos quatro parâmetros seguintes:

- a_i : é a distância (em módulo) entre z_{i-1} e z_i , medida ao longo do eixo x_i , que é a normal comum entre z_{i-1} e z_i , ou seja, é a distância $H_i O_i$.
- d_i : é a distância (com sinal) entre os eixos x_{i-1} e x_i , medida sobre o eixo z_{i-1} (que é a normal comum entre x_{i-1} e x_i), partindo-se de O_{i-1} e indo em direção à H_i . O sinal de d_i é positivo se, para ir de O_{i-1} até H_i , caminha-se no sentido positivo de z_{i-1} , e negativo se caminha-se no sentido oposto de z_{i-1} .
- α_i : é o ângulo (com sinal) entre o eixo z_{i-1} e o eixo z_i , medido em torno do eixo x_i , segundo a regra da mão direita, ou seja, é o ângulo de rotação em torno do eixo x_i , que o eixo z_{i-1} deve girar para que fique paralelo ao eixo z_i .
- θ_i : é o ângulo (com sinal) entre eixo x_{i-1} e o eixo x_i , medido em torno do eixo z_{i-1} segundo a regra da mão direita, ou seja, é o ângulo de rotação em torno do eixo z_{i-1} , que o eixo x_{i-1} deve girar para que fique paralelo ao eixo x_i .

Vamos formular a relação cinemática entre os elos adjacentes utilizando matrizes homogêneas. Usando a propriedade de equivalência pode-se representar a posição do sistema de referência i relativo ao sistema de referência $i - 1$ com transformação de coordenadas.

$$\mathbf{A}_i^{i-1} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-1)$$

onde a matriz \mathbf{A}_i^{i-1} representa a posição e orientação relativa entre os sistemas de referências associados a dois elos sucessivos. Observe que esta matriz do sistema de referência i para o sistema de referência $i - 1$ é uma função apenas da variável de junta i , isto é, a_i para a junta de revolução ou d_i para junta prismática. Os três primeiros vetores coluna 3×1 da matriz \mathbf{A}_i^{i-1} indicam a direção dos três vetores unitários do sistema de referência i em relação ao sistema de referência $i - 1$, enquanto a última coluna especifica a posição do sistema de coordenada O_i em relação ao sistema de referência $i - 1$.

A convenção de Denavit-Hartenberg permite construir a função de cinemática direta pela composição das transformações de coordenadas diretas expressa pela equação anterior dentro de um matriz de transformação de coordenada que descreve a posição e orientação do sistema de referência n em

relação ao sistema de referência da base dada por:

$$\mathbf{T} = \mathbf{A}_1^0(\mathbf{q}_1)\mathbf{A}_2^1(\mathbf{q}_2)\mathbf{A}_3^2(\mathbf{q}_3)\dots\mathbf{A}_n^{n-1}(\mathbf{q}_n) \quad (3-2)$$

onde a matriz \mathbf{T} representa a posição e orientação da ferramenta do robô relativa ao sistema de referência da base, como se apresenta na figura 3.3.

Como esperado, o cálculo da função da cinemática direta é recursiva e é obtida de uma maneira sistemática pelo simples produto de contribuição de cada grau de mobilidade. Este procedimento pode ser aplicado em qualquer cadeia cinemática aberta, e pode ser facilmente reescrito em uma operação formal.

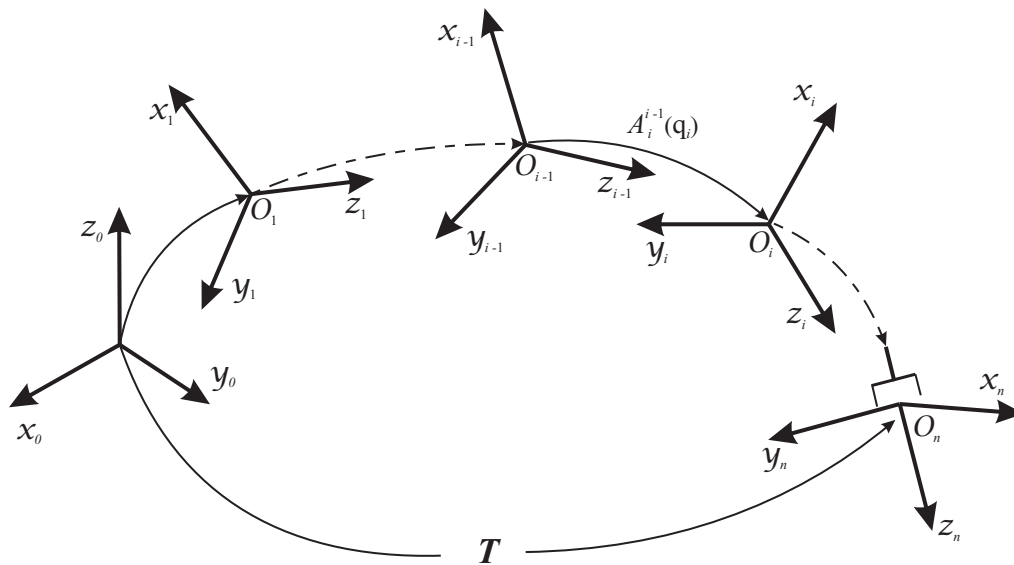


Figura 3.3: Posição e Orientação do efetuador terminal representado pela matriz \mathbf{T} [51].

3.2.2 Modelagem de Denavit-Hartenberg

O robô Titan IV está constituído por um grupo de corpos rígidos, ou elos. Cada elo se conecta a dois outros membros, logo cada elo tem dois eixos, como apresenta a figura 3.4. Também mostra-se o manipulador com seus seis sistemas de coordenadas. A seguir obtêm-se parâmetros de Denavit-Hartenberg para cada sistema de coordenadas.

A Tabela 3.1 apresenta os parâmetros Denavit-Hartenberg (DH) para o robô Titan IV, e a partir deles podem-se encontrar as matrizes A_i que calculam a posição da extremidade final do manipulador, explicitadas no Apêndice A.

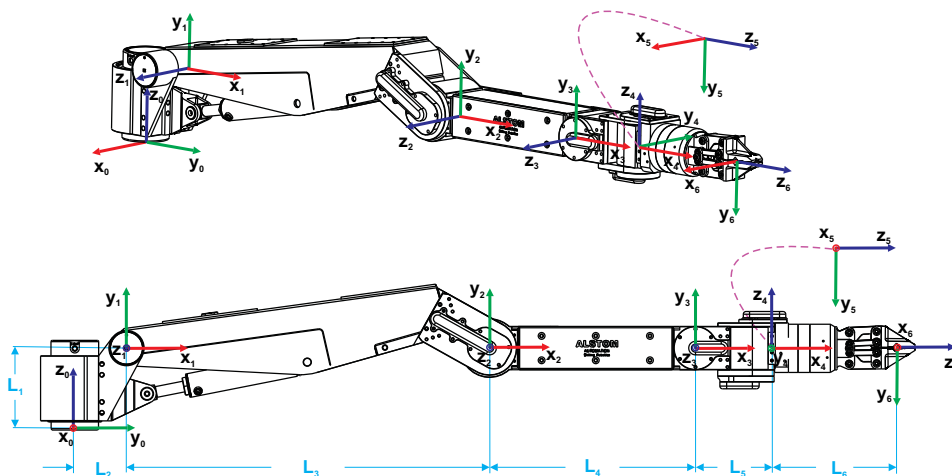


Figura 3.4: Robô Schilling Titan IV e seu sistema de coordenadas.

Tabela 3.1: Parâmetros do robô Manipulador Schiling Titan IV

Elo_i	α_i	a_i	d_i	θ_i
1	90°	l_2	l_1	θ_1
2	0°	l_3	0	θ_2
3	0°	l_4	0	θ_3
4	-90°	l_5	0	θ_4
5	-90°	0	0	$\theta_5 - 90^\circ$
6	0°	0	l_6	θ_6

Uma vez calculada a matriz homogênea \mathbf{T} , com qualquer ângulo dado nas juntas, pode-se obter a posição e orientação do efetuador terminal do robô.

3.3 Cinemática Inversa

O problema da cinemática inversa é mais difícil de resolver que o problema da cinemática direta, pois não há um procedimento sistemático explícito como o algoritmo D-H. Na cinemática inversa, procura-se determinar, a partir da posição e orientação do efetuador terminal, os ângulos das juntas correspondentes.

Um resultado disto é que cada robô, ou classes de robôs, tem que ser tratado separadamente. No entanto, a solução para o problema da cinemática inversa é mais útil que a cinemática direta, pois no controle da trajetória do robô a cinemática inversa se faz necessária para escolha da melhor configuração das juntas para um movimento de um ponto a outro com o mínimo de esforço, ou seja, com o movimento de menos juntas possíveis. Além disso, a chave para fazer robôs mais versáteis implica em usar realimentações com sensores

externos, os quais alimentam informações em termos de variáveis do espaço de configuração. Estas informações são necessárias na determinação dos valores apropriados das juntas do robô, para que assim possa realizar determinadas tarefas.

Mesmo para uma posição possível da extremidade do manipulador há, frequentemente, mais do que uma solução possível para os ângulos das juntas. Na figura 3.5 apresenta-se um exemplo típico, para um manipulador com três juntas consecutivas paralelas.

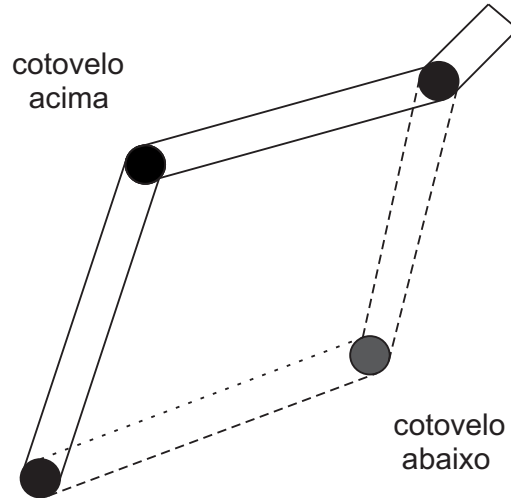


Figura 3.5: Duas soluções possíveis para os ângulos de junta, dada uma mesma posição.

A cinemática inversa do robô manipulador Titan IV será calculada caso a caso. Como o manipulador possui os eixos 4 e 5 de suas últimas juntas (5 e 6) encontrando-se em um único ponto, a solução analítica é possível de ser encontrada. Então considere a equação (3-3), que modela a posição do efetuador terminal do manipulador em relação à base pela multiplicação de matrizes de transformação obtidas anteriormente.

$$T_6^0 = A_1^0 A_2^1 A_3^2 A_4^3 A_5^4 A_6^5 \quad (3-3)$$

esta equação pode ser manipulada para encontrar a posição do eixo cinco P_5^1 em relação ao eixo um,

$$P_5^1 = T_5^1 = A_2^1 A_3^2 A_4^3 A_5^4 = (A_1^0)^{-1} T_6^0 (A_6^5)^{-1} \quad (3-4)$$

Seja representada de forma geral a matriz T

$$T_6^0 = \begin{bmatrix} n_x & t_x & b_x & p_x \\ n_y & t_y & b_y & p_y \\ n_z & t_z & b_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-5)$$

então pode-se expressar assim as matrizes

$$P_5^1 = A_2^1 A_3^2 A_4^3 A_5^4 = \begin{bmatrix} & L_3 C_2 + L_4 C_{23} + L_5 C_{234} \\ \mathbf{R}_{3 \times 3} & L_3 S_2 + L_3 S_{23} + L_3 S_{234} \\ & 0 \\ 0 & 1 \end{bmatrix} \quad (3-6)$$

$$(A_1^0)^{-1} T_6^0 (A_6^5)^{-1} = \begin{bmatrix} & (p_x - L_6 b_x) C_1 + (p_y - L_6 b_y) S_1 - L_2 \\ \mathbf{R}_{3 \times 3} & p_z - b_z L_6 - L_1 \\ & (p_x - L_6 b_x) S_1 - (p_y - L_6 b_y) C_1 \\ 0 & 1 \end{bmatrix} \quad (3-7)$$

Igualando-se as matrizes (3-6) e (3-7), obtém-se o ângulo θ_1 por,

$$\begin{aligned} 0 &= (p_x - L_6 b_x) S_1 - (p_y - L_6 b_y) C_1 \\ \tan \theta_1 &= \frac{p_y - L_6 b_y}{p_x - L_6 b_x} \\ \theta_1 &= \arctan\left(\frac{p_y - L_6 b_y}{p_x - L_6 b_x}\right) \end{aligned} \quad (3-8)$$

Seja $M = (A_1^0)^{-1} T_6^0 (A_6^5)^{-1}$ de (3-4), então pode-se encontrar outra relação igualando-se as componentes das matrizes T_5^1 e M

$$\begin{aligned} T_5^1(3, 3) &= M(3, 3) \\ S_5 &= b_y C_1 - b_x S_1 \end{aligned} \quad (3-9)$$

Resolvendo-se a equação (3-9), obtém-se o ângulo θ_5 , mas convenientemente transformaremos a função trigonométrica inversa arcsin a arctan. Utilizamos a variável W para obter a função trigonométrica arctan como se mostra na figura 3.6 (b), pelo teorema de Pitágoras, encontramos o valor de W em função das variáveis mostradas na equação (3-9).

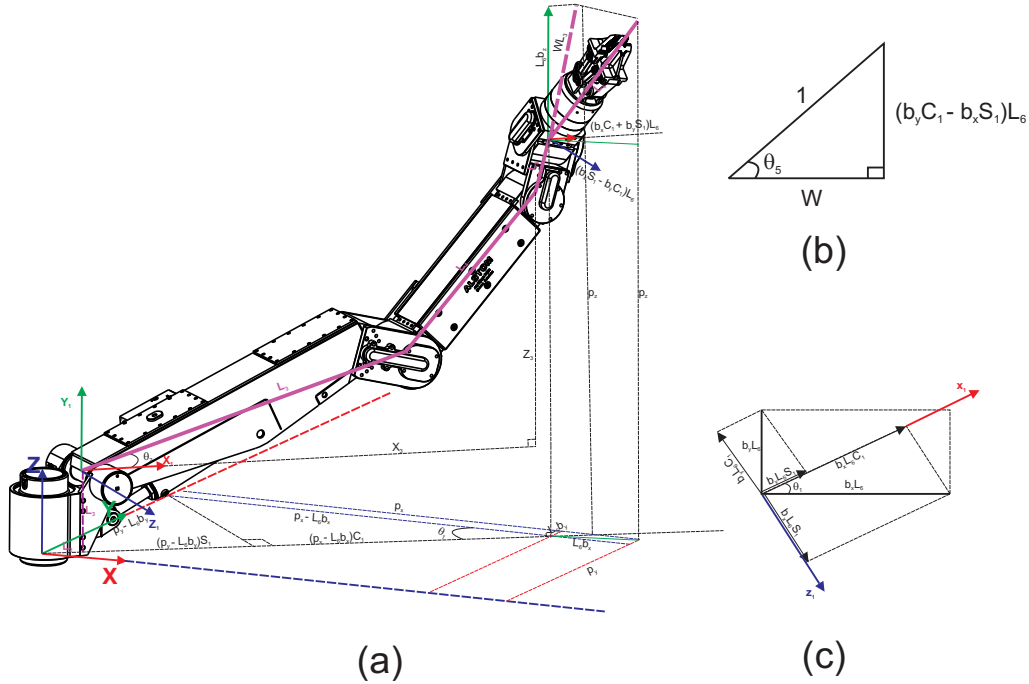


Figura 3.6: Método geométrico para calcular a cinemática inversa do robô Titan IV.

Observa-se que o ângulo θ_5 deve encontrar-se no intervalo $-\frac{\pi}{2} < \theta_5 < \frac{\pi}{2}$.

$$\begin{aligned}
 W &= \sqrt{1 - (b_y C_1 - b_x S_1)^2} \\
 W &= \sqrt{b_x^2 C_1^2 + b_y^2 S_1^2 + 2b_x C_1 b_y S_1 + b_z^2} \\
 W &= \sqrt{(b_x C_1 + b_y S_1)^2 + b_z^2}
 \end{aligned} \tag{3-10}$$

As funções trigonométricas arcsin e arccos apresentam múltiplas soluções, por isso é recomendável utilizar a função arctan para solucionar o problema da cinemática inversa.

$$\begin{aligned}
 \tan \theta_5 &= \frac{b_y C_1 - b_x S_1}{W} \\
 \theta_5 &= \arctan\left(\frac{b_y C_1 - b_x S_1}{W}\right)
 \end{aligned} \tag{3-11}$$

A relação geométrica entre os elos do robô manipulador, vista na figura 3.7 (a), mostra um sistema típico de um robô de dois graus de liberdade cuja solução pode-se encontrar utilizando as variáveis Z_3 e X_3 . Da figura 3.6 (a) e

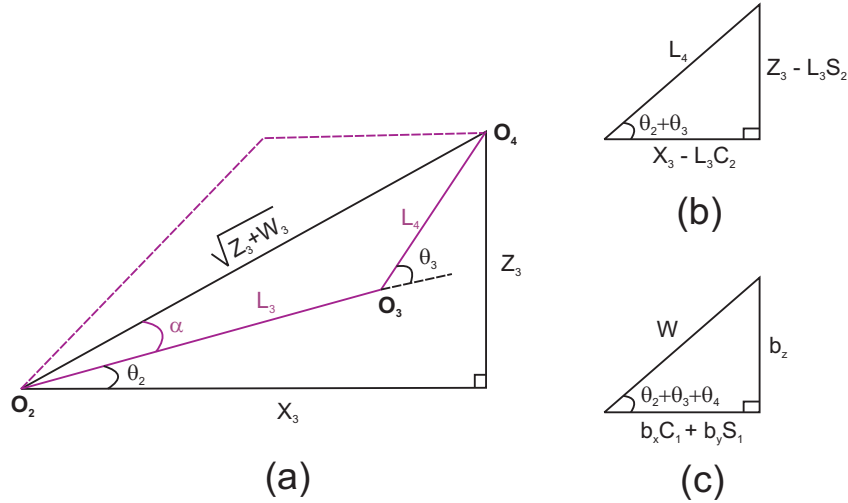


Figura 3.7: Eixos O_2 , O_3 e O_4 , sobre a forma cinemática inversa de dois graus de liberdade.

(b) encontra-se o valor das variáveis Z_3 e X_3 utilizando o método geométrico

$$X_3 = (p_x - L_6 b_x)C_1 + (p_y - L_6 b_y)S_1 - L_2 - \frac{L_5}{W}(b_x C_1 + b_y S_1) \quad (3-12)$$

$$Z_3 = p_z - L_6 b_z - L_1 - \frac{L_5}{W} b_z \quad (3-13)$$

Agora, utilizando-se a lei de cossenos, encontramos o valor de α visto na figura 3.7 (a)

$$\begin{aligned} L_4^2 &= Z_3^2 + X_3^2 + L_3^2 - 2L_3\sqrt{Z_3^2 + X_3^2} \cos \alpha \\ \alpha &= \arccos\left(\frac{Z_3^2 + X_3^2 + L_3^2 - L_4^2}{2L_3\sqrt{Z_3^2 + X_3^2}}\right) \end{aligned} \quad (3-14)$$

então

$$\begin{aligned} \tan(\alpha + \theta_2) &= \frac{Z_3}{X_3} \\ \theta_2 &= \arctan\left(\frac{Z_3}{X_3}\right) - \alpha \\ \theta_2 &= \arctan\left(\frac{Z_3}{X_3}\right) \pm \arccos\left(\frac{Z_3^2 + X_3^2 + L_3^2 - L_4^2}{2L_3\sqrt{Z_3^2 + X_3^2}}\right) \end{aligned} \quad (3-15)$$

O ângulo θ_2 apresenta duas soluções, assim podem-se obter duas possibilidades de posição na configuração das juntas do robô, a segunda delas representada por linhas tracejadas na figura 3.7 (a).

Também pode-se obter o valor do ângulo θ_3 pelo método geométrico seguindo o triângulo retângulo na figura 3.7 (b), cujos catetos encontram-se

da figura 3.6 (a) e (c).

$$\begin{aligned}\tan(\theta_2 + \theta_3) &= \frac{Z_3 - L_3 S_2}{X_3 - L_3 C_2} \\ \theta_3 &= \arctan\left(\frac{Z_3 - L_3 S_2}{X_3 - L_3 C_2}\right) - \theta_2\end{aligned}\quad (3-16)$$

Por último, temos a outra relação geométrica vista na figura 3.7 (c).

$$\begin{aligned}\tan(\theta_2 + \theta_3 + \theta_4) &= \frac{b_z}{b_x C_1 + b_y S_1} \\ \theta_4 &= \arctan\left(\frac{b_z}{b_x C_1 + b_y S_1}\right) - (\theta_2 + \theta_3)\end{aligned}\quad (3-17)$$

Igualando-se as matrizes T_5^1 e M , achamos θ_6 da relação

$$\begin{aligned}T_5^1(3, 2) &= M(3, 2) \\ 0 &= (n_x S_1 - n_y C_1)S_6 + (t_x S_1 - t_y C_1)C_6 \\ \theta_6 &= \arctan\left(\frac{t_y C_1 - t_x S_1}{n_x S_1 - n_y C_1}\right)\end{aligned}\quad (3-18)$$

As equações da cinemática inversa do manipulador possuem vários termos definidos por funções trigonométricas, o que leva a múltiplas soluções, num total de oito para o robô Titan IV. Além disto, existem questões a serem vistas sobre a existência de soluções

Considerando-se a equação (3-8) referente ao eixo 1, há a princípio a possibilidade de duas soluções. Não obstante, limitações físicas de movimento desta junta permitem que ela se mova entre -120° e 120° , permitindo apenas as soluções neste intervalo.

Quanto às equações (3-15) e (3-16), referentes aos eixos 2 e 3 respectivamente, têm-se a princípio duas soluções possíveis. Sabe-se que o manipulador funciona com o cotovelo voltado para cima devido às restrições das juntas. Isto dependerá só da forma de configuração onde se quer movimentar.

Na implementação do simulador, utilizamos a função arco tangente de dois argumentos, ou simplesmente, $ATan2$, para obter soluções que estão dentro do intervalo da configuração do robô. O código implementado encontra-se mostra no Apêndice C.

3.3.1 Dificuldades dos Métodos Geométricos

Os métodos geométricos apresentam alguns problemas cuja resolução requer um estudo cuidadoso:

- Como se viu o método pode fornecer oito soluções diferentes para uma dada situação do efetuador terminal, e torna-se necessário escolher a que deve ser usada.
- Como a solução para cada junta resulta, em última análise, do cálculo de uma função $ATan2$, surge o problema dos ângulos obtidos virem sempre restringidos ao intervalo $[-180^\circ, +180^\circ]$. Este fato pode levantar problemas, caso o intervalo de variação dos ângulos das juntas não esteja contido nesse intervalo.

3.3.2

Problema da escolha da Solução

Depois de calculadas todas as possíveis soluções, como o intervalo de variação das juntas é conhecido, o primeiro passo deverá ser eliminar quaisquer soluções que não estejam dentro desses intervalos. Podem então ocorrer três situações:

- Todas as soluções estão dentro dos limites de variação impostos às juntas. A dimensão do problema não é reduzida.
- Apenas algumas das soluções não violam os limites de variação impostos às juntas. A dimensão do problema foi reduzida.
- Todas as soluções violam os limites de variação impostos às juntas. Não existe solução. Isto significa que se pretende mover o manipulador para um ponto fora do seu espaço de trabalho.

3.3.3

Função Custo

Um procedimento possível para selecionar uma possível solução é escolher aquela que minimiza uma determinada função custo. Por exemplo, escolher a solução que minimiza o erro quadrático médio entre o vetor de coordenadas no espaço das juntas atual e o vetor de coordenadas no espaço das juntas candidato à próxima solução. É importante notar que, neste caso, se tenta fazer com que as trajetórias, no espaço das juntas, sejam contínuas no tempo.

Uma outra possibilidade consiste em escolher a solução que minimiza o erro quadrático médio entre o vetor de coordenadas (no espaço das juntas) candidato a próxima solução e o vetor de coordenadas no espaço das juntas cujas componentes são os pontos médios dos intervalos de variação impostos às juntas. Neste caso, tenta-se manter as juntas o mais afastado possível dos seus limites.

3.4 Cinemática Diferencial

Nas seções anteriores, as equações de cinemática direta e inversa estabeleceram as relações entre as coordenadas de juntas e a posição e orientação do efetuador terminal. Na cinemática diferencial, são estudadas as relações entre a velocidade descrita pelo efetuador terminal em relação às velocidades angulares e lineares das coordenadas da junta. Tal relação é descrita por uma matriz denominada Matriz Jacobiana, a qual é dependente da configuração do manipulador. O Jacobiano se constitui numa das mais importantes ferramentas para caracterização de robôs manipuladores; de fato, ele é utilizado para encontrar configurações singulares, análise de redundâncias, determinação de algoritmos de cinemática inversa, descrição do mapeamento de forças (estáticas) aplicadas nas juntas a partir da carga no efetuador terminal, além de servir como base para obtenção da equação da dinâmica e esquema de projeto de controladores no espaço operacional

Definindo-se a velocidade linear e a velocidade angular do efetuador terminal do robô manipulador por:

$$\dot{\mathbf{p}} = \begin{bmatrix} \mathbf{v}_e \\ \mathbf{w}_e \end{bmatrix} \quad (3-19)$$

obtem-se

$$\dot{\mathbf{p}} = \mathbf{J}\dot{\mathbf{q}} \quad (3-20)$$

onde $\dot{\mathbf{q}} = [\dot{q}_1, \dots, \dot{q}_n]$, é o vetor de velocidade $n \times 1$ da junta, e \mathbf{J} é a matriz Jacobiana.

A dimensão geral da uma Matriz Jacobiana é $m \times n$, onde m é o número de linhas, que é igual ao número de graus de liberdade do campo de trabalho do robô, e n é o número de colunas, que é igual ao número de articulações do robô. Para um robô que trabalha no espaço, m será no máximo igual a 6 e, para um robô que trabalha no plano, m será no máximo igual a 3. Os 6 graus de liberdade do espaço correspondem aos três graus de liberdade de posicionamento e aos três de orientação de um corpo rígido. Assim, observa-se que o número de linhas da Matriz Jacobiana não é fixa, devendo ser definida pelo interesse do problema e, principalmente, em função do que o robô é capaz de realizar.

Para o caso do robô Titan IV, de seis graus de liberdade, e com um espaço de trabalho $m = 6$, pode-se obter a Matriz Jacobiana 6×6 , apresentada no Apêndice A.

3.5 Dinâmica Inversa

O conhecimento de modelos dinâmicos realistas é fundamental para testar, em simulação, eficiência de leis de controle e técnicas de planejamento de trajetórias para robôs manipuladores, sem necessidade de usar um sistema físico. A análise do modelo dinâmico é útil para se projetar protótipos da estrutura mecânica dos elos.

O problema da dinâmica inversa está ligado diretamente ao problema de controle do manipulador, consiste em determinar os esforços nos atuadores, dadas as acelerações, velocidades e posições das juntas correspondentes. Os modelos mecânicos são desenvolvidos a partir das leis da Mecânica Newtoniana e Lagrangiana, obtendo-se assim as equações do movimento em termos dos parâmetros geométricos e inerciais dos elementos do robô. Para isto existem vários métodos de formulações como Lagrange, Newton-Euler, Kane, etc. A partir das quais, foram obtidos diferentes formas de representar a dinâmica do robô manipulador. Todas estas formulações cumprem o mesmo objetivo, mas apresentam diferentes estruturas e métodos de avaliação de acordo as necessidades.

Para a modelagem dinâmica do robô Titan IV, utiliza-se o método recursivo de Newton Euler desenvolvido originalmente por Luh-Walker-Paul's [53]. Este algoritmo permite obter um conjunto de equações recursivas que permitem reduzir o tempo de cálculo, permitindo um controle em tempo real.

Computar os torques nas juntas correspondentes a uma trajetória de junta especificada é um problema que aparece frequentemente no controle dinâmico de manipuladores. A partir das posições de junta e de suas primeiras e segundas derivadas (q_2 , \dot{q}_2 e \ddot{q}_2), é possível calcular o vetor torque τ de esforços de junta correspondente usando as equações recursivas do algoritmo Newton-Euler. A expressão analítica vetorial resultante é denominada equação de Dinâmica Inversa ou de forma fechada, a qual é constituída por um sistema de equações diferenciais de segunda ordem, não lineares e acopladas. Esta equação pode ser decomposta nos termos.

$$\tau = \tau_I(\mathbf{q}, \ddot{\mathbf{q}}) + \tau_C(\mathbf{q}, \dot{\mathbf{q}}) + \tau_G(\mathbf{q}) + \tau_F(\dot{\mathbf{q}}) \quad (3-21)$$

onde:

τ = vetor $n \times 1$ torque nas juntas.

\mathbf{q} = vetor $n \times 1$ de variáveis nas juntas.

$\tau_I(\mathbf{q}, \ddot{\mathbf{q}})$ = vetor $n \times 1$ de reações inerciais.

$\tau_C(\mathbf{q}, \dot{\mathbf{q}})$ = vetor $n \times 1$ de reações coriolis e centrífugas.

$\tau_G(\mathbf{q})$ = vetor $n \times 1$ de reações gravitacionais.

$\tau_F(\dot{\mathbf{q}})$ = = vetor $n \times 1$ de reações de atrito

O vetor de reações inerciais incorpora torques resistentes às acelerações e pode ser expresso como:

$$\boldsymbol{\tau}_I = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}$$

onde \mathbf{H} é a matriz de inércia $n \times n$, simétrica e positiva definida. Observe que $\boldsymbol{\tau}_I$ é proporcional às acelerações das juntas.

O vetor de reações coriolis e centrífugas incorpora torques resistentes ao movimento correspondentes a acelerações devidas à rotação relativa entre elos. O vetor $\boldsymbol{\tau}_C$ inclui termos onde aparecem produtos de velocidades de juntas (reações coriolis) e velocidades de junta elevadas ao quadrado (reações centrífugas).

O vetor de reações gravitacionais incorpora torques resistentes ao movimento devido ao peso dos elos do manipulador.

Se for necessário considerar o atrito nas juntas, este pode ser incorporado diretamente às equações de saída do algoritmo Newton-Euler, visto que estas reações ocorrem no eixo das juntas. Um modelo simples para as reações de atrito consiste em uma componente de atrito viscoso somada a uma componente de atrito "Coulomb"

$$\boldsymbol{\tau}_F = \mathbf{k}_{Fv}\dot{\mathbf{q}} + \mathbf{k}_{Fc}sgn(\dot{\mathbf{q}})$$

onde \mathbf{k}_{Fv} e \mathbf{k}_{Fc} são os coeficientes de atrito viscoso e coulomb de cada junta. A função descontínua $sgn(\dot{\mathbf{q}})$ retorna o sinal da velocidade da junta.

3.6 Dinâmica Direta

O problema da dinâmica direta é fundamental em aplicações que envolvem a simulação dinâmica do robô manipulador, e consiste em determinar a resposta dinâmica do mesmo, ou seja, a evolução temporal das posições e velocidade das juntas em função dos esforços aplicados pelos atuadores.

Computar as trajetórias da junta que correspondem à resposta aos esforços aplicados pelos atuadores é um problema que deve ser resolvido quando se deseja simular computacionalmente o comportamento dinâmico. A expressão analítica vetorial que descreve a aceleração das juntas em função dos esforços aplicados pelos atuadores é denominada equação de dinâmica direta, a qual, de

modo análogo à equação de dinâmica inversa, é constituída por um sistema de equações diferenciais de segunda ordem, não lineares e acopladas. Esta equação pode ser decomposta e obtida diretamente a partir da dinâmica inversa.

A eficiência computacional depende não só do algoritmo mas também da máquina onde esse algoritmo vai ser implementado, e do próprio compilador. Assim, na tentativa de determinar os fatores mais decisivos na eficiência computacional do sistema, realizaram-se diversos estudos que envolveram: tempo de computação em diversos sistemas computacionais para varias operações matemáticas e lógicas, comportamento de um modelo matemático, e tempo de computação no mesmo processador para diversas operações em vírgula flutuante, com quatro e oito bytes de precisão.

Para simplificar o problema da dinâmica direta, define-se a forma da equação geral do movimento por:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau} \quad (3-22)$$

onde;

$\mathbf{H}(\mathbf{q})$ é a matriz de inércia não singular $n \times n$ do manipulador.

$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ é a matriz $n \times n$ dos esforços das acelerações centrífuga e de coriolis.

$\mathbf{G}(\mathbf{q})$ é o vetor $n \times 1$ dos esforços da gravidade.

\mathbf{q} é o vetor $n \times 1$ das posições das juntas.

$\boldsymbol{\tau}$ é o vetor $n \times 1$ dos torques nas juntas.

Para solucionar esta equação, Walter e Orin [54] propõem 4 métodos diferentes, que diferem entre si pela eficiência computacional. O Apêndice A descreve os métodos 1 e 3, onde este último dá a maior eficiência computacional. Então, o método 3 será usado para achar a dinâmica direta do robô manipulador, requisito necessário para as seguintes etapas no desenvolvimento da geração de trajetórias e o controle do manipulador .

No próximo capítulo, os modelos desenvolvidos para o robô Titan IV, são usados para implementar seu controle de movimento.

4

Controle de Movimento

Neste capítulo apresenta-se uma abordagem por Controle de Torque Computado e por Modos Deslizantes aplicado a robôs manipuladores. Primeiramente, apresentamos os conceitos básicos da abordagem para o Controle de posição, de trajetória, Torque Computado, e por Modos Deslizante, para depois implementá-lo nos programas de simulação. Além disso, vamos comparar a eficiência das diferentes técnicas dos algoritmos de controle para o robô manipulador Titan IV, descrito no capítulo anterior.

4.1

Controle no Espaço das Juntas

O principal objetivo do controle no espaço da junta é projetar um controlador com realimentação, de tal forma que as coordenadas na junta $\mathbf{q}(t) \in R^n$ sigam o movimento desejado $\mathbf{q}_d(t)$ tão próximo possível. Para este fim, considere equação (3-22) de movimento de um manipulador de n graus de liberdade expressado no espaço da junta ([55]). Neste caso, o controle do robô manipulador é naturalmente realizado no espaço das juntas, visto que as saídas do controle são os torques das juntas. No entanto o usuário especifica um movimento em termos das coordenadas do efetuador terminal, e portanto é necessário entender a seguinte estratégia.

Na figura 4.1 mostra-se um esquema básico dos métodos de controle no espaço da junta. Primeiramente, o movimento desejado, que é descrito em termos das coordenadas do efetuador terminal, é transformado para uma correspondente trajetória na junta usando a cinemática inversa do manipulador. Então o controlador com realimentação determina os torques nas juntas necessários para movimentar o manipulador ao longo da trajetória desejada, especificada na coordenadas da junta inicialmente, com as informações do estado atual das juntas([56] e [57]).

Desde que sempre se assumiu que a tarefa desejada é dada em termos de sequências de tempo do movimento articular, os esquemas de controle no espaço da junta são bastante adequadas em situações onde as tarefas do manipulador podem ser pré-planejadas com precisão e será necessário pouco

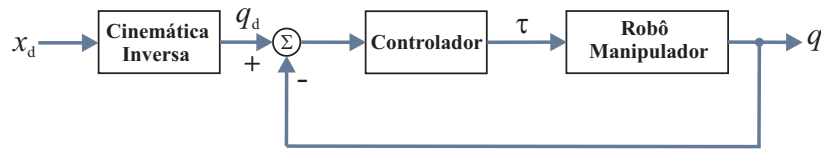


Figura 4.1: Esquema genérico do conceito de Controle no espaço da junta.

ou nenhum ajuste da trajetória ([58]). Tipicamente, a cinemática inversa é realizada para alguns pontos intermédios da tarefa, e as trajetórias nas juntas é interpolado utilizando soluções determinadas não triviais. Embora a trajetória comandada consista de movimentos em linha reta entre os pontos de interpolação representado nas coordenadas do efetuador terminal, o movimento resultante das juntas consiste de segmentos curvilíneos que correspondem à trajetória do efetuador final nos pontos de interpolação.

De fato, o controle no espaço das juntas incluem sempre controladores PD, PID, controle por Dinâmica Inversa, Controle Baseado em Lyapunov, Controle por Torque Computado, etc.

4.2 Controle de Posição

Considere o robô manipulador mostrado na figura 4.2, que encontra-se no espaço tridimensional, e a tarefa é movimentar a uma posição final desejada especificada pelo vetor constante \mathbf{q}_d (ângulos das juntas), sem importar a trajetória realizada. Isto é fisicamente realizável por uma lei de controle proporcional-derivativo (PD) que controla cada junta independentemente, baseado na medida local do erro de posição $\tilde{q}_j = q_{d_j} - q_j$ e das velocidades das juntas \dot{q}_j para $(j = 1, 2, \dots, n)$, obtendo

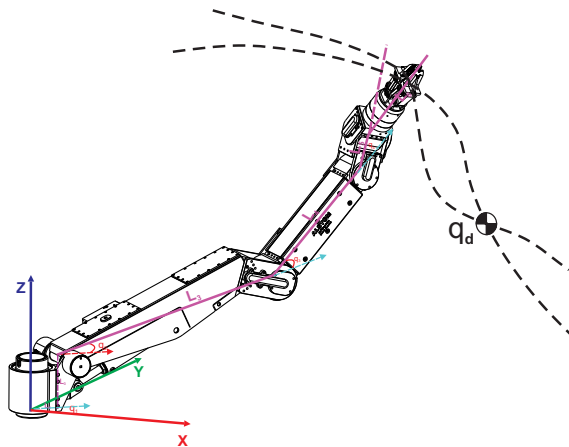


Figura 4.2: Robô Manipulador serial de 6 Graus de Liberdade.

$$\tau_j = k_{P_j} \tilde{q}_j - k_{D_j} \dot{q}_j \quad (4-1)$$

onde k_{P_j} e k_{D_j} são constantes estritamente positivas. Esta expressão é simplesmente a representação de configurar cada uma das articulações do manipulador como um dispositivo mecânico passivo composto por uma mola e um amortecedor, e tendo q_{d_j} como a posição desejada de repouso. O sistema físico passivo resultante simplesmente exibe oscilações amortecidas para a posição de repouso \mathbf{q}_d .

Em sua forma geral, a lei de controle PD pode ser descrita por:

$$\boldsymbol{\tau} = \mathbf{K}_P \tilde{\mathbf{q}} - \mathbf{K}_D \dot{\mathbf{q}} \quad (4-2)$$

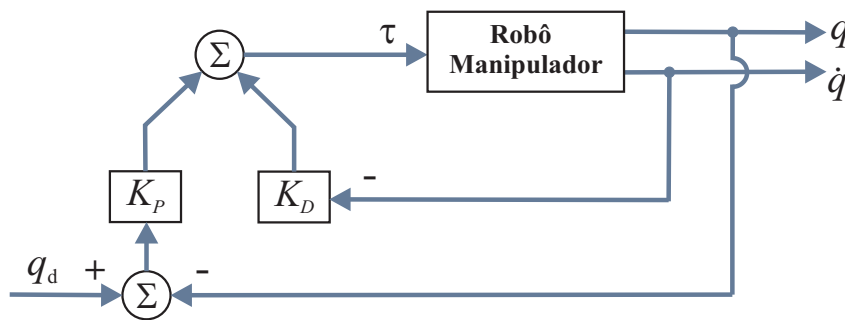


Figura 4.3: Diagrama de Blocos: Controle PD.

onde \mathbf{K}_P e \mathbf{K}_D são matrizes constantes simétrica definidas positivas (não necessariamente diagonais).

Para poder simular os algoritmo de controle, inicialmente se utilizou o ambiente Matlab, devido a possuir funções gráficas prontas para apresentar os resultados e, assim poder ter uma calibração rápida dos parâmetros do controlador. Nas características da simulação, considerou-se uma taxa de amostragem de 1KHz-5KHz, e o método numérico de Euler para solucionar as equações diferencias do sistema.

Agora vamos simular o algoritmo de controle PD, utilizando uma uma taxa de amostragem de 1KHz, sobre o robô manipulador Schilling de 6 graus de liberdade. cujos parâmetros do controlador são:

$$K_P = \begin{bmatrix} 1500 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1500 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1500 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 \end{bmatrix}$$

$$K_D = \begin{bmatrix} 350 & 0 & 0 & 0 & 0 & 0 \\ 0 & 350 & 0 & 0 & 0 & 0 \\ 0 & 0 & 350 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Os valores destas matrizes foram obtidos empiricamente depois de uma série de testes, até atingir erros muito pequenos.

