

Universidade Estadual de Campinas

Leonardo Rocha Olivi

NAVEGAÇÃO DE ROBÔS MÓVEIS ASSISTIVOS POR CONTROLE  
COMPARTILHADO BASEADO EM CAMPOS VETORIAIS

Campinas  
2014



Universidade Estadual de Campinas  
Faculdade de Engenharia Elétrica e de Computação

Leonardo Rocha Olivi

NAVEGAÇÃO DE ROBÔS MÓVEIS ASSISTIVOS POR  
CONTROLE COMPARTILHADO BASEADO EM CAMPOS VETORIAIS

Tese de doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos exigidos para a obtenção do título de Doutor em Engenharia Elétrica.

Área de concentração: Automação.

Orientador: Prof. Dr. Eleri Cardozo

Este exemplar corresponde à versão final da tese defendida pelo aluno Leonardo Rocha Olivi, e orientada pelo Prof. Dr. Eleri Cardozo

---

Campinas  
2014

Ficha catalográfica  
Universidade Estadual de Campinas  
Biblioteca da Área de Engenharia e Arquitetura  
Rose Meire da Silva - CRB 8/5974

OL4m Olivi, Leonardo Rocha, 1982-  
Navegação de robôs móveis assistivos por controle compartilhado baseado em campos vetoriais / Leonardo Rocha Olivi. – Campinas, SP : [s.n.], 2014.

Orientador: Eleri Cardozo.  
Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Campos vetoriais. 2. Navegação de robôs móveis. 3. Robótica na medicina.  
I. Cardozo, Eleri, 1954-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

**Título em outro idioma:** Navigation of assistive mobile robots by shared control based on vector fields

**Palavras-chave em inglês:**

Vector fields

Navigation of mobile robots

Medical robotics

**Área de concentração:** Automação

**Titulação:** Doutor em Engenharia Elétrica

**Banca Examinadora:**

Eleri Cardozo [Orientador]

Teodiano Freire Bastos Filho

Glauco Augusto de Paula Caurin

Paulo Roberto Gardel Kurka

José Raimundo de Oliveira

**Data de defesa:** 18-11-2014

**Programa de Pós-Graduação:** Engenharia Elétrica



# COMISSÃO JULGADORA - TESE DE DOUTORADO

**Candidato:** Leonardo Rocha Olivi

**Data da Defesa:** 18 de novembro de 2014

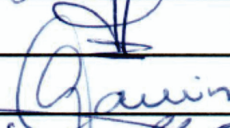
**Título da Tese:** "Navegação de Robôs Móveis Assistivos por Controle Compartilhado Baseado em Campos Vetoriais"

Prof. Dr. Eleri Cardozo (Presidente): \_\_\_\_\_

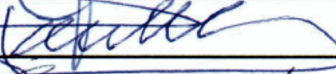


Prof. Dr. Teodiano Freire Bastos Filho: \_\_\_\_\_

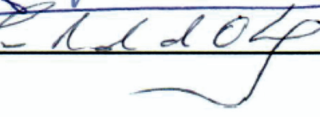
Prof. Dr. Glauco Augusto de Paula Caurin: \_\_\_\_\_



Prof. Dr. Paulo Roberto Gardel Kurka: \_\_\_\_\_



Prof. Dr. José Raimundo de Oliveira: \_\_\_\_\_





# Resumo

A mobilidade é uma competência humana extremamente valiosa, e pode ser perdida por diversas razões, tais como traumas na coluna vertebral, acidentes vasculares cerebrais, dentre outras. Algumas tecnologias desenvolvidas para as áreas médicas, como eletroencefalografia e eletromiografia, podem ser empregadas no desenvolvimento de Interfaces Humano-Máquina (*Human-Machine Interface*, ou HMI) com o objetivo de permitir que pessoas com capacidades motoras severamente comprometidas possam interagir com os mais diversos equipamentos. Por exemplo, esses usuários podem conduzir robôs móveis por meio de uma HMI adaptada para suas capacidades motoras e cognitivas. Entretanto, essas interfaces apresentam erros na identificação da ação pretendida pelo usuário, os quais comprometem sobremaneira a experiência desse usuário na interação com o mundo exterior. Para o caso dos robôs móveis assistivos, o conceito de controle compartilhado (*shared control*) foi concebido para compensar as taxas de erros das HMIs por meio da divisão de responsabilidade pela navegação do robô entre o usuário e um sistema supervisor autônomo. As abordagens mais comuns na literatura propõem estratégias de comutação entre ações do usuário e ações autônomas em situações pré-definidas tais como passagens estreitas e presença de obstáculos. Essa comutação é fonte de instabilidades pela impossibilidade de se prever todas as situações que o robô poderá encontrar. Esta tese propõe uma nova técnica de controle compartilhado, denominada de navegação assistida por campos vetoriais, a qual não possui modos de operação comutados. Nesta nova técnica, o sistema de controle embarcado sempre irá executar a ação identificada pela HMI, mas em função dos estados do robô e do ambiente e visando principalmente a segurança do usuário. Os resultados mostrados nesta tese evidenciam ganhos significativos com relação ao estado da arte, tornando a técnica proposta uma solução eficaz para o problema de navegação de robôs móveis assistivos.

**Palavras-chave:** campos vetoriais, navegação de robôs móveis, robótica na medicina.



# Abstract

Human mobility is an extremely valued skill that can be lost due to various reasons, such as spinal cord injuries, strokes, amputations, among others. Technologies developed for the medical areas, such as electroencephalography and electromyography, can be employed in the development of Human-Machine Interfaces (HMI) with the objective of allowing people with severe mobility impairments to communicate and to operate a wide range of equipments. Therefore, these users are allowed to conduct assistive robots allow through a HMI designed according to the user's mobility and cognitive skills. However, these interfaces still present low performance in the correct identification of the intended action by the user, which severely compromise the experience of the user when interacting with external world. In the specific case of assistive mobile robots, a shared control concept was developed in order to compensate the high error rates produced by the HMIs, where the responsibility for the navigation of the mobile robot is shared among the user and an autonomous supervisor system. The mainly approaches shown in literature switch a few modes of specific behavior for individualized situations, such as narrow corridors and obstacle avoidance. This work presents a novel shared control technique, named assistive navigation by vector fields, which does not employ switching modes, avoiding any instabilities and covering the unlimited environment configurations, significantly minimizing the HMI errors, facilitating the user's navigation. The vector fields offer both manual and autonomous navigation, increasing the user's interaction with the environment. In this novel technique, the embedded control system incorporates the HMI command with the robot and environment states, aiming the maximization of user's security and control capabilities. Results shown a novel manner for treating the problem, obtaining substantial gains over the state of art works, with low computational complexity, high flexibility concerning environments and users and effective optimization of errors, contributing for the user's mobility retrieval.

**Keywords:** vector fields, mobile robot's navigation, medical robotics.



# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Fundamentação técnica</b>	<b>7</b>
2.1	Comunicação humano-máquina assistiva . . . . .	7
2.1.1	Sinais biológicos . . . . .	8
2.1.2	Eletromiografia . . . . .	9
2.1.3	Interfaces cérebro-computador . . . . .	10
2.2	Robótica móvel assistiva . . . . .	13
2.3	Controle compartilhado . . . . .	16
2.4	Integração das navegações autônoma e compartilhada . . . . .	20
<b>3</b>	<b>Comunicação humano-máquina assistiva</b>	<b>23</b>
3.1	Interface principal . . . . .	25
3.2	Interface de controle compartilhado . . . . .	27
3.2.1	Controle compartilhado via sEMG . . . . .	27
3.2.2	Controle compartilhado via BCI . . . . .	29
3.3	Interface de controle autônomo . . . . .	31
3.4	Discussões . . . . .	33
<b>4</b>	<b>Navegação por controle compartilhado</b>	<b>39</b>
4.1	Navegação assistida . . . . .	41
4.2	Campo vetorial para controle da velocidade linear . . . . .	44
4.3	Campo vetorial para controle dos ângulos de giro . . . . .	47
4.4	Mapeamento durante a navegação assistida . . . . .	50
4.5	Pseudocódigo do método de navegação proposto . . . . .	52
4.6	Calibração dos parâmetros . . . . .	53
4.7	Discussões . . . . .	54
<b>5</b>	<b>Navegação autônoma</b>	<b>57</b>
5.1	Armazenamento do mapa . . . . .	58
5.2	Algoritmo de busca . . . . .	59
5.3	Controlador autônomo . . . . .	62

5.4	Análise da estabilidade do controlador por campos vetoriais . . . . .	63
5.5	Manutenção dos mapas e recálculo de rotas . . . . .	66
5.6	Discussões . . . . .	68
<b>6</b>	<b>Avaliação Experimental</b>	<b>71</b>
6.1	Experimentos de navegação assistida . . . . .	72
6.1.1	Navegação por um corredor estreito . . . . .	72
6.1.2	Navegação pelo diagrama de Voronoi . . . . .	75
6.1.3	Ganho de desempenho na utilização do campo atrativo . . . . .	77
6.1.4	Minimização de erros dos classificadores . . . . .	80
6.1.5	Navegação por ambientes de grandes dimensões . . . . .	87
6.1.6	Utilização de sensores de menor resolução . . . . .	90
6.2	Experimentos de navegação autônoma . . . . .	93
6.2.1	Análise do controlador autônomo com campos vetoriais . . . . .	93
6.2.2	Comparação com controlador PID . . . . .	99
6.2.3	Análise do recálculo de rotas . . . . .	102
6.3	Avaliação das implementações do projeto DesTINE . . . . .	106
6.4	Navegação em ambientes de grandes dimensões . . . . .	108
<b>7</b>	<b>Conclusões</b>	<b>111</b>
	<b>Bibliografia</b>	<b>115</b>
<b>A</b>	<b>Contribuições do autor</b>	<b>123</b>



# Agradecimentos

Agradeço,

ao Prof. Eleri Cardozo, cuja valorosa orientação permitiu minha evolução intelectual e pessoal;

aos amigos do grupo de robótica: Eric Rohmer, Ricardo Souza, Fernando Paolieri, Fernando Pinho, Guilherme Feliciano, Lucio Rocha, Diego Rodrigues, Fábio Teixeira e Luisa Uribe;

aos amigos e professores do Laboratório de Computação e Automação Industrial (LCA);

à agência CAPES, o apoio financeiro concedido durante todo o período de doutoramento;

à FEEC/UNICAMP, a ótima estrutura que oferece aos estudantes e pesquisadores;

a meus pais, Wilma e João, pelo amor e apoio incondicionais;

às minhas avós, Hilda e Elvira (*in memorian*), sem as quais não chegaria aqui;

aos meus tios, Marina e Luigi, e meu primo Ermanno;

aos meus tios, Martha (*in memorian*), Maria Silvia e Roberto, e as minhas primas Mariana e Heloísa;

à minha querida Regina (Nina), por dividir uma vida de amor e carinho comigo;

a todos que, de alguma forma, contribuíram para a realização deste trabalho.



# Lista de Figuras

1.1	Robôs utilizados nesta tese: (a) Pioneer P3-DX e (b) Seekur Jr. . . . .	3
2.1	O aparelho (a) Emotiv e seu (b) SDK, para leitura de EMG facial. . . . .	9
2.2	(a) Região cerebral de ocorrência e (b) exemplo de sinal P300. . . . .	11
2.3	(a) Sistema Internacional 10-20 e (b) aparelho g.tec. . . . .	12
2.4	Etapas de um sistema de BCI (exemplificado com um aparelho g.tec). . . . .	12
2.5	Cadeira de rodas robótica, com diversos componentes. . . . .	13
2.6	Cadeiras robóticas de (a) Levine et al. (1999) e (b) Phillips et al. (2007). . . . .	16
2.7	Cadeiras robóticas de (a) Trieu et al. (2008) e (b) Iturrate et al. (2009). . . . .	17
2.8	Robôs utilizados por (a) Tonin et al. (2010) e (b) Cruz et al. (2011). . . . .	18
2.9	Cadeiras robóticas de (a) Li et al. (2011) e (b) Carlson et al (2012). . . . .	19
3.1	A interface principal ( <i>main interface</i> ) de comunicação. . . . .	25
3.2	Método de obtenção dos pulsos P300. . . . .	26
3.3	A interface de controle compartilhado via EMG. . . . .	28
3.4	A interface de controle compartilhado via BCI/P300. . . . .	30
3.5	A interface de controle autônomo. . . . .	32
3.6	A interface de controle <i>point-to-go</i> desenvolvida no projeto DesTINE. . . . .	33
4.1	Ilustração de controle compartilhado . . . . .	39
4.2	(a) Configuração padrão e (b) modificação de comportamento . . . . .	42
4.3	Campos vetoriais atuando sobre o robô Pioneer P3-DX. . . . .	43
4.4	(a) Área coberta pelo <i>laser</i> (b) pose do robô associada a um vetor do campo. . . . .	45
4.5	Cálculo dos vetores utilizados no campo repulsivo. . . . .	46
4.6	Áreas seguras de um ambiente. . . . .	48
4.7	(a) Robô imerso no campo atrativo $\mathbf{N}$ e (b) obtenção do vetor $\vec{n}$ . . . . .	49
4.8	Ambiente simulado para treinamento de usuários . . . . .	53
5.1	Interação entre o usuário e o controle autônomo . . . . .	57
5.2	(a) Mapa métrico e (b) mapa de células, com perda de resolução. . . . .	59
5.3	Política proposta para o mapa do ambiente. . . . .	61
5.4	Ângulo utilizado na lei de controle do controlador autônomo. . . . .	64
5.5	Transformação polar para o modelo do robô diferencial. . . . .	65

5.6	Controle autônomo de posição por campos vetoriais. . . . .	66
5.7	(a) Detecção de anormalidade nas células e (b) estado das células alterado. . . .	67
5.8	Recálculo de rotas obstruídas. . . . .	68
6.1	Navegação por um corredor estreito. . . . .	73
6.2	Resultados qualitativos da navegação com campo repulsivo. . . . .	74
6.3	Resultados para o rastreamento do diagrama de Voronoi. . . . .	75
6.4	Desvio reduzido pelo campo vetorial atrativo (a) perto e (b) longe do diagrama.	77
6.5	Resultados do experimento (a) com e (b) sem o campo vetorial atrativo. . . . .	78
6.6	Erros do experimento (a) com e (b) sem o campo vetorial atrativo. . . . .	79
6.7	Taxa de acerto de 60%: resultados com e sem o método de campos vetoriais . .	82
6.8	Taxa de acerto de 70%: resultados com e sem o método de campos vetoriais . .	83
6.9	Taxa de acerto de 80%: resultados com e sem o método de campos vetoriais . .	84
6.10	Taxa de acerto de 90%: resultados com e sem o método de campos vetoriais . .	85
6.11	Mapeamento do LCA. . . . .	88
6.12	Alguns ambientes do LCA. . . . .	89
6.13	Disposição dos sonares no robô Pioneer P3-DX. . . . .	90
6.14	Situação 1: mapa local com diagrama de Voronoi . . . . .	91
6.15	Situação 2: mapa local com diagrama de Voronoi . . . . .	92
6.16	Situação 3: mapa local com diagrama de Voronoi . . . . .	92
6.17	Caminhos utilizados na avaliação do controlador autônomo. . . . .	94
6.18	Erros do controlador ao navegar pelos caminhos (a) 1, (b) 2 e (c) 3. . . . .	94
6.19	Velocidades para o caminho 1. . . . .	96
6.20	Velocidades para o caminho 2. . . . .	97
6.21	Velocidades para o caminho 3. . . . .	98
6.22	Erros do controlador PID ao navegar pelos caminhos 1, 2 e 3. . . . .	101
6.23	Velocidades angulares do controlador PID ao navegar pelos caminhos. . . . .	101
6.24	Recálculo de rotas com $A^*$ dinâmico. . . . .	103
6.25	Erros produzidos pelo controlador autônomo. . . . .	103
6.26	Recálculo de rotas com $A^*$ dinâmico para o segundo caminho. . . . .	105
6.27	Erros produzidos pelo controlador autônomo para o caminho 2. . . . .	105
6.28	Rotas dos métodos desenvolvidos no projeto DesTINe. . . . .	107
6.29	Mapeamento do segundo andar da FEEC/UNICAMP. . . . .	109
7.1	Vistas frontal, lateral e traseira da cadeira, mostrando seus vários dispositivos. .	114

# Lista de Tabelas

3.1	Classificação facial usando Emotiv. . . . .	28
3.2	Classificação de quatro opções usando o sistema BCI2000. . . . .	31
6.1	Valores dos parâmetros usados nos experimentos com robôs reais. . . . .	72
6.2	Resultados quantitativos para a navegação no corredor estreito. . . . .	73
6.3	Resultados para a navegação pelo diagrama de Voronoi. . . . .	75
6.4	Resultados para a avaliação com (esquerda) e sem (direita) o campo atrativo. . .	79
6.5	Resultados para acerto de 60% com (esquerda) e sem (direita) campos vetoriais.	82
6.6	Taxa de acerto de 70%: resultados com e sem o método de campos vetoriais . .	83
6.7	Taxa de acerto de 80%: resultados com e sem o método de campos vetoriais . .	84
6.8	Taxa de acerto de 90%: resultados com e sem o método de campos vetoriais . .	85
6.9	Navegação e mapeamento do LCA. . . . .	87
6.10	Resultados para o controlador por campos vetoriais. . . . .	95
6.11	Resultados para o controlador PID. . . . .	100
6.12	Desempenho do controlador para o caminho original. . . . .	104
6.13	Desempenho do controlador para o caminho recalculado. . . . .	104
6.14	Desempenho do controlador para o segundo caminho. . . . .	104
6.15	Desempenho do controlador para o segundo caminho recalculado. . . . .	105
6.16	Resultados de comparação entre os métodos do projeto DesTINe. . . . .	107



# Lista de Abreviaturas e Siglas

EMG	Eletromiografia
sEMG	Eletromiografia de superfície (não invasiva)
EEG	Eletroencefalografia
MEG	Magnetoencefalografia
MRI	<i>Magnetic Resonance Imaging</i> (Ressonância Magnética)
fMRI	<i>Functional Magnetic Resonance Imaging</i> (Ressonância Magnética Funcional)
NIRS	<i>Near-Infrared Spectroscopy</i> (Espectroscopia de Infravermelho Próximo)
HMI	<i>Human-Machine Interface</i> (Interface Homem-Máquina)
BCI	<i>Brain-Computer Interface</i> (Interface Cérebro-Computador)
SLAM	<i>Simultaneous Localization And Mapping</i> (Localização e Mapeamento Simultâneos)
SDK	<i>Software Development Kit</i> (Localização e Mapeamento Simultâneos)
SSVEP	<i>Steady State Visually Evoked Potentials</i> (Potencial Evocado Visual em Regime Permanente)
VFH	<i>Vector Field Histogram</i> (Histograma de Campos Vetoriais)
VFF	<i>Vector Force Field</i> (Campos de Forças Vetoriais)
RNA	Rede Neural Artificial
SNR	<i>Signal-to-Noise Ratio</i> (Relação Sinal-Ruído)
API	<i>Application Programming Interface</i> (Interface de Programação de Aplicativos)
ERS	<i>Event-Related Synchronization</i> (Sincronização Relacionada a Eventos)
ERD	<i>Event-Related Desynchronization</i> (Dessincronização Relacionada a Eventos)
SWLDA	<i>Stepwise Linear Discriminant Analysis</i> (Análise de Discriminante Linear Passo a passo)
CAR	<i>Common Average Reference</i> (Referência Média Comum)
EKF	<i>Extended Kalman Filter</i> (Filtro de Kalman Estendido)
RFID	<i>Radio-Frequency Identification</i> (identificação por radiofrequência)





## Introdução

Mobilidade pode ser definida como a capacidade de uma pessoa se deslocar e manipular objetos. Trata-se de uma das competências humanas mais valorizadas, permitindo ao indivíduo desenvolver as atividades físicas que desejar. De fato, a perda deste atributo, como é de conhecimento geral, pode ser acompanhada de frustrações e depressão, problemas que também podem se estender para o seu círculo pessoal.

Pessoas podem perder sua mobilidade por diversas razões. Há indivíduos com problemas inatos, tais como malformações ou doenças congênitas; doenças graduais, como esclerose múltipla e osteoartrite; causas súbitas, tais como traumas na coluna vertebral, acidentes vasculares cerebrais, amputações de membros; dentre muitas outras. Estas ocorrências são todas graves a ponto de paralisar parcial ou totalmente o indivíduo.

No Brasil, de acordo com o censo do Instituto Brasileiro de Geografia e Estatística (IBGE 2010), 23,9% da população brasileira (o que corresponde, atualmente, a cerca de 45,5 milhões de pessoas) têm algum tipo de deficiência, seja ela visual, motora, cognitiva, dentre outras. O mesmo censo aponta que a privação motora atinge 7% da população brasileira, aproximadamente 13,2 milhões de indivíduos. Tratam-se de percentuais bastante elevados, mostrando que uma parcela significativa da população demanda cuidados especiais.

Nos casos envolvendo deficiência motora, a recuperação da mobilidade é uma meta imperativa e um ponto de interesse deste trabalho. Existem casos em que se pode restabelecê-la por meio de procedimentos médicos e fisioterápicos. No entanto, há situações bastante agravadas, em que o indivíduo é confinado permanentemente numa cadeira de rodas. Em todos esses casos, de alguma forma, várias tecnologias podem ser empregadas para auxiliar na recuperação da mobilidade

Dentre as várias tecnologias existentes, a robótica pode exercer um papel importante por meio de sua área de pesquisa voltada para a acessibilidade. Para pessoas com deficiência motora severa, a Robótica Móvel Assistiva é um campo de investigação crescente, que tem o propósito de melhorar a mobilidade pessoal e, conseqüentemente, a qualidade de vida destas pessoas, promovendo sua inclusão no meio social.

Seguindo a linha de pesquisa relatada, o foco principal deste trabalho de pesquisa está no desenvolvimento de novas tecnologias robóticas voltadas para a melhoria da acessibilidade de pessoas que utilizam cadeiras de rodas, tornando-as usuárias de sistemas robóticos assistivos.

Uma grande quantidade de cadeiras de rodas robóticas vem sendo desenvolvida por grupos de pesquisa, tanto em empresas quanto na academia, em várias partes do mundo. É possível encontrar uma grande variedade de abordagens para a interação entre o usuário e o robô, variando desde simples controles de velocidades a complexos comportamentos autônomos. O objetivo desses trabalhos é englobar na cadeira robótica algumas das tarefas diárias de uma pessoa, de forma que a tecnologia possa assisti-la na execução dessas tarefas. Abordagens mais comuns consideram tarefas corriqueiras como cruzar passagens estreitas e desviar de obstáculos.

Um outro aspecto na robótica móvel assistiva trata justamente da comunicação entre o robô e o usuário. Uma vez que este se encontra debilitado, é necessário desenvolver um sistema capaz de extrair as vontades e necessidades da pessoa para que esta informação sirva de entrada ao sistema robótico. Incapaz de fornecer alguns dos comandos normais, tal qual a operação de um controle (*joystick*), os sinais biológicos adquiridos do usuário são uma boa fonte de informações a seu respeito, sendo que os mais usuais na literatura são sinais gerados por atividade muscular (quando se dispõe de um ou mais tipos de movimentos, como os faciais) e por atividade cerebral (Millán et al. 2010, Cruz et al. 2011, Cowan et al. 2012, Bastos-Filho et al. 2014).

A obtenção de sinais corporais, técnicas comumente utilizadas na área médica, pode se mostrar útil para a robótica assistiva. Para a captação dos sinais musculares utiliza-se a eletromiografia de superfície (sEMG) e para os sinais cerebrais a eletroencefalografia (EEG), ambas por meio de eletrodos. Estas técnicas podem ser invasivas, quando os eletrodos são inseridos no corpo do usuário, ou não-invasivas, com eletrodos superficiais. Ainda que sEMG e EEG sejam as abordagens mais utilizadas, conforme veremos nos capítulos seguintes, existem várias outras como a detecção de movimentos faciais e oculares por visão computacional ou acelerometria e detecção do fluxo respiratório, além das combinações entre essas técnicas (abordagens híbridas).

Os métodos de captação de sinais biológicos são empregados na construção de Interfaces Humano-Máquina (HMI, do inglês *Human-Machine Interface*) assistivas. Em se tratando de sinais cerebrais, há um segmento da área denominado Interface Cérebro-Computador (BCI, do inglês *Brain-Computer Interface*). A extração e classificação de sinais biológicos em comandos envolve muitas áreas do conhecimento, tais como processamento e filtragem de sinais, reconhecimento de padrões, aprendizado de máquina, dentre outras. Um maior detalhamento das HMIs será apresentado nos Capítulos 2 e 3 desta tese.

As equipes que pesquisam a área de interação entre o ser humano e a máquina, tais como técnicas para a classificação de sinais biológicos (sEMG, BCI e outros), encontram dificuldades em diversas etapas. Neste ramo, especialmente em abordagens não-invasivas, há uma grande interferência de ruídos, pois a relação sinal-ruído (SNR, do inglês *signal-to-noise ratio*) é, geralmente, baixa, indicando uma grande contaminação do sinal pelo ruído. A fadiga muscular/mental pelo uso prolongado do sistema é outro fator limitante, além de outros fatores que oferecem grandes dificuldades para a classificação correta dos padrões e comandos enviados pelo usuário.

Outras pesquisas se concentram na área de controle do equipamento robótico, investindo no desenvolvimento e integração de diversas técnicas para assegurar o cumprimento das tarefas atribuídas ao robô. Uma das grandes preocupações é a garantia da segurança do usuário durante toda a navegação, uma vez que este não pode intervir fisicamente em resposta a eventuais acidentes causados por falhas de segurança.

O objetivo desta tese é contribuir com a área de navegação robótica no contexto da robótica móvel assistiva. Nesta pesquisa, uma cadeira de rodas robótica foi emulada pelos robôs móveis Pioneer P3-DX e Seekur Jr., ambos fabricados pela empresa Adept MobileRobots (MobileRobots 2014) e mostrados na Figura 1.1, pois possuem a mesma estrutura. Este trabalho é parte integrante do projeto DesTINE (Desenvolvimento de Tecnologias da Informação para Neurologia), desenvolvido na Universidade Estadual de Campinas (UNICAMP), e patrocinado pelas agências de fomento à pesquisa brasileiras Financiadora de Estudos e Projetos (FINEP) e Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

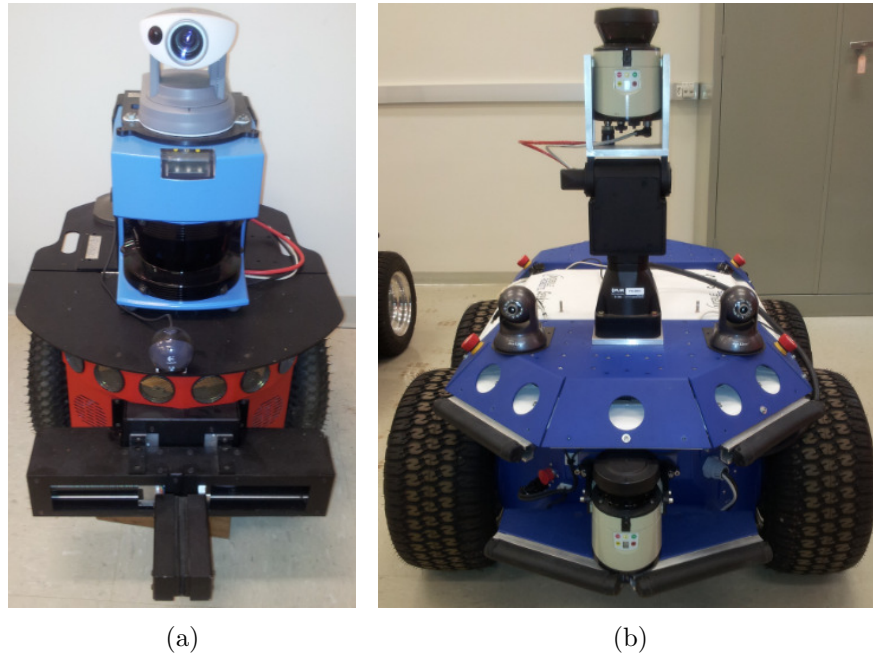


Figura 1.1: Robôs utilizados nesta tese: (a) Pioneer P3-DX e (b) Seekur Jr.

O projeto DesTINE iniciou em 2010 com a submissão de proposta à FINEP para uma chamada em Tecnologias Assistivas lançada por esta agência de fomento em 2009. O projeto teve duração de 4 anos e participação das seguintes instituições: Universidade Estadual de Campinas, Centro de Tecnologias da Informação Renato Archer, Instituto Venturus e Universidade Federal do ABC. Da UNICAMP participam a Faculdade de Engenharia Elétrica e Computação (FEEC), o Instituto de Física Gleb Wataghin (IFGW) e a Faculdade de Ciências Médicas (FCM). A equipe de pesquisa é altamente interdisciplinar composta de especialistas em neurociências, imagens médicas, processamento digital de sinais e robótica. O projeto tem como meta central o desenvolvimento de tecnologias da informação para auxiliar pessoas com severas deficiências de comunicação e locomoção causadas por doenças neurológicas, notadamente o acidente vascular cerebral (AVC).

A principal contribuição deste trabalho está na proposição e desenvolvimento de um sistema de controle para robôs assistivos, sendo este dividido em controle compartilhado e autônomo. O controle compartilhado (do inglês, *shared control*) representa a situação em que o usuário estabelece a trajetória do robô por meio de comandos diretos, como “vire à direita”, “vire à esquerda”, “acelere”, da mesma forma que faria ao utilizar um *joystick* ou teclado de computador.

A diferença substancial entre o controle manual e o compartilhado é que, no controle compartilhado, há um sistema supervisor que auxilia o usuário na condução do robô. Levando em consideração que o usuário é uma pessoa com deficiência, esse sistema supervisor se faz necessário para propiciar segurança na navegação. Por outro lado, tendo em vista a dificuldade de classificação de sinais dos sistemas sEMG/BCI, o controle compartilhado também deve atuar de forma a minimizar os efeitos negativos de possíveis comandos errôneos emitidos tanto pelo próprio usuário ou incorretamente classificados pela HMI.

Assim, os processos do sistema, tais como os comandos escolhidos pelo usuário, a execução dos comandos, a proteção e conforto do usuário, bem como vários outros aspectos relativos à navegação, ficam sob incumbência de um sistema de controle compartilhado, que atua junto ao usuário, de maneira a protegê-lo e melhorar suas experiências de navegação. Neste trabalho de pesquisa, propõe-se uma nova técnica de controle compartilhado, a qual foi denominada por “navegação assistida”, e baseada na utilização de campos vetoriais, conforme relatado em Olivi et al. (2013) e Olivi et al. (2014).

Ainda sobre a contribuição deste trabalho de pesquisa, tem-se a abordagem desenvolvida para controle autônomo em que, o usuário, por meio da HMI assistiva, deve escolher uma localização (sala, escritório etc.) previamente conhecida (mapeada) como ponto-objetivo. A partir do momento em que o robô possui um objetivo, vários algoritmos de controle e busca são empregados para que a tarefa seja cumprida autonomamente, isto é, sem requisitar decisões adicionais do usuário, o qual pode ocupar-se com outras atividades durante a navegação.

Para este tipo de abordagem, um conjunto de técnicas devem ser desenvolvidas e integradas para o cumprimento dos objetivos. Como exemplos dessas técnicas, cita-se Localização e Mapeamento Simultâneos (SLAM, do inglês *Simultaneous Localization And Mapping*), métodos de busca iterativa em mapas para o planejamento de caminhos (do inglês, *path planning*), técnicas de controle de movimentos (do inglês, *path following*), heurísticas de proteção, dentre outras. Algumas destas técnicas utilizadas nesta tese foram previamente publicadas em Olivi et al. (2012).

Para o desenvolvimento do comportamento autônomo, utilizaram-se técnicas de busca em mapas já conhecidas na literatura, tais como os algoritmos  $A^*$  e  $D^*$ , cuja tarefa é obter caminhos entre a posição atual do robô e seu ponto-objetivo designado pelo usuário. Ainda assim, é necessário observar que os ambientes diários são altamente dinâmicos, portanto, deve haver a preocupação com a manutenção dos mapas, de forma a atualizar as rotas sempre que necessário.

Outro aspecto importante na abordagem autônoma remete à movimentação do robô pelos caminhos encontrados por planejadores de caminhos. Para realizar essa tarefa no contexto da pesquisa desenvolvida, o controle compartilhado passa a ser comandado integralmente pela unidade computacional. Ou seja, o papel do usuário em direcionar o robô é assumido pela máquina, que se comportará como uma HMI perfeita, sem interferências dos ruídos e erros de classificação advindos das técnicas de sEMG/BCI. Por fim, a obtenção de mapas que representem com fidelidade o ambiente são obtidos por meio de SLAM ou inseridos a priori no sistema.

O presente texto está dividido em sete capítulos. O Capítulo 2 contém a fundamentação dos principais assuntos abordados, tais como técnicas de obtenção dos sinais do usuário e controle de robôs móveis assistivos, bem como sua evolução e desenvolvimentos atuais.

Além da fundamentação da área, o Capítulo 2 tem como objetivo situar a proposta desse trabalho de pesquisa no contexto acadêmico, estabelecendo sua contribuição.

O Capítulo 3 apresenta detalhes da HMI assistiva desenvolvida no âmbito desta pesquisa. Diversas interfaces foram desenvolvidas no Projeto DesTINe para a aquisição de comandos do usuário. Detalhes sobre as interfaces desenvolvidas, bem como os resultados alcançados, serão apresentados nesse capítulo.

O Capítulo 4 descreve a proposta de controle compartilhado utilizando uma nova técnica desenvolvida nessa tese, a navegação assistida por campos vetoriais. A partir das informações obtidas dos sensores de distância criam-se comportamentos para o robô assistivo, de forma a auxiliar o usuário a navegar por diversos ambientes, conhecidos ou não. Nenhuma ação autônoma é requerida nesse método, denotando uma navegação totalmente controlada pelo usuário.

O Capítulo 5 apresenta a proposta de controle autônomo, apresentando todas as técnicas desenvolvidas e utilizadas, bem como sua integração. Atualização de mapas, obtenção de rotas navegáveis e seguras para o robô e controle de movimentos especificamente projetados para ambientes assistivos são os temas abordados neste capítulo.

O Capítulo 6 se dedica a apresentar todos os resultados dos diversos experimentos planejados para avaliar as metodologias desenvolvidas nos capítulos 4 e 5, além das contribuições alcançadas para a área de robótica móvel assistiva por meio desta pesquisa.

Finalmente, o Capítulo 7 apresenta as conclusões finais, bem como as perspectivas de trabalhos futuros.



## Fundamentação técnica

Neste capítulo são abordados os principais tópicos concernentes à área de robótica móvel assistiva, mostrando sua evolução e desenvolvimentos atuais. O primeiro tópico se refere à comunicação do usuário com o sistema robótico por meio da interface humano-máquina assistiva, envolvendo o processo de extração de comandos do usuário, o aprendizado e a utilização de plataformas já consolidadas na área, de forma a obter uma via de comunicação entre o usuário e o sistema robótico.

Na sequência, são apresentados os aspectos relativos ao controle de uma cadeira de rodas robótica. Discutem-se as diferentes metodologias de navegação, tanto compartilhadas quanto autônomas, com a finalidade de cumprir os comandos adquiridos do usuário. Por fim, situa-se a contribuição desta tese, com comentários a respeito do método proposto e suas justificativas.

### 2.1 Comunicação humano-máquina assistiva

O processo de obtenção de vias de comunicação para pessoas cujas capacidades motoras estejam severamente debilitadas é um processo árduo. Cada indivíduo é um caso a parte, apresentando diferentes tipos de impedimentos. Conforme mencionado no capítulo anterior, existem debilidades bastante graves que podem paralisar o indivíduo quase totalmente.

Para a utilização das tecnologias desenvolvidas neste trabalho, considera-se o caso de pessoas portadoras de deficiências que, ao menos, possuam um nível de competência cerebral suficiente para tomadas de decisões coerentes, isto é, conscientes. Desta forma, é possível utilizar seus sinais cerebrais por meio de EEG. Há casos em que alguns grupos musculares, como os faciais, ainda estão preservados, permitindo o emprego de sEMG. Existem ainda opções diferentes, como por exemplo, o rastreamento de olhos por meio de câmeras, comandos por voz (quando aplicável ao usuário), dentre outros, além da combinação de técnicas, formando sistemas híbridos (Lee et al. 2013, Bastos-Filho et al. 2014).

O ponto de partida em tecnologias assistivas é a prévia análise das condições do usuário. Assim, é possível ponderar a estratégia que maximize a interação entre o usuário e a máquina, até mesmo com a personalização de arquiteturas para um determinado indivíduo. Nesta tese considera-se a possibilidade da utilização tanto de sEMG quanto de EEG, visto que são as opções mais difundidas na literatura (Millán et al. 2010, Bi et al. 2013, Bastos-Filho et al. 2014).

### 2.1.1 Sinais biológicos

A principal comunicação de um indivíduo com deficiência que impede a manipulação de interfaces de interação convencionais é por meio de seus sinais biológicos. Vários tipos de células possuem a capacidade de gerar uma descarga elétrica, fenômeno denominado como “potencial de ação”. Em especial, o sistema nervoso produz sinais elétricos, por meio de complexas atividades eletroquímicas celulares que percorrem todo o corpo transportando informação. A função desses sinais é, de forma abrangente, controlar as funções do corpo humano, tais como as movimentações voluntárias e involuntárias de órgãos e conjuntos musculares (Sanei & Chambers 2007).

Como os principais sinais biológicos geram sinais elétricos, pode-se obter a sua leitura por meio de eletrodos. De fato, Carlo Matteucci e Emil Du Bois-Reymond, no século XIX, foram os primeiros a registrar sinais elétricos emitidos por nervos musculares utilizando um galvanômetro. Nesse advento estabeleceu-se o conceito de neurofisiologia, desenvolvendo a formulação da bioeletricidade (Wolpaw et al. 2000, Sanei & Chambers 2007). Com o desenvolvimento de novas tecnologias, novos paradigmas foram criados para a obtenção de EEG/EMG e outros tipos de sinais biológicos, como a variação do fluxo sanguíneo e a captação de imagens internas do corpo, dentre outros. Seguem alguns exemplos utilizados atualmente:

- Magnetoencefalografia (MEG): técnica desenvolvida para a obtenção de imagens das áreas ativas do cérebro (neuroimagens). São geradas a partir da detecção dos campos magnéticos criados pelas correntes elétricas cerebrais, efeito eletromagnético natural. Os campos magnéticos, por sua vez, estão relacionados com as áreas ativas do cérebro, as quais podem, portanto, ter sua atividade monitorada (Hansen et al. 2010);
- Ressonância magnética nuclear (MRI) e funcional (fMRI): são outras técnicas para a geração de neuroimagens. Emitem fortes campos magnéticos e medem a absorção de energia dos tecidos analisados, caracterizando-se como uma espectroscopia. Analisa especialmente as mudanças de direção do fluxo sanguíneo, uma vez que este também é correlacionado com a atividade cerebral (Huettel et al. 2009);
- Espectroscopia de infravermelho próximo (NIRS): técnica de espectroscopia de absorção, baseada na emissão de luz na faixa próxima do infravermelho (de 800 *nm* a 2,5  $\mu m$ ). É utilizada para detectar mudanças nas concentrações de hemoglobina. Sua acumulação está correlacionada com as atividades cerebrais, pois é ela quem permite o transporte de oxigênio no sangue (Siesler et al. 2002).

Essas técnicas não são invasivas. A identificação de áreas ativas do cérebro durante determinadas atividades ajuda a entender seu funcionamento, bem como o estudo de doenças como epilepsia, esclerose múltipla, acidentes vasculares cerebrais e doenças de Alzheimer, Parkinson e Huntington (Sanei & Chambers 2007).

As técnicas mais utilizadas, conforme relatado anteriormente, são EEG e sEMG, que utilizam como base de funcionamento os mesmos eletrodos. São difundidas, pois medem correntes elétricas e utilizam materiais menos custosos, seu tempo de resposta é mais rápido que as outras relatadas (na ordem de milissegundos), além de serem constituídas por materiais mais leves e menores. Em outras palavras, são facilmente transportáveis por uma cadeira de rodas robótica.



As técnicas sEMG e EEG são utilizadas como foco de interação humano-máquina neste trabalho, assim, um relato mais detalhado será fornecido a seu respeito nas seções subsequentes.

## 2.1.2 Eletromiografia

A eletromiografia monitora essencialmente as atividades elétricas geradas nas membranas celulares dos nervos e músculos. A detecção é feita por meio de eletrodos, que podem ser invasivos ou superficiais (não-invasivos). Neste trabalho optou-se por eletrodos não-invasivos, uma vez que processos invasivos oferecem altos riscos e não podem ser praticados sem a presença de equipes médicas. Existem vários tipos de eletrodos não-invasivos, sendo alguns ativos e outros passivos, podem requerer soluções condutoras ou não (eletrodos molhados e secos), e mesmo sua construção pode variar para a melhoria na captação do sinal do usuário (G.tec 2014).

Dependendo da posição dos eletrodos, pode-se detectar quando um grupo muscular entra em atividade. É necessário colocar eletrodos por toda uma região de interesse, pois ao ativar-se um determinado grupo muscular, os eletrodos na vizinhança detectarão os sinais elétricos emitidos. A partir disso, é possível, por meio de técnicas de processamento de sinais, identificar qual foi o tipo de movimento executado pelo usuário, para, por exemplo, analisar o comportamento muscular em estudos de reabilitação (Sanei & Chambers 2007, Biswas et al. 2012).

Para a captação e classificação de movimentos faciais, utilizou-se o aparelho comercial Emotiv (Emotiv 2014), que pode ser observado na Figura 2.1(a), enquanto seu *kit* de desenvolvimento de software (SDK, do inglês *Software Development Kit*) é mostrado na Figura 2.1(b). Este aparelho utiliza eletrodos que requerem soluções salinas (eletrodos molhados) e a localização destes é fixa por meio de hastes plásticas, não podendo ser alteradas.

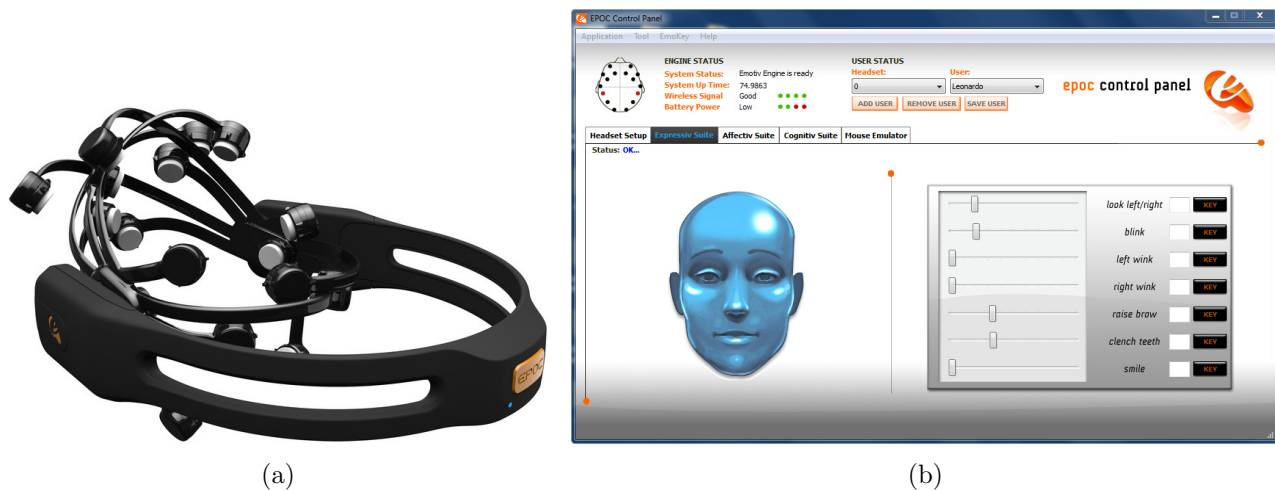


Figura 2.1: O aparelho (a) Emotiv e seu (b) SDK, para leitura de EMG facial.

Com este SDK é possível classificar vários tipos de movimentos faciais, tais como “olhar à esquerda”, “olhar à direita”, “mastigar”, “levantar as sobrancelhas” e “piscar”. O tempo de resposta é considerado rápido para essas aplicações, na ordem de milissegundos. No entanto, a classificação dos movimentos não é perfeita e a taxa de acerto varia para cada indivíduo. Mais detalhes a respeito da utilização do Emotiv serão fornecidos no Capítulo 3.

### 2.1.3 Interfaces cérebro-computador

As interfaces cérebro-computador (BCI) são interfaces humano-máquina desenvolvidas, primariamente, a partir da eletroencefalografia (EEG), sendo designadas para detectar e classificar certos padrões cerebrais do usuário. A detecção e, principalmente, a classificação de atividades cerebrais é bastante complexa, pois há uma grande gama de reações que o cérebro humano pode gerar quando estimulado.

No entanto, basicamente pode-se dividir essas reações em sinais espontâneos ou propositais. O primeiro abrange as respostas naturais do órgão a estímulos externos, como luzes piscando, e são chamados de potenciais evocados. O segundo é relativo a esforços mentais do próprio usuário como imaginação de movimentos e de música (Sanei & Chambers 2007, Graimann et al. 2010, Wolpaw & Wolpaw 2012). Conforme as tecnologias da área médica avançam, novos paradigmas vão sendo desenvolvidos. A seguir, citam-se alguns destes métodos:

- **Potencial Evocado Visual em Regime Estacionário (SSVEP, do inglês *Steady-State Visually Evoked Potentials*):** esse tipo de sinal é uma resposta natural a estímulos visuais em frequências específicas. Quando a retina é excitada por estímulos visuais (como exemplo, luzes piscando) entre 3,5 e 100  $Hz$ , o cérebro gera uma atividade elétrica na região do córtex visual (ou, lobo occipital), na mesma frequência ou seus múltiplos. Por meio dos sinais de EEG e técnicas de filtragem de sinais, é possível identificar a frequência para a qual o usuário está olhando. Se essas determinadas frequências são associadas a comandos, então pode-se executar o comando escolhido pelo usuário.
- **Potencial Evocado P300:** este sinal também é evocado e se origina da reação do usuário a um estímulo que possua um significado desejado. Esse estímulo deve ter procedência aleatória para forçar o usuário à concentração durante o processo. Quando o estímulo ocorre, o usuário responde com um esforço, usualmente uma contagem mental, todas as vezes que a opção desejada é destacada. Esse paradigma é conhecido por seu termo em inglês, *oddball*. Como abordagem mais comum, têm-se várias opções dispostas em formato matricial, as quais piscam aleatoriamente. O usuário deve escolher uma de seu interesse e contar mentalmente quantas vezes ela foi destacada. Esse sinal é de potencial positivo (P) e ocorre a aproximadamente 300 milissegundos após a reação do usuário. Na Figura 2.2(a), a qual foi obtida da referência (BCI2000 2014), tem-se a região cerebral de maior ocorrência deste sinal, que se encontra em sua maior parte no lóbulo parietal. Na Figura 2.2(b), reproduzida do Capítulo 2 de Somerset (2010), apresenta-se um exemplo do sinal P300, onde vê-se seu formato ligeiramente triangular.
- **Ritmo  $\mu$ :** este sinal corresponde a ondas que ocorrem no córtex motor, em uma faixa espectral entre 12 e 30  $Hz$ . Essas ondas são ocorrências naturais do cérebro, quando o usuário está em repouso. É possível suprimi-las, fato denominado por “dessincronização da onda”, na ocorrência de alguma ação motora, ou então, com prática, por imaginação de movimentos. Esta supressão pode ser detectada e empregada em sistemas BCI, especialmente as que envolvem decisões binárias.

- Imaginação de movimentos: são atividades mentais provocadas ativamente pelo usuário, como por exemplo, imaginar que está movendo o braço esquerdo, a perna direita, ou a língua. Além da supressão do ritmo  $\mu$ , essas atividades mentais podem criar padrões classificáveis, pois ativam diferentes partes do cérebro. Assim, pode-se associá-los a diversos contextos, como excussões de comandos, seleções de opções, e assim por diante.

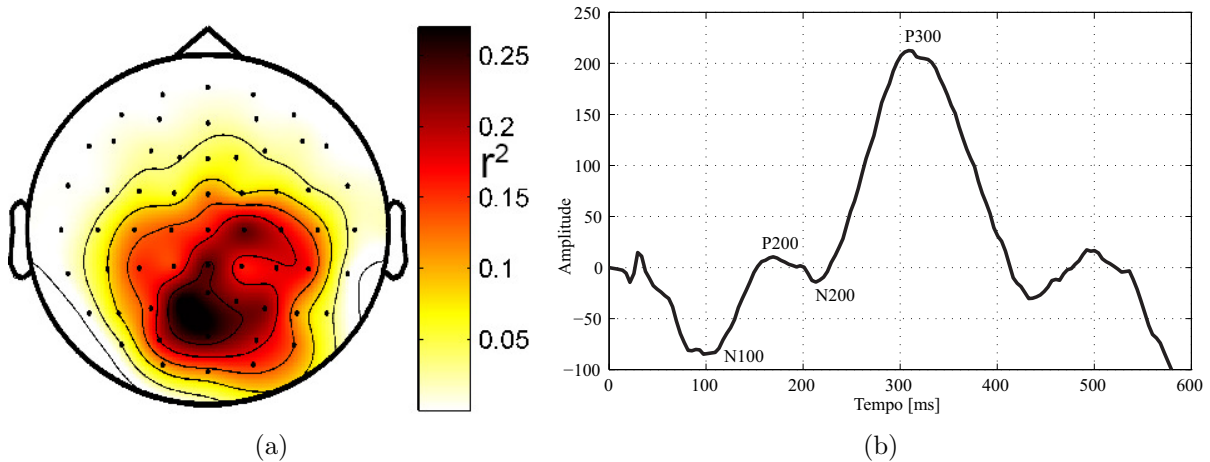


Figura 2.2: (a) Região cerebral de ocorrência e (b) exemplo de sinal P300.

Existem outros exemplos, como a imaginação de música, outros tipos de ritmos cerebrais, tais como os ritmos  $\alpha$  e  $\beta$ , e potenciais evocados negativos e positivos como o N100, N200, P200, os quais podem ser vistos na Figura 2.2(b). Os trabalhos na literatura são numerosos com diversas aplicações (Friedman et al. 2001, Serby et al. 2005, Graimann et al. 2010, Wolpaw & Wolpaw 2012, Postelnicu & Talaba 2013, Soriano et al. 2013).

Existem várias abordagens de BCI envolvendo a seleção de opções pelo usuário. Quando a situação oferece poucas opções (tipicamente três ou quatro), métodos mais diretos de obtenção de padrões, como a imaginação de movimentos, podem ser utilizados. No entanto, se o cenário propõe várias opções, como ocorre em alfabetos, então é necessário utilizar técnicas como o P300, que propiciam uma escolha dentre muitas disponíveis. No entanto, o tempo de execução destes algoritmos de várias opções é muito superior aos de poucas opções, e por isso, algumas estratégias devem ser ponderadas nesses casos (Sanei & Chambers 2007, Bi et al. 2013).

Para a captação de sinais cerebrais via EEG, os eletrodos devem ser colocados em locais estratégicos, o que decorre das diferentes regiões cerebrais onde se localizam as ocorrências das técnicas utilizadas. Por exemplo, o SSVEP ocorre no lobo occipital (córtex visual), já o P300 ocorre no lobo parietal, e o ritmo  $\mu$  ocorre no lobo central (córtex motor). É necessário, portanto, identificar as regiões cerebrais de maneira sistemática.

O Sistema Internacional 10-20 é uma padronização para nomeação e posicionamento de eletrodos em atividades envolvendo EEG, e pode ser observado na Figura 2.3(a). Para a obtenção de EEG, este trabalho de pesquisa utilizou o aparelho comercial g.USBamp fornecido pela empresa g.tec (G.tec 2014), e uma touca disposta de dezesseis eletrodos secos, que podem ser observados na Figura 2.3(b).

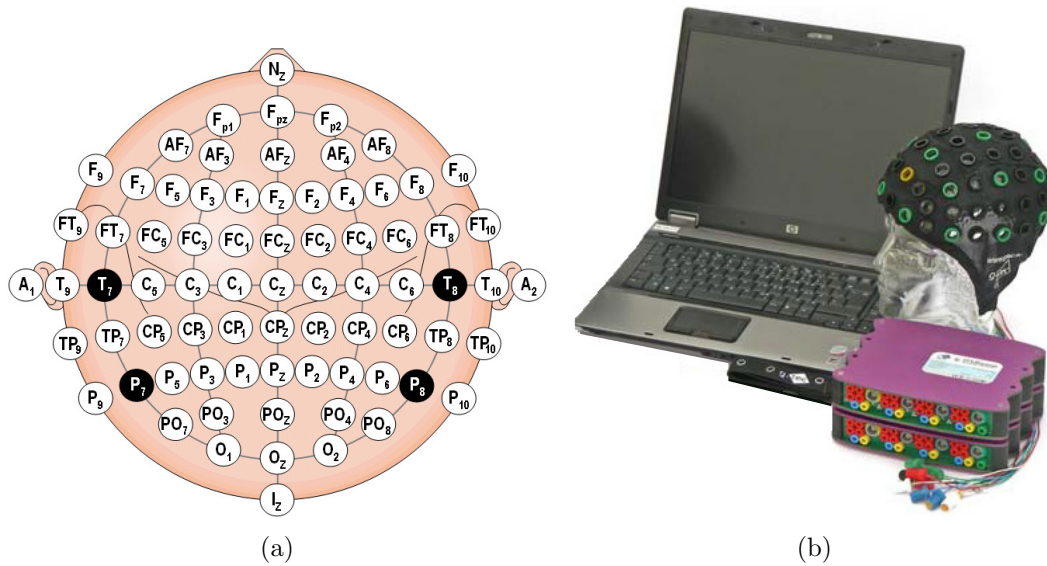


Figura 2.3: (a) Sistema Internacional 10-20 e (b) aparelho g.tec.

Neste trabalho, para a classificação de padrões, escolheu-se a técnica P300. Conforme discutido, trata-se de uma técnica reativa e que praticamente dispensa longos treinamentos e atividades mentais complexas. Com isso, utilizou-se o software BCI2000 (Schalk et al. 2004, BCI2000 2014), que é uma plataforma aberta para pesquisas em BCI disponível na *internet* e bastante difundida no meio acadêmico. Para a interação do usuário com o sistema, projetou-se uma interface para escolha de opções que engloba o paradigma P300. Os detalhes acerca da utilização dessas técnicas e aparelhos será abordada com maior profundidade no Capítulo 3.

Um sistema de BCI pode ser resumido na forma mostrada pela Figura 2.4. Há um estímulo externo, seja por uma tela de computador ou mesmo pelo contexto em que o usuário está inserido, que provoca uma resposta cerebral (proposital ou espontânea). Esses dados são obtidos por eletrodos e enviados a um sistema de filtragem e classificação. Desta forma, pode-se selecionar uma opção desejada, a qual está associada a um determinado comando. Para esta seleção, o sistema pode utilizar várias técnicas de filtragem e aprendizado de máquina. Finalmente, envia-se o resultado ao sistema final, como por exemplo, uma letra do alfabeto impressa, ou um local a ser alcançado por um robô.



Figura 2.4: Etapas de um sistema de BCI (exemplificado com um aparelho g.tec).

