



## MÉTODO DE CONTROLE E DETECÇÃO DE OBSTÁCULOS PARA ROBÔS MANIPULADORES APLICADO À INTERAÇÃO HUMANO-ROBÔ

Thiago Braga de Almeida Antonio

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientadores: Fernando Cesar Lizarralde  
Liu Hsu

Rio de Janeiro  
Setembro de 2014

MÉTODO DE CONTROLE E DETECÇÃO DE OBSTÁCULOS PARA ROBÔS  
MANIPULADORES APLICADO À INTERAÇÃO HUMANO-ROBÔ

Thiago Braga de Almeida Antonio

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO  
ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE  
ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE  
JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A  
OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA  
ELÉTRICA.

Examinada por:

---

Prof. Fernando Cesar Lizarralde, D.Sc.

---

Prof. Liu Hsu, Dr. d'État

---

Prof. Bruno Vilhena Adorno, Ph.D.

---

Dr. Antonio Candea Leite, D.Sc.

RIO DE JANEIRO, RJ – BRASIL  
SETEMBRO DE 2014

Antonio, Thiago Braga de Almeida

Método de Controle e Detecção de Obstáculos para Robôs Manipuladores Aplicado à Interação Humano-Robô/Thiago Braga de Almeida Antonio. – Rio de Janeiro: UFRJ/COPPE, 2014.

XII, 99 p.: il.; 29, 7cm.

Orientadores: Fernando Cesar Lizarralde

Liu Hsu

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2014.

Referências Bibliográficas: p. 88 – 94.

1. Robótica Controle. 2. Desvio de Obstáculos. 3. Campos Potenciais Artificiais. 4. Interação Humano-Robô. 5. Detecção de Obstáculos. I. Lizarralde, Fernando Cesar *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

# Agradecimentos

Primeiramente gostaria de agradecer a minha família que me apoiou por toda a minha vida e que continuará me apoiando no futuro.

Agraço também aos meus orientadores Fernando Cesar Lizarralde e Liu Hsu pelo apoio e pela paciência durante a pesquisa. Aos membros da banca Antonio Candea Leite e Bruno Vilhena Adorno, ofereço um obrigado pelas críticas e elogios, e também pela paciência em ler este trabalho.

Agradecimento especial aos amigos da graduação e do mestrado, pelos momentos de diversão e estudo, pois sem eles não teria conseguido chegar aonde cheguei. Menciono especialmente o Gabriel Casulari, meu companheiro de trabalho no LABCON e do projeto da PETROBRAS, e meus companheiros de trabalho do LEAD, Fernando Coutinho, Maurício Dias e Silva e Ighor Marcovistz.

Agradeço ao CNPq, a FAPERJ e a CAPES pelo suporte financeiro, através de bolsas e apoios aos projetos desenvolvidos no laboratório, que permitiram a realização deste trabalho.

E finalmente, agradeço aos demais membros do LABCON e do LEAD que contribuíram para a realização deste trabalho.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

## MÉTODO DE CONTROLE E DETECÇÃO DE OBSTÁCULOS PARA ROBÔS MANIPULADORES APLICADO À INTERAÇÃO HUMANO-ROBÔ

Thiago Braga de Almeida Antonio

Setembro/2014

Orientadores: Fernando Cesar Lizarralde  
Liu Hsu

Programa: Engenharia Elétrica

Este trabalho apresenta um método de controle cinemático e detecção de obstáculos para robôs manipuladores redundantes operando em ambientes frequentados por humanos. O algoritmo de controle baseia-se no conceito de campos potenciais artificiais aproveitando a característica de redundância do robô para desviar de obstáculos e atingir o objetivo de controle simultaneamente. O algoritmo de detecção de obstáculos utiliza um sensor de profundidade baseado na tecnologia de luz estruturada para obter uma descrição 2 1/2 D do ambiente a partir de uma nuvem de pontos. Para evitar que o algoritmo de detecção considere o robô como um obstáculo, um método é apresentado para eliminar o robô do cenário obtido pelo sensor. Campos repulsivos são gerados em torno dos obstáculos detectados permitindo que o robô realize a tarefa de interesse sem colisões. Simulações e resultados experimentais, obtidos com um robô Motoman DIA10 e um dispositivo de interface natural Microsoft Kinect<sup>TM</sup>, são apresentados para ilustrar a viabilidade do método proposto.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

CONTROL METHOD AND OBSTACLE DETECTION FOR ROBOT  
MANIPULATORS APPLIED TO HUMAN-ROBOT INTERACTION

Thiago Braga de Almeida Antonio

September/2014

Advisors: Fernando Cesar Lizarralde

Liu Hsu

Department: Electrical Engineering

This work presents a kinematic control and collision detection method for redundant robot manipulators operating in human-populated environments. The control algorithm is based on the concept of artificial potential fields taking advantage of the robot redundancy for the purpose of obstacle avoidance and control goal achievement. The obstacle detection algorithm uses a depth sensor based on the structured light to obtain a 2 1/2 D description of the surroundings from a point cloud. In order to avoid that the detection algorithm considers the robot as an obstacle, a method is presented to remove the robot from the scenario obtained by the sensor. Repulsive fields are created around the detected obstacles, allowing for the robot to perform the task of interest without collisions. Simulations and experimental results, obtained from a Motoman DIA10 robot and a natural user interface Microsoft Kinect<sup>TM</sup>, illustrate the feasibility of the proposed scheme.

# Sumário

<b>Lista de Figuras</b>	<b>ix</b>
<b>Lista de Tabelas</b>	<b>xii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Estado da Arte . . . . .	2
1.2 Objetivo . . . . .	3
1.3 Metodologia . . . . .	5
1.4 Formulação do Problema . . . . .	5
1.5 Organização da Dissertação . . . . .	6
<b>2 Modelagem Cinemática</b>	<b>8</b>
2.1 Cinemática . . . . .	9
2.1.1 Cinemática de um Manipulador . . . . .	9
2.1.2 Posição e Orientação de um Corpo Rígido . . . . .	10
2.2 Cinemática Direta . . . . .	12
2.2.1 Cinemática Direta para Manipuladores . . . . .	13
2.2.2 Espaço das Juntas e Espaço Operacional . . . . .	15
2.3 Cinemática Inversa . . . . .	18
2.4 Cinemática Diferencial . . . . .	20
2.4.1 Jacobiano Geométrico . . . . .	20
2.4.2 Singularidades Cinemáticas . . . . .	23
2.4.3 Jacobiano Analítico . . . . .	24
2.4.4 Jacobiano da Representação . . . . .	25
2.5 Conclusões . . . . .	25
<b>3 Controle Cinemático e Desvio de Obstáculos</b>	<b>26</b>
3.1 Controle Cinemático . . . . .	26
3.1.1 Restrições das Juntas . . . . .	29
3.1.2 Controle de Posição . . . . .	30
3.1.3 Controle de Orientação . . . . .	31
3.1.4 Controle Cinemático de Manipuladores Redundantes . . . . .	33

3.2	Desvio de Obstáculos . . . . .	36
3.2.1	Desvio de Obstáculos para as Elos . . . . .	37
3.2.2	Distância entre Elos e Obstáculos . . . . .	38
3.2.3	Desvio de Obstáculos para o Efetuador . . . . .	39
3.2.4	Mudança da Direção do Vetor Repulsivo . . . . .	40
3.2.5	Comparação entre as Forças Repulsivas . . . . .	42
3.3	Resultados de Simulação . . . . .	43
3.3.1	Simulação I . . . . .	44
3.3.2	Simulação II . . . . .	45
3.3.3	Simulação III . . . . .	48
3.3.4	Simulação IV . . . . .	50
3.3.5	Simulação V . . . . .	52
3.3.6	Simulação VI . . . . .	54
3.4	Conclusões . . . . .	56
<b>4</b>	<b>Detecção de Obstáculos</b>	<b>58</b>
4.1	Calibração Remota . . . . .	58
4.1.1	Simulação da Calibração Remota . . . . .	63
4.2	Obtendo a Descrição do Ambiente . . . . .	63
4.3	Filtragem da Nuvem de Pontos . . . . .	65
4.3.1	Eliminando o Robô da Nuvem de Pontos . . . . .	65
4.3.2	Comparação entre os Métodos de Filtragem . . . . .	71
4.3.3	Eliminando os pontos fora do alcance do robô . . . . .	74
4.4	Resultados Experimentais . . . . .	75
4.5	Conclusões . . . . .	83
<b>5</b>	<b>Considerações Finais e Discussões</b>	<b>85</b>
5.1	Conclusões . . . . .	85
5.2	Propostas para Trabalhos Futuros . . . . .	87
	<b>Referências Bibliográficas</b>	<b>88</b>
	<b>A Geometria Vetorial</b>	<b>95</b>
	<b>B Quatérnio Unitário</b>	<b>97</b>
	<b>C Propriedades das Elipses</b>	<b>99</b>

# Lista de Figuras

1.1	Exemplos da Interação Humano-Robô. . . . .	4
1.2	Formulação do problema. . . . .	6
2.1	Posição e Orientação de um Corpo Rígido. . . . .	10
2.2	Sistemas de coordenadas inercial $\bar{E}_0$ e do corpo $\bar{E}_1$ . . . . .	11
2.3	Descrição dos Elos e Juntas de um Manipulador. . . . .	13
2.4	Transformação de Coordenadas de uma Cadeia Cinemática aberta. . . . .	14
2.5	Cálculo do espaço de trabalho de uma robô planar de três elos. . . . .	17
3.1	Diagrama em blocos da malha de controle de velocidade. . . . .	27
3.2	Representação Gráfica de $H_i(\theta_i)$ . . . . .	30
3.3	Magnitude da força repulsiva $f_{L,O}$ . . . . .	38
3.4	Magnitude da função repulsiva para $b$ variando de 1 a 10. . . . .	39
3.5	Força repulsiva e sua variação. . . . .	41
3.6	Comparação entre as forças repulsivas. . . . .	42
3.7	Manipulador redundante e obstáculos simulados no Matlab. . . . .	43
3.8	Simulação I- Evolução no tempo da posição do efetuador. . . . .	44
3.9	Simulação I - Erro de posição. . . . .	44
3.10	Simulação I - Sinal de Controle Cartesiano. . . . .	45
3.11	Simulação I - Sinal de Controle nas juntas. . . . .	45
3.12	Simulação I - Menor Distância aos obstáculos. . . . .	45
3.13	Simulação II - Evolução no tempo da posição do efetuador. . . . .	46
3.14	Simulação II - Erro de posição e norma do erro de orientação. . . . .	46
3.15	Simulação II - Sinal de Controle Cartesiano. . . . .	47
3.16	Simulação II - Sinal de Controle nas juntas. . . . .	47
3.17	Simulação II - Menor Distância aos obstáculos. . . . .	48
3.18	Simulação III - Erro de posição. . . . .	48
3.19	Simulação III - Sinal de Controle Cartesiano. . . . .	49
3.20	Simulação III - Sinal de Controle nas juntas. . . . .	49
3.21	Simulação III - Rastreamento de Trajetória do Efetuador. . . . .	49
3.22	Simulação III - Menor Distância aos obstáculos. . . . .	50

3.23	Simulação IV - Erro de posição e Norma do Erro de Orientação . . . . .	50
3.24	Simulação IV - Sinal de Controle Cartesiano . . . . .	51
3.25	Simulação IV - Sinal de Controle nas Juntas. . . . .	51
3.26	Simulação IV - Rastreamento de Trajetória do Efetuador. . . . .	51
3.27	Simulação IV - Menor Distância aos obstáculos. . . . .	52
3.28	Simulação V - Erro de posição. . . . .	52
3.29	Simulação V - Sinal de Controle Cartesiano de posição. . . . .	53
3.30	Simulação V - Sinal de Controle nas Juntas. . . . .	53
3.31	Simulação V - Rastreamento de Trajetória do Efetuador. . . . .	53
3.32	Simulação V - Menor Distância aos obstáculos. . . . .	54
3.33	Simulação VI - Erro de posição e norma do erro de orientação. . . . .	55
3.34	Simulação VI - Sinal de Controle Cartesiano de posição e orientação. . . . .	55
3.35	Simulação VI - Sinal de Controle nas Juntas. . . . .	55
3.36	Simulação VI - Rastreamento de Trajetória do Efetuador. . . . .	56
3.37	Simulação VI - Menor Distância aos obstáculos. . . . .	56
4.1	Sistemas de coordenadas no Método de Calibração Remota. . . . .	62
4.2	Valores estimados para a posição e a orientação . . . . .	64
4.3	Obtenção da descrição do ambiente. . . . .	65
4.4	Método de remoção do robô baseado no uso de esferas. . . . .	66
4.5	Método de remoção do robô baseado no uso de cilindros. . . . .	67
4.6	Método de remoção do robô baseado no uso de elipsoides. . . . .	70
4.7	Esquematização do método das esferas. . . . .	72
4.8	Esquematização do método dos elipsoides. . . . .	73
4.9	Esquematização do método dos cilindros. . . . .	73
4.10	Filtragem dos pontos da nuvem. . . . .	74
4.11	Ambiente Experimental. . . . .	76
4.12	Sequência de imagens do robô - Regulação. . . . .	76
4.13	Trajetória do Efetuador - Regulação. . . . .	77
4.14	Erro de posição e norma do erro de orientação - Regulação. . . . .	77
4.15	Sinal de Controle Cartesiano - Regulação. . . . .	78
4.16	Sinal de Controle nas juntas - Regulação. . . . .	78
4.17	Sequência de imagens do robô - Movimento ponto-a-ponto. . . . .	79
4.18	Trajetória do Efetuador - Movimento ponto-a-ponto. . . . .	80
4.19	Erro de posição e norma do error de orientação - Movimento ponto-a-ponto. . . . .	80
4.20	Sinal de Controle Cartesiano - Movimento ponto-a-ponto. . . . .	81
4.21	Sinal de Controle nas juntas - Movimento ponto-a-ponto. . . . .	81
4.22	Trajetória do Efetuador - Rastreamento. . . . .	82

4.23	Erro de posição e norma do erro de orientação - Rastreamento. . . . .	82
4.24	Sinal de Controle Cartesiano - Rastreamento. . . . .	83
4.25	Sinal de Controle nas juntas - Rastreamento. . . . .	83
B.1	Rotação de um ângulo $\vartheta$ ao longo de um eixo arbitrário $r$ . . . . .	97

# Lista de Tabelas

3.1	Descrição das simulações realizadas. . . . .	43
-----	--	----

# Capítulo 1

## Introdução

Atualmente, um dos principais desafios da robótica é lidar com a interação entre humanos e robôs, possibilitando que eles trabalhem juntos de forma natural, efetiva e segura (Haddadin et al. 2008, Santis et al. 2008). Para buscar essa interação o ambiente ao redor do robô deve ser pelo menos parcialmente conhecido, garantindo que o mesmo se movimente sem causar danos aos operadores próximos a ele. Portanto, o robô deve ser capaz de evitar colisões com obstáculos localizados no seu espaço de trabalho e alcançar os objetivos de controle simultaneamente.

Nesse contexto, existem diversas aplicações onde robôs realizam tarefas previamente programadas ou são tele-operados, interagindo com seres humanos e inclusive com outros robôs como, por exemplo: (i) busca e resgate (Luo et al. 2011); (ii) educação (Chin et al. 2011); (iii) doméstica e de apoio a idosos e deficientes (Lohse et al. 2008); (iv) social, para entretenimento, companhia e guias em exposições (Sadazuka et al. 2007); (v) serviços, na área de agricultura e exploração (Kashiwazaki et al. 2010); (vi) médica e terapia assistida (Cleary 2005).

Nos últimos anos, a interação humano-robô (IHR) vem se tornando um tema de pesquisa relevante na área de robótica. O objetivo básico é criar e investigar interfaces que habilitam modos de interação eficazes e naturais com as tecnologias robóticas atuais. IHR é um tema altamente interdisciplinar e multidisciplinar, que reúne metodologias e técnicas de robótica, inteligência artificial, interação humano-computador, ciência cognitiva, fatores humanos, ciências sociais e muitas outras (Arkin et al. 2003, Dautenhahn 2007, Fong et al. 2003, Steinfeld et al. 2006).

Dentre as diversas abordagens desenvolvidas para auxiliar na interação entre humanos e robôs, destaca-se a manipulação interativa que pode ser aplicada para tarefas de cooperação, colaboração e coordenação. Na cooperação, o operador e o robô operam em conjunto (co-operam), enquanto trabalham em objetivos particulares ainda comuns. Por outro lado, na colaboração, o operador e o robô trabalham em conjunto (co-trabalham) em um único objetivo compartilhado, existindo uma hierarquia mestre-escravo. Enquanto isso, na coordenação, o operador e o robô tra-

balham em conjunto (co-trabalham) em um único objetivo compartilhado, mas a hierarquia mestre-escravo não está presente.

Um exemplo de manipulação cooperativa onde um operador segura um copo de plástico e o robô rastreia a mão do operador servindo-lhe água é apresentada em (Adorno, Bó, Fraisse & Poignet 2011). Um exemplo de manipulação colaborativa é baseada no conceito de manipulação simultânea usando movimentos espelhados onde o operador controla o robô e simultaneamente interage com o mesmo por meio de um objeto manipulado em comum (Adorno, Bó, Fraisse & Poignet 2011, Takubo et al. 2002). Um exemplo de manipulação coordenativa é apresentada em (Adorno, Bó & Fraisse 2011, Green et al. 2008), onde o sistema de controle do robô comanda não apenas o seu braço mas também o braço do operador, para manter a coordenação dos movimentos e melhorar a execução da tarefa de interesse.

## 1.1 Estado da Arte

Na área de robótica diversos métodos de planejamento de movimento e desvio de obstáculos foram propostos ao longo dos anos com o intuito de gerar um caminho ou trajetória livre de colisões entre dois pontos de interesse (LaValle 2011). Entretanto, como a proposta do presente trabalho é promover a interação entre robôs e seres humanos em um ambiente dinâmico e imprevisível, o uso de técnicas *off-line* que consideram os modelos do robô e do cenário perfeitamente conhecidos não é recomendado. Por outro lado, técnicas *off-line* de planejamentos de rotas, como o método RRT (do inglês, *Rapidly-exploring Random Tree*), se projetadas de maneira cuidadosa, podem ser utilizadas de maneira *on-line* na presença de obstáculos dinâmicos e incertezas paramétricas (Bernardes et al. 2014). Outra alternativa é utilizar técnicas *on-line* ou reativas capazes de lidar com a presença de objetos desconhecidos e móveis em ambientes incertos (Minguez et al. 2008).

Uma abordagem de desvio de obstáculos é baseada no conceito de campos potenciais artificiais, onde campos virtuais atrativos e repulsivos são criados no espaço de trabalho do robô, e dessa forma o efetuador pode mover-se na direção de um ponto desejado, enquanto evita colidir com obstáculos do ambiente (Chung et al. 1997, Khatib 1986). Outra abordagem consiste em utilizar campos artificiais circulares, onde o efetuador contorna o campo criado pelos obstáculos ao invés de apenas ser repellido como ocorre com o campo potencial repulsivo (Singh et al. 1996). Campos virtuais circulares e potenciais também podem ser combinados a fim de aproveitar as vantagens de ambas as abordagens (Haddadin et al. 2011). Nesses trabalhos, os autores assumem que o movimento dos obstáculos é conhecido a priori ignorando a necessidade de sensoriamento externo.

Para lidar com essa limitação, métodos de detecção de obstáculos e para obten-

ção de uma descrição geométrica do ambiente foram analisados. Em (Flacco et al. 2012) um método de detecção de obstáculos baseado em dados obtidos a partir de um dispositivo Microsoft Kinect<sup>TM</sup> é proposto para evitar colisões em tarefas de interação entre humanos e robôs. O Microsoft Kinect<sup>TM</sup>, ou apenas Kinect, é um sensor de movimento desenvolvido pela Microsoft para o vídeo game Xbox 360. Ele é composto principalmente por uma câmera RGB e um sensor de profundidade, possuindo também um microfone embutido, um processador e *software* próprios. Em (Rakprayoon et al. 2011) é apresentado um método de calibração para o Kinect bem como um método para distinguir entre o manipulador e os obstáculos, quando eles compartilham o mesmo espaço de trabalho. Neste contexto, alguns trabalhos dedicados ao problema de navegação de robôs móveis em ambientes internos e externos foram desenvolvidos usando o Kinect para detecção de obstáculos ao invés de uma abordagem baseada em *laser scanner* 3D ou câmeras *stereo* (Correa et al. 2012, Greuter et al. 2011), visando a utilização de sensores de menor custo. Ainda no âmbito da interação entre humanos e robôs, alguns autores têm utilizado o Kinect para reconhecer diferentes gestos corporais e gerar uma interface visual de interação humano-robô ou auxiliar na execução de tarefas de colaboração cujas etapas são iniciadas e finalizadas por gestos e comandos de voz (Cheng et al. 2012, De Luca & Flacco 2012, Motta-Ribeiro et al. 2012). A Figura 1.1 apresenta uma ilustração da ideologia de interação socialmente aceita<sup>1</sup> entre humanos e robôs e alguns exemplos de interação humano-robô usando um dispositivo de interface natural (Breazeal et al. 2008, Sisbot et al. 2010).

## 1.2 Objetivo

Na presente dissertação, apresenta-se um método de controle cinemático e detecção de obstáculos para manipuladores redundantes operando em ambientes frequentados por humanos (Almeida-Antonio et al. 2014). O algoritmo de controle baseia-se no conceito de campos potenciais artificiais e utiliza a pseudo-inversa da matriz Jacobiana com um fator de ponderação para os limites mecânicos das juntas, aproveitando a característica de redundância do robô para desviar de obstáculos. Um sensor de profundidade baseado na tecnologia de luz estruturada é utilizado para obter uma descrição 2 1/2 D do ambiente a partir de uma nuvem de pontos e detectar os obstáculos presentes no espaço de trabalho do robô. Para evitar que o algoritmo de detecção considere os elos e as juntas do robô como obstáculos, um método baseado no uso de figuras geométricas é apresentado para eliminar o robô do

---

<sup>1</sup>Interação socialmente aceita: O robô tem que se comportar de forma segura, confortável e natural, de tal forma que não cause medo, surpresa ou desconforto ao operador (Sisbot et al. 2010).

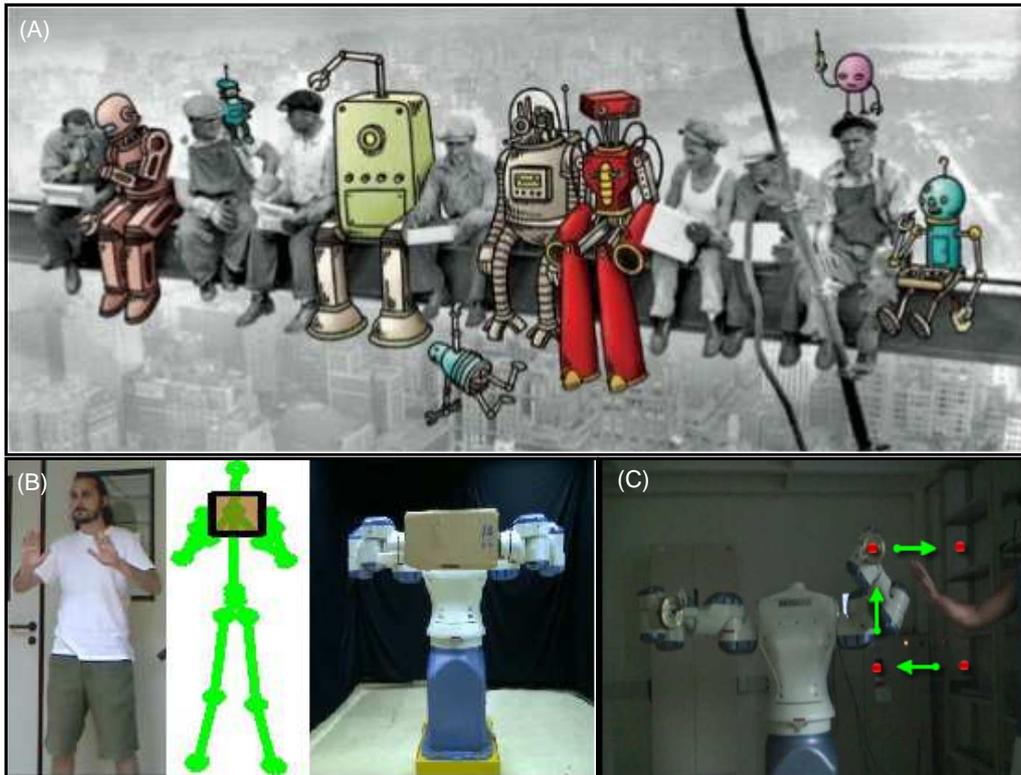


Figura 1.1: Exemplos da Interação Humano-Robô: (a) Ilustração de uma interação socialmente aceita entre humanos e robôs, (b) Tarefa de manipulação colaborativa e (c) detecção de obstáculos durante uma tarefa de rastreamento de trajetórias, ambas usando o Kinect.

cenário obtido pelo sensor. Campos repulsivos são gerados ao redor dos obstáculos detectados, permitindo que o robô realize a tarefa de interesse sem colisões e garantindo a segurança dos operadores. Simulações e resultados experimentais, obtidos a partir de um robô Motoman DIA10 e um dispositivo de interface natural (DIN) Microsoft Kinect<sup>TM</sup>, ilustram a viabilidade do método proposto.

A contribuição deste trabalho é combinar um método de controle cinemático com uma metodologia de detecção e desvio de obstáculos para buscar a interação entre humanos e robôs de forma segura e eficiente. Um método de calibração remota baseado no uso de sensores internos e externos é utilizada para estimar os parâmetros extrínsecos do DIN e evitar a necessidade de calibração *in-situ*. A análise de estabilidade e convergência do sistema completo é desenvolvida a partir da Teoria de Estabilidade de Lyapunov. Para avaliar a interação humano-robô, a abordagem proposta é implementada em um sistema robótico real realizando tarefas de regulação e rastreamento de trajetórias na presença de operadores humanos.

## 1.3 Metodologia

Para implementar os algoritmos de controle, de detecção e desvio de obstáculos apresentados nesta dissertação, a seguinte metodologia de trabalho foi estabelecida:

- Primeiramente foi realizada uma revisão bibliográfica sobre os métodos de controle e desvios de obstáculos, para compreender suas vantagens e desvantagens (Almeida-Antonio et al. 2012, Siciliano et al. 2009). O método de controle cinemático de posição e orientação, baseado na pseudo-inversa da matriz Jacobiana com um termo adicional de prioridade secundária, foi selecionado devido ao seu comprovado desempenho e eficiência.
- A partir da revisão bibliográfica, um algoritmo de desvio de obstáculos baseado no uso de campos potenciais artificiais repulsivos foi selecionado e aplicado para (i) desviar de objetos próximos às juntas, usando o método de projeção no espaço nulo do Jacobiano, e (ii) desviar de objetos próximos ao efetuador, inserindo um campo repulsivo na lei de controle cartesiana de posição.
- O algoritmo de controle cinemático e desvio de obstáculos foi simulado numericamente para avaliar a viabilidade de sua implementação. Em seguida essa metodologia foi implementada em um robô manipulador real na presença de obstáculos simulados, fixos e móveis.
- Uma revisão bibliográfica sobre metodologias de detecção de obstáculos estáticos e dinâmicos baseada no uso de dispositivos de interface natural foi realizada e alguns trabalhos foram selecionados (Flacco et al. 2012, Rakprayoon et al. 2011). Uma vez que o algoritmo de detecção pode considerar os elementos do robô como obstáculos, um método de delimitação do robô - baseado no uso de esferas, cilindros e elipsoides - é proposto para remover o robô do cenário obtido pelo sensor.
- Finalmente, a viabilidade da metodologia proposta foi verificada por meio de simulações e testes experimentais com um manipulador real na presença de obstáculos reais e seres humanos.

## 1.4 Formulação do Problema

Neste trabalho, considera-se que um robô manipulador redundante deve executar uma determinada tarefa em ambientes frequentados por humanos, na presença de pessoas trabalhando e movendo-se livremente dentro do espaço de trabalho do robô. A tarefa de interesse consiste em mover o efetuador de um ponto inicial  $P_0$  até o

ponto final  $P_f$ , seguindo uma trajetória de referência  $r(t)$ , enquanto mantém uma orientação desejada invariante no tempo (vide Figura 1.2). Assume-se aqui que a tarefa é previamente planejada para evitar singularidades e que a descrição do ambiente de trabalho é parcialmente conhecida.

Para adquirir uma representação geométrica 2 1/2 D do ambiente, um dispositivo de interface natural (DIN) equipado com um sensor de profundidade é posicionado em frente ao robô garantindo que o mesmo esteja dentro do seu campo de visão. Dessa forma, objetos fixos e móveis (incluindo pessoas) podem ser identificados no formato de uma nuvem de pontos e o robô pode realizar a tarefa de maneira segura sem colidir com esses obstáculos.

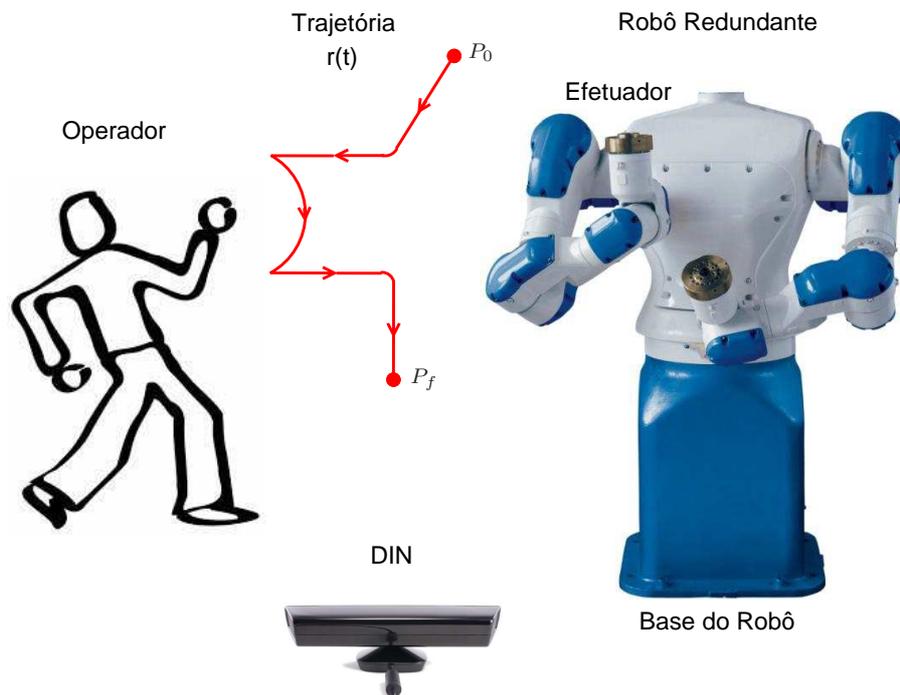


Figura 1.2: Robô evitando colisões com um operador e um DIN posicionado para obter a representação geométrica 2 1/2 D do ambiente.

Como a maioria dos robôs industriais possui uma malha interna de controle de velocidade para acionamento direto das juntas, uma lei de controle puramente cinemática baseada em comandos de velocidade pode ser implementada para garantir o desempenho satisfatório do manipulador durante a execução da tarefa. Na Figura 1.2 pode-se observar uma representação do manipulador realizando uma tarefa na presença e um operador.

## 1.5 Organização da Dissertação

Esta dissertação está organizada da seguinte forma:

- **Capítulo 2:** São apresentados alguns conceitos e definições relacionados à cinemática direta, cinemática inversa, cinemática diferencial e singularidades de um robô manipulador;
- **Capítulo 3:** A abordagem de controle cinemático de posição e orientação baseada na pseudo-inversa da matriz Jacobiana com uma restrição adicional de prioridade secundária é apresentada. Os conceitos e definições relacionados ao método de desvio de obstáculos baseado no uso de campos potenciais artificiais são apresentados e discutidos;
- **Capítulo 4:** É apresentado o método de detecção de obstáculos a partir de uma nuvem de pontos obtidas pelo sensor de profundidade juntamente com um método de delimitação do robô baseado no uso de figuras geométricas para remover o robô do cenário obtido pelo sensor. A descrição do ambiente experimental é apresentada juntamente com os resultados experimentais obtidos com um robô Motoman DIA10 e um dispositivo de interface natural Microsoft Kinect<sup>TM</sup>.
- **Capítulo 5:** São apresentadas as conclusões da dissertação e as propostas para desenvolvimento de trabalhos futuros.

## Capítulo 2

# Modelagem Cinemática

Na maioria das aplicações da robótica, completar uma tarefa genérica requer a execução de um movimento específico prescrito para o efetuador de um manipulador. O movimento pode ser *livre* se não existe interação física entre o efetuador e o ambiente, ou *restrito* se existem forças de contato entre ambos. A execução correta desse movimento é atribuída ao sistema de controle que fornece aos atuadores das juntas comandos consistentes para alcançar um ponto desejado (regulação) ou seguir uma trajetória desejada (rastreamento).

O controle de movimento do efetuador requer uma análise precisa das características da estrutura mecânica, dos atuadores e dos sensores. A finalidade desta análise é obter os modelos matemáticos descrevendo as relações de entrada/saída que caracterizam os componentes do manipulador. A modelagem de um manipulador é, portanto, uma premissa necessária para encontrar estratégias de controle de movimento adequadas para a estrutura do manipulador (Siciliano et al. 2009).

A modelagem de manipuladores pode ser dividida em modelagem *cinemática* e *dinâmica*. A cinemática de um robô estabelece a relação geométrica entre o movimento do robô no espaço das juntas e o movimento do efetuador no espaço da tarefa. O modelo cinemático de um manipulador é útil para determinar as relações entre os torques e as forças aplicados às juntas bem como os momentos e as forças aplicados no efetuador em uma configuração de equilíbrio estático. Por outro lado, a dinâmica do robô descreve a relação entre as forças generalizadas atuando sobre o robô e o movimento da estrutura governado por acelerações, velocidades e posições das juntas. O modelo dinâmico é útil para o desenvolvimento do projeto mecânico da estrutura, escolha dos atuadores, determinação de estratégias de controle e simulação do movimento do manipulador.

Como a maioria dos robôs industriais possui uma malha interna de controle de velocidade para acionamento direto das juntas, uma lei de controle puramente cinemática baseada em comandos de velocidade pode ser implementada para garantir o desempenho satisfatório do manipulador durante a execução da tarefa. Dessa

forma, neste trabalho a dinâmica do manipulador não será abordada e apenas a cinemática do manipulador será considerada.

## 2.1 Cinemática

A análise cinemática da estrutura mecânica de um manipulador é baseada na descrição de movimentos com respeito a um sistema de coordenadas fixo, ignorando os momentos e as forças que causam o movimento da estrutura.

Neste contexto, é importante distinguir entre cinemática e cinemática diferencial. Em referência a um robô manipulador, a *cinemática* descreve a relação analítica entre as posições das juntas e a posição e orientação do efetuador. A *cinemática diferencial* descreve a relação analítica entre o movimento das juntas e o movimento do efetuador em termos das velocidades, por meio de matrizes Jacobianas.

A elaboração da relação cinemática permite o estudo de dois problemas relevantes da robótica, ou seja, o problema da *cinemática direta* e o problema da *cinemática inversa*. Enquanto o primeiro diz respeito à determinação de um método geral e sistemático para descrever o movimento do efetuador em função do movimento das juntas por meio de ferramentas de álgebra linear, o segundo refere-se ao problema inverso e sua solução é de fundamental importância para transformar o movimento desejado do efetuador no espaço de trabalho em correspondente movimento das juntas.

### 2.1.1 Cinemática de um Manipulador

Um manipulador pode ser esquematicamente representado, de um ponto de vista mecânico, como uma cadeia cinemática de corpos rígidos (elos) conectados por meio de juntas. Em uma extremidade da cadeia encontra-se a base do robô, enquanto que na outra extremidade encontra-se o efetuador. O movimento resultante da estrutura é obtido por meio da composição dos movimentos de cada um dos elos com respeito ao elo anterior.

Então, para poder manipular ou interagir com um objeto no espaço é necessário descrever a posição e orientação do efetuador com respeito a um sistema de coordenadas de referência. Para realizar essa descrição de um modo padronizado, algumas definições baseadas em geometria vetorial são apresentadas no Apêndice A. Como resultado, estabelece-se a seguinte nomenclatura:

- $\bar{E}_a = [\vec{x}_a \ \vec{y}_a \ \vec{z}_a]$  denota o sistema de coordenadas ortonormal  $a$  e  $\vec{x}_a, \vec{y}_a, \vec{z}_a$  denotam os vetores unitários dos eixos de coordenadas.
- Considere um vetor  $\vec{v} = v_1 \vec{x}_a + v_2 \vec{y}_a + v_3 \vec{z}_a$  representado no sistema de

coordenadas  $\bar{E}_a$ . As coordenadas de  $\vec{v}$  expressas em  $\bar{E}_a$  são definidas como  $v = [v_1 \ v_2 \ v_3]^T$ .

### 2.1.2 Posição e Orientação de um Corpo Rígido

Um movimento rígido de um objeto é um movimento que preserva a distância entre os seus pontos. Este movimento é definido em termos de uma transformação de corpo rígido que preserva o produto escalar e o produto vetorial entre os referidos pontos. O estudo da cinemática e do controle de sistemas robóticos está intrinsecamente associado ao movimento dos corpos rígidos (Murray et al. 1994).