Na figura 4.4 e 4.5 mostra a resposta do comportamento dos deslocamentos nas juntas do robô manipulador, em relação a um deslocamento de referência como entrada. Pode-se observar, que o deslocamento nas juntas atinge à referência (trajetória constante) dada pelo usuário em um tempo razoável para os ganhos arbitrários.

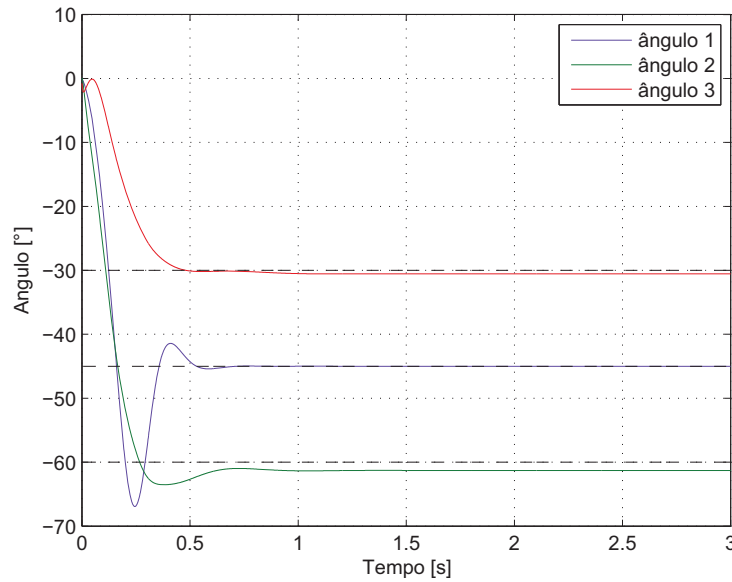


Figura 4.4: Referência e resposta de posição: juntas q_1 , q_2 e q_3 .

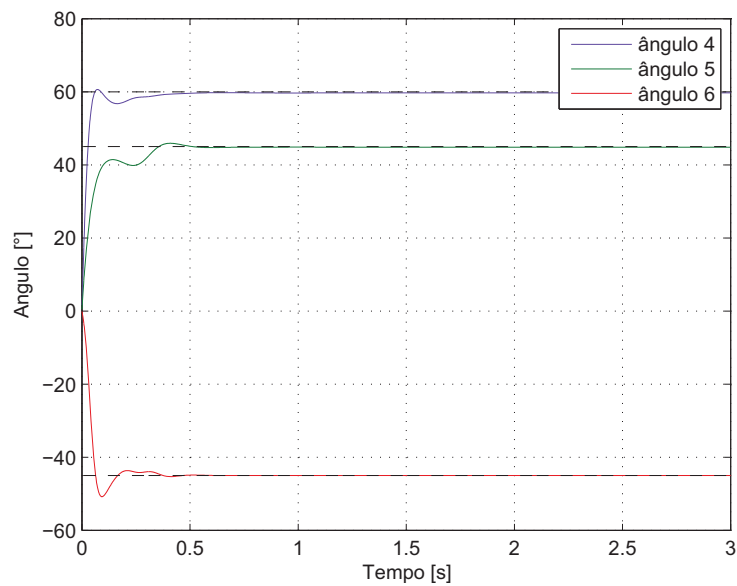


Figura 4.5: Referência e resposta de posição: juntas q_4 , q_5 e q_6 .

Na figura 4.6 e 4.7 mostra os erros de posicionamento das juntas. Observa-se que os valores dos erros ficam no intervalo de $\pm 3^\circ$

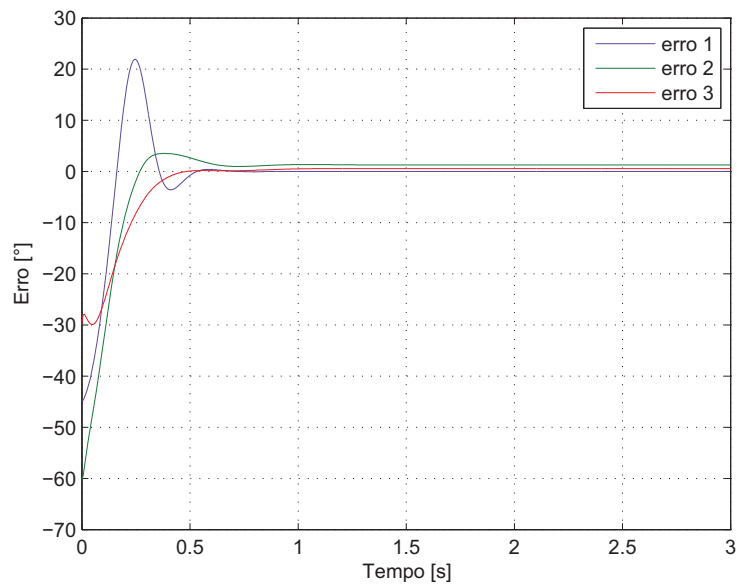


Figura 4.6: Erros de posicionamento: juntas \tilde{q}_1 , \tilde{q}_2 e \tilde{q}_3 .

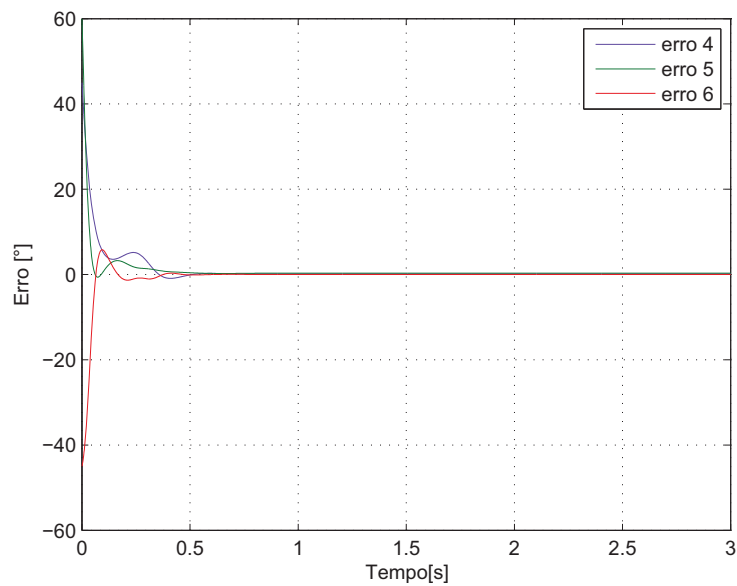


Figura 4.7: Erros de posicionamento: juntas \tilde{q}_4 , \tilde{q}_5 e \tilde{q}_6 .

Observa-se que o algoritmo de controle PD se comporta favoravelmente para controlar posição, mas com pequenos erros. Também há que ter em consideração que a trajetória de referência é constante degrau.

Para minimizar os erros, pode-se adicionar um componente integrador ao controle PD resultando um Controle PID.

4.2.1 Controle PD + Compensação de Gravidade

Um simples método de projeto para controlar manipuladores é utilizar um esquema de controle linear em base à linearização do sistema em torno a um ponto de operação. Um exemplo deste método é um controle PD com uma compensação de gravidade. A compensação de gravidade atua como um bias de correção, compensando apenas para a quantidade de torque que criam um *overshooting* e um comportamento transiente assimétrico.

Formalmente, a lei de controle tem a seguinte forma:

$$\tau = \mathbf{K}_P \tilde{\mathbf{q}} - \mathbf{K}_D \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) \quad (4-3)$$

A figura 4.8 representa com um diagrama de blocos o controlador PD com compensação de gravidade

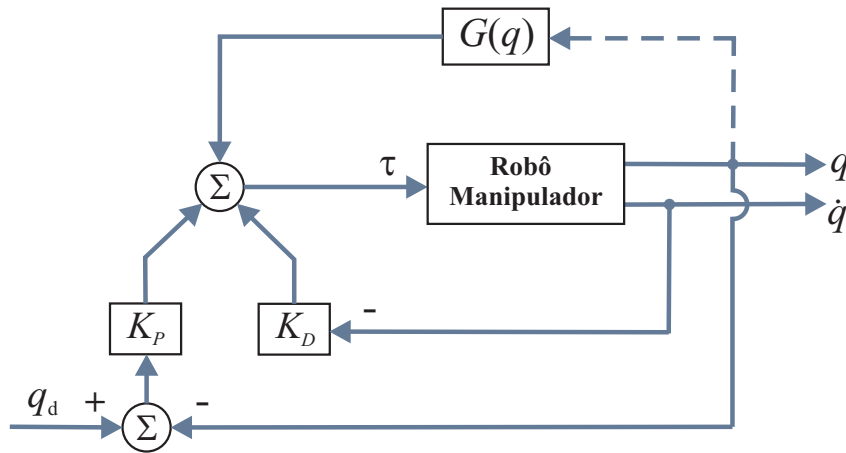


Figura 4.8: Diagrama de Blocos: Controle PD com compensação de gravidade.

Apresenta-se a seguir resultados da simulação, cujas características e parâmetros (matrizes K_P e K_D) são consideradas as mesmas que utilizou-se na simulação do controle PD.

A figura 4.9 e 4.10 mostram o comportamento do controle PD com compensação de gravidade para juntas do robô manipulador em torno a posições de referência (trajetória constante).

O controle PD com compensação de gravidade atua como um corretor do erro no posicionamento das juntas (mesma função de um integrador). Esta forma de corrigir os erros é mais eficiente e rápida de calibrar.

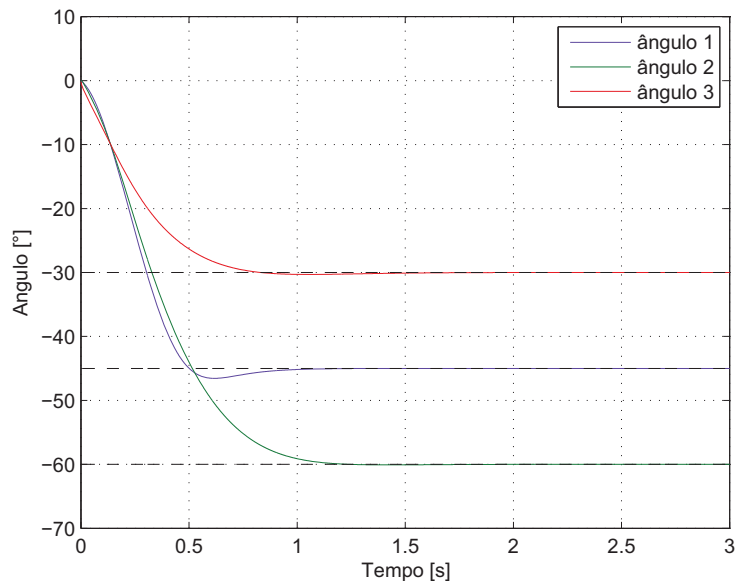


Figura 4.9: Referência e resposta de posição: juntas q_1 , q_2 e q_3 .

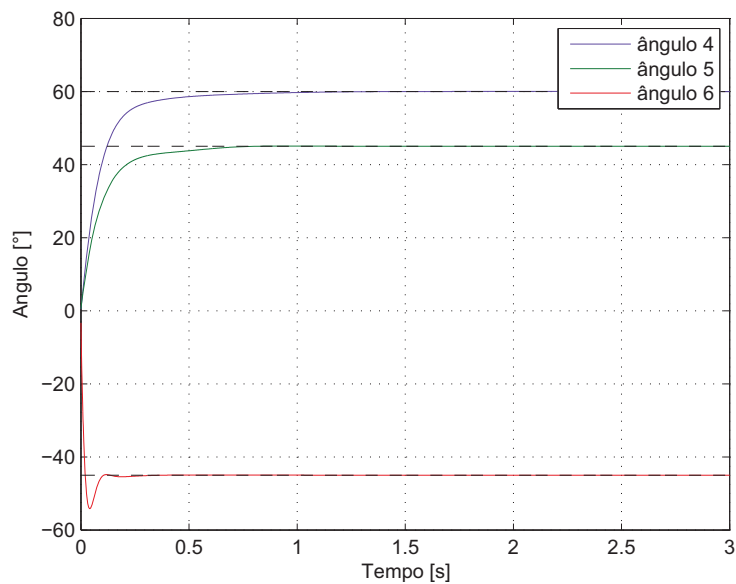


Figura 4.10: Referência e resposta de posição: juntas q_4 , q_5 e q_6 .

Observe que os erros de posicionamento (figura 4.11 e 4.12) são mais próximos a zero que os erros obtido pelo controle PD.

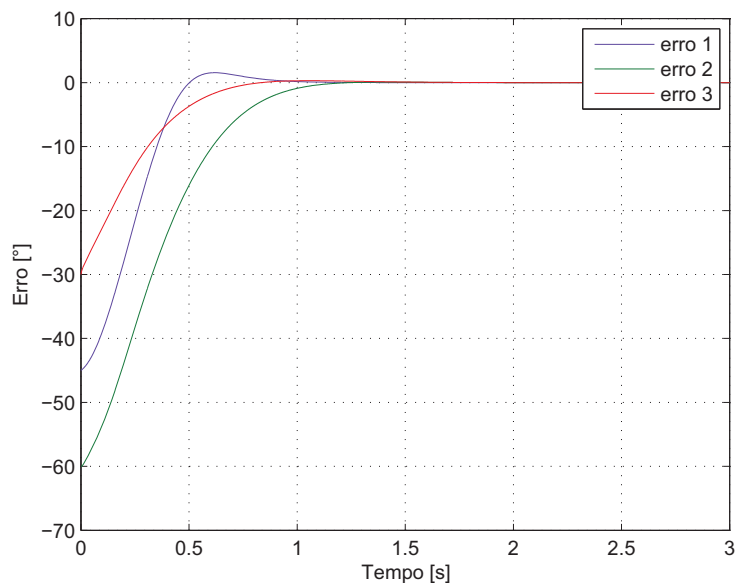


Figura 4.11: Erros de posicionamento: juntas \tilde{q}_1 , \tilde{q}_2 e \tilde{q}_3 .

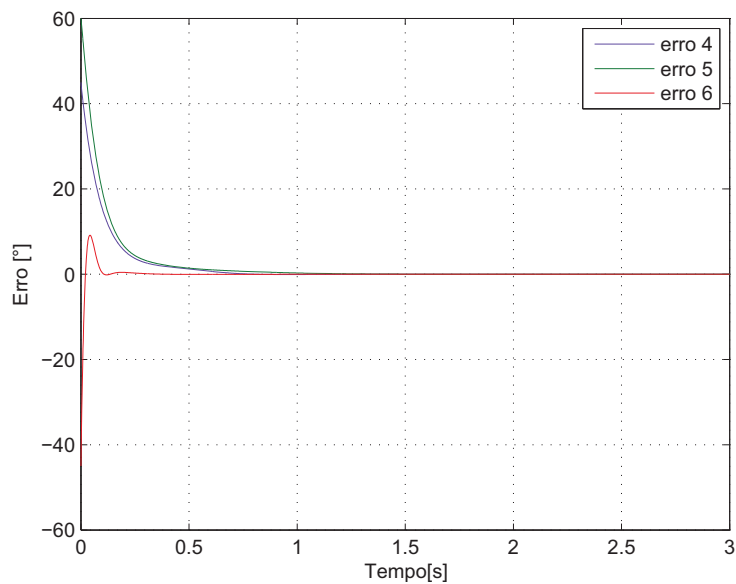


Figura 4.12: Erros de posicionamento: juntas \tilde{q}_4 , \tilde{q}_5 e \tilde{q}_6 .

Se a estimação do vetor de gravidade $\mathbf{G}(\mathbf{q})$ é aproximadamente igual ao da modelagem dinâmica, este robô se comportaria como se operara em um sistema de gravidade zero. Na simulação, se considerou a estimação do vetor $\mathbf{G}(\mathbf{q})$ com uma incerteza de 10%.

4.3 Controle Seguimento de Trajetória

Agora consideremos o caso quando o robô manipulador é requerido seguir uma trajetória desejada para conseguir atingir uma posição também desejada. Há muitas tarefas que precisam de uma capacidade efetiva para o seguimento de trajetória, tais como soldagem, corte, ou operações de grande velocidades na presença de obstáculos. Portanto, para melhorar o desempenho no seguimento da trajetória, os controladores devem ter em conta o modelo dinâmico do manipulador.

O problema de controle de seguimento de trajetória consiste em seguir uma determinada trajetória variante no tempo \mathbf{q}_d e suas derivadas sucessivas $\dot{\mathbf{q}}_d$ e $\ddot{\mathbf{q}}_d$, que descrevem a velocidade e aceleração, respectivamente. Para obter um desempenho eficiente, um esforço significativo tem sido dedicado ao desenvolvimento de estratégias de controle baseados no modelo dinâmico[59].

4.3.1 Controle de Torque Computado

Ao longo dos anos, muitos tipos de esquemas de controle têm sido propostos. A maioria deles pode ser considerada como casos especiais da classe controle por torque computado (figura 4.13), que é a aplicação da técnica *feedback linearization* para um sistema não linear em geral.

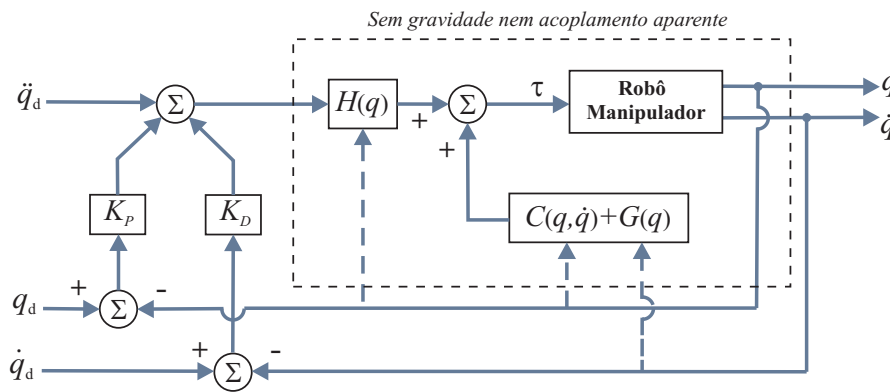


Figura 4.13: Diagrama de Blocos: Controle por Torque Computado.

Considere-se a lei de controle de entrada

$$\tau = \hat{H}(\mathbf{q})\mathbf{v} + \hat{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \hat{G}(\mathbf{q}) \quad (4-4)$$

sendo $\hat{H}(\mathbf{q})$, $\hat{C}(\mathbf{q}, \dot{\mathbf{q}})$, e $\hat{G}(\mathbf{q})$ as estimações das matrizes e o vetor correspon-

dente à equação dinâmica do robô manipulador, e \mathbf{v} é representada por:

$$\mathbf{v} = \ddot{\mathbf{q}}_d + \mathbf{K}_P \tilde{\mathbf{q}} + \mathbf{K}_D \dot{\tilde{\mathbf{q}}} \quad (4-5)$$

Nota-se que a lei de controle (4-4) contém os termos $\mathbf{K}_P \tilde{\mathbf{q}} + \mathbf{K}_D \dot{\tilde{\mathbf{q}}}$, o qual é um tipo PD. No entanto, esse termos são realmente pré-multiplicados pela matriz de inércia $\mathbf{H}(\mathbf{q})$. Portanto, isto não é um controlador linear como o PD, dado que os ganhos da posição e velocidades não são constantes, mas dependem do erro de posição $\tilde{\mathbf{q}}$.

Controle por torque computado foi um dos primeiros controladores baseado sem modelo, cuja abordagem foi criada para manipuladores. Isto é, no qual se faz uso explícito do conhecimento das matrizes $\mathbf{H}(\mathbf{q})$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$, e o vetor $\mathbf{G}(\mathbf{q})$. Além disso, observamos que a trajetória desejada de movimento \mathbf{q}_d e suas derivadas $\dot{\mathbf{q}}_d$ e $\ddot{\mathbf{q}}_d$, bem como as medidas de posição e velocidade \mathbf{q} e $\dot{\mathbf{q}}$, são usados para calcular a ação de controle.

Para fins práticos, a concepção de matrizes \mathbf{K}_D e \mathbf{K}_P podem ser escolhidos diagonal. Isto significa que a equação de malha fechada representa um sistema dissociado multivariável linear, isto é, o comportamento dinâmico dos erros de cada posição conjunta, governada por uma equação diferencial linear de segunda ordem, são independentes uns dos outros. Neste cenário, a seleção das matrizes \mathbf{K}_D e \mathbf{K}_P pode ser feita especificamente como:

$$\begin{aligned} \mathbf{K}_P &= \text{diag}\{\omega_1^2, \dots, \omega_n^2\} \\ \mathbf{K}_D &= \text{diag}\{2\omega_1, \dots, 2\omega_n\} \end{aligned} \quad (4-6)$$

Com esta escolha, cada junta responde como um sistema linear criticamente amortecido com largura de banda ω_i . A largura de banda define a velocidade da junta em questão e, conseqüentemente, a taxa de decaimento exponencial dos erros $\tilde{\mathbf{q}}(t)$ e $\dot{\tilde{\mathbf{q}}}(t)$.

Vamos agora simular o algoritmo de controle por torque computado, mas com uma taxa de amostragem de 3KHz, e com uma trajetória de referência variável. Os parâmetros do controlador (matrizes K_P e K_D) são encontrados depois de vários testes de calibração, tendo em conta a equação (4-6).

$$K_P = \begin{bmatrix} 1000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1500 & 0 & 0 & 0 \\ 0 & 0 & 0 & 900 & 0 & 0 \\ 0 & 0 & 0 & 0 & 800 & 0 \\ 0 & 0 & 0 & 0 & 0 & 800 \end{bmatrix}$$

$$K_D = \begin{bmatrix} 70 & 0 & 0 & 0 & 0 & 0 \\ 0 & 70 & 0 & 0 & 0 & 0 \\ 0 & 0 & 80 & 0 & 0 & 0 \\ 0 & 0 & 0 & 60 & 0 & 0 \\ 0 & 0 & 0 & 0 & 50 & 0 \\ 0 & 0 & 0 & 0 & 0 & 55 \end{bmatrix}$$

As figuras 4.14 e 4.15 mostram como o controle por torque computado atua de forma eficiente para uma trajetória variável. Nas estimações das matrizes H, C, G se considerou um 10% de incerteza em relação à dinâmica modelada real do robô. Os erros são mostrados nas figuras 4.16 e 4.17.

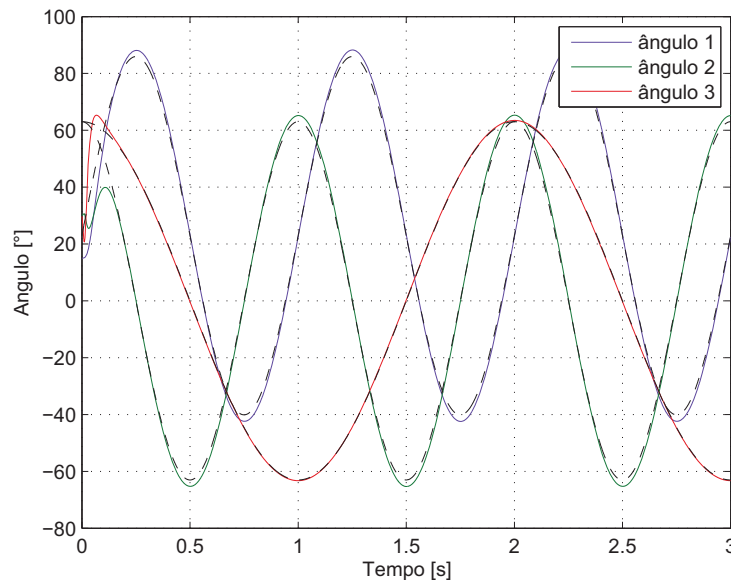


Figura 4.14: Referência e resposta do seguimento de trajetória: juntas q_1, q_2 e q_3 .

Se a estimação da modelagem dinâmica e aproximadamente equivalente à dinâmica real do robô, o sistema se comportaria como um sistema desacoplado.

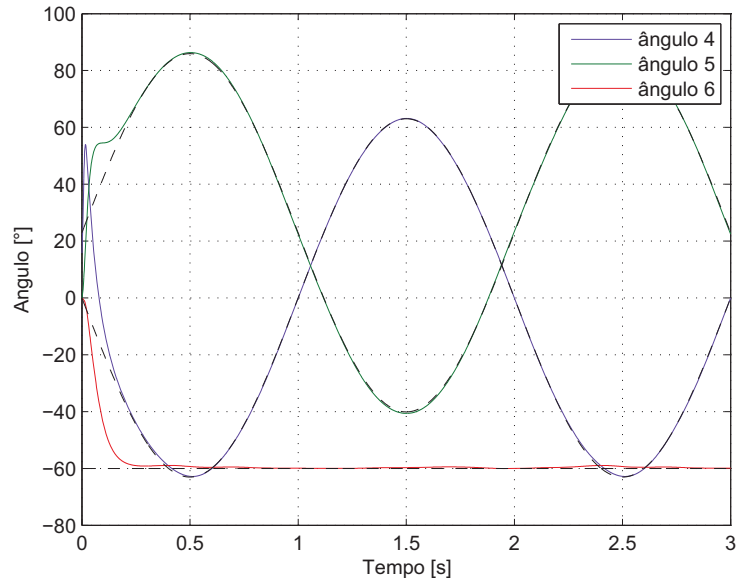


Figura 4.15: Referência e resposta do seguimento de trajetória: juntas q_4 , q_5 e q_6 .

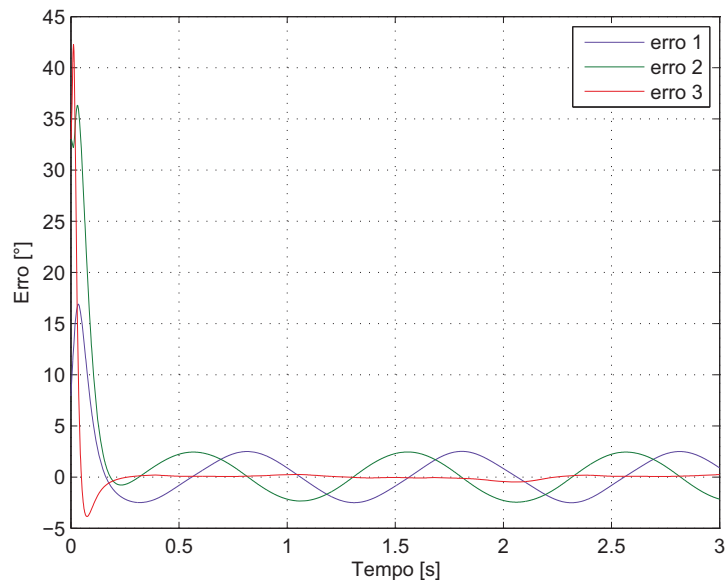


Figura 4.16: Erros de posicionamento: juntas \tilde{q}_1 , \tilde{q}_2 e \tilde{q}_3 .

Outro aspecto importante na implementação do controle por torque computado é a taxa de amostragem utilizada na discretização do mesmo. O limite inferior absoluto para a taxa de amostragem é dado pelo espectro do sinal de entrada de referência utilizado.

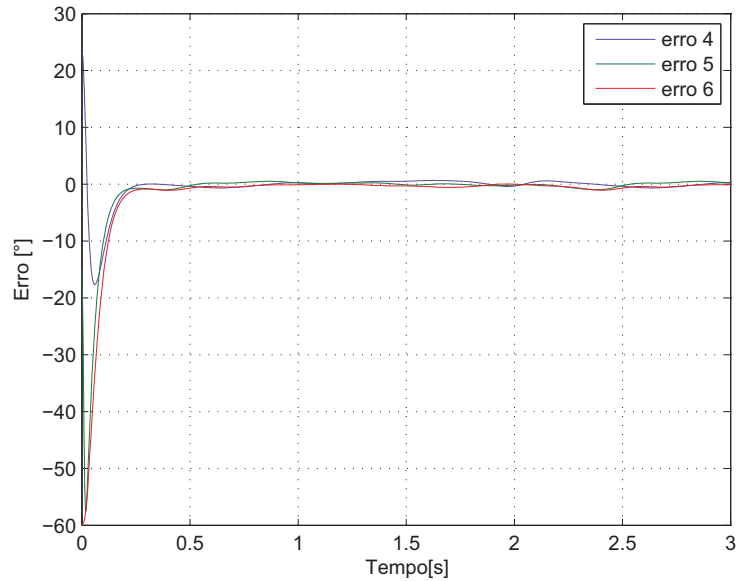


Figura 4.17: Erros de posicionamento: juntas \tilde{q}_4 , \tilde{q}_5 e \tilde{q}_6 .

A taxa de amostragem deve ser, no mínimo, duas vezes maior do que a largura de banda da entrada de referência. O limite superior da frequência de amostragem é dado pela capacidade de rejeição de perturbações e pela capacidade de supressão de vibrações mecânicas naturais. Se o período de amostragem é maior do que o tempo de correlação do ruído devido a perturbações randômicas, estas perturbações não serão suprimidas. Assim, uma boa regra empírica é escolher um período de amostragem pelo menos dez vezes menor do que o tempo de correlação do ruído.

Para evitar excitar modos de ressonância mecânicos, a frequência de amostragem deve ser, no mínimo, duas vezes maior do que a frequência natural dessas ressonâncias. Uma boa regra empírica é escolher a frequência de amostragem pelo menos dez vezes maior do que a frequência natural de ressonância mecânica.

4.4

Controle Robusto de Trajetória

Os casos tratados anteriormente abrangiam situações com incertezas nos parâmetros inerciais. Na prática, podemos encontrar outros tipos de incertezas não paramétricas, tais como a dinâmica não modelada de alta frequência (dinâmica dos atuadores ou vibrações estruturais), e de baixa frequência (atrito de Coulomb). Estas incertezas podem fazer com que a lei de controle convencional não distinga as informações de parâmetros e que o sistema não convirja adequadamente. O Controle Robusto tenta conseguir um desempenho consistente, muitas vezes exigindo uma estimação a priori dos limites dos parâmetros. Além disso, tem a vantagem de lidar com interferência, parâmetros de variação rápida e a dinâmica não modelada.

Isso motivou o uso de Controle a Estrutura Variável (VSC - Variable Structure Control) com modos deslizantes. Os primeiros estudos sobre VSC foram realizados na década de 60 por Emel'yanov e Barbashin na extinta União Soviética. Porém os estudos só foram difundidos entre os ocidentais em meados dos anos 1970, quando o livro Itkis [60] e o artigo Utkin [61] foram publicados em inglês.

A principal vantagem do controle VSC é a obtenção de um sistema de controle robusto a variações paramétricas e distúrbios externos. Os controladores VSC forçam a trajetória dos estados para um lugar no espaço de estados, cuja dinâmica é escolhida pelo projetista, e onde o sistema é imune a perturbações. Deste modo, ao contrário dos controladores baseados em dinâmica inversa, o efeito de incertezas paramétricas é anulado e o objetivo de controle alcançado, visto que o VSC funciona como um controlador de ganho infinito quando a trajetória de estados desliza sobre a superfície de chaveamento.

4.4.1

Controle por Modos Deslizantes

Para uma notação simplificada, o conceito será apresentado para a seguinte classe de sistemas não lineares. Considere o sistema dinâmico não linear de ordem n , definido por:

$$\dot{x}^{(n)} = f(\mathbf{x}, t) + g(\mathbf{x}, t)u + d(\mathbf{x}, t) \quad (4-7)$$

onde o escalar x é a saída de interesse (um ângulo da junta do robô), $u \in \mathbb{R}^m$ é a entrada de controle (torque aplicado nas juntas) e $\mathbf{x} \in \mathbb{R}^n$ é o vetor das variáveis de estados do sistema $\mathbf{x} = [x, \dot{x}, \dots, x^{(n-1)}]^T$.

A função $f(\mathbf{x}, t)$ (em geral não linear) não é exatamente conhecida, mas sua faixa de imprecisão em relação ao seus parâmetros é limitada superiormente por uma função contínua conhecida de \mathbf{x} . Da mesma forma, o ganho do controle $g(\mathbf{x}, t)$ é outra função não exatamente conhecida, mas é de sinal conhecido e limitado por uma função contínua conhecida de \mathbf{x} . A função $d(\mathbf{x}, t)$ representa perturbação e incertezas desconhecidas, que são tipicamente a inércia de uma sistema mecânico, que só é conhecida até uma certa aproximação, e os modelos de atrito que descrevem apenas parte das forças reais do atrito no sistema.

Um sistema de controle por modos deslizantes se compõe de duas fases principais:

- Fase de aproximação: é quando a trajetória das variáveis do sistema é conduzida em tempo finito desde qualquer estado inicial, para atingir à superfície de deslizamento.
- Fase de modo deslizante: o sistema entra em modo deslizante, ocorrendo uma redução na ordem da dinâmica do sistema, que passa a ser dada pela equação da superfície de deslizamento. Isto é possível devido à superfície criar uma atração do sistema para si mesma.

Então, o problema do controle é que o vetor de estado \mathbf{x} faça um seguimento para um específico vetor de estado desejado $\mathbf{x}_d = [x_d, \dot{x}_d, \dots, x_d^{(n-1)}]^T$ com a presença de uma modelagem imprecisa em f , g e d .

Para que a tarefa de seguimento seja realizável utilizando um controle finito u , o estado inicial desejado $\mathbf{x}_d(0)$ deve ser tal que:

$$\mathbf{x}_d(0) = \mathbf{x}(0) \quad (4-8)$$

Em um sistema de segunda ordem, por exemplo, a posição ou velocidade não pode saltar, de modo que qualquer trajetória desejada possível no tempo $t = 0$, necessariamente, inicia-se com o mesmo valor da posição e velocidade. Caso contrário, o seguimento pode ser atingido apenas após do transitório.

A metodologia considerada é, em base às fases mencionadas para o controle por modos deslizantes, da seguinte forma:

- Obtenção de uma superfície de deslizamento $s(\mathbf{x}; t)$, cuja dinâmica garanta a convergência do erro de seguimento para zero.
- Obtenção de uma lei de controle u descontínua, que garanta atingir a superfície considerada em um tempo finito.

Seja $\tilde{x} = x - x_d$ o erro de seguimento na variável x e seja,

$$\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_d = [\tilde{x}, \dot{\tilde{x}}, \dots, \tilde{x}^{(n-1)}]^T \quad (4-9)$$

o vetor do erro de seguimento. Além disso, vamos definir uma superfície $S = \{\mathbf{x} : s(\mathbf{x}, t) = 0\}$ variante no tempo no espaço de estado \mathbb{R}^n , por uma equação escalar $s(\mathbf{x}; t) = 0$, onde:

$$s(\mathbf{x}; t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} \tilde{\mathbf{x}} \quad (4-10)$$

e λ é uma constante estritamente positiva (que é interpretada como largura de banda do controle desejado). Assim, uma vez que o sistema entra na superfície de deslizamento, o erro de seguimento converge exponencialmente para a origem, com a constante de tempo regulada pelo valor dos ganhos λ , pois a partir daí, o sistema passará a respeitar a dinâmica dada $s(\mathbf{x}, t) = 0$, [62].

De acordo com a equação (4-10), um sistema de 1ª ordem apresenta um único ponto de deslizamento em \mathbb{R}^1 , um sistema de 2ª ordem apresenta uma linha de deslizamento em \mathbb{R}^2 , um sistema de 3ª ordem apresenta um plano de deslizamento em \mathbb{R}^3 , e sistemas de ordem superior a três apresentam um hiperplano de deslizamento em \mathbb{R}^n , como mostra a figura 4.18.

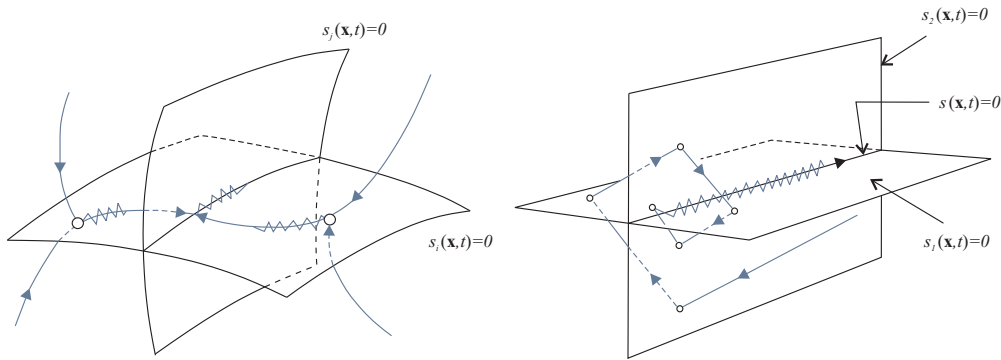


Figura 4.18: Superfícies de Deslizamento.

Por exemplo, se $n = 2$,

$$s = \dot{\tilde{\mathbf{x}}} + \lambda \tilde{\mathbf{x}}$$

desta equação, no deslizamento ($s = 0$), será $\dot{\tilde{\mathbf{x}}} = -\lambda \tilde{\mathbf{x}}$ onde λ indica a dinâmica do erro para a origem.

Dada a condição inicial (4-8), o problema de seguimento $\mathbf{x} \equiv \mathbf{x}_d$ é equivalente ao de permanecer na superfície S para todo $t > 0$; certamente

$s \equiv 0$ representa uma equação diferencial linear cuja única solução é $\tilde{\mathbf{x}} \equiv 0$, dadas as condições iniciais (4-8).

A variável s representa uma medida do desempenho do sistema em seguir a referência. Em Slotine e Li [62] definiu-se a relação entre o valor do escalar s e o erro de seguimento x por:

$$|s| \leq \Phi \Rightarrow |\tilde{x}^i(t)| \leq (2\lambda)^i \varepsilon \quad i = 0, 1, 2, \dots, n \quad (4-11)$$

onde $\varepsilon = \Phi/\lambda^{n-1}$ é a largura da camada limite, e o termo Φ é chamado de espessura da camada limite, e representa a distância da resposta do sistema em relação à superfície de deslizamento S . Para um sistema de segunda ordem, a relação será

$$|\tilde{x}(t)| \leq \frac{\Phi}{\lambda} \quad (4-12)$$

Seja o sinal de controle u descontínuo dado por:

$$u = \begin{cases} u^+ & \text{si } s(\mathbf{x}, t) > 0 \\ u^- & \text{si } s(\mathbf{x}, t) < 0 \end{cases} \quad (4-13)$$

onde $u^+(x)$ e $u^-(x)$ são funções contínuas. Nota-se que u não é definido em $s(\mathbf{x}, t) = 0$

Para que o sistema entre em modo deslizante, a superfície de deslizamento deve ser pelo menos localmente atrativa. Por exemplo, deve existir um domínio envolvendo a superfície no qual as trajetórias do sistema apontem na sua direção. Este fato pode ser matematicamente expresso da seguinte forma:

$$\lim_{\dot{s} \rightarrow 0^+} \dot{s} < 0 \quad \text{e} \quad \lim_{\dot{s} \rightarrow 0^-} \dot{s} > 0 \quad (4-14)$$

em algum domínio $\Omega \in \mathbb{R}^n$. Neste caso, a superfície de deslizamento seria:

$$\mathfrak{D} = S \cap \Omega = \{\mathbf{x} : s(\mathbf{x}, t) = 0\}$$

A expressão dada na equação (4-14) é frequentemente substituída pelo critério mais sucinto, porém equivalente, dado por:

$$\dot{s}s < 0 \quad (4-15)$$

As condições (4-14) e (4-15) são chamadas de condições de alcançabilidade. No caso da condição (4-15) ser satisfeita globalmente, por exem-

plano $\Omega = \mathbb{R}^n$, a função

$$V(s) = \frac{1}{2}s^2 < 0 \tag{4-16}$$

será uma função de Lyapunov para o estado s , uma vez que $\dot{V}(s) = \dot{s}s < 0$.

Embora as condições (4-14) e (4-15) sejam frequentemente encontradas na literatura, elas não garantem a existência de um modo deslizante ideal, já que estas condições garantem apenas que a superfície de deslizamento seja alcançada assintoticamente.

