

CENTRO UNIVERSITÁRIO FEI
DANILO HERNANI PERICO

**NAVEGAÇÃO GUIADA DE UM AGENTE AUTÔNOMO POR MEIO DE RELAÇÕES
PROJETIVAS OBTIDAS POR MÚLTIPLAS PERSPECTIVAS ESPACIAIS**

São Bernardo do Campo

2017

DANILO HERNANI PERICO

**NAVEGAÇÃO GUIADA DE UM AGENTE AUTÔNOMO POR MEIO DE RELAÇÕES
PROJETIVAS OBTIDAS POR MÚLTIPLAS PERSPECTIVAS ESPACIAIS**

Tese de Doutorado apresentada ao Centro Universitário FEI para obtenção do título de Doutor em Engenharia Elétrica, orientada pelo Prof. Dr. Reinaldo Augusto da Costa Bianchi.

São Bernardo do Campo

2017

Perico, Danilo Hernani.

Navegação guiada de um agente autônomo por meio de relações
projetivas obtidas por múltiplas perspectivas espaciais / Danilo Hernani
Perico. São Bernardo do Campo, 2017.

158 p. : il.

Tese - Centro Universitário FEI.

Orientador: Prof. Dr. Reinaldo Augusto da Costa Bianchi.

1. Navegação Guiada. 2. Filtro de Partículas. 3. Raciocínio Espacial
Qualitativo. 4. Agentes Autônomos. I. Bianchi, Reinaldo Augusto da
Costa, orient. II. Título.

Aluno: Danilo Hernani Perico

Matrícula: 513302-0

Título do Trabalho: Navegação guiada de um agente autônomo por meio de relações projetivas obtidas por múltiplas perspectivas espaciais.

Área de Concentração: Inteligência Artificial Aplicada à Automação e Robótica

Orientador: Prof. Dr. Reinaldo Augusto da Costa Bianchi

Data da realização da defesa: 15/12/2017

ORIGINAL ASSINADA

Avaliação da Banca Examinadora

São Bernardo do Campo, 15 / 12 / 2017.

MEMBROS DA BANCA EXAMINADORA

Prof. Dr. Reinaldo Augusto da Costa Bianchi

Ass.: _____

Prof. Dr. Flavio Tonidandel

Ass.: _____

Prof.^a Dr.^a Esther Luna Colombini

Ass.: _____

Prof. Dr. Fábio Gagliardi Cozman

Ass.: _____

Prof.^a Dr.^a Roseli Aparecida Francelin Romero

Ass.: _____

A Banca Examinadora acima-assinada atribuiu ao aluno o seguinte:

APROVADO

REPROVADO

VERSÃO FINAL DA TESE

**ENDOSSO DO ORIENTADOR APÓS A INCLUSÃO DAS
RECOMENDAÇÕES DA BANCA EXAMINADORA**

Aprovação do Coordenador do Programa de Pós-graduação

_____ Prof. Dr. Carlos Eduardo Thomaz

A Deus, aos meus pais, à minha esposa Lucivânia e aos meus filhos Mariana e Miguel.

AGRADECIMENTOS

Agradeço a Deus, por tudo.

A toda minha família, em especial aos meus pais, Valter e Ana Clara, por sempre terem me apoiado em todas as minhas decisões; a minha esposa, Lucivânia, por todo amor, incentivo, carinho e por ter compreendido as minhas ausências durante a realização deste trabalho, dedicou-me esta conquista.

Ao meu orientador, Prof. Dr. Reinaldo Augusto da Costa Bianchi, pela amizade, pela paciência, pelas sugestões, pelos conselhos, pela motivação e por todas as oportunidades que tem me dado.

Aos professores Dr. Plinio Thomaz Aquino Júnior, Dr. Carlos Eduardo Thomaz e Dr. Paulo Sérgio Rodrigues que, no decorrer de meus estudos, compartilharam comigo um bem de tão inestimável valor: o conhecimento. Agradeço, em particular, ao Prof. Dr. Paulo E. Santos, pela atenção que dedicou e tem ainda dedicado a mim; e ao Prof. Dr. Flavio Tonidandel pela confiança, pelas conversas e pelas oportunidades de aprendizado que sempre me proporciona.

Aos amigos, membros da equipe RoboFEI de Futebol de Robôs do Centro Universitário FEI: Isaac J. da Silva, Thiago P. D. Homem, Claudio de O. Vilão Jr., Vinícius N. Ferreira e Aislan C. Almeida, que contribuíram de forma decisiva na realização deste trabalho.

Agradeço ao Centro Universitário FEI por fornecer toda a infraestrutura necessária, como equipamentos, computadores e laboratórios.

Agradeço à CAPES pela bolsa de estudos de doutorado que me foi concedida, a qual foi determinante para o bom desenvolvimento de minha pesquisa.