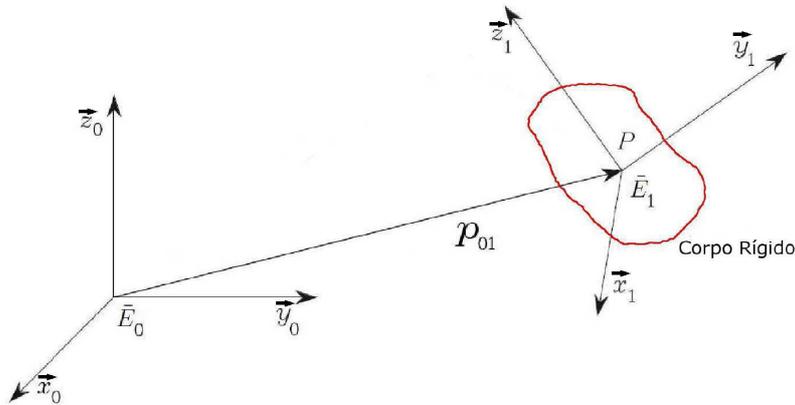


Figura 2.1: Posição e Orientação de um Corpo Rígido.

A localização de um corpo rígido no espaço é completamente descrita pela sua posição e orientação com respeito a um sistema de coordenadas de referência. Considere um corpo rígido e um sistema de coordenadas ortonormal de referência  $\bar{E}_0 = [\vec{x}_0 \ \vec{y}_0 \ \vec{z}_0]$  onde  $\vec{x}_0, \vec{y}_0, \vec{z}_0$  são os seus eixos unitários (Figura 2.1). A posição do ponto  $P$  fixado sobre o corpo rígido com respeito ao sistema de coordenadas  $\bar{E}_0$  é expressa pela relação:

$$p_{01} = p_x \vec{x}_0 + p_y \vec{y}_0 + p_z \vec{z}_0, \quad (2.1)$$

onde  $p_x, p_y, p_z$  são as coordenadas do vetor  $p_{01}$ , expressas no sistema de coordenadas  $\bar{E}_0$ , definidas por:

$$p_{01} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}. \quad (2.2)$$

Para descrever a orientação do corpo rígido é conveniente considerar um sistema de coordenadas ortonormal fixado ao corpo e expressar seus eixos unitários com respeito ao sistema de coordenadas de referência (Figura 2.1). Considere dois sistemas

de coordenadas  $\bar{E}_0 = [\vec{x}_0 \ \vec{y}_0 \ \vec{z}_0]$  e  $\bar{E}_1 = [\vec{x}_1 \ \vec{y}_1 \ \vec{z}_1]$  representando o sistema de coordenadas inercial e o sistema de coordenadas do corpo, respectivamente (Figura 2.2). Então, as coordenadas dos vetores  $\vec{x}_1, \vec{y}_1, \vec{z}_1$  são expressas no sistema de coordenadas  $\bar{E}_0$  usando as seguintes relações:

$$x_{01} = \bar{E}_0^* \vec{x}_1, \quad y_{01} = \bar{E}_0^* \vec{y}_1, \quad z_{01} = \bar{E}_0^* \vec{z}_1, \quad (2.3)$$

onde  $\bar{E}_0^* = [\vec{x}_0 \cdot \vec{y}_0 \cdot \vec{z}_0]^\top$  é a adjunta de  $\bar{E}_0$  como definido no Apêndice A.

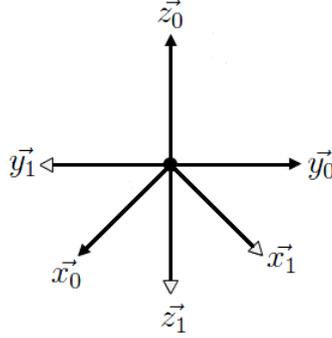


Figura 2.2: Sistemas de coordenadas inercial  $\bar{E}_0$  e do corpo  $\bar{E}_1$ .

A partir dessas equações, pode-se escrever  $\bar{E}_1$  como

$$\bar{E}_1 = \left[ \bar{E}_0 x_{01} \quad \bar{E}_0 y_{01} \quad \bar{E}_0 z_{01} \right] = \bar{E}_0 \overbrace{\left[ x_{01} \quad y_{01} \quad z_{01} \right]}^{R_{01}} = \bar{E}_0 R_{01}, \quad (2.4)$$

onde  $R_{01} \in SO(3)$  é denominada matriz de rotação, orientação ou atitude, e é definida por

$$R_{01} = \bar{E}_0^* \bar{E}_1, \quad (2.5)$$

onde  $SO(3)$  é denominado *Grupo Especial Ortonormal de dimensão 3*, definido por

$$SO(3) = \{R \in \mathbb{R}^{3 \times 3} : R^T R = I, \det(R) = 1\}. \quad (2.6)$$

A partir de (2.2) e (2.5), pode-se representar a posição e orientação de um corpo rígido por meio de uma matriz de transformação homogênea (Siciliano et al. 2009). Então, uma forma compacta para representar a configuração do sistema de coordenadas  $\bar{E}_1$  com respeito ao sistema de coordenadas  $\bar{E}_0$ , é dada por

$$T_{01}(\theta) = \begin{bmatrix} R_{01} & (p_{01})_0 \\ 0_{1 \times 3} & 1 \end{bmatrix}, \quad (2.7)$$

onde  $(p_{01})_0$  e  $R_{01}$  são a posição e a orientação do sistema de coordenadas  $\bar{E}_1$  em relação ao sistema de coordenadas  $\bar{E}_0$  respectivamente.

Note que a matriz de rotação  $R_{01}$  fornece uma descrição redundante da orientação do sistema de coordenadas e é caracterizada por nove elementos que não são independentes, mas relacionados por seis restrições devido a condição de ortogonalidade definida  $R_{01}^T R_{01} = I$ . Isso implica que três parâmetros são suficientes para descrever a orientação do corpo rígido no espaço. A representação da orientação em termo de três parâmetros independentes constitui uma *representação mínima*.

Uma representação não-mínima da orientação pode ser obtida recorrendo-se a uma representação com quatro parâmetros como, por exemplo, o quatérnio unitário (Apêndice B), expressando a rotação de um determinado ângulo em torno de um eixo do espaço. Neste contexto, a orientação do sistema de coordenadas  $\bar{E}_1$  em relação ao sistema de coordenadas  $\bar{E}_0$  pode ser expressa por:

$$q_{01} = ((q_s)_{01}, (q_v)_{01}), \quad (2.8)$$

onde  $q_{01} \in \mathbb{H}^1$  é o quatérnio unitário associado a matriz de rotação  $R_{01}$ ,  $(q_s)_{01} \in \mathbb{R}$  é a parte escalar e  $(q_v)_{01} \in \mathbb{R}^3$  é a parte vetorial, sujeitas a restrição de norma unitária  $(q_s)_{01}^2 + (q_v)_{01}^T (q_v)_{01} = 1$  (Siciliano et al. 2009).

## 2.2 Cinemática Direta

Como dito anteriormente, um manipulador consiste de uma série de corpos rígidos (elos) conectados por meios de juntas, onde as mais comuns são do tipo revolução e prismática. A estrutura completa forma uma cadeia cinemática: em uma extremidade da cadeia encontra-se uma base, enquanto que na outra encontra-se um efetuador, permitindo a manipulação de objetos no espaço (Figura 2.3).

De um ponto de vista topológico, a cadeia cinemática pode ser considerada de dois tipos: *aberta*, quando existe apenas uma sequência de elos conectando a base e o efetuador; *fechada*, quando um laço é formado e, dessa forma, mais de uma sequência de elos conecta a base e o efetuador. Nesta seção, considera-se que os manipuladores de interesse possuem cadeia cinemática aberta e, portanto, manipuladores compostos por cadeias cinemáticas *fechadas* não serão abordados neste trabalho.

A estrutura mecânica de um manipulador é caracterizada pelo número de graus de mobilidade, que determinam unicamente a sua configuração. Cada grau de mobilidade é tipicamente associado com a articulação da junta e constitui uma *variável de junta*, podendo ser um ângulo ou um deslocamento, dependendo do tipo de junta. Então, o objetivo da cinemática direta é computar a *posição e orientação* do efetuador em função das suas variáveis de juntas com respeito a um sistema de coordenadas

---

<sup>1</sup>O símbolo  $\mathbb{H}$  denota o grupo de quatérnio unitário satisfazendo a álgebra de quatérnio (Wen & Kreutz-Delgado 1991).

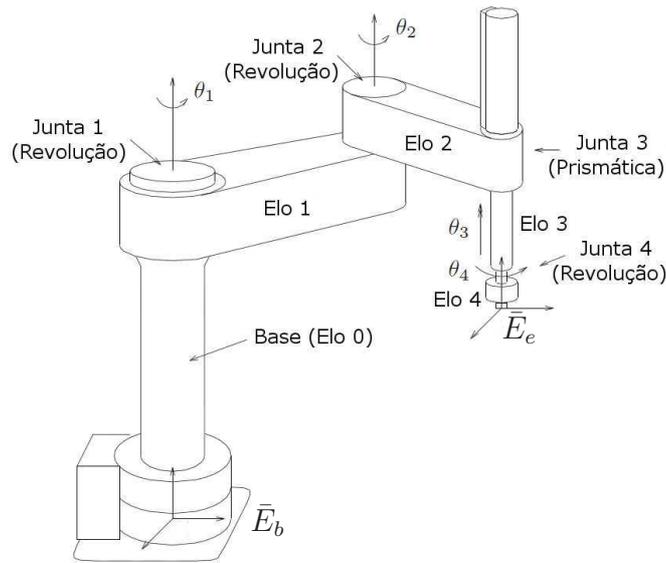


Figura 2.3: Descrição dos Elos e Juntas de um Manipulador.

de referência (Siciliano et al. 2009).

É válido mencionar que, como visto na seção anterior, a posição de um corpo rígido com respeito a um sistema de coordenadas de referência é descrita por meio do vetor posição de um ponto de interesse localizado sobre o corpo rígido, enquanto que a orientação é descrita pela matriz de rotação de um sistema de coordenadas fixado nesse ponto, com respeito ao mesmo sistema de coordenadas de referência.

### 2.2.1 Cinemática Direta para Manipuladores

A primeira abordagem usada para calcular a cinemática direta de um manipulador é analisar a geometria da estrutura do sistema robótico, a partir de um conhecimento básico de trigonometria. Porém, para os casos em que a estrutura do manipulador é complexa e o número de juntas aumenta, é preferível adotar uma solução menos direta e baseada em um procedimento sistemático, por exemplo, a convenção de *Denavit-Hartenberg* (Siciliano et al. 2009), a Teoria dos Helicóides (Murray et al. 1994) ou a abordagem de Quatérnio-Dual (Figueredo et al. 2013).

Então, considera-se um manipulador de cadeia *aberta* constituído de  $n+1$  elos conectados por  $n$  juntas, onde o elo 0 é convencionalmente fixado na base. Além disso, assume-se que cada junta contribui com um grau de mobilidade para a estrutura mecânica, correspondendo a uma variável de junta.

A construção de um procedimento operacional para a computação da cinemática direta é naturalmente derivada a partir da análise de uma típica cadeia cinemática *aberta* da estrutura de um manipulador. De fato, desde que cada junta conecte apenas dois elos consecutivos, é razoável considerar primeiro a relação cinemática entre

elos consecutivos e então obter uma descrição geral da cinemática do manipulador de um modo recursivo.

Dessa forma, é importante definir um sistema de coordenadas fixado a cada elo  $k$ , a partir do elo 0 até o elo  $n$ . Então, a transformação de coordenadas descrevendo a posição e orientação do sistema de coordenadas  $\bar{E}_k$  com respeito ao sistema de coordenadas  $\bar{E}_0$  (Figura 2.4) é dada por:

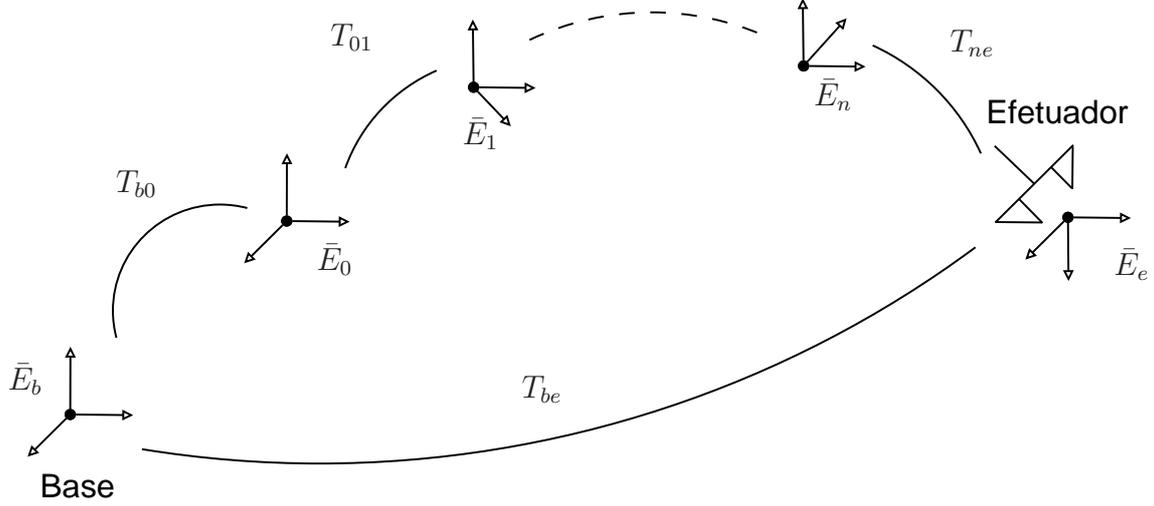


Figura 2.4: Transformação de Coordenadas de uma Cadeia Cinemática aberta.

$$T_{0k}(\theta) = T_{01}(\theta_1) \cdot \dots \cdot T_{k-1k}(\theta_k), \quad k = 1, \dots, n. \quad (2.9)$$

onde  $\theta = \begin{bmatrix} \theta_1 & \dots & \theta_k \end{bmatrix}^T$  é o vetor das variáveis das juntas.

Note que a cinemática direta é calculada de modo recursivo e é obtida de uma forma sistemática através da simples multiplicação das matrizes de transformação homogênea (Siciliano et al. 2009), onde cada matriz é uma função de uma única variável de junta. Neste contexto, a transformação homogênea descrevendo a posição e orientação do sistema de coordenadas do efetuador  $\bar{E}_e$  com respeito ao sistema de coordenadas da base  $\bar{E}_b$  pode ser obtido por meio de

$$T_{be}(\theta) = T_{b0} \cdot T_{0n}(\theta) \cdot T_{ne}, \quad (2.10)$$

onde  $T_{b0}$  e  $T_{ne}$  são (tipicamente) transformações homogêneas constantes descrevendo a posição e orientação do sistema de coordenadas  $\bar{E}_0$  com respeito ao sistema de coordenadas da base  $\bar{E}_b$  e do sistema de coordenadas do efetuador  $\bar{E}_e$  com respeito ao sistema de coordenadas  $\bar{E}_n$ , respectivamente.

Portanto, com respeito ao sistema de coordenadas da base  $\bar{E}_b$ , a posição e orientação do sistema de coordenadas do efetuador  $\bar{E}_e$  são expressas em uma forma mais

compacta, por meio da matriz de transformação homogênea dada por:

$$T_{be}(\theta) = \begin{bmatrix} R_{be} & (p_{be})_b \\ 0_{1 \times 3} & 1 \end{bmatrix}, \quad (2.11)$$

onde  $(p_{be})_b$  e  $R_{be}$  denotam a posição e orientação do sistema de coordenadas  $\bar{E}_e$  em relação ao sistema de coordenadas  $\bar{E}_b$  respectivamente.

## 2.2.2 Espaço das Juntas e Espaço Operacional

Como descrito na seção anterior, a equação da cinemática direta de um manipulador permite que a posição e orientação do sistema de coordenadas do efetuador possa ser expressa como uma função das variáveis das juntas com respeito a um sistema de coordenadas de referência (fixo), por exemplo, o sistema de coordenadas da base. Contudo, quando uma tarefa é designada ao efetuador, é necessário especificar a sua posição e orientação, eventualmente como uma função do tempo (trajetória). Para o caso da posição, isso é bem simples, mas especificar a orientação por meio de seus vetores unitários é um procedimento difícil, pois as componentes dos vetores devem satisfazer às restrições impostas pela condição de *ortonormalidade* (2.6) a cada instante de tempo.

O problema de descrever a orientação do efetuador admite uma solução natural se uma *representação mínima* ou outra parametrização adequada for adotada, por exemplo, as representações não-mínimas como os ângulos de Euler ou o quaternio unitário (Siciliano et al. 2009). Nesse caso, uma trajetória de movimento pode ser atribuída ao conjunto de ângulos escolhidos para representar orientação. Portanto, a posição pode ser dada por um número mínimo de coordenadas, com respeito à geometria da estrutura, enquanto que a orientação pode ser especificada em termos de uma representação conveniente descrevendo a rotação do sistema de coordenadas do efetuador com respeito ao sistema de coordenadas da base.

Deste modo, é possível descrever a posição e orientação do efetuador por meio de um vetor ( $m \times 1$ )

$$x = \begin{bmatrix} p \\ \phi \end{bmatrix} \in \mathbb{R}^m, \quad (2.12)$$

onde  $p$  descreve a posição do efetuador e  $\phi$  uma representação mínima da sua orientação. Essa representação de posição e orientação permite uma descrição da tarefa do efetuador em termos de uma série de parâmetros independentes. O vetor  $x$  é definido no espaço em que a tarefa do manipulador foi especificada e, por isso, esse espaço é chamado de *espaço operacional* e possui dimensão  $m$ . Por outro lado, o

*espaço das juntas* representa o espaço em que o vetor  $(n \times 1)$  das variáveis das juntas

$$\theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}, \quad (2.13)$$

é definido, e  $n$  é a sua dimensão. Considerando a dependência da posição e da orientação em termos das variáveis das juntas, a equação da cinemática direta pode ser escrita como

$$x = k(\theta). \quad (2.14)$$

A função vetorial  $k(\cdot)$ , de dimensão  $(m \times 1)$ , é em geral não-linear e permite a determinação das variáveis do espaço operacional a partir do conhecimento das variáveis do espaço das juntas.

### **Espaço de Trabalho**

O espaço de trabalho representa a região do ambiente de operação que o efetuador do robô pode acessar. Sua forma e volume dependem da estrutura do manipulador bem como da existência de limites mecânicos das juntas. Em referência ao espaço operacional, a *dimensão do espaço de trabalho* pode ser considerada como um índice de desempenho do sistema robótico. Esta é a região descrita pela origem do sistema de coordenadas do efetuador quando todas as juntas do manipulador executam todos os possíveis movimentos (Murray et al. 1994, Siciliano et al. 2009).

Normalmente deve-se distinguir entre *espaço de trabalho alcançável* e *espaço de trabalho destro*. O primeiro é a região em que a origem do sistema de coordenadas do efetuador pode alcançar com pelo menos uma orientação, enquanto o último é a região em que a origem do sistema de coordenadas do efetuador pode descrever enquanto alcança diferentes orientações. Obviamente, o espaço de trabalho destro é um subespaço do espaço de trabalho alcançável. Note que, um manipulador com menos de 6 graus de mobilidade não pode alcançar uma posição e orientação arbitrária no espaço.

O espaço de trabalho é caracterizado pela geometria do manipulador e os limites mecânicos das juntas. Para um manipulador com  $n$  graus de mobilidade, o espaço alcançável é o lugar geométrico dos pontos que podem ser alcançados considerando a equação cinemática direta apenas para a parcela de posição, isto é:

$$p = p(\theta) \quad \theta_{im} \leq \theta_i \leq \theta_{iM} \quad i = 1, \dots, n, \quad (2.15)$$

onde  $\theta_{im}$  ( $\theta_{iM}$ ) é o limite mínimo (máximo) da junta  $i$ . Essa região é finita, fechada,

conectada ( $p(\theta)$  é uma função contínua) e, por isso, é definida pela sua superfície de borda (Figura 2.5). Como as juntas são de revolução ou prismáticas, é intuitivo reconhecer que essa superfície é composta por elementos do tipo planar, esférico, toroidal e cilíndrico.

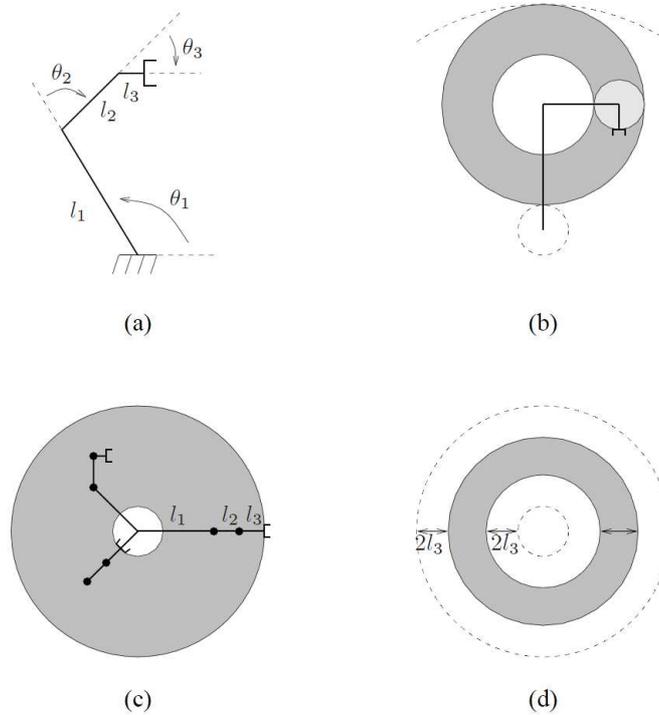


Figura 2.5: Cálculo do espaço de trabalho de uma robô planar de três elos (a). A construção do espaço de trabalho em (b). O espaço de trabalho alcançável (c) e o espaço de trabalho destro (d).

## Redundância Cinemática

Um manipulador é denominado *cinematicamente redundante* quando tem um número de graus de atuação maior que o número de variáveis que são necessárias para descrever uma determinada tarefa (Chiaverini et al. 2008, Siciliano et al. 2009). De acordo com os espaços definidos anteriormente, um manipulador é *intrinsecamente redundante* quando a dimensão do seu espaço operacional é menor que a dimensão do seu espaço de juntas ( $m < n$ ). Então, redundância é um conceito *relativo* à tarefa designada para o manipulador, que pode ser redundante com respeito à uma tarefa e não-redundante com respeito a outra. Mesmo no caso em que  $m = n$ , um manipulador pode ser *funcionalmente redundante* quando apenas um número  $r$  de componentes do espaço operacional forem necessários para a realização de uma determinada tarefa, com  $r < m$ .

Considerando o caso de um manipulador planar com três graus de mobilidade, se apenas a posição do efetuador for especificada, a estrutura apresenta uma re-

redundância funcional ( $n = m = 3, r = 2$ ). Se a orientação do efetuador no plano também for especificada, o manipulador deixa de ser funcionalmente redundante ( $n = m = r = 3$ ). Entretanto, se um manipulador com quatro graus de mobilidade for usado para realizar a mesma tarefa, sua estrutura apresenta redundância intrínseca ( $n = 4, m = 3$ ). Considerando um típico robô industrial com seis graus de atuação pode-se concluir que sua estrutura não é intrinsecamente redundante ( $n = m = 6$ ), mas pode tornar-se funcionalmente redundante dependendo da tarefa a ser executada.

Uma questão importante é compreender porque utilizar intencionalmente um manipulador redundante. A resposta é que a característica de redundância pode fornecer ao manipulador destreza e versatilidade ao seu movimento. De fato, com um manipulador redundante, é possível manter o efetuador em uma mesma posição enquanto alguma junta é movimentada, dependendo de quantos graus de mobilidade adicionais o manipulador possui.

Dessa forma, é possível realizar uma tarefa enquanto o manipulador, por exemplo, desvia de um determinado obstáculo. Além disso, se uma junta alcança o seu limite mecânico, é possível que outras juntas permitam a execução da tarefa desejada.

## 2.3 Cinemática Inversa

A equação da cinemática direta estabelece a relação funcional entre as variáveis das juntas e a posição e orientação do efetuador. Por outro lado, o problema de *cinemática inversa* consiste na determinação das variáveis das juntas correspondendo a uma dada posição e orientação do efetuador (Murray et al. 1994, Siciliano et al. 2009). A solução desse problema é de importância fundamental para transformar as especificações de movimento do efetuador no espaço operacional em correspondentes movimentos no espaço de juntas que permitem a execução de um determinado movimento.

No que diz respeito à equação de cinemática direta em (2.11), a posição e a matriz de rotação do efetuador são calculados de uma maneira única, desde que as variáveis das juntas sejam conhecidas. Por outro lado, o problema de cinemática inversa é muito mais complexo devido às seguintes razões:

- As equações para solucionar são geralmente não-lineares, e por isso não é sempre possível obter uma solução na forma fechada;
- Podem existir *múltiplas soluções*, inclusive com baixo número de graus de mobilidade;
- Podem existir *infinitas soluções*, como no caso de manipuladores cinematicamente redundantes;

- Inexistência de *soluções admissíveis*, devido à estrutura cinemática do manipulador.

A existência de soluções é garantida somente se uma determinada posição e orientação do efetuador pertencem ao espaço de trabalho destro do manipulador (Murray et al. 1994, Siciliano et al. 2009). Por outro lado, o problema de possuir múltiplas soluções não depende apenas do número de graus de mobilidades, mas também do número de parâmetros não-nulos de *Denavit-Hartenberg*. Em geral, quanto maior o número de parâmetros não-nulos, maior é o número de soluções admissíveis. Por exemplo, para um manipulador com seis graus de mobilidade sem limite mecânico das juntas, existem em geral 16 soluções admissíveis.

O cálculo de soluções fechadas requer ou *intuição algébrica* para encontrar as soluções significativas contendo as incógnitas ou *intuição geométrica* para encontrar pontos significativos na estrutura do manipulador, com respeito aos quais é possível e conveniente expressar a sua posição e orientação como uma função de um número reduzido de incógnitas.

Para entender a dificuldade de solucionar o problema de cinemática inversa, considerando um manipulador de  $n$  juntas, a partir da transformação homogênea (2.11) verifica-se que a parcela de orientação possui 9 equações, enquanto que o termo de posição apresenta 3 equações. Portanto, deve-se solucionar um sistema de equações com  $n$  incógnitas e 12 equações. Por outro lado, em todos os casos onde não existem soluções na forma fechada, ou essas soluções são difíceis de encontrar, é apropriado recorrer a técnicas de solução numérica. Essas técnicas são vantajosas pois podem ser aplicadas a qualquer estrutura cinemática, mas em geral não permitem a computação de todas as soluções admissíveis (Murray et al. 1994, Siciliano et al. 2009). Dentre os métodos propostos para solucionar o problema de cinemática inversa destacam-se:

- **Desacoplamento Cinemático** - na maioria do manipuladores industriais, compostos por punhos esféricos, é possível desacoplar as soluções da cinemática inversa da posição e da orientação (Siciliano et al. 2009);
- **Algoritmo Iterativo** - é possível calcular a cinemática inversa de uma forma recursiva, utilizando a inversa da matriz Jacobiana e a equação de cinemática direta, quando uma ferramenta computacional está disponível (Siciliano et al. 2009);
- **Decomposição em Sub-problemas de Paden-Kahan** - para manipuladores com determinadas estruturas a solução da cinemática inversa pode ser obtida por meio da solução de diversos subproblemas simples e de identificação imediata (Murray et al. 1994).

## 2.4 Cinemática Diferencial

A cinemática diferencial estabelece a relação entre as velocidades das juntas e as correspondentes *velocidades linear e angular* do efetuador. Essa relação é mapeada por uma matriz, denominada *Jacobiano geométrico*, que depende da configuração do manipulador (Murray et al. 1994, Siciliano et al. 2009). Por outro lado, se a posição e orientação do efetuador estão expressos em termos de uma *representação mínima* no espaço operacional, é possível calcular a matriz Jacobiana através da diferenciação da equação da cinemática direta com respeito às variáveis das juntas. Essa diferenciação resulta no *Jacobiano analítico* que, em geral, é diferente do Jacobiano geométrico.

O Jacobiano constitui uma das ferramentas mais importantes para estabelecer as características de um manipulador. Ele é útil para encontrar *configurações singulares*, analisar *redundâncias*, determinar *algoritmos de controle* baseados em cinemática inversa, além de outras utilidades (Siciliano et al. 2009).

### 2.4.1 Jacobiano Geométrico

O objetivo da cinemática diferencial é encontrar uma relação entre as velocidades das juntas e as velocidades linear e angular do efetuador. Para isso, deve-se expressar a velocidade linear  $\dot{p}$  e a velocidade angular  $\omega$  como uma função das velocidades das juntas  $\dot{\theta}$  por meio das seguintes relações:

$$\dot{p} = J_P(\theta)\dot{\theta}, \quad (2.16)$$

$$\omega = J_O(\theta)\dot{\theta}. \quad (2.17)$$

Em (2.16)  $J_P$  é a matriz ( $3 \times n$ ) relacionando a contribuição das velocidades das juntas  $\dot{\theta}$  para a velocidade linear do efetuador  $\dot{p}$ , enquanto que em (2.17)  $J_O$  é a matriz ( $3 \times n$ ) relacionando a contribuição das velocidades das juntas  $\dot{\theta}$  para a velocidade angular do efetuador  $\omega$ . De uma forma compacta, essas duas equações podem ser escritas como

$$v = \begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = J(\theta)\dot{\theta}, \quad (2.18)$$

que representa a equação da *cinemática diferencial* do manipulador. A matriz ( $6 \times n$ )  $J$  é o *Jacobiano geométrico* do manipulador

$$J = \begin{bmatrix} J_P \\ J_O \end{bmatrix}, \quad (2.19)$$

que, em geral, é uma função das variáveis das juntas.

## Cálculo do Jacobiano

Para o cálculo do Jacobiano, é conveniente separar a parcela da velocidade linear da parcela da velocidade angular. Para a contribuição da *velocidade linear*, a derivada no tempo do vetor posição  $p$  pode ser escrita como

$$\dot{p} = \sum_{i=1}^n \frac{\partial \dot{p}}{\partial \dot{\theta}_i} \dot{\theta}_i = \sum_{i=1}^n J_{P_i} \dot{\theta}_i. \quad (2.20)$$

Cada termo do somatório representa a contribuição da velocidade de uma única junta  $i$  para a velocidade linear do efetuador enquanto todas as outras juntas estão paradas. Note que, a velocidade das juntas  $\dot{\theta}_i$  produz uma velocidade espacial no sistema de coordenadas  $\bar{E}_n$ , solidário ao último elo. Distinguindo essa contribuição pelo tipo de junta, as seguintes relações são válidas:

(i) Se a junta  $i$  for prismática,  $\theta_i = d_i$  e

$$J_{P_i} \dot{\theta}_i = \vec{h}_i \dot{d}_i \quad \Rightarrow \quad J_{P_i} = \vec{h}_i, \quad (2.21)$$

onde  $\vec{h}_i$  é o vetor unitário na direção do eixo de rotação da  $i$ -ésima junta.

(ii) Se a junta  $i$  for de revolução,  $\theta_i = \vartheta_i$  e

$$J_{P_i} \dot{\theta}_i = \omega_n \times r_{in} = \dot{\vartheta}_i \vec{h}_i \times (p - p_i) \quad \Rightarrow \quad J_{P_i} = \vec{h}_i \times (p - p_i), \quad (2.22)$$

onde  $r_{in}$  é o vetor posição do efetuador em relação ao eixo da  $i$ -ésima junta.

Para a contribuição da *velocidade angular* tem-se que:

$$\omega = \omega_n = \sum_{i=1}^n \omega_{i-1,i} = \sum_{i=1}^n J_{O_i} \dot{\theta}_i, \quad (2.23)$$

e distinguindo essa contribuição pelo tipo de junta, as seguintes relações são válidas:

(i) Se a junta  $i$  for prismática,  $\theta_i = d_i$  e

$$J_{O_i} \dot{\theta}_i = 0 \quad \Rightarrow \quad J_{O_i} = 0. \quad (2.24)$$

(ii) Se a junta  $i$  for de revolução,  $\theta_i = \vartheta_i$  e

$$J_{O_i} \dot{\theta}_i = \dot{\vartheta}_i \vec{h}_i \quad \Rightarrow \quad J_{O_i} = \vec{h}_i. \quad (2.25)$$

Então, o Jacobiano em (2.19) pode ser particionado em vetores coluna ( $3 \times 1$ )

$J_{P_i}$  e  $J_{O_i}$  como

$$J = \begin{bmatrix} J_{P_i} & \dots & J_{P_n} \\ J_{O_i} & & J_{O_n} \end{bmatrix}, \quad (2.26)$$

onde

$$\begin{bmatrix} J_{P_i} \\ J_{O_i} \end{bmatrix} = \begin{cases} \begin{bmatrix} \vec{h}_i \\ 0 \end{bmatrix} & \text{(junta prismática).} \\ \begin{bmatrix} \vec{h}_i \times (p - p_i) \\ \vec{h}_i \end{bmatrix} & \text{(junta de revolução).} \end{cases} \quad (2.27)$$

A expressão em (2.27) permite que o cálculo do Jacobiano seja realizado de uma forma simples, sistemática e baseada nas relações da cinemática direta. Os vetores  $\vec{h}_i$ ,  $p$  e  $p_i$  são funções das variáveis das juntas, e podem ser definidos por:

- (a)  $\vec{h}_i$  é obtido a partir da terceira coluna da matriz de rotação  $R_{0i-1}$ ;
- (b)  $p$  é obtido a partir dos três primeiros elementos da quarta coluna da transformação homogênea  $T_{0e}$ ;
- (c)  $p_i$  é obtido a partir dos três primeiros elementos da quarta coluna da transformação homogênea  $T_{0i}$ .

As equações acima podem ser usadas para calcular as velocidades translacionais e rotacionais em qualquer ponto da estrutura do manipulador desde que a sua cinemática direta seja conhecida. Então, as velocidades linear e angular do efetuador expressas com respeito ao sistema de coordenadas da base  $\bar{E}_b$  são dadas por:

$$\begin{bmatrix} (\dot{p}_{be})_b \\ (\omega_{be})_b \end{bmatrix} = (J_{be})_b \dot{\theta}, \quad (2.28)$$

onde  $(J_{be})_b$  representa o Jacobiano geométrico do efetuador com respeito ao sistema de coordenadas da base.

Para o caso em que o manipulador de interesse é do tipo *antropomórfico*, possuindo um braço articulado e um punho esférico (Siciliano et al. 2009), a parametrização da cinemática direta confere ao Jacobiano geométrico do punho em relação à base  $(J_{bw})_b$  a seguinte forma particular:

$$(J_{bw})_b = \begin{bmatrix} J_{11} & 0_{3 \times 3} \\ J_{21} & J_{22} \end{bmatrix}, \quad (2.29)$$

que permite o *desacoplamento* entre a posição e a orientação do efetuador.

Finalmente, é importante notar que a matriz Jacobiana depende de qual sistema de coordenadas a velocidade do efetuador está expressa. Desta forma, se é desejado representar o Jacobiano em um sistema de coordenadas diferente  $\bar{E}_k$ , é suficiente conhecer a matriz de rotação relativa  $R_{kb} = \bar{E}_k^* \bar{E}_b$ . Então, as relações entre as velocidades dos dois sistemas de coordenadas é dada por

$$\begin{bmatrix} (\dot{p}_{be})_k \\ (\omega_{be})_k \end{bmatrix} = \begin{bmatrix} R_{kb} & 0_{3 \times 3} \\ 0_{3 \times 3} & R_{kb} \end{bmatrix} \begin{bmatrix} (\dot{p}_{be})_b \\ (\omega_{be})_b \end{bmatrix}, \quad (2.30)$$

que, de acordo com (2.18) e considerando  $\bar{E}_k$  invariante no tempo, resulta em

$$(J_{be})_k = \begin{bmatrix} R_{kb} & 0_{(3 \times 3)} \\ 0_{(3 \times 3)} & R_{kb} \end{bmatrix} (J_{be})_b, \quad (2.31)$$

onde  $(J_{be})_k$  representa o Jacobiano geométrico do efetuador em relação à base, expresso com respeito ao sistema de coordenadas  $\bar{E}_k$  (Siciliano et al. 2009).

## 2.4.2 Singularidades Cinemáticas

Em geral, o Jacobiano é uma função de uma configuração de ângulos de juntas (2.13). As configurações em que a matriz  $J$  perde posto são chamadas de *singularidades cinemáticas*. Encontrar as singularidades de um manipulador é de grande interesse devido aos seguintes fatores (Murray et al. 1994, Siciliano et al. 2009):

1. Singularidades representam configurações em que a mobilidade da estrutura é reduzida, ou seja, não é possível impor um movimento arbitrário ao efetuador;
2. Quando a estrutura está numa singularidade, podem existir infinitas soluções para o problema de cinemática inversa;
3. Na vizinhança de uma singularidade, pequenas velocidades no espaço operacional podem causar grandes velocidades no espaço das juntas.