Para garantir que a superfície de deslizamento seja alcançada em tempo finito, uma condição mais restritiva deve ser satisfeita. Uma condição muito utilizada na literatura é a condição de alcançabilidade $-\eta$ dada por:

$$\dot{s}s \leq -\eta|s| \tag{4-17}$$

onde η é uma constante estritamente positiva responsável pela velocidade de convergência do sistema.

Reescrevendo a equação (4-16) como

$$\frac{1}{2} \frac{d}{dt} s^2 \leq -\eta|s| \tag{4-18}$$

que é chamada a condição de deslizamento, e integrando de 0 a t_s , segue que:

$$|s(t_s)| - |s(0)| \leq -\eta t_s$$

O comportamento típico do sistema imposta pela condição de deslizamento é apresentada na figura 4.19.

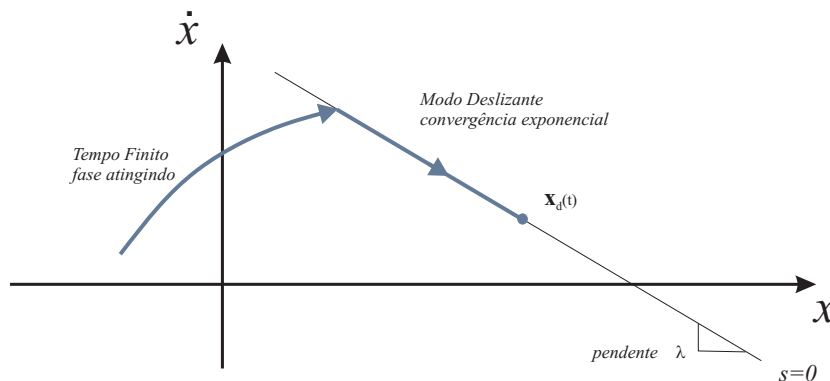


Figura 4.19: Interpretação gráfica das equações (4-10) e (4-18) ($n = 2$).

Deste modo, o tempo necessário para atingir a superfície $s = 0$, repre-

sentado por t_s , satisfaz:

$$t_s \leq \frac{|s(0)|}{\eta} \quad (4-19)$$

Nos casos analisados anteriormente, esta condição é satisfeita, garantindo assim o aparecimento do deslizamento ideal [63]. Além disso, a condição de deslizamento impõe que a distância entre uma trajetória qualquer fora da superfície de deslizamento e a mesma, medida por s^2 , decresce ao longo do tempo, com uma velocidade de convergência η .

A lei de controle u é projetada de forma a garantir que x atinge a superfície $s(x, t) = 0$ em um intervalo de tempo finito, e uma vez atingida tal superfície, permaneça deslizando nela indefinidamente. Alguns métodos utilizados para escolher o sinal de controle descontínuo são:

- Controle Modo Deslizante pelo método de controle equivalente proposto [61].

$$u = u_{eq} + u_c \quad (4-20)$$

onde u_c é um sinal de controle descontínuo com chaveamento, e se escolhe geralmente com o formato tipo $-K \operatorname{sgn}(s)$, com $K > 0$.

- Controle Modo Deslizante tipo controle *Bang-Bang* ([64]). Escolhendo um sinal de controle que pode envolver a técnica de chaveamento.

$$u = -K \operatorname{sgn}(s) \quad (4-21)$$

onde $K > 0$ se escolhe suficientemente maior, como para eliminar as perturbações.

Uma forma alternativa de se representar modos deslizantes é através do conceito de controle equivalente. Este método consiste, basicamente, em encontrar um controle contínuo u_{eq} , chamado controle equivalente, que para uma dada condição inicial sobre a superfície deslizante, a trajetória resultante do sistema

$$\dot{x}^{(n)} = f(\mathbf{x}, t) + g(\mathbf{x}, t)u_{eq} + d(t)$$

coincida com a trajetória descrita pelo sistema a estrutura variável (4-7) e (4-13), e permita conhecer a dinâmica do sistema durante o deslizamento.

Para encontrar o controle equivalente, é suficiente notar que $\dot{s}(\mathbf{x}, t) = 0$ é uma condição necessária para que as trajetórias de estado permaneçam na

superfície de deslizamento $s(\mathbf{x}, t) = 0$, pois

$$\dot{s}(\mathbf{x}, t) = \frac{\partial s}{\partial \mathbf{x}} \dot{\mathbf{x}} = \frac{\partial s}{\partial \mathbf{x}} [f(\mathbf{x}, t) + g(\mathbf{x}, t)u_{eq} + d(\mathbf{x}, t)] = 0 \quad (4-22)$$

Resolvendo a equação para u_{eq} ,

$$u_{eq} = - \left(\frac{\partial s}{\partial \mathbf{x}} g(\mathbf{x}, t) \right)^{-1} \frac{\partial s}{\partial \mathbf{x}} [f(\mathbf{x}, t) + d(\mathbf{x}, t)] \quad (4-23)$$

Para que exista o controle equivalente, é necessário que $\left(\frac{\partial s}{\partial \mathbf{x}} g(\mathbf{x}, t) \right)$ seja inversível.

Além disso, seja a função $d(\mathbf{x}, t)$ dentro do domínio de $g(\mathbf{x}, t)$, então desta condição significa que existe um sinal de controle u_d , tal que

$$d(\mathbf{x}, t) = g(\mathbf{x}, t)u_d \quad (4-24)$$

Esta condição é conhecida como *matching condition* (ou condição de encaixe). Se a condição de encaixe for cumprida, então poderemos escrever a dinâmica do sistema como

$$\dot{\mathbf{x}} = f(\mathbf{x}, t) + g(\mathbf{x}, t)u_{eq} + g(\mathbf{x}, t)u_d \quad (4-25)$$

e substituindo u_{eq} da equação obtemos

$$\dot{\mathbf{x}} = f(\mathbf{x}, t) - g(\mathbf{x}, t) \left(\frac{\partial s}{\partial \mathbf{x}} g(\mathbf{x}, t) \right)^{-1} \left(\frac{\partial s}{\partial \mathbf{x}} g(\mathbf{x}, t) \right) f(\mathbf{x}, t) \quad (4-26)$$

Da equação se comprova como a dinâmica sobre a superfície deslizante é imune para a perturbação $d(\mathbf{x}, t)$ para qualquer superfície $s(\mathbf{x}, t) = 0$, o que mostra que se a condição de encaixe é satisfeita, o controle por modos deslizantes é invariante sob perturbações paramétricas e, portanto, robusto [65].

A lei de controle na equação (4-20) possui um termo descontínuo que depende do valor da variável s . Este termo pode provocar oscilação elevada de alta frequência na ação de controle quando o sistema está próximo à superfície S . A figura 4.20 ilustra este comportamento, denominado *chattering* (ou chaveamento).

Estas oscilações podem excitar modos não modelados (modos de altas frequências) além de provocar o desgaste dos atuadores. Este comportamento tende a ser mais pronunciado quanto maior forem as incertezas do sistema, ou seja, erro de modelagem e perturbação.

Segundo Slotine e Li, para evitar o *chattering*, deve-se suavizar a função

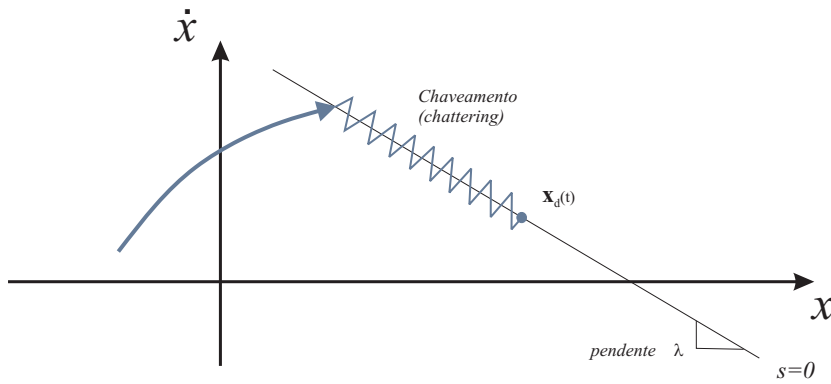


Figura 4.20: Fenômeno de *Chattering* (ou Chaveamento).

discontínua sgn utilizada na lei de controle, definindo-se uma camada limite em torno da superfície S dentro da qual ocorrerá a transição de sinal

A lei de controle suavizada será dada por:

$$u = u_{eq} - K \text{sat}(s/\Phi) \tag{4-27}$$

onde o termo $\text{sat}(s/\Phi)$ é definido como

$$\text{sat}(s/\Phi) = \begin{cases} s/\Phi & \text{se } |s/\Phi| \leq 1 \\ \text{sgn}(s/\Phi) & \text{se } |s/\Phi| > 1 \end{cases} \tag{4-28}$$

como se pode observar na figura 4.21

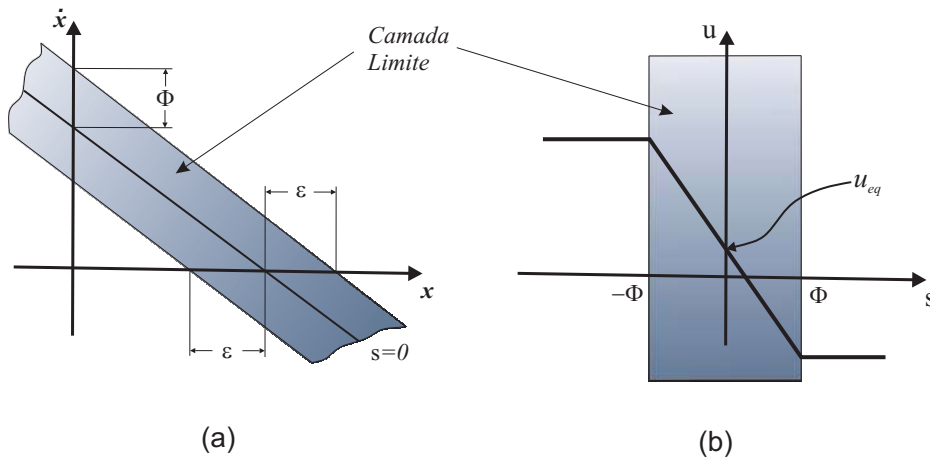


Figura 4.21: (a)Camada Limite para um sistema de segunda ordem. (b)Controle de Interpolação na camada limite.

Portanto, o controle fica introduzido dentro da camada limite quando $\text{sat}(s/\Phi) = s/\Phi$, conforme ilustra a figura 4.22

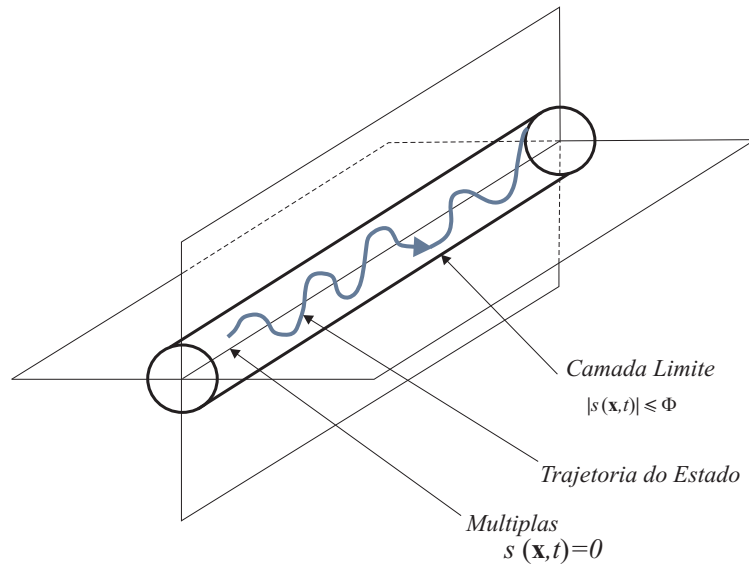


Figura 4.22: Camada Limite.

Fora da camada limite, o controle u satisfaz a condição de deslizamento, e a trajetória converge para dentro da camada limite. O erro de seguimento \tilde{x} fica limitado pela largura da camada limite, conforme

$$|\tilde{x}(t)| < \frac{\Phi}{\lambda^{n-1}} \quad (4-29)$$

Suavizar a lei de controle equivale a introduzir um filtro passa-baixa na dinâmica da variável s e, conseqüentemente, eliminam-se os chaveamentos (*chattering*). Dessa forma, obtém-se uma camada limite, com valor constante ao longo do tempo, uma vez que o parâmetro λ é constante e representa a largura de banda do filtro aplicado sobre a variável s .

Considere-se agora o projeto do controle robusto de trajetória, para robô manipulador. No caso de um sistema MIMO, se tem muitas possibilidades disponíveis para especificar uma condição deslizante análoga a (4-18). Pode-se requerer uma condição deslizante da forma:

$$\frac{1}{2} \frac{d}{dt} s_i^2 \leq -\eta_i |s_i| \quad (\eta > 0)$$

Para ser verificado individualmente ao longo de cada grau de liberdade i , com $s_i = \dot{q}_i + \lambda_i \tilde{q}_i$, se pode interpretar s^2 como a norma quadrada da distância para a superfície $s = 0$, e requerer no caso MIMO que:

$$\frac{1}{2} \frac{d}{dt} \mathbf{s}^T \mathbf{s} \leq -\eta (\mathbf{s}^T \mathbf{s})^{1/2} \quad (\eta > 0)$$

Seja um caso generalizado de um sistema de única entrada, o vetor \mathbf{s} define-se como

$$\mathbf{s} = \dot{\tilde{\mathbf{q}}} + \Lambda \tilde{\mathbf{q}} \quad (4-30)$$

onde $\tilde{\mathbf{q}} = \mathbf{q} - \mathbf{q}_d$, e Λ é uma matriz simétrica positivo-definida, ou também uma matriz de Hurwitz.

Na verdade, neste sistema de segunda ordem, podemos interpretar \mathbf{s} de (4-30) como termo da "velocidade de erro"

$$\mathbf{s} = \dot{\tilde{\mathbf{q}}} + \Lambda \tilde{\mathbf{q}} = \dot{\mathbf{q}} - \dot{\mathbf{q}}_r \quad (4-31)$$

onde,

$$\dot{\mathbf{q}}_r = \dot{\mathbf{q}}_d - \Lambda \tilde{\mathbf{q}}$$

A "velocidade de referência", vetor $\dot{\mathbf{q}}_r$, é fornecida pelo deslocamento da velocidade desejada $\dot{\mathbf{q}}_d$ de acordo com o erro de posição $\tilde{\mathbf{q}}$. Esta simples representação é uma manipulação notacional que permite traduzir as propriedades relacionadas com a energia nas propriedades do controle de trajetória.

De forma semelhante ao caso de entrada única, o vetor \mathbf{s} transmite informação sobre a limitação e convergência de \mathbf{q} e $\dot{\mathbf{q}}$, uma vez que a definição em (4-31) de \mathbf{s} também pode ser vista como uma equação diferencial estável de primeira ordem em $\tilde{\mathbf{q}}$, com \mathbf{s} como uma entrada. Assim, assumindo condições iniciais limitadas, ao mostrar a limitação de \mathbf{s} também se mostram as limitações de $\tilde{\mathbf{q}}$ e $\dot{\tilde{\mathbf{q}}}$, e portanto de \mathbf{q} e $\dot{\mathbf{q}}$; similarmente, se \mathbf{s} tende a 0 quando t tende ao infinito, então os vetores $\tilde{\mathbf{q}}$ e $\dot{\tilde{\mathbf{q}}}$ também tendem a 0.

Na presença de torques de gravidade externas $\mathbf{G}(\mathbf{q})$, a conservação de energia pode ser escrita (generalizando a equação) como

$$\dot{\mathbf{q}}^T (\boldsymbol{\tau} - \mathbf{G}) = \frac{d}{dt} \frac{1}{2} [\dot{\mathbf{q}}^T \mathbf{H} \dot{\mathbf{q}}] \quad (4-32)$$

No entanto, como detalhado abaixo, definindo a matriz \mathbf{C} pelas componentes

$$C_{ij} = \frac{1}{2} \sum_{k=1}^n \frac{\partial H_{ij}}{\partial q_k} \dot{q}_k + \sum_{k=1}^n \left(\frac{\partial H_{ik}}{\partial q_j} - \frac{\partial H_{jk}}{\partial q_i} \right) \dot{q}_k \quad (4-33)$$

então a matriz $(\dot{\mathbf{H}} - 2\mathbf{C})$ é certamente anti-simétrica, um fato que pode ser visto como uma expressão matriz da conservação da energia (Apêndice B).

Na prática, para robôs manipuladores com poucos graus de liberdade, essa igualdade pode ser suficiente para obter \mathbf{C} sem usar explicitamente a

expressão (4-33).

Com uma maior formalização da conservação da energia mecânica, estamos prontos para resolver o problema de controle robusto de trajetória. Vamos definir a função:

$$V(t) = \frac{1}{2}[\mathbf{s}^T \mathbf{H} \mathbf{s}] \quad (4-34)$$

Derivando a equação (4-34) se obtém,

$$\dot{V}(t) = \mathbf{s}^T \mathbf{H} \dot{\mathbf{s}} + \frac{1}{2} \mathbf{s}^T \dot{\mathbf{H}} \mathbf{s} \quad (4-35)$$

agora derivando também a equação (4-31),

$$\dot{\mathbf{s}} = \ddot{\mathbf{q}} - \ddot{\mathbf{q}}_r$$

e substituindo na equação (4-35) obtemos

$$\dot{V}(t) = \mathbf{s}^T (\mathbf{H} \ddot{\mathbf{q}} - \mathbf{H} \ddot{\mathbf{q}}_r) + \frac{1}{2} \mathbf{s}^T \dot{\mathbf{H}} \mathbf{s}$$

Da equação dinâmica do robô, despejamos $\mathbf{H} \ddot{\mathbf{q}}$ e substituímos no sistema dinâmico assim,

$$\mathbf{H} \ddot{\mathbf{q}} = \boldsymbol{\tau} - \mathbf{C} \dot{\mathbf{q}} - \mathbf{G} = \boldsymbol{\tau} - \mathbf{C}(\mathbf{s} - \dot{\mathbf{q}}_r) - \mathbf{G}$$

produzindo a seguinte expressão

$$\dot{V}(t) = \mathbf{s}^T (\boldsymbol{\tau} - \mathbf{H} \ddot{\mathbf{q}}_r - \mathbf{C} \dot{\mathbf{q}}_r - \mathbf{G}) \quad (4-36)$$

Onde a matriz anti-simétrica $(\mathbf{H} - 2\mathbf{C})$ tem sido utilizada para eliminar o termo $\frac{1}{2} \mathbf{s}^T \dot{\mathbf{H}} \mathbf{s}$.

De forma semelhante ao caso de entrada única, vamos agora definir a entrada de controle para ser da forma

$$\boldsymbol{\tau} = \hat{\boldsymbol{\tau}} - \mathbf{k} \text{sgn}(\mathbf{s}) \quad (4-37)$$

onde $\mathbf{k} \text{sgn}(\mathbf{s})$ é definido como o vetor de componentes $k_i \text{sgn}(s_i)$, e similarmente para o termo $\hat{\boldsymbol{\tau}}$, é o vetor de entrada de controle que faria a \dot{V} igual a zero se as dinâmicas foram exatamente conhecidas.

$$\hat{\boldsymbol{\tau}} = \hat{\mathbf{H}} \ddot{\mathbf{q}}_r + \hat{\mathbf{C}} \dot{\mathbf{q}}_r + \hat{\mathbf{G}}$$

Temos então:

$$\dot{V}(t) = \mathbf{s}^T [\tilde{\mathbf{H}}(\mathbf{q})\ddot{\mathbf{q}}_r + \tilde{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r + \tilde{\mathbf{G}}(\mathbf{q})] - \sum_{i=1}^n k_i |s_i| \quad (4-38)$$

Dados uns intervalos de erros na modelagem $\tilde{\mathbf{H}}$, $\tilde{\mathbf{C}}$ e $\tilde{\mathbf{G}}$ onde,

$$\tilde{\mathbf{H}} = \hat{\mathbf{H}} - \mathbf{H} \quad \tilde{\mathbf{C}} = \hat{\mathbf{C}} - \mathbf{C} \quad \tilde{\mathbf{G}} = \hat{\mathbf{G}} - \mathbf{G}$$

vemos que escolhendo as componentes k_i do vetor \mathbf{k} , tal que

$$k_i \geq |[\tilde{\mathbf{H}}(\mathbf{q})\ddot{\mathbf{q}}_r + \tilde{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r + \tilde{\mathbf{G}}(\mathbf{q})]_i| + \eta_i \quad (4-39)$$

(onde a constante η_i é estritamente positivo) Permite-se satisfazer a condição de deslizamento.

$$\dot{V} \leq - \sum_{i=1}^n \eta_i |s_i| \quad (4-40)$$

Vamos agora simular o algoritmo de controle robusto por modos deslizantes, com uma taxa de amostragem igual a 5KHz, pois para este tipo de controlador é recomendável trabalhar com uma taxa de amostragem entre 5KHz a 10Khz. Os parâmetros do controlador utilizados, calibrados empiricamente, são:

$$\Lambda = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 \end{bmatrix} \quad \text{e} \quad \eta = \begin{bmatrix} 2000 \\ 2000 \\ 800 \\ 500 \\ 500 \\ 10 \end{bmatrix}$$

Os parâmetros da função continua contida na lei do controlador são:

$$\Phi = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \end{bmatrix}$$

As figuras 4.23 e 4.24 mostram o resultado do controlador robusto, que atua de forma eficiente para uma trajetória variável. Na estimação das matrizes \hat{H} , \hat{C} , \hat{G} , se considerou 15% de incerteza em relação à dinâmica modelada real do robô.

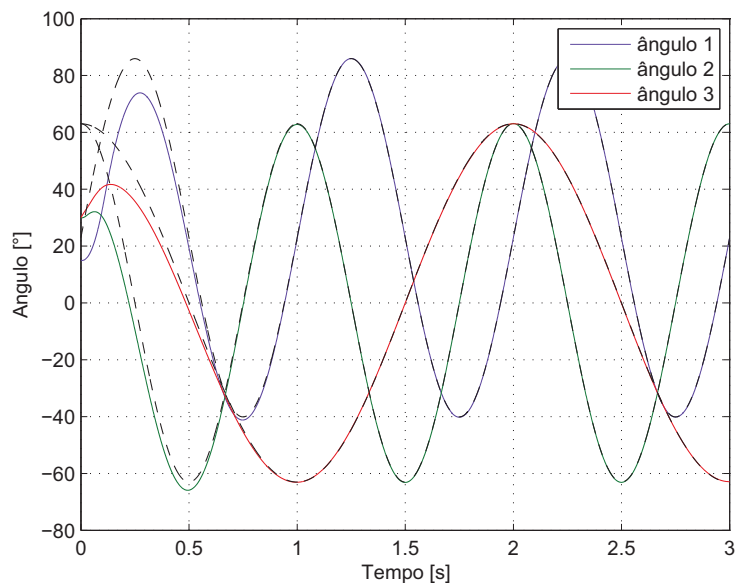


Figura 4.23: Referência e resposta do seguimento de trajetória: juntas q_1 , q_2 e q_3 .

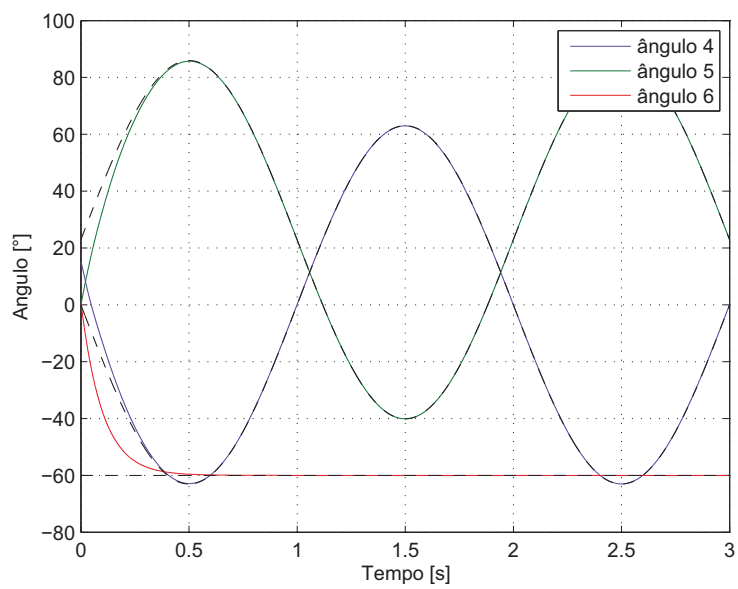


Figura 4.24: Referência e resposta do seguimento de trajetória: juntas q_4 , q_5 e q_6 .

Os erros são apresentados nas figuras 4.25 e 4.26.

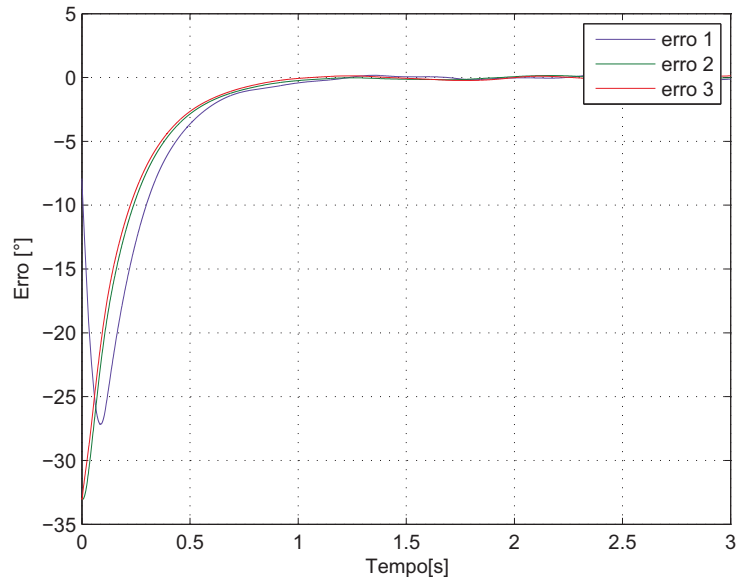


Figura 4.25: Erros de posicionamento: juntas \tilde{q}_1 , \tilde{q}_2 e \tilde{q}_3 .

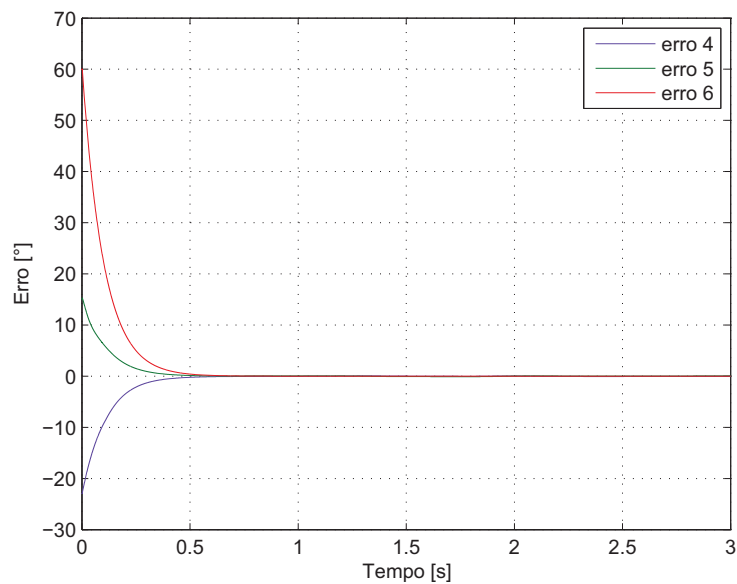


Figura 4.26: Erros de posicionamento: juntas \tilde{q}_4 , \tilde{q}_5 e \tilde{q}_6 .

Na figura 4.27 e 4.28 mostramos os resultados dos torques produzidos pelo controlador robusto. Observa-se como o sistema é atingido na superfície deslizante porque no início da sinal aparece um pequeno chaveamento, com isto podemos comprovar que a superfície de deslizamento esta cumprindo sua função. Em alguns casos se a superfície deslizante não fora a correta então o

controle poderia comportar-se simplesmente como um controlador PD, porque sua operação poderia estar somente atuando dentro da camada limite.

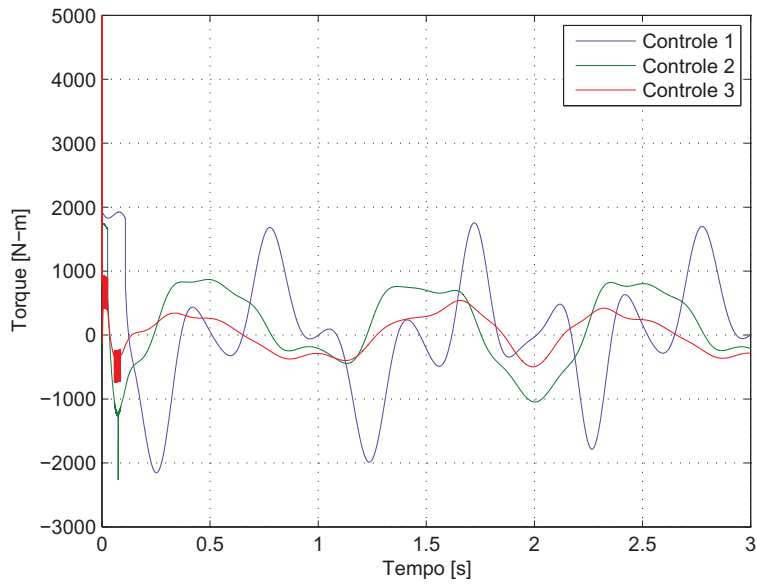


Figura 4.27: Gráfico dos torques produzidos: τ_1 , τ_2 e τ_3 .

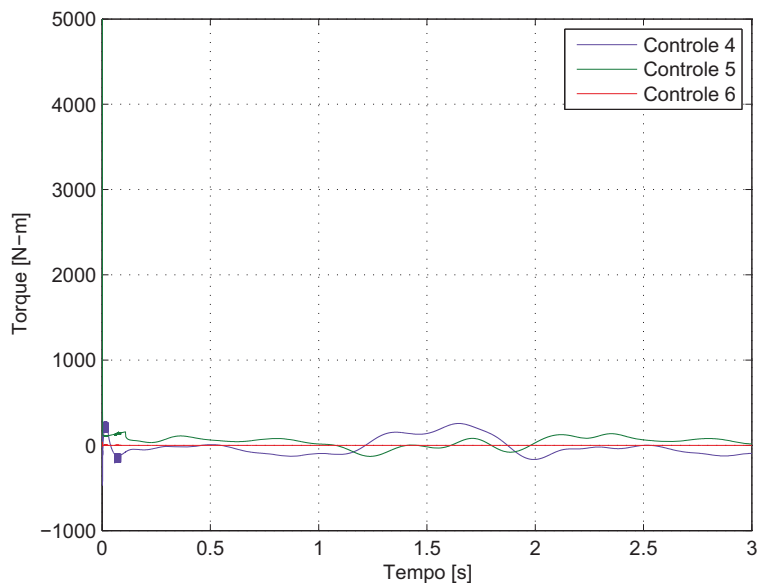


Figura 4.28: Gráfico dos torques produzidos: τ_4 , τ_5 e τ_6 .

As figuras 4.29 e 4.30 mostram as velocidades de referência utilizadas na representação das propriedades da energia no projeto do algoritmo do

controlador. Observa-se que a velocidade de referência faz um seguimento eficiente da trajetória da velocidade desejada.

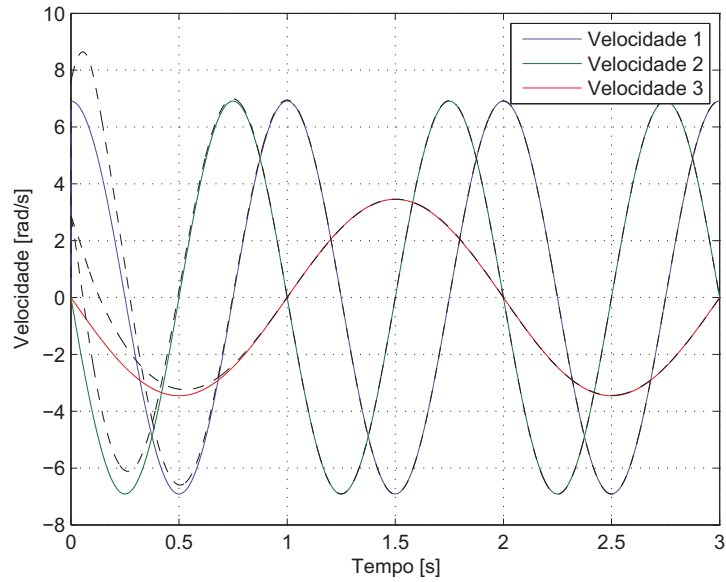


Figura 4.29: Referência e Resposta da Velocidade: \dot{q}_1 , \dot{q}_2 e \dot{q}_3 .

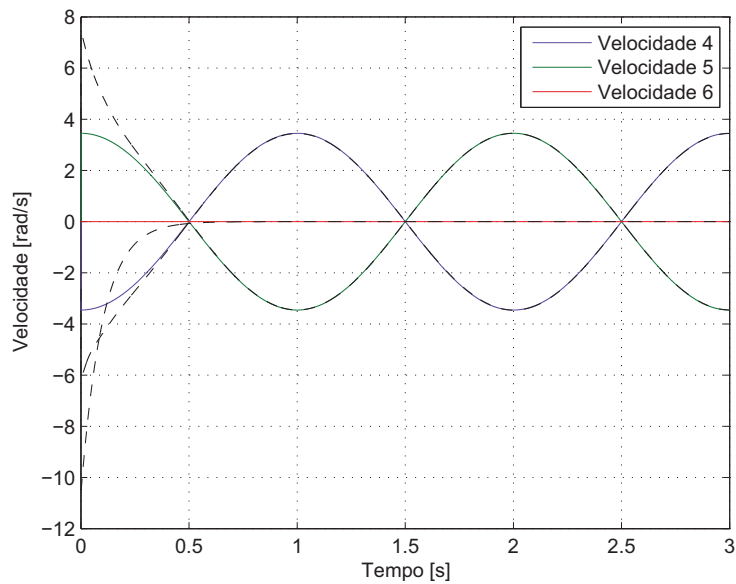


Figura 4.30: Referência e Resposta da Velocidade: \dot{q}_4 , \dot{q}_5 e \dot{q}_6 .

Outro aspecto que deve ser ressaltado é a robustez deste tipo de controlador. Apesar das incertezas e das perturbações existentes, as trajetórias

do sistema continuarem apontando em direção a superfície de deslizamento. O sistema continuará entrando em modo deslizante, apresentando o mesmo desempenho, governado pela dinâmica referente a equação da superfície deslizante.

Na figura 4.31, mostra as superfícies de deslizamento desenhadas no algoritmo de controle, e como pode-se ver os estados atingem a superfície de deslizamento dentro do entorno limitado por Φ .

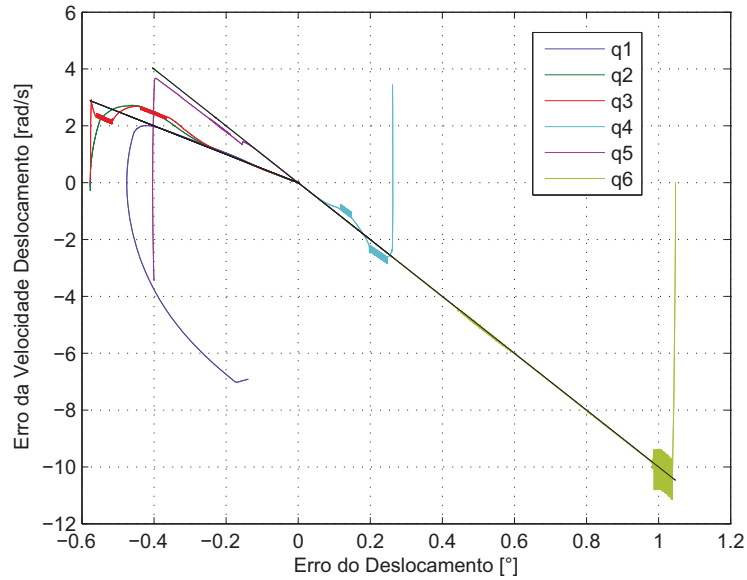


Figura 4.31: Superfícies Deslizantes do Sistema.

Neste movimento, denominado de modo deslizante, a trajetória do estado se desloca por uma superfície denominada superfície de deslizamento, denotada por $s(x, t) = 0$. Por outro lado, no espaço de estado, o chaveamento ocorre em uma superfície denominada de superfície de chaveamento. Embora no caso analisado estas duas superfícies se confundam, este fato nem sempre é verdadeiro.

5

Dispositivo Háptico de 5 Graus de Liberdade

Neste capítulo, são descritas as características funcionais da interface háptica Novint Falcon[®], sendo uns dos dispositivos hápticos mais baratos do mercado. Além disso, vamos fazer um estudo de sua modelagem, cuja configuração é semelhante ao mecanismo de um robô paralelo tipo delta. Também mostraremos a construção de um Dispositivo Háptico de 5 graus de liberdade, usando dois Falcon's acoplados mecanicamente com elos rígidos e rótulas nas juntas. Neste novo sistema, finalmente implementaremos um controle para compensar a massa adicionada aos efetadores terminais dos Falcon's.

5.1

Descrição do Falcon

Desenvolvido pela Novint Technologies Inc., o Falcon é um dispositivo de interação que permite uma manipulação force-feedback (retroalimentação de força física) por parte do computador. É utilizado principalmente para interagir com objetos virtuais no ambiente do jogo, basicamente, o Falcon é um pequeno “motor” que permite ao usuário sentir e interagir dentro de jogos de computador 3D.

O Novint Falcon[®] deixa o usuário controlar o jogo em três dimensões e, ao mesmo tempo, sentir a resposta das forças também em três dimensões, de forma fiel à realidade. Este aparelho move-se para a esquerda, direita, cima e baixo tal e qual como um mouse mas também para dentro e para fora de sua estrutura de modo a possibilitar uma experiência mais realista de toque 3D dentro de um jogo. Os motores estão sincronizados com os dados gráficos e físicos do jogo, deixando o utilizador sentir texturas, formas, peso e dimensões. É possível alterar a extremidade onde o usuário segura o Falcon. A peça padrão é uma esfera. Na figura 5.1 se mostra este dispositivo, com uma esfera e com um pistola. Se comunica com o computador através de uma ligação USB 2.0, e suas características principais são:

- Área de trabalho 3D igual 101,6mm x 101,6mm x 10,16cm.
- Capacidade de Força 10 N.



Figura 5.1: O Dispositivo Háptico Novint Falcon®

- Taxa de Quadros 1KHz.
- Resolução da posição 400dpi.
- Rigidez 5N/mm.
- Efeitos Dinâmicos (inércia, peso, momento).
- Potência 30W, tensão de alimentação 100V-240V.
- Tamanho 22.86cm x 22.86cm x 22.86cm
- Peso 2.72 Kg.
- Custo: US\$100 a US\$300.

Na figura 5.2 se apresentam as partes que compõem toda a estrutura do Novint Falcon®. Este dispositivo tem três motores de corrente contínua, que são o que produzem as forças de sensações para o usuário.

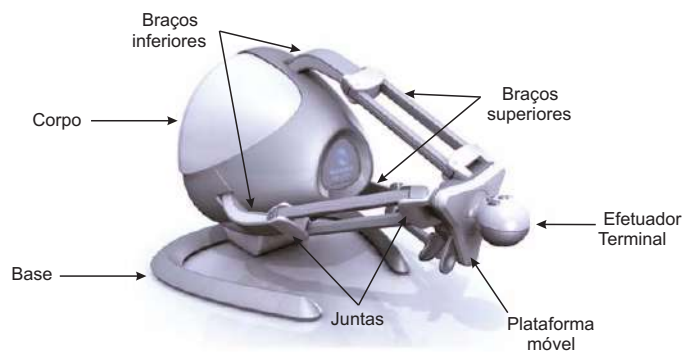


Figura 5.2: Membros do dispositivo háptico Novint Falcon®

5.2 Modelagem do Falcon

Nesta seção, vamos desenvolver a modelagem do Novint Falcon[®]. a configuração desse dispositivo háptico foi em base ao robô manipulador paralelo desenvolvido pela Universidade de Maryland ([66]). Esse Robô manipulador, mostrado na figura 5.3, utilizava apenas juntas rotativas para restringir a saída da plataforma móvel para o movimento translacional.



