

UNIVERSIDADE FEDERAL DE PELOTAS
Centro de Desenvolvimento Tecnológico
Programa de Pós-Graduação em Computação



Tese

**Em Busca de Navegação Socialmente Aceita e Confortável
com Foco em Cadeiras de Rodas Autônomas**

Kristofer Stift Kappel

Pelotas, 2024

Kristofer Stift Kappel

**Em Busca de Navegação Socialmente Aceita e Confortável
com Foco em Cadeiras de Rodas Autônomas**

Tese apresentada ao Programa de Pós-Graduação em Computação do Centro de Desenvolvimento Tecnológico da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Doutor em Ciência da Computação.

Orientador: Prof. Dr. Paulo Roberto Ferreira Jr.

Pelotas, 2024

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação da Publicação

K17e Kappel, Kristofer Stift

Em busca de navegação socialmente aceita e confortável com foco em cadeiras de rodas autônomas [recurso eletrônico] / Kristofer Stift Kappel ; Paulo Roberto Ferreira Junior, orientador. — Pelotas, 2024.
127 f. : il.

Tese (Doutorado) — Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2024.

1. Navegação social. 2. Cadeiras de rodas autônomas. 3. Mapas de custo. 4. Interação humano-robô. I. Ferreira Junior, Paulo Roberto, orient. II. Título.

CDD 005

Kristofer Stift Kappel

**Em Busca de Navegação Socialmente Aceita e Confortável
com Foco em Cadeiras de Rodas Autônomas**

Tese aprovada, como requisito parcial, para obtenção do grau de Doutor em Ciência da Computação, Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas.

Data da Defesa: 23 de Setembro de 2024

Banca Examinadora:

Prof. Dr. Paulo Roberto Ferreira Jr. (orientador)

Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Fagner de Assis Moura Pimentel

Doutor em Engenharia Elétrica pelo Centro Universitário FEI.

Prof. Dr. Felipe de Souza Marques

Doutor em Ciência da Computação pela UFRGS.

Prof. Dr. João Alberto Fabro

Doutor em Engenharia Elétrica e Informática Industrial pela UTFPR.

Dedico aos meus pais e à minha namorada pelo incansável esforço para me ajudar.

AGRADECIMENTOS

Agradeço primeiramente à Deus pela oportunidade de enfrentar mais uma etapa.

Em seguida agradeço aos meus pais, Liane e Claiton, pela condições que sempre me deram e pelos incentivos ao estudo, sem eles eu não chegaria neste ponto do caminho.

Agradeço à minha namorada, Gabriela, por não medir esforços em me ajudar sempre que precisei e por ser uma companheira incontestável.

Agradeço aos meus amigos e colegas que fizeram parte desta jornada, todos que de alguma forma ou de outra contribuíram para minha formação. Em especial à aqueles que acreditaram no projeto @home do PinguimBots comigo, Felipe Machado, Herick Dall'agnol e Vitória Fabrício, vocês tornaram os momentos de laboratório memoráveis.

Por fim agradeço aos meus mestres nessa jornada acadêmica. Em especial ao Prof. Dr. Paulo Ferreira Jr., que mesmo com todas as dificuldades que enfrentamos durante esse período sempre demonstrou seu apoio.

Não sei o que posso parecer para o mundo, mas para mim mesmo pareço ter sido apenas como um garoto brincando beira-mar e me desviando de vez em quando para encontrar uma pedra mais lisa ou uma concha mais bonita do que o normal, enquanto o grande oceano da verdade estava por descobrir diante de mim

— ISAAC NEWTON

RESUMO

KAPPEL, Kristofer Stift. **Em Busca de Navegação Socialmente Aceita e Confortável**

com Foco em Cadeiras de Rodas Autônomas. Orientador: Paulo Roberto Ferreira Jr.. 2024. 127 f. Tese (Doutorado em Ciência da Computação) – Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2024.

Robôs de serviço em ambientes públicos estão ficando cada vez mais comuns. Um dos próximos passos para a robótica é o desenvolvimento de robôs autônomos de navegação pessoal, tais como as Cadeiras de Rodas Motorizadas Autônomas (CRMA). Uma CRMA difere drasticamente de outros robôs com relação à navegação, pois é necessário levar em conta o fato de que o robô está ativamente carregando uma pessoa, que está exposta a diversas interações sociais, transformando o robô em um agente que precisa lidar com regras sociais durante a navegação. Dessa maneira, um robô autônomo tradicional não está sujeito, por exemplo, a desconforto durante a navegação, que é um fator que pode agravar a experiência do usuário em uma CRMA. O estudo aborda, principalmente, a dupla necessidade das CRMAs de navegar de forma eficiente, ao mesmo tempo em que aderem às normas sociais tanto de interação com pedestre quanto de interações com o seu usuário. A semântica existente nessas interações sociais é codificada em camadas de mapas de custo que representam o ambiente em um mapa de duas dimensões em forma de grade, cada célula nessa grade guarda um valor que é associado à semântica descrita. Além dos mapas de custo tradicionais para a navegação de robôs autônomos utilizamos uma camada referente ao espaço pessoal de terceiro, uma camada que calcula o conforto de se tráfegar em um ambiente com relação à distância das paredes e uma camada que calcula a visibilidade do local. Um problema recorrente é a avaliação isolada dessas camadas de custo, não averiguando quais as relações provenientes da combinação delas. O principal objetivo do trabalho é realizar a análise de como se comportam essas mesclas de mapas de custo de diferentes semânticas, tanto com relação ao usuário da CRMA como às demais pessoas. Por conseguinte, foram desenvolvidos dois cenários simulados diferentes, onde uma CRMA terá que navegar entre dois pontos. Um dos cenários reflete um ambiente de testes utilizado por um trabalho bem fundamentado da área, o outro é uma aproximação de um cenário real. Deste modo, os fatores avaliados a partir das simulações foram a taxa de conclusão da navegação, a distância total percorrida pelo robô, a taxa de invasão do espaço pessoal dos humanos, o conforto na navegação da CRMA e a visibilidade do usuário. Com relação às métricas, a combinação de todas as camadas utilizadas apresentou melhores resultados em todas as métricas, com exceção da distância total percorrida.

Dentre os comportamentos resultantes, por exemplo, foi possível observar que a inclusão da camada de conforto e de visibilidade teve impacto positivo no coeficiente de suavidade na navegação, trazendo assim, ainda mais comodidade para a navegação do usuário. Entretanto, nos resultados do coeficiente de visibilidade, houve uma redução após a inclusão das camadas de conforto e visibilidade, sem a camada de espaço pessoal, evidenciando a necessidade de analisar como a alteração nos ganhos de cada função de custo das camadas contribuiu no resultado final dos mapas de custo.

Palavras-chave: Navegação Social; Cadeiras de Rodas Autônomas; Mapas de Custo; Interação Humano-Robô.

ABSTRACT

KAPPEL, Kristofer Stiff. **Seeking for Socially Acceptable and Comfortable Navigation Focused on Autonomous Wheelchairs**. Advisor: Paulo Roberto Ferreira Jr.. 2024. 127 f. Thesis (Doctorate in Computer Science) – Technology Development Center, Federal University of Pelotas, Pelotas, 2024.

Service robots in public environments are becoming increasingly common. One of the next steps for robotics is the development of autonomous personal navigation robots, such as Autonomous Motorized Wheelchairs (AMW). An AMW differs drastically from other robots in terms of navigation, as it must take into account the fact that the robot is actively carrying a person who is exposed to various social interactions, turning the robot into an agent that needs to adhere to social rules during navigation. For example, a traditional autonomous robot is not subject to discomfort during navigation, which can significantly impact the user's experience in an AMW. This study primarily addresses the dual need for AMWs to navigate efficiently while adhering to social norms, both in interactions with pedestrians and with the wheelchair's user. The semantics embedded in these social interactions are encoded in layers of costmaps that represent the environment on a two-dimensional grid map, with each cell in this grid holding a value associated with the described semantics. In addition to the traditional cost maps for autonomous robot navigation, we use a layer for third-party personal space, a layer that calculates comfort when traveling in an environment concerning the distance from walls, and a layer that calculates visibility of the location. A recurring problem is the isolated evaluation of these cost layers, without considering the interactions resulting from their combination. The main objective of this work is to analyze how these combined cost maps with different semantics behave concerning both the AMW user and other people. Consequently, two different simulated scenarios were developed, where an AMW must navigate between two points. One scenario reflects a test environment used by a well-established study in the field, while the other approximates a real-world scenario. Thus, the factors evaluated from the simulations were the navigation completion rate, the total distance traveled by the robot, the rate of invasion of humans' personal space, the comfort of AMW navigation, and the visibility of the user. Regarding the metrics, the combination of all the layers used showed better results in all metrics, except for the total distance traveled. Among the resulting behaviors, for example, it was observed that the inclusion of the comfort and visibility layers had a positive impact on the smoothness coefficient of navigation, thus bringing even more comfort to the user's navigation. However, in the visibility coefficient results, there was a reduction after the inclusion of the comfort and visibility layers without the personal space layer, highlighting the need to analyze how changes in the gain of

each cost function of the layers contribute to the final result of the costmaps.

Keywords: Social Navigation; Autonomous Wheelchairs; Costmaps; Human-Robot Interaction.

LISTA DE FIGURAS

Figura 1	Exemplo de CRPM da marca Freedom.	28
Figura 2	Exemplo de CRM da marca Freedom.	29
Figura 3	CRMA proposta por Grewal; Matthews; Tea; George (2017)	31
Figura 4	Versão posterior da cadeira exposta na Figura 3, apresentada por Contractor; Grewal; Jayaprakash; Matthews; George (2018)	32
Figura 5	CRMA desenvolvida por Ng; Goh (2020)	33
Figura 6	Plataforma desenvolvida por Leaman; La; Nguyen (2016)	33
Figura 7	Diferentes Sensores	34
Figura 8	Câmeras de profundidade	36
Figura 9	Sip-and-puff produzido pela Origin Instruments	37
Figura 10	Execução do processo de construção do mapa através de SLAM	43
Figura 11	Partículas do AMCL	45
Figura 12	Especificação padrão dos valores dos mapas de custo	47
Figura 13	Execução do processo de atualização das camadas dos mapas de custo	49
Figura 14	Exemplo de navegação social gerada com a presença de proxemics, a CRMA tende a evitar invadir as zonas pessoal ou íntima das pessoas.	52
Figura 15	Adição de proxemics no mapa de custo	53
Figura 16	Representação gráfica do espaço de navegação focando no conforto do usuário de uma CRMA proposto por Morales; Kallakuri; Shinozawa; Miyashita; Hagita (2013).	55
Figura 17	Teste de navegação social da CRMA realizado por Morales; Kallakuri; Shinozawa; Miyashita; Hagita (2013).	56
Figura 18	HCoM gerado a partir do mapa do teste de Morales; Kallakuri; Shinozawa; Miyashita; Hagita (2013).	57
Figura 19	Rota gerada pelo A* para o menor caminho e o caminho mais confortável. (retirada de Morales; Kallakuri; Shinozawa; Miyashita; Hagita (2013))	58
Figura 20	Exemplo de visibilidade em esquinas demonstrado por Morales; Even; Kallakuri; Ikeda; Shinozawa; Kondo; Hagita (2014)	59
Figura 21	Exemplos de visibilidade no cálculo de razão de visibilidade (exemplo de Morales; Even; Kallakuri; Ikeda; Shinozawa; Kondo; Hagita (2014))	60
Figura 22	Visibilidade no mapa utilizado para os testes do trabalho de Morales; Even; Kallakuri; Ikeda; Shinozawa; Kondo; Hagita (2014))	61

Figura 23	Trajectoria dos participantes durante um dos testes de Morales; Even; Kallakuri; Ikeda; Shinozawa; Kondo; Hagita (2014)	62
Figura 24	Exemplo do algoritmo HCoPP proposto por Morales; Watanabe; Ferreri; Even; Ikeda; Shinozawa; Miyashita; Hagita (2015)	62
Figura 25	Fases da construção do mapa de conforto do HCoM	63
Figura 26	Visão superior do mapa gerado pelo HCoM, com as áreas de maior conforto para o usuário	64
Figura 27	Dois exemplos do índice de visibilidade	64
Figura 28	Exemplo do cálculo do índice de visibilidade para uma posição no mapa de custo	65
Figura 29	Mapa com o cálculo do índice de visibilidade	66
Figura 30	CRMA com sistema de pistas sociais proposto por Watanabe; Ikeda; Morales; Shinozawa; Miyashita; Hagita (2015)	69
Figura 31	Arquitetura proposta por Watanabe; Ikeda; Morales; Shinozawa; Miyashita; Hagita (2015) para geração das pistas sociais.	69
Figura 32	Teste realizado por Watanabe; Ikeda; Morales; Shinozawa; Miyashita; Hagita (2015) para validação do sistema de pistas sociais.	70
Figura 33	Resultados das trajetórias com pistas sociais geradas pelo projetor (Watanabe; Ikeda; Morales; Shinozawa; Miyashita; Hagita, 2015).	71
Figura 34	Resultados das trajetórias sem pistas sociais geradas pelo projetor (Watanabe; Ikeda; Morales; Shinozawa; Miyashita; Hagita, 2015). Pedestre anda próximo à parede para desviar da CRMA.	71
Figura 35	Resultados das trajetórias sem pistas sociais geradas pelo projetor (Watanabe; Ikeda; Morales; Shinozawa; Miyashita; Hagita, 2015). Pedestre desvia perto da CRMA.	72
Figura 36	Diferentes métodos para definição de espaços proxêmicos.	73
Figura 37	Modelo de CRMA desenvolvido por (Patil, 2018) e modificada pelo autor, visualizada no Gazebo	84
Figura 38	Modelo de CRMA desenvolvido por (Patil, 2018) e modificada pelo autor, visualizada no RViz	85
Figura 39	Modelo do mapa de corredores visualizado no Gazebo	86
Figura 40	Modelo do mapa de custo gerado pelo GMapping do mapa de corredores, visualizado no RViz	87
Figura 41	Representação do terceiro andar do Campus Anglo no Gazebo	88
Figura 42	Mapa de Custo gerado a partir da planta baixa do terceiro andar	88
Figura 43	Representação de metade do terceiro andar do Campus Anglo no Gazebo	89
Figura 44	Mapa de Custo gerado a partir de metade da planta baixa do terceiro andar	89
Figura 45	Atores posicionados ao longo do corredor principal do terceiro andar	90
Figura 46	Posições iniciais (círculo vermelho) e finais (triângulo vermelho) dos trajetos em ambos os mapas	92
Figura 47	Comparação entre diferentes conjuntos de camadas de mapas de custo; (a) Configuração padrão do <i>Navigation Stack</i> ; (b) Configuração padrão com a adição de um ator e camada de proxemics; (c) Inclusão de camadas de conforto e visibilidade, mas sem o ator e a camada de proxemics; (d) Todas as camadas propostas;	94

Figura 48	Navegação em diferentes configurações de mapas de custo no cenário de corredores.	95
Figura 49	Box plot da distância percorrida no cenário de corredores	96
Figura 50	Box plot da suavidade alcançada no cenário de corredores	97
Figura 51	Soma dos valores da camada de proxemics no cenário de corredores	98
Figura 52	Soma dos valores da camada de conforto no cenário de corredores	98
Figura 53	Soma dos valores da camada de visibilidade no cenário de corredores	99
Figura 54	Navegação em diferentes configurações de mapas de custo no cenário do terceiro andar.	100
Figura 55	Box plot da distância percorrida no cenário do terceiro andar	101
Figura 56	Box plot da suavidade no cenário do terceiro andar	102
Figura 57	Box plot da soma dos valores da camada de proxemics no cenário do terceiro andar	103
Figura 58	Box plot da soma dos valores da camada de conforto no cenário do terceiro andar	104
Figura 59	Box plot da soma dos valores da camada de visibilidade no cenário do terceiro andar	104
Figura 60	Mapas de custo no cenário de corredores.	124
Figura 61	Mapas de custo no cenário do terceiro andar.	125

LISTA DE TABELAS

Tabela 1	Classificação das distâncias definidas pelo espaço proxêmico. . . .	51
Tabela 2	Parâmetros utilizados no teste do HCoPP em ambiente real	66
Tabela 3	Descrição das Ferramentas de Desenvolvimento	83

LISTA DE ABREVIATURAS E SIGLAS

CRPM	Cadeira de Rodas de Propulsão Manual
CRM	Cadeira de Rodas Motorizada
CRMA	Cadeira de Rodas Motorizada Autônoma
IA	Inteligência Artificial
IHR	Interação Humano-Robô
ELA	Esclerose Lateral Amiotrófica
LME	Lesões na Medula Espinhal
OMS	Organização Mundial da Saúde
LiDAR	Light Detection and Ranging
EEG	Eletroencefalograma
EMG	Eletromiograma
ECG	Eletrocardiograma
AAL	Ambient Assisted Living
ICC	Interface Cérebro-Computador
PCL	Point Cloud Library
SLAM	Simultaneous Localization and Mapping
V-SLAM	Visual-Simultaneous Localization and Mapping
SSVEP	Steady State Visually Evoked Potential
IM	Imagens Motoras
ROS	Robot Operating System
GPU	Graphics Processing Unit
SBC	Single Board Computer
IMU	Inertial Measurement Unit
MCL	Monte Carlo Localization
KF	Kalman Filter
AMCL	Adaptative Monte Carlo Localization

EC	Espaço Cartesiano
EJ	Espaço de Juntas
GdL	Graus de Liberdade
EKF	Extende Kalman Filter
DWA	Dynamic Window Approach
TEB	Time Elastic Band
HCoM	Human-Comfort Factor Map
HCoPP	Human-Comfortable Path Planner
TC	Taxa de Conclusão
DP	Distância Percorrida
SC	Coeficiente de Suavidade
CS	Coeficiente Social
CC	Coeficiente de Conforto
CV	Coeficiente de Visibilidade

SUMÁRIO

1	INTRODUÇÃO	20
1.1	Hipótese	23
1.2	Objetivos	23
1.3	Estrutura da Tese	25
2	FUNDAMENTAÇÃO TEÓRICA	26
2.1	Da Cadeira de Rodas ao Robô Autônomo	26
2.1.1	Cadeiras de Rodas de Propulsão Manual	26
2.1.2	Cadeiras de Rodas Motorizadas	27
2.1.3	Cadeiras de Rodas Motorizadas Autônomas	28
2.2	Princípios da Navegação Autônoma	39
2.2.1	Mapeamento	39
2.2.2	Localização	40
2.2.3	Planejamento de Rotas	41
2.2.4	ROS e o Navigation Stack	41
2.3	Navegação Social	49
2.3.1	Navegação Social no Navigation Stack	52
3	TRABALHOS RELACIONADOS	54
3.1	Conforto do Humano	54
3.2	Visibilidade	58
3.3	Combinação de Fatores	61
3.4	Aprendizado de Normas Sociais	66
3.5	Relação Cadeira-Pedestre	67
3.6	Pistas Sociais	68
3.7	Utilização de Proxemics	72
3.8	Coordenação Social	73
3.9	Seleção dos Fatores Sociais	74
4	METODOLOGIA	76
4.1	Proposta	76
4.1.1	Camadas Sociais Adicionais	77
4.2	Ferramentas de Desenvolvimento	82
4.3	Modelo de CRMA Simulado	83
4.4	Cenários de Simulação	84
4.5	Variáveis de Interesse	86
4.6	Procedimentos dos Experimentos	90

5	RESULTADOS E DISCUSSÃO	93
5.1	Análise dos Resultados e Discussão	93
5.2	Primeiro Experimento - Cenário de Corredores	94
5.3	Segundo Experimento - Cenário do Anglo	99
5.4	Considerações Finais	105
6	CONCLUSÕES	106
6.1	Trabalhos Futuros	108
	REFERÊNCIAS	109
	APÊNDICE A IMAGENS DOS MAPAS DE CUSTOS GERADOS	124
	APÊNDICE B ARQUIVOS DE PARÂMETROS DOS MAPAS DE CUSTO	126

1 INTRODUÇÃO

A presença de robôs em espaços públicos tem crescido. Como mostrado pelo aumento de robôs em espaços como shopping centers, estações de trens e áreas urbanas (While; Marvin; Kovacic, 2020; Mintrom; Sumartojo; Kulić; Tian; Carreno-medrano; Allen, 2021). Com isso, se espera que a expansão das atividades e dos números dos robôs em ambientes públicos tenha implicações diretas em pilares como a saúde pública e produtividade humana nesses ambientes, aumentando, por conseguinte, a qualidade de vida (Salvini; Paez-granados; Billard, 2021).

Mesmo com muito progresso ainda há inúmeros desafios em aberto relacionados as áreas de navegação social, navegação em ambientes populosos e design para alta aceitabilidade de robôs convivendo com humanos na sociedade (Salvini; Paez-granados; Billard, 2021; Rebecchi; Farina; Andreoni; Capolongo; Corno; Perego; Lettieri, 2023; Babel; Kraus; Baumann, 2022). Esses desafios mostram a importância de garantir que robôs autônomos possam navegar em público de forma segura e efetiva, enquanto são aceitos pelos usuários.

Um dos principais sistemas robóticos que tem despertado interesse são as Cadeiras de Rodas Motorizadas Autônomas (CRMA). Para Pineau; West; Atrash; Villemure; Routhier (2011) a popularização das CRMAs seria revolucionário se tratando de robótica assistiva. No contexto das tecnologias assistivas, as Cadeiras de Rodas Autônomas têm se destacado como uma solução inovadora para pessoas com mobilidade reduzida.

Além disso, muitas CRMAs são cadeiras de rodas motorizadas convencionais adaptadas com uma interface de controle, um dispositivo de processamento e alguns sensores (Kappel; Ferreira jr, 2020). Poucos trabalhos apresentam uma cadeira de rodas construída sobre uma base robótica propriamente preparada para a tarefa (Tang; Liu; Hu; Zhou, 2018).

Um fator crucial das CRMAs é a forma como se dá a interação entre o usuário e a cadeira. Diferente do convencional, em que a cadeira de rodas recebe informações de entrada constantemente durante uma trajetória, para realizar o controle, e não precisa dar nenhum *feedback*, as CRMAs utilizam de uma metodologia inversa, na qual o

usuário fornece o mínimo de dados de entrada para que a cadeira consiga planejar o percurso, porém, a cadeira precisa, constantemente, fornecer dados de resposta para que o usuário se sinta seguro e confiante, sem se sentir desconfortável durante o trajeto.

Deste modo, é crucial que as CRMAs sejam capazes de comportar uma série de sensores (Ng; Goh, 2020). Dentre esses sensores, os principais são os odômetros e o Light Detection and Ranging (LiDAR)(Grewal; Matthews; Tea; George, 2017), assim permitindo navegação autônoma, a qual, para um robô tradicional, é uma tarefa de suma importância. Porém, quando se está tratando de robôs carregando humanos a preocupação é ainda maior.

A navegação autônoma convencional para robôs é comumente realizada por meio de três diferentes componentes que executam paralelamente, o planejamento de rotas, a localização e o mapeamento. Um dos módulos centrais que garantem a estabilidade desses métodos são os mapas de custo. Esses mapas são representações em duas dimensões do cenário em forma de grade, atribuindo custos específicos para cada região do mapa, onde esses custos representam uma informação semântica a ser atribuída.

A necessidade de uma navegação eficiente e segura para usuários de CRMA é um dos principais desafios enfrentados. A integração de sistemas de inteligência artificial (IA) e de interação humano-robô (IHR) desempenha um papel crucial na criação de cadeiras de rodas que não apenas se movem autonomamente no ambiente, mas também tem a habilidade interagirem de forma intuitiva com o ambiente e com as pessoas ao redor.

Além disso, robôs que precisam interagir em espaços públicos muitas vezes recorrem à navegação social. Esse tipo de navegação comumente se municia de informações como o espaço proxêmico das pessoas. Desta forma, este espaço proxêmico, seguindo as observações de Hall; Birdwhistell; Bock; Bohannan; Diebold jr; Durbin; Edmonson; Fischer; Hymes; Kimball et al. (1968), atribui ao redor dos humanos diferentes zonas as quais outros humanos, ou robôs nesse caso, podem adentrar sem afetar o conforto. Ou seja, a que ponto o robô pode se aproximar de alguém sem adentrar seu espaço íntimo.

No entanto, há uma diferença na forma como as CRMAs e os robôs de serviço convencionais abordam suas interação com seres humanos. De maneira simplificada, o robô de serviço, durante a navegação social, apenas se preocupa com aspectos relativos à sua interação direta com terceiros, estando mais perto de problemas como a invasão do espaço pessoal desses humanos.

Em contrapartida, quando se trata das CRMAs, a perspectiva muda, visto que elas precisam se preocupar com dois cenários, tanto com uma relação social com seu passageiro, como com os demais pedestres com os quais vai cruzar durante a

navegação. Além disso, a exposição do passageiro vinculado ao fato do constante tráfego em ambientes públicos como calçadas e ambientes internos é um agravante, pois faz com que o robô tenha que se preocupar com as interações provindas do passageiro com o pedestre, ou vice-versa.

Ou seja, no contexto de cadeiras de rodas autônomas a navegação social não pode se restringir apenas à maneira como o robô interage com terceiros, mas também como ele toma as decisões para que não prejudique a experiência de seu próprio passageiro.

Segundo Morales; Miyashita; Hagita (2017), existem três tipos distintos de interações com a CRMA: passageiro-cadeira, pedestre-cadeira e grupo-de-pedestres-cadeira. Onde o passageiro é o humano que utiliza a cadeira, e, portanto, a movimentação da mesma deve ser adequada para proporcionar conforto ao passageiro. Sendo assim, a interação passageiro-cadeira engloba as peculiaridades que melhoram a experiência do usuário na utilização da CRMA, incluindo desde escolhas relacionadas à rota até o *feedback* apresentado ao usuário por meio de alguma interface que melhore o entendimento das decisões tomadas pela CRMA.

Na interação pedestre-cadeira, os pedestres são os outros humanos que compartilham o mesmo ambiente em que a cadeira está navegando. Com isso a cadeira deve garantir uma navegação que não cause desconforto aos demais. Dessa forma, esse tipo de interação envolve ações que a CRMA toma para garantir a melhor interação com os pedestres, aumentando a satisfação deles durante a interação. Por fim, a interação grupo-de-pedestres-cadeira se refere aos cuidados que a CRMA deve tomar para não interromper as interações entre grupos de pedestres, como evitar passar no meio de pedestres que estão conversando, uma vez que essa ação pode ser considerada desrespeitosa e causar desconforto tanto para os pedestres quanto para o usuário da CRMA.

Quando um robô está transportando uma pessoa em um ambiente interno, é crucial considerar o próprio robô como um agente social suscetível ao desconforto. Portanto, a CRMA deve identificar e implementar mecanismos para garantir o conforto de todos os humanos envolvidos em qualquer um das interação.

Isto posto, existem análises de métodos que influenciam no conforto do passageiro das CRMAs. Essas propostas, por exemplo, avaliam o melhor local para se trafegar em ambientes internos ou a visibilidade que o passageiro têm, fatores que afetam diretamente no bem-estar do passageiro durante o processo de navegação. Entretanto, a maioria desses trabalhos avaliam uma estratégia isoladamente, ou apenas métodos voltados ao usuário ou à terceiros, não uma combinação de diferentes tipos de regras sociais.

É importante observar que, se a CRMA não respeitar alguma norma social, isso resulta instantaneamente em um estado de desconforto ou ansiedade para o usuário

que está sendo transportado. Como a CRMA transporta o passageiro, a cadeira se torna uma extensão do humano. Este fenômeno é denominado por Tajfel (1970) como grupo interno.

Assim, quando uma decisão da CRMA perturba outros humanos, referidos por Tajfel (1970) como grupo externo, o passageiro também se sente desconfortável, uma vez que ele sente empatia pelos pedestres.

De modo geral, esses conceitos podem ser estendidos para outros robôs que trafegam em ambientes públicos e que estejam carregando pessoas. Podemos pensar, futuramente, em *hoverboards* ou bases individuais autônomas capazes de suportar o peso de um passageiro.

Esses conceitos serão testados em um ambiente simulado, com atores estáticos e isolados simulando a presença de humanos, utilizando métricas para estimar os fatores sociais e comparar como diferentes combinações de camadas de custo interagem entre si.

1.1 Hipótese

A principal hipótese a ser analisada no estudo é de que a navegação social de CRMAs pode surgir a partir da combinação de diferentes camadas que priorizem diferentes estruturas semânticas com relação ao passageiro e ao pedestre, em conjunto com as camadas de navegação eficiente já dispostas em robôs autônomos.

Levando isso em conta, tanto fatores sociais que refletem o conforto nas interações da CRMA com pedestres, quanto aqueles que levam em consideração o conforto do usuário com o ambiente, precisam ser incorporados e analisados em conjunto, uma vez que a combinação dessas camadas pode afetar o desempenho umas das outras.

Para abordar a hipótese foram realizados testes em ambiente simulado em que o robô tem de navegar por diferentes ambientes. Com isso, os dados das camadas de navegação são coletadas para analisar qual o desempenho do robô seguindo métricas que abordam cada uma das variáveis de interesse, como conforto no local e os valores do espaço proxêmico.

1.2 Objetivos

De modo geral, navegação social em robôs autônomos envolve a maneira como o robô responde à interação sociais com os humanos. Porém, uma cadeira de rodas autônoma, uma vez que carrega um humano, precisa ser tratado como um agente social. Portanto, ela deve se preocupar tanto com questões práticas da navegação, como não gerar rotas que façam com que o passageiro se sinta desconfortável, como priorizar, ainda mais, as regras de espaços proxêmicos, uma vez que os dois agentes

estão sujeitos ao desconforto.

Porém, muitos trabalhos na literatura abordam e testam regras sociais individualmente, não em conjunto, mas na busca de uma solução avançada para navegação social é imprescindível a combinação desses diferentes fatores em suas camadas de mapas de custo.

Para isso o objetivo geral dessa tese é propor e avaliar uma solução de navegação para veículos autônomos carregando pessoas, com foco em CRMAs. Essa solução deve ser integrada ao *Navigation Stack* com base nos mapas de custos, levando em consideração tanto os fatores sociais com relação ao pedestre, como ao passageiro.

O foco central é a combinação dessas diferentes camadas, que mapeiam as semânticas da navegação social, possibilitando assim uma análise para avaliar como elas interagem e quais as peculiaridades que podem emergir dessas interações. Tendo isso em vista, a avaliação terá de incluir uma série de métricas que refletem quesitos sociais importantes, como distância, conforto, visibilidade, suavidade e proximidade dos pedestres, avaliados em diferentes cenários simulados com a presença de atores representando humanos.

Como objetivos específicos provenientes podemos citar:

1. Explorar o que constitui uma cadeira de rodas autônoma, principalmente com relação aos principais componentes, como os sensores;
2. Enumerar os problemas e alternativas de fatores sociais que podem ser convertidos em mapas de custos para a integração em cadeiras de rodas autônomas tanto com relação aos pedestres como com relação aos passageiros;
3. Propor uma solução de múltiplas camadas de custo para navegação social de CRMA, que seja aberta e integrada ao *Navigation Stack* para que futuramente, outros autores possam seguir avançando na área;
4. Estabelecer um conjunto de métricas para avaliação de navegação social que podem ser quantificáveis em ambiente simulado;
5. Testar diferentes configurações das combinações dos mapas de custo escolhidos para entender como as interações entre eles modificam as nuances da navegação social das CRMAs;
6. Coletar dados e comparar como as diferentes configurações de mapas de custo e a navegação autônoma tradicional disponível no *Navigation Stack* se saem em métricas sociais e vice-versa.

1.3 Estrutura da Tese

A tese começará com uma fundamentação teórica sobre tópicos como cadeiras de rodas motorizadas autônomas e navegação, passando pelos trabalhos relacionados na área, avançando para um capítulo onde a abordagem proposta é brevemente introduzida, juntamente com a metodologia e chegando até a descrição dos experimentos, os quais culminam nos resultados finais e conclusão.

De maneira geral o trabalho está dividido da seguinte maneira.

- No Capítulo 2 a fundamentação teórica é exposta, onde é feita, primeiramente, uma introdução detalhada aos principais temas relacionados à cadeiras de rodas motorizadas autônomas, focando em suas partes principais, como são construídas e possíveis alternativas para a interface com o seu passageiro. Após, são exibidos os três pilares da navegação autônoma tradicional: mapeamento, localização e planejamento de rotas. Adicionalmente é introduzido o ROS e o *Navigation Stack* e como cada um desses pilares funciona em conjunto com eles. Por último, é conceitualizada a navegação social, variante da navegação tradicional que visa acrescentar informações semânticas para melhorar a interação das pessoas com os robôs;
- No Capítulo 3 são introduzidos os trabalhos relacionados. Juntamente com os conceitos e fatores que norteiam esse trabalho são apresentadas diferentes formas de adicionar navegação social à cadeiras de rodas autônomas;
- No Capítulo 4 a metodologia é descrita detalhadamente, dando mais contexto sobre as camadas de mapas de custo selecionadas e suas implementações. Em seguida, de modo contíguo, são expostas as ferramentas utilizadas e como foram realizados os experimentos em simulação;
- No Capítulo 5 os resultados dos experimentos conduzidos são detalhados e analisados, dando evidência às interações entre as camadas que podem ser observadas;
- No Capítulo 6, por fim, as conclusões e expectativas de trabalhos futuros são descritos;

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Da Cadeira de Rodas ao Robô Autônomo

A evolução das cadeiras de rodas tem uma ligação intrínseca com os avanços tecnológicos. Com o passar dos anos, os mecanismos e funcionalidades de cadeiras de rodas passaram por mudanças significativas, influenciadas diretamente por melhoramentos contínuos realizados por terapeutas físicos, médicos especialistas e engenheiros (Trzaskoma; Molik; Morgulec-adamowicz; Truszczyńska-baszak; Ogonowska-słodownik; Marszałek; Stępień; Krawczyk; Wójcik, 2017).

Desta forma, essa evolução não focou apenas em melhorar aspectos mecânicos, mas também a experiência do usuário e medidas de segurança (Ardigò; Goosey-tolfrey; Minetti, 2005). Com esse progresso, o design e as capacidades de uma cadeira de rodas evoluíram de forma a proporcionar as necessidades devidas a usuários de acordo com suas necessidades específicas (Ferreira; Souza; Nascimento; Tartaruga; Portela; Mascarenhas; Queiróga, 2017; Cooper, 2009).

2.1.1 Cadeiras de Rodas de Propulsão Manual

Segundo a Organização Mundial da Saúde (OMS), em 2018, cerca de 75 milhões de pessoas necessitariam de cadeiras de rodas, mas apenas uma pequena fração teria acesso a elas ¹.

Esses números estão diretamente relacionados com o aumento rápido da expectativa de vida mundial, sem indícios de melhora das condições de saúde², aliados a problemas como distúrbio de marcha (Pirker; Katzenschlager, 2017), amputações de membros inferiores e doenças como Esclerose Lateral Amiotrófica (ELA) e Lesões na Medula Espinhal (LME)³.

Essa redução de mobilidade tende a levar a problemas psicológicos, os quais podem incluir: baixa autoestima, redução de confiança e motivação, além de uma maior

¹<https://www.who.int/news-room/fact-sheets/detail/disability-and-health>
<https://www.who.int/news-room/fact-sheets/detail/assistive-technology>
<https://www.wheelchairfoundation.org/fth/analysis-of-wheelchair-need/>

²<https://www.who.int/news-room/fact-sheets/detail/10-facts-on-ageing-and-health>

³<https://www.who.int/news-room/fact-sheets/detail/spinal-cord-injury>

propensão a casos de depressão, estresse e ansiedade (Lee; Uihlein, 2019; Vancini; Gomes; Oliveira; De Lira; Rufo-tavares; Andrade; Sarro; Sampaio; Viana; Nikolaidis et al., 2019; Rosenberg; Bombardier; Artherholt; Jensen; Motl, 2013), tendo impacto negativo na qualidade de vida (Urdiales; Poncela; Sanchez-tato; Galluppi; Olivetti; Sandoval, 2007). Ademais, a independência na mobilidade é uma das principais preocupações relatadas pelos pacientes, ficando apenas atrás de tópicos como família e saúde (Kübler; Winter; Ludolph; Hautzinger; Birbaumer, 2005).

Como primeira alternativa para problemas de mobilidade, os usuários frequentemente recorrem às cadeiras de rodas convencionais, ou Cadeiras de Rodas de Propulsão Manual (CRPM). Enquanto algumas das CRPMs podem ter a adição de pequenos motores elétricos para reduzir o esforço físico dos usuários (Joshi; Herrera; Springett; Weedon; Ramírez; Holloway; Ayaz; Dawes, 2019), a grande maioria ainda depende exclusivamente da propulsão gerada pelo humano, girando manualmente as rodas.

De acordo com a Organização Mundial da Saúde (Who, 2018), as CRPM não apenas aumentam o acesso à educação e ao mercado de trabalho, como também contribuem para a redução de problemas associados, como úlceras de pressão e contraturas musculares, graças ao movimento constante necessário para a locomoção. Além disso, as CRPM são caracterizadas por sua leveza e compacidade, o que as torna uma opção portátil e prática. Um exemplo de CRPM pode ser visualizado na Figura 1.

