

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
INSTITUTO TERCIO PACITTI DE APLICAÇÕES E PESQUISAS
COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

MARIAM DOS PASSOS AFONSO DA
CONCEIÇÃO

**SISTEMA SLAM COLABORATIVO
PARA CADEIRAS DE RODAS
INTELIGENTES**

Rio de Janeiro
2015

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
INSTITUTO TÉRCIO PACITTI DE APLICAÇÕES E PESQUISAS
COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

MARIAM DOS PASSOS AFONSO DA
CONCEIÇÃO

**SISTEMA SLAM COLABORATIVO
PARA CADEIRAS DE RODAS
INTELIGENTES**

Dissertação de Mestrado submetida ao
Corpo Docente do Departamento de Ci-
ência da Computação do Instituto de Ma-
temática, e Instituto Tércio Pacitti de
Aplicações e Pesquisas Computacionais da
Universidade Federal do Rio de Janeiro,
como parte dos requisitos necessários para
obtenção do título de Mestre em Informá-
tica.

Orientador: Josefino Cabral Melo Lima

Co-orientador: Adriano Joaquim de Oliveira Cruz

Rio de Janeiro
2015

CBIB Conceição, Mariam dos Passos Afonso da
Sistema SLAM Colaborativo para Cadeiras de Rodas Inteligentes / Mariam dos Passos Afonso da Conceição. – Rio de Janeiro, 2015.
95 f.: il.

Orientador: Josefino Cabral Melo Lima; Coorientador: Adriano Joaquim de Oliveira Cruz.

Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro, Instituto de Matemática, Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Programa de Pós-Graduação em Informática, Rio de Janeiro, 2015.

1. Cadeiras de Rodas Inteligentes. 2. Features Lineares. 3. IntellWheels. 4. SLAM. 5. SLAM Colaborativo. – Teses. I. Lima, Josefino Cabral Melo (Orient.). II. Cruz, Adriano Joaquim de Oliveira (Co-orient.). III. Universidade Federal do Rio de Janeiro. IV. Título

CDD:

MARIAM DOS PASSOS AFONSO DA CONCEIÇÃO

**Sistema SLAM Colaborativo para Cadeiras de Rodas
Inteligentes**

Dissertação de Mestrado submetida ao Corpo Docente do Departamento de Ciência da Computação do Instituto de Matemática, e Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários para obtenção do título de Mestre em Informática.

Aprovado em: Rio de Janeiro, ____ de _____ de _____.

Prof. Docteur Josefino Cabral Melo Lima (Orientador)

Prof. Ph.D. Adriano Joaquim de Oliveira Cruz (Co-orientador)

Prof. Ph.D. Felipe Maia Galvão França

Prof. D.Sc. Jonas Knopman

Prof. D.Sc. Claudio Miceli de Farias

Dedico este trabalho a minha família e a todas as pessoas que, a cada dia, lutam
para superar seus limites.

"Viva com um grande ideal. Todavia, esteja também atento às pequeninas coisas."
Meishu-Sama

AGRADECIMENTOS

Primeiramente, quero agradecer ao Supremo Deus pela conclusão de mais esta etapa, cheia de surpresas e desafios, e por tudo o que Ele me permitiu realizar. Agradeço a minha mãe, Maria Amélia, por suas orações, seu apoio incondicional, seu incentivo, e por compreender minha ausência. Mais uma vez, enfim conseguimos!

Minha gratidão às pessoas cujo suporte foi tão importante nesta trajetória. Dentre elas, os Professores Cabral Lima, Luis Paulo Reis, Mónica Faria, Rosaldo Rossetti, Claudio Miceli e, em especial, Adriano Cruz e Jonas Knopman, sem os quais eu não teria um tema de pesquisa bem estruturado. O amigo Juan Dell'Oso, por ter cedido o código de sua dissertação, o qual serviu de base para o simulador desenvolvido neste trabalho. Igualmente importantes foram os amigos do LabIC, do PPGI, de Portugal, e todos aqueles que amenizaram o estresse do Mestrado e a saudade de casa, ou não deixaram que eu me sentisse sozinha – mesmo que isso custasse dez horas de viagem. Também não posso deixar de citar Adriana e Aníbal, da secretaria do PPGI, sempre tão atenciosos e prestativos.

Por fim, meus agradecimentos ao CNPq, ao Programa Erasmus Mundus e à UFRJ/Cenpes, cujo financiamento foi indispensável à realização desta pesquisa.

RESUMO

CONCEIÇÃO, Mariam dos Passos Afonso da. **Sistema SLAM Colaborativo para Cadeiras de Rodas Inteligentes**. 2015. 95 f. Dissertação (Mestrado em Informática) - PPGI, Instituto de Matemática, Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2015.

Idosos e pessoas com algum tipo de limitação são grupos cada vez maiores na população mundial, o que demanda a necessidade de prover-lhes independência e melhor qualidade de vida. As cadeiras de rodas inteligentes (CRIs) surgiram com o intuito de atender às diferentes necessidades de tal público de forma melhor que as cadeiras de rodas tradicionais. CRIs são dispositivos de locomoção dotados de um sistema de controle artificial que auxilia ou substitui o controle dos usuários.

Diversas técnicas da Robótica Móvel já foram empregadas em CRIs, a exemplo de algoritmos para o problema de Localização e Mapeamento Simultâneos (SLAM – *Simultaneous Localization and Mapping*). O SLAM tradicional é aquele em que se utiliza um único robô. Um campo de estudo mais recente é o SLAM Colaborativo, que usa mais de um agente robótico e visa a um mapeamento mais rápido e preciso, explorando o ambiente de forma paralela e reduzindo erros através da integração de dados. Até a data de publicação desta dissertação, na literatura sobre cadeiras de rodas inteligentes foram encontradas referências somente para o uso do SLAM tradicional. Isto é, CRIs capazes de realizar mapeamento o fazem individualmente, e não compartilham o conhecimento gerado com outros dispositivos inteligentes que possam beneficiar-se dele.

Este trabalho vem propor um sistema capaz de apoiar o mapeamento colaborativo entre cadeiras de rodas inteligentes, mantendo o material gerado o mais atualizado possível e disponível para qualquer CRI ou outro dispositivo que dele necessite. Tal sistema deve ser capaz de armazenar informações de mapeamento, que serão utilizadas como dados iniciais em visitas subsequentes de outras CRIs, sendo aumentadas e refinadas gradativamente. Uma das bases de sua elaboração foram as características da Plataforma IntellWheels, uma plataforma para desenvolvimento de CRIs. O ambiente é representado por *features* lineares – uma forma de se lidar com características que podem ser descritas em termos de linhas e segmentos, como paredes, portas e mobília.

Palavras-chave: Cadeiras de Rodas Inteligentes, Features Lineares, IntellWheels, SLAM, SLAM Colaborativo.

ABSTRACT

CONCEIÇÃO, Mariam dos Passos Afonso da. **Sistema SLAM Colaborativo para Cadeiras de Rodas Inteligentes**. 2015. 95 f. Dissertação (Mestrado em Informática) - PPGI, Instituto de Matemática, Instituto Tércio Pacitti, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2015.

Elderly people and patients with some kind of disability represent a growing percentage of the world's population that demands autonomy, independency and quality of life. Intelligent wheelchairs (IWs) are meant to fulfil the diverse needs of such a public in a better way than traditional wheelchairs would do. IWs are locomotion devices to which an artificial control system is added in order to assist or replace their users' control.

Techniques derived from Mobile Robotics have already been employed to IWs, such as algorithms for the Simultaneous Localization and Mapping (SLAM) problem. Whereas traditional SLAM uses only one robot, a more recent field of study, called Collaborative SLAM, uses more than one robotic agent and aims for a quicker and more precise mapping. Those are accomplished by exploring the environment in a parallel fashion and decreasing errors through data fusion. To the best of our knowledge, only the traditional SLAM was employed to intelligent wheelchairs until now. In other words, IWs capable of mapping perform it individually, and they do not share the resulting knowledge with other intelligent devices.

This work proposes a system to support the collaborative mapping among intelligent wheelchairs. It should also maintain the produced material up to date and available to any IW or other devices. Such a system needs to store the mapping information, which may be used as prior data on subsequent visits from other IWs and, as a consequence, can be gradually augmented and refined. One of its conceptual bases was IntellWheels, a generic platform for research and development of intelligent wheelchairs. The environment is represented by linear features – a suitable way of dealing with characteristics that can be described in terms of lines and segments, such as walls, doors and furniture.

Keywords: Intelligent Wheelchairs, Linear Features, IntellWheels, SLAM, Collaborative SLAM.

LISTA DE FIGURAS

Figura 2.1:	Ilustração da imprecisão em sistemas robóticos móveis. [18]	18
Figura 2.2:	Modelo gráfico do problema SLAM. [50]	20
Figura 2.3:	Os campos da Robótica Exploratória e suas interseções: (I) SLAM, (II) exploração clássica, (III) localização ativa e (IV) SPLAM. [17]	23
Figura 2.4:	Protótipos de Madarasz [38] (esq.) e do projeto OMNI [22] (dir.).	28
Figura 2.5:	Tin Man I e II, respectivamente. [11]	29
Figura 2.6:	Protótipo Smartchair. [11]	30
Figura 2.7:	Versões do protótipo do projeto VAHM. [11]	31
Figura 2.8:	Protótipo ACCoMo. [6]	32
Figura 2.9:	Cadeira que reconhece e segue um acompanhante humano [52] (esq.) e a Essex Wheelchair [34] (dir.).	33
Figura 2.10:	Projeto Connected Wheelchair da Intel. (Foto: Divulgação/Intel)	34
Figura 2.11:	Arquitetura da interface multimodal flexível da IWP [22].	37
Figura 2.12:	Interface gráfica da IWP [22].	40
Figura 2.13:	Interface gráfica da IWP, com destaque para as indicações de direção e velocidade da CRI [22].	40
Figura 2.14:	Arquitetura do framework físico da Plataforma IntellWheels [43].	41
Figura 2.15:	Detalhes do protótipo da cadeira IntellWheels [43].	42
Figura 2.16:	Detalhes das partes frontal (esq.) e traseira (dir.) da barra esquerda de sensores [43].	42
Figura 2.17:	Foto da APPC à direita e seu modelo virtual à esquerda [22].	44
Figura 2.18:	Cadeira de rodas virtual [22].	44
Figura 2.19:	Perspectiva em primeira pessoa [22].	45
Figura 3.1:	Fluxograma do sistema proposto.	50
Figura 3.2:	Disposição dos sensores de proximidade ao redor da IntellWheels. Adaptado de [43].	51
Figura 3.3:	Ilustração da pose do robô no FRG e das coordenadas globais da <i>feature</i> em $\langle G \rangle$ e $\langle R \rangle$. Adaptado de [8].	54
Figura 3.4:	Observação mútua no <i>rendez-vous</i> . [9]	56
Figura 4.1:	A1: ambiente virtual simples.	65
Figura 4.2:	A2: ambiente virtual mais complexo.	66
Figura 4.3:	Caminho de R_1 antes do <i>rendez-vous</i> ; primeiro cenário de A1.	68
Figura 4.4:	Caminho de R_2 antes do <i>rendez-vous</i> ; primeiro cenário de A1.	68
Figura 4.5:	Caminho percorrido por R_1 e R_2 após o <i>rendez-vous</i> ; primeiro cenário de A1.	68
Figura 4.6:	Navegação inicial de R_1	69

Figura 4.7: Mapa M_1 inicial estimado por R_1	69
Figura 4.8: Navegação inicial de R_2	70
Figura 4.9: <i>Rendez-vous</i> e alinhamento dos mapas.	70
Figura 4.10: Mapa M_2 aumentado.	71
Figura 4.11: Mapa M_1 final estimado por R_1	71
Figura 4.12: Mapa M_2 final estimado por R_2	72
Figura 4.13: Caminho de R_1 (antes do <i>rendez-vous</i>); segundo cenário de A1.	74
Figura 4.14: Caminho de R_2 (antes do <i>rendez-vous</i>); segundo cenário de A1.	74
Figura 4.15: Caminho de R_3 , que entra no ambiente com o mapa aumentado, ora M_1 , ora M_2 ; segundo cenário de A1.	75
Figura 4.16: Caminho de R_1	77
Figura 4.17: Caminho de R_2	77
Figura 4.18: Estados finais de M_1 com menor (acima) e maior (abaixo) erro médio.	79
Figura 4.19: Estados finais de M_2 com menor (acima) e maior (abaixo) erro médio.	79
Figura 4.20: Caminho de R_3 no terceiro cenário de A2, com início e término no mesmo ponto.	81

LISTA DE TABELAS

Tabela 4.1: Valores médios das métricas para M_1 e M_2 : ambiente A1, primeiro cenário, configuração de sensoriamento tipo 1.	73
Tabela 4.2: Valores médios das métricas para M_1 e M_2 : ambiente A1, primeiro cenário, configuração de sensoriamento tipo 2.	73
Tabela 4.3: Valores médios das métricas para M_1 e M_2 : ambiente A1, segundo cenário, configuração de sensoriamento tipo 1.	76
Tabela 4.4: Valores médios das métricas para M_1 e M_2 : ambiente A1, segundo cenário, configuração de sensoriamento tipo 2.	76
Tabela 4.5: Valores médios das métricas para M_1 e M_2 : ambiente A2, primeiro cenário, configuração de sensoriamento tipo 1.	78
Tabela 4.6: Valores médios das métricas para M_1 e M_2 : ambiente A2, primeiro cenário, configuração de sensoriamento tipo 2.	78
Tabela 4.7: Valores médios das métricas para M_1 e M_2 : ambiente A2, segundo cenário, configuração de sensoriamento tipo 1.	80
Tabela 4.8: Valores médios das métricas para M_1 e M_2 : ambiente A2, segundo cenário, configuração de sensoriamento tipo 2.	80
Tabela 4.9: Valores médios das métricas para M_1 e M_2 : ambiente A2, terceiro cenário, configuração de sensoriamento tipo 1.	81
Tabela 4.10: Valores médios das métricas para M_1 e M_2 : ambiente A2, terceiro cenário, configuração de sensoriamento tipo 2.	82

LISTA DE ABREVIATURAS E SIGLAS

APPC	Associação do Porto de Paralisia Cerebral
C-SLAM	SLAM Colaborativo (ou Cooperativo)
Co-SLAM	SLAM Colaborativo (ou Cooperativo)
CRI	Cadeira de Rodas Inteligente
EKF	Filtro de Kalman Extendido (<i>Extended Kalman Filter</i>)
FEUP	Faculdade de Engenharia da Universidade do Porto
FOV	Campo de Visão (<i>Field of View</i>)
FRG	<i>Frame</i> de Referência Global
FRL	<i>Frame</i> de Referência Local
FRR	<i>Frame</i> de Referência do Robô
GUI	Interface Gráfica do Usuário (<i>Graphical User Interface</i>)
IBGE	Instituto Brasileiro de Geografia e Estatística
IDE	Ambiente Integrado para Desenvolvimento (<i>Integrated Development Environment</i>)
IntellSim	<i>IntellWheels Simulator</i>
IWP	Plataforma IntellWheels
LabIC	Laboratório de Inteligência Computacional da UFRJ
LIACC	Laboratório de Inteligência Artificial e Ciência de Computadores da Universidade do Porto
LRF	<i>Laser Range Finder</i>
M-Space	Subespaço de Medição (<i>Measurement Subspace</i>)
MASI	Modelos e Arquiteturas para Sistemas Inteligentes
MMI	Interface Multimodal (<i>Multimodal Interface</i>)
MR-SLAM	SLAM com múltiplos robôs (<i>Multi-robot SLAM</i>)
NAROBISI	Navegação Robótica Bio-Inspirada em Superfícies Irregulares
ONU	Organização das Nações Unidas

RFID	Identificação por Radiofrequência (<i>Radio-Frequency Identification</i>)
ROS	<i>Robot Operating System</i>
SLAM	Localização e Mapeamento Simultâneos (<i>Simultaneous Localization and Mapping</i>)
SMA	Sistema Multiagentes
SPLAM	Planejamento, Localização e Mapeamento Simultâneos (<i>Simultaneous Planning, Localization and Mapping</i>)
UFRJ	Universidade Federal do Rio de Janeiro
UNFPA	Fundo de População das Nações Unidas (<i>United Nations Population Fund</i>)

SUMÁRIO

1	INTRODUÇÃO	13
2	REFERENCIAL TEÓRICO	17
2.1	SLAM – LOCALIZAÇÃO E MAPEAMENTO SIMULTÂNEOS	17
2.1.1	SLAM Colaborativo	22
2.2	CADEIRAS DE RODAS INTELIGENTES	26
2.2.1	Plataforma IntellWheels	35
3	SISTEMA DE MAPEAMENTO COLABORATIVO COM CADEIRAS DE RODAS INTELIGENTES	47
3.1	CONCEITO DO SISTEMA E ESCOLHA DO MÉTODO C-SLAM	48
3.2	EKF-SLAM COLABORATIVO COM FEATURES LINEARES	53
3.2.1	SLAM com Múltiplos Robôs e Fusão de Mapas	55
4	IMPLEMENTAÇÃO, TESTES E RESULTADOS	59
4.1	TESTES E ANÁLISE DOS RESULTADOS	65
4.1.1	Ambiente A1	67
4.1.2	Ambiente A2	76
5	CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	84
	REFERÊNCIAS	88

1 INTRODUÇÃO

Assuntos relacionados à independência e à autonomia de idosos e de pessoas com algum tipo de limitação ganharam cada vez mais a atenção da sociedade nas últimas décadas, principalmente pela crescente representatividade desses grupos na população mundial. Tal tendência é apontada em diversos estudos, a exemplo de relatórios emitidos por entidades como a ONU [27], a UNFPA (Fundo de População das Nações Unidas) [1] e o IBGE [16]. Relacionada de forma estreita à integração social e ao bem-estar do indivíduo, a mobilidade é de extrema importância para sua saúde física e mental, aumentando sua autoestima, autoconfiança e felicidade [31].

Um dos dispositivos tecnológicos mais utilizados para aumentar a independência e a qualidade de vida de pessoas com limitações é a cadeira de rodas. Ela pode beneficiar pessoas com condições diversas, desde paraplégicos e tetraplégicos até idosos e pacientes com Alzheimer, Parkinson ou Paralisia Cerebral. Ao mesmo tempo, uma vez que esses potenciais usuários possuem diferentes necessidades, muitos podem não ser capazes de utilizar cadeiras de rodas tradicionais. Nesses casos, uma cadeira de rodas inteligente (CRI) costuma ser uma solução adequada. Uma CRI pode ser definida como um dispositivo de locomoção dotado de um sistema de controle artificial que auxilia ou substitui o controle de seu usuário, isentando-o total ou parcialmente da necessidade de conduzir a cadeira. Em geral, CRIs contam com técnicas oriundas da Robótica Móvel, costumam ter um conjunto de sensores e são controladas por um computador. [12] [22] [44]

Em Robótica existe o chamado problema de Localização e Mapeamento Simultâneos (SLAM – *Simultaneous Localization and Mapping*) [47] [49]. Neste âmbito, um campo de estudo mais recente é o SLAM Colaborativo (C-SLAM), que pode

ser definido como uma técnica que utiliza mais de um agente robótico e visa ao mapeamento mais preciso e em menos tempo, com a exploração paralela do ambiente e a redução de erros através da integração dos dados dos vários agentes envolvidos [17] [19] [33] [36]. A aplicação de algoritmos SLAM tradicionais (também conhecidos como *Single-robot SLAM* ou SLAM com um único robô) a cadeiras de rodas inteligentes já foi abordada em alguns trabalhos, a exemplo dos estudos de Cheein *et al.* [3], Wu *et al.* [52] e de um grupo da Universidade de Essex [34, 35]. Até a data de publicação desta dissertação, entretanto, não foram encontradas referências na literatura sobre o uso de C-SLAM em tal escopo.

A partir da definição de cadeira de rodas inteligente apresentada, pode-se dizer que ela é um agente robótico móvel, capaz de colher informações sobre o ambiente em que se encontra (como a localização de seus elementos) e de, possivelmente, utilizá-las para construir mapas. Uma CRI capaz de realizar mapeamento o faz individualmente, e o conhecimento gerado não é compartilhado com outras entidades que possam beneficiar-se dele. Podem-se destacar três desvantagens dessa situação, a primeira delas com relação à precisão do mapa obtido sem que se tire proveito da fusão dos dados de mapeamento na presença de mais de uma cadeira. Em segundo lugar, todo agente dotado da funcionalidade de mapeamento que visite certo ambiente terá de mapeá-lo, não importa quantos outros já o tenham feito anteriormente. Além disso, existe a possibilidade de que os mapas armazenados pelas cadeiras tornem-se defasados com o passar do tempo. Os dois primeiros fatores apontam que CRIs, assim como outros dispositivos robóticos móveis, podem tirar proveito de técnicas de colaboração para melhorar a acurácia dos mapas que constroem. Mais que isso, a disponibilidade de mapas os mais recentes possíveis seria uma vantagem adicional.

De fato, o mais comum é que o problema C-SLAM seja tratado sob a forma de um time distribuído de agentes robóticos autônomos dedicados à função de ma-

peamento por completo de um ambiente. Em outras palavras, a finalidade de toda a operação dos robôs é o mapeamento de ambientes [17] [19] [33] [36]. Desta forma, à primeira vista o contexto de CRIs pode não parecer adequado para a aplicação de SLAM Colaborativo. Por outro lado, se a técnica for tomada sob a ótica de um processo que simplesmente permite a troca e a integração de dados de mapeamento entre agentes, visando apenas à melhoria dos mapas construídos por eles, ela pode, sim, ser aproveitada por CRIs. O mais importante, sem colocar em risco suas funcionalidades principais, como a garantia da segurança do usuário e a robustez da navegação.

Assim, este trabalho vem propor, como objetivo geral, um sistema capaz de apoiar o mapeamento colaborativo entre cadeiras de rodas inteligentes, mantendo o material gerado o mais atualizado possível e disponível para qualquer CRI ou outro dispositivo que dele necessite. Tal sistema deve ser capaz de armazenar informações de mapeamento, que serão utilizadas como dados iniciais em visitas subsequentes de outras CRIs, sendo aumentadas e refinadas gradativamente. Ele foi elaborado com vistas à sua incorporação na Plataforma IntellWheels (IWP), uma plataforma genérica para desenvolvimento de CRIs com arquitetura modular que facilita a integração de diferentes sensores, dispositivos de entrada, métodos de cooperação etc. [22] [43]. Entretanto, muitas etapas de pesquisa são necessárias para seja finalmente possível a implantação de um módulo deste tipo na IWP. Por isso, este trabalho se restringe às três etapas iniciais como objetivos específicos: (i) proposta da dinâmica do sistema; (ii) escolha de uma estratégia C-SLAM compatível com a proposta e com o modelo IntellWheels; e (iii) avaliação de quais informações devem ser passadas às cadeiras na forma de um mapa prévio e da qualidade dos mapas obtidos por tal processo. Levantou-se a hipótese de que seria suficiente uma representação simples, composta pelo vetor de estado e pela matriz de covariância finais do último mapa armazenado no sistema e pela pose inicial do agente que esteja entrando no ambiente. Tal hipótese, a princípio, seria justificável pela padronização do conjunto

de sensores da Plataforma IntellWheels. Foram realizados apenas experimentos simulados, a fim de testar o terceiro dos objetivos específicos e verificar a hipótese levantada.