Neste contexto, as singularidades podem ser classificadas como:

- *Singularidades de Fronteira*, que ocorrem quando o manipulador está esticado ou retraído, e podem ser facilmente evitadas se o manipulador não se mover para perto dos limites do espaço de trabalho alcançável;
- *Singularidades Internas*, que ocorrem dentro do espaço de trabalho alcançável e são geralmente causadas pelo alinhamento de dois ou mais eixos de movimento, ou pela realização de uma configuração particular do efetuador. Ao contrário da singularidade de fronteira, essas singularidades constituem um sério problema e podem ser encontradas em qualquer lugar do espaço de trabalho alcançável para uma tarefa planejada no espaço operacional.

É válido ressaltar que já existem soluções alternativas para lidar com o problema de singularidade cinemática como, por exemplo, o Método DLS (do inglês, *damped least-square*) proposto em (Nakamura & Hanafusa 1986), o Método FIK (do inglês, *feedback inverse kinematics*) proposto em (Nakamura & Hanafusa 1986) e o Método da Inversa Filtrada (Vargas 2013, Vargas et al. 2014).

### 2.4.3 Jacobiano Analítico

Se a posição e orientação do efetuador estão especificadas em termos de um número mínimo de parâmetros no espaço operacional (2.12), é possível obter o Jacobiano usando uma *técnica analítica*, por meio da diferenciação da função da cinemática direta com respeito às variáveis das juntas.

A velocidade translacional do efetuador pode ser expressa pela derivada temporal do vetor posição do efetuador com respeito ao sistema de coordenadas base:

$$\dot{p} = \frac{\partial p}{\partial \theta} \dot{\theta} = J_p(\theta) \dot{\theta}. \quad (2.32)$$

Considerando que a orientação do efetuador  $R \in SO(3)$  pode ser representada por uma parametrização  $\phi$  adequada, a velocidade rotacional do efetuador é dada por

$$\dot{\phi} = \frac{\partial \phi}{\partial \theta} \dot{\theta} = J_\phi(\theta) \dot{\theta}. \quad (2.33)$$

Note que, o Jacobiano  $J_\phi$  não é obtido diretamente a partir de  $\frac{\partial \phi}{\partial \theta}$ , pois a função  $\phi$  em geral não é expressa diretamente em termos de  $\theta$ , necessitando do cálculo dos elementos da matriz de rotação  $R(\theta)$  relativa.

Dessa forma, a equação da cinemática diferencial pode ser obtida por meio da derivada temporal da equação da cinemática direta (2.14), resultando em

$$\dot{x} = \begin{bmatrix} \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} J_p(\theta) \\ J_\phi(\theta) \end{bmatrix} \dot{\theta} = J_a(\theta) \dot{\theta}, \quad (2.34)$$

onde

$$J_a(\theta) = \frac{\partial k(\theta)}{\partial \theta} \quad (2.35)$$

é chamado de *Jacobiano Analítico*, e em geral é diferente do Jacobiano geométrico  $J$ , uma vez que a velocidade angular do efetuador  $\omega$  com respeito ao sistema de coordenadas da base  $\bar{E}_b$  não é dada por  $\dot{\phi}$ .

## 2.4.4 Jacobiano da Representação

Considerando uma representação mínima de orientação para  $R \in SO(3)$  é possível encontrar uma relação entre a velocidade angular  $\omega$  e a velocidade rotacional  $\dot{\phi}$  (ou derivada temporal da representação de orientação) por meio de

$$\omega = J_r(\phi)\dot{\phi}, \quad (2.36)$$

onde  $J_r(\phi)$  é o *Jacobiano da Representação* e depende da representação da orientação escolhida.

A partir de (2.36) pode-se estabelecer a relação entre o Jacobiano analítico e o Jacobiano geométrico por

$$v = \begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & J_r(\phi) \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{\phi} \end{bmatrix} = T_r(\phi)\dot{x}, \quad (2.37)$$

e, portanto, em vista de (2.18) e (2.34) chega-se a:

$$J(\theta) = T_r(\phi)J_a(\theta). \quad (2.38)$$

É válido mencionar que outras representações de orientação podem ser utilizadas para expressar a relação entre a velocidade angular e a derivada temporal da representação de orientação como, por exemplo, o quatérnio unitário conforme será apresentado no Apêndice B.

## 2.5 Conclusões

Neste capítulo são apresentados alguns conceitos e definições utilizados para determinar a cinemática direta e a cinemática diferencial de um sistema robótico. É apresentada uma introdução sobre outros tópicos relevantes relacionados à cinemática de um robô como, por exemplo, o problema da cinemática inversa e singularidades cinemáticas. No próximo capítulo serão apresentados os conceitos e definições relacionados às abordagens de controle cinemático e desvio de obstáculos.

# Capítulo 3

## Controle Cinemático e Desvio de Obstáculos

Neste capítulo apresenta-se uma estratégia de controle cinemático e desvio de obstáculos para robôs manipuladores redundantes. O algoritmo de controle de posição e orientação utiliza a pseudo-inversa da matriz Jacobiana com um fator de ponderação para os limites mecânicos das juntas e aproveita a redundância cinemática do robô para desviar de obstáculos.

A abordagem de desvio de obstáculo é baseada no uso de dois tipos de campos potenciais artificiais repulsivos, atuando nas juntas e no efetuador do robô. O primeiro é definido em termos de uma restrição adicional de prioridade secundária e o segundo é utilizado como um sinal de velocidade repulsiva adicionado à lei de controle de posição convencional.

### 3.1 Controle Cinemático

Considere o problema de controle cinemático para um robô manipulador com  $n$  graus de atuação. Neste contexto, as seguintes hipóteses são assumidas:

(H1) A cinemática do robô é conhecida;

(H2) O efeito da dinâmica do robô pode ser desprezada.

A hipótese (H2) é válida para robôs manipuladores que apresentam elevados fatores de redução nas engrenagens e/ou quando baixas velocidades (ou acelerações lentas) são utilizadas para realizar a tarefa de interesse, e nesse caso o efeito do acoplamento dinâmico é considerado desprezível (Siciliano et al. 2009).

Em geral, a maioria dos robôs industriais disponíveis comercialmente possui uma malha de controle de velocidade interna para acionamento direto das juntas. A Figura 3.1 apresenta um esquema ilustrativo de uma típica malha de controle de velocidade de um manipulador, onde  $x$  representa a sua posição e orientação,  $\dot{x}$  denota

a sua velocidade de translação e rotação,  $\tau$  denota o vetor de torques aplicado nas juntas e  $J_a(\theta)$  é o Jacobiano analítico. O bloco *Driver* fornece a potência necessária para o acionamento dos atuadores do robô a partir de um sinal de controle  $v$  de entrada.

O sinal de controle  $v$  é gerado pelo controlador proporcional com ganho  $K$  que amplifica o sinal do erro  $e$ , medido entre o vetor de velocidades das juntas do robô  $\dot{\theta}$  e o sinal externo de referência  $\dot{\theta}_d$ . Então, para uma entrada  $u = \dot{\theta}_d$  e uma malha de controle de alto ganho ( $K$  elevado), tem-se que  $e \rightarrow 0$  e conseqüentemente  $u \approx \dot{\theta}$ .

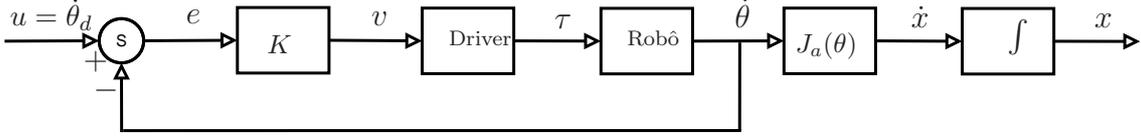


Figura 3.1: Diagrama em blocos da malha de controle de velocidade.

Portanto, considerando a abordagem de controle cinemático o movimento do manipulador pode ser descrito por:

$$\dot{\theta}_i = u_i, \quad i = 1, \dots, n, \quad (3.1)$$

onde  $\dot{\theta}_i$  é a velocidade angular da  $i$ -ésima junta,  $u_i$  é um sinal de controle de velocidade aplicado ao *driver* do motor da  $i$ -ésima junta e  $n$  é o número de juntas. A partir da modelagem cinemática, as variáveis no espaço das juntas são relacionadas com as variáveis no espaço operacional por meio dos seguintes mapeamentos de cinemática direta e cinemática diferencial:

$$p = k(\theta), \quad \dot{p} = J_p(\theta) \dot{\theta}, \quad (3.2)$$

onde  $k(\theta)$  é uma função  $m$ -dimensional, em geral não-linear, e  $J_p = \frac{\partial k}{\partial \theta} \in \mathbb{R}^{m \times n}$  é o Jacobiano analítico de posição,  $p, \dot{p} \in \mathbb{R}^m$  denotam a posição e a velocidade linear do efetuador do robô, expressos no sistema de coordenadas da base, e  $\theta, \dot{\theta} \in \mathbb{R}^n$  denotam a posição e a velocidade angular das juntas do manipulador.

Considere que a orientação do efetuador é descrita pela representação de quatérnio unitário denotada por um vetor de dimensão  $(4 \times 1)$   $q = (q_s, q_v) \in \mathbb{H}$ , onde  $q_s \in \mathbb{R}$  é a parte escalar e  $q_v \in \mathbb{R}^3$  é a parte vetorial, sujeitas a restrição de norma unitária  $q_s^2 + q_v^T q_v = 1$  (Siciliano et al. 2009).

A equação de propagação do quatérnio relaciona a derivada temporal do quatérnio unitário  $\dot{q} \in \mathbb{H}$  com a velocidade angular do efetuador do robô  $\omega \in \mathbb{R}^3$  de acordo com o *enfoque inercial* como:

$$\dot{q} = \frac{1}{2} B(q) \omega, \quad B(q) = \begin{bmatrix} -q_v^T \\ q_s I - Q(q_v) \end{bmatrix}, \quad (3.3)$$

onde  $Q(\cdot) : \mathbb{R}^3 \mapsto \text{SO}(3)$  denota o operador matriz anti-simétrica e  $J_r(\phi) = 2B^\top(q) \in \mathbb{R}^{3 \times 4}$  é o conhecido *Jacobiano da representação*.

A equação de cinemática diferencial fornece a relação entre o vetor de velocidades nas juntas e a correspondente velocidade linear e angular do efetuador do robô como:

$$\begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = \begin{bmatrix} J_p(\theta) \\ J_o(\theta) \end{bmatrix} \dot{\theta} = J(\theta) \dot{\theta}, \quad (3.4)$$

onde  $J(\theta) \in \mathbb{R}^{m \times n}$  é o *Jacobiano geométrico* do robô manipulador. Note que, a orientação do efetuador do robô pode ser fornecida em termos dos ângulos das juntas do manipulador como  $q = g(\theta)$ , where  $g(\cdot)$  é uma função vetorial, em geral não-linear. Então, considerando (3.3) e (3.4) tem-se que  $J_o(\theta) = J_r(q) (\partial g(\theta) / \partial \theta)$ . É válido mencionar que o modelo cinemático possui a seguinte propriedade muito útil para a análise de estabilidade de robôs manipuladores com juntas de revolução.

Propriedade (P1) (Dixon 2004):  $J(\theta)$  é limitado para todos os possíveis valores de  $\theta(t)$ , isto é, o Jacobiano depende de  $\theta(t)$  como argumentos de funções trigonométricas limitadas e  $\|J(\theta)\|_\infty \leq c_1$ , onde  $c_1 \in \mathbb{R}$  é uma constante positiva conhecida.

Então, a partir de (3.4) e considerando a abordagem de controle cinemático (3.1), obtém-se o seguinte sistema de controle:

$$\begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = J(\theta) u. \quad (3.5)$$

Para o caso onde a matriz Jacobiana é retangular ( $m < n$ ), o sinal de controle de velocidade  $u \in \mathbb{R}^n$  é dado por

$$u(t) = J^\dagger(q) v, \quad v = \begin{bmatrix} v_p \\ v_o \end{bmatrix}, \quad (3.6)$$

onde  $J^\dagger = J^\top (J J^\top)^{-1}$  é a pseudo-inversa à direita de  $J$  e  $v \in \mathbb{R}^m$  é o sinal de controle Cartesiano a ser projetado. Note que,  $v_p$  e  $v_o$  são os sinais de controle de posição e de orientação, projetados para comandar a pose do efetuador do robô simultaneamente.

O sinal de controle (3.6) minimiza localmente a norma das velocidades das juntas, desde que a cinemática do robô seja conhecida e  $v(t)$  não conduza o robô para *configurações singulares*, onde a matriz Jacobiana possui posto deficiente. A falha desta última condição ainda é um problema em investigação na área de robótica. Note que, no caso onde a matriz Jacobiana é quadrada ( $m = n$ ) e não-singular, o sinal de controle de velocidade é obtido por  $u(t) = J^{-1}(q) v$ .

### 3.1.1 Restrições das Juntas

Neste trabalho, o sinal de controle (3.6) pode ser modificado para evitar que os movimentos do robô alcancem os limites mecânicos das juntas. Considerando o problema de controle cinemático para robôs manipuladores redundantes, a seguinte relação baseada na *pseudo-inversa ponderada* da matriz Jacobiana pode ser utilizada (Bjerkeng et al. 2011):

$$u = W^{-1} J^\top (J W^{-1} J^\top)^{-1} v = J_w^\dagger v, \quad (3.7)$$

onde  $W \in \mathbb{R}^{n \times n}$  é uma matriz de ponderação, simétrica positiva definida, e  $J J_w^\dagger = I$ .

Note que, a relação (3.7) é válida também desde que  $v(t)$  não conduza o robô para configurações singulares. A matriz de ponderação  $W$  é projetada de forma que quando uma junta do robô aproxima-se do seu limite, os elementos da diagonal principal tendem a infinito. Isso garante que as velocidades das juntas tendam a zero na vizinhança de um limite mecânico (Chan & Dubey 1995). Então, os elementos diagonais de  $W^{-1}$  tendem a zero conforme a junta aproxima-se do seu limite, e são iguais a 1 se a junta está no meio da sua faixa. Desta forma, a matriz de ponderação  $W$  é definida para cada elemento da diagonal por:

$$W_{i,i} = 1 + \left| \frac{\partial H(\theta)}{\partial \theta_i} \right|, \quad (3.8)$$

onde  $H(\theta)$  é um critério de desempenho para evitar os limites mecânicos das juntas, dado por:

$$H(\theta) = \sum_{i=1}^n \frac{1}{4} \frac{(\theta_{i_{max}} - \theta_{i_{min}})^2}{(\theta_{i_{max}} - \theta_i)(\theta_i - \theta_{i_{min}})}, \quad (3.9)$$

onde  $\theta_{i_{max}}$  e  $\theta_{i_{min}}$  são os limites superior e inferior da  $i$ -ésima junta respectivamente. De (3.9) é intuitivo obter que

$$\frac{\partial H(\theta)}{\partial \theta_i} = \frac{(\theta_{i_{max}} - \theta_{i_{min}})^2 (2\theta_i - \theta_{i_{max}} - \theta_{i_{min}})}{4(\theta_i - \theta_{i_{max}})^2 (\theta_i - \theta_{i_{min}})^2}, \quad (3.10)$$

De fato, se  $\theta_i \rightarrow \theta_{i_{min}}$  ou  $\theta_i \rightarrow \theta_{i_{max}}$  tem-se que  $\left| \frac{\partial H(\theta)}{\partial \theta_i} \right| \rightarrow \infty$ . Além disso, se  $\theta_i = (\theta_{i_{max}} + \theta_{i_{min}})/2 = \theta_i^*$ , implica que  $\left| \frac{\partial H(\theta)}{\partial \theta_i} \right| = 0$ . Uma representação gráfica de  $\left| \frac{\partial H(\theta)}{\partial \theta_i} \right|$  é apresentada na Figura 3.2.

Note que em (3.8), não há diferença se a junta está se aproximando ou se afastando do limite. Dessa forma, quando uma junta está próxima do seu limite mecânico, ela permanecerá próxima deste limite indefinidamente. Uma forma de lidar com essa limitação é redefinir (3.8) de tal forma que a ponderação não atue caso a

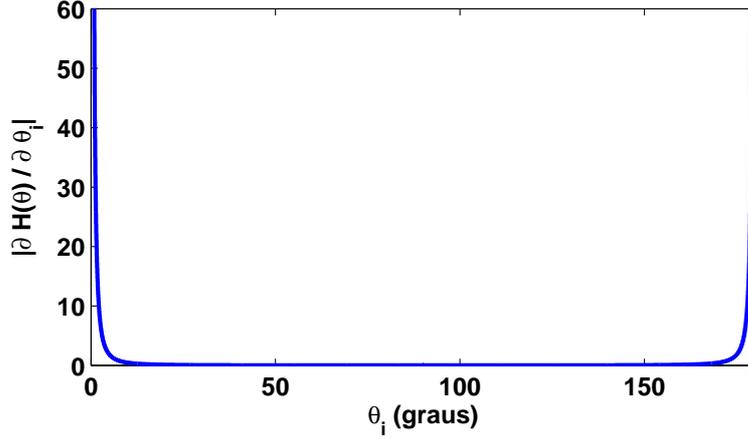


Figura 3.2: Representação Gráfica de  $\left| \frac{\partial H(\theta)}{\partial \theta_i} \right|$  para  $\theta_{i_{max}} = 180^\circ$  e  $\theta_{i_{min}} = 0^\circ$ .

junta esteja se afastando do limite (Chan & Dubey 1995):

$$W_{i,i} = \begin{cases} 1 + \left| \frac{\partial H(\theta)}{\partial \theta_i} \right| & \text{se } \Delta \left| \frac{\partial H(\theta)}{\partial \theta_i} \right| \geq 0, \\ 1 & \text{se } \Delta \left| \frac{\partial H(\theta)}{\partial \theta_i} \right| < 0, \end{cases} \quad (3.11)$$

onde  $\Delta \left| \frac{\partial H(\theta)}{\partial \theta_i} \right|$  é a variação no tempo de  $\left| \frac{\partial H(\theta)}{\partial \theta_i} \right|$ .

### 3.1.2 Controle de Posição

Considere que o objetivo do controle para uma determinada tarefa é seguir uma posição desejada variante no tempo  $p_d(t)$  a partir de uma posição inicial  $p$ , isto é:

$$p \rightarrow p_d(t), \quad e_p = p_d(t) - p \rightarrow 0, \quad (3.12)$$

onde  $e_p \in \mathbb{R}^3$  é o erro de posição. A partir de (3.5), (3.6) e (3.7) chega-se a  $\dot{p} = v_p$ , obtendo-se a seguinte equação de erro:

$$\dot{e}_p = \dot{p}_d - v_p. \quad (3.13)$$

Usando uma lei de controle baseada em uma ação proporcional e um termo *feed-forward* dada por

$$v_p = K_p e_p + \dot{p}_d, \quad (3.14)$$

onde  $K_p$  é a matriz de ganho de posição, a dinâmica do erro de posição é governada por:

$$\dot{e}_p + K_p e_p = 0. \quad (3.15)$$

Então, escolhendo-se  $K_p = K_p^\top > 0$ , o sistema de controle em malha-fechada é exponencialmente estável e, conseqüentemente  $\lim_{t \rightarrow \infty} e_p(t) = 0$ .

### 3.1.3 Controle de Orientação

Considere que o objetivo de controle para uma determinada tarefa é alcançar uma orientação desejada invariante no tempo  $R_d$  a partir de uma orientação inicial  $R$ , isto é:

$$R \rightarrow R_d, \quad R_q = R_d R^\top \rightarrow I, \quad (3.16)$$

onde  $R_q \in SO(3)$  é a matriz de erro de orientação no enfoque inercial. A representação em quatérnio unitário de  $R_q$  é dada por  $e_q = [e_{qs} \ e_{qv}^\top]^\top$ , com  $e_q = q_d * q^{-1}$ , onde  $q = (q_s, q_v)$  e  $q_d = (q_{sd}, q_{vd})$  são as representações em quatérnio unitário de  $R$  e  $R_d$  respectivamente e “ $*$ ” é o operador produto de quatérnio.

Note que,  $e_q = [1 \ 0^\top]^\top$  se e somente se  $R$  e  $R_d$  estão alinhados. Portanto, pode-se definir o erro de orientação como:

$$e_o = e_{qv} = q_s q_{vd} - q_{sd} q_v - Q(q_{vd}) q_v, \quad (3.17)$$

onde  $Q(\cdot)$  é o operador produto vetorial definido no Apêndice B.

A partir de (3.5), (3.6) e (3.7) chega-se a  $\omega = v_o$ . Então, usando uma lei de controle baseada em uma ação proporcional e um termo *feedforward* dada por

$$v_o = \omega_d + K_o e_o, \quad (3.18)$$

onde  $K_o \in \mathbb{R}^{3 \times 3}$  é a matriz de ganho de orientação e  $\omega_d \in \mathbb{R}^3$  é a velocidade angular desejada, obtêm-se a seguinte dinâmica para o erro de orientação:

$$\tilde{\omega} + K_o e_o = 0, \quad (3.19)$$

onde  $\tilde{\omega} = (\omega_d - \omega) \in \mathbb{R}^3$  é o erro de velocidade angular do efetuador. Então, o seguinte teorema pode ser enunciado.

**Teorema 1.** (*Estabilidade do Esquema de Controle de Orientação*) *Considere o sistema de controle em malha fechada descrito por (3.5) e (3.7) com a lei de controle de orientação (3.18). Assuma que  $q_d$  é a representação em quatérnio unitário para a matriz de orientação desejada  $R_d \in SO(3)$ . Baseado nas hipóteses (H1)-(H2) as seguintes propriedades são válidas: (i) todos os sinais do sistema em malha fechada são uniformemente limitados; (ii)  $\lim_{t \rightarrow \infty} e_{qv}(t) = 0$ ,  $\lim_{t \rightarrow \infty} e_{qs}(t) = \pm 1$ . Portanto, o sistema de controle em malha fechada é quase globalmente assintoticamente estável.*

**Prova:** Analisando a expressão (3.3) e (3.19), pode-se observar que a equação de

erro de orientação é *não-linear* uma vez que contém o erro de velocidade angular do efetuador, ao invés da derivada temporal do erro de orientação. Portanto, para verificar a estabilidade do sistema deve-se utilizar uma abordagem baseada na Teoria de Estabilidade de Lyapunov (Slotine & Li 1991). Então, para analisar a estabilidade do sistema (3.19), considera-se a seguinte função candidata de Lyapunov, definida positiva:

$$V(e_{qs}, e_{qv}) = [e_{qs} - 1]^2 + e_{qv}^T e_{qv} > 0, \quad (3.20)$$

e diferenciando (3.20) com respeito ao tempo ao longo das trajetórias do sistema, tem-se que

$$\dot{V} = 2[e_{qs} - 1]\dot{e}_{qs} + 2e_{qv}^T \dot{e}_{qv}. \quad (3.21)$$

Substituindo a equação de propagação do erro de quatérnio (Yuan 1988)

$$\dot{e}_{qs} = -\frac{1}{2} e_{qv}^T \tilde{\omega}, \quad (3.22)$$

$$\dot{e}_{qv} = \frac{1}{2} [e_{qs} I - Q(e_{qv})] \tilde{\omega}, \quad (3.23)$$

em (3.21) obtém-se:

$$\dot{V} = -[e_{qs} - 1]e_{qv}^T \tilde{\omega} + e_{qv}^T [e_{qs} I - Q(e_{qv})] \tilde{\omega}. \quad (3.24)$$

Após algumas manipulações algébricas chega-se a:

$$\dot{V} = e_{qv}^T \tilde{\omega} = -e_{qv}^T K_o e_{qv}. \quad (3.25)$$

Note que, para  $K_o = K_o^T > 0$  a derivada temporal de  $V$  dada por (3.25) é negativa semi-definida, isto é,  $\dot{V} \leq 0$ .

Então, uma vez que  $V$  é continuamente diferenciável, radialmente ilimitada, definida positiva e  $\dot{V} \leq 0$  em todo o espaço de estados, a partir do *Princípio da Invariância de LaSalle* (Khalil 2002) tem-se que todas as trajetórias convergem para o maior conjunto invariante  $\bar{\Omega}$  definido por:

$$\bar{\Omega} = \{(e_{qs}, e_{qv}) : \dot{V} = 0\} = \{(e_{qs}, e_{qv}) : e_{qv} = 0_{1 \times 3}\}. \quad (3.26)$$

Sabendo que  $\tilde{\omega} = -K_o e_{qv}$ , no conjunto invariante  $\bar{\Omega}$  tem-se que  $\tilde{\omega} = 0$ , uma vez que  $e_{qv} = 0$ . Portanto, a partir de (3.22) e (3.23) tem-se que  $\dot{e}_{qs} = 0$  e  $\dot{e}_{qv} = 0$ , resultando em:

$$\bar{\Omega} = \{(e_{qs}, e_{qv}) : e_{qs} = \pm 1, e_{qv} = 0_{1 \times 3}\}, \quad (3.27)$$

e, finalmente, implicando que  $(e_{qs}, e_{qv}) = (\pm 1, 0_{1 \times 3})$  é um ponto de equilíbrio quase-global assintoticamente estável (Wen & Kreutz-Delgado 1991). Note que  $e_{qs} = \pm 1$

para satisfazer a restrição  $e_{qs}^2 + e_{qv}^T e_{qv} = 1$ . Neste trabalho, utiliza-se o termo quase-global para indicar que o domínio de atração é o espaço de estado completo, exceto para um conjunto de medida de Lebesgue nula (Monzon 2003).  $\square$

### 3.1.4 Controle Cinemático de Manipuladores Redundantes

Quando o manipulador é redundante, a matriz Jacobiana possui mais colunas do que linhas e, por isso, infinitas soluções podem existir para a equação de cinemática diferencial  $v = J(\theta)\dot{\theta}$ . Uma metodologia para encontrar soluções viáveis consiste em formular o problema de controle cinemático como um problema de otimização linear com restrição (Chiaverini et al. 2008).

A característica de redundância do robô permite gerar movimentos internos nas juntas que reconfiguram a estrutura do manipulador sem modificar a posição e orientação do efetuador. Isso pode ser alcançado adicionando-se um termo, denominado solução homogênea, à lei de controle (3.7) como (Bjerkeng et al. 2011):

$$u = J_w^\dagger v + (I - J_w^\dagger J) W^{-1} \sigma, \quad (3.28)$$

onde  $\sigma \in \mathbb{R}^n$  é um vetor de velocidade das juntas arbitrária.

Note que a lei de controle (3.28) é composta de dois termos: o primeiro é relativo à norma mínima da velocidade das juntas, enquanto que o segundo, chamado de *solução homogênea*, satisfaz uma restrição adicional ponderada, especificada por meio de  $W^{-1}\sigma$ . Note que a matriz  $(I - J_w^\dagger J)$  projeta o vetor  $W^{-1}\sigma$  no espaço nulo do Jacobiano sem violar a restrição (3.6), que é denominada de objetivo primário. Uma justificativa para a utilização do termo  $W^{-1}\sigma$  ao invés de  $\sigma$ , baseada na análise da dinâmica da junta quando ela está próxima do seu limite, é apresentada em (Bjerkeng et al. 2011). Então, considerando o problema de controle de posição e orientação para um manipulador redundante, o seguinte teorema pode ser enunciado:

**Teorema 2.** *(Estabilidade do Esquema de Controle de Posição e Orientação) Considere o sistema de controle em malha fechada descrito por (3.5) e (3.28), com a lei de controle de posição (3.14) e a lei de controle de orientação (3.18). Assuma que o sinal de referência  $p_d(t)$  é contínuo por partes e uniformemente limitado e  $q_d$  é a representação em quatérnio unitário para a matriz de orientação desejada  $R_d \in SO(3)$ . Baseado nas hipóteses (H1)-(H2) as seguintes propriedades são válidas: (i) todos os sinais do sistema em malha fechada são uniformemente limitados; (ii)  $\lim_{t \rightarrow \infty} e_p(t) = 0$  e  $\lim_{t \rightarrow \infty} e_{qv}(t) = 0$ ,  $\lim_{t \rightarrow \infty} e_{qs}(t) = \pm 1$ . Portanto, o sistema de controle em malha fechada é quase-globalmente assintoticamente estável.*

**Prova:** Para analisar a estabilidade e a convergência da lei de controle de posição e

orientação considera-se a seguinte função candidata de Lyapunov, definida positiva:

$$V(e_p, e_{qs}, e_{qv}) = \frac{1}{2} e_p^\top e_p + [e_{qs} - 1]^2 + e_{qv}^\top e_{qv} > 0. \quad (3.29)$$

A partir do sistema de controle  $v = J u$ , onde  $J = [J_p \ J_o]^\top$  e considerando a equação de propagação do erro de quatérnio (3.22) e (3.23) obtém-se a seguinte dinâmica para o erro de posição:

$$\dot{e}_p = \dot{p}_d - J_p u, \quad (3.30)$$

e para o erro de orientação:

$$\dot{e}_{qs} = \frac{1}{2} e_{qv}^\top J_o u, \quad (3.31)$$

$$\dot{e}_{qv} = -\frac{1}{2} [e_{qs} I - Q(e_{qv})] J_o u. \quad (3.32)$$

Lembrando que  $J J_w^\dagger = I$  chega-se as seguintes relações:

$$J_p J_w^\dagger = [I \ 0], \quad (3.33)$$

$$J_o J_w^\dagger = [0 \ I]. \quad (3.34)$$

Então, substituindo a lei de controle cinemático (3.28) com  $v = [v_p \ v_o]^\top$  em (3.30) tem-se:

$$\dot{e}_p = \dot{p}_d - J_p J_w^\dagger \begin{bmatrix} v_p \\ v_o \end{bmatrix} - J_p \left( I - J_w^\dagger \begin{bmatrix} J_p \\ J_o \end{bmatrix} \right) W^{-1} \sigma, \quad (3.35)$$

e de forma análoga, aplicando a lei de controle cinemático (3.28) em (3.31) e (3.32) tem-se:

$$\begin{aligned} \dot{e}_{qs} &= \frac{1}{2} e_{qv}^\top J_o J_w^\dagger \begin{bmatrix} v_p \\ v_o \end{bmatrix} + \frac{1}{2} e_{qv}^\top J_o \left( I - J_w^\dagger \begin{bmatrix} J_p \\ J_o \end{bmatrix} \right) W^{-1} \sigma, \\ \dot{e}_{qv} &= -\frac{1}{2} [e_{qs} I - Q(e_{qv})] J_o J_w^\dagger \begin{bmatrix} v_p \\ v_o \end{bmatrix} - \frac{1}{2} [e_{qs} I - Q(e_{qv})] J_o \left( I - J_w^\dagger \begin{bmatrix} J_p \\ J_o \end{bmatrix} \right) W^{-1} \sigma. \end{aligned}$$

Utilizando as relações (3.33) e (3.34) chega-se finalmente a (i)  $\dot{e}_p = \dot{p}_d - v_p$ , (ii)  $\dot{e}_{qs} = \frac{1}{2} e_{qv}^\top v_o$  e (iii)  $\dot{e}_{qv} = -\frac{1}{2} [e_{qs} I - Q(e_{qv})] v_o$ .

Então, diferenciando (3.29) com respeito ao tempo ao longo das trajetórias do sistema e utilizando as relações (i) para a lei de controle de posição (3.14) e (ii)-(iii) para a lei de controle de orientação (3.18) tem-se que:

$$\dot{V} = -e_p^\top K_p e_p - e_{qv}^\top K_o e_{qv} \leq 0, \quad (3.36)$$

onde  $K_p = K_p^\top > 0$  e  $K_o = K_o^\top > 0$ .

Uma vez que  $V$  é uma função definida positiva com derivada não-positiva tem-se que  $V(t) \leq V(0)$  e, que  $e_p$ ,  $e_{qs}$  e  $e_{qv}$  são limitados. Entretanto, a abordagem de conjuntos invariantes (*Princípio da Invariância de LaSalle*) não pode ser utilizada para determinar a convergência de  $e_p$  e  $e_{qv}$ , pois a dinâmica do erro do sistema completo é não-autônoma. Para aplicar o *Lema de Barbalat* (Khalil 2002), é preciso verificar se  $\dot{V}$  é uniformemente contínua. Isso pode ser realizado analisando a derivada de  $\dot{V}$ :

$$\ddot{V} = -2 e_p^\top K_p \dot{e}_p - 2 e_{qv}^\top K_o \dot{e}_{qv}. \quad (3.37)$$

A partir da dinâmica do erro (3.30) e (3.32) expressas em termos de  $J_p$  e  $J_o$  e sabendo que, de acordo com a propriedade (P1),  $J$  é limitado, conclui-se que  $\dot{e}_p$  e  $\dot{e}_{qv}$  são limitados. Isso demonstra que  $\ddot{V}$  é limitado e conseqüentemente  $\dot{V}$  é uniformemente contínua. Então, aplicando o *Lema de Barbalat* (Khalil 2002) implica que  $\lim_{t \rightarrow \infty} e_p(t) = 0$ ,  $\lim_{t \rightarrow \infty} e_{qv}(t) = 0$  e conseqüentemente  $\lim_{t \rightarrow \infty} e_{qs}(t) = \pm 1$ , provando a estabilidade assintótica quase-global do sistema de controle em malha fechada.  $\square$

### Definição da Restrição Adicional

Para uma utilização eficiente do graus de mobilidade redundantes, a forma como o vetor  $\sigma$  é definido depende do objetivo secundário de interesse. Uma escolha típica de  $\sigma$  é dada por (Siciliano et al. 2009)

$$\sigma = k_o \left( \frac{\partial w(\theta)}{\partial \theta} \right)^\top, \quad (3.38)$$

onde  $k_o > 0$  e  $w(\theta)$  é uma função objetivo contínua em termos dos ângulos das juntas. Uma vez que a solução move-se ao longo da direção do gradiente da função objetivo, ela tenta maximizá-la localmente com respeito ao objetivo primário (*e.g.*, controle cinemático). Neste contexto, típicas funções objetivo são:

1. *Medida da Manipulabilidade:*

$$w(\theta) = \sqrt{\det(J(\theta) J^\top(\theta))}, \quad (3.39)$$

que decai para zero em uma configuração singular. Por isso, maximizando essa medida, a característica de redundância permite que o manipulador se movimente para longe das singularidades cinemáticas;

2. *Distância do limite mecânico das juntas:*

$$w(\theta) = -\frac{1}{2n} \sum_{i=1}^n \left( \frac{\theta_i - \bar{\theta}_i}{\theta_{iM} - \theta_{im}} \right)^2, \quad (3.40)$$

onde  $\theta_{iM}(\theta_{im})$  define o máximo (mínimo) limite da junta e  $\bar{\theta}_i$  é o seu valor médio. Ao maximizar essa distância, a redundância permite que o manipulador mantenha as suas juntas o mais perto possível do seu valor médio;

3. *Distância a um obstáculo:*

$$w(\theta) = \min_{p,p_o} \|p(\theta) - p_o\|, \quad (3.41)$$

onde  $p_o \in \mathbb{R}^3$  é o vetor posição de um ponto adequado localizado sobre o obstáculo (o seu centro, por exemplo, se o obstáculo é modelado como uma esfera) e  $p$  é o vetor posição de um ponto genérico na estrutura do manipulador. Ao maximizar essa distância, a redundância permite que o manipulador evite colisões com o obstáculo.

## 3.2 Desvio de Obstáculos

A utilização de robôs manipuladores redundantes para realizar tarefas enquanto interage com operadores humanos, é uma proposta que vem sendo estudada por diversos grupos de pesquisa nos últimos anos (Flacco et al. 2012, Haddadin et al. 2008, 2011). Neste contexto, a ideia é utilizar os graus de mobilidade não-redundantes do manipulador para realizar uma tarefa de prioridade primária (*e.g.*, rastreamento de trajetórias), enquanto que os graus de mobilidade redundantes são utilizados para realizar uma tarefa de prioridade secundária como, por exemplo, o desvio de obstáculos.

A maioria dos métodos de planejamento *off-line* baseados, por exemplo, em técnicas abordagens de retração, decomposição em células e/ou probabilística (Siciliano et al. 2009) são adequados para planejamento de movimentos *off-line* pois requerem um conhecimento prévio da geometria e da postura dos obstáculos no espaço de trabalho do robô. Esta hipótese é aceitável em diversos casos, por exemplo, quando um manipulador industrial movimenta-se em uma célula robotizada, que é um ambiente conhecido e estruturado. Técnicas *off-line* de planejamentos de rotas, como o método RRT (do inglês, *Rapidly-exploring Random Tree*), se projetadas de maneira cuidadosa, podem ser utilizadas de maneira *on-line* na presença de obstáculos dinâmicos e incertezas paramétricas (Bernardes et al. 2014). Entretanto, em tarefas de interação entre humanos e robôs, o robô deve ser capaz de planejar o seu movi-

mento *online*, usando informações plenas ou parciais do espaço de trabalho, obtidas durante a execução da tarefa, por meio de diversos tipos de sensores.