Embora as Cadeiras de Rodas de Propulsão Manual (CRPM) ofereçam diversas vantagens, como portabilidade e aumento do acesso à educação e ao mercado de trabalho, seu uso contínuo pode acarretar problemas significativos. Pacientes com tetraplegia ou mobilidade reduzida na parte superior do corpo frequentemente dependem de assistência de outra pessoa para operar uma CRPM.

Além disso, o uso repetitivo das CRPM está associado a problemas como estresse em demasia nos músculos e articulações, tendinites e bursites principalmente na região dos ombros, seguido pelos cotovelos e pulsos, além da fadiga inerente ao seu uso (Lee; Uihlein, 2019; Morgan, 2015; Bertolaccini; Filho; Christofolletti; Paschoarelli; Médola, 2017; Morrow; Hurd; Kaufman; An, 2010). Mesmo com modificações destinadas a melhorar a eficiência e ergonomia das CRPM (Gorce; Louis, 2012), os danos, especialmente na região do ombro, persistem a longo prazo (Walford; Requejo; Mulroy; Neptune, 2019).

2.1.2 Cadeiras de Rodas Motorizadas

Para contornar as limitações das Cadeiras de Rodas de Propulsão Manual (CRPM), existem as Cadeiras de Rodas Motorizadas (CRM). As CRM são operadas, comumente, por meio de um joystick acoplado em um dos apoios para braços, utili-

Figura 1 – Exemplo de CRPM da marca Freedom.



Fonte: <https://freedom.ind.br>

zando energia elétrica armazenada em baterias recarregáveis para alimentar o motor e permitir o deslocamento. Um exemplo pode ser visto na Figura 2, onde a presença do joystick no apoio de braço esquerdo é observável.

No entanto, pessoas com condições como ELA, LME ou tetraplegia, ou ainda com problemas que afetam a cognição ou sensibilidade, que sofrem com espasmos, ou que não possuem os movimentos finos necessários para a operação do joystick acabam enfrentando dificuldades para realizar sua própria locomoção (Yuliana; Riyadi, 2022; Leaman; La, 2017; Jang, 2024; Zhang; Li; Zhang; Liu, 2024; Marchal-crespo; Furumasu; Reinkensmeyer, 2010).

Dado esse contexto, vários autores propõem modelos de Cadeiras de Rodas Motorizadas Autônomas (CRMA) como uma solução alternativa para esses casos específicos.

2.1.3 Cadeiras de Rodas Motorizadas Autônomas

O avanço da robótica com foco em ambientes internos, ou ambientes domésticos, tem tido impactos significativos com relação ao nível de mobilidade, autonomia e experiência dos usuários com os quais tem contato. Robôs em ambientes internos atacam problemas que englobam desde veículos autônomos em logística, até robôs de serviço

Figura 2 – Exemplo de CRM da marca Freedom.



Fonte: <https://freedom.ind.br>

operando em hospitais, os quais ampliam as fronteiras para inovações em navegação e automação em ambientes indoor (Simpson; Lopresti; Hayashi; Nourbakhsh; Miller, 2004).

A evolução de robôs desenvolvidos especificamente para o ambiente doméstico tem impulsionado, por sua significativa intersecção, o desenvolvimento de Cadeiras de Rodas Motorizadas Autônomas (CRMA). Por conseguinte, esse progresso tem sido guiado por avanços contínuos em áreas como Navegação Autônoma, Automação, Inteligência Artificial (IA) e Interação Humano-Robô (IHR), as quais auxiliam no desenvolvimento de sistemas inteligentes para melhorar a mobilidade e independência de indivíduos portadores de algum tipo de deficiência (Cui; Shang; Yu; Wang, 2024; Zal; Chen; Chi; Chen, 2013).

O desenvolvimento e a popularização de CRMAs, segundo Pineau; West; Atrash; Villemure; Routhier (2011), seria um processo tão disruptivo quanto a evolução das CRPMs para CRMs.

As CRMAs visam simplificar a interface entre o usuário e a cadeira de rodas, permitindo que o usuário forneça comandos de alto nível que são convertidos em ações complexas pelo sistema de controle da cadeira (Hartman; Nandikolla, 2019).

Estas cadeiras utilizam sensores variados, como LiDARs para mapeamento do ambiente e encoders para odometria, permitindo a navegação autônoma e o desvio de obstáculos (Grewal; Jayaprakash; Matthews; Shrivastav; George, 2018; Tang; Liu; Hu; Zhou, 2018).

A CRMA pode ser considerada autônoma quando executa tarefas em um cenário dinâmico sem necessidade de interação contínua com humanos. Contudo, o nível de autonomia pode variar, necessitando, em algumas tarefas, da supervisão constante do usuário (Urdiales; Poncela; Sanchez-tato; Galluppi; Olivetti; Sandoval, 2007). Para garantir a usabilidade e conforto, as CRMA utilizam diferentes interfaces de controle, como Eletroencefalogramas (EEG) e Eletromiogramas (EMG), que são especialmente úteis para usuários com tetraplegia (Xiong; Hotter; Nadin; Patel; Tartakovsky; Wang; Patel; Axon; Bosiljevac; Brandenberger et al., 2019; Chaudhary; Xia; Silvoni; Cohen; Birbaumer, 2017; Nijboer; Sellers; Mellinger; Jordan; Matuz; Furdea; Halder; Mochty; Krusienski; Vaughan et al., 2008; Hochberg; Serruya; Friehs; Mukand; Saleh; Caplan; Branner; Chen; Penn; Donoghue, 2006).

Além das interfaces biológicas, telas sensíveis ao toque são frequentemente empregadas, permitindo ao usuário visualizar o mapa gerado pela CRMA e selecionar pontos de destino para navegação autônoma. A inclusão de conceitos de navegação social é crucial para a aceitação das CRMA, garantindo que a interação com outros pedestres seja confortável e segura (Kruse; Pandey; Alami; Kirsch, 2013). Trabalhos na literatura destacam a importância de incorporar regras sociais na navegação das CRMA, assegurando que o usuário se sinta seguro e confortável durante a locomoção (Morales; Kallakuri; Shinozawa; Miyashita; Hagita, 2013; Morales; Even; Kallakuri; Ikeda; Shinozawa; Kondo; Hagita, 2014).

Adicionalmente, as CRMA também são parte integrante da área de Ambient Assisted Living (AAL), que busca integrar sensores, métodos e tecnologias para melhorar a qualidade de vida, especialmente de pessoas idosas ou com deficiências físicas (Bardaro; Bascetta; Ceravolo; Farina; Gabellone; Matteucci, 2018; Cavanini; Benetazzo; Freddi; Longhi; Monteriu, 2014).

2.1.3.1 Exemplos de CRMAs

Em um trabalho realizado previamente, definimos, por meio de uma revisão de literatura, quais componentes, comumente, são utilizados para a definição de CRMAs. Foi observado que, além da base principal, outros três quesitos englobam a construção dessas cadeiras autônomas. São eles: sensores, interface de controle e dispositivo de processamento (Kappel; Ferreira jr, 2020). Entretanto, em um pequeno número de trabalhos foi observado o acoplamento de atuadores como manipuladores em robôs (Tang; Liu; Hu; Zhou, 2018).

Para a construção de Cadeiras de Rodas Motorizadas Autônomas (CRMA), duas metodologias principais são empregadas. A mais comum envolve a adaptação de uma CRM existente com uma série de sensores, uma interface para permitir a comunicação entre o usuário e a cadeira, e um dispositivo de processamento. Alternativamente, uma base robótica adequada pode ser equipada com um assento para transportar

uma pessoa, juntamente com uma interface de controle para comunicação entre as duas partes (Leaman; La; Nguyen, 2016). Ademais, o design externo de uma CRMA pode variar conforme a aplicação.

Uma abordagem básica para montar uma CRMA, capaz de realizar mapeamento, planejamento de rotas e localização, é descrita por Grewal; Matthews; Tea; George (2017), apresentada na Figura 3. Este modelo utiliza uma base de CRM, dois encoders para odometria, um LiDAR para mapeamento, um laptop e dois microcontroladores Arduino para computação. Além disso, o laptop também serve como interface, onde o usuário seleciona o destino no mapa gerado pelo LiDAR, usando uma ferramenta para visualização 3D.

Figura 3 – CRMA proposta por Grewal; Matthews; Tea; George (2017)

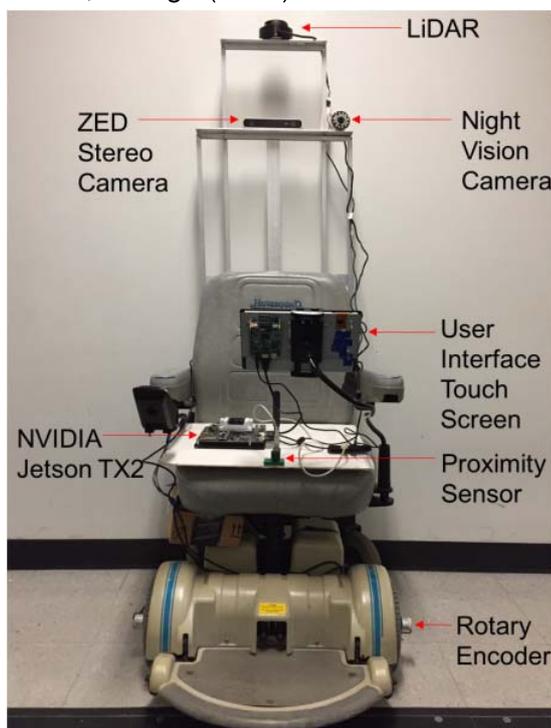


Fonte: retirada de Grewal; Matthews; Tea; George (2017)

O projeto de Grewal; Matthews; Tea; George (2017) foi adaptado em outros estudos. Em Grewal; Jayaprakash; Matthews; Shrivastav; George (2018) os autores adicionaram uma câmera e um medidor de distância a laser para identificar pontos de interesse. Já em Contractor; Grewal; Jayaprakash; Matthews; George (2018) os autores substituíram a câmera e o LiDAR por uma câmera de visão noturna, uma câmera estéreo e um sensor ultrassônico, além de trocar o laptop por uma Jetson TX2 e usar uma tela sensível ao toque como interface (Figura 4).

Em outra adaptação, Grewal; Matthews; Tea; George (2017) substituíram a interface por um sistema sip-and-puff, permitindo que usuários tetraplégicos interagem com a cadeira por meio de soprar ou sorver ar em um tubo somente.

Figura 4 – Versão posterior da cadeira exposta na Figura 3, apresentada por Contractor; Grewal; Jayaprakash; Matthews; George (2018)



Fonte: retirada de Contractor; Grewal; Jayaprakash; Matthews; George (2018)

Um projeto de CRMA semelhante aos apresentados anteriormente, com LiDAR e encoders, foi adaptado para pacientes sem movimento dos membros superiores utilizando uma Interface Cérebro-Computador (ICC) (Ng; Goh, 2020). Diferentemente dos modelos anteriores que usavam telas sensíveis ao toque, este projeto elimina a necessidade de interação física com a CRMA.

O design da cadeira de Ng; Goh (2020) é mostrado na Figura 5. Por meio de ICC o usuário pode controlar a cadeira e navegar pelo mapa. Além disso, o usuário pode “salvar” destinos e usar o modo autônomo da cadeira para retornar a esses pontos. Uma ferramenta de visualização 3D permite ver o mapa em tempo real, além de informações como posição atual e destino.

Em contraste com os designs de CRMA baseados em cadeiras de rodas motorizadas, Tang; Liu; Hu; Zhou (2018) apresentaram uma CRMA construída a partir de um chassi com rodas omnidirecionais, equipada com dois LiDARs, encoders nas rodas, um Kinect para detecção de objetos e uma ICC para controle.

Leaman; La; Nguyen (2016) propuseram uma plataforma removível destinada a transformar qualquer CRM em uma CRMA com poucos ajustes (Figura 6). Esta plataforma inclui diferentes opções de interface de controle, sensores para mapeamento, localização e navegação, e espaço para acomodar um laptop para realizar o proces-

Figura 5 – CRMA desenvolvida por Ng; Goh (2020)



Fonte: retirada de Ng; Goh (2020)

samento.

Figura 6 – Plataforma desenvolvida por Leaman; La; Nguyen (2016)



Fonte: imagens retiradas de Leaman; La; Nguyen (2016)

2.1.3.2 Componentes Principais de Uma CRMA

Os principais componentes que constituem uma CRMA podem ser distribuídos em três categorias diferentes: Sensores, Interfaces de Controle e Dispositivos de processamento

2.1.3.2.1 Sensores

Sensores são essenciais para a coleta de dados do ambiente, evitando problemas como colisões das CRMA com pessoas ou objetos, e garantindo a segurança do usuário e de terceiros. Diferentes sensores têm funções específicas, e a combinação de

vários sensores aprimora a percepção espacial das CRMAs.

Sensores como LiDAR (Figura 7(a)), infravermelho (Figura 7(c)), e ultrassônicos (Figura 7(d)) medem distâncias entre o emissor e objetos ao redor. O LiDAR, com maior ângulo de cobertura, é utilizado para mapear o ambiente. Encoders (Figura 7(b)) fornecem dados de odometria para estimar o movimento da cadeira. Acelerômetros e giroscópios, ou uma IMU, auxiliam na estimativa da posição em tempo real (Ibrahim; Peeie, 2020; Sollehudin; Heerwan, 2020).

Figura 7 – Diferentes Sensores

EAI



(a) LIDAR modelo YDLidar X4



(b) Encoder modelo B58N



(c) Sensor Infravermelho



(d) Sensor Ultrassônico modelo HC-SR04

Fonte: imagens retiradas da internet

Sensores ultrassônicos (Figura 7(d)) são comuns em CRMAs semi-autônomas ou de baixo custo para detectar objetos ao redor (Okwu; Tartibu; Ayomoh; Ighalo, 2020; Yashoda; Piumal; Polgahapitiya; Mubeen; Muthugala; Jayasekara, 2018; Kumar; Ma-

Ihotra; Sharma, 2020). Xiong; Hotter; Nadin; Patel; Tartakovsky; Wang; Patel; Axon; Bosiljevac; Brandenberger et al. (2019) propôs uma CRM semi-autônoma controlada por biosinais, que utiliza sensores ultrassônicos para desviar de obstáculos, seguir paredes e evitar escadas ou descidas íngremes.

Adicionalmente, câmeras também podem ser usadas para odometria visual, detecção de objetos e obtenção de informações semânticas do local. Alkhatib; Swaidan; Marzouk; Sabbah; Berjaoui; Diab (2019) utilizaram câmeras para detectar QR Codes que guiam a navegação da CRMA, enquanto sensores infravermelho evitam possíveis colisões. Baiju; Varghese; Alapatt; Joju; Sagayam (2020) usam câmeras com aprendizado de máquina para navegação em cenários pré-treinados. Lee; Lim; Eu; Goh; Tew (2017) emprega visão computacional para detectar obstáculos no chão. Hartman; Nandikolla (2019) utilizam câmeras e Laser Rangefinders para identificar obstáculos no terreno.

A combinação de LiDAR e encoders é uma abordagem simples para transformar uma CRM em CRMA, permitindo mapeamento e localização (Grewal; Matthews; Tea; George, 2017; Ng; Goh, 2020; Tang; Liu; Hu; Zhou, 2018). Como exemplo, no trabalho de Grewal; Matthews; Tea; George (2017), os autores adicionaram uma câmera e um LiDAR fixo para identificar pontos de interesse. Subsequentemente, no trabalho de Contractor; Grewal; Jayaprakash; Matthews; George (2018), substituíram esses sensores por uma câmera de visão noturna, uma câmera estéreo e um sensor ultrassônico, melhorando a detecção em diferentes condições de luz.

Câmeras estéreo ou sensores de percepção 3D, como ZED⁴ (Figura 8(a)) e Real-sense⁵ (Figura 8(b)), ou o Kinect (sensor da Microsoft) tem se tornado cada vez mais acessíveis (Rusu; Cousins, 2011). Esses sensores, permitem a percepção tridimensional do ambiente por meio de mapas de profundidade ou nuvens de pontos. A Point Cloud Library (PCL) é uma biblioteca amplamente utilizada para processar e visualizar essas nuvens de pontos, habilitando diversas ferramentas para filtragem, segmentação e reconhecimento de objetos, facilitando a integração desses dados em sistemas de navegação robótica.

Outra aplicação popular para esse tipo de câmera é a odometria visual (Nistér; Naroditsky; Bergen, 2004). Essa prática utiliza a sequência de imagens capturadas por câmeras para estimar o movimento do robô ao longo do tempo, enquanto o Visual-SLAM (Visual Simultaneous Localization and Mapping)(Mur-artal; Tardós, 2017) vai além, construindo um mapa do ambiente simultaneamente à localização do robô dentro desse mapa. O V-SLAM corrige erros de localização (drift) que podem ocorrer devido a imprecisões acumuladas ao longo do tempo, permitindo uma navegação com

⁴<https://www.stereolabs.com/zed/>

⁵<https://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html>

Figura 8 – Câmeras de profundidade



(a) ZED - StereoLabs

(b) Realsense - Intel

Fonte: imagens retiradas da internet

mais precisão e confiabilidade.

2.1.3.2.2 Interfaces de Controle

As CRPM tradicionais davam mobilidade aos usuários, porém, havia o revés de fadiga com o uso, isso excluindo o fato de que nem todos os portadores de deficiência eram dotados da capacidade de propelir a cadeira.

Com o advento das CRM a interface padrão para o movimento virou o joystick. No entanto, ainda havia o problema de que nem todo portador de deficiência dispunha de uma forma de utilizar o joystick ou dos movimentos finos muitas vezes necessários para conduzir a cadeira.

Diferentes interfaces foram propostas para facilitar o controle dessas cadeiras por usuários acometidos por diferentes deficiências. As CRMAs, por sua vez, também trazem a vantagem de terem a capacidade de serem adaptáveis à diferentes interfaces, uma vez que a interface apenas traduz comandos executados em alto-nível pelo usuário para comandos em baixo nível para o sistema de controle da cadeira, que executa a ação de forma autônoma (Hartman; Nandikolla, 2019).

Um exemplo desse tipo de aplicação é o trabalho de Grewal; Matthews; Tea; George (2017), que utiliza uma mesma base em diferentes trabalhos, mas interfaces de interação diferentes, como laptop (mouse e teclado) (Grewal; Matthews; Tea; George, 2017), sip-and-puff (Grewal; Matthews; Tea; Contractor; George, 2018) e touchscreen (Contractor; Grewal; Jayaprakash; Matthews; George, 2018).

Usar um laptop como interface é uma maneira simples de projetar comandos de entrada, exibir informações e ainda ter uma plataforma de processamento (Grewal; Matthews; Tea; George, 2017; Bardaro; Bascetta; Ceravolo; Farina; Gabellone; Matteucci, 2018). No entanto, muitas vezes é difícil para que o condutor faça uso da interface.

Algumas das principais interfaces de controle alternativas utilizadas em CRMAs são:

Telas sensíveis ao toque: Telas sensíveis ao toque apresentam a vantagem de permitir a interação do usuário com a CRMA de uma maneira gráfica e intuitiva. O usuário pode ver em tempo real as informações coletadas pelos sensores e tomar decisões a partir delas. Um exemplo é o trabalho de Contractor; Grewal; Jayaprakash; Matthews; George (2018), que coleta possíveis pontos de interesse do cenário e os apresenta em forma de lista ao usuário para que ele possa escolher, com facilidade, o destino a partir desses pontos. A desvantagem desse mecanismo é que mesmo muitas vezes não requerendo o mesmo nível de coordenação e de movimentos finos quando comparado ao joystick, pessoas com tetraplegia, por exemplo, ainda não conseguem fazer uso das telas sensíveis ao toque.

Sip-and-puff: Ao contrário dos joystick ou telas sensíveis ao toque, onde o usuário necessita ter algum nível de movimento nos membros superiores para a operação, a interface sip-and-puff⁶ não necessita, tornando ela mais indicada para paciente tetraplégicos, por exemplo. O sip-and-puff (Figura 9) é operado a partir do sorver ou soprar de ar em um tubo. É possível mapear esses atos para ações no sistema e com isso possibilitar o controle da cadeira. A desvantagem dessa ferramenta é o custo elevado e tempo de habituação do usuário.

Figura 9 – Sip-and-puff produzido pela Origin Instruments



Fonte: site da Origin Instruments³

Biosinais: Para casos em que o usuário não dispõe de movimentos dos membros superiores, provenientes, por exemplo de LME ou doenças neurodegenerativas como a ELA, os biosinais são uma alternativa de interface para controle da cadeira autônoma (Xiong; Hotter; Nadin; Patel; Tartakovsky; Wang; Patel; Axon; Bosiljevac; Brandenberger et al., 2019). Existem várias maneiras de obter esses biosinais. As maneiras mais comuns talvez sejam o Eletroencefalograma (EEG), o Eletromiograma (EMG) e o Eletrocardiograma (ECG). Além disso, esses métodos podem ser combinados para

⁶https://www.orin.com/access/sip_puff/

diferentes aplicações específicas.

2.1.3.2.3 Dispositivos de processamento

Diversos dispositivos de processamento podem ser empregados para atender às necessidades de uma CRMA. O ponto crucial é a capacidade do dispositivo de executar nodos do Robot Operating System (ROS), facilitando a integração dos diferentes componentes do sistema.

O dispositivo, ou grupo de dispositivos, escolhido deve ser capaz de processar simultaneamente dados dos sensores, interfaces de controle e algoritmos de navegação, mapeamento e localização.

Computadores portáteis, como os IntelNUC, são uma opção lógica para fonte central de processamento. O tamanho compacto facilita a integração em robôs de diversos tamanhos. Além disso, o desempenho é adequado para garantir a execução de diversos processos simultaneamente. Ademais, mini PCs oferecem diversas opções de conectividade. Entretanto, o custo e o consumo energético são mais elevados em comparação a microcontroladores ou Computadores de Placa Única (*Single Board Computer* - SBC).

Laptops são amplamente utilizados devido à sua flexibilidade e múltiplas interfaces de entrada, além de fornecer feedback visual ao usuário através de um monitor, que pode ser sensível ao toque. Outra vantagem significativa é a presença de uma bateria interna, evitando a possível necessidade de drenagem da bateria da cadeira .

Soluções mais portáteis, como a Raspberry Pi, podem executar nodos do ROS e outras aplicações simultaneamente, com baixo consumo de energia, embora não ofereçam a mesma flexibilidade de um laptop.

A Nvidia Jetson, similar à Raspberry Pi, integra uma Unidade de Processamento Gráfico (*Graphics Processing Unit* - GPU). Com o aumento do uso de aplicações que contam com aprendizado de máquina na robótica, como na detecção ou segmentação de objetos, ou com processamento de dados gerados pelos sensores, como nuvens de pontos, a Nvidia Jetson oferece uma alternativa compacta para essas tarefas.

Microcontroladores como Arduinos, ESPs, ou similares são amplamente usados em diversos projetos devido à sua capacidade de comunicar sensores e outros dispositivos de processamento, apesar de sua limitação em processamento para tarefas intensivas. No entanto, esses microcontroladores precisam dispor de alguma forma de integração ao ROS, com o intuito de facilitar a troca de mensagens entre componentes e a conversão de dados dos sensores para serviços ou mensagens em tópicos do ROS com latência baixa.

2.2 Princípios da Navegação Autônoma

Navegação autônoma se refere à habilidade de um robô, ou veículo autônomo, de se mover independentemente, de um ponto ao outro, percebendo o ambiente, sem intervenção humana. Sendo assim, a navegação autônoma é composta por uma série de tecnologias e metodologias que permitem a resolução dessa tarefa (Lu, 2014).

De um modo geral, navegação é uma tarefa resolvida para a maioria dos robôs. Os algoritmos necessários para habilitar navegação conseguem lidar com grandes volumes de dados gerados pelos sensores e manter registro dos locais com espaços livres ou ocupados com a precisão necessária (Lu, 2014).

Esses algoritmos são o suficiente para a maioria dos casos de uso em que um robô precisa se deslocar do ponto A ao ponto B, em um dado cenário, quando o que se busca é somente a eficiência temporal ou espacial para achar caminhos livres de colisão (Lu, 2014).

Para garantir um fluxo de execução do processo de navegação, muitas vezes, a tarefa como um todo, é dividida em subproblemas. De maneira geral, esses subproblemas podem ser dividido em três partes: Mapeamento, Localização e Planejamento de Rota.

2.2.1 Mapeamento

O mapeamento, na robótica, especialmente quando o contexto são ambiente internos, envolve criar a representação do ambiente para que um robô possa executar uma série de tarefas, que abrangem a tomada de decisão por meio da localização e navegação. Essa representação pode incluir informações espaciais do ambiente, obstáculos, zonas de interesse, ou objetivos, e outras características relevantes do cenário (Kostavelis, 2017; Slimane; Khireddine; Chafaa, 2013; Lee; Lim; Eu; Goh; Tew, 2017).

Existem uma gama de maneiras de representar um ambiente como um mapa na robótica, essas podem incluir, por exemplo, mapeamento baseado em grade, usando informações topológicas, ou baseada em dicionário (Wang; Cui; Zhao; Yang; Xu, 2009; Tomomi; Kanji, 2010; Lee; Park; Yu; Kim et al., 2011).

Cada uma das técnicas envolve algoritmos diferentes e extração de dados de sensores diferentes para representações fieis do cenário.

Em contrapartida, em alguns casos, o cenário pode ser muito amplo, ou as capacidades de obtenção de dados do robô podem não ser ricas o bastante. Desta forma, múltiplos robôs podem colaborar para criar um mapa global. Essa abordagem oferece benefícios como uma exploração mais rápida, maior tolerância a falhas e uma riqueza de dados maior por conta dos diversos sensores captando esses dados (Anderson, 2019; Hidayat; Trilaksono; Hindersyah, 2017). O processo envolve cada robô adquirir

dados do mapa, transmitir esses dados para os múltiplos robôs envolvidos e combinar esses mapas gerados (Lu; Yang; Zhao; Cheng, 2021). Um método probabilístico para combinar esses mapas alinha cada mapa individualmente em um mapa global coeso (Lee; Lee; Choi; Lee, 2011).

Há ainda avanços na área de SLAM que permitem que os robôs determinem sua posição ao mesmo tempo que adquirem dados de um mapa em um ambiente desconhecido (Mur-artal; Montiel; Tardós, 2015; Yuan, 2024).

Em suma, mapeamento é crucial na robótica para percepção espacial, garantindo assim navegação e interação com o ambiente de forma efetiva.

2.2.2 Localização

A localização na robótica abrange o processo pelo qual os robôs determinam sua posição dentro de um ambiente. Essa determinação pode ser alcançada por meio de vários métodos. Esses métodos podem ser divididos sobre duas metodologias diferentes.

A primeira é a localização absoluta, onde os robôs estabelecem sua pose em relação a um sistema de coordenadas global, por exemplo, por meio de avaliação de dados recebidos pelos sensores sobre as características do ambiente (Issa; Rashid, 2019).

A segunda é a localização relativa, que conta com leituras de sensores como a *Inertial Mesurement Unit* (IMU) ou os odômetros para estimar a pose do robô ao longo do tempo. Essa estimativa se dá por meio do resultado da integração dos movimentos de translação e orientação providos pelos sensores. Esse método está, naturalmente, mais sujeito a sofrer com erros acumulados ao longo do tempo.

Sendo assim, localização é crucial para permitir que os robôs naveguem de forma autônoma no seu ambiente. Métodos como a *Monte Carlo Localization* (MCL) utilizam uma técnica de localização por filtros de partícula para rastrear a pose do robô a partir de um mapa previamente conhecido (Hussein; Maged; Adel; Sabry; Abobakr; Mustafa; Dakroury; Morsy, 2021). As partículas representam possíveis posições do robô no mapa. A cada iteração o algoritmo cria um ranqueamento das partículas mais prováveis de acordo com as leituras dos sensores e previsões do movimento. Com isso, a partícula mais provável é assumida como a localização absoluta do robô (Dellaert; Fox; Burgard; Thrun, 1999).

Um famoso método para realizar localização relativa é o *Kalman Filter* (KF). Este método opera em um sistema de loop fechado, em que, dada uma estimativa inicial e um modelo matemático do movimento do robô, o método pode prever a sua posição ao longo do tempo. O KF, desta forma, compara o resultado do modelo com os dados recebidos pelos sensores. Uma vez que esses dados tendem a ser ruidosos o KF ajusta suas estimativas com as correções necessárias, de acordo com a confiabilidade

de cada sensor. Tendo assim uma previsão da posição do robô a cada instante de tempo (Kalman, 1960).

2.2.3 Planejamento de Rotas

O planejamento de rotas envolve determinar um caminho ótimo ou sub-ótimo para um robô navegar de um ponto inicial a um ponto final enquanto evita obstáculos em seu caminho (Zhang; Wang; Chen, 2018; Zhang; Zhao, 2022). Isso torna o planejamento de rotas um dos problemas mais fundamentais e essenciais para várias aplicações na robótica, garantindo uma operação segura e eficiente (Ding, 2020; Suwoyo, 2024).

Vários algoritmos visam encontrar as trajetórias mais eficientes para robôs, considerando fatores como evasão de obstáculos, critérios de otimização e ambientes dinâmicos (Suwoyo, 2024). Exemplos populares são o Dijkstra (Dijkstra et al., 1959) e o A* (Hart; Nilsson; Raphael, 1968).

Existem diferentes abordagens ao planejamento de rotas na robótica. Uma categorização comum é entre planejamento de rotas global e local (Liu; Wang; Wu; Wei, 2023; Wang; Cui; Zhao; Yang; Xu, 2009). O planejamento de rotas global envolve a determinação de um caminho que cobre todo o espaço de busca, traçando uma rota do ponto inicial ao final. Diferente do planejamento de rota local, que foca na evasão imediata de obstáculos e navegação em um espaço contido ao redor da posição atual do robô.

Na maioria das vezes, quando pensamos nesses algoritmos imaginamos os robôs traçando uma trajetória em um Espaço Cartesiano (EC), geralmente representados por um sistema de células em uma grade, alterando a pose do robô com o tempo. Adicionalmente, na robótica, podemos definir trajetórias também no Espaço de Juntas (EJ), o que é comum quando trabalhamos com manipuladores com diversos Graus de Liberdade (GdL) (Luo; Zhang; Zhuang; Ma; Li, 2023). Neste trabalho, posteriormente, sempre que citarmos o planejamento de rotas, estaremos mencionando um planejador operando no EC para movimentação da base do robô.

2.2.4 ROS e o Navigation Stack

O ROS é um conjunto de bibliotecas e ferramentas que tem a proposta de facilitar o desenvolvimento de sistemas robóticos (Quigley; Conley; Gerkey; Faust; Foote; Leibs; Wheeler; Ng, 2009). Por conseguinte, o ROS proporciona um meta-sistema operacional para robôs, incluindo funcionalidades como abstração de hardware, controle de dispositivos de baixo nível, passagem de mensagens entre processos, gerenciamento de pacotes e independência de linguagem⁷.

Ademais, é extensamente utilizado tanto na academia quanto na indústria, demo-

⁷<http://wiki.ros.org/ROS/Introduction>

cratizando a criação de *middlewares* para robôs que atuam em variadas aplicações (Dehnavi; Koedam; Nelson; Goswami; Goossens, 2021). Essa facilitação do processo de desenvolvimento se dá pela natureza de software livre, modularidade e escalabilidade, o que permite com que novos pacotes sejam feitos e disponibilizados rapidamente (Leu; Ristić-durrant; Slavnić; Glackin; Salge; Polani; Badii; Khan; Raval, 2013; Santos; Cunha; Macedo, 2019).

Uma das características chave do ROS é o sistema de mensagens que pode ser utilizado a partir de várias linguagens. Isso faz com que o sistema como um todo possa funcionar como uma coleção independente de vários executáveis chamados de nodos. Um dos meios de troca de mensagens disponíveis são os tópicos, os quais funcionam em um esquema de publicar e inscrever. Cada tópico tem um tipo de mensagem estruturado e padronizado. Sendo assim, câmeras podem publicar mensagens do tipo *Image* em um tópico correspondente, um LiDAR publica mensagens do tipo *LaserScan* e odômetros do tipo *Odometry*, independente do modelo do sensor.

Desta forma, isso faz com que nodos possam ser executados independentemente, apenas aderindo ao tipo de mensagem descrito. Cada nodo pode ser escrito em uma linguagem diferente e mandar mensagens através de diferentes dispositivos (Lu, 2014).

Por exemplo, um robô pode dispor de uma câmera e um primeiro nodo publica as imagens dessa câmera em um tópico do tipo *Image*. Em seguida, um outro nodo pode consumir os dados desse tópico e esse segundo nodo pode realizar detecção de objetos nas imagens recebidas. Porém, se em algum momento, seja necessário trocar esse segundo nodo por um novo, em outra linguagem, que realiza segmentação nas imagens, não seria necessário alterar o código do nodo que publica a captura. De mesmo modo, seria possível trocar o hardware da câmera, ou aplicar um filtro na primeira imagem, antes da publicação, sem a necessidade de alterar o segundo nodo. Em um terceiro caso, também é possível executar um desses nodos, ou os dois, em um robô diferente sem mudanças no fluxo empregado.

Os nodos do ROS são organizados em pacotes, os pacotes por sua vez em *stacks*. Uma dessas *stacks* é o *Navigation Stack*, que se tornou componente fundamental do ROS por prover, dentre outras funções, mapeamento, localização e planejamento de rota.

Uma parte importante, dentro do *Navigation Stack*, é o nodo chamado *move_base*, o qual, após configurado de acordo com o robô, disponibiliza uma interface para interagir com o *Navigation Stack* e possibilita o acesso aos planejadores de rota, bem como outras funcionalidades.

2.2.4.1 Mapeamento Com Navigation Stack

O processo de mapear um ambiente utilizando o *Navigation Stack*, geralmente envolve a utilização de algum tipo de SLAM, muitos desses já fornecidos dentro de pacotes. Algoritmos de SLAM abordam o desafio de iniciar um robô em uma localização desconhecida dentro de um ambiente desconhecido e construir um mapa de forma incremental enquanto simultaneamente o utiliza para determinar a localização absoluta do robô (Dissanayake; Newman; Clark; Durrant-whyte; Csorba, 2001; Labbé; Michaud, 2018).

Um dos pacotes comumente usados no ROS para mapear ambientes é o GMapping. Esse pacote é amplamente empregado para gerar um mapa de ocupação 2D baseado em grade de um ambiente com base no movimento e nas leituras sucessivas do LiDAR. A Figura 10 apresenta um modelo de mapa sendo criado usando o GMapping, as áreas em cinza escuro são regiões desconhecidas, as áreas em cinza claro são regiões livres, ou seja, que o LiDAR não encontrou obstáculos ou objetos, já as áreas em preto são obstáculos detectados.

Figura 10 – Execução do processo de construção do mapa através de SLAM



Fonte: Imagem gerada pelo autor.

Esse processo de mapeamento é crucial para as tarefas de navegação subsequentes, pois captura a estrutura estática do ambiente em dado momento, permitindo assim que algoritmos de planejamento de rotas evitem objetos dinâmicos de forma

reativa (Pavón-pulido; Blasco-garcía; López-riquelme; Feliú-batlle; Oterino-bono; Hertero, 2023).

2.2.4.2 *Localização Com Navigation Stack*

Uma vez com um mapa gerado o robô precisa se localizar nesse mapa. Na seção 2.2.2 vimos que o MCL é um método que permite ao robô desempenhar a localização com base em leituras de sensores e comparação com o ambiente. Um dos nodos do ROS disponibiliza o *Adaptative Monte Carlo Localization* (AMCL), que é uma versão melhorada do MCL.

Ambas as técnicas utilizam um filtro de partículas para estimar a posição do robô. Porém, a principal diferença entre ambos é o número de partículas utilizadas. O MCL mantém um número fixo, já o AMCL tem uma quantidade variável. Desta forma, o AMCL adapta o número de partículas com base na incerteza da localização do robô. Quando a incerteza é alta, mais partículas são usadas para cobrir uma maior área do espaço possível. Quando a incerteza é baixa, menos partículas são necessárias, economizando recursos computacionais (Thrun, 2002). Esse comportamento pode ser observado na Figura 11, onde na Figura 11(a), pelo fato do robô não ter se movido, tendo poucas leituras sua incerteza era alta, culminando em um número elevado de partículas, os quais foram reduzindo, conforme o número de observações do cenário ia aumentando nas Figuras 11(b) e 11(c).

Na seção 2.2.2 vimos também estratégias para obter a posição relativa do robô, com base em seus sensores. Um dos principais métodos disponíveis no ROS e que pode ser integrado ao *Navigation Stack* é o Extended Kalman Filter (EKF). Diferente do KF, relatado anteriormente, o EKF é uma versão avançada, adaptado para lidar com sistemas não lineares. Em muitas situações reais, como a navegação de um robô, o movimento e as medições não seguem linhas retas ou simples, mas têm curvas e relações complexas. O EKF aproxima essas não linearidades podendo assim fornecer estimativas precisas (Smith; Schmidt; Mcgee, 1962).