## 2.2 Robótica móvel assistiva

A robótica móvel está cada vez mais presente em todos os níveis da sociedade. Robôs têm sido empregados tanto em processos industriais quanto em ambientes hospitalares e domésticos. Há um grande campo a ser explorado em tecnologias assistivas devido à grande quantidade de pessoas com deficiência, conforme relatado anteriormente.

No caso das cadeiras de rodas robóticas, o grupo de pessoas a tirar proveito dessa tecnologia incluem quadriplégicos, lesionados da coluna vertebral, com doenças neuromusculares, paralisias cerebrais, com deficiências cognitivas, pessoas afetadas por acidentes vasculares cerebrais, com doenças congênitas, dentre outras (Wolpaw et al. 2000, Katevas 2001, Sanei & Chambers 2007, Millán et al. 2010, Cowan et al. 2012). A Figura 2.5 é uma representação de uma cadeira de rodas robótica que, em comparação com as cadeiras motorizadas, possui diversos outros componentes de processamento, sensoriamento e interface de comunicação.

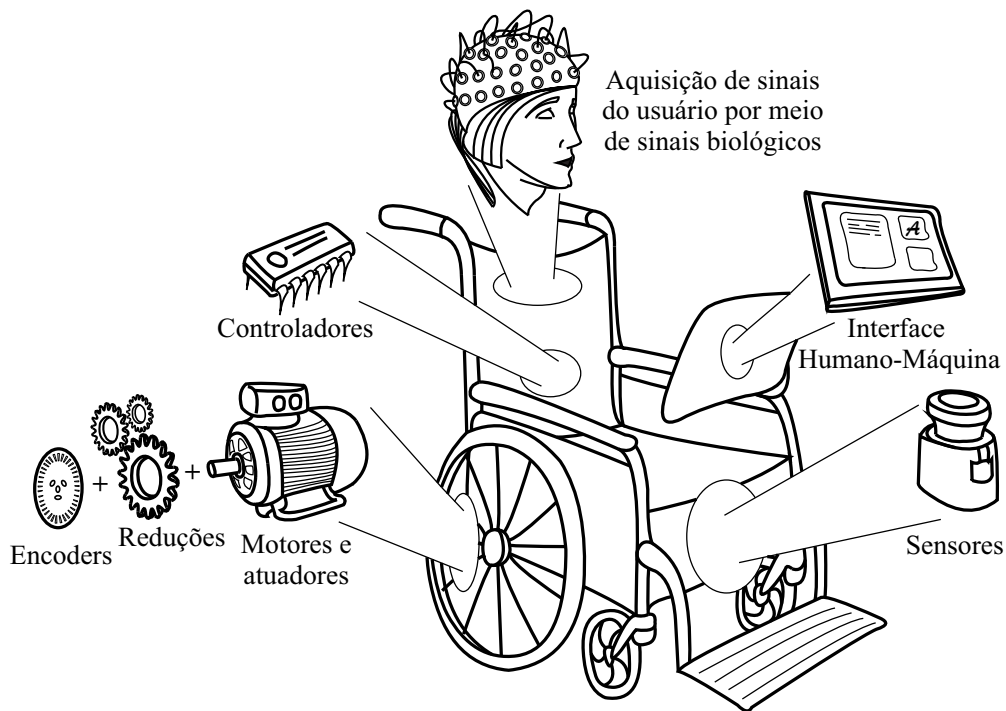


Figura 2.5: Cadeira de rodas robótica, com diversos componentes.

Uma grande gama de cadeiras de rodas robóticas vem sendo desenvolvida por diversos grupos de pesquisas para atender a esse público alvo. Suas funções variam, desde simples controles de velocidade a sofisticados comportamentos autônomos. Dentre os vários trabalhos reportados na literatura, podemos citar o trabalho pioneiro de Yanco (2001) que emprega sEMG.

Como exemplos de trabalhos que utilizam HMIs baseadas em EEG, há o emprego de SSVEP nos trabalhos de Luth et al. (2007) e Mandel et al. (2009), de P300 em Rebsamen et al. (2007), de imaginação de movimentos em Tanaka et al. (2005), Choi & Cichocki (2008) e Benevides et al. (2012), dessincronização do ritmo  $\alpha$  em Ferreira et al. (2007), e SSVEP em Müller et al. (2013). Outros trabalhos relacionados à proposta desenvolvida nesta tese serão examinados mais detalhadamente ao longo do texto.

Uma vez que se dispõe de uma interface de comunicação do usuário em um ambiente assistivo, é necessário um sistema de controle que interprete corretamente as requisições recebidas, para que o dispositivo robótico possa executá-las. O sistema de controle deve prezar não somente pela execução correta do comando, mas também pela integridade física do usuário.

Assim como nas HMIs assistivas, o principal ponto de partida ao se desenvolver soluções de robótica assistiva é a entrevista as pessoas com deficiências. As técnicas desenvolvidas, por mais sofisticadas que sejam, devem primariamente atender às necessidades diretas do público alvo. Há vários trabalhos publicado na literatura, relatando as opiniões e dificuldades dos usuários, assim como as abordagens empregadas (Crisman & Cleary 1998, Bourhis et al. 2001, Cruz et al. 2011, Freed et al. 2011, Benevides et al. 2012, Bi et al. 2013, Celeste et al. 2013, Bastos-Filho et al. 2014).

Um dos pontos principais da área envolve o conceito de participação do usuário durante a navegação de uma cadeira de rodas robótica, que pode ocorrer de várias formas, com diferentes graus de autonomia do robô. Essa concepção ficou conhecida na literatura como controle compartilhado, termo proveniente do inglês *shared control* (Millán et al. 2010, Cowan et al. 2012).

Um sistema de controle compartilhado tradicional é definido como uma tecnologia assistiva destinada ao usuário que navega um robô móvel. Esta tecnologia emprega, geralmente, uma grande quantidade de algoritmos, modos de operação, diferentes níveis de assistência ao usuário, dentre vários outros aspectos. Nessa área, cada autor propõe sua própria solução, a qual é geralmente voltada a um problema específico, tal como desvio de obstáculos, navegação por passagens estreitas, diminuição dos erros provenientes dos sistemas de classificação empregados nas HMIs (EMG, BCI, etc.), e assim por diante.

No tocante à navegação da cadeira de rodas, as várias abordagens podem ser agrupadas em duas categorias principais, controles autônomo e compartilhado. Evidentemente, é possível fazer um híbrido semiautônomo, no entanto, uma abordagem híbrida pode ser inclusa na categoria autônoma sem perda de generalidade. Uma caracterização geral de ambas as categorias é apresentada a seguir.

- **Controles autônomo e semiautônomo:** nessa abordagem, o usuário, por meio de uma HMI assistiva, escolhe um local disponível (mapeado) para o qual deseja ir. A partir desta requisição, o sistema robótico calculará todos os passos necessários para alcançar o ponto-objetivo autonomamente, isto é, durante a navegação não haverá necessidade de interação do usuário com a máquina, que ficará livre para outras atividades. Caso as distâncias navegadas autonomamente sejam curtas, de poucos metros, exigindo diversas interações do usuário para a continuação da navegação, pode-se classificar a abordagem como semi-autônoma. Já no caso de autonomia por distâncias grandes, onde o robô atravessa diversos ambientes, classifica-se como uma abordagem plenamente autônoma.

Justificativa: a comunicação assistiva utiliza esforços musculares (EMG) ou mentais (BCI). Este tipo de atividade, quando feita por tempos prolongados, pode levar o usuário à exaustão. Por esta razão, deseja-se minimizar as interações do usuário com a interface. Sendo o controle autônomo, requer-se uma menor quantidade de decisões para que uma navegação de longo termo seja planejada. Conseqüentemente, sua fadiga será menor (Crisman & Cleary 1998, Bourhis et al. 2001, Millán et al. 2010, Olivi et al. 2012).

Dificuldades: em robótica móvel, processos autônomos requerem uma integração de técnicas, como mapeamento prévio do ambiente, controladores de movimentos, sistemas de localização, uma eventual estruturação dos locais permitidos, dentre outros (Thrun et al. 2005, Morales et al. 2013).

- **Controle compartilhado:** nessa abordagem o usuário possui um grande poder de decisão no caminho do robô durante a navegação. Por meio da interface assistiva, o usuário dirige o robô diretamente, isto é, escolhendo comandos que direcionam o robô, tais como “virar à esquerda”, “virar à direita”, “acelerar” e “parar”. Dessa forma, a navegação do robô é totalmente afetada pelo julgamento do usuário, podendo inclusive alcançar locais que não estão disponíveis em abordagens autônomas, por exemplo, um local não mapeado. Juntamente com o usuário, há um sistema supervisor atuando em paralelo, corrigindo possíveis falhas das HMIs assistivas e ajudando o usuário a alcançar seus objetivos, denotando uma navegação assistida.

Justificativa: apesar de a interação com a interface assistiva ser cansativa, em trabalhos citados na literatura os usuários expressam o desejo de participar ativamente do processo de controle, pois, as atividades diárias, tais como se deslocar por ambientes, são bastante valorizadas pelas pessoas com restrições de locomoção. Além disso, tem-se que o esforço mental beneficia o desenvolvimento cognitivo dessas pessoas que, em alguns casos, estão se recuperando de traumas ou enfermidades (Müller-Putz et al. 2006, Dobrea & Dobrea 2009, Tonin et al. 2010).

Dificuldades: aparelhos de EMG/BCI possuem altos custos e a contaminação por ruídos tem uma influência grande na classificação de comandos, e geralmente há poucos canais disponíveis o que prejudica a quantidade de comandos disponíveis ao usuário (Leeb et al. 2007, Sanei & Chambers 2007).

As pesquisas em robótica assistiva, especialmente no âmbito de cadeiras de rodas robóticas, tiveram crescimento a partir da década de 1990. Em decorrência da tecnologia computacional, rapidez de comunicação, processamento de algoritmos, dentre outros aspectos técnicos, as abordagens mais comuns eram as autônomas, por questões de flexibilidade de implementações, disponibilidade de equipamentos, e assim por diante. Nesse contexto autônomo, podemos citar alguns trabalhos da época, tais como Nisbet et al. (1996) e Simpson & Levine (1997).

A partir do final da década de 1990, acompanhando o desenvolvimento tecnológico da computação, com o ganho de memória, velocidade e capacidade de armazenamento, as técnicas semiautônomas e manuais (compartilhadas) puderam florescer. O desenvolvimento das tecnologias médicas também favoreceram as cadeiras de rodas robóticas, com melhorias na captação de sinais de sEMG e EEG, com melhores filtros e maior velocidade de aquisição do sinal, com materiais mais leves, com menor consumo de energia e com maior precisão e acurácia dos equipamentos.

É verdade que os preços desses materiais, ainda nos dias atuais, não acompanharam o desenvolvimento citado, possuindo custos elevados. Porém, há um recente direcionamento dos fabricantes no sentido de aumentar o acesso a seus produtos, a fim de que os produtos voltados às tecnologias assistivas possam alcançar o grande público nos próximos anos.



## 2.3 Controle compartilhado

Considerando os primeiros anos da área de controle compartilhado, tem-se o trabalho de Levine et al. (1999), com sua cadeira robótica denominada *NavChair*, exibida na Figura 2.6(a). Mesmo sendo um trabalho antigo, já emprega o conceito de modos de operação, sendo que, destes, dispõe de três. O primeiro trata-se de um desvio de obstáculos por meio de Histograma de Campos Vetoriais (VFH, do inglês *Vector Field Histogram*). O segundo dedica-se ao problema de passagens de portas, ou seja, passagens estreitas. Nesta atividade, foi necessário modificar o algoritmo VFH para atender especialmente a esta modalidade. O modo de controle final se destina a seguir paredes, como maneira de deslocar-se por ambientes. Para todos os modos, o autor utiliza medidas de distâncias provenientes de sonares. Para alternar entre seus modos de controle compartilhado, combina técnicas probabilísticas, sendo que o controle é retirado do usuário para executar autonomamente as ações. Segundo os autores, o trabalho obteve desempenho adequado, resultando em sucesso em situações difíceis e sem qualquer colisão.

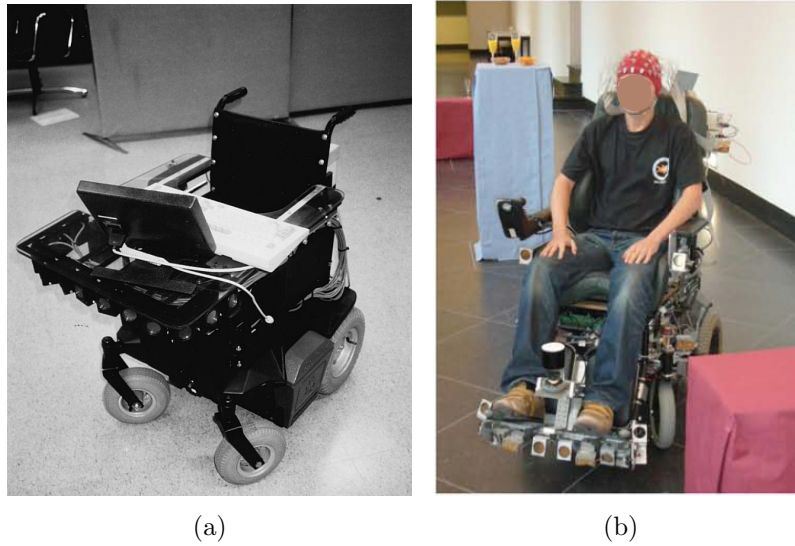


Figura 2.6: Cadeiras robóticas de (a) Levine et al. (1999) e (b) Phillips et al. (2007).

Houve outros trabalhos no início dos anos 2000, muito embora, uma real preocupação com a área deu-se a partir da segunda metade da década passada, como o trabalho de Philips et al. (2007), exibido na Figura 2.6(b). Esse trabalho empregou um nível de assistência constante, ou seja, o usuário sempre seria beneficiado por uma ação de controle compartilhado caso necessário. Ações essas que, assim como no trabalho anterior, são três. Nelas empregou a política do vencedor-leva-tudo (do inglês, *winner-takes-all policy*) após um cálculo estatístico de qual comportamento seria o mais apropriado para a situação atual. As informações do ambiente foram obtidas por meio de *lasers*. O primeiro comportamento evitava colisões, agindo como um “botão de emergência”, parando a cadeira quando muito próxima de obstáculos. O segundo, uma variação do primeiro, que desviava a cadeira dos obstáculos de um certo ângulo. A terceira ação envolvia o conceito de cumprimento de objetivos: tratava-se de uma recuperação angular apropriada quando o usuário encontrava-se muito desviado de seu ponto-objetivo. Segundo os autores, a abordagem ofereceu bons resultados, tanto qualitativa quanto quantitativamente.



Com uma abordagem em inteligência computacional, Trieu et al. (2008) desenvolveu um tipo de controle compartilhado voltado para o desvio de obstáculos com Redes Neurais Artificiais (RNAs) Bayesianas. As informações do sistema robótico são obtidas por *lasers*, e esta abordagem também utiliza a assistência constante. A técnica desenvolvida é capaz de detectar e passar por portas estreitas e corredores. O autor também se preocupou em obter mapas do ambiente. O usuário dispõe de quatro comandos básicos (“esquerda”, “direita”, “para frente” e “para trás”). Ao escolher um comando, sua intenção é combinada com o mapa do local, assim, obtém-se uma região segura na direção apontada pelo usuário, para a qual o robô desloca-se autonomamente. Por conta do deslocamento autônomo, esta abordagem se caracteriza como semiautônoma, pois são muitas as interações com o usuário para navegar por pequenas distâncias. Sua cadeira robótica é exibida na Figura 2.7(a).

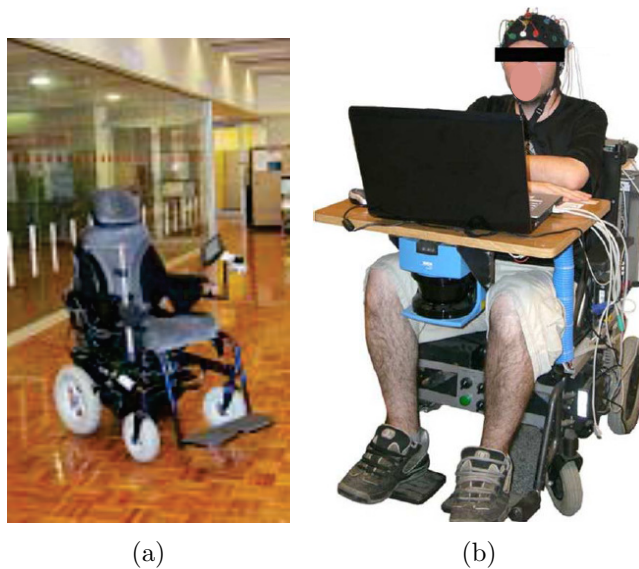


Figura 2.7: Cadeiras robóticas de (a) Trieu et al. (2008) e (b) Iturrate et al. (2009).

A abordagem de Iturrate et al. (2009), cuja cadeira robótica é exibida na Figura 2.7(b), utilizou numa interface baseada em P300. A tela do usuário mostra uma visão do ambiente com vários pontos próximos marcados como alcançáveis, obtidos via *laser*, disponíveis para escolha via HMI assistiva. Após o objetivo escolhido, um comportamento autônomo desloca o robô até a marca e o procedimento de escolha se repete. Trata-se de uma abordagem do tipo “aponte-para-ir” (tradução livre do inglês *point-to-go*). Da mesma forma que o trabalho anterior, é uma abordagem semiautônoma, dada a grande quantidade de interações com o usuário para curtas distâncias de navegação. Durante a navegação, os autores se preocupam com a segurança do usuário por meio de detecção e desvio de obstáculos, utilizando o planejador de rotas D\*.

Uma abordagem de navegação compartilhada é reportada pelos autores Tonin et al. (2010). Esta implementação foca na detecção e desvio de obstáculos. Sempre que o robô está em face a um obstáculo e não possui capacidade de decidir autonomamente para qual direção deve prosseguir, mantém-se a navegação sob responsabilidade do usuário. Quando este escolhe uma direção (“esquerda” ou “direita”) para o desvio, o robô gira a passos fixos de 30° a cada decisão tomada. As informações do ambiente chegam ao robô por meio de sensores infravermelhos.

Há, ainda, um comportamento tipo “botão de emergência”, que pára o robô quando este perde a comunicação com o sistema de BCI. O trabalho menciona bons resultados, com navegação por ambientes diários de um usuário, como corredores e passagens de portas, perfazendo-os sem colisões. O robô utilizado nesta pesquisa está exibido na Figura 2.8(a).

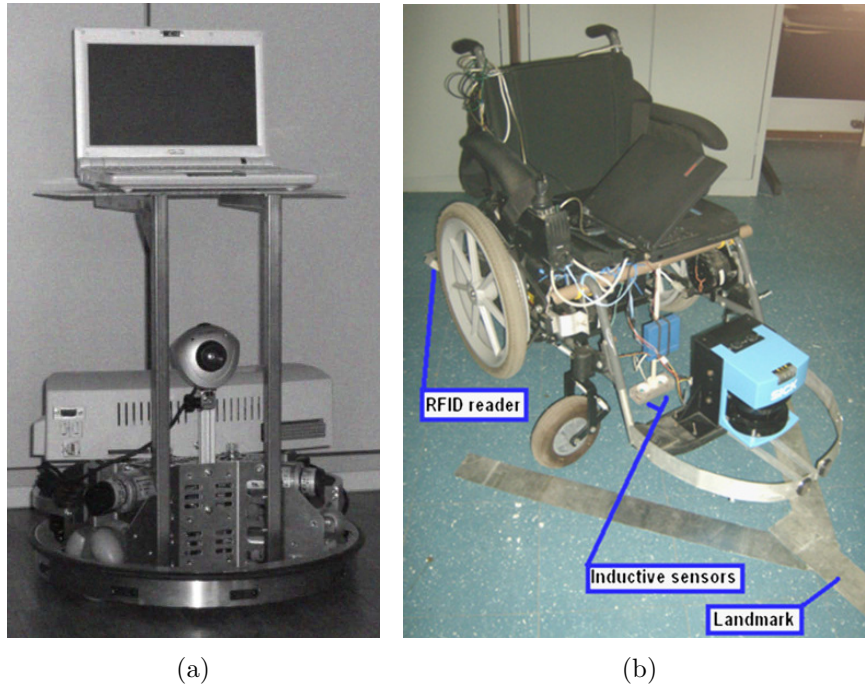


Figura 2.8: Robôs utilizados por (a) Tonin et al. (2010) e (b) Cruz et al. (2011).

A partir de 2010, abordagens mais sofisticadas passaram a ser desenvolvidas, uma vez que os problemas de classificação das abordagens de EMG/BCI persistem, além da dependência da aptidão do usuário para utilização dos dispositivos. Em 2011, uma abordagem totalmente autônoma foi reportada em Cruz et al. (2011). Nessa abordagem, os autores desenvolveram e integraram diversas técnicas para que o usuário, ao escolher uma posição de destino, fosse levado de maneira totalmente independente até o local desejado. O robô possui duas maneiras de navegar no ambiente. Quando em passagens apertadas, como portas, há um trilho metálico no chão, conforme pode-se visualizar na Figura 2.8(b), em que o robô assume o comportamento de seguir o trilho com a ajuda de sensores indutivos. No entanto, em passagens livres, o robô possui um caminho calculado por meio do algoritmo de Dijkstra, que leva o robô a cruzar ambientes em busca do ponto-objetivo. Para a localização do robô, a qual é fundamental para navegações autônomas, os autores utilizam um SLAM por Filtro de Kalman Estendido (EKF-SLAM), o qual é auxiliado por identificação de etiquetas por rádio-frequência (*tags RFID*) colocadas estrategicamente no ambiente, junto aos trilhos magnéticos. Trata-se, portanto, de uma abordagem que depende da instrumentação do ambiente. Mais recentemente, os autores estenderam o trabalho incorporando navegação semiautônoma Bastos-Filho et al. (2014). Na abordagem semiautônoma, um sistema supervisor oferece ao usuário quatro comportamentos autônomos auxiliares para o deslocamento do robô pelo ambiente: auxílio para cruzar passagens estreitas, auxílio a manobras em espaços reduzidos, paradas de emergência e controle de velocidade linear.

Também em 2011, Li et al. (2011), Figura 2.9(a), propõem uma função de custo com multiobjetivos. A função de custo possui três índices (“segurança”, “conforto” e “obediência”) definidos pelos autores. Estes índices avaliam a performance da cadeira robótica, que, então, calcula o peso de participação do usuário nas tarefas. A cadeira possui defesas reativas a obstáculos, baseadas em VFH e Campos de Forças Vetoriais (VFF, do inglês *Vector Force Field*). O artigo apresenta resultados coerentes em passagens de portas e por corredores estreitos, perfazendo-os sem colisões, sempre tendo em vista a segurança do usuário.

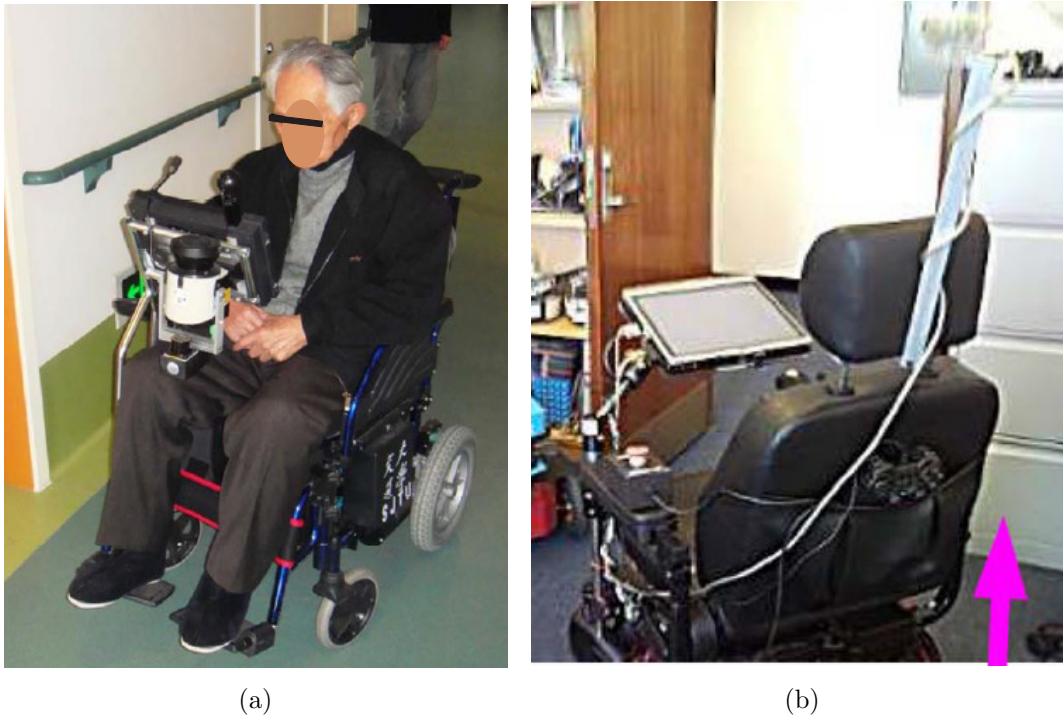


Figura 2.9: Cadeiras robóticas de (a) Li et al. (2011) e (b) Carlson et al (2012).

Seguindo na linha de cálculo de probabilidades, cita-se o trabalho de Carlson & Demiris (2012), Figura 2.9(b). Os autores desenvolveram um sistema de múltiplas hipóteses para a predição das intenções do usuário, ou seja, as escolhas mais prováveis dado um determinado ambiente, para, quando necessário, fazer ajustes nos sinais de controle para que os objetivos escolhidos sejam alcançados de maneira segura. Intenções, de acordo com o trabalho, são decisões de atravessar passagens e corredores. Quando há mais de um tipo de intenção, uma política de vencedor-leva-tudo decide qual ação será tomada. Quando uma intenção é detectada pelo preditor, traça-se uma trajetória curta utilizando VFH+ (versão aprimorada de VFH), caracterizando-se como uma navegação semiautônoma. Os resultados do trabalho relatam um longo circuito percorrido com sucesso pelo usuário.

Há vários outros tipos de implementações que podem ser encontrados na literatura, como por exemplo Derry & Argall (2013) que utilizaram nuvens de pontos em 3D juntamente com visão computacional a fim de elaborar comportamentos no robô para a detecção de portas, passagens estreitas, dentre outras situações difíceis que podem se apresentar ao longo do ambiente, de forma a ajudar o usuário em sua navegação.

Considerando os trabalhos relatados anteriormente, embora todos estes métodos sejam bastante diferentes, possuem, em essência, um ponto em comum. Em algum momento da navegação o controle retira o comando do usuário para que o robô execute alguma ação autônoma, usualmente desvio de obstáculos. Embora o desvio de obstáculos seja indispensável, o fato de retirar o usuário do laço de controle, por conta de ações autônomas, pode levar a situações e configurações desfavoráveis, levando-o à frustração.

De fato, essa linha de implementações que optam por diversos modos de funcionamento (por exemplo, um modo para desvio de obstáculos, outro modo para passagens estreitas etc.), com o chaveamento entre eles é tão difundida na literatura que, em 2010, Millán et al. (2010) classificou as abordagens de controle compartilhado como sendo pertencentes a apenas duas categorias: esses modos seriam chaveados entre si de maneira autônoma (quando condições fossem satisfeitas por meio de dados coletados), ou, pelo próprio usuário de maneira manual.

Vários dos trabalhos descritos concordam com a definição de controle compartilhado dada por Millán et al. (2010). Os autores Cowan et al. (2012) contestam esta definição, mostrando que há várias outras maneiras de controle compartilhado além dessas duas relatadas. O trabalho desenvolvido nesta tese concorda com Cowan et al. (2012), cujas justificativas serão apresentadas na próxima seção.

## 2.4 Integração das navegações autônoma e compartilhada

Conforme discutido na seção anterior, há, essencialmente, duas abordagens principais para a navegação de robôs assistivos reportadas na literatura, a navegação autônoma e a navegação compartilhada, ambas com benefícios e reveses. Propõe-se, portanto, uma abordagem que integre ambas as concepções de forma a trabalharem em conjunto. Desta maneira, minimiza-se as desvantagens individuais de cada abordagem, discutidas na Seção 2.2, página 13. O usuário poderá valer-se de ambas, e assim o sistema poderá ajudá-lo da maneira mais apropriada para cada situação que se apresenta.

Para a abordagem manual foi desenvolvido um sistema de controle compartilhado baseado na utilização de campos vetoriais. No final da seção anterior, argumentou-se que este trabalho concorda com a contestação de Cowan et al. (2012) ao argumento de Millán et al. (2010) sobre os modos de realizar o controle compartilhado. Isso se dá visto que o presente trabalho não utiliza quaisquer tipos de chaveamentos entre modos de operação.

Essa escolha se deu porque sistemas chaveados (*switched systems*) são um problema fundamental e não totalmente resolvido da área de controle. Mesmo que este chaveamento se dê entre dois ou mais sistemas estáveis, não se pode garantir a estabilidade do sistema como um todo sem uma análise apropriada, a qual geralmente é bastante complexa (Liberzon 2003). Nenhum dos trabalhos pesquisados na literatura apresentou essa análise de estabilidade, mostrando que o assunto é totalmente negligenciado. A partir destas considerações, conclui-se que optar por chaveamentos entre modos de operação sem a devida análise de estabilidade do sistema pode expor o usuário a situações de instabilidades e, conseqüentemente, perda de controle do sistema. Por todas essas questões apresentadas, neste trabalho esse tipo de abordagem foi evitada, não havendo, portanto, quaisquer tipos de chaveamentos entre modos de operação.

Além disso, modos de operação e ações autônomas são, em última análise, táticas para vencer situações específicas do ambiente. Por exemplo, o modo de atravessar portas serve unicamente a este propósito. Dessa forma, particulariza-se uma quantidade limitada de situações possíveis, geralmente as mais comuns, para que essas sejam chaveadas entre si conforme se apresentam no ambiente. No entanto, eventualidades não previstas pelo projetista podem inutilizar todos os modos de operação disponíveis, por mais numerosos que sejam. Em outras palavras, sempre existirá uma situação nova para o sistema, pois as configurações possíveis para um ambiente diário são praticamente ilimitadas.

Adicionalmente, não se garante que um modo de operação contemplará todas as situações para as quais se propõe. Por exemplo, uma ação de cruzar portas pode não englobar todos os tipos de portas existentes, fazendo com que o sistema entre em modo de falha mesmo numa situação para a qual ele possui um modo, novamente colocando o usuário em uma situação insegura.

Além de levar em consideração os trabalhos discutidos na seção anterior, escolheu-se para o método proposto total independência do ambiente. Dessa forma, qualquer avaliação da performance do usuário para o cálculo da intervenção do controle compartilhado, como em Philips et al. (2007) ou Li et al. (2011), não podem ser levadas em consideração. Uma análise desta natureza demanda o cumprimento de, ao menos, um objetivo, o que requer conhecimento prévio do ambiente. Além disso, esta abordagem não utiliza modos de operação uma vez que tais modos nunca são suficientes para satisfazer todos os possíveis objetivos do usuário, tampouco instrumentações do ambiente. Ao contrário, considera-se que o usuário deve ter a liberdade para navegar para onde melhor lhe aprouver.

Similarmente, nesta proposta de pesquisa não se executam quaisquer predições de intenções, como as feitas em Carlson & Demiris (2012), uma vez que isto pressupõe uma particularização (modelagem) de cada usuário, e nesse trabalho de pesquisa optou-se por não levar em conta especificidades dos usuários, pois, particularizações podem limitar o alcance do sistema.

Finalmente, a proposta jamais retira o controle do usuário, dada a inexistência de ações autônomas. Esta proposição foi considerada após a análise e implementação de alguns dos trabalhos desenvolvidos na literatura. Notou-se que, em diversas situações, após o controle ser retirado do usuário para efetuação de uma ação autônoma, o robô terminava o movimento em uma configuração desfavorável para o usuário, o que gerava frustração. Dessa forma, para evitar esse tipo de situação para o usuário, eliminou-se as ações autônomas e os modos de operação.

Como consequência dessas escolhas, obtém-se total liberdade de deslocamento para o usuário (Olivi et al. 2012, Olivi et al. 2013, Olivi et al. 2014). Dentre os principais efeitos positivos dessa abordagem, destacam-se a redução dos efeitos negativos causados por erros advindos do classificador da HMI assistiva, e, a promoção da navegação segura, privilegiando zonas seguras com maior velocidade e flexibilidade, sem contudo prejudicar a acessibilidade do usuário a todos os locais navegáveis do ambiente. Além disso, menciona-se o fato de que a estratégia proposta de controle compartilhado, se munida do ferramental necessário, pode produzir mapas dos ambientes explorados, o que possibilita a navegação autônoma sem o conhecimento prévio do ambiente. O ponto negativo decorre do fato que a navegação compartilhada exige esforço extra do usuário, o que pode ser cansativo por longos períodos de tempo.

Mesmo esse ponto negativo pode ser relativizado, pois, em casos de BCI, o esforço mental pode ajudar no desenvolvimento cognitivo do usuário que estiver em fase de recuperação (Müller-Putz et al. 2006, Dobrea & Dobrea 2009, Tonin et al. 2010), tornando o ponto negativo, neste caso particular, positivo. No entanto, no caso geral, o cansaço do usuário deve ser levado em consideração, razão pela qual desenvolveu-se a estratégia de controle autônomo.

Em relação ao planejamento do controle autônomo desta tese, seguiu-se a linha já consolidada na literatura. Ao usuário é oferecido um menu, controlado por sEMG ou EEG, a partir do qual ele pode escolher um destino num mapa conhecido. Tendo um objetivo, o qual é a única ação necessária do usuário, o robô o alcança autonomamente sem demais requisições. Destaca-se a colaboração da navegação por controle compartilhado, que possibilita obter um mapa do ambiente e, com o mesmo tipo de controlador empregado na navegação compartilhada, navegar o robô de forma autônoma. O usuário pode utilizar este modo de navegação sempre que estiver fatigado ou diante de trajetos rotineiros. Os pontos negativos dizem respeito a necessidade de mapas do ambiente e no vasto conjunto de técnicas robóticas requeridas, tais como mapeamento, localização, planejamento de caminhos e navegação, conforme relatadas em Olivi et al. (2012).

As estratégias de navegação assistida por controle compartilhado e de navegação autônoma serão detalhadas nos capítulos seguintes. Deseja-se avaliar tanto sua eficácia quanto sua contribuição agregada ao contexto das tecnologias assistivas em geral, validando seu desenvolvimento. No entanto, antes de detalhá-las, é necessário obter uma via de comunicação entre o usuário e a máquina. Comumente designadas por Interfaces Humano-Máquina (HMIs), esses canais de comunicação colhem dados biológicos dos usuários e obtêm os comandos para o sistema robótico. Nesta pesquisa utilizou-se uma HMI assistiva de comunicação, a qual será objeto de estudo no próximo capítulo.



## Comunicação humano-máquina assistiva

Este capítulo apresenta o desenvolvimento das interfaces de comunicação entre o usuário e o sistema robótico assistivo. A HMI é essencial para desenvolver a proposta de controle do sistema robótico mencionado na Seção 2.4, página 20. Primeiramente, é necessário estabelecer a comunicação entre o usuário e a máquina. Nos capítulos precedentes, explicitou-se que as pessoas para as quais essas tecnologias de acessibilidade são desenvolvidas sofrem de severos impedimentos físicos. Como estão incapacitadas de utilizar comunicação tradicional, são necessárias interfaces de comunicação baseadas em sinais biológicos. As técnicas mais utilizadas na literatura, e também neste trabalho de pesquisa, são sEMG e BCI. Para o desenvolvimento, utiliza-se, respectivamente, os equipamentos comerciais Emotiv (Emotiv 2014) e g.tec USBamp (G.tec 2014).

Conforme discutido na Seção 2.2, página 13, o custo de equipamentos à base de sEMG e EEG ainda são altos. No entanto, quando comparados com outros equipamentos como o NIRS, seus custos são relativamente acessíveis. Além disso, são passíveis de serem embarcados em veículos assistivos, dado o seu reduzido volume e peso. Outra vantagem é a possibilidade de utilizar eletrodos secos não-invasivos, reduzindo a manutenção do equipamento e o tempo de preparação (*set-up*) do usuário. Ressalta-se ainda que o tempo de resposta deste tipo de equipamento é bastante reduzido quando comparado com os demais (Bi et al. 2013).

O equipamento Emotiv, utilizado para captação e classificação de sEMG, conforme discutido na Seção 2.1.2, página 9, utiliza eletrodos que necessitam de líquidos condutores, também chamados de eletrodos molhados (do inglês, *wet electrodes*). No entanto, seu tempo de colocação no usuário é relativamente rápido, uma vez que seus eletrodos possuem posições fixadas por meio de hastes plásticas, não oferecendo dificuldades de colocação. Utilizou-se o próprio SDK do aparelho para classificação de movimentos faciais (vide Figura 2.1, página 9). Seu tempo de classificação é dos mais rápidos, na ordem de milissegundos (Emotiv 2014).

Com o aparelho da g.tec, utilizado para BCI e mostrado na Seção 2.1.3, página 10, foram utilizados eletrodos secos (do inglês, *dry electrodes*), juntamente com o módulo de amplificação g.USBamp. Os eletrodos secos deste aparelho oferecem boas vantagens de colocação e flexibilidade, uma vez que sessões de EEG com eletrodos molhados necessitam de mais tempo para a preparação do usuário. Juntamente com o equipamento g.tec, utilizou-se sua própria Interface de Programação de Aplicativos (API, do inglês *Application Programming Interface*) para MATLAB (MathWorks 2014) e também o *framework* BCI2000 (BCI2000 2014).

Foram consideradas algumas abordagens de HMIs reportadas na literatura antes de planejar a que foi desenvolvida nesta pesquisa. Existem várias estratégias, com diferentes tipos de coleta e processamento de sinais biológicos (Bourhis et al. 2001, Müller-Putz et al. 2006, Ramsey et al. 2006, Leeb et al. 2007, Vanacker et al. 2007, Philips et al. 2007, Trieu et al. 2008, Iturrate et al. 2009, Geng et al. 2010, Morales et al. 2013, Park et al. 2013, Zal et al. 2013). Em sua grande maioria, são variações de teclados virtuais, mais conhecidos por seu termo em inglês, *spellers*. A concepção geral baseia-se em apresentar várias opções (ao menos duas) de interesse do usuário em uma única tela. Para a seleção de opções e comandos, vários dos paradigmas apresentados nas Seções 2.1.2 e 2.1.3, páginas 9 e 10, podem ser utilizados.

De acordo com Bi et al. (2013), os três paradigmas mais utilizados para controle de robôs, em se tratando de BCI, são o P300, o SSVEP e a abordagem por Sincronização Relacionada a Eventos (ERS do inglês, *Event-Related Synchronization*) e Dessincronização Relacionadas a Eventos (ERD do inglês, *Event-Related Desynchronization*). Conforme o autor, tanto o P300 e o SSVEP possuem a vantagem de praticamente não necessitar de treinamento, sendo que o SSVEP possui tempo de classificação tipicamente entre 2 e 4 segundos.

Ainda de acordo com Bi et al. (2013), o tempo do P300 depende da quantidade de opções disponibilizadas no menu para a escolha, podendo requerer poucos segundos para poucas opções até 20 segundos ou mais em caso em que o número de opções seja alto (por exemplo, o nosso alfabeto). Em relação à técnica de ERD/ERS, necessita-se de um tempo maior na fase de treinamento para que o usuário se acostume com a técnica, mas seu tempo de classificação fica entre 0,5 e 2 segundos, conforme relata o autor.

Em se tratando de controle de robôs móveis, além dos exemplos citados na Seção 2.2, página 13, ainda pode-se exemplificar o trabalho de Escolano et al. (2009) utilizando P300 e controle compartilhado tipo chaveamento de ações, Geng et al. (2010) com ERD/ERS e uma abordagem com controle compartilhado tipo chaveamento, e finalmente, Silva et al. (2013), utilizando EMG e uma abordagem de navegação por controle compartilhado com uma cadeira de rodas virtual. Após uma análise dos paradigmas apresentados na literatura, optou-se pela interface apresentada nas seções a seguir.

O projeto DesTINe, do qual este trabalho de pesquisa faz parte, possui em seu escopo o desenvolvimento de uma HMI assistiva. Grande parte do trabalho com as opções da HMI, tal como a parte de estruturação visual (*layout*), bem como algumas das opções foram desenvolvidas em conjunto com o Dr. Eric Rohmer, atual pesquisador em modalidade de pós-doutorado do Laboratório de Automação e Computação (LCA) da Faculdade de Engenharia Elétrica e de Computação (FEEC) da Universidade Estadual de Campinas (UNICAMP). Existem diversas opções, para as mais diferentes aplicações assistivas. As opções que não dizem respeito ao conteúdo desse trabalho de pesquisa não serão abordadas.

Para a obtenção dos resultados apresentados nesse trabalho de pesquisa, desenvolveram-se duas opções específicas para os controles assistido (compartilhado) e autônomo. Essas interfaces de controle foram desenvolvidas integralmente pelo autor desta tese, e estão baseadas na revisão da literatura apresentada anteriormente. Suas funcionalidades, maneiras de utilização e resultados serão abordados nas próximas seções, mostrando como um usuário com deficiência pode se comunicar com o mundo exterior por meio da interface.



## 3.1 Interface principal

A interface principal é a primeira tela com a qual o usuário tem contato. É nela onde se dispõem os principais agrupamentos de opções existentes. Essa interface pode ser observada na Figura 3.1, contendo, ao todo, nove ícones dispostos em formato matricial.

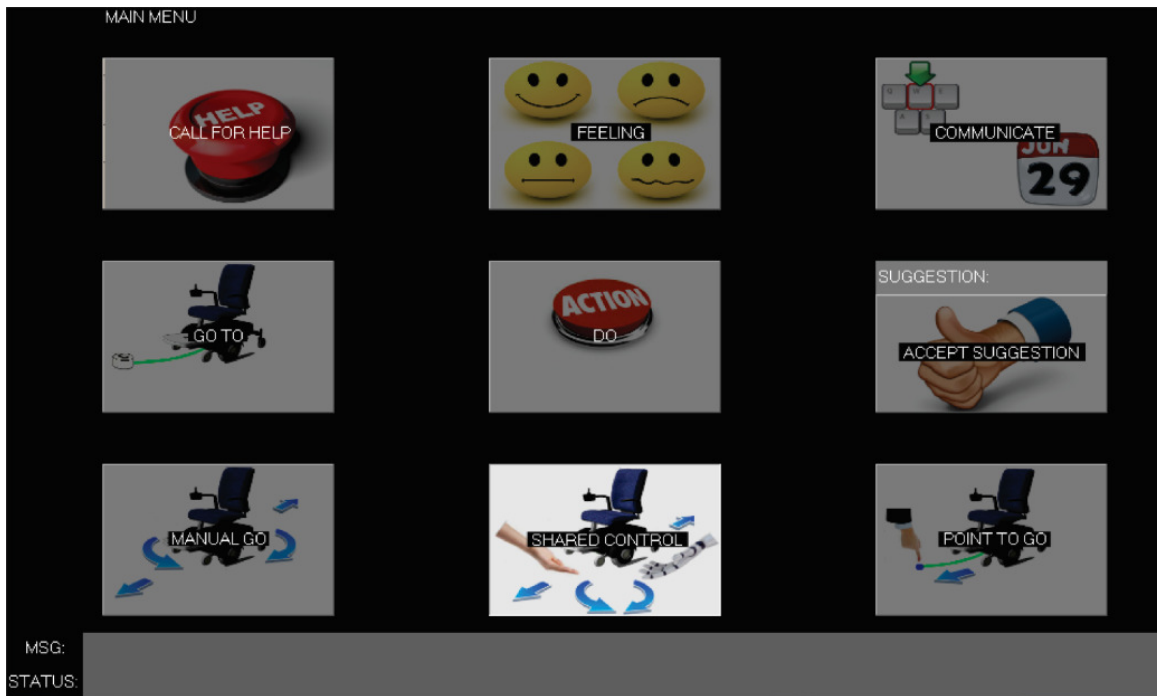


Figura 3.1: A interface principal (*main interface*) de comunicação.

A maneira de interação com essa interface, e todas as outras telas de opções pode ser tanto com EMG quanto com BCI, sendo que para a última optou-se pelo paradigma P300. Além desses, há ainda a alternativa de empregar o SSVEP, que também está incluso no escopo do projeto DesTINE. Nesse tema, Uribe (2013) desenvolveu suas contribuições para o projeto reportadas na sua dissertação de mestrado.

A escolha de opções via SSVEP baseia-se no estímulo visual em frequências distintas. Na dissertação de Uribe (2013), foram utilizados diodos emissores de luz (LED, do inglês *Light Emitting Diode*) afixados na periferia do monitor disponibilizado ao usuário. Cada um dos LEDs pisca a uma frequência pré-determinada, e está associado a uma opção diferente do menu. A tarefa do usuário é olhar diretamente para o LED correspondente à sua opção desejada. Conforme relatado na Seção 2.1.3, página 10, e de acordo com vários autores citados anteriormente, há uma resposta da região do lobo occipital cerebral, o qual está associado à visão. Esta resposta se dá na mesma frequência do LED, ou em seus múltiplos. Por meio da captação desta frequência, a opção escolhida pelo usuário pode ser classificada. Esse método não será abordado nesta tese, que se concentrará na utilização de sEMG e BCI/P300.

A escolha das opções via P300 na HMI do projeto DesTINE foi elaborada conforme a maneira clássica apontada na literatura, na qual as opções disponíveis na tela piscam de maneira aleatória, para que estejam conforme o paradigma *oddball* (vide Seção 2.1.3, página 10).

Todas as opções piscam uma mesma quantidade de vezes, sendo possível regular o período de tempo entre cada destaque das opções. Foram utilizadas 8 exibições para cada opção, com 130 milissegundos de período de tempo para cada. O usuário deve concentrar-se em uma das opções e contar mentalmente todas as vezes que sua opção desejada for destacada. Ao final, há um processo de identificação da opção escolhida pela obtenção dos pulsos P300 emitidos em decorrência de seus esforços mentais.

Nesta tese, utilizou-se a plataforma BCI2000. O processo de obtenção dos pulsos P300 pode ser resumido pela Figura 3.2. Observa-se que, a partir do sinal de EEG total, obtêm-se janelas temporais de intervalo definido, a partir de todos os estímulos apresentados ao usuário. Faz-se, então, a média destas janelas para cada um dos canais de EEG. Depois, mediam-se todos os canais de interesse. O pulso se apresenta para a opção que o usuário contou mentalmente.

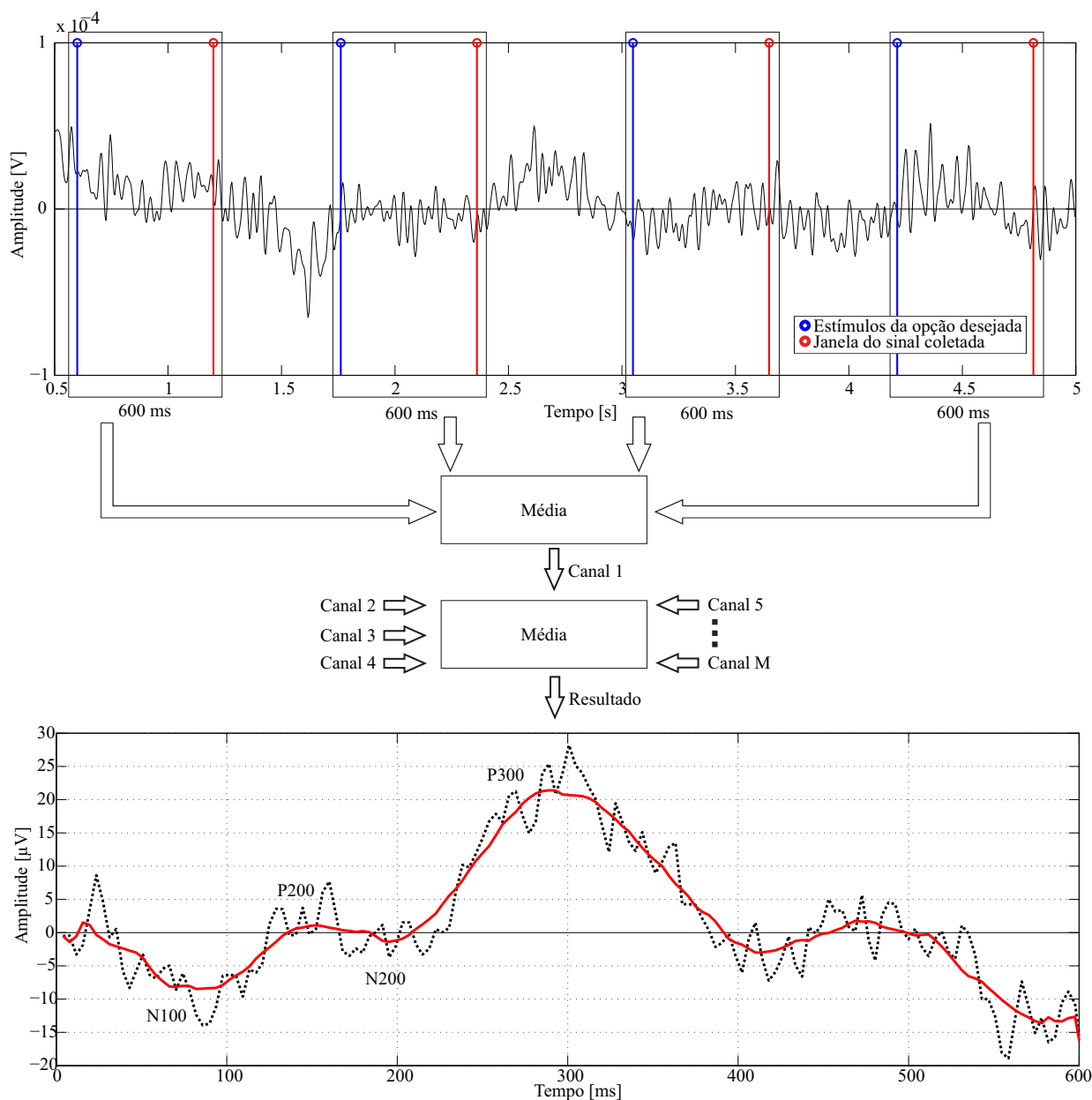


Figura 3.2: Método de obtenção dos pulsos P300.

No que concerne à utilização de sEMG com Emotiv, a escolha de opções se dá por meio de quatro canais, representados por quatro movimentos musculares com funções fixas, independentemente da interface que se apresenta ao usuário. No SDK do Emotiv há uma lista de vários movimentos faciais disponíveis. Os escolhidos e suas funções são relatados a seguir, e a justificativa dessas escolhas se dará na Seção 3.2.1.

Os movimentos de “olhar à esquerda” e “olhar à direita” são utilizados para seleções de opções, selecionando linhas e colunas. O movimento de “mastigar” é utilizado para confirmação, e o de “levantar as sobrancelhas” para cancelamento. Inicialmente, o usuário seleciona a linha onde está a opção desejada. Assim, confirma a seleção, passando para a fase de seleção da coluna, destacando a opção e confirmando-a. Se, porventura, acontecer algum erro de seleção, pode-se utilizar o comando de cancelar, e o usuário é remetido para a etapa imediatamente anterior.

## 3.2 Interface de controle compartilhado

A interface de controle compartilhado refere-se à opção de navegação assistida, para a qual desenvolveu-se uma técnica inédita na literatura, baseada em campos vetoriais, e previamente relatada em Olivi et al. (2013) e Olivi et al. (2014). Esta técnica, denominada navegação assistida, é independente de quaisquer métodos de aquisição de sinais. Desta forma, pôde-se oferecer ambos os paradigmas, sEMG e BCI, atendendo uma gama maior de usuários. No entanto, cada paradigma funciona diferentemente, e então, fez-se necessário desenvolver uma interface de comunicação conveniente para cada um.

### 3.2.1 Controle compartilhado via sEMG

A interface que oferece o controle via sEMG é mostrada na Figura 3.3. Nela podem-se observar os atalhos para o menu anterior (*main menu*) e para a chamada de ajuda (*call for help*). Em seguida há dois botões de giro, à esquerda e à direita. Estes dois comandos são os principais desse método, sendo a via de controle do caminho do robô. Quando utilizados, alteram a orientação (*heading*) do robô em  $\Delta\theta$ . O cômputo deste ângulo de giro será aprofundado no Capítulo 4. Por fim, existem os botões de aumento (*faster*) e diminuição (*slower*) da velocidade linear máxima que a cadeira de rodas robótica pode atingir.

Os movimentos faciais para controle da interface de sEMG são os mesmos relatados na seção anterior: “olhar à esquerda”, “olhar à direita”, “mastigar” e “levantar sobrancelhas”. Estes movimentos faciais foram escolhidos, dentre vários outros oferecidos pelo SDK do aparelho, por duas razões: a primeira é a intuitividade para os movimentos de giro do robô e de seleção de opções; a segunda é a alta taxa de acerto na confirmação e cancelamento de comandos.

Os comandos confirmar e cancelar são necessários para que o usuário possa observar livremente o ambiente por onde navega, sem que o robô faça movimentos indesejados. Portanto, uma ação somente será efetuada mediante confirmação. Conforme discutido na Seção 2.1.2, a classificação não é perfeita. Desta forma, efetuou-se uma série de experimentos para determinar a taxa de acertos do aparelho. As porcentagens de classificação do equipamento obtidos para o autor desta tese estão explicitadas na Tabela 3.1.

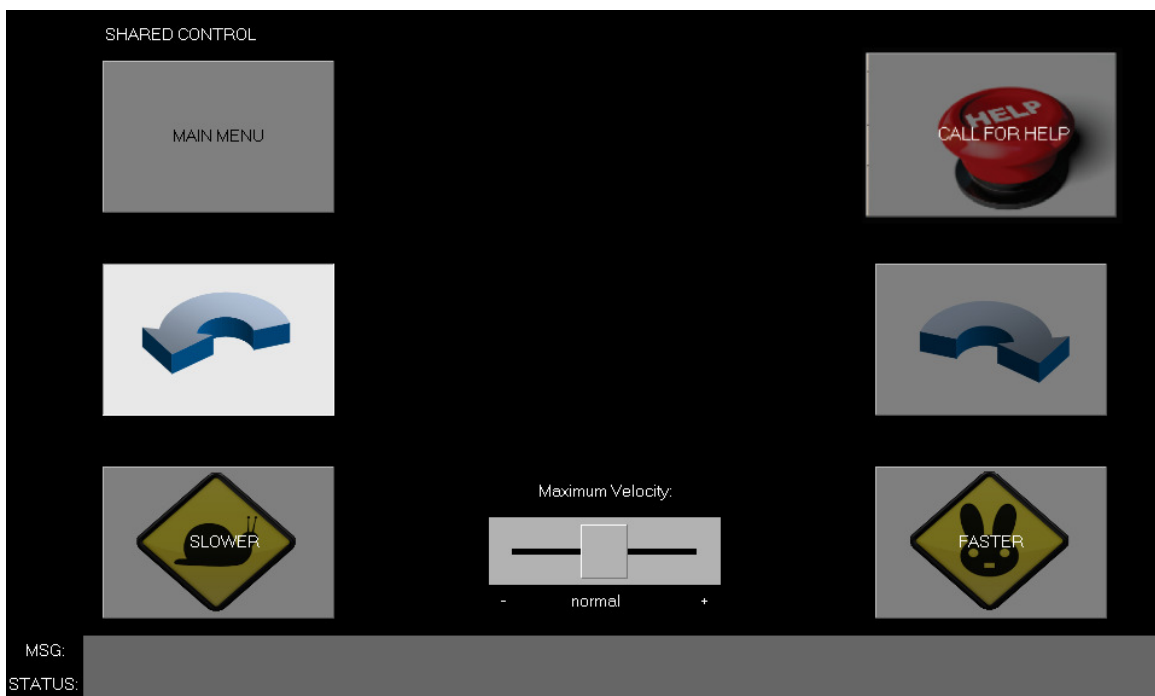


Figura 3.3: A interface de controle compartilhado via EMG.