Neste sistema, o ambiente é representado por *features* lineares. Uma *feature* nada mais é do que uma característica observável do ambiente em questão. Em especial, a representação por *features* lineares é uma forma de se lidar com características que podem ser descritas em termos de linhas e segmentos, como paredes, portas e mobília em ambientes internos. Cada CRI executa localmente o *Single-robot EKF-SLAM* (SLAM tradicional baseado no Filtro de Kalman Estendido) até encontrar outra cadeira, quando ocorre a fusão dos mapas gerados por cada uma. Após a fusão, ambas possuem a mesma cópia do mapa do ambiente, unindo as informações que elas detinham até então. Cada uma prossegue sua navegação realizando o EKF-SLAM novamente de forma individual, mas utilizando o mapa aumentado. Tal estratégia baseou-se no trabalho de Benedettelli, Garulli, Giannitrapani [9], adaptado para que as *features* lineares fossem descritas pelas coordenadas de seus pontos extremos. O modelo do robô e seu sensoriamento são simplificações do modelo IntellWheels, mas cuidou-se para que os parâmetros de sensoriamento e os ruídos fossem mantidos em valores representativos de tal modelo.

Esta dissertação está organizada conforme segue. O Capítulo 2 trata do referencial teórico, discorrendo sobre o problema SLAM e sua versão colaborativa, cadeiras de rodas inteligentes e a Plataforma IntellWheels. As características do sistema proposto e o método C-SLAM escolhido são assunto do Capítulo 3, enquanto a implementação, os testes realizados e seus resultados estão no Capítulo 4. Por fim, considerações finais e sugestões de trabalhos futuros são o tema do Capítulo 5.

2 REFERENCIAL TEÓRICO

A primeira seção deste capítulo apresenta o conceito e as principais características do problema de Localização e Mapeamento Simultâneos tradicional (SLAM) e de sua versão colaborativa (C-SLAM). A seção seguinte discorre sobre cadeiras de rodas inteligentes, incluindo alguns exemplos de projetos encontrados na literatura. Em particular, a Seção 2.2.1 é dedicada a apresentar a Plataforma IntellWheels.

2.1 SLAM – Localização e Mapeamento Simultâneos

Uma das grandes questões em Robótica Móvel é o problema de Localização e Mapeamento Simultâneos, conhecido pela sigla SLAM (*Simultaneous Localization and Mapping*). Um sistema operando com SLAM requer, no mínimo, um agente móvel equipado com ao menos um sensor exteroceptivo. Isto é, ele deve ser capaz de extrair informação do seu entorno. A dificuldade deste tipo de problema reside no fato de que o robô precisa, ao mesmo tempo, construir uma representação interna do ambiente e localizar-se com base nesse modelo. Em outras palavras, o agente desconhece tanto o mapa do ambiente como suas próprias **poses** (posição e orientação com respeito ao sistema de coordenadas do ambiente). Ele obtém somente medições e dados de controle. [47] [49]

Publicações envolvendo autores como Thrun e Grisetti [29] [49] [50] foram bastante citadas na literatura sobre Localização e Mapeamento Simultâneos dos últimos anos. Em especial, uma boa base teórica sobre o assunto pode ser encontrada em Thrun, Burgard, Fox [49] e em Thrun, Leonard [50]. Em português, pode-se citar o trabalho de Dell’Oso [21]. Essas três referências servirão como principais fontes

do breve detalhamento sobre SLAM nos parágrafos a seguir.

Um dos problemas geralmente encontrados em sistemas robóticos móveis são incertezas geradas por sensores ruidosos e dispositivos imprecisos. Tais incertezas afetam diretamente os dados sobre a localização do robô e dos elementos (*features*) que ele observa no mundo desde o início da navegação. Isso faz com que a realidade e a representação que o robô possui internamente da sua trajetória e da configuração do ambiente (dados estimados) não coincidam, o que é muito bem ilustrado pela Fig. 2.1.

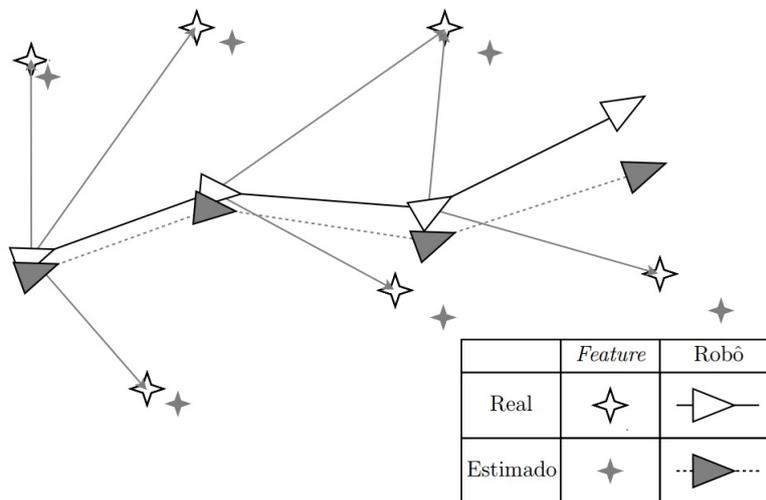


Figura 2.1: Ilustração da imprecisão em sistemas robóticos móveis. [18]

Em outras palavras, a imprecisão nas informações obtidas e na execução dos comandos de movimento faz com que a pose real do agente e a localização real das *features* não correspondam às respectivas estimativas. Com o acúmulo dessas incertezas, as estimativas divergem cada vez mais da realidade, levando à construção de mapas equivocados [4] [47] [49]. SLAM é uma técnica que visa à diminuição dessa discrepância. Para tanto, uma das estratégias utilizadas é a reamostragem: quando o robô retorna a pontos previamente visitados, as informações podem ser

refinadas através da comparação entre estimativas atuais e antigas (que possuem uma incerteza menor).

O problema SLAM pode ser descrito em termos probabilísticos. Em particular, a vantagem de algoritmos desse tipo é a possibilidade de se modelar tanto ruídos de fontes diversas quanto seus efeitos nas medições [4]. Sejam o tempo t e a pose x_t do robô. Num cenário onde a altura dos objetos pode ser ignorada, x_t geralmente é um vetor coluna tridimensional composto pelo par ordenado que designa a posição do robô no plano e pelo valor que indica sua orientação com respeito ao sistema de coordenadas do plano. Uma sequência de poses é chamada de **trajetória**, denotada por X_T , onde T (que pode ser ∞) é o instante de tempo que marca seu término. A pose inicial $x_0 = [0, 0, 0]^T$ é conhecida, mas as demais não o são.

$$X_T = \{x_0, x_1, x_2, \dots, x_T\} \quad (2.1)$$

Através da odometria o robô obtém informações relativas a duas poses consecutivas, isto é, ao movimento entre duas poses. Esses dados são obtidos com o uso de *encoders* (dispositivos eletromecânicos capazes de computar a quantidade de voltas realizadas pelas rodas do robô) ou a partir dos sinais de controle transmitidos aos motores. Seja u_t a odometria que caracteriza sua movimentação entre os instantes de tempo $t - 1$ e t . Assim, a movimentação relativa do robô é dada pela sequência U_T .

$$U_T = \{u_1, u_2, u_3, \dots, u_T\} \quad (2.2)$$

Se a odometria fosse livre de ruído, X_T poderia facilmente ser recuperada a partir de U_T , mas este não é o caso quando se trata de aplicações reais.

Seja m o mapa verdadeiro¹ do ambiente, que descreve a localização de objetos, **marcos** (ou *landmarks*) e outros elementos que façam parte dele. Marcos são características distinguíveis, estacionárias e reobserváveis do ambiente, que podem

¹Por “verdadeiro”, entenda-se “completo e preciso”.

ser reconhecidas com certo grau de confiança. Eles podem ser de diversos tipos, dependendo do ambiente em que o robô esteja inserido. Quinas de paredes, portas, sinais WiFi ou marcas propositais são alguns exemplos [25] [30]. O mapa m geralmente é tido como estático – não muda ao longo do tempo. Tais informações sobre o ambiente são obtidas graças à capacidade que o robô possui de perceber seu entorno. Assim, são essas medições que estabelecem correspondências entre os elementos em m e a localização x_t do robô. É possível assumir, sem perda de generalidade, que o robô realiza exatamente uma medição a cada instante de tempo. A sequência dessas medições é dada por Z_T .

$$Z_T = \{z_1, z_2, z_3, \dots, z_T\} \quad (2.3)$$

O problema SLAM pode ser enunciado, então, como aquele em que se deseja recuperar um modelo do mundo m mais verdadeiro possível e a sequência X_T de poses do robô a partir dos dados de odometria U_T e observação do ambiente Z_T .

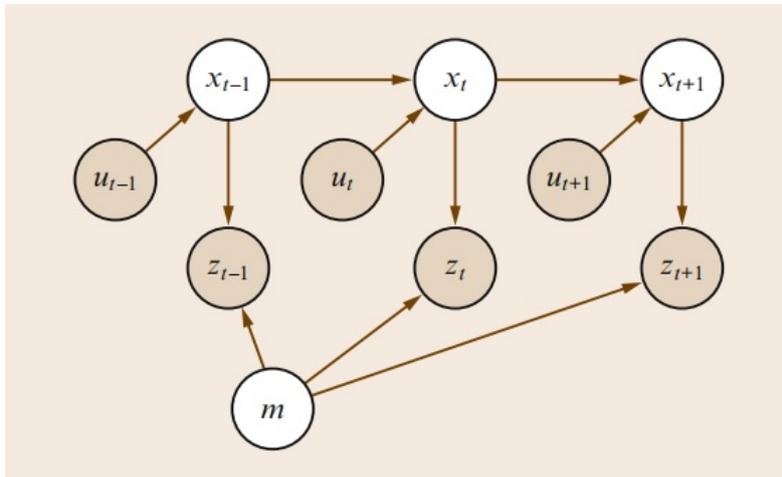


Figura 2.2: Modelo gráfico do problema SLAM. [50]

Outra maneira de se representar e explicar o problema SLAM, e que facilita a compreensão da dependência entre suas variáveis, é o modelo gráfico, ilustrado

pela Fig. 2.2. Os dados de hometria (u) e observação (z), representados pelos nós sombreados, são as chamadas **variáveis observáveis**, pois são diretamente observáveis pelo robô. No problema SLAM, elas são utilizadas como base para recuperar o mapa (m) e as poses (x), representados pelos nós brancos e que, por sua vez, são denominados **variáveis inobserváveis**. Esse tipo de diagrama ressalta a sequência em que as variáveis ocorrem no sistema e sua relação causal através das arestas direcionais. Tome-se como exemplo a Fig. 2.2, no instante de tempo t . (i) A pose x_t depende da pose anterior x_{t-1} e da hometria atual u_t . (ii) Por sua vez, a observação z_t depende da pose do robô naquele mesmo instante de tempo (x_t) e da configuração do mapa m . No instante de tempo seguinte $t + 1$, o robô obtém nova hometria u_{t+1} . Desse dado e da pose anterior x_t depende a pose atual x_{t+1} . Tal cadeia de dependências se repete ao longo de toda a navegação do robô; a Fig. 2.2 ilustra apenas um breve trecho do que seria sua navegação completa.

As relações de dependência explicitadas no parágrafo anterior são importantes para a solução do problema SLAM: são os modelos de (i) movimento e (ii) medição. Em geral, eles são pensados como distribuições de probabilidade, conforme 2.4 e 2.5, respectivamente.

$$p(x_t | x_{t-1}, u_t) \tag{2.4}$$

$$p(z_t | x_t, m) \tag{2.5}$$

As formas de se interpretar a questão de Localização e Mapeamento Simultâneos podem ser classificadas em dois tipos. Uma delas é chamada de **full SLAM** e leva em consideração o conjunto completo de medições para estimar toda a trajetória do robô, conforme 2.6.

$$p(X_T, m | Z_T, U_T) \tag{2.6}$$

Ou seja, tem-se o problema do cálculo da probabilidade posterior conjunta de X_T e m dadas as variáveis observáveis Z_T e U_T . Ela também é conhecida por **smoothing**

SLAM, pois geralmente é abordada com estratégias de suavização como a minimização de erros pelo método dos mínimos quadrados. Além disso, seu processamento costuma ser feito em lote.

A outra classificação é denominada **online SLAM** e consiste numa estimação online de estados, sendo o estado do sistema composto pelo mapa m e pela pose x_t do robô no instante t .

$$p(x_t, m | Z_T, U_T) \quad (2.7)$$

A estimativa é refinada incrementalmente, conforme novas medições tornam-se disponíveis. Esse tipo também é nomeado **filtering SLAM**, uma vez que sua solução é comumente baseada em filtros estatísticos. Uma das abordagens mais comuns é a utilização do **Filtro de Kalman Estendido**, ou **EKF** (*Extended Kalman Filter*).

Apesar de todas as vantagens trazidas com o desenvolvimento do SLAM, de acordo com Zou, Tan [55], um de seus principais pontos fracos é o fato de suas abordagens convencionais não lidarem bem com ambientes dinâmicos. Em geral, pontos identificados como sendo móveis são considerados outliers e descartados, o que funciona bem para uma quantidade limitada desse tipo de elemento. Entretanto, a chance das técnicas falharem aumenta quando o conjunto de pontos dinâmicos é grande.

2.1.1 SLAM Colaborativo

De acordo com Camara Neto, Campos [17] e Liu, Fan, Zhang [36], embora o problema **SLAM tradicional** (*Single-robot SLAM*, ou **SLAM com um único robô**) tenha sido amplamente estudado nas duas últimas décadas e alcançado um alto nível de maturidade, por muito tempo as técnicas desenvolvidas deixaram de lado questões acerca de estratégias exploratórias. Isto é, considerava-se que a tra-

jetória dos agentes seria aleatória, ou determinada por teleoperação humana, seguimento de paredes etc.. Isso se torna um problema grave no contexto da Robótica Cooperativa, onde as ações dos robôs devem ser autônomas, adaptativas e dinâmicas. Esse tipo de problema precisa lidar com fatores como a concorrência dos agentes por espaço físico, tratamento de colisão, interferência entre sensores e comunicação para troca de dados. A fim de preencher essa lacuna no campo de Localização e Mapeamento Simultâneos, deu-se início ao estudo da junção entre SLAM, cooperação e planejamento em duas frentes independentes: (i) SPLAM, sigla para Planejamento, Localização e Mapeamento Simultâneos (*Simultaneous Planning, Localization and Mapping*); e (ii) **SLAM Colaborativo**.

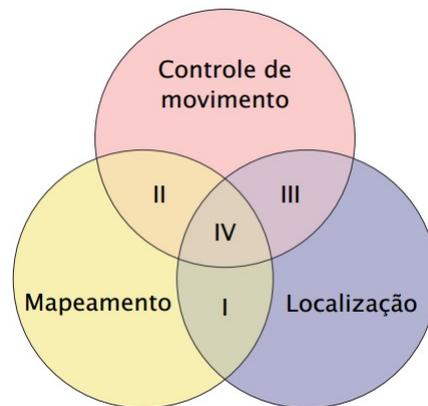


Figura 2.3: Os campos da Robótica Exploratória e suas interseções: (I) SLAM, (II) exploração clássica, (III) localização ativa e (IV) SPLAM. [17]

SPLAM, também conhecido como *Integrated Exploration* (Exploração Integrada), *Adaptive Concurrent Mapping and Localization* (Localização e Mapeamento Adaptativos Concorrentes, em livre tradução) ou, ainda, *Active SLAM*, é a interseção entre as três principais áreas da Robótica Exploratória: mapeamento, localização e controle de movimento ou de trajetória (Fig. 2.3). O planejamento de trajetória depende essencialmente do conhecimento prévio do mapa do ambiente, o que torna a integração do aspecto exploratório ao SLAM algo não trivial. Por isso a técnica SPLAM visa ao desenvolvimento de estratégias de planejamento adaptativas

e com baixo custo computacional, incorporando os dados sensoriais constantemente coletados pelo agente.

SLAM Colaborativo, por sua vez, não possui uma definição consensual. De forma geral, pode-se delinear três classes de problemas com base no aspecto do processo SLAM beneficiado pela cooperação. A primeira é a Localização Cooperativa, também conhecida por Posicionamento Cooperativo ou Interlocalização. Neste caso, cada robô conhece sua posição relativa aos demais e sua hometria é corrigida antes de ser enviada ao módulo SLAM do sistema. A segunda classe é a Exploração Distribuída, que consiste na extensão do SPLAM para um cenário com múltiplos robôs. Por fim, a terceira classe é o Processamento Distribuído e Fusão de Dados, onde cada agente trata o SLAM localmente, isto é, no seu espaço de vizinhança. Assim, uma solução global de mapeamento é estimada com frequência a partir dessas estimativas individuais. Além disso, na literatura há diferentes termos utilizados para designar SLAM Colaborativo, e qualquer um deles poderá ser encontrado ao longo deste texto: *Collaborative SLAM*, *Cooperative SLAM* (SLAM Cooperativo), *C-SLAM*, *Co-SLAM* e *Multi-robot SLAM*, ou MR-SLAM (SLAM com múltiplos robôs).

O mais comum é que o problema C-SLAM seja tratado sob a forma de um time de agentes autônomos operando de forma distribuída, dedicados ao mapeamento de um ambiente por completo. Uma tarefa é atribuída a cada robô, cujas ações são coordenadas visando à otimização do processo de mapeamento e do resultado obtido. Tal resultado é o mapa compartilhado do ambiente, construído a partir da combinação dos dados coletados pelos agentes. Em geral eles também coordenam uma estratégia de exploração com base nesse mapa, de forma a torná-la mais eficiente. Exemplos de exploração distribuída e coordenada são técnicas de Escalonamento por Reversão de Arestas (*Scheduling by Edge Reversal*, ou SER), um algoritmo que interpreta o ambiente como um grafo conexo e se baseia nas vizinhan-

ças dos vértices então formados [2] [28]. De forma resumida, SLAM Colaborativo busca construir um mapa de forma mais rápida e precisa; explorar o ambiente de forma paralela, coordenada e, conseqüentemente, mais veloz; e reduzir os erros de mapeamento, graças à integração de dados vindos de vários agentes. [17] [19] [33] [36]

Embora o problema SLAM Colaborativo possa ser abordado de diferentes formas, podem-se ressaltar algumas decisões que lhes são comuns. Primeiramente, alguma estratégia oriunda das técnicas de *Single-robot SLAM* é escolhida como base para o problema de Localização e Mapeamento Simultâneos em si – EKF-, *Fast-*, *Visual SLAM* etc.. Outra questão, um dos pontos-chave do problema, é a **fusão de mapas**, também conhecida como **fusão de dados** ou *map merging*. Essa etapa consiste na formação de um mapa resultante da união dos dados levantados pelos robôs e exige a determinação de um *frame* de referência comum, pois cada agente coleta dados com relação a seu *frame* local de referência. A fusão de mapas é crucial para a qualidade do mapa gerado, uma vez que influencia diretamente sua consistência e, conseqüentemente, o comportamento do sistema [9]. Por fim, tem-se a forma como as informações são trocadas no sistema durante a execução, o que pode ocorrer diretamente entre os agentes ou através de uma entidade externa ao time. No primeiro caso, uma das estratégias mais comuns é o *rendez-vous*, no sentido de **encontro**²: a troca de informações é feita entre pares de robôs no momento em que eles se encontram. Os trabalhos de Benedettelli, Garulli, Giannitrapani [9] e de Liu, Fan, Zhang [36] são alguns que usam essa técnica. Já no segundo caso, pode-se utilizar a ideia de *Cloud Robotics* (Robótica em Nuvem), a exemplo do trabalho de Mohanarajah *et al.* [39], onde uma entidade externa (a “nuvem”) é responsável por receber os dados coletados pelos robôs, executar a fusão de dados e devolver-lhes o mapa atualizado. [17] [19] [33] [36]

²<http://www.aulete.com.br/rendez-vous>, acessado em 16-06-2015.

2.2 Cadeiras de Rodas Inteligentes

A autonomia e a independência de idosos e de pacientes com algum tipo de limitação são assuntos que têm despertado cada vez mais o interesse da sociedade nas últimas décadas. Isso se deve fortemente à porcentagem da população mundial que estes grupos representam, e à velocidade do envelhecimento global. Segundo levantamento realizado pela ONU [27], em 2007 já havia mais de 650 milhões de pessoas ao redor do mundo com algum tipo de deficiência ou limitação, correspondendo a aproximadamente 10% da população na época. Um relatório da UNFPA (Fundo de População das Nações Unidas) divulgado cinco anos depois [1] fez algumas projeções sobre a população mundial a partir dos 60 de idade. Estimou-se que este grupo somaria um bilhão de pessoas até 2022, e dois bilhões de pessoas até 2050, quando 80% delas viveriam em países em desenvolvimento. Outra pesquisa, realizada em 2013 pelo IBGE [16], indicou que tal grupo deverá corresponder a 13.8% e 33.7% da população brasileira em 2020 e 2060, respectivamente.

A mobilidade é um aspecto de grande importância para a saúde física e mental do ser humano, pois está fortemente atrelada à sua integração social e ao seu bem-estar [31]. A possibilidade de se movimentar com independência é um fator que aumenta a autoconfiança, a autoestima e o nível de felicidade dos indivíduos. A tecnologia pode auxiliar pessoas com limitações a executarem tarefas que seriam difíceis ou até impossíveis para elas, contribuindo para sua independência e melhor qualidade de vida. Um dos dispositivos mais comuns neste sentido são as cadeiras de rodas. Mais do que paraplégicos e tetraplégicos, elas também ajudam pessoas idosas e pacientes com Alzheimer, Parkinson, Paralisia Cerebral e outras condições que impedem ou dificultam sua locomoção. Ao longo do tempo, pesquisadores têm trabalhado na melhoria das características das cadeiras de rodas tradicionais (manuais e elétricas), bem como na sua adaptação às diferentes necessidades de seus usuários. Ainda assim, dependendo do tipo e do grau de sua limitação, um indiví-

duo pode não ser capaz de utilizá-las. Nesses casos, uma cadeira de rodas inteligente (CRI) pode ser uma solução adequada. [12] [22] [44]

Pode-se dizer que o conceito de cadeiras de rodas inteligentes é uma consequência natural dessas iniciativas de melhorias e adequações das cadeiras de rodas tradicionais. O trabalho de Faria [22] definiu a CRI como um dispositivo de locomoção dotado de um sistema de controle artificial que auxilia ou substitui o controle de seus usuários. O objetivo é isentá-los total ou parcialmente da necessidade de conduzir a cadeira. Em geral ela conta com técnicas oriundas da robótica móvel, costuma ter um conjunto de sensores e é controladas por um computador. De acordo com Braga *et al.* [12], as principais características de uma CRI são (i) a interação com o usuário, que pode ser feita através de diferentes dispositivos de controle (joystick, comandos de voz, movimentos da cabeça ou dos olhos etc.); (ii) a navegação autônoma, segura, flexível e capaz de desviar de obstáculos; e (iii) a comunicação com outros dispositivos inteligentes, como portas automáticas e outras CRIs.

De fato, a adaptação de cadeiras de rodas ao seu público alvo (indivíduos acometidos de algum tipo de limitação física) é um grande desafio, pois deve ter em conta conceitos de usabilidade, acessibilidade e robustez. Mais ainda, é preciso garantir a segurança dos usuários [22]. Merece destaque o conceito de que uma CRI deve interferir minimamente no ambiente em que opera para ter um bom desempenho. Além disso, é preciso levar em consideração que seus usuários não necessariamente serão pessoas com conhecimentos técnicos para utilizar interfaces e controles complicados. [11]

O desenvolvimento de métodos de navegação e autolocalização para CRIs é outra questão crítica, pois, novamente, envolve quesitos de segurança. As técnicas mais comuns empregadas nos últimos anos para ambientes internos são *dead reckoning* (navegação estimada) e SLAM. Já a navegação em ambientes externos

comumente utiliza dispositivos de GPS.