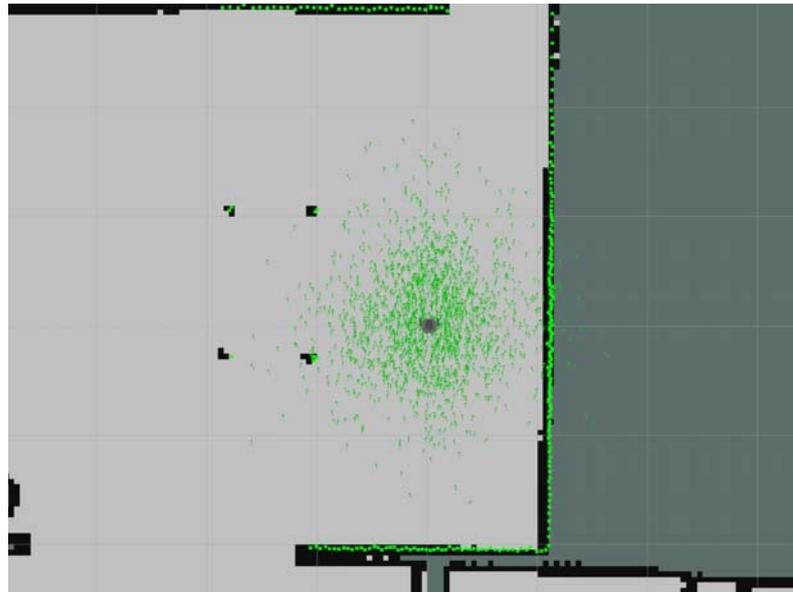
É necessário apontar que na maioria das aplicações do *Navigation Stack* com mapas pré-definidos o AMCL é uma escolha mais recorrente do que o EKF pela sua precisão em apontar a posição absoluta do robô.

2.2.4.3 *Planejamento de Rotas com o Navigation Stack*

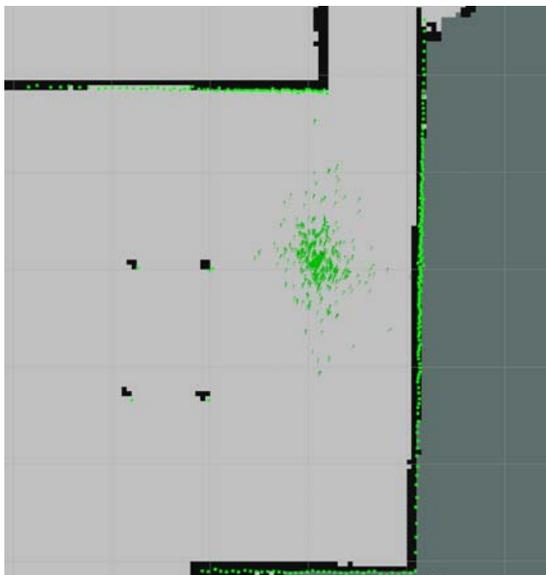
O planejamento de rotas depende tanto do algoritmo de planejamento como do mapa de custos. Como vimos, o mapa de custos é uma representação em duas dimensões em que cada célula tem um valor. Valores acima de um limite definido são tratados como valores de colisão e devem ser evitados ao máximo.

Como visto anteriormente, há dois tipos de algoritmos de planejamento no ROS, o Global e o Local. Cada um dispõe de um mapa de custo para calcular a trajetória e

Figura 11 – Partículas do AMCL



(a) Antes de navegar



(b) Após navegar até o primeiro ponto



(c) Após navegar até o segundo ponto

Fonte: Imagens geradas pelo autor.

diferentes camadas podem ser aplicados a cada um deles independentemente.

Os algoritmos padrão do *Navigation Stack* para o planejador de rotas global são o Dijkstra (Dijkstra et al., 1959) e o A* (Hart; Nilsson; Raphael, 1968) e são tipicamente usados como planejadores de rota.

Entretanto, existe um problema típico relacionado ao custo das células. Se o robô apenas considerar isso ao passar por uma célula pode ser que ele tome um caminho exacerbadamente longo para minimizar o custo, evitando assim qualquer célula com custo maior que zero. Para atacar esse problema, um algoritmo de *wavefront* é usado para adicionar uma constante P para cada célula, criando um gradiente que se estende do ponto inicial do robô ao destino (Lu, 2014).

Assim como Lu (2014), podemos definir $C(p)$ como o custo total de um trajeto p e podemos achar o melhor trajeto minimizando a equação 1 em que $f(x, y)$ é o custo da célula e P é a constante adicionada pelo *wavefront*.

$$\min C(p) = \min \sum_{(x,y) \in p} [f(x, y) + P] \quad (1)$$

Existem também os planejadores locais, que se baseiam em usar o ambiente ao redor do robô, para, com base no planejador global, tomar a decisão do próximo passo a ser tomado. Esses planejadores recebem como entrada o objetivo do planejador global e, por meio de um mapa de custo local, dão como saída para o robô uma mensagem de *Twist* que contém velocidades lineares em x , y e z e angulares em roll, pitch e yaw.

Para os algoritmos de planejadores de rotas locais, dois principais algoritmos são implementados no ROS. O primeiro é o Dynamic Window Approach (DWA), que discretiza o espaço de controle do robô e para cada amostra simula diferentes velocidades em x , y e θ . Após, avalia cada trajetória com métricas como: a proximidade com obstáculos, proximidade com o objetivo final, concordância com o planejador global e velocidade, descartando as ações inválidas. Tendo selecionado a melhor trajetória o DWA repete o processo até encontrar o destino final (Fox; Burgard; Thrun, 1997).

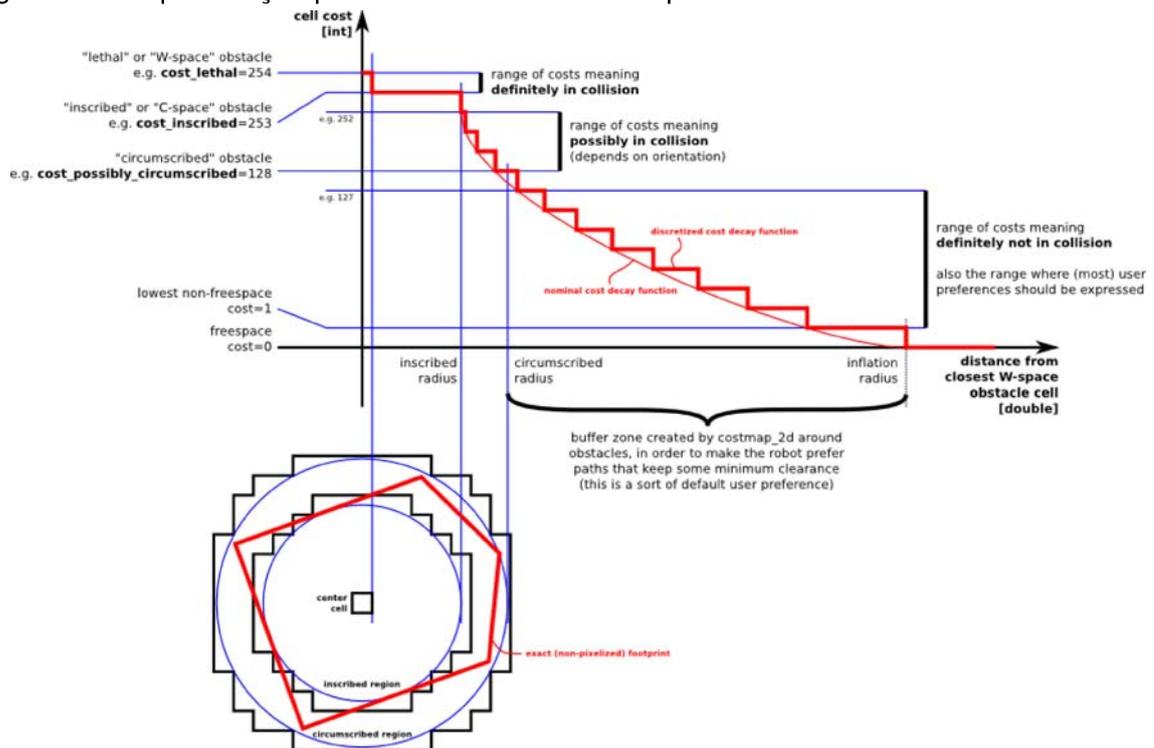
Já o *Time Elastic Band* (TEB), outro algoritmo de planejador local utiliza uma estratégia baseada no conceito de elásticos. Inicialmente, um caminho é recebido, representado por uma “banda elástica” onde cada ponto é tratado como um nó, e as conexões entre esses pontos são elásticos, permitindo estiramento e compressão. A função de custo penaliza proximidade a obstáculos, mudanças bruscas de direção e violações das capacidades de movimento do robô. Esse elástico é iterativamente ajustado para minimizar a função de custo, incluindo otimização temporal para ajustar velocidades e acelerações, esse processo lembra forças que simulam as propriedades elásticas, ajustando as posições dos pontos para um caminho mais otimizado (Rösmann; Feiten; Wösch; Hoffmann; Bertram, 2013).

2.2.4.4 Mapas de Custo

Já sabemos que o *Navigation Stack*, ao gerar um mapa com um método de SLAM, por exemplo, tem como saída um mapa de ocupação 2D baseado em grade, que chamaremos deste ponto em diante de mapa de custo. Um mapa de custo é uma representação 2D de um ambiente dividido em células, onde cada célula armazena o custo agregado em navegar no local representado.

Inicialmente imaginaremos esses valores dos mapas de custo como sendo a ocupação do local no mapa. Existem espaços livres, espaços ocupados e espaços desconhecidos, todos representados nas células. No ROS os valores nos mapas de custo variam de 0 a 255, sendo 0 representando um espaço livre, e 255 representando uma célula onde existe um obstáculo, como pode ser visto na Figura 10, onde os espaços livres são representados pela cor cinza claro e os obstáculos pela cor preto. Os valores intermediários representam, nesse contexto, a possibilidade de estar em contato com um objeto. A representação padrão para interpretação dos custos nas células de um mapa de custo pode ser visto na Figura 12.

Figura 12 – Especificação padrão dos valores dos mapas de custo



Fonte: Imagem obtida em ROS COSTMAP (2018).

Em uma abordagem para um mapa de custo tradicional, todos os valores são guardados em apenas um mapa de custo central. Essa metodologia é conhecida por Mapa de Custo Monolítico. Esse tipo de mapa tem como vantagem a simplicidade, uma vez

que é necessário apenas uma leitura para obter o dado referente a uma célula específica.

Consequentemente, para problemas de navegação simples, em que se deseja sair de um ponto A até um ponto B apenas se preocupando em evitar colisões um Mapa de Custo Monolítico funciona bem. Porém, quando se deseja introduzir informação dinâmica, ou dados do contexto do ambiente, grande parte da informação semântica é perdida.

Um exemplo clássico é com relação à respeitar o espaço pessoal de humanos. Esse tipo de informação, pelo seu significado semântico, não é apenas mais um obstáculo a se desviar. Ora, passar muito perto de uma mesa é algo comum, já no caso de um uma pessoa é socialmente indesejável. Com isso, nem todos os obstáculos podem ser tratados semanticamente de mesma maneira. Existem muitos outros cenários em que escolher apenas o caminho mais curto sem obstáculos é menos preferível do que um mais longo.

Então, para contornar esses problemas Lu; Hershberger; Smart (2014), propõe um modelo de mapas de custo com múltiplas camadas. A grande diferença para o mapa monolítico é a forma como os valores são populados. Portanto, ao invés de guardar diretamente no mapa de custo final, são guardadas diferentes camadas intermediárias, cada uma com uma funcionalidade específica. Esses dados são então populados em uma camada chamada de *master*.

Esse modelo acaba por envolver duas fases diferentes, uma fase de atualização de bordas e uma de atualização de valores. Na fase de atualização de borda, cada camada expõe o quanto do mapa de custo geral ela deseja atualizar. Essas bordas começam vazias e são acumuladas a cada camada, que pode inflar a borda o quanto for necessário.

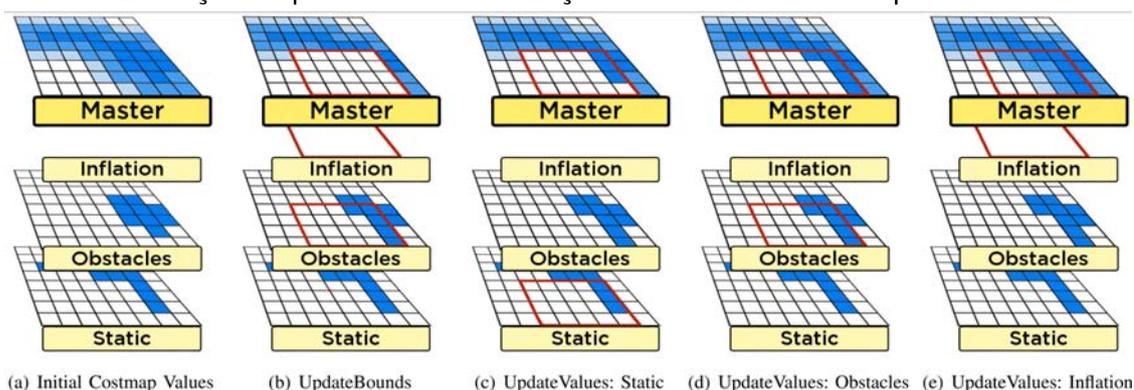
Após isso, na segunda fase, cada camada atualiza os pesos dentro da área da borda final da camada *master*. Cada camada pode guardar os dados que achar necessário e atualizar os valores da última camada de maneiras diferentes. Esse processo pode ser observado na Figura 13.

Dessa maneira podem ser preservadas informações semânticas relevantes à cada camada. No exemplo da Figura 13, a primeira camada, a camada *static*, se refere ao mapa de custo gerado, por exemplo, por um algoritmo de SLAM. É uma recriação do estado das ocupações dos obstáculos no mapa no momento da leitura pelos sensores, portanto o bloco mais básico.

A segunda camada na Figura 13 é a camada *obstacles*. Ela é responsável por identificar obstáculos no ambiente, tanto os já presentes na camada *static*, como novos obstáculos móveis, ou mudanças no mapa estático.

Já a terceira camada, chamada de *inflation*, tem o objetivo de “inflar” os obstáculos presentes tanto na camada *static*, como detectados pela camada *obstacles*. A moti-

Figura 13 – Execução do processo de atualização das camadas dos mapas de custo



Fonte: Lu; Hershberger; Smart (2014).

vação é reduzir a chance de um robô se aproximar de um obstáculo, diminuindo sua chance de colisão.

Esse é o fluxo de criação de mapas básico, presente no ROS. Assim que configurado, o *Navigation Stack* opera com essas camadas para habilitar a navegação no robô. Os arquivos com os parâmetros de configuração podem ser vistos no Apêndice B. Um dos principais parâmetros que podemos encontrar no Apêndice B, tanto no arquivo *local_costmap_params.yaml*, quanto no *global_costmap_params.yaml* é a lista de *plugins*, essa lista define a ordem em que as camadas dos mapas de custo serão processados. Desta forma, a primeira camada da lista é a primeira a ser computada e aplicada, seguindo o processo de execução da Figura 13 e assim por diante.

2.3 Navegação Social

Os seres humanos, ao navegar por ambientes, geralmente seguem um conjunto de normas ou regras sociais, muitas vezes de forma inconsciente. Diariamente indivíduos se movem em multidões sem colidir uns com os outros, ou se agrupam conforme normas como caminhar em um lado preferencial da rua. Portanto, ao contrário de obstáculos quaisquer, um robô não pode simplesmente seguir o caminho mais curto ao se movimentar entre humanos, pois isso pode gerar reações negativas, como raiva, receio e desconforto. Além disso, contextualmente, essa pode não ser a ação adequada. Esse problema é inerentemente interdisciplinar, pois envolve áreas como robótica e Interação Humano-Robô (IHR), e frequentemente abrange também psicologia e sociologia (Mavrogiannis; Baldini; Wang; Zhao; Trautman; Steinfeld; Oh, 2021).

Baseado nisso, a navegação social envolve tomar decisões, durante a navegação, baseado em interação sociais com humanos (Riedl, 2001). Para os robôs, o objetivo da navegação social é garantir a segurança, conforto e movimentações naturais, se integrando a ações sociais e interação proxêmicas (Fernandes; Leite; Díaz-amado;

Cardinale, 2023; Scales; Aycard; Aubergé, 2024).

Basicamente, existem três maneiras de realizar a interação com humanos dentro da área de IHR, interações verbais, não-verbais e espaciais. As interações verbais envolvem a utilização de linguagem falado para expressar comandos ou intenções (Aly; Tapus, 2012). Porém, a dependência em comunicação verbal pode limitar a interação, uma vez que pode não expor todas as nuances de emoção e intenção humana (Mavridis, 2015). Já a interação não-verbal usualmente é atribuída a gestos, expressões faciais ou linguagem corporal para facilitar o engajamento na experiência de interação (Zörner; Arts; Vasiljevic; Srivastava; Schmalzl; Mir; Bhatia; Strahl; Peters; Alpay; Wermter, 2021).

Não obstante, nesse trabalho será abordado apenas a interação espacial. Essa interação envolve o posicionamento físico e movimentos de robôs em relação à humanos e vice-versa, os quais podem afetar a qualidade da interação e conseqüentemente o conforto. Entender o espaço pessoal é um fator indispensável para a navegação social e o projeto de robôs capazes de interagir com humanos (Pimentel; Aquino-jr, 2021; Narayanan; Spalanzani; Luo; Babel, 2016).

Os autores Kruse; Pandey; Alami; Kirsch (2013) apresentam diversos desafios da navegação social, discutindo principalmente o conforto humano, que é essencial para garantir que a interação com o robô seja aceitável. Além do conforto, Kruse; Pandey; Alami; Kirsch (2013) mencionam a naturalidade e a sociabilidade como fatores principais. Entretanto, todos esses aspectos são difíceis de avaliar e implementar em um robô, mas são necessários para que a interação possa se aproximar à aquela realidade por humanos.

Segundo Mavrogiannis; Baldini; Wang; Zhao; Trautman; Steinfeld; Oh (2021), a tarefa de modelar o comportamento dos pedestres é crucial. Para isso, sinais sociais são utilizados para enriquecer os algoritmos com informações sobre possíveis comportamentos dos pedestres, em vez de tratá-los apenas como simples obstáculos. Nesse mesmo estudo, os autores citam três abordagens para modelar regras sociais: espaço proxêmico, intenções e formações e espaços sociais.

De acordo com Charalampous; Kostavelis; Gasteratos (2017), sociabilidade, nesse contexto, se referem principalmente a decisões abstratas na trajetória do robô que englobam noções éticas e culturais dos humanos. Os autores também apontam que o método mais estudado para alcançar uma navegação socialmente aceitável é empregando as teorias do espaço proxêmico, inicialmente propostas por Hall (Hall; Birdwhistell; Bock; Bohannan; Diebold jr; Durbin; Edmonson; Fischer; Hymes; Kimball et al., 1968).

Em um dos trabalhos mais famosos do tópico, Hall; Birdwhistell; Bock; Bohannan; Diebold jr; Durbin; Edmonson; Fischer; Hymes; Kimball et al. (1968) apresentam um estudo detalhado sobre a relação espacial que existe quando as pessoas interagem

entre si. Este estudo mapeou e quantificou essas relações espaciais, denominando-as de proxemics, ou espaço proxêmico.

A Tabela 1 exibe as medidas quantificadas dos espaços proxêmicos definidas por (Hall; Birdwhistell; Bock; Bohannan; Diebold jr; Durbin; Edmonson; Fischer; Hymes; Kimball et al., 1968). Essas medidas são frequentemente utilizadas na formulação de diversos modelos de regras sociais que guiam a interação de robôs com humanos durante a navegação. Para assegurar o conforto do pedestre, recomenda-se que o robô evite invadir as zonas Pessoal ou Íntima, apenas se limitando a navegar nas áreas social ou pública, exceto quando uma interação específica exigir a entrada do robô nessas zonas (Kruse; Pandey; Alami; Kirsch, 2013).

Tabela 1 – Classificação das distâncias definidas pelo espaço proxêmico.

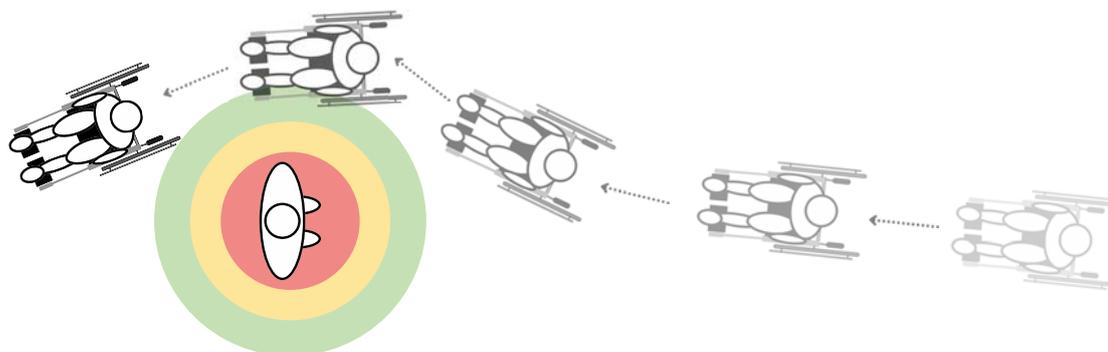
Classificação	Distância
Íntima	0 a 45cm
Pessoal	45 a 120cm
Social	120 a 360cm
Pública	>360cm

Fonte: Métricas observadas por Hall; Birdwhistell; Bock; Bohannan; Diebold jr; Durbin; Edmonson; Fischer; Hymes; Kimball et al. (1968).

No domínio da robótica, proxemics tem um papel crucial nos sistemas de navegação, onde podem ser integrados à navegação clássica para permitir uma navegação que obedece as distâncias sociais (Daza; Barrios-aranibar; Díaz-amado; Cardinale; Vilasboas, 2021). Robôs desenvolvidos para exibir comportamentos socialmente aceitos precisam perceber e interpretar informações de posicionamento dos seus arredores com precisão, particularmente em proximidade com humanos (Sequeira; Gameiro, 2017). Entender o espaço proxêmico é essencial para desenvolver sistemas de navegação conscientes do ser humano, considerando convenções social e definições espaciais.

As zonas proxêmicas ao redor dos humanos são comumente representadas como círculos, o que é apropriado ao considerar o pedestre como um obstáculo estacionário, como exemplificado na Figura 14. No entanto, alguns estudos utilizam zonas proxêmicas em formato elíptico na direção frontal do indivíduo, baseando-se na observação de que os pedestres valorizam mais o espaço pessoal frontal do que o espaço pessoal nas laterais. Além disso, muitos estudos incorporam mudanças dinâmicas no formato dessas regiões dependendo da interação que está ocorrendo, como a expansão de formas elípticas na direção em que o indivíduo está se movendo ou na direção do objeto com o qual está interagindo no momento (Kruse; Pandey; Alami; Kirsch, 2013; Mavrogiannis; Baldini; Wang; Zhao; Trautman; Steinfeld; Oh, 2021).

Figura 14 – Exemplo de navegação social gerada com a presença de proxemics, a CRMA tende a evitar invadir as zonas pessoal ou íntima das pessoas.



Fonte: Autor

2.3.1 Navegação Social no Navigation Stack

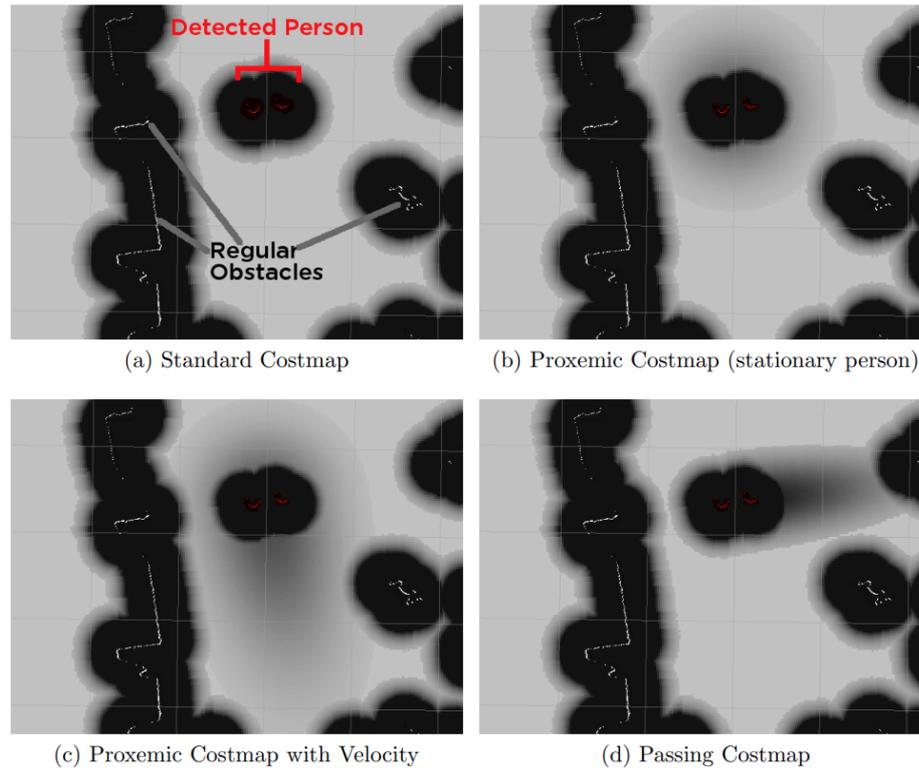
O ROS disponibiliza um pacote com camadas sociais para mapas de custos vistos na Seção 2.2.4.1. Essa camada adiciona uma distribuição Gaussiana ao redor das pessoas nos mapas de custo. O objetivo é criar áreas ao redor das pessoas que sejam custosas o suficiente para que o robô evite determinado trajeto (Kirby; Simmons; Forlizzi, 2009; Lu, 2014).

Em seu trabalho Lu (2014) utilizou uma mistura de gaussianas proposta por Kirby; Simmons; Forlizzi (2009), para criar a camada que adiciona proxemics ao redor dos humanos. Exemplos podem ser observados na Figura 15.

Podemos observar, na Figura 15(a) o comportamento normal dos mapas de custo. A Figura 15(b) representa o humano com os custos do espaço proxêmico inflados em uma distribuição gaussiana ao seu redor. Já a Figura 15(b), mostra o mesmo humano, caso estivesse se deslocando para frente. Nesse caso a mistura de gaussianas proposta por Kirby; Simmons; Forlizzi (2009) faz com que essa área se estenda mais a frente. Por fim, o mesmo conceito, aplicado ao lado do humano, para incentivar o robô à passar ao lado esquerdo da pessoa 15(d) em um corredor, por exemplo.

Adicionalmente, após a obtenção do pacote *social_navigation_layers*, disponibilizado por Lu (2014), para a utilização do mesmo é necessário a inserção dele à lista de mapas de custos ativos. A lista final dos mapas de custos utilizados e sua ordem pode ser encontrado no Apêndice B.

Figura 15 – Adição de proxemics no mapa de custo



Fonte: Lu; Hershberger; Smart (2014).

O conteúdo do Apêndice B explicita a configuração utilizada na camada de proxemics, não alterando os parâmetros pré estabelecidos, com exceção do parâmetro *active_layer* utilizado para determinar quais camadas estão influenciando o resultado final da camada *master*. Os demais possíveis parâmetros são: *cutoff* que determina o limite do menor valor utilizado, *amplitude* que determina a extensão da zona proxêmica no seu pico, *covariance* que afeta a covariância dos ajustes, *factor* que denota uma constante de multiplicação da velocidade em casos em que o humano está se movendo, e *keep_time* o qual remete ao tempo em que a camada mantém os humanos detectados.

3 TRABALHOS RELACIONADOS

Diferente de outros robôs, como os robôs de serviço, as Cadeiras de Rodas Motorizadas Autônomas (CRMA) possuem um fator adicional a ser considerado durante a navegação social. Dado que esse robô está transportando um humano, é crucial que a navegação social leve em conta não apenas o bem-estar e a aceitação dos pedestres, mas também o conforto e a satisfação do usuário da cadeira (passageiro) (Morales; Miyashita; Hagita, 2017). Neste capítulo são expressos os fatores relevantes para a navegação social de CRMAs.

3.1 Conforto do Humano

Conforto, para os humanos, é um estado de bem-estar e ausência de estresse. Medir o conforto não é uma tarefa simples, dado que os humanos percebem mais comumente sua ausência em momentos de desconforto (Morse; Bottorff; Hutchinson, 1994). O conforto humano apresenta uma complexidade multidimensional que depende de fatores internos e externos (Morales; Watanabe; Ferreri; Even; Shinozawa; Hagita, 2018).

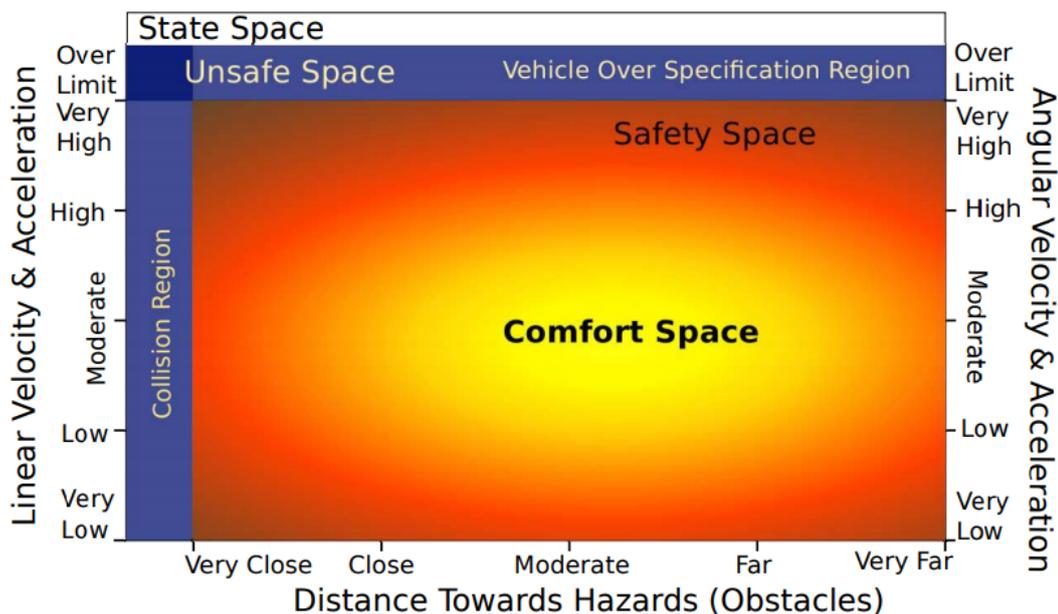
A navegação social em CRMA tem preocupação não apenas com a sensação de conforto das pessoas externas à CRMA, que irão interagir com a rota tomada pela cadeira, mas principalmente com o conforto do usuário que a cadeira transporta.

Uma navegação confortável para o humano, nesse caso, requer que o movimento gerado seja não apenas seguro, mas também que não provoque estresse ou ansiedade no usuário da cadeira (Morales; Kallakuri; Shinozawa; Miyashita; Hagita, 2013).

Uma representação da relação de conforto proposta por Morales; Kallakuri; Shinozawa; Miyashita; Hagita (2013) pode ser observada na Figura 16. O espaço de navegação na figura consiste de três zonas distintas. As zonas não seguras, representadas em azul, são áreas onde pode haver colisão. A zona amarela, denominada zona de conforto, é a região preferida pelo passageiro, onde ele tem maior propensão a sentir-se seguro e confortável. Entre a região azul e a amarela, representada por tons alaranjados, está o espaço de segurança, onde a CRMA está livre de colisões,

mas o usuário pode ou não estar confortável.

Figura 16 – Representação gráfica do espaço de navegação focando no conforto do usuário de uma CRMA proposto por Morales; Kallakuri; Shinozawa; Miyashita; Hagita (2013).



Fonte: Morales; Kallakuri; Shinozawa; Miyashita; Hagita (2013).

As regiões mais escuras dentro do espaço de segurança tendem a representar áreas onde o usuário está menos confortável. O canto superior direito da Figura 16 simboliza o desconforto causado por altas velocidades, enquanto a parte inferior direita representa a ansiedade associada ao movimento em baixa velocidade. De forma semelhante, a parte inferior esquerda indica o desconforto de andar perto de obstáculos, e a parte superior esquerda, o receio de andar perto de obstáculos em alta velocidade.

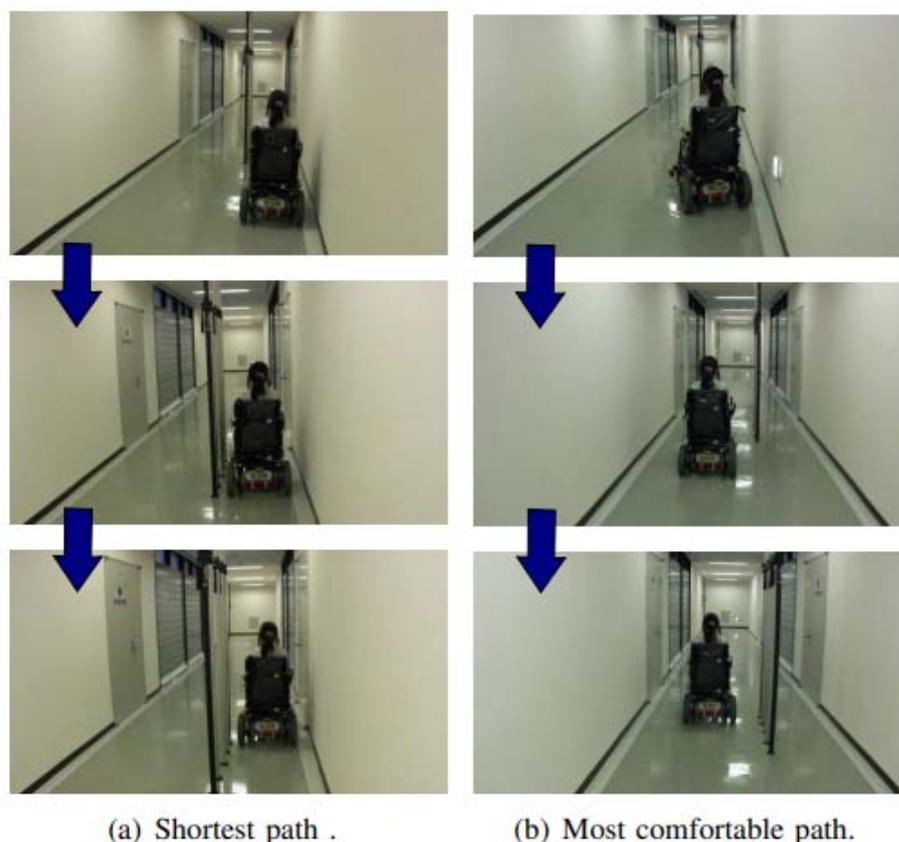
Os autores em Morales; Kallakuri; Shinozawa; Miyashita; Hagita (2013) destacam, através da Figura 16, que uma navegação segura sem conforto é viável, mas o contrário não é. Eles também demonstram que obter conforto na navegação não é simplesmente uma questão de maximizar a distância e a velocidade. Com base nisso, os autores introduziram um mapa de fator de conforto humano, ou *Human-Comfort Factor Map* (HCoM), capaz de calcular as regiões no mapa com maior confortabilidade para o passageiro.

Dois experimentos em um corredor foram realizados pelos autores para identificar parâmetros como distância de obstáculos, velocidade angular e linear, e aceleração angular e linear, em relação à sua confortabilidade (Morales; Kallakuri; Shinozawa; Miyashita; Hagita, 2013).

O algoritmo utiliza um A* como planejador global (Hart; Nilsson; Raphael, 1968), otimizando a rota com base no conforto do usuário. Para testar a viabilidade da na-

vegação, foi realizado um teste onde a CRMA navegou em um corredor com uma divisória ao meio (Figura 17). Ao final do teste, os participantes preencheram um questionário que relacionava a sensação de conforto a cada uma das rotas.

Figura 17 – Teste de navegação social da CRMA realizado por Morales; Kallakuri; Shinozawa; Miyashita; Hagita (2013).



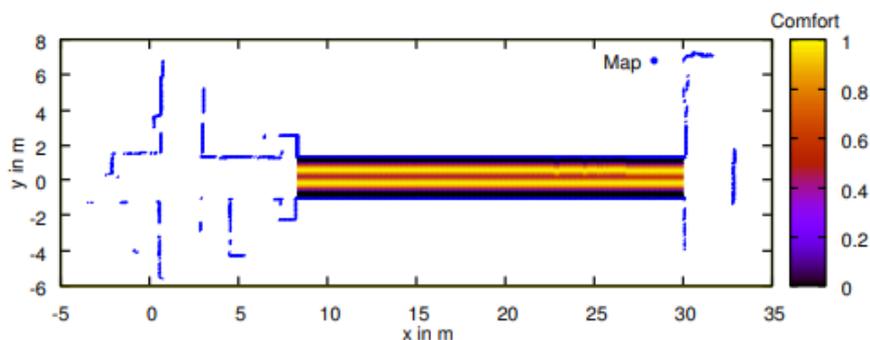
Fonte: Morales; Kallakuri; Shinozawa; Miyashita; Hagita (2013).