Tabela 3.1: Classificação facial usando Emotiv.

Ação	Correto	Errado	Não classificado
Olhar à esquerda	74,67%	17,00%	8,33%
Olhar à direita	59,00%	19,67%	21,33%
Mastigar	96,33%	0,00%	3,67%
Levantar sobrancelhas	89,67%	0,66%	9,67%
Média	79,92%	9,33%	10,75%

Na tabela acima existem três possibilidades de classificação: o “acerto”, o “erro” e a “não identificação”. O acerto refere-se à classificação correta do movimento realizado pelo usuário. O erro ocorre quando qualquer outro movimento diferente do executado é identificado. A terceira opção não retorna qualquer saída quando o movimento é realizado. Estes resultados foram obtidos pela média de 300 aquisições para cada movimento facial, todos obtidos em uma mesma sessão de coleta de dados.

É importante ressaltar que, em se tratando de tecnologias assistivas, os resultados podem variar bastante entre usuários. Vários fatores influenciam esses tipos de resultados, tais como a colocação do aparelho no escalpo, a umidificação dos eletrodos, a oxidação dos contatos dos eletrodos, além dos fatores individuais dos usuários. Há relatos sobre usuários que não conseguem utilizar uma determinada técnica, enquanto outros obtêm altos índices de desempenho.

Com relação à utilização da interface, esta inicia seu funcionamento a partir da seleção da opção “*Shared Control*” no menu principal. Da mesma forma que o controle do menu principal, utilizam-se os movimentos de “olhar à esquerda” e “direita” para selecionar as linhas e colunas das opções. Ao confirmar a linha dos comandos de giro, inicia-se a navegação do robô.

A navegação assistida por controle compartilhado será abordada em sua totalidade no Capítulo 4. Contudo, adianta-se que, uma vez iniciada a navegação o robô desloca-se com velocidade linear máxima escolhida previamente nos botões de velocidade, sempre em frente. O usuário, com os comandos de giro, decidirá o caminho a ser percorrido. Conforme discutido, os comandos de giro são selecionados com os movimentos dos olhos.

Ainda assim, não é permitido que a cadeira de rodas execute o comando se não receber uma confirmação. Isso decorre da necessidade do usuário movimentar os olhos para observar o ambiente. Logo, o robô não deve executar comandos de giro sem o aval do usuário. O usuário pode finalizar sua navegação por meio do comando cancelar (no caso, “levantar sobranceiras”). Assim, retorna para o modo de seleção de opções, onde pode alterar a velocidade linear, retornar ao menu principal, ou então, utilizar o botão de emergência.

Comenta-se, por fim, que quaisquer um dos comandos faciais utilizados poderiam ser substituídos por outros fornecidos pelo SDK, como por exemplo, “piscadas”, “sorrisos”, dentre outros. Novamente, justifica-se a escolha dos comandos com movimentos dos olhos, por serem intuitivos e por serem comuns à maioria dos usuários que possuem deficiências motoras, uma vez que não estão ligados ao sistema nervoso central (Snell & Lemp 1997). Já os comandos de confirmação e cancelamento estão baseados na alta taxa de acerto, especialmente do movimento de mastigar, conforme mostrado na Tabela 3.1. De toda forma, o usuário tem a possibilidade de escolher outros movimentos faciais que lhe sejam mais aprazíveis ou precisos.

### 3.2.2 Controle compartilhado via BCI

Para a abordagem utilizando BCI, por meio do equipamento g.tec juntamente com a técnica de identificação de sinais P300, desenvolveu-se a interface exibida na Figura 3.4. Como se pode observar, esta interface requer uma disposição de opções diferenciada daquela concebida para sEMG. Esta configuração foi obtida por meio da plataforma BCI2000. O usuário escolhe dentre quatro opções: **L** para virar à esquerda (do inglês, *Left*), **R** para virar à direita (do inglês, *Right*), **S** para finalizar a navegação (do inglês, *Stop*) e **N** para o robô não executar quaisquer tipos de movimentos de giro (do inglês, *None*). Ou seja, quando o usuário desejar seguir em frente, ele deve escolher a opção **N**.

Conforme relatado nas seções anteriores, utilizam-se BCIs baseados na identificação de pulsos P300, pois esta é uma técnica que permite uma escolha dentre várias opções. No entanto, a quantidade de opções influenciará no tempo necessário para a classificação. Portanto, muitas opções poderiam demandar muito tempo, tornando a navegação robótica inviável, pois a velocidade do robô deveria ser muito diminuta para que o usuário tivesse tempo hábil de executar os comandos. Por este motivo, foram utilizados apenas quatro comandos.

O método de BCI P300, ao contrário de sEMG, sempre possui uma saída ao final da rodada de classificação, independentemente do usuário interagir com a interface. Por isso, foi necessário incluir a opção **N**, que faz com que o robô siga em frente, e o usuário deve escolhê-la, caso não queira alterar sua rota. Nota-se, inclusive, que esta é a única interface que não contém o botão de emergência. Isto ocorre justamente porque não há a opção de confirmação ou cancelamento, como na interface baseada em sEMG. Como a classificação não é perfeita, é possível que a opção emergencial seja acionada contra a vontade do usuário, e por isso foi retirada.

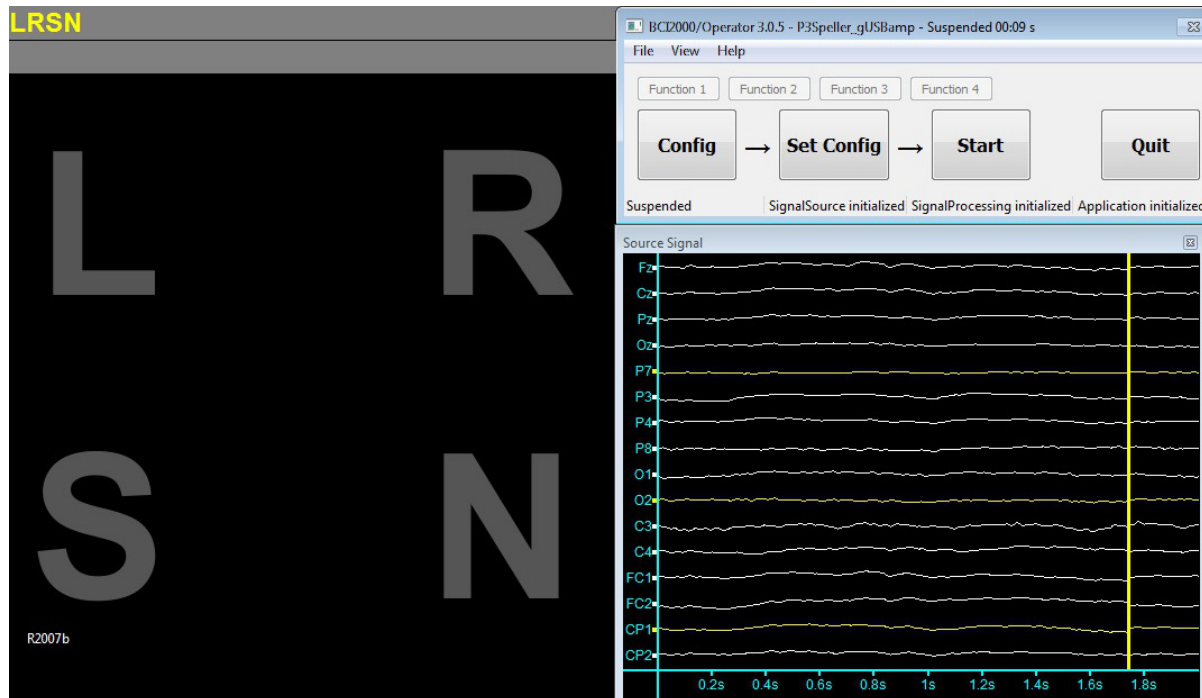


Figura 3.4: A interface de controle compartilhado via BCI/P300.

O aparelho g.tec utilizado disponibiliza 16 eletrodos secos para aquisição de EEG. As posições utilizadas, no Sistema Internacional 10-20 (ver Seção 2.1.3, página 10), conforme pode-se observar na Figura 3.4, são os seguintes:  $F_z$ ,  $C_z$ ,  $P_z$ ,  $O_z$ ,  $P_3$ ,  $P_4$ ,  $P_7$ ,  $P_8$ ,  $O_1$ ,  $O_2$ ,  $C_3$ ,  $C_4$ ,  $FC_1$ ,  $FC_2$ ,  $CP_1$ ,  $CP_2$ , sendo que a referência (REF) e o neutro (GND) ficam nos mastóides direito e esquerdo, respectivamente. Estas posições, de acordo com a literatura, são usuais para a captação de sinais P300. Apesar da opção pelos mastóides, também é comum utilizar a referência e neutro nos lóbulos das orelhas (Sanei & Chambers 2007, Ferreira et al. 2007, Trieu et al. 2008, Millán et al. 2010, BCI2000 2014).

O processo de utilização do método P300 passa por ajustes de filtros do aparelho g.USBamp e a coleta de dados de treinamento para o classificador. Os filtros do g.USBamp são aplicados diretamente no sinal dos eletrodos, dos quais ativaram-se dois. O primeiro é um filtro passa-faixa tipo *Butterworth* (Bianchi & Sorrentino 2007) de grau 8 com frequências entre 0,5 e 30  $Hz$ . Sua finalidade é excluir ruídos e outros sinais que estejam fora da faixa de resposta P300. O segundo é um filtro rejeita-faixa (*notch*) tipo *Chebyshev* (Bianchi & Sorrentino 2007) de grau 4, com frequências entre 58 e 62  $Hz$ , para retirar os artefatos introduzidos pela rede elétrica.

A fase de ajuste do classificador seguiu os passos do tutorial disponível na página eletrônica do BCI2000. Em suma, obtêm-se vários dados utilizando a interface em questão (Figura 3.4), de forma a reunir um grupo de treinamento. Assim, um processo de regressão é utilizado para calibrar os pesos de um filtro espacial a ser utilizado na etapa de utilização direta do método. Para a obtenção dos pesos, utilizou-se o método de Análise de Discriminante Linear Passo a Passo (SWLDA, do inglês *Stepwise Linear Discriminant Analysis*) (BCI2000 2014). O filtro espacial é o de Referência Média Comum (CAR, do inglês *Common Average Reference*) (BCI2000 2014), que é bastante popular em aplicações em BCI/P300 (Sanei & Chambers 2007).

Os resultados de classificação para cada uma das opções de controle estão listadas na Tabela 3.2. Conforme já discutido, no método P300 não há a possibilidade de não-classificação, portanto, os resultados aqui exibidos possuem apenas duas categorias, “correta” ou “errada”. Assim como a classificação de sEMG, para a obtenção das médias foram obtidas 300 aquisições de cada um dos comandos. Nesse caso, os dados foram adquiridos em dias diferentes, por tratar-se de um processo que demanda mais tempo que o sEMG, enfatizando-se que, assim como no sEMG, os resultados são individuais, e podem variar consideravelmente entre usuários.

Tabela 3.2: Classificação de quatro opções usando o sistema BCI2000.

Ação	Correto	Errado
<b>L</b>	89,67%	10,33%
<b>R</b>	85,00%	15,00%
<b>Q</b>	87,00%	13,00%
<b>N</b>	88,33%	11,67%
Média	87,50%	12,50%

Observa-se que não há grandes variações na taxa de acerto de todos os comandos, fato que não acontece na classificação via sEMG. Este fato reflete que o processo possui uma constância entre as opções, ou seja, pode-se modelar uma taxa de erros parecida para todos os comandos. O classificador P300 obteve rendimento médio superior ao sEMG, vide Tabela 3.1, página 28, com um ganho percentual de 8,66%.

Com relação à configuração das opções para este menu, os parâmetros foram calibrados de maneira empírica. Optou-se, finalmente, por 8 exibições com duração de 130 milissegundos para cada um dos comandos. Como o tempo do classificador é da ordem de milissegundos, o tempo total de cada rodada de classificação gira em torno de 4,2 segundos. É um tempo bastante alto quando comparado com a abordagem de sEMG. No entanto, para menores quantidades de exibições das opções, ou com tempos de exposição reduzidos, observou-se que a porcentagem de acerto era reduzida a níveis inaceitáveis, sendo, portanto, a configuração relatada a que fornece o melhor compromisso entre tempo por rodada de classificação e taxa de acerto. Mesmo com a taxa média de acertos maior que a do sEMG, o alto tempo de classificação obriga a velocidade do robô a ser menor, para que, conforme argumentado anteriormente, o usuário tenha tempo hábil para emitir um comando corretamente.

### 3.3 Interface de controle autônomo

A interface desenvolvida para a modalidade de controle autônomo é ativada a partir da escolha da opção “Go To” do menu principal (vide Figura 3.1, página 25). Conforme relatado nas seções anteriores, a abordagem autônoma envolve a escolha de locais pré-determinados, ou seja, previamente mapeados. Dessa forma, o robô, por meio de diversas técnicas, alcançará os locais conhecidos sem que haja necessidade do usuário fazer esforços mentais adicionais, deixando-o livre para exercer outras atividades. Portanto, a interface autônoma trata-se de uma matriz contendo diferentes locais previamente mapeados, conforme observa-se na Figura 3.5.

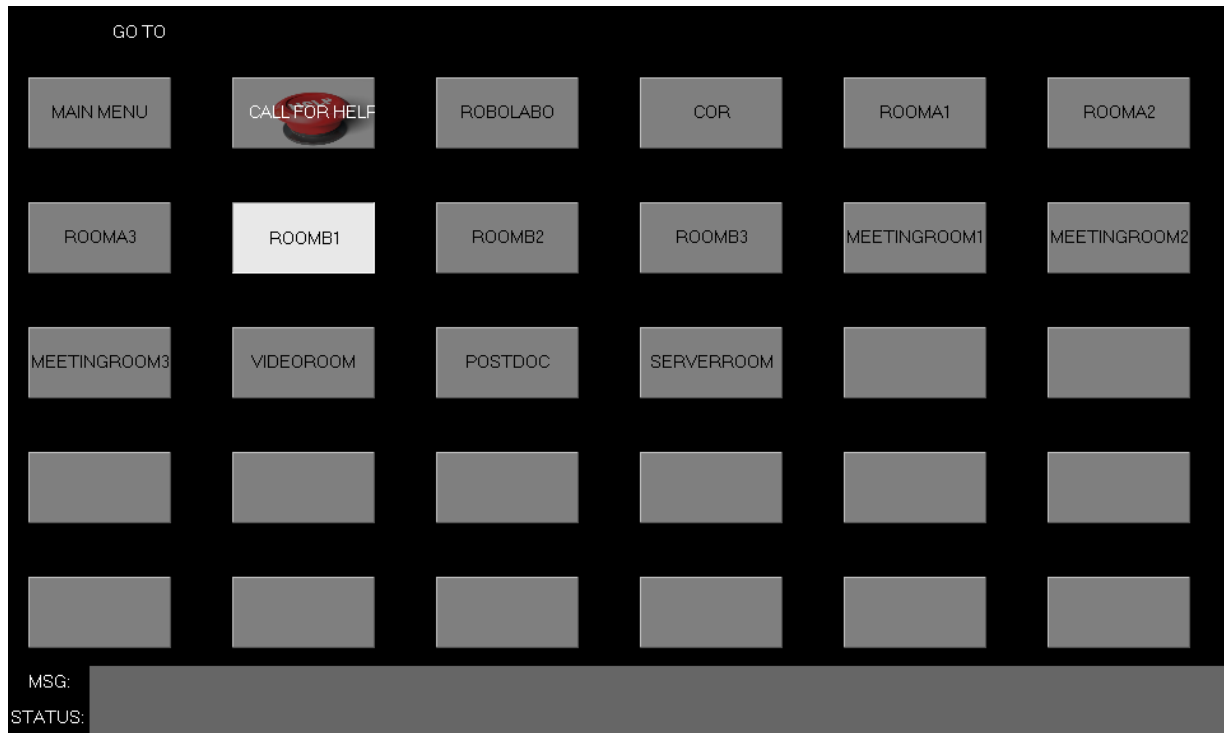


Figura 3.5: A interface de controle autônomo.

Na Figura 3.5 exibem-se diferentes locais do Laboratório de Computação e Automação (LCA). O mapa do laboratório em questão foi obtido por meio da etapa de navegação por controle compartilhado, com a utilização de um algoritmo de SLAM. Todos os detalhes acerca da obtenção desse mapa serão explicitados no Capítulo 4.

Todos os pormenores relacionados ao controle autônomo serão aprofundados no Capítulo 5, o qual mostra a aplicação dessa interface, juntamente com todos os seus resultados. No momento, exibe-se a configuração (*layout*) e a maneira de utilização desta interface, uma vez que ela é o meio de comunicação entre o usuário e o robô assistivo.

Conforme se observa na Figura 3.5, trata-se de uma interface simples, principalmente quando comparada com as utilizadas no controle compartilhado. Ela possui apenas as opções de locais pré-definidos alcançáveis pelo robô e acrescidos ao mapa levantado durante a navegação por controle compartilhado num processamento a posteriori. Da mesma forma que as outras interfaces, pode ser controlada tanto por sEMG quanto por BCI/P300, das maneiras relatadas nas Seções 3.2.1 e 3.2.2, respectivamente. Dessa forma, o usuário deve selecionar uma das opções, as quais representam locais do ambiente.

Quando o usuário escolhe um dos locais disponíveis nesta interface, podem haver duas situações relativas à posição do robô. Na primeira, apenas uma única posição com orientação,  $C_{x,y,\theta}$ , está associada àquele cômodo, por exemplo, sua porta de entrada. A partir disso, o robô calculará a rota para dirigir-se até o objetivo. Ao atingir este objetivo, o usuário poderá comutar para a navegação por controle compartilhado para se posicionar no interior da sala. Na segunda situação, pode-se abrir um mapa da sala escolhida e apresentar certas posições de interesse na sala, por exemplo, uma posição defronte à televisão.



Esta interface poderia ainda oferecer ao usuário a opção de adicionar uma determinada posição no mapa da sala que lhe seja conveniente durante a navegação. Isto é, uma vez que o usuário decida que está numa posição do ambiente a qual considera ser importante, ele poderia adicioná-la aos pontos de interesse da sala. Este ponto poderia ser selecionado posteriormente por um ícone contendo uma foto obtida com a câmera de bordo do robô, ou de uma câmera posicionada no ambiente, a partir do ponto de interesse, por exemplo.

Outra alternativa seria, ao entrar em uma sala, o usuário navegaria com o auxílio de uma interface tipo aponte-para-ir (*point-to-go*, Seção 2.3), apresentada na Figura 3.6. Nesta figura cada ponto retangular azul é uma posição que pode ser escolhida com o mesmo paradigma empregado na interface (sEMG ou BCI). Escolhido o ponto, o robô se desloca até ele autonomamente. Entretanto, estas facilidades não foram implementadas nesta pesquisa de tese.

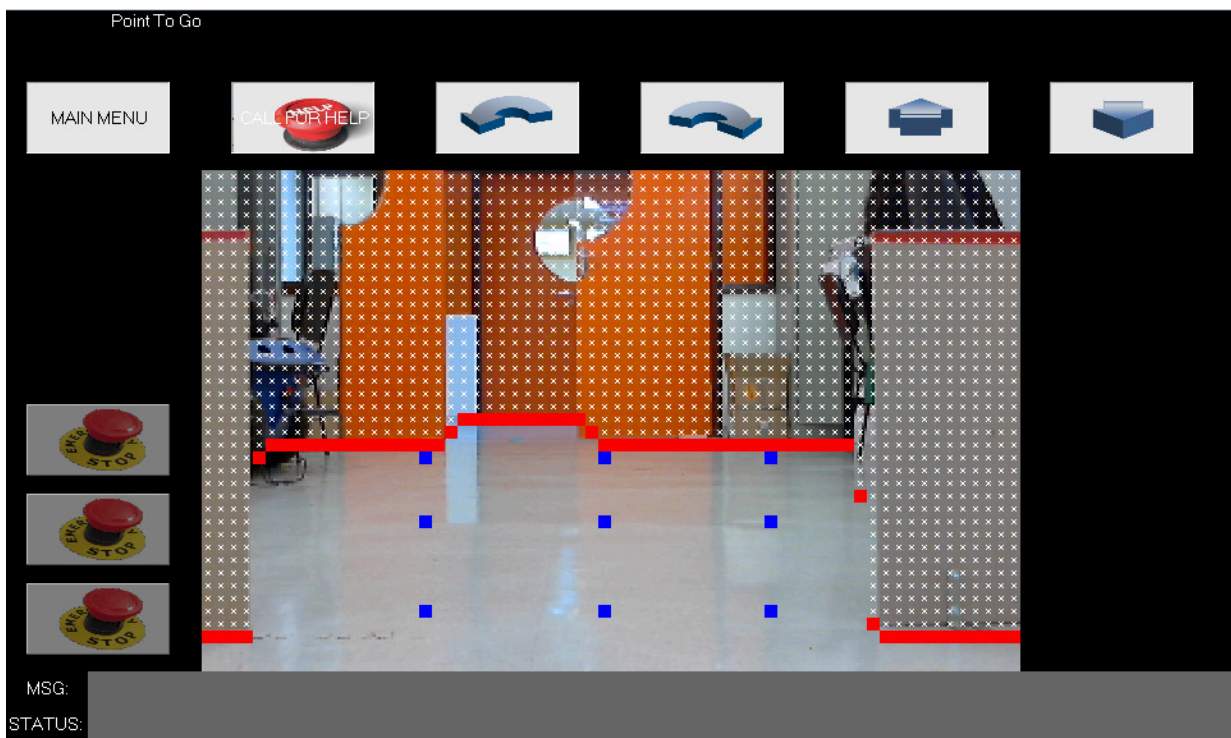


Figura 3.6: A interface de controle *point-to-go* desenvolvida no projeto DesTINE (cortesia do Dr. Eric Rohmer).

## 3.4 Discussões

Neste capítulo foi apresentada a interface de comunicação entre o usuário e o sistema robótico. A interface levou em consideração outras abordagens relatadas na literatura para o mesmo fim (Bourhis et al. 2001, Müller-Putz et al. 2006, Ramsey et al. 2006, Leeb et al. 2007, Vanacker et al. 2007, Philips et al. 2007, Trieu et al. 2008, Iturrate et al. 2009, Geng et al. 2010, Morales et al. 2013, Park et al. 2013, Zal et al. 2013). Desta forma, a HMI assistiva apresentada segue as mesmas diretrizes que têm apresentado bons resultados em diversos trabalhos.

A interface desenvolvida para o Projeto DesTINe, mencionado no Capítulo 1, preocupou-se essencialmente com a acessibilidade do usuário às opções disponíveis. Suas opções foram desenvolvidas em língua inglesa, no entanto, versões em português podem ser facilmente obtidas. A interface principal propicia o acesso aos principais grupos de opções disponíveis. Na navegação autônoma ficam disponíveis ao usuário os locais adicionados previamente à interface por especialistas. Em desenvolvimentos futuros, os locais poderão ser adicionados pelo próprio usuário, durante a navegação por controle compartilhado.

Para esta tese, desenvolveram-se integralmente duas opções da interface, as quais são a de controle autônomo e assistido, tanto para sEMG quanto para EEG. Ambas são técnicas desenvolvidas para navegações de longo termo, ou seja, em que o usuário pode navegar por distâncias grandes com menores esforços e interações com a interface. Ainda assim, a navegação por distâncias pequenas não estão excluídas dessas opções. Essas implementações são o objeto de discussão dos próximos capítulos. Todas as interfaces disponibilizadas são dispostas, essencialmente, em formatos matriciais, pois, dessa forma podem se adaptar tanto aos paradigmas de sEMG ou BCI. Para a utilização da interface desenvolvida, foram empregados dois tipos de aparelhos comerciais, em que ambos diferem significativamente nos seus respectivos preços comerciais para o uso de EEG e sEMG. Opções menos custosas devem ser avaliadas para posterior implementação no projeto.

O primeiro equipamento utilizado é o Emotiv (Emotiv 2014), que pode ser visto na Figura 2.1(a), página 9. Este foi utilizado para aquisição e classificação de sEMG para movimentos faciais, utilizando eletrodos molhados. Esse aparelho também é dotado da capacidade de processar sinais de eletroencefalograma (EEG). No entanto, na versão utilizada seu código-fonte é proprietário, o qual faz a transmissão dos sinais de EEG encriptados. Dessa forma não é possível utilizá-lo fora de seu SDK. O Emotiv possui eletrodos fixados por hastes, o que impossibilita utilizar o Sistema Internacional 10-20. Seus melhores desempenhos ficam em classificar sEMG. O SDK associa cada um dos movimentos a um padrão caractere do teclado. Dessa forma, sua utilização por outros programas depende de comandos que leiam caracteres, como o *“getchar”* do C++ ou o *“getkey”* disponível no MATLAB.

Os resultados percentuais do Emotiv podem ser observados na Tabela 3.1, página 28, com uma média de aproximadamente 80% de acerto. Ainda sim, existe uma grande disparidade de acertos entre os movimentos faciais utilizados, entre 59% e 79,67% de sucesso. Isso pode levar um usuário a, naturalmente, utilizar um movimento mais frequentemente que outro, moldando o caminho do robô a essas condições. Ainda com relação a essas taxas de acerto, observou-se que os resultados do usuário (no caso, o autor desta tese) não melhoraram com o passar do tempo, ou seja, o treinamento exaustivo não provocou melhoras no uso do sEMG já que esta melhora depende de fatores físicos que não podem ser alterados ao longo do tempo. Para técnicas de EEG costuma-se observar melhoria de desempenho com o passar do tempo.

Ainda assim, optou-se por estes comandos pois, conforme relatado na Seção 3.2.1, são intuitivos para escolher as direções de giro do robô, além de serem menos cansativos quando comparados com outros movimentos, como movimentos de “piscadelas”, “sorrisos” e “piscadas de olhos”. Além disso, movimentos oculares são comumente preservados em usuários com deficiência (Snell & Lemp 1997).

Ainda assim, existem casos extremos como por exemplo as pessoas em uma situação conhecida na literatura pelo seu termo em inglês, *locked-in*, em que sequer os movimentos oculares são existentes. Para essas pessoas, abordagens de sEMG não são eficazes, pois não há nenhum movimento muscular disponível, sendo necessário recorrer às abordagens que utilizam EEG/BCI, como o P300, dentre outros.

As taxas de acertos dos comandos de “mordida” e de “levantar as sobancelhas” são bem maiores, com 96,33% e 89,67%, respectivamente, e, especialmente, com baixos índices de não-classificação ou de classificação errônea. Por serem mais confiáveis, foram empregados em confirmações e cancelamentos, sendo determinantes para uma navegação com maiores acertos para os comandos enviados pelo usuário.

A principal vantagem da utilização de sEMG se encontra no tempo de classificação. O processo de identificação de movimentos musculares é feita por meio de detecção de artefatos. Um artefato, de maneira geral, é uma interferência na gravação do sinal. Um movimento muscular gera correntes elétricas que interferem na gravação dos sinais pelos eletrodos, introduzindo ruídos. Alguns movimentos possuem artefatos característicos que podem ser utilizados para identificação, como piscadas e mordidas. Ao contrário de ondas cerebrais, estes artefatos são detectados muito rapidamente, em cerca de 10 milissegundos (Emotiv 2014). Essa identificação rápida torna mais dinâmica a navegação da cadeira de rodas robótica, que pode atingir maiores velocidades.

O Emotiv ainda oferece outros tipos de entradas. A primeira é a de “pensamentos conscientes” (*conscious thoughts*), envolvendo a classificação de pensamentos propositais, como “puxar”, “empurrar”, “rotacionar”, “virar”, dentre outros. O equipamento oferece um jogo onde o usuário pode treinar estes pensamentos. Pode-se utilizá-los da mesma maneira que os movimentos faciais, por meio de associações a caracteres de teclado. A segunda é a classificação de “emoções” *affectiv suite*, onde se pode medir níveis de excitação, atenção (*engagement*), tédio, meditação e frustração. Entretanto, a confiabilidade na detecção destes padrões, os quais são obtidos por meio de EEG, em um equipamento de baixo custo como o Emotiv é extremamente baixa, o que inviabiliza sua utilização prática para controle de robôs.

O segundo equipamento utilizado nesta pesquisa é o g.tec, juntamente com o amplificador g.USBamp e 16 eletrodos secos. Este equipamento foi utilizado para as abordagens envolvendo EEG, com o paradigma P300. No entanto, é um equipamento de uso geral, que pode ser utilizado com quaisquer paradigmas envolvendo EEG e sEMG. Seria possível utilizar somente o aparelho g.tec para todo o trabalho desenvolvido, no entanto, este tem um custo comercial muito elevado, e por isso se faz necessário buscar também por alternativas menos custosas como o Emotiv. O amplificador g.USBamp possui alguns filtros disponíveis para isolar os sinais desejados dos demais, além da possibilidade de rejeitar algumas faixas espectrais específicas, como a de 60 *Hz* produzida pela rede elétrica.

Para o equipamento g.tec há ainda uma API para MATLAB, de onde podem-se obter diretamente os sinais de EEG. Com esse equipamento utilizou-se uma plataforma bastante difundida no meio acadêmico, o BCI2000 (Schalk et al. 2004, BCI2000 2014). Por meio desse sistema projetou-se a HMI com o paradigma P300, mostrada na Figura 3.4, página 30, contendo quatro comandos, conforme relatado anteriormente.

A porcentagem de acerto do BCI2000 consta na Tabela 3.2, página 31, com acerto médio de 87,50%. Diferentemente do Emotiv, nesta abordagem os comandos não possuem grandes discrepâncias de acerto entre si, o que determina um grau de confiabilidade maior entre os comandos. No Emotiv, o olhar à esquerda obteve maior taxa que à direita, podendo influenciar o usuário. No P300 isto não ocorre, uma vez que todos possuem relativamente o mesmo nível de acerto, sendo escolhido o que melhor se encaixa na situação que se apresenta, e não o que apresenta maior facilidade.

Deve-se notar que as taxas de acerto desses métodos são obtidas mediante treinamento. A literatura sobre P300 assinala que este método apresenta bons resultados para diversas situações, onde uma das maiores vantagens é a diminuta fase de treinamento que os usuários são submetidos. Sendo assim, trata-se de um método muito difundido na área de tecnologias assistivas (Serby et al. 2005, Iturrate et al. 2009, Millán et al. 2010, Bi et al. 2013, Postelnicu & Talaba 2013). No caso de abordagens neurológicas, também é comum evidenciar a melhoria de desempenho do usuário com o passar do tempo, desenvolvendo maiores habilidades em gerar os sinais adequados aos classificadores.

Nesta pesquisa, a taxa média de acerto do P300 foram maiores que a do sEMG, mas ainda maiores quando se compara os comandos de “esquerda” e “direita”, com ganhos de desempenho significativos, de 16,73% e 30,59% respectivamente. No entanto, por não possuir o comando de confirmação, o P300 sempre emitirá uma saída do classificador, e, por essa razão, o botão de emergência não foi incluído na interface.

O ponto negativo do P300 é seu tempo de classificação, pois essa abordagem necessita de uma coleta de EEG para que seja analisada, ao contrário do sEMG, que apresenta resultados praticamente instantâneos. Em decorrência disso, minimizou-se a quantidade de comandos a serem apresentados, ficando, ao final, apenas quatro opções, exibidas 8 vezes cada a um período de 130 milissegundos. Ainda assim, o tempo obtido foi muito maior que o do sEMG, com cerca de 4,2 segundos para cada escolha de comandos para o usuário. No entanto, o P300 se faz mais útil que o sEMG em situações como a mostrada na Figura 3.5, página 32, em que muitas opções são disponibilizadas para o usuário, reduzindo significativamente a quantidade de interações entre o usuário e a interface.

A primeira consideração a ser feita é com relação à identificação do sinal. Mesmo com uma etapa de treinamento, o paradigma P300 depende da existência de sinal. Ou seja, o usuário deve ser capaz de emitir o sinal para que este seja localizado. Sabe-se, por meio da literatura, que nem todos os usuários são capazes de gerar os sinais de maneira eficaz. Nesse aspecto, filtros paramétricos que utilizam exemplos de treinamento para o aprendizado, como o CAR, levam vantagem, pois eles incorporam as características individuais de cada usuário. O sEMG já não possui este tipo de vantagem, ou seja, as características peculiares de cada indivíduo proporcionarão diferentes taxas de acerto no classificador.

A segunda consideração pode ser remetida ao próprio aparelho de aquisição utilizado, fornecido pela g.tec. As diferenças de impedância, filtragens, dentre outros aspectos técnicos como os próprios eletrodos, podem ter provocado as diferenças observadas nos resultados entre sEMG e EEG. Os materiais do equipamento g.tec são superiores, propiciando uma coleta de dados com menor influencia de ruídos.

Um outro dado a ser analisado são os custos entre estes equipamentos, que são bastante discrepantes, o que se reflete especialmente no material com os quais são fabricados. Além disso, os eletrodos do g.tec são secos, e conseqüentemente, mais fáceis de serem ajustados, enquanto os do Emotiv são molhados, requerendo uma solução salina para condução elétrica, com maiores fontes de problemas, como oxidação e falta de líquido condutor, causando a deterioração do desempenho do equipamento.

As vantagens do Emotiv se dão com relação à sua portabilidade e baixo custo. O aparelho é praticamente auto-contido em seu capacete (*headset*), com autonomia de bateria para várias horas, e necessitando apenas de um dispositivo (*dongle*) conectado ao computador via USB para transmissão e recepção de sinais via radio-frequência. O equipamento g.tec possui vários aparatos, sendo de utilização menos prática, especialmente com relação a conexões por meio de cabos.

As interfaces discutidas neste capítulo levam em consideração a acessibilidade do usuário aos menus de opções, tais como sugestões e atalhos para as mais diversas atividades diárias. O objetivo fundamental foi oferecer uma via de comunicação para a utilização dos métodos de controle a ser relatados nos próximos capítulos.



## Navegação por controle compartilhado

Este capítulo é dedicado à navegação de um robô móvel assistivo por meio de controle compartilhado. Conforme descrito no Capítulo 1 e, principalmente, na Seção 2.2, página 13, essa é a abordagem onde o usuário mais possui influência sobre o caminho do robô, utilizando opções do tipo “vire à esquerda”, “vire à direita”, “acelere”, “pare”, dentre outras. A Figura 4.1 ilustra o conceito de controle compartilhado. Nesta figura, um usuário, por meio de uma interface assistiva baseada em EEG, envia comandos para um robô móvel. Tais comandos são executados em função das informações adquiridas do ambiente e do próprio robô. Desta forma, o responsabilidade de uma navegação confortável e segura é dividida entre o usuário e o robô.

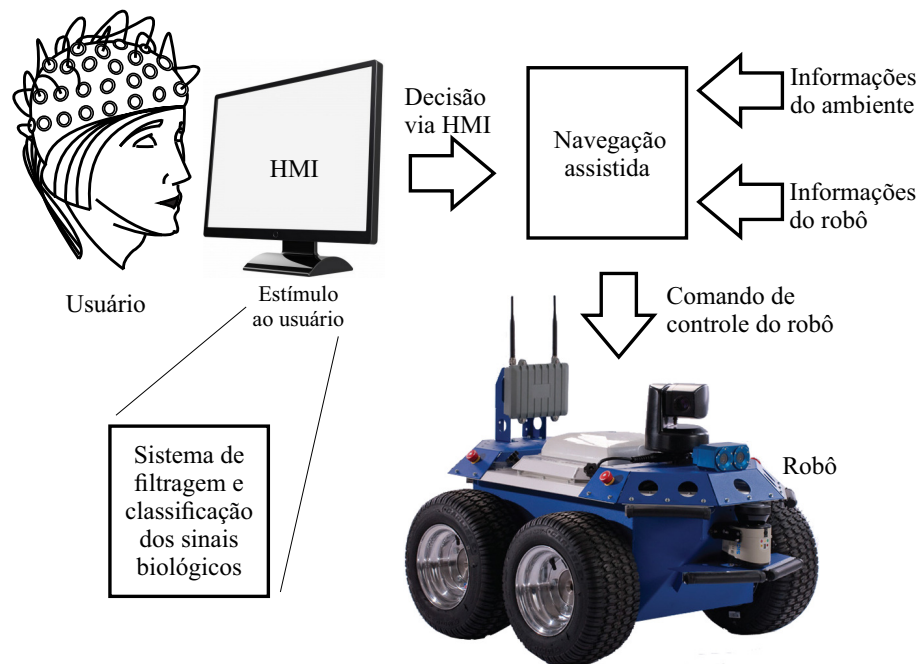


Figura 4.1: Ilustração de controle compartilhado, com Seekur Jr.

Conforme a Seção 2.3, existem muitas abordagens de controle compartilhado, mas, todas têm como diretiva proteger o usuário durante a navegação, executar medidas de correção quando há objetivos, minimizar os erros advindos da HMI assistiva, dentre outras, diminuindo a sobrecarga de funções do usuário.

No Capítulo 2 apresentou-se a evolução da área de controle compartilhado, mostrando algumas de suas abordagens, onde as mais usuais são desvios de obstáculos e ajuda para atravessar passagens estreitas (portas e corredores). Algumas utilizam algoritmos simples, enquanto as mais atuais têm utilizado técnicas sofisticadas com inteligência artificial, algoritmos adaptativos com predição de intenções e otimizações multiobjetivos.

Argumentou-se que grande parte destas técnicas possuíam um ponto em comum, o de retirar o controle do usuário para executar ações autônomas. Essas são estratégias dedicadas a situações específicas e possuem limitações, uma vez que situações não previstas sempre podem aparecer. Além disso, há o inconveniente de, eventualmente, não permitirem ao usuário executar determinada ação desejada, o que pode ir contra ações pré-programadas no robô. Finalmente, estas técnicas estão em acordo com a afirmação de Millán et al. (2010), que estabelecem apenas duas maneiras de realizar o controle compartilhado: os modos de operação são comutados, ou pelo próprio usuário, ou automaticamente por detecções de limiares.

Essa tese propõe uma técnica de controle compartilhado baseada na utilização de campos vetoriais a qual denominamos navegação assistida. Esta abordagem propõe-se a ser generalista, dispensando modelos do usuário, modelos de predições, cálculos de probabilidades, modelos paramétricos, estruturação do ambiente, dentre outros. Esta técnica proposta jamais retira o comando do usuário, pois não existem quaisquer ações integralmente autônomas.

Como o controle compartilhado é realizado sempre em conjunto com o usuário, este não fica impedido de acessar com segurança os locais que desejar. Dessa forma, este trabalho concorda com Cowan et al. (2012), mostrando uma terceira maneira de realizar o controle compartilhado. Esta técnica não se limita somente a cadeiras de rodas robóticas controladas por sEMG/BCI. Quaisquer tipos de veículos robóticos teleoperados que possuírem sensores para extrair distâncias de obstáculos, por exemplo, *lasers*, sonares, infravermelhos e câmeras, podem se valer desta abordagem.

Antes de apresentar a técnica de navegação assistida, faz-se um breve comentário acerca da possível associação que o leitor pode vir a fazer com a conhecida estratégia de controle “Campos Potenciais” (*Potential Fields*), em decorrência de ambos os nomes serem similares. Este método, proposto inicialmente por Andrews & Hogan (1983) para robôs manipuladores, foi adaptado para robótica móvel por Khatib (1985) e, desde então, tem sido bastante popular em aplicações e cursos de robótica.

O método de Campos Potenciais é um planejador de caminhos (*path planner*) e controlador autônomo, pois coordena os movimentos do robô até que este atinga um determinado ponto-objetivo. Os movimentos do robô são influenciados por meio de gradientes, que são produzidos pelos obstáculos existentes num mapa previamente disponível ao robô. O ponto-objetivo atrai o robô, enquanto os obstáculos o repelem, e desta forma, o robô navega pelo ambiente evitando obstáculos e, ao mesmo tempo, buscando o ponto-objetivo.

No entanto, além de necessitar um mapa do ambiente para calcular os gradientes dos obstáculos, o método dos Campos Potenciais possui sérios problemas quando opera em ambientes onde existem obstáculos de formato não-convexos, o que propicia o aparecimento de mínimos locais no caminho do robô, o que, conseqüentemente, faz o algoritmo convergir para um ponto indesejado do ambiente.



O método proposto nesta tese não se trata de um controlador autônomo, pois este papel deve ser de responsabilidade do usuário, eliminando os problemas com mínimos locais, sendo que o robô não é defletido ou atraído diretamente pelos campos vetoriais para não gerar situações adversas. A influência desses ocorre tão somente nos parâmetros de navegação (velocidades e direções). Por fim, esta técnica não necessita diretamente de mapas, embora também possa ser utilizada em conjunto com estes.

## 4.1 Navegação assistida

Esta seção relata os conceitos gerais da técnica de navegação assistida que emprega a estratégia de método de controle compartilhado baseado na utilização de campos vetoriais. Essa técnica se propõe a auxiliar a navegação do robô, que está sendo controlado por um usuário, via alguma HMI assistiva baseada, por exemplo em sinais cerebrais (BCI) ou musculares (sEMG). A proteção da integridade do robô é o tópico principal, porém, sem limitar os movimentos e locais alcançáveis pelo mesmo.

O conceito principal desta técnica é o da elaboração de comportamentos para o robô, o que se dá por meio da imersão do veículo em um ou mais campos vetoriais. Ambientes cotidianos, tais como casas, hospitais, locais de trabalho etc., são dinâmicos e oferecem as mais diversas situações ao robô. Este, por sua vez, deve ser capaz de moldar sua conduta à situação que se apresenta sem necessitar de ações autônomas. Uma maneira eficiente para delinear estes tipos de ações é por meio de campos vetoriais.

Campos de vetores podem informar ao robô as condições e restrições do ambiente de maneira direta. Os campos podem ser atrativos ou repulsivos, conforme a situação. Obstáculos próximos não são situações favoráveis à segurança, e portanto, devem emanar campos repulsivos, propiciando uma desaceleração ao robô, tornando a navegação lenta. Já áreas livres de obstáculos maximizam a segurança, sendo desejáveis de ser alcançadas e de se permanecer nelas. Dessa forma, estas devem possuir campos atrativos, propiciando maiores velocidades de navegação. É possível, para um projetista, criar campos que se adaptem às mais diversas situações, sendo este um método flexível.

O robô possui um conjunto de vetores pré-definidos que geram um comportamento padrão (*default*). Estes vetores padrão serão alterados pela interação com os campos vetoriais das imediações do robô. Assim, o comportamento do robô será modificado, adaptando-se às condições impostas pelo ambiente. Estes conceitos podem ser visualizados por meio das ilustrações contidas na Figura 4.2.

A Figura 4.2(a) mostra os parâmetros de controle do robô. Há quatro parâmetros obtidos a partir do centro do robô, sendo três vetores e um escalar. Os vetores são o “vetor de velocidade linear” ( $\vec{v}$ ), o “vetor de giro à esquerda” ( $\vec{s}_l$ ), o “vetor de giro à direita” ( $\vec{s}_r$ ) e, finalmente, o “raio da zona de proteção” do robô ( $R$ ). Os sobre/subscritos estão em língua inglesa para associarem-se às suas definições em Olivi et al. (2013) e Olivi et al. (2014). A figura mostram os vetores em suas configurações padrão (*default*), sendo, respectivamente,  $\vec{v}^{(d)}$ ,  $\vec{s}_l^{(d)}$ ,  $\vec{s}_r^{(d)}$  e  $R^{(d)}$ . Entre os vetores  $\vec{v}$  e  $\vec{s}_l$  calcula-se o ângulo de giro à esquerda ( $\theta_l$ ). Da mesma maneira, entre  $\vec{v}$  e  $\vec{s}_r$  encontra-se o ângulo de giro à direita ( $\theta_r$ ).

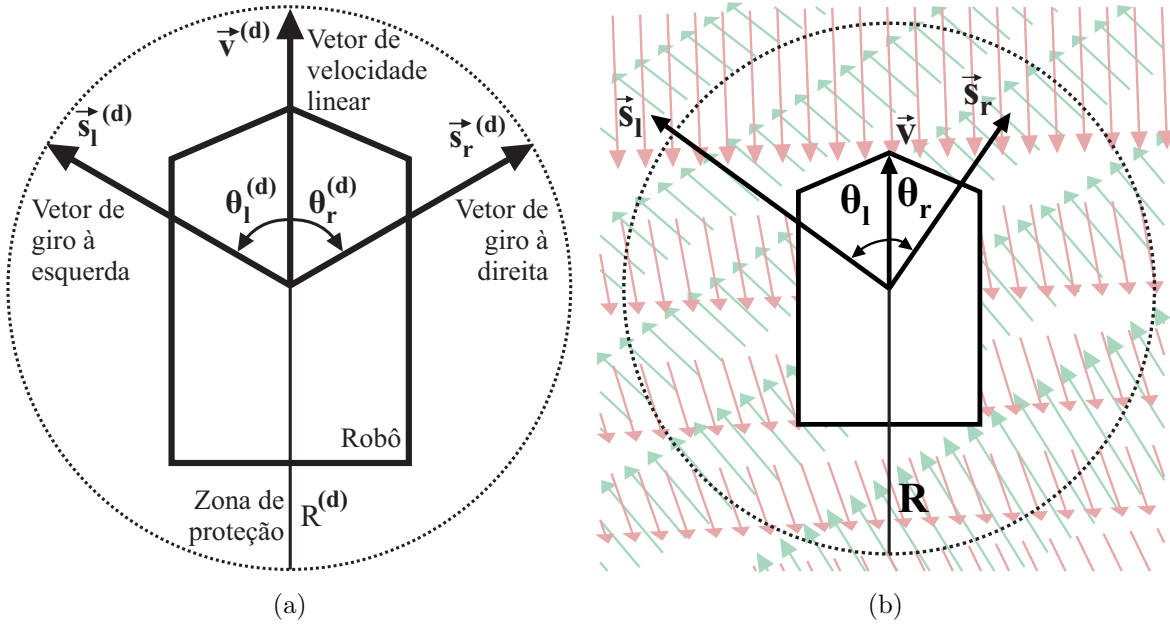


Figura 4.2: (a) Configuração padrão dos parâmetros do robô, e (b) campos vetoriais alterando o comportamento original do robô.

Conforme mencionado, os vetores padrão do robô imprimem um determinado comportamento. Ao entrar em contato com os campos vetoriais do ambiente, os vetores do robô são alterados, conforme pode-se notar na Figura 4.2(b). Nessa figura em questão, todos os vetores estão alterados, inclusive o raio da zona de proteção. Nota-se, inclusive, a diferença entre os ângulos de giro, cujos valores padrão eram iguais, o que configura um outro comportamento de navegação ao robô.

O vetor de velocidade linear,  $\vec{v}$ , conforme o próprio nome sugere, é responsável por regular a velocidade linear ( $v$ ) do robô. A norma padrão deste vetor,  $\|\vec{v}^{(d)}\|$ , é dada pela velocidade máxima ( $v_{max}$ ) escolhida pelo usuário para sua navegação, conforme mostrado na Figura 3.3, página 28. Esta velocidade máxima impõe um limitante inferior e superior das velocidades do robô ( $\|\vec{v}\| \leq v_{max}$ ), isto é, deve-se carregar o sinal (positivo ou negativo) relativo ao referencial do robô. A velocidade linear é, então, passada para o controlador de velocidades do robô para ser executada. Este vetor não será alterado por ações do usuário durante a navegação, e sim, por um campo vetorial resultante do ambiente.

Os vetores de giro à esquerda e à direita ( $\vec{s}_l$  e  $\vec{s}_r$ ) são utilizados para os cálculos dos ângulos de giro ( $\theta_l$  e  $\theta_r$ ). Nesta etapa, o usuário participa ativamente da navegação, onde escolhe para quais lados deseja orientar-se. No entanto, dependendo da situação, pode ser mais conveniente girar-se com um ângulo maior (ex: alcançar áreas livres) ou menor (ex: orientar-se para áreas com obstáculos). Esta é a razão de existirem dois ângulos de giro distintos, ao invés de apenas um. Esta situação será abordada com maior profundidade nas seções subsequentes.

A zona de proteção é a região ao redor do robô onde se procura por obstáculos. Esta zona possui um raio dinâmico  $R$ . Seu valor padrão,  $R^{(d)}$ , é relativo a seu limitante inferior, ou seja,  $R \geq R^{(d)}$ . Seu comprimento é dado pela distância Euclidiana da pose atual do robô ( $P_{x,y,\theta}$ ) até a zona segura (livre de obstáculos) mais próxima.

Dessa forma, a segurança do robô está intimamente ligada ao comprimento do raio  $R$ . Se o robô estiver próximo a uma zona segura (com baixa densidade de obstáculos), então seu comprimento será pequeno. Na situação contrária, se  $R$  é grande, então o robô está próximo de uma zona perigosa (com alta quantidade de obstáculos), e, por esta razão, a área de busca por obstáculos deve ser maior. Os detalhes serão aprofundados na Seção 4.3.

De acordo com as descrições dos parâmetros, pode-se condensar estas informações na forma de uma lei de controle para o robô. Seguindo a regra da mão direita, tem-se que

$$\begin{aligned} \text{movimento linear:} \quad & v^{\{t\}} = \pm \|\vec{v}^{\{t\}}\| && \text{onde } -v_{max} \leq v^{\{t\}} \leq v_{max} \\ \text{movimento angular:} \quad & \begin{cases} P_{\theta}^{\{t+1\}} = P_{\theta}^{\{t\}} + \theta_l^{\{t\}} \\ P_{\theta}^{\{t+1\}} = P_{\theta}^{\{t\}} - \theta_r^{\{t\}} \end{cases} && \begin{array}{l} \text{se o usuário quiser girar à esquerda} \\ \text{se o usuário quiser girar à direita} \end{array} \end{aligned} \quad (4.1)$$

onde  $P_{\theta}$  é a orientação (*heading*) do robô e  $t$  é o tempo atual.

A este método insere-se um último parâmetro, denominado de “raio crítico” ( $R_{crit}$ ). Trata-se de uma distância bastante próxima à colisão. Caso alguma das distâncias obtidas pelos sensores equipare-se a este raio, a velocidade linear do robô deve ser anulada ( $v = 0$ ) no mesmo instante. Em casos de campos vetoriais bem parametrizados, apenas a lei de controle anterior é suficiente para prover a segurança do robô. No entanto, recomenda-se adotar este procedimento como uma medida cautelar redundante, para o caso de uma eventual má calibração dos parâmetros de controle.

A Figura 4.3 mostra o método funcionando em uma situação real. Tem-se um robô navegando por um ambiente, adentrando um corredor estreito. Os obstáculos emanam um campo vetorial repulsivo  $\mathbf{N}$ . A região denotada por  $\mathbf{V}$  mostra as zonas seguras desse meio, a qual ajusta a zona de proteção  $R$ . A combinação dos vetores padrão do robô com os vetores dos campos moldam o comportamento do robô, que passa a ter ângulos de giro pequenos, propícios para este ambiente afilado, e velocidade linear reduzida, permitindo o avanço seguro.

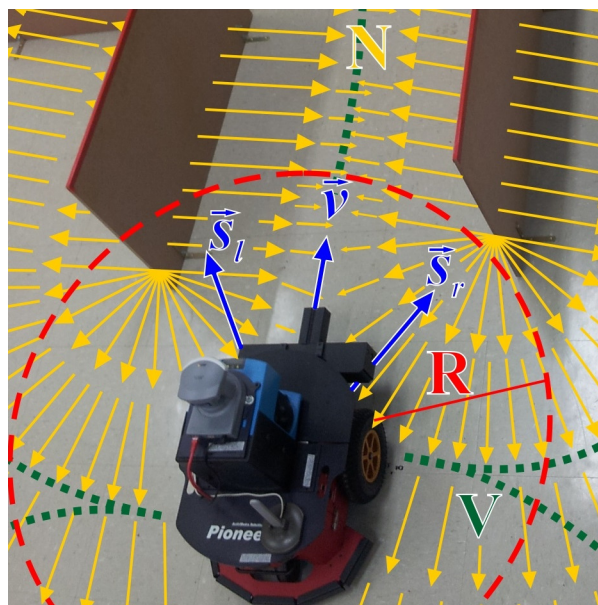


Figura 4.3: Campos vetoriais atuando sobre o robô Pioneer P3-DX.

A Equação 4.1 mostra que o comportamento desta abordagem faz com que o robô siga em frente. Os campos vetoriais não alteram a direção do robô automaticamente, o robô somente altera sua orientação caso o usuário deseje. Dessa forma, a influência deste no trajeto é determinante. Inclusive, este é o motivo pelo qual o método não possui quaisquer dificuldades com obstáculos não-convexos. Sempre que o usuário resolver alterar sua direção, o robô o fará por meio de operações que orientem o robô, as quais serão relatadas nas seções subsequentes.

Todos os parâmetros do método podem ser configurados antes da navegação começar, cuja calibração impõe sua influência diretamente à navegação. As normas padrões dos vetores de giro determinam a influência que os campos vetoriais terão nos ângulos de giro. Para facilitar a calibração dos campos vetoriais, adotou-se que a norma padrão dos vetores de giro é a mesma do vetor de velocidade linear. Não há obrigatoriedade neste procedimento, sendo apenas uma medida conveniente. O mesmo se pode estabelecer a respeito dos ângulos de giro padrão. Configuram-se ambos como sendo iguais, para que sua distribuição beneficie os movimentos do robô igualmente para ambos os sentidos quando não há influências de campos.

Conforme se pode notar, este método de controle compartilhado atua constantemente junto ao usuário. Enquanto este se preocupa em direcionar seu caminho pelo ambiente, o controle compartilhado apenas molda o comportamento do robô conforme o ambiente impõe suas restrições. A prioridade deste controle não é a otimização da navegação com relação à economia de energia, ou com relação ao menor caminho possível, e sim, com relação à segurança do robô e seu usuário, em que permite-se uma velocidade maior do robô em zonas seguras. Os conceitos de zonas seguras e perigosas, embora intuitivos, serão formalmente definidos nas próximas seções.

Observa-se que o controle compartilhado baseado em campos vetoriais não limita os locais alcançáveis pelo usuário, mesmo os considerados perigosos. Há vários campos de vetores que podem ser projetados. Esta tese apresenta o desenvolvimento de dois campos para cumprir os objetivos relativos à navegação de uma cadeira de rodas robótica (simulada pelos robôs da Figura 1.1, página 3) controlada por sEMG/BCI, que é maximizar a segurança sem limitar o acesso do usuário ao ambiente. Os resultados apresentados no Capítulo 6 foram considerados satisfatórios para todas as situações em que o robô foi submetido. A seguir tem-se a caracterização dos campos vetoriais utilizados na pesquisa.