Atualmente também existem simuladores para cadeiras de rodas inteligentes e comuns (manuais ou elétricas). Cada um possui suas especificidades, mas, de forma geral, o objetivo desses softwares é o treinamento dos usuários na condução das cadeiras ou a realização de testes de desempenho do equipamento físico.

Os trabalhos de Barbosa [6], Braga [11], Faria [22] e Petry [43] apresentam diversos projetos de cadeiras de rodas inteligentes, incluindo simuladores, presentes na literatura. A seguir, parte das características e funcionalidades de alguns deles é brevemente resumida. Mais detalhes podem ser encontrados nas referências supracitadas.

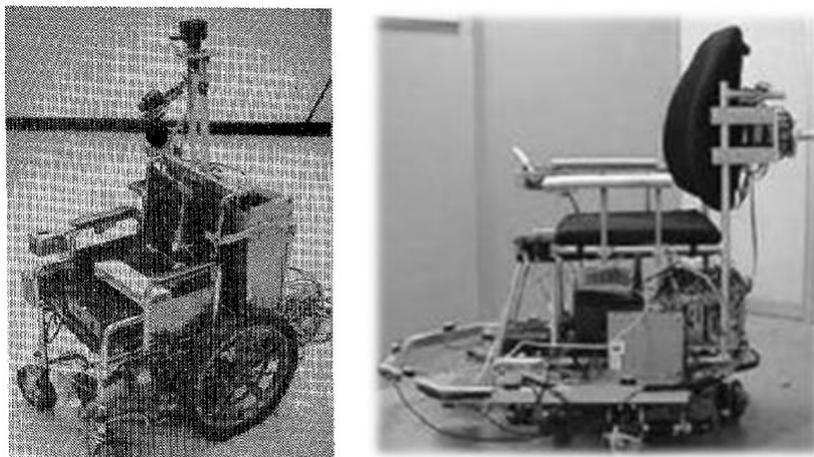


Figura 2.4: Protótipos de Madarasz [38] (esq.) e do projeto OMNI [22] (dir.).

Um dos primeiros projetos de cadeira de rodas autônoma data de 1986, proposto por Madarasz *et al.* [38] (Fig. 2.4, esq.). Seu objetivo era o desenvolvimento de uma cadeira de rodas que pudesse operar sem intervenção humana em ambientes populosos, minimizando ao máximo as colisões com qualquer tipo de obstáculo (pessoas e objetos). Estas funcionalidades dependiam de uma câmera digital acoplada à CRI. Elas baseavam-se no reconhecimento de marcos de referência e de objetos

móveis, bem como na localização e no reconhecimento de pontos identificados anteriormente pelo sistema.

Em 1993 foi proposto o simulador Powered Wheelchair Mobility, uma ferramenta para treinamento e avaliação de usuários de cadeiras de rodas manuais. Os resultados obtidos indicaram que o comportamento dos pacientes era similar tanto com o simulador quanto com o dispositivo físico. O projeto OMNI (*Office Wheelchair with High Manoeuvrability and Navigational Intelligence for People with Severe Handicap*, Fig. 2.4, dir.), do mesmo ano, propunha uma CRI omnidirecional. Seu sistema, dividido em três módulos inteligentes independentes, contava com planejadores de trajetória e tarefas, desvio de obstáculos, passagem através de portas e modos de navegação autônomo e semiautônomo. Em 1995, os autores de Tin Man I buscaram construir uma cadeira de rodas inteligente de baixo custo. Ela possuía três modos de operação: (i) condução executada pelo usuário com desvio automático de obstáculos; (ii) navegação através de um caminho pré-determinado; ou (iii) navegação até um ponto específico. Três anos depois foi divulgada sua segunda versão, chamada Tin Man II. Esta, por sua vez, dispunha de funcionalidades mais avançadas, como armazenamento de dados da navegação, regresso ao ponto de partida e passagem através de portas. Tais protótipos são exibidos na Fig. 2.5.



Figura 2.5: Tin Man I e II, respectivamente. [11]

A Smartchair (Fig. 2.6), desenvolvida pela Universidade da Pensilvânia (EUA), também desviava de obstáculos, movimentava-se para um destino específico e cruzava portas. O projeto VAHM (*Véhicule Autonome pour Handicapé Moteur*, Fig. 2.7), que teve início em torno de 1998, tinha como alvo a construção de uma interface que tornasse mais eficiente o controle de cadeiras de rodas elétricas. Assim como a OMNI e a Tin Man, a VAHM também era dotada de navegação semiautônoma com desvio de obstáculos. Além disso, era capaz de navegar autonomamente com base em mapas internos. Em 2000 foi incorporado ao projeto um simulador para testes, que empregava realidade virtual. A ideia era que eles fossem realistas e seguros o suficiente para os usuários.

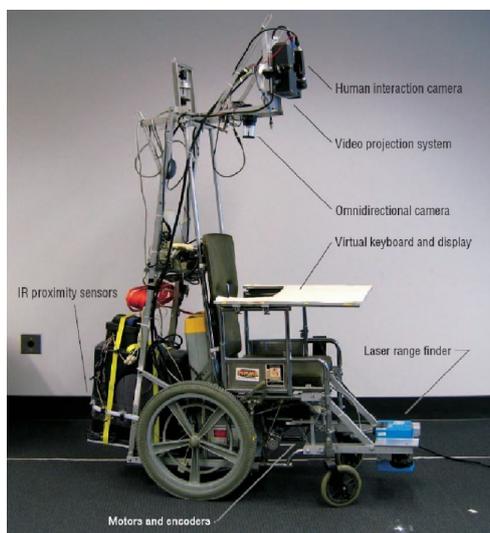


Figura 2.6: Protótipo Smartchair. [11]

Ainda em 1998, o Centro Nacional de Reabilitação para Deficientes do Japão propôs um simulador envolvendo uma plataforma que reproduzia acelerações e desacelerações como as de uma cadeira de rodas elétrica. O estudo realizado mostrou que os usuários acharam a condução em ambientes reais e virtuais bastante parecida, apesar da segunda modalidade apresentar mais dificuldades. Quatro anos depois, uma publicação conjunta do Royal Hospital for Neuro-disability e da University of East London sobre o papel da tecnologia de realidade virtual para avaliação e



Figura 2.7: Versões do protótipo do projeto VAHM. [11]

treinamento de novos usuários de cadeiras de rodas elétricas obteve conclusões similares. O mesmo pode ser dito do simulador McGill, de 2011, que se baseou no Unreal Development Kit e foi empregado para comparar ambientes reais e virtuais utilizando uma cadeira de rodas elétrica.

O simulador Virtual Environment Mobility, de 2005, efetuou testes de condução de uma cadeira elétrica em diferentes cenários virtuais com crianças, e citou a necessidade de adaptação do ambiente virtual às necessidades de cada usuário, que podem ser bastante diferentes. O projeto Virtual Intelligent Wheelchair, de 2007, se dedicava à avaliação da navegação de uma CRI num circuito pré-determinado, mas os testes não utilizavam pessoas para conduzir a cadeira, que tinha movimentos automáticos. A Universidade de Pittsburgh, por sua vez, publicou no ano seguinte um estudo conduzido com vítimas de traumatismo cranioencefálico, utilizando um simulador em duas dimensões. Os objetivos eram testar suas habilidades para guiar cadeiras de rodas e avaliar o desempenho de controles alternativos. Também existe o simulador comercial WheelSim © Life Tool, para treinamento de usuários de cadeiras de rodas comuns. Outro projeto japonês é o ACCoMo (*Autonomous, Cooperative, Collaborative Mobile*, Fig. 2.8), um protótipo voltado para locomo-

ção em ambientes fechados que adotou conceitos de colaboração com seu usuário e cooperação com outras cadeiras. O sistema baseou-se em algoritmos de aprendizagem por reforço, redes neurais e algoritmos genéticos, e realizava a localização e a interação com outros dispositivos por meio de RFID.

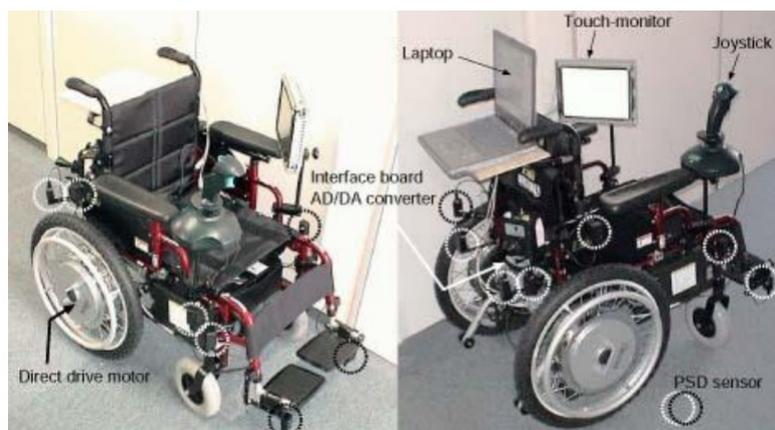


Figura 2.8: Protótipo ACCoMo. [6]

Em Portugal, um dos projetos desenvolvidos foi o RobChair (não confundir com RoboChair, mencionado anteriormente), de 2002, “definido como um robô móvel construído a partir de uma cadeira de rodas motorizada comercial, dotada de sensores” [11]. Ele era capaz de controlar sua velocidade, evitar colisões e desviar de obstáculos. Os dois últimos comportamentos foram implementados com o método reativo de campos potenciais, que, basicamente, projeta campos de repulsão ao redor dos obstáculos utilizando conceitos análogos aos dos campos elétricos da Física. [42] [37]

A seguir são apresentadas, de forma resumida, outras publicações (algumas mais recentes) na área de tecnologias assistivas. Segundo Krishnan e Pugazhenthii [31], grande parte das cadeiras de rodas inteligentes têm sua navegação baseada em EKF SLAM. Ainda assim, eles afirmam que a navegação de CRIs é uma grande área de pesquisa, onde novos e diferentes métodos estão evoluindo.



Figura 2.9: Cadeira que reconhece e segue um acompanhante humano [52] (esq.) e a Essex Wheelchair [34] (dir.).

Cheein *et al.* [3], da Universidade Nacional de San Juan, Argentina, aplicaram o EKF SLAM a um equipamento robótico cuja navegação é comandada através de sinais eletromiográficos, obtidos por meio uma interface músculo-computador (MCI). Os mapas gerados são armazenados e podem ser usados para uma navegação mais segura em revisitações. O robô móvel utilizado possui modelo cinemático similar ao de uma cadeira de rodas motorizada. Desta forma, os pesquisadores acreditam ser possível explorar a aplicação desenvolvida para a navegação autônoma desses equipamentos. Wu *et al.* [52] também empregaram o EKF SLAM para a localização de uma CRI, usando LRFs (*laser range finders*), *encoders* ópticos e um sensor de bússola. Eles propuseram um sistema que a torna capaz de reconhecer e seguir um acompanhante humano, usando dados multissensoriais. O acompanhante é considerado um alvo, que é reconhecido com o uso do algoritmo SURF (*Speeded Up Robust Features*) e de uma câmera PTZ (*Pan-Tilt-Zoom*). Ele é seguido com base nos dados de reconhecimento e mapeamento. A cadeira desenvolvida nesse projeto é mostrada na Fig. 2.9, à esquerda. Já a Essex Wheelchair [34, 35] tem navegação interativa, baseada em sensores e em mapas de múltiplas camadas (*multi-layered maps*), em vez de mapas geométricos (*geometric maps*). A implementação utiliza o framework ROS e permite a criação de mapas 2D com a técnica de *occupancy grid*

(grade de ocupação), usando os dados dos sensores a laser e da hometria da CRI. A cadeira, exibida à direita da Fig. 2.9, também é capaz de planejar rotas e tarefas.

Em setembro de 2014, a Intel anunciou o projeto Cadeira de Rodas Conectada (*Connected Wheelchair Project*³, Fig. 2.10), do seu grupo de Internet das Coisas (*Intel IoT group*) em parceria com o Dr. Stephen Hawking. Tal cadeira foi projetada para funcionar com o kit de desenvolvimento da placa Intel Galileo. Ela é capaz de monitorar sinais vitais do usuário, como batidas cardíacas, respiração, pressão sanguínea e temperatura, e a interação é feita através de um aplicativo instalado em um tablet com tela sensível ao toque.⁴



Figura 2.10: Projeto Connected Wheelchair da Intel. (Foto: Divulgação/Intel)

Na literatura também há publicações brasileiras na área de cadeiras de rodas inteligentes. Em 2010, o Grupo de Robótica de Reabilitação da Universidade Federal do Espírito Santo (UFES) propôs uma estratégia que dava a uma CRI a capacidade de desviar de obstáculos estáticos e móveis num ambiente semiestruturado [15]. As

³<http://www.intel.com/content/www/us/en/internet-of-things/videos/dr-hawkings-connected-wheelchair-video.html>, acessado em dezembro de 2014.

⁴Fonte: <http://www.techtudo.com.br/noticias/noticia/2014/09/stephen-hawking-revela-cadeira-de-rodas-inteligente-da-intel.html>, acessado em dezembro de 2014.

equações elaboradas consideravam suas dimensões físicas, e o simulador de cadeiras de rodas robóticas da UFES foi utilizado no trabalho. Bissoli *et al.* [10] simularam, em 2013, a navegação de uma CRI num ambiente residencial fictício. Tal ambiente era estruturado, e a cadeira tinha conhecimento prévio do mapa. Ela podia planejar caminhos e gerar trajetórias a partir de comandos de voz. A principal preocupação do projeto foi proporcionar uma movimentação confortável para o usuário.

Por sua vez, uma equipe formada por pesquisadores da Universidade Federal de Uberlândia (UFU) em parceria com a Universidade de Lorraine, na França [46], está desenvolvendo um sistema que possibilite controlar uma cadeira de rodas inteligente através de sinais elétricos gerados pelos músculos e pela atividade cerebral do usuário. Seu objetivo é beneficiar pacientes com paralisias que os impossibilitem de mover um joystick. Já a Universidade Federal de Santa Catarina (UFSC) [14] apresentou recentemente um projeto que questiona o fato de que a grande maioria dos projetos no contexto de CRIs preocupa-se somente com o desenvolvimento de tecnologias de hardware e software, sem levar em consideração questões de design, ou seja, da relação entre usuário e objeto. Seu objetivo foi apresentar uma abordagem metodológica para o desenvolvimento do design das cadeiras, tendo em conta os requisitos tecnológicos e de acessibilidade desse tipo de produto.

2.2.1 Plataforma IntellWheels

Apesar da existência de diversos projetos para desenvolvimento de CRIs, Petry [43] afirmou que eles eram restritos a arquiteturas de hardware e software específicas para a cadeira utilizada na pesquisa. Em outras palavras, não havia um modelo genérico para transformar cadeiras de rodas elétricas em CRIs. Outros pontos negativos na maioria desses projetos eram os custos elevados e a falta de simplicidade das configurações, o que diminuía seu público em potencial.

O projeto IntellWheels surgiu na tentativa de se melhorar este quadro, com foco na criação de uma plataforma para desenvolvimento de cadeiras de rodas inteligentes com interface multimodal flexível. Tal interface foi concebida de forma a facilitar o desenvolvimento e os testes de novas técnicas e metodologias, exigindo o mínimo possível de modificações nas cadeiras de rodas elétricas às quais fosse integrada. O projeto foi uma iniciativa da Faculdade de Engenharia da Universidade do Porto (FEUP) em parceria com as seguintes instituições: Laboratório de Inteligência Artificial e Ciência da Computação (LIACC); INESC Tecnologia e Ciência (INESC-TEC); Instituto de Engenharia Eletrônica e Telemática de Aveiro (IEETA); Escola Superior de Tecnologia da Saúde do Porto (ESTSP); Universidade do Minho (UMINHO); e Associação do Porto de Paralisia Cerebral (APPC).

A modelagem do IntellWheels baseou-se na arquitetura de Sistema Multiagentes (SMA). Dentre outros benefícios, ela facilita a incorporação de novas funcionalidades à cadeira e possibilita a auto-organização do sistema [11] [43]. Este último é, basicamente, um mecanismo apresentado por um SMA que lhe permite reconfigurar sua organização através da interação e do comportamento individual de componentes muito simples, sem comandos externos. Deste processo emana um comportamento coletivo que resulta, por exemplo, no cumprimento das tarefas estipuladas para o sistema. [45]

O paradigma de SMA adota os conceitos de micro e macro perspectivas. Assim, de acordo com a primeira, cada cadeira é vista como um conjunto de vários micro agentes (controle, interface etc.). Já na segunda, cada cadeira como um todo é considerada um macro agente, assim como outros elementos do sistema. Ou seja, portas inteligentes também são macro agentes nesta perspectiva, por exemplo. Tal estratégia é interessante por viabilizar a utilização de diversos módulos independentes, mas interoperantes.

Uma vez que as limitações físicas que podem acometer os usuários de cadeiras de rodas são bastante diversas e específicas, um dos principais focos do projeto IntellWheels foi a adaptabilidade dos comandos da CRI. O conceito de Interface Multimodal (MMI – *Multimodal Interface*) foi adotado na plataforma a fim de atingir este objetivo [43]. Oviatt [41] afirmou que este tipo de interface busca prover meios de interação humano-computador mais transparentes, flexíveis e eficientes. Sua base é a identificação da linguagem e do comportamento naturais do ser humano com a incorporação de tecnologias de reconhecimento (como fala, toque, movimentos corporais etc.). Desta forma, espera-se que as MMIs sejam mais fáceis e intuitivas de se usar, bem como mais estáveis e robustas do que interfaces unimodais.

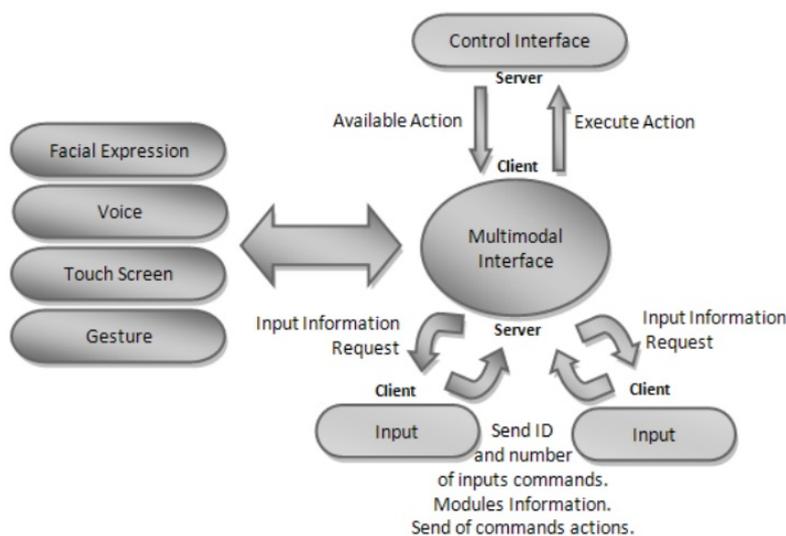


Figura 2.11: Arquitetura da interface multimodal flexível da IWP [22].

A MMI desenvolvida para o IntellWheels permite a conexão simultânea e a interação de diversos dispositivos de entrada. De forma geral, eles podem ser configurados para fornecer à CRI informações sobre velocidade e direção, ou para requisitar ações mais complexas, que incluem comandos como “seguir em frente”, “virar à direita”, “virar à esquerda”, “ir ao quarto” e “ir ao elevador”. O esquema da arquitetura da MMI flexível da Plataforma IntellWheels pode ser visto na Fig. 2.11. [13] [43]

Foram analisadas oito modalidades de entradas de usuário para a plataforma, descritas por Petry [43] e mencionadas a seguir. Uma possibilidade é (i) o joystick tradicional, embutido na própria cadeira, ou (ii) um joystick USB tipicamente usado em jogos, cujos botões podem ser customizados para disparar ações da cadeira. (iii) Uma tela sensível ao toque e (iv) um teclado são as interfaces para configuração da CRI, o que pode ser feito pelo próprio usuário ou pelos profissionais de saúde e apoio. (v) Os movimentos de cabeça são mais um tipo de entrada, e são percebidos por um acelerômetro de três eixos implantado em um boné. Este dispositivo, que se comunica com o computador via Bluetooth, pode ser usado para disparar comandos ou para conduzir a cadeira. Neste caso, as velocidades linear e angular da CRI são determinadas pela posição da cabeça do usuário. O sistema conta, ainda, com (vi) reconhecimento de expressões faciais simples, que podem ser relacionadas a ações. Este módulo de entrada utiliza uma câmera digital comum e um perceptron multicamadas (MLP) para identificar as expressões. Para o usuário, os comandos se traduzem em piscar o olho direito ou esquerdo, abrir a boca, levantar a sobrancelha direita ou esquerda etc. [24]. Apesar de proporcionar uma condução muito confortável de forma geral, tal módulo depende de iluminação estável no ambiente para operar com robustez [22]. Além desses, a plataforma aceita (vii) comandos de voz, cujo processamento é feito pelo Microsoft Speech API (SAPI) e que podem ser usados para a execução de ações. Por fim, também foi testada a incorporação de (viii) uma interface cérebro-computador comercial para reconhecimento de expressões faciais e pensamentos. Até a conclusão deste trabalho, entretanto, as pesquisas indicaram que tal dispositivo não era suficientemente preciso para proporcionar um controle robusto e seguro da IntellWheels. [23]

Mais do que dar liberdade para o usuário escolher quais módulos de entrada deseja utilizar, a Plataforma IntellWheels também permite que ele defina sequências de entradas para comandar a cadeira. Em outras palavras, uma sequência de estímulos de entrada do usuário pode ser associada a uma ação da cadeira. Além disso,

a sequência pode ser formada por entradas de diferentes dispositivos. Por exemplo, piscar o olho direito uma vez e apertar o botão 1 do joystick pode indicar à cadeira o comando virar à direita. Com isso, o usuário é capaz de estabelecer combinações entre os estímulos que consegue realizar para customizar o comando da CRI, criando sua própria linguagem de comunicação. A intenção foi proporcionar ao usuário uma condução mais confortável e adequada às suas condições. [43]

A interação com o usuário é feita através de uma interface gráfica (GUI – *Graphical User Interface*), apresentada nas Figuras 2.12 e 2.13. No seu topo há botões que permitem, por exemplo, desligá-la ou personalizar as configurações da IWP: definir as sequências de estímulos de entrada e associá-las aos comandos da CRI, bem como alterar associações já gravadas. Logo abaixo está o assistente de navegação, personalizado amigavelmente como Anna, cuja função é exibir informações como funcionalidades ativas e em que modo a cadeira está operando – manual, automático ou híbrido. Já o grupo de ícones no canto inferior direito da GUI indica quais dispositivos de entrada podem ser usados. Na Fig 2.12, por exemplo, o joystick externo não está disponível. Na Fig 2.13 é possível observar como as indicações de direção da cadeira são fornecidas, e um arco cujos setores coloridos indicam a velocidade com que a CRI está se movendo. Além disso, abaixo desse arco é informada graficamente qual sequência de entradas está sendo executada no momento (imagem à esquerda).