O HCoM gerado a partir do mapa do corredor pode ser observado na Figura 18. Quanto mais escura a região no mapa, menor o conforto da referida região. A Figura 18(a) representa o mapa sem nenhum obstáculo, enquanto a Figura 18(b) apresenta o mapa conforme o teste realizado na Figura 17.

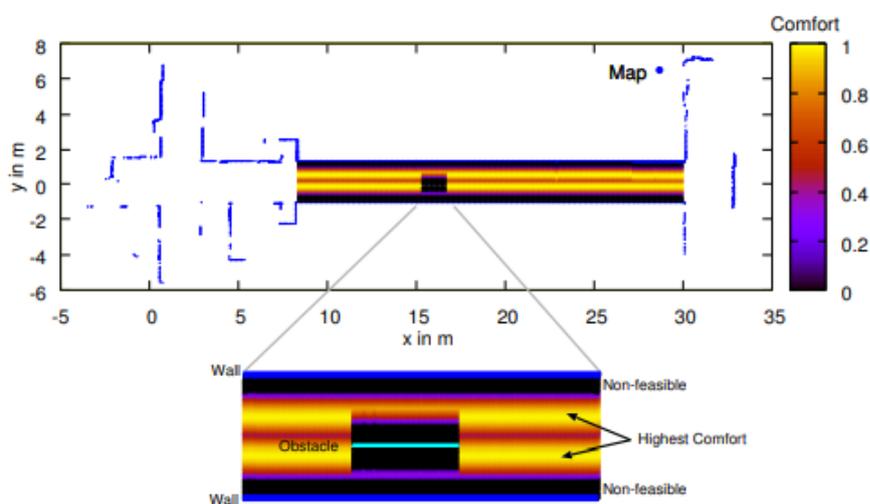
Ao aplicar o algoritmo A* para calcular a trajetória no mapa da Figura 18(b), obtém-se o resultado ilustrado na Figura 19. O caminho em cinza representa a rota otimizada em termos de distância, enquanto o caminho em vermelho representa a rota mais confortável de acordo com o HCoM.

A principal diferença observada é que a rota de menor caminho, calculada pelo algoritmo A* clássico, traça praticamente uma linha reta, enquanto o caminho calculado utilizando o HCoM contorna o obstáculo. Isso evita levar o usuário por uma passagem estreita e possivelmente desconfortável. Como resultado dos testes aplicados

Figura 18 – HCoM gerado a partir do mapa do teste de Morales; Kallakuri; Shinozawa; Miyashita; Hagita (2013).



(a) Comfort map of a straight corridor environment.



(b) Comfort map of a straight corridor environment with an obstacle.

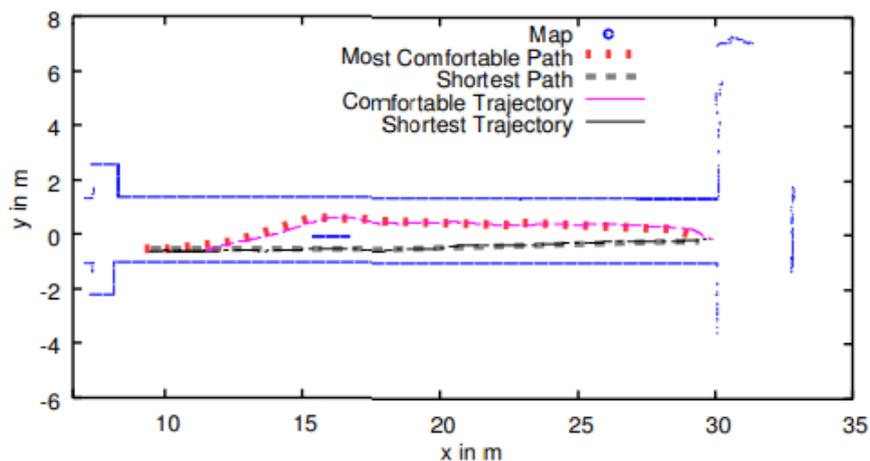
Fonte: Morales; Kallakuri; Shinozawa; Miyashita; Hagita (2013).

aos participantes humanos, Morales; Kallakuri; Shinozawa; Miyashita; Hagita (2013) constataram que, dos 29 participantes, mais de 90% preferiram a abordagem proposta em relação à abordagem de menor caminho.

A efetividade da aplicação de uma CRMA com comportamento social foi avaliada por Shiomi; Iio; Kamei; Sharma; Hagita (2015). O experimento foi realizado em uma casa de repouso para idosos no Japão. Os autores buscaram responder duas perguntas relacionadas ao tema. A primeira questiona a preferência pelo uso da CRMA em comparação ao auxílio do cuidador humano. A segunda aborda o comportamento da cadeira, investigando se ela deveria exibir comportamentos sociais considerados relaxantes e confortáveis pelos idosos durante a navegação.

Essas questões foram baseadas em observações do comportamento social dos cuidadores feitas pelos autores. Os comportamentos foram divididos em duas cate-

Figura 19 – Rota gerada pelo A* para o menor caminho e o caminho mais confortável. (retirada de Morales; Kallakuri; Shinozawa; Miyashita; Hagita (2013))



Fonte: Morales; Kallakuri; Shinozawa; Miyashita; Hagita (2013).

gorias: orientados a indivíduos, que incluem ações pessoais para cada passageiro, como saudá-los pelo nome ou trafegar na velocidade média preferida; e orientados a locais, que envolvem comunicações sobre ações futuras, como anunciar o local de destino.

Três cenários foram aplicados com os participantes: o primeiro com a CRMA sem comportamentos sociais; o segundo com a CRMA exibindo comportamentos sociais; e o terceiro com o cuidador humano, como realizado no dia a dia. As avaliações favoreceram o uso da CRMA com comportamentos sociais.

Os autores também discutem considerações éticas, como a possível dependência do uso da CRMA, a novidade e habituação, já que, na época, o dispositivo era novo e isso poderia impactar as avaliações, levando o usuário a se cansar do uso repetitivo da cadeira. Além disso, foram destacadas limitações do estudo, como a impossibilidade de isolar todos os comportamentos nos testes devido a restrições de tempo, a não avaliação das interações entre a cadeira e os pedestres, e limitações no cenário utilizado (Shiomi; Iio; Kamei; Sharma; Hagita, 2015).

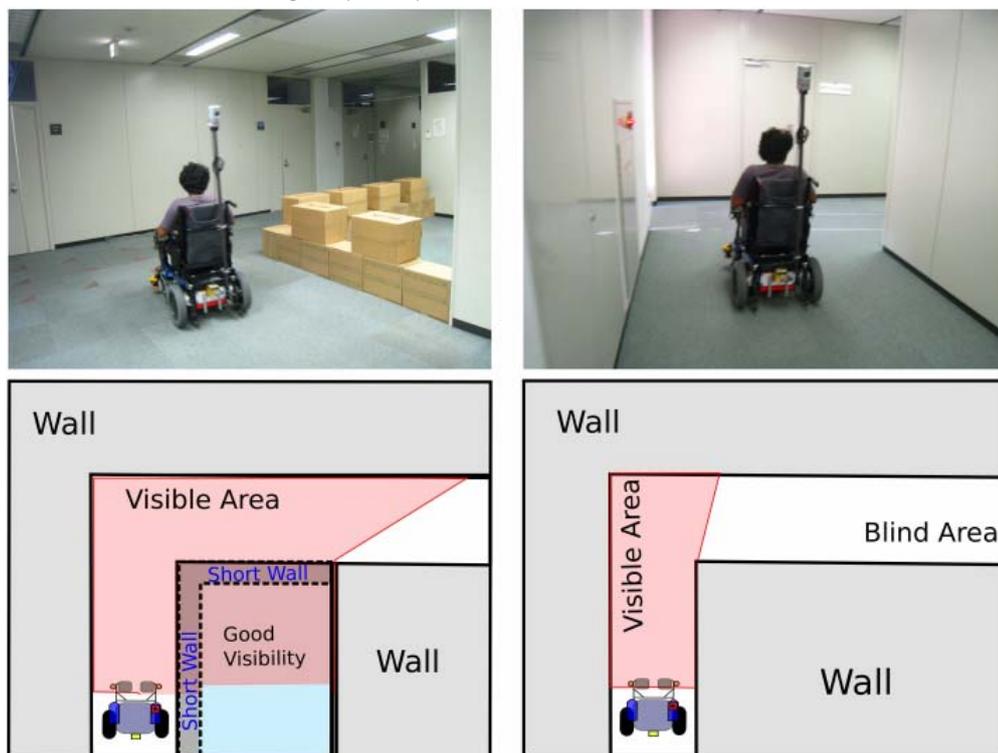
3.2 Visibilidade

De modo geral, quando um humano navega por um cenário, muitos fatores influenciam a sua locomoção e escolhas de rota, sendo o próprio cenário um desses fatores impactantes (Morales; Even; Kallakuri; Ikeda; Shinozawa; Kondo; Hagita, 2014).

Ao se aproximar de uma esquina com baixa visibilidade, as pessoas tendem a reduzir a velocidade e se afastar um pouco do canto para obter uma melhor visibilidade ao fazer a curva (Morales; Even; Kallakuri; Ikeda; Shinozawa; Kondo; Hagita, 2014).

Na Figura 20, Morales; Even; Kallakuri; Ikeda; Shinozawa; Kondo; Hagita (2014) demonstram exemplos de como um humano se comporta ao encarar esquinas com boa visibilidade (esquerda) e com nenhuma visibilidade (direita). Pode-ser observar, que na primeira imagem, a pessoa se desloca mais perto da parede, em comparação com a segunda imagem.

Figura 20 – Exemplo de visibilidade em esquinas demonstrado por Morales; Even; Kallakuri; Ikeda; Shinozawa; Kondo; Hagita (2014)



Fonte: Morales; Even; Kallakuri; Ikeda; Shinozawa; Kondo; Hagita (2014).

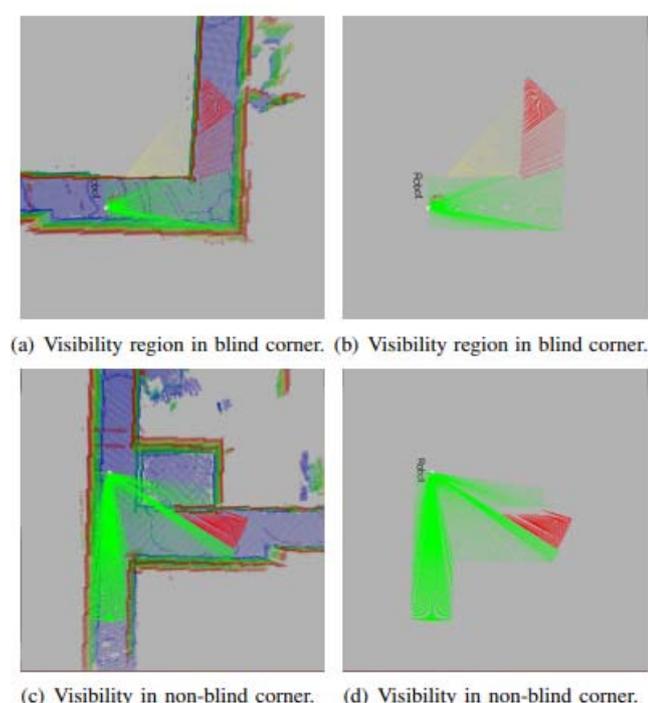
A representação do cenário no trabalho de Morales; Even; Kallakuri; Ikeda; Shinozawa; Kondo; Hagita (2014) é feita em duas camadas. A primeira camada consiste no mapa de ocupação gerado durante o mapeamento do cenário, enquanto a segunda camada contém as áreas atravessáveis. Com essas camadas, é realizada uma análise da razão de visibilidade, cujo resultado é a proporção dos feixes de laser visíveis projetados pela CRMA.

Com base nessa análise, os autores Morales; Even; Kallakuri; Ikeda; Shinozawa; Kondo; Hagita (2014) listaram uma classificação de quatro tipos diferentes de feixes de laser.

- Visível e atravessável, representado pelos feixes verdes em áreas atravessáveis no mapa (podem ser vistos nas Figuras 21(a), 21(b), 21(c) e 21(d));

- Visível mas não atravessável, como os feixes verdes na direita das Figuras 21(c) e 21(d);
- Não visível mas atravessável, feixes vermelhos nas Figuras 21(a), 21(b), 21(c) e 21(d);
- Não visível e não atravessável, feixes vermelho claro, que podem ser vistos nas Figuras 21(a) e 21(b);

Figura 21 – Exemplos de visibilidade no cálculo de razão de visibilidade (exemplo de Morales; Even; Kallakuri; Ikeda; Shinozawa; Kondo; Hagita (2014))

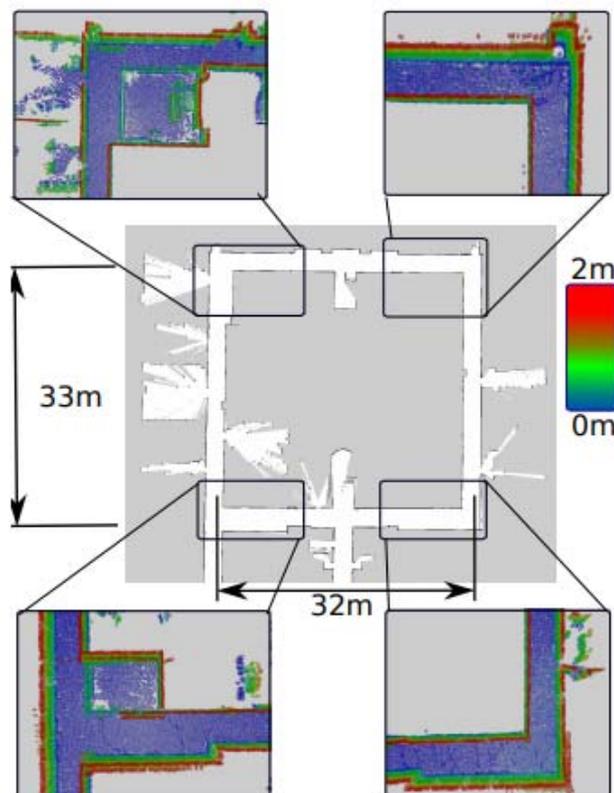


Fonte: Morales; Even; Kallakuri; Ikeda; Shinozawa; Kondo; Hagita (2014).

Os testes em cenário real realizados por Morales; Even; Kallakuri; Ikeda; Shinozawa; Kondo; Hagita (2014) foram conduzidos no ambiente representado na Figura 22, que apresenta esquinas com boa e baixa visibilidade. Inicialmente, os participantes tiveram um tempo para se habituar à cadeira e, posteriormente, foram solicitados a completar quatro voltas no sentido horário e quatro no sentido anti-horário no referido cenário.

Na fase seguinte, a cadeira operou autonomamente pelo cenário, repetindo, sem que os passageiros soubessem, duas voltas no sentido horário e duas no sentido anti-horário que os próprios participantes haviam realizado anteriormente. Além disso, foram repetidas duas voltas realizadas por um piloto experiente em alta velocidade, com a ordem das execuções sendo aleatória.

Figura 22 – Visibilidade no mapa utilizado para os testes do trabalho de Morales; Even; Kallakuri; Ikeda; Shinozawa; Kondo; Hagita (2014))



Fonte: Morales; Even; Kallakuri; Ikeda; Shinozawa; Kondo; Hagita (2014).

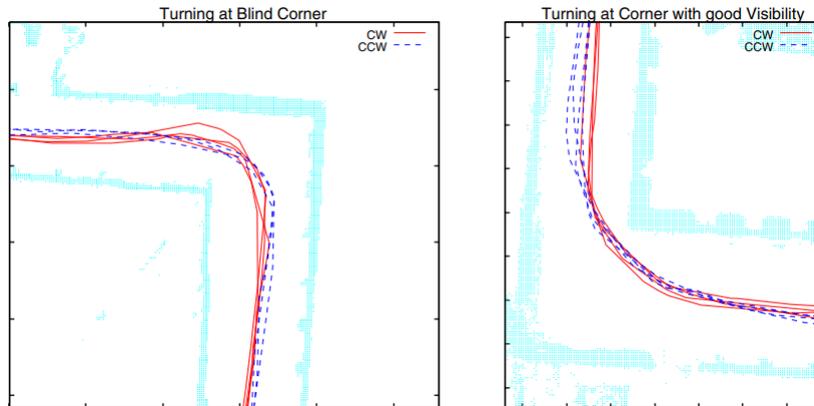
Durante toda a execução, os participantes dispunham de um botão para pressionar sempre que sentissem estresse, ansiedade ou desconforto. Informações como a posição, a visibilidade do local, a aceleração, e as velocidades linear e angular eram registradas sempre que o botão era pressionado.

Os resultados apresentados pelos autores indicam que os participantes se sentem menos confortáveis em regiões com menor visibilidade. Na Figura 23, observa-se que, ao encontrar uma curva com pouca visibilidade (esquerda), o usuário tenta fazer uma curva mais aberta para ganhar mais visão, diferentemente da imagem à direita, onde o usuário tem boa visibilidade e faz a curva de forma mais fechada.

3.3 Combinação de Fatores

Baseando-se nos trabalhos de Morales; Kallakuri; Shinozawa; Miyashita; Hagita (2013) e Morales; Even; Kallakuri; Ikeda; Shinozawa; Kondo; Hagita (2014), os autores, em uma iteração subsequente, através do estudo de Morales; Watanabe; Ferreri; Even; Ikeda; Shinozawa; Miyashita; Hagita (2015), propuseram o *Human-Comfortable Path Planner* (HCoPP).

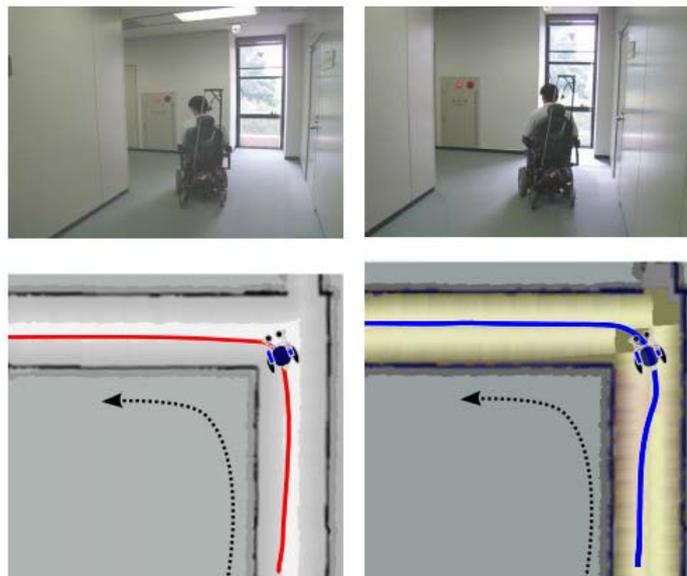
Figura 23 – Trajetória dos participantes durante um dos testes de Morales; Even; Kallakuri; Ikeda; Shinozawa; Kondo; Hagita (2014)



Fonte: Morales; Even; Kallakuri; Ikeda; Shinozawa; Kondo; Hagita (2014).

Um exemplo da aplicação do HCoPP pode ser visto na Figura 24. À esquerda, observa-se um algoritmo tradicional de menor caminho, enquanto à direita, vê-se o resultado do caminho planejado pelo HCoPP, que leva em consideração o conforto do usuário, ambos em uma seção com pouca visibilidade. Conforme ilustrado, o algoritmo de menor caminho toma uma rota mais próxima da curva, enquanto o HCoPP prefere se movimentar de forma a aumentar a visibilidade.

Figura 24 – Exemplo do algoritmo HCoPP proposto por Morales; Watanabe; Ferreri; Even; Ikeda; Shinozawa; Miyashita; Hagita (2015)



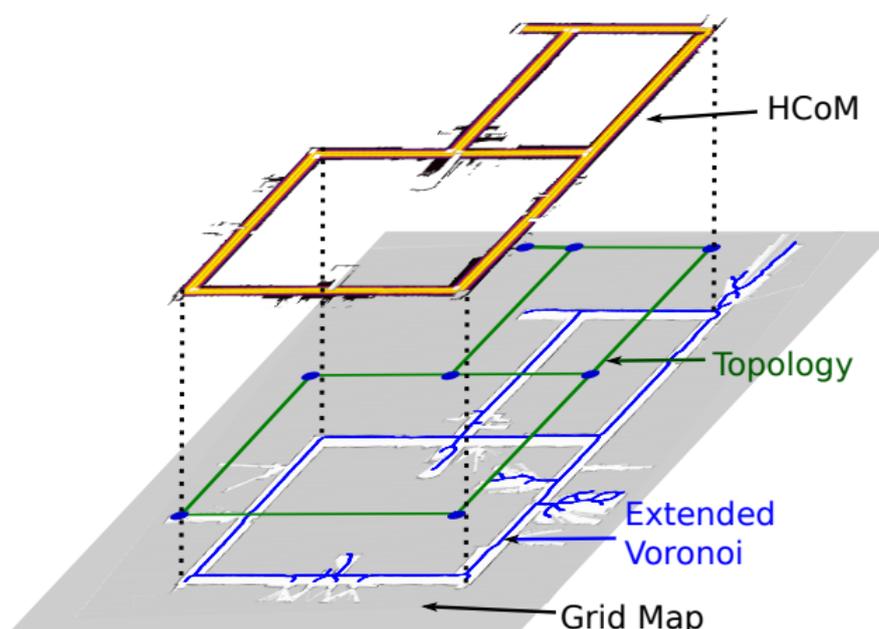
Fonte: Morales; Watanabe; Ferreri; Even; Ikeda; Shinozawa; Miyashita; Hagita (2015).

A primeira parte do HCoPP (Morales; Watanabe; Ferreri; Even; Ikeda; Shinozawa; Miyashita; Hagita, 2015) baseia-se no mapeamento resultante do HCoM (Morales;

Kallakuri; Shinozawa; Miyashita; Hagita, 2013), que visa manter o usuário confortável em corredores, evitando que ele ande próximo a paredes ou obstáculos, enquanto mantém uma posição no corredor que não atrapalhe o tráfego de pedestres no sentido contrário.

A construção do HCoM é realizada a partir da aplicação de um Diagrama de Voronoi Estendido sobre o mapa gerado. Após isso, a topologia dos corredores é obtida e o algoritmo calcula um mapa de conforto sobre essas camadas. A construção do HCoM pode ser observada na Figura 25. A Figura 26 mostra as áreas de maior conforto do usuário no mapa gerado pelo HCoM na Figura 25.

Figura 25 – Fases da construção do mapa de conforto do HCoM

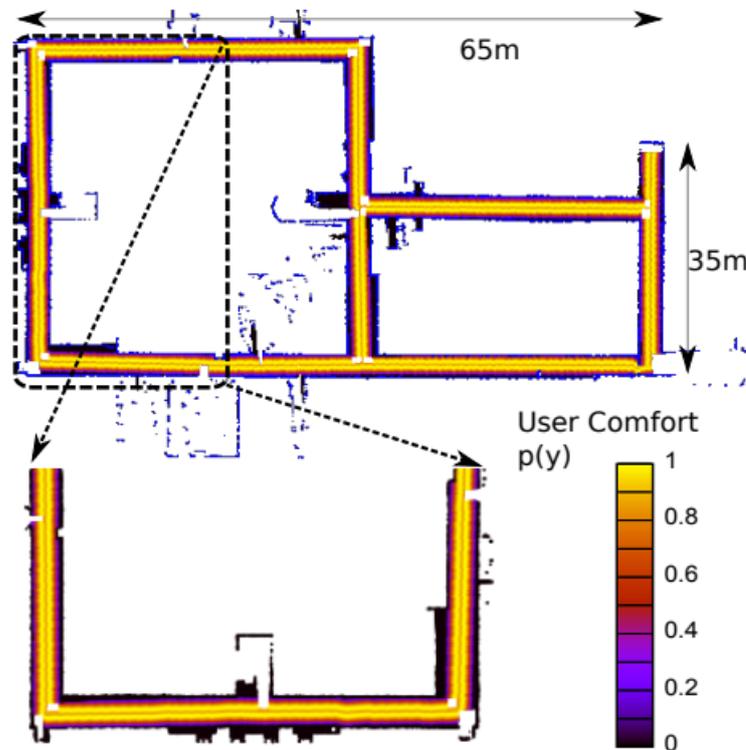


Fonte: Morales; Watanabe; Ferreri; Even; Ikeda; Shinozawa; Miyashita; Hagita (2015).

Posteriormente, o índice de visibilidade, obtido a partir do estudo citado, é integrado ao HCoM. Esse estudo demonstra que indivíduos se sentem menos confortáveis em curvas com menor visibilidade em comparação com curvas de maior visibilidade. No trabalho de Morales; Watanabe; Ferreri; Even; Ikeda; Shinozawa; Miyashita; Hagita (2015), o índice de visibilidade é incorporado como uma camada adicional com o objetivo de fornecer ao planejador de rotas uma noção da visibilidade do usuário durante o percurso, visando maximizar esse fator. A Figura 27 ilustra um exemplo de duas posições distintas da CRMA e seus respectivos índices de visibilidade.

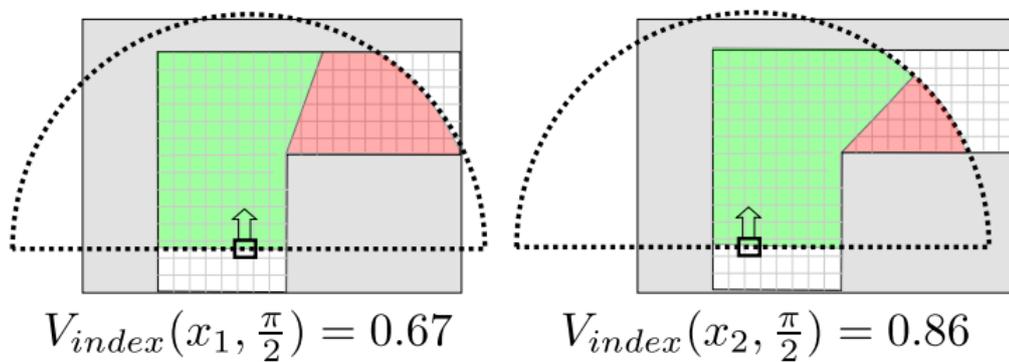
Para integrar o índice de visibilidade no planejador de rotas, os autores calcularam esse índice no mapa de custo. Esse processo envolve a simulação do robô apontando para cada célula vizinha no cenário, com o índice de visibilidade sendo calculado para cada célula. Os valores resultantes para cada posição são então registrados. Um

Figura 26 – Visão superior do mapa gerado pelo HCoM, com as áreas de maior conforto para o usuário



Fonte: Morales; Watanabe; Ferreri; Even; Ikeda; Shinozawa; Miyashita; Hagita (2015).

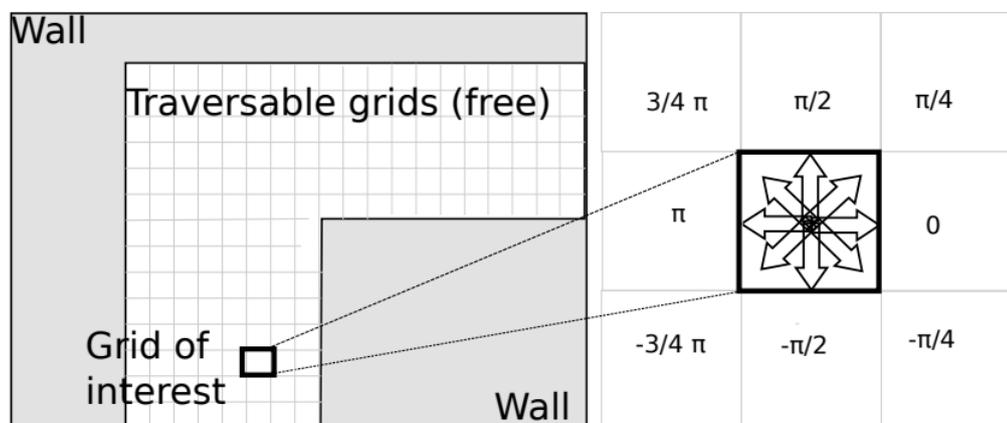
Figura 27 – Dois exemplos do índice de visibilidade



Fonte: Morales; Watanabe; Ferreri; Even; Ikeda; Shinozawa; Miyashita; Hagita (2015).

exemplo dessa abordagem pode ser encontrado na Figura 28.

Figura 28 – Exemplo do cálculo do índice de visibilidade para uma posição no mapa de custo



Fonte: Morales; Watanabe; Ferreri; Even; Ikeda; Shinozawa; Miyashita; Hagita (2015).

Após o cálculo do índice de visibilidade no mapa, obtemos uma representação como a mostrada na Figura 29. Nesta representação, a visibilidade em cada ponto do mapa é calculada como a média da visibilidade daquele ponto nas oito direções indicadas na Figura 28.

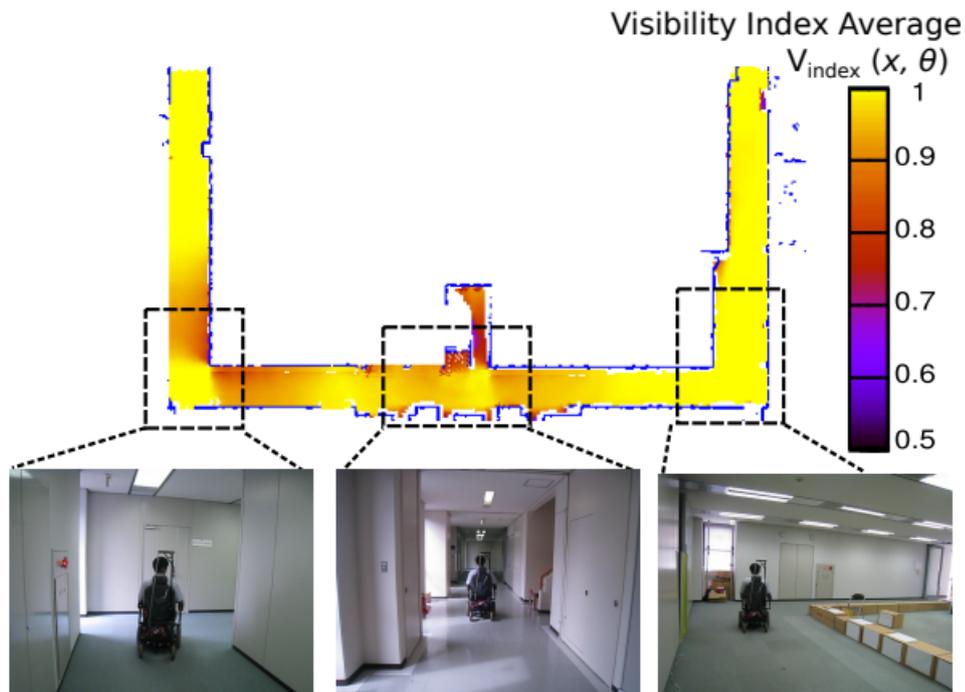
É possível observar na Figura 29 que esquinas geometricamente semelhantes podem apresentar níveis distintos de visibilidade. Por exemplo, a esquina no canto inferior esquerdo e a interseção na parte inferior central têm visibilidade menor em comparação com a esquina no canto inferior direito (Morales; Watanabe; Ferreri; Even; Ikeda; Shinozawa; Miyashita; Hagita, 2015).

O planejador global utiliza informações como distância, conforto e visibilidade para traçar a rota mais adequada. A função do planejador pode ser ajustada para priorizar um dos três parâmetros distintos, como, por exemplo, escolher entre o caminho mais curto ou o mais confortável.

Para avaliar a eficácia do HCoPP, os participantes realizaram duas avaliações como passageiros na CRMA. Uma rodada utilizou o algoritmo de Dijkstra (Dijkstra et al., 1959) como planejador global, enquanto a outra rodada empregou o HCoPP (Morales; Watanabe; Ferreri; Even; Ikeda; Shinozawa; Miyashita; Hagita, 2015). Os parâmetros utilizados no teste pelos autores estão detalhados na Tabela 2.

Os resultados indicam que 80% dos 24 participantes preferiram o HCoPP para navegação, enquanto apenas seis dos 24 participantes consideraram o algoritmo de menor caminho mais confortável. Os autores observam que os parâmetros do HCoPP podem ser ajustados para atender às preferências individuais dos usuários. No entanto, neste teste, todos os participantes utilizaram os mesmos parâmetros fixos.

Figura 29 – Mapa com o cálculo do índice de visibilidade



Fonte: Morales; Watanabe; Ferreri; Even; Ikeda; Shinozawa; Miyashita; Hagita (2015).

3.4 Aprendizado de Normas Sociais

Outra abordagem pela qual uma Cadeira de Rodas Motorizada Autônoma (CRMA) pode aderir às normas sociais, como o distanciamento habitual das paredes ou a execução de manobras durante a navegação, é por meio do Aprendizado de Normas Sociais.

Um exemplo dessa metodologia é apresentado por Johnson; Kuipers (2018), onde os autores desenvolvem um ambiente simulado que permite a um robô aprender comportamentos típicos de humanos. Este ambiente utiliza um mapa topológico que cate-

Tabela 2 – Parâmetros utilizados no teste do HCoPP em ambiente real

Parâmetro	Valor
Peso da distância	0.9
Peso do desconforto	0.019
Peso da falta de visibilidade	1.5
Valor da velocidade linear	0.7 m/s
Valor da aceleração linear	0.4 m/s ²
Valor da velocidade angular	0.6 rad/s
Valor da aceleração angular	1.0 rad/s ²

Fonte: Elaborada por (Morales; Watanabe; Ferreri; Even; Ikeda; Shinozawa; Miyashita; Hagita, 2015).

goriza passagens ou áreas através de *gateways*. Com o suporte desse mapa, a CRMA aprende um conjunto de normas que orientam o movimento dos agentes em relação às paredes, em segmentos retos, e a travessia dos *gateways* no cenário, facilitando a transição entre diferentes áreas ou segmentos retos.

Além disso, o processo de aprendizado pode incorporar as reações dos agentes à presença de outros agentes ao seu redor durante a execução das ações definidas, permitindo não apenas a assimilação de uma regra de distância, mas também o aprendizado de comportamentos condicionados a diversas situações.

Para o treinamento, a CRMA foi exposta a um cenário real, realizando trajetórias em um ambiente pré-mapeado do campus. Neste ambiente, a CRMA interagiu com pedestres que não estavam cientes do teste e continuavam com suas atividades cotidianas normalmente.

Os resultados indicaram que a CRMA foi capaz de internalizar a preferência por manter-se do lado direito do corredor durante a navegação. Quando confrontada com um pedestre vindo na direção oposta à habitual, a CRMA tendia a passar pelo lado esquerdo, conforme esperado. Além disso, observou-se que o comportamento socialmente adaptado da CRMA influenciou os pedestres que passavam, reforçando comportamentos normativos semelhantes.

3.5 Relação Cadeira-Pedestre

Um dos objetivos da navegação social é garantir que um robô, ao interagir com pedestres ou grupos de pedestres, exiba um comportamento semelhante ao esperado de um pedestre, de modo a evitar desconforto.

Estudos indicam que, em um corredor, humanos tendem a se posicionar no centro quando há poucos pedestres. Quando há pessoas se deslocando em direções opostas, os indivíduos tendem a escolher um lado para a locomoção (Bršćić; Zanlungo; Kanda, 2014). Geralmente, o lado escolhido coincide com o fluxo de trânsito (Helbing; Molnár; Farkas; Bolay, 2001).

O estudo de (Morales; Miyashita; Hagita, 2017) amplia o trabalho de (Morales; Kallakuri; Shinozawa; Miyashita; Hagita, 2013), incorporando o conforto nas interações entre a cadeira e os pedestres aos fatores de conforto previamente discutidos entre a cadeira e o passageiro. Observou-se que os passageiros se sentem mais confortáveis quando a cadeira ocupa um lado do corredor, deixando o outro lado livre, mesmo na ausência de pedestres.

Focando em corredores e reconhecendo as irregularidades do mundo real, o estudo utiliza informação topológica para definir o comportamento da CRMA em corredores de larguras variáveis. Inicialmente, os autores extraem um Diagrama de Voronoi Estendido (Beeson; Jong; Kuipers, 2005) e computam nodos e arestas no mapa, onde

o número de nodos é proporcional ao tamanho do robô; robôs maiores geram menos nodos.

Utilizando o Diagrama de Voronoi, (Morales; Miyashita; Hagita, 2017) computam a largura dos corredores para criar um mapa de conforto com base no espaço disponível e na possível direção de tráfego dos pedestres.

Nos testes realizados, os participantes foram submetidos a uma rodada como passageiros e uma rodada como pedestres. Cada rodada envolvia atravessar o percurso uma vez usando o algoritmo de Dijkstra para menor caminho (Dijkstra et al., 1959), e uma vez utilizando o caminho global gerado com o mapa social proposto.