## 4.2 Campo vetorial para controle da velocidade linear

Nesta seção descreve-se o campo vetorial utilizado para o controle da velocidade linear do robô. Conforme discutido na seção anterior, o objetivo principal do método é criar comportamentos para o robô por meio destes campos de vetores. A velocidade linear não é controlada pelo usuário, e sim, pelo campo associado a ela. O único controle da velocidade linear permitido ao usuário é o de determinar sua velocidade máxima. Dessa forma, esta abordagem deve ser capaz de controlar a velocidade de forma a privilegiar áreas seguras, permitir a navegação em áreas estreitas e evitar colisões do robô.

Para gerar o comportamento desejado, deve-se interagir com as informações sobre obstáculos, às quais são obtidas por meio de sensores. A descrição intuitiva do comportamento a ser gerado para o robô, com relação à posição dos obstáculos dispostos no ambiente é descrita a seguir.

Sempre que obstáculos estiverem a uma determinada distância a frente do robô, deseja-se que este pare para evitar colisões. Se, por algum fator externo, obstáculos se aproximarem demasiadamente da frente do veículo, é necessário que este se afaste. No entanto, caso os obstáculos estejam localizados nas laterais do robô (ex: corredores estreitos, passagens de portas etc.), quer-se permitir a navegação com uma velocidade linear reduzida, de forma que o usuário tenha tempo hábil para decidir o caminho a ser seguido.

Para gerar o comportamento mencionado, projetou-se um campo vetorial repulsivo  $\mathbf{C}$  com relação às distâncias do robô aos obstáculos. No caso específico desta pesquisa, as informações com relação aos obstáculos são obtidas por meio de sensores *laser*. Utilizaram-se dois tipos do sensor, o LMS-100 (robô Seekur Jr.) e o LMS-200 (robô Pioneer P3-DX), ambos da empresa SICK (SICK 2014).

Com ambos os *lasers* mencionados adquiriam-se as distâncias frontais do robô, com distâncias referentes aos ângulos  $\lambda \in [-90^\circ, 90^\circ]$  com passos de  $1^\circ$ , em decorrência desse ser o passo limite do modelo LMS-200. A Figura 4.4(a) ilustra, por meio do simulador MobileSIM (MobileRobots 2014), qual é a área coberta pelos *lasers* nas condições relatadas. Por estar imerso no campo, a pose atual do robô ( $P_{x,y,\theta}$ ) está associada a um vetor  $\vec{c}$  do campo  $\mathbf{C}$ , conforme fica exemplificado na Figura 4.4(b).

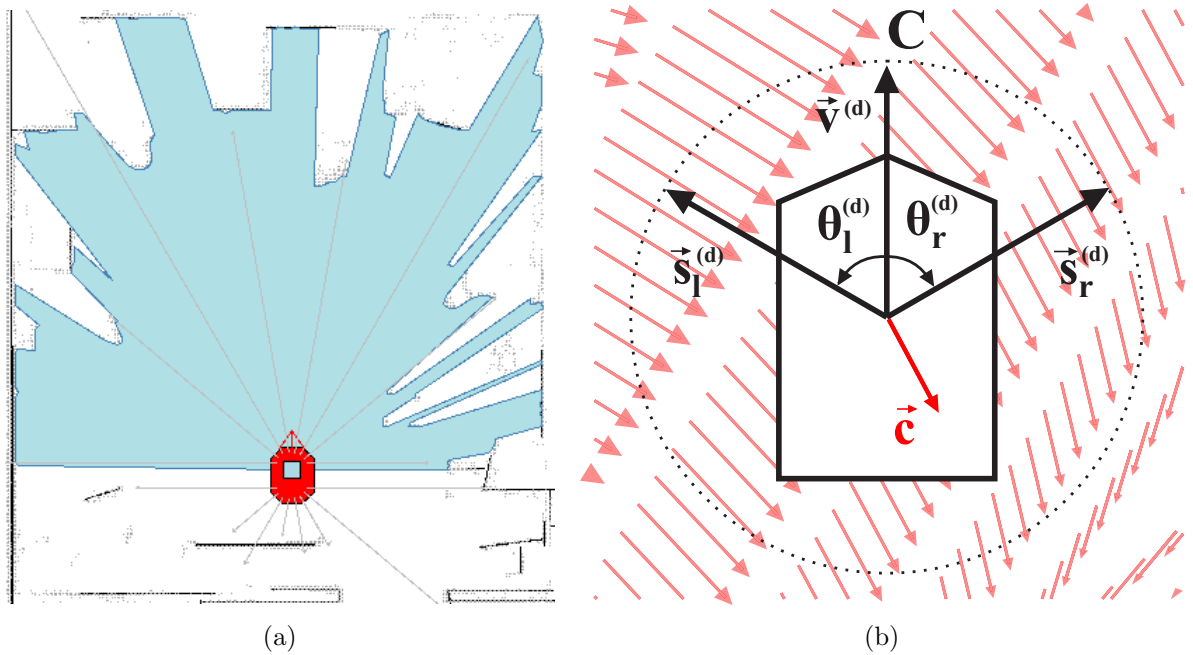


Figura 4.4: (a) Área coberta pelo *laser* (b) pose do robô associada a um vetor do campo.

As distâncias ( $d_i$ ) obtidas dos obstáculos contidos na zona de proteção formam vetores na direção contrária (repulsivos) dos obstáculos sobre o segmento de reta que conecta o robô e o obstáculo. O vetor repulsivo resultante  $\vec{c}$  também deve fornecer desacelerações suaves para o usuário. Atendendo a essas requisições, gera-se o vetor bidimensional a seguir:

$$\vec{c} = -\beta \sum_{i=1}^M \frac{R}{d_i} \begin{bmatrix} \cos(\lambda_i + P_\theta) \\ \sin(\lambda_i + P_\theta) \end{bmatrix} \quad (4.2)$$

onde  $M$  é o número de leituras de *laser* cujas distâncias  $d_i$  estão dentro da zona de proteção de raio  $R$ . Os ângulos de cada uma das  $M$  leituras de *laser*, no referencial (*frame*) do robô, são dados por  $\lambda_i$ . Por fim,  $\beta$  é um fator de escala a ser calibrado de acordo com o robô em uso.

A Figura 4.5 exemplifica o cálculo dos vetores utilizados nesta seção. Nesta figura tem-se uma varredura de distâncias nos ângulos do intervalo  $\lambda \in [-90^\circ, 90^\circ]$  no referencial do robô. Por meio das leituras de distâncias, pôde-se identificar  $M=3$  obstáculos contidos na zona de proteção de raio  $R$  à esquerda. Sobre as retas que passam pelo robô e os obstáculos, em sentido oposto, tem-se três vetores escalados pelo fator  $\beta$ . A soma destes três vetores fornece o vetor resultante  $\vec{c}$ . Portanto,  $\vec{c}$  é o vetor do campo vetorial  $\mathbf{C}$  associado àquela determinada pose do robô  $P_{x,y,\theta}$ . Conclui-se que a configuração de  $\vec{c}$  depende diretamente da pose atual do robô, o qual está em movimento pelo ambiente.

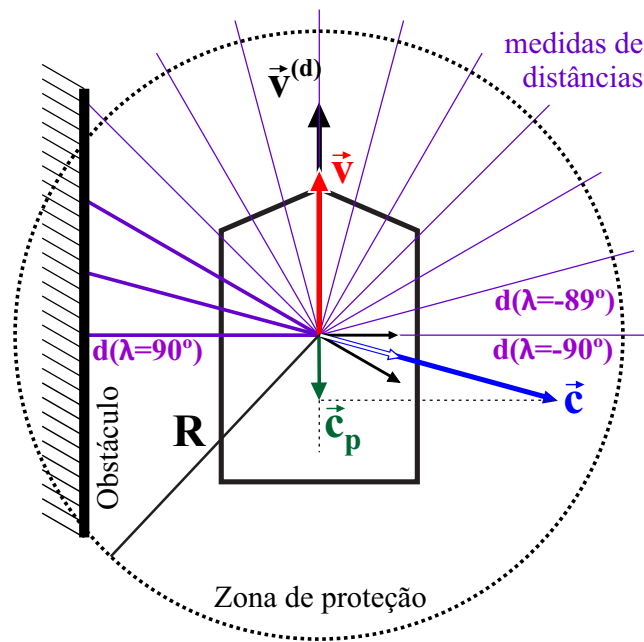


Figura 4.5: Cálculo dos vetores utilizados no campo repulsivo.

O método é flexível e permite combinar  $\vec{c}$  e  $\vec{v}^{(d)}$  conforme o comportamento planejado exigir. No entanto, a utilização de  $\vec{c}$  não possibilita cumprir todos os requisitos do comportamento desejado para a velocidade do robô. Exceto para o caso onde  $\vec{c}$  e  $\vec{v}^{(d)}$  são colineares, qualquer operação entre ambos produz uma resultante fora do eixo longitudinal do robô. Como a velocidade linear está sempre sobre a abscissa do robô, esta resultante não possui sentido geométrico para representar  $v$ . No entanto, a projeção de  $\vec{c}$  em  $\vec{v}^{(d)}$ , à qual denominamos  $\vec{c}_p$ , também mostrada na Figura 4.5, atende a todos os requisitos do comportamento planejado.

O vetor  $\vec{c}_p$  possui propriedades interessantes. Quando os obstáculos estão à frente do robô, a norma de  $\vec{c}_p$  atinge seus maiores valores, enquanto os obstáculos nas laterais produzem normas de baixos valores. É obtido pela seguinte composição de produtos vetoriais

$$\vec{c}_p = \left( \frac{\vec{c} \cdot \vec{v}^{(d)}}{\vec{v}^{(d)} \cdot \vec{v}^{(d)}} \right) \vec{v}^{(d)} \quad (4.3)$$

Combina-se  $\vec{c}_p$  com  $\vec{v}^{(d)}$  por meio da seguinte soma vetorial

$$\vec{v} = \vec{v}^{(d)} + \vec{c}_p \quad (4.4)$$

Essa soma produz resultados pertinentes ao comportamento. Para o caso de  $\vec{c}_p$  possuir normas grandes, o comprimento da resultante  $\vec{v}$  é pequeno, fazendo com que a velocidade do robô diminua a medida que se aproxima das zonas perigosas. A uma determinada distância suficientemente próxima dos obstáculos (a ser calibrada por  $\beta$ ), a norma de  $\vec{v}$  chega a zero, fazendo o robô parar completamente. E se, porventura, algum obstáculo aproximar-se além do limite que produz velocidade zero, o campo produz  $\vec{v}$  negativo. Neste caso, o robô poderá recuar até atingir novamente uma distância segura. Este recuo pode se dar, por exemplo, perfazendo o trajeto inverso até a distância segura ser alcançada.

A projeção  $\vec{c}_p$  é especialmente útil em corredores estreitos e passagens de portas, onde os obstáculos são laterais. Nesses casos, o vetor possui normas reduzidas, produzindo velocidades baixas, porém não nulas, o que permite a navegação segura do veículo, propiciando tempo suficiente para o usuário tomar decisões a respeito do caminho que deseja percorrer. Portanto, a projeção  $\vec{c}_p$  auxilia a acessibilidade do usuário a locais de difícil acesso, permitindo que seja este a executar os movimentos de passagem, dispensando atividades autônomas.

### 4.3 Campo vetorial para controle dos ângulos de giro

Esta seção se dedica ao campo vetorial que molda os ângulos de giro empregados nos comandos de giro emitidos pelo usuário. Da mesma forma que a seção anterior, esta se propõe a gerar um comportamento para o veículo robótico. Conforme descrito na Seção 4.1, os ângulos  $\theta_l$  e  $\theta_r$  possuem valores pré-definidos antes da navegação começar. Os ângulos de giro são a via de controle do usuário sobre seu caminho. Sendo assim, esta seção tem como objetivo a alteração dos ângulos de giro pré-definidos de forma a facilitar o controle do usuário.

Quando estes ângulos estão fixos (por exemplo,  $15^\circ$ ), é possível teleoperar um robô sem dificuldades. No entanto, é possível adicionar flexibilidade ao sistema de controle do usuário, pois, ao se alterar estes ângulos, criam-se comportamentos diferentes. Para o caso-foco desta pesquisa, que é a teleoperação via sEMG/BCI, quer-se maximizar a segurança do veículo, o qual conduz um usuário cuja capacidade de reação a situações de emergência está comprometida.

Para isso, manipulam-se os ângulos  $\theta_l$  e  $\theta_r$  de forma a privilegiar as áreas seguras identificadas pelo sensoriamento de distâncias. No contexto desta tese, considera-se como “áreas seguras” aquelas regiões que maximizam suas distâncias com relação a todos os obstáculos detectados. Em decorrência desta definição, as “áreas perigosas” são as áreas distantes das áreas seguras, ou seja, minimizam suas distâncias com os obstáculos conhecidos.

O comportamento que se deseja criar é facilitar o acesso das áreas seguras. Desta forma, os vetores de giro devem ser atraídos por estas áreas, proporcionando ângulos de giro melhor adaptados (geralmente maiores que os pré-definidos) para alcançá-las mais rapidamente, ou seja, com uma menor quantidade de interações. As áreas perigosas, conseqüentemente, passam a dispor de ângulos de giros menos propícios (geralmente menores que os pré-definidos) para serem atingidas.



Embora as áreas perigosas não sejam beneficiadas, não estão proibidas. O usuário ainda pode acessá-las caso desejar. Entretanto, deverá fazê-lo com passos mais comedidos, maximizando sua segurança, uma vez que manobrar em áreas repletas de obstáculos pode ocasionar colisões. A Figura 4.6 reutiliza o mesmo ambiente mostrado na Figura 4.4(a), página 45, para exemplificar qualitativamente o conceito de áreas seguras.

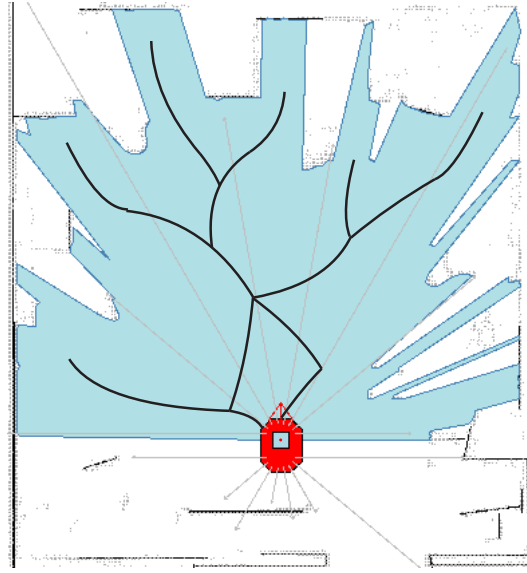


Figura 4.6: Áreas seguras de um ambiente.

Nesta tese utilizou-se o método do diagrama de Voronoi (Okabe et al. 2000, Siegwart & Nourbakhsh 2004), desenvolvido por Georgy Voronoy, para obter as zonas livres. Trata-se de uma maneira de subdividir um determinado espaço em um número de regiões (ou células), a partir de pontos fixos. As divisões das células são colocadas sempre na média entre os pontos adjacentes. Quando aplicado à robótica, considerando que os pontos fixos são obstáculos, obtém-se que as bordas das células são os locais mais distantes possíveis de todos os obstáculos, e portanto, as regiões mais seguras de um dado mapa. Outras técnicas para a obtenção das áreas seguras poderiam ter sido adotadas, não sendo o método de controle compartilhado proposto nesta tese, portanto, dependente direto do diagrama de Voronoi.

Considera-se o robô imerso num campo vetorial  $\mathbf{N}$  atrativo com relação às zonas seguras. Há, portanto, um vetor  $\vec{n}$  do campo  $\mathbf{N}$  associado à pose do robô, conforme mostrado na Figura 4.7(a). O passo inicial para obter  $\vec{n}$ , nesta tese, é por meio do diagrama de Voronoi. Tem-se que as leituras de sensores do robô fornecem uma boa estimativa das condições nas redondezas do robô. Portanto, pode-se utilizá-las para calcular o diagrama e decidir qual é o ponto do diagrama mais próximo do robô. Utiliza-se o ponto mais próximo pois, caso o usuário precise dirigir-se rapidamente a uma zona segura, este ponto é o ponto alvo mais indicado. Assim, remete-se à afirmação da Seção 4 sobre o método não depender de mapas globais, uma vez que medidas do referencial do robô (independentes de odometria) podem ser utilizadas.

No entanto, mapas globais são utilizados nesta pesquisa para suporte à navegação autônoma como descrito na Seção 2.4. Tais mapas são computados durante a navegação por controle compartilhado por meio de um algoritmo de SLAM como descrito na Seção 4.4.



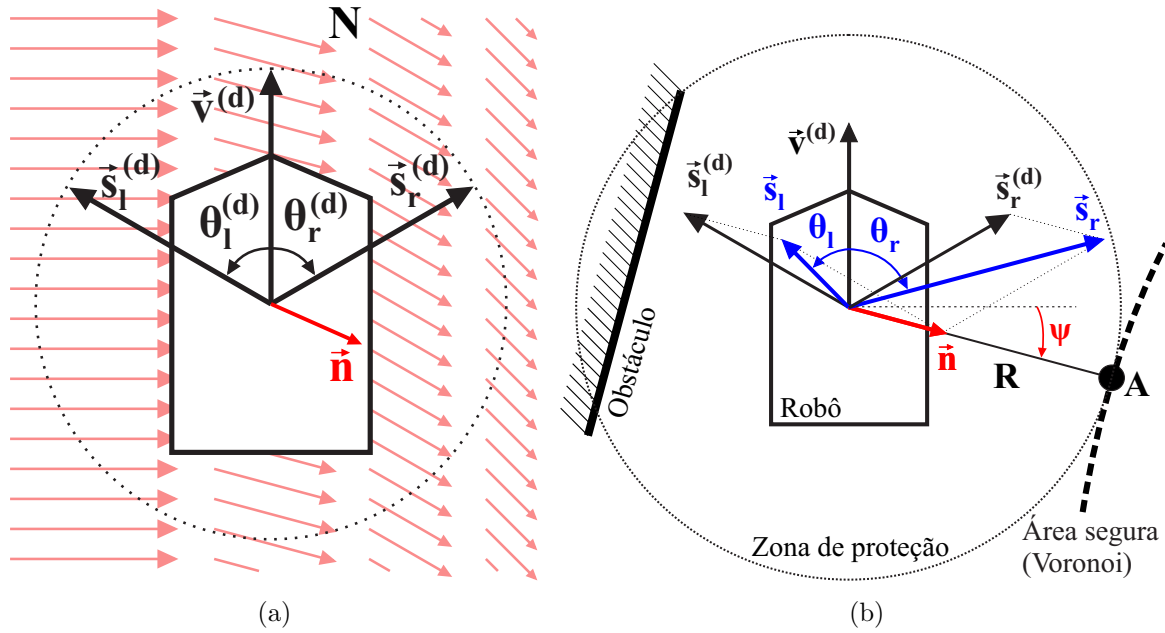


Figura 4.7: (a) Robô imerso no campo atrativo  $\mathbf{N}$  e (b) obtenção do vetor  $\vec{n}$ .

Estes mapas globais são armazenados juntamente com o diagrama de Voronoi já computado. Assim, caso o robô se encontre em uma região já mapeada, o diagrama de Voronoi não necessita ser novamente calculado, pois já está disponível no mapa global desta região. Uma vez que se dispõe do diagrama de Voronoi, o próximo passo é obter o raio da zona de proteção  $R$  do robô. Este raio é dado pela distância Euclidiana entre a pose do robô  $P_{x,y,\theta}$ , que deve ser conhecido globalmente, e o ponto do diagrama de Voronoi mais próximo, denotado na Figura 4.7(b) por  $A$ . Sendo assim,  $R$  é dado pelo máximo entre os seguintes valores

$$R = \max \left( \sqrt{(A_x - P_x)^2 + (A_y - P_y)^2}, R^{(d)} \right) \quad (4.5)$$

Se não houverem mapas globais,  $P_{x,y,\theta}$  é a origem do sistema do robô.

Além de  $R$ , calcula-se o ângulo entre a pose do robô e o ponto  $A_{x,y}$  por meio de

$$\psi = \text{atan2}(A_y - P_y, A_x - P_x) \quad (4.6)$$

Com as informações de  $R$  e  $\psi$ , é possível calcular o vetor  $\vec{n}$ , o qual é dado por

$$\vec{n} = \alpha R \begin{bmatrix} \cos(\psi) \\ \sin(\psi) \end{bmatrix} \quad (4.7)$$

em que  $\alpha$  é um fator de escala a ser calibrado de acordo com o robô utilizado e ao efeito desejado no comportamento. Quanto maior  $\alpha$ , maiores serão os ângulos de giro comparados com seus valores padrão. Verifica-se, por meio da Figura 4.7(b) e da Equação 4.7, que  $\vec{n}$  encontra-se sobre a reta que passa por  $P_{x,y}$  e  $A_{x,y}$ . Portanto, se  $\alpha = 1$ , então  $\|\vec{n}\| = R$ .

O passo final é combinar  $\vec{n}$  com os vetores de giro padrão,  $\vec{s}_l^{(d)}$  e  $\vec{s}_r^{(d)}$ , para obter os vetores de giro sob influência do campo  $\mathbf{N}$ . Estes novos vetores são dados pela seguinte soma vetorial

$$\vec{s}_{l,r} = \vec{s}_{l,r}^{(d)} + \vec{n} \quad (4.8)$$

Na Figura 4.7(b) observa-se que os vetores  $\vec{s}_l$  e  $\vec{s}_r$  produzidos são influenciados pelo diagrama de Voronoi, produzindo ângulos de giro maiores para as zonas seguras, e menores para as zonas próximas de obstáculos. Desta forma o usuário, caso deseje, pode alcançar as zonas livres mais rapidamente, obtendo velocidades de navegação maiores, conforme explicitado na seção anterior.

A obtenção dos ângulos de giro se dá por

$$\theta_{l,r} = \text{acos} \left( \frac{\vec{s}_{l,r} \cdot \vec{v}}{\|\vec{s}_{l,r}\| \|\vec{v}\|} \right) \quad (4.9)$$

Conforme pode-se observar pela Equação 4.9, os ângulos de giro estão contidos no intervalo  $0 \leq \theta_{l,r} \leq 180^\circ$ . Os maiores ângulos se dão quando o ponto  $A$  encontra-se na traseira do robô, de forma que o usuário possa alcançar as zonas seguras rapidamente. Finalmente, quando  $A$  encontra-se na dianteira, ambos os ângulos  $\theta_l$  e  $\theta_r$  são menores que os ângulos padrão, de forma que o usuário não possa divergir rapidamente das zonas seguras.

## 4.4 Mapeamento durante a navegação assistida

Conforme mencionado anteriormente, existe a cooperação entre a navegação assistida e autônoma, onde na assistida obtêm-se os mapas globais para serem utilizados nos procedimentos autônomos. Após a obtenção do mapa pelo robô, existem alguns procedimentos que devem ser feitos para configurar a navegação autônoma, tais como a extração de locais a serem alcançados no referencial do mapa, nomeação das salas, eventuais extrações de mapas topológicos para algoritmos de sugestão, como os de Raizer et al. (2013), dentre outros.

Existem várias modalidades de mapa que podem ser utilizados e construídos por robôs. Dentre eles, citam-se os mapas métricos, mapas de características (*features*), mapas topológicos, mapas em células, dentre outros (Siegwart & Nourbakhsh 2004). A construção de mapas emprega o sensoriamento do ambiente, no entanto, o próprio robô e os sensores possuem erros que, durante a navegação, vão se acumulando.

Para corrigir estes erros acumulados e mantê-los em níveis baixos, técnicas de localização e, especialmente, de SLAM são utilizadas. Muitas abordagens neste tema foram desenvolvidas, tais como o SLAM por Filtro de Kalman Estendido (EKF, do inglês *Extended Kalman Filter*), o GraphSLAM, o SLAM por Informação Estendida (*Extended Information SLAM*), dentre outros (Thrun et al. 2005). Nesta pesquisa, optou-se por utilizar o FastSLAM versão 2.0 (Montemerlo et al. 2002, Thrun 2002, Thrun et al. 2005).

Este método utiliza mapas de características. Utilizaram-se também mapas tipo células para armazenamento e exibição de resultados. Essencialmente, o FastSLAM é uma combinação do método de filtragem de Monte Carlo sequencial, conhecido como Filtro de Partículas juntamente com o Filtro de Kalman Estendido. Cada uma das partículas e dos pontos de referência (ou, simplesmente, referências, do inglês *landmarks*) são, de fato, EKFs de baixa dimensão.

Neste tópico será evitada a reprodução dos detalhes técnicos à respeito do FastSLAM. Tais informações podem ser encontradas na literatura, por tratar-se de uma técnica bastante difundida. Para maiores detalhes, recomenda-se, em particular, seu artigo original por Montemerlo et al. (2002) e o livro de Thrun et al. (2005), com excelentes fundamentações teóricas, bem como exemplos de implementação.

Por ser baseado no Filtro de Partículas, o FastSLAM é um método multimodal, possuindo várias hipóteses (partículas), e assim, mantém probabilidades posteriores relativas a múltiplas associações de dados. Esta é sua vantagem em relação aos métodos de SLAM unimodais, como os baseados em Filtros de Bayes e EKF, que mantém apenas a informação com máxima verossimilhança. Cada partícula é uma representação contínua do robô, sendo vantagem também com relação a métodos discretos, como o SLAM baseado em cadeias de Markov.

Sabe-se que o Filtro de Partículas possui complexidade computacional exponencial. No entanto, em Thrun et al. (2005) mostra-se que a abordagem utilizada no FastSLAM, conhecida por *Rao-Blackwellized Particle Filter*, permite sua implementação em tempo logarítmico com relação ao número de características. Outra vantagem se dá pelo fato de poder utilizar modelos não-lineares dos robôs sem necessitar de linearizações, ao contrário das técnicas como o EKF SLAM. Isto é importante quando a cinemática é altamente não-linear, ou a incerteza da pose é alta. Por fim, em decorrência da combinação do Filtro de Partículas com o Filtro de Kalman Estendido, o FastSLAM resolve ambos os problemas de *full SLAM* e de *online SLAM*, sendo um poderoso método de localização e mapeamento simultâneos (Thrun et al. 2005).

Para esta tese, utilizou-se o FastSLAM 2.0 sem correspondências conhecidas. Isto é, o robô parte de um ponto qualquer no ambiente com sua odometria nula ( $P_{x,y,\theta} = [0, 0, 0]^T$ ) e sem qualquer conhecimento prévio sobre a localização dos pontos de referência utilizados. As referências escolhidas são os cantos das paredes. Isto se dá por serem facilmente detectáveis com sensor *laser* e por, geralmente, serem espaçados uns dos outros, propiciando sua distinção.

Para a detecção das referências utilizou-se o algoritmo *Split-And-Merge* (Siegwart & Nourbakhsh 2004), que gera retas a partir das medidas de distâncias dos sensores. Os cruzamentos das retas fornecem as referências em questão. As características utilizadas são dadas por

$$Z = \begin{bmatrix} \sqrt{(L_x - P_x)^2 + (L_y - P_y)^2} \\ \text{atan2}(L_y - P_y, L_x - P_x) - P_\theta \end{bmatrix} \quad (4.10)$$

em que  $L_{x,y}$  é a posição espacial da referência. A primeira linha de  $Z$  é a distância Euclidiana (*range*) e a segunda é o ângulo (*bearing*) entre o robô e o ponto de referência. É comum acrescentar mais uma linha na Equação 4.10 com uma informação extra, denominada de “assinatura” (*signature*) da referência. No entanto, este parâmetro não foi necessário, sendo mais comum na identificação de marcas (referências) visuais por meio de níveis de cinza, matizes, cores, etc.

Os erros de odometria são representados pela seguinte matriz de covariâncias

$$\Sigma = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix} \quad (4.11)$$

onde utilizou-se  $\sigma_x=10$  mm,  $\sigma_y=25$  mm, e  $\sigma_\theta=1,25^\circ$ , obtidos empiricamente. Os erros do sensor laser estão representados na matriz de covariâncias de *range* e *bearing*, dada por

$$Q = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_b^2 \end{bmatrix} \quad (4.12)$$

onde, para a covariância da distância referida como “*range*” utilizou-se  $\sigma_r=5$  mm, e para o ângulo “*bearing*”,  $\sigma_b=0,25^\circ$ . O número de partículas foi 150, obtido por experimentação. Cada rodada durava, em média, 73,7 ms para 150 *landmarks*.

## 4.5 Pseudocódigo do método de navegação proposto

Nesta seção faz-se, por fim, um resumo do método na forma de um pseudo-algoritmo. Conforme discutido previamente, antes de iniciar a navegação há uma fase de definição dos parâmetros padrão de controle, a qual pode ser feita tanto pelo usuário por meio da HMI quanto pelo projetista. Assim, escolhe-se qual será a velocidade linear máxima ( $v_{max}$ ), o comprimento mínimo da zona de proteção ( $R^{(d)}$ ), os ângulos de giro ( $\theta_l$  e  $\theta_r$ ) e os parâmetros de escala dos vetores de campo,  $\alpha$  e  $\beta$ , os quais devem ser calibrados de acordo com o robô utilizado. O laço de controle é mostrado em seguida por meio de pseudocódigo.

---

**Algoritmo 1** Pseudocódigo do método de navegação por controle compartilhado baseado em campos vetoriais (navegação assistida).

---

```

1: enquanto a navegação não for terminada pelo usuário faça
2:   Adquira as medidas de distâncias ( $d$ ) dos sensores (ex: lasers, sonares, etc.);
3:   Determine a quantidade de distâncias contidas na zona de proteção ( $M$ );
4:   se  $M \geq 1$  então
5:     Calcule o vetor repulsivo dos obstáculos:  $\vec{c} = -\beta \sum_{i=1}^M \frac{R}{d_i} \begin{bmatrix} \cos(\lambda_i + P_\theta) \\ \sin(\lambda_i + P_\theta) \end{bmatrix}$ ;
6:     Calcule o vetor de projeção:  $\vec{c}_p = \left( \frac{\vec{c} \cdot \vec{v}^{(d)}}{\vec{v}^{(d)} \cdot \vec{v}^{(d)}} \right) \vec{v}^{(d)}$ ;
7:     Calcule o vetor de velocidade linear:  $\vec{v} = \vec{v}^{(d)} + \vec{c}_p$ ;
8:   senão
9:     A velocidade linear é máxima:  $\vec{v} = \vec{v}^{(d)}$ ;
10:  fim se
11:  se alguma distância  $d$  é menor ou igual ao raio crítico:  $d_k \leq R_{crit}$  então
12:    A velocidade linear deve ser nula:  $v = 0$ ;
13:  senão
14:    Aplique a velocidade linear:  $v = \|\vec{v}\|$ , onde  $-v_{max} \leq v \leq v_{max}$ ;
15:  fim se
16:  Obtenha as zonas seguras a partir das distâncias (ex: diagrama de Voronoi);
17:  Obtenha o ponto seguro mais próximo do robô:  $A_{x,y}$ ;
18:  Calcule o novo raio da zona de segurança:  $R = \max \left( \sqrt{(A_x - P_x)^2 + (A_y - P_y)^2}, R^{(d)} \right)$ ;
19:  Calcule o ângulo entre  $A$  e o robô:  $\psi = \text{atan2}(A_y - P_y, A_x - P_x)$ ;
20:  Calcule o vetor atrativo das zonas seguras:  $\vec{n} = \alpha R \begin{bmatrix} \cos(\psi) \\ \sin(\psi) \end{bmatrix}$ ;
21:  Calcule os novos vetores de giro:  $\vec{s}_{l,r} = \vec{s}_{l,r}^{(d)} + \vec{n}$ ;
22:  Calcule os novos ângulos de giro:  $\theta_{l,r} = \text{acos} \left( \frac{\vec{s}_{l,r} \cdot \vec{v}}{\|\vec{s}_{l,r}\| \|\vec{v}\|} \right)$ ;
23:  se existe algum comando de giro dado pelo usuário então
24:    Aplique o giro:  $\begin{cases} P_\theta^{\{t+1\}} = P_\theta^{\{t\}} + \theta_l^{\{t\}} & \text{se o usuário deseja girar à esquerda} \\ P_\theta^{\{t+1\}} = P_\theta^{\{t\}} - \theta_r^{\{t\}} & \text{se o usuário deseja girar à direita} \end{cases}$ 
25:  senão
26:     $P_\theta^{\{t+1\}} = P_\theta^{\{t\}}$ 
27:  fim se
28: fim enquanto

```

---

## 4.6 Calibração dos parâmetros

De acordo com o descrito neste capítulo, percebe-se que a calibração do método se dá de forma empírica. Isto se deve ao fato de cada usuário possuir um desempenho diferente no uso das HMIs, conforme relatado no capítulo anterior. Os parâmetros devem ser, portanto, adaptados de acordo com o desempenho de cada usuário.

Se um usuário possui alto desempenho na utilização das HMIs, é possível calibrar parâmetros de forma a permitirem maiores velocidades ao robô. No entanto, para usuários que possuem baixos desempenhos, parâmetros mais conservadores devem ser aplicados, restringindo altas velocidades de navegação para que, assim, o usuário tenha tempo hábil para decidir as direções para as quais deseja navegar.

Quando uma pessoa vai utilizar um aparelho robótico via métodos assistivos tipo sEMG/BCI, é necessário que haja uma fase de treinamento, onde esta pessoa se tornará um usuário por meio da prática, uma vez que os algoritmos de classificação utilizados nas HMIs dependem de parametrizações via dados de treinamentos adquiridos do próprio usuário, tais como os descritos na Seção 3.2.2, página 29. A pessoa em questão deverá aprender a utilizar as funcionalidades do sistema antes de utilizar o aparelho robótico real.

Durante essa fase de treinamento, é possível oferecer ao usuário um simulador do robô assistivo de forma que possa se familiarizar com a maneira de operação do mesmo por meio de uma HMI assistiva. A Figura 4.8 ilustra um ambiente doméstico cotidiano, sintetizado no simulador 3D V-REP (Freese 2014).

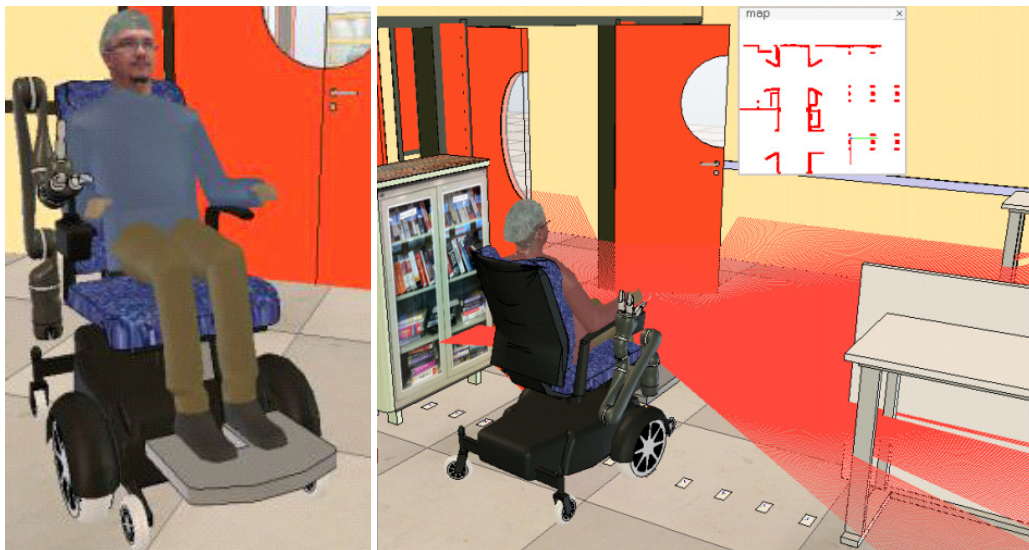


Figura 4.8: Ambiente simulado para treinamento de usuários do sistema robótico, modelado no simulador 3D V-REP (cortesia do Dr. Eric Rohmer).

Nesta fase de treinamento, é possível avaliar a facilidade do usuário para operar a HMI, ou seja, seu nível de acerto obtido na classificação dos comandos. Para realizar esta avaliação em ambientes simulados são submetidas tarefas para o usuário, tais como alcançar um determinado ponto-objetivo ou percorrer determinada rota pré-definida.

O desempenho do usuário pode ser inferido por um profissional de saúde verificando, por exemplo, a regularidade do trajeto percorrido e o nível de estresse do usuário (via, por exemplo, sensores corpóreos que meçam batimentos cardíacos e condutividade da pele). Por fim, uma interface de ajuste dos parâmetros de navegação pode ser construída para que, por meio desta, profissionais de saúde possam calibrar os parâmetros do método de navegação proposto.

Para esta tese, a regulação dos parâmetros se deu a partir de utilização do simulador MobileSim. Estabeleceram-se alguns caminhos para atividades de rastreamento (*tracking*), em que o usuário deve manter-se sobre uma rota pré-definida. Também utilizou-se o robô real para o ajuste fino dos parâmetros obtidos via simulação, de forma a permitir a navegação por passagens de portas e corredores estreitos sem que a navegação fosse impedida por esses ambientes.

## 4.7 Discussões

Este capítulo apresentou a técnica de controle compartilhado desenvolvida para esta pesquisa de doutorado e para o projeto DesTINE. Trata-se de um novo método que oferece liberdade de controle para o usuário e se baseia no emprego de campos vetoriais para imprimir comportamentos ao robô. A principal contribuição desta técnica é permitir que uma pessoa com deficiências motoras severas conduza um veículo robótico sem quaisquer intervenções autônomas, com total liberdade em escolher seus percursos.

Todas as ações autônomas discutidas na Seção 2.3 foram eliminadas, e resultados significativos foram obtidos frente às técnicas pesquisadas na literatura. Os resultados serão apresentados no Capítulo 6, os quais comprovam que, por meio desta técnica, o usuário consegue navegar por áreas estreitas e na presença de obstáculos sem impedimentos e com segurança. Os campos vetoriais moldam o comportamento do robô de forma a proteger a integridade do usuário. Caso o usuário anseie por retornar a zonas seguras, o comportamento angular permite alcançá-las rapidamente, com menos ações de comando e com maior velocidade.

No Capítulo 6 será demonstrado por experimentação que os campos vetoriais aqui exibidos atenderam às restrições impostas por ambientes diários e complexos, por onde um robô assistivo deve navegar. Adianta-se que os resultados mostraram que o ganho proveniente da utilização do método é significativo, atingindo porcentagens elevadas, especialmente em casos onde o erro das técnicas de classificação empregadas em sEMG/BCI são mais proeminentes. Além disso, mostrou-se que também a técnica é passível de ser utilizada com sensores de menor custo, como o sonar, o que propicia seu desenvolvimento em larga escala, possibilitando que diversos usuários tenham acesso à técnica e, conseqüentemente, melhorem suas condições de vida.

Mesmo os campos vetoriais mostrados nesse capítulo, apesar de atenderem às especificações de todas as situações apresentadas, são particularizações de um método generalista, conforme discutido previamente. De fato, uma grande quantidade de campos vetoriais podem ser combinados para controlar os parâmetros e imprimir os mais diversos comportamentos ao robô. Inclusive, o método pode ser utilizado com ou sem um mapa global, como se mostrou na seção que emprega sensores tipo sonar. Em matéria de compatibilidade, este método pode ser utilizado com quaisquer paradigmas de sEMG/BCI, ou mesmo em conjunto com outros tipos de teleoperação, sejam por comandos de voz, rastreamento de olhos, fluxo respiratório etc.

Com relação a trabalhos futuros, o método ainda permite a oportunidade de otimizar os parâmetros  $\alpha$  e  $\beta$ , que podem assumir a forma de funções, conforme o exemplo a seguir

$$\beta(\lambda_i) = a_1\lambda_i^2 + a_2\lambda_i + a_3 \quad (4.13)$$

em que  $\lambda_i$  são todos os ângulos correspondentes às  $i$  distâncias contidas dentro do raio de proteção  $R$ , obtidas, por exemplo, por *laser rangefinder*. Esta forma quadrática dependente, portanto, dos  $i$  ângulos das leituras de distâncias, no referencial do robô, por exemplo: os ângulos  $\lambda = [-90^\circ, -71^\circ, 18^\circ, 33^\circ]$  estão contidos na zona de proteção  $R$ , logo, deverão ser considerados para a quadrática. Da forma como está exposta, a Equação 4.13 aumenta a importância dos obstáculos laterais, possibilitando ao robô se proteger destes obstáculos a distâncias maiores. No entanto, más parametrizações podem impedir o robô de cruzar portas ou navegar por corredores estreitos. Outros exemplos podem ser derivados deste, onde técnicas adaptativas e de otimização podem ser empregadas, dentre outras.

Um dado importante a ser destacado é com relação à implementação do próprio código do método, mostrado no Algoritmo 1, página 52. Nota-se que nenhuma das operações relativas ao método é computacionalmente complexa ou que demande alto poder de processamento. Desta forma, a técnica pode ser implementada em processadores menos poderosos, baixando os custos de fabricação do robô assistivo.

Como no uso de qualquer tecnologia, falhas são passíveis de ocorrer, como por exemplo atrasos na interpretação ou aplicação dos comandos enviados pelo usuário. Tais atrasos podem colocar o robô em risco frente a obstáculos, como, por exemplo, um comando de desvio não emitido ou executado no momento requisitado. Para esse tipo de ocorrência introduziu-se um raio crítico de proteção ( $R_{crit}$ ), apresentado no final da Seção 4.1. Seu objetivo é parar o robô caso exista algum tipo de falha, seja por questões de atraso de comandos, por má parametragem dos campos vetoriais, ou algum outro motivo não previsto. Finalmente, detectores mecânicos de choque (*bumpers*) capazes de frear o veículo quando acionados são componentes fundamentais em robôs assistivos.

Por meio de todos os experimentos realizados para esse capítulo, pode-se afirmar que o método de controle compartilhado por campos vetoriais, quando comparado com os métodos citados na literatura, é uma técnica superior às até então propostas.





## Navegação autônoma

Este capítulo trata do último tópico desta pesquisa, o controle autônomo, que transporta o usuário de um robô assistivo por ambientes previamente conhecidos. A justificativa, conforme descrita na Seção 2.2, consiste no fato que as interações com HMI assistivas são cansativas para o usuário com o uso prolongado. Dessa forma, com a navegação autônoma, o usuário pode escolher um destino desejado, e ater-se a outras atividades. Esta é a cooperação com o controle compartilhado, em que o usuário não é requisitado enquanto o robô executa todas as operações necessárias. A Figura 5.1 mostra a interação entre o usuário e a interface de controle autônomo, a partir da qual os algoritmos assumem a navegação do robô até o ponto-objetivo.

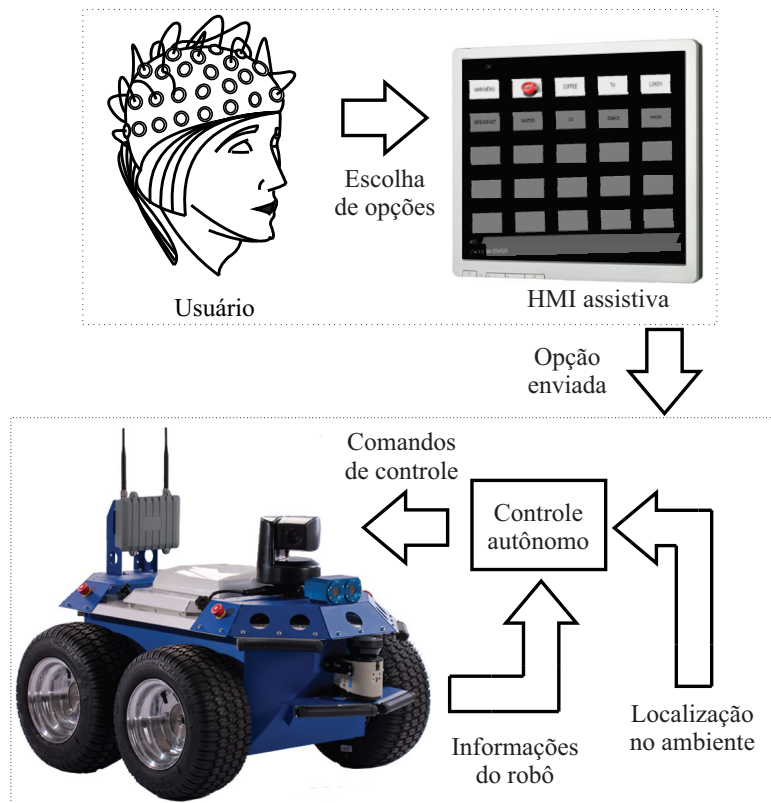


Figura 5.1: Interação entre o usuário e o controle autônomo (Seekur Jr., MobileRobots).

No entanto, navegar por ambientes de maneira autônoma requer seu conhecimento prévio. Do capítulo anterior obteve-se o mapeamento, por meio de FastSLAM, de sorte que ambientes antes desconhecidos passam a ser acessíveis por vias autônomas. Ainda assim, um grande conjunto de técnicas devem ser utilizadas para garantir os resultados esperados, tais como pontos de referência, técnicas de localização, controle e busca.

Por meio da interface assistiva principal (Seção 3.1, Figura 3.1), escolhe-se a opção “Go To” para entrar no modo de navegação autônoma. A próxima interface exhibe várias opções de localidades conhecidas (Seção 3.3, Figura 3.5), dentre as quais, o usuário pode selecionar uma localização específica como objetivo. Este é todo o esforço que o usuário deve fazer. A partir desse momento, o comportamento autônomo assume a navegação do robô e não exige outras interações do usuário até o cumprimento da tarefa ou seu cancelamento pelo usuário. Este capítulo tratará de todas as técnicas utilizadas e desenvolvidas no processo de navegação.

## 5.1 Armazenamento do mapa

Em problemas de navegação autônoma, geralmente os objetivos são determinados com relação a posições espaciais em um mapa. Desta forma, para que se decida um destino para o robô, é necessário conhecer o mapa do local a ser navegado. No capítulo anterior, a navegação por controle compartilhado obteve um mapa métrico do ambiente graças à correção da localização por meio do FastSLAM (Montemerlo et al. 2002).

No entanto, a determinação de um caminho entre dois pontos de um mapa, o ponto atual e o ponto-objetivo, necessita de um algoritmo de busca. Muitos desses algoritmos utilizam busca em grafos, em outras palavras, busca discreta. O algoritmo utilizado para esta tese, o  $A^*$ , a ser tratado nas próximas seções, realiza este tipo de procedimento. Contudo, é necessário discretizar o mapa métrico advindo da navegação por controle compartilhado (navegação assistida) em um mapa de células tipo grade (*grid*).

Existem diversas abordagens para a discretização em células. Para esta tese foram utilizadas células quadradas de lado  $L$  para fazer a divisão do ambiente. Testaram-se alguns valores para  $L$ , de forma a escolher uma opção que não fosse demasiadamente grande, a qual resulta em exacerbada perda de resolução do mapa, como fechar passagens possíveis. Observou-se também que seu valor não fosse demasiadamente pequeno de forma a evitar um tempo computacional elevado para o algoritmo encontrar a solução desejada. Dessa forma, para a navegação assistida de ambos os robôs utilizados (Pioneer 3-DX e Seekur Jr.), obteve-se o valor de  $L = 200$  milímetros.

Ao discretizar o mapa, obtém-se uma matriz que pode ser preenchida de diversas maneiras, como o mapa de ocupação (*occupancy grid*) (Moravec & Elfes 1985), que oferece um mapa de probabilidades dos obstáculos. Esse tipo de técnica é eficiente em corrigir o alto nível de erros dos sensores. No entanto, nesta tese foram utilizados sensores *laser* para obter medidas, os quais possuem alta precisão e acurácia, além da utilização de um poderoso algoritmo de SLAM, o FastSLAM. Tendo isto em consideração, utilizou-se um mapa de células com apenas dois níveis de distinção, livre ou ocupada (por obstáculos). A Figura 5.2 mostra o exemplo de discretização do mapa, onde se visualiza a perda de resolução no processo.

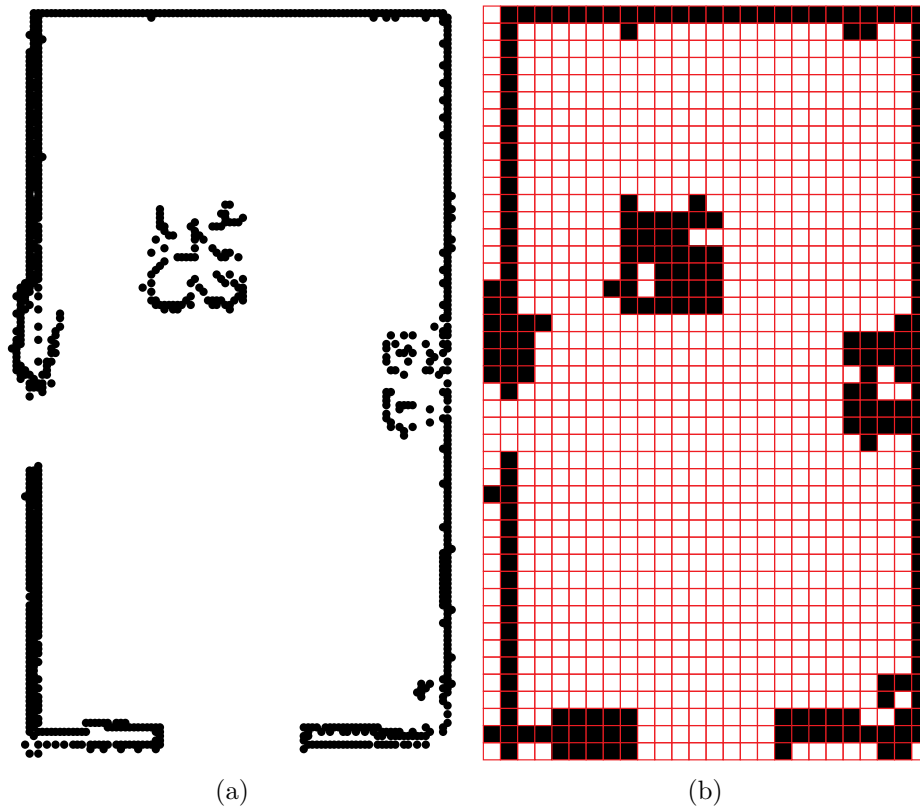


Figura 5.2: (a) Mapa métrico e (b) mapa de células, com perda de resolução.

## 5.2 Algoritmo de busca

Quando um ponto-objetivo é informado pelo usuário, é necessário planejar uma rota (*path planning*) que leve o robô de seu ponto atual ao objetivo. Na seção anterior, mencionou-se que esse trabalho de pesquisa utiliza o algoritmo de busca  $A^*$ . Esse método, desenvolvido por Hart et al. (1968), faz uma busca em grafos do tipo informada por “melhor escolha” (do inglês, *best-first search*), em outras palavras, a busca expande sempre na direção do menor custo. Trata-se de uma extensão computacionalmente mais eficiente do algoritmo de Dijkstra (1959).

Todo algoritmo de busca, para obter garantia de êxito, deve necessariamente possuir uma propriedade denominada “completude probabilística”. Esta propriedade assegura que a probabilidade de encontrar uma solução, caso ela exista, aproxima-se assintoticamente de 100% quando o tempo cresce ao infinito (Karaman & Frazzoli 2011). O algoritmo  $A^*$ , possui a propriedade denominada “completude”, que é mais forte que a completude probabilística pois garante a obtenção de uma resposta em tempo finito (Thrun et al. 2005, Russell & Norvig 2009), seja ela a solução desejada, ou então, uma mensagem de falha caso não exista solução. Aqueles algoritmos de busca que não possuem estas propriedades são denominados “incompletos”, onde não há garantias de obtenção de um caminho.

Além da completude, o  $A^*$  possui a vantagem de ser computacionalmente eficiente. Esta eficiência não se dá somente em comparação com o algoritmo de Dijkstra, mas num senso geral. Seu desempenho depende da heurística utilizada, onde seu pior caso de complexidade espacial é  $O(|V|)$ , ou seja, linear com o número de vértices (Russell & Norvig 2009, Lerner et al. 2009).

Ainda mais, o  $A^*$  não possui quaisquer dificuldades em lidar com mínimos locais, retornando sempre o caminho ótimo (no sentido de comprimento, o menor possível) sempre que ele existir. No entanto, esta última vantagem não vai ao encontro com um dos objetivos mencionados ao longo desta pesquisa, o da segurança do usuário.

O menor caminho possível, frequentemente, se aproxima em excesso de obstáculos, onde não é incomum obter caminhos margeando-os. Caso este caminho mínimo seja adotado, põe-se a segurança do usuário em risco, uma vez que os ambientes são altamente dinâmicos, além de possíveis incongruências na odometria serem passíveis de ocorrer. Colisões passam a ser um risco, mesmo com as mais cuidadosas medidas de segurança, como os campos potenciais, uma vez que falhas inesperadas e atrasos de execução de comandos podem ocorrer, tal como relatado na Seção 4.7. Por este motivo, fez-se uma modificação na heurística do  $A^*$ , de forma que se calcule o menor caminho possível contido nas regiões seguras, uma vez este é o melhor local para o usuário estar em situações de falha.

Num mapa de células, onde as células navegáveis (sem obstáculos) possuem um valor distinto das células obstruídas (com obstáculos), o objetivo do  $A^*$  é partir de uma célula inicial, explorar células vizinhas, até chegar à célula objetivo. É possível estabelecer uma conectividade de exploração entre as células. Nesta pesquisa utilizou-se conectividade 8, ou seja, exploram-se as 8 células adjacentes (vizinhas) à célula central, pois o robô utilizado é do tipo diferencial. Para a distinção entre as células navegáveis e obstruídas, as primeiras possuirão valores positivos, enquanto que as com obstáculos são denotadas por um valor negativo, para facilitar sua identificação, pois não serão exploradas pelo algoritmo. Isso se dá pois, caso os obstáculos possuam um valor positivo muito alto, ainda assim são passíveis de serem navegados.

Conforme discutido, a heurística de busca do algoritmo  $A^*$  é do tipo informada, indo sempre em direção ao nó de menor custo. Para calcular os custos, o  $A^*$  utiliza a seguinte função

$$\min F = g + h \quad (5.1)$$

onde  $g$  é dada por

$$g = d_{\text{inicio} \rightarrow \text{atual}} + d_{\text{atual} \rightarrow \text{vizinho}} \quad (5.2)$$

e  $h$  é obtida com

$$h = d_{\text{vizinho} \rightarrow \text{objetivo}} \quad (5.3)$$

onde  $d$  denota a distância Euclidiana entre as células indexadas. Há, ainda, a manipulação de duas listas, de células abertas e fechadas. Esta tese não fornecerá maiores aprofundamentos sobre o algoritmo  $A^*$ , por tratar-se de um método difundido. Para mais informações, recomenda-se o livro de Russell & Norvig (2009).

Conforme mencionado, fez-se uma modificação para que o  $A^*$  calcule as rotas ótimas dentro das áreas seguras. Isto se deve à vantagem de conhecê-las por meio do método de campos vetoriais do capítulo anterior, onde utilizou-se o diagrama de Voronoi. Desta forma, altera-se os valores das células navegáveis de acordo com o diagrama. As células mais próximas do diagrama recebem valores menores, que vão aumentando radialmente conforme seu distanciamento das zonas seguras. O diagrama de Voronoi próximo de obstáculos em um raio de 500 milímetros é retirado por questões de segurança. Nesta pesquisa, utilizou-se a seguinte política

$$C_k = 2C_{k-1} \quad (5.4)$$

em que  $C$  é o valor da célula, e  $k$  é o valor da camada de células cujos custos aumentam ao se distanciar do Voronoi. Este conceito fica exemplificado por meio da Figura 5.3.