O projeto utilizou uma cadeira de rodas elétrica comum: estrutura metálica, assento, joystick, bateria e quatro rodas, duas delas dotadas de motores independentes. Entretanto, o conceito de cadeira de rodas inteligente exige, no mínimo, as capacidades de perceber seu entorno e de responder a comandos do usuário e a mudanças que ocorram no ambiente. Para tanto, sensores e dispositivos de controle computacional podem-lhe ser agregados. No projeto IntellWheels esse hardware adicional foi chamado de framework físico [43]. Ao mesmo tempo, houve a preocupação



Figura 2.12: Interface gráfica da IWP [22].



Figura 2.13: Interface gráfica da IWP, com destaque para as indicações de direção e velocidade da CRI [22].

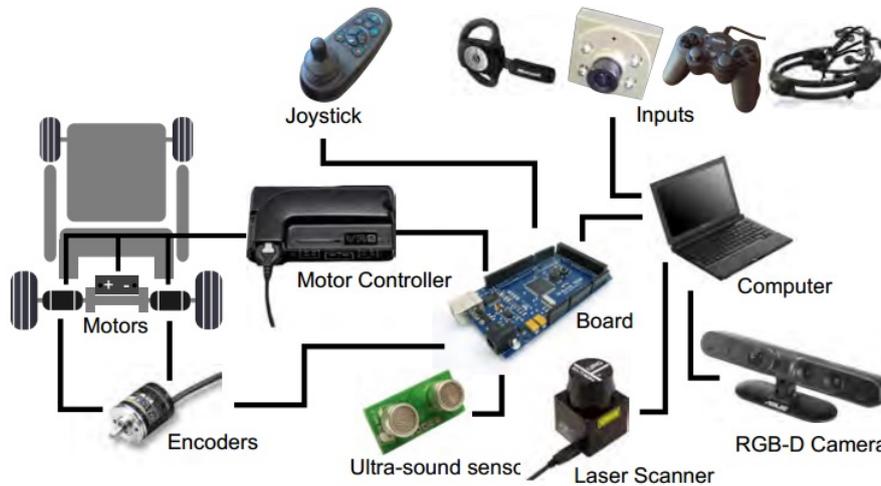


Figura 2.14: Arquitetura do framework físico da Plataforma IntellWheels [43].

de se alterar minimamente o design da cadeira de rodas, o que restringiu a quantidade e o tipo desses elementos adicionais, mas conferiu ao framework a capacidade de ser facilmente acoplado a cadeiras elétricas comuns [22]. Exemplo disso são suas duas barras laterais de sensores feitas em alumínio [43]. Cada uma delas abriga oito sensores ultrassônicos, e a barra esquerda também contém um *laser range finder* (LRF). Sua cor preta e seu formato tornam o conjunto discreto e harmonioso com o design usual das cadeiras de rodas motorizadas. O framework físico também conta com dois *encoders* acoplados a cada um dos motores. Os dados obtidos por eles são usados para estimar o deslocamento e a localização relativa da cadeira. Seu sistema de localização se baseia em técnicas de navegação estimada (*dead reckoning*), que usa as informações do histórico de leituras dos sensores e do estado inicial da CRI para estimar sua localização atual. A Fig. 2.14 mostra os elementos da arquitetura do framework físico da IWP, enquanto alguns detalhes do protótipo da cadeira IntellWheels são destacados nas Figuras 2.15 e 2.16.

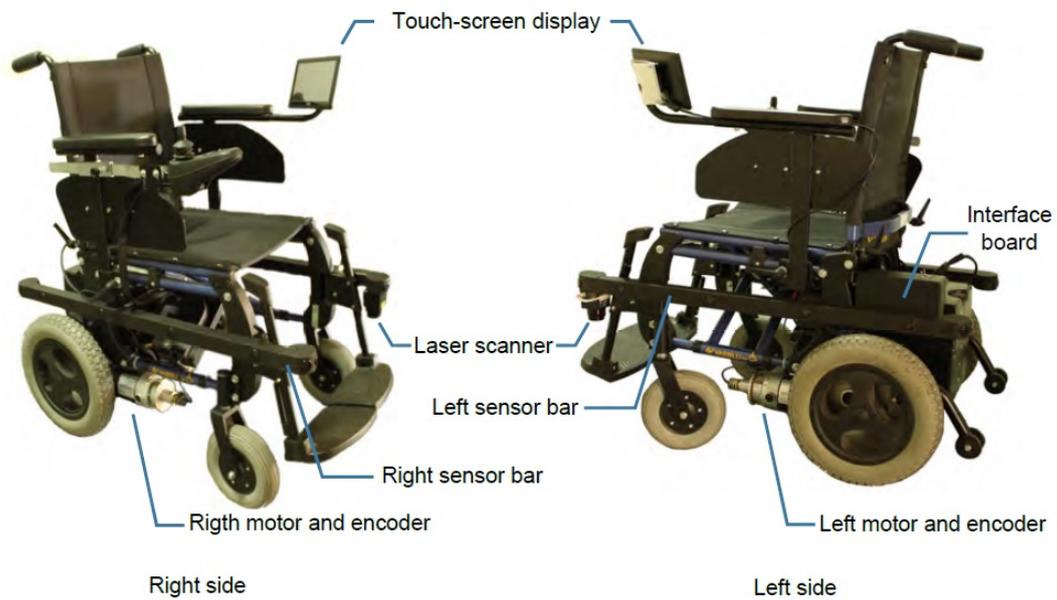


Figura 2.15: Detalhes do protótipo da cadeira IntelWheels [43].



Figura 2.16: Detalhes das partes frontal (esq.) e traseira (dir.) da barra esquerda de sensores [43].

A Plataforma IntellWheels também conta com um simulador, o IntellSim. Muito embora o objetivo primordial do projeto fosse integrar os componentes elaborados a uma CRI real, também era importante mantê-los compatíveis com o simulador IntellSim. Isso poderia facilitar a realização de testes, inclusive com a participação de pessoas. Sua versão original era um modelo 2D adaptado do sistema de simulação de código aberto Ciber-Rato (ou Cyber-Mouse, em inglês). Este sistema é a modalidade virtual da competição anual de robótica Micro-Rato, organizada pela Universidade de Aveiro. Seus competidores são desafiados a desenvolver um time de robôs móveis autônomos capaz de solucionar um labirinto, levando em consideração dois objetivos. Primeiro o time precisa encontrar um farol localizado na área alvo da arena. Uma vez que todos a tenham alcançado, cada um dos robôs precisa retornar a sua posição de partida. O sistema se baseia em computação distribuída, com uma arquitetura cliente-servidor estruturada em três entidades: o simulador, que atua como servidor; o visualizador; e os agentes robóticos – estes sendo considerados clientes. [5] [51] [32]

A versão atual do IntellSim é um modelo em três dimensões construído com o Unreal Tournament 2004 e o USARSim 3.1.3 (*Unified System for Automatic and Robot Simulation*), enquanto seu cenário 3D baseou-se no Unreal 2004 Editor. O USARSim é um simulador com alta fidelidade para robôs e ambientes, baseado no motor de jogos Unreal Tournament. A versão empregada na IWP baseia-se no Unreal Engine 2.5 junto com o motor de física Karma. As locações modeladas para o IntellSim são instalações do prédio da APPC (Associação do Porto de Paralisia Cerebral, Fig. 2.17), pois os usuários que participaram dos testes são pacientes da Associação. A intenção foi proporcionar a eles uma experiência de condução da CRI virtual em ambientes familiares, o que facilitaria o processo de aprendizagem. [22]

Ainda segundo Faria [22], no primeiro simulador (baseado no Ciber-Rato) o ambiente carecia de mais realismo e o comportamento da CRI virtual precisava



Figura 2.17: Foto da APPC à direita e seu modelo virtual à esquerda [22].

ser melhorado. A escolha do Usarsim deveu-se, em especial, a tais necessidades e ao princípio de diminuir ao máximo possível os custos do projeto IntellWheels. Algumas de suas características também influenciaram tal preferência, como o suporte avançado a robôs com rodas; a capacidade de se programar e controlar os robôs via rede, permitindo experimentos com realidade mista; e a possibilidade de se importar objetos e robôs modelados em outras plataformas. Desta forma, diversos componentes do cenário 3D (reprodução de instalações da APPC) e a própria cadeira virtual (Fig. 2.18) puderam ser modelados no 3DStudioMax e incorporados no Unreal Editor como malhas estáticas. [22] [43]



Figura 2.18: Cadeira de rodas virtual [22].

Outra característica interessante do USARSim é a possibilidade de se observar as cenas a partir de uma perspectiva egocêntrica (do ponto de vista do robô) ou em terceira pessoa. Seu sistema é estruturado como uma arquitetura cliente-servidor, que permite executar mais de um robô simultaneamente. Do lado cliente estão o cliente Unreal e o controlador, ou outra aplicação incorporada pelo usuário. Já no lado servidor estão presentes o motor Unreal, o Gamebots (uma modificação do Unreal Tournament), mapas e modelos de objetos (como os robôs). [22]

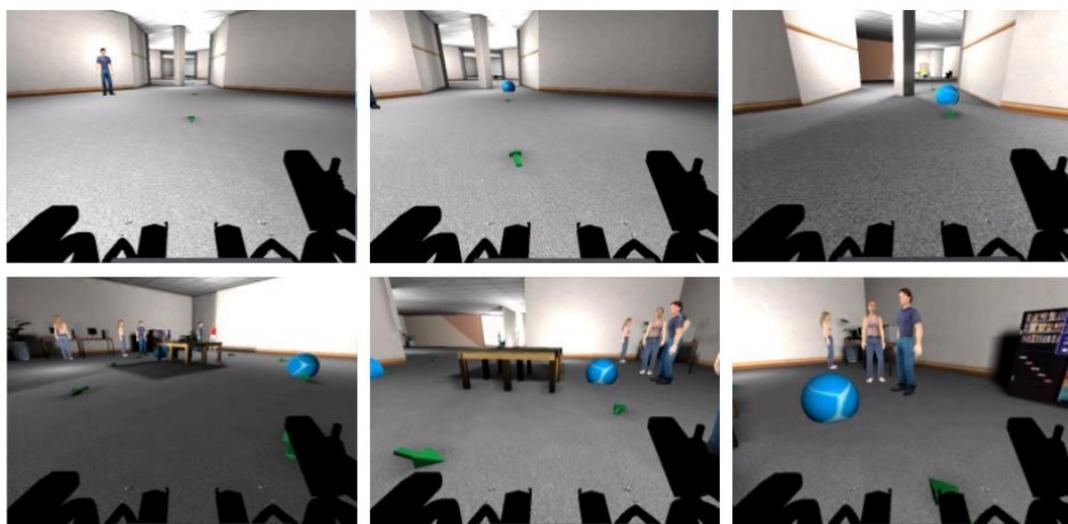


Figura 2.19: Perspectiva em primeira pessoa [22].

Foi construído um jogo sério voltado para o ensino das habilidades necessárias para a condução de cadeiras de rodas inteligentes. Ele contribuiu para que o IntellSim fosse um simulador 3D realista, permitindo diferentes modos de condução em cenários verossímeis. O jogo foi criado com Processing, uma linguagem de programação de código aberto e IDE de contexto visual [48]. Ele permite a definição de circuitos e a colocação de marcos ao longo do caminho. Esses objetos, em formato de bola ou estrela, devem ser capturados pelo usuário para ganhar pontos. Algumas medidas de desempenho dos usuários também podem ser coletadas através do jogo,

como o tempo gasto para completar o percurso, a precisão com que este é feito e o número de colisões. A Fig. 2.19 mostra cenas desse jogo e a perspectiva em primeira pessoa, citada anteriormente.

É importante citar que a Plataforma IntellWheels, além de permitir o trabalho em modos puramente real ou virtual, incorporou o conceito de realidade mista: as cadeiras reais e virtuais podem interagir, com seus ambientes sendo considerados como um só. Isso viabiliza uma maior diminuição de custos ao dispensar a construção de muitos exemplares da cadeira real para a execução de testes mais complexos. [12] [43]

3 SISTEMA DE MAPEAMENTO COLABORATIVO COM CADEIRAS DE RODAS INTELIGENTES

A ideia inicial para esta dissertação era estudar a agregação de um módulo SLAM a uma cadeira de rodas inteligente – especificamente, a Plataforma IntellWheels. Entretanto, o levantamento bibliográfico revelou que esse tema já vem sendo explorado nos últimos anos, a exemplo dos trabalhos de Cheein *et al.* [3], Li *et al.* [35, 34] e Wu *et al.* [52]. Além disso, fatores como o tempo para execução da pesquisa e a disponibilidade de ferramentas fariam necessárias simplificações que diminuiriam a contribuição a ser gerada.

A partir disso, pensou-se na possibilidade de se aplicar alguma técnica de SLAM Colaborativo a CRIs. Foi realizado um novo levantamento bibliográfico, o qual sugeriu que a ideia de “colaboração” nesse contexto está ligada à colaboração entre a cadeira, outros dispositivos robóticos, a equipe de cuidados e o próprio cadeirante, visando ao seu bem-estar [34] [52] [53]. Mais do que isso, não foram encontradas referências que tratassem da colaboração entre CRIs para fins de mapeamento.

O capítulo começa pela descrição do conceito do sistema de mapeamento elaborado nesta dissertação (Seção 3.1). Em seguida, na mesma seção, são apontados os critérios de escolha da estratégia C-SLAM utilizada. Por fim, na Seção 3.2, as características mais relevantes de tal estratégia são resumidas.

3.1 Conceito do Sistema e Escolha do Método C-SLAM

Cadeiras de rodas inteligentes, em essência, são agentes robóticos móveis que possuem sensores capazes de obter dados do ambiente. Uma CRI pode utilizar essas informações para construir mapas dos locais por onde navega e armazená-los. Entretanto, o mapeamento é feito individualmente e, em geral, seu resultado não é compartilhado com outras CRIs e tampouco com o próprio ambiente, o que apresenta certas desvantagens. Um motivo simples é que toda cadeira que passe pela primeira vez por determinada área terá de mapeá-la – um esforço desnecessário caso o conhecimento produzido anteriormente fosse compartilhado. Além disso, conforme exposto na Seção 2.1, dispositivos de sensoriamento são sujeitos a ruídos, o que interfere diretamente no mapeamento. Quando se recorre a técnicas como a fusão de dados de mais de um agente, as chances de que o mapa gerado seja mais preciso são maiores. Também é relevante o fato de que o ambiente em si poderia se beneficiar desses mapas, cuja manutenção muitas vezes é interessante do ponto de vista administrativo. Pode-se mencionar, ainda, a possibilidade de que os mapas guardados pelas CRIs tornem-se defasados em decorrência de reformas e mudanças na disposição da mobília, diminuindo sua confiabilidade.

Diante desses fatores, decidiu-se averiguar a possibilidade de se construir mapas de forma colaborativa entre cadeiras de rodas inteligentes, num sistema capaz de apoiar tal colaboração, mantendo esse material o mais atualizado possível e disponível para qualquer CRI ou outro dispositivo que dele necessite. Esse sistema de colaboração é composto por CRIs e por uma **entidade de supervisão**, o que, inicialmente, requer que o ambiente esteja preparado para dar suporte a tal estrutura. Ao ingressar no ambiente, a cadeira efetua um procedimento de “credenciamento” no sistema, chamado de *check-in*. Nessa etapa anterior à navegação da CRI acontece a troca de informações com a supervisão, quando lhe é atribuído um identificador. Caso já tenha havido visitas de CRIs anteriormente, ela também tem acesso a um

mapa do ambiente (que pode ou não estar completo) e ao seu indicador de confiabilidade. O indicador é gerado pela supervisão, e seu papel é ajudar a cadeira a determinar se pode ou não basear-se naquele mapa durante sua navegação. O mapeamento é realizado por uma ou mais CRIs simultaneamente através da abordagem de SLAM Colaborativo, cujo processo de escolha é apresentado mais à frente nesta Seção. Alguns critérios tiveram de ser levados em consideração, como a possibilidade de se realizar SLAM mesmo que houvesse apenas um agente em operação. Quando uma CRI sai do ambiente é realizado seu “descredenciamento” do sistema, o *check-out*: ela entrega à entidade de supervisão os dados obtidos durante sua navegação. Dessa forma, espera-se manter uma relação de troca, onde cadeiras de rodas inteligentes construam e atualizem mapas de ambientes ao mesmo tempo em que, com certo nível de confiança, se beneficiam da existência desse material para suas funções de navegação e planejamento.

Outra questão abordada neste trabalho foi a verificação da hipótese levantada quanto às informações necessárias para que os agentes se apropriassem dos mapas prévios em sua navegação. Supôs-se que seria suficiente uma representação simples, composta pelo vetor de estado e pela matriz de covariância finais do mapa armazenado pela supervisão e pela pose inicial do agente que estivesse entrando no ambiente. Tal hipótese, a princípio, se justificaria pelo fato de que os modelos sensorial e de hometria da Plataforma IntellWheels são padronizados. O levantamento bibliográfico trouxe uma referência que utiliza estratégia parecida com esta, num artigo de 2009 de Barkby *et al.* [7], porém com o mapa estruturado em grid.

Obviamente, a prioridade de uma cadeira de rodas não é e nem deve ser a função de mapeamento, devido à sua finalidade e por questões como segurança e robustez. Portanto, não é esperado que todo o ambiente seja mapeado. Ainda assim, as CRIs poderão contar com tais mapas nas áreas que forem processadas, melhorando seu desempenho. Também é relevante mencionar um conceito citado na

Seção 2.2, de que “uma CRI deve interferir minimamente no ambiente em que opera para ter um bom desempenho”. De fato, o sistema proposto nesta pesquisa exige que o ambiente seja preparado para suportá-lo, mas o funcionamento propriamente dito das cadeiras não interfere nele.

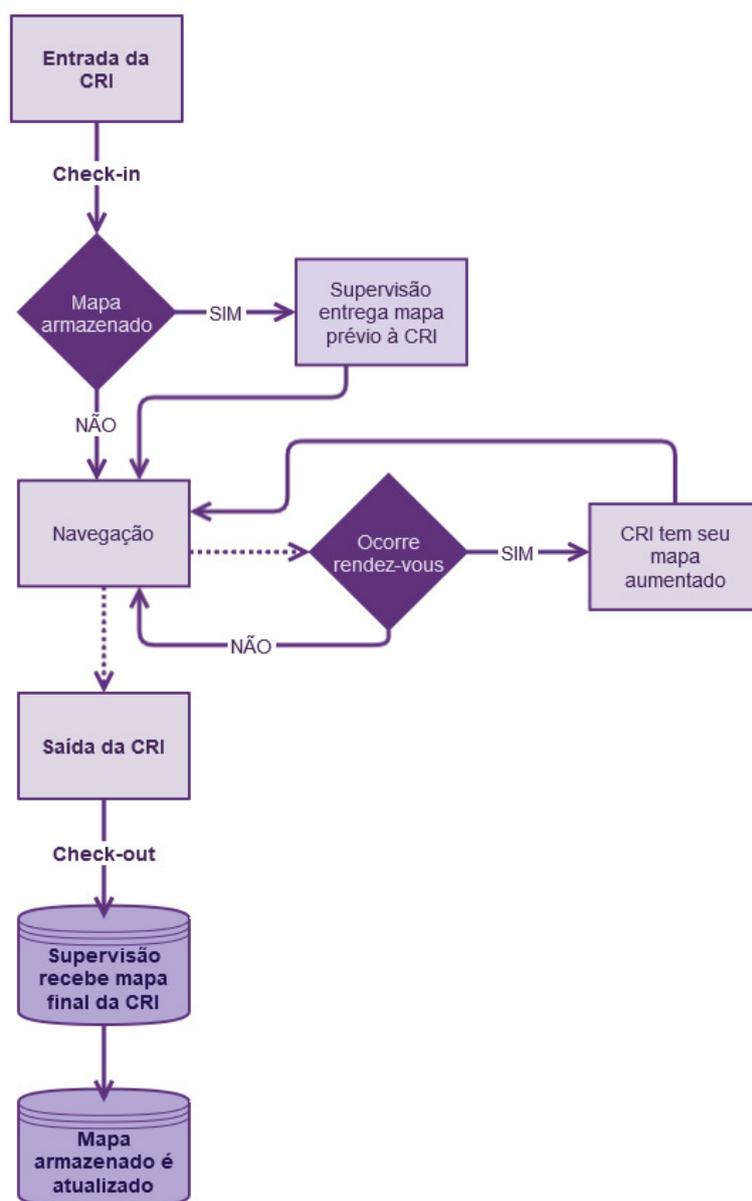


Figura 3.1: Fluxograma do sistema proposto.

Um fluxograma ilustrando o funcionamento do sistema com relação a uma CRI qualquer que dê entrada no ambiente é mostrado na Fig. 3.1. As linhas pontilhadas a partir do evento “Navegação” denotam que, durante esse processo, pode ou não ocorrer o evento de *rendez-vous* (um encontro entre duas CRIs que pode propiciar a fusão de mapas, conforme será detalhado adiante), até que a CRI deixe o ambiente. Com o *check-out*, a supervisão recebe o mapa final gerado pela CRI em questão, que é utilizado para atualizar o mapa armazenado em sua base de dados.

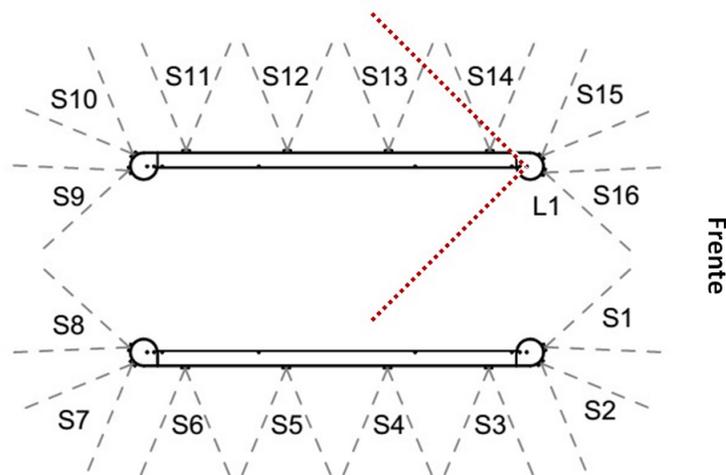


Figura 3.2: Disposição dos sensores de proximidade ao redor da IntellWheels. Adaptado de [43].

Após a conceituação do sistema, o passo seguinte foi a definição da estratégia de SLAM Colaborativo para a realização do mapeamento. Algumas referências foram coletadas, mas era necessário estabelecer critérios que permitissem escolher uma delas. O ponto de partida foi o modelo robótico da cadeira IntellWheels, que serve de base para o agente robótico do sistema proposto neste trabalho e cujas características deveriam ser compatíveis com a estratégia C-SLAM adotada. Sua hodometria é obtida através de 2 *encoders* conectados ao eixo do motor de cada uma das 2 rodas traseiras. A percepção do ambiente se dá através de 17 sensores de proximidade – 1 scanner 2D a laser e 16 sonares, dispostos nas barras laterais conforme o esquema da Fig. 3.2 [43]. L1, na parte frontal da barra esquerda, é o sensor a laser, que possui um FOV (do inglês *field of view*, ou campo de visão) de 270°,

representado pelo pontilhado vermelho. Os demais são os sensores ultrassônicos, com FOV de 45° (tracejados pretos).

Dentre os artigos pesquisados havia abordagens baseadas em FastSLAM [19], Grafos [33] e *Random Finite Set* (RFS) [40], que não foram escolhidas mas cujas referências são deixadas aqui a título de informação, caso o leitor tenha interesse em conhecer outras possibilidades para SLAM Colaborativo. O método selecionado foi desenvolvido por Benedettelli, Garulli, Giannitrapani [9], que adapta o **EKF-SLAM** para a **representação M-Space de *features* lineares**. Duas outras fontes utilizadas de forma auxiliar foram os trabalhos de Benedettelli [8] e de Zhou, Roumeliotis [54].