Os resultados do experimento mostraram que os pedestres acharam o caminho gerado utilizando o mapa social mais confortável do que o gerado pelo algoritmo de menor caminho. Além disso, um teste adicional foi realizado em um lar de idosos para avaliar o desempenho do algoritmo em um ambiente real, onde pedestres passam regularmente. Mesmo sem instruções específicas para os pedestres, não foram observados problemas na interação (Morales; Miyashita; Hagita, 2017).

3.6 Pistas Sociais

Uma abordagem para comunicar a intenção da cadeira ao pedestre é através de um método denominado pistas sociais (*social cues*) (Watanabe; Ikeda; Morales; Shinozawa; Miyashita; Hagita, 2015). A navegação entre humanos é uma tarefa complexa (Trautman; Ma; Murray; Krause, 2015), e, para evitar colisões, os humanos frequentemente utilizam pistas sociais, como mudanças no olhar e na maneira de andar (Helbing; Molnár; Farkas; Bolay, 2001).

Para uma cadeira autônoma, é crucial demonstrar a intenção de suas ações para outros humanos a fim de prevenir colisões. No entanto, a aplicação de pistas sociais não é trivial, pois, se mal interpretadas, podem levar a acidentes devido ao erro na interpretação das ações do robô (Takayama; Dooley; Ju, 2011; Lu; Smart, 2013). Além disso, o uso inadequado de pistas sociais pode resultar em constrangimento e desconforto para o passageiro.

O estudo de Watanabe; Ikeda; Morales; Shinozawa; Miyashita; Hagita (2015) utiliza um projetor para emitir pistas sociais quando detecta a presença de pedestres, como ilustrado na Figura 30. De acordo com os autores, trabalhos anteriores também empregaram robôs (sem passageiros) equipados com projetores ou dispositivos emissores de luz para indicar a intenção de ações (Matsumaru, 2006; Matsumaru; Kusada; Iwase, 2006).

Utilizando dados provenientes dos sensores, o sistema proposto por Watanabe; Ikeda; Morales; Shinozawa; Miyashita; Hagita (2015) é capaz de calcular a rota planejada e projetá-la à frente. Para evitar inconsistências na projeção, uma máscara é

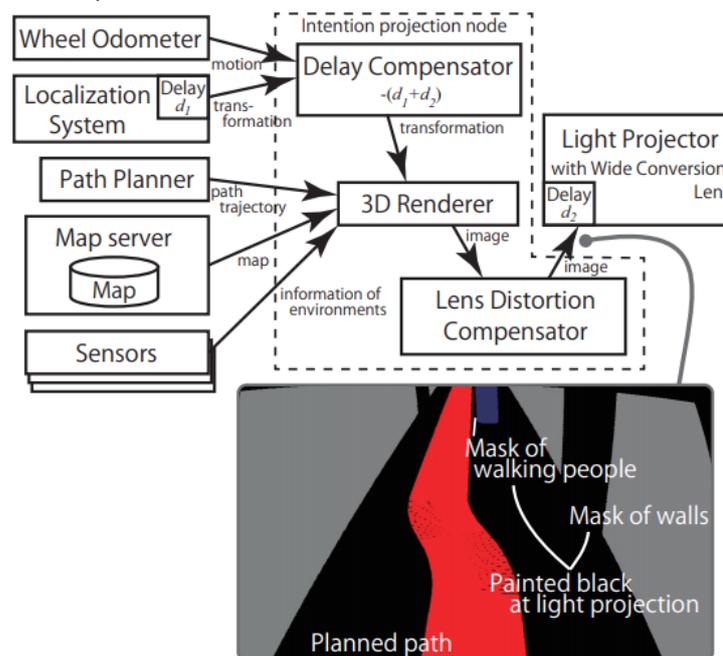
Figura 30 – CRMA com sistema de pistas sociais proposto por Watanabe; Ikeda; Morales; Shinozawa; Miyashita; Hagita (2015)



Fonte: Watanabe; Ikeda; Morales; Shinozawa; Miyashita; Hagita (2015).

empregada para separar a rota das paredes, portas, objetos e pessoas. A arquitetura resultante deste sistema pode ser visualizada na Figura 31.

Figura 31 – Arquitetura proposta por Watanabe; Ikeda; Morales; Shinozawa; Miyashita; Hagita (2015) para geração das pistas sociais.

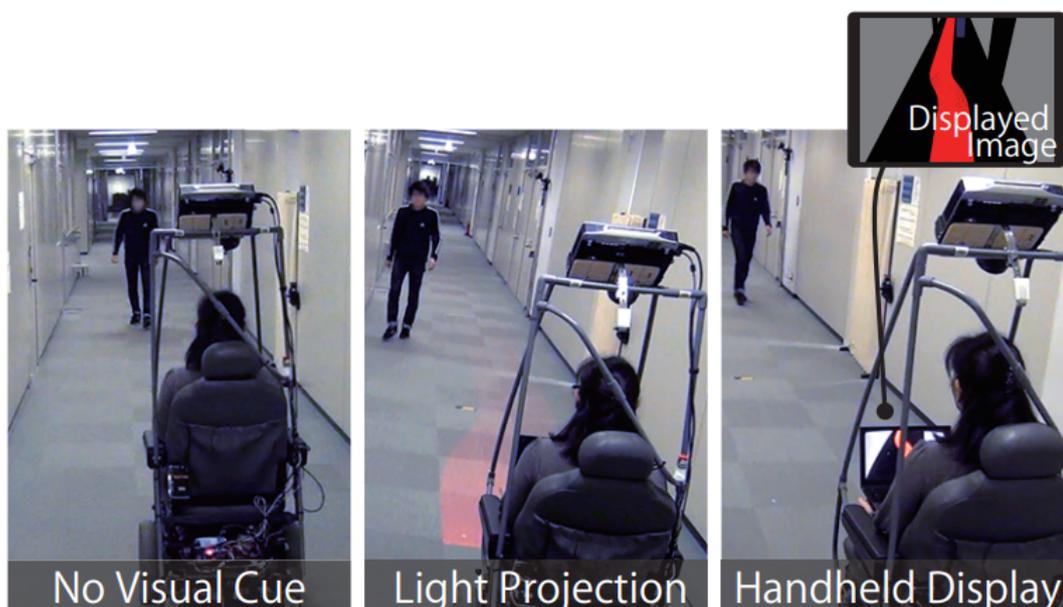


Fonte: Watanabe; Ikeda; Morales; Shinozawa; Miyashita; Hagita (2015).

A avaliação conduzida pelos autores seguiu o princípio de alternar os papéis dos participantes entre passageiros e pedestres. Enquanto atuavam como pedestres, os participantes não tinham conhecimento sobre a utilização de pistas sociais pela CRMA, nem sobre a forma dessas pistas, se seriam projetadas ou exibidas por um display em mãos, conforme ilustrado na Figura 32.

Após cada execução, os participantes preencheram um formulário avaliando diver-

Figura 32 – Teste realizado por Watanabe; Ikeda; Morales; Shinozawa; Miyashita; Hagita (2015) para validação do sistema de pistas sociais.



Fonte: Watanabe; Ikeda; Morales; Shinozawa; Miyashita; Hagita (2015).

Além desses aspectos, como a confortabilidade, a facilidade de compreensão dos movimentos e a impressão de conforto em relação aos outros indivíduos.

Para uma avaliação quantitativa do comportamento dos pedestres, foram instalados quatro LIDARs no corredor, e a posição de cada pedestre foi determinada utilizando um filtro de partículas baseado na altura do torso (Glas; Miyashita; Ishiguro; Hagita, 2009). Informações sobre o cenário estão ilustradas na Figura 32.

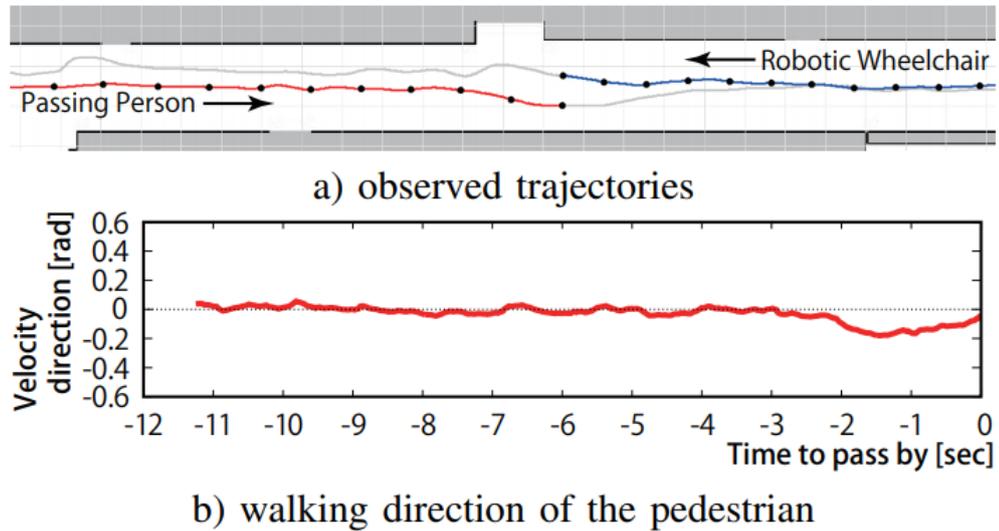
As Figuras 33, 34 e 35 apresentam exemplos dos dados coletados durante os testes realizados pelos autores. A primeira parte da imagem, como exemplificado na Figura 33(a), mostra a trajetória de cada participante. A segunda parte, exemplificada na Figura 33(b), ilustra a direção do percurso do pedestre até o ponto de interseção com a CRMA.

A análise dos dados revela que o percurso assistido pelo projetor, conforme ilustrado na Figura 33(a), resulta em um movimento mais suave tanto para a CRMA quanto para o pedestre, em comparação com os dois exemplos onde o projetor não foi utilizado.

Na trajetória apresentada na Figura 34(a), o pedestre demonstra receio ao caminhar muito próximo à parede, deixando mais espaço para a CRMA. Em contraste, a trajetória na Figura 35(a) mostra uma mudança brusca de direção do pedestre, ocorrendo muito próximo à CRMA.

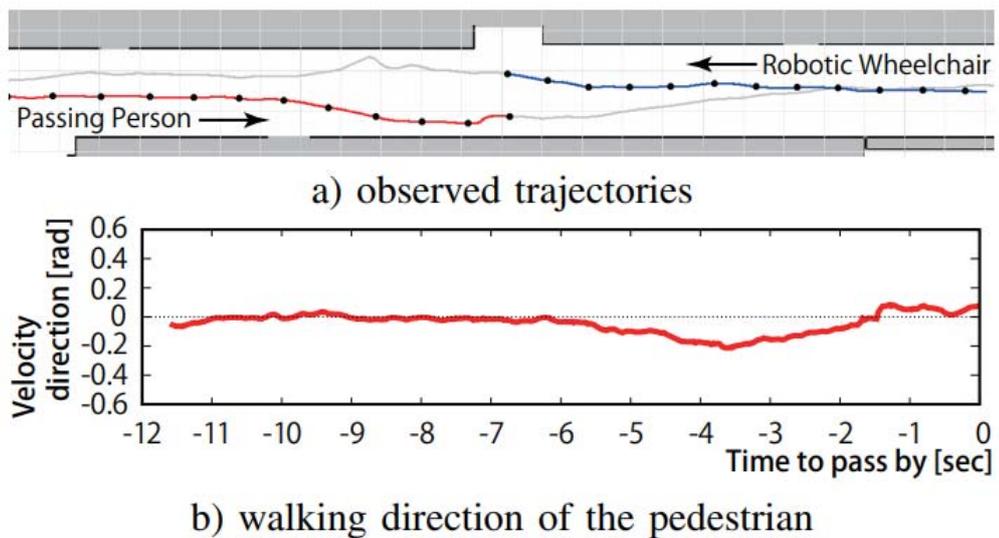
As Figuras 33(b), 34(b) e 35(b) confirmam essas observações. Na Figura 33(b),

Figura 33 – Resultados das trajetórias com pistas sociais geradas pelo projetor (Watanabe; Ikeda; Morales; Shinozawa; Miyashita; Hagita, 2015).



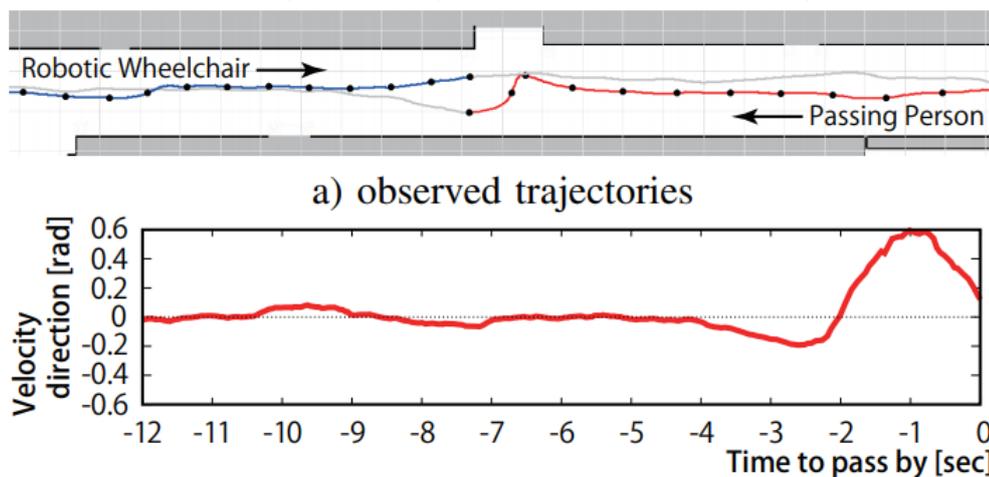
Fonte: Watanabe; Ikeda; Morales; Shinozawa; Miyashita; Hagita (2015).

Figura 34 – Resultados das trajetórias sem pistas sociais geradas pelo projetor (Watanabe; Ikeda; Morales; Shinozawa; Miyashita; Hagita, 2015). Pedestre anda próximo à parede para desviar da CRMA.



Fonte: Watanabe; Ikeda; Morales; Shinozawa; Miyashita; Hagita (2015).

Figura 35 – Resultados das trajetórias sem pistas sociais geradas pelo projetor (Watanabe; Ikeda; Morales; Shinozawa; Miyashita; Hagita, 2015). Pedestre desvia perto da CRMA.



Fonte: Watanabe; Ikeda; Morales; Shinozawa; Miyashita; Hagita (2015).

a direção do pedestre é mais constante à medida que se aproxima da CRMA. No entanto, na Figura 34(b), essa mudança é mais abrupta, e essa abruptude é ainda mais pronunciada na Figura 35(b).

Os resultados finais apresentados por Watanabe; Ikeda; Morales; Shinozawa; Miyashita; Hagita (2015) indicam que, do ponto de vista dos pedestres, ambos os métodos sem a utilização de pistas foram avaliados negativamente. Por outro lado, do ponto de vista dos passageiros, tanto o display manual quanto o projetor receberam avaliações positivas, com relatórios de maior conforto, possivelmente devido à empatia com os pedestres.

3.7 Utilização de Proxemics

O conceito de *proxemics*, introduzido por Hall; Birdwhistell; Bock; Bohannon; Diebold jr; Durbin; Edmonson; Fischer; Hymes; Kimball et al. (1968), define as zonas de espaço que um indivíduo mantém ao seu redor, especificando as distâncias nas quais as interações interpessoais ocorrem.

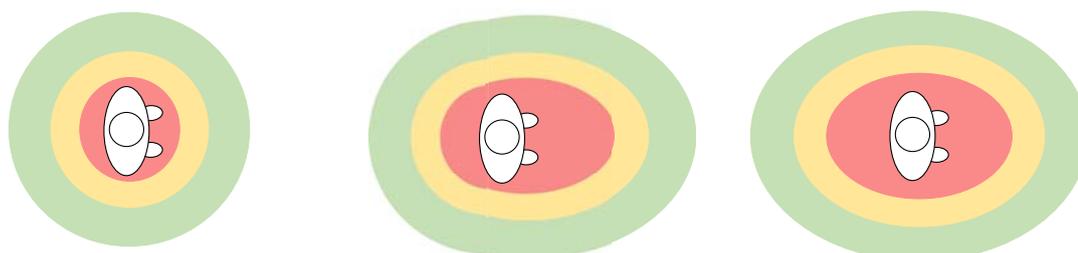
Essas zonas proximicas são classificadas com base na distância física, conforme detalhado na Tabela 1. Normalmente, um robô deve evitar adentrar a zona do espaço pessoal para não causar desconforto ao pedestre. Essas zonas são frequentemente representadas por círculos ao redor do pedestre. Entretanto, alguns trabalhos exploram diferentes formatos para essas representações. Um exemplo de tal representação é apresentado na Figura 36a, onde a zona vermelha representa o espaço íntimo, a zona amarela o espaço pessoal e a zona verde o espaço social.

No estudo de Tomari; Kobayashi; Kuno (2012), que foi expandido em Tomari; Ko-

bayashi; Kuno (2014), os autores aplicam os conceitos de *proxemics* no contexto de navegação de CRMA. Além disso, desenvolvem um módulo para detectar humanos e a orientação de seus rostos. Com base nessas informações, um semi-círculo é projetado na parte posterior do humano para identificar o espaço pessoal que a CRMA deve evitar atravessar. Na parte frontal, é utilizado um semi-elipse que se estende para frente, como ilustrado na Figura 36b. Esse formato assimétrico é adotado porque, do ponto de vista social, as pessoas tendem a proteger mais seu espaço frontal (Amaoka; Laga; Saito; Nakajima, 2009).

Em contraste, o trabalho de Ratsamee; Mae; Ohara; Kojima; Arai (2013) não foca especificamente na navegação de CRMA, mas utiliza um conceito semelhante ao de Tomari; Kobayashi; Kuno (2012), acrescentando uma nova dimensão. Neste trabalho, o robô decide se deve desviar do humano ou interagir com ele com base na duração do contato visual do pedestre. Além disso, a elipse representando o espaço pessoal se estende para ambos os lados, tanto para frente quanto para trás do pedestre, como mostrado na Figura 36c .

Figura 36 – Diferentes métodos para definição de espaços proxêmicos.



(a) Espaço proxêmico proposto por Hall; Birdwhistell; Bock; Bohannon; Diebold jr; Durbin; Edmonson; Fischer; Hymes; Kimball et al. (1968)

(b) Espaço proxêmico utilizado por Tomari; Kobayashi; Kuno (2012)

(c) Espaço proxêmico proposto por Ratsamee; Mae; Ohara; Kojima; Arai (2013)

Fonte: Imagem gerada pelo autor.

3.8 Coordenação Social

Outro método empregado para a interação da CRMA com o pedestre é a coordenação social. Em cenários onde dois agentes competem por espaço físico, como em um corredor estreito, é necessário estabelecer uma prioridade para a passagem.

Similar a alguns trabalhos discutidos na subseção anterior (Tomari; Kobayashi; Kuno, 2012; Ratsamee; Mae; Ohara; Kojima; Arai, 2013), o estudo de Akita; Satake; Shiomi; Imai; Kanda (2018) também utiliza a detecção facial do pedestre. No entanto, os autores empregam esse recurso para identificar possíveis destinos do pedestre e, com base nessa informação, planejar a rota potencial do pedestre. Caso haja uma

interseção entre a rota planejada pelo pedestre e a rota da CRMA, o robô decide se deve ceder espaço para o pedestre ou se deve prosseguir com seu percurso.

Esse processo é baseado em um conceito de prioridade. Por exemplo, se um pedestre se encontrar com uma CRMA transportando um usuário, a prioridade seria da CRMA, por questões de empatia, e a CRMA tomaria a iniciativa de completar seu percurso, permitindo que o pedestre aguarde sua passagem. No entanto, se a CRMA não estiver transportando um usuário, a prioridade seria do pedestre, resultando na inversão dos papéis.

3.9 Seleção dos Fatores Sociais

Para integrar o presente trabalho alguns dos fatores citados neste capítulo foram inicialmente selecionados, uma vez que podem ser representados e disponibilizados como pacotes de mapas de custo no ROS. Além disso, o conjunto dos fatores sociais selecionados deveriam abranger tanto aos componentes relativos aos pedestres quanto aos passageiros da CRMA, contemplando assim ambos os lados das interações abordadas neste trabalho.

Com relação ao pedestre, por já ser um pacote disponível e de fácil acesso, além de ter muito suporte da literatura, serão utilizadas proxemics, em especial o modelo proposto por Lu (2014). Com relação ao passageiro da CRMA serão utilizadas as abordagens apresentadas neste capítulo que demonstram o melhor encapsulamento para uma camada de custo, tanto a camada de conforto como a camada de visibilidade, ambas propostas por Morales; Watanabe; Ferreri; Even; Ikeda; Shinozawa; Miyashita; Hagita (2015), o qual é um dos principais trabalhos na área, foram selecionadas.

O principal critério de seleção das camadas utilizadas foi a viabilidade de inclusão como uma camada independente no *Navigation Stack*, possibilitando fácil distribuição e viabilização com relação aos testes em ambiente simulado. Neste caso, fatores como as pistas sociais não foram selecionados nesta etapa.

Acerca das métricas selecionadas, os trabalhos apresentados utilizavam questionários aplicados aos participantes para avaliar a sensação de conforto proporcionada aos usuários durante os percursos. Sem a possibilidade de aplicar os questionários, devido aos testes feitos em simulação, a avaliação se dá por meio de métricas referentes aos fatores sociais propostos. Com isso, as métricas são compostas pela distância, que é o fator comparativo para com a navegação clássica, a suavidade, que denota o quão natural é a movimentação da cadeira de rodas, também importante quando relacionada ao conforto do usuário durante o percurso, o conforto com relação à distância das paredes, proveniente do trabalho de Morales; Kallakuri; Shinozawa; Miyashita; Hagita (2013), o índice de visibilidade, presente em Morales; Even;

Kallakuri; Ikeda; Shinozawa; Kondo; Hagita (2014), e por fim a proximidade com os pedestres, de acordo com o quanto o robô invade as zonas de proxemics, segundo o modelo de Lu (2014).

4 METODOLOGIA

Após identificarmos os conceitos que constituem uma cadeira de rodas autônoma e também identificarmos os fatores de navegação social que podem compor essa navegação é importante delimitarmos, à seguir, como serão realizados os experimentos para coleta dos resultados. Começando pela implementação das camadas de mapas de custo utilizadas, seguindo pelos componentes que constituem a simulação. Alguns dos elementos que compõe esses experimentos são: o modelo do robô, os cenários de simulação, os parâmetros utilizados e a configuração da máquina em que foram executados. O resultado final está disponível no github¹.

4.1 Proposta

Tendo em vista que a navegação social para CRMAs se dá por conta tanto das interações da cadeira com o passageiro, como da cadeira com os pedestres, ambos os casos precisam ser englobados nos mapas de custo.

A maioria dos trabalhos da literatura buscam isolar apenas um aspecto da navegação social para análise. Em casos em que mais de um dos fatores da navegação social são combinados, com frequência esses métodos compõe o mesmo grupo, portanto, ou ambas se referem à navegação em relação ao passageiro, ou ao pedestre, como no trabalho de Morales; Watanabe; Ferreri; Even; Ikeda; Shinozawa; Miyashita; Hagita (2015). Portanto, a análise da combinação de diferentes fatores é uma lacuna bibliográfica que pode ser explorada.

Para abordar esse problema é proposta a combinação de diferentes métodos para a avaliação da navegação do robô com base em diferentes informações semânticas do cenário.

Desta forma, chegamos no objetivo principal da tese, como citado no Capítulo 1.2. Para atacar o objetivo buscamos avaliar como essas diferentes camadas adicionais interagem entre si, estimando a viabilidade da combinação das camadas e observando quais os padrões que podem surgir das interações cruzadas entre elas, visando am-

¹https://github.com/kriskappel/wheelchair_comfortable_social_nav

pliar o estado-da-arte na área específica.

4.1.1 Camadas Sociais Adicionais

Como visto na Capítulo 3, existem vários fatores que podem compor os elementos sociais envolvidos nas interações de CRMAs. No entanto, existe uma lacuna nos trabalhos que unem tanto a preocupação com o conforto do pedestre e do passageiro ao mesmo tempo. Como pôde ser observado, grande parte do trabalhos que aborda a navegação autônoma para cadeiras de rodas se preocupa com apenas um dos casos citados.

Consequentemente, na busca por um modelo de navegação social para CRMAs é necessário, então, selecionar mapas de custo que traduzam tanto informações semânticas relacionadas aos passageiros, quanto informações semânticas relacionadas aos pedestres.

Em um trabalho anterior descrevemos o processo de seleção e implementação dessas camadas (Kappel; Ferreira, 2023), o qual envolve basicamente aquelas cujas regras sociais melhor se adéquam como camadas de mapas de custo no *Navigation Stack*. Neste estudo os mapas de custo escolhidos foram: primeiramente a camada de conforto, que mapeia a os corredores e espaços internos e gera uma camada que define o valor de cada célula com relação à sua proximidade com as paredes, buscando a posição ótima para navegar nesses ambientes; Além disso mantivemos uma camada de visibilidade, que mapeia o visibilidade ao redor, a partir de cada célula; Por último, para cobrir padrões de interação com pedestre, utilizamos a camada de navegação social por espaço proxêmico disponível no ROS;

As camadas de conforto e visibilidade tiveram que ser reimplementadas, com relação às propostas por Morales; Watanabe; Ferreri; Even; Ikeda; Shinozawa; Miyashita; Hagita (2015). Esse processo se deve à adaptação ao sistema de mapa de custo utilizadas pelo *Navigation Stack* para garantir a possibilidade de futura integração com diferentes camadas, ou em diferentes projetos. Além disso, essas camadas precisaram ser adequar a funcionar de acordo com os recursos de sensoriamento comumente utilizadas por CRMAs, que são LiDARs e odômetros, como foi descrito na Capítulo 2.1.

4.1.1.1 Camada de Conforto

Quando um robô está transportando um humano em um ambiente interno, surgem diferenças no comportamento. Em contraste, um robô social que não está carregando ninguém só precisa considerar o conforto dos pedestres. Isso permite que o robô se locomova próximo às paredes ou objetos, ou até mesmo fique de frente para eles, o que geralmente é desconfortável para os humanos. Portanto, um CRMA precisa ter essa percepção e agir de acordo, maximizando seu próprio conforto para levar em conta o humano que está transportando.

O conforto humano representa um estado de bem-estar e ausência de estresse. Contudo, mensurar o mesmo apresenta desafios, uma vez que as pessoas frequentemente o percebem com mais clareza em situações de desconforto do que em momentos de conforto (Morse; Bottorff; Hutchinson, 1994). Ademais, o conforto é um conceito multidimensional, influenciado por fatores internos e externos (Morales; Watanabe; Ferreri; Even; Shinozawa; Hagita, 2018). No contexto das CRMAs, a navegação social precisa priorizar esses contextos durante o deslocamento. Com isso, uma experiência confortável exige não apenas segurança, mas também a ausência de estresse ou ansiedade para o passageiro (Morales; Kallakuri; Shinozawa; Miyashita; Hagita, 2013).

Nesse sentido, (Morales; Kallakuri; Shinozawa; Miyashita; Hagita, 2013) propõem um modelo que representa a relação entre segurança e conforto na navegação de CRMAs. Os autores demonstram que é possível realizar uma navegação segura sem conforto, mas o conforto não pode ser alcançado sem segurança. Eles introduziram o HCoM, que identifica regiões que proporcionam maior conforto ao passageiro, com base em fatores como a distância de obstáculos e a velocidade (Morales; Kallakuri; Shinozawa; Miyashita; Hagita, 2013).

Dois experimentos realizados em um corredor permitiram aos pesquisadores ajustar parâmetros como distância dos obstáculos, velocidade angular e linear, além da aceleração. Nos testes, a cadeira de rodas navegou em um corredor com um divisor central, e os participantes avaliaram seu conforto em diferentes rotas. Os resultados mostraram que os participantes preferiram amplamente a rota calculada com base no HCoM, que evitava passagens estreitas e desconfortáveis (Morales; Kallakuri; Shinozawa; Miyashita; Hagita, 2013).

Pra a implementação dessa camada primeiramente utilizamos a função de conforto em corredor provida por Morales; Miyashita; Hagita (2017).

A equação 2 representa o conforto $p_{Wc}(y)$ de um robô na posição y do corredor. Essa posição diz respeito à relação da distância do robô para com as paredes laterais, esse valor varia de 0 a 1, sendo 0 mais próximo possível de uma das paredes e 1 mais próximo da outra. Já a função $U(y)$ é exposta na equação 3 e o termo η é uma constante de valor 0,604 utilizada para normalizar o resultado do conforto para valor entre 0 e 1, que posteriormente são transformados para valores de 0 a 252 no mapa de custo.

$$p_{Wc}(y) \propto e^{-U(y)/\eta} \quad (2)$$

$$U(y) = \frac{a}{y} + \frac{a}{L-y} + \left(\frac{y-\delta}{b}\right)^2 \quad \delta = \begin{cases} c & \text{if } y \leq L/2 \\ 1-c & \text{otherwise} \end{cases} \quad (3)$$

As constantes a , b e c são definidas como 0,1; 0,3 e 0,25 respectivamente, esses parâmetros são definidos pelos autores através dos estudos em Morales; Kallakuri; Shinozawa; Miyashita; Hagita (2013). Adicionalmente, c representa a posição de preferência no corredor, onde o conforto é maximizado. Por fim, L é a largura do corredor, também normalizada de 0 a 1. Com isso, essa camada pode ser facilmente ajustada para dar preferência a apenas um lado do corredor, apenas alterando a condição de δ .

Com a função de custo em mãos o próximo passo é definir como aplicar ela no ambiente representado pelo mapa de custo. Em Morales; Watanabe; Ferreri; Even; Ikeda; Shinozawa; Miyashita; Hagita (2015) os autores utilizaram a metodologia mostrada na Figura 25.

Uma desvantagem desse método apresentado na Figura 25 ocorre quando o mapa tem poucos corredores que não se conectam através desses nós. Para resolver isso, na construção da camada os corredores estão sendo detectados de maneira diferente. As representações das paredes podem ser segmentadas do mapa quando o robô ao inicializar o *Navigation Stack*, usando o algoritmo *Hough Lines* (Duda; Hart, 1972). O algoritmo usado para construir a camada de conforto proposta é mostrado no pseudo-código representado pelo Algoritmo 1.

Algorithm 1 Corridor Detection

Require: A matrix M_{ij} and three values: *angleThreshold*, *maxDistanceThreshold*, and *minDistanceThreshold*

Ensure: A list of pairs of lines $L < P_{ij}, P_{kl} >$

```

1: linesList  $\leftarrow$  HoughLines( $M_{ij}$ )
2: linesList  $\leftarrow$  SortLinesByLength( $M_{ij}$ )
3: Create empty list of pairs parallelLines
4: for all lines  $l1$  in linesList do
5:   for all remaining lines  $l2$  in linesList do
6:     angleDiff  $\leftarrow$  AngleDifference( $l1, l2$ )
7:     if angleDiff  $<$  angleThreshold and
       angleDiff  $<$   $-1 * \textit{angleThreshold}$  then
8:       distance  $\leftarrow$  Distance( $l1, l2$ )
9:       if distance  $<$  maxDistanceThreshold
         and distance  $<$  minDistanceThreshold then
10:        parallelLines  $\leftarrow$   $\langle l1, l2 \rangle$ 
11:      end if
12:    end if
13:  end for
14: end for
15: return parallelLines

```

O algoritmo recebe uma matriz M_{ij} , que é o mapa de custos atual, e três parâmetros personalizáveis distintos, um para a diferença máxima de ângulo entre as linhas

paralelas e dois para os limiares mínimo e máximo de distância para que duas linhas diferentes sejam consideradas um corredor. O mesmo algoritmo retorna uma lista de pares de linhas indicando prováveis corredores dentro do mapa. Basicamente, ele funciona da seguinte maneira: o algoritmo HoughLines é aplicado sobre o mapa de custos atual e todas as linhas detectadas são ordenadas pelo comprimento, sendo as linhas mais longas colocadas primeiro. Após ordenar as linhas, o algoritmo itera sobre elas para obter pares de linhas que respeitem os parâmetros de limiar mencionados. Finalmente, o algoritmo retorna os pares de linhas detectados.

Tendo detectado essas áreas agora é necessário aplicar a equação 2 nas células. Para isso podemos utilizar o pseudo-código mostrado no algoritmo 2.

Algorithm 2 Costmap Comfort

Require: A list of pairs of lines $L < P_{ij}, P_{kl} >$

- 1: **pop** *linePair* from $L < P_{ij}, P_{kl} >$
 - 2: $poly\!-\!gon \leftarrow convexFillCells(linePair)$
 - 3: **for all** cells c in *polygon* **do**
 - 4: $position \leftarrow positionInCorridor(c)$
 - 5: $comfort \leftarrow calculateComfortFunction(position)$
 - 6: $costmap_c \leftarrow comfort$
 - 7: **end for**
-

O algoritmo 2 recebe uma lista de pares de linhas, pega cada um desses pares, calcula as células dentro deles. Para cada célula calcula a posição normalizada dentro do corredor, calcula a função de conforto e coloca esse número na célula relativa no mapa de custo.

4.1.1.2 Camada de Visibilidade

A navegação humana em um ambiente é influenciada por diversos fatores, sendo o layout físico do espaço um dos mais impactantes. Na maioria das vezes, os humanos, especialmente ao conduzir uma cadeira de rodas em um ambiente interno, tendem a naturalmente “abrir” mais a curva em uma esquina para ganhar visibilidade antes de realmente virar, particularmente em esquinas com baixa visibilidade (Morales; Even; Kallakuri; Ikeda; Shinozawa; Kondo; Hagita, 2014).

Portanto, o comportamento humano se modifica ao navegar por esquinas com diferentes níveis de visibilidade. No estudo de (Morales; Even; Kallakuri; Ikeda; Shinozawa; Kondo; Hagita, 2014), a análise da “taxa de visibilidade” foi realizada com base nas áreas transitáveis, determinando a proporção de feixes de laser visíveis para a CRMA.

Adicionalmente, (Morales; Even; Kallakuri; Ikeda; Shinozawa; Kondo; Hagita, 2014) realizaram testes em um ambiente real com esquinas de alta e baixa visibilidade. Durante os testes, os participantes podiam apertar um botão sempre que sentissem es-

três, ansiedade ou desconforto. Dados sobre a posição, visibilidade local, aceleração e velocidades linear/angular foram registrados nesses momentos. Os resultados do estudo corroboraram com a hipótese que os participantes se sentiam menos confortáveis em regiões de baixa visibilidade.

Porém, em Morales; Even; Kallakuri; Ikeda; Shinozawa; Kondo; Hagita (2014), os autores adotaram um sistema de visibilidade usando um LiDAR 3D, para mapear o cenário e determinar o fator de visibilidade de cada célula no mapa, calculando a proporção de segmentos de linha visíveis em relação ao número total de segmentos de linha. No entanto, muitas CRMAs na literatura não incorporam um LiDAR 3D, como pode ser visto no Capítulo 2.1. Portanto, o algoritmo foi modificado para calcular a visibilidade em um mapa gerado por um LiDAR 2D, que é uma opção mais acessível.

Semelhante ao citado, mas utilizando o LiDAR 2D, um índice de visibilidade foi calculado para cada célula no cenário. O índice de visibilidade representa a proporção de células vizinhas que são visíveis a partir da célula desejada dentro de um alcance limitado. Como mostrado na Equação 4, o índice de visibilidade é a razão entre células visíveis e atravessáveis $vtc(x)$ e a soma de todas as células atravessáveis $vtc(x) + nvtc(x)$. A equação considera um conjunto de células x dentro de um determinado alcance a partir da célula y .

$$VI(y) = \frac{\sum_{x \in cells} vtc(x)}{\sum_{x \in cells} vtc(x) + \sum_{x \in cells} nvtc(x)} \quad (4)$$

O pseudo-código do Algoritmo 3 estrutura o processo. Deste modo, o algoritmo recebe um alcance máximo da visibilidade, uma parte do mapa de custo e uma lista de células que são as bordas desse mapa de custo. Então, para cada célula é calculado quantas são as células visíveis e não visíveis a partir dessa célula com um alcance definido. Esses valores são aplicados na Equação 4, retornando assim o *visibilityIndex*.

Algorithm 3 Visibility

Require: A value *raytraceRange*, a costmap *window*, a list of cells *borders*

Ensure: A value *visibilityIndex*

- 1: **for all** cells c in *window* **do**
 - 2: $countVisible, countInvisible \leftarrow raytraceLine(c, borders, raytraceRange)$
 - 3: **end for**
 - 4: $visibilityIndex \leftarrow calculateVisibily(countVisible, countInvisible)$
 - 5: **return** *visibilityIndex*
-

Esse resultado da função pode ainda sofrer uma amplificação, com a aplicação de um fator de multiplicação. Esse fator serve para dar mais peso à visibilidade, que

muitas vezes pode ser ignorada quando posta junto com múltiplas camadas, pelo fato de que em alguns cenários mais vazios essa camada pode gerar pesos muito baixos comparados aos pesos de outras camadas.