Figura 5.3: Robô Manipulador Paralelo da Universidade de Maryland

O dispositivo Novint Falcon[®] apresenta um mecanismo modificado em relação ao robô manipulador de Maryland [66], Para nosso estudo nós referenciaremos em parte desse robô para poder achar a correta modelagem. A figura 5.4 mostra o mecanismo do Falcon em CAD, fornecida pela Novint.



Figura 5.4: Modelo CAD do Novint Falcon[®]

5.2.1 Geometria do Manipulador

Um diagrama esquemático do Novint Falcon[®] está esboçado na figura 5.5, onde a base fixa é chamada elo 0, e a plataforma móvel é chamada elo 16. Três membros idênticos são conectados à plataforma base fixa. Cada membro consiste de um braço na parte superior, e um braço na parte inferior. Os braços inferiores são chamados elos 1, 2 e 3. Cada braço superior é um um paralelogramo de quatro barras planas, conectando os elos 4, 7, 10 e 13 para a primeira parte; 5, 8, 11 e 14 para a segunda parte; e 6, 9, 12 e 15 para a terceira parte. Todos os elos e plataformas são considerados corpos rígidos.

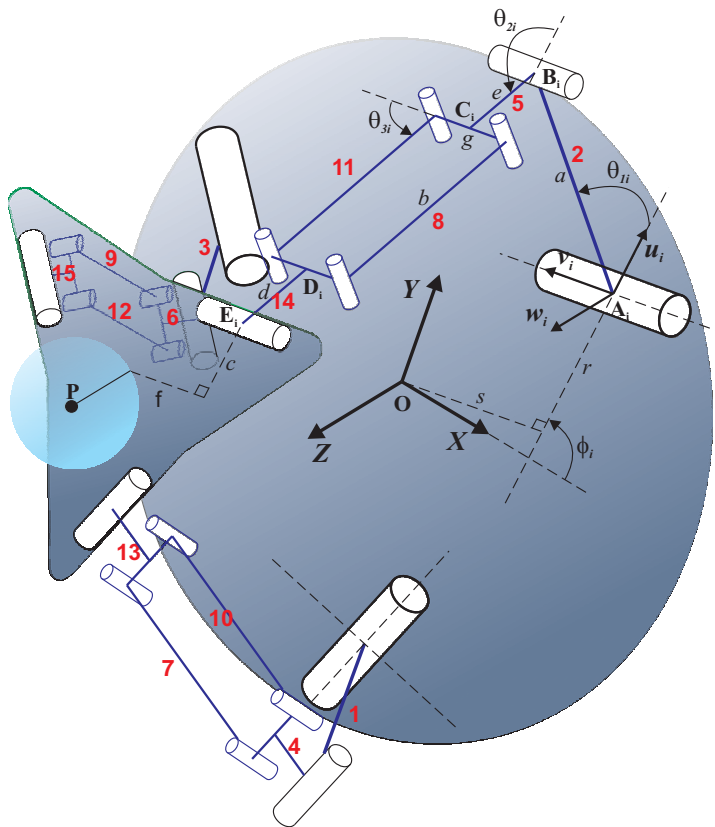


Figura 5.5: Representação esquemática do Novint Falcon[®]

Para cada membro, os braços superiores e inferiores e as duas plataformas são conectados por três juntas rotativas paralelas nos pontos A , B , e E . Os eixos dessas juntas rotativas são perpendiculares aos eixos do paralelogramo de quatro barras para cada membro. Existe também um pequeno desvio entre a montagem do braço superior e os eixos das juntas rotativas em B e E . Estes desvios são uma função da geometria dos elos 4, 5, 6, 13, 14 e 15. Os eixos A_1 , A_2 e A_3 ficam em um plano ligado à plataforma fixa. Da mesma forma, os

eixos E_1 , E_2 e E_3 ficam em um plano ligado à plataforma móvel. O sistema de referência XYZ é fixado à base (fixa) no ponto O , localizado no centro da plataforma (fixa). Os eixos x , e y ficam no mesmo plano, como se definiu para os eixos A_1 , A_2 e A_3 . O ângulo ϕ_i para i -ésimo suporte, mostrado na figura 5.6, define a orientação angular do suporte relativo ao sistema XYZ na plataforma fixa. Outro sistema de coordenada $U_iV_iW_i$ é ligado à base fixa em A_i para cada suporte, tal que o eixo u_i é perpendicular ao eixo de rotação da junta em A_i e faz um ângulo ϕ_i a partir do eixo x , no plano da plataforma fixa. O eixo v_i está ao longo do eixo da junta em A_i .

O i -ésimo suporte do manipulador é mostrado na figura 5.6, o vetor $\bar{\mathbf{p}}$ é o vetor posição do ponto P no sistema coordenado XYZ , onde P está ligado no centro da plataforma móvel. O ângulo θ_{1i} é medido a partir de $\bar{\mathbf{u}}$ em relação a \overline{AB} . O ângulo θ_{2i} é definido desde a direção $\bar{\mathbf{u}}$ até \overline{BC} . O ângulo θ_{3i} é definido pelo ângulo desde a direção de $\bar{\mathbf{v}}$ até \overline{CD} . Os comprimentos dos elos são também mostrados na figura 5.6.

Para este robô manipulador, os ângulos θ_{11} , θ_{22} e θ_{33} são considerados os ângulos dos atuadores fixados nas juntas. Esta configuração oferece a vantagem de poder localizar todos os atuadores na base fixa, reduzindo a inércia das partes móveis do sistema.

5.2.2

Cinemática Inversa

O objetivo de solucionar a cinemática inversa é definir um mapeamento da posição da plataforma, movendo-se em um espaço cartesiano, para o conjunto dos ângulos das juntas que permitam atingir essa posição. Para esse análise, considera-se a posição da plataforma móvel conhecida, dada pela posição do vetor $\bar{\mathbf{p}}$, que define a localização de P no centro da plataforma móvel no sistema coordenado XYZ . A análise da cinemática inversa produz um conjunto de três ângulos nas juntas para cada suporte (θ_{1i} , θ_{2i} e θ_{3i} para o i -ésimo suporte) que definem as possíveis posições de cada suporte para determinada posição da plataforma em movimento. Basicamente, para esse robô manipulador paralelo, o problema é encontrar os ângulos dos atuadores (θ_{11} , θ_{22} e θ_{33}) necessários para posicionar a plataforma móvel no valor desejado. Uma abordagem intuitiva para a solução é a de considerar o problema geometricamente.

Surpreendentemente, o este problema da cinemática inversa para o Falcon está determinado pelas restrições redundantes do mecanismo. A figura 5.6 mostra um pequeno desvio s para cada suporte, medido ao longo do eixo da junta do atuador (direção de $\bar{\mathbf{v}}$). Outras características e pequenas distâncias que

vão desde a plataforma da base (fixa) até na terra para o centro da plataforma móvel (onde fica o efetuator terminal) são também consideradas. Os valores daquelas distâncias estão em [67]. Da figura 5.6 a equação de malha fechada pode ser escrita para cada membro por:

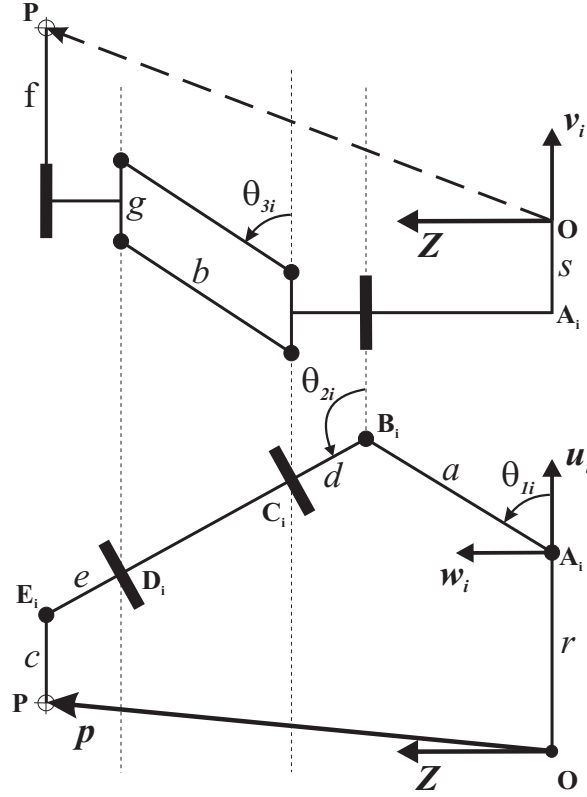


Figura 5.6: Vista frontal e superior de um suporte do Novint Falcon®

$$\overline{A_i B_i} + \overline{B_i E_i} = \overline{O_i P_i} + \overline{P_i E_i} - \overline{O_i A_i} \quad (5-1)$$

A seguinte transformação expressa a posição de P no sistema coordenado UVW ligado ao ponto A para cada suporte i :

$$\begin{bmatrix} p_{ui} \\ p_{vi} \\ p_{wi} \end{bmatrix} = \begin{bmatrix} \cos \phi_i & \sin \phi_i & 0 \\ -\sin \phi_i & \cos \phi_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} + \begin{bmatrix} -r \\ s \\ 0 \end{bmatrix} \quad (5-2)$$

A partir da equação (5-1), considerando-se para o Falcon $d = e$, então as

expressões para p_{ui} , p_{vi} e p_{wi} são

$$p_{ui} = a \cos \theta_{1i} - c + [2d + b \sin \theta_{3i}] \cos \theta_{2i} \quad (5-3)$$

$$p_{vi} = b \cos \theta_{3i} + f \quad (5-4)$$

$$p_{wi} = a \sin \theta_{1i} + [2d + b \sin \theta_{3i}] \sin \theta_{2i} \quad (5-5)$$

Da equação (5-4), duas soluções são imediatamente encontradas para θ_{3i} :

$$\theta_{3i} = \pm \arccos \left(\frac{p_{vi} - f}{b} \right) \quad (5-6)$$

Agora, com θ_{3i} conhecido, vamos gerar uma equação com θ_{1i} como a única incógnita. Para isto, isolamos os termos θ_{2i} nas equações (5-3) e (5-5), aplicamos a soma dos quadrados nas duas equações de modo que θ_{2i} seja eliminado, e aplicando a relação de Pitágoras, se obtém

$$\begin{aligned} (p_{ui} + c)^2 + p_{wi}^2 + a^2 - 2a(p_{ui} + c) \cos \theta_{1i} - 2ap_{wi} \sin \theta_{1i} \\ = 4d^2 + 4db \sin \theta_{3i} + b^2 \sin^2 \theta_{3i} \end{aligned} \quad (5-7)$$

Para transformar a equação (5-7) numa expressão polinomial, utilizamos a função tangente de ângulo metade, definida por:

$$t_{1i} = \tan \left(\frac{\theta_{1i}}{2} \right) \quad (5-8)$$

produzindo a relação

$$\sin \theta_{1i} = \frac{2t_{1i}}{1 + t_{1i}^2} \quad e \quad \cos \theta_{1i} = \frac{1 - t_{1i}^2}{1 + t_{1i}^2} \quad (5-9)$$

Substituindo as funções trigonométricas do ângulo metade na equação (5-7), resulta na equação simplificada

$$l_{2i}t_{1i}^2 + l_{1i}t_{1i} + l_{0i} = 0, \quad (5-10)$$

onde

$$l_{0i} = p_{wi}^2 + p_{ui}^2 + 2cp_{ui} - 4d^2 - 2ap_{ui} + (a - c)^2 - b^2 \sin^2 \theta_{3i} - 4bd \sin \theta_{3i} \quad (5-11)$$

$$l_{1i} = -4ap_{w1} \quad (5-12)$$

$$l_{2i} = p_{wi}^2 + p_{ui}^2 + 2cp_{ui} + 2ap_{ui} - 4d^2 - b^2 \sin^2 \theta_{3i} - 4bd \sin \theta_{3i} + (a + c)^2 \quad (5-13)$$

A equação (5-10) pode ser solucionada para t_{1i} , produzindo dois possíveis valores para θ_{1i} , dados por

$$t_{1i} = \pm \frac{-l_{1i} + \sqrt{l_{1i}^2 - 4l_{2i}l_{0i}}}{2l_{2i}} \quad (5-14)$$

e da equação (5-8), se obtém:

$$\theta_{1i} = 2 \arctan(t_{1i}) \quad (5-15)$$

Então, se obtêm duas soluções de θ_{1i} , para cada uma das duas soluções de θ_{3i} . Assim, para uma posição dada da plataforma móvel, existem quatro configurações possíveis em cada suporte. Mas somente é aplicável uma solução para o Falcon, devido ao domínio limitado de θ_{1i} (próximo a 90°). Então, somente se utilizará o valor positivo em ambas soluções.

Finalmente, a combinação das equações (5-3) e (5-5) se obtém:

$$\theta_{2i} = \arctan \left(\frac{p_{wi} - a \sin \theta_{1i}}{p_{ui} - a \cos \theta_{1i} + c} \right) \quad (5-16)$$

5.2.3

Cinemática Direta

O objetivo de solucionar a cinemática direta é definir um mapeamento de um conjunto conhecido de ângulos produzidos pelos atuadores, para obter uma posição desconhecida da plataforma móvel. Para este manipulador, os ângulos das juntas que são considerados conhecidos são os ângulos formados pelas ligações de entrada na base do manipulador, θ_{11} , θ_{12} e θ_{13} . A posição desconhecida da plataforma móvel é descrita pela posição do vetor $\bar{\mathbf{p}}$, o qual define a localização do ponto P no centro da plataforma móvel no sistema coordenado XYZ .

Em comparação aos robôs manipuladores seriais, a cinemática direta é mais complexa em solucionar para os robôs paralelos que o problema da

cinemática inversa. Muitas vezes, o problema da cinemática direta para robôs paralelos recai em resolver um grande sistema de equações polinomiais. Três abordagens comuns para resolver esse sistema de equações são descritos por Raghavan e Roth [68]: Eliminação Dialítica, Continuação Polinomial e Bases de Grobner.

A abordagem aplicada que resolve este caso é o procedimento por Eliminação Dialítica de Sylvester (Salmon [69]). Este procedimento é utilizado para eliminar uma ou mais incógnitas a partir de um sistema de equações. Este método foi utilizado por Husain e Waldrom [70] para resolver o problema da cinemática direta e inversa de uma plataforma de três suportes paralelos com duas juntas do atuador e quatro juntas passivas associado com cada membro do suporte. Além disso, este procedimento também foi aplicado para solucionar a cinemática direta da plataforma de Stewart ([71]).

Neste trabalho, não se desenvolverá a cinemática direta, uma vez que não é necessária para a aplicação, que só requer a cinemática inversa.

5.2.4 Jacobiano

Para os robôs manipuladores paralelos, a matriz Jacobiana proporciona uma transformação da velocidade do efetuador terminal no espaço cartesiano para a velocidade dos atuadores, mas no espaço das juntas do robô [72]. A matriz Jacobiana tem uma representação diferente da tradicional definida para os manipuladores seriais. Essa mudança da matriz Jacobiana para manipuladores paralelos segue uma naturalidade da dualidade entre manipuladores paralelos e seriais ([73]), e é feita por questão de conveniência.

Como uma extensão desta definição, Gosselin e Angeles [72], apresentaram um Jacobiano em duas partes. Seja a variável dos atuadores nas juntas denotada por um vetor \mathbf{q} de n -dimensões, e a localização da plataforma movimentada descrita por um vetor \mathbf{x} de m -dimensões, então a restrição cinemática imposta pelos elos em relação às variáveis pode ser escrita na seguinte forma:

$$\mathbf{f}(\mathbf{x}, \mathbf{q}) = \mathbf{0} \quad (5-17)$$

onde \mathbf{f} é uma função implícita de n -dimensões de \mathbf{q} e \mathbf{x} , e $\mathbf{0}$ um vetor de zeros de n -dimensões. Derivando a equação (5-17) com respeito ao tempo, obtemos uma relação entre as velocidade de entrada da junta e a velocidades de saída do efetuador terminal,

$$\mathbf{J}_q \dot{\mathbf{q}} = \mathbf{J}_x \dot{\mathbf{x}} \quad (5-18)$$

onde,

$$\mathbf{J}_x = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \quad e \quad \mathbf{J}_q = -\frac{\partial \mathbf{f}}{\partial \mathbf{q}}$$

são duas matrizes separadas, onde \mathbf{J}_x é uma matriz Jacobiano $n \times m$ e \mathbf{J}_q é uma matriz Jacobiana $n \times n$ ambas tem configurações dependentes, é dizer $\mathbf{J}_x = \mathbf{J}_x(\mathbf{x}, \mathbf{q})$ e $\mathbf{J}_q = \mathbf{J}_q(\mathbf{x}, \mathbf{q})$ por essa razão a matriz Jacobiana \mathbf{J} considerada pode ser escrito como

$$\dot{\mathbf{q}} = \mathbf{J}\dot{\mathbf{x}} \quad (5-19)$$

onde $\mathbf{J} = \mathbf{J}_q^{-1}\mathbf{J}_x$ é uma matriz $n \times m$. Nota-se que a matriz Jacobiana definida na equação (5-19) para um robô manipulador paralelo corresponde à matriz Jacobiana inversa de um robô manipulador serial.

A vantagem da representar em duas partes a matriz Jacobiana é que se pode identificar os diferentes tipos de singularidade do robô. Independente da forma que assuma, A matriz Jacobiana fornece muitas informações úteis para o desempenho de um manipulador, e frequentemente utilizada para fins de geração de trajetórias. A análise do Jacobiano também é usada para determinar as posições singulares de um manipulador, similarmente o Jacobiano é usado para descrever os limites do espaço de trabalho de um manipulador ([74]).

Para o manipulador paralelo considerado neste trabalho, a matriz Jacobiana é uma matriz quadrada

Para esta análise, \mathbf{q} é o vetor dos deslocamentos angulares do atuador nas juntas, e \mathbf{x} é o vetor posição da plataforma móvel:

$$\mathbf{q} = \begin{bmatrix} \theta_{11} \\ \theta_{12} \\ \theta_{13} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad (5-20)$$

As matrizes Jacobianas são obtidas diferenciando a equação da malha fechada para cada suporte do manipulador paralelo, e depois solucionando o sistema de equações resultantes, de modo que

$$\mathbf{J}_q \begin{bmatrix} \dot{\theta}_{11} \\ \dot{\theta}_{12} \\ \dot{\theta}_{13} \end{bmatrix} = \mathbf{J}_x \begin{bmatrix} V_{p,x} \\ V_{p,y} \\ V_{p,z} \end{bmatrix} \quad (5-21)$$

onde $V_{p,x}, V_{p,y}$ e $V_{p,z}$ são as componentes da velocidade do ponto P fixado na plataforma móvel, representado no sistema de coordenados XYZ , e onde \mathbf{J}_q e \mathbf{J}_x são as matrizes 3×3 do Jacobiano. A matriz Jacobiana é função esta em termos dos ângulos das juntas e portanto, depende da posição e pose do

manipulador.

Observe-se pela figura 5.5, que:

$$\overline{OA}_i + \overline{AB}_i = \overline{OP}_i + \overline{PE}_i + \overline{ED}_i + \overline{DC}_i + \overline{CB}_i \quad (5-22)$$

Diferenciando a equação (5-22) com respeito ao tempo, e expressando a equação resultante no sistema coordenado UVW , obtém-se o seguinte:

$$\bar{\omega}_{1i} \times \overline{AB}_i = \bar{V}_{P,uvw} + \bar{\omega}_{3i} \times \overline{DC}_i + \bar{\omega}_{2i} \times (\overline{ED}_i + \overline{CB}_i) \quad (5-23)$$

onde $\bar{\omega}_{ni}$ é a velocidade angular do n -ésimo elo do i -ésimo suporte no sistema coordenado UVW , dado por:

$$\bar{\omega}_{1i} = \begin{bmatrix} 0 \\ -\dot{\theta}_{1i} \\ 0 \end{bmatrix}, \quad \bar{\omega}_{2i} = \begin{bmatrix} 0 \\ -\dot{\theta}_{2i} \\ 0 \end{bmatrix} \quad e \quad \bar{\omega}_{3i} = \begin{bmatrix} \dot{\theta}_{3i} \sin \theta_{2i} \\ -\dot{\theta}_{2i} \\ -\dot{\theta}_{3i} \cos \theta_{2i} \end{bmatrix} \quad (5-24)$$

Reescrevendo a equação (5-23) em termos dos parâmetros dos elos e ângulos das juntas,

$$\begin{bmatrix} a\dot{\theta}_{1i} \sin \theta_{1i} \\ 0 \\ -a\dot{\theta}_{1i} \cos \theta_{1i} \end{bmatrix} = \begin{bmatrix} V_{p,u} - b\dot{\theta}_{3i} \cos \theta_{2i} \cos \theta_{3i} + \dot{\theta}_{2i} \sin \theta_{2i} [2d + b \sin \theta_{3i}] \\ V_{p,v} + b\dot{\theta}_{3i} \sin \theta_{3i} \\ V_{p,w} - b\dot{\theta}_{3i} \sin \theta_{2i} \cos \theta_{3i} - \dot{\theta}_{2i} \cos \theta_{2i} [2d + b \sin \theta_{3i}] \end{bmatrix} \quad (5-25)$$

As matrizes Jacobianas são determinadas considerando a equação (5-25) como um sistema de três equações e três incógnitas, $(\dot{\theta}_{1i}, \dot{\theta}_{2i}$ e $\dot{\theta}_{3i})$, e solucionando $\dot{\theta}_{1i}$ para o i -ésimo suporte. Isso é conseguido solucionando, primeiramente a segunda linha da equação (5-25) para $\dot{\theta}_{3i}$:

$$\dot{\theta}_{3i} = -\frac{V_{p,v}}{b \sin \theta_{3i}} \quad (5-26)$$

Essa expressão para $\dot{\theta}_{3i}$ é substituído na primeira e terceira linha da equação (5-25), deixando duas equações e duas incógnitas $(\dot{\theta}_{1i}$ e $\dot{\theta}_{2i})$. Essas duas equações são isoladas, de modo a eliminar $\dot{\theta}_{2i}$, obtendo

$$\begin{aligned} a\dot{\theta}_{1i} \sin(\theta_{2i} - \theta_{1i}) \sin \theta_{3i} &= V_{p,u} \cos \theta_{2i} \sin \theta_{3i} \\ &V_{p,v} \cos \theta_{3i} + V_{p,w} \sin \theta_{2i} \sin \theta_{3i} \end{aligned} \quad (5-27)$$

A equação (5-27) pode ser transformada do sistema coordenado UVW , do

suporte i , para o sistema coordenado XYZ do manipulador, utilizando

$$\begin{bmatrix} V_{p,ui} \\ V_{p,vi} \\ V_{p,wi} \end{bmatrix} = \begin{bmatrix} \cos \phi_i & \sin \phi_i & 0 \\ -\sin \phi_i & \cos \phi_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_{p,x} \\ V_{p,y} \\ V_{p,z} \end{bmatrix} \quad (5-28)$$

Esta transformação é repetida para cada suporte, e as equações resultantes são reorganizadas, resultando em.

$$\mathbf{J}_q \begin{bmatrix} \dot{\theta}_{11} \\ \dot{\theta}_{12} \\ \dot{\theta}_{13} \end{bmatrix} = \mathbf{J}_x \begin{bmatrix} V_{p,x} \\ V_{p,y} \\ V_{p,z} \end{bmatrix}$$

onde

$$\mathbf{J}_x = \begin{bmatrix} \dot{j}_{x11} & \dot{j}_{x12} & \dot{j}_{x13} \\ \dot{j}_{x21} & \dot{j}_{x22} & \dot{j}_{x23} \\ \dot{j}_{x31} & \dot{j}_{x32} & \dot{j}_{x33} \end{bmatrix}, \quad \mathbf{J}_q = \begin{bmatrix} \dot{j}_{q11} & 0 & 0 \\ 0 & \dot{j}_{q12} & 0 \\ 0 & 0 & \dot{j}_{q13} \end{bmatrix} \quad (5-29)$$

e

$$\begin{aligned} \dot{j}_{xi1} &= \cos \theta_{2i} \sin \theta_{3i} \cos \phi_i - \cos \theta_{3i} \sin \phi_i \\ \dot{j}_{xi2} &= \cos \theta_{3i} \cos \phi_i + \cos \theta_{2i} \sin \theta_{3i} \sin \phi_i \\ \dot{j}_{xi3} &= \sin \theta_{2i} \sin \theta_{3i} \\ \dot{j}_{q1i} &= a \sin(\theta_{2i} - \theta_{1i}) \sin \theta_{3i} \end{aligned} \quad (5-30)$$

para $i = 1, 2$ e 3 .

Algumas vezes, a matriz Jacobiana para um manipulador paralelo não é dimensionalmente uniforme, ou seja, mistura grandeza de comprimento e ângulo. Isso cria alguns problemas ao usar a matriz Jacobiana para outros tipos de análise, tais como para examinar sua rigidez. No entanto, no caso do Falcon temos do

$$\mathbf{J} = \mathbf{J}_q^{-1} \mathbf{J}_x \quad (5-31)$$

Com o Jacobiano J dimensionalmente uniforme para cada elemento tendo a dimensão de $(\text{comprimento})^{-1}$. Isso ocorre porque o manipulador tem apenas 3 graus de liberdade translacionais na plataforma móvel, e todas as juntas são rotativas. Assim, todos os elementos do vetor de saída x para este manipulador têm a dimensão do comprimento, e todos os elementos do vetor de entrada q são

rotações, resultando em uma matriz Jacobiana dimensionalmente uniforme.

5.3 Estática

Nesta seção, vamos descrever a estática para robôs manipuladores paralelos (Falcon). Devido à existência de várias malhas fechadas, o método de análise recursiva não é mais aplicável. Em geral, é necessário obter equações do equilíbrio de forças e momentos para cada elo, e resolver as equações simultaneamente. No entanto, se apenas as forças ou momentos são de interesse, o princípio do trabalho virtual pode ser aplicado [75].

Existem dois tipos de juntas em manipuladores paralelos: junta ativa e junta passiva. Assumindo que as juntas não têm atrito e os efeitos gravitacionais são desprezíveis, então as forças de reação nas juntas passivas não contribuem com trabalho virtual.

Seja $\mathbf{F} = [\mathbf{f}, \mathbf{n}]^T$ um vetor que representa a força e momento de saída do efetuador terminal, e o vetor $\boldsymbol{\tau} = [\tau_1, \tau_1, \dots, \tau_n]^T$ representa os torque produzidos pelos atuadores nas juntas (ou forças). O vetor $\delta\mathbf{q} = [\delta q_1, \delta q_2, \dots, \delta q_n]^T$ representa o deslocamento virtual associado com os atuadores das juntas, e o vetor $\delta\mathbf{x} = [\delta q_1, \delta q_1, \dots, \delta q_1]^T$ representa o deslocamento virtual associado ao efetuador terminal.

Então, o trabalho virtual exercido por todas as forças ativas pode ser escrito da seguinte forma:

$$\boldsymbol{\tau}^T \delta\mathbf{q} - \mathbf{F}^T \delta\mathbf{x} = 0 \quad (5-32)$$

Os deslocamentos virtuais, $\delta\mathbf{q}$ e $\delta\mathbf{x}$, são relacionados pela matriz Jacobiana

$$\delta\mathbf{q} = \mathbf{J}\delta\mathbf{x} \quad (5-33)$$

onde $\mathbf{J} = \mathbf{J}_q^{-1}\mathbf{J}_x$, foi definido na equação (5-18). Assim, substituindo a equação (5-33) em (5-32), obtém-se

$$(\boldsymbol{\tau}^T \mathbf{J} - \mathbf{F}^T) \delta\mathbf{x} = 0 \quad (5-34)$$

Uma vez que a equação (5-34) é válida para qualquer deslocamento virtual, conclui-se que

$$\boldsymbol{\tau}^T \mathbf{J} - \mathbf{F}^T = 0 \quad (5-35)$$

Finalmente, obtemos

$$\mathbf{F} = \mathbf{J}^T \boldsymbol{\tau} \quad (5-36)$$

$$\mathbf{J} = \mathbf{J}_q^{-1} \mathbf{J}_x \quad (5-37)$$

5.4

Descrição do Hardware

O Falcon é um robô paralelo de 3 graus de liberdade. Conectando-se dois Falcons através de uma barra com duas juntas esféricas (vide figuras 5.7 e 5.9), pode-se obter um dispositivo que movimenta-se em 6 graus de liberdade, segundo [76]. Isto se demonstra com a seguinte equação de mobilidade:

$$M = 6(n - g) + \sum_{i=1}^g f_i \quad (5-38)$$

onde: conectividade de cada Falcon $f_1 = f_2 = 3$ (movimentos em x,y,z sem rotação), conectividade de cada junta esférica $f_3 = f_4 = 3$, número de partes moveis: $n = 3$, número de juntas: $g = 4$, logo

$$M = 6(3 - 4) + 3 + 3 + 3 + 3$$

$$M = -6 + 12 = 6$$

Com o resultado da equação da mobilidade para nosso sistema, o maior número de graus de liberdade que se pode obter acoplando os dois Falcons desta forma seria 6. Mas o sistema somente poderia operar em 5 graus de liberdade, devido à falta de um sensor para medir o ângulo giro da barra em torno de seu próprio eixo. Então, o dispositivo háptico fica limitado a controlar 5 graus de liberdade.

A figura 5.7 mostra um esquema do dispositivo háptico antes de sua construção.

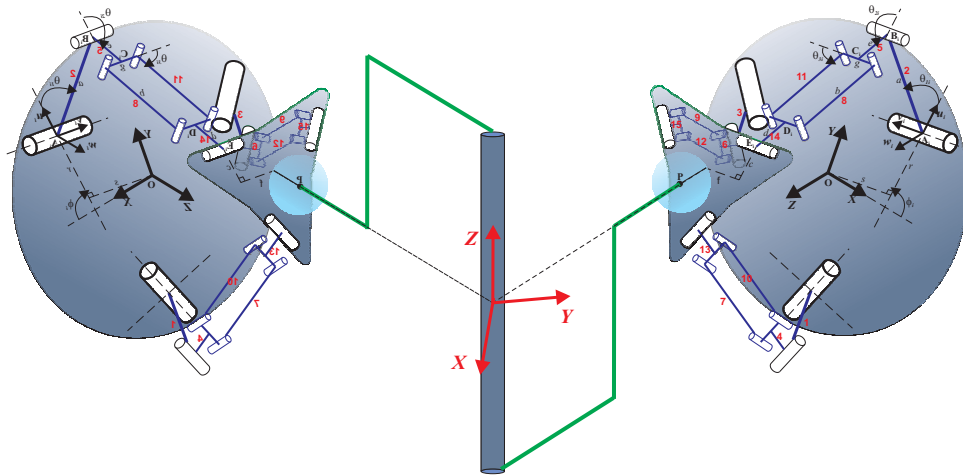


Figura 5.7: Esquema do Dispositivo Háptico de 5 graus de liberdade

Na figura 5.8, mostram-se os Falcons orientados a 45° fixos em um suporte retangular.



Figura 5.8: Falcons fixos, ainda desacoplados

A figura 5.9 apresenta as rótulas mecânicas utilizadas na construção.



Figura 5.9: Componentes mecânicos

Finalmente, a figura 5.10 apresenta o dispositivo háptico construído.



Figura 5.10: Dispositivo Háptico de 5 graus de liberdade

5.5 Compensação de Gravidade

O Falcon não tem implementada uma compensação de gravidade em sua configuração normal. Isso quer dizer que o efetuator terminal do Falcon, não fica em uma posição estática dentro da sua área de trabalho, tendendo a cair até seu limite inferior de curso. Além disso, na construção do dispositivo háptico, se acoplaram novos componentes ao sistema (elos, parafusos e rótulas), que também adicionam uma certa massa.

O software do Falcon não permite enviar torques para o sistema, mas utilizando a biblioteca CHAI3D podemos enviar forças desejadas para atuador. Então podemos achar primeiramente os torques necessários que permitam compensar a força de gravidade, e depois utilizando a equação (5-36) encontramos as forças necessárias para enviar aos atuadores dos Falcons.

A figura 5.11 mostra o diagrama de corpo livre de um elo do Falcon, para achar o torque equivalente. Observe que no primeiro componente do elo acoplado ao atuador, a posição do centro de massa encontra-se a uma distância chamada l_{cg} até o eixo do atuador ([76] e [77])

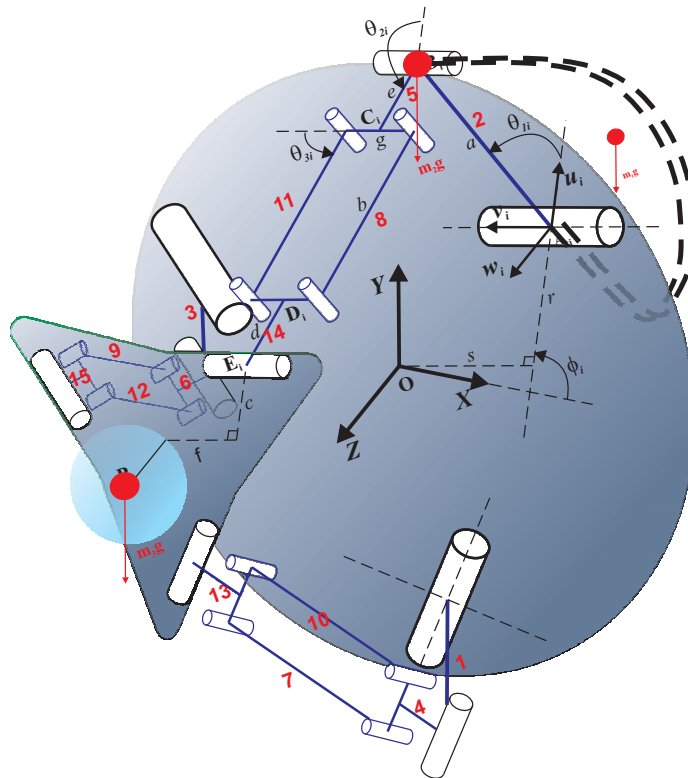


Figura 5.11: Diagrama de Corpo Livre de um elo do Falcon

O torque equivalente para compensar a gravidade é

$$\tau_{equi} = \tau_{laco} + \tau_{efe} \quad (5-39)$$

A figura 5.12 mostra a decomposição da força de gravidade nas coordenadas $u_i v_i w_i$, a qual um torque ao atuador

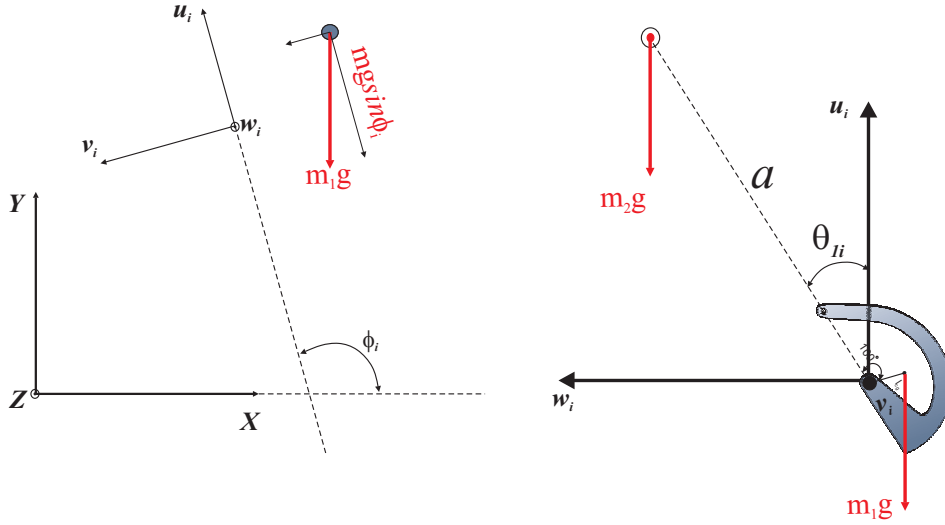


Figura 5.12: Forças de Gravidade em um elo do Falcon

Então, o torque do i -ésimo elo do Falcon $\tau_{laco i}$ é:

$$\tau_{laco i} = m_1 g l_{cg} \sin(100^\circ - \theta_{1i}) \sin \phi_i + m_2 g a \sin(\theta_{1i}) \sin \phi_i \quad (5-40)$$

onde m_1 é a massa do elo curvo, e m_2 é a massa conjunta do elo paralelo e da articulação.

Agora, substituindo em uma expressão geral para todos os laços do Falcon,

$$\tau_{equi} = m_1 g l_{cg} \begin{bmatrix} \sin(100^\circ - \theta_{11}) \sin \phi_1 \\ \sin(100^\circ - \theta_{21}) \sin \phi_2 \\ \sin(100^\circ - \theta_{31}) \sin \phi_3 \end{bmatrix} - m_2 g a \begin{bmatrix} \sin(\theta_{11}) \sin \phi_1 \\ \sin(\theta_{21}) \sin \phi_2 \\ \sin(\theta_{31}) \sin \phi_3 \end{bmatrix} + m_3 g l_{cg} \mathbf{J}^{-T} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad (5-41)$$

onde m_3 é a massa conjunta do efetuator terminal mais a plataforma móvel e as articulações dos três laços.

Finalmente, substituímos a equação (5-41) em (5-36), para obter a força de compensação de gravidade do sistema

$$\mathbf{F} = \mathbf{J}^T \boldsymbol{\tau}_{equi} \quad (5-42)$$

Os valores da massa dos elementos do Falcon foram obtidos de [77], e o valor da massa dos elementos acoplados ao novo dispositivo háptico são apresentados na Tabela 5.1

Tabela 5.1: Valores da massa e comprimento dos Parâmetros

Nome dos Componentes	Massa(Kg)	Comprimento(m)
Elo Curvo	$m_a = 0,052$	$a = 0.060$
Elo paralelo	$m_b = 0,008$	$b = 0.1025$
Plataforma Móvel	$m_c = 0,033$	$c = 0.157$
Articulação	$m_d = 0,010$	$d = 0.0115$
Distância na Plataforma		$r = 0.0366$
Deslocamento na Plataforma		$s = 0.0272$
Deslocamento na Plataforma		$f = 0.0262$
Deslocamento - z		$p_{z,0} = 0.134$
Efetuator terminal	$m_e = 0,052$	$L = 0.15$

5.6 Controle de 5 Graus de Liberdade

Para sincronizar o Dispositivo Háptico com o robô manipulador dentro do ambiente virtual, temos que utilizar algumas técnicas de teleoperação. Existem distintos esquemas de controle utilizados na teleoperação. Peñin [78] realiza um análise abrangente dos sistemas clássicos de teleoperação, usando para isso a teoria de controle clássico.

O esquema de controle bilateral Posição - Posição, mostrado na figura 5.13, é baseado na troca de leituras de posição tanto do mestre como do escravo. No lado do mestre, realiza-se a leitura de sua posição, logo enviada ao escravo, onde existe uma malha fechada de controle de posição cuja entrada de referência é a posição do mestre. No lado do escravo, também se faz a leitura da posição, que é enviada ao mestre, onde também existe uma malha fechada de controle de posição cuja entrada de referência é a posição do escravo.

Este esquema é muito fácil de implementar, dado que as reflexões de forças são baseadas nas diferenças entre as posições do mestre e escravo. Se não houver amplificação, os ganhos (G_m , G_s) serão equivalentes, obtendo-se um controle simétrico. A desvantagem principal deste sistema de controle é

não ser possível ter uma percepção (sentir) a rigidez de objetos muito rígidos ao interagir, com o robô escravo.