No método escolhido, o ambiente é representado na forma de *features* lineares, apropriada para lidar com características que podem ser descritas em termos de linhas e segmentos. É o caso daquelas extraídas de paredes, portas e mobília em ambientes internos (*indoor*). Há diferentes maneiras de se definir um segmento de linha, e a referência utilizada o faz através da representação M-Space [26]. Entretanto, é possível adotar a mesma abordagem C-SLAM descrevendo as linhas por seus pontos extremos. Essa, por sua vez, é a representação empregada nesta dissertação. Deste ponto do texto em diante será usada a notação a seguir.

- Vetores: símbolos em negrito.
- *Frames* de referência: $\langle F \rangle$.
- Estimativas de vetores: $\hat{\mathbf{x}}$.
- Erros de estimativa: $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$
- Elementos expressos no *frame* de um robô específico: ${}^f \mathbf{x}_r$. O índice sobrescrito

f , à esquerda, indica que \mathbf{x} está sendo considerado no *frame* $\langle R_f \rangle$, correspondente ao robô R_f . Já o índice subscrito r informa que \mathbf{x} diz respeito ao robô R_r .

- Matrizes diagonais são denotadas sob a forma $diag(a_1, \dots, a_n)$, sendo os parâmetros a_1, \dots, a_n os elementos escalares de sua diagonal.

3.2 EKF-SLAM Colaborativo com Features Lineares

Começando pela maneira como as características do ambiente são representadas, uma *feature* linear nada mais é do que um segmento de linha. Pode-se dizer que a forma mais simples e intuitiva de descrevê-la é determinar as coordenadas de seus pontos extremos com relação a um **frame de referência global (FRG)** ou **espaço da feature**, chamado de $\langle G \rangle$:

$$\mathbf{x}_f = [x_A \ y_A \ x_B \ y_B]^T. \quad (3.1)$$

Assim, \mathbf{x}_f é o vetor de **coordenadas globais da feature**.

Considerando-se um robô autônomo navegando em um ambiente 2D, sua pose pode ser expressa no FRG $\langle G \rangle$ na forma

$$\mathbf{x}_r = [x_r \ y_r \ \theta_r]^T, \quad (3.2)$$

sendo $[x_r \ y_r]^T$ sua posição e θ_r sua orientação relativa a $\langle G \rangle$.

A percepção de mundo do robô, efetuada através de sensores, é dada com relação a ele mesmo, configurando outro *frame* de referência, o qual será chamado de $\langle R \rangle$: o **frame de referência do robô (FRR)**. As coordenadas globais da *feature*, por sua vez, podem ser expressas em $\langle R \rangle$, quando passam a ser chamadas de $\mathbf{x}_o = [x_A^r \ y_A^r \ x_B^r \ y_B^r]^T$. A Fig. 3.3 ilustra o que foi exposto até então.

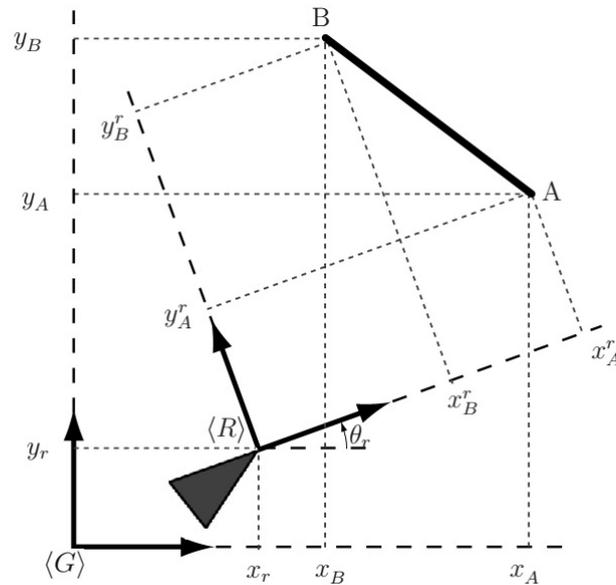


Figura 3.3: Ilustração da pose do robô no FRG e das coordenadas globais da *feature* em $\langle G \rangle$ e $\langle R \rangle$. Adaptado de [8].

A representação especial chamada M-Space (do inglês *Measurement Subspace*), ou subespaço de medição, proposta por Folkesson, Jensfelt, Christensen [26], introduz um *frame* de referência local a cada *feature*, fazendo com que *features* diferentes possuam *frames* de referência distintos, e trabalha com subespaços de medição – uma abstração da parte do espaço da *feature* que é detectada pelos sensores do robô. Seu intuito é prover uma alternativa para se lidar com certos problemas, comuns em robótica, gerados por outros tipos de representação. Um deles é a observação parcial, que acontece se o robô não for capaz de detectar um segmento de ponta a ponta numa única leitura de seus sensores. Outro problema é o dito “efeito do braço de alavanca” (ou “*lever arm effect*”), que diz respeito ao deslocamento do segmento estimado em decorrência de erros nas medições dos sensores [20] [26]. Quando o M-Space é utilizado, o EKF-SLAM precisa sofrer algumas adaptações a fim de que se inclua a descrição do vetor de estado no subespaço de medição. Além disso, enquanto a incerteza sobre a pose do robô é expressa no FRG, a incerteza sobre as *features* é expressa no M-Space.

O método C-SLAM elaborado por Benedettelli, Garulli, Giannitrapani [9] pode ser compreendido como uma sequência de ciclos de três etapas. Inicialmente o agente navega enquanto executa o *Single-robot EKF-SLAM* localmente, construindo independentemente um mapa local, até encontrar outro robô. Esse momento marca a segunda etapa: o *rendez-vous*, quando ocorre a fusão dos mapas locais dos dois agentes envolvidos. Assim, cada um fica com uma cópia do mapa resultante. Em seguida tem-se a terceira etapa, quando os agentes prosseguem a navegação, novamente executando o *Single-robot SLAM*. A diferença é que, agora, seus mapas locais são o resultado do processo de fusão de dados.

3.2.1 SLAM com Múltiplos Robôs e Fusão de Mapas

Estejam dois robôs, R_1 e R_2 , a explorar um ambiente, cada um executando o *Single-robot SLAM* localmente. Além de ser possível que o mapa de cada um contenha diferentes informações, em geral os *frames* de referência utilizados são diferentes. Isso acontece porque cada robô adota um FRG com base em sua posição inicial, que não necessariamente é a mesma para todos. No sistema apresentado, a troca de informações entre os robôs e a subsequente fusão de mapas acontecem sob o conceito do *rendez-vous*, que costuma ser executado em pares, independentemente da quantidade total de agentes. A fusão de mapas, entretanto, depende de que se expresse o mapa construído por um robô no *frame* do outro robô. Assim, é essencial que se defina a relação entre os *frames* dos envolvidos, o que pode ser feito com dados de proximidade mútua. Em outras palavras, R_1 realiza medições com relação ao seu *frame* de referência $\langle R_1 \rangle$ que permitem estimar a localização do parceiro R_2 e expressar seu mapa local em $\langle R_1 \rangle$, e vice-versa.

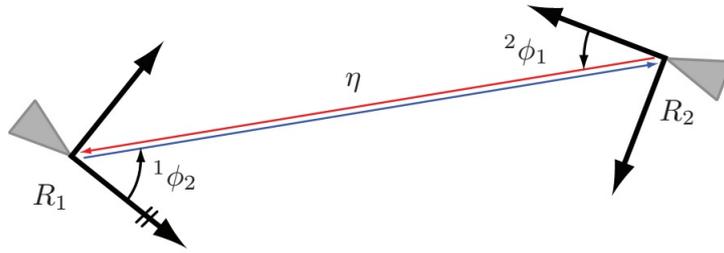


Figura 3.4: Observação mútua no *rendez-vous*. [9]

Pode-se dividir a fusão de mapas em três etapas. O primeiro passo é o **alinhamento dos mapas**, quando a roto-translação entre os *frames* $\langle G_1 \rangle$ e $\langle G_2 \rangle$ é encontrada. Em seguida, a **matriz de covariância do erro de estimativa** do novo mapa é atualizada para cada robô, mais uma vez considerando a relação entre *frames*. Por fim, os mapas gerados são refinados visando ao aumento de sua precisão através da **remoção de dados duplicados**. Isso é útil e necessário porque há chance de que os mapas originais acabem se sobrepondo em algumas regiões.

Sejam ${}^1\mathbf{x}_1$ e ${}^2\mathbf{x}_2$ as estimativas do vetor de estado dos robôs R_1 e R_2 , respectivamente. Assim, suas dimensões são $m_1 = 3 + 4n_1$ e $m_2 = 3 + 4n_2$, onde n_1 e n_2 são a quantidade de marcos dos respectivos mapas. Nos parágrafos que seguem o processo de fusão de mapas será considerado, como exemplo, do ponto de vista do robô R_1 . É necessário estimar ${}^1\mathbf{x}_2$ – o mapa de R_2 expresso no *frame* $\langle G_1 \rangle$. A observação mútua entre R_1 e R_2 é ilustrada pela Fig. 3.4, onde η é a distância entre ambos e ${}^i\phi_j$, $i, j = 1, 2$, $i \neq j$ é o ângulo sob o qual R_i observa R_j . A distância η é composta pela média das posições relativas dos robôs, ${}^1\eta_2$ e ${}^2\eta_1$, ponderada pelos valores de suas variâncias, conforme a Eq. (3.3).

$$\eta = \frac{\sigma_{2\eta_1}^2 * {}^1\eta_2}{\sigma_{1\eta_2}^2 + \sigma_{2\eta_1}^2} + \frac{\sigma_{1\eta_2}^2 * {}^2\eta_1}{\sigma_{1\eta_2}^2 + \sigma_{2\eta_1}^2} \quad (3.3)$$

Há, ainda, os erros de medição ε_η e $\varepsilon_{i\phi_j}$ – ruídos brancos de média 0. Esses elementos de medição mútua dão origem ao **vetor de medição combinada**

$$\bar{\mathbf{z}} = [\eta \quad {}^1\phi_2 \quad {}^2\phi_1]^T + [\varepsilon_\eta \quad \varepsilon_{1\phi_2} \quad \varepsilon_{2\phi_1}]^T = \mathbf{z} + \varepsilon_{\mathbf{z}}, \quad (3.4)$$

onde $\varepsilon_{\mathbf{z}}$ é um ruído branco de medição cuja matriz de covariância é

$$R_{\mathbf{z}} = E[\varepsilon_{\mathbf{z}} \varepsilon_{\mathbf{z}}^T] = \text{diag}(\sigma_\eta^2, \sigma_{1\phi_2}^2, \sigma_{2\phi_1}^2). \quad (3.5)$$

Como as posições relativas dos agentes são medições independentes, a variância σ_η^2 pode ser obtida pela relação a seguir:

$$\sigma_\eta^2 = \frac{\sigma_{1\eta_2}^2 * \sigma_{2\eta_1}^2}{\sigma_{1\eta_2}^2 + \sigma_{2\eta_1}^2}. \quad (3.6)$$

Enfim, transformações geométricas permitem estimar a roto-translação $t(\cdot)$ entre os *frames* $\langle R_1 \rangle$ e $\langle R_2 \rangle$, sendo similar o procedimento para estimar ${}^2\mathbf{x}_1$:

$${}^1\hat{\mathbf{x}}_2 = t({}^1\hat{\mathbf{x}}_1, {}^2\hat{\mathbf{x}}_2, \bar{\mathbf{z}}). \quad (3.7)$$

Após o alinhamento os dois robôs possuem o mesmo mapa, expresso em seus próprios *frames* de referência. Antes que eles possam retornar à execução individual do *Single-robot SLAM*, entretanto, é necessário atualizar suas incertezas para que correspondam ao novo mapa. Em outras palavras, também é preciso alinhar as matrizes de covariância antigas, de acordo com as transformações aplicadas aos mapas, para que se encontre a matriz de covariância do novo vetor de estado.

Inicialmente, o que se tem são as informações que dizem respeito aos mapas locais anteriores à fusão. Assim, pode-se juntar os dois vetores de estado no vetor $\mathbf{X} = [{}^1\mathbf{x}_1^T \quad {}^2\mathbf{x}_2^T]^T$, descrito nos espaços das *features*. A matriz de covariância do erro de estimativa é dada por

$$P = E[\tilde{\mathbf{X}} \tilde{\mathbf{X}}^T]. \quad (3.8)$$

Continuando do ponto de vista do robô R_1 , defina-se o **vetor de estado alinhado (ou aumentado)** $\mathbf{X}^a = [{}^1\mathbf{x}_1^T \quad {}^1\mathbf{x}_2^T]^T$. Seu erro de estimativa é $\tilde{\mathbf{X}}^a$, cuja matriz de covariância é a que segue na Eq. (3.9).

$$P^a = E[\tilde{\mathbf{X}}^a (\tilde{\mathbf{X}}^a)^T] \quad (3.9)$$

É necessário, então, encontrar a matriz de covariância do estado correspondente ao mapa gerado pelo alinhamento, P^a , no que é utilizada a transformação $t(\cdot)$.

Tem-se, agora, outra questão: a existência de dados duplicados no novo mapa. Isso pode acontecer se R_1 e R_2 tiverem mapeado áreas coincidentes, o que é provavelmente verdade ao menos nos arredores do ponto em que se encontraram. O resultado é que o vetor de estado alinhado \mathbf{X}^a poderá conter *features* repetidas, prejudicando a acurácia do SLAM. Dessa forma, é preciso remover as duplicatas do mapa e propagar tais alterações para o vetor de estado e a matriz de covariância. Primeiramente é necessário determinar quais *features* encontram-se repetidas em \mathbf{X}^a . De posse desta informação, as duplicatas que não pertencem ao mapa original devem ser descartadas – no caso de R_1 , são removidos os dados oriundos de R_2 . Isso ocorre porque cada robô possui maior confiança nos dados gerados por si próprio em detrimento dos demais [9]. Em seguida, as linhas e colunas correspondentes da matriz de covariância do estado aumentado P^a também são removidas.

O procedimento de fusão dos mapas é realizado por ambos os agentes envolvidos (R_1 e R_2). Uma vez concluída esta etapa eles voltam a navegar normalmente, executando o *Single-robot EKF-SLAM* de forma independente sobre seus mapas aumentados. Até o final de suas navegações eles podem ou não encontrar-se um com o outro novamente, ou com outros agentes quaisquer.

4 IMPLEMENTAÇÃO, TESTES E RESULTADOS

Após a elaboração da proposta do sistema, introduzido no Capítulo 3, foi feita uma análise das possibilidades de trabalho sobre o tema que se adequariam ao tempo disponível para a pesquisa. Optou-se por abordar as tarefas da dinâmica do sistema SLAM colaborativo para cadeiras de rodas inteligentes proposto, com foco no tratamento dos dados e incertezas vinculados ao uso de um mapa prévio no início da navegação, sem o compromisso com a representação M-Space das *features* lineares na implementação.

A simulação do sistema foi construída no Matlab, a partir de um algoritmo *Single-robot EKF-SLAM* para marcos pontuais num ambiente de duas dimensões [21]. Um agradecimento especial a Juan Dell'Oso por tão gentilmente ter cedido seu código, que serviu como ponto de partida para esta implementação. Os parâmetros do ambiente são armazenados num arquivo separado, permitindo sua alteração conforme for necessário. Tais parâmetros representam os pontos extremos das *features* lineares, que são computadas como os marcos realmente utilizados pelo EKF-SLAM. A ideia seguida nesta implementação foi a geração de dados suficientes para que o sistema proposto pudesse ser testado, o que se mostrou possível mesmo sem a utilização por hora da representação M-Space. O principal motivo para o uso dessa representação especial é amenizar o impacto de certas limitações de sensoriamento comuns na robótica. Por outro lado, a complexidade dos ambientes de testes não é tão grande, o que levou a crer que tais efeitos estariam presentes, mas provavelmente não seriam capazes de inviabilizar a validade dos resultados obtidos.

Sendo assim, o algoritmo SLAM implementado trabalha com o vetor de estado \mathbf{X} , de dimensões $3 + 2m$ (Eq. 4.1), e a matriz de covariância apropriada. No

vetor de estado, \mathbf{x}_r é a pose do robô, dada pela Eq. (3.2), e o mapa é constituído por m pontos $\mathbf{x}_i = [x_i \ y_i]^T$, $1 \leq i \leq m$.

$$\mathbf{X} = [\mathbf{x}_r^T \ \mathbf{x}_1^T \ \mathbf{x}_2^T \ \dots \ \mathbf{x}_m^T]^T \quad (4.1)$$

A partir desses dados o sistema computa informações complementares relativas à característica linear do ambiente, que ainda se deseja aproveitar. O vetor \mathbf{X}_f (Eq. 4.2), de tamanho $3 + 4n$, também contém a pose do robô e o mapa estimado em termos de *features* lineares, compostas pelos pares ordenados dos seus pontos extremos A e B : $\mathbf{x}_{f_i} = [x_A \ y_A \ x_B \ y_B]^T$ (conforme Eq. 3.1), $1 \leq i \leq n$, onde n é a quantidade de segmentos no mapa.

$$\mathbf{X}_f = [\mathbf{x}_r^T \ \mathbf{x}_{f_1}^T \ \mathbf{x}_{f_2}^T \ \dots \ \mathbf{x}_{f_n}^T]^T \quad (4.2)$$

A movimentação dos robôs entre os instantes de tempo t e $t + 1$ é dada pela Eq. (4.3), com hodometria $\mathbf{u}(t) = [\Delta_t \ \phi_t]^T$. O deslocamento é Δ_t , fixado num passo de 10 cm, e ϕ_t é o giro. Os erros de hodometria são introduzidos por um ruído aleatório, sendo $\sigma_\Delta = 0.5$ cm e $\sigma_\phi = 1^\circ$. Os robôs são dotados de um sensor de proximidade a laser, cujas medições são compostas por dois valores: a distância (*range*) e a direção (*bearing*) do marco observado com relação à pose do robô. Tais dados também são ruidosos, com $\sigma_r = 1$ cm e $\sigma_b = 1^\circ$, respectivamente. Os testes, detalhados adiante, utilizaram diferentes valores de campo de visão (FOV) e alcance para o sensor. Todos os valores numéricos acima tiveram como referência o modelo IntellWheels.

$$\mathbf{x}_r(t + 1) = \mathbf{x}_r(t) + \begin{pmatrix} \Delta_t * \cos(\theta + \phi_t) \\ \Delta_t * \sin(\theta + \phi_t) \\ \phi_t \end{pmatrix} \quad (4.3)$$

Cada agente percorre um caminho pré-definido, com pose inicial conhecida, cujos pontos são guardados em outro arquivo, que também pode ser alterado. As configurações do ambiente serão abordadas mais à frente, durante as considerações

sobre os testes realizados. Os trajetos são independentes entre diferentes robôs, que navegam enquanto executam o algoritmo *Single-robot EKF-SLAM* até que dois deles se encontrem. Esse momento marca o *rendez-vous*, quando ocorre a fusão dos mapas dos robôs envolvidos. Sejam eles chamados de R_1 e R_2 . Os desvios padrão dos erros das medidas de observação mútua possuem valores iguais para ambos os agentes, propositalmente escolhidos para serem maiores que os valores utilizados para os erros de *range* e *bearing* da detecção de marcos. O intuito foi representar na simulação o fato de que os dados de observação mútua são ainda menos precisos que as leituras dos sensores de proximidade dos agentes [9]. Para o erro do ângulo ${}^i\phi_j$ sob o qual R_i observa R_j , com $i, j = 1, 2, i \neq j$, tem-se que

$$\sigma_{1\phi_2} = \sigma_{2\phi_1} = 4^\circ.$$

No caso da distância η entre R_1 e R_2 , obtém-se $\sigma_\eta = 2$ cm conforme a Eq. (3.6) a partir das variâncias dos erros das posições relativas ${}^1\eta_2$ e ${}^2\eta_1$:

$$\sigma_{1\eta_2}^2 = \sigma_{2\eta_1}^2 = 8 \text{ cm}^2.$$

Além disso, como seus valores são iguais, a distância η é dada pela média aritmética de ${}^1\eta_2$ e ${}^2\eta_1$ (Eq. 3.3). Cabe ressaltar que a observação mútua é computada com relação às poses reais (e não estimadas) de R_1 e R_2 , assim como é feito com as observações do ambiente.

A fusão dos mapas de R_1 e R_2 possui três etapas, conforme indicado na Seção 3.2.1: (i) alinhamento dos mapas; (ii) atualização das incertezas e (iii) remoção de dados duplicados. No trabalho de Benedettelli, Garulli, Giannitrapani [9], como as matrizes de covariância são descritas no M-Space, este trecho está fortemente atrelado a tal representação das *features* lineares do ambiente. Apesar disso, os dados necessários para realizar a fusão não dependem de dados no M-Space e podem ser obtidos dos mapas por pontos, o que pode ser visto no material elaborado por Zhou, Roumeliotis [54].