“Navegar é preciso, viver não é preciso.”

Fernando Pessoa

RESUMO

Navegação é uma habilidade essencial para que agentes móveis possam ser completamente autônomos e capazes de desempenhar tarefas complexas que dependam de diversas competências cognitivas. Contudo, o problema de navegação em agentes que não podem perceber o mundo e não tem modelo de movimento completamente definido foi, até agora, pouco estudado. Uma maneira de tornar possível a navegação de um agente sem estas informações é a utilização da navegação guiada, na qual outros agentes autônomos, dotados de percepção, podem, por meio da união de suas diferentes perspectivas, inferir de forma recursiva a localização e as ações que o agente guiado deve tomar para realizar um trajeto de sua origem até um destino. Devido, ainda, a falta de conhecimento que pode existir sobre as características físicas e sensoriais do agente que será conduzido, um sistema de navegação guiada deve ser capaz de trabalhar com um nível de abstração que possibilite o uso de um modelo genérico de movimento e que transmita instruções de ação em alto nível, permitindo sua interpretação por qualquer tipo de agente autônomo. Esta abstração pode ser alcançada pelo uso de instruções baseadas em proposições espaciais projetivas, como direita, esquerda, frente e atrás. Assim, este trabalho apresenta dois modelos híbridos, qualitativo-probabilístico, para se guiar um agente sem percepção e sem modelo de movimento definido por meio do uso de proposições projetivas. Estes dois modelos utilizam os fundamentos encontrados no Filtro de Partículas, utilizado no contexto de navegação, e a representação qualitativa formalizada por meio da linguagem de restrição para relações de direção relativas StarVars. Os dois métodos propostos foram sistematicamente avaliados por experimentos realizados em um simulador de robôs humanoides e em duas provas de conceito, uma para cada método, realizadas com robôs humanoides reais. Os resultados obtidos demonstraram que os dois modelos de navegação guiada propostos foram bem sucedidos.

Palavras-chave: Navegação Guiada. Filtro de Partículas. Raciocínio Espacial Qualitativo. Agentes Autônomos

ABSTRACT

Navigation is an essential ability for mobile agents to be completely autonomous and able to perform complex tasks that depend on various cognitive skills. However, the problem of navigation in agents that can not perceive the world and does not have a fully defined model of movement has so far been little studied. One way to make an agent without these information to navigate is to use guided navigation, in which other autonomous agents, endowed with perception, can, through the union of their different perspectives, infer recursively the localization and the actions that the guided agent must take to make a route from its origin to a destination. Due to the lack of knowledge that can exist about the physical and sensorial characteristics of the agent to be conducted, a guided navigation system must be able to work with a level of abstraction that allows the use of a generic model of movement and that transmits instructions of action at high level, that can be interpreted by any type of autonomous agent. This abstraction can be achieved by using instructions based on projective spatial prepositions such as right, left, front, and back. Thus, this work presents two hybrid models, qualitative-probabilistic, to guide an agent without perception and without model of movement defined through the use of projective propositions. These two models use the foundations found in the Particle Filter, used in the context of navigation, and the qualitative representation formalized by means of the restriction language for relative relations of direction StarVars. The two proposed methods were systematically evaluated by experiments carried out on a 2D humanoid robot simulator and two proofs of concept, one for each method, were performed with real humanoid robots. The results obtained demonstrated that the two guided navigation models proposed were successful.

Keywords: Guided Navigation. Particle Filter. Qualitative Spatial Reasoning. Autonomous Agents