4.1.1.3 Proxemics

O conceito de espaço proxêmico foi introduzido pela primeira vez por Hall (Hall; Birdwhistell; Bock; Bohannan; Diebold jr; Durbin; Edmonson; Fischer; Hymes; Kimball et al., 1968). Com isso, ele define um conjunto de zonas que os indivíduos têm ao seu redor, indicando as distâncias às quais as pessoas se relacionam. Como resultado, a proxêmica quantifica a distância física nas relações espaciais, como pode ser visto na Tabela 1.

Tipicamente, essas zonas são representadas como círculos ao redor da pessoa com uma distribuição gaussiana. Essa abordagem assume que os humanos são obstáculos estacionários. No entanto, a proxêmica também pode ser representada por formas elípticas, formadas por misturas de gaussianas, na direção de uma área desejada.

Neste trabalho, aplicamos o modelo de proxêmica apresentado por Lu (2014), pois ele fornece uma opção pronta para uso que pode ser facilmente integrada com outras camadas de mapa de custos. Esse pacote, chamado *social_navigation_layers*, pode ser facilmente obtido no ROS. Sua versão com configurações base e zonas proxêmicas com uma distribuição gaussiana em formato circular ao redor dos humanos foi utilizada, como mostrado na Figura 37a.

4.2 Ferramentas de Desenvolvimento

Na Tabela 3 estão detalhadas as configurações básicas do ambiente utilizado para desenvolvimento da maior parte do projeto. Porém, é importante salientar que a proposta é de um sistema que pode ser portado e executado sem adversidades para qualquer ambiente que contém o ROS.

O principal motivo para a utilização destas versões de ROS e Ubuntu se deve ao fato de que cada versão do ROS acompanha e é recomendada para cada novo lançamento das versões do Ubuntu. Como a versão do Ubuntu previamente instalada era o 20.04.6, a versão do ROS recomendada é a Noetic. Uma vez que o ROS Noetic apenas receberá suporte até 2025, algumas mudanças precisam ser feitas nas camadas de mapas de custo para possibilitar a portabilidade para o ROS 2. Atualmente essas mudanças, em resumo, são as seguintes: substituir as bibliotecas do ROS 1 pelas do ROS 2, assim como as chamadas de funções específicas; e mudar a classe base para *nav2_costmap_2d::Layer*. O restante do corpo dos códigos pode permanecer o mesmo.

Tabela 3 – Descrição das Ferramentas de Desenvolvimento

Componente	Descrição
ROS	Noetic
Sistema Operacional	Ubuntu 20.04.6
Gazebo	11.11.0
Memória	48 GiB
Processador	AMD Ryzen 5 5600x
GPU	RTX 2060 12GiB
g++	9.4
Python	3.8.10
RViz	1.14.20

Fonte: Elaborada pelo autor.

O Gazebo é um simulador de robótica em 3D que permite testar e desenvolver algoritmos de robôs em um ambiente virtual realista. Ele oferece uma simulação de física e uma interface robusta para integração com o ROS e outras ferramentas de desenvolvimento.

Como essas simulações contam com muitos nodos, tópicos e serviços funcionando ao mesmo tempo, com diferentes tipos de mensagens sendo trocadas, é necessária uma ferramenta de visualização que organize esses dados para facilitar a visualização. Essa é a função do RViz, que apresenta os dados de uma forma organizada para facilitar o acompanhamento do fluxo de dados.

A memória, processador e GPU são as descrições da máquina em que os experimentos foram realizados. Adicionalmente, as versões do g++ e do Python foram aquelas utilizadas para o desenvolvimento dos algoritmos dos mapas de custo e scripts utilizados para a execução do projeto e coleta de dados.

4.3 Modelo de CRMA Simulado

O ponto inicial é a utilização de um robô simulado que reflita as limitações do robô físico, como descrito no Capítulo 2.1, principalmente com relação aos principais sensores. No trabalho de Afonso; Ferreira (2023), os autores utilizam o modelo da cadeira de rodas disponível em Patil (2018).

O projeto conta com um modelo simulado para execução no Gazebo, bem como os arquivos de Unified Robot Description Format (URDF) que descrevem o robô em formato XML. Por meio do URDF de um robô é possível descrever suas partes, propriedades físicas, juntas, sensores, dentre outros detalhes.

A CRMA na simulação conta com um LiDAR Hokuyo, para obtenção das leituras de scans laser, encoder para realizar a odometria e um sensor Kinect² na parte superior,

²<https://support.xbox.com/pt-BR/help/hardware-network/kinect/kinect-and-privacy>

capaz de captar imagens RGB-D. No entanto, nas simulações finais apenas o LiDAR e os odômetros foram utilizados.

Porém, algumas pequenas alterações foram feitas no modelo inicial da cadeira de rodas. O atrito das rodas traseiras, chamadas de rodízios giratórios, que não são tracionárias, foi levemente reduzido. Isso foi feito pois em algumas manobras a cadeira ficava travada. Geralmente acontecia quando os rodízios giratórios estavam voltados para lados opostos, o que era comum quando o robô girava em yaw.

Não obstante, o ângulo do LiDAR teve de ser levemente corrigido também, para que ficasse mais paralelo ao solo.

Por fim, o footprint do robô foi reduzido, mais perto das medidas reais, sem acrescentarmos desnecessários, para poder proporcionar uma navegação mais precisa ao modelo final. Deste modo, o modelo final, visto no Gazebo, é apresentado na Figura 37.

Figura 37 – Modelo de CRMA desenvolvido por (Patil, 2018) e modificada pelo autor, visualizada no Gazebo



Fonte: Imagem gerada pelo autor.

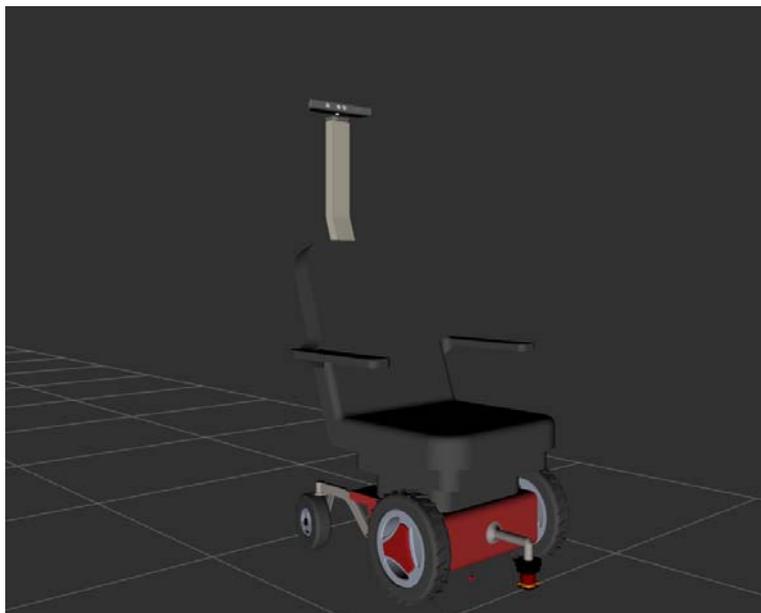
Além disso, podemos visualizar, na Figura 38, o modelo de descrição da CRMA desta vez no RViz.

4.4 Cenários de Simulação

Os cenários utilizados para os experimentos foram desenvolvidos no Gazebo.

O primeiro cenário tem a função de ser um cenário de controle e faz alusão ao cenário adotado por Morales; Watanabe; Ferreri; Even; Ikeda; Shinozawa; Miyashita; Hagita (2015) em seus testes.

Figura 38 – Modelo de CRMA desenvolvido por (Patil, 2018) e modificada pelo autor, visualizada no RViz



Fonte: Imagem gerada pelo autor.

Esse cenário é composto basicamente, por quatro corredores, formando um quadrado. Cada corredor tem três metros de largura e vinte metros de comprimento.

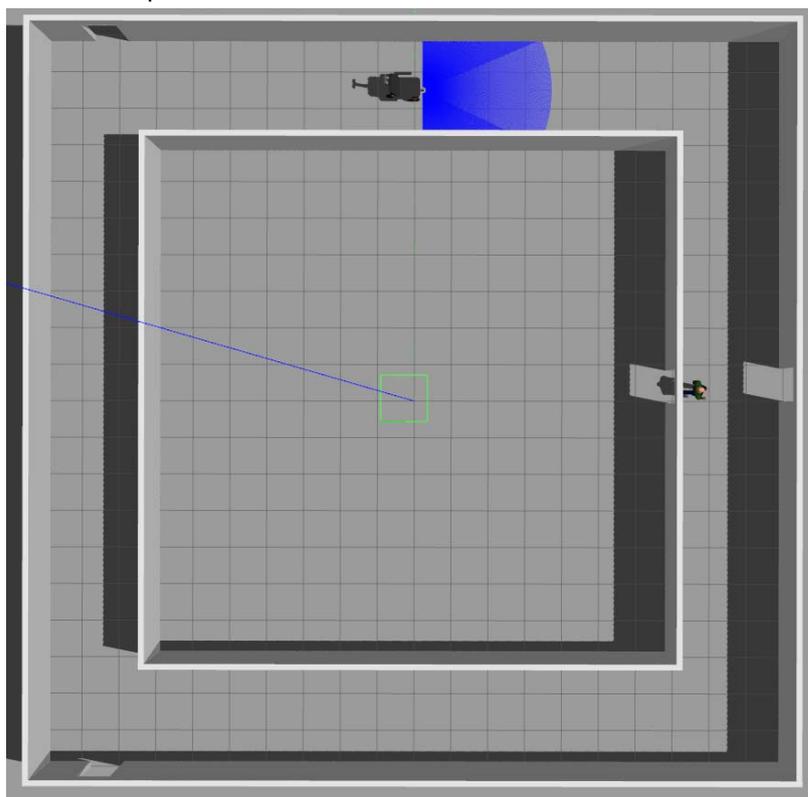
O primeiro cenário, chamado convenientemente de “corredores”, têm quatro portas dispostas uma na parte superior esquerda, uma inferior esquerda, e duas, uma de frente para a outra, no meio do corredor direito. Há também um ator, representando um humano, que está parado na porta interna do corredor direito. Essa configuração pode ser observada na Figura 39. O mapa de custo gerado pelo GMapping é visível na Figura 40. Esse mapa de custo foi tratado em um ferramenta de edição de imagem após mapeamento para eliminar artefatos que poderiam prejudicar o experimento.

O segundo cenário tinha a intenção de refletir um cenário público da vida real. Com isso em mente foi escolhido o terceiro andar do campus Anglo da Universidade Federal de Pelotas (UFPel), que é o andar que abriga os cursos de Computação.

Deste modo, com uma planta baixa do terceiro andar foi possível recriar uma versão aproximada do mesmo. A qual pode ser vista na Figura 41. Como o mapeamento de todo o perímetro seria uma tarefa muito onerosa, uma vez que se tinha a planta baixa, se optou por utilizar um script em Python que transformasse a imagem em uma versão binarizada e construísse o mapa de custos em cima dessa imagem. O resultado pode ser visto na Figura 42.

Infelizmente, devido a uma série de limitações do *Navigation Stack* relativos ao tamanho do mapa e o espaço de busca, se optou por utilizar uma versão reduzida do terceiro andar, mantendo apenas uma metade do mapa, como pode ser notado pela

Figura 39 – Modelo do mapa de corredores visualizado no Gazebo



Fonte: Imagem gerada pelo autor.

Figura 43. O seu mapa de custos gerado pelo mesmo script, o qual binariza a imagem e transforma em um mapa de custos é apresentada na Figura 44.

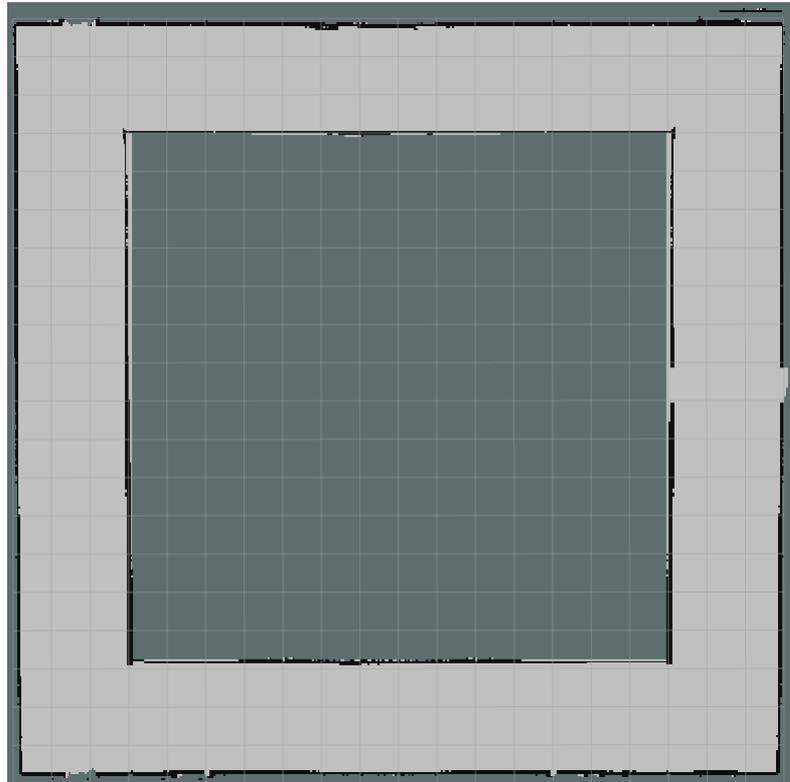
Há também a inclusão de dois atores, que simulam pessoas no cenário, posicionadas ao longo do corredor principal. As posições finais de ambos os atores foram geradas aleatoriamente. Eles podem ser observados em mais detalhe na Figura 45.

4.5 Variáveis de Interesse

Diante do fato de que serão realizados apenas testes em ambiente simulado, uma coleção de métricas teve de ser definida, uma vez que não seria possível realizar questionários em testes reais como é de praxe na literatura. Sendo assim, foram definidas um total de seis métricas numéricas para avaliação das simulações. Os variáveis de interesse foram obtidas por meio da gravação de ROSBags e analisadas posteriormente com Python.

1. Taxa de Conclusão (TC): Quantas das tentativas de navegação obtiveram sucesso;
2. Distância Percorrida (DP): Total de distância, em metros, percorrido pelo robô;

Figura 40 – Modelo do mapa de custo gerado pelo GMapping do mapa de corredores, visualizado no RViz



Fonte: Imagem gerada pelo autor.

3. Coeficiente de Suavidade: Calcula a média das diferenças dos ângulos que compõem as linhas da trajetória (Pimentel; Aquino, 2019). Esse indicador, também conhecido como Smoothing Coefficient (SC), tem o objetivo de avaliar o quão suave é a navegação, uma vez que isso torna a navegação mais confortável (Equação 5). Valores mais próximos de 1 sugerem mais suavidade. No entanto os valores podem se aproximar de 1 pela esquerda ou pela direita. Para facilitar a análise, usando valores mais próximos de 0 como medida de conforto podemos diminuir 1 caso o valor de SC for maior do que 1, ou computar $1 - SC$, caso o valor seja menor;

$$SC = 1 - \frac{\sum_{i=1}^n (\arctan(y_i - y_{i-1}, x_i - x_{i-1}) / \pi)}{n - 1} \quad (5)$$

4. Coeficiente Social (CS): Uma soma dos valores da célula ocupada pelo *base link* do robô na camada de proxemics. Avalia o quanto o robô invadiu o espaço proxêmico dos humanos;
5. Coeficiente de Conforto (CC): Uma soma dos valores da célula ocupada pelo *base link* do robô na camada de conforto. Avalia o quão confortável foi a trajetória;

Figura 41 – Representação do terceiro andar do Campus Anglo no Gazebo



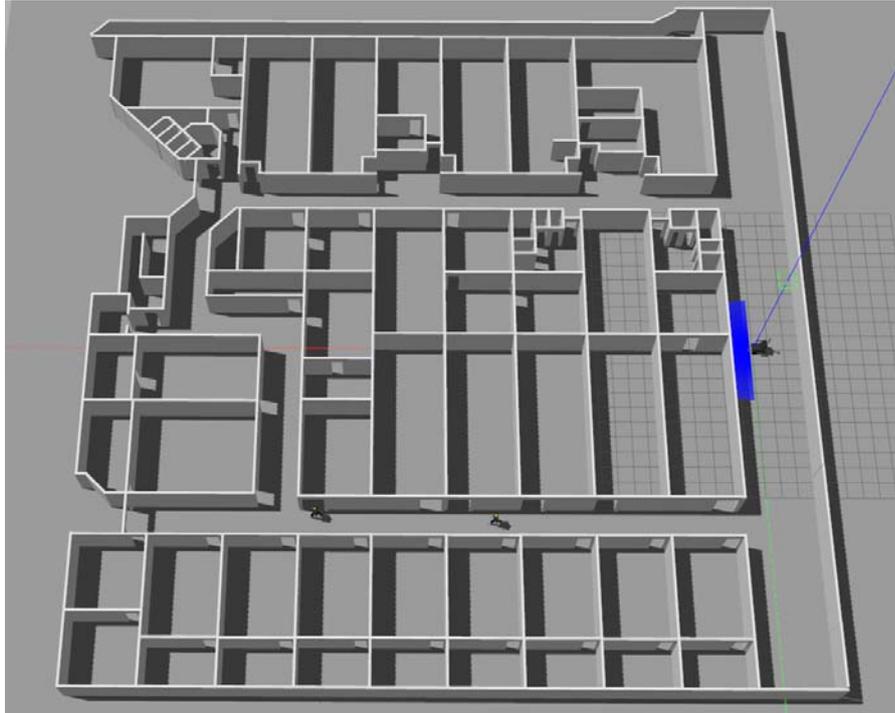
Fonte: Imagem gerada pelo autor.

Figura 42 – Mapa de Custo gerado a partir da planta baixa do terceiro andar



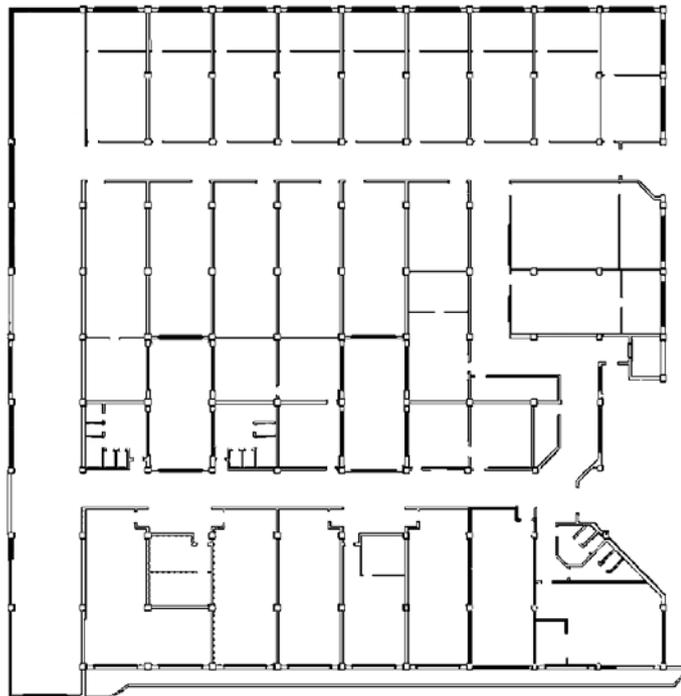
Fonte: Imagem gerada pelo autor.

Figura 43 – Representação de metade do terceiro andar do Campus Anglo no Gazebo



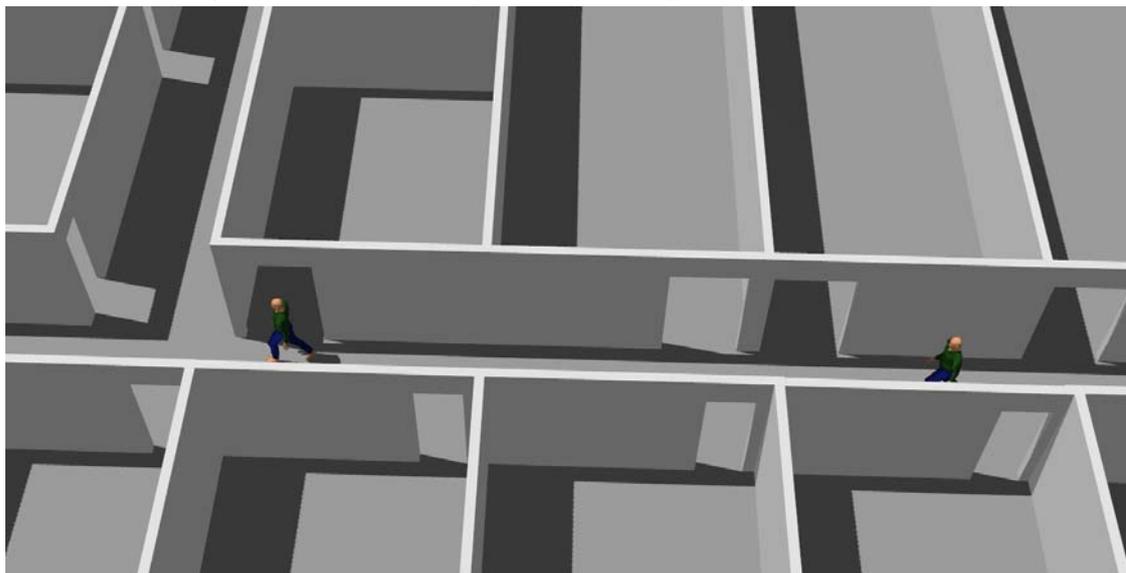
Fonte: Imagem gerada pelo autor.

Figura 44 – Mapa de Custo gerado a partir de metade da planta baixa do terceiro andar



Fonte: Imagem gerada pelo autor.

Figura 45 – Atores posicionados ao longo do corredor principal do terceiro andar



Fonte: Imagem gerada pelo autor.

6. Coeficiente de Visibilidade (CV): Uma soma dos valores da célula ocupada pelo *base link* do robô na camada de visibilidade. Avalia o quão boa foi a visibilidade durante a trajetória;

Para o cálculo da métrica TC foram coletadas as mensagens do tópico */move_base/result*, o qual para cada resultado final possível da navegação registra um número inteiro diferente como código. Além disso, para calcular tanto a DP quanto o SC, foram guardadas as mensagens dos tópicos */amcl_pose* e */odom*, respectivamente, uma vez que ambos tem dados da posição do robô ao longo do tempo.

Além disso, cada uma das métricas restantes (CS, CC e CV) está relacionada a uma camada do mapa de custos. Para esse caso as camadas foram alteradas com a adição de um parâmetro do ROS que permite desativar a camada, mas ainda registrar o custo dela. Com isso, cada uma das camadas ganhou um parâmetro a mais, chamado de *active_layer*. Esse parâmetro faz com que a camada fique ativa e publicando, porém, se o parâmetro for setado para falso, a camada não atualizará os pesos da camada master. Não obstante, não importando o estado do *active_layer* a camada publicará o valor da célula atual ocupada pelo *base link* do robô no seu respectivo tópico *current_value*.

4.6 Procedimentos dos Experimentos

Para compararmos os resultados e o desempenho de utilizarmos, ou não, as diferentes camadas, dois experimentos foram compostos. Cada um deles em cada um

dos cenários apresentados anteriormente.

Em cada um dos experimentos quatro configurações diferentes são testadas:

- Nav - Apenas a navegação padrão do *Navigation Stack*;
- Social Nav - Navegação padrão do *Navigation Stack* com camada de proxemics;
- Confort Nav - Navegação padrão do *Navigation Stack* com camada de visibilidade e camada de conforto;
- Social Confort Nav - Navegação com camada de visibilidade, conforto e proxemics;

A ordem de processamento das camadas das diferentes configurações utilizadas podem ser vistas no Apêndice B, descritas no parâmetro *plugins* de ambos os arquivos. A configuração Nav representa a configuração clássica do *Navigation Stack*, enquanto a Social Confort Nav atua com todas as camadas listadas ativas.

Para a execução automática e sequencial de uma bateria de experimentos foi escrito um script em Python com a API do Roslaunch. Esse script executava o *launcher* da simulação e o *launcher* do *Navigation Stack*, além de uma série de nodos úteis, como o que registrava as ROSBags.

Cada execução das diferentes configurações de camadas de mapas de custo foi executada um total de 50 vezes em cada um dos dois cenários simulados.

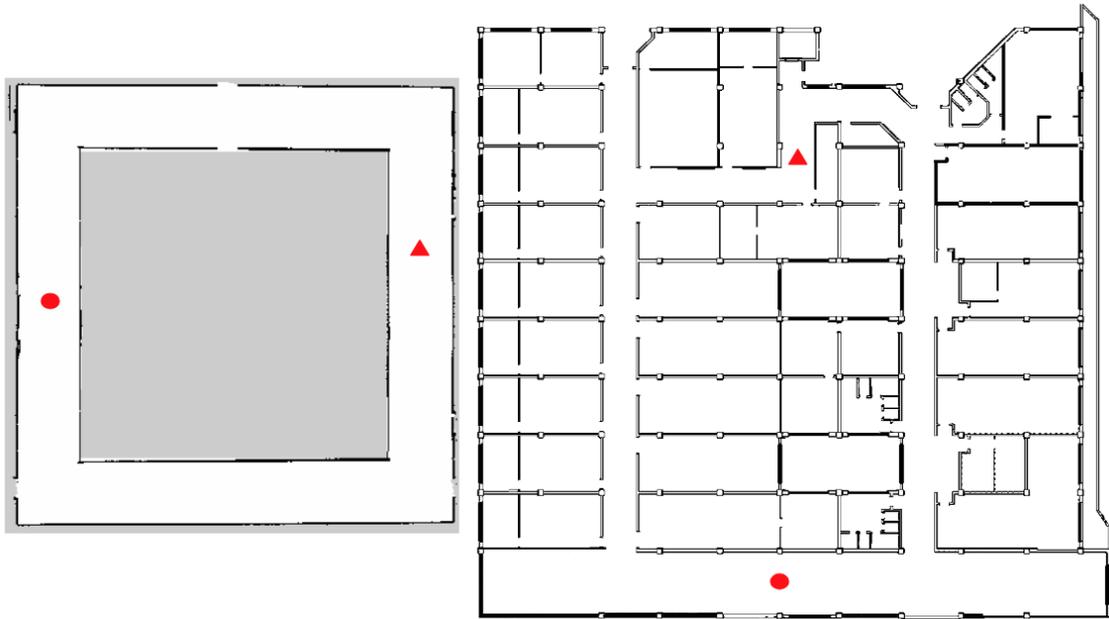
Os experimentos foram executados com as interfaces gráficas do Gazebo e do RViz desabilitadas e grande parte das mensagens de log no terminal suprimida, substituídas por mensagens de andamento do script. Com isso, na máquina descrita na Tabela 3, uma bateria de 50 execuções no mapa de corredores levava por volta de duas horas, enquanto que no mapa do terceiro andar levava em torno de quatro horas.

Em ambos os cenários os robôs tem como objetivo um ponto no mapa que é praticamente o ponto contrário ao de início. Ambos os mapas com os pontos de início da navegação e objetivo podem ser vistos na Figura 46, onde os círculos vermelhos representam os pontos de partida e os triângulos vermelhos os pontos de destino.

Para simular a detecção do ator em tempo real um nodo ativo verifica quando o robô está a uma distância que o laser pode ter contato com o humano e publica a pose dele em um tópico do tipo *Person*. O alcance delimitado para detecção do ator é de 4 metros, este que é o valor aproximado referente ao raio da maior distância atingida pelo tamanho utilizado para definição do mapa de custo local (os quais podem ser encontrados no Apêndice B).

Ao final de cada unidade de execução, o script da simulação executava o nodo *hector_trajectory_server*, que exportava uma imagem da trajetória realizada pelo robô

Figura 46 – Posições iniciais (círculo vermelho) e finais (triângulo vermelho) dos trajetos em ambos os mapas



Fonte: Imagem gerada pelo autor.

em cima do mapa estático. Posteriormente, foi desenvolvido um script para extrair e sobrepor as linhas da imagem que representam o caminho executado

5 RESULTADOS E DISCUSSÃO

No decorrer deste capítulo serão apresentados os resultados coletados a partir dos experimentos realizados, utilizando os arquivos de parametrização do Apêndice B, variando apenas o número de camadas dos mapas de custo ativas a cada cenário de testes. Esses resultados são analisados e uma discussão em torno dos pontos de interesse é levantada a seguir.

5.1 Análise dos Resultados e Discussão

O primeiro teste foi relacionado a viabilidade da junção de camadas, como uma prova de conceito (Kappel; Ferreira, 2023). O principal objetivo era testar, de modo prático, se a integração das camadas era possível e expressaria o efeito desejado. Um exemplo da diferença visual das configurações de camadas interagindo com um ambiente em que se precisa passar por um humano pode ser observada na Figura 47.

Na Figura 47a, o robô faz seu trajeto padrão, evitando as paredes por conta da camada Inflation. Já na Figura 47b, o robô evita o humano não invadindo seu espaço proxêmico. No entanto, na Figura 47c o robô permanece em um dos lados ao longo do corredor. Ao se aproximar do final do corredor, o robô sai do percurso. Esse comportamento é atribuído à Camada de Visibilidade, que guia o robô para uma região com melhor visibilidade, tornando a curva menos desconfortável para o passageiro.

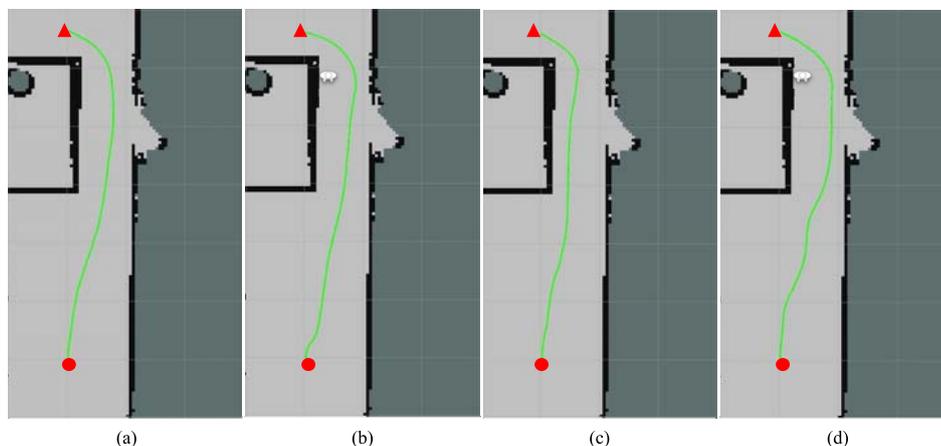
Além disso, na Figura 47d, a adição de um humano no final do corredor tem um efeito distinto na camada de conforto. Que troca o lado do corredor, respeitando as áreas das proxemics.

Após esse trabalho inicial, as camadas passaram por melhorias extensas na performance, aplicação e correções de bugs. Com isso, nesse trabalho poderemos ter uma avaliação mais minuciosa, com métricas mais concisas, sobre as características da combinação das camadas.

Com isso em mente, duas principais hipóteses conduziram esta análise.

- **H1:** A falta de uma camada que está relacionada com uma interação da cadeira com o pedestre afetará negativamente o resultado de uma camada que está

Figura 47 – Comparação entre diferentes conjuntos de camadas de mapas de custo; (a) Configuração padrão do *Navigation Stack*; (b) Configuração padrão com a adição de um ator e camada de proxemics; (c) Inclusão de camadas de conforto e visibilidade, mas sem o ator e a camada de proxemics; (d) Todas as camadas propostas;



Fonte: Imagem gerada pelo autor.

relacionada com uma interação da cadeira com o passageiro (ou vice-versa);

- **H2:** A inclusão das camadas de conforto irá trazer resultados positivos em relação a métrica de suavidade, quando comparada com os resultados da navegação tradicional;

Para facilitar a visualização e a comparação entre as metodologias apresentaremos as métricas como gráficos de caixa. Com relação à Taxa de Conclusão, todos os experimentos encerraram com conclusão na navegação, por isso os resultados dessa métrica não precisarão ser apresentados.

Todos os dados apresentados nas seções a seguir estão disponíveis para acesso no github¹.

5.2 Primeiro Experimento - Cenário de Corredores

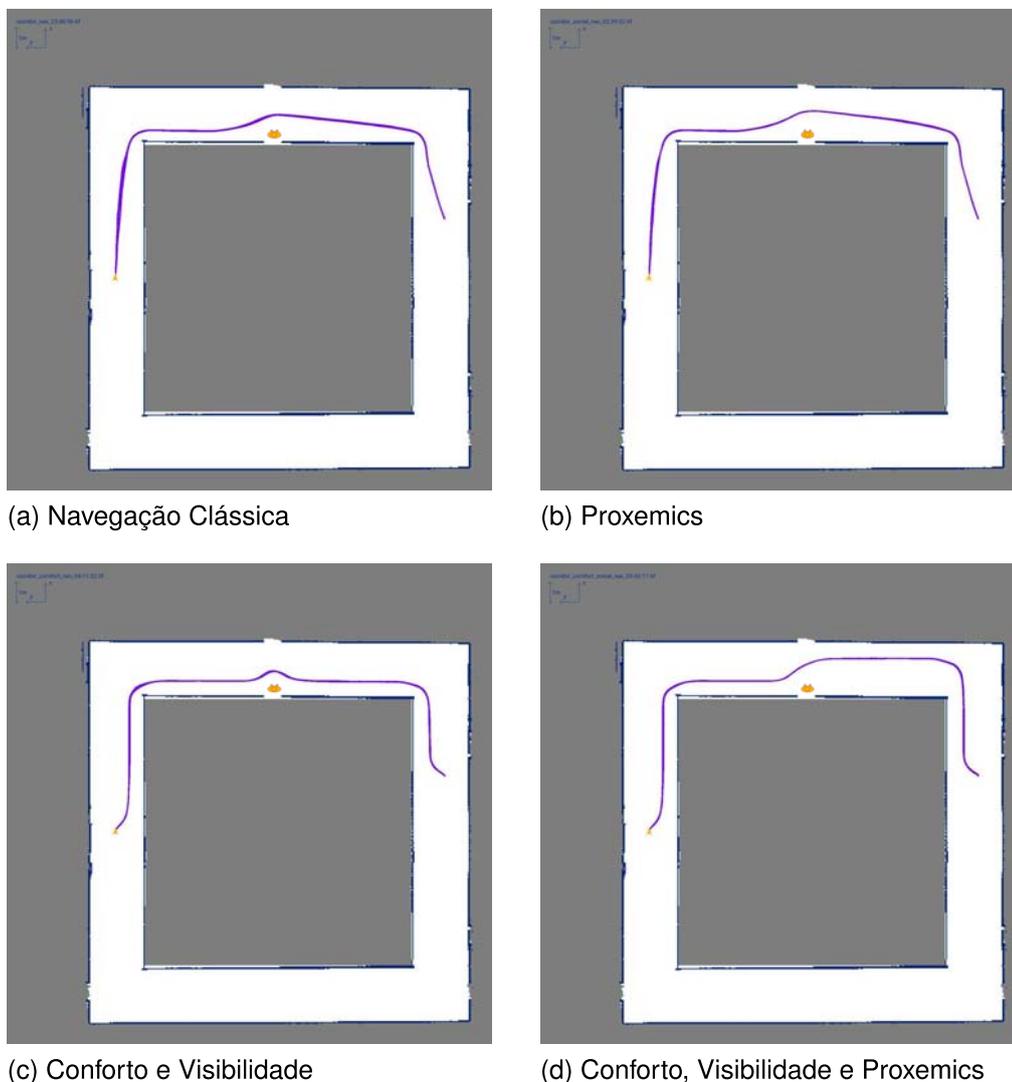
Esse cenário foi pensado como um meio de comparação por evidenciar as diferenças entre as configurações de camadas, além de ser baseado no ambiente de testes de (Morales; Watanabe; Ferreri; Even; Ikeda; Shinozawa; Miyashita; Hagita, 2015).

Para facilitar os debates e melhorar a avaliação, apresentaremos, antes das métricas, a visualização da sobreposição dos caminhos realizados pelas diferentes configurações de mapas de custo na Figura 48. É importante ressaltar que o ponto em que

¹https://github.com/kriskappel/wheelchair_comfortable_social_nav

as trajetórias fazem uma curva, no meio do segundo corredor, é o ponto onde o ator se encontra, sua posição aproximada está representada pelo símbolo em laranja.

Figura 48 – Navegação em diferentes configurações de mapas de custo no cenário de corredores.



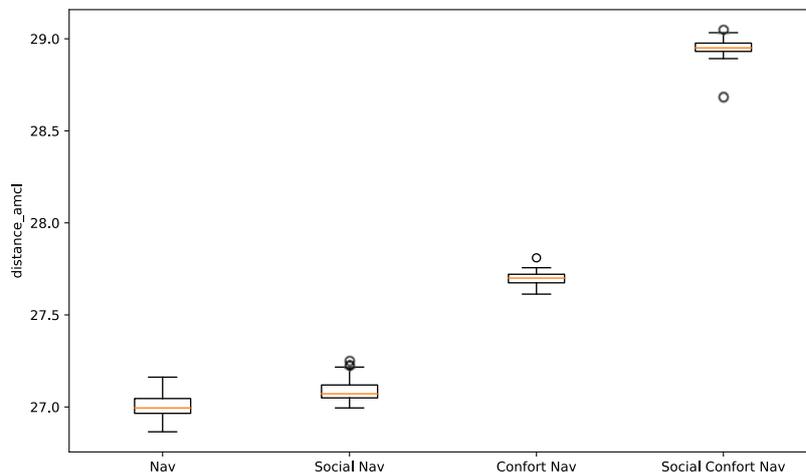
Fonte: Imagem gerada pelo autor.