O alinhamento é realizado sobre os mapas de pontos em \mathbf{X}_1 e \mathbf{X}_2 , e nada mais é do que a determinação da transformação entre os *frames* de R_1 e R_2 . Em outras palavras, com $i, j = 1, 2, i \neq j$, do ponto de vista do robô R_i , é preciso expressar o mapa de R_j contido em ${}^{G_j}\mathbf{X}_j$ no espaço das *features* $\langle G_i \rangle$ através de

$${}^{G_i}\mathbf{X}_j = \mathbf{t}({}^{G_i}\mathbf{X}_i, {}^{G_j}\mathbf{X}_j, \bar{\mathbf{z}}). \quad (4.4)$$

A roto-translação \mathbf{t} é definida pelas equações a seguir, começando pela determinação da posição do robô R_j no *frame* do robô R_i :

$${}^{R_i}\mathbf{p}_{R_j} = \eta * \begin{bmatrix} \cos({}^i\phi_j) \\ \sin({}^i\phi_j) \end{bmatrix}. \quad (4.5)$$

Em seguida, obtêm-se a posição (Eq. 4.7) e a orientação (Eq. 4.8) do robô R_j no espaço das *features* $\langle G_i \rangle$. \mathbf{C} é a matriz de rotação responsável por aplicar as devidas transformações ao argumento recebido, denotado por ψ na Eq. 4.6.

$$\mathbf{C}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix} \quad (4.6)$$

$${}^{G_i}\mathbf{p}_{R_j} = {}^{G_i}\mathbf{p}_{R_i} + \mathbf{C}(\theta_i) * {}^{R_i}\mathbf{p}_{R_j} \quad (4.7)$$

$${}^{G_i}\theta_{R_j} = \theta_i + \pi + {}^i\phi_j - {}^j\phi_i \quad (4.8)$$

A transformação do *frame* de coordenadas globais $\langle G_j \rangle$ para $\langle G_i \rangle$ é feita pela expressão (4.10), sendo o ângulo ${}^{G_i}\theta_{G_j}$ dado por (4.9).

$${}^{G_i}\theta_{G_j} = {}^{G_i}\theta_{R_j} - \theta_j \quad (4.9)$$

$${}^{G_i}\mathbf{p}_{G_j} = {}^{G_i}\mathbf{p}_{R_j} - \mathbf{C}({}^{G_i}\theta_{G_j}) * {}^{R_j}\mathbf{p}_{R_j} \quad (4.10)$$

Por fim, para se obter ${}^{G_i}\mathbf{X}_j$, é necessário aplicar a transformação (4.11) a cada ponto \mathbf{x}_k , $1 \leq k \leq m_j$, onde m_j é a quantidade de pontos em \mathbf{X}_j .

$${}^{G_i}\mathbf{x}_k = {}^{G_i}\mathbf{p}_{G_j} + \mathbf{C}({}^{G_i}\theta_{G_j}) * {}^{G_j}\mathbf{x}_k \quad (4.11)$$

A esta altura, R_i já possui o estado aumentado \mathbf{X}_i^a , dado por

$$\mathbf{X}_i^a = \begin{bmatrix} \mathbf{X}_i \\ G_i \mathbf{X}_j \end{bmatrix} = \begin{bmatrix} \mathbf{X}_i \\ G_i \mathbf{p}_{R_j} \\ G_i \theta_{R_j} \\ G_i \mathbf{x}_1 \\ \vdots \\ G_i \mathbf{x}_{m_j} \end{bmatrix}. \quad (4.12)$$

Na segunda etapa (atualização das incertezas) é necessário computar o erro de estimativa do estado aumentado \mathbf{X}_i^a , o que é feito a partir dos erros de estimativa dos componentes de $G_i \mathbf{X}_j$. Como a roto-translação \mathbf{t} não é linear, precisa ser linearizada nos valores estimados de seus parâmetros: \mathbf{X}_i , $G_j \mathbf{X}_j$ e $\bar{\mathbf{z}}$. Portanto, a atualização das incertezas envolve o cálculo dos Jacobianos dos componentes de $G_i \mathbf{X}_j$ a fim de determinar seu erro de estimativa. Obtidos tais valores, pode-se determinar a matriz de covariância do estado aumentado (Eq. 4.13). Caso o leitor tenha interesse, as equações necessárias para a determinação dos Jacobianos podem ser encontradas em Zhou, Roumeliotis [54].

$$P_i^a = E[\tilde{\mathbf{X}}_i^a (\tilde{\mathbf{X}}_i^a)^T] = \begin{bmatrix} P_{i,i} & P_{i,j}^a \\ P_{j,i}^a & P_{j,j}^a \end{bmatrix} \quad (4.13)$$

A fusão dos mapas é encerrada pela remoção de dados duplicados. Quando há correspondência, são mantidos os dados originalmente gerados por R_i , pois sua confiança nas informações oriundas de R_j é menor [9]. Para cada ponto removido de \mathbf{X}_i^a , as linhas e colunas correspondentes de P_i^a também são removidas. Por fim, são removidas as informações sobre a pose de R_j e o vetor de *features* lineares \mathbf{X}_{f_i} de R_i é atualizado. Nas referências utilizadas, a associação de dados se baseia na **distância de Mahalanobis** [20] [8] [54] e no grau de sobreposição entre cada par de *features* avaliado. Na implementação realizada, contudo, os marcos são identificáveis, permitindo que R_i saiba quais são as duplicatas em seu mapa aumentado.

Após a fusão, R_1 e R_2 prosseguem navegando independentemente, executando o *Single-robot EKF-SLAM*. Podem ocorrer novos encontros entre ambos, ou entre qualquer um deles e outros agentes que também estejam presentes. Ao sair do ambiente, cada um realiza o *check-out*, entregando à entidade de supervisão seu mapa final.

Uma vez havendo um mapa armazenado pela supervisão, este é fornecido aos novos robôs que entrem no ambiente. Ou seja, em vez de começar a navegação sem mapa algum, o robô R^* subsequente recebe como informação prévia a última versão disponível do mapa. Para tanto, são armazenados o vetor de estado do último mapa, sua matriz de covariância e o vetor de *features* lineares correspondente, que são utilizados por R^* . Localmente, no vetor de estado e no vetor de *features* a pose do robô é alterada para os valores da pose inicial de R^* ; os demais dados não são alterados. Ou seja,

$$\mathbf{X}^* = \begin{bmatrix} \mathbf{x}_{r^*} \\ \mathbf{x}'_1 \\ \mathbf{x}'_2 \\ \vdots \\ \mathbf{x}'_{m'} \end{bmatrix} \quad \mathbf{X}_f^* = \begin{bmatrix} \mathbf{x}_{r^*} \\ \mathbf{x}'_{f_1} \\ \mathbf{x}'_{f_2} \\ \vdots \\ \mathbf{x}'_{f_{n'}} \end{bmatrix} \quad P^* = P',$$

onde \mathbf{x}'_k , e \mathbf{x}'_{f_k} são, respectivamente, os pontos e as *features* lineares do último mapa armazenado pelo sistema, e P' é a matriz de covariância correspondente.

4.1 Testes e Análise dos Resultados

Foram elaborados dois ambientes de testes, ilustrados pelas Figs. 4.1 e 4.2, ambos com 5x5 m de dimensão ¹. Todas as extremidades de segmentos de linhas, incluindo quinas, foram computadas como marcos pontuais. As *features* lineares foram caracterizadas como todo segmento mínimo determinado por dois pontos do ambiente, e que fossem observáveis pelos robôs. Isto é, segmentos oclusos por móveis etc. não foram computados como *features*, a exemplo do que ocorre na Fig. 4.2. O primeiro ambiente, chamado de A1, possui uma configuração pouco complexa, com apenas 22 *features* lineares. O objetivo primordial dos testes nesse ambiente foi avaliar a corretude e o comportamento geral do sistema. Já o outro ambiente, A2, é mais complexo, com 49 *features*. Aqui o intuito foi realizar testes com um volume maior de dados, a fim de analisar melhor o desempenho do mapeamento de um robô que recebe um mapa prévio no início de sua navegação, de acordo com a estratégia adotada.

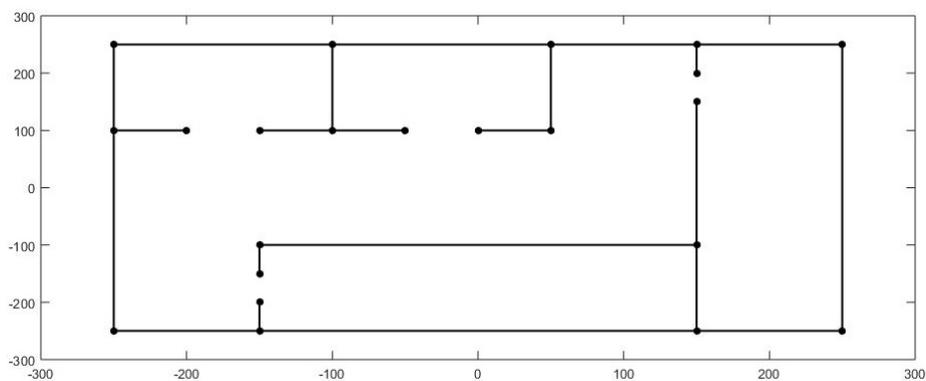


Figura 4.1: A1: ambiente virtual simples.

¹Todas as figuras utilizadas na Seção 4.1 e subseções foram geradas no Matlab com configurações padrão para gráficos.

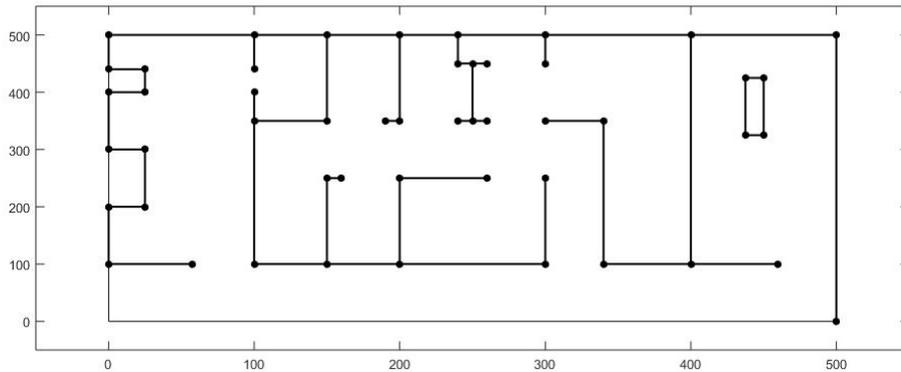


Figura 4.2: A2: ambiente virtual mais complexo.

No ambiente A1 as simulações foram compostas por dois cenários de teste diferentes (Seção 4.1.1), enquanto o ambiente A2 contou com três cenários distintos (Seção 4.1.2). Todos eles possuíam dois robôs, R_1 e R_2 , que começavam suas navegações de forma independente e se encontravam a determinada altura, realizando os procedimentos de fusão de mapas. Em seguida se observou ou a continuação da navegação de R_1 e R_2 com seus mapas aumentados M_1 e M_2 , ou o comportamento de um terceiro robô R_3 , que entrava em cena ora com M_1 , ora com M_2 como mapa prévio, no qual baseava sua navegação. Em outras palavras, o segundo caso visou à avaliação das dinâmicas de *check-out* e do uso de um mapa prévio por um novo robô que adentrasse o ambiente. As poses iniciais de cada robô eram conhecidas, mas não necessariamente iguais. Além disso, foram definidas duas configurações de sensoriamento distintas, comuns no sensoriamento a laser. Na configuração tipo 1 o FOV é de 180° e o alcance é de 100 cm; na configuração tipo 2, o FOV é de 270° e o alcance é de 80 cm (em especial, baseada no modelo IntellWheels). Assim, cada cenário de teste foi executado 200 vezes: 100 rodadas com a configuração tipo 1, e outras 100 com a configuração tipo 2.

No decorrer desta e das seções seguintes, o termo **mapa final** denota o mapa obtido ao final de cada cenário teste, para cada uma de suas 100 rodadas com cada um dos dois tipos de configuração de sensoriamento. Para cada mapa estimado (M_1 e M_2) foram avaliados quatro parâmetros. (i) O **total de *features* detectadas por completo** (aquelas que tiveram ambos os pontos extremos detectados, e não apenas um deles). (ii) O **erro médio do mapa estimado** e (iii) o **desvio padrão do erro médio do mapa estimado**. O erro médio do mapa estimado nada mais é do que o valor médio da distância entre os pontos extremos de cada *feature* real e suas estimativas no mapa, somente com relação às *features* já detectadas. O último parâmetro foi (iv) a **incerteza média do mapa estimado**, composta pela média dos valores absolutos das incertezas das estimativas, contidas na diagonal principal da matriz de covariância. As métricas foram determinadas com relação às *features* completas, e não com relação ao conjunto de pontos detectados, que pode incluir *features* não inicializadas. Foi utilizado um intervalo de confiança de 95%. As coletas foram feitas em três instantes: (i) antes do *rendez-vous*; (ii) depois do *rendez-vous*; e (iii) ao final do cenário. Também foram armazenados, para os mapas finais, os valores mínimos e máximos de (i) quantidade de *features* lineares detectadas; (ii) erro médio na posição das *features* no mapa; e (iii) incerteza média sobre a posição das *features* no mapa – estes dois últimos computados conforme os parâmetros de avaliação (ii) e (iv) descritos anteriormente.

4.1.1 Ambiente A1

O primeiro cenário de testes no ambiente A1 teve apenas dois robôs, R_1 e R_2 , cujos caminhos anteriores ao *rendez-vous* são ilustrados nas Figs. 4.3 e 4.4. O losango marca o ponto inicial da trajetória e o quadrado, seu término. Após a fusão dos mapas, ambos continuaram a navegação percorrendo o mesmo caminho (Fig. 4.5), diferente das trajetórias iniciais, cada um com seu respectivo mapa aumentado.

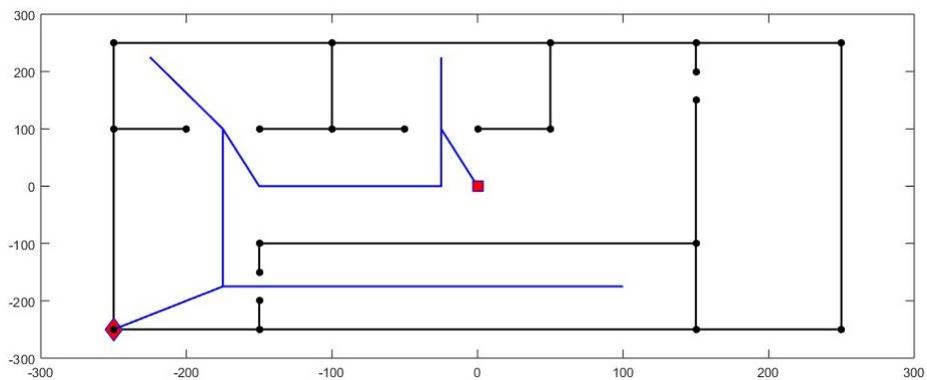


Figura 4.3: Caminho de R_1 antes do *rendez-vous*; primeiro cenário de A1.

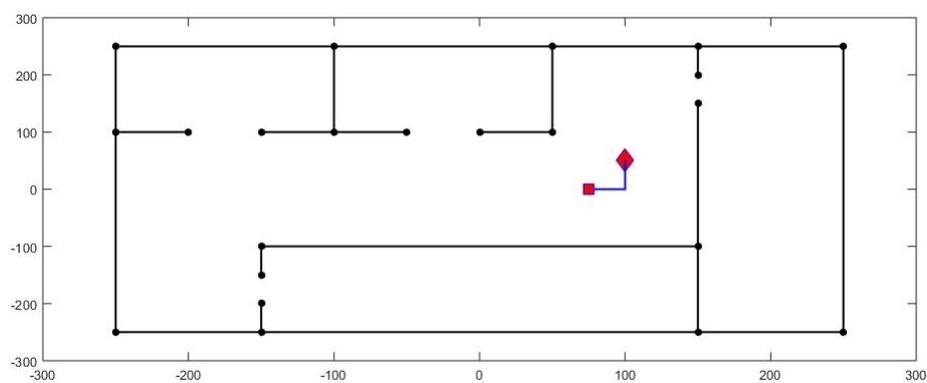


Figura 4.4: Caminho de R_2 antes do *rendez-vous*; primeiro cenário de A1.

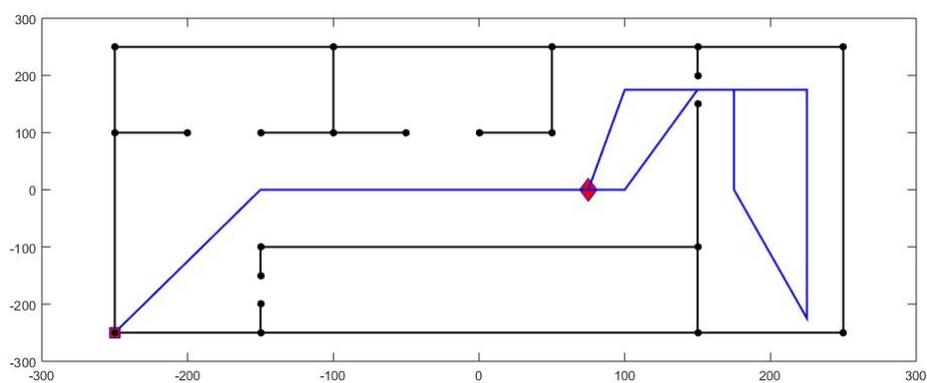


Figura 4.5: Caminho percorrido por R_1 e R_2 após o *rendez-vous*; primeiro cenário de A1.

Antes da discussão do levantamento estatístico, para melhor compreensão da dinâmica do sistema e da caracterização deste teste preliminar, são mostradas a seguir imagens com as etapas deste cenário na configuração de sensoriamento tipo 1: $FOV = 180^\circ$ e alcance = 100 cm. A navegação inicial de R_1 é retratada na Fig. 4.6, e o mapa então obtido, na Fig. 4.7. O traço vermelho representa o caminho realmente seguido pelo robô, enquanto o traço verde mostra seu caminho estimado. O robô é representado pelo triângulo (azul para estimado, vermelho para real); sua região de sensoriamento, pelo tracejado vermelho; e a incerteza de sua posição, pela elipse. Os landmarks pontuais reais são marcados com cruzeiras vermelhas, e suas posições estimadas são as cruzeiras verdes.

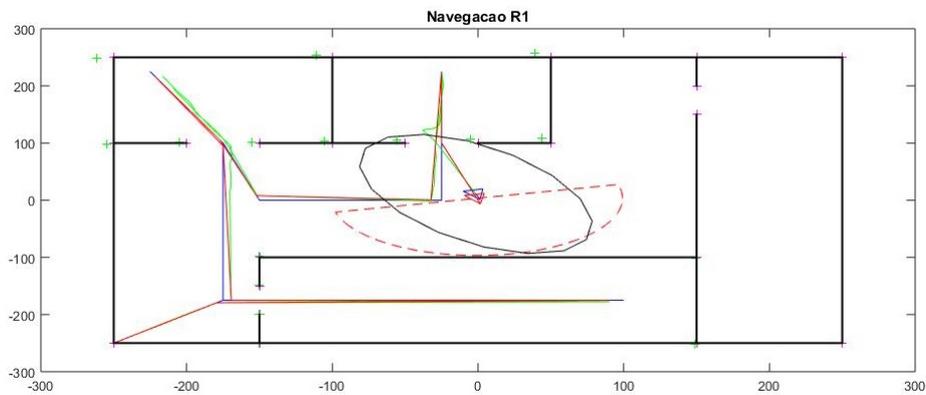


Figura 4.6: Navegação inicial de R_1 .

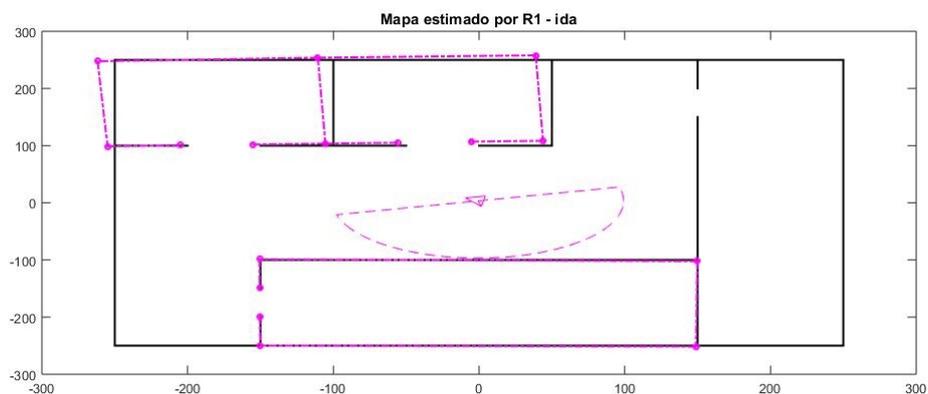


Figura 4.7: Mapa M_1 inicial estimado por R_1 .

Na Fig. 4.8 tem-se a navegação inicial de R_2 . Note-se que nenhum marco foi detectado, situação esta proposital. O objetivo foi avaliar como R_2 se comportaria após a fusão ao navegar com um mapa sem qualquer informação originalmente sua. Isso já seria um indicativo do que se poderia esperar da performance de um robô ao navegar com um mapa prévio conforme a representação utilizada.

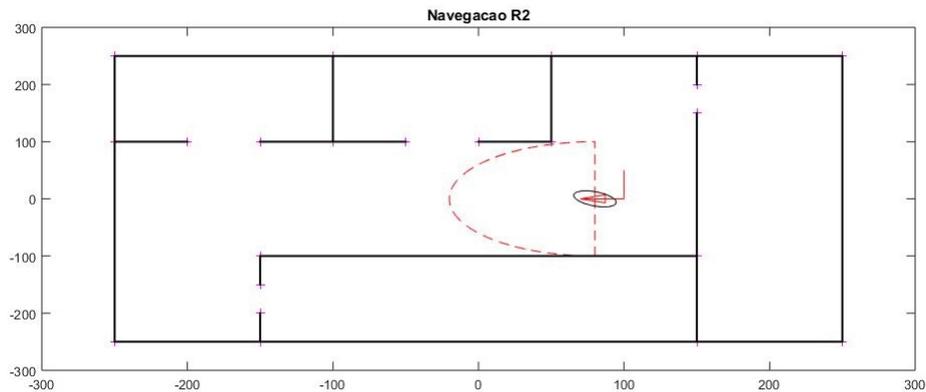


Figura 4.8: Navegação inicial de R_2 .

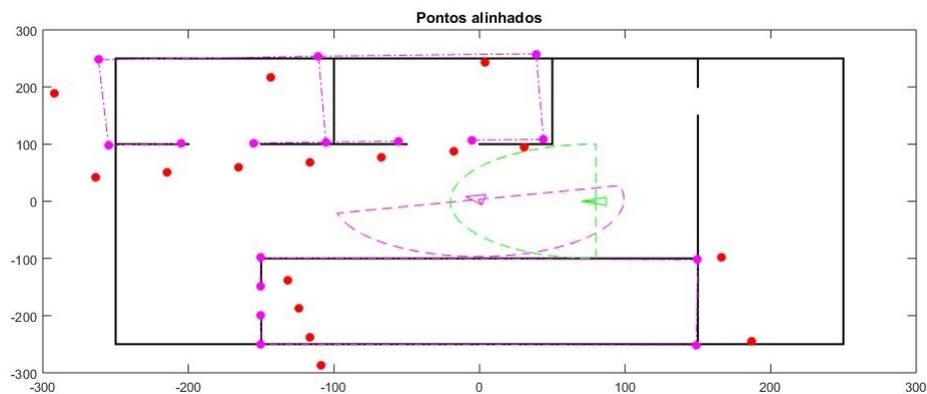


Figura 4.9: *Rendez-vous* e alinhamento dos mapas.

As posições dos robôs no momento do *rendez-vous* e o alinhamento dos mapas estão na Fig. 4.9, onde os pontos vermelhos são os pontos de M_1 no *frame* de R_2 .

Obviamente, M_1 não sofreu quaisquer alterações. Já R_2 “herdou” todos os pontos estimados por R_1 . Na Fig. 4.10 pode-se ver que há uma distância considerável entre o mapa real e M_2 , armazenado por R_2 . Neste exemplo, após prosseguir navegando, R_1 obteve um mapa final bastante apurado e completo (Fig. 4.11). A grande surpresa foi que R_2 , apesar de contar com um mapa com valores altos de erro e incerteza, conseguiu uma boa correção de sua estimativa (Fig. 4.12).

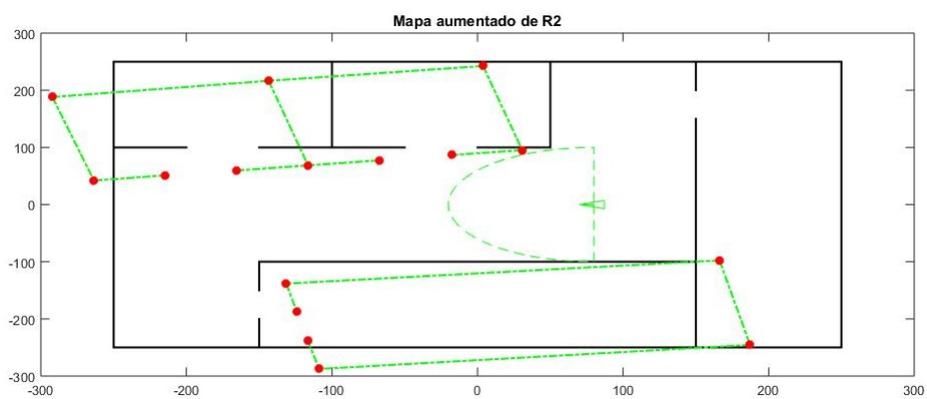


Figura 4.10: Mapa M_2 aumentado.