Uma abordagem eficiente para planejamento de movimento *online* baseia-se no conceito de que um determinado ponto do robô pode mover-se no espaço cartesiano sob a influência de campos potenciais artificiais. Neste contexto, campos atrativos conduzem o robô na direção de um ponto desejado, enquanto que campos repulsivos são criados ao redor de obstáculos para evitar colisões (Khatib 1986). A desvantagem dessa abordagem é a ocorrência de mínimos locais, onde o somatório de forças é nulo em uma posição que não é a mínima global. Funções de navegação com apenas um mínimo global são propostas em (Rimon & Koditschek 1992) para tratar o problema de mínimos locais em tarefas planares. Nos casos tridimensionais com a presença de obstáculos dinâmicos, a ocorrência de mínimos locais não é tão frequente quanto no caso planar. Neste trabalho, dois tipos de campos potenciais repulsivos são utilizados: um é definido em termos de uma restrição adicional onde as juntas são consideradas como os pontos de interesse para o cálculo da distância (Chung et al. 1997); o outro atua sobre o efetuador e modifica a lei de controle de posição, garantindo que o mesmo siga uma trajetória de referência livre de colisões (Flacco et al. 2012).

### 3.2.1 Desvio de Obstáculos para as Elos

Para um elo  $L_i$  e um obstáculo  $O_k$  arbitrário, o campo potencial repulsivo  $U_{L,O}(\theta)$  é definido por (Khatib 1986):

$$U_{L,O}(\theta) = \begin{cases} \frac{k_{L,O}}{2} \left( \frac{1}{\eta_{L,O}(\theta)} - \frac{1}{\eta_{lim}} \right)^2, & \eta_{L,O}(\theta) \leq \eta_{lim} \\ 0, & \eta_{L,O}(\theta) > \eta_{lim} \end{cases}$$

onde  $k_{L,O} > 0$  é um fator de ganho,  $\eta_{L,O}(\theta)$  é a distância entre o elo e o obstáculo e  $\eta_{lim}$  é o raio de influência do obstáculo ou a distância limite de detecção do obstáculo. Note que, quanto menor é  $\eta_{L,O}(\theta)$ , maior é a intensidade do campo potencial repulsivo. Dessa forma, se  $\eta_{L,O}(\theta) \rightarrow 0$ , implica que  $U_{L,O}(\theta) \rightarrow \infty$ . Então, a restrição adicional resultante  $\sigma$  pode ser definida por:

$$\sigma = \sum_{k=1}^{n_o} \nabla U_{L,O_k}(\theta), \quad (3.42)$$

onde  $n_o$  é o número de obstáculos e  $\nabla U_{L,O}(\theta)$  é o gradiente do potencial repulsivo

definido por:

$$\nabla U_{L,O}(\theta) = \begin{cases} -\frac{k_{L,O}}{\eta_{L,O}^2(\theta)} \left( \frac{1}{\eta_{L,O}(\theta)} - \frac{1}{\eta_{lim}} \right) \nabla \eta_{L,O}(\theta), & \eta_{L,O}(\theta) \leq \eta_{lim} \\ 0, & \eta_{L,O}(\theta) > \eta_{lim} \end{cases}$$

onde  $\nabla \eta_{L,O}(\theta)$  é o gradiente da distância e a força repulsiva resultante de  $U_{L,O}$  é dada por:

$$f_{L,O} = -\nabla U_{L,O}(\theta). \quad (3.43)$$

Na Figura 3.3 é apresentado um exemplo da magnitude da força repulsiva  $f_{L,O}$ .

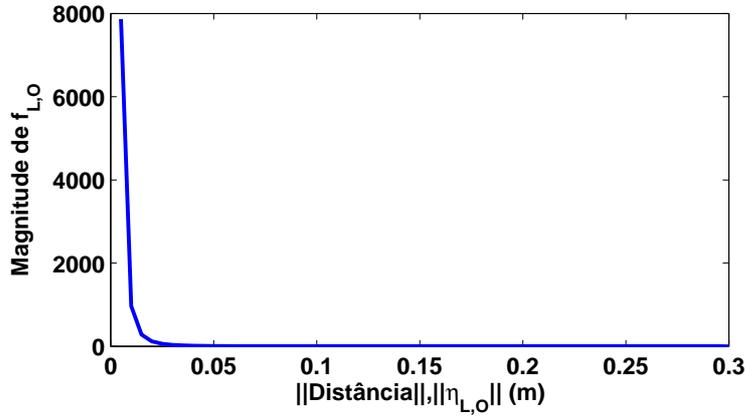


Figura 3.3: Magnitude da força repulsiva  $f_{L,O}$  com  $k_{L,O}=0,001$  e  $\eta_{lim}=\rho=0,3$  m.

### 3.2.2 Distância entre Elos e Obstáculos

Considere  $P_L$  um ponto arbitrário localizado em um elo  $L_i$  do robô, conectando dois pontos  $P_1$  e  $P_2$  localizados em duas juntas consecutivas  $J_1$  e  $J_2$  respectivamente. Agora, considere  $p_l = (x_l, y_l, z_l)$  a posição do ponto  $P_L$  expressa em um sistema de coordenadas inercial dada por:

$$\begin{aligned} x_l &= x_1 + \lambda(x_2 - x_1), \\ y_l &= y_1 + \lambda(y_2 - y_1), \\ z_l &= z_1 + \lambda(z_2 - z_1), \end{aligned} \quad (3.44)$$

onde  $p_1 = (x_1, y_1, z_1)$  e  $p_2 = (x_2, y_2, z_2)$  são as posições dos pontos  $P_1$  e  $P_2$  respectivamente, e  $\lambda > 0$  é uma constante paramétrica a ser definida. A distância entre um obstáculo pontual  $O_k$  e um elo  $L_i$  pode ser obtida pelo cálculo da distância entre esse obstáculo e o ponto mais próximo localizado no elo. A partir da minimização de (3.44), o valor de  $\lambda$  que corresponde ao ponto mais próximo do elo  $L_i$ , é obtido

por:

$$\lambda = -\frac{(p_1 - p_o) \cdot (p_2 - p_1)}{\|p_2 - p_1\|^2}, \quad (3.45)$$

onde  $p_o$  é a posição do obstáculo pontual  $O_k$ . A expressão para o valor de  $p_l$  varia de acordo com o valor de  $\lambda$ . Se  $\lambda \leq 0$ ,  $p_l = p_1$ , se  $\lambda \geq 1$ ,  $p_l = p_2$ , e nos demais casos,  $p_l$  é obtido de (3.44). Então, a distância resultante  $\eta_{L,O}$  é calculada em termos de  $\lambda$  como (Khatib 1986):

$$\eta_{L,O} = \begin{cases} \frac{\|(p_o - p_1) \times (p_o - p_2)\|}{\|p_2 - p_1\|} & , 0 < \lambda < 1 \\ \|(p_o - p_1)\| & , \lambda \leq 0 \\ \|(p_o - p_2)\| & , \lambda \geq 1. \end{cases} \quad (3.46)$$

### 3.2.3 Desvio de Obstáculos para o Efetuador

Seja  $U_{E,O}(\theta)$  o campo potencial repulsivo associado ao efetuador  $E$  e a um obstáculo  $O_k$  arbitrário. A força repulsiva correspondente a  $U_{E,O}(\theta)$  é definida por (Flacco et al. 2012)

$$f_{E,O} = -\nabla U_{E,O}(\theta) = v_{mag}(\eta_{E,O}) \frac{\eta_{E,O}(\theta)}{\|\eta_{E,O}(\theta)\|}, \quad (3.47)$$

em termos da direção de um vetor unitário  $\vec{\eta}_{E,O}(\theta)$  e da magnitude

$$v_{mag}(\eta_{E,O}) = \frac{v_{max}}{1 + e^{(\|\eta_{E,O}(\theta)\| - \rho)^{-1}b}}, \quad (3.48)$$

onde  $\eta_{E,O}(\theta)$  é a distância entre o efetuador e o obstáculo,  $v_{max}$  é a magnitude máxima,  $\rho$  é o raio de influência do obstáculo e  $b > 0$  é uma constante que define a inclinação da curva (vide Figura 3.4). Note que, quando  $\eta_{E,O}(\theta) \rightarrow 0$ ,  $v_{mag}(\eta_{E,O}) \rightarrow v_{max}/(1+e^{-b})$ , quando  $\eta_{E,O}(\theta) \rightarrow \rho$ ,  $v_{mag}(\eta_{E,O}) \rightarrow v_{max}/(1+e^b)$  e quando  $\eta_{E,O}(\theta) \rightarrow \bar{\rho}$ ,  $v_{mag}(\eta_{E,O}) \rightarrow 0$ , onde  $\bar{\rho} \in [\rho, \infty)$ .

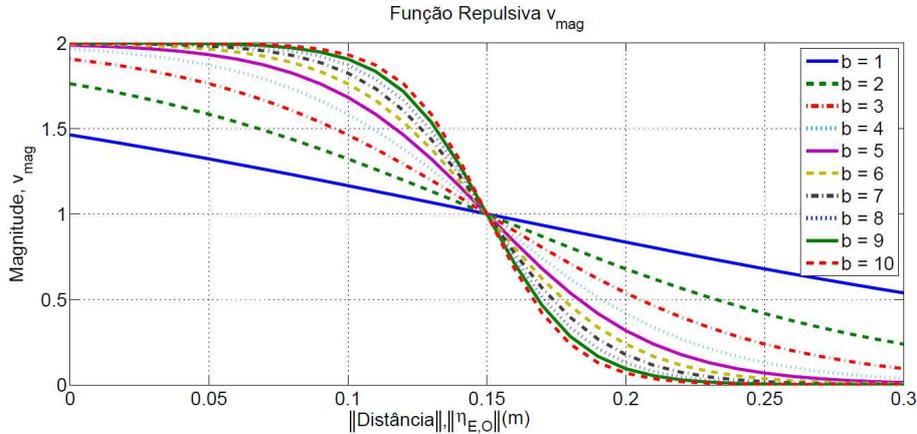


Figura 3.4: Magnitude da função repulsiva para  $b$  variando de 1 a 10, com  $v_{max} = 2 \text{ m s}^{-1}$  e  $\rho = 0,3 \text{ m}$ .

Então, considerando a contribuição de todos os obstáculos, a força repulsiva resultante é dada por:

$$f_t = \sum_{k=1}^{n_o} f_{E,O_k}. \quad (3.49)$$

Entretanto, na presença de múltiplos obstáculos o uso de (3.49) pode resultar em uma força repulsiva resultante com magnitude elevada. Isso significa que quanto maior o número de obstáculos  $n_o$  maior será a magnitude de  $f_t$ . Uma alternativa para evitar esse problema é obter a direção da força repulsiva resultante e considerar apenas a magnitude da força devido ao obstáculo mais próximo do efetuador  $v_{mag}(\eta_{E,O_{min}})$ . Dessa forma obtêm-se uma força repulsiva  $f_r$  com variações mais suaves (Flacco et al. 2012) descrita por:

$$f_r = -\nabla U_r = v_{mag}(\eta_{E,O_{min}}) \frac{f_t}{\|f_t\|}, \quad (3.50)$$

onde  $f_t/\|f_t\|$  é o vetor unitário na direção da força repulsiva resultante. Então, considerando-se o termo  $f_r$  como um vetor de velocidade repulsiva pode-se adicionar (3.50) na lei de controle de posição (3.14) como:

$$v_{p,r} = K_p e_p + \dot{p}_d + f_r, \quad (3.51)$$

garantindo que o efetuador realize a tarefa de interesse enquanto desvia de obstáculos presentes no seu espaço de trabalho. A limitabilidade do sinal de controle (3.51) pode ser discutida analisando-se as magnitudes do termo de velocidade repulsiva  $\nabla U_r$  e do termo proporcional ao erro de posição  $K_p e_p$ . Quando a distância entre o efetuador e o obstáculo diminui, a magnitude do campo repulsivo aumenta, e tenderá ao seu valor máximo se essa distância tender a zero, isto é,  $\eta_{E,O} \rightarrow 0$  e  $v_{mag}(0) \rightarrow v_{max}/(1 + e^{-b})$ . Neste caso, o termo de velocidade repulsiva prevalecerá sobre o termo proporcional ao erro de posição ( $K_p e_p \ll \nabla U_r$ ), e o efetuador se afastará do obstáculo. Por outro lado, quando a distância entre o efetuador e o obstáculo aumenta, a magnitude do campo repulsivo diminui e tenderá ao seu valor mínimo se essa distância tender ao raio de influência do obstáculo  $\rho$ , isto é,  $\eta_{E,O} \rightarrow \rho$  e  $v_{mag}(\rho) \rightarrow 0$ . Neste caso, o termo proporcional ao erro de posição prevalece sobre o termo de velocidade repulsiva ( $K_p e_p \gg \nabla U_r$ ), garantindo a convergência do erro de rastreamento, desde que outro obstáculo não se aproxime do efetuador.

### 3.2.4 Mudança da Direção do Vetor Repulsivo

De acordo com (3.50), verifica-se que a força repulsiva é baseada apenas na direção da distância mínima entre o efetuador e o obstáculo. Então, quando o obstáculo aproxima-se do efetuador com uma velocidade maior do que a capacidade

de movimento do robô, pode ser difícil evitar colisões retraindo-se o efetuador na mesma direção da velocidade do obstáculo.

Uma estratégia de reação mais efetiva, similar a reação humana, consiste em escapar da colisão movendo o efetuador em uma direção aproximadamente normal a velocidade do obstáculo. Esse efeito pode ser obtido utilizando-se a variação temporal da força repulsiva  $f_r$ , denotada por  $\dot{f}_r = df_r/dt$  para modificar a direção do vetor repulsivo na vizinhança de um ponto de interesse localizado no efetuador. O seguinte algoritmo proposto em (Flacco et al. 2012) modifica a direção da força repulsiva de acordo com a sua variação (Figura 3.5):

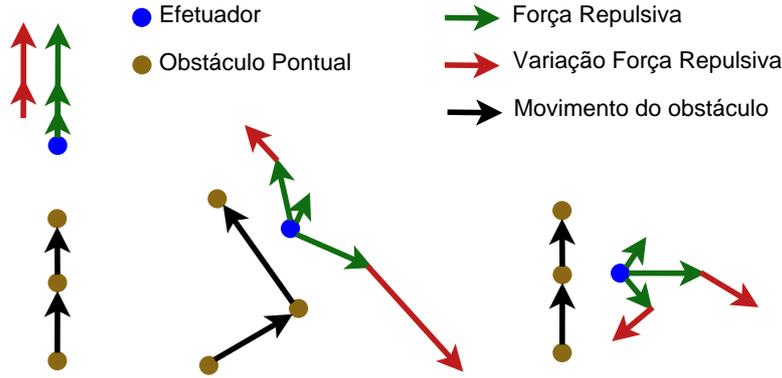


Figura 3.5: Força repulsiva e sua variação.

---

Algoritmo I - Mudança da Direção do Vetor Repulsivo

---


$$\vec{s} = \frac{\dot{f}_r(\eta_{E,O})}{\|\dot{f}_r(\eta_{E,O})\|}; \quad \vec{r} = \frac{f_r(\eta_{E,O})}{\|f_r(\eta_{E,O})\|}; \quad \beta = \arccos(\vec{s}^T \vec{r});$$

if  $\beta < \pi/2$

$$\vec{n} = \vec{s} \times \vec{r}; \quad \vec{t} = \vec{n} \times \vec{s} / \|\vec{n} \times \vec{s}\|;$$

$$\gamma = \beta + \frac{\beta - \pi/2}{1 + e^{-(\|\dot{f}_r(\eta_{E,O})\| (2/\dot{f}_{r,max}) - 1) c}};$$

$$f_{r,mod}(\eta_{E,O}) = \|f_r(\eta_{E,O})\| (\cos(\gamma) \vec{s} + \sin(\gamma) \vec{t});$$

else

$$f_{r,mod}(\eta_{E,O}) = f_r(\eta_{E,O});$$

end

---

onde  $\beta$  é o ângulo entre a força repulsiva e a sua variação. Note que, apenas quando o obstáculo está se aproximando do efetuador ( $\beta < \pi/2$ ), a força repulsiva é modificada. Quando  $f_r$  e  $\dot{f}_r$  estão alinhadas ( $\beta = 0$ ), o vetor  $\vec{n}$  é nulo e o vetor  $\vec{t}$  não é definido. Nesse caso,  $\vec{t}$  é obtido ao rotacionar  $\vec{U}_r$  de  $\pi/2$  radianos e a ortogonalidade é mantida.

A direção de  $f_{r,mod}$  é uma combinação da direção de  $f_r$  e da direção do vetor  $\vec{t}$ , que está no mesmo plano que  $f_r$  e  $\dot{f}_r$ . O ângulo  $\gamma$ , que define essa nova direção, é igual a  $\beta$  se  $\dot{f}_r = 0$ , e tende a  $\pi/2$  se  $\dot{f}_r \rightarrow \dot{f}_{r,max}$ , que é a máxima variação permitida

para o campo repulsivo. O escalar positivo  $c$  funciona como o escalar  $b$  em (3.48), e modifica a inclinação da função  $\|\dot{f}_r\|$ .

### 3.2.5 Comparação entre as Forças Repulsivas

Conforme foi apresentado, dois tipos de campos repulsivos diferentes denotados por  $U_{L,O}$  e  $U_{E,O}$  foram utilizados neste trabalho para o propósito de desvio de obstáculos. A decisão de não utilizar apenas um único tipo de campo repulsivo para as juntas e o efetuador deve-se, principalmente, a maneira em que eles são definidos e suas dimensões. O campo repulsivo  $U_{L,O}$  é definido no espaço das juntas e possui dimensão  $(n \times 1)$ , enquanto que o campo repulsivo  $U_{E,O}$  é definido no espaço Cartesiano e possui dimensão  $(3 \times 1)$ .

Além disso, conforme ilustrado na Figura 3.6 verifica-se que o campo repulsivo utilizado nas juntas é ilimitado para uma distância  $\eta_{L,O}$  nula e possui um valor elevado para valores próximos a essa distância. Isso significa que o sinal de controle é elevado quanto mais próximo de um obstáculo e o movimento não é suave. Caso o obstáculo esteja muito próximo da junta, uma saturação do atuador pode ocorrer.

Por outro lado, o campo potencial atuando no efetuador possui valor limitado quando a distância  $\eta_{E,O}$  se aproxima de 0 e sua forma garante que o movimento do efetuador seja suave e natural. Note que, movimentos bruscos do efetuador podem impedir uma interação humano-robô segura e eficiente.

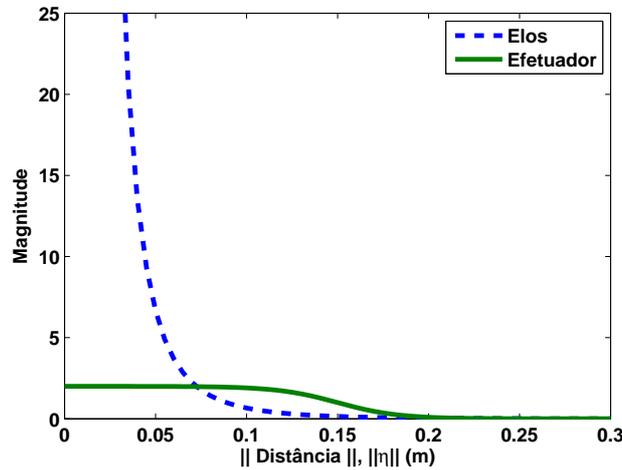


Figura 3.6: Comparação entre as forças repulsivas com  $k_{L,O} = 0,001$ ,  $v_{max} = 2 \text{ m s}^{-1}$ ,  $\eta_{lim} = \rho = 0,3 \text{ m}$  e  $b = 9$ .

### 3.3 Resultados de Simulação

Nesta seção, apresenta-se os resultados de simulação para avaliar a metodologia de controle cinemático e desvio de obstáculos para um manipulador redundante, considerando a presença de obstáculos no espaço de trabalho do robô. Assume-se aqui que os obstáculos, estáticos e dinâmicos, são perfeitamente detectados no cenário onde o robô está inserido.

As simulações foram realizadas no Matlab (The MathWorks, Inc.) utilizando o *toolbox* de robótica (Corke 2011) e os obstáculos pontuais foram implementados via código (Figura 3.7). Os obstáculos estáticos foram simulados próximos ao braço esquerdo do Motoman e os obstáculos dinâmicos foram simulados movendo-se próximo ao efetuador, paralelo ao plano da tarefa. A pose inicial do efetuador é  $p_0 = [0,816 \ 0,656 \ 1,337]^T m$ . Os parâmetros de controle utilizados são:

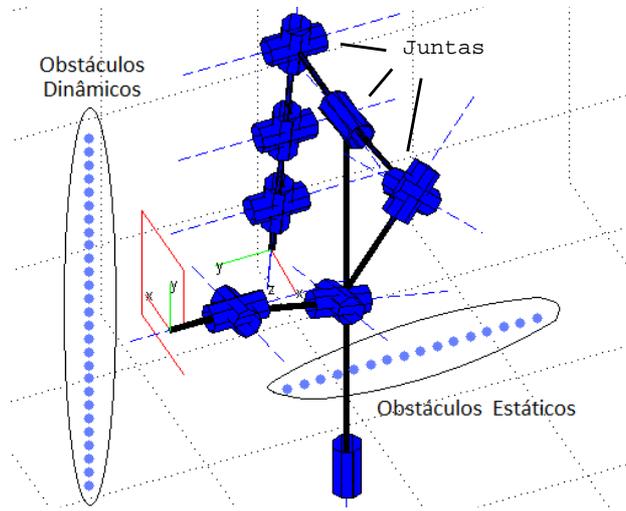


Figura 3.7: Manipulador redundante e obstáculos simulados no Matlab.

$K_p = 50I \ s^{-1}$ ,  $K_o = 0,05I \ rad \ s^{-1}$ ,  $k_{L,O} = 0,05$ ,  $\eta_{lim} = 0,6 \ m + \text{raio do elo}$ . Outros parâmetros utilizados são:  $v_{max} = 0,35 \ m \ s^{-1}$ ,  $\rho = 0,6 \ m$ ,  $b = 9$ ,  $\dot{V}_{r_{max}} = 1 \ m \ s^{-1}$ ,  $c = 9$ .

As simulações realizadas foram organizadas de acordo com os tipos de tarefa, objetivo de controle e campo potencial utilizado, conforme ilustrado na Tabela 3.1 a seguir.

Tabela 3.1: Descrição das simulações realizadas.

Simulação	Tarefa	Controle	Desvio
I	Movimento ponto-a-ponto	Posição	Elos/Efetuator
II	Movimento ponto-a-ponto	Posição/Orientação	Elos/Efetuator
III	Rastreamento de trajetória	Posição	Elos
IV	Rastreamento de trajetória	Posição/Orientação	Elos
V	Rastreamento de trajetória	Posição	Elos/Efetuator
VI	Rastreamento de trajetória	Posição/Orientação	Elos/Efetuator

Os mesmos obstáculos estáticos são utilizados para as simulações I e II, enquanto que as simulações III, IV, V e VI utilizam diferentes obstáculos estáticos.

### 3.3.1 Simulação I

A tarefa realizada nessa simulação consiste no movimento do efetuador através de quatro pontos de passagem (*waypoints*) formando uma trajetória retangular em um plano. Para simplificar, apenas o problema de controle de posição foi considerado.

A evolução no tempo da posição do efetuador durante a execução da tarefa na simulação I é apresentada na Figura 3.8 e a Figura 3.9 apresenta a evolução no tempo do erro de posição.

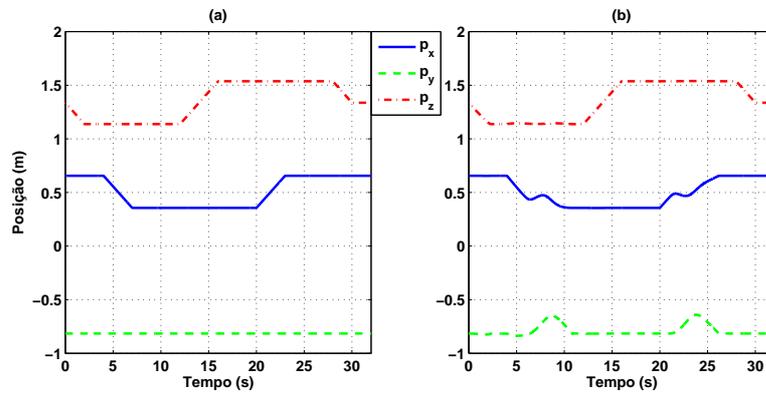


Figura 3.8: Simulação I - Evolução no tempo da posição do efetuador (a) sem desvio de obstáculos e (b) com desvio de obstáculos.

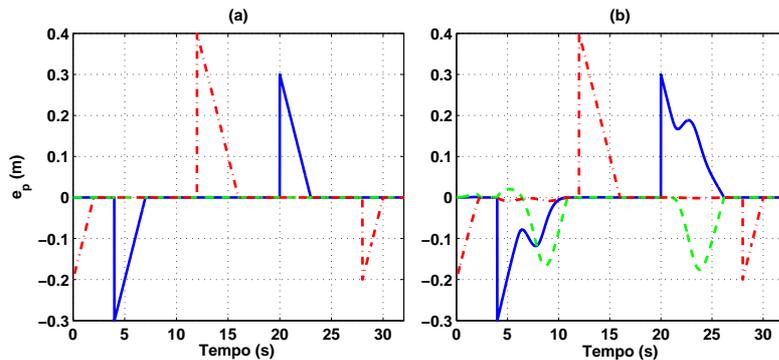


Figura 3.9: Simulação I - Erro de posição (a) sem desvio de obstáculos e (b) com desvio de obstáculos. Legenda:  $e_{px}$  (—),  $e_{py}$  (---),  $e_{pz}$  (-·-).

Na Figura 3.10 observa-se o comportamento do sinal de controle Cartesiano de posição  $v_p$  e o sinal de controle nas juntas para a simulação I é apresentado na Figura 3.11. A comparação entre a menor distância entre os elos e os obstáculos na simulação I é apresentada na Figura 3.12. Note que, a tarefa foi realizada com êxito e os *waypoints* foram alcançados, enquanto os elos se mantiveram mais afastados dos obstáculos.

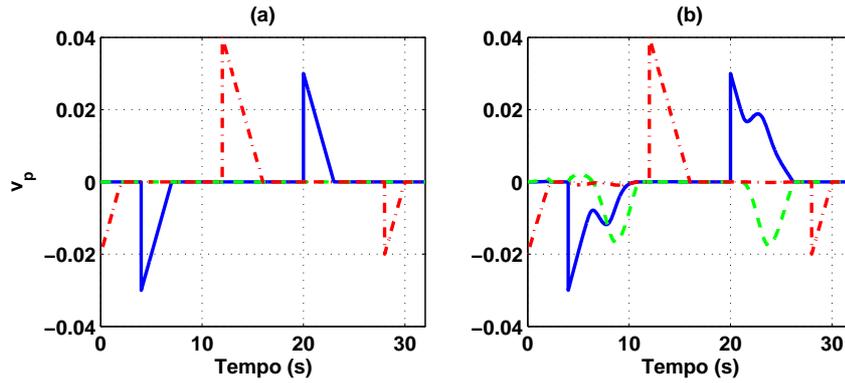


Figura 3.10: Simulação I - Sinal de controle Cartesiano de posição (a) sem desvio de obstáculos e (b) com desvio de obstáculos.

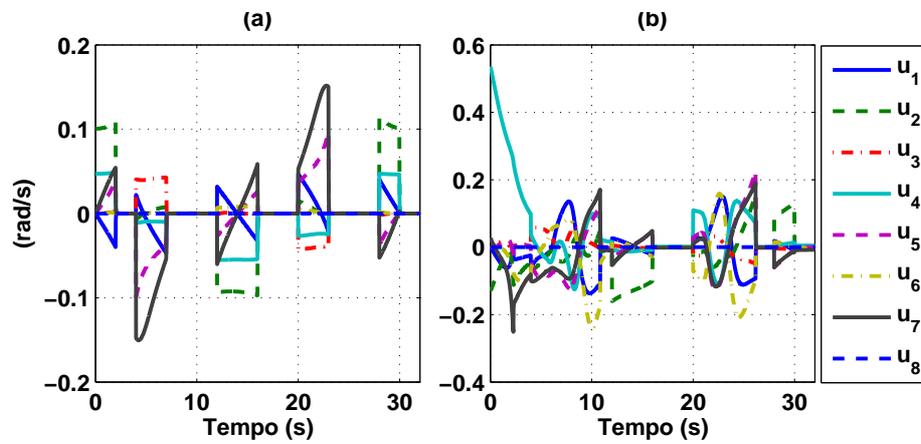


Figura 3.11: Simulação I - Sinal de controle nas juntas (a) sem desvio de obstáculos e (b) com desvio de obstáculos.

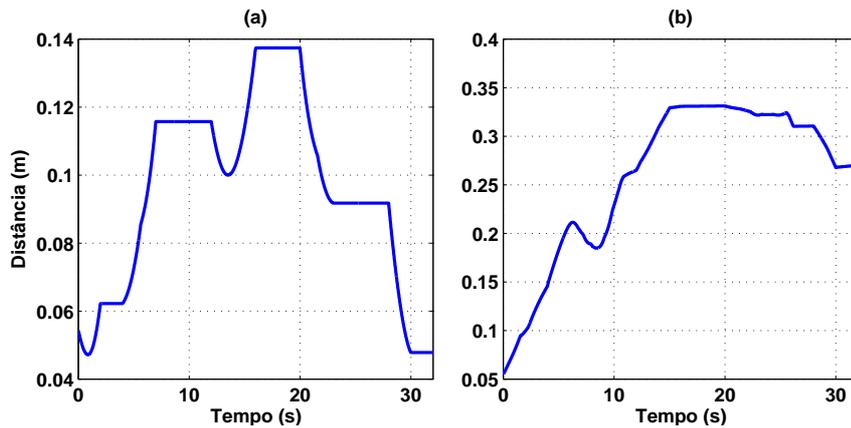


Figura 3.12: Simulação I - Menor distância entre os elos e os obstáculos (a) sem desvio de obstáculos e (b) com desvio de obstáculos.

### 3.3.2 Simulação II

A tarefa realizada nessa simulação consiste no movimento do efetuador através de quatro pontos de passagem (*waypoints*) formando uma trajetória retangular em

um plano. Porém, aqui considera-se o problema de controle de posição e orientação.

A evolução no tempo da posição do efetuador durante a execução da tarefa na simulação II é apresentada na Figura 3.13 e na Figura 3.14 observa-se o comportamento do erro de posição e da norma do erro de orientação respectivamente.

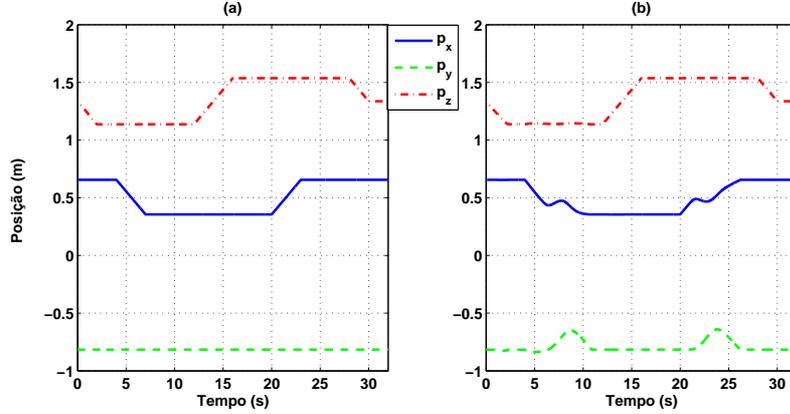


Figura 3.13: Simulação II - Evolução no tempo da posição do efetuador (a) sem desvio de obstáculos e (b) com desvio de obstáculos.

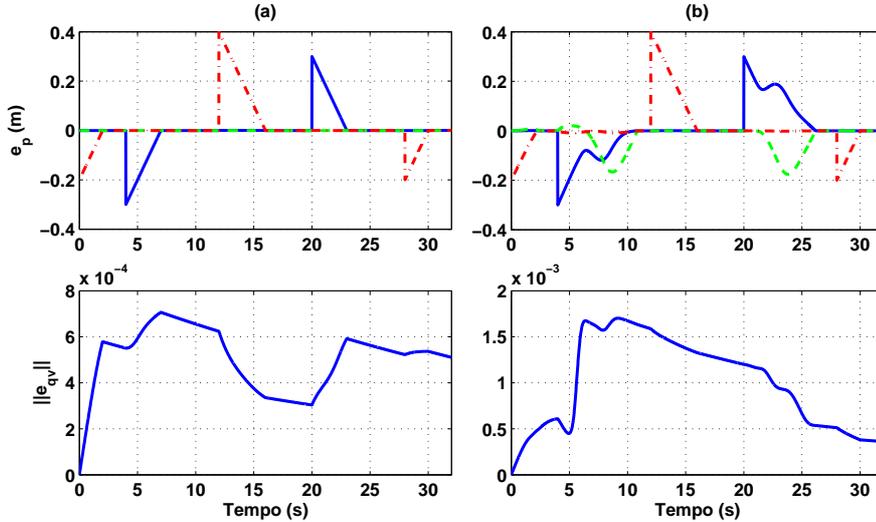


Figura 3.14: Simulação II - Erro de posição e norma do erro de orientação (a) sem desvio de obstáculos e (b) com desvio de obstáculos. Legenda:  $e_{px}$  (—),  $e_{py}$  (— —),  $e_{pz}$  (— ·).

Na Figura 3.15 observa-se o comportamento dos sinais de controle Cartesiano de posição  $v_p$  e orientação  $v_o$  respectivamente, enquanto que o sinal de controle nas juntas para a simulação II é apresentado na Figura 3.16. A comparação entre a menor distância entre os elos e os obstáculos na simulação II é apresentada na Figura 3.17. Note que, a tarefa foi realizada com êxito e os *waypoints* foram alcançados, enquanto os elos se mantiveram mais afastados dos obstáculos.

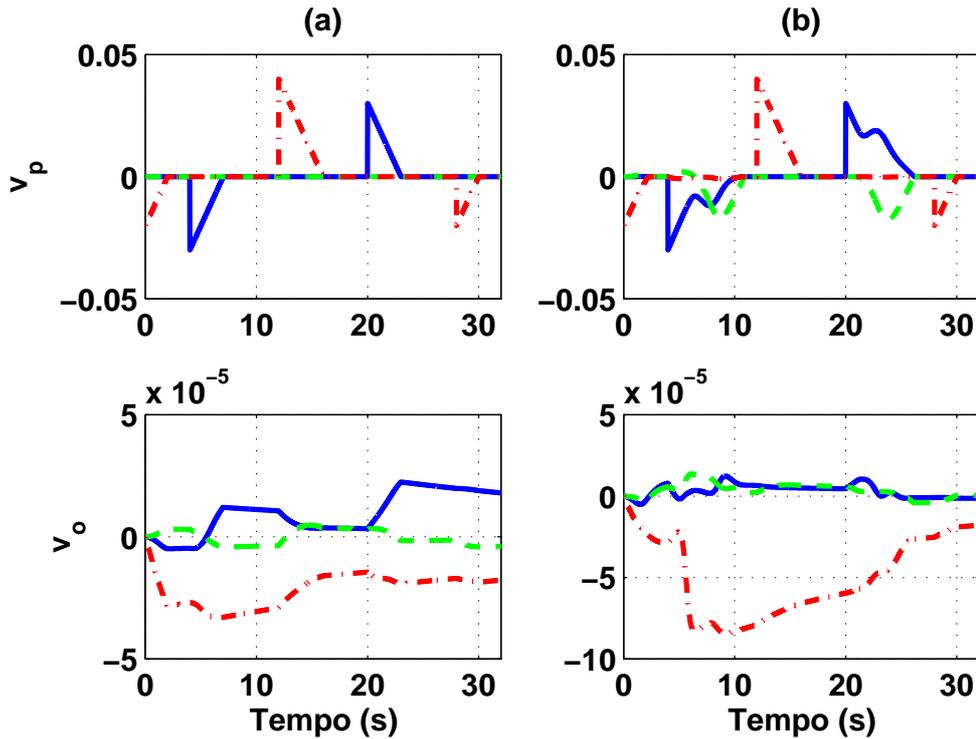


Figura 3.15: Simulação II - Sinal de controle Cartesiano de posição e orientação (a) sem desvio de obstáculos e (b) com desvio de obstáculos.

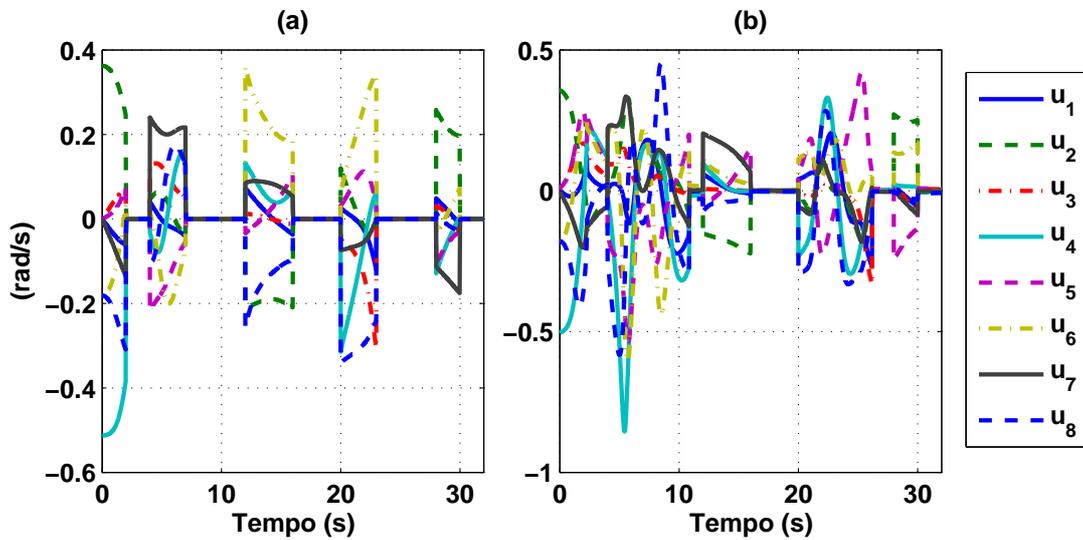


Figura 3.16: Simulação II - Sinal de controle nas juntas (a) sem desvio de obstáculos e (b) com desvio de obstáculos.