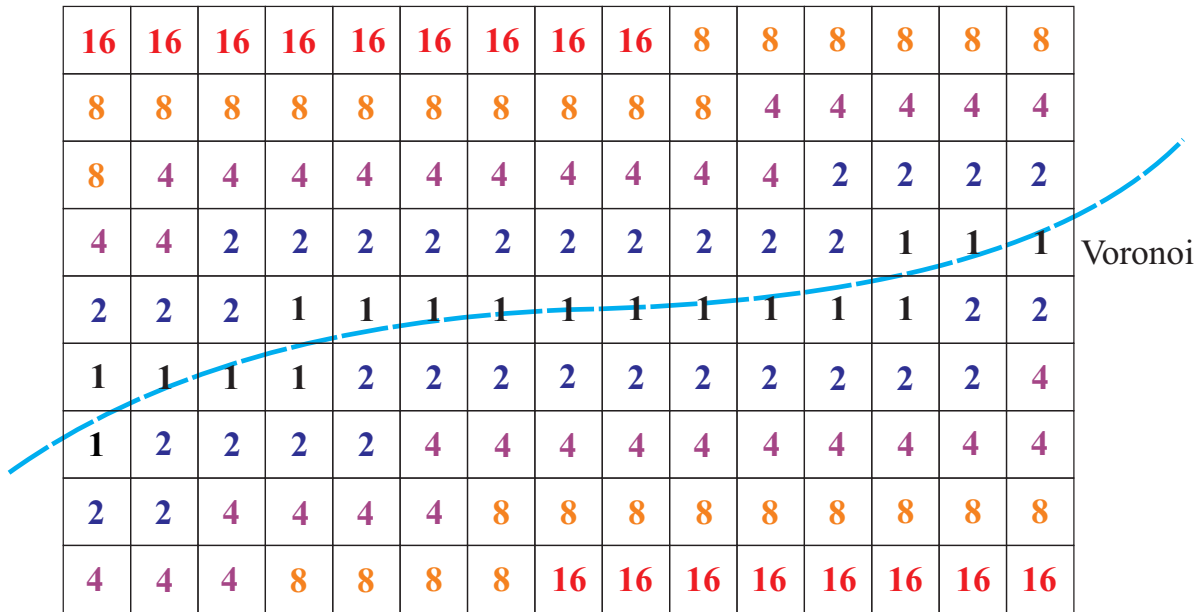


Figura 5.3: Política proposta para o mapa do ambiente.

Com a política da Equação 5.4, os pesos das camadas das células crescem rapidamente na direção radial, tornando as áreas perigosas bastante custosas para o algoritmo. Em ambientes grandes, esses valores de custo podem ficar tão altos que tornam-se impeditivos para a navegação, mantendo o robô o mais distante dos obstáculos. Para inserir este método na função de custo do  $A^*$ , modifica-se a Equação 5.2 da seguinte maneira

$$g = d_{\text{inicio} \rightarrow \text{atual}} + C_{\text{vizinho}} d_{\text{atual} \rightarrow \text{vizinho}} \quad (5.5)$$

Com esta modificação, o  $A^*$  passa a inserir no custo de transitar da célula atual para a célula vizinha um valor diferente da distância métrica percorrida entre as células, como o usual. Dessa forma, força-se o algoritmo a explorar primeiramente as células contidas nas áreas seguras em direção à célula objetivo. O algoritmo somente percorrerá as camadas de células mais próximas das áreas com obstáculos caso seja estritamente necessário, como no caso da célula objetivo estar situada em uma camada de maior custo. Uma vez que a função  $h$  da Equação 5.5 continua sendo otimista (distância Euclidiana), todas as propriedades do método, principalmente a completude, são preservadas.

A modificação apresentada do algoritmo  $A^*$  altera o conceito original de rota ótima. Assim, para o algoritmo, a rota ótima passa a ser a que propicia a navegação segura, ou seja, a menor rota segura possível, contida dentro do diagrama de Voronoi. Assim, todas as rotas passarão, em sua maior parte, pelo diagrama de Voronoi, ou o mais próximo deste, garantindo assim a integridade física do usuário durante a navegação.

### 5.3 Controlador autônomo

O cálculo do caminho a ser seguido pelo robô foi abordado nas seções anteriores. No entanto, para que o robô possa seguir uma determinada rota, é necessário que se tenha um controlador de movimentos. Há uma grande quantidade de controladores relatados na literatura, desde os mais simples, como o liga-e-desliga (*on-off*), os clássicos, como proporcionais (P), e proporcionais-integrais-derivativos (PID), além de outros mais sofisticados, como avanço-atraso de fase, por modelos preditivos (MPC), dentre outros (Siegwart & Nourbakhsh 2004, Ogata 2004).

Todas as abordagens de controle relatadas podem controlar um robô móvel sem dificuldades. De fato, abordagens P e PID são muito comuns na literatura e na indústria, dada sua facilidade de implementação e eficácia no controle. No entanto, é necessário considerar que o robô em questão transporta uma pessoa com deficiência, a qual não pode defender-se ou sequer impedir uma situação desfavorável, como um movimento abrupto do robô.

Dessa forma, o controlador autônomo, além de possuir a capacidade de navegar pelo caminho planejado com acurácia também necessita de uma estratégia que considere os obstáculos transientes que surgem nos ambientes. Os recintos diários possuem diversos obstáculos móveis sobre os quais o sistema robótico não possui controle, tais como pessoas, mobílias leves, portas etc. Caso esses obstáculos não mapeados se moverem em direção ao robô, ou forem colocados no caminho calculado pelo  $A^*$ , a colisão deve ser evitada pelo controlador, especialmente quando o planejador de caminhos não tiver condições de recalculer rotas de desvio a tempo.

Uma alternativa para manter as informações dos obstáculos nas redondezas do robô é realizar o contínuo monitoramento do ambiente por meio de SLAM. Contudo, a manutenção de mapas implica num alto custo computacional, e há a necessidade de que o controle possua uma resposta rápida, especialmente numa situação emergencial, como, por exemplo, uma pessoa que cruza o caminho do robô.

Todas essas necessidades apontadas para o controlador autônomo tornam a tarefa de um controle clássico, do tipo P ou PID, bastante dificultadas. Mesmo controladores mais sofisticados, como os baseados em redes neurais artificiais (Haykin 2001) ou em sistemas nebulosos (*fuzzy*) (Pedrycz & Gomide 2007), conhecidos por sua grande capacidade de generalizar sistemas não-lineares, ofereceriam dificuldades de calibração frente a todas essas restrições.

No entanto, um controlador que atende a todos esses requisitos foi desenvolvido nesta pesquisa. O método de controle compartilhado proposto, conforme discutido no capítulo anterior, possui controles de velocidade angular e linear definidos por campos vetoriais que podem ser calibrados para atender os requisitos de acurácia ao navegar pelo caminho calculado pelo  $A^*$ . Nessa abordagem, um controlador autônomo emula um usuário interagindo com uma HMI ideal, ou seja, que não é afetada por ruídos ou erros de classificação. Desta forma, o robô é direcionado como se estivesse em navegação por controle compartilhado.

Além disso, o campo vetorial repulsivo possui a capacidade de identificar obstáculos não mapeados, pois são independentes de qualquer mapeamento, conforme demonstrado nas Seções 4.1 e 4.2. Sendo assim, o campo pode controlar a velocidade linear do robô e fazê-lo parar frente a um obstáculo não mapeado, protegendo a integridade do usuário. E mais, há ainda a zona de proteção crítica, dada pelo raio crítico  $R_{crit}$ , cuja função é parar o robô imediatamente caso algum obstáculo inesperado se aproxime demasiadamente, sendo uma proteção emergencial.

Os campos vetoriais também podem diminuir o esforço computacional na manutenção do mapa. Conforme discutido, não é necessário preocupar-se com o rastreamento frequente dos obstáculos transientes, pois os campos vetoriais podem se encarregar de evitá-los. Basta, portanto, escolher uma política para incluir esses obstáculos no mapa somente quando forem detectados pelos campos vetoriais, o que diminui o esforço computacional sobre o processador do robô. A política escolhida para essa abordagem será discutida na sequência.

Uma vez que o mapa global e a posição dos pontos de referência (*landmarks*) são conhecidos, não é necessário que se aplique novamente uma técnica de SLAM completa, bastando a aplicação de uma técnica de localização para a correção de odometria do robô. Nesta tese não foi necessário utilizar outra técnica de localização diferente do FastSLAM, bastando utilizar este algoritmo no modo de “correspondências conhecidas”. Dessa forma, utiliza-se apenas a localização, que consiste de diversas partículas ( $N=150$  nesta tese) compostas de Filtros de Kalman Estendidos. Assim, mantêm-se os erros de odometria em níveis aceitáveis com esforço computacional reduzido.

O controlador autônomo possui uma modificação na Equação 4.1, página 43. O robô pode ser controlado por uma velocidade angular ao invés de ângulos de giro, para maior acurácia na navegação do caminho calculado. Contudo, é necessário planejar as velocidades angulares do robô de forma a impedir que movimentos curvilíneos ejetem o usuário do robô. Desta forma, determina-se uma velocidade angular máxima  $\omega_{max}$ . Para esta tese, utilizou-se, por meio de calibração,  $\omega_{max} = 15^\circ/s$ . Por questões de conforto, não é conveniente navegar o robô com velocidades lineares negativas, para que o usuário navegue sempre de maneira frontal. Assim, a velocidade mínima permitida ao robô é zero. A nova lei de controle é dada por

$$\begin{aligned} \text{velocidade linear: } v^{\{t\}} &= \|\vec{v}^{\{t\}}\| & \text{onde } 0 \leq \|\vec{v}\| \leq v_{max} \\ \text{velocidade angular: } \omega^{\{t\}} &= \alpha\phi^{\{t\}} & \text{onde } -\omega_{max} \leq \omega \leq \omega_{max} \end{aligned} \quad (5.6)$$

sendo que  $\phi$  está exibido na Figura 5.4, dado por

$$\phi^{\{t\}} = \text{atan2}(A_y^{\{k\}} - P_y^{\{t\}}, A_x^{\{k\}} - P_x^{\{t\}}) - P_\theta^{\{t\}} \quad (5.7)$$

em que  $A_{x,y}^{\{k\}}$  é o ponto-objetivo,  $\xi$  é uma distância de tolerância que denota chegada ao objetivo e  $\alpha$  é uma constante a ser calibrada. Nesta pesquisa, adotou-se empiricamente  $\xi = 3$  cm.

## 5.4 Análise da estabilidade do controlador por campos vetoriais

Um controlador autônomo deve ser estável, ou seja, deve garantir que a referência desejada seja alcançada. Em robótica móvel, de um modo geral, e no contexto desta pesquisa, a referência desejada é a posição espacial (cartesiana) dos pontos-objetivo dados pelo caminho computado pelo algoritmo  $A^*$ . Ou seja, se o controlador propiciar ao robô alcançar os pontos do caminho, um a um, ao final o robô terá percorrido todo o caminho desejado, alcançando seu destino final.

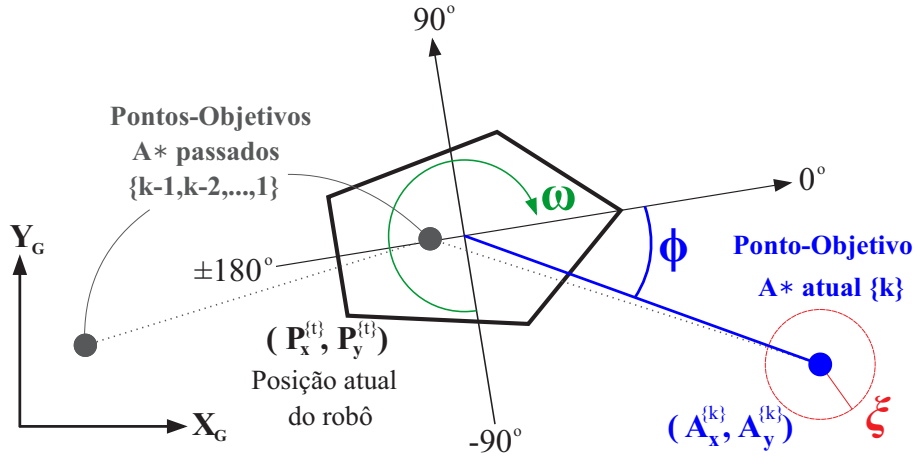


Figura 5.4: Ângulo utilizado na lei de controle do controlador autônomo.

De acordo com Siegwart & Nourbakhsh (2004), o modelo de um robô diferencial é dado por:

$$\begin{bmatrix} \dot{P}_x \\ \dot{P}_y \\ \dot{P}_\theta \end{bmatrix} = \begin{bmatrix} \cos(P_\theta) & 0 \\ \text{sen}(P_\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (5.8)$$

onde  $v$  e  $\omega$  são as ações de controle linear e angular para o robô, e  $P_{x,y,\theta}$  é a pose atual do robô.

Quando se utilizam abordagens angulares como a do campo vetorial da Equação 5.6, é conveniente reescrever o modelo do robô por meio de coordenadas polares. Dessa maneira, obtém-se:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\phi} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} -\cos(\phi) & 0 \\ \frac{\text{sen}(\phi)}{\rho} & -1 \\ \frac{\text{sen}(\phi)}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (5.9)$$

em que  $\rho = \sqrt{\Delta x^2 + \Delta y^2}$ ,  $\Delta x = A_x - P_x$ ,  $\Delta y = A_y - P_y$ ,  $\phi$  é dado na Equação 5.7,  $\gamma = \phi + P_\theta$ . Todos esses dados podem ser visualizados na Figura 5.5, onde ainda destaca-se que  $\lambda = \cos(\phi)\rho$ .

Aplicando-se a lei de controle da Equação 5.6 em 5.9, obtém-se o seguinte sistema em malha fechada

$$\begin{bmatrix} \dot{\rho} \\ \dot{\phi} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} -\alpha \cos(\phi) \|\vec{v}\| \\ \alpha \|\vec{v}\| \frac{\text{sen}(\phi)}{\rho} - \alpha \phi \\ \alpha \|\vec{v}\| \frac{\text{sen}(\phi)}{\rho} \end{bmatrix} \quad (5.10)$$

Conforme se pode verificar, a lei de controle proposta na Equação 5.10 não soluciona a singularidade inerente na Equação 5.9 para a situação  $\rho = 0$ , ou seja, para quando  $P_x = A_x$  e  $P_y = A_y$ , situação esta já bastante conhecida na literatura com relação a controladores de posição (Brockett 1983, Aicardi et al. 1995). Uma maneira de resolver essa situação é acrescentar  $\rho$  à velocidade linear, conforme descreve Siegwart & Nourbakhsh (2004), fazendo-a proporcional à distância entre o robô e o ponto-objetivo. No entanto, esta medida é inconveniente pelo fato de reduzir a velocidade linear do robô conforme este se aproxima do alvo, e por essa razão foi descartada nesta tese. No entanto, o fato da singularidade em  $\rho = 0$  não impede o controlador de possuir estabilidade com relação ao controle de posição (Brockett 1983, Aicardi et al. 1995).



Sendo assim, para resolver esta situação, utilizou-se um círculo de raio  $\xi$ , conforme mostrado nas Figuras 5.4 e 5.5. É possível fazer  $\xi > 0$  tão pequeno quanto se queira e, dessa forma, possibilitar ao robô aproximar-se do ponto-objetivo desejado o tão próximo quanto seja necessário para satisfazer os requisitos de proximidade de uma navegação real.

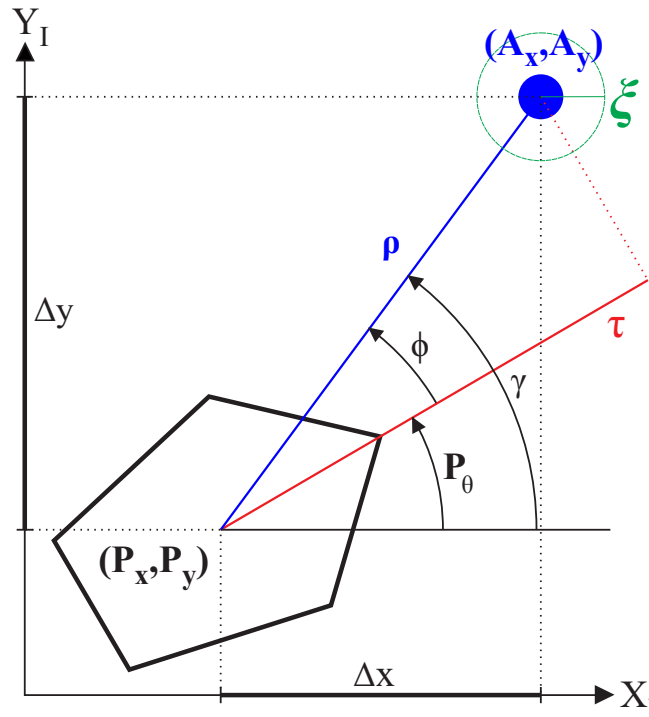


Figura 5.5: Transformação polar para o modelo do robô diferencial.

Uma vez que o problema da singularidade fica resolvido pelo círculo de raio  $\xi$ , resta ao controlador garantir a minimização da distância  $\rho$  ao longo do tempo, uma vez que, conforme a Figura 5.5, o problema de controle de posição será resolvido se  $\rho \rightarrow \xi$  quando  $t \rightarrow \infty$  (Brockett 1983, Aicardi et al. 1995, Lizarralde 1998, Vieira 2005).

De acordo com a Figura 5.5, conforme citado anteriormente, tem-se que

$$\rho = \frac{\tau}{\cos(\phi)} \quad (5.11)$$

ou seja,  $\rho \rightarrow \xi$  implica diretamente em  $\tau \rightarrow \xi$ . Nesse sentido,  $\tau \rightarrow \xi$  acontecerá quando a velocidade linear, positiva ou negativa, estiver na direção do objetivo, e,  $\phi \neq \pm 90^\circ$ . Em uma situação de pior caso, onde o ângulo  $\phi$  permanece muito próximo a  $\pm 90^\circ$ , o robô será levado ao ponto-objetivo por meio de um comportamento espiral. No entanto, independentemente do caminho, ainda assim, garante-se que  $\rho \rightarrow \xi$  quando  $t \rightarrow \infty$ , fato que garante, portanto, a estabilidade do controlador (Aicardi et al. 1995, Lizarralde 1998, Vieira 2005).

O fato relatado anteriormente leva em consideração que o robô possa assumir velocidades lineares negativas. No caso específico do controlador por campos vetoriais, tem-se que  $v \geq 0$ , conforme relatado na Equação 5.6, para que o usuário nunca navegue de costas por questão de conforto, lembrando-se que a velocidade linear nula somente acontece na presença de obstáculos obstruindo a passagem do robô.

Tendo em vista a condição extra dos campos vetoriais descrita no parágrafo anterior, e considerando as condições para garantir a convergência do robô ao ponto-objetivo, conclui-se, portanto, ser necessário e suficiente que  $\phi \in (-90^\circ, 90^\circ)$ . Conseqüentemente, a convergência global para  $\xi$  estará garantida para um controlador angular que faça  $\phi \rightarrow 0$ . E, para cumprir essa condição, é necessário que a velocidade angular  $\omega$  do robô seja diretamente proporcional a  $\phi$  (Aicardi et al. 1995, Lizarralde 1998, Vieira 2005).

Com base nas conclusões mostradas ao longo da seção, a lei de controle exibida nas Equações 5.6 e 5.7 garantem, portanto, a convergência do robô ao ponto-objetivo, fazendo do controlador autônomo por campos vetoriais globalmente estável, garantindo, portanto, que o ponto objetivo estipulado para a navegação autônoma será efetivamente atingido. Por fim, o esquema de controle desenvolvido para a navegação autônoma pode ser observado na Figura 5.6.

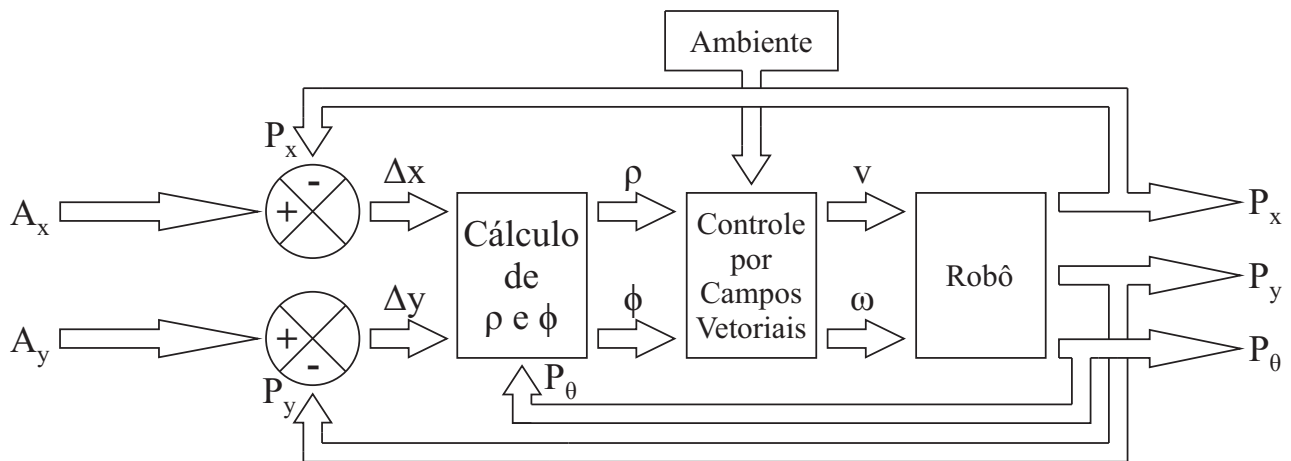


Figura 5.6: Controle autônomo de posição por campos vetoriais.

## 5.5 Manutenção dos mapas e recálculo de rotas

Os ambientes por onde o robô se desloca são dinâmicos, com diversos obstáculos móveis, tais como pessoas, ou que tiveram suas posições alteradas desde o mapeamento, tais como mobília. Portanto, o mapa original não se manterá estático, havendo diferenças a serem ajustadas. Para atualizar o mapa, é preciso uma política de verificação da posição dos obstáculos, que depende de sensores apropriados. Nesta pesquisa, conforme apresentado anteriormente, utilizaram-se sensores de distância do tipo *laser* SICK LSM-100/200. A plataforma robótica de Cardozo et al. (2010), utilizada para a comunicação sem-fio com o robô, permite a aquisição dos valores de todas as distâncias fornecidas pelos sensores a uma taxa de aproximadamente 18 *Hz*.

Estando o robô em uma dada posição  $P_{x,y,\theta}$ , é possível reunir as informações a respeito do ambiente real por meio dos sensores. Assim, obtêm-se as configurações espaciais reais dos obstáculos no momento atual, que podem estar, inclusive, sobre o caminho calculado para o robô. Seria plausível, portanto, fazer uma comparação entre a situação apresentada pelo mapa e pelos sensores para concluir sobre possíveis diferenças e, em caso afirmativo, realizar as devidas alterações no mapa e no caminho do robô.

No entanto, ao invés de proceder conforme o relatado acima, permite-se que o robô continue navegando no caminho original, mesmo que este contenha obstáculos não previstos. Ao encontrar obstáculos não mapeados, o campo vetorial repulsivo diminuirá naturalmente a velocidade do robô de acordo com sua proximidade. Se o obstáculo for transitivo, o robô voltará a navegar com velocidades maiores assim que o caminho for liberado. Para um obstáculo permanente, o controlador irá proteger o robô, fazendo com que este pare diante da obstrução não prevista e não prossiga em seu caminho original.

A escolha pelo procedimento relatado se dá em razão do gasto de esforço computacional extra que faz a verificação entre sensores e mapa demanda, o qual pode ser direcionado para a localização e controlador. Assim sendo, a correção do mapa, com a adição de novos obstáculos se dará somente quando a velocidade linear do robô chegar a zero, denotando que um novo obstáculo permanente foi adicionado ao ambiente. A Figura 5.7 exibe a situação relatada, sendo que 5.7(a) exibe a diferença entre as situações do mapa e as medidas recebidas dos sensores. Dada a necessidade de correção do mapa, altera-se o estado da célula, conforme a Figura 5.7(b).

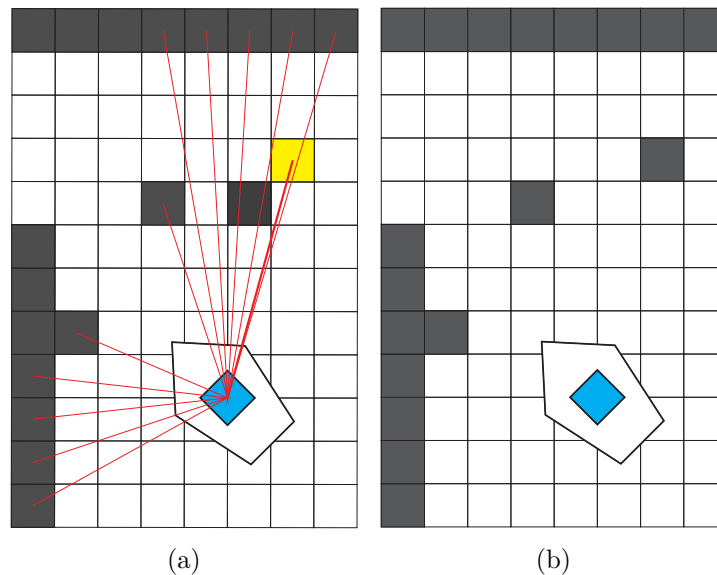


Figura 5.7: (a) Detecção de anormalidade nas células e (b) estado das células alterado.

A situação relatada mostra, portanto, que o robô foi impedido de navegar pelo caminho originalmente calculado por um obstáculo não previsto no mapa original. Com a velocidade zerada, não há outra alternativa a não ser remodelar o ambiente, para que uma nova rota seja calculada. No entanto, sempre que o ambiente é remodelado, as zonas perigosas e seguras do ambiente local são alteradas. Conforme descrito na Seção 5.2, o caminho  $A^*$  é calculado sempre levando em consideração as zonas seguras.

Portanto, a obtenção de um novo caminho depende diretamente da atualização do diagrama de Voronoi do ambiente. Assim, a partir da célula que se encontra no ponto médio das células que tiveram seu estado alterado para ocupadas/livres, invoca-se o algoritmo de Voronoi, numa região limitada por um raio de 10 células (2 metros neste trabalho), para efetuar a atualização do diagrama na área em questão. Este raio de alteração foi escolhido empiricamente, o qual mostrou-se suficiente, dadas as dimensões dos robôs.

Uma vez que o procedimento de atualização do diagrama de Voronoi esteja concluído, é necessário recalcular um caminho alternativo para o robô, pois o obstáculo recém adicionado impede o robô de percorrer a rota pré-estabelecida. Caminhos  $A^*$  são sequências de células que, uma a uma, devem ser seguidas para que o robô atinja seu ponto-objetivo. Na situação descrita, reporta-se ao  $A^*$  a nova posição inicial do caminho (a posição atual do robô), e a nova distribuição de pesos locais das células, calculados a partir do diagrama de Voronoi.

Assim, a nova rota é imediatamente calculada e o robô passa a utilizá-la, abandonando a antiga. A Figura 5.8 exhibe esta abordagem, a qual foi utilizada em Olivi et al. (2012) e Teixeira et al. (2012), além desta pesquisa e dos trabalhos relativos ao projeto DesTINE. Uma vez que se possibilita a atualização imediata (*online*), tanto do mapa quanto dos caminhos do robô, evolui-se o  $A^*$  para sua variante dinâmica, o  $D^*$ , acrônimo de *Dynamic A\** (Stentz 1994).

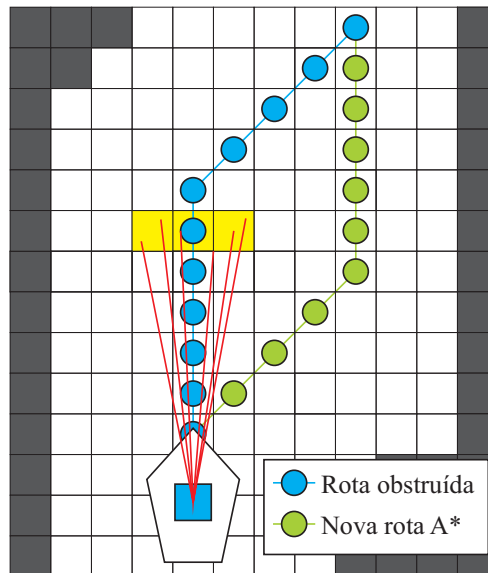


Figura 5.8: Recálculo de rotas obstruídas.

## 5.6 Discussões

Apresentou-se neste capítulo a estratégia de controle autônomo utilizada para transportar um robô assistivo a um ponto-objetivo designado pelo usuário. Este ponto é escolhido por meio de uma interface humano-máquina assistiva, conforme mostrado na Seção 3.3, página 31. A partir disso, vários algoritmos são empregados para que a tarefa seja concluída com sucesso. Um dado fundamental para que se desenvolva as atividades é o mapa, obtido durante a navegação assistida, descrita no Capítulo 4, com o auxílio do algoritmo FastSLAM (Seção 4.4).

Uma vez que o mapa métrico do ambiente estava disponível, planejou-se o modo de armazená-lo, por meio de células quadradas de lado  $L=200\text{ mm}$ , conforme mencionado na Seção 5.1, página 58. Esse tipo de mapa em grades facilita a utilização de algoritmos de busca discretos, dentre os quais, escolheu-se o algoritmo  $A^*$  por suas propriedades de completude e rapidez computacional. Para o  $A^*$  desenvolveu-se uma alteração na heurística de busca, de forma a produzir rotas ótimas que maximizam a segurança do usuário, conforme descrito na Seção 5.2.

Para que o robô pudesse navegar pelo ambiente seguindo o caminho  $A^*$ , desenvolveu-se um controlador para direcionar o robô pelos caminhos calculados. Para esta tese, utilizou-se a própria técnica de controle compartilhado proposta. A razão de se escolher os campos vetoriais para o controlador autônomo reside na facilidade de moldar comportamentos para o robô por meio dos campos vetoriais, os quais permitem a concepção de estratégias de controle altamente não-lineares com relativa facilidade, permitindo melhor ajuste e adaptação que as técnicas clássicas. O projeto deste controlador encontra-se relatado na Seção 5.3.

A proteção emergencial, no caso de obstáculos não previstos, falhas e atrasos na execução de comandos, já está prevista pelo controlador por meio do raio crítico de proteção ( $R_{crit}$ ), e deve parar o robô frente a colisões iminentes todas as vezes que este raio de proteção for violado.

Mesmo um eficiente algoritmo de busca como o  $A^*$  depende de mapas bem estruturados e atualizados para fornecer soluções coerentes. Com este objetivo, desenvolveu-se uma estratégia para realizar a manutenção do mapa, principalmente no quesito de adicionar obstáculos não previstos, conforme descrita na Seção 5.5. Essa heurística está baseada no tipo de sensor empregado, no caso, sensores de distâncias por *laser*.

A manutenção de mapas por meio do controlador autônomo resolve o problema da dinamicidade dos ambientes, e da necessidade de implementar outras técnicas paralelas para esta finalidade. No entanto, as rotas obtidas pelo algoritmo  $A^*$  também devem ser atualizadas quando novos obstáculos forem adicionados ao ambiente. Para isto, invoca-se o cálculo do diagrama de Voronoi nas redondezas dos novos obstáculos e, assim, recalculam-se as rotas de forma a evitar colisões, conforme descrito na Seção 5.5. Com essas estratégias implementadas, evoluiu-se o  $A^*$  para sua variante dinâmica, o  $D^*$ .

Os resultados referentes a este capítulo serão exibidos no Capítulo 6, mostrando que tais políticas de manutenção do mapa, cálculo e atualização de rotas mostraram-se eficientes, produzindo resultados coerentes com os objetivos principais, notadamente a maximização da segurança. Por meio de todos os experimentos de navegação autônoma realizados, pode-se afirmar que o método de controle autônomo por campos vetoriais obteve desempenho satisfatório, especialmente quando comparado com os métodos até então citados na literatura.



## Avaliação Experimental

Este trabalho apresentou, nos Capítulos 3, 4 e 5, soluções para o controle de um veículo assistivo cujo usuário possui severas deficiências físicas. Uma vez impossibilitada de comunicação convencional, por meio da operação de controles manuais (*joysticks*) ou teclados, a pessoa ainda dispõe de sinais biológicos que podem traduzir seus desígnios e necessidades.

Para que essa comunicação assistiva seja possível, dispõe-se de uma interface de comunicação entre a pessoa e a máquina especialmente adaptada para a utilização de sinais biológicos. Esta interface foi descrita no Capítulo 3, na qual uma HMI assistiva foi desenvolvida para o projeto DesTINe, mostrando a possibilidade da utilização de sinais de eletromiografia (sEMG) e eletroencefalografia (EEG) para a extração de comandos do usuário em aplicações na área da robótica móvel assistiva.

A partir da obtenção dos comandos advindos do usuário via HMI, é possível controlar um robô móvel diretamente. No entanto, a aquisição e classificação dos comandos adquiridos do usuário está sujeita a erros. Para minimizá-los, bem como proteger o usuário, uma técnica de controle compartilhado é utilizada.

O Capítulo 4 mostrou um novo tipo de controle compartilhado, baseado na utilização de campos vetoriais. Nesta abordagem, constroem-se comportamentos para o robô por meio de vetores de controle, os quais podem ser manipulados por campos vetoriais computados a partir do sensoriamento do ambiente com os dispositivos do robô ou distribuídos pelo próprio ambiente. Dessa forma, o robô pode moldar seu comportamento de acordo com as situações impostas pelo ambiente, situações estas determinadas por meio dos campos vetoriais.

O Capítulo 5 apresentou a abordagem autônoma desta pesquisa, uma vez que o uso prolongado de tecnologias assistivas é conhecidamente cansativo. Nesta abordagem os diversos aspectos concernentes à navegação autônoma por ambientes conhecidos foram equacionados. A manutenção dos mapas de ambientes dinâmicos, o cálculo das rotas entre a posição atual do robô e o ponto-objetivo selecionado pelo usuário, além do controle do robô pelo caminho calculado foram apresentados neste capítulo.

Os experimentos planejados para validar as técnicas desenvolvidas e integradas nos Capítulos 4 e 5 serão apresentados neste capítulo, bem como a análise dos resultados obtidos de maneira qualitativa e quantitativa.

## 6.1 Experimentos de navegação assistida

Esta seção apresenta os experimentos conduzidos para validar o método de controle compartilhado baseado em campos vetoriais. Designou-se um experimento para avaliar os campos vetoriais utilizados no controle da velocidade linear (Seção 4.2) e no controle do ângulo de giro (Seção 4.3), bem como a avaliação conjunta destes campos. Avalia-se também o mapeamento obtido pelo FastSLAM, essencial para a navegação autônoma descrita no Capítulo 5.

Para a avaliação do método de navegação assistida proposto, uma política de pior caso foi adotada para o controle do robô. Por esta razão, utilizou-se o controle via sEMG, com o sensor Emotiv, em virtude de ser o classificador que obteve o pior desempenho de classificação. Experimentos com simulação também foram concebidos para avaliação de erros com independência do desempenho de usuários.

A comunicação entre os robôs e o computador do usuário é sem-fio, e utiliza a plataforma robótica de Cardozo et al. (2010). Todos os algoritmos foram desenvolvidos por meio da plataforma MATLAB (MathWorks 2014). A Tabela 6.1 mostra todos os valores padrão (*default*) e parâmetros do método utilizados nos experimentos que utilizam o equipamento de sEMG. Estes valores, conforme descrito na Seção 4.6, foram obtidos por meio de calibração.

Tabela 6.1: Valores dos parâmetros usados nos experimentos com robôs reais.

Parâmetro	Valor	Parâmetro	Valor
$\ \vec{v}^{(d)}\ $	100 [mm/s]	$\theta_l^{(d)}$	30°
$\ \vec{s}_l^{(d)}\ $	100	$\theta_r^{(d)}$	30°
$\ \vec{s}_r^{(d)}\ $	100	$R^{(d)}$	1600 [mm]
$\alpha$	0,02	$\beta$	$\eta$

### 6.1.1 Navegação por um corredor estreito

O primeiro experimento com sEMG trata da navegação do robô por um corredor estreito com uma passagem de porta no início. Este experimento foi especificamente projetado para avaliar a estratégia de controle da velocidade linear descrito na Seção 4.2, página 44. O comportamento esperado é que ao robô seja permitido navegar nestes ambientes, onde o robô utilizará velocidades menores, evitando quaisquer colisões. Na avaliação desta seção utilizou-se o robô Pioneer P3-DX. O corredor e o robô podem ser observados na Figura 6.1.

Conforme se pode observar, o corredor oferece uma possibilidade de giro à esquerda e duas à direita. Utilizou-se esta configuração, novamente em decorrência da política de pior caso, dada a baixa taxa de acerto proporcionada pelo comando “girar à direita”, conforme mostrado na Tabela 3.1. O corredor possui 92 centímetros de largura, e pode-se observar que, em proporção ao tamanho do robô, mostra-se uma passagem bastante estreita, oferecendo grandes dificuldades de navegação para uma estratégia de controle compartilhado com interface baseada em sEMG.

Ao iniciar-se a navegação, o robô adentra o corredor enquanto o campo vetorial repulsivo o desacelera paulatinamente até uma velocidade abaixo de 50 mm/s. Nessa velocidade é possível que o usuário observe o ambiente e tome suas decisões com cautela.





Figura 6.1: Navegação por um corredor estreito.

Mesmo as baixas taxas de classificação do Emotiv na escolha dos comandos à esquerda e à direita não são decisivas, uma vez que o robô interrompe sua navegação em face a um obstáculo. Dessa maneira, é possível que o usuário empregue seu esforço em escolher os comandos, sem se preocupar com sua integridade física. Os resultados quantitativos e qualitativos encontram-se, respectivamente, na Tabela 6.2 e na Figura 6.2.

Tabela 6.2: Resultados quantitativos para a navegação no corredor estreito.

Parâmetro	Valor	Parâmetro	Valor
$v_{medio}$	24,12 [mm/s]	$n^{\circ}$ de referências	4
$\Delta S$	5287,13 [mm]	$n^{\circ}$ ações	20
$\Delta t$	204,86 [s]	$n^{\circ}$ giros esq.	3
$\beta$	0,42	$n^{\circ}$ giros dir.	6

Conforme a Figura 6.2, o comportamento geral descrito nas seções anteriores foi atingido, sendo que o veículo necessitou navegar paralelamente às paredes para obter velocidade linear positiva. Sempre que o robô aproximava demasiadamente sua parte frontal de uma parede, sua velocidade linear decrescia até zero. O robô navegou por 5,28 metros em 3 minutos e 24 segundos, uma navegação lenta, com média de 24,12 mm/s, o que confirma o comportamento pretendido e valida o resultado. O método obteve êxito em proteger o robô, evitando colisões com sucesso. O FastSLAM comportou-se adequadamente, encontrando os 4 pontos de referência corretamente. O mapa mostrado foi produzido por células de 20 cm de lado.

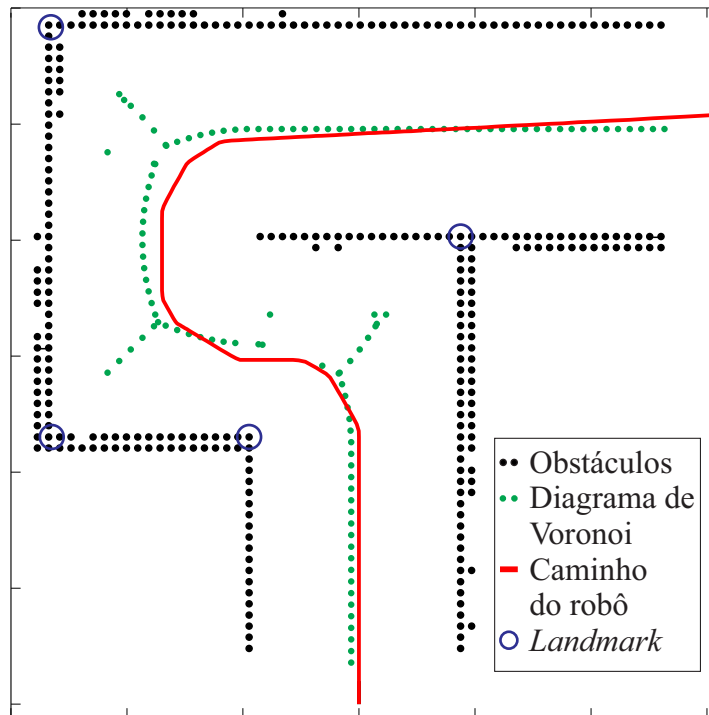


Figura 6.2: Resultados qualitativos da navegação com campo repulsivo.

Dos trabalhos relatados na Seção 2.3, o que mais se alinha com os resultados alcançados nesse experimento é o de Tonin et al. (2010), que produziu um controlador para o usuário que utiliza os mesmos ângulos de giro exibidos na Tabela 6.2, página 73. Os resultados quantitativos exibidos na Figura 3 de Tonin et al. (2010), página 1464, mostram um ambiente contendo algumas salas e um corredor estreito, com passagens de portas. Os autores estabelecem a posição do robô (R), alguns pontos-objetivos (T1, T2, T3 e T4) e caminhos possíveis (*possible paths*) conectando esses pontos objetivos. Tais caminhos visam evitar obstáculos que venham dificultar a navegação do usuário.

Os caminhos possíveis mostrados na Figura 3 de Tonin et al. (2010) possuem situações similares às desta seção, como o da posição R até T1, denominado como o caminho 1 (*path 1*), o qual é um corredor estreito com passagem de porta. Não há dados quantitativos fornecidos especificamente em relação aos desempenhos do robô pelos caminhos exibidos e, por essa razão, não é possível comparar numericamente os métodos.

No entanto, por meio da Figura 3 de Tonin et al. (2010) é possível visualizar, qualitativamente, que a segurança é severamente comprometida, uma vez que o robô se aproxima demasiadamente dos obstáculos repetidas vezes por todos os caminhos percorridos. Dessa forma, conclui-se que a abordagem por campos vetoriais apresentada nesta seção possui melhores desempenhos em relação à segurança sem necessitar de quaisquer ações autônomas.

Os resultados numéricos disponibilizados pelo citado autor em sua Figura 4, página 1465, são referentes ao tempo total para completar os caminhos, onde, para o caminho 1 comentado anteriormente, o tempo relatado teve média em torno de 5 minutos, utilizando sua estratégia de controle compartilhado. Ainda assim, a distância dos caminhos não foi descrita pelos autores, não sendo possível concluir acerca das velocidades desenvolvidas pelo robô.

### 6.1.2 Navegação pelo diagrama de Voronoi

Neste tópico apresenta-se a avaliação do campo vetorial relatado na Seção 4.3 com um usuário utilizando o equipamento de sEMG. O comportamento desejado é que os ângulos de giro do robô tenham preferência por zonas seguras, ou seja, pelo diagrama de Voronoi, facilitando o acesso a estas áreas e maximizando a velocidade de navegação.

Para essa seção, impõe-se uma tarefa ao usuário, que deve alcançar uma posição específica do mapa por meio do diagrama, isto é, mantendo o robô o máximo possível sobre a referência exibida. Trata-se de uma tarefa de rastreamento em que o campo vetorial deverá facilitar a atividade do usuário. O mapa com resultados qualitativos pode ser visualizado na Figura 6.3, enquanto os resultados quantitativos estão dispostos na Tabela 6.3.

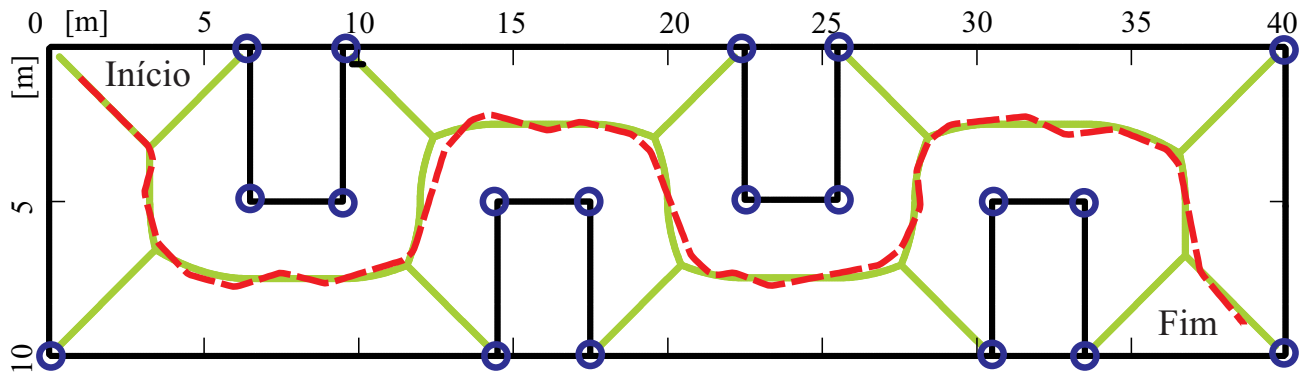


Figura 6.3: Resultados para o rastreamento do diagrama de Voronoi.

Tabela 6.3: Resultados para a navegação pelo diagrama de Voronoi.

Parâmetro	Valor	Parâmetro	Valor
$v_{medio}$	96,44 [mm/s]	$n^{\circ}$ de referências	19
$\Delta S$	55556,50 [mm]	$n^{\circ}$ ações	70
$\Delta t$	534,02 [s]	$n^{\circ}$ giros esq.	17
$\beta$	0,6	$n^{\circ}$ giros dir.	17
$\Delta E_{vf}$	150,39 [mm]	$\Delta E_{w/o}$	237,82 [mm]
$IC_{90\%}$	21,48 [mm]	$\sigma_{\Delta E}$	106,68 [mm]

Ao usuário é disponibilizado o mapa do ambiente com o diagrama de Voronoi, por meio do simulador MobileSIM (MobileRobots 2014). Trata-se de um mapa relativamente grande, com  $400 \text{ m}^2$ . O usuário inicia sua navegação em um dos cantos e seu objetivo é atingir o canto extremo oposto. A Figura 6.3 exibe o desempenho do usuário no rastreamento.

O diagrama de Voronoi é representado pelas linhas sólidas, enquanto o caminho percorrido pelo robô é mostrado pela linha tracejada. Percebe-se que a navegação do usuário esteve próxima do diagrama de Voronoi durante a maior parte do trajeto. Este resultado reflete-se na alta velocidade média do experimento, que manteve-se em  $96,44 \text{ mm/s}$ , muito próxima da velocidade máxima, de  $100 \text{ mm/s}$ . Nesse experimento o robô navegou por uma distância de  $55,55$  metros em  $8$  minutos e  $54$  segundos.

O FastSLAM produziu bons resultados, onde o mapa exibido na Figura 6.3 possui as referências obtidas pelo método. Ao todo, existem 20 referências no ambiente (cantos de parede), dos quais o FastSLAM identificou corretamente 19, representados pelos círculos na figura em questão. A única referência não identificada não estava disponível ao alcance do sensor *laser* do robô, uma vez que encontrava-se na parte traseira do veículo no início da navegação, e sua região nunca foi alcançada pelo sensor *laser* durante o trajeto. Dessa forma, esta referência jamais entrou na área de identificação, razão pela qual não foi identificada.

O número de comandos de giro dados pelo usuário mostrou-se balanceado, conforme mostra a Tabela 6.3, com 17 comandos para cada direção. Esse fato decorreu do caminho possuir certos padrões repetitivos, moldados pelo próprio ambiente, o qual possui uma estrutura cadenciada, fato que levou a esta simetria de comandos.

Considerando os resultados expostos na Figura 6.3 e Tabela 6.3, conclui-se que a estratégia de campos vetoriais atrativos às áreas seguras do ambiente tiveram uma participação fundamental no bom desempenho da navegação do usuário. Assim, confirma-se que os objetivos pretendidos nessa seção foram alcançados, onde o controle compartilhado obteve desempenho satisfatório.

No que concerne aos objetivos expostos nesta seção, o trabalho que mais se aproxima dentre aqueles relatados na Seção 2.3 é o de Philips et al. (2007). De fato, a escolha do ambiente mostrado na Figura 6.3 é proposital para a comparação entre os resultados do controle compartilhado. O controle compartilhado do trabalho mencionado possui três ações, sendo que uma delas é a de recuperação de orientação, a qual serve os mesmos propósitos do campo vetorial atrativo. O objetivo de Philips et al. (2007) é partir de um extremo e chegar ao outro do ambiente, não importando o caminho percorrido. Conforme descrito no início dessa seção, ao controle compartilhado por campos vetoriais adicionou-se a tarefa de rastreamento do diagrama de Voronoi. Essa tarefa não está incluída no trabalho de Philips et al. (2007).

Os principais resultados mostrados em Philips et al. (2007) são qualitativos, dispostos nas Figuras 4, 5 e 6 desta referência, que mostram os trajetos do robô conduzido pelo usuário pelo ambiente. A Figura 4, página 411, dedica-se a mostrar o ambiente, que é praticamente o mesmo exibido na Figura 6.3, sendo o ambiente de Philips et al. (2007) um pouco menor em comprimento, com a mesma largura. Nessa figura exibe-se o melhor desempenho obtido pelo seu controle compartilhado. Ainda assim, seu trajeto aproximou-se demasiadamente das paredes, sendo um caminho considerado perigoso para o usuário. A Figura 5, página 411, mostra os resultados com o controle compartilhado desabilitado, onde os trajetos são praticamente aleatórios, mostrando que seu controle compartilhado possui efeito positivo. Na Figura 6, página 412, tem-se outro resultado para o controle compartilhado, sendo este, no entanto, pior quando comparado com o apresentado em sua Figura 4.

É possível constatar pelos resultados mostrados em Philips et al. (2007) que a ação de recuperação de orientação possui grande influência sobre o trajeto do robô. Apesar de existir a preocupação com a segurança, percebe-se que os trajetos passam próximos a obstáculos com frequência, indo inclusive de encontro às paredes onde claramente percebe-se que esta não é a intenção do usuário. Desta forma, para todos os resultados apresentados por Philips et al. (2007), conclui-se que os resultados apresentados nessa seção para os campos vetoriais são bastante superiores em termos da regularidade e suavidade do caminho obtido.

Quantitativamente, a performance do usuário com o sistema de BCI relatada, de 58%, são parecidas com o desempenho do Emotiv para os comandos de giro. Com relação à distância navegada, Philips et al. (2007) relata que o melhor resultado apresentado fica por conta do sujeito 2, com 60,67 metros, contra 69,23 metros do sujeito 1, em média. Nos campos vetoriais, os caminhos a serem percorridos pelo usuário são livres, porém, deseja-se que haja uma predileção pelos caminhos mais seguros, para que as velocidades do robô sejam maiores.

Nessa seção, impôs-se que o robô seguisse o diagrama de Voronoi que, por ser o caminho mais seguro, é naturalmente um caminho bastante longo. Ainda assim, os resultados alcançados com relação a distância navegada é de 55,55 metros, sendo um resultado superior ao do trabalho comparado, pois o controle compartilhado obteve melhor desempenho em ajudar o usuário a seguir pelo caminho designado.

### 6.1.3 Ganho de desempenho na utilização do campo atrativo

Nesta seção utiliza-se uma abordagem autônoma para comparar o desempenho do controle por campos vetoriais com relação à estratégia da manipulação dos ângulos de giro. A abordagem escolhida exime os resultados de erros cometidos pelo usuário ou pelo aparelho de sEMG/BCI, isto é, em acompanhamento ao controle compartilhado está um usuário ideal, no que tange a emissão de comandos 100% corretos. Assim, é possível avaliar de forma sistemática o ganho real do método sem a influência de um usuário real.

Uma vez que o robô encontra-se sobre o diagrama de Voronoi, como mostrado na Figura 6.4(a), o vetor atrativo do campo atrativo  $\vec{n}$  fica alinhado com o eixo longitudinal do veículo, o que produz ângulos de giro menores que os ângulos pré-definidos. Quando o usuário encontra-se distante do diagrama, os ângulos de giro produzidos propiciam o retorno ao diagrama em detrimento das zonas perigosas, conforme a Figura 6.4(b) exemplifica. Este comportamento permite que o usuário mantenha-se sobre o diagrama (ou retorne a ele) mais facilmente.

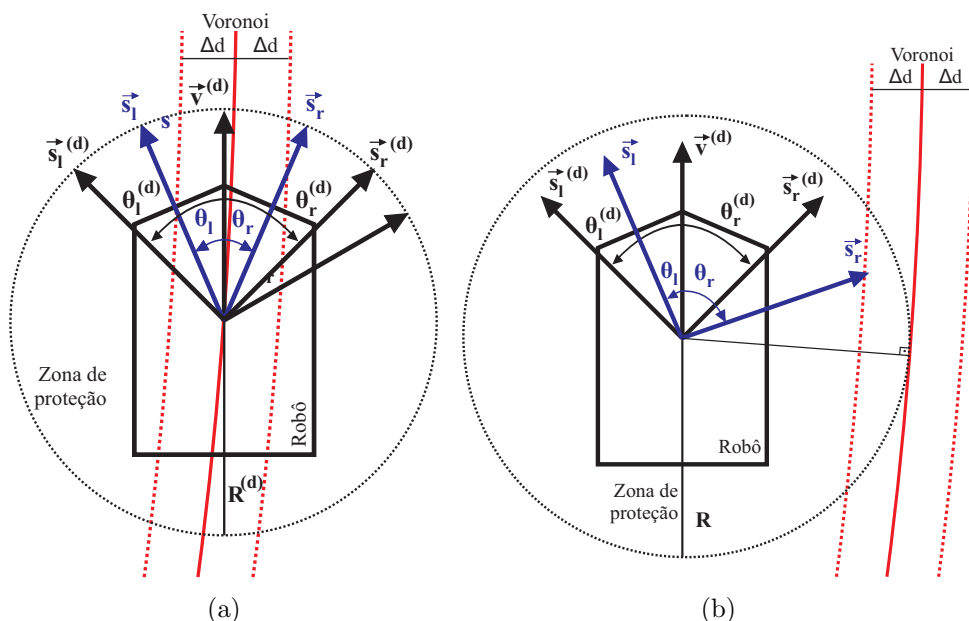


Figura 6.4: Desvio reduzido pelo campo vetorial atrativo (a) perto e (b) longe do diagrama.

Para essa avaliação, somente o campo vetorial atrativo será considerado, uma vez que o campo que controla a velocidade linear teve sua eficácia comprovada no experimento da Seção 6.1.1. Nessa abordagem, impôs-se uma distância limite  $\Delta d$  de 100 milímetros do robô ao diagrama de Voronoi, de acordo com o exposto na Figura 6.4. Uma vez ultrapassado este limite, o robô gira automaticamente em direção ao diagrama caso estiver se afastando deste, e gira para se alinhar com o diagrama, caso estiver se aproximando deste.

Esse comportamento foi imposto por ser o comportamento natural de um usuário conduzindo o robô executando o mesmo objetivo. A restrição da distância  $\Delta d$  foi necessária para garantir uma zona de ação independente de um usuário humano, para que a estratégia seja avaliada de forma sistemática, conforme mencionado anteriormente. Nesse experimento, o mesmo cenário da seção anterior será utilizado, no qual o robô sempre parte da posição  $[1000 \text{ mm}, 1000 \text{ mm}, 90^\circ]^T$  e deve alcançar a região retangular iniciada no ponto  $[9000, 39000]^T \text{ mm}$ . Os resultados do trajeto obtido pelo robô estão exibidos na Figura 6.5.