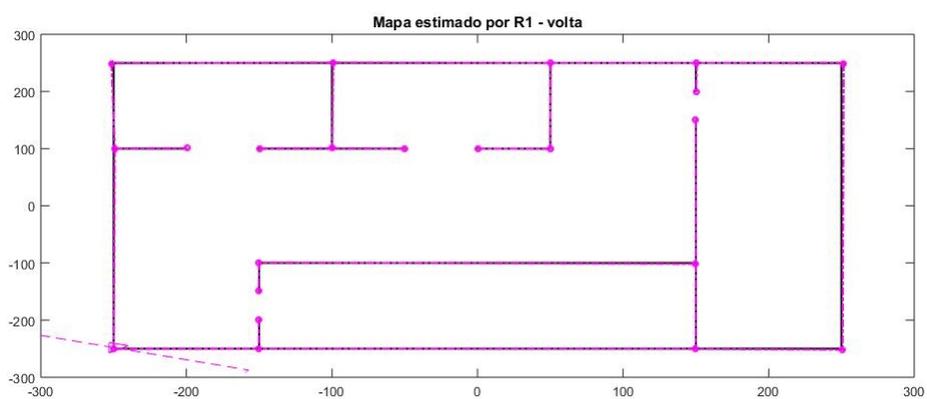


Figura 4.11: Mapa M_1 final estimado por R_1 .

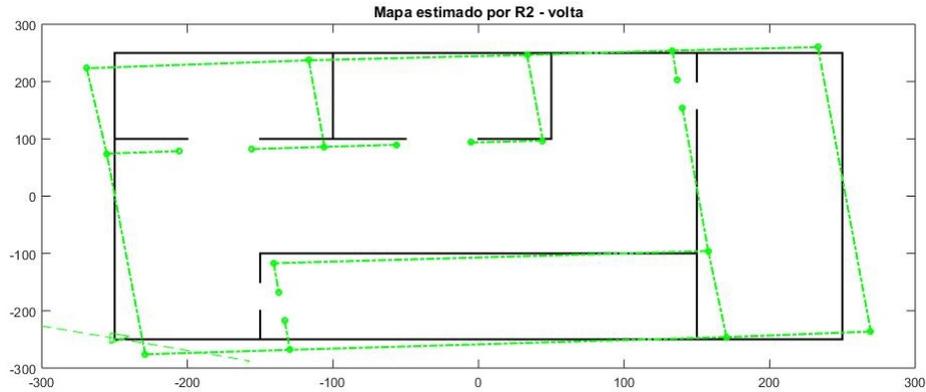


Figura 4.12: Mapa M_2 final estimado por R_2 .

Os valores médios das métricas obtidas nas 100 rodadas deste primeiro cenário de teste, em ambas as configurações de sensoriamento, são mostrados nas Tabelas 4.1 e 4.2. Na média, o tempo de execução de cada rodada na configuração de sensoriamento tipo 1 foi de 0.80 s, e de 0.69 s na configuração tipo 2. Como o estado do mapa M_1 é o mesmo antes e depois do *rendez-vous*, e M_2 está vazio antes do encontro, para fins práticos não foram consideradas as informações coletadas no primeiro instante. Relembrando, neste cenário, tanto R_1 quanto R_2 permaneceram no ambiente após o *rendez-vous*, navegando com seus mapas aumentados, sem se encontrarem novamente.

Observe que, na média, ocorreu uma melhora significativa na precisão dos mapas aumentados ao final do cenário. Uma atenção especial deve ser dispensada às informações sobre M_2 , cuja incerteza média começou bastante alta, por causa das condições em que se deu a fusão. Tais resultados, os primeiros obtidos nesta pesquisa, apontaram que a aplicação desta estratégia para SLAM com informação prévia seria promissora. Também se pode notar que a configuração de sensoriamento tipo 2 (FOV = 270° e alcance = 80 cm) foi afetada diretamente pela diminuição do alcance do sensor. Foi uma redução de 20% que ainda assim não pôde ser

compensada pelo aumento de 50% do FOV. Acredita-se que a queda de desempenho tenha sido fruto da combinação de dois fatores: (i) o aumento da área não coberta pelos robôs ao passarem pelos corredores largos, uma vez que suas trajetórias não eram próximas às paredes; e (ii) a influência da diminuição de observações sobre a correção do caminho estimado.

MAPA M_1	Pós fusão	Final
Total de <i>features</i>	14	22
Erro médio do mapa	6.9963	8.0065
Desvio padrão do erro médio	8.6647	10.0024
Incerteza média do mapa	94.1568	48.3684
MAPA M_2	Pós fusão	Final
Total de <i>features</i>	14	22
Erro médio do mapa	20.8895	11.2091
Desvio padrão do erro médio	23.7069	13.8664
Incerteza média do mapa	502.3334	68.9169

Tabela 4.1: Valores médios das métricas para M_1 e M_2 : ambiente A1, primeiro cenário, configuração de sensoriamento tipo 1.

MAPA M_1	Pós fusão	Final
Total de <i>features</i>	7.9300	14.8300
Erro médio do mapa	6.8857	6.5987
Desvio padrão do erro médio	8.6856	7.6556
Incerteza média do mapa	114.4648	113.9223
MAPA M_2	Pós fusão	Final
Total de <i>features</i>	7.9300	15.1000
Erro médio do mapa	19.5243	10.3122
Desvio padrão do erro médio	23.4196	12.3571
Incerteza média do mapa	456.6270	108.9506

Tabela 4.2: Valores médios das métricas para M_1 e M_2 : ambiente A1, primeiro cenário, configuração de sensoriamento tipo 2.

No segundo cenário de teste para este ambiente foram testadas duas situações análogas. Na primeira, supôs-se que R_1 realizou o *check-out* logo após o encontro com R_2 , entregando à supervisão seu mapa aumentado pós-fusão M_1 . Em seguida chegou um novo robô R_3 , que recebeu M_1 como mapa prévio. Já na segunda situação, supôs-se que quem saiu após a fusão foi R_2 . Neste caso, R_3 recebeu M_2 antes de iniciar sua navegação. Em ambas as situações, o **mapa final** diz respeito ao mapa obtido ao final da navegação de R_3 , que percorreu o mesmo caminho em ambos os casos, diferente daqueles percorridos por R_1 e R_2 . As trajetórias de R_1 , R_2 e R_3 estão nas Figuras 4.13, 4.14 e 4.15, respectivamente.

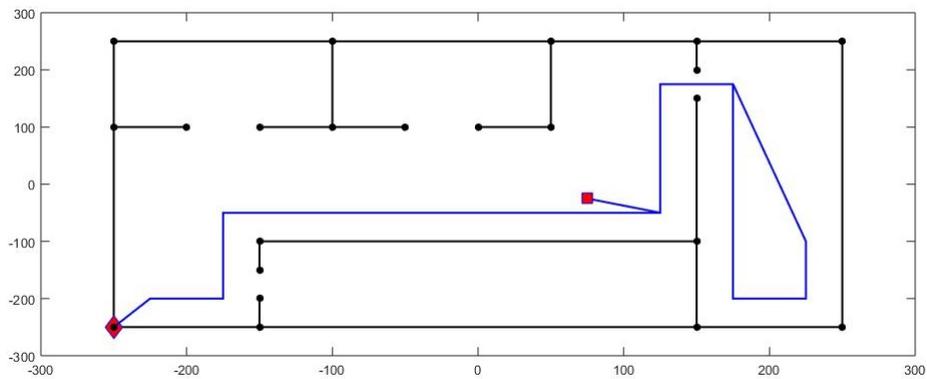


Figura 4.13: Caminho de R_1 (antes do *rendez-vous*); segundo cenário de A1.

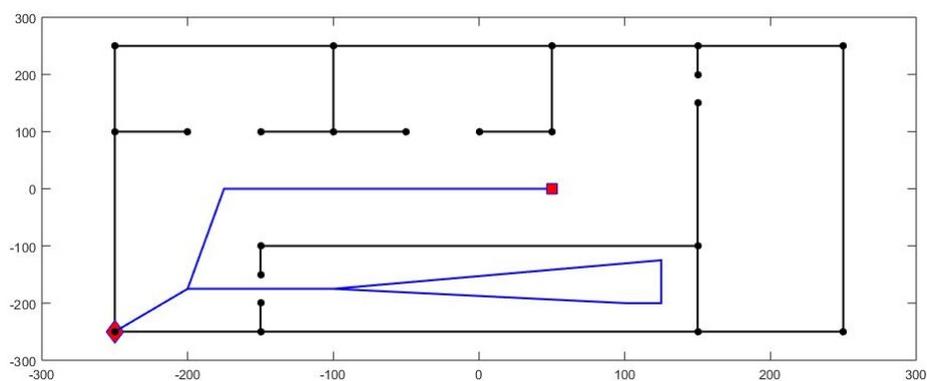


Figura 4.14: Caminho de R_2 (antes do *rendez-vous*); segundo cenário de A1.

MAPA M_1	Pré fusão	Pós fusão	Final
Total de <i>features</i>	8	9.0800	19
Erro médio do mapa	9.6345	11.4338	6.5106
Desvio padrão do erro médio	11.5969	13.1599	7.2124
Incerteza média do mapa	177.8800	283.1756	106.4028
MAPA M_2	Pré fusão	Pós fusão	Final
Total de <i>features</i>	6.0400	9.0800	19
Erro médio do mapa	2.4044	9.2731	3.5216
Desvio padrão do erro médio	2.7440	10.9560	3.7917
Incerteza média do mapa	27.2942	361.9284	50.8138

Tabela 4.3: Valores médios das métricas para M_1 e M_2 : ambiente A1, segundo cenário, configuração de sensoriamento tipo 1.

MAPA M_1	Pré fusão	Pós fusão	Final
Total de <i>features</i>	9	9	19
Erro médio do mapa	7.9864	7.9864	2.7633e+03
Desvio padrão do erro médio	9.9403	9.9403	2.7601e+04
Incerteza média do mapa	198.7158	198.7158	95.5052
MAPA M_2	Pré fusão	Pós fusão	Final
Total de <i>features</i>	5	9	19
Erro médio do mapa	2.4487	12.1798	27.8488
Desvio padrão do erro médio	3.0336	14.3531	178.4883
Incerteza média do mapa	62.3426	635.3493	113.0611

Tabela 4.4: Valores médios das métricas para M_1 e M_2 : ambiente A1, segundo cenário, configuração de sensoriamento tipo 2.

4.1.2 Ambiente A2

Para o ambiente mais complexo, A2, já foram desenhadas trajetórias mais próximas das paredes. Isso também acabou por ser uma consequência do aumento de detalhes na área, fazendo com que os corredores fossem mais estreitos. O seguimento de paredes é um comportamento comum em cadeiras de rodas inteligentes e, portanto, uma trajetória mais próxima delas na simulação é justificável.

Os três cenários de teste de A2 são semelhantes. Primeiramente, R_1 e R_2 navegaram por caminhos distintos até se encontrarem, quando realizaram a fusão de seus mapas. Então, novamente, duas situações foram verificadas. Na primeira ocorreu o *check-out* de R_1 logo após o *rendez-vous*, quando ele entregou à supervisão seu mapa aumentado pós-fusão M_1 . Em seguida entrou um novo robô, R_3 , que recebeu M_1 como informação prévia para sua navegação. A segunda situação é análoga, mas com relação à saída de R_2 .

No primeiro cenário, o caminho de R_3 foi o mesmo percorrido por R_1 , exibido na Fig. 4.16. O caminho de R_2 é mostrado na Fig. 4.17.

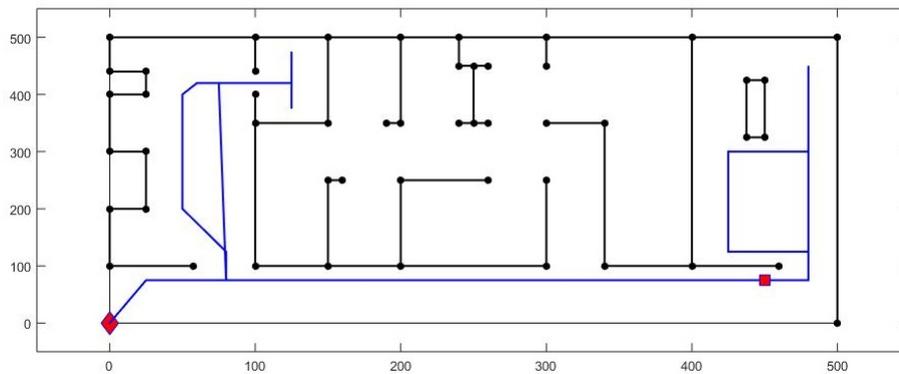


Figura 4.16: Caminho de R_1 .

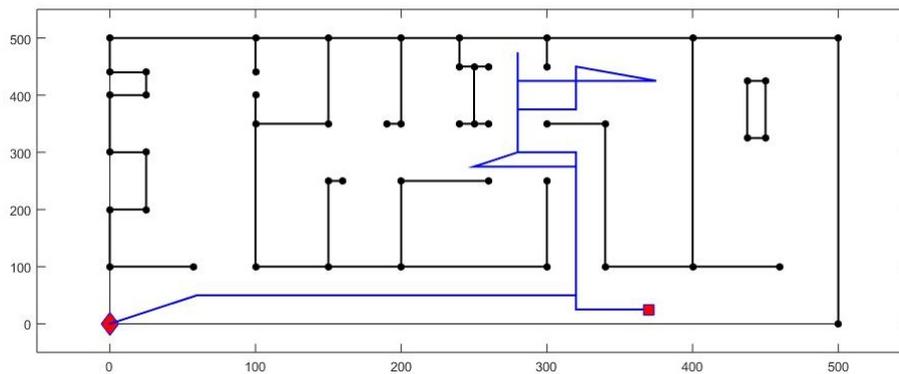


Figura 4.17: Caminho de R_2 .

Nas Tabelas 4.5 e 4.6 constam os valores médios das métricas para este primeiro cenário de A2, respectivamente nas configurações de sensoriamento tipos 1 (tempo médio de execução de 2.02 s) e 2 (tempo médio de execução de 1.98 s). Em ambas, enquanto o mapa M_1 teve uma incerteza média alta e erros médios ligeiramente maiores, M_2 obteve resultados melhores. Para comparação, são mostrados os mapas finais M_1 (Fig. 4.18) e M_2 (Fig. 4.19) com menor e maior erro médio na configuração de sensoriamento tipo 1 (FOV = 180° e alcance = 100 cm).

MAPA M_1	Pré fusão	Pós fusão	Final
Total de <i>features</i>	32.0200	47.0400	47.0800
Erro médio do mapa	6.3710	13.9089	13.7557
Desvio padrão do erro médio	7.4501	15.7807	15.6666
Incerteza média do mapa	74.9168	360.5471	234.6752
MAPA M_2	Pré fusão	Pós fusão	Final
Total de <i>features</i>	24	47.0400	47.0800
Erro médio do mapa	10.0272	21.4526	9.9937
Desvio padrão do erro médio	12.4511	24.6262	13.3243
Incerteza média do mapa	96.5749	551.8557	2.3889

Tabela 4.5: Valores médios das métricas para M_1 e M_2 : ambiente A2, primeiro cenário, configuração de sensoriamento tipo 1.

MAPA M_1	Pré fusão	Pós fusão	Final
Total de <i>features</i>	27.2800	42.9700	43.4800
Erro médio do mapa	7.3814	15.2251	14.5281
Desvio padrão do erro médio	8.6640	16.8614	16.2935
Incerteza média do mapa	106.9332	481.0910	279.7136
MAPA M_2	Pré fusão	Pós fusão	Final
Total de <i>features</i>	18.3600	42.9700	43.4200
Erro médio do mapa	9.7157	22.3046	11.4935
Desvio padrão do erro médio	11.8483	25.2479	14.3285
Incerteza média do mapa	117.6924	684.4578	9.4347

Tabela 4.6: Valores médios das métricas para M_1 e M_2 : ambiente A2, primeiro cenário, configuração de sensoriamento tipo 2.



Figura 4.18: Estados finais de M_1 com menor (acima) e maior (abaixo) erro médio.

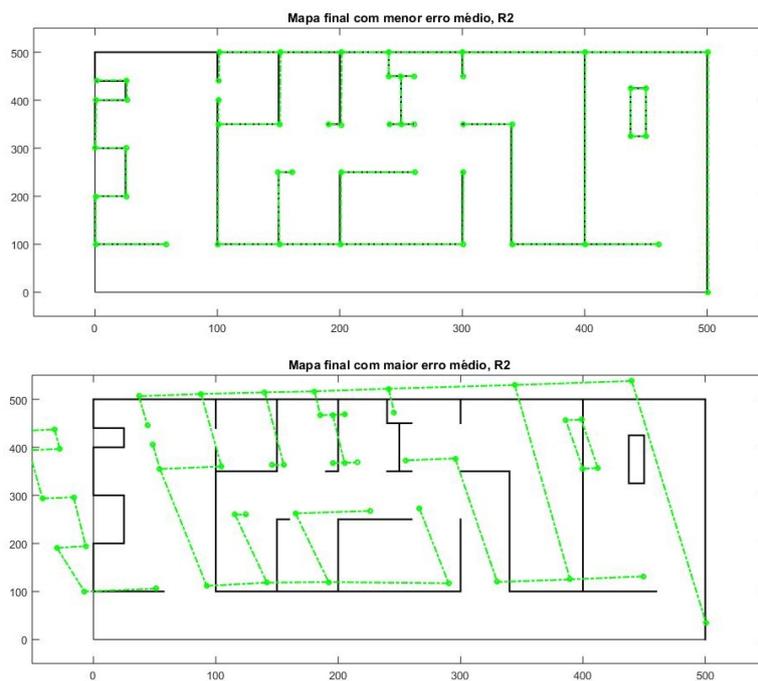


Figura 4.19: Estados finais de M_2 com menor (acima) e maior (abaixo) erro médio.

No segundo cenário de testes, o caminho percorrido por R_3 passou a ser o mesmo de R_2 (Fig. 4.17). Os dados levantados nas configurações de sensoriamento tipos 1 (tempo médio de execução de 1.64 s) e 2 (tempo médio de execução de 1.62 s) estão respectivamente nas Tabelas 4.7 e 4.8, e observa-se a mesma tendência de relação entre valores existente no cenário anterior, mas, desta vez, com os papéis trocados entre M_1 e M_2 .

MAPA M_1	Pré fusão	Pós fusão	Final
Total de <i>features</i>	32	47	47
Erro médio do mapa	6.8025	13.0568	8.7332
Desvio padrão do erro médio	8.1447	14.9422	11.5386
Incerteza média do mapa	74.6110	359.3223	2.1821
MAPA M_2	Pré fusão	Pós fusão	Final
Total de <i>features</i>	24	47	47
Erro médio do mapa	10.3910	19.3864	19.1171
Desvio padrão do erro médio	13.0614	22.2246	451.2734
Incerteza média do mapa	95.4203	548.9629	21.9559

Tabela 4.7: Valores médios das métricas para M_1 e M_2 : ambiente A2, segundo cenário, configuração de sensoriamento tipo 1.

MAPA M_1	Pré fusão	Pós fusão	Final
Total de <i>features</i>	27.2300	42.9400	43.8300
Erro médio do mapa	8.3422	15.5612	13.3719
Desvio padrão do erro médio	9.6082	17.6854	15.7567
Incerteza média do mapa	107.5995	489.0415	9.7040
MAPA M_2	Pré fusão	Pós fusão	Final
Total de <i>features</i>	18.5600	42.9400	43.9400
Erro médio do mapa	10.3073	22.9009	20.7516
Desvio padrão do erro médio	12.2959	26.1511	25.5060
Incerteza média do mapa	121.8729	685.7772	410.1260

Tabela 4.8: Valores médios das métricas para M_1 e M_2 : ambiente A2, segundo cenário, configuração de sensoriamento tipo 2.

Por fim, no terceiro cenário, R_3 realizou uma trajetória distinta daquelas feitas por R_1 e R_2 (Fig. 4.20). As métricas obtidas são exibidas nas Tabelas 4.9 e 4.10. Na média, o tempo de execução de cada rodada foi de 2.89 s e 2.75 s para cada tipo de configuração de sensoriamento respectivamente. Estes casos tiveram o melhor desempenho médio dentre as seis combinações simuladas, com incertezas e erros médios satisfatoriamente baixos.

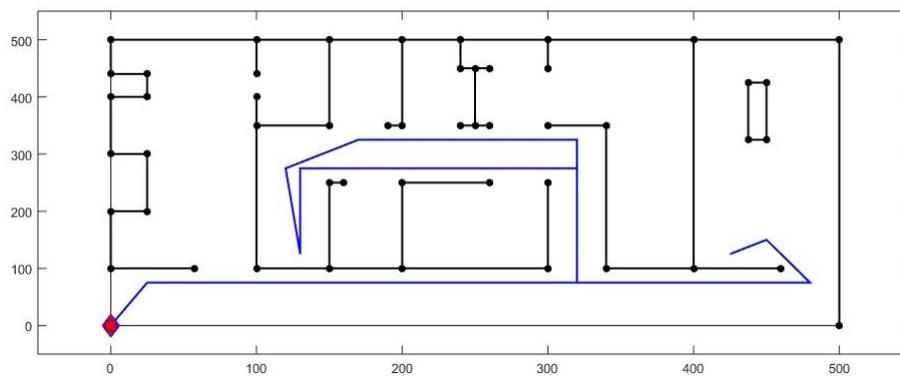


Figura 4.20: Caminho de R_3 no terceiro cenário de A2, com início e término no mesmo ponto.

MAPA M_1	Pré fusão	Pós fusão	Final
Total de <i>features</i>	32.0600	47.0600	47.0600
Erro médio do mapa	7.0606	14.6043	8.5324
Desvio padrão do erro médio	8.1286	16.2237	10.5306
Incerteza média do mapa	74.7051	362.9478	5.1182
MAPA M_2	Pré fusão	Pós fusão	Final
Total de <i>features</i>	24	47.0600	47.0600
Erro médio do mapa	10.8003	21.5320	8.7218
Desvio padrão do erro médio	12.8470	24.0869	10.6050
Incerteza média do mapa	95.2705	552.2097	3.8504

Tabela 4.9: Valores médios das métricas para M_1 e M_2 : ambiente A2, terceiro cenário, configuração de sensoriamento tipo 1.

MAPA M_1	Pré fusão	Pós fusão	Final
Total de <i>features</i>	27.3300	42.8000	43.2200
Erro médio do mapa	7.9841	16.3844	10.3698
Desvio padrão do erro médio	9.4984	18.5577	12.7619
Incerteza média do mapa	109.3383	488.1194	5.8427
MAPA M_2	Pré fusão	Pós fusão	Final
Total de <i>features</i>	18.1300	42.8000	43.2200
Erro médio do mapa	10.1492	24.4637	10.9321
Desvio padrão do erro médio	12.1808	27.9766	14.2837
Incerteza média do mapa	122.0444	693.2387	5.9594

Tabela 4.10: Valores médios das métricas para M_1 e M_2 : ambiente A2, terceiro cenário, configuração de sensoriamento tipo 2.