Em comparação à simulação I, devido a inclusão do controle de orientação, o manipulador tem menos liberdade para realizar a tarefa e evitar obstáculos e, dessa forma, os elos se afastam menos dos obstáculos.

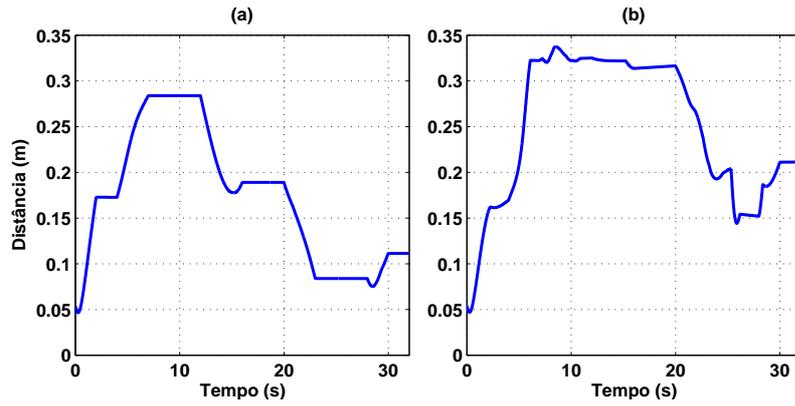


Figura 3.17: Simulação II - Menor distância entre os elos e os obstáculos (a) sem desvio de obstáculos e (b) com desvio de obstáculos.

### 3.3.3 Simulação III

A tarefa realizada nessa simulação consiste no rastreamento de uma trajetória circular em um plano. Aqui, apenas o problema de controle de posição e o desvio de obstáculos para os elos foram considerados. Quando um obstáculo se aproxima do efetuador a tarefa de rastreamento é interrompida e alterada para regulação da posição até o obstáculo se afastar do manipulador.

A Figura 3.18 apresenta a evolução no tempo do erro de posição para a simulação III e na Figura 3.19 observa-se o comportamento do sinal de controle Cartesiano de posição  $v_p$ . Note que picos ocorrem no erro de posição devido ao chaveamento da tarefa de rastreamento para regulação e vice versa.

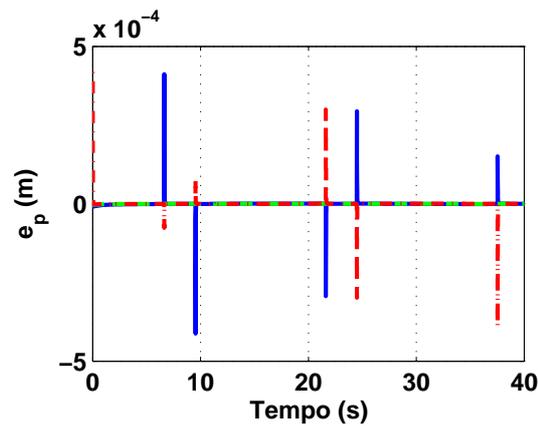


Figura 3.18: Simulação III - Erro de posição. Legenda:  $e_{px}$  (—),  $e_{py}$  (---),  $e_{pz}$  (-·-).

O sinal de controle nas juntas para a simulação III é apresentado na Figura 3.20 e na Figura 3.21 encontra-se o resultado do rastreamento de trajetória realizada pelo efetuador no espaço Cartesiano. A comparação entre a menor distância entre os elos e os obstáculos para a simulação III é apresentada na Figura 3.22.

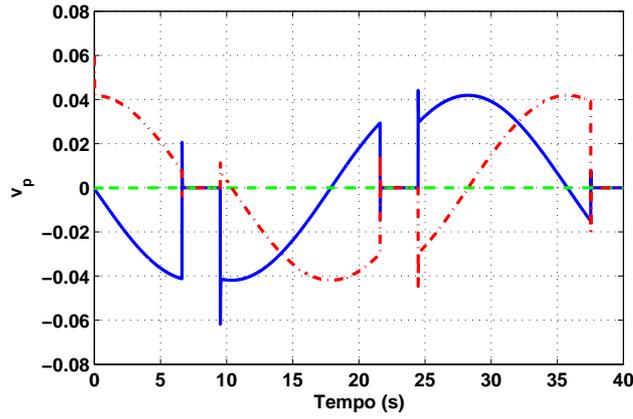


Figura 3.19: Simulação III - Sinal de controle Cartesiano de posição.

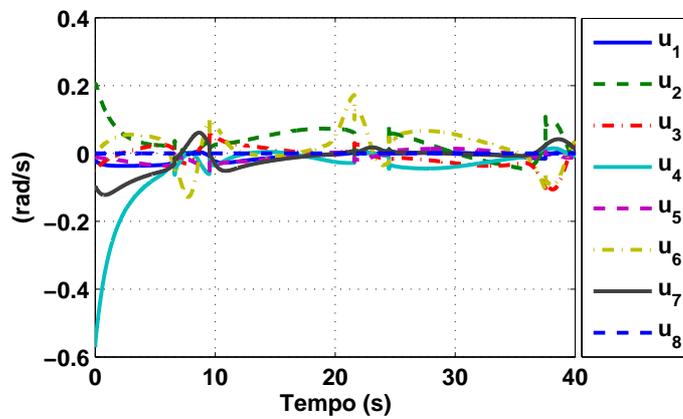


Figura 3.20: Simulação III - Sinal de controle nas juntas.

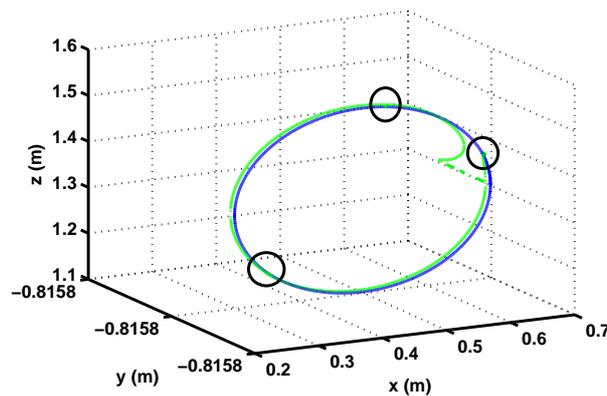


Figura 3.21: Simulação III - Rastreamento de trajetória realizado pelo efetuador. Legenda:  $p_d$  (—),  $p$  (---), pontos de chaveamento (O).

O rastreamento de trajetória foi realizado com êxito e os elos mantiveram-se mais afastadas dos obstáculos que para o mesmo caso sem desvio de obstáculos.

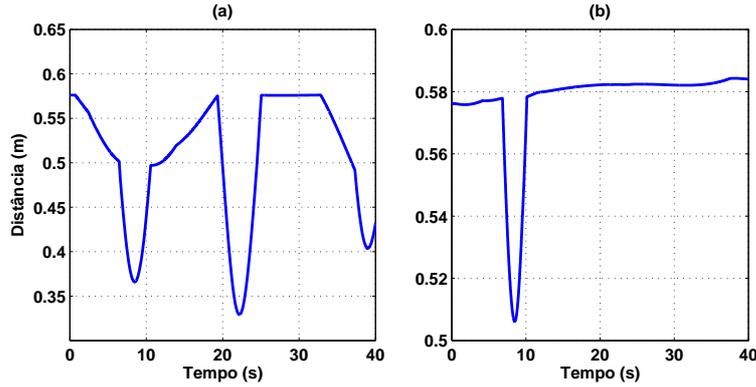


Figura 3.22: Simulação III - Menor distância entre os elos e os obstáculos (a) sem desvio de obstáculos e (b) com desvio de obstáculos.

### 3.3.4 Simulação IV

A tarefa realizada nessa simulação consiste no rastreamento de uma trajetória circular em um plano, ao mesmo tempo que realiza a regulação da orientação. Apenas o desvio de obstáculos para os elos foi considerado e quando um obstáculo se aproxima do efetuador a tarefa de rastreamento é modificada para regulação da posição até o obstáculo se afastar do manipulador.

A Figura 3.23 apresenta a evolução no tempo do erro de posição para a simulação IV e na Figura 3.24 observa-se o comportamento dos sinais de controle Cartesiano de posição  $v_p$  e de orientação  $v_o$ . Note que, alguns picos ocorrem no erro de posição devido ao chaveamento da tarefa de rastreamento para regulação e vice-versa.

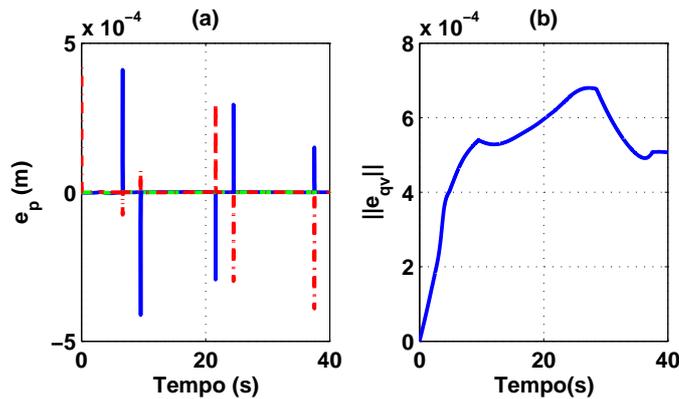


Figura 3.23: Simulação IV - Erro de posição e norma do erro de orientação. Legenda:  $e_{px}$  (—),  $e_{py}$  (— —),  $e_{pz}$  (— ·).

O sinal de controle nas juntas para a simulação IV é apresentado na Figura 3.25 e na Figura 3.26 encontra-se o resultado do rastreamento de trajetória realizado pelo efetuador no espaço Cartesiano. A comparação entre a menor distância entre os elos e os obstáculos para a simulação IV é apresentada na Figura 3.27.

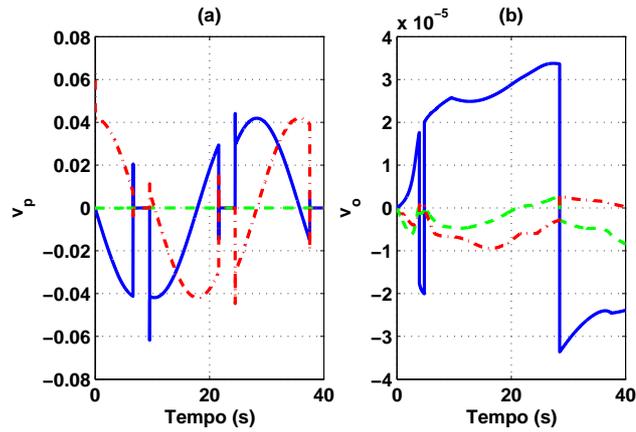


Figura 3.24: Simulação IV - Sinal de controle Cartesiano de posição e orientação.

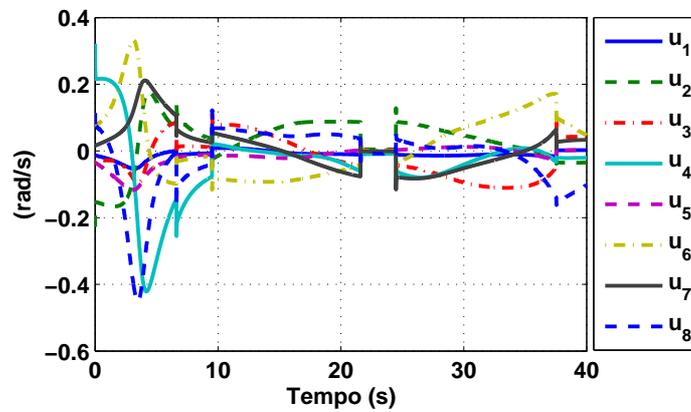


Figura 3.25: Simulação IV - Sinal de controle nas juntas.

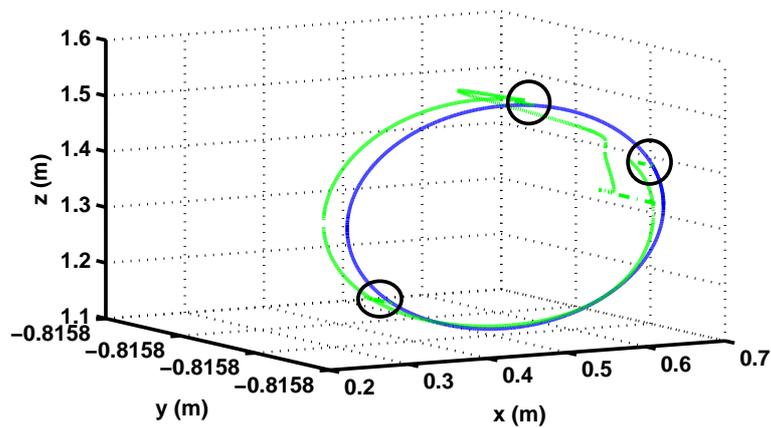


Figura 3.26: Simulação IV - Rastreamento de trajetória realizado pelo efetuador. Legenda:  $p_d$  (—),  $p$  (—·), pontos de chaveamento (O).

O rastreamento de trajetória foi realizado com êxito e os elos mantiveram-se mais afastados dos obstáculos que para o mesmo caso sem desvio de obstáculos. Entretanto os elos não se afastaram tanto quanto para o caso da simulação III.

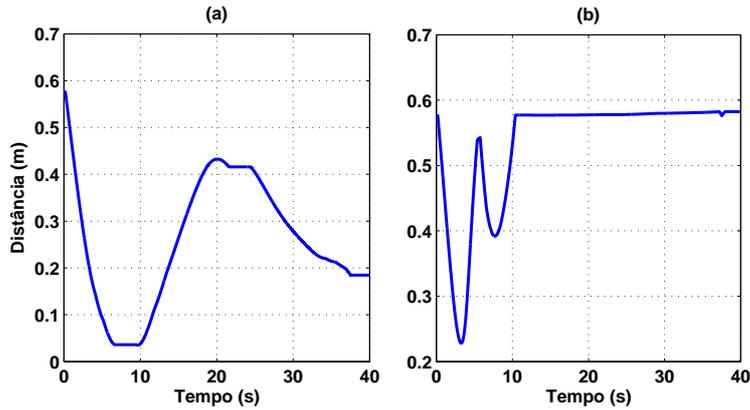


Figura 3.27: Simulação IV - Menor distância entre os elos e os obstáculos (a) sem desvio de obstáculos e (b) com desvio de obstáculos.

### 3.3.5 Simulação V

A tarefa realizada nessa simulação consiste no rastreamento de uma trajetória circular em um plano. Aqui, apenas o problema de controle de posição é considerado e o desvio de obstáculos para os elos e para o efetuador também é implementado. Quando um obstáculo se aproxima do efetuador a tarefa de rastreamento é modificada para regulação da posição até o efetuador evitar o obstáculo ou o mesmo se afastar do manipulador.

A Figura 3.28 apresenta a evolução no tempo do erro de posição para a simulação V e na Figura 3.29 observa-se o comportamento do sinal de controle Cartesiano de posição  $v_p$ . Note que, alguns picos ocorrem no erro de posição devido ao chaveamento da tarefa de rastreamento para regulação e vice-versa.

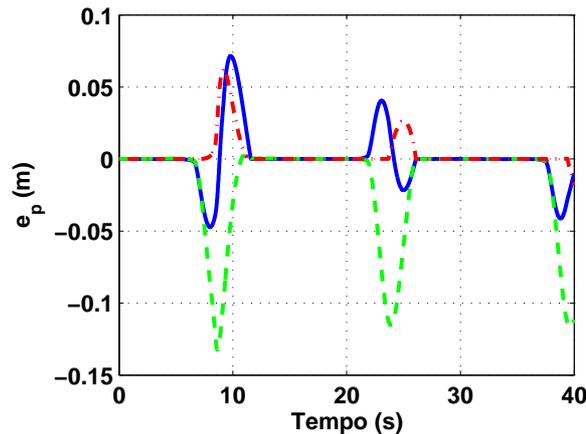


Figura 3.28: Simulação V - Erro de posição. Legenda:  $e_{px}$  (—),  $e_{py}$  (---),  $e_{pz}$  (-·-).

O sinal de controle nas juntas para a simulação V é apresentado na Figura 3.30 e na Figura 3.31 encontra-se o resultado para o rastreamento de trajetória realizado pelo efetuador no espaço Cartesiano.

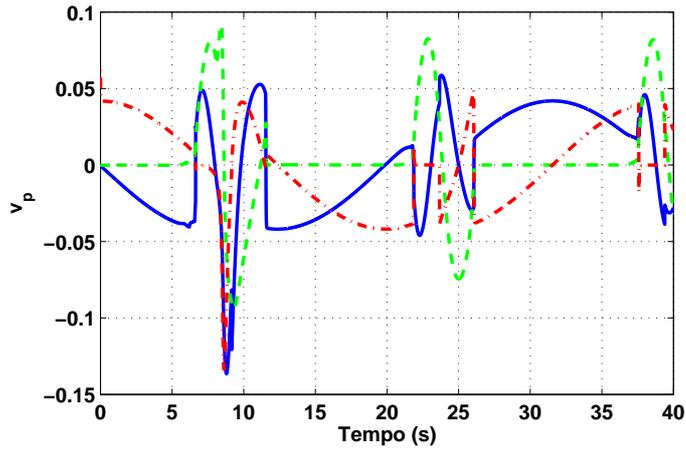


Figura 3.29: Simulação V - Sinal de controle Cartesiano de posição.

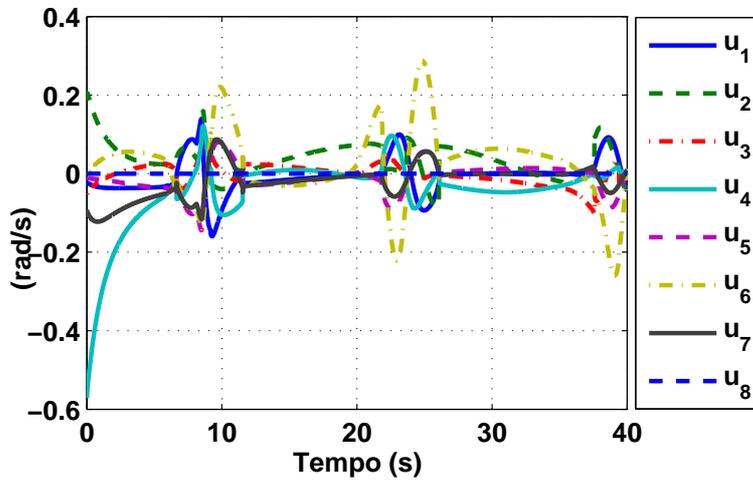


Figura 3.30: Simulação V - Sinal de controle nas juntas.

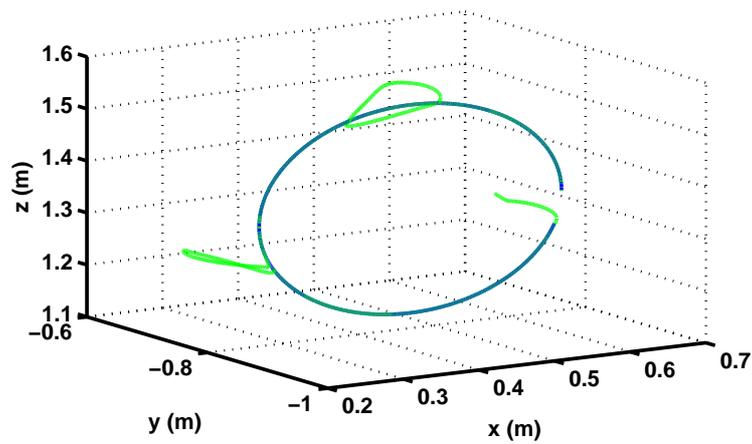


Figura 3.31: Simulação V - rastreamento de trajetória realizado pelo efetuador.  
 Legenda:  $p_d$  (—),  $p$  (—•).

A comparação entre a menor distância entre os elos e os obstáculos para a simulação V é apresentada na Figura 3.32.

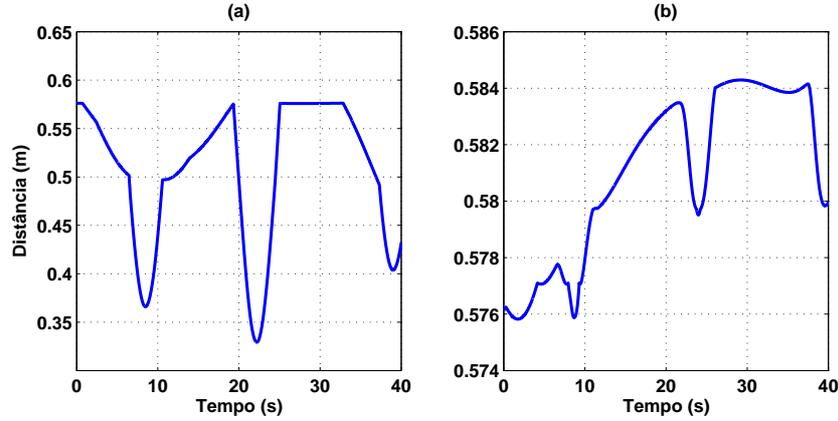


Figura 3.32: Simulação V - Menor distância entre os elos e os obstáculos (a) sem desvio de obstáculos e (b) com desvio de obstáculos.

O rastreamento de trajetória foi realizado com êxito e os elos mantiveram-se mais afastadas dos obstáculos que para o mesmo caso sem desvio de obstáculos. Além disso, com o acréscimo do termo de desvio de obstáculos para o efetuador, os elos se afastaram mais do que para o caso da simulação III.

### 3.3.6 Simulação VI

A tarefa realizada nessa simulação consiste no rastreamento de uma trajetória circular em um plano, ao mesmo tempo que realiza a regulação da orientação. Aqui, o problema de controle de posição e orientação é considerado bem como o desvio de obstáculos, para os elos e para o efetuador. Quando um obstáculo se aproxima do efetuador a tarefa de rastreamento é modificada para regulação da posição até o efetuador evitar o obstáculo ou o mesmo se afastar do manipulador.

A Figura 3.33 apresenta a evolução no tempo do erro de posição para a simulação VI e na Figura 3.34 observa-se o comportamento dos sinais de controle Cartesiano de posição  $v_p$  e de orientação  $v_o$ . Note que, alguns picos ocorrem no erro de posição devido ao chaveamento da tarefa de rastreamento para regulação e vice-versa. O sinal de controle nas juntas para a simulação VI é apresentado na Figura 3.35 e na Figura 3.36 encontra-se o rastreamento de trajetória realizado pelo efetuador. A comparação entre a menor distância entre os elos e os obstáculos para a simulação V é apresentada na Figura 3.37.

O rastreamento de trajetória foi realizado com êxito e os elos mantiveram-se mais afastadas dos obstáculos que para o mesmo caso sem desvio de obstáculos. Entretanto, nesse caso, o acréscimo do termo de desvio de obstáculos para o efetuador aparentemente não alterou a menor distância para os elos.

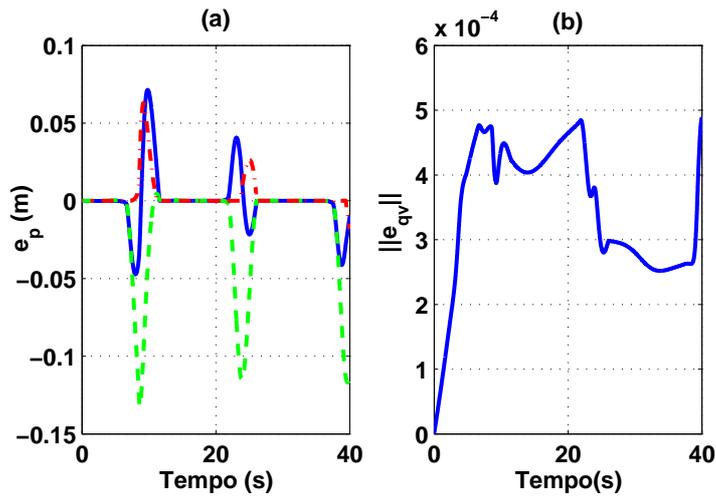


Figura 3.33: Simulação VI - Erro de posição e norma do erro de orientação. Legenda:  $e_{px}$  (—),  $e_{py}$  (---),  $e_{pz}$  (-·-).

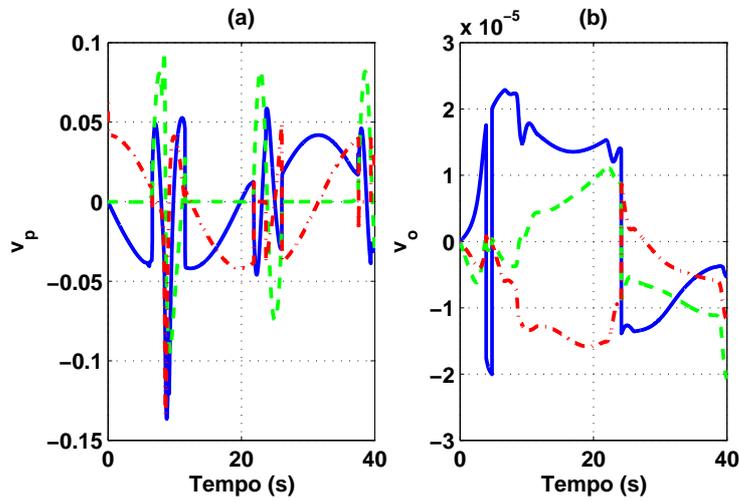


Figura 3.34: Simulação VI - Sinal de controle Cartesiano de posição e orientação.

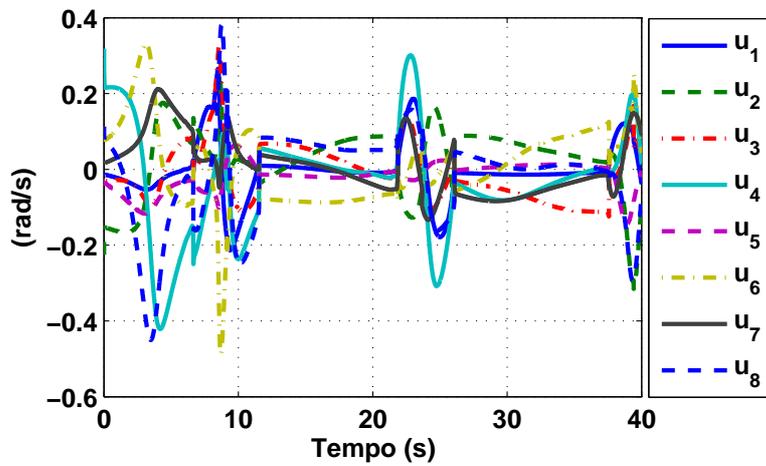


Figura 3.35: Simulação VI - Sinal de controle nas juntas.

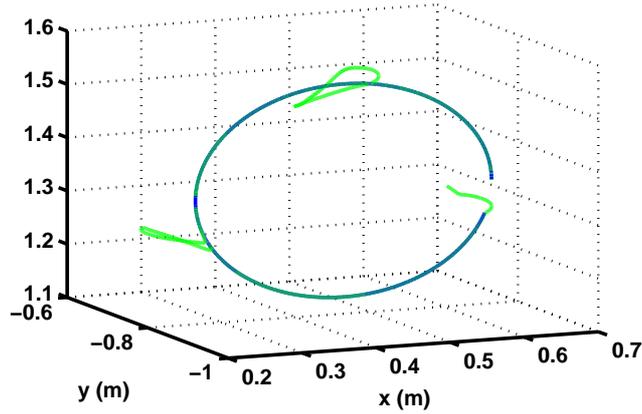


Figura 3.36: Simulação VI - Rastreamento de trajetória realizada pelo efetuador. Legenda:  $p_d$  (—),  $p$  (—•).

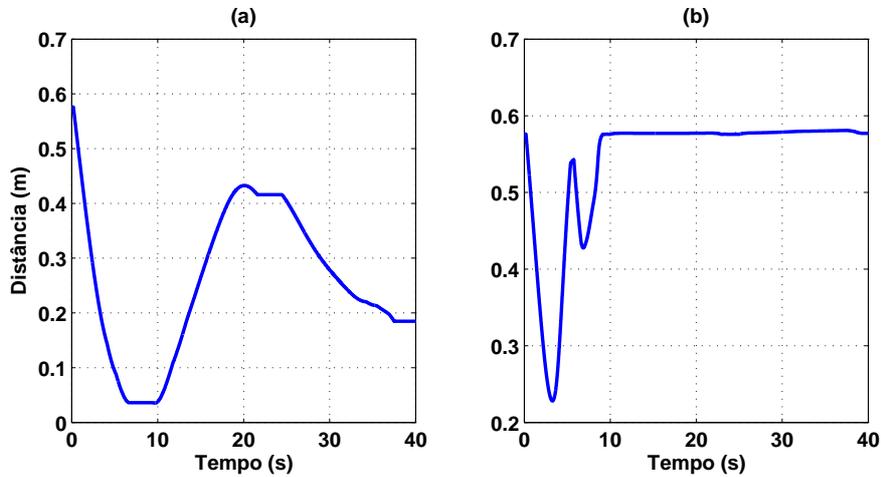


Figura 3.37: Simulação VI - Menor distância entre os elos e os obstáculos (a) sem desvio de obstáculos e (b) com desvio de obstáculos.

No caso em que os obstáculos se aproximam constantemente do robô, o efetuador se afasta até o obstáculo parar ou se afastar. Caso isso não ocorra, o manipulador se aproxima cada vez mais de uma configuração singular e, por questões de segurança, o movimento do robô é interrompido.

### 3.4 Conclusões

Neste capítulo, apresenta-se os conceitos e definições relacionados ao controle cinemático de posição e de orientação de um manipulador robótico. A análise de estabilidade e convergência do algoritmo de controle cinemático é realizada a partir da Teoria de Estabilidade de Lyapunov. Um método de desvio de obstáculos, baseado no uso de campos potenciais repulsivos atuando nos elos e no efetuador, é introduzido e uma análise das vantagens e desvantagens de cada campo é apresentada.

A estabilidade da lei de controle Cartesiana de posição com um termo repulsivo é discutida a partir da análise das magnitudes do termo de velocidade repulsiva e do termo proporcional ao erro de posição. Simulações são apresentadas para avaliar a viabilidade e o desempenho da metodologia proposta.

A partir da análise dos resultados de simulação conclui-se que as tarefas de movimento ponto-a-ponto e rastreamento na presença de obstáculos foram realizadas com êxito, validando a metodologia proposta. De acordo com os resultados, conclui-se também que o algoritmo de controle cinemático somente com o desvio de obstáculos para os elos é suficiente para evitar colisões e alcançar o objetivo de controle simultaneamente. Porém, a inclusão do desvio de obstáculos para o efetuador, em geral, faz com os elos se afastem mais dos obstáculos. Além disso, nesse caso, a segurança do operador passa a ter maior prioridade e a tarefa não é realizada até que o operador se afaste do manipulador a uma distância segura. Para o caso onde a execução da tarefa tem maior prioridade é recomendado evitar a aproximação de operadores do manipulador. Devido a presença de obstáculos dinâmicos, o problema de mínimos locais não foi recorrente e a tarefa foi realizada sem a existência de situações de equilíbrio de forças.

# Capítulo 4

## Detecção de Obstáculos

Neste trabalho, utiliza-se um sensor de profundidade de um dispositivo de interface natural Microsoft Kinect<sup>TM</sup> para detectar os obstáculos presentes no espaço de trabalho do robô, a partir de uma nuvem de pontos. Uma detecção de obstáculos adequada permite que o robô opere na presença de humanos de forma segura e eficiente. Uma vez que o algoritmo de detecção pode considerar o robô como um obstáculo, apresenta-se um método de delimitação dos elos e das juntas baseado no uso de figuras geométricas para remover o robô do cenário obtido pelo sensor. Para evitar a necessidade de calibração do Kinect no ambiente de operação um método de calibração remota baseado no uso de sensores internos e externos é introduzido para estimar os parâmetros extrínsecos do sensor de profundidade.

### 4.1 Calibração Remota

A qualidade da detecção de obstáculos no ambiente de operação depende do conhecimento prévio dos parâmetros de calibração do sensor. Em geral, os parâmetros intrínsecos são obtidos a partir da folha de dados do sensor que é disponibilizada pelo fabricante, mas os parâmetros extrínsecos são difíceis de obter, particularmente se o sensor está localizado em um ambiente remoto e de difícil acesso.

Como o sensor de profundidade do Kinect pode ser modelado como uma câmera *pin-hole*, os métodos de calibração *off-line* para câmeras de vídeo convencionais propostos na literatura podem ser utilizados para estimar os parâmetros intrínsecos do sensor. Em geral, esses métodos são baseados no uso de um tabuleiro de xadrez ou de alguma técnica de calibração interativa baseada no método de mínimos quadrados (Herrera C. et al. 2012, Rakprayoon et al. 2011). Por outro lado, os parâmetros extrínsecos do sensor – sua posição e orientação com respeito ao sistema de coordenadas inercial – podem ser obtidos a partir medições diretas *in-situ*, sujeitas a incertezas devido a erros aleatórios de medição.

Nesta seção, apresenta-se um método de calibração remota baseado no uso de

sensores internos e externos para estimar os parâmetros extrínsecos do sensor de profundidade. Os sensores interno e externo a serem considerados são os *encoders* das juntas de um robô manipulador e uma câmera RGB do dispositivo Kinect.

A ideia básica do método de calibração consiste em fixar um objeto alvo na extremidade do robô (e.g., no punho ou no efetuador) e extrair uma característica da imagem do alvo usando a câmera RGB, para diversas configurações de juntas ou posições no espaço Cartesiano, e de alguma forma obter os parâmetros extrínsecos do sensor de profundidade. Neste contexto, as seguintes hipóteses são consideradas:

- (H1) A cinemática direta do manipulador é conhecida;
- (H2) Os parâmetros intrínsecos da câmera e do sensor de profundidade são ambos conhecidos;
- (H3) O objeto alvo tem geometria esférica e permanece sempre dentro do campo de visão da câmera;

As hipóteses (H1) e (H2) são aceitáveis uma vez que os parâmetros cinemáticos do robô bem como os parâmetros intrínsecos da câmera e do sensor de profundidade podem ser encontrados na folha de dados dos respectivos fabricantes. A hipótese (H3) é formulada para que a área projetada no espaço da imagem seja invariante com respeito a rotações do objeto no espaço Cartesiano. Em geral, a matriz de parâmetros intrínsecos de uma câmera RGB é dada por (Siciliano et al. 2009):

$$K_v = \begin{pmatrix} f s_x & 0 & c_x \\ 0 & f s_y & c_y \end{pmatrix}, \quad (4.1)$$

onde  $f$  é a distância focal da câmera,  $(s_x, s_y)$  são os fatores de escalamento em *pixels* por milímetros e  $(c_x, c_y)$  são as coordenadas do centro do plano da imagem.

Seja  $p_b = (x_b, y_b, z_b)$  a posição do objeto alvo expressa no sistema de coordenadas da base do robô  $\bar{E}_b$ . Sem perda de generalidade, assume-se aqui que (i) o alvo está fixado no efetuador do robô e (ii) a posição do efetuador é mensurável por meio da equação de cinemática direta, isto é,  $p_b = k(\theta)$  onde  $\theta \in \mathbb{R}^n$  é o vetor de ângulos das juntas. A posição do objeto alvo expressa no sistema de coordenadas da câmera  $\bar{E}_c$  é denotada por  $p_c = (x_c, y_c, z_c)$ . Então, a relação entre os sistemas de coordenadas da base do robô  $\bar{E}_b$  e da câmera  $\bar{E}_c$  é descrita por:

$$p_b = p_{bc} + R_{bc} p_c, \quad (4.2)$$

onde  $p_{bc} \in \mathbb{R}^3$  é a posição relativa entre as origens dos sistemas de coordenadas e  $R_{bc} \in SO(3)$  é a matriz de rotação, descrevendo a orientação da câmera com respeito a base do robô.