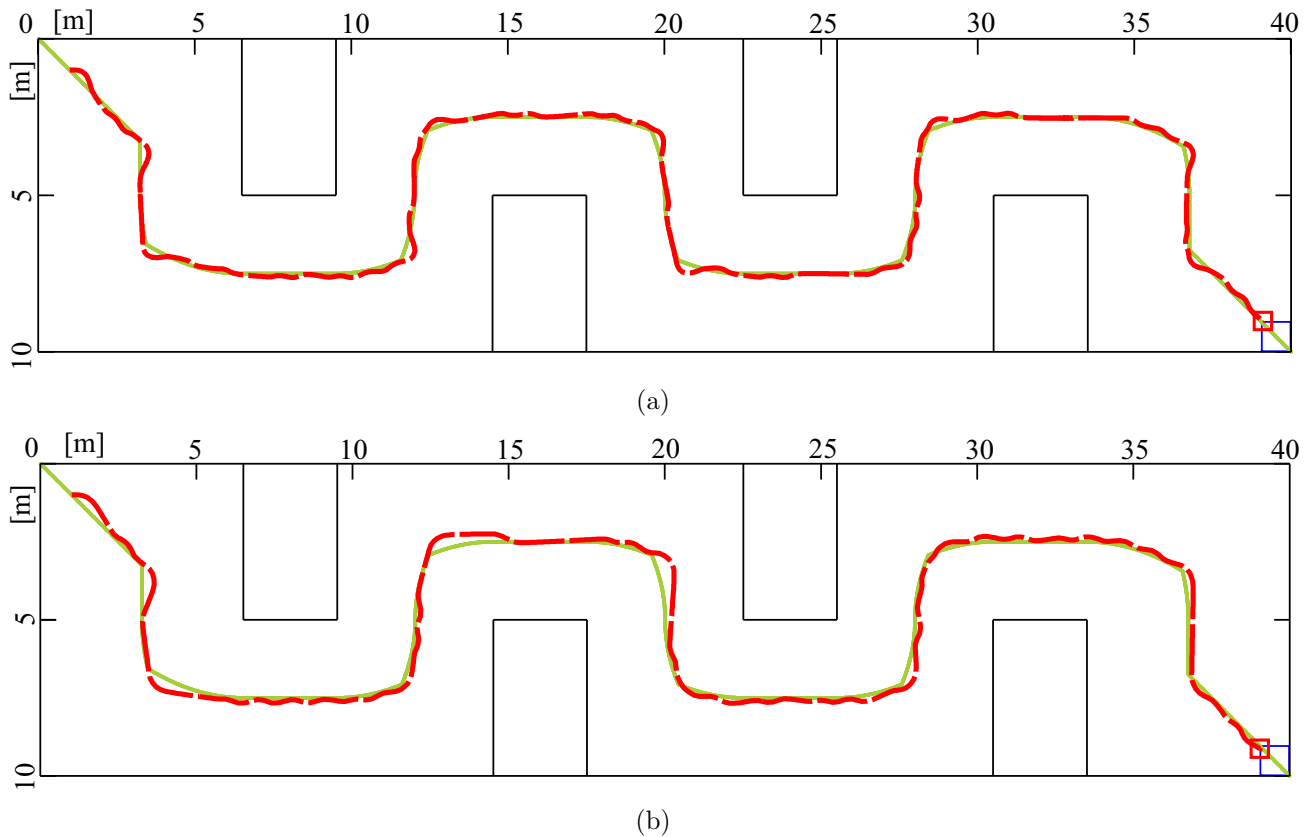


Figura 6.5: Resultados do experimento (a) com e (b) sem o campo vetorial atrativo.

Conforme é possível notar na Figura 6.5, o caminho pretendido é o próprio diagrama de Voronoi, o qual foi ligeiramente modificado, onde todas as partes que não fazem parte do caminho central foram excluídas. Isso foi necessário para evitar que o comportamento autônomo utilizado tomasse um caminho diferente daquele que não levasse à região-objetivo. Em ambos os casos, o robô apresentou um bom comportamento, conseguindo percorrer o caminho do início (canto superior esquerdo) até a região objetivo (canto inferior direito), mantendo-se bastante próximo do diagrama de Voronoi por todo o percurso.

O ambiente utilizado foi inicialmente planejado para ser o mesmo do utilizado no trabalho de Philips et al. (2007), para comparações entre os métodos. Por mostrar-se um bom ambiente para avaliações, em razão de suas propriedades repetitivas, foi utilizado no experimento desta seção. Os resultados quantitativos concernentes aos trajetos dos robôs estão contidos na Figura 6.6 e na Tabela 6.4.

A velocidade máxima permitida para a navegação do robô foi de  $200 \text{ mm/s}$ , pois velocidades muito acima desse patamar não são comuns em tecnologias assistivas. Para a manipulação dos ângulos de giro, utilizou-se um parâmetro  $\alpha$  no valor de 0,3 para o campo vetorial atrativo. Assim, os ângulos de giro para as áreas perigosas tornam-se bastante diminutos quando o robô se afasta do diagrama de Voronoi.

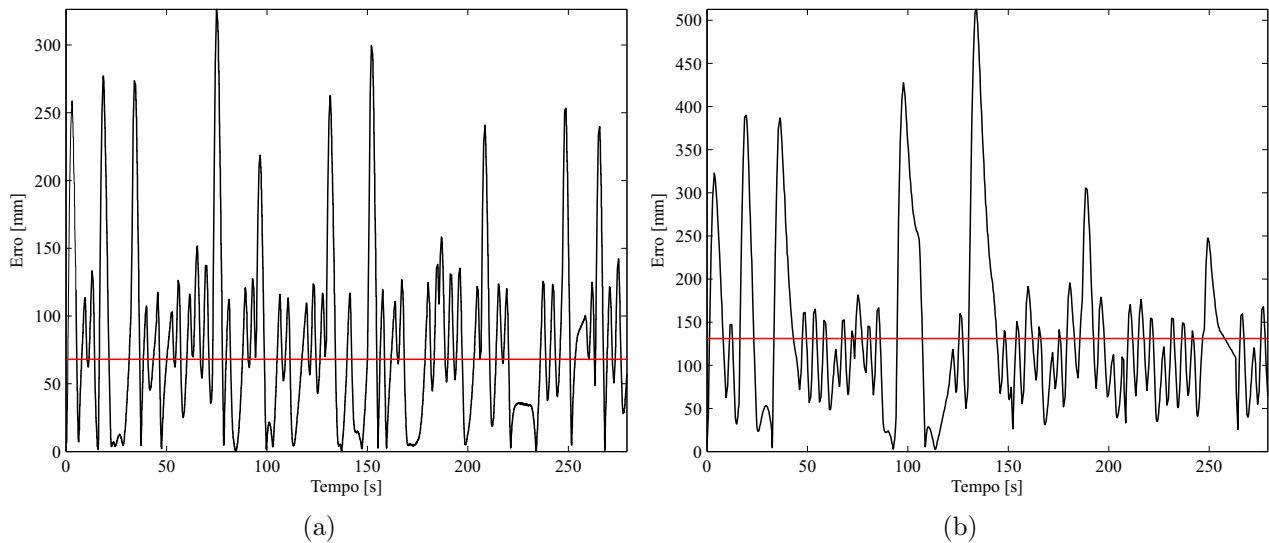


Figura 6.6: Erros do experimento (a) com e (b) sem o campo vetorial atrativo.

Nota-se nas figuras que os erros possuem um comportamento similar, no entanto, a magnitude dos erros produzidos no experimento que utiliza o campo vetorial são menores. Isso produziu um erro médio ( $\Delta E_{medio}$ ) menor, de  $68,12 \text{ mm}$ , contra um erro médio de  $130,95 \text{ mm}$  da estratégia sem o campo vetorial atrativo. Esta comparação revela um ganho de, aproximadamente, 92,24% do experimento com o campo vetorial, o que é um percentual bastante significativo em termos de erros, confirmando novamente o êxito no comportamento planejado.

Tabela 6.4: Resultados para a avaliação com (esquerda) e sem (direita) o campo atrativo.

Parâmetro	Valor	Parâmetro	Valor
$v_{max}$	200 [mm/s]	$v_{max}$	200 [mm/s]
$\Delta E_{medio}$	68,1168 [mm]	$\Delta E_{medio}$	130,9509 [mm]
$\Delta E_{max}$	326,3659 [mm]	$\Delta E_{max}$	512,7416 [mm]
$\Delta t$	279,17 [s]	$\Delta t$	281,14 [s]
$\alpha$	0,3	$\alpha$	0
$\sigma$	60,3898 [mm]	$\sigma$	95,9403 [mm]
$IC_{95\%}$	1,6776 [mm]	$IC_{95\%}$	6,6698 [mm]

Por se tratarem de resultados obtidos em simulação, é possível calcular medidas estatísticas do nível de acerto (*ground truth*) dos experimentos. Obteve-se que estes resultados para o controle compartilhado apresentaram um desvio padrão ( $\sigma_{\Delta E}$ ) de 60,39 *mm* com intervalo de confiança de 95% ( $IC_{95\%}$ ) no valor de 1,68 *mm*.

Com o nível de acerto obtido nesses resultados, mostra-se que, para os campos vetoriais o robô não se distanciou mais que 25 *cm* do diagrama de Voronoi, que é um comprimento menor que o próprio robô. Conclui-se, portanto, que os resultados produzidos pelo método de controle compartilhado para este experimento foram plenamente satisfatórios.

#### 6.1.4 Avaliação do impacto dos erros produzidos pelas HMIs

O experimento da seção anterior considerou um controle autônomo para avaliar o ganho do campo vetorial atrativo na navegação. A taxa de acerto dos comandos foi de 100%, ou seja, todos os comandos enviados foram executados corretamente. No entanto, é interessante avaliar o ganho dos campos vetoriais para diferentes taxas de acertos, uma vez que os aparelhos e técnicas utilizados em tecnologias assistivas não possuem uma taxa de acerto de 100%.

Para tanto, o mesmo mapa e objetivos da seção anterior foram utilizados mas, desta vez, com uma probabilidade de erros na execução dos comandos adicionada. Nota-se que, quando a região objetivo encontra-se na traseira do robô, isso significa que a navegação está acontecendo no sentido contrário ao desejado, e então o robô gira para o lado que minimiza seu erro angular, independente de onde está o diagrama de Voronoi no dado momento.

Os erros na execução dos comandos do controle autônomo foram adicionados por meio da utilização de um algoritmo tipo roleta com duas opções de sorteio, “acerto” e “erro”, onde a chance de cada é regulada conforme a necessidade do experimento. Dessa forma, caso seja sorteada a opção “erro”, o próximo comando do controlador autônomo será executado ao contrário, de maneira a levar o robô para o lado oposto que seria necessário para corrigir seu caminho.

Diferentemente da seção anterior, que não considerou o campo que controla a velocidade linear, esta seção utiliza ambos os campos atrativo e repulsivo, com o objetivo de avaliar o comportamento do método por completo. Portanto, tanto a manipulação dos ângulos de giro quanto da velocidade linear foram comparados com a estratégia que não os utiliza, ou seja, os parâmetros  $\alpha$  e  $\beta$  das equações 4.2, página 45, e 4.7, página 49, são zero, para a avaliação dos ganhos do método de campos vetoriais.

Para essa seção, utilizou-se o mesmo parâmetro de manipulação dos ângulos de giro ( $\alpha$ ) da seção anterior, o qual obteve bons resultados. Com relação a manipulação da velocidade linear, utilizou-se um aumento considerável do parâmetro que a regula ( $\beta$ ), de forma a modelar um campo repulsivo de grande intensidade, para cumprir especificamente os objetivos dessa seção.

O aumento no parâmetro  $\beta$ , para o valor de 4,37, possui um efeito de frenagem do robô conforme este se afasta do diagrama de Voronoi. Isto é, garante-se que a velocidade do robô será diminuída drasticamente todas as vezes que se afastar das zonas de segurança, efeito este que pode ser melhor observado nas figuras de resultados a seguir. Dessa forma, caso o robô execute comandos errados em localidades que estejam longe das zonas seguras, os efeitos negativos recairão com maior intensidade em sua orientação, uma vez que as baixas velocidades lineares produzidas pelo campo repulsivo alterarão sua posição espacial de maneira pouco significativa.



Assim, a posição do robô com relação ao diagrama de Voronoi não será alterada em grandes proporções, detendo o avanço do erro da posição com relação ao referido diagrama. Para a avaliação dos desempenhos da estratégia dessa seção, utilizou-se, portanto, quatro diferentes taxas de acerto, sendo elas 60%, 70%, 80% e 90%, respectivamente. Estas taxas de acerto foram consideradas em razão das diferentes estratégias relatadas na literatura, que usualmente variam dentro deste alcance. Taxas de acerto menores a 60% não foram utilizadas por produzir resultados insatisfatórios em aplicações de tecnologias assistivas.

Não foi considerada uma taxa de acerto de 100%, pois esses resultados assemelham-se aos apresentados da seção anterior. Para todos os experimentos, assim como na seção anterior, o robô parte da posição  $[1000 \text{ mm}, 1000 \text{ mm}, 90^\circ]^T$  e deve alcançar a região retangular iniciada no ponto  $[9000 \text{ mm}, 39000 \text{ mm}]^T$ . A distância máxima permitida ao robô para se distanciar do diagrama de Voronoi ( $\Delta d$ ), assim como na seção anterior, foi de 100 milímetros. A partir dessa distância, o controle compartilhado passa a atuar no robô. A velocidade máxima permitida ao robô também é a mesma da seção anterior, de 200 milímetros por segundo.

Considera-se que o robô está navegando em direção ao objetivo quando o ponto-objetivo, aqui dado por  $[9000 \text{ mm}, 39000 \text{ mm}]^T$ , encontra-se na parte frontal do robô, ou seja, quando o ângulo  $\phi$ , dado por

$$\phi = \psi - P_\theta \quad (6.1)$$

onde  $\psi$  é dado na Equação 4.6, página 49, e está no intervalo  $[-90^\circ, 90^\circ]$  com relação ao robô. Nessas condições, o robô se orientará em direção ao diagrama de Voronoi quando se distanciar deste numa distância maior que  $\Delta d$ . No entanto, dessa vez haverá o fator de “acerto” ou “erro” sorteado, conforme mencionado nos parágrafos anteriores, possibilitando erros que podem levar o robô para longe da região de segurança.

Se, porventura, o objetivo estiver localizado na parte traseira do robô, em outras palavras, se  $\phi$  estiver no intervalo  $(-180^\circ, -90^\circ] \cap [90^\circ, 180^\circ]$ , considera-se que este navega na direção contrária ao objetivo. Nessa situação, deseja-se que o robô volte a ter o ponto-objetivo na sua parte frontal o mais rápido possível. Portanto, o robô se orienta pelo lado (esquerdo ou direito) que possuir o menor ângulo com relação ao ponto-objetivo sem considerar a posição corrente do diagrama de Voronoi. Da mesma forma descrita no parágrafo anterior, o fator de erro pode levar a orientação do robô para longe do objetivo.

O objetivo dos experimentos dessa seção é verificar o impacto dos erros de classificação das HMIs assistivas no trajeto do robô. Esses erros podem se manifestar tanto no distanciamento do robô com relação ao diagrama de Voronoi quanto no seu direcionamento com relação ao ponto-objetivo. Assim, quer-se verificar se o campo vetorial da velocidade linear poderá evitar que o robô se distancie demasiadamente das regiões seguras, e se o campo dos vetores que controlam os ângulos de giro poderão manter o robô direcionado ao diagrama, de modo a completar seu trajeto.

A seguir, mostram-se todos os resultados obtidos para as taxas de acerto mencionadas anteriormente, cujos resultados serão comentados na sequência do texto. Cada uma das figuras mostra os resultados qualitativos dos caminhos com (a) e sem (b) o método, bem como os erros cometidos pelo robô (c e d) com relação ao diagrama de Voronoi durante o trajeto. Para cada experimento há uma tabela contendo os resultados quantitativos da navegação.

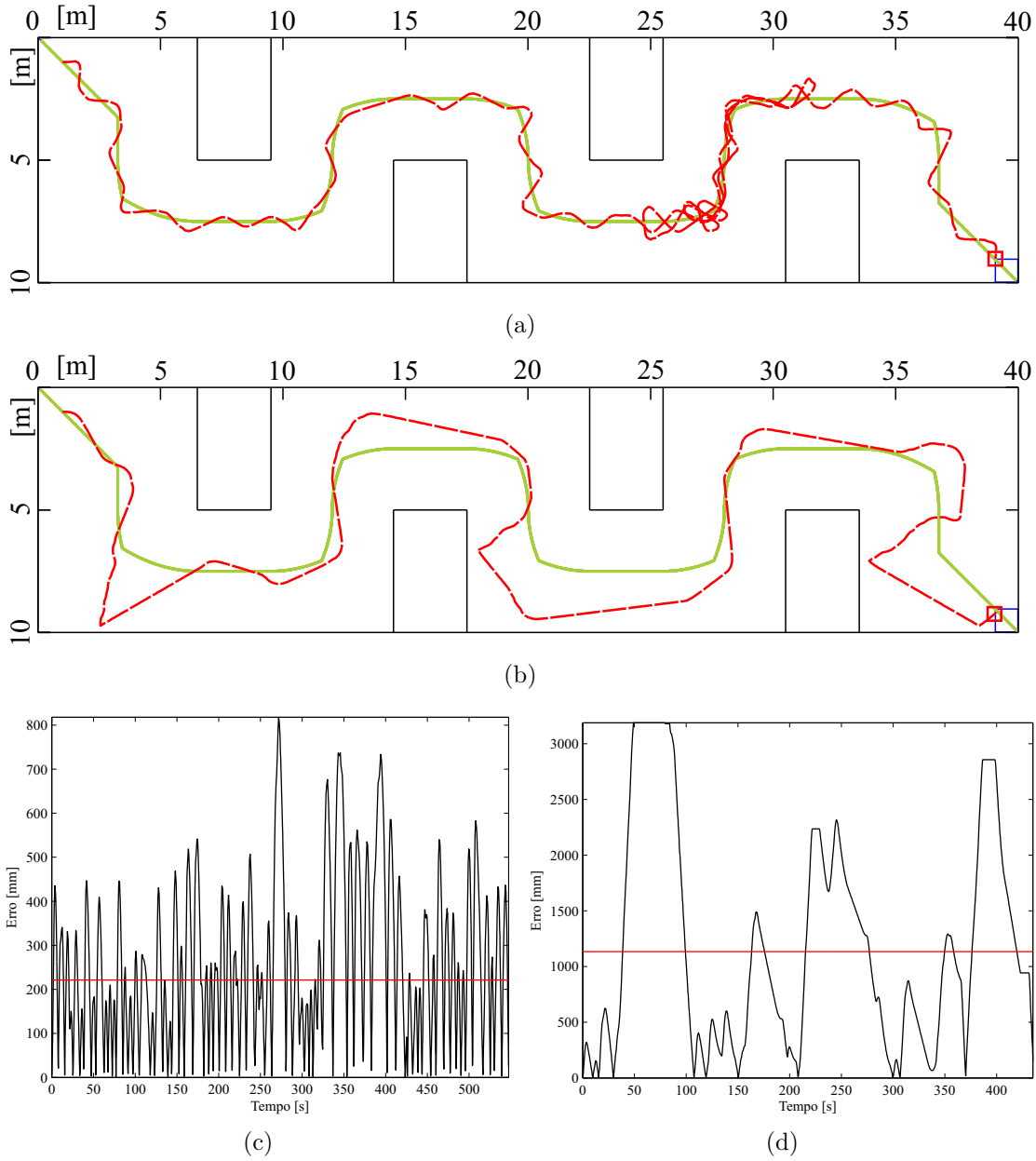


Figura 6.7: Taxa de acerto de 60%: caminhos (a) e (b) e erros (c) e (d) com e sem o método de campos veroriais, respectivamente.

Tabela 6.5: Resultados para acerto de 60% com (esquerda) e sem (direita) campos vectoriais.

Parâmetro	Valor	Parâmetro	Valor
$\Delta E_{medio}$	221,1866 [mm]	$\Delta E_{medio}$	1134,4493 [mm]
$\Delta E_{max}$	817,9824 [mm]	$\Delta E_{max}$	3188,8817 [mm]
$\Delta t$	547,2989 [s]	$\Delta t$	435,1475 [s]
$\alpha$	0,3	$\alpha$	0
$\beta$	4,37R	$\beta$	0
$\sigma$	172,4733 [mm]	$\sigma$	986,7631 [mm]
$IC_{95\%}$	8,7023 [mm]	$IC_{95\%}$	54,2069 [mm]

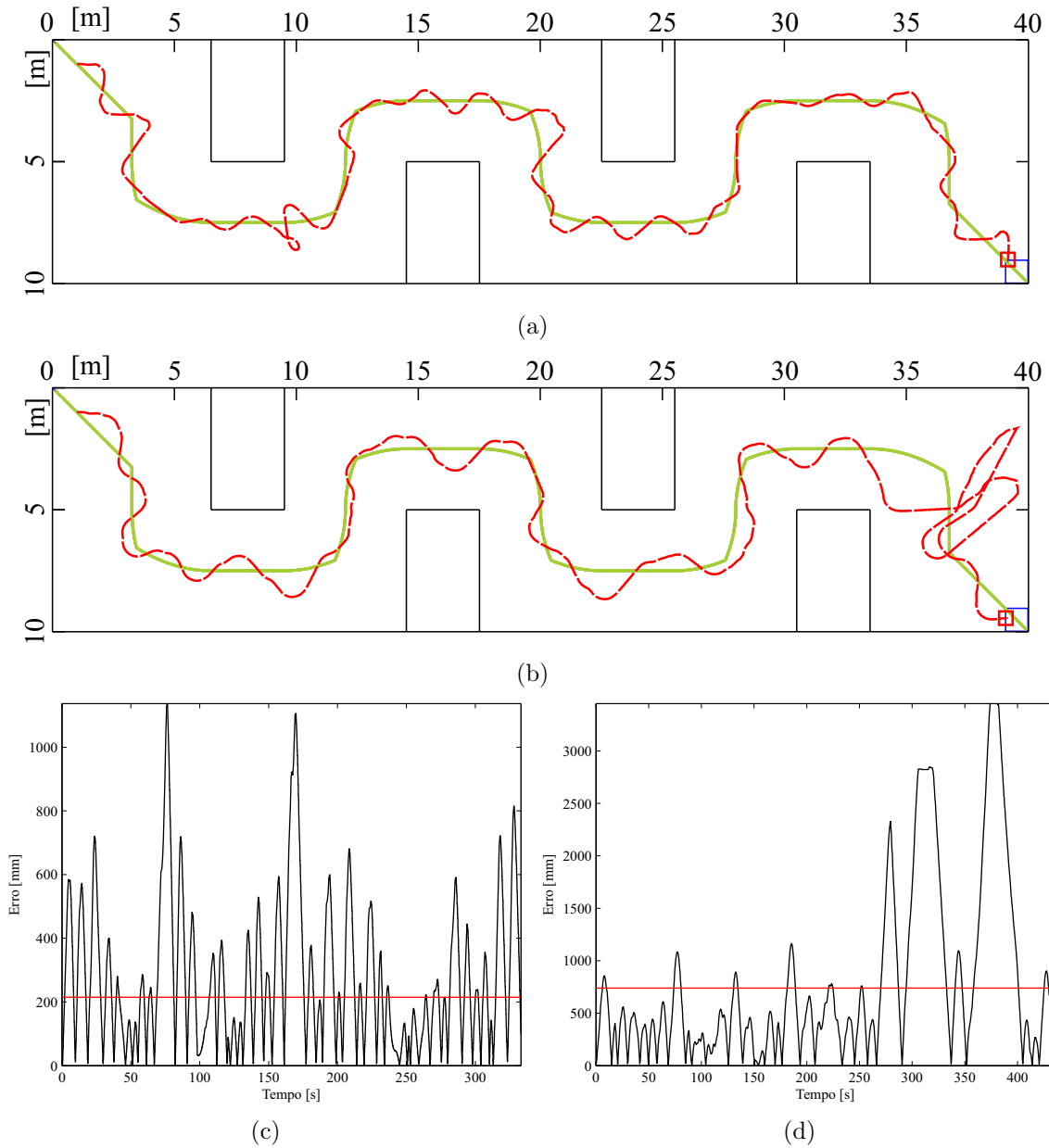


Figura 6.8: Taxa de acerto de 70%: caminhos (a) e (b), e erros (c) e (d) com e sem o método de campos vetoriais, respectivamente.

Tabela 6.6: Resultados com taxa de acerto de 70% para experimentos com (esquerda) e sem (direita) campos vetoriais.

Parâmetro	Valor	Parâmetro	Valor
$\Delta E_{medio}$	214,6676 [mm]	$\Delta E_{medio}$	739,3654 [mm]
$\Delta E_{max}$	1136,7229 [mm]	$\Delta E_{max}$	3450,7132 [mm]
$\Delta t$	333,1362 [s]	$\Delta t$	435,5207 [s]
$\alpha$	0,3	$\alpha$	0
$\beta$	4,37R	$\beta$	0
$\sigma$	210,2529 [mm]	$\sigma$	853,2763 [mm]
$IC_{95\%}$	5,8367 [mm]	$IC_{95\%}$	54,7233 [mm]

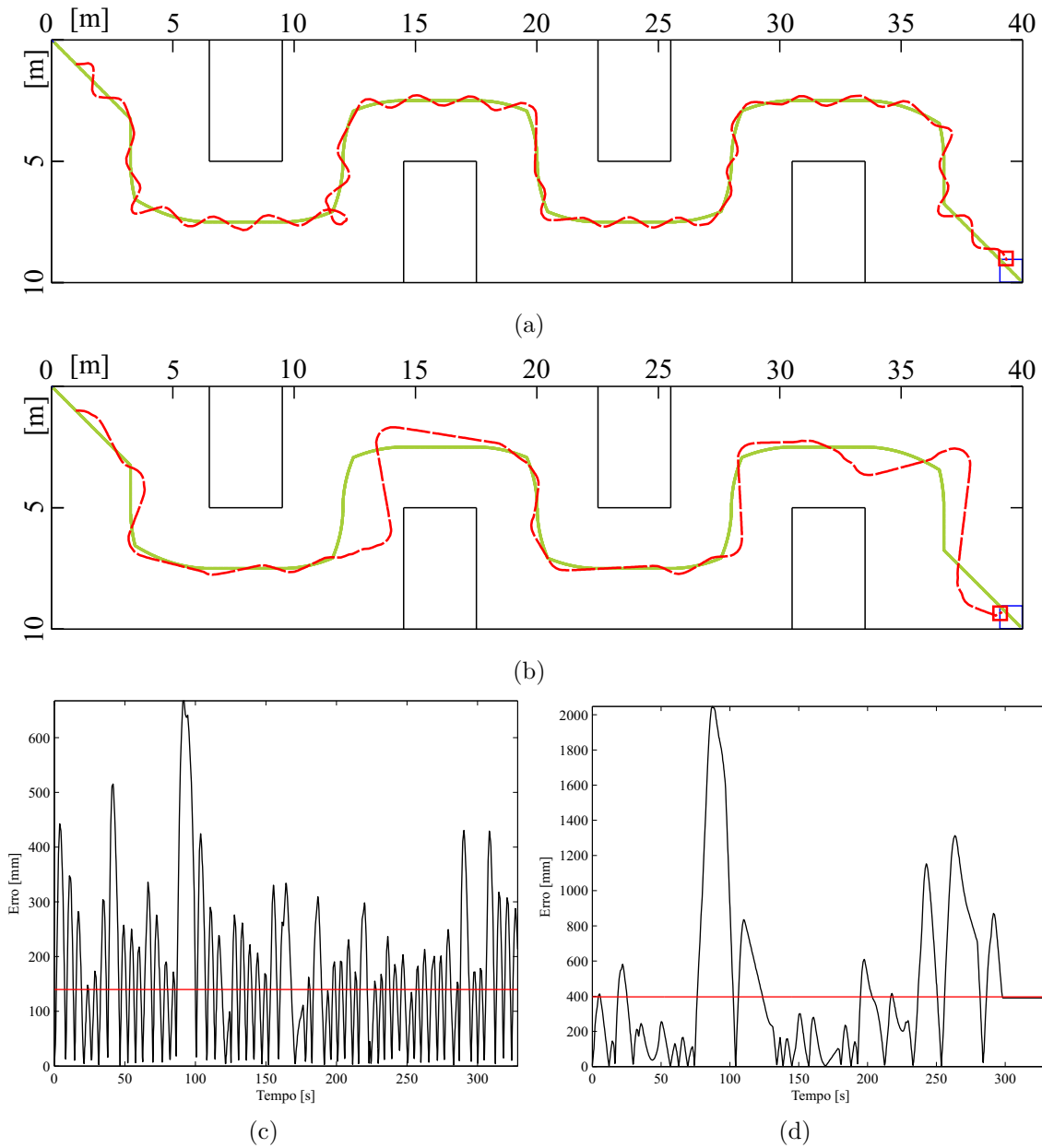


Figura 6.9: Taxa de acerto de 80%: caminhos (a) e (b), e erros (c) e (d) com e sem o método de campos vectoriais, respectivamente.

Tabela 6.7: Resultados com taxa de acerto de 80% para experimentos com (esquerda) e sem (direita) campos vectoriais.

Parâmetro	Valor	Parâmetro	Valor
$\Delta E_{medio}$	139,8601 [mm]	$\Delta E_{medio}$	396,1504 [mm]
$\Delta E_{max}$	667,4697 [mm]	$\Delta E_{max}$	2047,4203 [mm]
$\Delta t$	328,4261 [s]	$\Delta t$	331,8880 [s]
$\alpha$	0,3	$\alpha$	0
$\beta$	4,37R	$\beta$	0
$\sigma$	113,9806 [mm]	$\sigma$	427,0392 [mm]
$IC_{95\%}$	7,1436 [mm]	$IC_{95\%}$	25,7203 [mm]

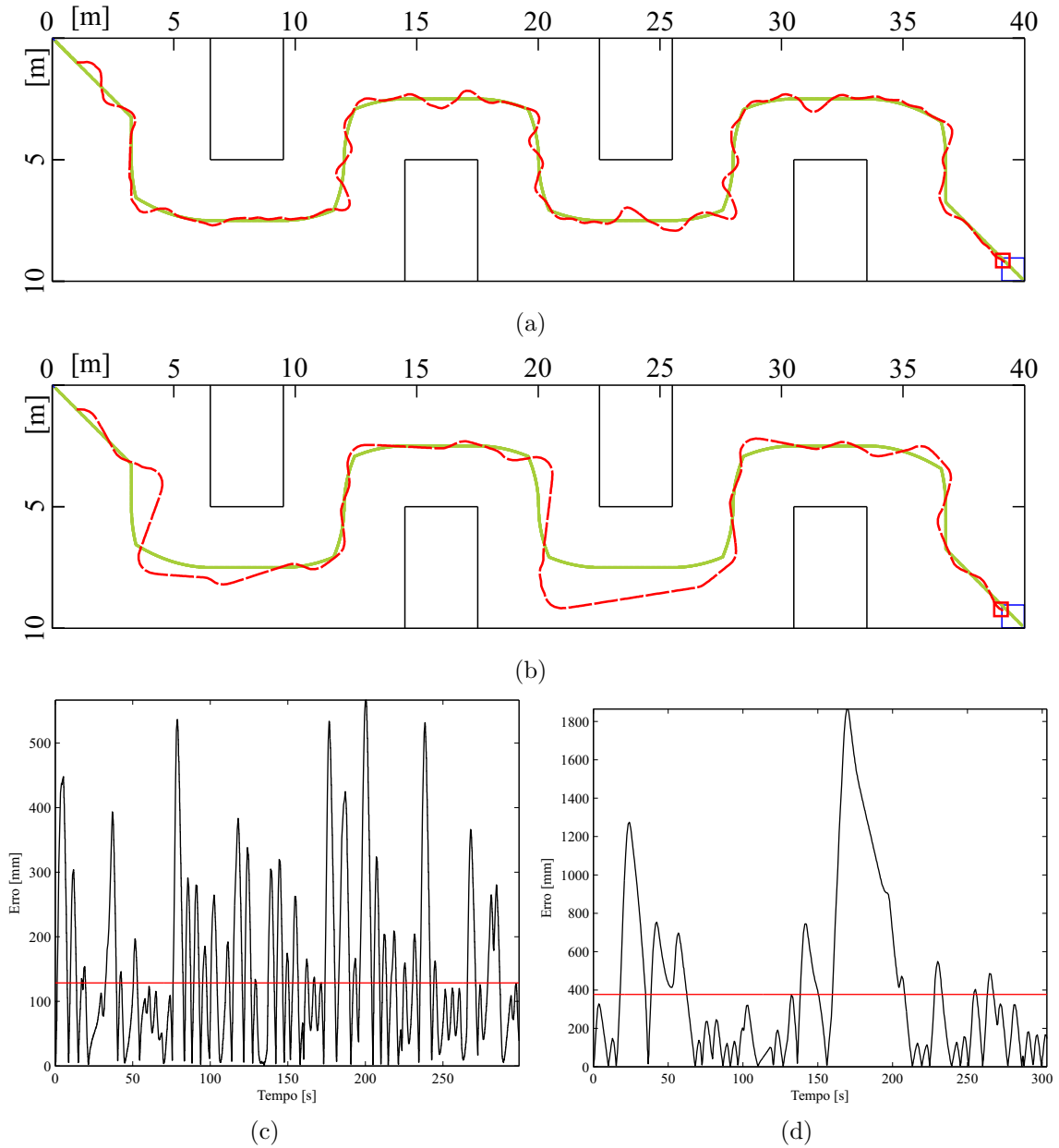


Figura 6.10: Taxa de acerto de 90%: caminhos (a) e (b), e erros (c) e (d) com e sem o método de campos vetoriais, respectivamente.

Tabela 6.8: Resultados com taxa de acerto de 90% para experimentos com (esquerda) e sem (direita) campos vetoriais.

Parâmetro	Valor	Parâmetro	Valor
$\Delta E_{medio}$	128,4769 [mm]	$\Delta E_{medio}$	377,1489 [mm]
$\Delta E_{max}$	566,6623 [mm]	$\Delta E_{max}$	1864,8603 [mm]
$\Delta t$	299,0080 [s]	$\Delta t$	303,0586 [s]
$\alpha$	0,3	$\alpha$	0
$\beta$	4,37R	$\beta$	0
$\sigma$	119,8594 [mm]	$\sigma$	429,2974 [mm]
$IC_{95\%}$	3,5743 [mm]	$IC_{95\%}$	27,2136 [mm]

O primeiro experimento possui a menor taxa de acerto, 60%. Isso significa que, para um comando de correção de caminhos, há 40% de chances do comando executado piorar a posição do robô, girando-o para o lado contrário do qual seria necessário para minimizar os erros. Os dados quantitativos do experimento estão na Tabela 6.5, e na Figura 6.7, mostrando, respectivamente, o trajeto e os erros da posição do robô com relação ao diagrama de Voronoi.

A Figura 6.7(a) mostra que a estratégia do campo repulsivo operou no sentido de impedir que o robô se distanciasse excessivamente do diagrama de Voronoi. Ainda assim, a alta taxa de erros inserida nos comandos fez com que o robô navegasse no sentido inverso do objetivo. Já a Figura 6.7(b) mostra que o robô se distanciou demasiadamente do diagrama por diversas vezes, nos mesmos moldes do trabalho de Philips et al. (2007), mostrado nas figuras de Philips et al. (2007), páginas 411 a 412. Nota-se que o ganho do método de campos vetoriais para essa taxa de erros é muito alto, sendo acima de 412,89%.

O segundo experimento utilizou uma taxa de acerto de 70%. Seus resultados estão na Figura 6.8 e na Tabela 6.6. Para esta navegação, o caminho sem o método de campos vetoriais (Figura 6.8(b)) obteve um resultado superior ao do experimento com 60% de acerto. Mesmo assim, o robô obteve erros grandes com relação ao diagrama de Voronoi e também deslocou-se no sentido inverso ao do objetivo. O trajeto com campos vetoriais (Figura 6.8(a)) obteve melhorias com relação ao obtido na taxa de 60%, com erros menores. Com relação ao ganho do método, verificou-se um índice médio de, aproximadamente, 244,42% com relação aos dados sem a participação do método de campos vetoriais.

Os dados referentes à taxa de acerto de 80% mantiveram o crescimento na taxa de acerto do robô com relação ao diagrama de Voronoi. Os resultados concernentes à navegação com o método (Figura 6.9(a)) apresentaram melhoria quando comparados com os dados anteriores. Com relação à estratégia sem campos vetoriais desse experimento (Figura 6.9(b)), o ganho médio do método desse trabalho de pesquisa foi de 183,25%, mantendo o bom índice de ganhos com o robô sempre próximo ao diagrama de Voronoi, e alcançando a área objetivo.

O último experimento dessa seção, com taxa de acerto de 90% (Figura 6.10) apresentou bons resultados, parecidos com os obtidos da seção anterior (Figura 6.5), que considera a taxa de acerto de 100%. Nenhum dos resultados (com e sem campos vetoriais) apresentou navegação contrária à região objetivo. Ainda assim, os resultados para os campos vetoriais foram bastante superiores, com ganho médio de 193,55% com relação ao erro ao diagrama de Voronoi. Mesmo o desvio padrão dos dados apresentados possui um nível bastante baixo, de aproximadamente 12 centímetros, quando comparado com as dimensões do robô. O índice de confiança, de 95%, ficou abaixo de 4 centímetros, demonstrando a confiabilidade dos resultados obtidos.

Conforme os experimentos dessa seção demonstraram, mesmo com taxas baixas de acerto, o método dos campos vetoriais diminuiu significativamente o impacto dos erros produzidos pela HMI assistiva, seja por má classificação dos comandos adquiridos do usuário, seja por comandos emitidos incorretamente pelo próprio usuário. Os ganhos médios foram favoráveis aos campos vetoriais em todos os experimentos, sempre com um ganho percentual acima de 100%. Dessa forma, o método desenvolvido nessa pesquisa demonstra sua tolerância a erros nos comandos emitidos, mantendo o usuário próximo das zonas mais seguras do mapa. Tal conclusão é constatada pelos resultados quantitativos mostrados nas tabelas de cada um dos experimentos.

### 6.1.5 Navegação por ambientes de grandes dimensões

Em todos os experimentos descritos anteriormente, o método de controle compartilhado por campos vetoriais foi validado experimentalmente para uma série de situações planejadas para a avaliação do método. Esta seção descreve um experimento para validar ambos os campos, atrativo e repulsivo, atuando em conjunto em situações que um usuário de cadeiras de rodas robótica pode confrontar no cotidiano. Além disso, faz-se necessário validar o desempenho do mapeamento por FastSLAM implementado para ambientes reais.

Este experimento tem como objetivo construir o mapa completo do laboratório LCA, situado na UNICAMP. Desta forma, o usuário deve navegar o robô por todos os cômodos do laboratório, enfrentando as mais diversas situações de um ambiente diário, tais como corredores estreitos, passagens de portas, mobília, obstáculos fixos e móveis, e assim por diante. Os campos vetoriais devem auxiliar o acesso aos ambientes, enquanto o FastSLAM deve corrigir a odometria do robô para que o mapa fique coerente com a construção real. Nessa navegação utilizou-se o robô Pioneer P3-DX em virtude da acessibilidade a todos os ambientes disponíveis. No LCA existem algumas passagens de portas muito estreitas, menores que a largura do robô Seekur Jr., impossibilitando seu acesso a todos os locais existentes.

Os resultados quantitativos podem ser observados na Tabela 6.9 enquanto os qualitativos estão na Figura 6.11. Conforme se observa no caminho percorrido pelo robô, o início e o final da navegação se dá no mesmo cômodo. Nessa navegação não havia objetivos quanto a seguir o Voronoi. O robô pôde navegar livremente pelo ambiente, se aproximando de obstáculos sem restrições. O objetivo era acessar a maior quantidade de locais possíveis para obter o mapa completo do laboratório para a etapa de controle autônomo. Na Figura 6.12 tem-se algumas imagens do laboratório em questão. Nessa navegação, todos os resultados da seção anterior fizeram-se presentes, fazendo com que o robô mantivesse seu curso de maneira segura, mesmo com a adversidade do aparelho utilizado (Emotiv) não possuir classificação com 100% de acerto.

Tabela 6.9: Navegação e mapeamento do LCA.

Parâmetro	Valor	Parâmetro	Valor
$v_{medio}$	70,96 [mm/s]	$n^o$ de referências	516
$\Delta S$	206832 [mm]	$n^o$ ações	460
$\Delta t$	2914,70 [s]	$n^o$ giros esq.	165
$\beta$	0,6R	$n^o$ giros dir.	60

Com o FastSLAM 2.0 foi possível produzir um mapa coerente do ambiente real, em que pequenas distorções por desvios de odometria, como paredes curvadas, foram corrigidas quando se converteram as medidas para um mapa de células de lado 10 centímetros. O fato do trajeto do robô retornar a regiões conhecidas antes de entrar em novos ambientes ajuda a reduzir os erros acumulados ao longo do caminho. Os landmarks utilizados, conforme mencionado na Seção 4.4, são os cantos de paredes. Eles são obtidos por meio de cruzamentos das retas dadas pelo algoritmo de extração *Split-And-Merge* (Siegwart & Nourbakhsh 2004). O número de referências encontradas, 516, se deve ao grande número de obstáculos cuja forma pode produzir esse tipo de marcação de ambiente, tais como mesas, cadeiras, armários, dentre outros.

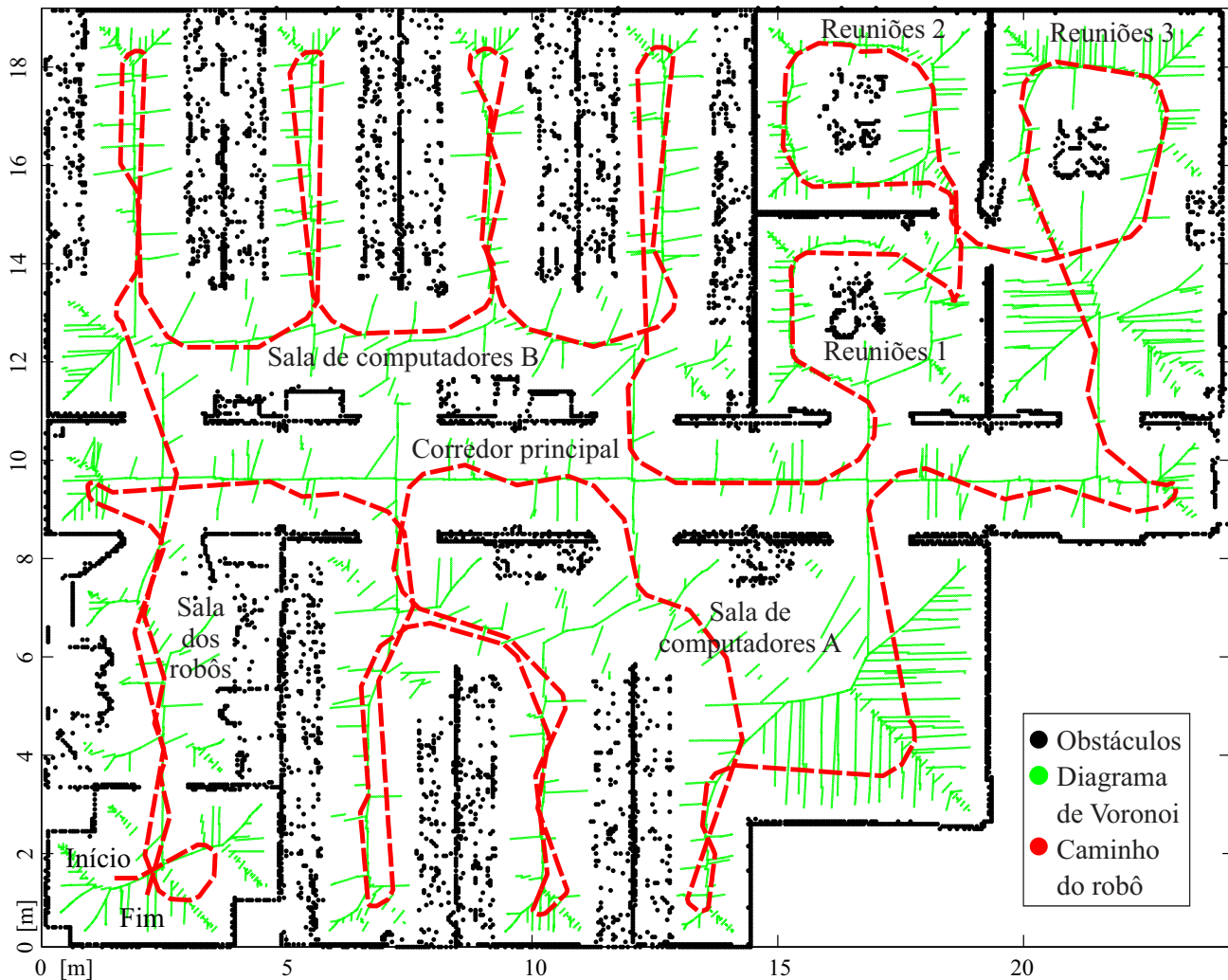


Figura 6.11: Mapeamento do LCA.

Em um tempo de navegação de 48 minutos e 35 segundos o robô, por meio dos campos vetoriais, navegou por todos os ambientes disponíveis do laboratório, percorrendo um caminho de 206,83 metros. Nenhuma situação em particular foi evitada, o que possibilitou gerar um mapa de, aproximadamente,  $408 m^2$ . Os resultados contidos na Tabela 6.9 mostram que os comandos de giro à esquerda foram preferidos com relação aos da direita em 73,33% das vezes. Este fato é resultado da diferença de classificação obtida pelo Emotiv (Tabela 3.1, página 28). Assim, pode-se concluir que os resultados alcançados por meio dos campos vetoriais mostraram-se satisfatórios, validando o método para ambientes não controlados.

Para comparação com a literatura citada no Capítulo 2, o trabalho que mais se alinha com os objetivos desta seção é o controle compartilhado semiautônomo de Trieu et al. (2008). Seu sistema provê ao usuário quatro comandos básicos (“girar à esquerda”, “girar à direita”, “ir para frente” e “ir para trás”), os quais são combinados com um mapa global do ambiente. Dessa forma, o robô pode navegar para um local que obedeça ao comando do usuário e que não se choque com nenhum obstáculo do ambiente. O deslocamento do robô na direção escolhida pelo usuário é feito de maneira autônoma.



O método de Trieu et al. (2008) também utilizou o mapeamento dos ambientes, sendo este feito de maneira global. Seus resultados nesse quesito estão exibidos de maneira qualitativa nas Figuras 8 e 9, página 4257 da referência citada, onde observam-se bons mapeamentos dos ambientes em que realizaram seus testes, mostrando diversos detalhes de mobílias e outros obstáculos. Por esses resultados, pode-se considerar que, no ponto do mapeamento, ambos os trabalhos se equivalem.

No entanto, as diferenças entre os trabalhos se dão no controle compartilhado. O controle compartilhado semiautônomo de Trieu et al. (2008) requer muitas interações do usuário para navegar curtas distâncias, tornando o método cansativo. Os caminhos obtidos mostram que o robô exibiu um bom comportamento no que se refere a passagem pelos ambientes. Observa-se ainda que o robô jamais se aproxima de obstáculos pela ação autônoma do controle compartilhado. Esta ação compromete a liberdade do usuário em navegar próximo aos obstáculos, caso queira. Isso não pode ser alcançado pelo método de Trieu et al. (2008), dadas as suas características, onde a combinação entre o mapa e a direção escolhida pode produzir uma posição final diferente daquela que o usuário espera, o que pode gerar frustração. Para o controle compartilhado por campos vetoriais, o usuário pode direcionar-se para quaisquer localidades sem ações autônomas, e a navegação segue o curso escolhido pelo usuário em sua integralidade e com segurança.



Figura 6.12: Alguns ambientes do LCA: (a) corredor principal, (b) sala de computadores B, (c) sala de reuniões 1 e (d) sala de reuniões 2.

### 6.1.6 Utilização de sensores de menor resolução

Durante o curso desta tese, foram mostrados resultados para sensores tipo *laser*. Tais sensores são bastante precisos e produzem resultados bastante satisfatórios, conforme as seções de resultados anteriores mostraram. No entanto, os custos desses sensores são, atualmente, bastante altos, e outros sensores de menores preços podem ser utilizados para reduzir os custos de um projeto. Um exemplo desse tipo de sensor menos custoso são os sonares, sensores de distâncias bastante difundidos cujos custos são muito menores que dos *lasers*.

O robô P3-DX, utilizado nessa pesquisa, possui 16 sonares, sendo 8 na parte frontal e o restante na parte traseira, de forma a cobrir o robô por todos os lados, conforme mostrado na Figura 6.13. Apesar da baixa resolução espacial dada por apenas 16 sonares, é possível utilizá-los para montar um mapa local do ambiente, e assim, calcular um diagrama de Voronoi para a utilização das áreas livres percebidas no dado instante de tempo. Dessa forma, não se confeccionará um mapa global do ambiente a partir das medidas de sonar, ao contrário do que foi realizado com o *laser*, dada a baixa resolução para a obtenção de referências e características. Ainda assim é possível navegar com segurança, utilizando mapas locais a cada interação do algoritmo, os quais serão utilizados para calcular o diagrama de Voronoi do ambiente atual.

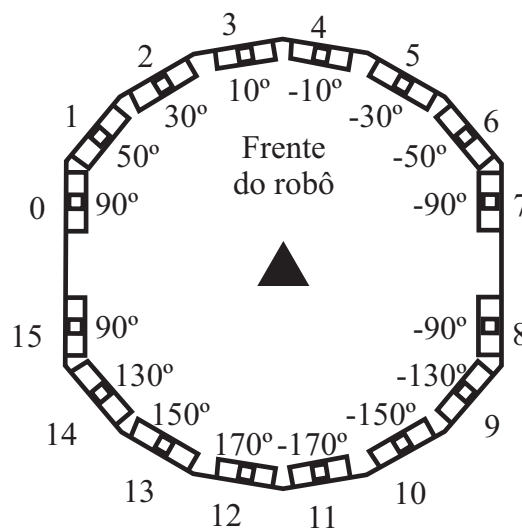


Figura 6.13: Disposição dos sonares no robô Pioneer P3-DX.

Para utilizar essa estratégia de mapas locais em um ambiente realista, o mapa global do laboratório LCA/UNICAMP, obtido na seção anterior com o FastSLAM, (Figura 6.11, página 88), foi transformado em um mapa virtual para o simulador MobileSim, de forma a possibilitar experimentos como os realizados nas Seções 6.1.3 e 6.1.4. Assim, é possível observar a situação real que se apresenta ao robô, com todos os obstáculos nas vizinhanças, e comparar com o mapa local construído a partir de seus sonares, para concluir acerca de sua eficácia.

A estratégia consiste em utilizar as distâncias obtidas dos sonares, convertê-las em pontos espaciais cartesianos no referencial do robô, e ligá-los por meio de retas, de forma a fechar o mapa local. A partir desse ambiente fechado, calcula-se o diagrama de Voronoi para o campo atrativo. Para o campo repulsivo, utilizam-se as medidas de distâncias dadas pelos sonares, uma vez que já se conhecem os ângulos referentes a cada um dos sensores (Figura 6.13).

A Figura 6.14 mostra a situação em que todos os sonares obtêm medidas próximas ao robô. A situação real em que o robô se encontra está mostrada na Figura 6.14(a), onde se podem observar todos os obstáculos presentes no ambiente local. Note-se que estes resultados foram obtidos por meio de simulação com o mapa do LCA obtido anteriormente. A Figura 6.14(b) mostra o mapa local construído com as 16 leituras de sonar, bem como seu diagrama de Voronoi resultante. Ressalta-se que, por questões de baixa resolução espacial, os obstáculos não ficam bem representados, como é possível observar. No entanto, para esse caso, as zonas seguras puderam ser bem representadas pelo diagrama de Voronoi, e assim, é possível utilizar a estratégia de manipulação dos ângulos de giro dos campos vetoriais.

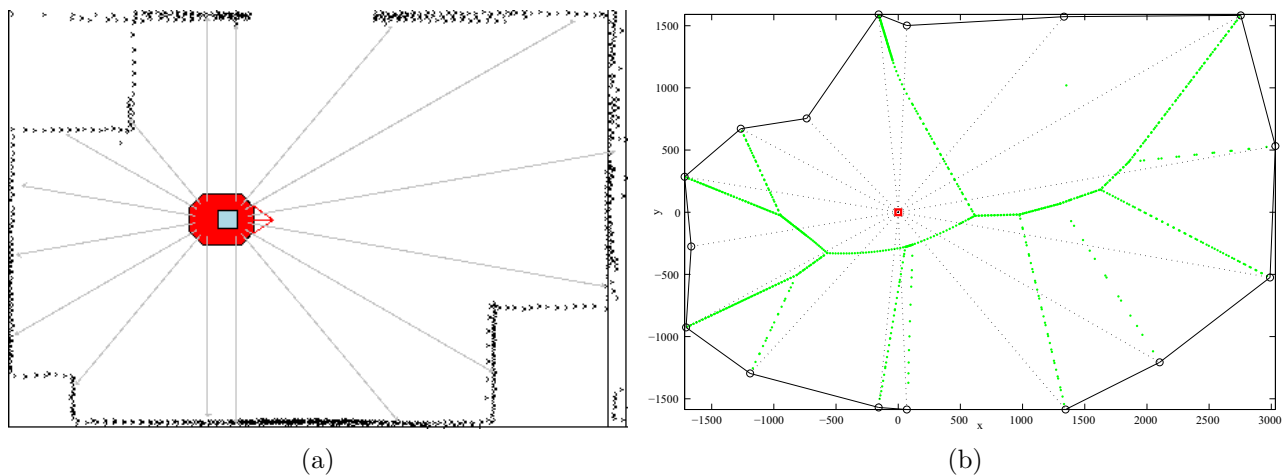


Figura 6.14: Situação 1: (a) ambiente do robô e (b) mapa local com diagrama de Voronoi obtido pelo sonar.

No entanto, sabe-se que os sonares produzem erros de distância maiores que os observados com sensores *lasers*. Um dos principais problemas é a saturação da distância máxima permitida pelo sonar. No caso dos sonares do Pioneer P3-DX utilizado nessa pesquisa, essa distância máxima é de 5 metros, tanto no simulador quanto no robô real. A Figura 6.15 representa a situação descrita. Há medidas de sonar que devem ser desprezadas, pois os obstáculos encontram-se fora do alcance do sonar (Figura 6.15(a)). Dessa forma, não se pode confiar na informação que esses sensores trazem, pois não representam reflexões causadas por obstáculos.

Assim, elimina-se essa informação na construção do mapa local, e calcula-se o diagrama de Voronoi correspondente a esse mapa, como mostrado na Figura 6.15(b). Da mesma forma que na figura anterior, não se representa com fidelidade as redondezas do robô pela falta de sensores em ângulos diferentes para captar mais informação do ambiente. Mesmo assim, observa-se que o diagrama de Voronoi passa pelos locais mais seguros e pode ser utilizado para influenciar o comportamento do robô, por meio de campos vetoriais.