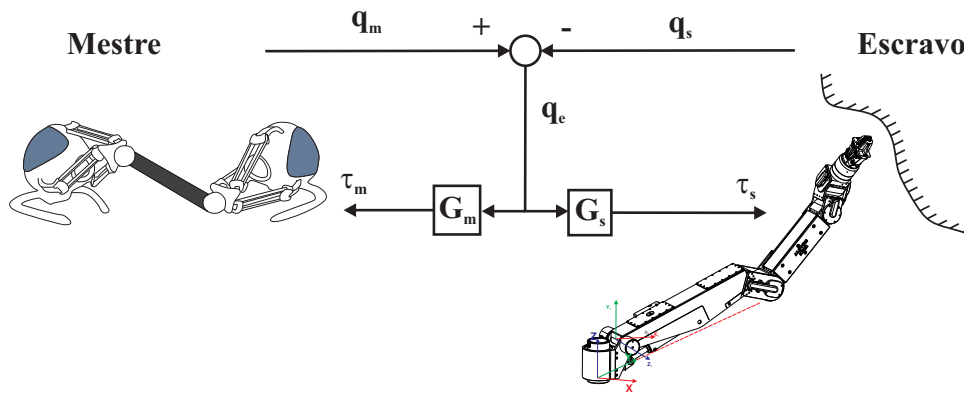


Figura 5.13: Esquema de controle bilateral Posição - Posição

Outro esquema é o controle bilateral com sensor de força, é mostrada na figura 5.14. Este esquema é parecido ao primeiro, mas a diferença é a existência de um sensor para obter a força que aplica o escravo dentro do ambiente remoto ou vice-versa. Essa informação é enviada ao mestre, a qual realimenta a força em malha aberta sobre o operador.

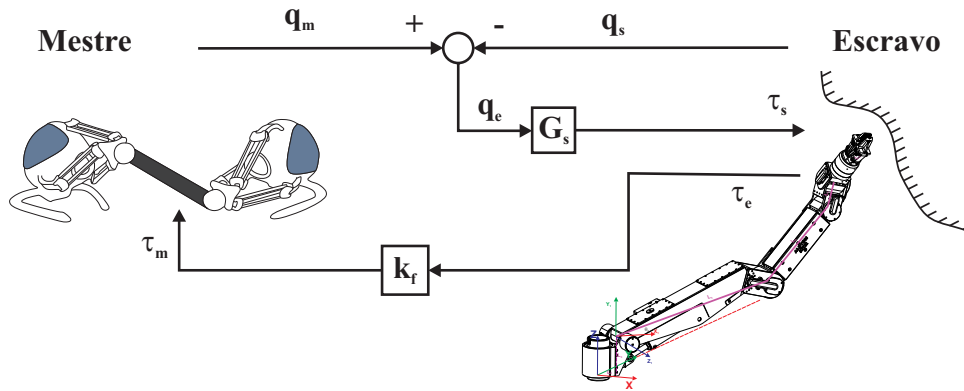


Figura 5.14: Esquema é o controle bilateral com sensor de força

Deve-se considerar também incluir uma modelagem do operador (humano) ao sistema de controle. Uma forma de fazer isto é considerando a interação com o ser humano um sistema massa-mola-amortecedor e logo aplicando os conhecimentos básicos de controle para obter um modelagem total do sistema.

Agora, vamos desenvolver um algoritmo de controle para o dispositivo háptico, que permita obter uma sensação nos 5 graus de liberdade.

A biblioteca CHAI3D fornece somente a posição do efetuador terminal do Falcon, assim $[p_{ox}, p_{oy}, p_{oz}]^T$.

A conversão de coordenadas segue da seguinte forma:

$$\begin{aligned} p_x &= p_{oy} \\ p_y &= p_{oz} \\ p_z &= p_{ox} + p_{z,0} \end{aligned}$$

onde $[p_x p_y p_z]^T$ é o vetor posição do efetuador terminal em relação ao sistema coordenado assumido na modelagem do Falcon.

Deve-se desenvolver uma coordenação das forças individuais em cada Falcon para interagir na barra que os une, dando assim a sensação de 5 graus de liberdade. Para isto, definimos um sistema de coordenadas no centro da barra (centro de gravidade), tal que o vetor posição zero do efetuador dos Falcons, seja também o vetor posição zero no centro da barra (configuração inicial).

A figura 5.15 mostra a configuração inicial e a posição dos sistemas coordenados dos dois Falcons e o sistema coordenada da barra.

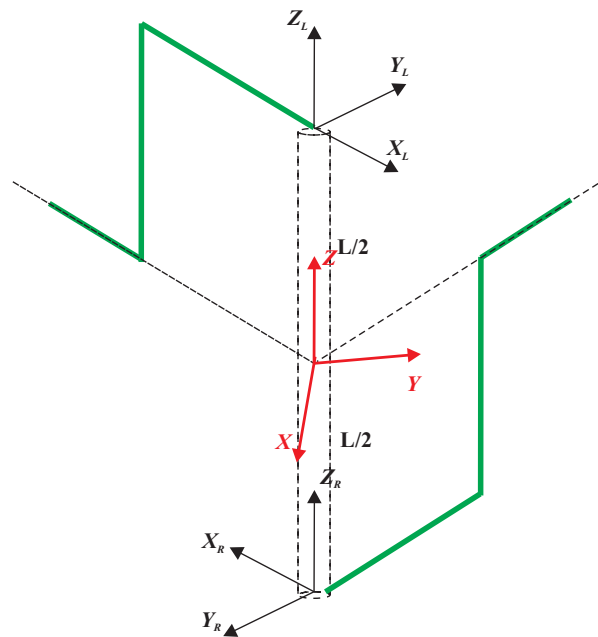


Figura 5.15: Sistema de Coordenadas

Então, para obter as posições do efetuador terminal de cada Falcon em relação ao sistema coordenado da barra, utilizamos a matriz de transformação

$$\begin{bmatrix} {}^b p_{xL} \\ {}^b p_{yL} \\ {}^b p_{zL} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\pi/4) & -\sin(\pi/4) & 0 & 0 \\ \sin(\pi/4) & \cos(\pi/4) & 0 & 0 \\ 0 & 0 & 0 & L/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_{xL} \\ p_{yL} \\ p_{zL} \\ 1 \end{bmatrix} \quad (5-43)$$

$$\begin{bmatrix} {}^b p_{xR} \\ {}^b p_{yR} \\ {}^b p_{zR} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\pi/4) & -\sin(\pi/4) & 0 & 0 \\ \sin(\pi/4) & \cos(\pi/4) & 0 & 0 \\ 0 & 0 & 0 & -L/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_{xR} \\ p_{yR} \\ p_{zR} \\ 1 \end{bmatrix} \quad (5-44)$$

O objetivo é mostrar uma força e torque efetivos em algum ponto da barra P_x , para isto vamos definir os vetores posição em relação ao sistema de coordenada da barra, como mostra na figura 5.16.

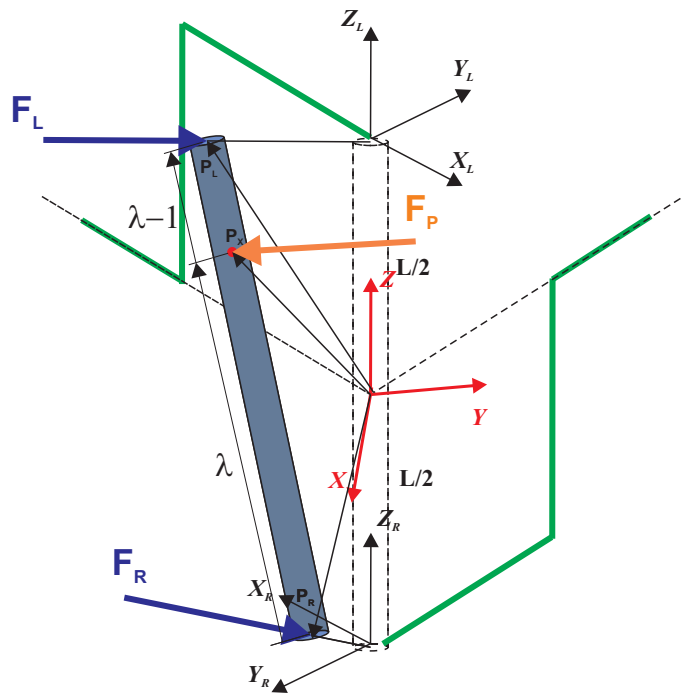


Figura 5.16: Efetuador Terminal do dispositivo Háptico de 5 graus de liberdade

Do gráfico, podemos achar a seguinte relação utilizando a propriedade de vetores

$$\mathbf{P}_x = \lambda \mathbf{P}_R + (1 - \lambda) \mathbf{P}_L \quad 0 \leq \lambda \leq 1 \quad (5-45)$$

Para calcular a força efetiva na barra, aplicamos a primeira lei de

equilíbrio

$$\begin{aligned} \mathbf{F}_L + \mathbf{F}_R + \mathbf{F}_P &= 0 \\ \sum \text{momentos}_{P_x} &= 0 \end{aligned} \tag{5-46}$$

então achando o momento em relação ao ponto P_x

$$(\mathbf{P}_L - \mathbf{P}_X) \times \mathbf{F}_L + (\mathbf{P}_R - \mathbf{P}_X) \times \mathbf{F}_R = 0 \tag{5-47}$$

obtém-se a seguinte expressão:

$$\lambda \mathbf{F}_L = (1 - \lambda) \mathbf{F}_R \tag{5-48}$$

Substituindo a equação (5-48) em (5-46) encontramos

$$\begin{aligned} \mathbf{F}_R &= \lambda \mathbf{F}_P \\ \mathbf{F}_L &= (1 - \lambda) \mathbf{F}_P \end{aligned} \tag{5-49}$$

o torque efetivo para a barra mostrado na figura 5.17, é tal que

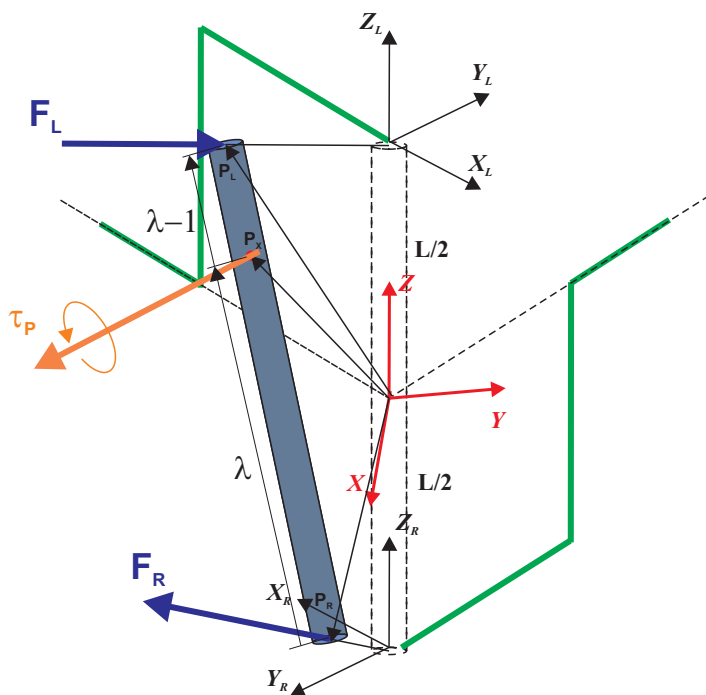


Figura 5.17: Efetuador Terminal do dispositivo Háptico de 5 graus de liberdade

$$\boldsymbol{\tau}_P + \boldsymbol{\tau}_{efe} = 0 \quad (5-50)$$

Definindo $\Delta = \mathbf{P}_L - \mathbf{P}_R$, e aplicando a lei de equilíbrio ao sistema, temos

$$\boldsymbol{\tau}_{efe} = \Delta \times \mathbf{F}_{efe} \quad (5-51)$$

$$\begin{aligned} \boldsymbol{\tau}_{efe} \times \Delta &= \Delta \times \mathbf{F}_{efe} \times \Delta \\ \mathbf{F}_{efe} &= \frac{\boldsymbol{\tau}_{efe} \times (\mathbf{P}_L - \mathbf{P}_R)}{|\Delta|^2} \\ \mathbf{F}_{efe} &= \frac{\boldsymbol{\tau}_{efe} \times \Delta}{L^2} \end{aligned} \quad (5-52)$$

Então, as forças efetivas para exercer o torque ao usuário são:

$$\mathbf{F}_R = \mathbf{F}_{efe} \quad (5-53)$$

$$\mathbf{F}_L = -\mathbf{F}_{efe} \quad (5-54)$$

Desse modo é possível prover ao usuário as sensações de força/torque, em 5 graus de liberdade, referentes à tarefa do sistema escravo. Mas devemos ter em conta que há redundância de força/torque, porque temos em total 6 atuadores por parte dos dois Falcon's. O problema de não poder observar ou sentir o sexto grau de liberdade, é porque uma componente de cada força que fornecem os falcon's encontra-se no eixo da barra e passam pelo centro de gravidade produzindo assim, um torque igual a zero. É por isso que o sexto grau de liberdade não é observável nem controlável em nosso sistema.

No próximo capítulo este dispositivo será acoplado a um ambiente virtual, desenvolvido especialmente para essa dissertação, controlando (virtualmente) um manipulador robótico Schilling Titan IV.

6 Desenvolvimento do Simulador Virtual

Neste capítulo, descreve-se o desenvolvimento do Simulador Virtual para demonstrar a importância das interfaces Hápticas. Além disso, falaremos alguns conceitos utilizados em informática úteis para explicar o funcionamento da biblioteca CHAI3D, e sua programação para este trabalho. O compilador utilizado para desenvolver o Simulador Virtual é o Microsoft Visual Studio 2008. Este ambiente apresenta diversas funcionalidades e facilidades para programar em *C/C++* e também poder embutir em um mesmo executável arquivos e bibliotecas *dll*'s necessários para compilar o CHAI3D.

6.1 Definições Prévias

6.1.1 Thread

Thread, ou linha de execução em português, é uma forma de um processo dividir a si mesmo em duas ou mais tarefas que podem ser executadas simultaneamente [79]. O suporte à thread é fornecido pelo próprio sistema operacional (SO), no caso da Kernel-Level Thread (KLT), ou implementada através de uma biblioteca de uma determinada linguagem, no caso de uma User-Level Thread (ULT). Uma linha de execução permite que o usuário do programa utilize uma funcionalidade do ambiente, enquanto outras linhas de execução realizam outros cálculos e operações.

Os sistemas que suportam apenas uma única linha de execução é chamado de monothread, e aqueles sistemas que suportam múltiplas linhas de execução são chamados de multithread.

6.1.2 Framework

Em desenvolvimento de software, um framework ou arcabouço é uma abstração que une códigos comuns entre vários projetos de software, provendo uma funcionalidade genérica. Um framework pode atingir uma funcionalidade específica, por configuração, durante a programação de uma aplicação. Ao

contrário das bibliotecas, é o framework quem dita o fluxo de controle da aplicação, chamado de Inversão de Controle [80].

6.1.3

API

API, de Application Programming Interface (ou Interface de Programação de Aplicativos), é um conjunto de rotinas e padrões estabelecidos por um software para a utilização das suas funcionalidades por programas aplicativos que não querem envolver-se em detalhes da implementação do software, mas apenas usar seus serviços. De modo geral, a API é composta por uma série de funções acessíveis somente por programação, e que permitem utilizar características do software menos evidentes ao utilizador tradicional [81].

6.1.4

Biblioteca

Na ciência da computação, biblioteca é uma coleção de sub-programas utilizados no desenvolvimento de software. Bibliotecas contêm código e dados auxiliares, que provêem serviços a programas independentes, o que permite o compartilhamento e a alteração de código e dados de forma modular. Alguns executáveis são tanto programas independentes quanto bibliotecas, mas a maioria das bibliotecas não são executáveis. Executáveis e bibliotecas fazem referências mútuas conhecidas como ligações, tarefa tipicamente realizada por um ligador [82].

6.1.5

Motores Gráficos

Motor gráfico (ou motor de renderização, ou motor 3D, ou Engine) é uma ferramenta usada para o desenvolvimento de simulações em tempo real, como os jogos, tanto para videogames ou computadores [79]. Eles possuem um conjunto de bibliotecas específicas para simplificar o desenvolvimento do mesmo. O que caracteriza um motor gráfico é um conjunto de subsistemas como o de renderização, que é usado tanto para gráficos 2D como para 3D, o motor de física ou detecção de colisão, e motores de suporte a sons, animação, inteligência artificial, rede e grafo de cena (scene graph), que é a representação hierárquica de todos os objetos que compõem um mundo virtual.

6.1.6 OpenGL

OpenGL é definido como "um programa de interface para hardware gráfico". Na verdade, OpenGL é uma biblioteca de rotinas gráficas e de modelagem, bi (2D) e tridimensional (3D), extremamente portátil e rápida. Usando OpenGL, é possível criar gráficos 3D com uma qualidade visual próxima de um efeitos visuais (*ray tracer*). Entretanto, a maior vantagem na sua utilização é a rapidez, uma vez que usa algoritmos cuidadosamente desenvolvidos e otimizados pela Silicon Graphics, Inc., líder mundial em Computação Gráfica e Animação [83].

OpenGL não é uma linguagem de programação, é uma poderosa e sofisticada API (Application Programming Interface) para criação de aplicações gráficas 2D e 3D. Seu funcionamento é semelhante ao de uma biblioteca C, uma vez que fornece uma série de funcionalidades. Normalmente, se diz que um programa é baseado em OpenGL, ou é uma aplicação OpenGL, o que significa que ele é escrito em alguma linguagem de programação que faz chamadas a uma ou mais bibliotecas OpenGL.

6.1.7 ODE

ODE de Open Dynamic Engine, é uma biblioteca de alto desempenho de código aberto, para simulação dinâmica de corpos rígidos. Seus dois principais componentes são o motor de dinâmicas de corpo rígido e o motor de detecção de colisão. Está repleto de recursos para plataformas independentes, e é fácil de usar em C/C++ API. Essa biblioteca implementa tipos comuns e de detecção de colisão integrada com o atrito. ODE é útil para os veículos que simulam objetos em ambientes de realidade virtual e criaturas virtuais. Atualmente, é usado em muitos jogos de computador, ferramentas de criação 3D e ferramentas de simulação.

A ODE é uma biblioteca gratuita, utilizada para a simulação articulada de corpos rígidos dinâmicos estruturados. Seu dois principais componentes dessa biblioteca são: o motor de dinâmicas de corpo rígido e o motor de detecção de colisão. Inicialmente implementada para a linguagem C, hoje é disponibilizada para usuários C++ e Delphi. Ela apresenta ótimos recursos para a simulação em tempo real de objetos em ambientes de realidade virtual. Segundo Smith [84], criador da biblioteca, por ser rápida, flexível e robusta, ela possibilita o desenvolvimento de aplicações de extrema complexidade e com alto nível de precisão sem perda de desempenho, tornando possível desenvolver softwares com simulação física muito próxima da realidade.

A biblioteca ODE permite manipular corpos (objetos) considerando forças, torques e energias, além de outros atributos da física, dando aos corpos da cena comportamento e reações físicas realísticas. Hecker [85] destaca que a simulação de fenômenos físicos faz a “solidez” de um jogo.

6.2 CHAI3D

O CHAI3D (Computer Haptics & Active Interface 3D) é um framework open source (código aberto) para a visualização e simulação de interação háptica em tempo real [86]. Verifica-se um crescimento notável, significativo assinalável da sua utilização nos 5 últimos anos desde sua criação, sendo utilizado em vários projetos e em diferentes domínios, tais como: jogos, simuladores, aplicações para o ensino ou aplicações médicas. Esta plataforma está escrita em C++ e foi concebida com o intuito de tornar fácil e intuitiva a criação de novas aplicações. Este framework fornece as estruturas de dados necessárias para que seja possível a implementação de corpos estáticos, dinâmicos e articulados. A componente gráfica do CHAI3D é conseguida através de um motor gráfico leve, baseado em OpenGL. Esta está também apoiada num modelo bem estruturado de classes, que tem como principal objetivo permitir eventuais extensões de forma simples. Está também disponível a importação de modelos 3D desenvolvidos em aplicações como o Autodesk 3D Studio Max e o Alias Wavefront. O CHAI3D foi concebido de forma a suportar módulos de extensão, sendo que atualmente alguns dos módulos mais relevantes são os motores físicos ODE e GEL que permitem simular corpos rígidos e deformáveis em tempo real. Refira-se ainda que este se encontra preparado para a utilização de diferentes dispositivos hápticos, e que é possível a sua utilização em ambiente Windows, Linux e Mac OSX (Multiplataforma).

Apesar das vantagens supra apresentadas, este framework utiliza para a simulação dos comportamentos físicos de objetos o motor de física ODE, sendo que este está atualmente a ser preterido em relação a outros motores físicos mais recentes. O mesmo não apresenta novas funcionalidades há algum tempo e, em consequência desse decréscimo de melhorias e do aparecimento de mais recentes e melhores motores físicos, tem se verificado a redução de utilizadores. Dada a especial relevância do comportamento físico dos objetos neste trabalho, o motor físico utilizado mostra-se como um dos fatores determinantes de sucesso, sendo que o ODE não dá as garantias pretendidas para o sucesso do trabalho que se pretende desenvolver.

O CHAI3D foi desenvolvido por grupo de pesquisadores da Universidade de Stanford USA, Universidade Católica de Brasília, Universidade de Siena

Itália e o Instituto EPFL no Lausanne, Suíça.

Podem-se destacar algumas características do CHAI3D:

- Renderização gráfica e encapsulamento das funcionalidades da API para o processamento gráfico com OpenGL.
- Renderização gráfica e háptica com detecção de colisões para malhas triangulares e controle de gravidade.
- Importação de modelos com extensão .obj e .eds.
- Interação com superfícies lisas virtuais, através do Proxy háptico.
- Comunicação com muitas placas eletrônicas digitais/análogas de E/S.
- Integração com ActiveX para aplicações portais-embarcados.
- Disponibilizar mecanismos de navegação como translação e rotação de objetos 3D.
- Disponibilizar mecanismos de mapeamento de texturas, câmera, luzes, transparência e uma interface de contato virtual.
- Integração com ODE (Open Dynamics Engine) para interação háptica com corpos rígidos.

O CHAI3D suporta os seguintes dispositivos hápticos:

- Delta.x e Omega.x da Force Dimension (<http://www.forcedimension.com/products>).
- PHAToM da SensAble Inc. (<http://www.sensable.com/products-haptic-device.htm>).
- Novint Falcon[®] Technologies Inc. (<http://www.novint.com/index.php/novintfalcon>).
- Freedom 6S de MPB Technologies Inc Technologies Inc. (http://www.mpb-technologies.ca/mpbt/mpbt_web_2009/_en/6dof/index.html).

O CHAI3D suporta os seguintes sistemas de compilação:

- Microsoft Visual Studio 6.
- Microsoft Visual Studio.Net.
- Borland C++ Builder.
- Cygwin - gcc.
- Linux - gcc.

6.2.1 Efeitos Hápticos

A cena gráfica do CHAI3D fornece um conjunto de efeitos hápticos que podem ser atribuídos para superfícies de objeto implícitos. Esses efeitos são computados utilizando o método *computeLocalInteraction* de local interação de cada objeto. A malha, ou qualquer outro objeto complexo que não utilize esse método, não é capaz de aplicar os efeitos hápticos, porque não há forma de calcular um ponto de interação projetada a partir de um algoritmo de objeto genérico. Só o algoritmo para o ponto *Proxy* é usado para esse objetos para calcular forças.

Os efeitos hápticos em base à classe *cGenericEffect* na API são o seguinte:

- Efeito Magnético, *cEffectMagnet* proporciona um campo magnético perto do objeto.
- Efeito Viscosidade, *cEffectViscosity* proporciona um efeito de uma ferramenta movendo-se através de um fluido.
- Efeito de Vibração, *cEffectVibration* proporciona um efeito de vibração com uma especificada frequência e amplitude.
- Efeito de superfície, *cEffectSurface* proporciona um efeito de superfície básica para uma ferramenta que pressiona um objeto.
- Efeito Stick-slip, proporciona um efeito de deslizamento em um objeto sobre outro com aderência produzido por atrito.

6.2.2 Módulo ODE

O CHAI3D não tem implementada sua própria simulação de corpos rígidos. No entanto, existe um módulo que conecta a cena gráfica do CHAI3D com a biblioteca ODE. A comunicação entre CHAI3D e ODE é controlada pelas classes *cODE*, *cODEWorld* e *cODEGenericBody*. O API contém bibliotecas ODE pré-compiladas tanto para ligações estáticas dinâmicas, com dupla precisão.

Cada objeto na simulação com ODE tem que ser adicionado a um específico mundo ODE. Tal objeto é definido como um corpo genérico ODE com propriedades de simulação física e um corpo CHAI3D modela a imagem na cena gráfica.

O módulo ODE habilita a criação de caixas dinâmicas, esferas, cápsulas e malhas. Planos estáticos também estão disponíveis e a gravidade global pode ser definida como um vetor tridimensional que descreve uma força (Figura 6.1).



Figura 6.1: Exemplo do módulo ODE em CHAI3D

6.2.3 Módulo GEL

A tecnologia háptica utiliza uma implementação de uma simulação de corpo deformável mais do que qualquer outra tecnologia. O CHAI3D oferece um módulo para criar tais objetos deformáveis na cena gráfica, o qual usa o motor gráfico dinâmico GEL desenvolvido pela Stanford University. Assim como no módulo ODE, o módulo GEL é implementado como um mundo separado (*cGELWorld*) de objetos deformáveis. A idéia principal por trás da deformação é uma modelo esqueleto feito de nós (*cGELSkeletonNode*) e elos (*cGELSkeletonLink*) entre eles.

Os Nós são representados como esferas com um determinado raio e massa ligados com molas definidos pelas propriedades de deflexão por alongamento e torção (Figura 6.2). Cada nó tem propriedades físicas (amortecimento linear, amortecimento angular, definição de campo de gravidade) e fornece métodos para controle de força e torque. O módulo GEL proporciona uma forma simples de adicionar objetos deformáveis para a cena gráfica.

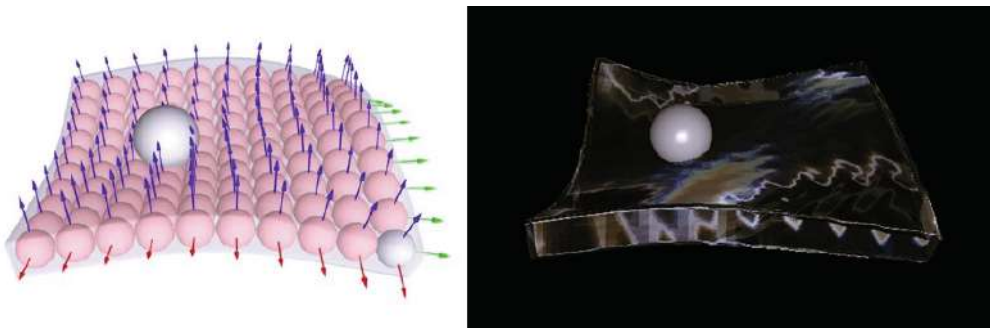


Figura 6.2: Exemplo do módulo GEL em CHAI3D

6.2.4 Arquitetura da Interface Háptica

Os algoritmos de renderização háptica calculam as forças de interação entre a representação da interface tátil dentro do ambiente virtual e os objetos virtuais que povoam o meio. Além disso, esses algoritmos garantem que o dispositivo háptico execute as forças corretamente sobre o operador humano.

Chamamos *avatar* à representação virtual na interface háptica através da qual o usuário interage fisicamente com o ambiente virtual. É evidente que a escolha do *avatar* depende do que está sendo simulado e também da capacidade do dispositivo háptico. O operador controla a posição do *avatar* dentro do ambiente virtual. O contato entre o *avatar* da interface e do ambiente virtual desencadeia forças de ação e reação que dependem também da geometria do *avatar* do tipo contato, que regula essas forças.

Dentro de um determinado aplicativo, o usuário pode escolher entre os diferentes *avatars*. Por exemplo, uma ferramenta cirúrgica pode ser tratada como um objeto volumétrico de troca de forças e posições com o usuário em espaço 6D, ou como um ponto para representação da ferramenta, para a troca de forças e posições em um espaço 3D. Vários componentes compõem tipicamente um algoritmo de renderização háptica, ilustrado na figura 6.3

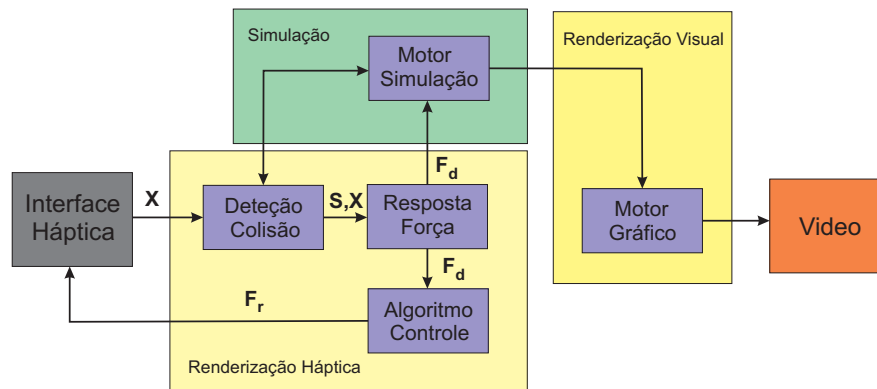


Figura 6.3: Arquitetura da Interface Háptica no CHAI3D

O algoritmo de *Collision-detection* detecta as colisões entre os objetos e *avatars* no ambiente virtual e fornece informações sobre onde, quando e, idealmente para que extensão de colisão (penetração, recortes, área de contato, e assim por diante) tem acontecido.

O algoritmo de *Force-response* calcula a força de interação entre os *avatars* e objetos virtuais, quando uma colisão é detectada. Esta força se aproxima, tanto quanto possível, às forças de contato que normalmente surgem durante o contato entre os objetos reais. Esse algoritmo opera normalmente

nas posições do *avatar*, nas posições dos objetos dentro do ambiente virtual, e no estado de colisão entre os *avatares* e objetos virtuais. Seus valores de retorno são normalmente vetores de força e torque aplicados ao corpo do dispositivo háptico.

Limitações de hardware impedem que os dispositivos hápticos apliquem a força exata calculada pelos algoritmos de força para o usuário. Algoritmos de controle comandam o dispositivo de tal forma que seja minimizado o erro entre as forças ideais e aplicáveis.

A natureza de tempo discreto dos algoritmos de renderização háptica muitas vezes torna isso difícil. Vetores desejados de força e torque calculados pelos algoritmos de força alimentam os algoritmos de controle que vai ser comandado para o dispositivo háptico. Um típico ciclo háptico consiste da seguinte sequência de eventos:

- Algoritmos de controle de baixo nível, que amostram a posição nas articulações pelos sensores do dispositivo háptico.
- Este algoritmo de controle combina a informação recolhida de cada sensor para obter a posição do dispositivo no espaço cartesiano, que seria a posição do *avatar* no ambiente virtual.
- O algoritmo de detecção de colisão usa a informação de posição para encontrar colisões entre objetos e *avatares* e relata o grau resultante da penetração ou recuo.
- O algoritmo de força computa a interação de forças entre *avatares* e objetos virtuais envolvidos na colisão.
- A resposta do algoritmo de força envia as forças de interação para os algoritmos de controle, aplicadas sobre o operador através do dispositivo háptico, enquanto se mantém um comportamento globalmente estável.

O mecanismo de simulação em seguida usa as mesmas técnicas de interação para calcular o seu efeito sobre objetos no ambiente virtual. Embora não existam regras sobre a frequência dos algoritmos hápticos, a taxa comum é de 1KHz. Essa taxa é aceitável, mas para maiores taxas pode-se proporcionar um maior realismo nas sensações de textura, vibração etc.

6.3 Desenvolvimento do Ambiente Virtual

Inicialmente se projetou um ambiente virtual simples, para interagir com o dispositivo Falcon. No ambiente virtual se encontra uma caixa fixa de cor azul, mostrada na figura 6.4, com um orifício na parte central. Além disso, se representa com uma esfera rígida de cor cinza (*Avatar*) o efetuator terminal do Falcon. Esse ambiente será usado posteriormente para avaliar a dificuldade de atravessar o avatar pelo orifício com ou sem realimentação de força.

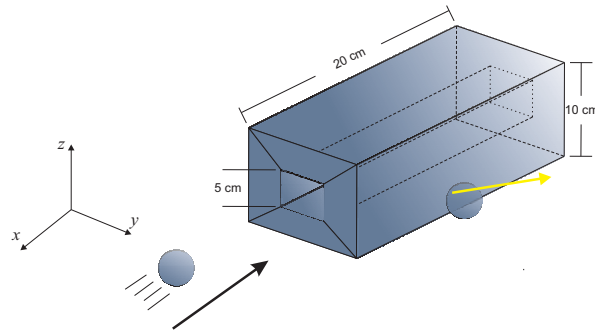


Figura 6.4: Desenho da caixa fixa com orifício na parte central

Esta esfera será aquela que interage com a caixa fixa, enviando uma realimentação de força para o usuário. Também adiciona-se ao programa uma visualização, a posição em tempo real da esfera, e as coordenadas do vetor força criado pela colisão dentro do ambiente virtual.

A figura 6.5 apresenta o ambiente virtual simples, implementado no programa.

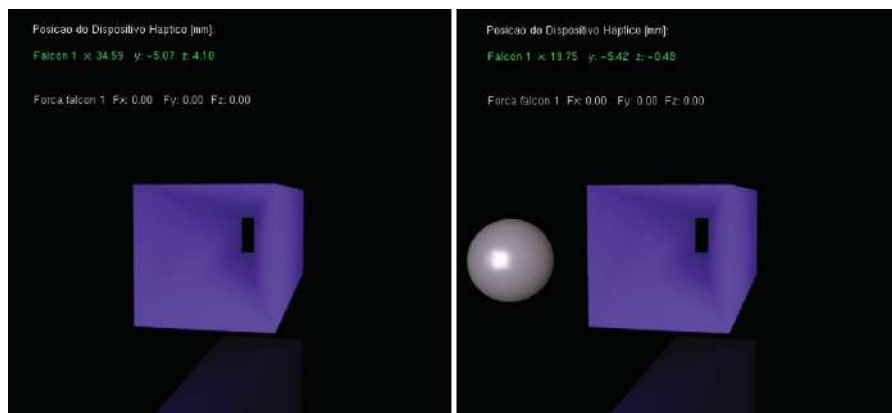


Figura 6.5: Implementação da caixa fixa com orifício na parte central

Para desenhar a caixa, implementou-se uma função *createObjeto()* que utiliza os recursos do OpenGL para desenhar superfícies. Com a classe *CMesh* do

CHAI3D podemos computar os vetores normais para a iluminação e também computar a detecção de colisões no objeto.

Adicionou-se ao ambiente uma representação gráfica no espaço do vetor força de reação, criado pelas colisões, mediante uma linha branca vide figura 6.6.

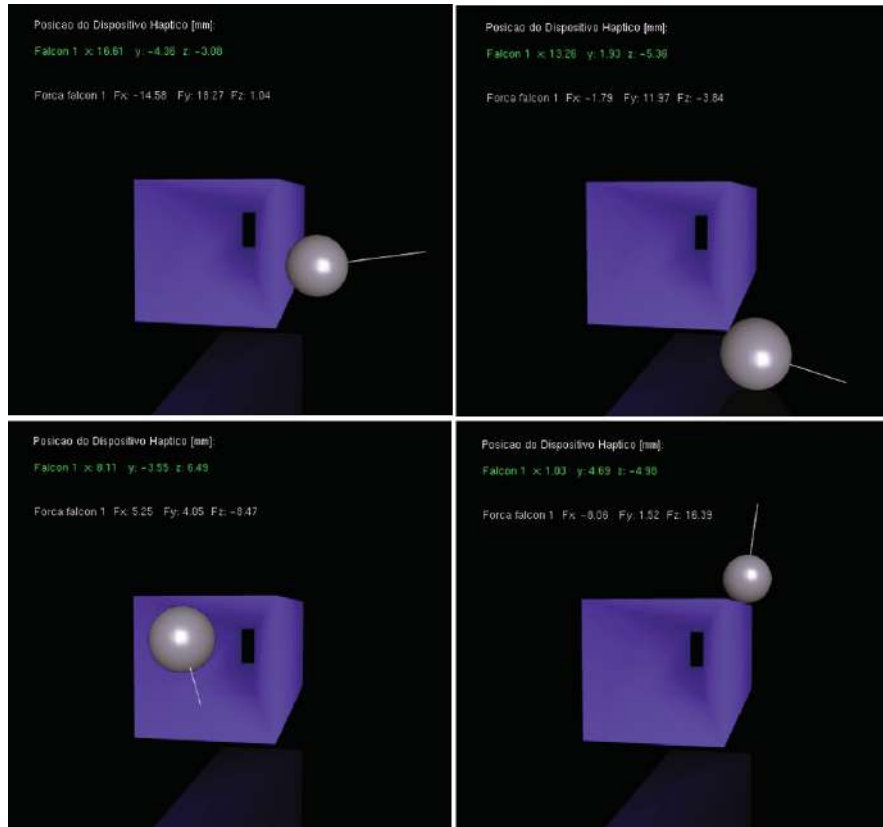


Figura 6.6: Implementação dos vetores de força

6.3.1 Simulador do Robô Manipulador

O algoritmo 1 mostra como foi realizada uma metodologia para desenhar e implementar a modelagem do robô manipulador no ambiente virtual.

Algorithm 1 Desenho e implementação da modelagem do Robô Manipulador

```
1: Desenho do Robô com a função Robot_Schilling  
2: Implementação da Cinemática Direta na função CinemaDireS  
3: Implementação da Cinemática Inversa na função CinemaInverS  
4: Implementação da Dinâmica Direta na função NewtonEulerS  
5: Implementação da Dinâmica Inversa na função WalkerOrinS  
6: while Encontra-se na área de trabalho do  
7:   if dado de entrada é ponto singular then  
8:     Computar  $q \rightarrow q_d + \Delta q$ .  
9:     return false  
10:  else  
11:    computar nova posição  $q \rightarrow q_d$   
12:  end if  
13:  Computar a dinâmica do robô.  
14: end while  
15: return true
```

A figura 6.7 mostra o robô manipulador usado na teleoperação, de 6 graus de liberdade. Cada peça que o compõe foi desenhada em três dimensões utilizando-se as bibliotecas de modelagem em 3D. Após o modelo tridimensional ser construído com as coordenadas e comprimentos de seu projeto, este é importado para o ambiente virtual.

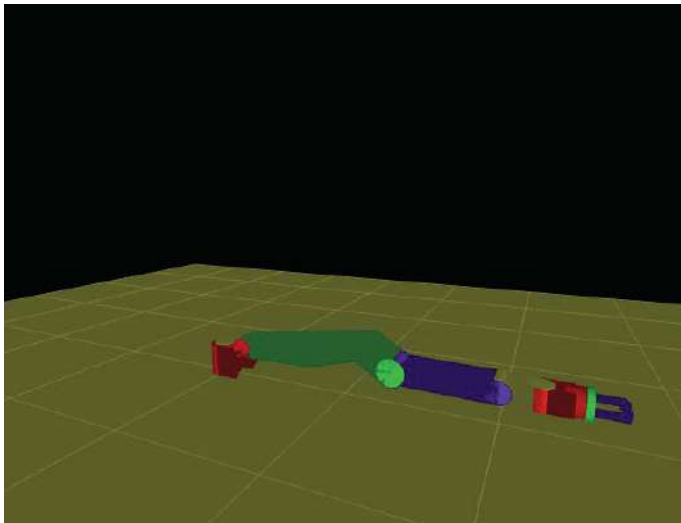


Figura 6.7: Desenho do Robô Manipulador Schilling Titan IV

Com as peças todas disponíveis são necessárias definir as regras dinâmicas do processo, os motores, acionadores, e sensores. Estes passos estão descritos por módulos da seguinte forma:

Para obter uma aproximação real do comportamento de um robô manipulador, inicialmente se definiram as orientações nos eixos de cada atuador de acordo o método de Denavit-Hatenberg.

Pode-se visualizar na figura 6.8 alguns pontos de posicionamento do robô como teste. Aqui, o robô está sendo dirigido para vários pontos coordenados com diversas orientações dentro do ambiente virtual.