Considere  $p_v = (x_v, y_v)$  e  $a_v \in \mathbb{R}^+$  a posição do centroide e a área projetada do objeto alvo, obtidas pela câmera RGB e expressas no sistema de coordenadas da imagem  $\bar{E}_v$ . A área projetada do objeto alvo  $a_v$  varia em função da profundidade  $z_c$ , ou da distância entre a câmera e o objeto, de acordo com a seguinte relação (Flandin et al. 2000):

$$\dot{a}_v = - \left( \frac{2 a_v}{z_c} \right) \dot{z}_c. \quad (4.3)$$

Seja  $a_{v_0} \in \mathbb{R}^+$  a área projetada conhecida de um objeto no tempo  $t_0$ , tal que  $a_{v_0} = a_v(t_0)$  corresponde a distância  $z_{c_0} = z_c(t_0)$ . Eliminando a dependência do tempo, pode-se integrar ambos os lados de (4.3) entre os intervalos  $[a_{v_0} a_v]$  e  $[z_{c_0} z_c]$  para obter a seguinte relação:

$$z_c = z_{c_0} \left( \frac{a_{v_0}}{a_v} \right)^{\frac{1}{2}}, \quad (4.4)$$

onde  $a_v$  é continuamente capturada pela câmera. Desta forma, pode-se obter a posição de um ponto genérico no sistema de coordenadas da câmera  $\bar{E}_c$  através de:

$$\begin{aligned} x_c &= ((x_v - c_x)z_c)/f s_x, \\ y_c &= ((y_v - c_y)z_c)/f s_y, \\ z_c &= z_{c_0} (a_{v_0}/a_v)^{\frac{1}{2}}. \end{aligned} \quad (4.5)$$

Agora, escolhendo uma sequência de configurações para as juntas do robô, pode-se obter um conjunto de pontos de calibração  $\mu_1, \mu_2, \dots, \mu_N$ . Note que, a posição de cada ponto de calibração  $\mu_i$  é mensurável com respeito ao sistema de coordenadas da base  $\bar{E}_b$  através da cinemática direta, e estimada com respeito ao sistema de coordenadas da câmera  $\bar{E}_c$  através da relação (4.5). Então, o seguinte sistema de equações pode ser obtido (Leite et al. 2012, Marques 2005):

$$\begin{aligned} p_{b\mu_1} &= p_{bc} + R_{bc} p_{c\mu_1}, \\ &\vdots \\ p_{b\mu_N} &= p_{bc} + R_{bc} p_{c\mu_N}. \end{aligned}$$

onde  $p_{b\mu_i} \in \mathbb{R}^3$  e  $p_{c\mu_i} \in \mathbb{R}^3$  para  $i = 1, 2, \dots, N$  denotam as posições dos pontos de calibração com respeito aos sistema de coordenadas  $\bar{E}_b$  e  $\bar{E}_c$  respectivamente.

Considere  $\mathbf{x}_{bc} = [p_{bc}^\top \ \phi_{bc}^\top]^\top$  o vetor de parâmetros extrínsecos da câmera, onde  $p_{bc} = [x_{bc} \ y_{bc} \ z_{bc}]^\top$  denota a posição relativa da origem do sistema de coordenadas  $\bar{E}_c$  expressa em  $\bar{E}_b$  e  $\phi_{bc} = [\zeta \ \iota \ \psi]^\top$  denota a orientação de  $\bar{E}_c$  em respeito a  $\bar{E}_b$  usando a parametrização de ângulos de Euler ZYX (Murray et al. 1994). Para determinar o vetor de parâmetros extrínsecos  $\mathbf{x}_{bc}$  usando um algoritmo de estimação baseado em um método de mínimos quadrados, como o Método de Newton, uma função

objetivo  $f$  é definida, descrevendo o sistema em termos dos parâmetros de posição e orientação:

$$f(p_{bc}, R_{bc}) = \begin{bmatrix} p_{b\mu_1} - p_{bc} - R_{bc} p_{c\mu_1} \\ \vdots \\ p_{b\mu_N} - p_{bc} - R_{bc} p_{c\mu_N} \end{bmatrix} \in \mathbb{R}^{3N}. \quad (4.6)$$

Note que, cada equação possui três coordenadas de posição e seis restrições para a orientação. Desta forma, o número mínimo de pontos necessários para realizar a calibração é três. Entretanto, como as medidas obtidas a partir de *encoders* e câmeras são contaminadas com ruído, é necessário medir diversos pontos de calibração para melhorar a acurácia dos elementos do vetor de parâmetros. Para minimizar a função objetivo (4.6), o Jacobiano da função  $f$  é utilizado:

$$J_f = \nabla f = \left[ \frac{\partial^T f}{\partial p_{bc}} \quad \frac{\partial^T f}{\partial \phi_{bc}} \right]^T. \quad (4.7)$$

Note que  $\frac{\partial f}{\partial p_{bc}} = -[I \cdots I]^T \in \mathbb{R}^{3N \times 3}$ . Então, para obter (4.7) é necessário calcular somente  $\frac{\partial f}{\partial \phi_{bc}}$  da seguinte forma:

$$\frac{\partial f}{\partial \phi_{bc}} = - \begin{bmatrix} \frac{\partial R_{bc}}{\partial \zeta} p_{c\mu_1} & \frac{\partial R_{bc}}{\partial l} p_{c\mu_1} & \frac{\partial R_{bc}}{\partial \psi} p_{c\mu_1} \\ \vdots & \vdots & \vdots \\ \frac{\partial R_{bc}}{\partial \zeta} p_{c\mu_N} & \frac{\partial R_{bc}}{\partial l} p_{c\mu_N} & \frac{\partial R_{bc}}{\partial \psi} p_{c\mu_N} \end{bmatrix}, \quad (4.8)$$

onde  $\frac{\partial R_{bc}}{\partial \zeta}$ ,  $\frac{\partial R_{bc}}{\partial l}$ ,  $\frac{\partial R_{bc}}{\partial \psi} \in \mathbb{R}^{3 \times 3}$  e  $\frac{\partial f}{\partial \phi_{bc}} \in \mathbb{R}^{3N \times 3}$ . O Método de Newton estima o vetor de parâmetros extrínsecos  $\mathbf{x}_{bc}$  em cada iteração  $k$  como:

$$\mathbf{x}_{bc}(k+1) = \mathbf{x}_{bc}(k) - \beta J_f^\dagger(k) f(p_{bc}(k), R_{bc}(k)), \quad (4.9)$$

com  $\beta > 0$  e  $J_f^\dagger(k) = (J_f^T(k) J_f(k))^{-1} J_f(k)^T$  sendo a pseudo-inversa a esquerda de  $J_f$ . Note que, como  $J_f \in \mathbb{R}^{3N \times 6}$ , a dimensão da matriz Jacobiana pode tornar-se muito grande se inúmeros pontos de calibração forem utilizados. Então, para evitar o mal-condicionamento numérico, estratégias de inversão de matrizes como, por exemplo, o algoritmo Levenberg-Marquardt (Motta et al. 2001) poderiam ser aplicadas. Além disso, não é possível garantir que o método iterativo (4.9) converge globalmente. Note que, ao invés da parametrização de ângulos de Euler, a representação por quatérnios unitários, ou outra representação de orientação, poderia ter sido utilizada, alterando a equação (4.8).

Uma vez que o vetor de parâmetros extrínsecos da câmera  $\mathbf{x}_{bc}$  é estimado, torna-se simples obter o vetor posição  $p_{bc}$  e a matriz de rotação  $R_{bc}$ . Entretanto, para obter uma descrição geométrica adequada do ambiente de operação utilizando o sensor de

profundidade deve-se obter os seus parâmetros de calibração extrínsecos, denotados por  $(R_{bd}, p_{bd})$ , uma vez que os parâmetros intrínsecos foram considerados previamente conhecidos. Uma representação do esquema de calibração remota com os sistemas de coordenadas e as transformações homogêneas envolvidas é a apresentado na Figura 4.1.

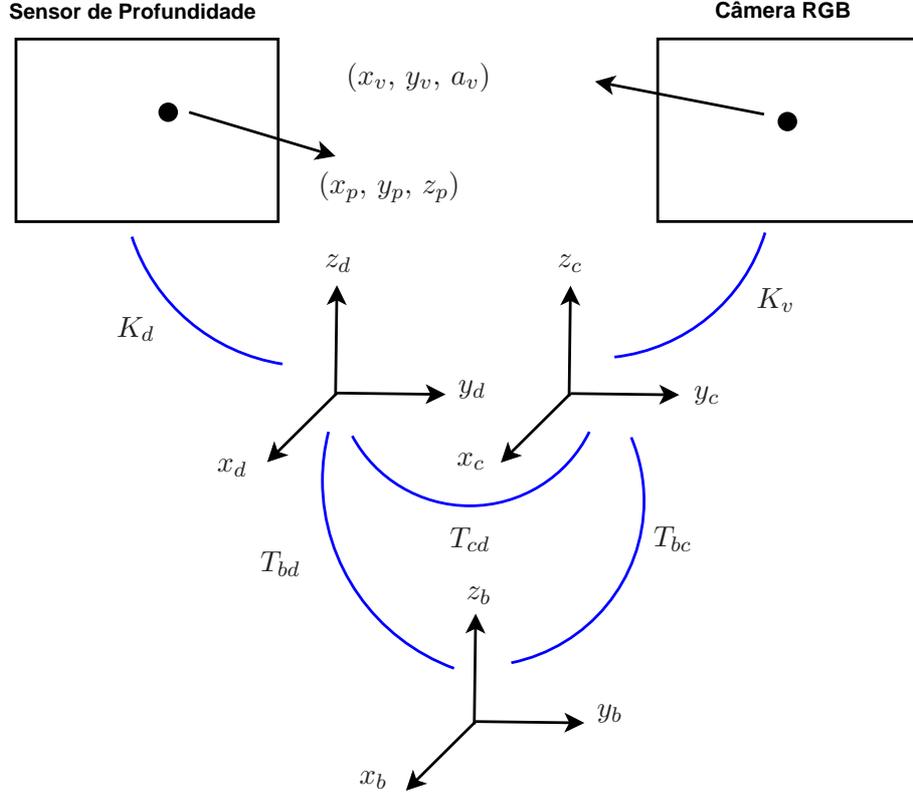


Figura 4.1: Representação dos sistemas de coordenadas utilizados no Método de Calibração Remota.

A partir de uma análise do dispositivo Microsoft Kinect<sup>TM</sup> sabe-se que os sistemas de coordenadas da câmera  $\bar{E}_c$  e do sensor de profundidade  $\bar{E}_d$  não são coincidentes. Porém, a matriz de transformação homogênea entre os seus sistemas de coordenadas, denotada por  $T_{cd}$  é conhecida. Então, pode-se obter a transformação homogênea entre o sistema de coordenadas do sensor de profundidade  $\bar{E}_d$  e o sistema de coordenadas da base do robô  $\bar{E}_b$ , através dos seguintes passos:

*Passo 1:* Estime  $p_{bc}$  e  $R_{bc}$  (via calibração remota) ;

*Passo 2:* Obtenha  $p_{cd}$  e  $R_{cd}$  (conhecidos);

*Passo 3:* Calcule  $p_{bd}$  e  $R_{bd}$  como:

$$p_{bd} = p_{bc} + R_{bc} p_{cd}, \quad (4.10)$$

$$R_{bd} = R_{bc} R_{cd}. \quad (4.11)$$

### 4.1.1 Simulação da Calibração Remota

Nesta seção, um resultado de simulação é apresentado para avaliar a viabilidade do método de calibração remota proposto. Considere uma câmera *pin-hole* posicionada em frente ao robô de tal forma que o mesmo esteja dentro do seu campo de visão. Então, assume-se que a posição e a orientação do sistema de coordenadas da câmera expressas com respeito ao sistema de coordenadas da base do robô são dadas respectivamente por:  $p_{bc} = [0, 2 \quad -2 \quad 1]^T \text{ mm}$  e  $\phi_{bc} = [0 \quad 0 \quad -\frac{\pi}{2}]^T \text{ rad}$ .

Para realizar a calibração remota utiliza-se 10 pontos de calibração  $\mu_i$ ,  $i = 1, \dots, 10$  onde suas respectivas posições expressas nos sistemas de coordenadas da base do robô e da câmera são denotadas respectivamente por  $p_{b\mu_i}$  e  $p_{c\mu_i}$ . Um ruído de medição de até 5% foi introduzido nas posições obtidas e o Método de Newton descrito por (4.9) foi utilizado, obtendo-se os seguintes valores aproximados para  $p_{bc}$  e  $\phi_{bc}$  após 50 iterações:  $p_{bc} = [0, 229 \quad -2, 008 \quad 0, 988]^T \text{ mm}$  e  $\phi_{bc} = [0, 010 \quad 0, 003 \quad -1, 566]^T \text{ rad}$ . Note que, os valores obtidos a partir do algoritmo de calibração remota estão bem próximos dos valores de referência. A norma do erro entre a posição estimada e a posição de referência foi de aproximadamente  $0,032 \text{ mm}$ . Por outro lado, a norma do erro entre a orientação estimada e a orientação de referência foi de aproximadamente  $0,011 \text{ rad}$ . Além disso, quanto maior é o nível de ruído considerado pior é a qualidade da estimação, e quanto maior é o número de pontos de calibração melhor será a qualidade da estimação. Na Figura 4.2 apresenta-se a convergência dos valores estimados para a posição  $p_{bc}$  e orientação  $\phi_{bc}$  após 50 iterações.

## 4.2 Obtendo a Descrição do Ambiente

Os dados obtidos pelo sensor de profundidade fornecem uma representação geométrica 2 1/2 D do ambiente, que consiste de uma tripla  $(x_p, y_p, z_p)$  onde o par  $(x_p, y_p)$  representa as coordenadas da projeção de um ponto cartesiano no plano e o elemento  $z_p$  representa a distância entre o ponto e o plano.

O sensor de profundidade pode ser modelado como uma câmera *pin-hole*, em termos de parâmetros intrínsecos de uma matriz  $K_d$  que modela a projeção de um ponto cartesiano no plano da imagem, e em termos dos parâmetros extrínsecos da matriz  $T_{bd}$ , calculada anteriormente. A matriz  $K_d$  é definida da forma:

$$K_d = \begin{pmatrix} f_d s_{xd} & 0 & c_{xd} \\ 0 & f_d s_{yd} & c_{yd} \\ 0 & 0 & 1 \end{pmatrix},$$

onde  $f_d$  é a distância focal da câmera,  $(s_{xd}, s_{yd})$  são fatores de escalamento em pixels

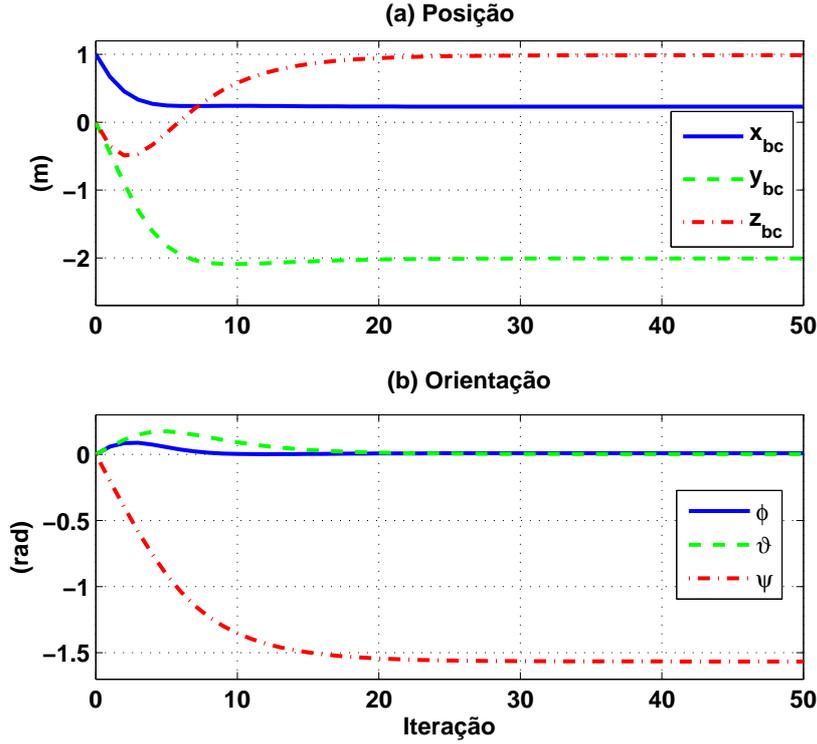


Figura 4.2: Valores estimados para a (a) posição e a (b) orientação do sistema de coordenadas da câmera com respeito ao sistema de coordenadas da base do robô.

por milímetros e  $(c_{xd}, c_{yd})$  são as coordenadas do centro do plano da imagem.

As coordenadas em pixel no plano da imagem e a profundidade de um ponto cartesiano genérico  $p_p$  são expressas no sistema de coordenadas do sensor como:

$$\begin{aligned} x_d &= ((x_p - c_{xd})z_p)/f s_{xd}, \\ y_d &= ((y_p - c_{yd})z_p)/f s_{yd}, \\ z_d &= z_p. \end{aligned}$$

As coordenadas  $(x_d, y_d, z_d)$  podem ser expressas no sistema de coordenadas da base do robô utilizando (4.11) e (4.10):

$$\begin{pmatrix} x_b \\ y_b \\ z_b \end{pmatrix} = R_{bd} \begin{pmatrix} x_d \\ y_d \\ z_d \end{pmatrix} + p_{bd}. \quad (4.12)$$

Um exemplo da obtenção da descrição do ambiente é apresentado na Figura 4.3, onde são apresentados os dados obtidos pelo Kinect (à esquerda), a nuvem de pontos no sistema de coordenadas do sensor (no centro) onde é possível notar a representação invertida da imagem devido a inversão do eixo  $y$ , e a nuvem de pontos no sistema de coordenadas do robô (à direita). Note que, como o sensor de profundidade do

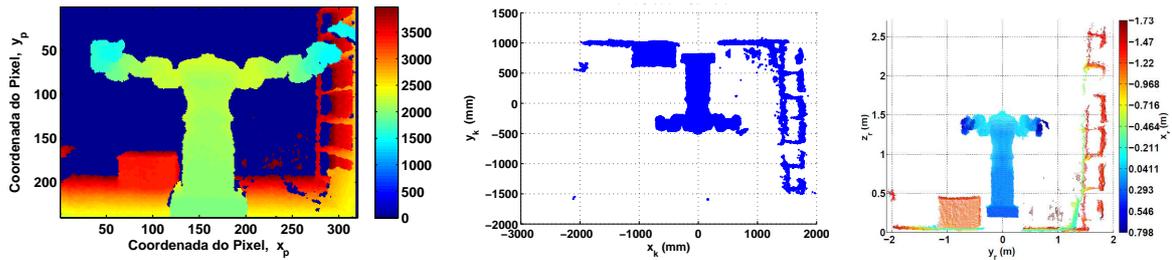


Figura 4.3: Dados obtidos pelo Kinect (esquerda). Nuvem de pontos no sistema de coordenadas do sensor (centro) nuvem de pontos no sistema de coordenada do robô (direita).

Kinect tem um alcance mínimo ( $0,4\text{ m}$ ) e máximo ( $4,0\text{ m}$ ), caso um objeto esteja localizado fora da faixa de detecção o valor da profundidade medida será zero. Dessa forma, esse objeto não pertencerá a nuvem de pontos e, portanto, não deverá ser considerado como um obstáculo.

Além disso, a nuvem de pontos cartesianos genéricos é composta por objetos próximos ou distantes do robô, bem como por elos e juntas. Então, quando o sensor de profundidade é inserido no ambiente de operação, o algoritmo de detecção pode considerar os elementos do robô como obstáculos. Neste contexto, um método será proposto para remover o robô do cenário obtido pelo sensor, delimitando os elos e as juntas por uma sequência de figuras geométricas.

### 4.3 Filtragem da Nuvem de Pontos

Nesta seção, uma metodologia é proposta para filtrar a nuvem de pontos obtida pelo sensor de profundidade, eliminando primeiramente os pontos relacionados as juntas e aos elos do robô, e em seguida removendo os pontos localizados fora do espaço de trabalho. É válido mencionar que o problema de auto-colisões pode ser evitado adicionando após a filtragem os pontos referentes aos elos que podem colidir internamente.

#### 4.3.1 Eliminando o Robô da Nuvem de Pontos

Para eliminar o robô do cenário obtido pelo sensor de profundidade, três métodos de delimitação das juntas e dos elos baseados em figuras geométricas são propostos: (i) Método das Esferas; (ii) Método dos Cilindros; e (iii) Método dos Elipsoides. Note que, para o algoritmo de filtragem, as juntas e os elos do robô são considerados como obstáculos, que devem ser removidos da nuvem de pontos.

## Método das Esferas

Nesse método, a ideia é delimitar todos os elos e as juntas do robô por uma sequência de esferas como visto na Figura 4.4 (Flacco et al. 2012), de acordo com os seguintes passos:

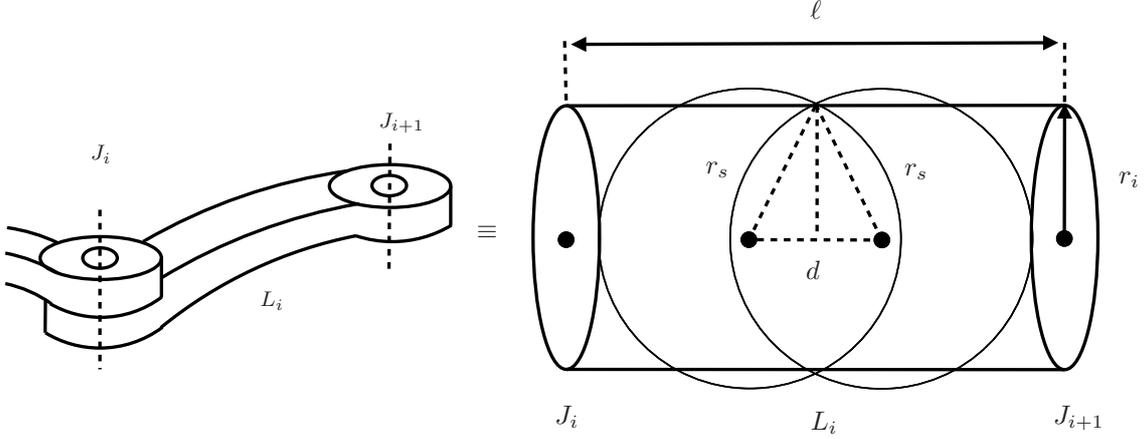


Figura 4.4: Método de remoção do robô baseado no uso de esferas.

*Passo 1:* Considere duas juntas consecutivas  $J_i$  e  $J_{i+1}$  conectadas por um elo  $L_i$  de comprimento  $\ell$  e raio  $r_i$ ;

*Passo 2:* Calcule a posição das juntas dos robôs por meio da equação de cinemática direta  $p_{J_i} = k(\theta_i)$  para  $i = 1, \dots, n$ ;

*Passo 3:* Defina pontos equidistantes dentro do elo  $L_i$  ao longo do menor segmento de reta interligando as juntas  $J_i$  e  $J_{i+1}$ . O número de pontos é definido por:

$$n_p = \left\lceil \frac{\ell}{d} \right\rceil + 1, \quad (4.13)$$

onde  $d = \left(\frac{\ell}{\lambda}\right)$  e  $\lambda \in \mathbb{N}^*$ ;

*Passo 4:* A partir do raio do elo  $r$  e da distância entre os pontos  $d$ , obtenha os raios das esferas  $r_s$  pelo Teorema de Pitágoras:

$$r_s^2 = \left(\frac{d}{2}\right)^2 + r_i^2. \quad (4.14)$$

*Passo 5:* Use as juntas e os pontos intermediários como centros de esferas (vide Figura 4.4) e crie esferas em todos os elos do robô;

*Passo 6:* Remova todos os pontos da nuvem localizados dentro das esferas.

Para facilitar o entendimento do método, o seguinte algoritmo de obtenção de esferas é apresentado resumindo os passos realizados (Algoritmo I).

---



---

### Algoritmo I: Filtragem baseado em Esferas

---



---

```

% Given the positions  $p_{J_i}$ , the radius of joints  $r_i$ 
and the vector with the obstacle positions  $p_o$  define:

filtering_by_spheres ( $p_{J_i}$ ,  $r_i$ ,  $p_o$ )
% Number of links
 $n = 8$ ;
% Number of points inside each link
 $\lambda = 5$ ;
for  $i = 1 : n - 1$ 
     $\ell = \|p_{J_{i+1}} - p_{J_i}\|$ ;     $d = \frac{\ell}{\lambda}$ ;
     $n_p = \frac{\ell}{d} + 1$ ;     $r_s = \sqrt{(\frac{d}{2})^2 + r_i^2}$ ;
    for  $k = 0 : \lambda$ 
         $\delta = p_{J_i} + \frac{k}{\lambda}(p_{J_{i+1}} - p_{J_i})$ ;
         $r_o = \|p_o - \delta\|$ ;    % Norm between  $\delta$  and  $p_o$ 
         $[j] = \text{find}(r_o < r_s)$ ;    % Find robot in the point cloud
        delete  $p_o(j)$ ;
    end
end
end

```

---



---

### Método dos Cilindros

Neste método, a ideia é delimitar todos os elos do robô por uma sequência de cilindros como visto na Figura 4.5 (Vogel et al. 2013), de acordo com os seguintes passos:

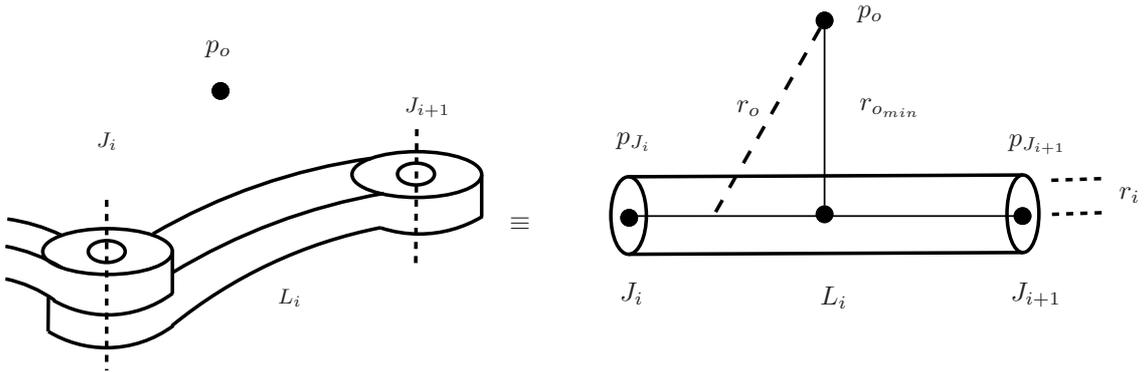


Figura 4.5: Método de remoção do robô baseado no uso de cilindros.

*Passo 1:* Considere duas juntas consecutivas  $J_i$  e  $J_{i+1}$  conectadas por um elo  $L_i$  de comprimento  $\ell$  e raio  $r_i$ ;

*Passo 2:* Calcule a posição das juntas dos robôs por meio da equação de cinemática direta  $p_{J_i} = k(\theta_i) = (x_i, y_i, z_i)$  para  $i = 1, \dots, n$ ;

*Passo 3:* Definir uma parametrização para os elos do robô:

$$\begin{aligned}x_{a_i} &= x_i + \lambda(x_{i+1} - x_i), \\y_{a_i} &= y_i + \lambda(y_{i+1} - y_i), \\z_{a_i} &= z_i + \lambda(z_{i+1} - z_i),\end{aligned}\tag{4.15}$$

onde  $x_{a_i}$ ,  $y_{a_i}$ ,  $z_{a_i}$  são as coordenadas de um ponto  $P_{A_i}$  no elo e  $\lambda \in \mathbb{R}$  é uma constante de parametrização a ser definida

*Passo 4:* Considerando  $p_{a_i} = [x_{a_i} \ y_{a_i} \ z_{a_i}]^\top$  a posição do ponto  $P_{A_i}$ , calcular a distância  $r_o = |p_{a_i} - p_o|$  entre o obstáculo  $o$  cuja posição é dada pelo vetor  $p_o = (x_o, y_o, z_o)$  e o ponto arbitrário  $P_{A_i}$  localizado no elo  $L_i$  (Weisstein 2013):

$$r_o = \sqrt{[(x_i - x_o) + \lambda(x_{i+1} - x_i)]^2 + [(y_i - y_o) + \lambda(y_{i+1} - y_i)]^2 + [(z_i - z_o) + \lambda(z_{i+1} - z_i)]^2}.$$

*Passo 5:* Calcular a distância mínima  $r_{o_{min}}$  entre o ponto arbitrário  $a_i$  e o obstáculo  $o$ . O valor do parâmetro  $\lambda$ , que define o ponto no elo mais próximo do obstáculo, é obtido a partir da minimização da equação de  $r_o$ . Então, calculando-se  $dr_o/d\lambda = 0$  e solucionando a equação resultante para  $\lambda$  obtém-se:

$$\lambda = -\frac{(p_{J_i} - p_o) \cdot (p_{J_{i+1}} - p_{J_i})}{\|p_{J_{i+1}} - p_{J_i}\|^2}.\tag{4.16}$$

**Observação 1.** Note que se o valor de  $\lambda$  for menor que 0, ou maior que 1, o ponto arbitrário  $P_{A_i}$  não se encontra no elo  $i$  e conseqüentemente o obstáculo não pertence ao próprio elo  $i$ .

A distância mínima é encontrada substituindo (4.16) na equação da distância  $r_o$  e, conseqüentemente, chega-se a:

$$r_{o_{min}}^2 = \frac{|(p_{J_i} - p_o)|^2 |(p_{J_{i+1}} - p_{J_i})|^2 - [(p_{J_i} - p_o) * (p_{J_{i+1}} - p_{J_i})]^2}{\|(p_{J_{i+1}} - p_{J_i})\|^2}\tag{4.17}$$

Utilizando a Identidade de Lagrange definida por  $(A \times B)^2 = A^2B^2 - (A \cdot B)^2$  no numerador de (4.17) e extraíndo-se a raiz quadrada da equação resultante, a relação (4.17) pode ser reescrita de uma forma simplificada:

$$r_{o_{min}} = \frac{\|(p_o - p_{J_i}) \times (p_o - p_{J_{i+1}})\|}{\|(p_{J_{i+1}} - p_{J_i})\|}\tag{4.18}$$

*Passo 6:* Após calcular a distância para todos os obstáculos, remover todos os pontos da nuvem onde  $r_{o_{min}}$  é menor que  $r$ .

---

---

**Algoritmo II: Filtragem baseado em Cilindros**

---

---

% Given the positions  $p_{J_i}$ , the radius of joints  $r_i$   
and the vector with the obstacle positions  $p_o$  define:

filtering\_by\_cylinders ( $p_{J_i}$ ,  $r_i$ ,  $p_o$ )

% Number of links

$n = 8$ ;

% Number of obstacles

$n_o = 10$ ;

for  $i = 1 : n - 1$

for  $k = 1 : n_o$ ,

$$\lambda = -\frac{(p_{J_i} - p_o) \cdot (p_{J_{i+1}} - p_{J_i})}{\|p_{J_{i+1}} - p_{J_i}\|^2};$$

if  $0 < \lambda < 1$

$$r_{o_{min}} = \frac{\|(p_o - p_{J_i}) \times (p_o - p_{J_{i+1}})\|}{\|(p_{J_{i+1}} - p_{J_i})\|};$$

if  $r_{o_{min}} < r_i$

delete  $p_o(i)$ ;

end

else

delete  $p_o(i)$

end

end

end

---

---

Para facilitar o entendimento do método, o seguinte algoritmo de obtenção de cilindros é apresentado resumindo os passos realizados (Algoritmo II).

### Método dos Elipsoides

Nesse método, a ideia é delimitar todos os elos e as juntas do robô por uma sequência de elipsoides, seguindo a seguinte equação:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1, \quad (4.19)$$

onde  $a$ ,  $b$  e  $c$  são os semi-eixos principais do elipsoide. Considerando  $a$  como o semi-eixo principal maior e  $b = c$  como semi-eixos menores, o elipsoide pode ser simplificado como uma elipse no plano, como visto na Figura 4.6. Então, os seguintes passos são realizados para delimitar o robô por uma sequência de elipsoides:

*Passo 1:* Considere duas juntas consecutivas  $J_i$  e  $J_{i+1}$  conectadas por um elo  $L_i$  de comprimento  $\ell$  e raio  $r_i$ ;



4. Finalmente, o semi-eixo  $b$  é obtido através das seguintes relações:  $\xi = \frac{P_2 F_2}{P_2 H}$  e  $\frac{f}{a} = \frac{r_i}{c_e - f}$ , resultando em:

$$b = \sqrt{\frac{r_i^2 + r_i \sqrt{r_i^2 + (d)^2}}{2}} \quad (4.23)$$

*Passo 5:* Agora, considerando a transformação homogênea de cada junta em relação ao sistema de coordenadas da base, denotada por  $T_{bi}$ , pode-se obter a nuvem de pontos em função de cada junta através da equação:

$$(p_o)_i = R_{bi}^T (p_o)_b - R_{bi}^T p_{bi}. \quad (4.24)$$

Dessa forma, a nuvem de pontos estará representada no sistema de coordenadas das juntas e a equação do elipsoide (4.19) pode ser utilizada. Transformando a posição da junta  $J_{i+1}$  para o sistema de coordenadas da junta  $J_i$ , pode-se descobrir a direção ( $x$ ,  $y$  ou  $z$ ) do elo e alterar a equação da elipse (4.19) de tal forma que o semi-eixo principal maior  $a$  esteja na direção do elo.

*Passo 6:* Remova todos os pontos da nuvem localizados dentro dos elipsoides, isto é, a equação (4.19) seja menor que 1.

Para facilitar o entendimento do método, o seguinte algoritmo de obtenção de elipsoides é apresentado resumindo os passos realizados (Algoritmo III).

### 4.3.2 Comparação entre os Métodos de Filtragem

É válido ressaltar que para delimitar completamente os elos e as juntas do robô a partir do método de remoção baseado em esferas (Figura 4.7) ou elipsoides (Figura 4.8) deve-se usar um grande número de pontos, implicando no aumento do custo computacional e na redução do espaço de trabalho livre de colisões. Além disso, os objetos localizados muito próximos do robô podem ser considerados como parte dele e, em geral, existe a sobreposição de esferas.

No caso do método de remoção baseado em cilindros (Figura 4.9) não existe uma grande redução no espaço de trabalho livre de colisões e existe um número menor de elementos geométricos, evitando sobreposições. Entretanto, o cálculo da parametrização demanda um alto custo computacional.

De acordo com os resultados práticos, o método de filtragem que exige menor custo computacional é o método das esferas devido a sua simplicidade de implementação. Nos experimentos de avaliação dos métodos das esferas e dos elipsoides, considera-se  $\lambda=5$  para todos os elos.

% Given the positions  $p_{J_i}$ , the radius of joints  $r_i$ , the homogenous transformations  $T_{bi}$  and the vector with the obstacle positions  $p_o$  define:

```

filtering_by_ellipsoids ( $p_{J_i}, r_i, p_o$ )
% Number of points inside each link
 $\lambda = 5$ ;
% Number of links
 $n = 8$ ;
% Number of obstacles
 $n_o = 10$ ;
 $\lambda = 5$ ;
for  $i = 1 : n - 1$ 
     $\ell = \|p_{J_{i+1}} - p_{J_i}\|$ ;     $d = \frac{\ell}{\lambda}$ ;
     $n_p = \frac{\ell}{d} + 1$ ;
     $a = \frac{r_i + \sqrt{r_i^2 + (d)^2}}{2}$ ;
     $b = \sqrt{\frac{r_i^2 + r_i \sqrt{r_i^2 + (d)^2}}{2}}$ ;
    for  $k = 1 : n_o$ 
         $p_o = R_{bi}^T p_o - R_{bi}^T p_{bi}$ ;
         $[j] = \text{find}(\frac{x_o^2}{a^2} + \frac{y_o^2}{b^2} + \frac{z_o^2}{b^2} < 1)$ ;    % In this case the link is aligned to the x-axis
        delete  $p_o(j)$ ;
    end
end
end

```

---



---

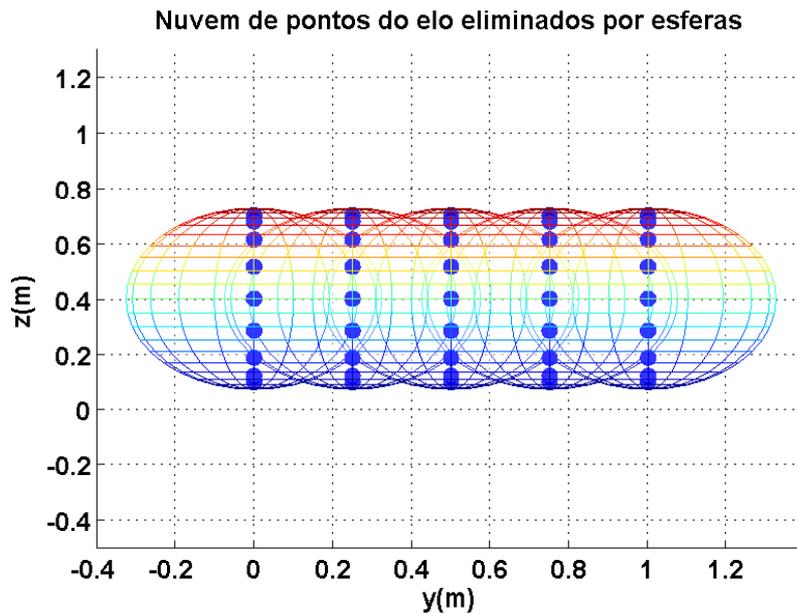


Figura 4.7: Esquematização do método das esferas.