Há, entretanto, situações específicas onde um conjunto com quantidade limitada de sensores (no caso, sonares) podem não capturar informações sobre obstáculos que estejam próximos ao robô, como mostrado na Figura 6.16. Isso acontece justamente pela pouca quantidade de sensores disponíveis. Somente com a adição de mais sensores pode-se aumentar a resolução espacial do conjunto, e suprir esse tipo de falha.

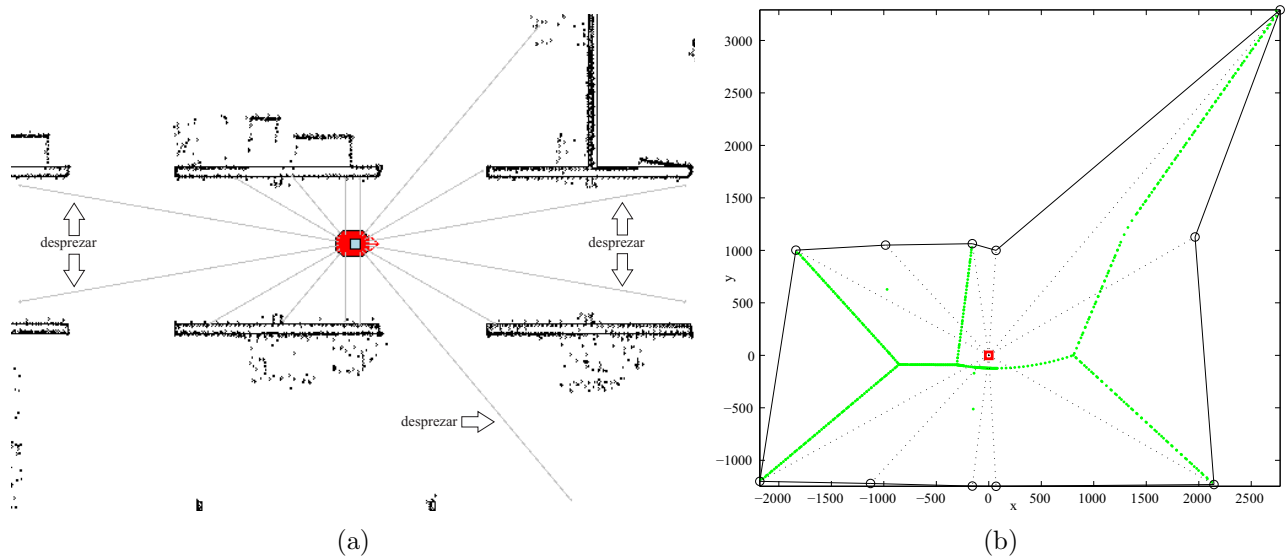


Figura 6.15: Situação 2: (a) ambiente do robô e (b) mapa local com diagrama de Voronoi obtido pelo sonar.

A Figura 6.16(a) mostra obstáculos bastante próximos ao robô que não puderam ser captados por conta de sua angulação específica com relação ao robô. Apesar disso, o diagrama de Voronoi ainda garante as regiões mais seguras para o cenário apresentado.

Apesar do caso apresentado na Figura 6.16, nota-se que para a maioria das situações os sonares captam os obstáculos próximos, fornecendo um bom mapa local e boas medidas para o campo repulsivo que controla a velocidade linear. Nesse sentido, mostra-se que é possível utilizar o método de campos vetoriais com sensores menos custosos e, ainda assim, obter bons resultados de desempenho no sentido da proteção do usuário.

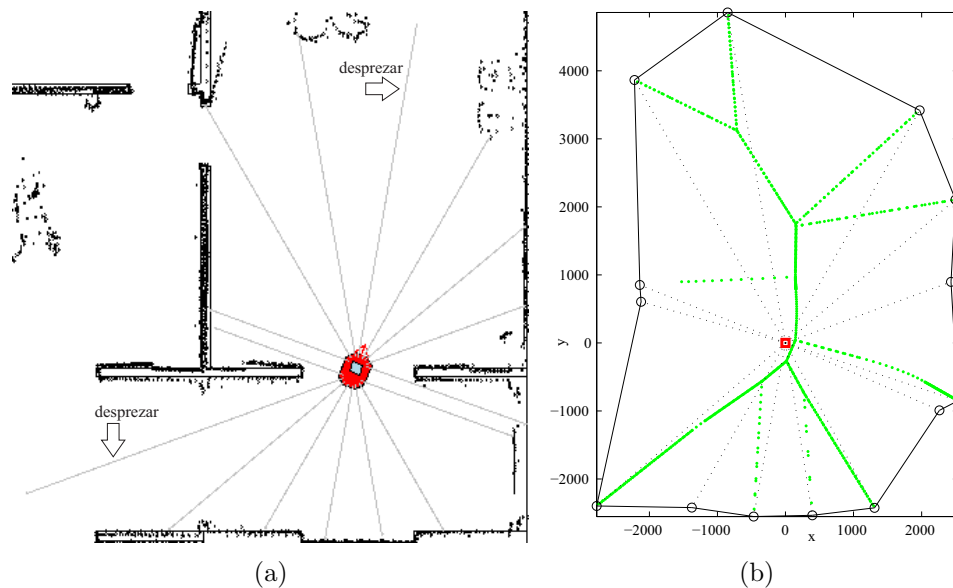


Figura 6.16: Situação 3: (a) ambiente do robô e (b) mapa local com diagrama de Voronoi obtido pelo sonar.

## 6.2 Experimentos de navegação autônoma

Esta seção apresenta os resultados para a estratégia de navegação autônoma proposta no Capítulo 5, mais especificamente, para a estratégia de cálculo e recálculo de caminhos  $A^*$ , conforme relatado na Seção 5.2. Analisa-se a efetividade do controlador por campos vetoriais desenvolvido para a tarefa de navegar o robô através do caminho obtido (Seção 5.3), bem como sua comparação com um controlador clássico, o PID (proporcional-integral-derivativo).

### 6.2.1 Análise do controlador autônomo com campos vetoriais

Esta seção avalia dois aspectos da estratégia desenvolvida: o cálculo dos caminhos  $A^*$  com a política de melhores caminhos no sentido da segurança, e o desempenho do controlador autônomo em percorrê-los corretamente. Esse experimento utilizou recursos de simulação para a obtenção de resultados quantitativos (*ground truth*). O mapa do ambiente real do laboratório LCA/UNICAMP, mostrado na Figura 6.17, foi obtido por meio da navegação por controle compartilhado, conforme descrito ao longo deste capítulo, e discretizado em células de lado igual a 200 mm, de forma a fornecer um mapa em grade para as buscas do algoritmo  $A^*$ .

A primeira análise é referente à estratégia em que se altera a política de custo do  $A^*$ . A Figura 6.17 mostra os três caminhos calculados com o  $A^*$  de forma a percorrer grandes distâncias dentro das possibilidades do ambiente. Nota-se uma pequena diferença entre os caminhos e o diagrama de Voronoi, as quais se deram por conta da perda de resolução do mapa na discretização do mesmo, sendo este um resultado esperado.

Ainda assim, as diferenças entre o diagrama de Voronoi e os caminhos  $A^*$  são pequenas. Os caminhos calculados estão distantes de todos os obstáculos e, ao mesmo tempo, bastante próximos ao diagrama de Voronoi, maximizando a segurança do usuário. Dessa forma, para a primeira etapa de avaliação, demonstra-se qualitativamente que a estratégia utilizada para a função de custo para a otimização do  $A^*$  obteve êxito em calcular os menores caminhos possíveis dentro das áreas seguras, validando a estratégia.

A segunda análise se dá com relação à efetividade do controlador para a tarefa de navegar o robô através do caminho obtido. Os valores dos parâmetros do controlador são  $\alpha = 0,59$  e  $\beta = 1,51$ , os quais foram obtidos por meio de ajuste empírico, no qual o objetivo é, ao mesmo tempo, obter uma boa acurácia para o caminho, e tornar o robô bastante sensível a possíveis obstáculos do ambiente, enfatizando a segurança do usuário. Os detalhes quantitativos dos três caminhos são explicitados na Tabela 6.10, e os erros relativos ao caminho estão na Figura 6.18.

A velocidade máxima permitida para os experimentos é de 150 mm/s, pois considera-se que seja uma velocidade coerente com as expectativas de uma navegação autônoma, em que o robô conduz um usuário com deficiência, além de estar nos padrões relatados na literatura (Iturrate et al. 2009, Bi et al. 2013). Todos os dados apresentados para os experimentos permaneceram estáveis em níveis aceitáveis, demonstrando que os resultados possuem um grau de repetição alto, e dessa forma, o nível de confiabilidade no desempenho é igualmente elevado.

Ao analisar os dados numéricos fornecidos na Tabela 6.10, nota-se que os erros médios do robô ao caminho planejado ( $\Delta E_{medio}$ ) foram de, respectivamente, 2,70 mm, 3,03 mm e 3,10 mm, sendo considerados valores baixos, denotando uma boa parametragem para o controlador.

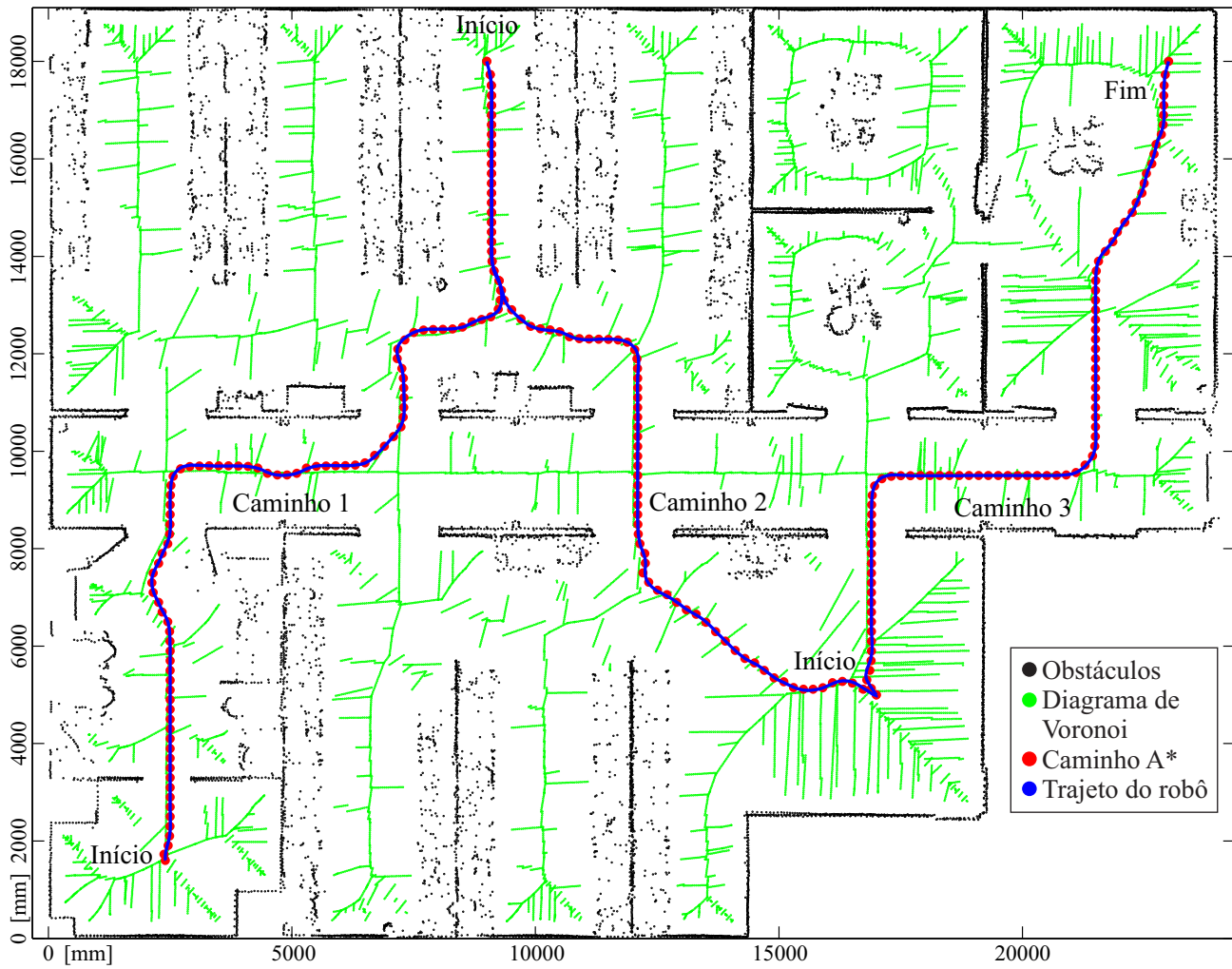


Figura 6.17: Caminhos utilizados na avaliação do controlador autônomo.

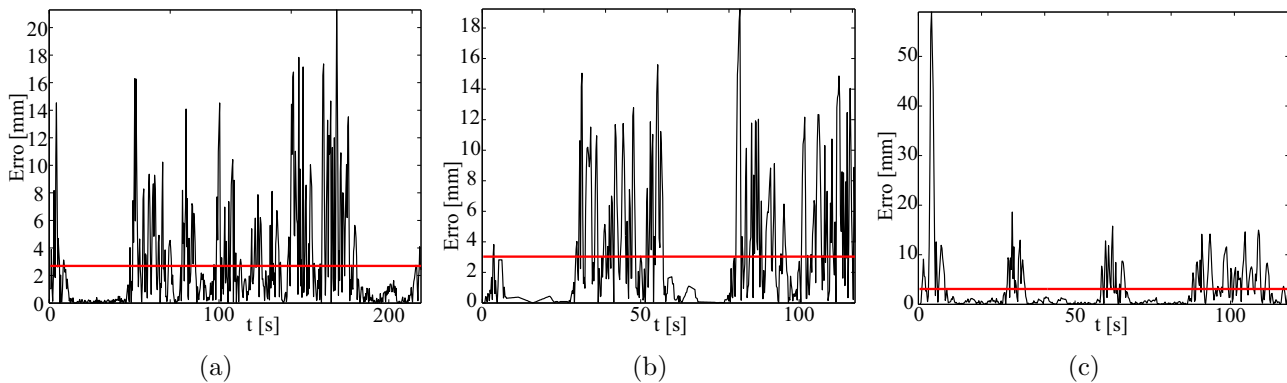


Figura 6.18: Erros do controlador ao navegar pelos caminhos (a) 1, (b) 2 e (c) 3.



Tabela 6.10: Resultados para o controlador por campos vetoriais.

Caminho 1		Caminho 2		Caminho 3	
Parâmetro	Valor	Parâmetro	Valor	Parâmetro	Valor
$v_{max}$	150 $mm/s$	$v_{max}$	150 $mm/s$	$v_{max}$	150 $mm/s$
$v_{media}$	100,52 $mm/s$	$v_{media}$	126,63 $mm/s$	$v_{media}$	119,01 $mm/s$
$w_{max}$	18,53°/s	$w_{max}$	19,94°/s	$w_{max}$	18,87°/s
$\Delta S$	22011,61 $mm$	$\Delta S$	18270,77 $mm$	$\Delta S$	17399,29 $mm$
$\Delta t$	218,97 $s$	$\Delta t$	144,28 $s$	$\Delta t$	146,20 $s$
$\Delta E_{max}$	21,29 $mm$	$\Delta E_{max}$	19,25 $mm$	$\Delta E_{max}$	58,96 $mm$
$\Delta E_{medio}$	2,70 $mm$	$\Delta E_{medio}$	3,03 $mm$	$\Delta E_{medio}$	3,10 $mm$
$\sigma$	3,65 $mm$	$\sigma$	3,91 $mm$	$\sigma$	6,07 $mm$
$IC_{99\%}$	0,36 $mm$	$IC_{99\%}$	0,46 $mm$	$IC_{99\%}$	0,73 $mm$

Os valores de desvio padrão ( $\sigma$ ), respectivamente, 3,65  $mm$ , 3,91  $mm$  e 6,07  $mm$ , permaneceram baixos e contidos, ou seja, não crescem indefinidamente, mantendo os erros entre a posição do robô e do caminho ( $\Delta E$ ) dentro de faixas bem definidas, conforme atestam as Figuras 6.18(a), 6.18(b) e 6.18(c), que apresentam todos os erros  $\Delta E$  desenvolvidos pelo robô ao longo do trajeto e do tempo. Este resultado também comprova a estabilidade do controlador.

Ainda sobre os dados numéricos, os intervalos de confiança de 99% ( $IC_{99\%}$ ) obtidos em todos os experimentos ficaram abaixo de 1  $mm$ , denotando que os experimentos possuem alto grau de confiabilidade, tornando esse controlador adequado para a tarefa de conduzir o usuário pelos caminhos designados. Combinando as medidas dos três experimentos, a distância percorrida é de 57681,67  $mm$ , num tempo total de 509,45  $s$ , resultando numa velocidade média de 113,22  $mm/s$ , mostrando a regulação da velocidade linear por meio dos campos vetoriais repulsivos.

De acordo com a Seção 4.2, as maiores velocidades se dão em espaços seguros, portanto, os caminhos  $A^*$  levaram o robô pelas regiões mais próximas das zonas seguras disponíveis no mapa. A terceira análise desta seção foi objeto de discussão do parágrafo anterior, a ação dos campos repulsivos na regulação da velocidade linear. Conforme relatado na Seção 5.3, página 62, uma diferença, e também vantagem, entre o controlador por campos vetoriais e os clássicos utilizados em controle de caminhos (*path following*) é, de fato, a regulação da velocidade linear por meio de informações do ambiente. Conforme argumentado, abordagens clássicas de controle não dispõem dessa característica, e portanto, não podem proteger o robô de obstáculos não previstos pelo caminho a ser seguido, enquanto os campos vetoriais podem.

Com os parâmetros  $\alpha$  e  $\beta$  relatados nesta seção, as velocidades linear e angular estão exibidas nas Figuras 6.19, 6.20, e 6.21, divididas em (a), (b), (c) e (d), mostrando os resultados individuais obtidos para os caminhos da Figura 6.17. As figuras rotuladas como (a) mostram as velocidades lineares obtidas com o passar do tempo, bem como a velocidade média  $v_{media}$ . As rotuladas como (b) exibem as velocidades angulares do robô com o passar do tempo. As figuras rotuladas como (c) mostram a visão em planta-baixa do caminho do robô, que aqui é representado por retângulos do tamanho do robô, na qual fica implícita a noção de velocidade do robô, por meio da densidade dos retângulos. Se em um trecho há uma alta densidade de retângulos, significa que o robô demorou para passar por lá, tendo, portanto, velocidade baixa.

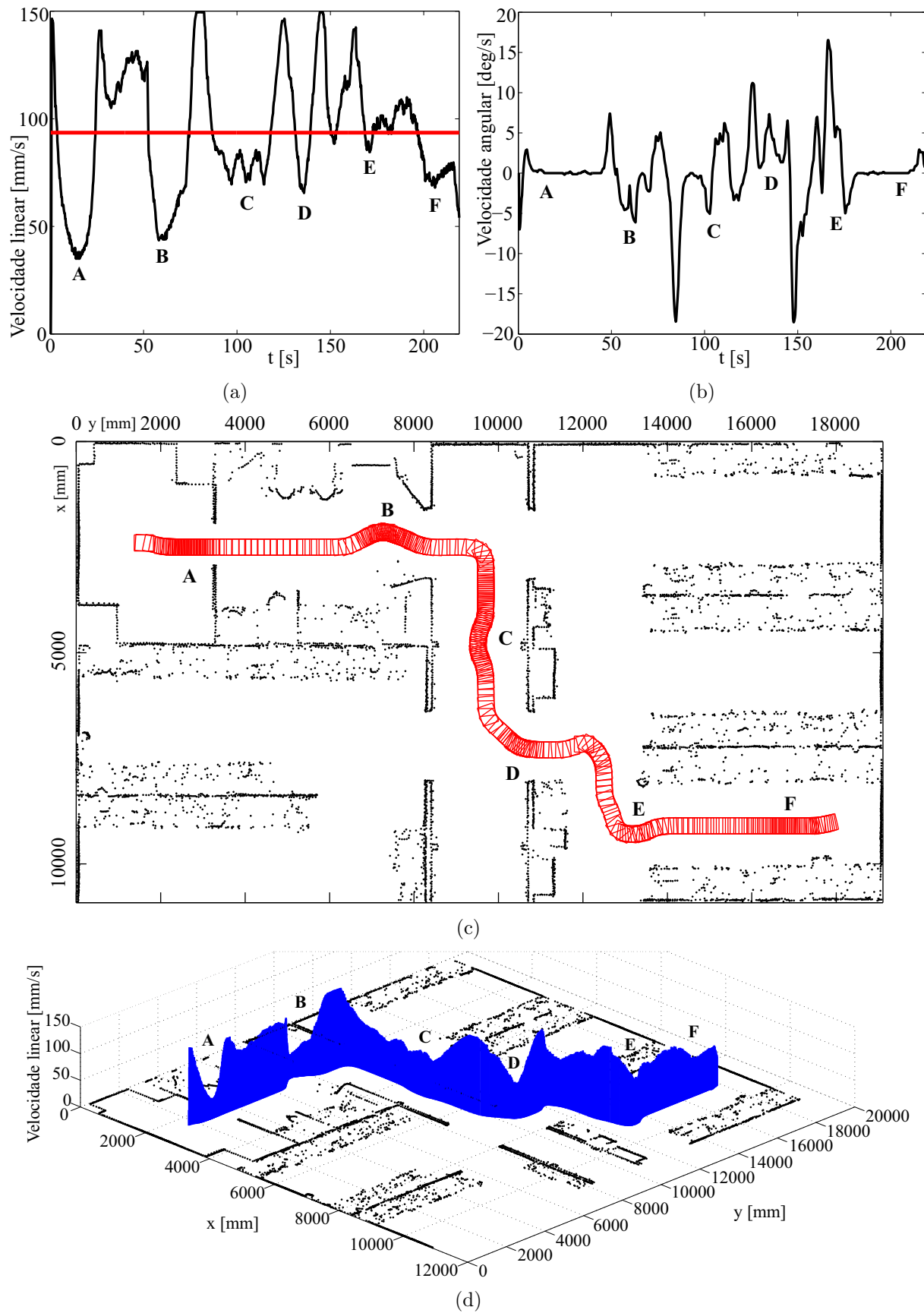


Figura 6.19: Velocidades para o caminho 1.



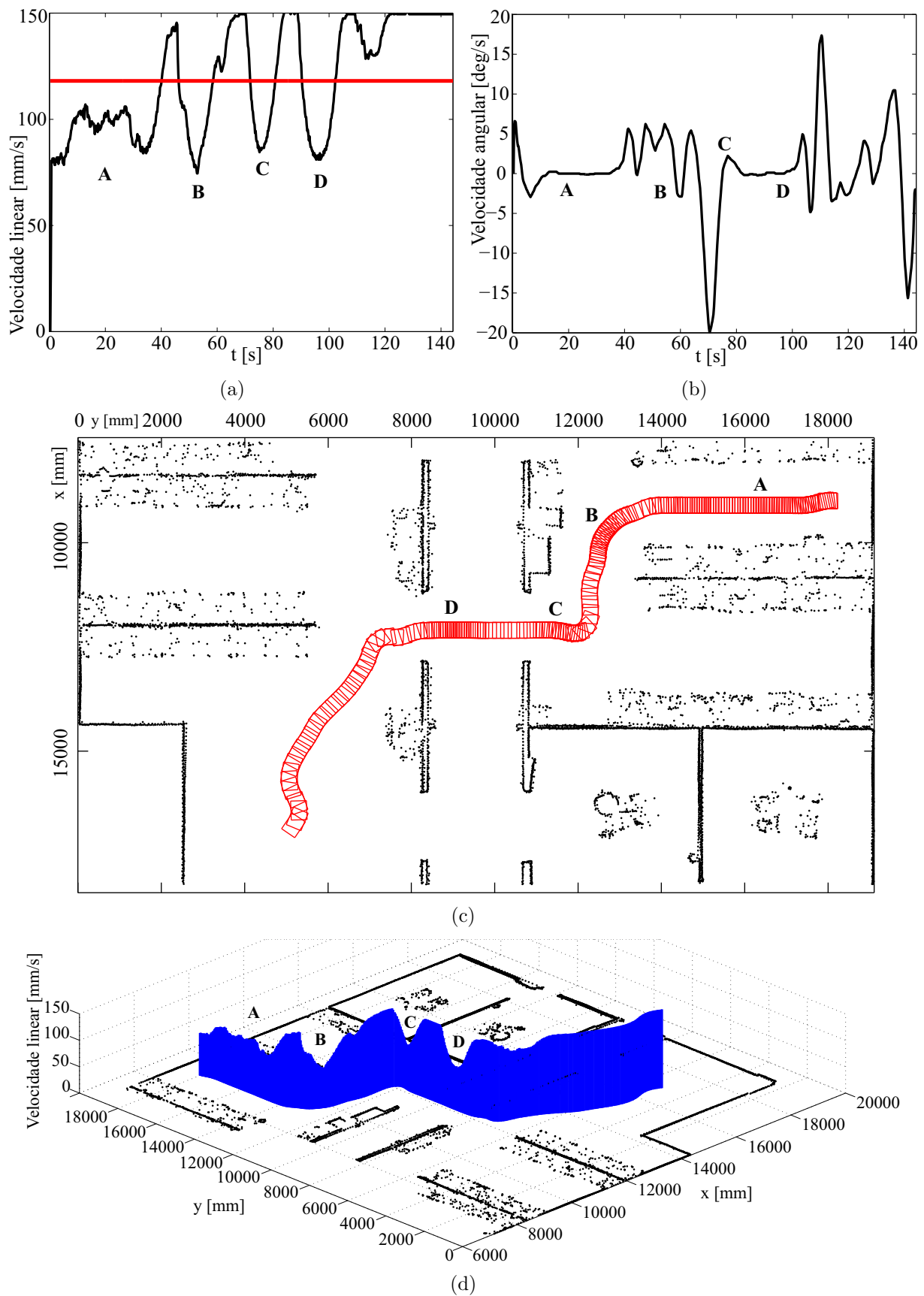


Figura 6.20: Velocidades para o caminho 2.

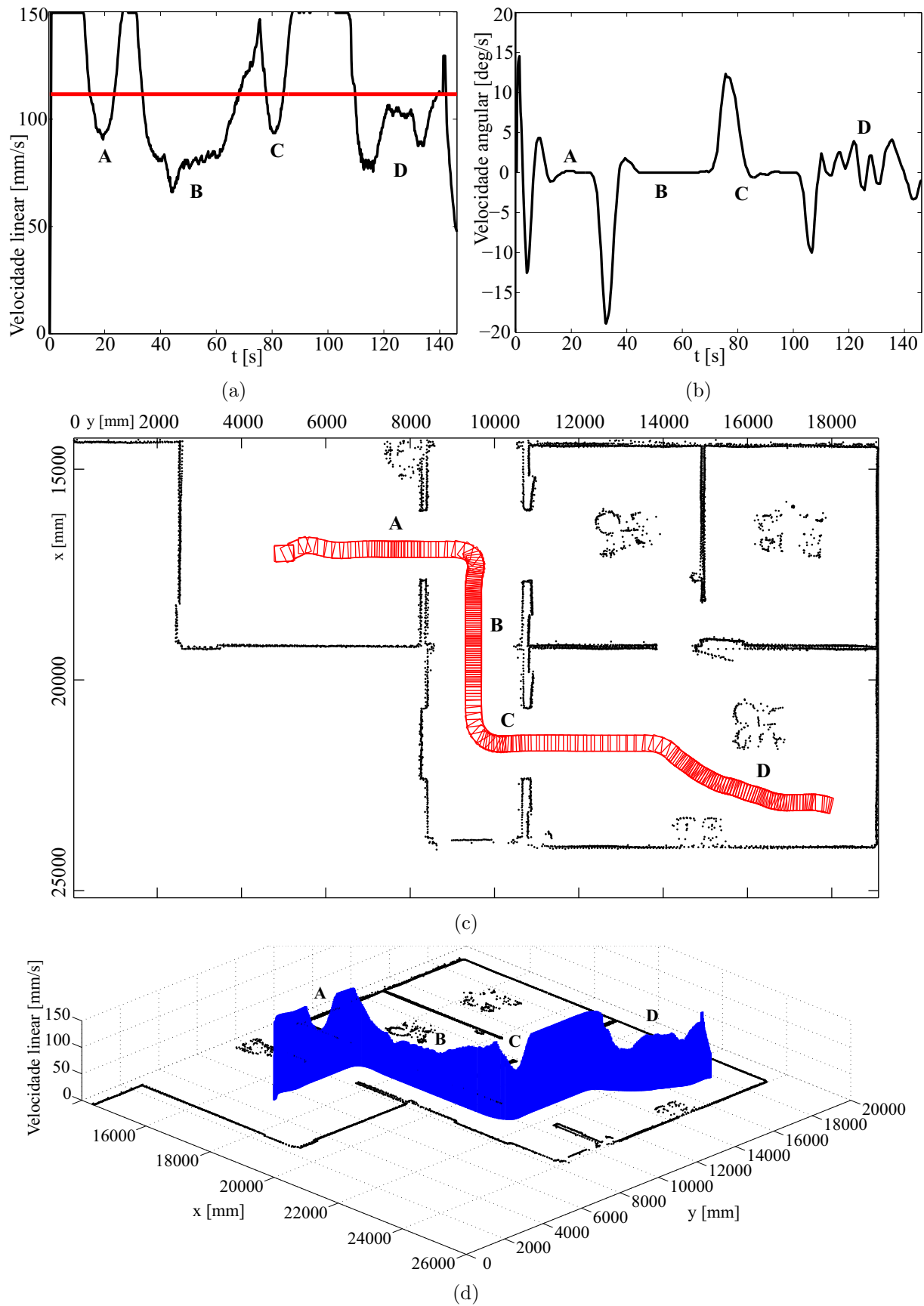


Figura 6.21: Velocidades para o caminho 3.

Se existem poucos retângulos numa região, denota-se que o robô passou rapidamente pelo trecho. As figuras rotuladas como (d) mostram os resultados combinados de (a) e (c), em três dimensões, em que as dimensões “x” e “y” mostram o caminho do robô, enquanto a dimensão “z” mostra a velocidade linear do robô para cada ponto do trajeto identificados nas figuras (a).

As Figuras 6.19, 6.20, e 6.21 destacam por letras maiúsculas as situações onde a velocidade linear fica abaixo da velocidade média. Estas situações denotam a proximidade de obstáculos, tais como passagem de portas ou proximidade de paredes. Conforme projetado, o controlador reduz a velocidade linear significativamente nessas situações, fazendo com que a passagem por esses trechos seja cautelosa, o que se reflete na redução da velocidade linear média.

Outro efeito diretamente ligado à variação da velocidade linear é a redução do esforço de controle angular. Uma vez que o robô desloca-se com menor velocidade em locais perigosos, demorando mais para atingir o próximo ponto-objetivo do caminho  $A^*$ , conseqüentemente o controle angular dispõe de mais tempo para ajustar a direção  $\phi$  do robô, tornando os comandos de giro mais suaves, trazendo benefícios de conforto e segurança para o usuário. Este fato pode ser comprovado por meio das Figuras 6.19(b), 6.20(b) e 6.21(b), onde vê-se que  $\omega$  jamais ultrapassou o valor de  $20^\circ/s$ , sendo esta uma velocidade angular baixa que, ainda assim, possibilitou um bom desempenho em termos de acurácia do controlador.

Esta seção apresentou a análise das estratégias de controle desenvolvidas no Capítulo 5, como o cálculo do caminho  $A^*$  e também o desempenho do controlador por campos vetoriais em conduzir o robô pelo caminho desejado. Discutiu-se a eficácia do método de forma qualitativa e quantitativa, mostrando diversos aspectos benéficos pelos quais conclui-se que o controlador autônomo desenvolvido comportou-se de maneira satisfatória na tarefa de transportar um usuário por meio de um caminho de maneira segura e acurada.

## 6.2.2 Comparação com controlador PID

Esta seção compara o desempenho do controlador por campos vetoriais com um controlador PID. Esse tipo de controlador é largamente utilizado nos mais diferentes tipos de aplicações da indústria e academia (Ogata 2004). Em robótica móvel, os controladores PID são populares em aplicações de controle de caminhos por serem de implementação simples e eficientes. Sendo assim, o controlador PID é, portanto, um excelente parâmetro de comparação para o desempenho do controlador autônomo por campos vetoriais.

A lei de controle designada para essa tarefa é relativa aos pontos dados pelo caminho  $A^*$ , para que este desenvolva o mesmo comportamento dado pela seção passada. Com dois pontos cartesianos, a pose atual do robô  $P_{x,y,\theta}$  e um ponto-objetivo, dado por  $G_{x,y}$ , um dos diversos pontos do caminho  $A^*$  calculado, aqui dado como a referência para o controlador, a velocidade linear  $v$  do robô é dada por

$$v = \begin{cases} v_{max} & \text{se } G_{x,y} \text{ não é o último ponto do caminho} \\ K_\rho \rho & \text{caso contrário} \end{cases} \quad (6.2)$$

onde  $K_\rho$  é uma constante a ser regulada e  $\rho = \sqrt{\Delta x^2 + \Delta y^2}$ , com  $\Delta x = G_x - P_x$  e  $\Delta y = G_y - P_y$ .

A velocidade angular é obtida por meio de

$$\omega = K_P e_t + K_I \sum_{k=0}^t e_k + K_D (e_t - e_{t-1}) \quad (6.3)$$

em que  $t$  denota o tempo atual, o erro é dado por  $e = \text{atan2}(\Delta y, \Delta x) - P_\theta$ , e os parâmetros do controlador são  $K_P$ ,  $K_I$  e  $K_D$  para, respectivamente, os termos proporcional, integral e derivativo, os quais devem ser calibrados. A calibração de um controlador PID possui várias abordagens disponíveis na literatura, relativas a determinados tipos de sistema.

A garantia da estabilidade desse controlador PID se dá pelo mesmo método de análise utilizado na Seção 5.4, página 63, uma vez que a lei de controle angular, mostrada no parágrafo anterior, tem como função zera o ângulo formado entre a frente do robô e o ponto-objetivo. O ângulo referido é, nesta seção, denotado por  $e$  (erro) para ficar em concordância com os termos clássicos utilizados no PID, sendo este o mesmo ângulo  $\phi$  do controlador por campos vetoriais.

Os parâmetros que proporcionaram o comportamento mais adequado do robô, tendo em vista a redução dos erros do robô ao caminho, foram  $K_p = 0,1$ ,  $K_P = 0,401$ ,  $K_I = 0,08$  e  $K_D = 0,07$ , obtidos por meio de calibração empírica, inclusive produzindo melhores resultados que os obtidos via Ziegler-Nichols, um método baseado em sensibilidade linear para obter parâmetros de controladores PID de maneira sistemática (Ogata 2004).

Assim, utiliza-se este controlador PID como critério para comparação de desempenho. Os erros aqui avaliados, conforme descrito anteriormente, são relativos à posição do robô com relação ao caminho  $A^*$ . Os resultados quantitativos dos experimentos são apresentados na Tabela 6.11 enquanto os resultados qualitativos acerca dos erros estão mostrados na Figura 6.22.

Tabela 6.11: Resultados para o controlador PID.

Caminho 1		Caminho 2		Caminho 3	
Parâmetro	Valor	Parâmetro	Valor	Parâmetro	Valor
$v_{max}$	150 mm/s	$v_{max}$	150 mm/s	$v_{max}$	150 mm/s
$v_{media}$	149,27 mm/s	$v_{media}$	149,14 mm/s	$v_{media}$	149,20 mm/s
$w_{max}$	52°/s	$w_{max}$	59°/s	$w_{max}$	83°/s
$\Delta S$	22308,78 mm	$\Delta S$	18240,83 mm	$\Delta S$	17462,07 mm
$\Delta t$	137,93 s	$\Delta t$	114,62 s	$\Delta t$	109,37 s
$\Delta E_{max}$	51,12 mm	$\Delta E_{max}$	29,36 mm	$\Delta E_{max}$	62,07 mm
$\Delta E_{medio}$	5,23 mm	$\Delta E_{medio}$	4,40 mm	$\Delta E_{medio}$	4,23 mm
$\sigma$	6,92 mm	$\sigma$	5,43 mm	$\sigma$	8,21 mm
$IC_{99\%}$	0,43 mm	$IC_{99\%}$	0,33 mm	$IC_{99\%}$	0,46 mm

Tem-se que a comparação dos controladores se dá pelas Tabelas 6.10 e 6.11. Nota-se que, em razão do controlador PID não possuir a mesma estratégia de campos repulsivos para controlar a velocidade linear, esta se mantém na velocidade máxima permitida, 150 mm/s, o tempo todo. Esse fato reflete-se na velocidade média,  $v_{media}$ , dos três experimentos, que se esteve próxima à velocidade máxima. Um efeito do alto valor de  $v_{media}$  são as altas velocidades angulares necessárias para controlar o robô pelo caminho, ou seja, alto esforço de controle.

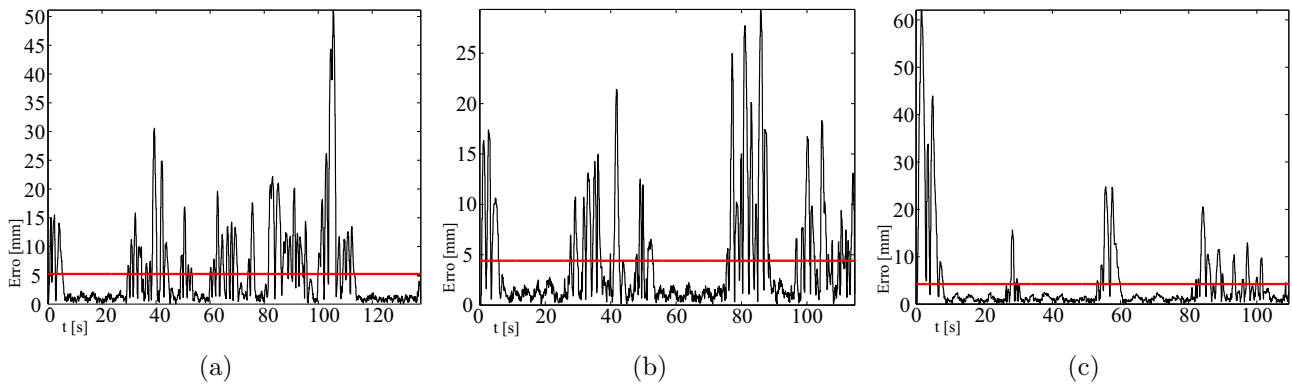


Figura 6.22: Erros do controlador PID ao navegar pelos caminhos (a) 1, (b) 2 e (c) 3 da Figura 6.17.

Conforme a tabela de resultados, a velocidade angular ( $\omega$ ) obtida pelo controlador PID foi maior que os resultados obtidos pelos campos vetoriais, com, respectivamente,  $52^\circ/s$ ,  $59^\circ/s$  e  $83^\circ/s$ . A evolução temporal das velocidades angulares dos três caminhos, mostradas na Figura 6.23, denotam alto esforço de controle, o que pode tornar desconfortável a navegação de um usuário, devido às altas acelerações/desacelerações impostas ao robô, e, dependendo da situação, pode inclusive oferecer riscos ao usuário no sentido de ejetá-lo do robô.

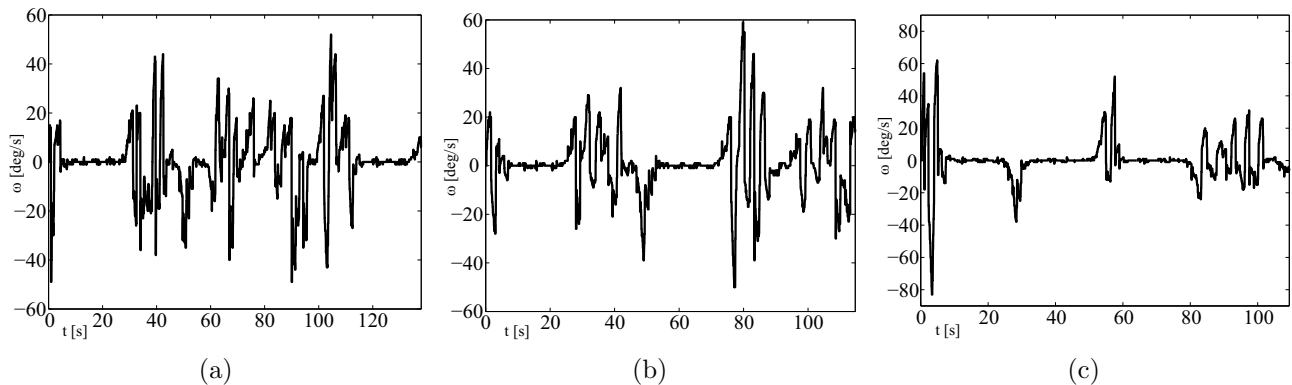


Figura 6.23: Velocidades angulares obtidas pelo controlador PID ao navegar pelos caminhos (a) 1, (b) 2 e (c) 3 da Figura 6.17.

Com relação aos resultados quantitativos, observa-se que o controlador por campos vetoriais obteve melhores desempenhos. O erro médio ( $\Delta E_{medio}$ ) do caminho 1 dos campos vetoriais obteve um ganho de 93,70%, mostrando que este permaneceu junto ao caminho com mais eficiência. Na Figura 6.17 percebe-se que o caminho 1 é o que possui a maior quantidade de curvas, sendo o que mais requisita ações de controle, oferecendo as maiores dificuldades. Por essa razão, o desempenho superior dos campos vetoriais frente ao controlador PID neste caminho revela maior confiabilidade do controlador proposto nesta tese.

Para o segundo caminho, o ganho dos erros médios dos campos vetoriais com relação ao PID foi de 45,21%. O segundo caminho é menor que o primeiro, e além disso, possui menos curvas, o que propiciou um melhor desempenho do PID, diminuindo a vantagem dos campos vetoriais sobre ele. Ainda assim, os campos vetoriais obtiveram melhores desempenhos quantitativos.

Nos resultados do terceiro caminho, os campos vetoriais conseguiram um ganho de 36,45% sobre os resultados do PID com relação ao erro médio. O grau de dificuldade do terceiro caminho é bastante parecido com o do segundo, conforme pode ser observado na Figura 6.18. Consequentemente, os níveis de ganho se mantiveram parecidos com o do segundo caminho, ainda assim, com superioridade dos campos vetoriais.

Os resultados para os outros parâmetros, tais como o erro máximo e desvio padrão também se mostraram superiores, com altos índices de ganho por parte do controlador por campos vetoriais. Com relação aos erros máximos ( $\Delta E_{max}$ ), os campos vetoriais obtiveram no primeiro caminho um ganho de vantagem de 140,11%, no segundo 52,52% e no terceiro 5,27%. Tem-se que, para o caminho 1, o ganho dos campos vetoriais com relação ao desvio padrão ( $\sigma$ ) foi de 89,59%, para o caminho 2 de 38,87% e para o caminho 3 de 35,26%. Esses resultados confirmam a argumentação anterior sobre as dificuldades maiores oferecidas pelo caminho 1, mantendo a coerência com relação aos dados numéricos.

Conforme mostrado, os resultados numéricos do controlador por campos vetoriais sobre o PID são significativos para o controle de trajetões. Mesmo com esses resultados positivos, ainda assim, leva-se em consideração que uma das maiores contribuições do controlador autônomo desenvolvido é, de fato, a proteção do robô por meio dos campos repulsivos. Métodos clássicos de controle, quando utilizados para controle de trajetões, não lidam com o problema de obstáculos não previstos, conforme mencionado em seções anteriores, uma vez que não possuem a realimentação das distâncias dos obstáculos, o que é feito nos campos vetoriais por definição.

Por essa razão, a proteção do robô contra obstáculos não previstos no caminho não é garantida por controladores como o PID. Neste quesito, outros controladores deveriam ser empregados, tais como os baseados em lógica nebulosa, mas provavelmente não teriam uma estrutura simples como a fornecida pelos campos vetoriais. De fato, esse tipo de problema requer a reestruturação desses controladores citados, inserindo novos fatores de análise, e aumentando sua complexidade. Os campos vetoriais já possuem essa proteção intrínseca, mantendo sua estrutura simples, podendo esta ser implementada em processadores de baixo poder computacional.

### 6.2.3 Análise do recálculo de rotas

Esta seção aborda o recálculo de rotas, tópico apresentado na Seção 5.5. Um obstáculo é adicionado ao ambiente sobre um caminho original calculado anteriormente. O caminho deste experimento é longo, indo de um extremo ao outro do mapa. A navegação percorre o caminho original até que a estratégia de manutenção de mapas, mencionada na Seção 5.5, detecta a presença de um obstáculo desconhecido no mapa. Esta detecção se dá com o decréscimo paulatino da velocidade linear do robô, chegando a zero a, aproximadamente, 1,38 m do obstáculo. Essa distância decorre diretamente dos parâmetros  $\alpha$  e, em especial  $\beta$ , cujos valores são os mesmos utilizados no experimento da Seção 6.2.1.

A partir da detecção e inclusão do novo obstáculo ao mapa, o diagrama de Voronoi para a área em torno do novo obstáculo é computado. Em seguida, a partir da posição atual do robô, o algoritmo  $A^*$  calcula uma nova rota até o ponto final, a qual o robô passa a seguir imediatamente. O resultado dessa estratégia é mostrado de maneira qualitativa na Figura 6.24, na qual os caminhos original e recalculado estão destacados.

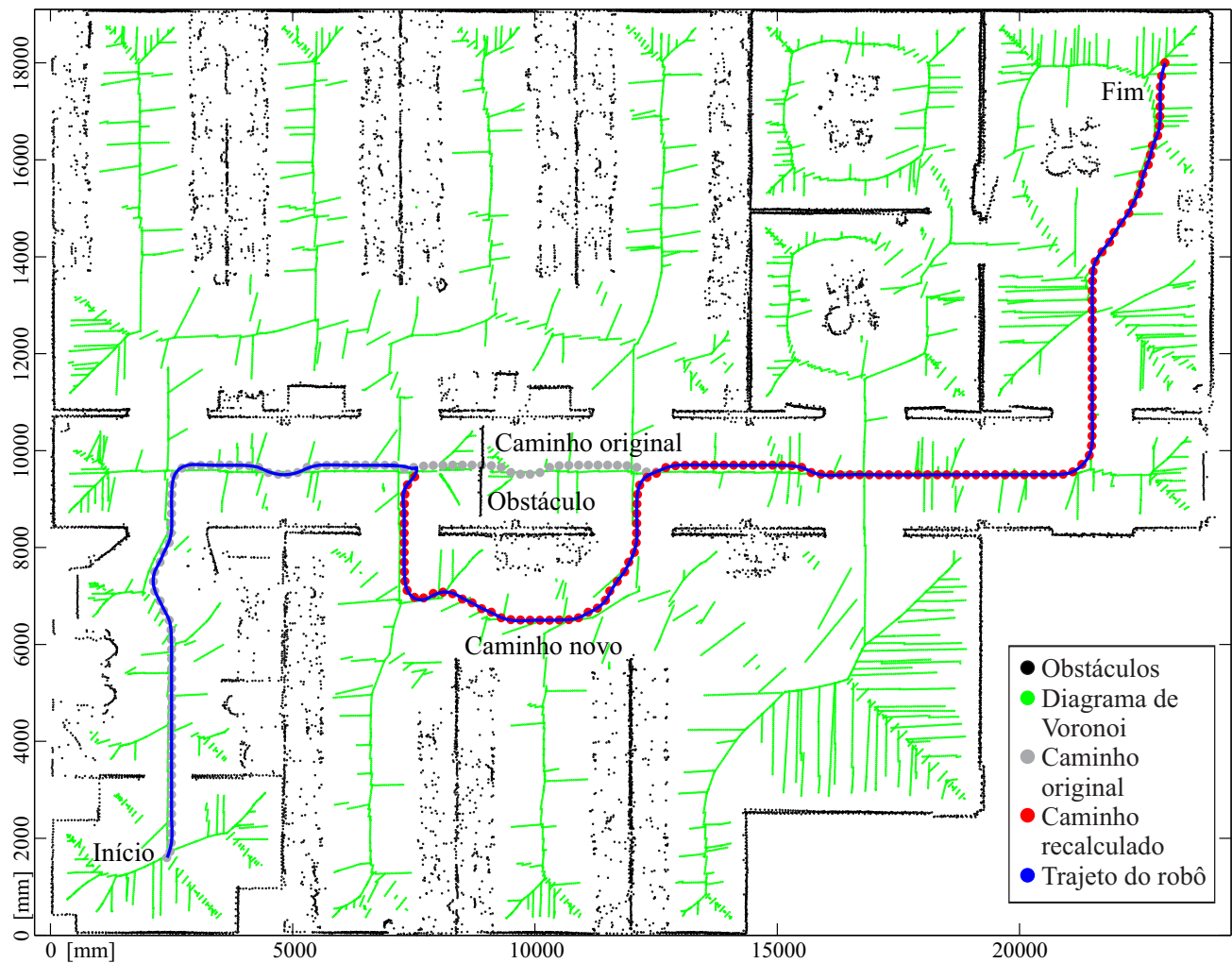


Figura 6.24: Recálculo de rotas com  $A^*$  dinâmico.

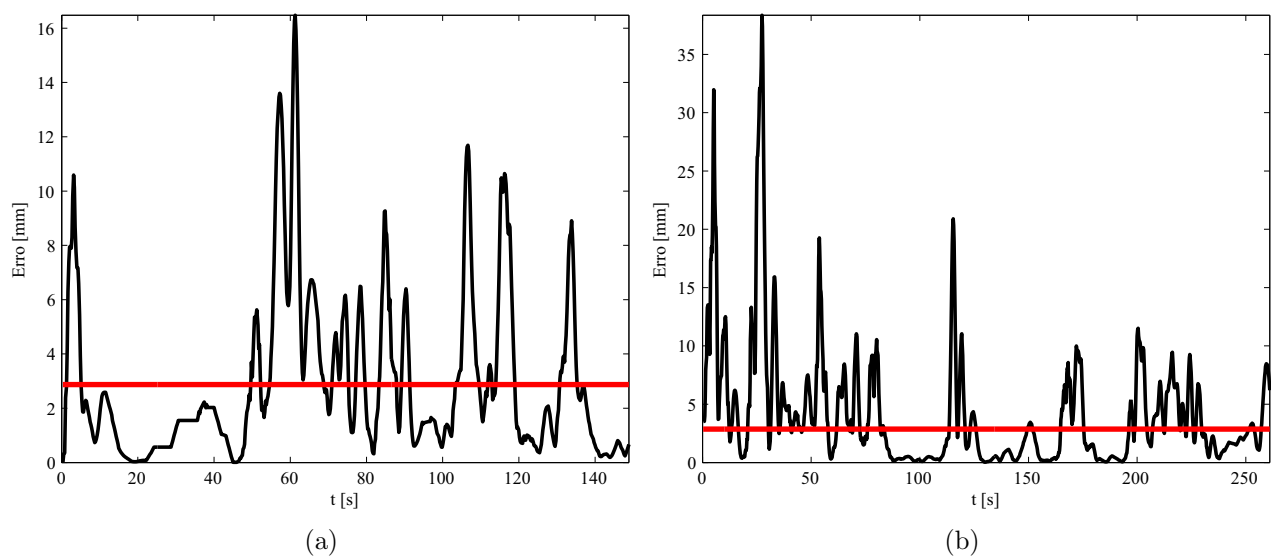


Figura 6.25: Erros produzidos pelo controlador autônomo para rotas (a) original e (b) recalculada.

Tabela 6.12: Desempenho do controlador para o caminho original.

Parâmetro	Valor	Parâmetro	Valor
$v_{max}$	150 $mm/s$	$\Delta E_{max}$	16,48 $mm$
$v_{media}$	89,22 $mm/s$	$\Delta E_{medio}$	2,87 $mm$
$w_{max}$	18,84°/s	$\Delta E_{min}$	0,00 $mm$
$\Delta S$	13281,37 $mm$	$\sigma$	3,09 $mm$
$\Delta t$	148,86 $s$	$IC_{99\%}$	0,30 $mm$

Tabela 6.13: Desempenho do controlador para o caminho recalculado.

Parâmetro	Valor	Parâmetro	Valor
$v_{max}$	150 $mm/s$	$\Delta E_{max}$	38,36 $mm$
$v_{media}$	106,02 $mm/s$	$\Delta E_{medio}$	3,45 $mm$
$w_{max}$	19,57°/s	$\Delta E_{min}$	0,00 $mm$
$\Delta S$	27670,19 $mm$	$\sigma$	5,29 $mm$
$\Delta t$	260,98 $s$	$IC_{99\%}$	0,39 $mm$

Observa-se, novamente, a efetividade da estratégia implementada para cálculo de caminhos seguros (analisados na seção anterior), que passam o mais longe possível dos obstáculos. Os dados quantitativos referentes ao experimento estão disponíveis na Figura 6.25 e nas Tabelas 6.12 e 6.13. Tem-se que os valores dos erros  $\Delta E$  mantiveram-se na mesma faixa de valores dos apresentados na seção anterior, reforçando a repetibilidade da ação conjunta do controlador autônomo com o FastSLAM. Com uma navegação totalizando 40951,56  $mm$  em 409,84  $s$  (seis minutos e cinquenta segundos), obteve-se uma velocidade média de 99,92  $mm/s$ , concordando com os resultados obtidos na Seção 6.2.1.

Um segundo caminho foi percorrido com uma configuração diferente do laboratório onde algumas portas foram fechadas impedindo o acesso do robô a certos ambientes. Um novo mapeamento para esta configuração foi realizado obtendo-se um mapa de, aproximadamente, 178  $m^2$ . Os resultados qualitativos estão na Figura 6.26, e os quantitativos na Figura 6.27 e nas Tabelas 6.14 e 6.15.

Tabela 6.14: Desempenho do controlador para o segundo caminho.

Parâmetro	Valor	Parâmetro	Valor
$v_{max}$	150 $mm/s$	$\Delta E_{max}$	21,26 $mm$
$v_{media}$	82,17 $mm/s$	$\Delta E_{medio}$	3,34 $mm$
$w_{max}$	11,02°/s	$\Delta E_{min}$	0,00 $mm$
$\Delta S$	6286,78 $mm$	$\sigma$	4,15 $mm$
$\Delta t$	76,51 $s$	$IC_{99\%}$	0,54 $mm$

Os resultados para o segundo caminho foram similares ao primeiro, conforme a Tabela 6.15. Os erros mantiveram-se no mesmo patamar, denotando a acurácia e repetibilidade da técnica de controle desenvolvida. Os caminhos combinados somam uma distância total de 25869,66  $mm$ , percorrida em 278,09  $s$ , produzindo uma velocidade média de 93,03  $mm/s$ .