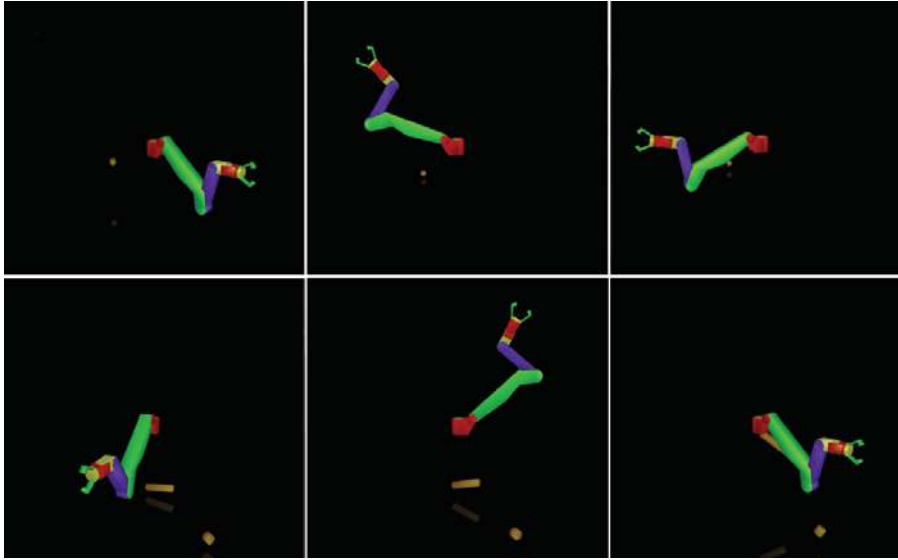


Figura 6.8: Implementação da Cinemática Inversa para o Robô Manipulador

Conforme estudado anteriormente, no Capítulo 3, o conhecimento do modelo dinâmico do robô é de extrema importância para o projeto dos controladores. Embora tenha desenvolvido matematicamente o modelo do robô utilizado, para obter um melhor realismo ainda, implementaram-se as funções de detecção de colisões com as funções da biblioteca ODE embarcadas no CHAI3D. Com essas funções, os elos do robô e os objetos que interagem dentro do ambiente virtual não poderão ocupar um mesmo lugar no espaço para cada instante da simulação. A figura 6.9 mostra como o manipulador agarra uma barra e se movimenta junto com ela.

Com as bibliotecas de ODE, é fácil mostrar na simulação trajetórias realistas quando dois objetos interagem, sem precisar implementar outras equações de movimento além das do manipulador.

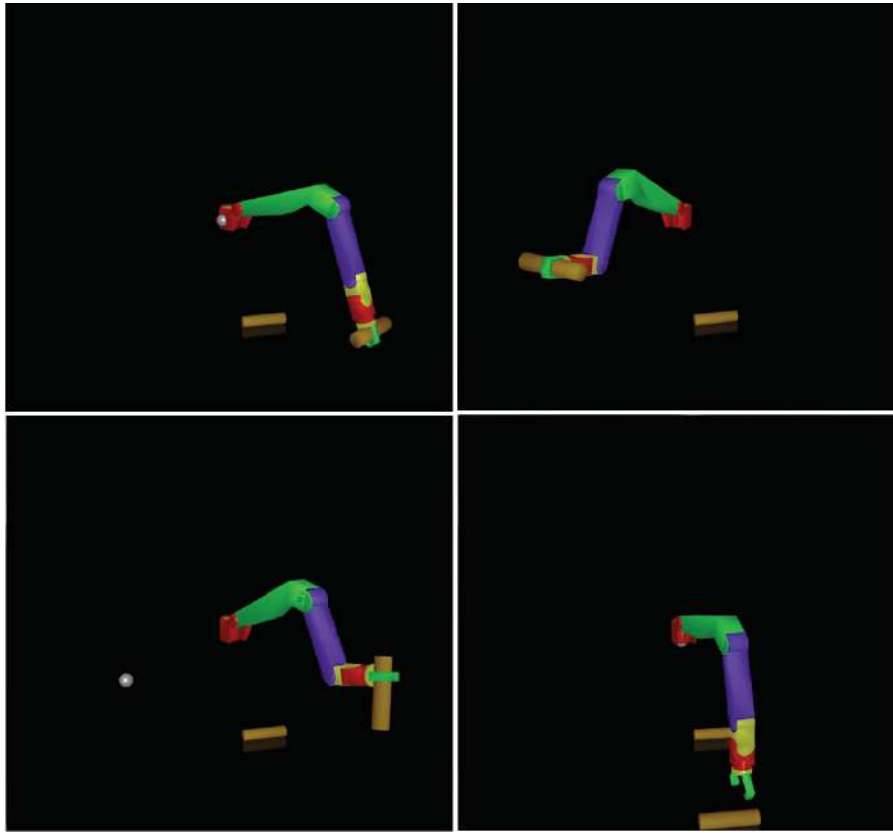


Figura 6.9: Implementação do algoritmo de colisões

6.3.2

Integração da Interface Háptica com o Ambiente Virtual

Depois de ter implementado os comportamentos dinâmicos e cinemáticos para o robô manipulador, com a modelagem matemática dentro do ambiente virtual, o passo seguinte é a integração com o dispositivo háptico, que no início será feita com só um dos dois Falcons. O movimento de um Falcon é somente em 3 graus de liberdade (posição x,y,z). O algoritmo 2 mostra como foi realizada uma metodologia para juntar o robô manipulador com a interface háptica.

O controle de forças foi feito com a função da classe *cGenericHapticDevice*, que envia as forças de reação ao dispositivo háptico por meio da porta USB.

Por fim, para o projeto da interfaces hápticas para teleoperar o robô com uma posição e orientação desejada, implementou-se o algoritmo 3 abaixo

A integração do robô manipulador com o dispositivo háptico possibilita enviar forças para gerar torques e perceber uma sensação de movimento em 5 graus de liberdade com o algoritmo desenvolvido no capítulo 5.

Algorithm 2 Robô Manipulador comandado com o Falcon

```
1: Incluir Algoritmo 1
2: while Encontra-se na área de trabalho do
3:   Ler posição do Falcon
4:   if dado de entrada é ponto singular then
5:     Computar  $q \rightarrow q_d + \Delta q$ .
6:     return false
7:   else
8:     computar nova posição  $q \rightarrow q_d$ 
9:   end if
10:  Computar a dinâmica do robô.
11:  Aplicar o algoritmo de controle para a teleoperação
12:  Computar colisões
13:  Enviar forças de reação da colisão
14: end while
15: return true
```

Algorithm 3 Integração da Interface Háptica com o Ambiente Virtual

```
1: Incluir Algoritmo 1
2: while Encontra-se na área de trabalho do
3:   Ler as posições dos Falcon's
4:   Transformar as posições do Falcon ao sistema de coordenadas global
5:   Sincronizar As áreas de trabalho
6:   if dado de entrada é ponto singular then
7:     Computar  $q \rightarrow q_d + \Delta q$ .
8:     return false
9:   else
10:    computar nova posição  $q \rightarrow q_d$ 
11:   end if
12:  Computar a dinâmica do robô.
13:  Aplicar o algoritmo de controle para a teleoperação
14:  Computar colisões
15:  Enviar forças de reação da colisão
16: end while
17: return true
```

7

Resultados Experimentais

Neste capítulo, apresentam-se os resultados da parte experimental obtidos ao teleoperar um Robô Manipulador com realimentação de força, dentro do ambiente remoto virtual. Tais experimentos têm como objetivo principal testar os diferentes algoritmos de controle, o comportamento da modelagem dinâmica, a implementação da compensação de gravidade e da força/torque de feedback para dar um maior realismo de 5 graus de liberdade ao dispositivo háptico construído com os Falcons. Além disso, inicialmente será feito um experimento em um ambiente simples para observar algumas dificuldades produzidas na teleoperação de objetos.

7.1

Teste da Interface Háptica

Antes de iniciar os experimentos para teleoperar robôs manipuladores com sistemas de interface háptica, serão apresentados testes envolvendo manipular objetos em ambientes remotos com ou sem realimentação de forças. Estes testes são realizados com a ajuda de 5 pessoas, que manipularam o Falcon para cumprir uma tarefa determinada dentro de um ambiente remoto (ambiente virtual).

Esta tarefa consiste em movimentar uma esfera (avatar) em 3 dimensões (x-y-z) dentro do ambiente virtual e passar por um orifício com pequenas folgas, que fica na parte central de uma caixa fixa desenhada para o experimento, descrito no capítulo 6. Os resultados são avaliados através da leitura de forças produzidas na interação dos objetos. Para esse experimento, cada pessoa deve passar 5 vezes a esfera através do orifício da caixa em duas ocasiões: A primeira sem realimentação de forças e a seguinte com realimentação de forças.

A figura 7.1 mostra os passos sequenciais que cada pessoa precisa fazer para cumprir a tarefa. Como podemos observar, cada vez que a esfera interage com a caixa fixa, aparece uma linha que representa a força de reação exercida na esfera pela caixa. O objetivo é atravessar o orifício exercendo a menor força máxima (em módulo) possível.

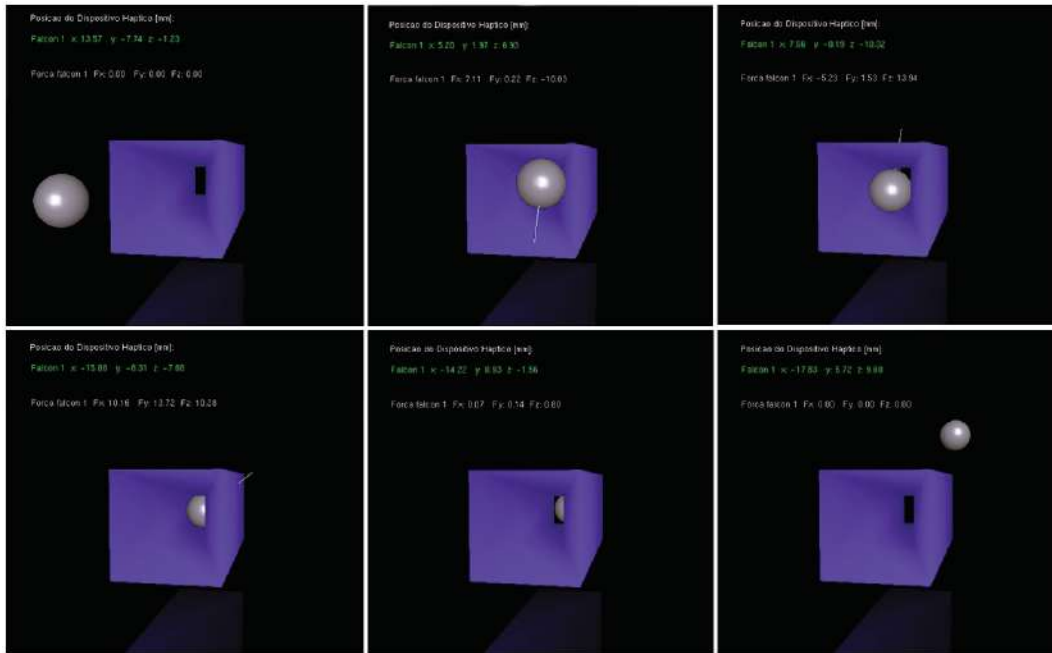


Figura 7.1: Objeto virtual para testes

As figuras 7.2 a 7.11 mostram as forças de reação nos experimentos com os 5 usuários sem e com realimentação de força.

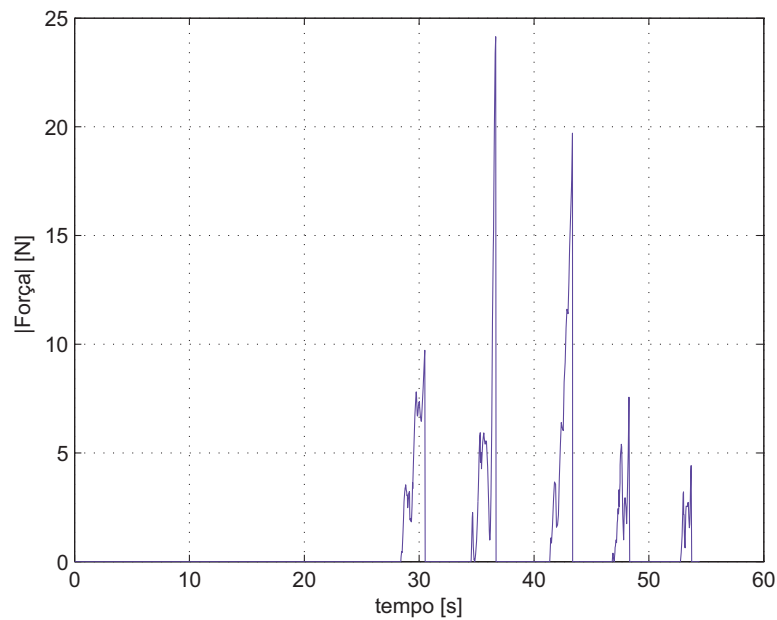


Figura 7.2: Usuário 1: Sem Sensação de Força, mostrando as 5 tentativas de inserção (correspondente aos 5 picos)

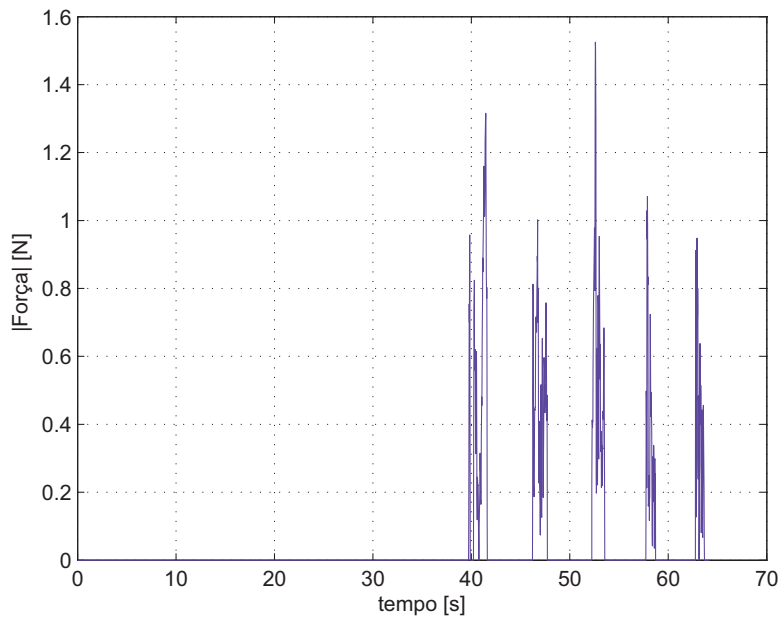


Figura 7.3: Usuário 1: Com Sensação de Força, mostrando as 5 tentativas de inserção (correspondente aos 5 picos)

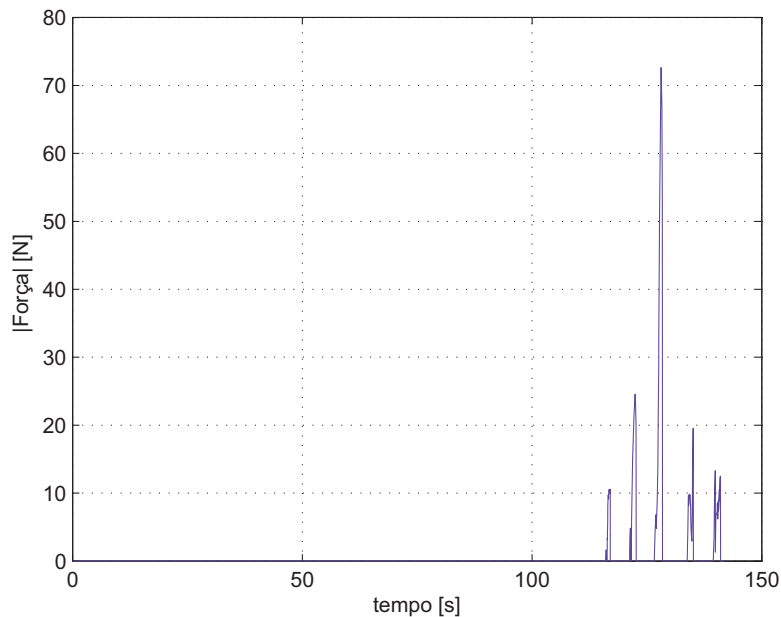


Figura 7.4: Usuário 2: Sem Sensação de Força, mostrando as 5 tentativas de inserção (correspondente aos 5 picos)

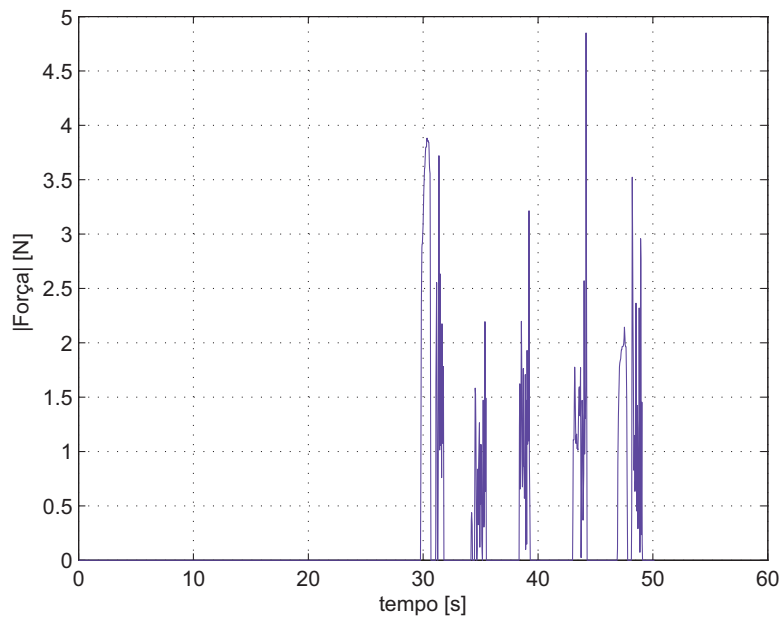


Figura 7.5: Usuário 2: Com Sensação de Força, mostrando as 5 tentativas de inserção (correspondente aos 5 picos)

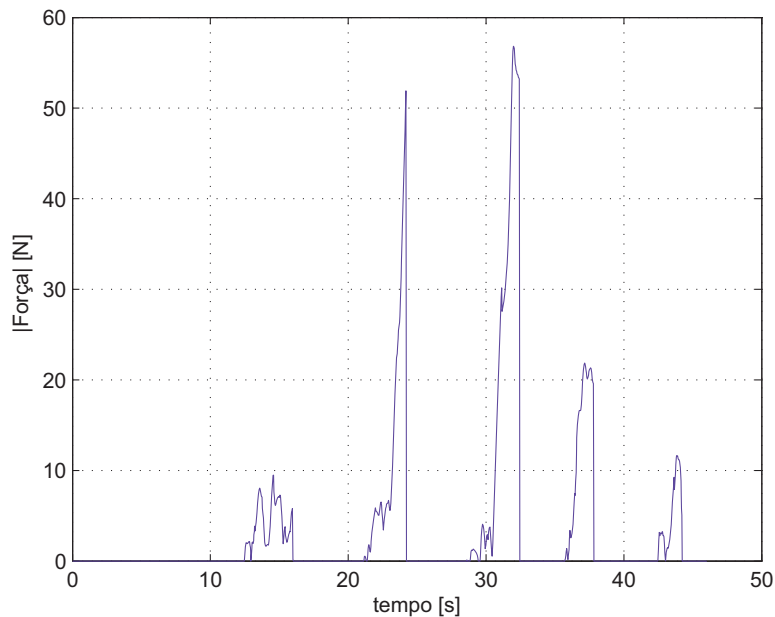


Figura 7.6: Usuário 3: Sem Sensação de Força, mostrando as 5 tentativas de inserção (correspondente aos 5 picos)

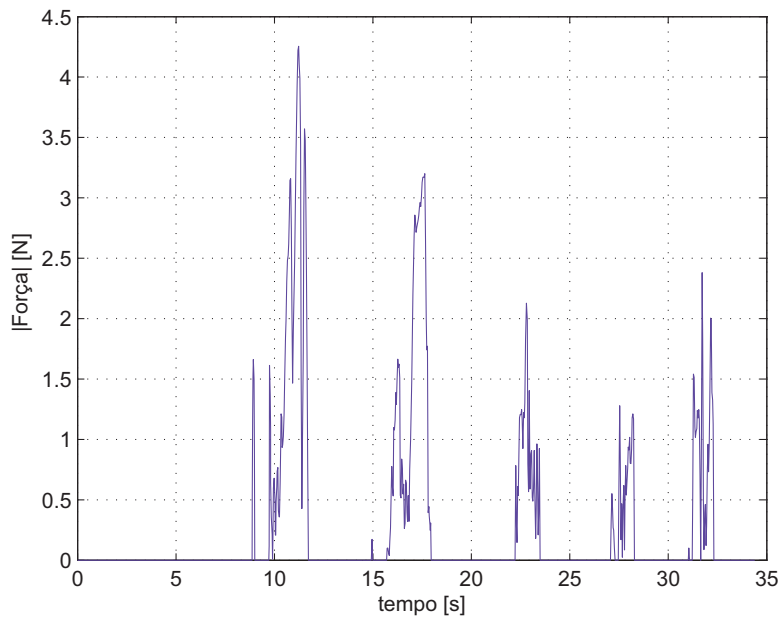


Figura 7.7: Usuário 3: Com Sensação de Força, mostrando as 5 tentativas de inserção (correspondente aos 5 picos)

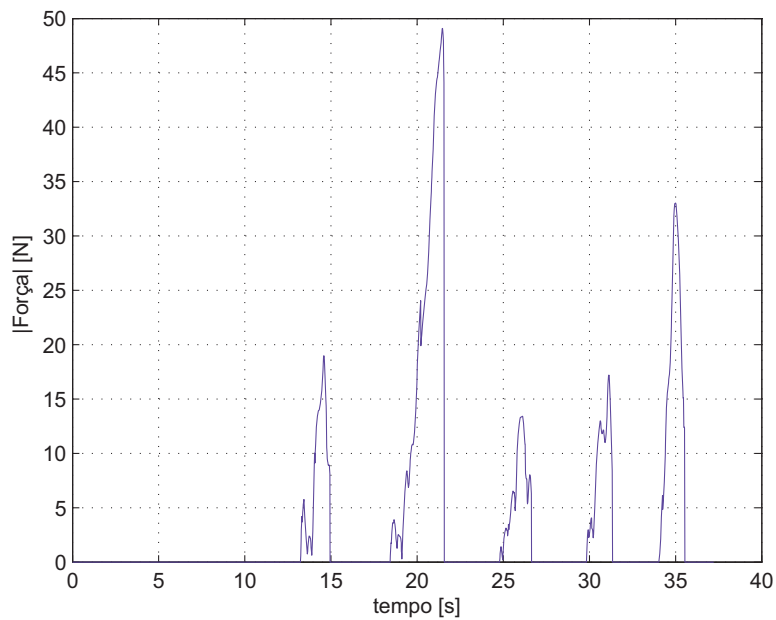


Figura 7.8: Usuário 4: Sem Sensação de Força, mostrando as 5 tentativas de inserção (correspondente aos 5 picos)

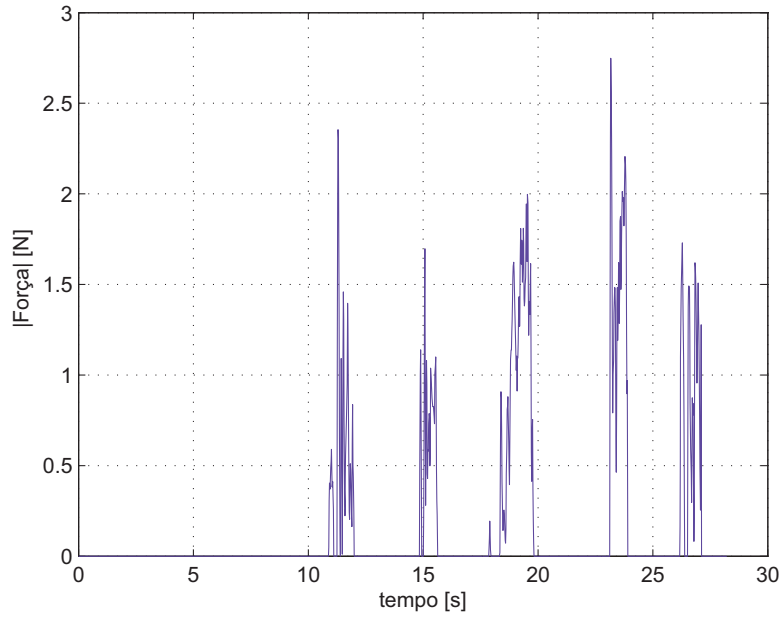


Figura 7.9: Usuário 4: Com Sensação de Força, mostrando as 5 tentativas de inserção (correspondente aos 5 picos)

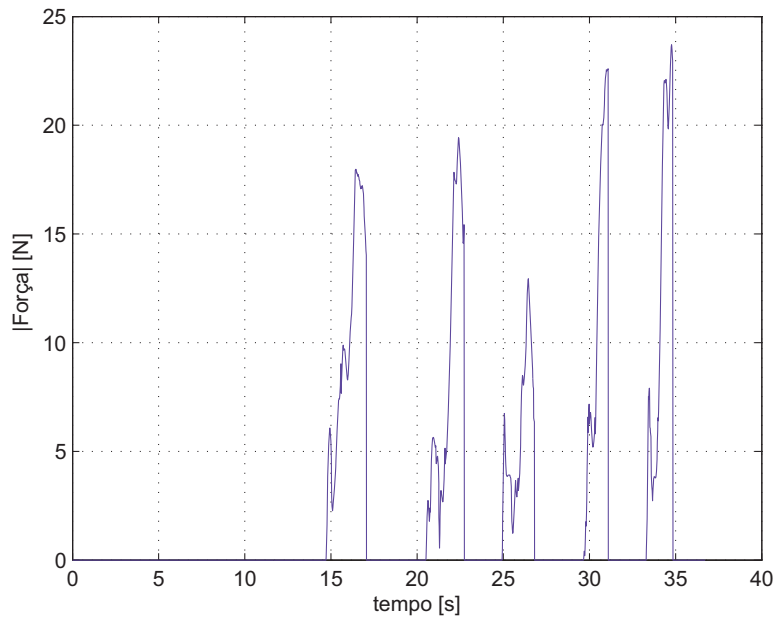


Figura 7.10: Usuário 5: Sem Sensação de Força, mostrando as 5 tentativas de inserção (correspondente aos 5 picos)

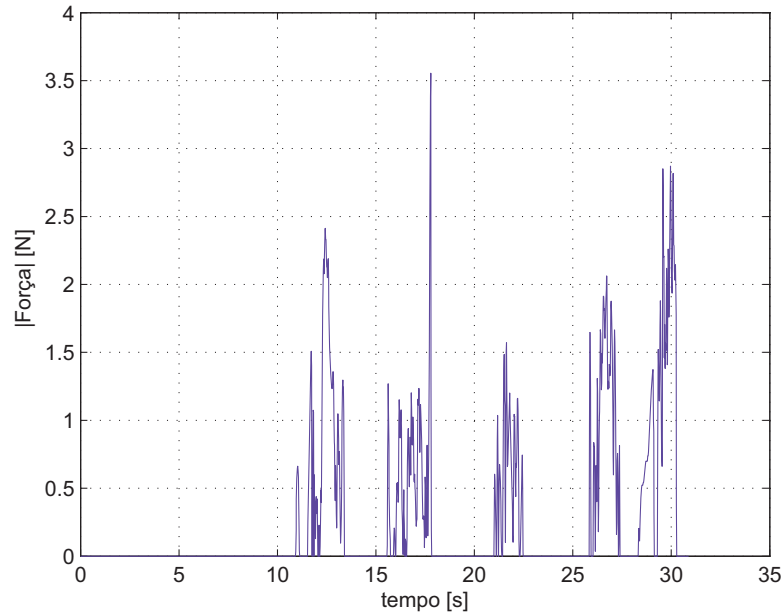


Figura 7.11: Usuário 5: Com Sensação de Força, mostrando as 5 tentativas de inserção (correspondente aos 5 picos)

Observa-se que, devido a pequena folga no orifício, as colisões são inevitáveis na teleoperação. No entanto, sem a sensação de força os picos atingem de 20 a 80N.

Quando a realimentação de força é acionada, as colisões no sistema escravo não geram forças maiores que 5N. Nota-se, também que, em quase todos os experimentos, o tempo de execução de uma mesma pessoa foi significativamente maior quando não havia a sensação de força.

Em fim, a tarefa é executada mais rapidamente e com a maior precisão quando a interface háptica é ativada.

Para se ter uma melhor visualização do experimento, a figura 7.12 mostra um gráfico com os módulos das forças máximas de reação, obtida dos 5 testes realizados pelas 5 pessoas.

As linhas tracejadas representam os testes com realimentação de força, e as contínuas sem.

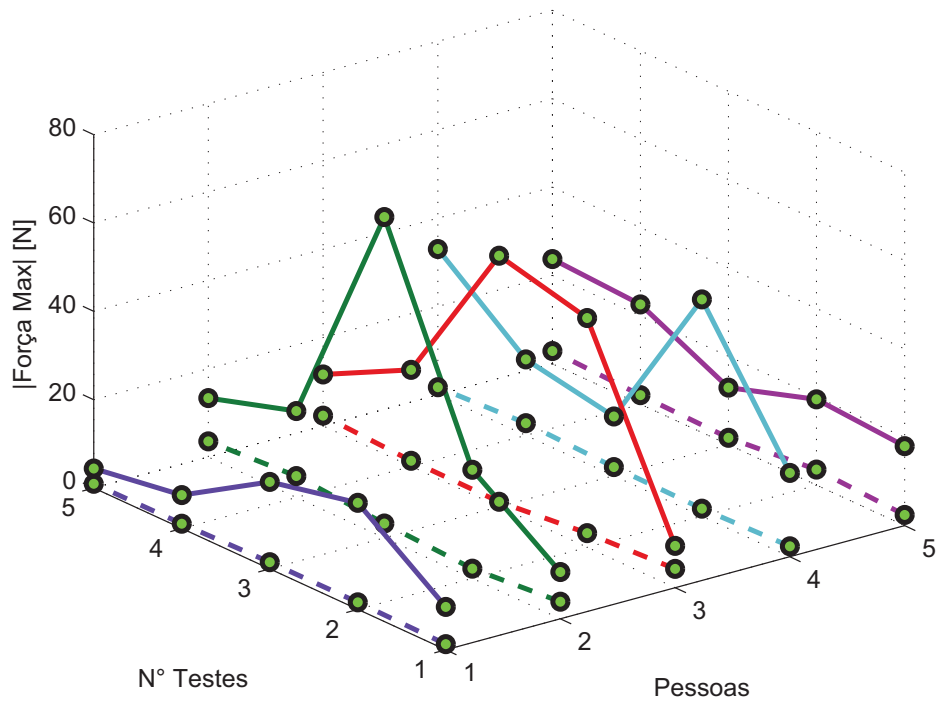


Figura 7.12: Gráficos dos testes feitos com 5 pessoas

7.2

Controle do Robô Manipulador

Nesta seção são apresentados os experimentos controlando o manipulador virtual de seis graus de liberdade figura 7.13. Nota que, como a interface possui apenas 5 graus de liberdade, a junta 6 não será controlado pelo dispositivo háptico projetado, e sim por comandos no teclado do computador. Diversos experimentos são conduzidos utilizando controladores PD, de Torque Computado, e por modos deslizantes, cujos ganhos são calibrados com os valores utilizados no Capítulo 4.

Os parâmetros utilizados para o controle bilateral foram:

$$K_f = \begin{bmatrix} 1500 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1500 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1500 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix}$$

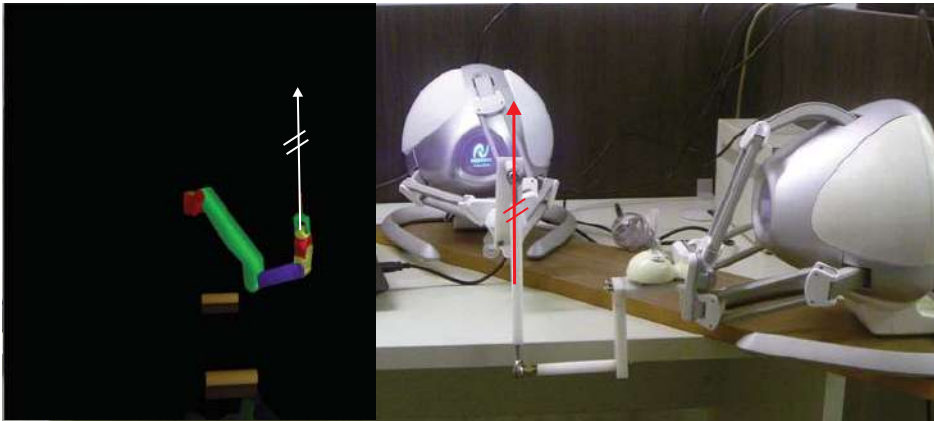


Figura 7.13: Referência e resposta de posição: juntas q_1 , q_2 e q_3 .

As figuras 7.14 - 7.25 mostram os valores de referência (linhas tracejadas) e as posições obtidas (linhas cheias) das juntas, assim com os erros associados, em graus ($^{\circ}$).

Os algoritmos de controle apresentado nesta dissertação, respondem de forma eficiente para um valor de referência variável comanda desde o dispositivo háptico, em toda a teleoperação do robô manipulador. Nota-se que os esforços de controle são reduzidos para o caso do controle por modos deslizante. Cabe ressaltar que a frequência de operação utilizada foi $1KHz$, pois as bibliotecas do ambiente virtual estão especificadas nessa frequência.

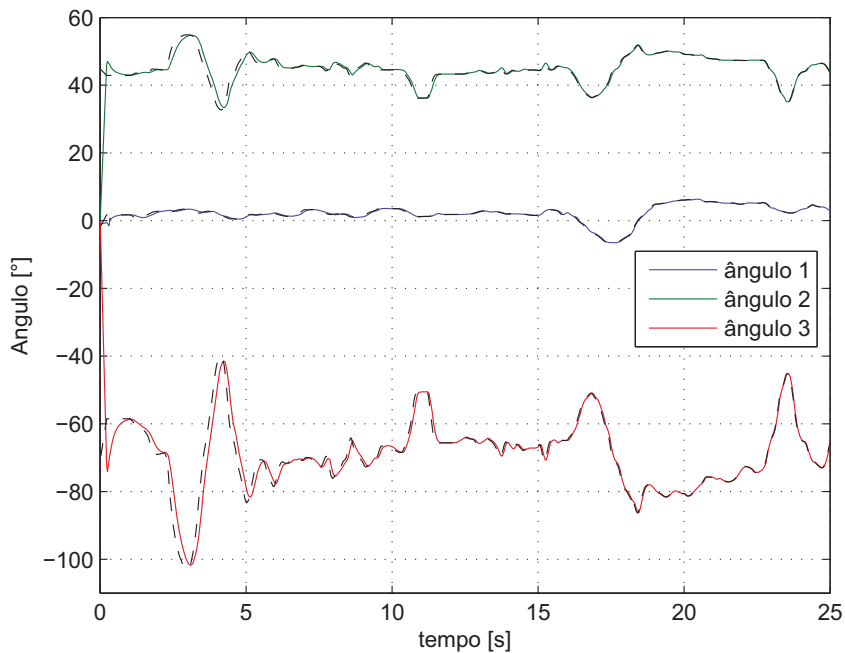


Figura 7.14: Referência e resposta de posição: juntas q_1 , q_2 e q_3 . (PD)

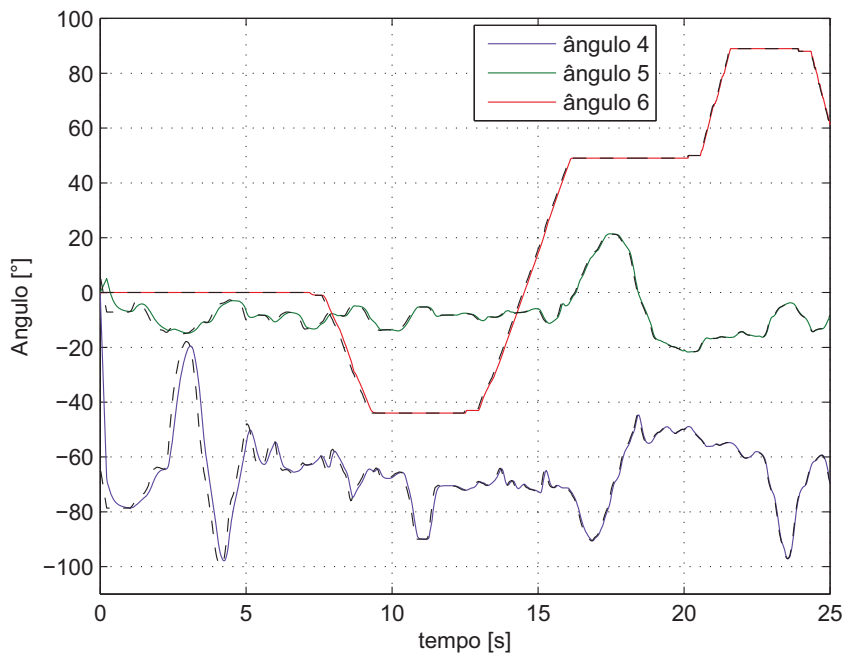


Figura 7.15: Referência e resposta de posição: juntas q_4 , q_5 e q_6 . (PD)

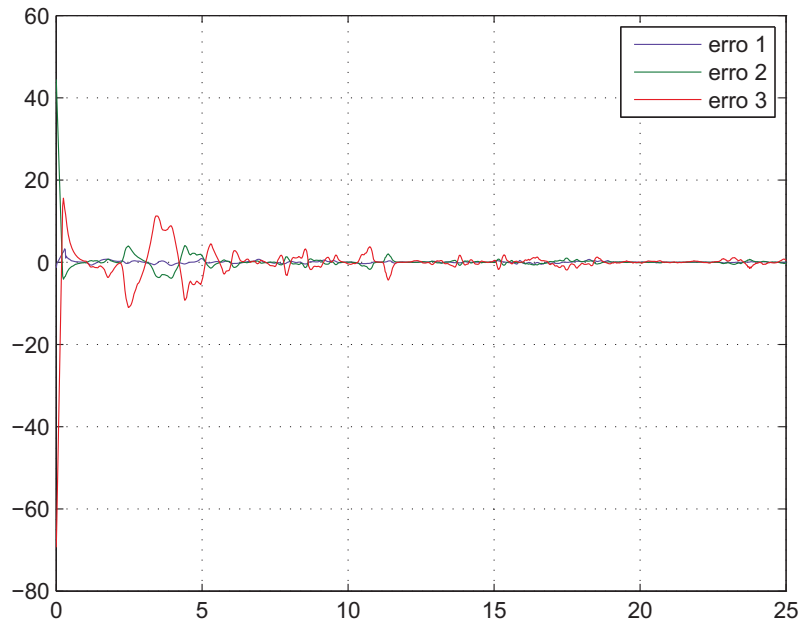


Figura 7.16: Erros de posicionamento: juntas \tilde{q}_1 , \tilde{q}_2 e \tilde{q}_3 . (PD)

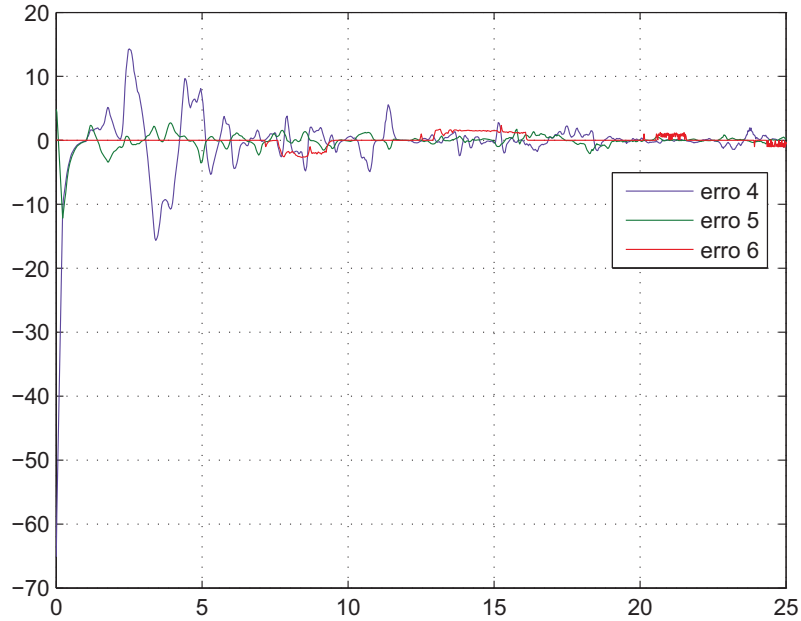


Figura 7.17: Erros de posicionamento: juntas \tilde{q}_4 , \tilde{q}_5 e \tilde{q}_6 . (PD)