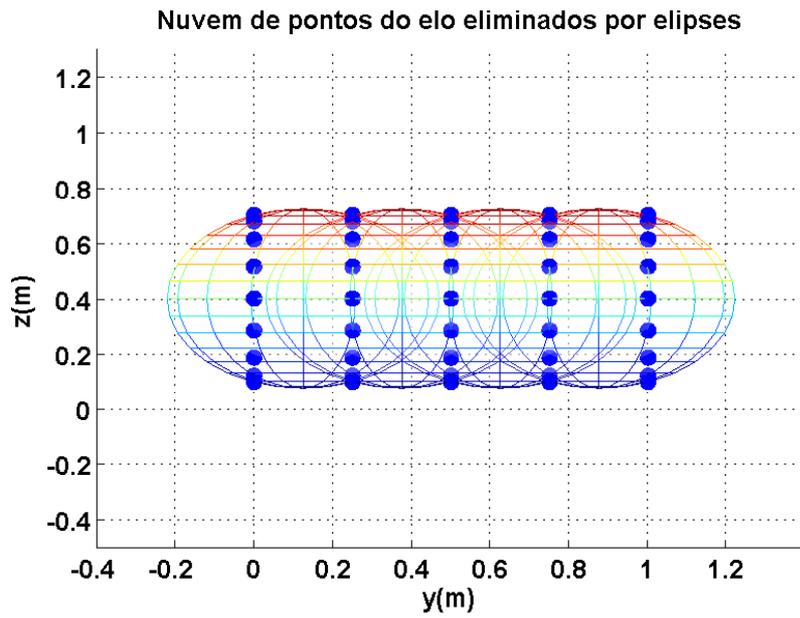


Figura 4.8: Esquematização do método dos elipsoides.

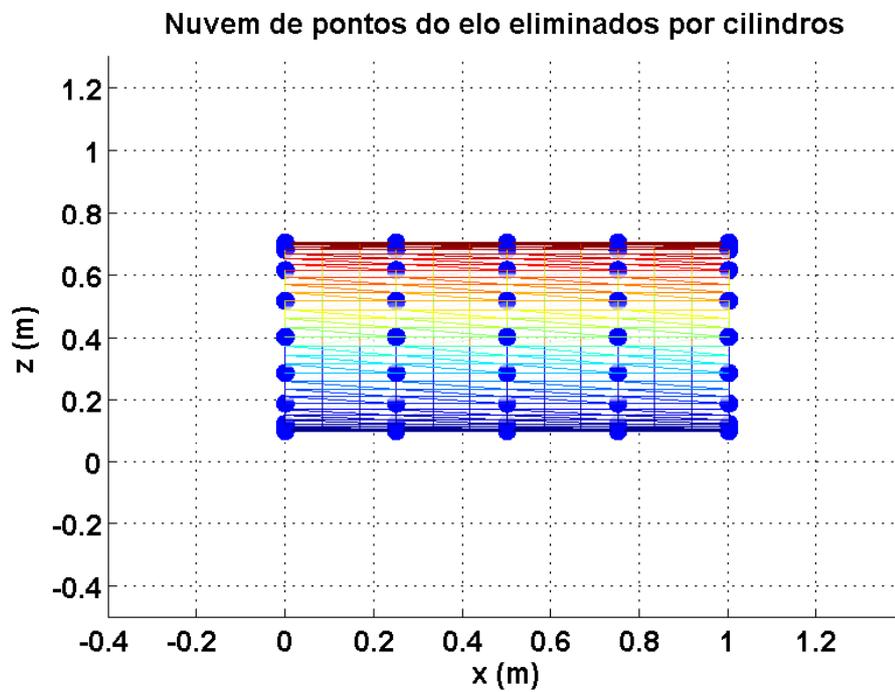


Figura 4.9: Esquematização do método dos cilindros.

### 4.3.3 Eliminando os pontos fora do alcance do robô

Para diminuir a carga computacional do método de detecção, pode-se descartar os obstáculos que estão fora do alcance do robô em um determinado instante de tempo. Primeiramente, os pontos localizados fora do espaço de trabalho do robô são removidos da nuvem de pontos. Em seguida, as figuras geométricas criadas para remover o robô da nuvem são reutilizadas aumentando-se as suas dimensões. Finalmente, remove-se os pontos da nuvem localizados fora das figuras geométricas. Um exemplo das etapas para obter essa nuvem de pontos final é apresentada na Figura 4.10:

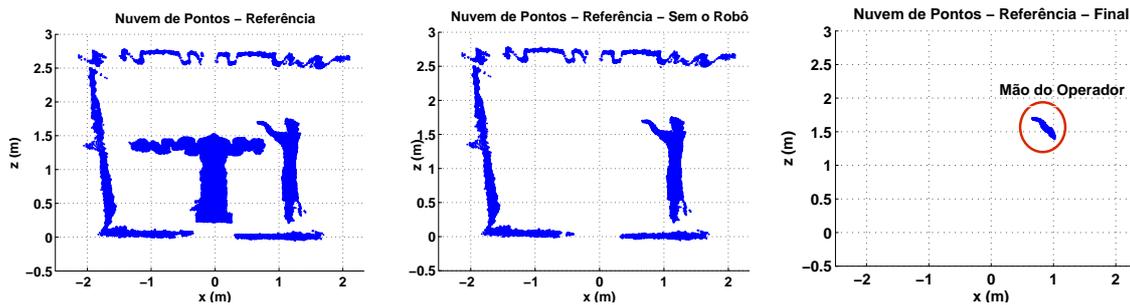


Figura 4.10: Filtragem dos pontos da nuvem.

**Observação 2.** *Uma desvantagem do método de detecção baseado em um único sensor de profundidade é a possibilidade de existência de zonas de oclusão ou de objetos oclusos, aumentando o risco de colisões. Uma solução alternativa para minimizar a ocorrência de oclusão é aumentar a densidade de sensores, utilizando mais de um sensor de profundidade com suas linhas de visão perpendiculares umas a outras, ou realizar uma fusão sensorial entre o Kinect e um sensor externo que forneça uma representação geométrica do ambiente na forma de uma nuvem de pontos. Entretanto, ao utilizar dois Kinects, interferências mútuas podem ocorrer entre os sensores. Ao utilizar um motor para vibrar um dos Kinects a interferência mútua pode ser eliminada (Maimone & Fuchs 2012).*

## 4.4 Resultados Experimentais

Nesta seção, resultados experimentais são apresentados para ilustrar a viabilidade do método proposto. O sistema robótico consiste de um robô Motoman DIA10 (Motoman Inc.) de dois braços com 15 graus de mobilidade (DoF, *degrees of freedom*), com 7-DoF por braço mais 1-DoF na base, um controlador industrial de baixo nível NX100 e um sistema HSC (*High-speed Synchronous Controller*) (Yasakawa 2009). Nos experimentos apenas o tronco e um braço será utilizado ( $n = 8$ ) e a tarefa de interesse consiste em alcançar uma posição e orientação desejadas no espaço Cartesiano ( $m = 6$ ), com apenas 2 graus de redundância ( $n - m = 2$ ).

Os sinais de referência são gerados por um computador externo, rodando Windows OS, que é conectado ao controlador NX100 pelo sistema HSC. Então, os sinais de referência são enviadas para uma malha de controle de posição das juntas, que é implementada em Simulink e executada a uma frequência de 500 Hz. O acesso ao sistema HSC é realizado através do Matlab usando uma arquitetura e biblioteca desenvolvida para aplicações de robótica e automação, denominada *Robot Raconteur*, distribuída pelas redes do CATS/RPI<sup>1</sup>.

Um dispositivo de interface natural Microsoft Kinect<sup>TM</sup> é posicionado de frente para o robô a uma distância de 2,25 m e a uma altura de 0,96 m. Esses valores são medidos com respeito ao sistema de coordenadas de referência fixado na base do robô. O sensor de profundidade do Kinect captura imagens com uma resolução de  $320 \times 240$  pixels a uma frequência de 30 Hz e os parâmetros intrínsecos do sensor são obtidos a partir de um método de calibração proposto em (Herrera C. et al. 2012). Na Figura 4.11 pode-se observar o ambiente experimental contendo o robô manipulador, o Kinect e o operador.

Os parâmetros de controle ajustados empiricamente são:  $K_p = 50I \text{ s}^{-1}$ ,  $K_o = 0,05I \text{ rad s}^{-1}$ ,  $k_{L,O} = 0,05$ ,  $\eta_{lim} = 0,6 \text{ m} + \text{raio do elo}$ . Outros parâmetros são:  $v_{max} = 0,35 \text{ m s}^{-1}$ ,  $\rho = 0,6 \text{ m}$ ,  $b = 9$ ,  $\dot{V}_{r_{max}} = 1 \text{ m s}^{-1}$ ,  $c = 9$ .

Três experimentos foram realizados para avaliar o comportamento do robô em tarefas de regulação, movimento ponto-a-ponto e rastreamento, considerando a presença de seres humanos no seu espaço de trabalho<sup>2</sup>. No Experimento 1, a tarefa de interesse consiste na regulação de posição e de orientação enquanto um operador aproxima-se do robô, fazendo-o reagir (vide Figura 4.12). Neste caso, a posição e orientação desejadas são iguais a posição e orientação iniciais.

A pose inicial do efetuador  $x_0 = [0,816 \ 0,656 \ 1,337]^T \text{ m}$  e os movimentos realizados pelo efetuador para evitar a colisão com o operador são apresentados na

<sup>1</sup>Center for Automation Technologies and Systems / Rensselaer Polytechnic Institute, Troy, NY, USA - <http://www.cats.rpi.edu/>

<sup>2</sup>Um vídeo apresentando os experimentos de regulação e movimento ponto-a-ponto pode ser encontrado em: <http://www.youtube.com/watch?v=yYi0jCC2i2M>



Figura 4.11: Ambiente experimental com o robô manipulador, o dispositivo de interface natural montado em cima de um tripé e o operador.

Figura 4.13. A Figura 4.14 apresenta a evolução no tempo do erro de posição e da norma do erro de orientação respectivamente, onde pode-se observar a variação do erro de posição devido ao comportamento relativo do robô.

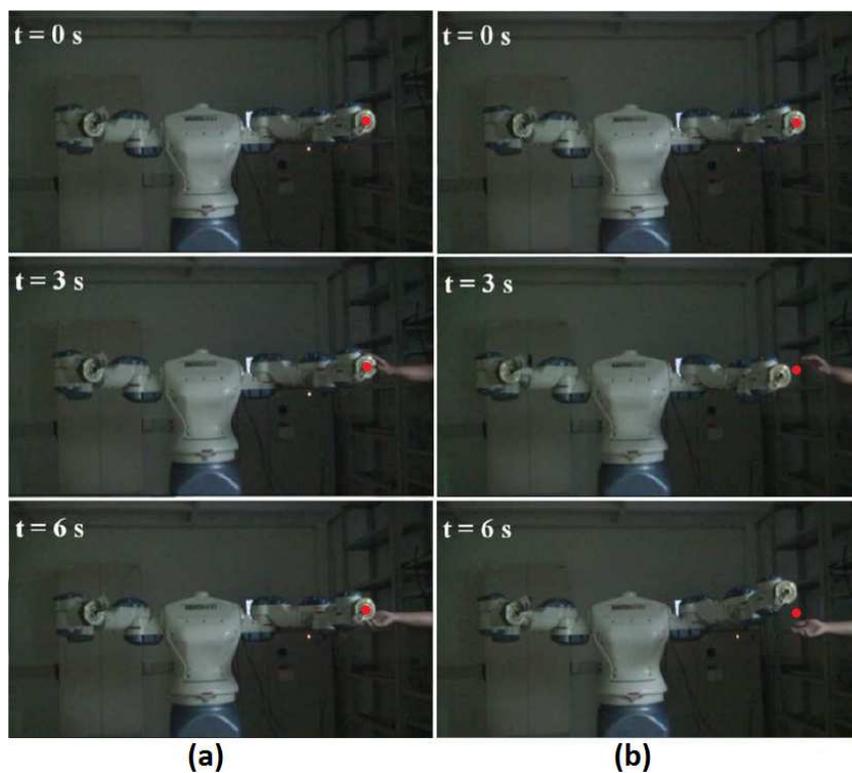


Figura 4.12: Sequência de imagens do robô para uma tarefa de regulação (a) sem desvio de obstáculos e (b) com desvio de obstáculos.

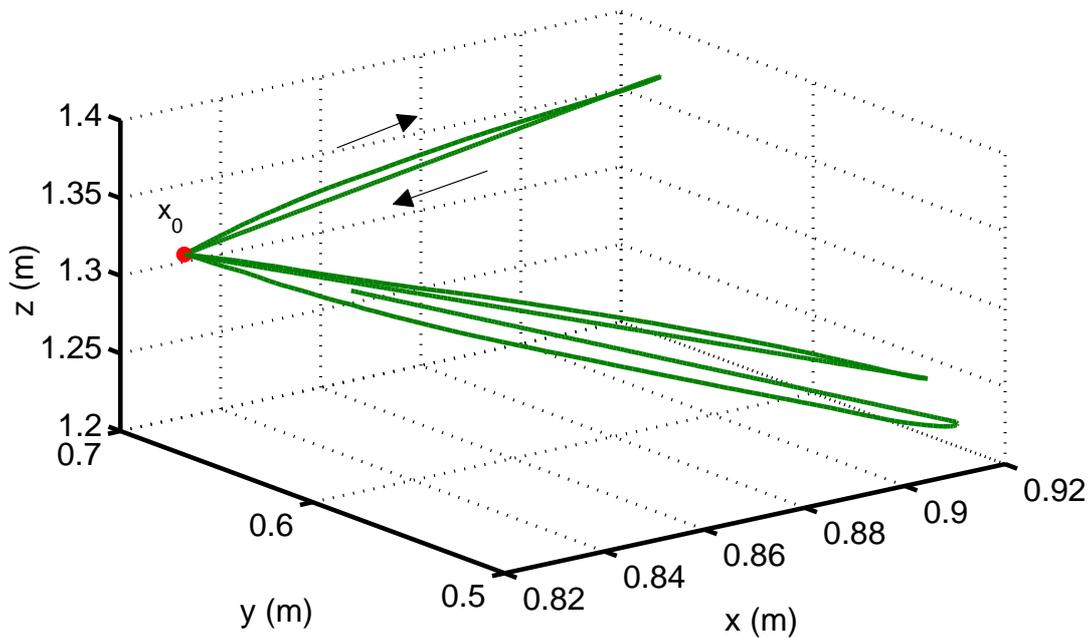


Figura 4.13: Posição inicial do efetuator  $x_0$  e movimentos do efetuator durante uma tarefa de regulação com desvio de obstáculos.

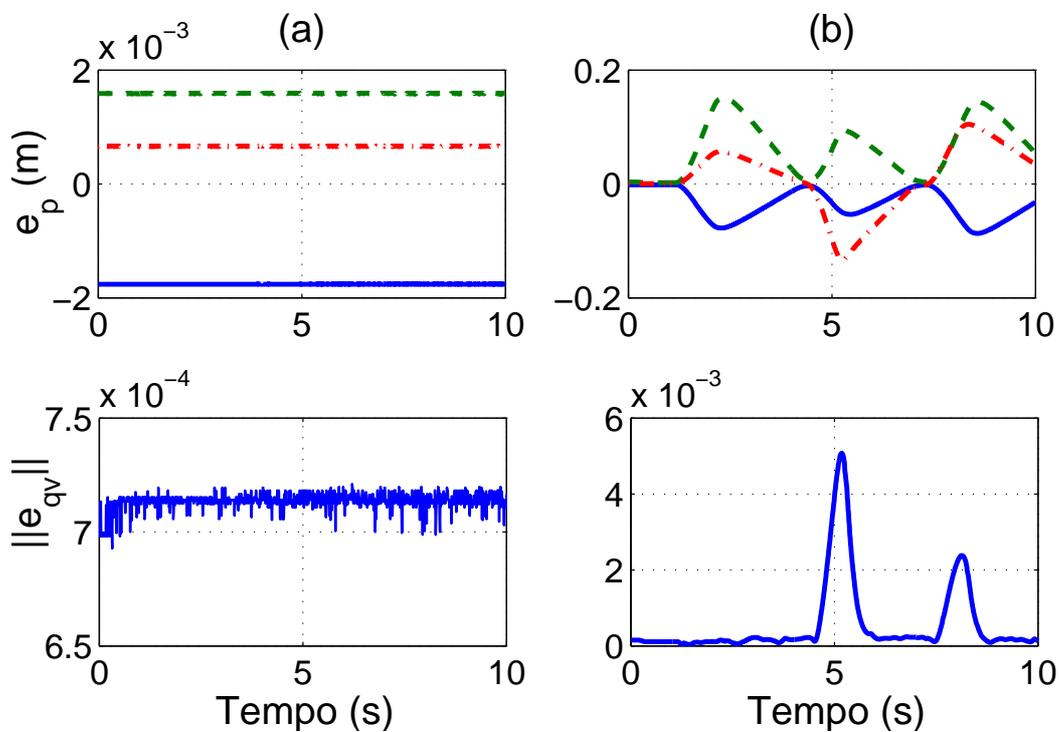


Figura 4.14: Erro de posição e norma do erro de orientação para uma tarefa de regulação (a) sem desvio de obstáculos e (b) com desvio de obstáculos. Legenda:  $e_{px}$  (—),  $e_{py}$  (---),  $e_{pz}$  (-·-).

O comportamento dos sinais de controle Cartesiano de posição  $v_p$  e de orientação  $v_o$  é observado na Figura 4.15 e a Figura 4.16 apresenta o sinal de controle nas juntas.

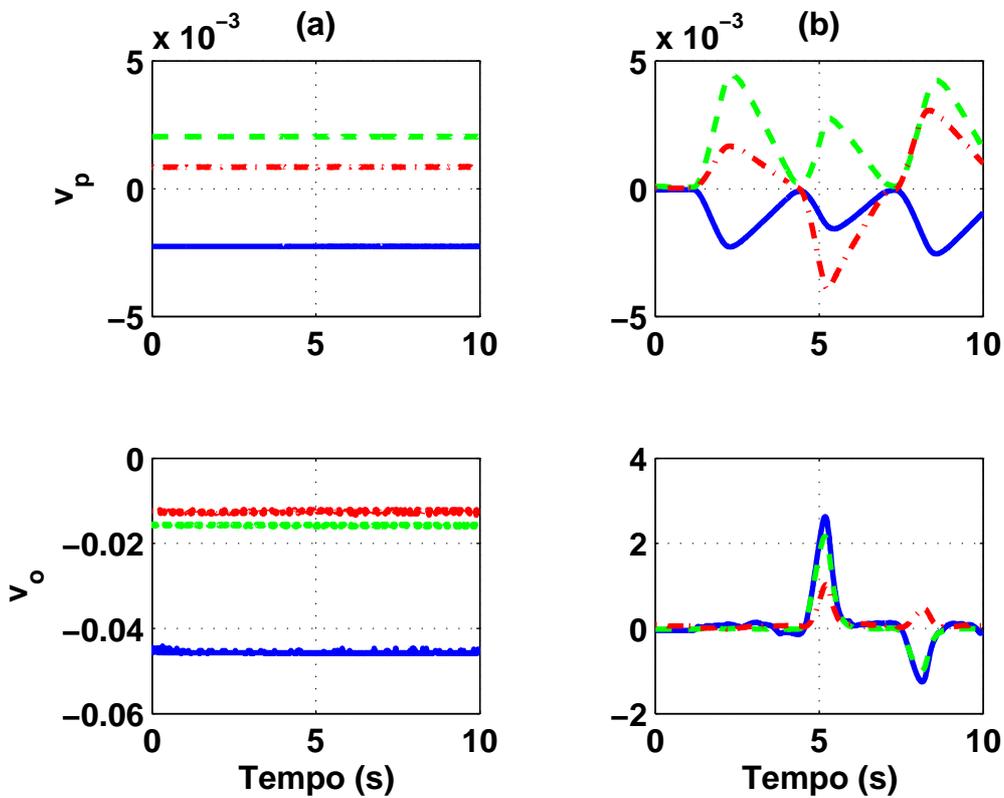


Figura 4.15: Sinal de Controle Cartesiano para uma tarefa de regulação (a) sem desvio de obstáculos e (b) com desvio de obstáculos.

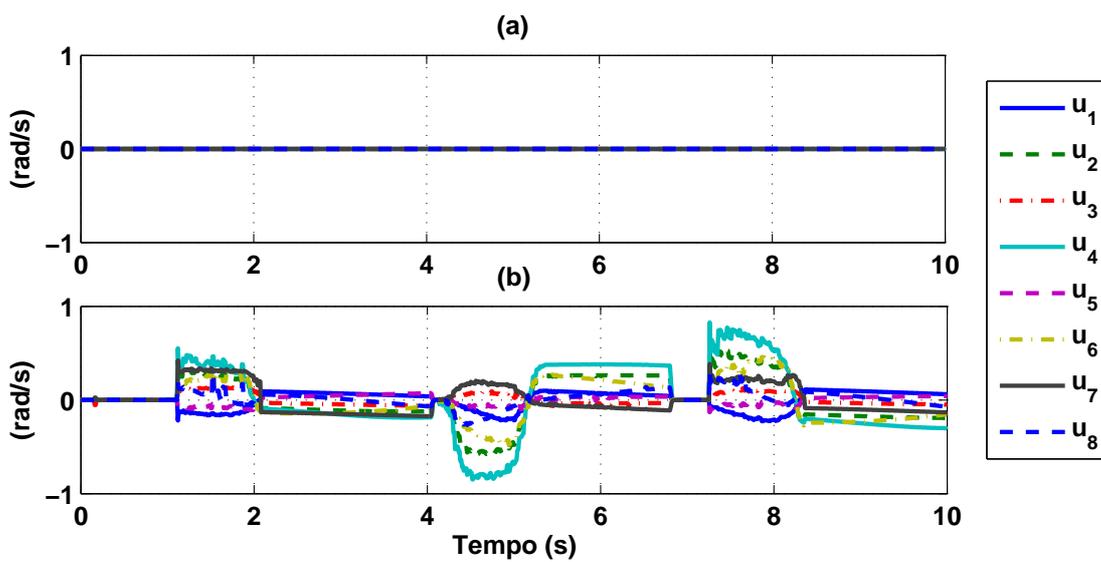


Figura 4.16: Sinal de Controle nas juntas para uma tarefa de regulação (a) sem desvio de obstáculos e (b) com desvio de obstáculos.

No Experimento 2, o efetuador movimenta-se através de quatro pontos de passagem (*waypoints*) formando uma trajetória retangular em um plano, enquanto o operador aproxima-se constantemente do robô (vide Figura 4.17). A evolução no tempo da posição do efetuador durante a execução da tarefa de rastreamento com desvio de obstáculos é apresentada na Figura 4.18. Na Figura 4.19 observa-se o comportamento do erro de posição e da norma do erro de orientação respectivamente, devido à ação reativa do robô para evitar colisões e alcançar os *waypoints* simultaneamente.

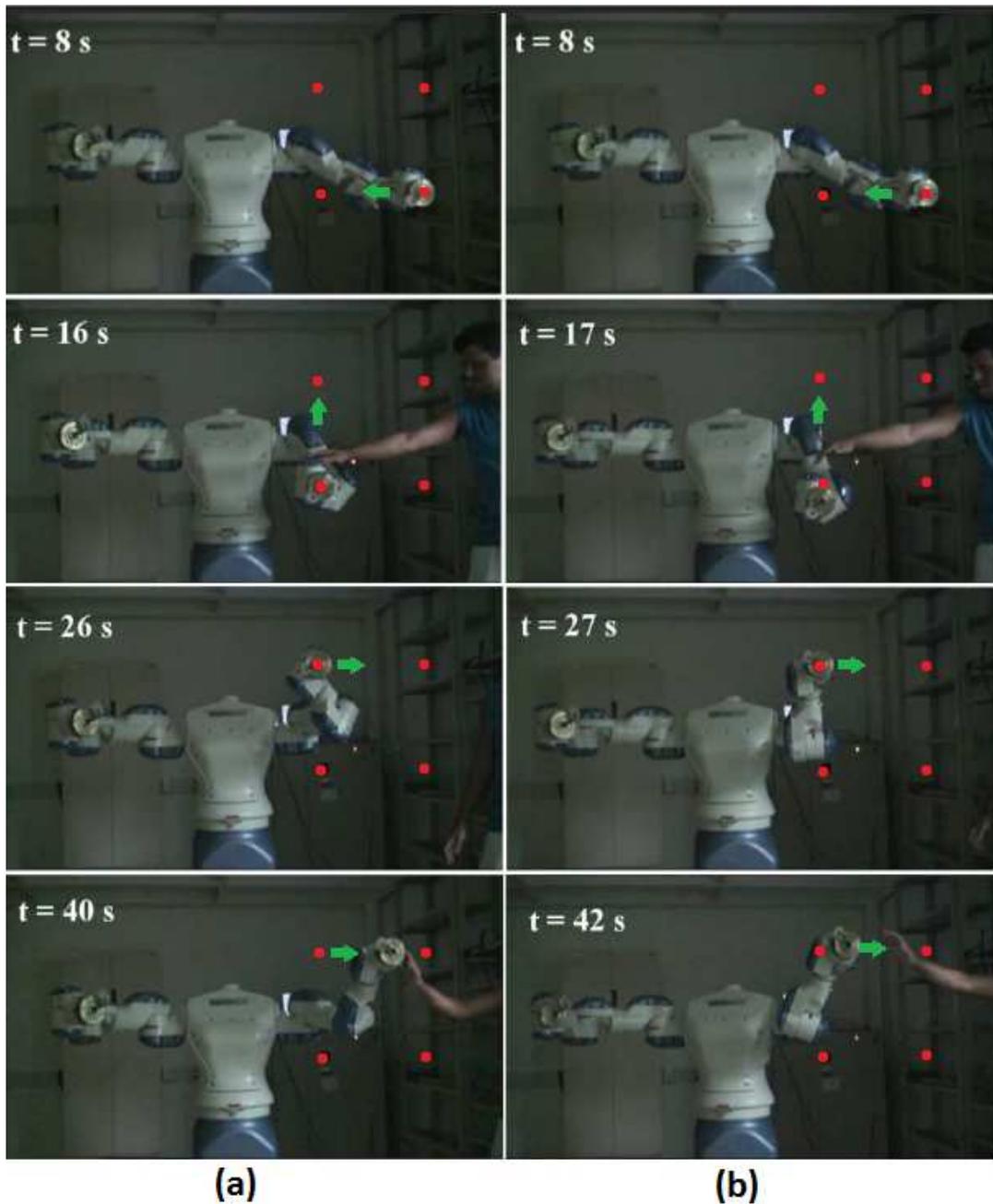


Figura 4.17: Sequência de imagens do robô para um movimento ponto-a-ponto (a) sem desvio de obstáculos (b) e com desvio de obstáculos.

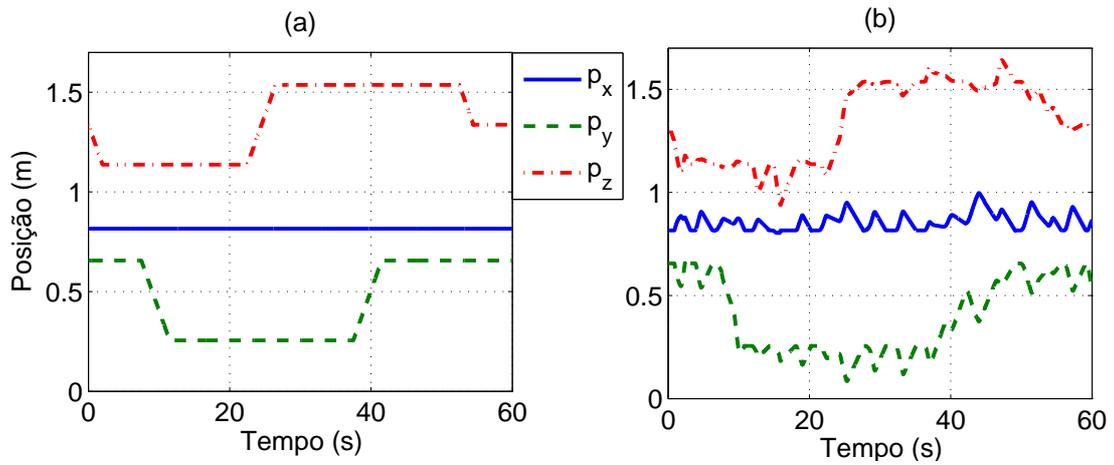


Figura 4.18: Evolução no tempo da posição do efetuador para o movimento ponto-a-ponto (a) sem desvio de obstáculos e (b) com desvio de obstáculos.

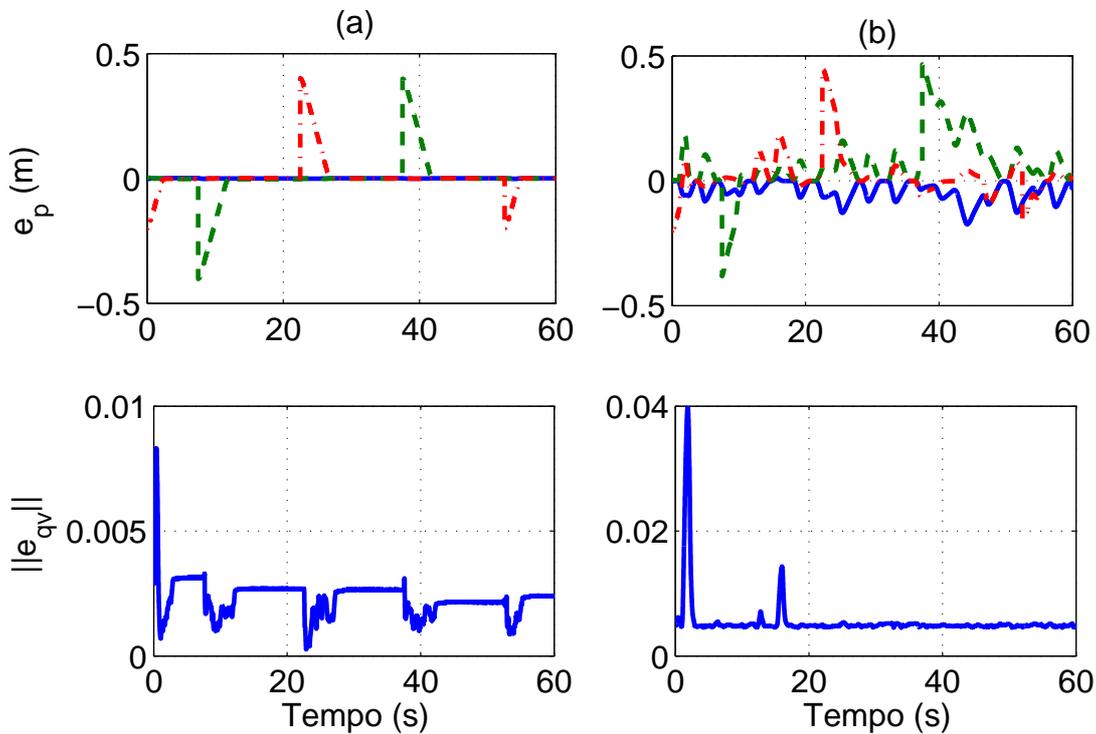


Figura 4.19: Erro de posição e norma do erro de orientação para o movimento ponto-a-ponto. (a) sem desvio de obstáculos e (b) com desvio de obstáculos. Legenda:  $e_{px}$  (—),  $e_{py}$  (---),  $e_{pz}$  (-·-).

O comportamento dos sinais de controle Cartesiano de posição  $v_p$  e de orientação  $v_o$  é observado na Figura 4.20 e a Figura 4.21 apresenta o sinal de controle nas juntas.

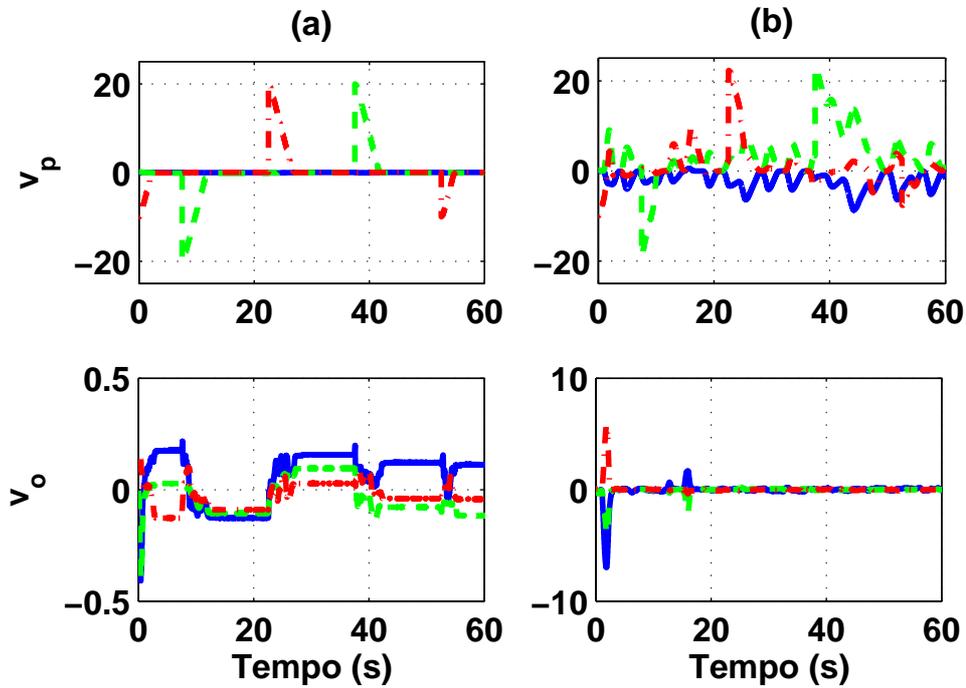


Figura 4.20: Sinal de Controle Cartesiano para o movimento ponto-a-ponto. (a) sem desvio de obstáculos e (b) com desvio de obstáculos.

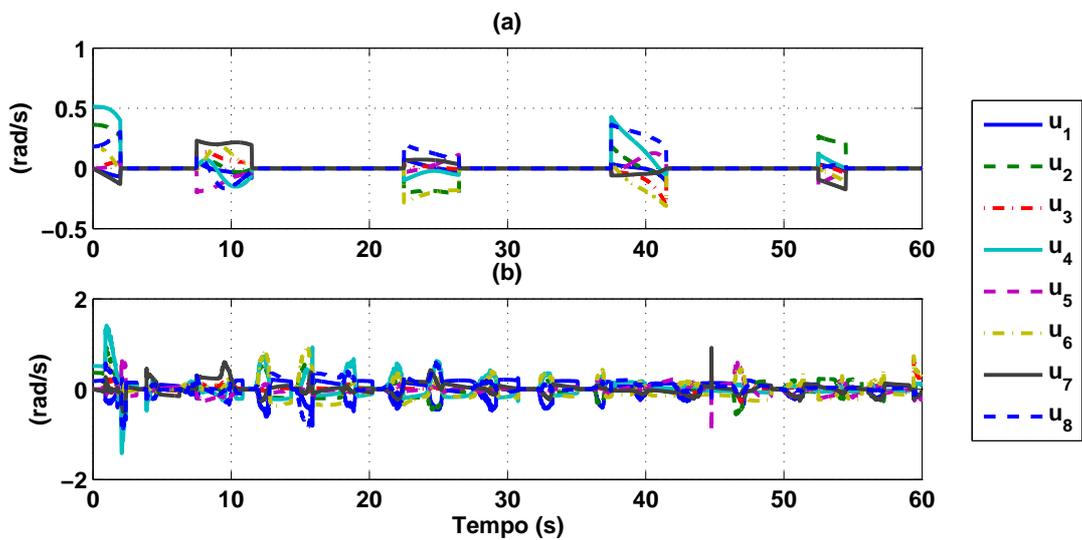


Figura 4.21: Sinal de Controle nas juntas para o movimento ponto-a-ponto. (a) sem desvio de obstáculos e (b) com desvio de obstáculos.

No Experimento 3, a tarefa de interesse consiste no rastreamento da posição e na regulação da orientação enquanto o operador aproxima-se constantemente do

robô. Neste caso, a efetuador movimentar-se formando uma trajetória circular em um plano. Os movimentos realizados pelo efetuador para evitar a colisão com o operador são apresentados na Figura 4.22. A Figura 4.23 apresenta a evolução no tempo do erro de posição e da norma do erro de orientação respectivamente.

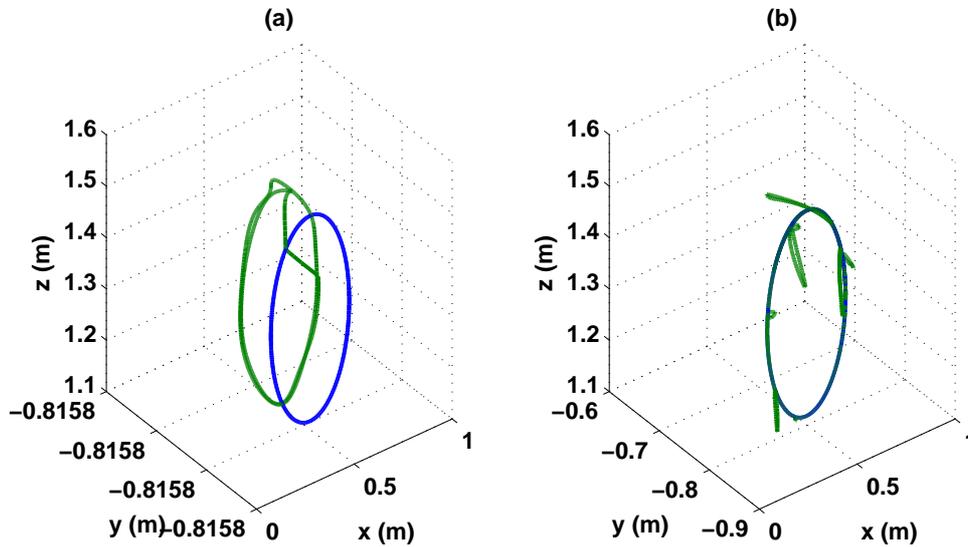


Figura 4.22: Movimentos do efetuador durante uma tarefa de rastreamento (a) sem desvio de obstáculos e (b) com desvio de obstáculos.