Um fato interessante é que, na média, quando R_3 fez o caminho de R_1 , o mapa final M_1 ficou menos preciso que o M_2 final. Acredita-se que tal comportamento se deva ao fato de que os valores referentes às informações originalmente geradas por R_1 não teriam como melhorar muito mais por causa das limitações dos sensores. Ou seja, essa parte do mapa podia não estar em seu estado ótimo, mas era o melhor que o sensoriamento do agente poderia obter. Nesse caso, como as reobservações aconteceram na região em que M_1 foi gerado antes de ser aumentado, houve uma melhora, mas esta foi menos significativa porque os dados beneficiados já se encontravam próximos dessa estabilidade. A melhora apresentada quando R_3 navegou de posse de M_2 já foi mais relevante, pois a reobservação influenciou as informações herdadas do mapa de R_1 durante a fusão (armazenadas com uma imprecisão maior por R_2). O equivalente ocorreu quando R_3 fez o caminho de R_2 . Já no terceiro cenário, onde R_3 percorreu um caminho diferente dos anteriores, as estimativas médias finais de M_1 e M_2 ficaram próximas em todos os casos.

As consequências da não utilização da representação M-Space puderam ser notadas, em especial, na grande inclinação de algumas *features* lineares quando a incerteza sobre seus pontos extremos era grande. Exemplos são os mapas finais

com erro médio alto nas Figuras 4.18 e 4.19. Apesar disso, o desempenho médio da estratégia avaliada foi satisfatório, contando inclusive com casos bastante apurados, conforme ilustrado em algumas figuras.

5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Inicialmente, pretendia-se tratar nesta dissertação da inclusão de um módulo *Single-robot SLAM* ao modelo de cadeira de rodas inteligentes da Plataforma IntellWheels. Viu-se, contudo, que essa contribuição não seria tão significativa quanto o esperado, dado o volume de estudos semelhantes realizados nos últimos anos. Por outro lado, não foram encontradas referências ao uso de SLAM Colaborativo com CRIs na literatura, o que pareceu ser uma boa alternativa para o direcionamento desta pesquisa.

Em última análise, cadeiras de rodas inteligentes são agentes robóticos móveis, capazes de perceber o ambiente em que se encontram e de possivelmente construir mapas com base em tais informações. Quando uma CRI é dotada da funcionalidade de mapeamento, ele é feito individualmente e seu produto não é compartilhado com outras entidades que possam beneficiar-se dele. Três desvantagens decorrem disso. A primeira delas diz respeito ao fato de que mapas gerados a partir da fusão dos dados de mais de um agente têm maiores chances de serem mais precisos. A segunda se refere aos sucessivos esforços de mapeamento eventualmente realizados por diferentes agentes sobre um mesmo ambiente. Enfim, a terceira é a possível defasagem dos mapas armazenados pelas cadeiras. Os dois primeiros fatores mostram que CRIs, assim como outros dispositivos robóticos móveis, podem tirar proveito de técnicas de SLAM Colaborativo. Mais que isso, a disponibilidade de mapas os mais recentes possíveis seria uma vantagem adicional.

Assim, o objetivo geral deste trabalho foi a elaboração de um sistema que, através da realização de SLAM Cooperativo entre cadeiras de rodas inteligentes,

fosse capaz de armazenar informações de mapeamentos parciais, que seriam utilizadas como dados iniciais em visitas subsequentes de outras CRIs, sendo aumentadas e refinadas gradativamente. Até que esta proposta esteja pronta para ser efetivamente utilizada pela Plataforma IntellWheels, são necessárias várias etapas. Por isso, o presente trabalho se restringiu às três etapas iniciais como objetivos específicos: (i) proposta da dinâmica do sistema; (ii) definição de uma estratégia C-SLAM compatível com a proposta e com o modelo IntellWheels; e (iii) avaliação de quais informações deveriam ser passadas às cadeiras na forma de um mapa prévio e da qualidade dos mapas obtidos por tal processo. Para este terceiro item especificamente, a hipótese era de que seria suficiente uma representação simples, composta pelo vetor de estado e pela matriz de covariância finais do último mapa armazenado pelo sistema e pela pose inicial do agente que estivesse entrando no ambiente. A justificativa para tanto era a padronização dos modelos sensorial e de odometria da Plataforma IntellWheels.

As características do sistema e a estratégia C-SLAM empregada foram descritas no Capítulo 3. Em resumo, o sistema é formado por uma entidade de supervisão e pelas CRIs, que se credenciam junto a tal entidade antes de começarem a navegar – o chamado *check-in*. É nesse momento que a cadeira recebe o mapa prévio, caso haja algum disponível. No método C-SLAM adotado (desenvolvido por Benedettelli, Garulli, Giannitrapani [9]), o mapeamento é realizado localmente por cada agente, e a colaboração ocorre através da fusão de mapas, quando dois deles se encontram (*rendez-vous*). O *rendez-vous* pode ocorrer diversas vezes entre diferentes CRIs, ou até mesmo não ocorrer. Por fim, toda cadeira, ao deixar o ambiente, realiza seu descredenciamento (*check-out*) junto à entidade de supervisão, a quem entrega seus dados de mapeamento.

O ambiente é representado por *features* lineares que, originalmente [9], são definidas através da representação M-Space [26] a fim de se amenizar o impacto de

certas limitações de sensoriamento comuns em robótica. Na simulação implementada neste trabalho, contudo, empregou-se a descrição dos segmentos de linha pelas coordenadas de seus pontos extremos. Tal decisão baseou-se no trabalho de Zhou, Roumeliotis [54], o qual demonstra que as informações necessárias para o pleno funcionamento da estratégia C-SLAM em si podem ser obtidas de mapas com marcos na forma de pontos. Assim, o algoritmo SLAM aqui desenvolvido trabalha com dois vetores de estado. Um deles contém as estimativas de todos os pontos detectados no ambiente, e o outro registra apenas as estimativas dos pontos extremos das *features* completamente identificadas pelo agente.

Detalhes sobre a implementação e os testes da hipótese levantada foram assuntos do Capítulo 4. O modelo do robô e seu sensoriamento são simplificações do modelo IntellWheels. Apesar disso, houve a preocupação de que os parâmetros de sensoriamento e os ruídos fossem mantidos em valores representativos do modelo. Os resultados dos testes realizados indicaram que a forma de representação do mapa prévio e a inclusão da pose inicial do robô recém-chegado no vetor de estado são suficientes para que ele realize SLAM. De fato, a qualidade dos mapas obtidos por meio desse processo superou as expectativas da autora, principalmente pelo fato de que foi necessário deixar de lado a representação M-Space das *features* lineares proposta pela principal referência deste trabalho. Era sabido que o uso da representação por pontos extremos poderia prejudicar a precisão dos mapas gerados; entretanto, para a finalidade dos testes conduzidos, os resultados foram satisfatórios.

A proposta introduzida por esta pesquisa abre espaço para outros trabalhos, na direção de que o sistema possa ser realmente implantado. A Plataforma IntellWheels possui plenas condições de ser utilizada para testes, tanto simulados quanto reais, onde o procedimento C-SLAM pode ser incorporado como um novo módulo lógico. Ao mesmo tempo, a preparação de um ambiente para dar suporte ao sistema e armazenar os mapas gerados não seria um processo muito intrusivo.

Uma sugestão é o uso de frameworks para *Cloud Robotics* (Robótica em Nuvem), levando o sistema para algo além da interação entre cadeiras de rodas inteligentes apenas. Além disso, a forma de representação do mapa prévio em especial pode ser útil em aplicações de robótica em geral, incluindo algumas realizadas pelo grupo de pesquisa do LabIC – UFRJ, como o Projeto NAROBISI, voltado para exploração e mapeamento autônomos em ambientes inóspitos.

Há, ainda, outras questões interessantes em aberto. A mais relevante delas diz respeito aos mapas deixados pelos agentes quando se retiram do ambiente (*check-out*). O ideal seria que a entidade de supervisão fosse capaz de realizar fusão sobre esse conjunto de mapas armazenados, integrando-os de forma inteligente para que sempre houvesse apenas uma versão do mapa guardada com ela. Entretanto, a estratégia utilizada é muito dependente da observação mútua entre robôs no *rendez-vous*, o que não seria possível de se realizar neste caso. O segundo item relevante é a incorporação da representação M-Space ao C-SLAM. Acredita-se que em ambientes mais complexos e ruidosos isso seria capaz de melhorar as estimativas dos mapas, além de ser uma representação compatível com o funcionamento de sensores a laser reais e do modelo sensorial da IWP em particular. Outro ponto está relacionado à diferenciação de mapas obtidos em andares diferentes de um mesmo prédio, que não foi abordada nesta pesquisa. Além destes, tem-se a inclusão de um indicador de confiabilidade entre os dados passados às cadeiras no *check-in*. Tal elemento informaria ao agente o quão precisa estaria aquela versão do mapa, permitindo que ele decida se deve ou não basear sua localização naqueles dados ao longo de sua navegação. Por fim, uma ideia interessante seria testar o comportamento do algoritmo mediante a mudança da solução para dados duplicados, abordados na última etapa da fusão de mapas. Originalmente, em caso de duplicata, o agente deve remover as informações herdadas de seu parceiro durante a fusão. Uma alternativa seria combinar os dados de ambos os agentes, numa estratégia semelhante àquela empregada para a observação mútua no *rendez-vous*.

REFERÊNCIAS

- [1] **AGEING in the Twenty-First Century: A Celebration and A Challenge**. New York: United Nations Population Fund, 2012. Disponível em: <<https://www.unfpa.org/sites/default/files/pub-pdf/Ageing%20report.pdf>>. Acesso em: 28 nov. 2014.
- [2] ALVES, D. S. et al. The Effect of Intelligent Escape on Distributed SER-Based Search. In: MURGANTE, B. (Ed.). **Computational Science and its Applications ICCSA 2012**. Berlin: Springer, 2012. p. 101–112. (Lecture Notes in Computer Science, v. 7333).
- [3] AUAT CHEEIN, F. A. et al. SLAM algorithm applied to robotics assistance for navigation in unknown environments. **Journal of NeuroEngineering and Rehabilitation**, [S.l.], v. 7, n. 1, 2010.
- [4] AULINAS, J. et al. The SLAM Problem: A Survey. In: ALSINET, T. Artificial Intelligence Research and Development: Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence. **Anais...** Amsterdam: IOS Press, 2008. p. 363–371.
- [5] AZEVEDO, J. et al. A Cooperative CiberMouse@RTSS08 Team. In: LOPES, L. S. et al. (Ed.). **Progress in Artificial Intelligence**. Berlin: Springer, 2009. p. 251–262. (Lecture Notes in Computer Science, v. 5816).
- [6] BARBOSA, M. J. T. **Development of an Intelligent Wheelchair 3D Simulator / Visualizer**. 2011. 90 f. Dissertação (Mestrado Integrado em Engenharia Informática e Computação) — Faculdade de Engenharia da Universidade do Porto, Porto, 2011.

- [7] BARKBY, S. et al. Incorporating prior maps with Bathymetric Distributed Particle SLAM for improved AUV navigation and mapping. In: OCEANS 2009, MTS/IEEE BILOXI, 2009, Mississippi. **Anais...** New York: IEEE, 2009. p. 1–7.
- [8] BENEDETTELLI, D. **Multi-Robot SLAM using M-Space feature representation**. 2009. 99 f. Dissertação (Mestrado) — Facoltà di Ingegneria, Università degli Studi di Siena, Siena, 2009.
- [9] BENEDETTELLI, D.; GARULLI, A.; GIANNITRAPANI, A. Cooperative SLAM Using M-Space Representation of Linear Features. **Robotics and Autonomous Systems**, [S.l.], v. 60, n. 10, p. 1267–1278, oct. 2012.
- [10] BISSOLI, A. L. C. et al. Geração e Suavização de Trajetórias Automáticas para uma Simulação Residencial de uma Cadeira de Rodas Comandada por Voz. In: SIMPÓSIO BRASILEIRO DE AUTOMACÃO INTELIGENTE, 2013, Fortaleza. **Anais...** [S.l.: s.n.], 2013.
- [11] BRAGA, R. A. M. **Plataforma de desenvolvimento de cadeiras de rodas inteligentes**. 2010. 246 f. Tese (Doutorado em Engenharia Informática) — Faculdade de Engenharia da Universidade do Porto, Porto, 2010.
- [12] BRAGA, R. A. M. et al. Concept and Design of the IntellWheels Platform for Developing Intelligent Wheelchairs. In: CETTO, J.; FERRIER, J.-L.; FILIPE, J. (Ed.). **Informatics in Control, Automation and Robotics**. Berlin: Springer, 2009. p. 191–203. (Lecture Notes in Electrical Engineering, v. 37).
- [13] BRAGA, R. A. M. et al. IntellWheels: Modular development platform for intelligent wheelchairs. **Journal of Rehabilitation Research and Development**, [S.l.], v. 48, n. 9, p. 1061–1076, 2011.
- [14] BRAGA, R. A. M. et al. Design de uma cadeira de rodas inteligente. In:

- CONGRESSO BRASILEIRO DE PESQUISA E DESENVOLVIMENTO EM DESIGN, 11, 2014, Gramado. **Anais...** [S.l.: s.n.], 2014. p. 3542–3553.
- [15] BRANDÃO, A. S. et al. Uma Estratégia de Desvio de Obstáculos Estáticos e Dinâmicos Aplicados a uma Cadeira de Rodas Robótica. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA, 18., 2010, Bonito. **Anais...** [S.l.: s.n.], 2010. p. 4381–4386.
- [16] Brasil. Ministério do Planejamento. **Síntese de indicadores sociais: uma análise das condições de vida da população brasileira: 2013.** Rio de Janeiro: IBGE, 2013. 266 p. (Estudo e Pesquisas. Informação demográfica e socioeconômica, 32). Disponível em: <<http://biblioteca.ibge.gov.br/visualizacao/livros/liv66777.pdf>>. Acesso em: 28 nov. 2014.
- [17] CAMARA NETO, V. F. da; CAMPOS, M. F. M. Localização e mapeamento simultâneos em robótica cooperativa com planejamento: uma revisão. In: SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE, 9., 2009, Brasília, DF. **Anais...** [S.l.: s.n.], 2009. p. 19–24.
- [18] Campos Filho, R. F. **Abordagem estocástica com fusão sensorial para mapeamento geográfico utilizando VANTs.** 2012. 75 f. Dissertação (Mestrado em Engenharia Mecânica) — Escola Politécnica, Universidade de São Paulo, São Paulo, 2012.
- [19] CHEN, S. et al. Multirobot FastSLAM Algorithm Based on Landmark Consistency Correction. **Mathematical Problems in Engineering**, [S.l.], v. 2014, p. 1–7, 2014.
- [20] CROWLEY, J. L. et al. Measurement and integration of 3-D structures by tracking edge lines. **International Journal of Computer Vision**, [S.l.], v. 8, n. 1, p. 29–52, 1992.

- [21] DELL'OSO, J. M. **Estudo de Métodos SLAM para Orientação de Robôs em Ambientes Internos**. 2014. 103 f. Dissertação (Mestrado em Informática) — Universidade Federal do Rio de Janeiro, Instituto de Matemática, Rio de Janeiro, 2014.
- [22] FARIA, B. M. **Patient Classification for Intelligent Wheelchair Adaptation**. 2013. 233 f. Tese (Doutorado em Engenharia Informática) — Universidade de Aveiro, Aveiro, 2013.
- [23] FARIA, B. M.; REIS, L. P.; LAU, N. Cerebral Palsy EEG Signals Classification: Facial Expressions and Thoughts for Driving an Intelligent Wheelchair. In: INTERNATIONAL CONFERENCE ON DATA MINING WORKSHOPS, 12., 2012, Brussels. **Anais...** New York: IEEE, 2012. p. 33–40.
- [24] FARIA, P. M. et al. Interface Framework to Drive an Intelligent Wheelchair Using Facial Expressions. In: INTERNATIONAL SYMPOSIUM ON INDUSTRIAL ELECTRONICS, 2007, Vigo. **Anais...** New York: IEEE, 2007. p. 1791–1796.
- [25] FERRIS, B.; FOX, D.; LAWRENCE, N. WiFi-SLAM Using Gaussian Process Latent Variable Models. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 20., 2007, Hyderabad. **Anais...** [S.l.: s.n.], 2007. p. 2480–2485.
- [26] FOLKESSON, J.; JENSFELT, P.; CHRISTENSEN, H. I. The M-Space Feature Representation for SLAM. **Transactions on Robotics**, [S.l.], v. 23, n. 5, p. 1024–1035, oct. 2007.
- [27] **FROM Exclusion to Equality**: Realizing the rights of persons with disabilities: Handbook for Parliamentarians on the Convention on the Rights of Persons with Disabilities and its Optional Protocol. New York: United Nations; Geneva: Office of the United Nations High Commissioner for Human Rights; Le Grand-Saconnex: Inter-Parliamentary

- Union, 2007. (Handbook for parliamentarians, 14). Disponível em: <<http://www.ohchr.org/Documents/Publications/training14en.pdf>>. Acesso em: 28 nov. 2014.
- [28] GONCALVES, V. C. F. **Descontaminação Distribuída de Grafos**. 2011. 72 f. Dissertação (Mestrado em Engenharia de Sistemas e Computação) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2011.
- [29] GRISETTI, G. et al. A Tutorial on Graph-Based SLAM. **Intelligent Transportation Systems Magazine**, [S.l.], v. 2, n. 4, p. 31–43, 2010.
- [30] HUANG, J. et al. Efficient, generalized indoor WiFi GraphSLAM. In: INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2011, Shanghai. **Anais...** New York: IEEE, 2011. p. 1038–1043.
- [31] KRISHNAN, R.; PUGAZHENTHI, S. Mobility assistive devices and self-transfer robotic systems for elderly, a review. **Intelligent Service Robotics**, [S.l.], v. 7, n. 1, p. 37–49, 2014.
- [32] LAU, N. et al. Ciber-Rato: Uma Competição Robótica num Ambiente Virtual. **Revista do DETUA**, [S.l.], v. 3, n. 7, p. 647–650, 2002.
- [33] LAZARO, M. T. et al. Multi-robot SLAM using condensed measurements. In: INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 2013, Tokyo. **Anais...** New York: IEEE, 2013. p. 1069–1076.
- [34] LI, R. et al. ROS Based Multi-sensor Navigation of Intelligent Wheelchair. In: INTERNATIONAL CONFERENCE ON EMERGING SECURITY TECHNOLOGIES, 4., 2013, Cambridge. **Anais...** New York: IEEE, 2013. p. 83–88.
- [35] LI, R. et al. Multi-layered map based navigation and interaction for an intelligent wheelchair. In: INTERNATIONAL CONFERENCE ON ROBOTICS AND BIOMIMETICS, 2013, Shenzhen. **Anais...** New York: IEEE, 2013. p. 115–120.

- [36] LIU, Y.; FAN, X.; ZHANG, H. A Fast Map Merging Algorithm in the Field of Multirobot SLAM. **The Scientific World Journal**, [S.l.], v. 2013, p. 1–8, 2013.
- [37] LÔBO, R. M. **Novas estratégias de planejamento no FIRN: um framework para navegação inteligente de robôs**. 2012. 174 f. Dissertação (Mestrado em Informática) — Universidade Federal do Rio de Janeiro, Instituto de Matemática, Rio de Janeiro, 2012.
- [38] MADARASZ, R. et al. The design of an autonomous vehicle for the disabled. **Journal of Robotics and Automation**, [S.l.], v. 2, n. 3, p. 117–126, 1986.
- [39] MOHANARAJAH, G. et al. Cloud-Based Collaborative 3D Mapping in Real-Time With Low-Cost Robots. **Transactions on Automation Science and Engineering**, [S.l.], v. 12, n. 2, p. 423–431, apr. 2015.
- [40] MORATUWAGE, D. et al. Extending Bayesian RFS SLAM to multi-vehicle SLAM. In: INTERNATIONAL CONFERENCE ON CONTROL AUTOMATION ROBOTICS VISION, 12., 2012, Guangzhou. **Anais...** New York: IEEE, 2012. p. 638–643.
- [41] OVIATT, S. Multimodal Interfaces. In: JACKO, J. A.; SEARS, A. (Ed.). **The Human-computer Interaction Handbook**. Mahwah: L. Erlbaum Associates, 2003. p. 286–304.
- [42] PACHECO, R. N.; COSTA, A. H. R. Navegação de robôs móveis utilizando o método de campos potenciais. In: SAKUDE, M. T. S.; A. C., C. C. de (Ed.). **Workshop de Computação**. São José dos Campos: Instituto Tecnológico de Aeronáutica, 2002. p. 125–130.
- [43] PETRY, M. **A Vision-based Approach Towards Robust Localization for Intelligent Wheelchairs**. 2013. 198 f. Tese (Doutorado em Engenharia Informática) — Faculdade de Engenharia da Universidade do Porto, Porto, 2013.

- [44] REIS, L. P. et al. IntellWheels MMI: A Flexible Interface for an Intelligent Wheelchair. In: BALTES, J. et al. (Ed.). **RoboCup 2009: Robot Soccer World Cup XIII**. Berlin: Springer, 2010. p. 296–307. (Lecture Notes in Computer Science, v. 5949).
- [45] SERUGENDO, G. D. M.; GLEIZES, M.-P.; KARAGEORGOS, A. Self-organization in Multi-agent Systems. **The Knowledge Engineering Review**, [S.l.], v. 20, n. 2, p. 165–189, 2005.
- [46] SILVEIRA, E. Mobilidade ampliada: Sistema permitirá a portadores de paralisias graves comandar cadeira de rodas com músculos da face. In: **Pesquisa FAPESP**. [S.l.: s.n.]. São Paulo, n. 208, jun. 2013. Engenharia Biomédica, p. 72.
- [47] SOLÀ, J. **Simultaneous localization and mapping with the extended Kalman filter**. Disponível em: <<http://www.iri.upc.edu/people/jsola/JoanSola/eng/course.html>>. Acesso: 28 nov. 2014.
- [48] TERZIDIS, K. **Algorithms for Visual Design Using the Processing Language**. Indiannapolis: Wiley, 2009.
- [49] THRUN, S.; BURGARD, W.; FOX, D. **Probabilistic Robotics**. Cambridge: MIT Press, 2006. (Intelligent Robotics and Autonomous Agents).
- [50] THRUN, S.; LEONARD, J. Simultaneous Localization and Mapping. In: SICILIANO, B.; KHATIB, O. (Ed.). **Springer Handbook of Robotics**. Berlin: Springer, 2008. p. 871–889.
- [51] UNIVERSIDADE DE AVEIRO. Departamento de Electrónica, Telecomunicações e Informática. **CiberRato 2013: Rules and Technical Specifications**. 2013, p. 1-15. Disponível em: <<https://sourceforge.net/projects/cpss/files/cpss/2.1.0/Ciber2013-rules.pdf>>. Acesso em: 28 nov. 2014.

- [52] WU, B.-F. et al. Accompanist recognition and tracking for intelligent wheelchairs. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN AND CYBERNETICS, 2014, San Diego. **Anais...** New York: IEEE, 2014. p. 2138–2143.
- [53] ZENG, Q. et al. A Collaborative Wheelchair System. **IEEE Transactions on Neural Systems and Rehabilitation Engineering**, [S.l.], v. 16, n. 2, p. 161–170, apr. 2008.
- [54] ZHOU, X. S.; ROUMELIOTIS, S. I. **Multi Robot SLAM Map Alignment with Rendezvous**. Minneapolis: Department of Computer Science and Engineering, University of Minnesota, 2005. 11 p. (Technical Report TR-2005-001). Disponível em: <www-users.cs.umn.edu/~zhou/paper/rendTechRep.pdf>. Acesso em: 25 jul. 2015.
- [55] ZOU, D.; TAN, P. CoSLAM: Collaborative Visual SLAM in Dynamic Environments. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, [S.l.], v. 35, n. 2, p. 354–366, feb. 2013.