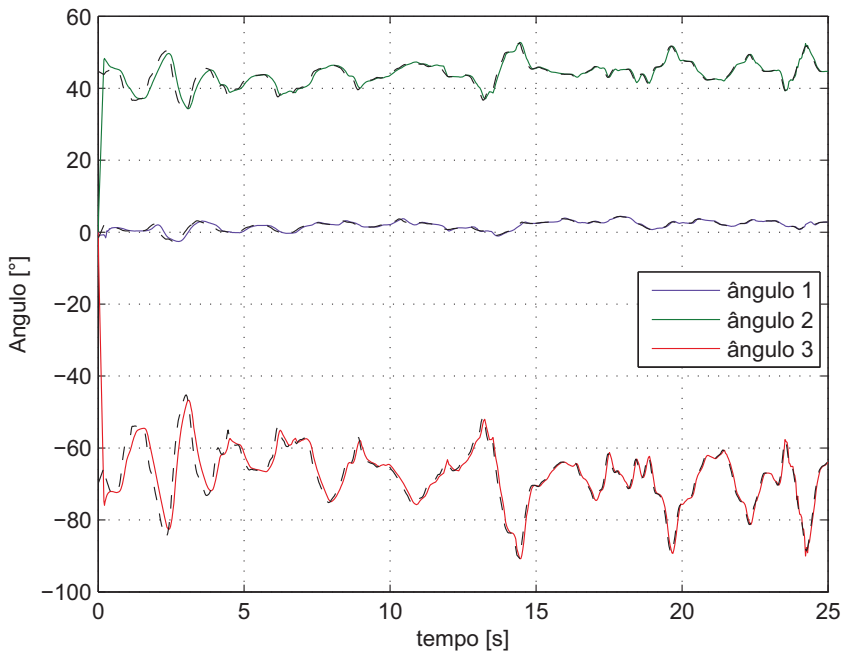


Figura 7.18: Referência e resposta de posição: juntas q_1 , q_2 e q_3 . (Torque Computado)

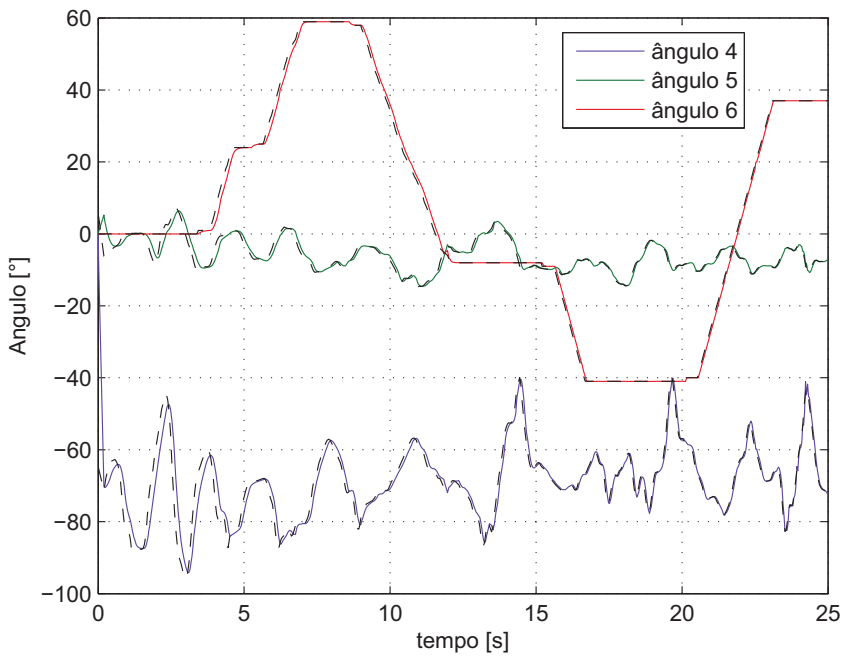


Figura 7.19: Referência e resposta de posição: juntas q_4 , q_5 e q_6 . (Torque Computado)

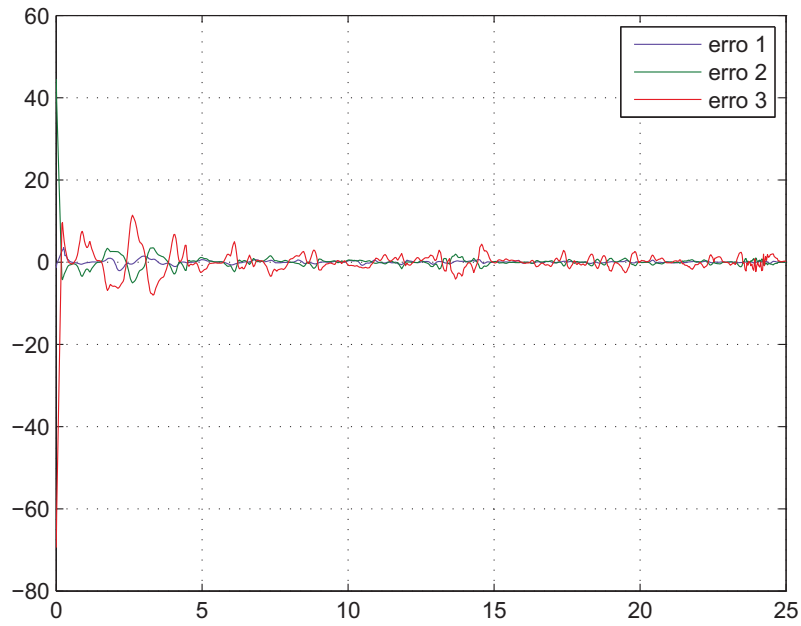


Figura 7.20: Erros de posicionamento: juntas \tilde{q}_1 , \tilde{q}_2 e \tilde{q}_3 . (Torque Computado)

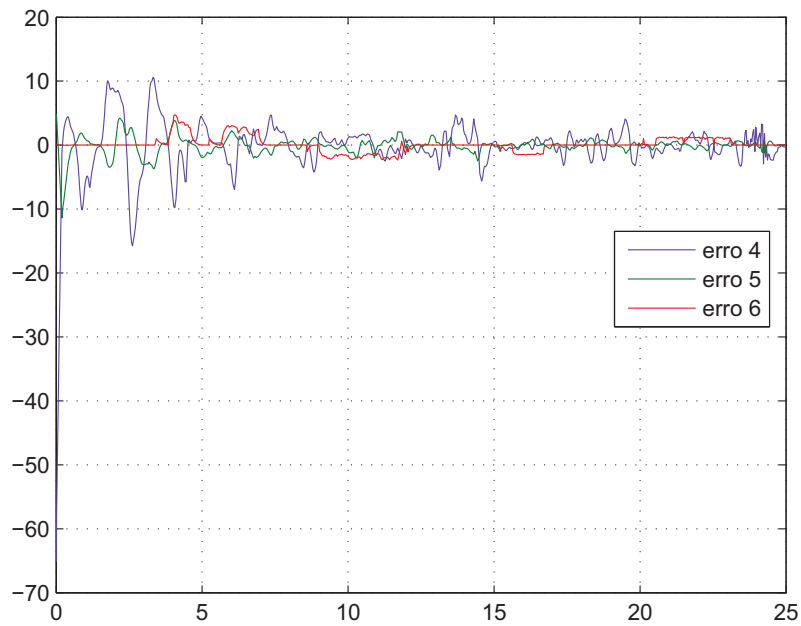


Figura 7.21: Erros de posicionamento: juntas \tilde{q}_4 , \tilde{q}_5 e \tilde{q}_6 . (Torque Computado)

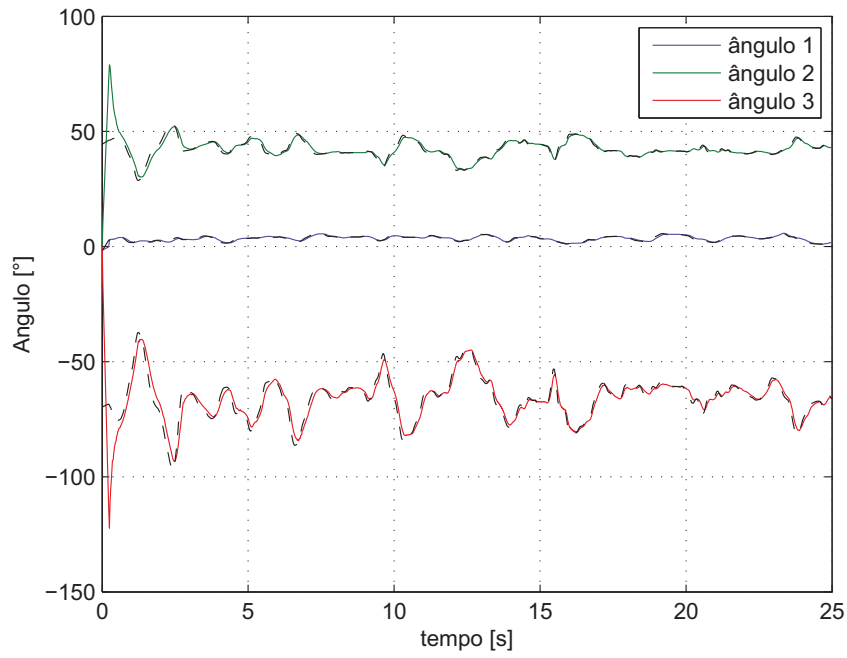


Figura 7.22: Referência e resposta de posição: juntas q_1 , q_2 e q_3 . (Modos Deslizantes)

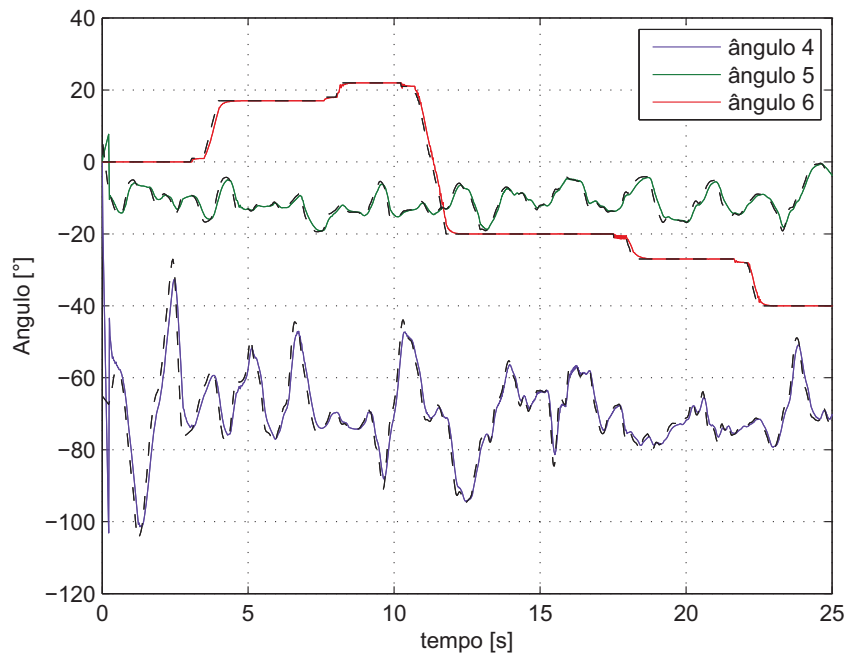


Figura 7.23: Referência e resposta de posição: juntas q_4 , q_5 e q_6 . (Modos Deslizantes)

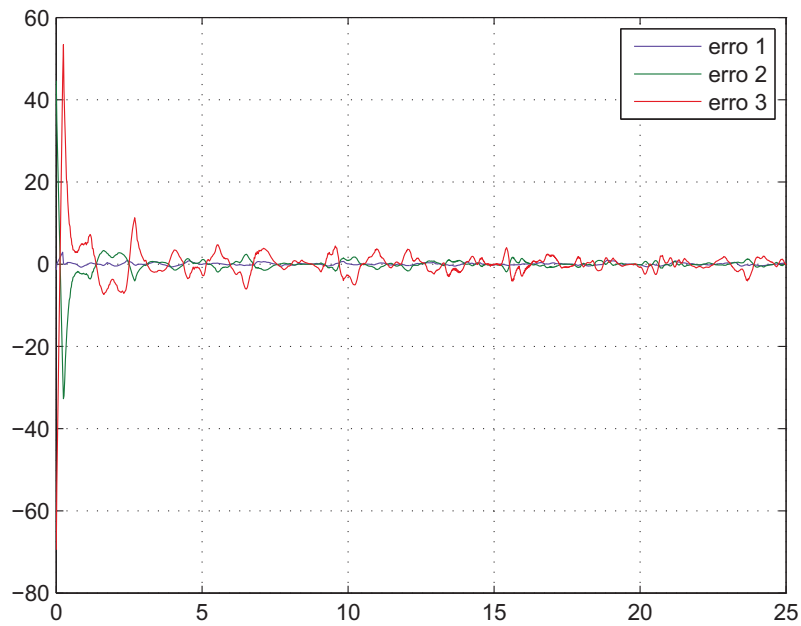


Figura 7.24: Erros de posicionamento: juntas \tilde{q}_1 , \tilde{q}_2 e \tilde{q}_3 . (Modos Deslizantes)

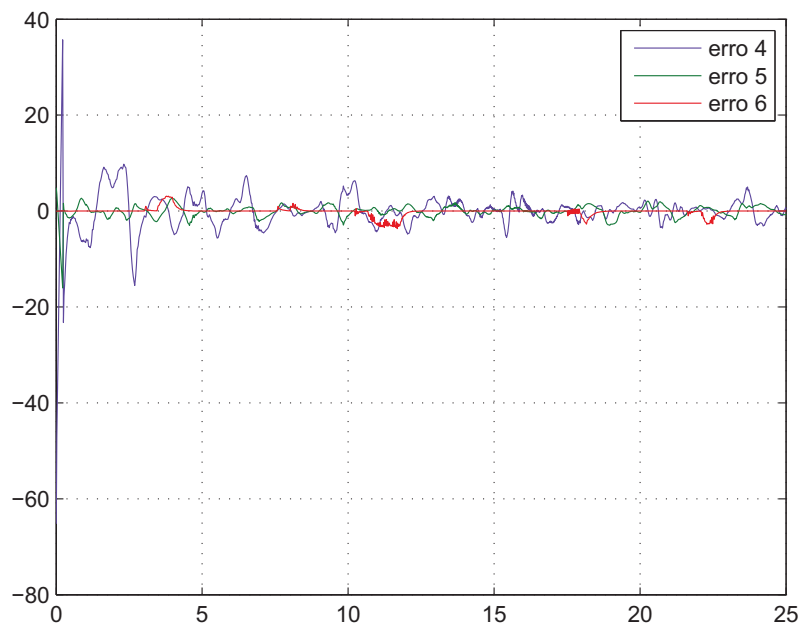


Figura 7.25: Erros de posicionamento: juntas \tilde{q}_4 , \tilde{q}_5 e \tilde{q}_6 . (Modos Deslizantes)

8

Conclusões e Sugestões

Neste Capítulo são discutidos o desempenho do robô virtual e dos controladores projetados, bem como o desempenho do dispositivo háptico construído para a manipulação do elemento terminal do robô. São sugeridos, também, trabalhos futuros.

8.1

Conclusões do Trabalho

O presente trabalho apresentou o desenvolvimento de um sistema háptico de 5 graus de liberdade, capaz de prover a um teleoperador as sensações de força e torque sofridas por um sistema escravo controlado. O sistema foi implementado a partir da conexão de 2 interfaces hápticas comerciais de 3 graus de liberdade "Novint Falcon", que possuem baixo custo e uma resolução e frequência suficientes para as tarefas propostas. O sistema foi modelado cinemática e estaticamente, e um controle de compensação de gravidade foi implementado para evitar que efeitos gravitacionais interferissem nas sensações de força pelo usuário.

Seu desempenho foi testado em ambientes gráficos tridimensionais especialmente desenvolvidos para essa Dissertação, utilizando bibliotecas de programação dentro do ambiente CHAI3D. No primeiro ambiente implementado, foram criados uma caixa virtual contendo um orifício central e uma esfera para fazer o papel de avatar. O objetivo do experimento foi testar a eficiência de um teleoperador em comandar a esfera para inseri-la e atravessa-la pelo orifício, que possui folgas muito pequenas e visualmente imperceptíveis, utilizando a interface com a realimentação de força ligada ou desligada. Testes com 5 usuários comprovaram que a realimentação de força permitiu uma diminuição significativa nas magnitudes das forças de colisões entre a esfera e o orifício, além de diminuir o tempo de execução da tarefa. A sensação de toque transmitiu informação rica e detalhada sobre o objeto com o qual o usuário interagiu. Quando combinada com outros sentidos, especialmente a visão, o torque aumentou dramaticamente a quantidade de informação enviada ao cérebro para processamento. O aumento na informação reduz erros do usuário, como foi

observado, permite a redução do tempo de execução.

Um ambiente virtual mais elaborado também foi desenvolvido, implementando a modelagem e controle de um manipulador virtual. O manipulador virtual utilizado na teleoperação foi o Schilling Titan IV, cujos modelos cinemáticos e dinâmicos foram obtidos e apresentados em detalhe. O controle local do manipulador virtual foi implementado utilizando técnicas PD, de Torque Computado, e por Modos Deslizantes baseado no erro mínimo. Simulações demonstraram que as 2 últimas técnicas, por serem não lineares, resultaram em menores erros de posicionamento e seguimento de trajetórias.

O ambiente virtual inclui também diversos objetos para treinamento de teleoperação, e em tese pode comportar qualquer outro manipulador, desde que seu modelo seja introduzido no software desenvolvido. Este ambiente de simulação pode vir a ter um papel importante no planejamento e execução de operações reais de robótica, além de poder ser utilizado em treinamento e planejamento de tarefas teleoperadas em ambientes inóspitos.

8.2

Sugestões para Trabalhos Futuros

Uma fonte de erro na realimentação de força do sistema desenvolvido vem da limitação da biblioteca CHAI3D, que permite enviar os comandos de força apenas de forma indireta aos atuadores do Falcon. Seria mais eficiente comanda-los de forma direta, evitando assim eventuais erros de calibragem na cinemática do Falcon. Recentemente, a empresa Novint Technology liberou uma biblioteca chamada HDAL, que permite controlar diretamente os atuadores, abrindo totalmente o acesso ao hardware interno do dispositivo. Trabalhos futuros poderiam migrar as rotinas desenvolvidas em CHAI3D para HDAL, melhorando a precisão das forças de feedback.

Para melhorar o software do ambiente virtual desenvolvido, poderiam também ser incluídos outros algoritmos de colisões, outros algoritmos para controladores de força, e principalmente a otimização do código para permitir uma maior modularização das rotinas.

Os problemas tecnológicos de atrasos de comunicações entre mestre e escravo foram estudados para os possíveis esquemas de controle bilateral, através do ambiente virtual simulado. No entanto, a migração do sistema desenvolvido para um ambiente real iria requerer um maior estudo destes efeitos, incluindo perturbações ou atrasos não modelados.

Neste trabalho não se utilizou um controle de impedância para o dispositivo háptico construído. Seria recomendável implementar estes tipos de controladores para obter um melhor desempenho na teleoperação e em espe-

cial melhorar a relação de inércias mestre-escravo.

Finalmente, notou-se que a rigidez da estrutura dos Falcon não era alta o suficiente para não interferir na sensação de força do usuário. O uso de outros dispositivos mais rígidos - mas também de custo mais elevado - como o PhantomOmni seria uma sugestão para futuras implementações.

Referências Bibliográficas

- [1] LARKUM, T.; BROOME, D.. **Advanced controller for an underwater manipulator**. Proceedings of the Third IEEE Conference, vol.2:p.1081–1086, 1994.
- [2] CONTE, G.; PERDON, A. ; ZANOLI, S.. **A navigation and inspection system for rov's**. Cahllenges of Our Global Environment:Conference Proceedings, vol.1:p.354–359, 1995.
- [3] BUTNER, S. E.; GHODOUSSI, M.. **Transforming a surgical robot for human telesurgery**. IEEE Transactions on Robotics and Automaton, vol.5(19):p.818–824, 2003.
- [4] BALLANTYNE, G.. **Trobotic surgery, telerobotic surgery, telepresence, and tementoring, surgical endoscopy**. Springer Verlag Verlag New York, vol.6:p.1389–1402, 2002.
- [5] TEAM, R.. **Characterization of the martian surface deposits by the mars pathfinder rover, sojourner**. Science, vol.278(5344):p.1765–1768, 1997.
- [6] MISHKIN, A.; MORRISON, J. ; NGUYEN, T.. **Experiences with operations and autonomy of the mars pathfinder microrover**. Aerospace Conference: Proceeding IEEE, vol.2:p.337–351, 1998.
- [7] OYAMA; TSUNEMOTO, T. ; INOUE. **Experimental study on remote manipulation using virtual reality**. Presence-Teleop. Virtua Environment, vol.2(2):p.101–110, 1993.
- [8] BLACKMON; STARK. **Model based supervisory control in telero-botics**. Presence-Teleop. Virtua Environment, vol.5(2):p.205–223, 1996.
- [9] THOLEY, G.. **A teleoperative Haptic Feedback Framework for Computer-Aided Minimally Invasive Surgery**. Tese de Doutorado, Drexel University, Philadelphia-USA, 2003.
- [10] OAKLEY, I.; MCGEE, M.; A, B. S. ; GRAY, P. D.. **Putting the feel un look and feel**. Relatório Técnico, 2000.
- [11] KLATZKY, R. L.; LEDERMAN, S. J.. **Modality specificity in cognition: The case of touch. the nature of remembering**. Relatório Técnico, 2000.

- [12] LAMOTTE, R. H.; SRINIVASAN, M. A.. **Surface microgeometry: Neural encoding and perception.** In: Information processing in the Somatosensory System, p. p.49–58, 1991.
- [13] CRAIG, J. C.; ROLLMAN, G. B.. **Somesrhesis annual review of psychology.** vol.50:p.305–331, 1999.
- [14] DARIAN, SMITH, I.. **The sense of touch:performance and peripher-
al neural processes. handbook of physiology:the nervous system
and sensory processes.** Bethesda: American Physiology Society, p. p.739–
746, 1984.
- [15] AKERMAN, P. L.. **A theory of intellectul development: process,
personality, interests and knowledge.** Intelligence, vol.22:p.227–257,
1996.
- [16] AUKSTAKALNIS, S. B. D.. **The art and science of virtual reality.**
Relatório Técnico, 1992.
- [17] BURDEA, G.; COIFFET, P.. **Virtual Reality Technology.** John Wiley
& Sons, NY, 1994.
- [18] JACOBSON, L.. **Virtual reality:a status report.** AI Expert, p. p.26–33,
1991.
- [19] KRUEGER, M. W.. **Artificial Reality II.** Addison Wesley, MA, 1991.
- [20] ROSEMBLU, L.; BURDEA, G. ; TACHI, S.. **Virtual reality reborn.** IEEE
Computer Graphics and Applications, vol.18(n.6):p.21–23, 1998.
- [21] MACHADO, L. S.. **Conceitos básicos da realidade virtual.** Relatório
Técnico, 1995.
- [22] ACKERMAN; DIANE. **Uma História Natural dos Sentido.** 2nd ed.
Bertrand, Br, 1990.
- [23] NETTO, A. V.; MACHADO, L. S. ; OLIVEIRA, M. C. F.. **Realidade
Virtual.** Visual Books, Florianópolis, 2002.
- [24] BROOKS, F. P.. **What’s real about virtual reality?** IEEE Computer
Graphics and Applications, vol.19(n.6):p.16–27, 1999.
- [25] MACHADO, L. S.. **A realidade virtual no modelamento e simulação
de procedimentos invasivos em oncologia pediátrica: Um estudo
de caso no transplante de medula óssea.** Tese de Doutorado, Escola
Politécnica da Universidade de São Paulo, São Paulo, 2003.

- [26] MCLAUGHLIN, M. L.; HESPANHA, J. ; SUKHATME, G.. **Touch in Virtual environments: Haptics and the design of interactive systems**. Prentice-Hall, USA, 2002.
- [27] BURDEA, G. C.; COIFFET, P.. **Virtual Reality Technology**. Wiley-IEEE Press, USA, 2003.
- [28] LOGITECH'S. **Logitech's wingman force feedback mouse**. Relatório Técnico, 2005.
- [29] MAHVASAH, M.; HAYWARD, V.. **Passivity-based high-fidelity haptic rendering of contact**. IEEE International Conference on Robotics and Automation, 3:p.3722–3728, 2003.
- [30] SALISBURY, K.; BROCK, D.; MASSIE, T.; SWARUP, N. ; ZILLES, C.. **Haptic rendering: Programming touch interaction with virtual objects**. In: ACM Symposium on Interactive 3D Graphics, 3:p.123–130, 1995.
- [31] MACLEAN, K. E.. **the haptic camera: a technique for characterizing and playing back haptic properties of real environments**. In: ASME Dynamic Systems and Control Division, vol.58:p.459–467, 1996.
- [32] MORGENBESSER, H. B.; SRINIVASAN, M. A.. **Force shading for haptic shape perception**. In: Symposium on Haptic Interfaces for a Virtual environment and Teleoperator Systems, p. p.407–412, 1996.
- [33] VOYLES, R. M.; FEDDER, G. ; KHOSLA, P. K.. **3-d firm grasp with the scrollic gripper**. IEEE International conference on Robotics and Automaton, p. p.1662–1667, 1995.
- [34] VOYLES, R. M.; FEDDER, G. ; KHOSLA, P. K.. **Desing of a modular tactile sensor and actuator based on an electroorheological gel**. In Proceedings of the IEEE Computer Graphics and Applications, p. p.13–17, 1996.
- [35] BICCHI, A.; SCILINGO, E. P.. **A sensorized minimally invasive surgery tool for detecting tissual elastic properties**. Proceedings of the IEEE International Conference on Robotics and Automaton, p. p.884–888, 1996.
- [36] MASCARO, S.; ASADA, H. H.. **Virtual switch human-machine interface using fingernail touch sensor**. Proceedings of the IEEE International conference on Robotics and Automaton, p. p.2533–2538, 1999.

- [37] FASSE, E. D.; HOGAN, N.. **Quantitative measurement of haptic perception**. Proceedings of the IEEE International Conference on Robotics and Automaton, p. p.3199–3204, 1994.
- [38] AUKSTAKALNIS, S.. **Silicon mirage: The art and science of virtual reality**. peatchpit press. Relatório Técnico, 1992.
- [39] SOUZA, D. F. L.; VALDEK, M. C. O. ; MORAES, R. M.. **Siteg - sistema interativo de treinamento em exame ginecológico**. In: Proc of Symposium on Virtual reality (SVR), p. p.445–456, 2006.
- [40] BASDOGAN, C.; SRINIVASAN, M. A.. **Haptic Rendering in Virtual Environments**, in **Virtual Environments HandBook**. Stannet, 2002.
- [41] SENSABLE. **Manual de especificações do phantom omni**.<http://www.sensable.com/haptic-phantom-omni.htm>. Relatório Técnico, 2009.
- [42] WATSON, K.. **Development of stereoscopic-haptic virtual environments**. 12th IEEE Symposium on Computer-Based Medical Systems, p. p.29–34, 1999.
- [43] VINCE, J.. **Virtual Reality Systems**. Addison Wesley, NY, 1995.
- [44] SUTHERLAND, I.. **The ultimate display**. In: IFIP Congress Proceedings, p. p.506–5008, 1965.
- [45] PIMENTEL, K.; TEIXEIRA, K.. **Virtual Reality Through the New Looking Glass 2nd ed**. McGraw Hill, NY, 1995.
- [46] SZÉKELY, G.; SATAVA, R.. **Virtual reality in medicine**. BMJ, vol.319:p.1–4, 1999.
- [47] SELA, G.. **Real-time incision simulation using fem-based discontinuous free form deformation**. Computer Aided Design, p. p.01–07, 2007.
- [48] D'AULIGNAC, D.. **Towards a realistic echographic simulator**. Medical Image Analysis, p. p.71–81, 2006.
- [49] BLYTH, P.; STOTT, N. S.. **A simulation-based training system for hip fracture fixation for use within the hospital environment**. Injury, Int. J. Care Injured, p. p.1197–1203, 2007.

- [50] CARDOSO, A.; LAMOUNIER, E. A.. **Aplicações na educação e treinamento in: Realidade virtual e aumentada - uma abordagem tecnológica.** Editora SBC - Sociedade Brasileira de Computação PA, vol.1:p.343–357, 2008.
- [51] ASADA, H.; SLOTINE, J.. **Robot Analysis and Control.** Jhon Wiley, New York, 1985.
- [52] SCHILLING. <http://www.schilling.com> **Robot Schilling Titan IV.** 2010.
- [53] LUH, J. Y. S.; WALKER, M. W. ; PAUL, R. P.. **On-line computational scheme for mechanical manipulators.** ASME Journal of Dynamic Systems Measurement and Control, vol.102:p.069–076, 1980.
- [54] WALKER, M. W.; E, O. D.. **Efficient dynamic computer simulation of robotic mechanisms.** Journal of Dynamics Systems Mesasurement and Control, vol.104:p.205–211, 1981.
- [55] SCIAVICCO, L.; SICILIANO, B.. **Modeling and Control of Robot Manipulator.** McGraw-Hill, New York, 1996.
- [56] AN, H.; ATKESON, G. ; HOLLERBACH, M.. **A Mathematical Introductionto Robotic Manipulation.** MIT Press, Cambridge, 1994.
- [57] MURRAY, R.; XI, Z. ; SASTRY, S.. **A Mathematical Introductionto Robotic Manipulation.** CRC Press, BocaRaton, 1994.
- [58] YSHIKAWA, T.. **Foundations of Robotics.** MIT Press, Cambridge, 1990.
- [59] AN, C.; ATKESON, C. ; HOLLERBACH, J.. **Model-based control of a robot manipulator.** MIT Press, 1988.
- [60] ITKI, U.. **Control Systems of Variable Strucure.** Wiley, 1976.
- [61] UTKIN, V.. **Variable struture systems with sliding modes (a survey).** IEEE Transaction on Automatic Control, vol.22(2):p.212–222, 1977.
- [62] SLOTINE, J.; LI, W.. **Applied Nonlinear Control.** Pretince Hall, New Jersey, 1991.
- [63] NUNES, L. E.. **Controle por Modos Deslizantes de Ordem Superior com Estabilidade Global.** Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro - BRAZIL, 2004.

- [64] KNOWLES, G.. **A Introduction to Applied Optimal Control.** Academic Press, USA, 1981.
- [65] DRAZENOVIC, B.. **The invariance conditions for variable structure systems.** *Automatica*, vol.5:p.287–295, 1969.
- [66] STAMPER, R. E.. **A Three Degree of Freedom Parallel Manipulator with Only Translational Degrees of Freedom.** PhD thesis, Department of Mechanical Engineering. Tese de Doutorado, The University of Maryland, Department of Mechanical Engineering, 1997.
- [67] STEVEN, M.; HILLIER, N.. **Characterisation of the novint falcon haptic device for application as a robot manipulator.** *Australasian Conference on Robotics and Automation (ACRA)*, vol.1:p.001–004, 2009.
- [68] RGAHAVAN, M.; ROTH, B.. **Solving polynomial systems for the kinematic analysis and synthesis of mechanisms and robot manipulators.** *Transactions of the ASME Special 50th Anniversary Design Issue*, vol.117:p.071–079, 1995.
- [69] SALMON, G.. **Lessons introductory to the modern higher algebra.** chelsea publishing co., ny. Relatório Técnico, 1964.
- [70] RGAHAVAN, M.; ROTH, B.. **Position kinematics of a three-limbed mixed mechanism.** *ASME Journal of Mechanical Design*, vol.116:p.924–929, 1994.
- [71] LIN, W.; GRIFFIS, M. ; DUFFY, J.. **Forward displacement analyses of the 4-4 stewart platforms.** *ASME Journal of Mechanical Design*, vol.114:p.444–450, 1992.
- [72] GOSSELIN, C.; ANGELES, J.. **The optimum kinematic design of a planar three-degree-of-freedom parallel manipulator.** *Journal of Mechanisms, Transmissions, and Automation in Design*, vol.110:p.35–41, 1988.
- [73] WALDRON, K.; HUNT, K.. **Series-parallel dualities in actively coordinated mechanisms.** *Proceedings of the 4th International Symposium on Robotic Research*, MIT Press, vol.110:p.175–181, 1988.
- [74] OBLACK, D.; KOHLI, D.. **Boundry surfaces, limit surfaces, crossable and noncrossable surfaces in workspace of mechanical manipulators.** *Journal of Mechanisms, Transmissions, and Automation in Design*, vol.110:p.389–396, 1988.

- [75] LUNG-WEN, T.. **Robot Analysis - The Mechanic of Serial and Parallel Manipulators**. Jhon Wiley, New York, 1999.
- [76] AMAN, V.; SCOTT, T.; MCCLAIN, W. ; ABBOUT, J.. **How to build an inexpensive 5-dof haptic devicece using novint falcons**. EuroHaptics, p. p.136–143, 2010.
- [77] MARTIN, S.; HILLIER, N.. **Characterisation of the novint falcon haptic device for application as a robot manipulator**. In Proc Australasian Conf. Robotics and Automation, vol.110:p.389–396, 2009.
- [78] PEÑIN, L. F.. **Control Bilateral de robots Teleoperador, contribuciones en reflexión de fuerzas**. Tese de Doutorado, Universidad Politecnica de Madrid, Madrid, 1998.
- [79] WIKIPEDIA. <http://pt.wikipedia.org/>.
- [80] CAY, H.. **Padrões e Projeto Orientados a Objetos**. Jhon Wiley, São Paulo - Brasil, 2006.
- [81] GUILHERME, S.. **Treinamento Profissional em Java**. Digerati Books, Brasil, 2006.
- [82] SEBESTA, R. W.. **Conceitos de Linguagens de Programação**. Jhon Wiley, São Paulo - Brasil, 2002.
- [83] COHEN, M.; HARB, I.. **OpenGL - Uma Abordagem Prática e Objetiva**. Novatec, São Paulo - Brasil, 2002.
- [84] RUSSELL, S.. **Open dynamics engine. use guide**, <http://www.ode.org/>. Relatório Técnico, 2004.
- [85] HECKER, C.. **The next frontier game developer**. Relatório Técnico, 1996.
- [86] CONTI, F.. **The chai library - project of computer science department, stanford university**. Relatório Técnico, 2006.

A

Apêndice A

Este apêndice contém as equações relacionadas à cinemática e dinâmica do robô manipulador Schilling Titan IV, bem como as matrizes do jacobiano que relaciona as velocidades da ferramenta com respeito às juntas do robô.

A.1

Matriz Homogênea

Considere os parâmetros D-H da tabela 3.1 obtidos da modelagem do robô mostrada na figura 3.4, então podem-se obter as matrizes homogêneas de transformação

$$A_1^0 = \begin{bmatrix} \cos \theta_1 & 0 & \sin \theta_1 & l_2 \cos \theta_1 \\ \sin \theta_1 & 0 & -\cos \theta_1 & l_2 \sin \theta_1 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A-1})$$

$$A_2^1 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & l_3 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & l_3 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A-2})$$

$$A_3^2 = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & l_4 \cos \theta_3 \\ \sin \theta_3 & \cos \theta_3 & 0 & l_4 \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A-3})$$

$$A_4^3 = \begin{bmatrix} \cos \theta_4 & 0 & -\sin \theta_4 & l_5 \cos \theta_4 \\ \sin \theta_4 & 0 & \cos \theta_4 & l_5 \sin \theta_4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A-4})$$

$$A_5^4 = \begin{bmatrix} \sin \theta_5 & 0 & \cos \theta_5 & 0 \\ -\cos \theta_5 & 0 & \sin \theta_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A-5})$$

$$A_6^5 = \begin{bmatrix} \cos \theta_6 & -\sin \theta_6 & 0 & 0 \\ \sin \theta_6 & \cos \theta_6 & 0 & 0 \\ 0 & 0 & 1 & l_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A-6})$$

Com estas matrizes, obtém-se a matriz \mathbf{T}

$$\mathbf{T} = \mathbf{A}_1^0 \mathbf{A}_2^1 \mathbf{A}_3^2 \mathbf{A}_4^3 \mathbf{A}_5^4 \mathbf{A}_6^5 \quad (\text{A-7})$$

Definindo

$$\mathbf{T} = \begin{bmatrix} \mathbf{n} & \mathbf{t} & \mathbf{b} & \mathbf{x} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Então obtemos a posição e orientação do efetuador terminal do robô.

$$\mathbf{n} = \begin{bmatrix} \cos \theta_6 \sin \theta_5 \cos \theta_1 \cos(\theta_2 + \theta_3 + \theta_4) + \cos \theta_6 \cos \theta_5 \sin \theta_1 + \sin \theta_6 \cos \theta_1 \sin(\theta_2 + \theta_3 + \theta_4) \\ \cos \theta_6 \sin \theta_5 \sin \theta_1 \cos(\theta_2 + \theta_3 + \theta_4) - \cos \theta_6 \cos \theta_5 \cos \theta_1 + \sin \theta_6 \sin \theta_1 \sin(\theta_2 + \theta_3 + \theta_4) \\ \cos \theta_6 \sin \theta_5 \sin(\theta_2 + \theta_3 + \theta_4) - \sin \theta_6 \cos(\theta_2 + \theta_3 + \theta_4) \end{bmatrix}$$

$$\mathbf{t} = \begin{bmatrix} -\sin \theta_6 \sin \theta_5 \cos \theta_1 \cos(\theta_2 + \theta_3 + \theta_4) - \sin \theta_6 \cos \theta_5 \sin \theta_1 + \cos \theta_6 \cos \theta_1 \sin(\theta_2 + \theta_3 + \theta_4) \\ -\sin \theta_6 \sin \theta_5 \sin \theta_1 \cos(\theta_2 + \theta_3 + \theta_4) + \sin \theta_6 \sin \theta_5 \cos \theta_1 + \cos \theta_6 \sin \theta_1 \sin(\theta_2 + \theta_3 + \theta_4) \\ -\sin \theta_6 \sin \theta_5 \sin(\theta_2 + \theta_3 + \theta_4) - \cos \theta_6 \cos(\theta_2 + \theta_3 + \theta_4) \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} \cos \theta_1 \cos \theta_5 \cos(\theta_2 + \theta_3 + \theta_4) - \sin \theta_1 \sin \theta_5 \\ \sin \theta_1 \cos \theta_5 \cos(\theta_2 + \theta_3 + \theta_4) + \cos \theta_1 \sin \theta_5 \\ \cos \theta_5 \sin(\theta_2 + \theta_3 + \theta_4) \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} [l_2 + l_3 \cos \theta_2 + l_4 \cos(\theta_2 + \theta_3) + (l_5 + l_6 \cos \theta_5) \cos(\theta_2 + \theta_3 + \theta_4)] \cos \theta_1 - l_6 \sin \theta_1 \sin \theta_5 \\ [l_2 + l_3 \cos \theta_2 + l_4 \cos(\theta_2 + \theta_3) + (l_5 + l_6 \cos \theta_5) \cos(\theta_2 + \theta_3 + \theta_4)] \sin \theta_1 + l_6 \cos \theta_1 \sin \theta_5 \\ l_1 + l_3 \sin \theta_2 + l_4 \sin(\theta_2 + \theta_3) + (l_5 + l_6 \cos \theta_5) \sin(\theta_2 + \theta_3 + \theta_4) \end{bmatrix}$$

A.2**Matriz Jacobiana**

A matriz Jacobiana é dado por

$$J = \begin{bmatrix} J_{L1} & J_{L2} & J_{L3} & J_{L4} & J_{L5} & J_{L6} \\ J_{A1} & J_{A2} & J_{A3} & J_{A4} & J_{A5} & J_{A6} \end{bmatrix}$$

onde

$$J_{L1} = \begin{bmatrix} -[l_2 + l_3 \cos \theta_2 + l_4 \cos(\theta_2 + \theta_3) + (l_5 + l_6 \cos \theta_5) \cos(\theta_2 + \theta_3 + \theta_4)] \sin \theta_1 - l_6 \cos \theta_1 \sin \theta_5 \\ [l_2 + l_3 \cos \theta_2 + l_4 \cos(\theta_2 + \theta_3) + (l_5 + l_6 \cos \theta_5) \cos(\theta_2 + \theta_3 + \theta_4)] \cos \theta_1 - l_6 \sin \theta_1 \sin \theta_5 \\ 0 \end{bmatrix}$$

$$J_{L2} = \begin{bmatrix} -[l_3 \sin \theta_2 + l_4 \sin(\theta_2 + \theta_3) + (l_5 + l_6 \cos \theta_5) \sin(\theta_2 + \theta_3 + \theta_4)] \cos \theta_1 \\ -[l_3 \sin \theta_2 + l_4 \sin(\theta_2 + \theta_3) + (l_5 + l_6 \cos \theta_5) \sin(\theta_2 + \theta_3 + \theta_4)] \sin \theta_1 \\ [l_3 \cos \theta_2 + l_4 \cos(\theta_2 + \theta_3) + (l_5 + l_6 \cos \theta_5) \cos(\theta_2 + \theta_3 + \theta_4)] \end{bmatrix}$$

$$J_{L3} = \begin{bmatrix} -[l_4 \sin(\theta_2 + \theta_3) + (l_5 + l_6 \cos \theta_5) \sin(\theta_2 + \theta_3 + \theta_4)] \cos \theta_1 \\ -[l_4 \sin(\theta_2 + \theta_3) + (l_5 + l_6 \cos \theta_5) \sin(\theta_2 + \theta_3 + \theta_4)] \sin \theta_1 \\ [l_4 \cos(\theta_2 + \theta_3) + (l_5 + l_6 \cos \theta_5) \cos(\theta_2 + \theta_3 + \theta_4)] \end{bmatrix}$$

$$J_{L4} = \begin{bmatrix} -(l_5 + l_6 \cos \theta_5) \sin(\theta_2 + \theta_3 + \theta_4) \cos \theta_1 \\ -(l_5 + l_6 \cos \theta_5) \sin(\theta_2 + \theta_3 + \theta_4) \sin \theta_1 \\ (l_5 + l_6 \cos \theta_5) \cos(\theta_2 + \theta_3 + \theta_4) \end{bmatrix}$$

$$J_{L5} = \begin{bmatrix} -l_6 \sin \theta_5 \cos(\theta_2 + \theta_3 + \theta_4) \cos \theta_1 - l_6 \sin \theta_1 \cos \theta_5 \\ -l_6 \sin \theta_5 \cos(\theta_2 + \theta_3 + \theta_4) \sin \theta_1 + l_6 \cos \theta_1 \cos \theta_5 \\ l_6 \sin \theta_5 \sin(\theta_2 + \theta_3 + \theta_4) \end{bmatrix}$$

$$J_{L6} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$J_{A1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$J_{A2} = \begin{bmatrix} \sin \theta_1 \\ -\cos \theta_1 \\ 0 \end{bmatrix}$$

$$J_{A3} = \begin{bmatrix} \sin \theta_1 \\ -\cos \theta_1 \\ 0 \end{bmatrix}$$

$$J_{A4} = \begin{bmatrix} \sin \theta_1 \\ -\cos \theta_1 \\ 0 \end{bmatrix}$$

$$J_{A5} = \begin{bmatrix} -\cos \theta_1 \sin(\theta_2 + \theta_3 + \theta_4) \\ -\sin \theta_1 \sin(\theta_2 + \theta_3 + \theta_4) \\ \cos(\theta_2 + \theta_3 + \theta_4) \end{bmatrix}$$

$$J_{A6} = \begin{bmatrix} \cos \theta_1 \cos(\theta_2 + \theta_3 + \theta_4) \cos \theta_5 - \sin \theta_1 \sin \theta_5 \\ \sin \theta_1 \cos(\theta_2 + \theta_3 + \theta_4) \cos \theta_5 + \cos \theta_1 \sin \theta_5 \\ \cos \theta_5 \sin(\theta_2 + \theta_3 + \theta_4) \end{bmatrix}$$

A.3

Algoritmo Walker-Orin

Em seções anteriores se definiu a equação geral assumida para o robô manipulador em estudo. Vamos simular a sua dinâmica direta assim,

$$\boldsymbol{\tau} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) \quad (\text{A-8})$$

Seja uma função $SUB(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}; \boldsymbol{\tau})$ que calcula $\boldsymbol{\tau}$, utilizando a equação da dinâmica inversa do robô manipulador de n graus de liberdade. Seus

argumentos são os vetores de deslocamento, velocidade e aceleração das juntas. Agora vamos explicar os métodos 1 e 3 do algoritmo de Walker e Orin.