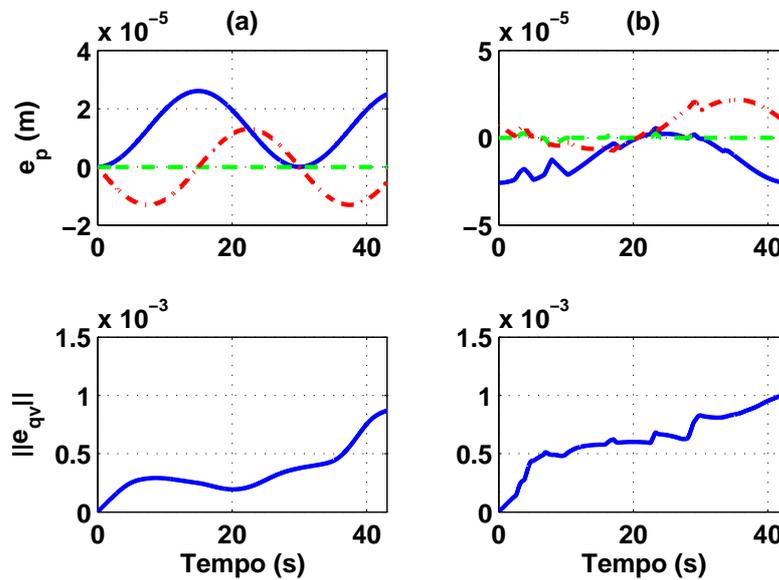


Figura 4.23: Erro de posição e norma do erro de orientação para uma tarefa de rastreamento (a) sem desvio de obstáculos e (b) com desvio de obstáculos. Legenda:  $e_{px}$  (—),  $e_{py}$  (— —),  $e_{pz}$  (— ·).

O comportamento dos sinais de controle Cartesiano de posição  $v_p$  e de orientação  $v_o$  é observado na Figura 4.24 e a Figura 4.25 apresenta o sinal de controle nas juntas.

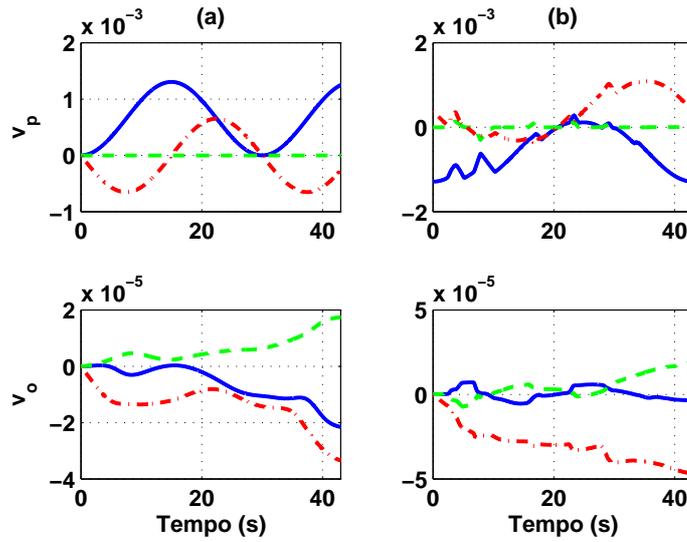


Figura 4.24: Sinal de Controle Cartesiano para uma tarefa de rastreamento (a) sem desvio de obstáculos e (b) com desvio de obstáculos.

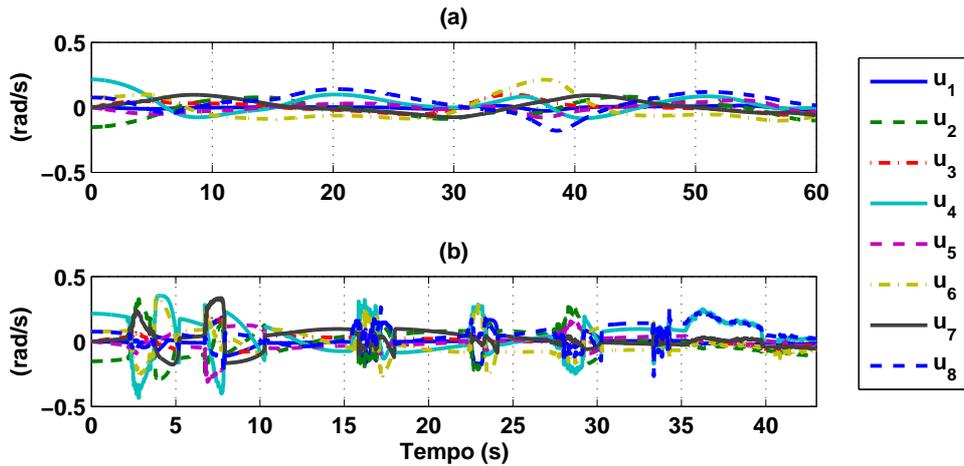


Figura 4.25: Sinal de Controle nas juntas para uma tarefa de rastreamento (a) sem desvio de obstáculos e (b) com desvio de obstáculos.

## 4.5 Conclusões

Nesta seção, apresenta-se um método de detecção de obstáculos que utiliza um sensor de profundidade de um dispositivo de interface natural Microsoft Kinect<sup>TM</sup> para detectar objetos presentes no espaço de trabalho do robô, a partir de uma nuvem de pontos. Uma metodologia de calibração remota, baseada no uso de sensores internos e externos, é introduzida para estimar os parâmetros extrínsecos do sensor de profundidade e evitar a necessidade de calibração *in-situ*.

Um método de filtragem da nuvem de pontos baseado no uso de figuras geométricas - esferas, cilindros e elipsoides - é apresentado para remover os elos e as juntas do robô do cenário obtido pelo sensor de profundidade bem como os obstáculos localizados fora do espaço de trabalho do robô. Uma comparação entre as características de cada método de filtragem é apresentada de acordo com a análise de dados experimentais obtidos pelo Kinect. Resultados práticos indicam que o método de filtragem com menor tempo de processamento é o baseado em esferas, entretanto o método mais adequado depende da geometria do robô.

Finalmente, resultados experimentais preliminares, obtidos com um robô Motoman DIA10 e um dispositivo Microsoft Kinect<sup>TM</sup>, são apresentados para ilustrar a viabilidade do método proposto. É válido ressaltar que pontos de equilíbrio de força (mínimos locais) ocorrem quando o operador está localizado entre o robô e a trajetória de referência, mas quando o operador se afasta suficientemente o robô volta a realizar a tarefa de interesse.

# Capítulo 5

## Considerações Finais e Discussões

Neste capítulo, apresenta-se as considerações finais e algumas propostas para o desenvolvimento de trabalhos futuros, de acordo com os tópicos de pesquisa investigados na dissertação.

### 5.1 Conclusões

1. Neste trabalho é apresentado um método de controle cinemático de posição e orientação, bem como uma metodologia de desvio e detecção de obstáculos para permitir uma interação natural, efetiva e segura entre seres humanos e robôs. Simulações e experimentos realizados com um robô Motoman DIA10 e um dispositivo de interface natural Microsoft Kinect<sup>TM</sup> demonstram a viabilidade da solução proposta.
2. O algoritmo de controle cinemático baseia-se na abordagem de campos potenciais artificiais, utilizando a pseudo-inversa ponderada da matriz Jacobiana para evitar que os movimentos do robô alcancem os limites mecânicos das juntas e aproveitando a característica de redundância do robô para desviar de obstáculos estáticos e dinâmicos.
3. O método de desvio de obstáculos é baseado no uso de dois tipos de campos potenciais repulsivos, para as juntas e para o efetuador. Um é definido em termos de uma restrição adicional onde as juntas são consideradas como os pontos de interesse para o cálculo da distância, e outro atua sobre o efetuador e modifica a lei de controle de posição convencional garantindo que o mesmo siga uma trajetória de referência livre de colisões. A razão de utilizar tipos diferentes de campos repulsivos está relacionada à maneira como eles são definidos, suas respectivas dimensões, suavidade de movimento do robô e existência de um valor máximo limitado. De acordo com os resultados de simulação e de experimentos conclui-se que o desvio de obstáculos somente

para os elos é suficiente para evitar colisões. Porém, a inclusão do desvio de obstáculos para o efetuador, em geral, faz com que os elos se afastem mais dos obstáculos além de evitar situações que podem impedir a realização da tarefa. Devido a presença de obstáculos dinâmicos, o problema de mínimos locais não foi recorrente e a tarefa foi realizada com êxito.

4. O algoritmo de detecção de obstáculos utiliza um sensor de profundidade baseado na tecnologia de luz estruturada para obter uma descrição geométrica 2 1/2 D do ambiente a partir de uma nuvem de pontos. Uma limitação do método de detecção é que durante a realização da tarefa podem ocorrer oclusões e, conseqüentemente, a segurança do operador pode não ser garantida. Uma solução alternativa para minimizar a ocorrência de oclusão de objetos é aumentar a densidade de sensores, utilizando mais de um sensor de profundidade com suas linhas de visão perpendiculares umas as outras ou realizar uma fusão sensorial entre o Kinect e um sensor externo que forneça uma representação geométrica do ambiente na forma de uma nuvem de pontos. Campos repulsivos são criados ao redor dos obstáculos identificados permitindo a execução bem sucedida da tarefa de interesse sem colisões.
5. Uma vez que o algoritmo de detecção pode considerar os elementos do robô como obstáculos, um método de filtragem aplicado à nuvem de pontos é proposto para eliminar o robô do cenário obtido pelo sensor de profundidade, delimitado os elos e as juntas por uma sequência de figuras geométricas - esferas, cilindros e elipsoides. Resultados práticos indicam que o método de filtragem baseado no uso de esferas requer menor esforço computacional (i.e., menor tempo de processamento) devido à simplicidade de implementação em relação aos outros métodos propostos. Entretanto, a escolha do método pode depender da dimensão e da geometria do robô.
6. Uma metodologia de calibração remota baseada no uso de sensores internos e externos - *encoders* do robô e a câmera RGB do Kinect - é introduzida para estimar os parâmetros extrínsecos do sensor de profundidade, evitando a necessidade de calibração *in-situ*. Entretanto, uma vez que as medidas obtidas pelos sensores estão sujeitas a ruídos, é necessário utilizar diversos pontos de calibração para melhorar a precisão dos parâmetros estimados. Por outro lado, um aumento no número de pontos de calibração pode aumentar a dimensão da matriz Jacobiana, dificultando o cálculo da sua inversa.

## 5.2 Propostas para Trabalhos Futuros

Para dar continuidade ao desenvolvimento dos temas de pesquisa apresentados nessa dissertação, seguem abaixo algumas propostas para serem exploradas em trabalhos futuros:

- Considerar a inclusão de algoritmos de planejamento de trajetórias na solução proposta visando solucionar o problema da escolha da posição inicial e da existência de mínimos locais, que bloqueiam o manipulador em uma configuração de equilíbrio de forças.
- Estudar a viabilidade de desenvolvimento e implementação de um algoritmo de detecção no espaço da imagem que identifique os tipos de obstáculos, localizados dentro do espaço de trabalho do robô, como humanos e objetos. Neste contexto, o algoritmo de detecção deve recorrer ao uso de técnicas de reconhecimento facial e de processamento de imagens para classificar os obstáculos e delimitar o espaço de trabalho na imagem. Dessa forma, pode-se estabelecer prioridades de desvio de acordo com o tipo de obstáculo presente no ambiente de operação. Por exemplo, o desvio de operadores movendo-se livremente pode ter maior prioridade em relação ao desvio de objetos estáticos do ambiente.
- Desenvolver e implementar um algoritmo de controle cinemático e de desvio de obstáculos, relaxando a hipótese de completo conhecimento da cinemática direta do robô. As incertezas na cinemática podem surgir, por exemplo, quando (i) o efetuador do robô manipula diversos objetos de dimensões diferentes para utilizar como ferramenta ou quando (ii) o efetuador é substituído por um trocador de ferramentas, mais adequado para a execução de tarefas em ambientes remotos e de difícil acesso. Neste caso, recomenda-se o uso de uma técnica de controle adaptativo ou robusto para lidar com a existência de incertezas paramétricas. Dessa forma, se uma nova ferramenta for utilizada na extremidade do robô, o algoritmo de controle cinemático e desvio de obstáculos pode ser utilizado sem a necessidade de conhecimento prévio da relação de cinemática direta do robô. Uma solução alternativa consiste em utilizar algoritmos de servovisão adaptativa devido a sua conhecida robustez a incertezas na cinemática do robô (Leite et al. 2009).

# Referências Bibliográficas

- Adorno, B., Bó, A. P. L., Fraisse, P. & Poignet, P. (2011), Towards a Cooperative Framework for Interactive Manipulation involving a Human and a Humanoid, *in* ‘Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)’, pp. 3777–3783.
- Adorno, B. V., Bó, A. P. L. & Fraisse, P. (2011), Interactive Manipulation between a Human and a Humanoid: When Robots Control Human Arm Motion, *in* ‘Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)’, pp. 4658–4663.
- Almeida-Antonio, T. B., Leite, A. C., From, P. J., Hsu, L. & Lizarralde, F. (2012), Algoritmos de Controle Cinemático com Desvio de Obstáculos aplicados à Robôs do tipo Pórtico, *in* ‘Anais do XIX Congresso Brasileiro de Automática’, Campina Grande, PB, pp. 3846–3853.
- Almeida-Antonio, T. B., Lizarralde, F., Hsu, L. & Leite, A. C. (2014), Método de Controle e Detecção de Obstáculos para Robôs Manipuladores Aplicado à Interação Humano-Robô, *in* ‘Anais do XX Congresso Brasileiro de Automática’, Belo Horizonte, MG, pp. 4036–4043.
- Arkin, R. C., Fujita, M., Takagi, T. & Hasegawa, R. (2003), ‘An Ethological and Emotional Basis for Human–robot Interaction’, *Robotics and Autonomous Systems* **42**(3–4), 191–201.
- Bernardes, M. C., Adorno, B. V., Borges, G. A. & Poignet, P. (2014), ‘3D Robust Online Motion Planning for Steerable Needles in Dynamic Workspaces Using Duty-Cycled Rotation’, *Journal of Control, Automation and Electrical Systems* **25**(2), 216–227.
- Bjerkeng, M., Transeth, A., Pettersen, K., Kyrkjebo, E. & Fjerdigen, S. (2011), Active Camera Control with Obstacle Avoidance for Remote Operations with Industrial Manipulators: Implementation and Experimental Results, *in* ‘Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)’, pp. 247–254.

- Breazeal, C., Takanishi, A. & Kobayashi, T. (2008), Social Robots that Interact with People, *in* B. Siciliano & O. Khatib, eds, ‘Springer Handbook of Robotics’, Springer Berlin Heidelberg, pp. 1349–1369.
- Chan, T. F. & Dubey, R. V. (1995), ‘A Weighted Least-norm Solution based Scheme for Avoiding Joint Limits for Redundant Joint Manipulators’, *IEEE Transactions on Robotics and Automation* **11**(2), 286–292.
- Cheng, L., Sun, Q., Su, H., Cong, Y. & Zhao, S. (2012), Design and Implementation of Human-robot Interactive Demonstration System based on Kinect, *in* ‘Proceedings of the 24th Chinese Control and Decision Conference (CCDC)’, pp. 971–975.
- Chiaverini, S., Oriolo, G. & Walker, I. D. (2008), Kinematically Redundant Manipulators, *in* B. Siciliano & O. Khatib, eds, ‘Springer Handbook of Robotics’, Springer Berlin Heidelberg, pp. 245–268.
- Chin, K.-Y., Wu, C.-H. & Hong, Z.-W. (2011), A Humanoid Robot as a Teaching Assistant for Primary Education, *in* ‘Proceedings of the 5th International Conference on Genetic and Evolutionary Computing (ICGEC)’, pp. 21–24.
- Chung, C. Y., Lee, B.-H. & Lee, J.-H. (1997), Obstacle Avoidance for Kinematically Redundant Robots using Distance Algorithm, *in* ‘Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)’, Vol. 3, pp. 1787–1793.
- Cleary, K. (2005), Medical Robotics and the Operating Room of the Future, *in* ‘Proceedings of 27th Annual International Conference of the Engineering in Medicine and Biology Society (EMBS)’, Shanghai, China, pp. 7250–7253.
- Corke, P. I. (2011), *Robotics, Vision & Control: Fundamental Algorithms in Matlab*, Springer.
- Correa, D., Sciotti, D., Prado, M., Sales, D., Wolf, D. & Osorio, F. (2012), Mobile Robots Navigation in Indoor Environments Using Kinect Sensor, *in* ‘Proceedings of the 2nd Brazilian Conference on Critical Embedded Systems (CBSEC)’, pp. 36–41.
- Dautenhahn, K. (2007), ‘Socially Intelligent Robots: Dimensions of Human–robot Interaction’, *Philosophical Transactions of the Royal Society B* **362**(1480), 679–704.

- De Luca, A. & Flacco, F. (2012), Integrated Control for pHRI: Collision Avoidance, Detection, Reaction and Collaboration, *in* ‘Proceedings of the 4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)’, pp. 288–295.
- Dixon, W. (2004), Adaptive Regulation of Amplitude Limited Robot Manipulators with Uncertain Kinematics and Dynamics, *in* ‘Proceedings of the 2004 American Control Conference (ACC)’, Vol. 4, Boston, MA, USA, pp. 3839–3844.
- Figueredo, L., Adorno, B., Ishihara, J. & Borges, G. (2013), Robust Kinematic Control of Manipulator Robots using Dual Quaternion Representation, *in* ‘Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)’, pp. 1949–1955.
- Flacco, F., Kroger, T., De Luca, A. & Khatib, O. (2012), A Depth Space Approach to Human-robot Collision Avoidance, *in* ‘Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)’, pp. 338–345.
- Flandin, G., Chaumette, F. & Marchand, E. (2000), Eye-in-hand/eye-to-hand Cooperation for Visual Servoing, *in* ‘Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)’, Vol. 3, pp. 2741–2746.
- Fong, T., Thorpe, C. & Baur, C. (2003), Collaboration, Dialogue, Human-Robot Interaction, *in* R. Jarvis & A. Zelinsky, eds, ‘Robotics Research’, Vol. 6 of *Springer Tracts in Advanced Robotics*, Springer Berlin Heidelberg, Berlin, pp. 255–266.
- Green, S. A., Billingham, M., Chen, X. & Chase, J. G. (2008), ‘Human-robot Collaboration: A Literature Review and Augmented Reality Approach in Design’, *International Journal of Advanced Robotic Systems* 5(1), 1–18.
- Greuter, M., Rosenfelder, M., Blaich, M. & Bittel, O. (2011), Obstacle and Game Element Detection with the 3D-Sensor Kinect, *in* D. Obdržálek & A. Gottscheber, eds, ‘Research and Education in Robotics - EURO-BOT 2011’, Vol. 161 of *Communications in Computer and Information Science*, Springer Berlin Heidelberg, Berlin, pp. 130–143.
- Haddadin, S., Albu-Schäfer, A., De Luca, A. & Hirzinger, G. (2008), Collision Detection and Reaction: A Contribution to Safe Physical Human-Robot Interaction, *in* ‘Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)’, pp. 3356–3363.

- Haddadin, S., Belder, R. & Albu-Schäffer, A. (2011), Dynamic Motion Planning for Robots in Partially Unknown Environments, *in* ‘Proceedings of the 18th IFAC World Congress’, Vol. 18, Milan, Italy, pp. 6842–6850.
- Herrera C., D., Kannala, J. & Janne, H. (2012), ‘Joint Depth and Color Camera Calibration with Distortion Correction’, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(10), 2058–2064.
- Kashiwazaki, K., Sugahara, Y., Iwasaki, J., Kosuge, K., Kumazawa, S. & Yamashita, T. (2010), Greenhouse Partner Robot System, *in* ‘Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)’, pp. 1–8.
- Khalil, H. K. (2002), *Nonlinear Systems*, 3rd edn, Prentice Hall Inc.
- Khatib, O. (1986), ‘Real-Time Obstacle Avoidance for Manipulators and Mobile Robots’, *The International Journal of Robotics Research* **5**(1), 90–98.
- LaValle, S. (2011), ‘Motion Planning’, *IEEE Robotics Automation Magazine* **18**(1), 79–89.
- Leite, A. C., Lizarralde, F., From, P. J., Costa, R. R. & Hsu, L. (2012), ‘Remote Calibration and Trajectory Replanning for Robot Manipulators Operating in Unstructured Environments’, *Proceedings of the 2012 IFAC Workshop on Automatic Control in Offshore Oil and Gas Production 1*, 59–65.
- Leite, A. C., Lizarralde, F. & Hsu, L. (2009), ‘Hybrid Adaptive Vision-Force Control for Robot Manipulators Interacting with Unknown Surfaces’, *The International Journal of Robotics Research* **28**(7), 911–926.
- Lohse, M., Hanheide, M., Wrede, B., Walters, M., Koay, K. L., Syrdal, D., Green, A., Huttenrauch, H., Dautenhahn, K., Sagerer, G. & Severinson-Eklundh, K. (2008), Evaluating Extrovert and Introvert Behaviour of a Domestic Robot - a Video Study, *in* ‘Proceedings of the 17th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)’, pp. 488–493.
- Luo, C., Espinosa, A., Pranantha, D. & De Gloria, A. (2011), Multi-robot Search and Rescue Team, *in* ‘Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)’, pp. 296–301.
- Maimone, A. & Fuchs, H. (2012), Reducing interference between multiple structured light depth sensors using motion, *in* ‘IEEE Virtual Reality Short Papers and Posters (VRW)’, pp. 51–54.

- Marques, G. C. (2005), Calibração Remota de Sistemas Robóticos Utilizando Sensores Internos e Externos, Master's thesis, Programa de Engenharia Elétrica COPPE/UFRJ.
- Minguez, J., Lamiroux, F. & Laumond, J.-P. (2008), Motion Planning and Obstacle Avoidance, *in* B. Siciliano & O. Khatib, eds, 'Springer Handbook of Robotics', Springer Berlin-Heidelberg, pp. 827–852.
- Monzon, P. (2003), 'On Necessary Conditions for Almost Global Stability', *IEEE Transactions on Automatic Control* **48**(4), 631–634.
- Motta, J., de Carvalho, G. & McMaster, R. (2001), 'Robot Calibration using a 3D Vision-based Measurement System with a Single Camera', *Robotics and Computer-Integrated Manufacturing* **17**(6), 487–497.
- Motta-Ribeiro, G. C., T. B., Leite, A. C., From, P. J., Lizarralde, F. & Hsu, L. (2012), Uma Arquitetura para Teleoperação integrando Interface Natural, Realimentação de Força e Servovisão, *in* 'Anais do XIX Congresso Brasileiro de Automática', Campina Grande, PB, pp. 3549–3556.
- Murray, R. M., Li, Z. & Sastry, S. S. (1994), *A Mathematical Introduction to Robotic Manipulation*, CRC Press LLC.
- Nakamura, Y. & Hanafusa, H. (1986), 'Inverse Kinematic Solutions with Singularity Robustness for Robot manipulator Control', *ASME Journal of Dynamic Systems* **108**(3), 163–171.
- Rakprayoon, P., Ruchanurucks, M. & Coundoul, A. (2011), Kinect-based Obstacle Detection for Manipulator, *in* 'Proceedings of the IEEE/SICE International Symposium on System Integration', pp. 68–73.
- Rimon, E. & Koditschek, D. (1992), 'Exact Robot Navigation using Artificial Potential Functions', *IEEE Transactions on Robotics and Automation*, **8**(5), 501–518.
- Sadazuka, K., Kuno, Y., Kawashima, M. & Yamazaki, K. (2007), Museum Guide Robot with Effective Head Gestures, *in* 'Proceedings of the International Conference on Control, Automation and Systems (ICCAS)', pp. 1168–1171.
- Santis, A. D., Siciliano, B., Luca, A. D. & Bicchi, A. (2008), 'An Atlas of Physical Human-robot Interaction', *Mechanism and Machine Theory* **43**(3), 253–270.

- Siciliano, B., Sciavicco, L., Villani, L. & Oriolo, G. (2009), *Robotics: Modelling, Planning and Control*, Advanced Textbooks in Control and Signal Processing, Springer, Verlag London Limited.
- Singh, L., Stephanou, H. & Wen, J. (1996), Real-time Robot Motion Control with Circulatory Fields, *in* ‘Proceedings of IEEE International Conference on Robotics and Automation (ICRA)’, Vol. 3, Minneapolis, MN, pp. 2737–2742.
- Sisbot, E. A., Marin-Urias, L. F., Broquère, X., Sidobre, D. & Alami, R. (2010), Synthesizing Robot Motions Adapted to Human Presence, *in* ‘Journal of Social Robotics’, Vol. 2, pp. 329–343.
- Slotine, J.-J. E. & Li, W. (1991), *Applied Nonlinear Control*, Prentice Hall.
- Steinfeld, A., Fong, T., Kaber, D., Lewis, M., Scholtz, J., Schultz, A. & Goodrich, M. (2006), Common Metrics for Human-robot Interaction, *in* ‘Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-robot Interaction’, ACM Digital Library, New York, NY, USA, pp. 33–40.
- Takubo, T., Arai, H., Hayashibara, Y. & Tanie, K. (2002), ‘Human-robot Cooperative Manipulation using a Virtual Nonholonomic Constraint’, *International Journal of Robotics Research* **21**(5-6), 541–553.
- Vargas, L. V. (2013), Inversa Filtrada: Uma Solução Alternativa para a Cinemática Inversa de Manipuladores Robóticos, Master’s thesis, Programa de Engenharia Elétrica COPPE/UFRJ.
- Vargas, L. V., Leite, A. C. & Costa, R. R. (2014), Overcoming Kinematic Singularities with the Filtered Inverse Approach, *in* ‘Proceedings of the 19th IFAC World Congress’, Cape Town, South Africa, pp. 8496–8502.
- Vogel, C., Walter, C. & Elkmann, N. (2013), A Projection-based Sensor System for Safe Physical Human-robot Collaboration, *in* ‘Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)’, pp. 5359–5364.
- Weisstein, E. W. (2013), ‘Point-line distance-3-dimensional’. (Acessado em 13/03/2013).  
**URL:** <http://mathworld.wolfram.com/Point-LineDistance3-Dimensional.html>
- Wen, J.-Y. & Kreutz-Delgado, K. (1991), ‘The Attitude Control Problem’, *IEEE Transactions on Automatic Control* **36**(10), 1148–1162.

Yaskawa, M. (2009), 'Manual do Motoman'. (Acessado em 12/09/2014).

**URL:** <http://www.motoman.com/motomedia/manuals/docs/154557-1cd.pdf>

Yuan, J. (1988), 'Closed-loop Manipulator Control using Quaternion Feedback', *IEEE Journal of Robotics and Automation* **4**(4), 434–440.

# Apêndice A

## Geometria Vetorial

Para realizar a descrição da posição e da orientação do efetuador com respeito a um sistema de coordenadas de referência de um modo padronizado, algumas definições baseadas em geometria vetorial são apresentadas a seguir.

**Definição 1** *Um sistema de coordenadas ortonormal  $\bar{E} = \begin{bmatrix} \vec{e}_1 & \vec{e}_2 & \vec{e}_3 \end{bmatrix}$  satisfaz as seguintes propriedades*

1. Normalidade:  $\|\vec{e}_i\| = 1$ ,  $i = 1, 2, 3$ ;
2. Ortogonalidade:  $\vec{e}_i \cdot \vec{e}_j = 0$ ,  $i \neq j$ ;
3. Produto Vetorial:  $\vec{e}_3 = \vec{e}_1 \times \vec{e}_2$ ;

onde  $\vec{e}_i \in \mathcal{V}$  e  $\mathcal{V}$  é um espaço vetorial normado (Siciliano et al. 2009).

Um sistema de coordenadas  $\bar{E} = \begin{bmatrix} \vec{e}_1 & \vec{e}_2 & \vec{e}_3 \end{bmatrix}$  pode ser interpretado como uma transformação linear  $\bar{E} : \mathbb{R}^3 \mapsto \mathcal{V}$ , ou seja:

$$\vec{v} = \bar{E} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} \vec{e}_1 & \vec{e}_2 & \vec{e}_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = v_1 \vec{e}_1 + v_2 \vec{e}_2 + v_3 \vec{e}_3; \quad v_i \in \mathbb{R}. \quad (\text{A.1})$$

Além disso, é importante definir o operador adjunto de  $\bar{E}$ , denotado por  $\bar{E}^*$ , representando um mapeamento  $\bar{E}^* : \mathcal{V} \mapsto \mathbb{R}^3$ , por

$$\bar{E}^* = \begin{bmatrix} \vec{e}_1 \cdot \\ \vec{e}_2 \cdot \\ \vec{e}_3 \cdot \end{bmatrix}, \quad (\text{A.2})$$

onde “ $\cdot$ ” denota o produto escalar e que satisfaz as seguintes propriedades:

1.  $g = \bar{E}^* \bar{E} = I_{3 \times 3}$ ,  $g : \mathbb{R}^3 \mapsto \mathbb{R}^3$ ;

$$2. g^{-1} = \bar{E} \bar{E}^* = \mathcal{I}, \quad g^{-1} : \mathcal{V} \mapsto \mathcal{V},$$

onde  $\mathcal{I}$  é o operador identidade (escalar).

**Definição 2** Considere o vetor  $\vec{v} = v_1 \vec{e}_1 + v_2 \vec{e}_2 + v_3 \vec{e}_3$  representado no sistema de coordenadas  $\bar{E} = \begin{bmatrix} \vec{e}_1 & \vec{e}_2 & \vec{e}_3 \end{bmatrix}$ . As coordenadas de  $\vec{v}$  em  $\bar{E}$  são definidas como  $\vec{v} = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}^T \in \mathbb{R}^3$  e então:

$$1. \vec{v} = \bar{E}v;$$

$$2. v = \bar{E}^* \vec{v};$$

# Apêndice B

## Quatérnio Unitário

As desvantagens de algumas representações de orientação podem ser superadas utilizando-se uma representação de quatro parâmetros denominada quatérnio unitário denotada por  $q = (q_s, q_v)$ , onde:

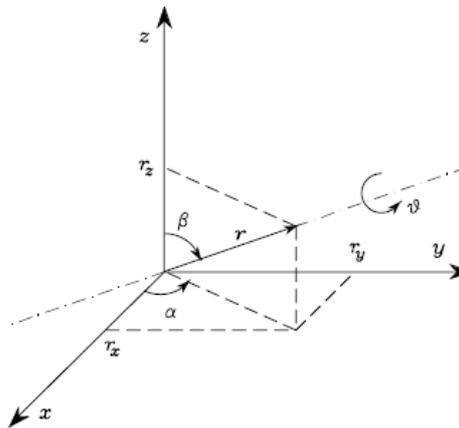


Figura B.1: Rotação de um ângulo  $\vartheta$  ao longo de um eixo arbitrário  $r$

$$q_s = \cos\left(\frac{\vartheta}{2}\right), \quad q_v = \sin\left(\frac{\vartheta}{2}\right) \vec{r}, \quad (\text{B.1})$$

$\vartheta$  é o ângulo de rotação ao redor do eixo  $\vec{r}$  (Figura B.1),  $q_s \in \mathbb{R}$  é denominada a parte escalar do quatérnio, e  $q_v = [q_{vx} \ q_{vy} \ q_{vz}]^T \in \mathbb{R}^3$  é chamada a parte vetorial do quatérnio, estando sujeitas à seguinte restrição:

$$\|q\|^2 = q^T q = q_s^2 + q_v^T q_v = 1. \quad (\text{B.2})$$

Por isso, essa representação é denominada *quatérnio unitário*. É importante ressaltar que a rotação de  $-\vartheta$  ao longo de  $-\vec{r}$  resulta na mesma representação de quatérnio que associada a rotação de  $\vartheta$  ao longo de  $\vec{r}$ . A matriz de rotação corres-

pondente ao quatérnio é obtida por

$$R = (2q_s^2 - 1)I + 2(q_v q_v^T + q_s Q(q_v)), \quad (\text{B.3})$$

onde  $Q(\cdot)$  é o operador produto vetorial, resultando em:

$$R(q_s, q_v) = \begin{bmatrix} 2(q_s^2 + q_{vx}^2) - 1 & 2(q_{vx} q_{vy} - q_s q_{vz}) & 2(q_{vx} q_{vz} + q_s q_{vy}) \\ 2(q_{vx} q_{vy} + q_s q_{vz}) & 2(q_s^2 + q_{vy}^2) - 1 & 2(q_{vy} q_{vz} - q_s q_{vx}) \\ 2(q_{vx} q_{vz} - q_s q_{vy}) & 2(q_{vy} q_{vz} + q_s q_{vx}) & 2(q_s^2 + q_{vz}^2) - 1 \end{bmatrix}.$$

Entretanto, quando é desejado resolver o *problema inverso*, ou seja, computar o quatérnio correspondente a uma matriz de rotação

$$R(q_s, q_v) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \quad (\text{B.4})$$

o seguinte resultado é útil:

$$q_s = \frac{1}{2} \sqrt{\text{tr}(R) + 1}, \quad (\text{B.5})$$

$$q_v = \frac{1}{2} \begin{bmatrix} \text{sgn}(r_{32} - r_{23}) \sqrt{r_{11} - r_{22} - r_{33} + 1} \\ \text{sgn}(r_{13} - r_{31}) \sqrt{r_{22} - r_{33} - r_{11} + 1} \\ \text{sgn}(r_{21} - r_{12}) \sqrt{r_{33} - r_{11} - r_{22} + 1} \end{bmatrix}, \quad (\text{B.6})$$

onde, por convenção,  $\text{sgn}(x) = 1$  para  $x \geq 0$  e  $\text{sgn}(x) = -1$  para  $x < 0$ . Nota-se que em (B.5), é implicitamente assumido que  $q_s \geq 0$ ; isso corresponde a um ângulo  $\vartheta \in [-\pi, \pi]$ , e portanto qualquer rotação pode ser descrita. Além disso, as soluções para  $q_s$  e  $q_v$  são livres de singularidades. O quatérnio extraído de  $R^{-1} = R^T$  é denominado  $q^{-1}$  e pode ser computado como:

$$q^{-1} = (q_s, -q_v). \quad (\text{B.7})$$

Considerando-se  $q_1 = ((q_s)_1, (q_v)_1)$  e  $q_2 = ((q_s)_2, (q_v)_2)$  os quatérnios correspondentes às matrizes de rotação  $R_1$  e  $R_2$  respectivamente, o quatérnio correspondente ao produto  $R_1 R_2$  é dado por:

$$q_1 * q_2 = ((q_s)_1 (q_s)_2 - (q_v)_1^T (q_v)_2, (q_s)_1 (q_v)_2 + (q_s)_2 (q_v)_1 + Q((q_v)_1) (q_v)_2), \quad (\text{B.8})$$

onde “\*” denota o operador produto de quatérnio. É fácil verificar que se  $q_2 = q_1^{-1}$ , então o resultado obtido a partir de (B.8) é denominado elemento identidade para o produto:

$$q_1 * q_2 = q_1 * q_1^{-1} = (1, 0). \quad (\text{B.9})$$

# Apêndice C

## Propriedades das Elipses

Para delimitar os elos e as juntas do robô por uma sequência de elipsoides, as seguintes propriedades são utilizadas:

- A soma das distâncias entre qualquer ponto ( $P_1$ ) localizado na elipse e os focos  $F_1$  e  $F_2$  é constante e igual ao maior eixo ( $2a$ ) cuja equação é descrita a seguir:

$$\overline{P_1F_1} + \overline{P_1F_2} = 2a. \quad (\text{C.1})$$

- A distância entre os pontos focais e o centro da elipse é chamada de excentricidade linear  $f$ . Utilizando a propriedade anterior e o Teorema de Pitágoras, a seguinte propriedade é definida:

$$(\overline{P_1F_1})^2 = b^2 + f^2, \quad (\text{C.2})$$

e sabendo que  $\overline{P_1F_1} = \overline{P_1F_2}$  e  $2\overline{P_1F_1} = 2a$  obtém-se:

$$a^2 = b^2 + f^2. \quad (\text{C.3})$$

- Cada foco está associado a uma linha paralela ao menor eixo denominada de diretriz ( $C$ ). A razão  $\xi$  entre a distância do foco ( $F_2$ ) a um ponto da elipse ( $P_1$ ) qualquer e a distância perpendicular entre o ponto ( $P_1$ ) e a diretriz é constante e menor que 1, ou seja:

$$\xi = \frac{\overline{P_1F_2}}{\overline{P_1C}} < 1. \quad (\text{C.4})$$