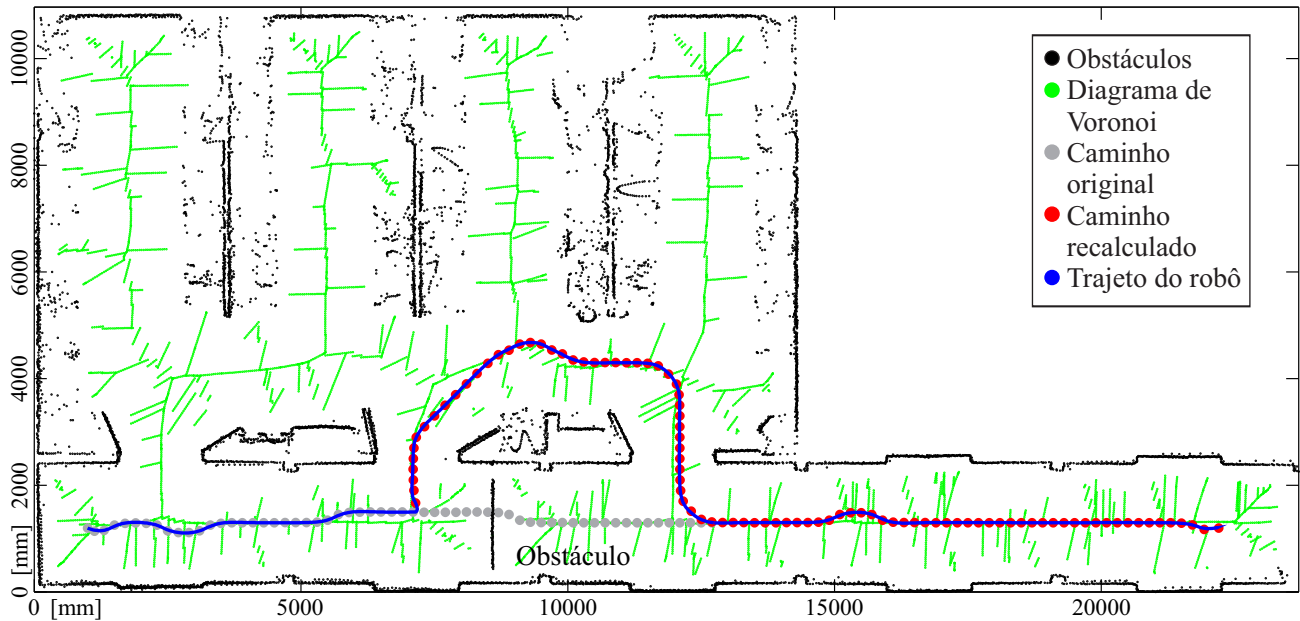


Figura 6.26: Recálculo de rotas com  $A^*$  dinâmico para o segundo caminho.

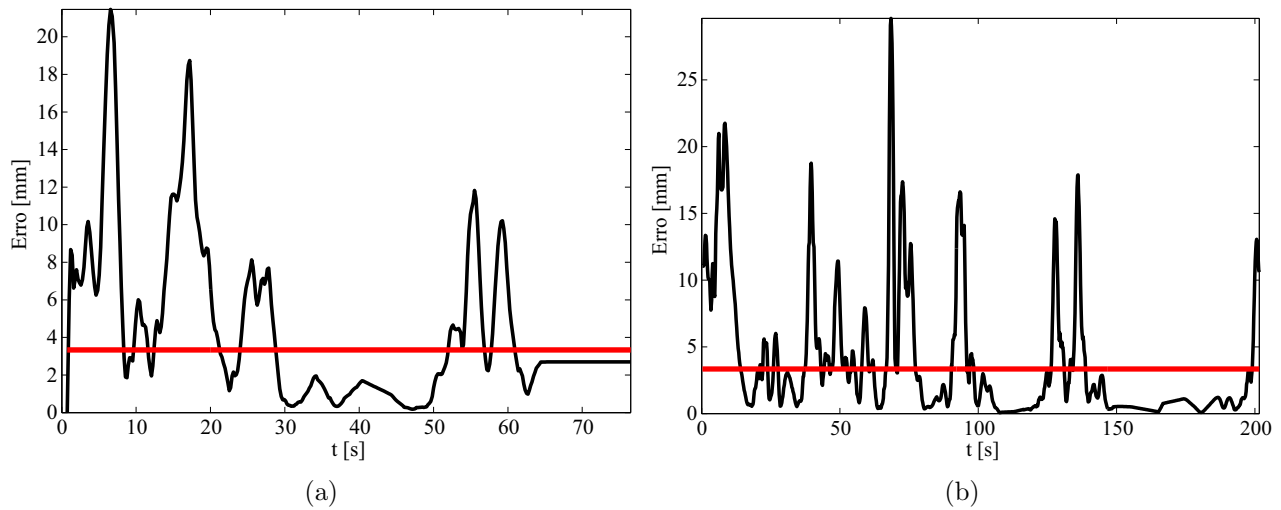


Figura 6.27: Erros produzidos pelo controlador autônomo para as rotas (a) original e (b) recalculada para o segundo caminho 2.

Tabela 6.15: Desempenho do controlador para o segundo caminho recalculado.

Parâmetro	Valor	Parâmetro	Valor
$v_{max}$	150 mm/s	$\Delta E_{max}$	29,61 mm
$v_{media}$	97,15 mm/s	$\Delta E_{medio}$	3,86 mm
$w_{max}$	22°/s	$\Delta E_{min}$	0,00 mm
$\Delta S$	19582,88 mm	$\sigma$	5,02 mm
$\Delta t$	201,58 s	$IC_{99\%}$	0,39 mm

A velocidade média anterior denota a ação dos campos vetoriais durante a navegação, reduzindo a velocidade linear em locais próximos a zonas perigosas. Os resultados obtidos ressaltam a segurança da navegação, com os caminhos recalculados passando sempre pelas zonas mais seguras do mapa, mostrando que a heurística desenvolvida para o algoritmo  $A^*$  é efetiva.

### 6.3 Avaliação das implementações do projeto DesTINE

O projeto DesTINE desenvolveu outros métodos para a utilização da plataforma robótica pelo usuário que não foram contemplados nesta tese. Conforme mostra a Figura 3.1, página 25, outras opções de controle podem ser escolhidas, como a “*Manual Go*” e a “*Point-to-Go*”, conforme citado anteriormente no Capítulo 3.

Esta seção mostra uma comparação entre os campos vetoriais e os outros métodos desenvolvidos exclusivamente para o projeto DesTINE por Rohmer (2014), mostrando as funcionalidades e os resultados entre os métodos “*Manual Go*”, “*Point-to-Go*”, e “*Autonomous Control*”. Todas as comparações foram feitas por meio do simulador V-REP (Freese 2014), e a HMI é controlada pelo Emotiv/sEMG, com quatro comandos: “*next*” (próximo/direita), “*previous*” (anterior/esquerda), “*validate*” (confirmação) e “*cancel*” (cancelamento). Os dados mostrados nesta seção são cortesia do Dr. Eric Rohmer.

A funcionalidade “*Manual Go*” implementa um controle (*joystick*) virtual para o usuário, disponibilizando quatro comandos: “para frente”, “para trás”, “para a esquerda” e “para a direita”. O método consiste em deslocar ou girar o robô por curtas distâncias e ângulos, a serem calibrados previamente. Neste método, a taxa de navegação por comando dado é baixa, exigindo do usuário diversas interações para navegar por grandes distâncias, sendo, neste caso, um método cansativo. No entanto, é eficaz para ajustar o robô em curtas distâncias.

O “*Point-to-Go*”, conforme descrito anteriormente na página 33, exibe uma imagem de câmera para o usuário, com diversos pontos alcançáveis. O usuário, utilizando a HMI, escolhe um dos pontos-objetivo exibidos via sEMG/BCI, e então o robô desloca-se autonomamente para o local. Ao alcançar o local, novas interações são necessárias para que o usuário continue navegando pelo ambiente até alcançar seu destino. Consideram-se para este método as mesmas observações feitas para o anterior.

A implementação do “*Autonomous Control*” não se trata do método por campos vetoriais apresentado nesta tese. O controlador utilizado nessa implementação é o do simulador V-REP (Freese 2014), o qual emprega um planejador de caminhos (*path planner*) topológico próprio da plataforma para calcular o caminho e um controlador PD (Proporcional-Derivativo) para controle de caminhos. Há também uma combinação desse controle autônomo com o trabalho de Raizer et al. (2013), em que sugestões são fornecidas ao usuário durante o trajeto, sendo esta mais uma fonte de comparação entre os métodos do projeto DesTINE.

A comparação envolve dois caminhos, dos quais o usuário sai de uma pose inicial  $P_{x,y,\theta}^1$ , desloca-se até uma pose intermediária  $P_{x,y,\theta}^2$  em outro cômodo e, a partir daí, deve navegar para a pose final  $P_{x,y,\theta}^3$  que está em outra sala do ambiente. Compara-se o número de interações, a distância percorrida e o tempo gasto para a navegação total. Os caminhos podem ser visualizados na Figura 6.28, e os dados quantitativos estão na Tabela 6.16.

Tabela 6.16: Resultados de comparação entre os métodos do projeto DesTINe.

	Controle Manual	Campos Vetoriais	Point to Go	Controle Autônomo	Autônomo com Sugestões
Total de interações	179	114	171	45	20
Next	26	26	67	15	7
Previous	16	12	29	1	3
Validate	131	66	47	23	8
Cancel	6	7	28	6	2
Distância total (m)	33,84	31,26	29,42	24,97	24,88
Tempo total (s)	624,3	367,6	452,9	302,9	298,7

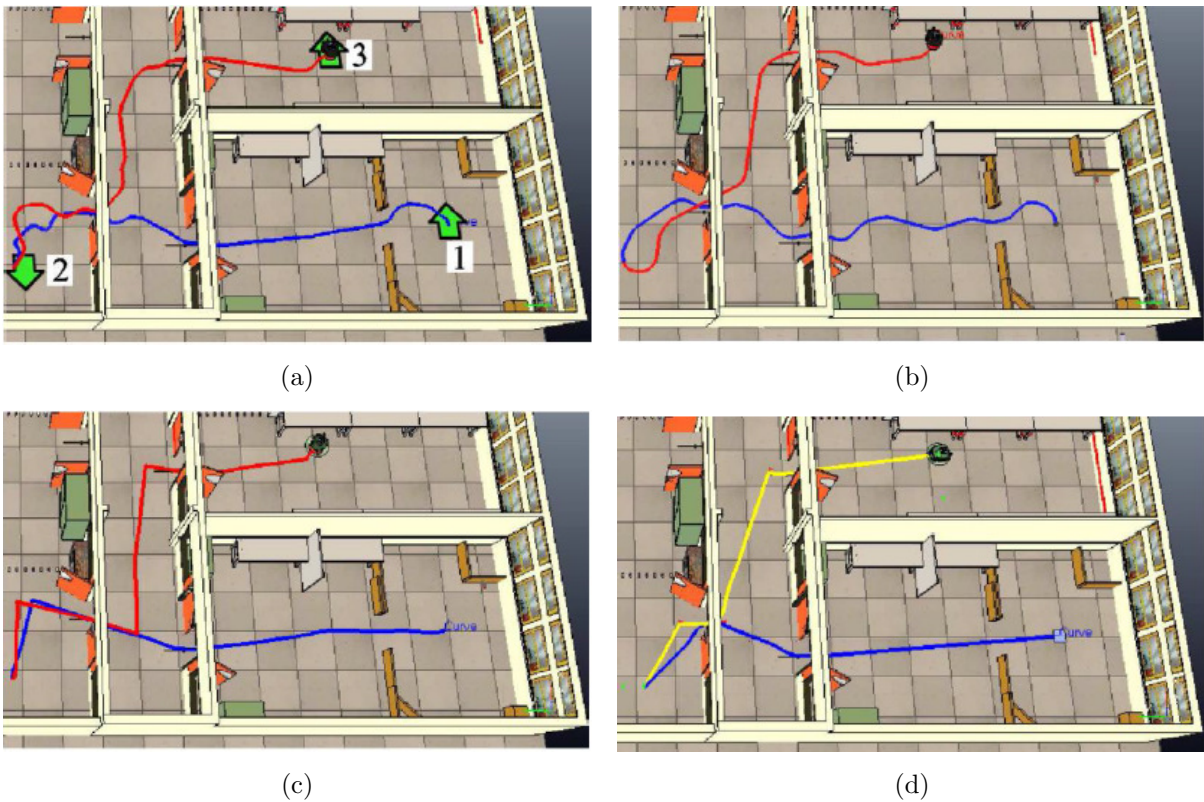


Figura 6.28: Rotas do (a) *Manual Go*, (b) *Campos Vetoriais*, (c) *Point-to-Go* e (d) *Autonomous Control* (Cortesia do Dr. Eric Rohmer).

Conforme percebe-se dos dados exibidos, os métodos que possuem os melhores resultados são os de navegação autônoma. Uma vez que não se tem a participação do usuário, o planejador de trajetórias calcula o menor caminho possível, o qual o robô segue por meio do controlador. Neste caso, as ações do usuário mostradas na Tabela 6.16 correspondem às ações do usuário em navegar pelos menus, escolher as localidades a serem alcançadas e ativar a navegação autônoma.

Percebe-se que a navegação autônoma acompanhada de sugestões ajuda ao usuário no sentido de diminuir a quantidade de interações com a HMI. O resultado exibido é, de fato, o esperado, uma vez que métodos de controle manuais, ou seja, com a participação ativa do usuário na navegação, tendem a ser menos eficientes, inserindo ruídos e erros de comandos no trajeto.

Dentre os métodos não autônomos, os melhores resultados foram obtidos pelo método dos campos vetoriais. No número total de interações, o este método obteve um ganho de 57,01% sobre o controle manual e de 50% com relação ao *Point-to-Go*. Com relação ao tempo de navegação entre os pontos-objetivo, os ganhos são de 69,83% sobre o controle manual e 23,20% sobre o *Point-to-Go*, ficando, inclusive, próximo aos tempos dos métodos autônomos.

## 6.4 Navegação em ambientes de grandes dimensões

Como última atividade, utilizou-se o robô Seekur Jr. controlado pelo sensor Emotiv para navegar por todo o segundo andar da Faculdade de Engenharia Elétrica e Computação (FEEC) da Universidade Estadual de Campinas (UNICAMP), o mesmo andar onde se encontra o Laboratório de Computação e Automação Industrial (LCA) de forma a utilizar os campos vetoriais para mapeá-lo. O resultado desta navegação final se encontra na Figura 6.29.

Para esta última tarefa, utilizou-se  $\theta_l = \theta_r = 45^\circ$ , e os mesmos parâmetros  $\alpha$  e  $\beta$  utilizados na tarefa de mapeamento do LCA, Figura 6.11, página 88. A escolha desses novos parâmetros para os ângulos de giro se deram em virtude do tamanho do ambiente, de forma a reduzir a quantidade de comandos de giro necessários para fazer as curvas inerentes ao ambiente e, conseqüentemente, reduzir o tempo de navegação total do experimento.

O robô navegou por 226,24 metros em 43,17 minutos, resultando em uma velocidade linear média de 87,34 *mm/s*. O mapa final, mostrado em detalhes na Figura 6.29, possui uma área útil, capturada pelos sensores do robô, de, aproximadamente, 742 *m*<sup>2</sup>. É possível observar algumas pequenas distorções no mapa, especialmente no corredor da sala de professores, onde nota-se que o alinhamento das paredes apresenta uma leve curvatura.

Alguns dos motivos plausíveis para a explicação dessas distorções são os fatos que o ambiente possui algumas portas de vidro, e outras com pinturas envernizadas, possuindo propriedades reflexivas, que inserem ruído extra na captura de distâncias pelo *laser*, além da própria deterioração da odometria, a qual o FastSLAM não pôde corrigir com precisão absoluta. No entanto, a deterioração da odometria, mesmo com um poderoso algoritmo de SLAM, é um resultado esperado quando o robô navega por vários ambientes novos sem revisitar os antigos, como foi o caso do mapeamento desta seção, uma vez que não se trata de um ambiente circular.

No entanto, em linhas gerais, considera-se que o mapa foi obtido com um nível adequado de precisão para cenários de robótica móvel assistiva. A navegação por campos vetoriais pode ser empregada em quaisquer tipos de ambientes, se mostrando especialmente útil naqueles ambientes que são previamente desconhecidos pelo sistema robótico. Mesmo que o robô não estivesse mapeando o ambiente, nem utilizando uma técnica de SLAM, ainda assim o usuário poderia navegar livremente pelo ambiente com segurança, pois o método dos campos vetoriais não depende de quaisquer outras aplicações para funcionar, conforme descrito anteriormente.

Com base nos experimentos descritos nesta seção, consideramos que o novo método de controle por campos vetoriais proposto nesta tese se mostrou eficiente, tanto na navegação assistida onde áreas distantes de obstáculos são privilegiadas, quanto na navegação autônoma onde a velocidade é regulada de acordo com o ambiente em torno do robô. Desta forma, acreditamos ter propiciado uma contribuição significativa para a área de robótica móvel assistiva.

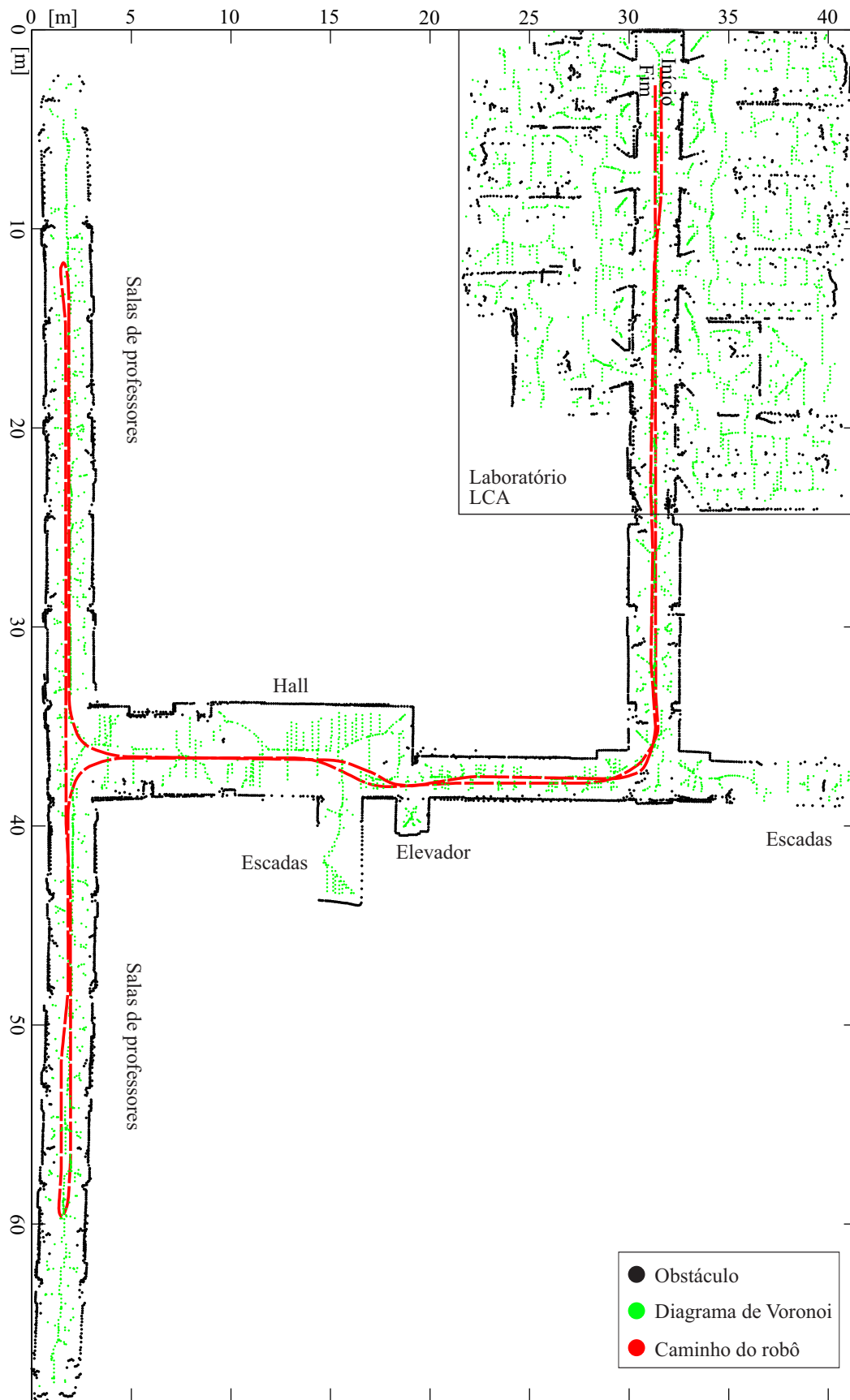


Figura 6.29: Mapeamento do segundo andar da FEEC/UNICAMP.



## Conclusões

Esta tese apresentou soluções para o controle de um veículo robótico assistivo cujo condutor possui severos impedimentos físicos. Para a interação com o usuário dispõe-se de uma HMI assistiva descrita no Capítulo 3. Esta interface processa sinais biológicos a partir dos quais são identificados comandos emitidos pelo usuário. No entanto, a aquisição e classificação dos comandos do usuário está sujeita a erros, os quais dificultam sobremaneira o controle da navegação. Para minimizar os efeitos destes erros, bem como proteger o usuário durante a navegação, foi desenvolvida uma técnica de navegação robótica onde a responsabilidade pela condução do veículo é dividida entre o condutor e o software embarcado no veículo. Estas técnicas são denominadas na literatura de controle compartilhado (*shared control*).

Desde meados dos anos 2000, diversas abordagens de controle compartilhado vêm sendo desenvolvidas. Algumas das principais técnicas foram apresentadas no Capítulo 2, inclusive para fins de comparação com o trabalho desenvolvido nesta pesquisa, onde foi proposta e validada uma nova técnica de controle compartilhado, a qual denominamos navegação assistida. Esta técnica se baseia no conceito de campos vetoriais computados a partir dos sensores instalados no veículo por meio de operações simples que podem ser realizadas em processadores de pequeno porte ou mesmo em microcontroladores.

O método proposto obteve êxito em cumprir os requisitos listados no decorrer dos capítulos, tais como minimizar os efeitos dos erros de classificação causados pelas HMIs assistivas e evitar colisões. Também possibilitou uma navegação totalmente controlada pelo usuário, dispensando todas as ações autônomas características das abordagens analisadas na revisão bibliográfica. Por último, o objetivo principal foi a proteção da integridade do usuário e seu veículo sem restringir a navegação como ocorre com técnicas que empregam ações autônomas.

Nessa nova abordagem, construíram-se comportamentos para o robô assistivo por meio de vetores de controle, os quais podem ser manipulados por campos vetoriais. Dessa forma, o robô pode moldar seu comportamento de acordo com as situações e dificuldades reveladas pelos campos vetoriais. O método de controle compartilhado por campos vetoriais apresentou resultados superiores os relatados na literatura, mostrando-se inclusive mais simples e de fácil implementação que os mais sofisticados métodos propostos até então. Os conceitos da estratégia, bem como sua maneira de implementação e resultados podem ser encontrados no Capítulo 4 e em Olivi et al. (2013) e Olivi et al. (2014).

Durante a navegação por controle compartilhado, onde o usuário se desloca por ambientes que podem ser desconhecidos, o mapeamento do ambiente é realizado com o auxílio de um método de SLAM, o FastSLAM 2.0. Este método produz mapas com excelentes níveis de detalhe e precisão. Estes mapas são importantes para a modalidade de navegação autônoma desenvolvida. A navegação autônoma alivia a carga de esforço do usuário, uma vez que a interação com HMIs assistivas pode ser cansativa com o uso prolongado.

O Capítulo 5 apresentou a abordagem proposta para navegação autônoma. Foi necessário atacar diversos aspectos concernentes à navegação autônoma por ambientes conhecidos, por exemplo, manutenção de mapas, planejamento e re-planejamento de rotas (*path planning*) e controle de caminhos (*path following*). Para o planejamento e re-planejamento de rotas foi empregado o algoritmo  $D^*$  (*Dynamic A\**) dotado de uma heurística que maximiza a segurança da rota. Para o controle de caminhos, o mesmo método de navegação por controle compartilhado baseado em campos vetoriais foi empregado. O método apresenta um desempenho ligeiramente superior ao controlador clássico PID no que tange à manutenção do caminho. Entretanto, uma vantagem significativa do controlador proposto é sua capacidade de ajustar a velocidade do robô de acordo com o ambiente, por exemplo, diminuindo a velocidade em passagens estreitas. Tal capacidade inexistente nos controladores clássicos de caminhos como o PID.

O Capítulo 6 foi dedicado à avaliação experimental das técnicas de navegação por controle compartilhado e autônoma proposta nesta tese. Procurou-se, sempre que possível, confrontar os resultados obtidos com os citados na literatura. Quando tais resultados não estão disponíveis, a técnica proposta foi confrontada com uma solução clássica, como no caso do controlador PID para controle de caminhos.

Por fim, conclui-se que o trabalho desenvolvido obteve resultados significativos na área de robótica móvel assistiva, fornecendo meios para a melhoria da mobilidade de pessoas com privação motora, de modo que estas recuperem algumas de suas capacidades de locomoção, tornem-se menos dependentes de terceiros, e assim adquiriram convivência em círculos sociais.

## Trabalhos Futuros

Em trabalhos futuros, pretende-se aprimorar cada uma das formas de navegação mostradas nessa pesquisa, a autônoma e a assistida. Para a navegação por controle compartilhado, é possível investigar novas formas de emanar campos dos ambientes, de forma a maximizar sua eficácia. O método abre possibilidades de otimização na utilização de vários de seus parâmetros, via utilização de técnicas de aprendizado de máquina, adaptativas, de inteligência computacional, dentre outras, fazendo, inclusive, com que a etapa de calibração dos parâmetros seja ainda mais adaptada ao usuário do sistema.

Com a navegação autônoma, é possível aparelhar e estruturar os ambientes de forma a facilitar o processo de localização e mapeamento do robô, como a utilização de identificação por radiofrequência (RFID, do inglês *Radio-Frequency IDentification*), de marcas visuais e reflexivas, de sensores de ultrassom, dentre outros.

Além dessas possibilidades, também é possível expandir as capacidades dos robôs via a interação com ambientes inteligentes dotados de sensores externos, redes de sensores sem fio, servidores e nós de alto poder de processamento, conforme relatado em Souza et al. (2013).



A expansão das capacidades do robô por meio de ambientes inteligentes levam a diversas situações interessantes, as quais podem vir a contribuir significativamente para as tecnologias assistivas, por exemplo, o fornecimento de mapas e rotas ao robô pelo próprio ambiente (Olivi et al. 2012), ou a execução de complexos algoritmos robóticos em nuvens computacionais.

Atualmente está em fase final de desenvolvimento a cadeira de rodas robótica desenvolvida na FEEC/UNICAMP, descrita brevemente na sequência.

## Cadeira de rodas robótica

A cadeira de rodas robótica foi desenvolvida a partir de uma cadeira motorizada comercial, o modelo SX do fabricante Freedom Veículos Elétricos Ltda. Esta cadeira vem equipada com um módulo de controle com *joystick* e módulo de potência que alimenta seus dois motores elétricos de corrente contínua. A energia é fornecida por duas baterias de 24V recarregáveis.

Neste projeto alguns requisitos importantes foram identificados. O primeiro requisito é o discutido por toda esta tese, a segurança da pessoa com deficiência. A cadeira robótica deve ser capaz de detectar degraus, desníveis e iminência de colisões, interrompendo a locomoção assim que uma situação de perigo é detectada. Tais ações devem ser processadas no nível mais baixo da hierarquia do software, ou seja, no nível de controle de locomoção.

O segundo requisito se refere ao consumo de energia. A instalação de processadores, sensores e periféricos certamente irão diminuir a autonomia da cadeira robótica pois drenarão energia da bateria para o seu funcionamento. Desta forma os processadores, sensores e periféricos devem ser selecionados visando um compromisso entre consumo de energia e utilidade.

O terceiro requisito se refere ao custo do produto final. A utilização de componentes de prateleira, muitos destes empregados em sistemas robóticos de baixo custo, podem reduzir consideravelmente o custo. Na montagem da cadeira de rodas robótica foram utilizados extensivamente tais componentes. Como parâmetro de preço, estimamos que o custo de uma cadeira de rodas robótica comercial não deva superar em uma vez e meia o custo de uma cadeira de rodas motorizada convencional.

O quarto requisito se refere à capacidade de configuração da cadeira de rodas robótica às limitações da pessoa com deficiência. A solução adotada foi permitir a incorporação à cadeira robótica de novos dispositivos de processamento como *notebooks*, *tablets* e *smartphones*. A interface humano-máquina executa nestes dispositivos e envia comandos para a cadeira utilizando um dos canais de interconexão nela presentes, dentre eles, rede *Ethernet* cabeada, rede *WiFi* e conexão *Bluetooth*. Assim, a cadeira de rodas robótica fica totalmente independente da interface humano-máquina assistiva que o usuário utiliza.

Finalmente, o quinto requisito diz respeito à capacidade de operação remota da cadeira por parte de um cuidador ou familiar do usuário. Vimos que as interfaces humano-máquina são cansativas com o uso prolongado. Na concepção da cadeira de rodas robótica prevemos a situação onde um cuidador ou familiar assuma o comando da cadeira por meio da *internet* ou de uma conexão direta com a cadeira robótica. Tal pessoa pode utilizar seu *smartphone* ou *tablet* para teleoperar a cadeira de qualquer lugar utilizando uma interface intuitiva, tal qual um *joystick* virtual. A pessoa no comando pode visualizar e eventualmente se comunicar com a pessoa com deficiência utilizando aplicativos de comunicação.

Para desenvolver todas essas atividades com interações via rede, a cadeira de rodas robótica desenvolvida possui capacidade para se conectar à internet utilizando seu próprio ponto de acesso *WiFi* ou uma conexão 3G/4G intermediada pelo *smartphone* do condutor, por exemplo. Essa abordagem, conforme mencionado anteriormente, propicia que a cadeira tenha suas capacidades expandidas enormemente.

Na montagem da cadeira de rodas robótica procurou-se não alterar a ergonomia da cadeira, mantendo-se todas as suas características originais tais como encosto de cabeça, apoio para os pés rebatíveis e os apoios de braço originais. Adicionalmente, foi instalada uma mesa rebatível para apoio de *notebook*, *tablet* ou *smartphone*. A única característica eliminada foi a dobragem da cadeira para transporte. Todos os dispositivos eletrônicos foram instalados na parte de trás da cadeira, exceto os controladores dos motores que foram instalados nas laterais da cadeira. O sensor principal, um medidor de distância a *laser* (*laser rangefinder*) foi instalado em base retrátil para não dificultar a acomodação do usuário. A cadeira de rodas robótica pode ser visualizada na Figura 7.1.



Figura 7.1: Vistas frontal, lateral e traseira da cadeira, mostrando seus vários dispositivos.

O *software* de interação entre os componentes da cadeira de rodas robótica consiste da plataforma RestThru (Cardozo et al. 2010, Souza et al. 2013), a mesma instalada nos robôs e simuladores utilizados nesta tese. Desta forma, todo o software desenvolvido nesta pesquisa de tese pode ser utilizado na cadeira robótica de imediato e sem qualquer alteração. De fato, a avaliação das técnicas de controle compartilhado e navegação autônoma propostos nesta tese se constitui em um trabalho em andamento e consequência direta da pesquisa aqui reportada.

# Bibliografia

- Aicardi, M., Casalino, G., Bicchi, A. & Balestrino, A. (1995). Closed loop steering of unicycle like vehicles via lyapunov techniques, *IEEE Robotics & Automation Magazine* **2**(1): 27–35.
- Andrews, J. R. & Hogan, N. (1983). Impedance control as a framework for implementing obstacle avoidance in a manipulator, *Control of Manufacturing Processes and Robotic Systems*, pp. 243–251.
- Bastos-Filho, T. F., Cheein, F. A., Müller, S. M. T., Celeste, W. C., Cruz, C. D. L., Cavaliere, D. C., Sarcinelli-Filho, M., Amaral, P. F. S., Perez, E., Soria, C. M. & Carelli, R. (2014). Towards a new modality-independent interface for a robotic wheelchair, *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **22**(3): 567–584.
- BCI2000 (2014). The bci2000 software, <http://www.schalklab.org/research/bci2000/>, acessado em 26-08-2014.
- Benevides, A. B., Bastos-Filho, T. F. & Sarcinelli-Filho, M. (2012). Pseudo-online classification of three mental tasks using kl divergence, *Journal of Medical and Biological Engineering* **6**(1): 411–416.
- Bi, L., Fan, X.-A. & Liu, Y. (2013). Eeg-based brain-controlled mobile robots a survey, **43**(2): 161–176.
- Bianchi, G. & Sorrentino, R. (2007). *Electronic Filter Simulation & Design*, 1st edn, McGraw-Hill Professional.
- Biswas, K., Mazumder, O. & Kundu, A. S. (2012). Multichannel fused emg based biofeedback system with virtual reality for gait rehabilitation, *4th International Conference on Intelligent Human Computer Interaction*, Kharagpur, India, pp. 1–6.
- Bourhis, G., Horn, O., Habert, O. & Pruski, A. (2001). An autonomous vehicle for people with motor disabilities, *IEEE Robotics and Automation Magazine* **8**(1): 20–28.
- Brockett, R. W. (1983). *Asymptotic Stability and Feedback Stabilization*, Birkhäuser Verlag, Boston, USA.

- Cardozo, E., Guimarães, E., Rocha, L., Souza, R., Paolieri, F. & Pinho, F. (2010). A platform for networked robotics, *International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, pp. 1000–1005.
- Carlson, T. & Demiris, Y. (2012). Collaborative control for a robotic wheelchair: Evaluation of performance, attention, and workload, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* **42**(3): 876–888.
- Celeste, W. C., Bastos-Filho, T. F., Sarcinelli-Filho, M., Cruz, C. D. L. & Carelli, R. (2013). A robust adaptive path-following controller for a robotic wheelchair, *Journal of Control, Automation and Electrical Systems* **24**(1): 397–408.
- Choi, K. & Cichocki, A. (2008). Control of a wheelchair by motor imagery in real time, *International Conference on Intelligent Robots and Systems*, Vol. 5326, Daejeon, South Korea, pp. 330–337.
- Cowan, R. E., Fregly, B. J., Boninger, M. L., Chan, L., Rodgers, M. M. & Reinkensmeyer, D. J. (2012). Recent trends in assistive technology for mobility, *Journal of NeuroEngineering and Rehabilitation* **9**(20).
- Crisman, J. & Cleary, M. (1998). Progress on the deictically controlled wheelchair, *Assistive Technology and Artificial Intelligence*, Springer-Verlag Berlin Heidelberg, New York, USA, pp. 137–148.
- Cruz, C. D. L., Celeste, W. C. & Bastos, T. F. (2011). A robust navigation system for robotic wheelchairs, *Control Engineering Practice* **19**(6): 575–590.
- Derry, M. & Argall, B. (2013). Automated doorway detection for assistive shared-control wheelchairs, *IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, pp. 1254–1259.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs, *Numerische Mathematik* **1**: 269–271.
- Dobrea, M.-C. & Dobrea, D. M. (2009). The selection of proper discriminative cognitive tasks - a necessary prerequisite in high-quality bci applications, *2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies*, Bratislava, Slovakia, pp. 1–6.
- Emotiv (2014). The emotiv epoc interface, <http://www.emotiv.com/>, acessado em 26-08-2014.
- Escolano, C., Antelis, J. & Minguez, J. (2009). Human brain-teleoperated robot between remote places, *IEEE International Conference on Robotics and Automation*, Kobe, Japan, pp. 4430–4437.
- Ferreira, A., Silva, R. L., Celeste, W. C., Bastos, T. F. & Filho, M. S. (2007). Human-machine interface based on muscular and brain signals applied to a robotic wheelchair, *Journal of Physics* **90**(1): 1541–1672.

- Freed, A., Chan, A. D. C., Lemaire, E. D. & Parush, A. (2011). Wearable emg analysis for rehabilitation (wear) - surface electromyography in clinical gait analysis, *IEEE International Workshop on Medical Measurements and Applications Proceedings*, pp. 601–604.
- Freese, M. (2014). V-rep: Virtual robot experimentation platform, <http://www.coppeliarobotics.com/>, acessado em 26-08-2014.
- Friedman, D., Cycowicz, Y. M. & Gaeta, H. (2001). The novelty p3: an event-related brain potential (erp) sign of the brain's evaluation of novelty, *Neuroscience and Biobehavioral Reviews* **25**(4): 355–373.
- Geng, T., Gan, J. Q. & Hu, H. (2010). A self-paced online bci for mobile robot control, *International Journal of Advanced Mechatronic Systems* **2**(1): 28–35.
- Graimann, B., Allison, B. Z. & Pfurtscheller, G. (2010). *Brain-Computer Interfaces: Revolutionizing Human-Computer Interaction*, 1st. edn, Springer-Verlag Berlin Heidelberg.
- G.tec (2014). g.tec medical engineering, <http://www.gtec.at/>, acessado em 26-08-2014.
- Hansen, P., Kringelbach, M. & Salmelin, R. (2010). *MEG: An Introduction to Methods*, 1st. edn, Oxford University Press.
- Hart, P. E., Nilsson, N. J. & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths, *IEEE Transactions on Systems Science and Cybernetics* **4**(2): 100–107.
- Haykin, S. S. (2001). *Redes Neurais: princípios e práticas*, 2nd. edn, Bookman Companhia.
- Huettel, S. A., Song, A. W. & McCarthy, G. (2009). *Functional Magnetic Resonance Imaging*, 2nd. edn, Massachusetts: Sinauer.
- IBGE (2010). Censo demográfico 2010, <http://censo2010.ibge.gov.br/>, acessado em 26-08-2014.
- Iturrate, I., Antelis, J. M., Kübler, A. & Minguez, J. (2009). A noninvasive brain-actuated wheelchair based on a p300 neurophysiological protocol and automated navigation, *IEEE Transactions On Robotics* **25**(3): 614–627.
- Karaman, S. & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning, *The International Journal of Robotics Research* **30**(7): 846–894.
- Katevas, N. (2001). *Mobile Robotics in Health Care Services*, 1st. edn, IOS Press.
- Khatib, O. (1985). Real-time obstacle avoidance for manipulators and mobile robots, *IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 500–505.

- Lee, J.-H., Lim, J.-H., Hwang, H.-J. & Im, C.-H. (2013). Development of a hybrid mental speller combining eeg-based brain-computer interface and webcam-based eye-tracking, *The 35th International Conference of the IEEE Engineering in Medicine and Biology Society*, Osaka, Japan, pp. 2240–2242.
- Leeb, R., Friedman, D., Müller-Putz, G. R., Scherer, R., Slater, M. & Pfurtscheller, G. (2007). Self-paced (asynchronous) bci control of a wheelchair in virtual environments: A case study with a tetraplegic, *Computational Intelligence and Neuroscience* pp. 1–8.
- Lerner, J., Wagner, D. & Zweig, K. A. (2009). *Algorithmics of Large and Complex Networks*, 1st edn, Springer.
- Levine, S. P., Bell, D. A., Jaros, L. A., Simpson, R. C., Koren, Y. & Borenstein, J. (1999). The navchair assistive wheelchair navigation system, *IEEE Transactions On Rehabilitation Engineering* **7**(4).
- Li, Q., Chen, W. & Wang, J. (2011). Dynamic shared control for human-wheelchair cooperation, *IEEE International Conference on Robotics and Automation*, Shanghai, China, pp. 4278–4283.
- Liberzon, D. (2003). *Switching in systems and control*, 1st edn, Birkhäuser Boston Inc.
- Lizarralde, F. C. (1998). *Stabilization of affine nonlinear control systems by a Newton type method*, PhD thesis, Universidade Federal do Rio de Janeiro (UFRJ/COPPE), Rio de Janeiro, Brasil.
- Luth, T., Ojdanic, D., Friman, O., Prenzel, O. & Graser, A. (2007). Low level control in a semi-autonomous rehabilitation robotic system via a braincomputer interface, *IEEE 10th International Conference on Rehabilitation Robotics*, pp. 721–728.
- Mandel, C., Luth, T., Laue, T., Rofer, T., Graser, A. & Krieg-Bruckner, B. (2009). Navigating a smart wheelchair with a brain-computer interface interpreting steady-state visual evoked potentials, *9th International Conference on Intelligent Data Engineering and Automated Learning*, St. Louis, USA, pp. 1118–1125.
- MathWorks (2014). Matlab: Matrix laboratory, <http://www.mathworks.com/>, acessado em 26-08-2014.
- Millán, J. d. R., Rupp, R., Müller-Putz, G. R., Murray-Smith, R., Giugliemma, C., Tangermann, M., Vidaurre, C., Cincotti, F., Kübler, A., Leeb, R., Neuper, C., Müller, K.-R. & Mattia, D. (2010). Combining brain-computer interfaces and assistive technologies: state-of-the-art and challenges, *Frontiers in Neuroscience* **161**(4).
- MobileRobots (2014). Mobile robot systems for research and development, <http://www.mobilerobots.com/>, acessado em 26-08-2014.

- Montemerlo, M., Thrun, S., Koller, D. & Wegbreit, B. (2002). Fastslam: A factored solution to the simultaneous localization and mapping problem, *The Association for the Advancement of Artificial Intelligence (AAAI) National Conference on Artificial Intelligence*, Association for the Advancement of Artificial Intelligence (AAAI), Edmonton, Canada, pp. 593–598.
- Morales, Y., Kallakuri, N., Shinozawa, K., Miyashita, T. & Hagita, N. (2013). Human-comfortable navigation for an autonomous robotic wheelchair, *IEEE International Conference on Intelligent Robots and Systems*, Tokyo, Japan, pp. 2737–2743.
- Moravec, H. & Elfes, A. E. (1985). High resolution maps from wide angle sonar, *IEEE International Conference on Robotics and Automation*, St. Louis, USA, pp. 116–121.
- Müller-Putz, G. R., Scherer, R., Pfurtscheller, G. & Rupp, R. (2006). Brain-computer interfaces for control of neuroprostheses: from synchronous to asynchronous mode of operation, *Biomedizinische Technik* **51**(2): 57–63.
- Müller, S., Bastos-Filho, T. F. & Sarcinelli-Filho, M. (2013). Proposal of a ssvep-bci to command a robotic wheelchair, *Journal of Control, Automation and Electrical Systems* **24**(1): 97–105.
- Nisbet, P., Craig, J., Odor, P. & Aitken, S. (1996). Smart wheelchairs for mobility training, *Technology and Disability* **5**(1): 49–62.
- Ogata, K. (2004). *Modern Control Engineering*, 4th edn, Pearson and Prentice Hall.
- Okabe, A., Boots, B., Sugihara, K. & Chiu, S. N. (2000). *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, 2nd edn, Wiley.
- Olivi, L., Souza, R., Paolieri, F., Guimaraes, E. & Cardozo, E. (2012). A distributed navigation strategy for mobile robots based on wireless sensor networks, *XVIII Brazilian Conference on Automation*, Campina Grande, PB, Brasil, pp. 1–8.
- Olivi, L., Souza, R., Rohmer, E. & Cardozo, E. (2013). Shared control for assistive mobile robots based on vector fields, *The 10th International Conference on Ubiquitous Robots and Ambient Intelligence*, Jeju, South Korea, pp. 96–101. Received the conference’s main prize **Best Paper Award**.
- Olivi, L., Souza, R., Rohmer, E. & Cardozo, E. (2014). Método e sistema de movimentação de um elemento eletromecânico, *Technical Report : Patent number BR 10 2014 018508 9*, Universidade Estadual de Campinas (UNICAMP), Campinas, Brasil. Patente depositada no Instituto Nacional de Propriedade Intelectual (INPI).
- Park, C., Looney, D., ur Rehman, N., Ahrabian, A. & Mandic, D. P. (2013). Classification of motor imagery bci using multivariate empirical mode decomposition, *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **21**(1): 10–22.
- Pedrycz, W. & Gomide, F. (2007). *Fuzzy Systems Engineering: Toward Human-Centric Computing*, 1st edn, Wiley-IEEE Press.

- Philips, J., Millán, J., Vanacker, G., Lew, E., Galán, F., Ferrez, P. W., Brussel, H. V. & Nuttin, M. (2007). Adaptive shared control of a brain-actuated simulated wheelchair, *IEEE 10th International Conference on Rehabilitation Robotics*, Noordwijk, Netherlands, pp. 408–414.
- Postelnicu, C.-C. & Talaba, D. (2013). P300-based brain-neuronal computer interaction for spelling applications, *IEEE Transactions on Biomedical Engineering* **60**(2): 534–543.
- Raizer, K., Rohmer, E., Paraense, A. L. O. & Gudwin, R. R. (2013). Effects of behavior network as a suggestion system to assist bci users, *IEEE Symposium on Computational Intelligence in Rehabilitation and Assistive Technologies*, Singapore, pp. 40–47.
- Ramsey, N. F., van de Heuvel, M. P., Kho, K. H. & Leijten, F. S. S. (2006). Towards human bci applications based on cognitive brain systems: an investigation of neural signals recorded from the dorsolateral prefrontal cortex, *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **14**(2): 214–217.
- Rebsamen, B., Teo, C. L., Zeng, Q., Ang, M. H., Burdet, E., Guan, C. & Laugier, H. Z. C. (2007). Controlling a wheelchair indoors using thought, *IEEE Intelligent Systems* **22**(2): 18–24.
- Rohmer, E. (2014). Environments for supporting the navigation of assistive mobile robots, *Technical Report 2011/091137*, Department of Computer Engineering and Automation, State University of Campinas, Campinas, Brazil. Tech Report for Post-Doctoral activity to Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPES).
- Russell, S. & Norvig, P. (2009). *Artificial Intelligence: A Modern Approach*, 3rd edn, Prentice Hall.
- Sanei, S. & Chambers, J. (2007). *EEG signal processing*, 1st edn, Wiley-Interscience.
- Schalk, G., McFarland, D. J., Hinterberger, T., Birbaumer, N. & Wolpaw, J. R. (2004). Bci2000: a general-purpose brain-computer interface (bci) system, *IEEE Transactions on Biomedical Engineering* **51**(6): 1034–1043.
- Serby, H., Yom-Tov, E. & Inbar, G. F. (2005). An improved p300-based brain-computer interface, *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **13**(1): 89–98.
- SICK (2014). Sick group: Sensor intelligence, <http://www.sick.com/>, acessado em 26-08-2014.
- Siegwart, R. & Nourbakhsh, I. R. (2004). *Introduction to Autonomous Mobile Robots*, 1st edn, The MIT Press.
- Siesler, H. W., Ozaki, Y., Kawata, S. & Heise, H. M. (2002). *Near-Infrared Spectroscopy: Principles, Instruments, Applications*, 1st. edn, Wiley-VCH.
- Silva, A. N., Morère, Y., Naves, E. L. M., Sá, A. A. R. & Soares, A. B. (2013). Virtual electric wheelchair controlled by electromyographic signals, *Biosignals and Biorobotics Conference*, Rio de Janeiro, Brazil, pp. 1–5.



- Simpson, R. & Levine, S. (1997). Development and evaluation of voice control for a smart wheelchair, *XX Annual Conference on Rehabilitation Engineering*, Pittsburgh, USA, pp. 417–419.
- Snell, R. S. & Lemp, M. A. (1997). *Clinical Anatomy of the Eye*, 2nd. edn, Wiley-Blackwell.
- Somerset, V. S. (2010). *Intelligent and Biosensors*, 1st edn, InTech Books.
- Soriano, D., Silva, E. L., Slenes, G. F., Lima, F. O., Uribe, L. F. S., Coelho, G. P., Rohmer, E., Venancio, T. D., Beltramini, G. C., Campos, B. M., Anjos, C. A. S., Suyama, R., Li, L. M., Castellano, G. & Attux, R. (2013). Music versus motor imagery for bci systems a study using fmri and eeg: Preliminary results, *Biosignals and Biorobotics Conference*, Rio de Janeiro, Brazil, pp. 1–6.
- Souza, R., Pinho, F., Olivi, L. & Cardozo, E. (2013). A restful platform for networked robotics, *The 10th International Conference on Ubiquitous Robots and Ambient Intelligence*, Jeju, South Korea, pp. 423–428.
- Stentz, A. (1994). Optimal and efficient path planning for partially-known environments, *IEEE International Conference on Robotics and Automation*, Vol. 10, San Diego, USA, pp. 3310–3317.
- Tanaka, K., Matsunaga, K. & Wang, H. O. (2005). Electroencephalogram-based control of an electric wheelchair, *IEEE Transactions on Robotics* **21**(4): 762–766.
- Teixeira, F. V., Souza, R. S., Olivi, L., Rohmer, E., Guimarães, E. G. & Cardozo, E. (2012). Infraestrutura de rede de sensores sem fio para ambientes assistivos, *XVIII Brazilian Conference on Automation*, Campina Grande, PB, Brasil, pp. 1–8.
- Thrun, S. (2002). Particle filters in robotics, *The 17th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, Edmonton, Canada.
- Thrun, S., Burgard, W. & Fox, D. (2005). *Probabilistic Robotics*, 1st edn, The MIT Press.
- Tonin, L., Leeb, R., Tavella, M., Perdikis, S. & Millán, J. d. R. (2010). The role of shared-control in bci-based telepresence, *IEEE International Conference on Systems, Man, and Cybernetics*, Istanbul, Turkey, pp. 1462–1466.
- Trieu, H. T., Nguyen, H. T. & Willey, K. (2008). Shared control strategies for obstacle avoidance tasks in an intelligent wheelchair, *IEEE International Conference of Engineering in Medicine and Biology Society*, Vancouver, Canada, pp. 4254–4257.
- Uribe, L. F. S. (2013). *Desenvolvimento de uma interface homem-máquina baseada em potenciais evocados visuais em regime estacionário*, Dissertação de Mestrado, Universidade Estadual de Campinas (UNICAMP), Campinas, Brasil.

- Vanacker, G., Millán, J. D. R., Lew, E., Ferrez, P. W., Moles, F. G., Philips, J., Brussel, H. V. & Nuttin, M. (2007). Context-based filtering for assisted brain-actuated wheelchair driving, *Computational Intelligence and Neuroscience* **7**: 1–12.
- Vieira, F. C. (2005). *Controle dinâmico de robôs móveis com acionamento diferencial*, Dissertação de Mestrado, Universidade Federal do Rio Grande do Norte (UFRN), Natal, Brasil.
- Wolpaw, J. R., Birbaumer, N., Heetderks, W. J., McFarland, D. J., Peckham, P. H., Schalk, G., Donchin, E., Quatrano, L. A., Robinson, C. J., & Vaughan, T. M. (2000). Brain-computer interface technology: a review of the first international meeting, *IEEE Transactions on Rehabilitation Engineering* **8**(2): 164–173.
- Wolpaw, J. & Wolpaw, E. W. (2012). *Brain-Computer Interfaces: Principles and Practice*, 1st. edn, Oxford University Press.
- Yanco, H. A. (2001). Development and testing of a robotic wheelchair system for outdoor navigation, *Conference of the Rehabilitation Engineering and Assistive Technology Society of North America*, Falls Church, USA, pp. 1–3.
- Zal, F., Chen, T.-S., Chi, S.-W. & Kuo, C.-H. (2013). Fuzzy controller based subsumption behavior architecture for autonomous robotic wheelchair, *International Conference on Advanced Robotics and Intelligent Systems*, Tainan, Taiwan, pp. 158–163.

## Contribuições do autor

### Capítulo de livro internacional

Rocha, L. A., **Olivi, L. R.**, Paolieri, F., Feliciano, G., Souza, R. S., Rodrigues, D., Pinho, F., Teixeira, F., Guimarães, E. G., Cardozo, E. (2011) A Cloud Architecture For Educational Enterprises: A Case Study In Robotics. In: Mahmood, Zaigham, Hill, Richard. (Org.). *Cloud Computing For Enterprise Architectures*. 1ed. New York: Springer Verlag, V. 1, P. 135-157.

### Patente

**Olivi, L.**, Souza, R., Rohmer, E., Cardozo, E. (2014) Método e sistema de movimentação de um elemento eletromecânico, *Technical Report: Patent number BR 10 2014 018508 9*, Universidade Estadual de Campinas (UNICAMP), Campinas, Brasil. Patente depositada no Instituto Nacional de Propriedade Intelectual (INPI).

### Trabalhos completos em anais de congressos nacionais e internacionais

**Olivi, L.**, Souza, R., Rohmer, E., Cardozo, E. (2013) Shared Control For Assistive Mobile Robots Based On Vector Fields. In: *The 10th International Conference On Ubiquitous Robots And Ambient Intelligence*, Seoul: Korea Robotics Society.

Souza, R., Pinho, F., **Olivi, L.**, Cardozo, E. (2013) A Restful Platform For Networked Robotics. In: *The 10th International Conference On Ubiquitous Robots And Ambient Intelligence*, Seoul: Korea Robotics Society.

Amaya, J., Souza, R., **Olivi, L.**, Cardozo, E. (2013) Control Of Assistive Mobile Robots Via Seng Signals. In: *XI Simpósio Brasileiro De Automação Inteligente (SBAI 2013)*, Fortaleza.

**Olivi, L.**, Souza, R., Paolieri, F., Guimarães, E., Cardozo, E. (2012) A Distributed Navigation Strategy For Mobile Robots Based On Wireless Sensor Networks. In: *XIX Congresso Brasileiro De Automática (CBA 2012)* Campina Grande.

Rodrigues, D., **Olivi, L.**, Souza, R., Guimarães, E., Cardozo, E. (2012) Continuous Topology Learning And Early Recognition For Mobile Robots Navigation. In: *XIX Congresso Brasileiro De Automática (CBA 2012)* Campina Grande.

Teixeira, F. V., Souza, R. S., **Olivi, L.**, Rohmer, E., Guimarães, E. G., Cardozo, E. (2012) Infraestrutura De Rede De Sensores Sem Fio Para Ambientes Assistivos. In: *XIX Congresso Brasileiro De Automática (CBA 2012)* Campina Grande.

Agostinho, L., Souza, R., Paolieri, F., **Olivi, L.**, Feliciano, G., Teixeira, F., Rodrigues, D., Guimarães, E., Cardozo, E. (2011) Um Modelo Federado Em Cloud Computing Para Infraestruturas Robóticas. In: *Quarto Encontro Dos Alunos E Docentes Do Departamento De Engenharia De Computação E Automação Industrial (IV Eadca)*, Campinas. V. 1. P. 52-55.

Souza, R. S., Agostinho, L., Teixeira, F., Rodrigues, D., **Olivi, L.**, Guimarães, E. G., Cardozo, E. (2011) Control Of Mobile Robots Through Wireless Sensor Networks. In: *XXIX Simpósio Brasileiro De Redes De Computadores E Sistemas Distribuídos (XXIX SBRC)*, Campo Grande, 2011. V. 1. P. 805-818.

Feliciano, G., Agostinho, L., **Olivi, L.**, Guimarães, E. G., Cardozo, E. (2011) Uma Arquitetura Para Gerência De Identidades Em Nuvens Híbridas. In: *IX Workshop Em Clouds, Grids E Aplicações (IX WCGA)*, Campo Grande, V. 1. P. 15-28.

Agostinho, Lucio, **Olivi, Leonardo**, Feliciano, Guilherme, Paolieri, Fernando, Rodrigues, Diego, Cardozo, Eleri, Guimarães, Eliane (2011) A Cloud Computing Environment For Supporting Networked Robotics Applications. In: *2011 IEEE 9th International Conference On Dependable, Autonomic And Secure Computing (IX DASC)*, Sydney, p. 1110-1116.

Agostinho, Lucio, Feliciano, Guilherme, **Olivi, Leonardo**, Cardozo, Eleri, Guimarães, Eliane (2011) A Bio-Inspired Approach To Provisioning Of Virtual Resources In Federated Clouds. In: *2011 IEEE 9th International Conference On Dependable, Autonomic And Secure Computing (IX DASC)*, Sydney, p. 598-604.

## Prêmio

O trabalho *Shared Control For Assistive Mobile Robots Based On Vector Fields* recebeu o prêmio de melhor trabalho (*Best Paper Award*) no *10th International Conference On Ubiquitous Robots And Ambient Intelligence (URAI 2013)*.