A.3.1 Método 1

Da equação (A-8), nota-se que os torques nas juntas são uma função linear das acelerações das juntas. Portanto, seja \mathbf{b} definido como um vetor “bias”, igual ao torque produzido só pelas acelerações gravitacionais, centrífugas e coriolis, assim

$$\mathbf{b} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) \quad (\text{A-9})$$

Então, as acelerações nas juntas podem ser obtidas computando a equação linear

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} = (\boldsymbol{\tau} - \mathbf{b}) \quad (\text{A-10})$$

O vetor bias \mathbf{b} pode ser facilmente calculado, definindo $\mathbf{q}, \dot{\mathbf{q}}$ do seu estado atual, fazendo $\ddot{\mathbf{q}} = 0$, é usando a função do programa $SUB(\mathbf{q}, \dot{\mathbf{q}}, 0; \mathbf{b})$. O torque computado pela função SUB com esses valores de entrada é igual ao vetor bias \mathbf{b} .

A dificuldade está em calcular a equação (A-10), para encontrar os valores da matriz \mathbf{H} . Neste primeiro método, isto é conseguido através da definição de \mathbf{q} em seu estado atual, fazendo $\ddot{\mathbf{q}} = \mathbf{e}_j$, e chamando a função $SUB(\mathbf{q}, 0, \mathbf{e}_j; \mathbf{h}_j)$, onde \mathbf{e}_j é um vetor $n \times 1$ com j -ésimo elemento igual a 1 e os demais iguais a 0, e \mathbf{h}_j é a j -ésima coluna de \mathbf{H} . Assim, \mathbf{h}_j é o torque nas juntas quando as velocidades são zero, sem efeitos gravitacionais, e as acelerações nas juntas $\ddot{\mathbf{q}}$ é iguais a \mathbf{e}_j .

Uma vez que todos os elementos da matriz \mathbf{H} são determinados, então as acelerações da junta são obtidas solucionando a equação (A-10)

A.3.2 Método 3

Neste método tem-se um procedimento diferente para calcular a matriz de inércia \mathbf{H} . Somente a diagonal superior precisa ser calculada (devido à simetria) na seguinte ordem: $H_{n,n}, H_{n-1,n}, H_{n-2,n}, \dots, H_{1,n}; H_{n-1,n-1}, H_{n-2,n-1}, \dots, H_{1,n-1}$; etc. Para fazer isto, nota-se que quando computamos para a j -ésima coluna da matriz de inércia no método 1, $\ddot{\mathbf{q}}_i = 0$ para $i > j$. Ou seja, todos os seguintes $n - j + 1$ elos aparecem como um único corpo rígido. Deste sistema composto

de elos com a sua massa total, a localização de seu centro de massa e seu momento de inércia seria,

$$\left. \begin{aligned} \mathbf{F}_j &= \mathbf{M}_j \mathbf{v}_j = \mathbf{M}_j (\mathbf{z}_j \times \mathbf{c}_j) = \mathbf{z}_{j-1} \times (\mathbf{M}_j \mathbf{c}_j) \\ \mathbf{N}_j &= \mathbf{E}_j \mathbf{z}_{j-1} \end{aligned} \right\} \text{ Junta } j \text{ rotacional} \quad (\text{A-11})$$

$$\left. \begin{aligned} \mathbf{F}_j &= \mathbf{E}_j \mathbf{z}_{j-1} \\ \mathbf{N}_j &= 0 \end{aligned} \right\} \text{ Junta } j \text{ translacional} \quad (\text{A-12})$$

onde os símbolos anteriormente não definidos são:

\mathbf{M}_j = massa total do elo j até o elo n .

$\dot{\mathbf{v}}_j$ = aceleração linear do centro de massa do sistema composto desde o elo j até o elo n .

\mathbf{c}_j = localização do centro de massa do sistema composto do elo j até o elo n com respeito à origem de coordenadas do elo $j - 1$.

\mathbf{E}_j = matriz do momento de inércia do sistema composto do elo j até o elo n .

Com essas acelerações, \mathbf{F}_i e \mathbf{N}_i são todos zeros para $i < j$. Utilizando isto,

$$\mathbf{f}_i = \mathbf{f}_{i+1} \quad (\text{A-13})$$

$$\mathbf{n}_i = \mathbf{n}_{i+1} + \mathbf{p}_i^* \times \mathbf{f}_{i+1} \quad \text{para } i = 1, \dots, j - 1 \quad (\text{A-14})$$

e

$$\mathbf{f}_j = \mathbf{f}_j \quad (\text{A-15})$$

$$\mathbf{n}_j = \mathbf{N}_j + \mathbf{c}_j + \mathbf{F}_j \quad \text{para } i = 1, \dots, j - 1 \quad (\text{A-16})$$

Assim, inicializando com $i = j$, equação ?? até ??, pode ser usado para obter todos os valores de \mathbf{n}_i e \mathbf{f}_i , para $i \leq j$.

Os componentes da matriz de momento de inércia ao longo da coluna j são então iguais aos torque ou forças geradas na junta. Assim, para $i \leq j$:

$$H_{ij} = \begin{cases} \mathbf{z}_{i-1} \cdot \mathbf{n}_i & \text{Junta } i \text{ rotacional} \\ \mathbf{z}_{i-1} \cdot \mathbf{f}_i & \text{Junta } i \text{ translacional} \end{cases} \quad (\text{A-17})$$

Para computar M , c_j , e E_j , inicia-se no elo n :

$$\mathbf{M}_n = \mathbf{m}_n \quad (\text{A-18})$$

$$\mathbf{c}_n = \mathbf{s}_n + \mathbf{p}_n^* \quad (\text{A-19})$$

$$\mathbf{E}_n = J_n \quad (\text{A-20})$$

onde,

\mathbf{m}_j = massa do elo j .

\mathbf{s}_j = localização do centro de massa do elo j em relação à coordenada do elo j .

\mathbf{J}_j = matriz do momento de inércia do elo j .

Para solucionar \mathbf{M}_j , \mathbf{c}_j e \mathbf{E}_j , pode-se utilizar as seguintes equações recursivas:

$$\mathbf{M}_j = \mathbf{M}_{j+1} + m_j \quad (\text{A-21})$$

$$\mathbf{c}_j = \frac{1}{M_j} [m_j(\mathbf{s}_j + \mathbf{p}_j^*) + M_{j+1}(\mathbf{c}_{j+1} + \mathbf{p}_j^*)] \quad (\text{A-22})$$

$$\begin{aligned} \mathbf{E}_j = & \mathbf{E}_{j+1} + M_{j+1}[(\mathbf{c}_{j+1} + \mathbf{p}_j^* - \mathbf{c}_j) \cdot (\mathbf{c}_{j+1} + \mathbf{p}_j^* - \mathbf{c}_j)\mathbf{I} \\ & - (\mathbf{c}_{j+1} + \mathbf{p}_j^* - \mathbf{c}_j)(\mathbf{c}_{j+1} + \mathbf{p}_j^* - \mathbf{c}_j)^T] \\ & + \mathbf{J}_j + m_j[(\mathbf{s}_j + \mathbf{p}_j^* - \mathbf{c}_j) \cdot (\mathbf{s}_j + \mathbf{p}_j^* - \mathbf{c}_j)\mathbf{I} \\ & - (\mathbf{s}_j + \mathbf{p}_j^* - \mathbf{c}_j)(\mathbf{s}_j + \mathbf{p}_j^* - \mathbf{c}_j)^T] \end{aligned} \quad (\text{A-23})$$

onde \mathbf{I} é a matriz identidade 3×3 . Da equação (A-22), M_j é a soma da massa do elo j e a massa total do sistema composto do elo $j+1$ até n . Da equação (A-23), o centro de massa do sistema composto do elo j até elo n pode ser obtido do conhecimento da localização do centro de massa do elo j , \mathbf{s}_j e a localização do centro de massa do sistema composto do elo $j+1$ até elo n , \mathbf{c}_{j+1} . De uma forma análoga, uma vez que \mathbf{c}_{j+1} , M_{j+1} , e \mathbf{E}_{j+1} foram determinados, usase-se o teorema de eixos paralelos para a matriz de momentos de inércia do sistema composto do elo j até elo n relativo a \mathbf{c}_j . O teorema de eixos paralelos é um método para obter a matriz de momentos de inércia J_o de um corpo rígido em relação a uma origem de coordenadas O arbitrária em termos da matriz de inércia J_G relativa ao centro de massa. Seja r o vetor coordenada de G relativo a O , como se mostra na figura A-1, e M a massa, então:

$$\mathbf{J}_0 = \mathbf{J}_G + M(r \cdot r\mathbf{I} - rr^T) \quad (\text{A-24})$$

Assim, os primeiros dois termos da equação (A-23) são a contribuição em \mathbf{E}_j da composição dos elos $j+1$ até n , e os seguintes dois termos são a contribuição \mathbf{E}_j pelo elo j , como se mostra na figura A-2.

Na equação (A-11) até (A-23), tem sido implicado que todos os vetores são referenciados ao sistema coordenado O , e por isso que eles são difíceis de calcular. Como descrito em [6], para aliviar o custo computacional, todas as propriedades de massa do sistema composto do elo j até o elo n serão

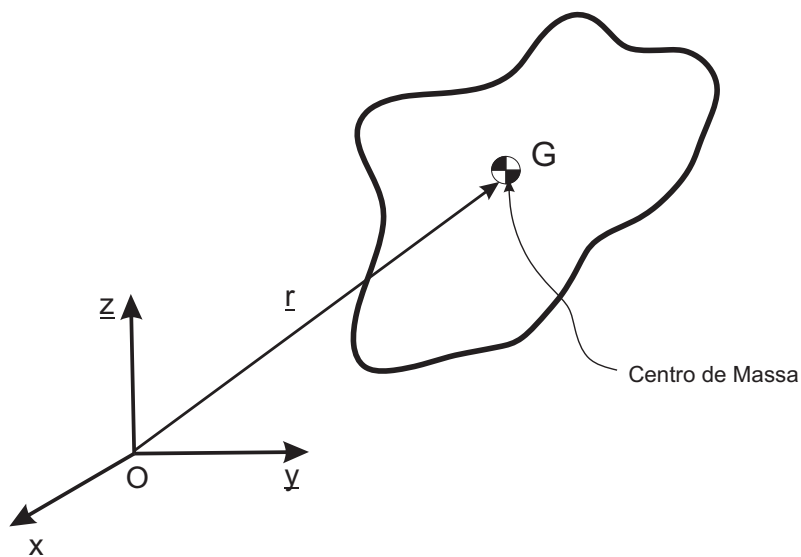


Figura A.1: Posição do centro de massa G com respeito à coordenada O

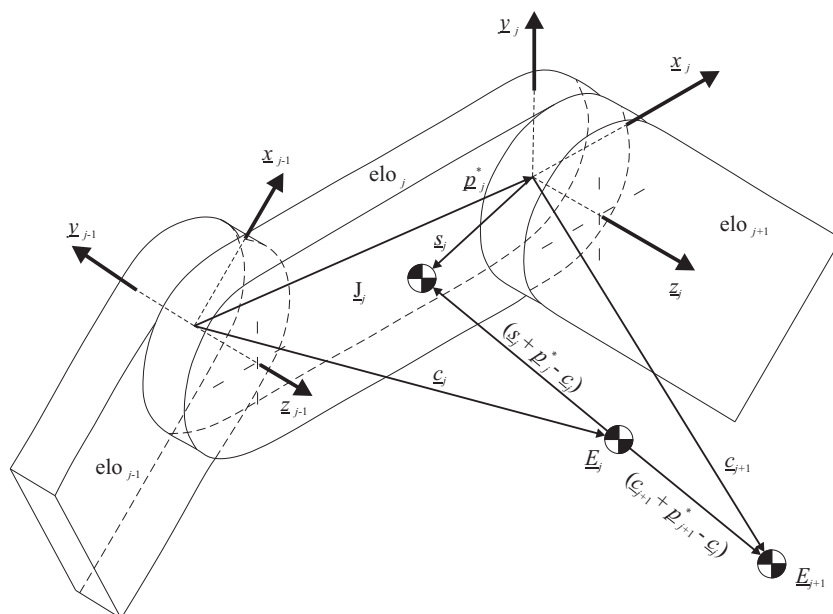


Figura A.2: Centro de massa do sistema composto do elo j até o elo n

referenciados para o sistema coordenado $j - 1$. Assim, a equação usada para computar as propriedades da massa do sistema composto do elo j até elo n , para $j = 1, \dots, n$, torna-se:

$$\mathbf{M}_n = \mathbf{m}\mathbf{J}_n \tag{A-25}$$

$${}^{n-1}\mathbf{c}_n = \mathbf{A}_{n-1}^n ({}^n s_n + \mathbf{p}_n^*) \tag{A-26}$$

$${}^{n-1}\mathbf{E}_n = \mathbf{A}_{n-1}^n \mathbf{E}_n \mathbf{A}_n^{n-1} \tag{A-27}$$

Usando este procedimento para computar a matriz de inércia H e o algoritmo descrito no método 1, encontram-se assim, as acelerações requeridas.

A continuação um resumo dos passos para achar a dinâmica direta aplicando o método 3 do algoritmo Walker-Orin.

Passo 1: Calcule-se M_j , ${}^{j-1}\mathbf{c}_j$ e ${}^{j-1}\mathbf{E}_j$ para $j = N...1$ com

$$M_i = M_{j+1} + m_j \quad (\text{A-28})$$

$${}^{j-1}\mathbf{c}_j = \mathbf{A}_{j-1}^j {}^j\mathbf{c}_j$$

$${}^{j-1}\mathbf{c}_j = \mathbf{A}_{j-1}^j \frac{1}{M_j} [m_j ({}^j\mathbf{s}_j + {}^j\mathbf{p}_j^*) + M_{j+1} ({}^j\mathbf{c}_{j+1} + {}^j\mathbf{p}_j^*)] \quad (\text{A-29})$$

$${}^{j-1}\mathbf{E}_j = \mathbf{A}_{j-1}^j {}^j\mathbf{E}_{j+1} + M_{j+1} [({}^j\mathbf{c}_{j+1} + {}^j\mathbf{p}_j^* - {}^j\mathbf{c}_j) \cdot ({}^j\mathbf{c}_{j+1} + {}^j\mathbf{p}_j^* - {}^j\mathbf{c}_j) \mathbf{I} - ({}^j\mathbf{c}_{j+1} + {}^j\mathbf{p}_j^* - {}^j\mathbf{c}_j) ({}^j\mathbf{c}_{j+1} + {}^j\mathbf{p}_j^* - {}^j\mathbf{c}_j)^T] + {}^j\mathbf{J}_j + m_j [({}^j\mathbf{s}_j + {}^j\mathbf{p}_j^* - {}^j\mathbf{c}_j) \cdot ({}^j\mathbf{s}_j + {}^j\mathbf{p}_j^* - {}^j\mathbf{c}_j) \mathbf{I} - ({}^j\mathbf{s}_j + {}^j\mathbf{p}_j^* - {}^j\mathbf{c}_j) ({}^j\mathbf{s}_j + {}^j\mathbf{p}_j^* - {}^j\mathbf{c}_j)^T] \mathbf{A}_j^{j-1} \quad (\text{A-30})$$

Passo 2: Calcular ${}^{j-1}\mathbf{F}_j$, ${}^{j-1}\mathbf{N}_j$ para $j = 1...N$ utilizando

$${}^{j-1}\mathbf{F}_j = \mathbf{z}_o \times M_j {}^{j-1}\mathbf{c}_j \quad (\text{A-31})$$

$${}^{j-1}\mathbf{N}_j = {}^{j-1}\mathbf{E}_j \mathbf{z}_o \quad (\text{A-32})$$

$$(\text{A-33})$$

Passo 3: Para $j = N...1$ calcule-se ${}^{i-1}\mathbf{f}_i$, ${}^{i-1}\mathbf{n}_i$, $i = j...1$ das equações seguintes para uma particular j ,

$${}^{j-1}\mathbf{f}_j = j_{j-1} \mathbf{F}_j \quad (\text{A-34})$$

$${}^{j-1}\mathbf{n}_j = {}^{j-1}\mathbf{N}_j + {}^{j-1}\mathbf{c}_j \times {}^{j-1}\mathbf{F}_j \quad (\text{A-35})$$

$${}^{i-1}\mathbf{f}_i = \mathbf{A}_{i-1}^i {}^i\mathbf{f}_{i+1} \quad (\text{A-36})$$

$${}^{i-1}\mathbf{n}_i = \mathbf{A}_{i-1}^i ({}^i\mathbf{n}_{i+1} + {}^i\mathbf{p}_i^* \times {}^i\mathbf{f}_{i+1}) \quad (\text{A-37})$$

Utilizando as equações (A-31) até (A-37), seria o procedimento para calcular a matriz H , e com a equação descrito no método 1 acharemos as acelerações para cada valor do torque.

$$\mathbf{H}_{ij} = \begin{cases} z & \text{componente de } {}^{i-1}\mathbf{n}_i & \text{junta } i \text{ rotativa} \\ z & \text{componente de } {}^{i-1}\mathbf{f}_i & \text{junta } i \text{ prismática} \end{cases} \quad (\text{A-38})$$

onde

$$\mathbf{z}_o = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (\text{A-39})$$

B

Apêndice B

Este apêndice contém as propriedades da modelagem dinâmica do robô manipulador descritas no capítulo 3, importante para os diferentes tipos de algoritmos de controle utilizados neste trabalho.

B.1

Modelagem Dinâmica

Seja a equação da modelagem dinâmica do robô manipulador de n graus de liberdade obtida da expressão (3-22), e seja o vetor \mathbf{q} de $n \times 1$ o deslocamento nas juntas, então:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau} \quad (\text{B-1})$$

B.1.1

Propriedade 1

A matriz de inércia é positiva-definida e simétrica, tal que:

$$\lambda_h \mathbf{I}_n \leq \mathbf{H}(\mathbf{q}) \leq \lambda_H \mathbf{I}_n \quad (\text{B-2})$$

onde λ_h e λ_H são constantes positivas.

B.1.2

Propriedade 2

A matriz $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ com dimensão $n \times n$ satisfaz:

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \leq c_o \dot{\mathbf{q}} \quad (\text{B-3})$$

limitada por alguma constante c_o

B.1.3

Propriedade 3

O vetor força/torque gravitacional satisfaz:

$$\mathbf{G}(\mathbf{q}) \leq g_o \quad (\text{B-4})$$

limitada por alguma constante g_o .

B.1.4**Propriedade 4**

Da propriedade (B-2), lembre-se que a matriz $\mathbf{H}(\mathbf{q})$ é simétrica ($H_{ij} = H_{ji}$) e seja $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ a matriz das forças de Coriolis e Centrífugas. Então, representando para o i -ésimo elemento $\mathbf{C}\dot{\mathbf{q}}$ como

$$\sum_{j=1}^n C_{ij} \dot{q}_j = \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n \frac{\partial H_{ij}}{\partial q_k} \dot{q}_j \dot{q}_k + \frac{1}{2} \sum_{k=1}^n \sum_{j=1}^n \left(\frac{\partial H_{ij}}{\partial q_k} - \frac{\partial H_{jk}}{\partial q_i} \right) \dot{q}_k \dot{q}_j$$

e simplificando a expressão obtém-se:

$$C_{ij} = \frac{1}{2} \sum_{k=1}^n \frac{\partial H_{ij}}{\partial q_k} \dot{q}_k + \frac{1}{2} \sum_{k=1}^n \left(\frac{\partial H_{ik}}{\partial q_j} - \frac{\partial H_{jk}}{\partial q_i} \right) \dot{q}_k$$

$$C_{ij} = \frac{1}{2} \dot{H}_{ij} + \frac{1}{2} \sum_{k=1}^n \left(\frac{\partial H_{ik}}{\partial q_j} - \frac{\partial H_{jk}}{\partial q_i} \right) \dot{q}_k$$

Seja $\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{H}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$, então

$$N_{ij} = \sum_{k=1}^n \left(\frac{\partial H_{jk}}{\partial q_i} - \frac{\partial H_{ik}}{\partial q_j} \right) \dot{q}_k$$

Da expressão anterior, obtemos que:

$$N_{ij} = -N_{ji}$$

Isto significa que a matriz $\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}})$ é uma matriz anti-simétrica, e assim

$$\begin{aligned} \dot{\mathbf{q}}^T \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} &\equiv 0 \\ \dot{\mathbf{q}}^T (\dot{\mathbf{H}} - 2\mathbf{C}) \dot{\mathbf{q}} &\equiv 0 \end{aligned} \quad (\text{B-5})$$

para qualquer vetor $\dot{\mathbf{q}}$ de dimensões $n \times 1$

B.1.5**Propriedade 5**

Um balanço de energia para o robô manipulador de juntas rígidas é dado por:

$$E_{\text{manipulador}} = E_{\text{cinética}} + E_{\text{potencial}}$$

onde

$$\begin{aligned} E_{cinética} &= \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{H} \dot{\mathbf{q}} \\ E_{potencial} &= \mathbf{U}(\mathbf{q}) \end{aligned}$$

então

$$E_{manipulador} = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{H} \dot{\mathbf{q}} + \mathbf{U}(\mathbf{q})$$

Derivando a expressão da energia,

$$\begin{aligned} \frac{d}{dt}(E_{manipulador}) &= \dot{\mathbf{q}}^T \mathbf{H} \ddot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{H}} \dot{\mathbf{q}} + \frac{\partial \mathbf{U}(\mathbf{q})}{\partial t} \frac{\partial q}{\partial q} \\ \frac{d}{dt}(E_{manipulador}) &= \dot{\mathbf{q}}^T \mathbf{H} \ddot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{H}} \dot{\mathbf{q}} + \dot{\mathbf{q}}^T \mathbf{G} \end{aligned}$$

Substituindo a equação dinâmica do manipulador, obtemos

$$\begin{aligned} \frac{d}{dt}(E_{manipulador}) &= \dot{\mathbf{q}}^T \boldsymbol{\tau} - \dot{\mathbf{q}}^T \mathbf{C} \dot{\mathbf{q}} - \dot{\mathbf{q}}^T \mathbf{G} + \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{H}} \dot{\mathbf{q}} + \dot{\mathbf{q}}^T \mathbf{G} \\ \frac{d}{dt}(E_{manipulador}) &= \dot{\mathbf{q}}^T \boldsymbol{\tau} + \dot{\mathbf{q}}^T \left(\frac{1}{2} \dot{\mathbf{H}} - \mathbf{C} \right) \dot{\mathbf{q}} \end{aligned}$$

Então, utilizando a propriedade (B-5), finalmente obtemos

$$\frac{d}{dt}(E_{manipulador}) = \dot{\mathbf{q}}^T \boldsymbol{\tau} \quad (\text{B-6})$$

Se observa que os termos de Coriolis e das forças centrífugas não contribuem para a variação temporal da energia total. Além disso, a potência necessária para o movimento do robô manipulador é aquela potência transmitida pelo atuador de cada junta. As forças de Coriolis e Centrífuga não contribuem na potência do sistema.

C

Apêndice C

C.1

Instalação do CHAI3D em MVS2008

Para experimentar os exemplos demo do CHAI3D no MVS2008 fazemos o seguinte:

- Execute o Visual Studio 2008 e abra um Projeto Solution assim, no menu File -> Open -> Project Solution... ou tecla Ctrl+Shift+O como se mostra na figura [C.1](#).

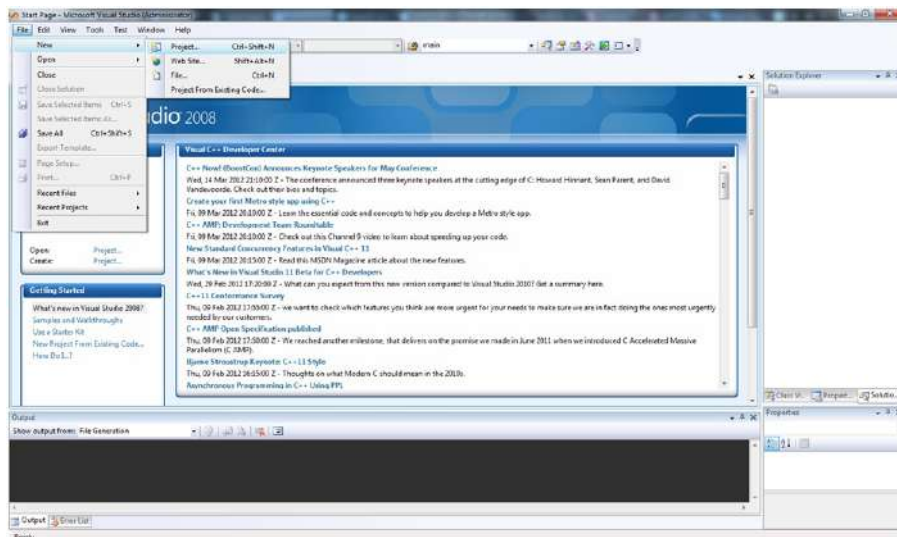


Figura C.1: Microsoft Visual Studio 2008

- Aparecerá uma nova janela para abrir o arquivo com extensão .sln, Esse arquivo se encontra na pasta de exemplos msvc9 do CHAI3D com o nome CHAI3D.sln. Logo compilamos o programa teclando Ctrl+F7 e todos os exemplo demo ficaram pronto para sua experimentação como mostra a figura [C.2](#) e [C.3](#).

Os seguintes passos descrevem a configuração um novo projeto utilizando as bibliotecas que o CHAI3D utiliza. Isto serve para poder programar um próprio código para desenvolver o entorno virtual com a aplicação descrita para este trabalho.

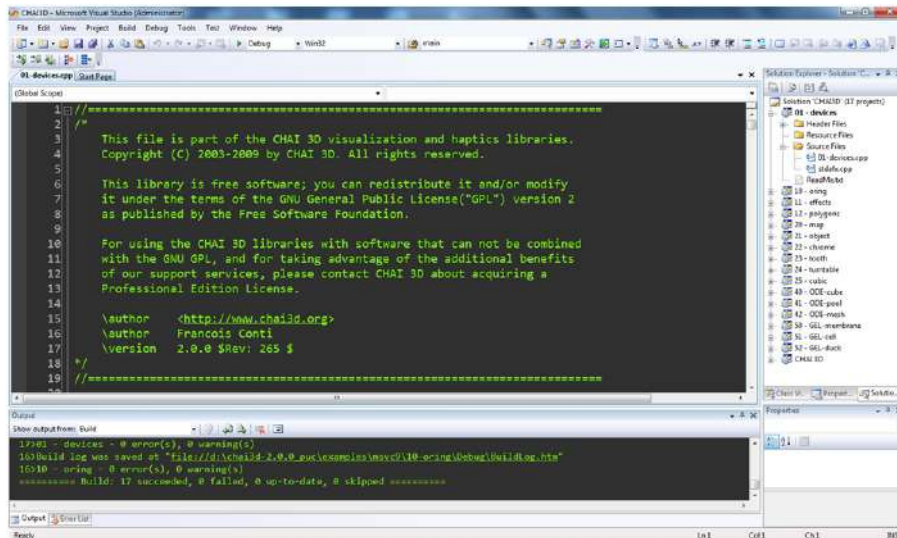


Figura C.2: Compilação dos exemplos demo do CHAI3D

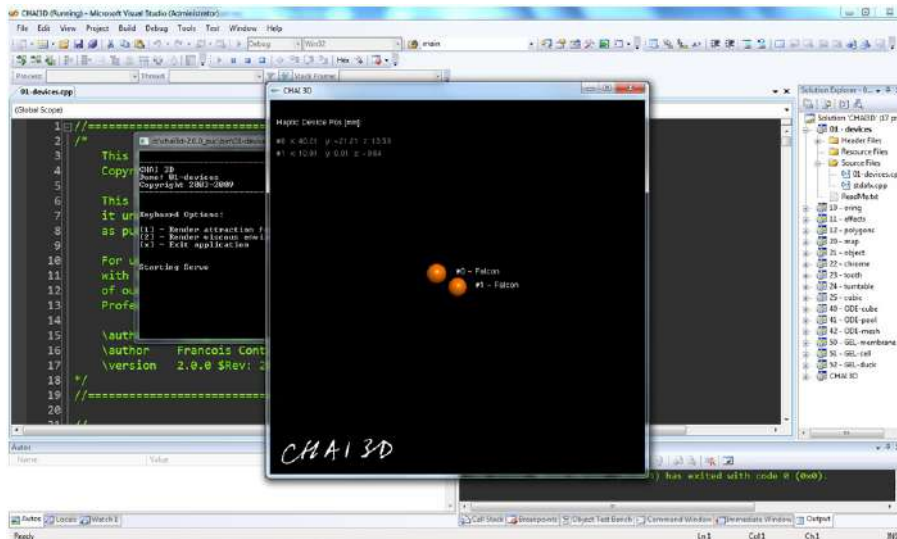


Figura C.3: Exemplo 01-devices do CHAI3D

1. Execute o Visual Studio 2008 e crie um novo projeto no menu File- >New- >Project ou tecla Ctrl+Shift+N. Guarde o nome projeto na pasta de exemplos msvc9 do CHAI3D. Para a aplicação utilizamos *Win32 Console Application* e com um *empty project*, como mostramos na figura [C.4](#).
2. Logo no MSV2008 na parte do menu Project- >Project Properties ou teclando Alt+F7 abrimos uma janela onde configuraremos os arquivos para utilizar o CHAI3D.
3. Em Configuration Properties- >C/C++- >General- >Additional Include Directories escrevemos D: chai3d-2.0.0 src e D: chai3d-

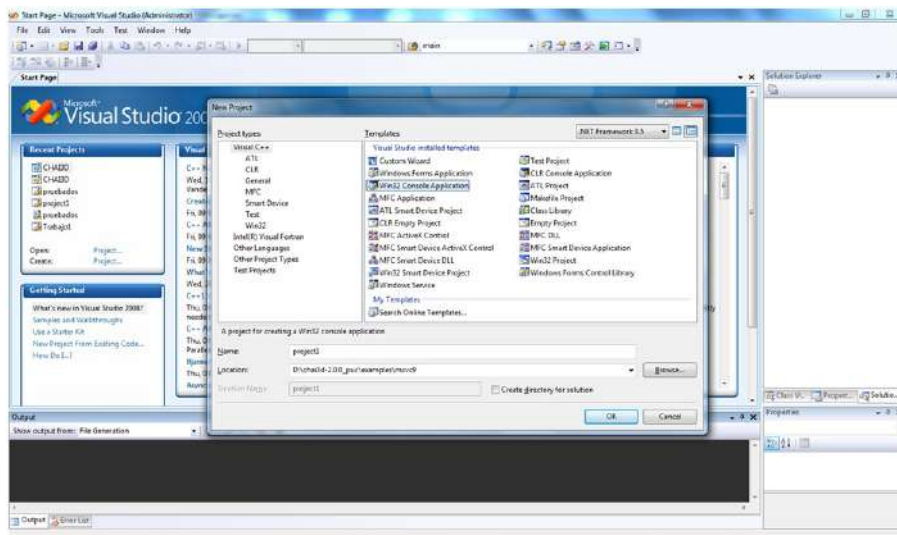


Figura C.4: Criando novo projeto no MVS2008

2.0.0 external OpenGL msvc D: chai3d-2.0.0 external ODE include ode
D: chai3d-2.0.0 modules ODE.

4. Em Configuration Properties -> C/C++ -> Preprocessor -> Preprocessor Definitions escrevemos:

```
WIN32;$_DEBUG;$_CONSOLE;$_MSVC;WINVER$=0X0500$;dDOUBLE
```

como mostra a figura [C.5](#).

5. Em Configuration Properties -> Linker -> General -> Additional Include Directories escrevemos D: chai3d-2.0.0 external ODE lib msvc
D: chai3d-2.0.0 external OpenGL msvc D: chai3d-2.0.0 lib msvc9
6. Em Configuration Properties -> Linker -> General -> Output File escrevemos D: chai3d-2.0.0 bin project1.exe, figura [C.6](#).

- 7 Em Configuration Properties -> Linker -> Input -> Additional Dependencies, escrevemos:

```
winmm.lib opengl32.lib chai3d-debug.lib  
glu32.lib odbc32.lib odbccp32.lib ode_double.lib
```

Finalmente compilamos nosso primeiro projeto, mostrado na figura [C.7](#).

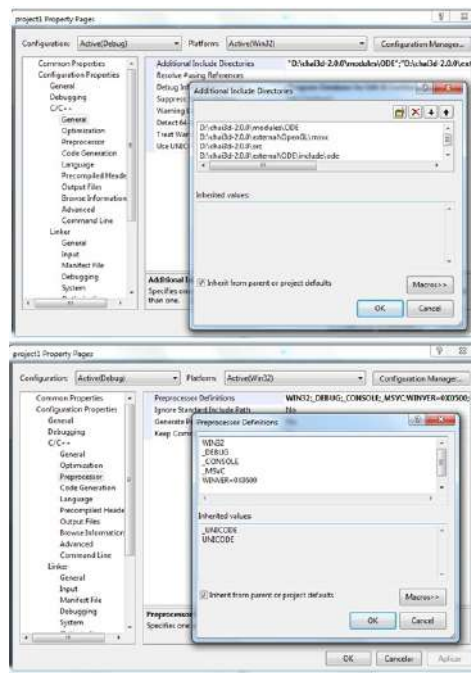


Figura C.5: Enlace de arquivos cabeçalhos .h (Passos 2, 3 e 4)

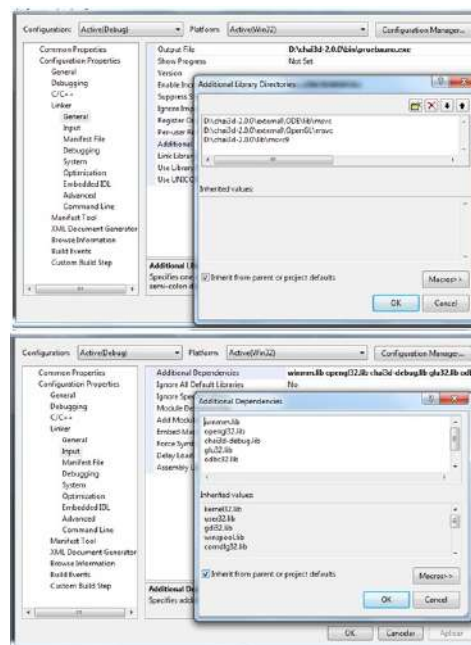


Figura C.6: Enlace de arquivos .lib (Passos 5 e 6)

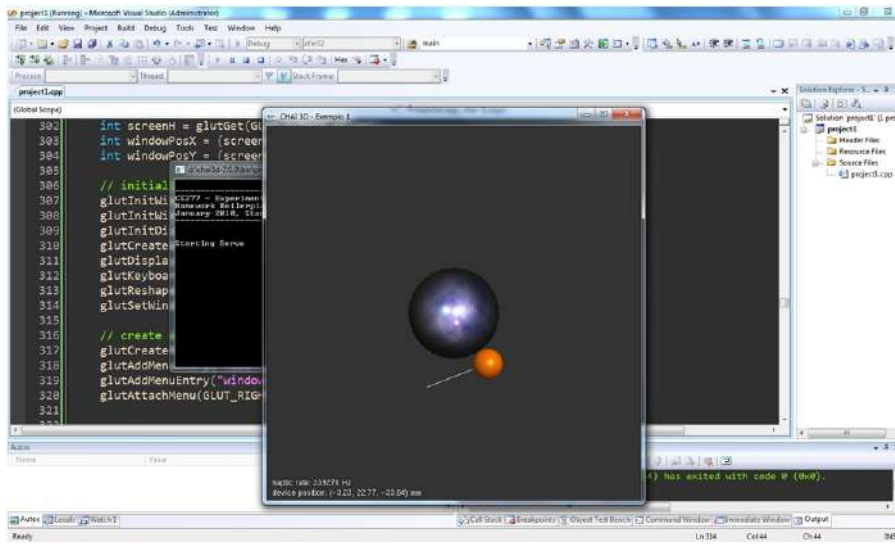


Figura C.7: Compilação de um projeto com Chai3D