Conforme a Figura 48, visualmente podemos constatar que a navegação clássica, tende a ter um comportamento mais diagonal em relação ao caminho tomado nos corredores, isso se dá, em grande parte, pelo fato de que essa navegação naturalmente busca o caminho mais curto e mais eficiente, como era esperado. A única diferença quando se é acrescentado as proxemics é um ângulo maior, evitando o espaço proxêmico do ator. Com relação à Figura 48c podemos observar que há uma aderência maior ao conceito de corredores anteriormente citado, sem as proxemics, no entanto, o desvio do ator é menos agudo e logo volta ao seu lado do corredor. Por fim, com todas as camadas a CRMA, ao fazer uma curva mais aguda com relação ao ator acaba

se aderindo à outra via do corredor. Esse comportamento, se desejado, pode ser alterado dando prioridade a uma via na função de custo.

A primeira métrica avaliada será a distância percorrida, com base nos dados gerados pelo AMCL (Figura 49). Como é de se esperar, a navegação comum apresentou vantagem sobre aquela com os mapas de conforto. Isso é o esperado, com relação à distância, uma vez que buscar o caminho mais eficiente é o ponto principal da navegação clássica. O motivo da discrepância fica evidente quando comparamos os caminhos gerados nas Figuras 48a e 48d. Enquanto na Figura 48a o robô pode simplesmente atravessar diagonalmente o corredor, na Figura 48d ele está “preso” à semântica da navegação em corredores.

Figura 49 – Box plot da distância percorrida no cenário de corredores



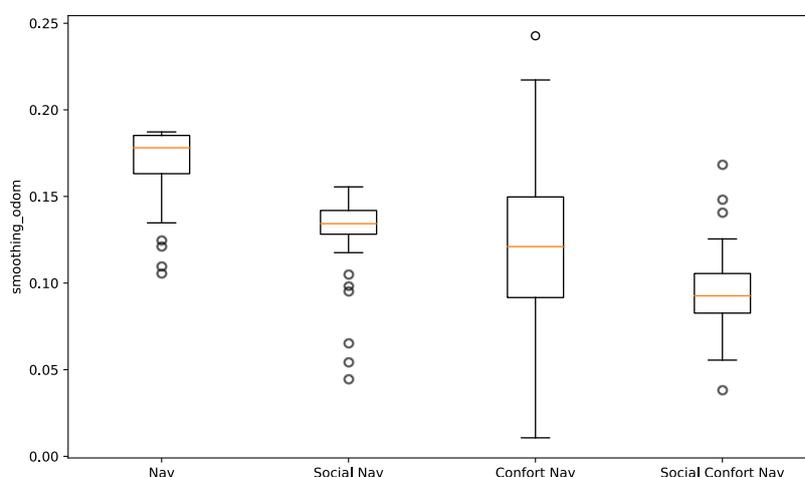
Fonte: Imagem gerada pelo autor.

Quando olhamos para os gráficos de suavidade vemos que a configuração com camadas de conforto e visibilidade teve uma maior variabilidade quando comparada com as outras configurações (Figura 50). Lembrando que essa métrica é sensível e um conjunto de cinquenta simulações pode ainda, causar as variações que são representadas.

Também vale a pena ressaltar que o Social Confort Nav teve os menores valores tanto nos quartis quanto na mediana, quando comparado aos outros. Isso está ligado com o fato de que esse método mantém a cadeira mais tempo andando em linha reta, principalmente após passar pelo ator. De mesmo modo, a curva para retornar ao lado do corredor que estava antes é o que acaba penalizando e reduzindo o bigode mínimo para a caixa do Confort Nav.

No gráfico da Figura 51, que representa o quanto do espaço proxêmico do ator

Figura 50 – Box plot da suavidade alcançada no cenário de corredores



Fonte: Imagem gerada pelo autor.

foi invadido, vemos que a Confort Nav, por querer se restringir novamente ao seu lado do corredor, acabou invadindo mais o espaço proxêmico do que qualquer outra configuração de camadas.

De forma análoga, a falta dessa restrição de localização no corredor que trouxe resultados favoráveis à Social Nav. Adicionalmente, esse é um indício de como a adição ou diferentes combinações de camadas influenciam no resultado final da navegação.

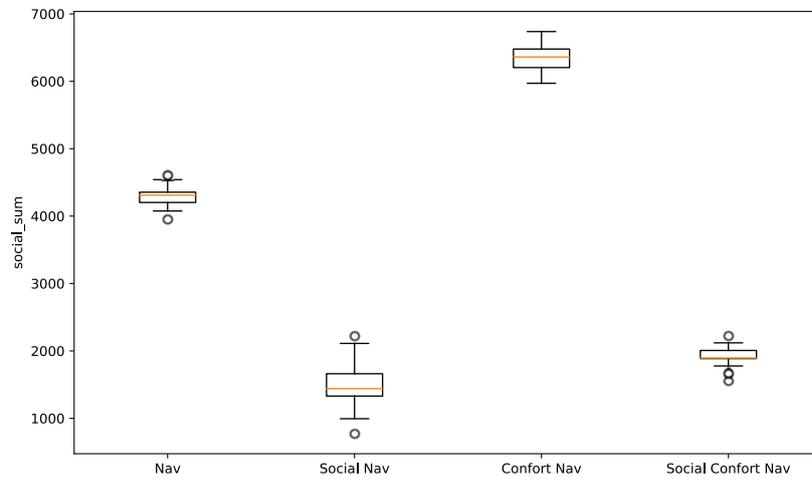
Como resultado, ao compararmos, os resultados da Social Nav e da Social Confort Nav, na Figura 51, reparamos que a tendência é que apenas a camada de proxemics, de maneira isolada, obtém resultados superiores, neste quesito, quando comparada com a mesma configuração, porém com as camadas de conforto do passageiro adicionadas.

Esse resultado abre espaço para discussões acerca da influência de camadas em demais quesitos, uma vez que uma camada diretamente relacionada ao passageiro está tendo efeito negativo, mesmo que baixo, em uma camada que trata do conforto dos pedestres. Assim como sugerido pela primeira hipótese apresentada no começo desse capítulo.

Adiante, no gráfico da Figura 52, vemos o comportamento esperado refletido. Com relação à distância das parede tanto a Confort Nav quanto a Social Confort Nav alcançam resultados superiores quando comparados com as outras duas configurações.

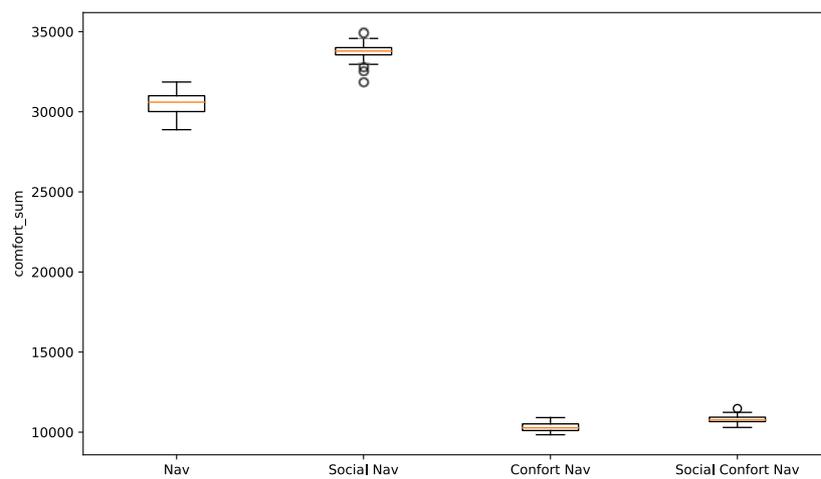
O gráfico da Figura 53 nos trás um resultado peculiar. A curva decrescente mostrada entre os gráficos de caixa nos sugere que há indícios de que, respeitando o espaço proxêmico, por consequência, o robô acaba ganhando mais visibilidade e o

Figura 51 – Soma dos valores da camada de proxemics no cenário de corredores



Fonte: Imagem gerada pelo autor.

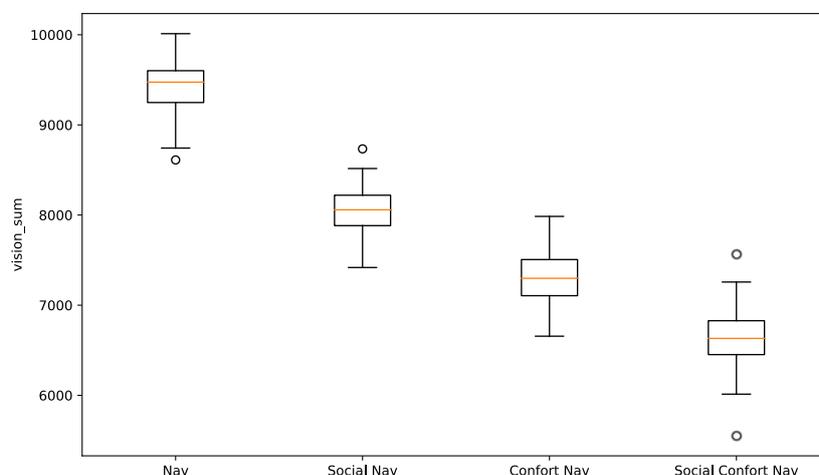
Figura 52 – Soma dos valores da camada de conforto no cenário de corredores



Fonte: Imagem gerada pelo autor.

custo nessa camada é reduzido.

Figura 53 – Soma dos valores da camada de visibilidade no cenário de corredores



Fonte: Imagem gerada pelo autor.

É interessante observarmos o comportamento do robô nas esquinas da Figura 48, que mostra as trajetórias no corredor. O comportamento, anteriormente mencionado, de tentar obter mais visibilidade em curvas não é tão proeminente. Uma hipótese é de que esse fato pode estar relacionado com a largura do corredor, que pode, naturalmente já oferecer um valor de visibilidade aceitável, logo, o robô prefere ficar continuar no mesmo lado.

5.3 Segundo Experimento - Cenário do Angulo

De forma análoga ao que foi feito na seção anterior, a Figura 54 apresenta as trajetórias das diferentes configurações de mapas de custo. A apresentação dessas imagens no início serve como base para as futuras discussões sobre as métricas. Vale lembrar que os dois momentos durante a navegação em que o robô muda seu curso no meio do corredor são os dois pontos de presença dos atores, suas posições aproximadas estão representadas pelos símbolos em laranja.

Neste gráfico, novamente, de forma visual, a navegação clássica e com proxemics, se assemelham muito, tendo como padrão o deslocamento diagonal em grande parte do percurso, havendo diferença apenas na forma como é realizada o desvio do ator. A grande diferença surge na Figura 54c, a qual não tem a camada de proxemics, em muitos casos, isso acarreta no robô acabando por preferir ir em linha reta, atravessando o espaço proxêmico do ator, pois o custo de desviar é maior do que o de passar por perto dele. Esse tipo de comportamento acarretaria em grande desconforto para

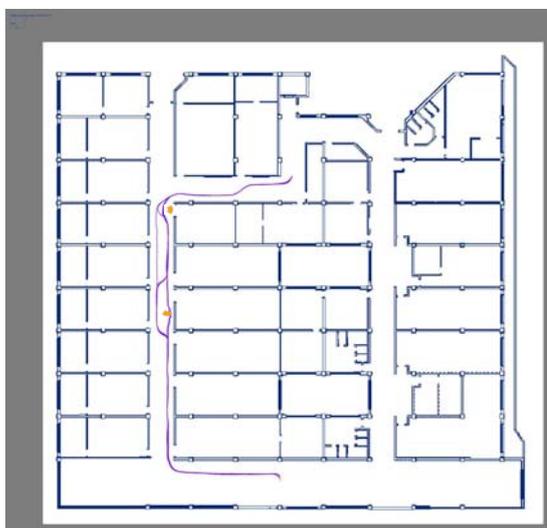
Figura 54 – Navegação em diferentes configurações de mapas de custo no cenário do terceiro andar.



(a) Navegação Clássica



(b) Proxemics



(c) Conforto e Visibilidade



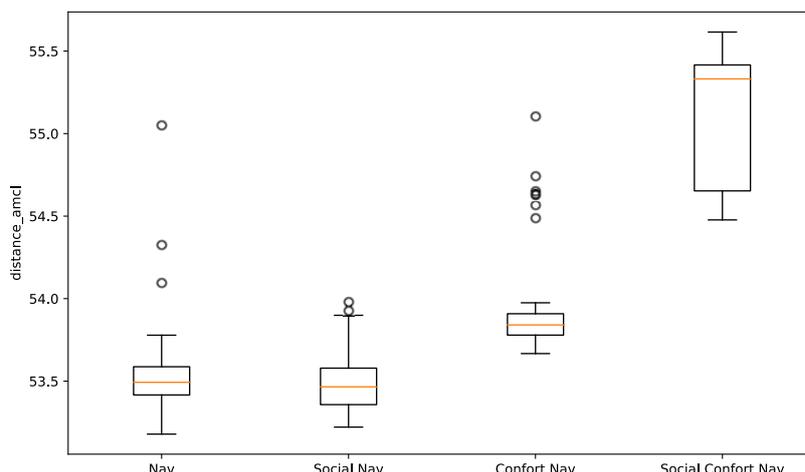
(d) Conforto, Visibilidade e Proxemics

Fonte: Imagem gerada pelo autor.

ambos o passageiro e o pedestre. Isso é corrigido com a adição da camada de proxemics (Figura 54d), onde agora a maior diferença é a preferência ou não do robô de prosseguir na via que estava. Lembrando que o comportamento de seguir em uma via específica pode ser alterado pela função de custo da camada de conforto.

Adiante, o gráfico da Figura 55 nos mostra um resultado parecido com o da Figura 49, ambos tratam da distância percorrida pelo robô no cenário. De mesmo modo, o que mais prejudica o Social Comfort Nav é o fato de se aderir ao outro lado do corredor quando desvia do primeiro ator. Diferente do Comfort Nav, que, pelo fato de valorizar muito a redução do custo geral de continuar no mesmo lado do corredor, acaba passando perto do humano, e como consequência, tem um redução na distância percorrida, quando comparado com o Social Comfort Nav.

Figura 55 – Box plot da distância percorrida no cenário do terceiro andar



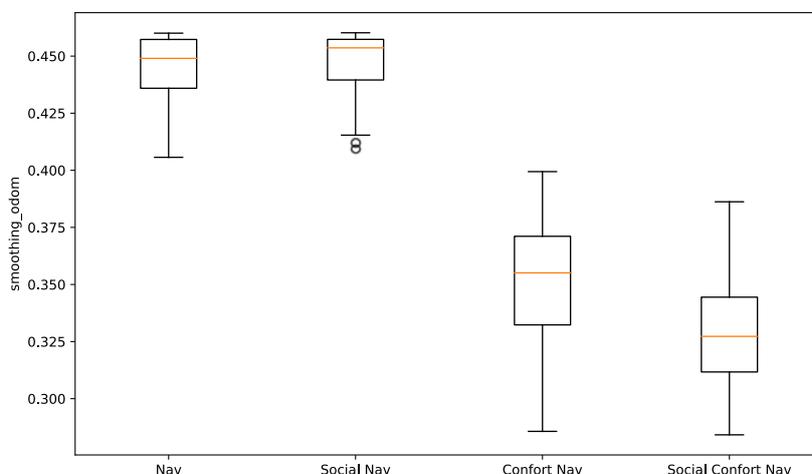
Fonte: Imagem gerada pelo autor.

Seguindo, desta vez, vimos resultados ligeiramente mais controlados nos valores de suavidade na Figura 56. É possível observar que em longos corredores a habilidade da camada de conforto de se aderir a um lado do corredor tem impacto positivo na suavidade, pelo número reduzido de manobras de virada. Diferente da navegação com as camadas clássicas que busca o menor caminho e acaba virando mais vezes, traçando um caminho mais diagonal no corredor.

De forma geral, os resultados na Nav e Social Nav são muito similares, assim como os resultados vistos na Comfort Nav e Social Comfort Nav. Sendo assim, coerente associar a existência das camadas de conforto ao aumento da suavidade, fato que corrobora a segunda hipótese levantada no começo deste capítulo.

Na Figura 57, no caso da camada de proxemics observamos a maior discrepância

Figura 56 – Box plot da suavidade no cenário do terceiro andar



Fonte: Imagem gerada pelo autor.

entre todos os gráficos. Adicionalmente, quando comparamos a Confort Nav com a Nav, ambas sem proxemics, vemos que a diferença da mediana é de pouco menos de 4000 na Nav para mais de 1000 na Confort Nav.

Como já citado anteriormente, o fato da Confort Nav se aderir à direita do corredor e não ter influência da camada de proxemics, ou seja, os custos da região em que ficaria o espaço proxêmico não serem altos o suficiente para alterar a tomada de decisão do robô com relação à rota, ele acaba invadindo o espaço proxêmico do primeiro ator por muito tempo, causando um acúmulo grande nos valores dessa camada.

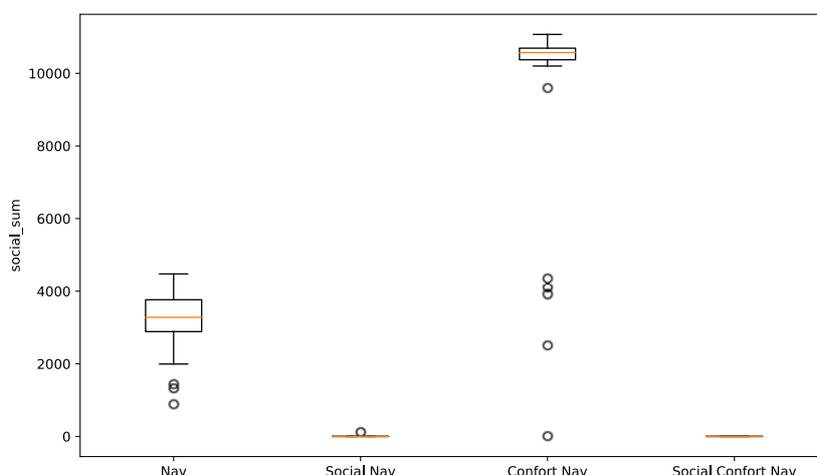
Quando analisamos a correlação de Pearson entre a distância e o coeficiente social, para os testes no Confort Nav, há uma forte correlação negativa entre ambos (-0.91), a qual reforça a observação de que a distância diminui conforme a CRMA invade o espaço pessoal dos atores. Além de que também é possível observar uma correlação muito positiva entre as somas dos valores das camadas de visibilidade e social (0.96) nos testes Confort Nav, a qual indica que o principal responsável pela invasão do espaço proxêmica seja a camada de conforto ao invés da de visibilidade, uma vez que os valores desta última acumulam juntamente com o da camada de proxemics.

É provável que os outliers positivos neste caso sejam os momentos em que a CRMA preferiu por evitar o ator, como se fosse um obstáculo comum, e acabou por não invadir tanto o espaço proxêmico, minimizando muito os valores, mas mesmo assim, os limites dos bigodes superior e inferior do gráfico, que representam os limites da distribuição dos dados, é muito alto.

Esse fato corrobora com a ideia principal do trabalho, de que diferentes métodos de interação social, tanto se tratando do pedestre como do passageiro, tem de ser adotados, ainda mais quando envolvendo a navegação social de CRMAs.

É importante sempre dar atenção a casos como este pois a preocupação com o conforto do passageiro ativamente piorou um caso de conforto do pedestre, quando não houve a combinação de ambas.

Figura 57 – Box plot da soma dos valores da camada de proxemics no cenário do terceiro andar



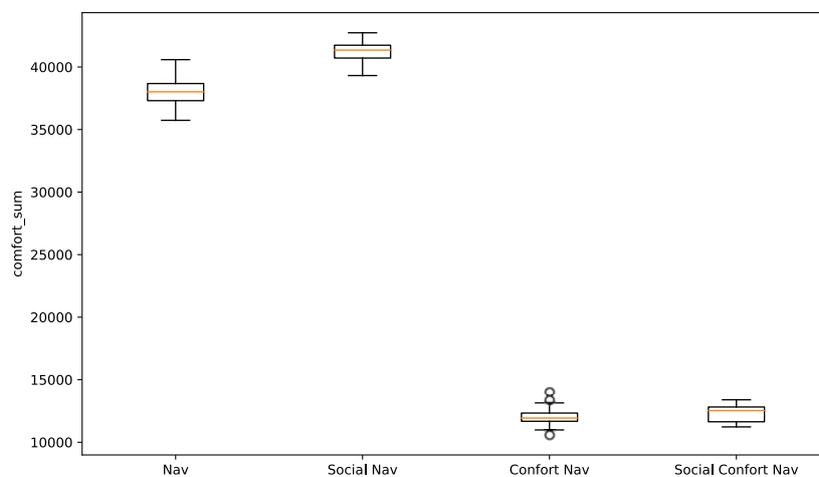
Fonte: Imagem gerada pelo autor.

Como era o esperado, na Figura 58, assim como na Figura 52, com relação ao conforto os resultados das configurações que envolveram a camada de conforto foram muito superiores. Porém, no teste do Social Confort Nav, os valores da soma da camada de conforto tem uma correlação altamente positiva com os valores de distância (0.897), o que provavelmente é proveniente dos momentos em que a cadeira troca a via do corredor em que está trafegando, aumentando a distância e consequentemente diminuindo o conforto.

Por final, um resultado inesperado. Na Figura 59 os algoritmos que implementaram a camada de visibilidade foram os que desempenham pior nesse quesito. Esse fato curioso é sustentado pela ideia de que, enquanto em um corredor longo, robôs que trafegam pelo meio tem sempre a melhor visibilidade, enquanto robôs que trafegam mais para as laterais acabam acumulando valores do mapa de custo de visibilidade, porém, pela compensação de custos criada pela camada de conforto (sustentada pelos resultados da Figura 58) o robô opta por seguir no trajeto.

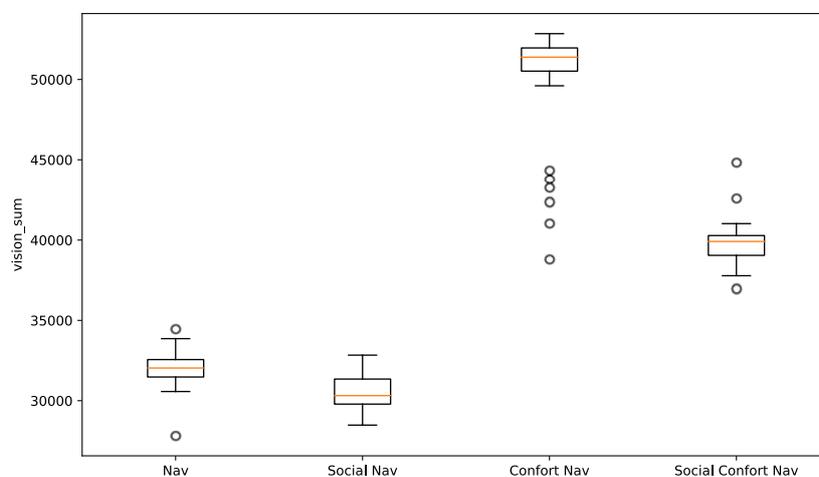
Ainda na Figura 59, a alta correlação negativa, de -0,899, entre os valores de distância e a soma dos valores da camada de visibilidade corroboram com o fato de que

Figura 58 – Box plot da soma dos valores da camada de conforto no cenário do terceiro andar



Fonte: Imagem gerada pelo autor.

Figura 59 – Box plot da soma dos valores da camada de visibilidade no cenário do terceiro andar



Fonte: Imagem gerada pelo autor.

a camada de conforto, que faz com que o robô fique mais próximo de paredes, consequentemente aumentando os valores de visibilidade, uma vez que as vezes em que o robô toma os caminhos maiores são os que ele muda de via e acaba se distanciando temporariamente da parede, aumentando a visibilidade.

Para tentar corrigir esse problema é possível balancear melhor as diferentes camadas atribuindo diferentes multiplicadores de pesos a cada uma. Seriam constantes

que multiplicariam o valor final da camada a fim de aumentar ou reduzir o impacto de cada uma no resultado final da navegação. Idealmente, esses valores de ganho teriam de ser ajustados para tentar encontrar valores ideais que maximizem uma série de métricas, ou de acordo com as preferências do usuário.

5.4 Considerações Finais

Testar diferentes combinações de camadas para diferentes interações entre o robô e os humanos é imprescindível para o avanço da área de navegação social. Sem contar pelo fato de que existe uma miríade de interações que poderiam ter sido abordadas nos teste realizados.

Adiante, a primeira hipótese levantada no começo desse capítulo (H1) sugeriria que a ausência de uma camada teria impactos negativos em outra. A resposta para essa hipótese não é totalmente conclusiva, se observarmos os dados não veremos um indício numérico que corrobore, pelo contrário (vide Figura 51). Desta forma, mesmo que o resultado no desempenho da navegação tenha sido positivo por um lado, certamente teria implicações graves no conforto do passageiro, o que seria um impacto negativo de qualquer forma nas camadas que modelam a relação da cadeira com o seu usuário.

Outrossim, nesse tópico é necessário também destacar os resultados vistos na Figura 57, em que a inclusão das camadas de conforto e visibilidade, sem a camada de proxemics, tem impacto severamente negativo no conforto do pedestre. Isso é algo que precisa ser evitado, evidenciando a necessidade de camadas de diferentes semânticas, com relação ao passageiro e ao pedestre, para garantir o conforto de ambos na navegação de uma CRMA.

Já a segunda hipótese (H2) teve resultados que corroboraram nos testes realizados, afinal, tanto na Figura 50, quanto na 56, a adição das camadas de conforto trouxeram de fato melhores resultados para os índices de suavidade. Isso se deve ao fato de que, seguindo uma única via no corredor, a cadeira se mantém mais estável e acaba realizando menos manobras de virada no geral, a qual acaba melhorando a experiência do usuário da CRMA.

Além disso, é importante avaliar o impacto real das camadas, por exemplo, na Figura 59 a adição da camada de visibilidade não teve contribuição no índice de visibilidade. Nesse caso seria necessário analisar a adição de um ganho na função de custo nos testes das camadas para que a contribuição delas no resultado final possa ser ajustada e avaliada.

6 CONCLUSÕES

Este trabalho apresentou uma combinação de diferentes fatores de interações sociais aplicados ao contexto de Cadeiras de Rodas Motorizadas Autônomas. Deste modo, focando tanto nas relações do grupo-interno (passageiro-cadeira), quanto do grupo-externo (pedestre-cadeira). O objetivo principal foi avaliar, sob um conjunto de métricas, a viabilidade de aplicações dessas diferentes maneiras de se propor navegação social, tanto de forma conjunta, como separada.

Foram realizados experimentos em dois cenários diferentes, um parecido com um dos ambientes utilizado por um dos trabalhos já bem fundamentados na área, para teste de base desses algoritmos, e outro que se aproxima de um cenário real em um ambiente público.

Os resultados apontam que uma navegação confortável tem impacto e deve também se preocupar com a suavidade do movimento, uma vez que isso reflete na experiência geral do usuário. Algumas peculiaridades na interação entre as camadas puderam ser observadas, expondo a necessidade de tratar regras sociais com diferentes semânticas no mesmo conjunto de navegação, uma vez que, e.g., algoritmos que priorizam o conforto com relação a uma posição no cenário podem acabar causando desconforto na interação com outros humanos se essa precaução não for levada em conta.

Os testes também iluminaram o fato de que camadas com regras sociais que têm princípios semelhantes devem ser devidamente balanceada em suas funções de custo. Entretanto, o ganho de cada camada deve ser testado para que cada uma dê a contribuição necessária no resultado final da navegação. Ainda que haja casos em que dar mais valor a uma camada do que a outra pode ser uma característica positiva e desejada na implementação do sistema.

A principal hipótese, de que são necessárias tanto as camadas que favorecem o conforto do passageiro como as do pedestre, uma vez que apenas a utilização delas separadamente não é o suficiente conforto para ambos em alguns casos.

O objetivo principal foi alcançado por meio dos testes realizados em simulação, que envolviam a interação tanto os fatores de interação com o pedestre, quanto com

o passageiro, na navegação de CRMAs. Por meio disto, é possível continuar a expansão da área de pesquisa para que as CRMAs possam ser popularizadas e aprimorar a experiência do conforto das mesmas, para pedestres e usuários, em ambientes públicos.

Com relação aos objetivos específicos, o primeiro foi completado pois foi possível elencar como se constitui uma CRMA, separando os componentes em três diferentes características: dispositivo de processamento, interface de controle e sensores embarcados, como pode ser visto no Capítulo 2.1.

De mesmo modo, com relação ao segundo objetivo específico, no Capítulo 3 se enumerou uma série de alternativas para fatores sociais disponíveis na literatura, visando tanto o passageiro como o pedestre.

Para completar o objetivo específico 3, a solução desenvolvida foi integrada ao *Navigation Stack* e disponibilizada publicamente no github.

Abordando o quarto objetivo específico, no Capítulo 4.5 foi definido um conjunto de métricas para avaliação dos fatores sociais em simulação de navegação social de CRMAs, uma vez que, na literatura, os testes são corriqueiramente relacionados a avaliações aplicados em testes reais com os indivíduos envolvidos.

Além disso, para o quinto objetivo específico, as diferentes combinações de mapas de custo foram testadas e os resultados das interações entre as mesmas foram discutidos no Capítulo 5.

Adicionalmente, juntamente com os testes das configurações de camadas de custo realizadas, um modelo proposto, com as camadas de conforto, visibilidade e social, foi comparado com a navegação clássica, também no Capítulo 5, com relação ao sexto objetivo específico. Deste modo, a configuração Social Comfort Nav, que engloba todas as camadas de custo, tanto as propostas quanto as clássicas, com exceção à variável que mede a distância percorrida, apresentou resultados superiores em todas as demais métricas.

O trabalho apresentou discussões positivas acerca dos resultados encontrados. Porém, há de se explicitar o fato de que os testes foram limitados em vários aspectos. Tanto no número de interações que ele abrangeu, quanto ao que se delimitaram os cenários de testes, como por exemplo, apenas atores estáticos e isolados.

O principal ponto negativo é a falta de validação de resultados com pessoas em ambiente real, uma vez que daria mais respaldo à navegação social e daria margem de comparação à outros resultados presentes na literatura. Um fator que dificultou a realização desses experimentos foi a pandemia de COVID-19 que interrompeu o acesso aos laboratórios físicos para desenvolvimento do robô.

6.1 Trabalhos Futuros

Como trabalhos futuros podem ser citados inúmeros testes. Por exemplo, a utilização de uma maior gama de regras sociais. Especialmente regras que tratem os espaços proxêmicos como uma mistura de gaussianas simulando a movimentação de humanos ou a interação deles com objetos ou outros humanos.

Além disso, as camadas de conforto e visibilidade podem ser analisadas separadamente na mesma bateria de testes realizada neste trabalho, com o objetivo de analisar qual das duas tem mais influência nos resultados apresentados.

Atualmente, apenas a função de custo de cada camada é calculada, sem nenhum ganho adicionado a essa função. Adicionalmente, podem ser avaliados diferentes ganhos para os custos de cada camada, avaliando como esses valores interferem no resultado final da navegação, buscando também os valores ideais para maximizar as métricas apresentadas. Outra opção são ganhos adaptativos, onde dependendo do contexto onde a CRMA se encontra, diferentes valores são utilizados, e.g., próximo à curvas é dado um valor maior ao resultado da função da camada visibilidade, já trafegando por corredor a camada de conforto ganha prioridade.

A maneira como uma CRMA pode abordar uma pessoa ou um grupo de pessoa também é um problema disruptivo e relevante.

Também podem ser realizados mais testes, com ambientes simulados diferentes ou até testes com robôs reais em ambientes internos, podendo avaliar assim como é a reação das demais pessoas aos comportamentos da CRMA.

Por fim, podem ser utilizadas técnicas de IA mais avançadas, como a segmentação e raciocínio sobre cenas para detectar pontos de interesse ou contextos para a troca do tipo de navegação com alterações das camadas utilizadas. Ou ainda, a aplicação de Aprendizado por Reforço para que conjuntos de regras sociais não precisem ser descritos matematicamente, mas ter seus padrões aprendidos por um modelo de forma autônoma.

REFERÊNCIAS

AFONSO, P. D. A.; FERREIRA, P. R. Autonomous Navigation of Wheelchairs in Indoor Environments using Deep Reinforcement Learning and Computer Vision. In: LATIN AMERICAN ROBOTICS SYMPOSIUM (LARS), 2023 BRAZILIAN SYMPOSIUM ON ROBOTICS (SBR), AND 2023 WORKSHOP ON ROBOTICS IN EDUCATION (WRE), 2023., 2023. **Anais...** [S.l.: s.n.], 2023. p.260–265.

AKITA, S.; SATAKE, S.; SHIOMI, M.; IMAI, M.; KANDA, T. Social coordination for looking-together situations. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p.834–841.

ALKHATIB, R.; SWAIDAN, A.; MARZOUK, J.; SABBAH, M.; BERJAOUI, S.; DIAB, M. O. Smart Autonomous Wheelchair. In: INTERNATIONAL CONFERENCE ON BIO-ENGINEERING FOR SMART TECHNOLOGIES (BIOSMART), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p.1–5.

ALY, A.; TAPUS, A. Prosody-driven robot arm gestures generation in human-robot interaction. **Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction**, [S.l.], 2012.

AMAOKA, T.; LAGA, H.; SAITO, S.; NAKAJIMA, M. Personal space-based modeling of relationships between people for new human-computer interaction. **International Workshop on Advanced Imaging Techniques (IWAIT) 2009**, [S.l.], 2009.

ANDERSONE, I. Heterogeneous map merging: state of the art. **Robotics**, [S.l.], v.8, p.74, 2019.

ARDIGÒ, L. P.; GOOSEY-TOLFREY, V. L.; MINETTI, A. E. Biomechanics and energetics of basketball wheelchairs evolution. **International Journal of Sports Medicine**, [S.l.], v.26, p.388–396, 2005.

BABEL, F.; KRAUS, J.; BAUMANN, M. Findings from a qualitative field study with an

autonomous robot in public: exploration of user reactions and conflicts. **International Journal of Social Robotics**, [S.l.], v.14, p.1625–1655, 2022.

BAIJU, P. V.; VARGHESE, K.; ALAPATT, J. M.; JOJU, S. J.; SAGAYAM, K. M. Smart Wheelchair for Physically Challenged People. In: INTERNATIONAL CONFERENCE ON ADVANCED COMPUTING AND COMMUNICATION SYSTEMS (ICACCS), 2020., 2020. **Anais...** [S.l.: s.n.], 2020. p.828–831.

BARDARO, G.; BASCETTA, L.; CERAVOLO, E.; FARINA, M.; GABELLONE, M.; MATTEUCCI, M. MPC-based control architecture of an autonomous wheelchair for indoor environments. **Control Engineering Practice**, [S.l.], v.78, p.160–174, 2018.

BEESON, P.; JONG, N. K.; KUIPERS, B. Towards autonomous topological place detection using the extended voronoi graph. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2005., 2005. **Proceedings...** [S.l.: s.n.], 2005. p.4373–4379.

BERTOLACCINI, G. d. S.; FILHO, I. F. P. d. C.; CHRISTOFOLETTI, G.; PASCHOARELLI, L. C.; MÉDOLA, F. O. The influence of axle position and the use of accessories on the activity of upper limb muscles during manual wheelchair propulsion. **International Journal of Occupational Safety and Ergonomics**, [S.l.], v.24, p.311–315, 2017.

BRŠČIĆ, D.; ZANLUNGO, F.; KANDA, T. Density and velocity patterns during one year of pedestrian tracking. **Transportation Research Procedia**, [S.l.], v.2, p.77–86, 2014.

CAVANINI, L.; BENETAZZO, F.; FREDDI, A.; LONGHI, S.; MONTERIU, A. SLAM-based autonomous wheelchair navigation system for AAL scenarios. In: IEEE/ASME 10TH INTERNATIONAL CONFERENCE ON MECHATRONIC AND EMBEDDED SYSTEMS AND APPLICATIONS (MESA), 2014., 2014. **Anais...** [S.l.: s.n.], 2014. p.1–5.

CHARALAMPOUS, K.; KOSTAVELIS, I.; GASTERATOS, A. Recent trends in social aware robot navigation: A survey. **Robotics and Autonomous Systems**, [S.l.], v.93, p.85–104, 2017.

CHAUDHARY, U.; XIA, B.; SILVONI, S.; COHEN, L. G.; BIRBAUMER, N. Brain–computer interface–based communication in the completely locked-in state. **PLoS biology**, [S.l.], v.15, n.1, p.e1002593, 2017.

CONTRACTOR, V.; GREWAL, H. S.; JAYAPRAKASH, N. T.; MATTHEWS, A.; GEORGE, K. PCL-Based Autonomous Wheelchair Navigation in Unmapped Indoor Environments. In: IEEE ANNUAL UBIQUITOUS COMPUTING, ELECTRONICS & MOBILE COMMUNICATION CONFERENCE (UEMCON), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p.291–296.

COOPER, R. A. Smartwheel. **Prosthetics Amp; Orthotics International**, [S.l.], v.33, p.198–209, 2009.

CUI, J.; SHANG, Y.; YU, S.; WANG, Y. Research on intelligent wheelchair multimode human–computer interaction and assisted driving technology. **Actuators**, [S.l.], v.13, p.230, 2024.

DAZA, M.; BARRIOS-ARANIBAR, D.; DÍAZ-AMADO, J.; CARDINALE, Y.; VILAS-BOAS, J. An approach of social navigation based on proxemics for crowded environments of humans and robots. **Micromachines**, [S.l.], v.12, p.193, 2021.

DEHNAVI, S.; KOEDAM, M.; NELSON, A. T.; GOSWAMI, D.; GOOSSENS, K. Compros: a composable ros2 based architecture for real-time embedded robotic development. **2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**, [S.l.], 2021.

DELLAERT, F.; FOX, D.; BURGARD, W.; THRUN, S. Monte carlo localization for mobile robots. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (CAT. NO. 99CH36288C), 1999., 1999. **Proceedings...** [S.l.: s.n.], 1999. v.2, p.1322–1328.

DIJKSTRA, E. W. et al. A note on two problems in connexion with graphs. **Numerische mathematik**, [S.l.], v.1, n.1, p.269–271, 1959.

DING, H. Motion path planning of soccer training auxiliary robot based on genetic algorithm in fixed-point rotation environment. **Journal of Ambient Intelligence and Humanized Computing**, [S.l.], v.11, p.6261–6270, 2020.

DISSANAYAKE, G.; NEWMAN, P.; CLARK, S.; DURRANT-WHYTE, H.; CSORBA, M. A solution to the simultaneous localization and map building (slam) problem. **IEEE Transactions on Robotics and Automation**, [S.l.], v.17, p.229–241, 2001.

DUDA, R. O.; HART, P. E. Use of the Hough transformation to detect lines and curves in pictures. **Commun. ACM**, New York, NY, USA, v.15, n.1, p.11–15, Jan. 1972.

FERNANDES, G.; LEITE, A. B.; DÍAZ-AMADO, J.; CARDINALE, Y. Social aware navigation based on proxemic interaction for an autonomous wheelchair. **Workshop Proceedings of the 19th International Conference on Intelligent Environments (IE2023)**, [S.l.], 2023.

FERREIRA, S. A.; SOUZA, W. C. d.; NASCIMENTO, M. A. d.; TARTARUGA, M. P.; PORTELA, B. S.; MASCARENHAS, L. P. G.; QUEIRÓGA, M. R. Características morfológicas, desempenho de força e de potência anaeróbia em jogadores de basquetebol

em cadeira de rodas. **Brazilian Journal of Kinanthropometry and Human Performance**, [S.l.], v.19, p.343, 2017.

FOX, D.; BURGARD, W.; THRUN, S. The dynamic window approach to collision avoidance. **IEEE Robotics & Automation Magazine**, [S.l.], v.4, n.1, p.23–33, 1997.

GLAS, D. F.; MIYASHITA, T.; ISHIGURO, H.; HAGITA, N. Laser-based tracking of human position and orientation using parametric shape modeling. **Advanced robotics**, [S.l.], v.23, n.4, p.405–428, 2009.

GORCE, P.; LOUIS, N. Wheelchair propulsion kinematics in beginners and expert users: Influence of wheelchair settings. **Clinical Biomechanics**, [S.l.], v.27, n.1, p.7–15, 2012.

GREWAL, H.; MATTHEWS, A.; TEA, R.; GEORGE, K. LIDAR-based autonomous wheelchair. In: IEEE SENSORS APPLICATIONS SYMPOSIUM (SAS), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p.1–6.

GREWAL, H. S.; JAYAPRAKASH, N. T.; MATTHEWS, A.; SHRIVASTAV, C.; GEORGE, K. Autonomous wheelchair navigation in unmapped indoor environments. In: IEEE INTERNATIONAL INSTRUMENTATION AND MEASUREMENT TECHNOLOGY CONFERENCE (I2MTC), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p.1–6.

GREWAL, H. S.; MATTHEWS, A.; TEA, R.; CONTRACTOR, V.; GEORGE, K. Sip-and-Puff Autonomous Wheelchair for Individuals with Severe Disabilities. In: IEEE ANNUAL UBIQUITOUS COMPUTING, ELECTRONICS & MOBILE COMMUNICATION CONFERENCE (UEMCON), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p.705–710.

HALL, E. T.; BIRDWHISTELL, R. L.; BOCK, B.; BOHANNAN, P.; DIEBOLD JR, A. R.; DURBIN, M.; EDMONSON, M. S.; FISCHER, J.; HYMES, D.; KIMBALL, S. T. et al. Proxemics [and comments and replies]. **Current anthropology**, [S.l.], v.9, n.2/3, p.83–108, 1968.

HART, P. E.; NILSSON, N. J.; RAPHAEL, B. A formal basis for the heuristic determination of minimum cost paths. **IEEE transactions on Systems Science and Cybernetics**, [S.l.], v.4, n.2, p.100–107, 1968.

HARTMAN, A.; NANDIKOLLA, V. K. Human-machine interface for a smart wheelchair. **Journal of Robotics**, [S.l.], v.2019, 2019.

HELBING, D.; MOLNÁR, P.; FARKAS, I. J.; BOLAY, K. Self-organizing pedestrian movement. **Environment and planning B: planning and design**, [S.l.], v.28, n.3, p.361–383, 2001.

HIDAYAT, F.; TRILAKSONO, B. R.; HINDERSYAH, H. Distributed multi - robot rao - blac k wellized particle filt er ing simultaneous localization and mapping with consenus calculation of particle weight and posterior parameters. **International Journal on Electrical Engineering and Informatics**, [S.l.], v.9, p.632–658, 2017.

HOCHBERG, L. R.; SERRUYA, M. D.; FRIEHS, G. M.; MUKAND, J. A.; SALEH, M.; CAPLAN, A. H.; BRANNER, A.; CHEN, D.; PENN, R. D.; DONOGHUE, J. P. Neuro- nal ensemble control of prosthetic devices by a human with tetraplegia. **Nature**, [S.l.], v.442, n.7099, p.164–171, 2006.

HUSSEIN, W.; MAGED, S. M.; ADEL, E. S.; SABRY, S.; ABOBAKR, O. A.; MUSTAFA, M. A.; DAKROURY, A. E.; MORSY, E. A. Modelling, simulation, and implementing ros for autonomous navigation of tracked robot. **International Journal of Advances in Scientific Research and Engineering**, [S.l.], v.07, p.01–13, 2021.

IBRAHIM, M. S. bin; PEEIE, M. H. bin. The dynamic behavior investigation of electric power wheelchair during the obstacle avoidance. In: IOP CONFERENCE SERIES: MATERIALS SCIENCE AND ENGINEERING, 2020. **Anais...** [S.l.: s.n.], 2020. v.788, n.1, p.012074.

ISSA, B.; RASHID, A. Rp lidar sensor for multi-robot localization using leader follower algorithm. **Iraqi Journal for Electrical and Electronic Engineering**, [S.l.], v.15, p.21–32, 2019.

JANG, E. A study on the uwb-based position estimation method using dead recko- ning information for active driving in a mapless environment of intelligent wheelchairs. **Applied Sciences**, [S.l.], v.14, p.620, 2024.

JOHNSON, C.; KUIPERS, B. Socially-aware navigation using topological maps and social norm learning. In: AAAI/ACM CONFERENCE ON AI, ETHICS, AND SOCIETY, 2018., 2018. **Proceedings...** [S.l.: s.n.], 2018. p.151–157.

JOSHI, S.; HERRERA, R. R.; SPRINGETT, D.; WEEDON, B. D.; RAMÍREZ, D. Z. M.; HOLLOWAY, C.; AYAZ, H.; DAWES, H. A cross-sectional study using wireless electro- cardiogram to investigate physical workload of wheelchair control in real world environ- ments. **Advances in Neuroergonomics and Cognitive Engineering**, [S.l.], p.14–25, 2019.

KALMAN, R. E. A New Approach to Linear Filtering and Prediction Problems. **Transac- tions of the ASME–Journal of Basic Engineering**, [S.l.], v.82, n.Series D, p.35–45, 1960.

KAPPEL, K.; FERREIRA, P. R. Towards Comfortable and Socially Acceptable Navigation in Autonomous Motorized Wheelchairs. In: LATIN AMERICAN ROBOTICS SYMPOSIUM (LARS), 2023 BRAZILIAN SYMPOSIUM ON ROBOTICS (SBR), AND 2023 WORKSHOP ON ROBOTICS IN EDUCATION (WRE), 2023., 2023. **Anais...** [S.l.: s.n.], 2023. p.319–324.

KAPPEL, K. S.; FERREIRA JR, P. R. Literature Review of Motorized Autonomous Wheelchairs. **IV Brazilian Humanoid Robot Workshop and V Brazilian Workshop on Service Robotics**, [S.l.], 2020.

KIRBY, R.; SIMMONS, R.; FORLIZZI, J. COMPANION: A Constraint-Optimizing Method for Person-Acceptable Navigation. In: RO-MAN 2009 - THE 18TH IEEE INTERNATIONAL SYMPOSIUM ON ROBOT AND HUMAN INTERACTIVE COMMUNICATION, 2009. **Anais...** [S.l.: s.n.], 2009. p.607–612.

KOSTAVELIS, I. Robot behavioral mapping: a representation that consolidates the human-robot coexistence. **Robotics and Automation Engineering Journal**, [S.l.], v.1, 2017.

KRUSE, T.; PANDEY, A. K.; ALAMI, R.; KIRSCH, A. Human-aware robot navigation: A survey. **Robotics and Autonomous Systems**, [S.l.], v.61, n.12, p.1726–1743, 2013.

KÜBLER, A.; WINTER, S.; LUDOLPH, A. C.; HAUTZINGER, M.; BIRBAUMER, N. Severity of depressive symptoms and quality of life in patients with amyotrophic lateral sclerosis. **Neurorehabilitation and neural repair**, [S.l.], v.19, n.3, p.182–193, 2005.

KUMAR, D.; MALHOTRA, R.; SHARMA, S. Design and Construction of a Smart Wheelchair. **Procedia Computer Science**, [S.l.], v.172, p.302–307, 2020.

LABBÉ, M.; MICHAUD, F. Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. **Journal of Field Robotics**, [S.l.], v.36, p.416–446, 2018.

LEAMAN, J.; LA, H. A comprehensive review of smart wheelchairs: past, present, and future. **Ieee Transactions on Human-Machine Systems**, [S.l.], v.47, p.486–499, 2017.

LEAMAN, J.; LA, H. M.; NGUYEN, L. Development of a smart wheelchair for people with disabilities. In: IEEE INTERNATIONAL CONFERENCE ON MULTISENSOR FUSION AND INTEGRATION FOR INTELLIGENT SYSTEMS (MFI), 2016., 2016. **Anais...** [S.l.: s.n.], 2016. p.279–284.

LEE, H.; LEE, S.; CHOI, M. H.; LEE, B. Probabilistic map merging for multi-robot rbpf-slam with unknown initial poses. **Robotica**, [S.l.], v.30, p.205–220, 2011.

LEE, K. K.; UIHLEIN, M. J. Adaptive sports in the rehabilitation of the disabled veterans. **Physical medicine and rehabilitation clinics of North America**, [S.l.], v.30, n.1, p.289–299, 2019.

LEE, Y.-C.; PARK, S.-H.; YU, W.; KIM, S.-H. et al. Topological map building for mobile robots based on GIS in urban environments. In: INTERNATIONAL CONFERENCE ON UBIQUITOUS ROBOTS AND AMBIENT INTELLIGENCE (URAI), 2011., 2011. **Anais...** [S.l.: s.n.], 2011. p.790–791.

LEE, Y. K.; LIM, J. M.; EU, K. S.; GOH, Y. H.; TEW, Y. Real time image processing based obstacle avoidance and navigation system for autonomous wheelchair application. In: ASIA-PACIFIC SIGNAL AND INFORMATION PROCESSING ASSOCIATION ANNUAL SUMMIT AND CONFERENCE (APSIPA ASC), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p.380–385.

LEU, A.; RISTIĆ-DURRANT, D.; SLAVNIĆ, S.; GLACKIN, C.; SALGE, C.; POLANI, D.; BADII, A.; KHAN, A.; RAVAL, R. Corbys cognitive control architecture for robotic follower. **Proceedings of the 2013 IEEE/SICE International Symposium on System Integration**, [S.l.], 2013.

LIU, Y.; WANG, C.; WU, H.; WEI, Y. Mobile robot path planning based on kinematically constrained a-star algorithm and dwa fusion algorithm. **Mathematics**, [S.l.], v.11, p.4552, 2023.

LU, D. V. **Contextualized robot navigation**. [S.l.]: Washington University in St. Louis, 2014.

LU, D. V.; HERSHBERGER, D.; SMART, W. D. Layered costmaps for context-sensitive navigation. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 2014., 2014. **Anais...** [S.l.: s.n.], 2014. p.709–715.

LU, D. V.; SMART, W. D. Towards more efficient navigation for robots and humans. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 2013., 2013. **Anais...** [S.l.: s.n.], 2013. p.1707–1713.

LU, H.; YANG, S.; ZHAO, M.; CHENG, S. Multi-robot indoor environment map building based on multi-stage optimization method. **Complex System Modeling and Simulation**, [S.l.], v.1, p.145–161, 2021.

LUO, S.; ZHANG, M.; ZHUANG, Y.; MA, C.; LI, Q. A survey of path planning of industrial robots based on rapidly exploring random trees. **Frontiers in Neurorobotics**, [S.l.], v.17, 2023.

MARCHAL–CRESPO, L.; FURUMASU, J.; REINKENSMEYER, D. J. A robotic wheelchair trainer: design overview and a feasibility study. **Journal of NeuroEngineering and Rehabilitation**, [S.l.], v.7, 2010.

MATSUMARU, T. Mobile robot with preliminary-announcement and display function of forthcoming motion using projection equipment. In: ROMAN 2006-THE 15TH IEEE INTERNATIONAL SYMPOSIUM ON ROBOT AND HUMAN INTERACTIVE COMMUNICATION, 2006. **Anais...** [S.l.: s.n.], 2006. p.443–450.

MATSUMARU, T.; KUSADA, T.; IWASE, K. Mobile robot with preliminary-announcement function of forthcoming motion using light-ray. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 2006., 2006. **Anais...** [S.l.: s.n.], 2006. p.1516–1523.

MAVRIDIS, N. A review of verbal and non-verbal human–robot interactive communication. **Robotics and Autonomous Systems**, [S.l.], v.63, p.22–35, 2015.

MAVROGIANNIS, C.; BALDINI, F.; WANG, A.; ZHAO, D.; TRAUTMAN, P.; STEINFELD, A.; OH, J. Core challenges of social robot navigation: A survey. **arXiv preprint arXiv:2103.05668**, [S.l.], 2021.

MINTROM, M.; SUMARTOJO, S.; KULIć, D.; TIAN, L.; CARRENO-MEDRANO, P.; ALLEN, A. Robots in public spaces: implications for policy design. **Policy Design and Practice**, [S.l.], v.5, p.123–139, 2021.

MORALES, Y.; EVEN, J.; KALLAKURI, N.; IKEDA, T.; SHINOZAWA, K.; KONDO, T.; HAGITA, N. Visibility analysis for autonomous vehicle comfortable navigation. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), 2014., 2014. **Anais...** [S.l.: s.n.], 2014. p.2197–2202.

MORALES, Y.; KALLAKURI, N.; SHINOZAWA, K.; MIYASHITA, T.; HAGITA, N. Human-comfortable navigation for an autonomous robotic wheelchair. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 2013., 2013. **Anais...** [S.l.: s.n.], 2013. p.2737–2743.

MORALES, Y.; MIYASHITA, T.; HAGITA, N. Social robotic wheelchair centered on passenger and pedestrian comfort. **Robotics and Autonomous Systems**, [S.l.], v.87, p.355–362, 2017.

MORALES, Y.; WATANABE, A.; FERRERI, F.; EVEN, J.; IKEDA, T.; SHINOZAWA, K.; MIYASHITA, T.; HAGITA, N. Including human factors for planning comfortable paths. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), 2015., 2015. **Anais...** [S.l.: s.n.], 2015. p.6153–6159.

MORALES, Y.; WATANABE, A.; FERRERI, F.; EVEN, J.; SHINOZAWA, K.; HAGITA, N. Passenger discomfort map for autonomous navigation in a robotic wheelchair. **Robotics and Autonomous Systems**, [S.l.], v.103, p.13–26, 2018.

MORGAN, K. Repetition-based training for efficient propulsion in new manual wheelchair users. **Physical Medicine, Rehabilitation Amp; Disabilities**, [S.l.], v.1, p.1–9, 2015.

MORROW, M. M.; HURD, W. J.; KAUFMAN, K. R.; AN, K. N. Shoulder demands in manual wheelchair users across a spectrum of activities. **Journal of Electromyography and Kinesiology**, [S.l.], v.20, p.61–67, 2010.

MORSE, J. M.; BOTTORFF, J. L.; HUTCHINSON, S. The phenomenology of comfort. **Journal of advanced nursing**, [S.l.], v.20, n.1, p.189–195, 1994.

MUR-ARTAL, R.; MONTIEL, J. M.; TARDÓ, J. D. Orb-slam: a versatile and accurate monocular slam system. **IEEE Transactions on Robotics**, [S.l.], v.31, p.1147–1163, 2015.

MUR-ARTAL, R.; TARDÓ, J. D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. **IEEE Transactions on Robotics**, [S.l.], v.33, n.5, p.1255–1262, 2017.

NARAYANAN, V. K.; SPALANZANI, A.; LUO, R. C.; BABEL, M. Analysis of an adaptive strategy for equitably approaching and joining human interactions. **2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)**, [S.l.], 2016.

NG, D. W.-K.; GOH, S. Y. Indirect Control of an Autonomous Wheelchair Using SSVEP BCI. **Journal of Robotics and Mechatronics**, [S.l.], v.32, n.4, p.761–767, 2020.

NIJBOER, F.; SELLERS, E.; MELLINGER, J.; JORDAN, M. A.; MATUZ, T.; FURDEA, A.; HALDER, S.; MOCHTY, U.; KRUSIENSKI, D.; VAUGHAN, T. et al. A P300-based brain–computer interface for people with amyotrophic lateral sclerosis. **Clinical neurophysiology**, [S.l.], v.119, n.8, p.1909–1916, 2008.

NISTÉR, D.; NARODITSKY, O.; BERGEN, J. Visual odometry. In: IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2004. CVPR 2004., 2004., 2004. **Proceedings...** [S.l.: s.n.], 2004. v.1, p.I–I.

OKWU, M. O.; TARTIBU, L. K.; AYOMOH, M.; IGHALO, D. Development of an Autonomous Wheelchair for The Disabled and Performance Analysis Using ANFIS Model. **ROBOMECH Journal**, [S.l.], 2020.

PATIL, A. **Nuric Wheelchair Simulation**. 2018. Accessed: 2022-06-4, https://github.com/patilnabhi/nuric_wheelchair_model_02.

PAVÓN-PULIDO, N.; BLASCO-GARCÍA, J.; LÓPEZ-RIQUELME, J.; FELIÚ-BATLLE, J.; OTERINO-BONO, R.; HERRERO, M. Juno project: deployment and validation of a low-cost cloud-based robotic platform for reliable smart navigation and natural interaction with humans in an elderly institution. **Sensors**, [S.l.], v.23, p.483, 2023.

PIMENTEL, F.; AQUINO, P. Performance evaluation of ROS local trajectory planning algorithms to social navigation. In: LATIN AMERICAN ROBOTICS SYMPOSIUM (LARS), 2019 BRAZILIAN SYMPOSIUM ON ROBOTICS (SBR) AND 2019 WORKSHOP ON ROBOTICS IN EDUCATION (WRE), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p.156–161.

PIMENTEL, F. d. A. M.; AQUINO-JR, P. T. Evaluation of ROS navigation stack for social navigation in simulated environments. **Journal of Intelligent & Robotic Systems**, [S.l.], v.102, n.4, p.87, 2021.

PINEAU, J.; WEST, R.; ATRASH, A.; VILLEMURE, J.; ROUTHIER, F. On the feasibility of using a standardized test for evaluating a speech-controlled smart wheelchair. **International Journal of Intelligent Control and Systems**, [S.l.], v.16, n.2, p.124–131, 2011.

PIRKER, W.; KATZENSCHLAGER, R. Gait disorders in adults and the elderly. **Wiener Klinische Wochenschrift**, [S.l.], v.129, n.3-4, p.81–95, 2017.

QUIGLEY, M.; CONLEY, K.; GERKEY, B.; FAUST, J.; FOOTE, T.; LEIBS, J.; WHEELER, R.; NG, A. Y. ROS: an open-source Robot Operating System. In: ICRA WORKSHOP ON OPEN SOURCE SOFTWARE, 2009. **Anais...** [S.l.: s.n.], 2009. v.3, n.3.2, p.5.

RATSAMEE, P.; MAE, Y.; OHARA, K.; KOJIMA, M.; ARAI, T. Social navigation model based on human intention analysis using face orientation. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 2013., 2013. **Anais...** [S.l.: s.n.], 2013. p.1682–1687.

REBECCHI, A.; FARINA, M.; ANDREONI, G.; CAPOLONGO, S.; CORNO, M.; PEREGO, P.; LETTIERI, E. Shaping accessible public spaces for visually impaired people. the budd-e research experience. **TECHNE - Journal of Technology for Architecture and Environment**, [S.l.], p.192–203, 2023.

RIEDL, M. O. A computational model and classification framework for social navigation. **Proceedings of the 6th International Conference on Intelligent User Interfaces**, [S.l.], 2001.

ROS Costmap. 2018. Accessed: 2024-06-30, http://wiki.ros.org/costmap_2d.

ROSENBERG, D. E.; BOMBARDIER, C. H.; ARTHURHOLT, S.; JENSEN, M. P.; MOTL, R. W. Self-reported depression and physical activity in adults with mobility impairments. **Archives of Physical Medicine and Rehabilitation**, [S.l.], v.94, n.4, p.731–736, 2013.

RÖSMANN, C.; FEITEN, W.; WÖSCH, T.; HOFFMANN, F.; BERTRAM, T. Efficient trajectory optimization using a sparse model. In: EUROPEAN CONFERENCE ON MOBILE ROBOTS, 2013., 2013. **Anais...** [S.l.: s.n.], 2013. p.138–143.

RUSU, R. B.; COUSINS, S. 3d is here: Point cloud library (pcl). In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2011., 2011. **Anais...** [S.l.: s.n.], 2011. p.1–4.

SALVINI, P.; PAEZ-GRANADOS, D.; BILLARD, A. Safety concerns emerging from robots navigating in crowded pedestrian areas. **International Journal of Social Robotics**, [S.l.], v.14, p.441–462, 2021.

SANTOS, A.; CUNHA, A.; MACEDO, N. Static-time extraction and analysis of the ros computation graph. **2019 Third IEEE International Conference on Robotic Computing (IRC)**, [S.l.], 2019.

SCALES, P.; AYCARD, O.; AUBERGÉ, V. Planning socially expressive mobile robot trajectories. **Sensors**, [S.l.], v.24, p.3533, 2024.

SEQUEIRA, J.; GAMEIRO, D. L. A probabilistic approach to rfid-based localization for human-robot interaction in social robotics. **Electronics**, [S.l.], v.6, p.32, 2017.

SHIOMI, M.; IIO, T.; KAMEI, K.; SHARMA, C.; HAGITA, N. Effectiveness of social behaviors for autonomous wheelchair robot to support elderly people in Japan. **PloS one**, [S.l.], v.10, n.5, p.e0128031, 2015.

SIMPSON, R.; LOPRESTI, E. F.; HAYASHI, S.; NOURBAKHSH, I.; MILLER, D. P. The smart wheelchair component system. **The Journal of Rehabilitation Research and Development**, [S.l.], v.41, p.429, 2004.

SLIMANE, N.; KHIREDINE, M. S.; CHAFAA, K. A metric approach for environments mapping. **2013 International Conference on Control, Decision and Information Technologies (CoDIT)**, [S.l.], 2013.

SMITH, G. L.; SCHMIDT, S. F.; MCGEE, L. A. **Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle**. [S.l.]: National Aeronautics and Space Administration, 1962. v.135.

SOLLEHUDIN, I.; HEERWAN, P. Speed estimation of the electric-powered wheelchair by proposing the state observer method based on experimental data. **SN Applied Sciences**, [S.l.], v.2, n.2, p.242, 2020.

SUWOYO, H. Problem solving path planning and path tracking in a 3 dof hexapod robot using the rrt* algorithm with path optimization and pose-to-pose. **Sinergi**, [S.l.], v.28, p.265, 2024.

TAJFEL, H. Experiments in intergroup discrimination. **Scientific american**, [S.l.], v.223, n.5, p.96–103, 1970.

TAKAYAMA, L.; DOOLEY, D.; JU, W. Expressing thought: improving robot readability with animation principles. In: HUMAN-ROBOT INTERACTION, 6., 2011. **Proceedings...** [S.l.: s.n.], 2011. p.69–76.

TANG, J.; LIU, Y.; HU, D.; ZHOU, Z. Towards BCI-actuated smart wheelchair system. **Biomedical engineering online**, [S.l.], v.17, n.1, p.111, 2018.

THRUN, S. Probabilistic robotics. **Communications of the ACM**, [S.l.], v.45, n.3, p.52–57, 2002.

TOMARI, R.; KOBAYASHI, Y.; KUNO, Y. Empirical framework for autonomous wheelchair systems in human-shared environments. In: IEEE INTERNATIONAL CONFERENCE ON MECHATRONICS AND AUTOMATION, 2012., 2012. **Anais...** [S.l.: s.n.], 2012. p.493–498.

TOMARI, R.; KOBAYASHI, Y.; KUNO, Y. Analysis of socially acceptable smart wheelchair navigation based on head cue information. **Procedia computer science**, [S.l.], v.42, p.198–205, 2014.

TOMOMI, N.; KANJI, T. Dictionary-based map compression using modified ransac map-matching. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND BIOMIMETICS, 2010., 2010. **Anais...** [S.l.: s.n.], 2010. p.980–985.

TRAUTMAN, P.; MA, J.; MURRAY, R. M.; KRAUSE, A. Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation. **The International Journal of Robotics Research**, [S.l.], v.34, n.3, p.335–356, 2015.

TRZASKOMA, Z.; MOLIĆ, B.; MORGULEC-ADAMOWICZ, N.; TRUSZCZYŃSKA-BASZAK, A.; OGONOWSKA-SŁODOWNIK, A.; MARSZAŁEK, J.; STĘPIEŃ, A.; KRAWCZYK, M.; WÓJCIK, A. Identification of the forces developed by upper limbs in various forms of human physical activity and in manual techniques used by physiotherapists – a brief review. **Advances in Rehabilitation**, [S.l.], v.31, p.59–70, 2017.

URDIALES, C.; PONCELA, A.; SANCHEZ-TATO, I.; GALLUPPI, F.; OLIVETTI, M.; SANDOVAL, F. Efficiency based reactive shared control for collaborative human/robot navigation. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 2007., 2007. **Anais...** [S.l.: s.n.], 2007. p.3586–3591.

VANCINI, R. L.; GOMES, A.; OLIVEIRA, H.; DE LIRA, C. A. B.; RUFO-TAVARES, W.; ANDRADE, M.; SARRO, K. J.; SAMPAIO, M.; VIANA, R. B.; NIKOLAIDIS, P. T. et al. Quality of life, depression, anxiety symptoms and mood state of wheelchair athletes and non-athletes: a preliminary study. **Frontiers in psychology**, [S.l.], v.10, p.1848, 2019.

WALFORD, S. L.; REQUEJO, P. S.; MULROY, S. J.; NEPTUNE, R. R. Predictors of shoulder pain in manual wheelchair users. **Clinical Biomechanics**, [S.l.], v.65, p.1–12, 2019.

WANG, B.; CUI, S.; ZHAO, L.; YANG, G.; XU, X. Mobile robot map building based on grid arrangement. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND COMPUTATIONAL INTELLIGENCE, 2009., 2009. **Anais...** [S.l.: s.n.], 2009. v.2, p.288–291.

WATANABE, A.; IKEDA, T.; MORALES, Y.; SHINOZAWA, K.; MIYASHITA, T.; HAGITA, N. Communicating robotic navigational intentions. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS), 2015., 2015. **Anais...** [S.l.: s.n.], 2015. p.5763–5769.

WHILE, A.; MARVIN, S.; KOVACIC, M. Urban robotic experimentation: san francisco, tokyo and dubai. **Urban Studies**, [S.l.], v.58, p.769–786, 2020.

WHO, W. H. O. **Assistive technology**. 2018. Disponível em: <<https://www.who.int/news-room/fact-sheets/detail/assistive-technology>>. Acesso em: 10-12-2020.

XIONG, M.; HOTTER, R.; NADIN, D.; PATEL, J.; TARTAKOVSKY, S.; WANG, Y.; PATEL, H.; AXON, C.; BOSILJEVAC, H.; BRANDENBERGER, A. et al. A Low-Cost, Semi-Autonomous Wheelchair Controlled by Motor Imagery and Jaw Muscle Activation. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN AND CYBERNETICS (SMC), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p.2180–2185.

YASHODA, H.; PIUMAL, A.; POLGAHAPITIYA, P.; MUBEEN, M.; MUTHUGALA, M.; JAYASEKARA, A. Design and development of a smart wheelchair with multiple control interfaces. In: MORATUWA ENGINEERING RESEARCH CONFERENCE (MERCEN), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p.324–329.

YUAN, R. X. Series of advanced slam processes and applications. **Applied and Computational Engineering**, [S.l.], v.41, p.242–250, 2024.

YULIANA, S.; RIYADI, M. A. Implementation of brain computer interface (bci) as a smart wheelchair motion commands. **Journal of Biomedical Science and Bioengineering**, [S.l.], v.2, p.14–17, 2022.

ZAL, F.; CHEN, T.; CHI, S.; CHEN, K. Fuzzy controller based subsumption behavior architecture for autonomous robotic wheelchair. **2013 International Conference on Advanced Robotics and Intelligent Systems**, [S.l.], 2013.

ZHANG, H.; WANG, L.; CHEN, A. Path planning for the mobile robot: a review. **Symmetry**, [S.l.], v.10, p.450, 2018.

ZHANG, S.; ZHAO, T. Mobile robot path planning in 2d space: a survey. **Highlights in Science, Engineering and Technology**, [S.l.], v.16, p.279–289, 2022.

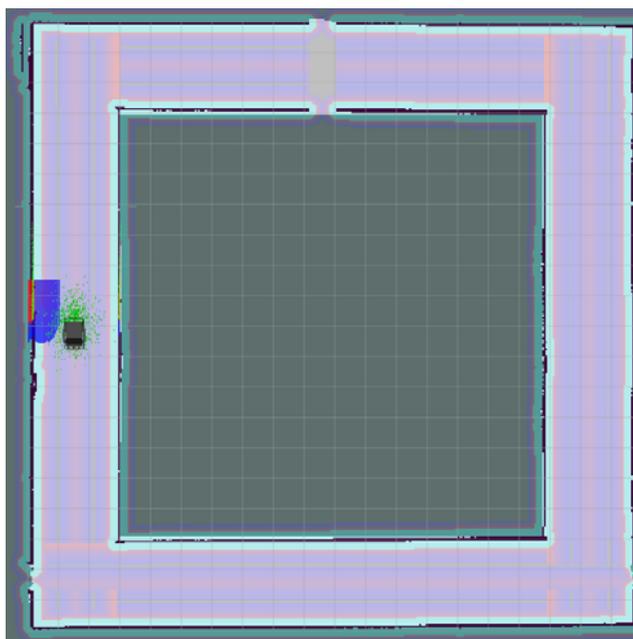
ZHANG, X.; LI, J.; ZHANG, R.; LIU, T. A brain-controlled and user-centered intelligent wheelchair: a feasibility study. **Sensors**, [S.l.], v.24, p.3000, 2024.

ZÖRNER, S.; ARTS, E.; VASILJEVIC, B.; SRIVASTAVA, A.; SCHMALZL, F.; MIR, G.; BHATIA, K.; STRAHL, E.; PETERS, A.; ALPAY, T.; WERMTER, S. An immersive investment game to study human-robot trust. **Frontiers in Robotics and AI**, [S.l.], v.8, 2021.

Apêndices

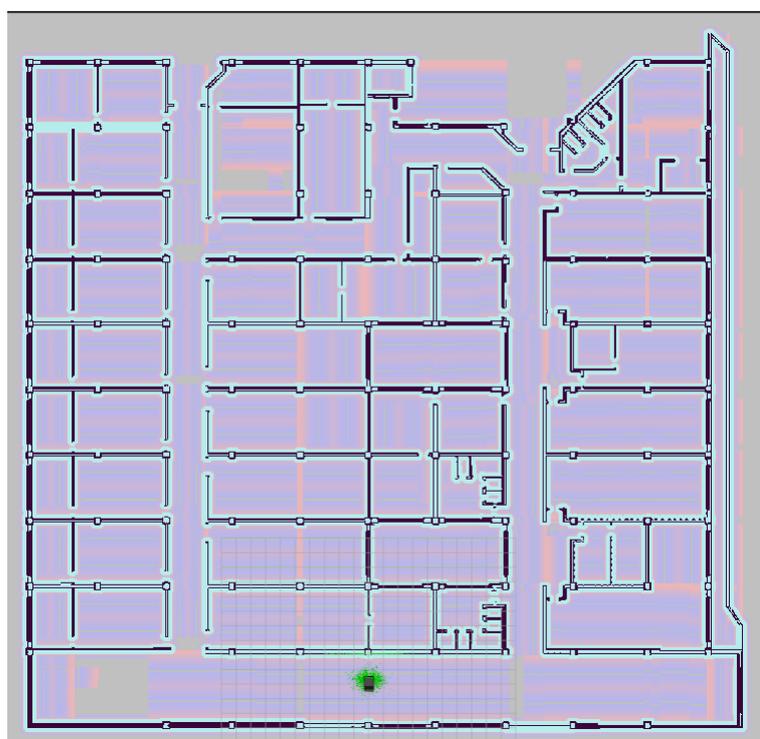
APÊNDICE A – Imagens dos mapas de custos gerados

Figura 60 – Mapas de custo no cenário de corredores.



Fonte: Imagem gerada pelo autor.

Figura 61 – Mapas de custo no cenário do terceiro andar.



Fonte: Imagem gerada pelo autor.

APÊNDICE B – Arquivos de Parâmetros dos Mapas de Custo

1. *costmap_common_params.yaml*

```
1  obstacle_range: 3.0
2  raytrace_range: 3.5
3
4  footprint: [[-0.7, -0.3], [-0.7, 0.3], [0.27, 0.3], [0.27, -0.3]]
5
6  inflation_layer:
7    inflation_radius: 0.5
8    cost_scaling_factor: 10.0
9
10 obstacle_layer:
11   observation_sources: scan
12   scan: {sensor_frame: laser_joint, data_type: LaserScan, topic:
13     ↪ scan, marking: true, clearing: true}
14
15 comfort_layer:
16   observation_sources: scan
17   activate_layer: true
18
19 vision_layer:
20   activate_layer: true
21
22 social_navigation_layer:
23   activate_layer: true
```

2. *local_costmap_params.yaml*

```
1  local_costmap:
2    global_frame: odom
3    robot_base_frame: base_footprint
4
5    update_frequency: 5.0
```

```

6   publish_frequency: 5.0
7   transform_tolerance: 0.5
8
9   rolling_window: true
10  width: 3
11  height: 3
12  resolution: 0.05
13
14  plugins:
15    - {name: obstacle_layer, type: "costmap_2d::ObstacleLayer"}
16    - {name: vision_layer, type: "vision_layer::VisionLayer"}
17    - {name: inflation_layer, type:
18      ↪ "costmap_2d::InflationLayer"}
19    - {name: social_navigation_layer, type:
20      ↪ "social_navigation_layers::ProxemicLayer"}

```

3. *global_costmap_params.yaml*

```

1  global_costmap:
2    global_frame: map
3    robot_base_frame: base_footprint
4
5    update_frequency: 5.0
6    publish_frequency: 5.0
7    transform_tolerance: 0.5
8
9    plugins:
10     - {name: static_layer, type: "costmap_2d::StaticLayer"}
11     - {name: comfort_layer, type:
12       ↪ "comfort_layer::ComfortLayer"}
13     - {name: obstacle_layer, type: "costmap_2d::ObstacleLayer"}
14     - {name: inflation_layer, type:
15       ↪ "costmap_2d::InflationLayer"}
16     - {name: social_navigation_layer, type:
17       ↪ "social_navigation_layers::ProxemicLayer"}

```