LISTA DE ILUSTRAÇÕES

Figura 1 – Modelo Oculto de Markov – <i>Hidden Markov Model</i> (HMM) que caracteriza a evolução das ações, estados e observações	39
Figura 2 – Modelo gráfico do problema de Localização e Mapeamento Simultâneos – <i>Simultaneous Localization and Mapping</i> (SLAM) <i>online</i>	44
Figura 3 – Política de ações para um MDP	46
Figura 4 – Sistemas de referência espaciais para relações de direção.....	51
Figura 5 – Relações encontradas no cálculo Flip-Flop	52
Figura 6 – Representação do Cálculo de Cruz Dupla – <i>Double Cross Calculus</i> (DCC) . .	52
Figura 7 – Relação de orientação entre dois dipolos.....	53
Figura 8 – Pontos orientados com m igual a 3 e 4	54
Figura 9 – A composição das relações $A_4 \angle^1 B$ e $B_4 \angle^9 C_3$ pode ter como uma solução $A_4 \angle^7_{13} C$ na OPRA ₄	56
Figura 10 – ST AR ₈ : $A(\mathbf{7})B \wedge B(\mathbf{3})A$	57
Figura 11 – StarVars ₈ : $\theta_A = \mathbf{2}\eta = \mathbf{90}^\circ$	59
Figura 12 – Representação gráfica de uma possível resposta obtida pelo StarVars ₈ , dada a entrada φ	61
Figura 13 – Aibo durante o experimento	64
Figura 14 – Duas perspectivas do Aibo durante o experimento realizado: $t\mathbf{1}$ e $t\mathbf{2}$	64
Figura 15 – Duas embarcações a motor (MVs).....	66
Figura 16 – Representação da OPRA ₄ nas embarcações e o comportamento idealizado para a regra exibida na Figura 15.....	66
Figura 17 – Sistema de transição completo para a regra exibida na Figura 16	67
Figura 18 – Diagrama Conceitual de Vizinhanças do RCC8	68
Figura 19 – Cálculo de Oclusão de Regiões – <i>Region Occlusion Calculus</i> (Cálculo de Oclusão de Regiões – <i>Region Occlusion Calculus</i> (ROC))	69
Figura 20 – Representação das relações básicas do PQRS	70
Figura 21 – Representação do mapa qualitativo com cinco regiões distintas. O ponto fixo de luz é representado por L . As linhas entre L , o objeto e a sua sombra definem fronteiras entre as regiões	70
Figura 22 – Representação do mapa qualitativo composto pela divisão do espaço ao redor dos objetos r , b e g : $\mathbf{28}$ regiões distintas foram formadas.	73
Figura 23 – Imagem do experimento realizado com o robô PeopleBot.	73
Figura 24 – Vista superior do corte no equador da Egosfera do robô.	74
Figura 25 – Simulador utilizado.....	75
Figura 26 – O robô aéreo e o robô terrestre na maquete de desastre.....	76
Figura 27 – Agentes autônomos utilizados durante a proposta.....	81

Figura 28 – União das seções angulares da $StarVars_m$ para formar as regiões nomeadas com posições projetivas.	83
Figura 29 – Relacionamento entre o objetivo e uma partícula que representa uma possível postura do agente guiado.....	87
Figura 30 – Modelo de mundo retornado pelo $StarVars_8$	89
Figura 31 – Modelo de mundo ψ retornado pelo $StarVars_8$ com $\varepsilon = -0.3$	90
Figura 32 – Modelo de mundo retornado pelo $StarVars_8$ com partículas representando as posições localizações do objetivo e do agente guiado.....	91
Figura 33 – Modelo qualitativo do movimento de translação aplicado às partículas que representam o agente guiado: andar para frente até trocar de região	93
Figura 34 – Exemplo de dessincronização entre o agente guiado real e as partículas que o representam.	98
Figura 35 – Exemplo do modelo desenvolvido para desconsiderar as fronteiras pares ou ímpares em cada agente observador.....	99
Figura 36 – Direções numéricas obtidas pelo agente A , que tem orientação 45° , com relação às entidades B e C	101
Figura 37 – Exemplo da obtenção dos ângulos internos dos triângulos formados pelos agentes espaciais	103
Figura 38 – Exemplos de robôs humanoides	113
Figura 39 – Robôs humanoides do time RoboFEI-HT que foram utilizados durante a prova de conceito com suas respectivas cores.....	114
Figura 40 – Arquitetura de software dos robôs humanoides: Arquitetura em Cruz.....	115
Figura 41 – <i>RoboFEI-HT Simulator</i> exibindo três robôs de cada time.	116
Figura 42 – Experimento para medir a fidelidade do simulador com relação à uma situação real	117
Figura 43 – Caminhos percorridos pelos robôs reais e virtuais (sem e com erros)	118
Figura 44 – Exemplo do domínio utilizado nos experimentos	120
Figura 45 – Exemplo de caminhos percorridos utilizando o $StarVars_{16}$ com instrução única para realizar a navegação guiada.....	127
Figura 46 – Exemplo de caminhos percorridos utilizando o $StarVars_{16}$ atualizado a cada troca de região para realizar a navegação guiada	128
Figura 47 – Exemplo de caminhos percorridos utilizando o $QPFGN_{16}$ para realizar a navegação guiada	129
Figura 48 – Exemplo de caminhos percorridos utilizando o $NPFGNQI_{16}^3$ para realizar a navegação guiada	130
Figura 49 – Processo de visão computacional utilizado para identificação dos objetos de interesse.....	132
Figura 50 – Configuração espacial inicial utilizada na prova de conceito.....	133
Figura 51 – Movimento do robô humanoide real sendo guiado pelo $QPFGN_8$	134

Figura 52 – Prova de conceito: predição e atualização das partíulas no QPFGN ₈	135
Figura 53 – Caminhos percorridos pelo robô guiado com o QPFGN ₈	136
Figura 54 – Prova de conceito: movimento do robô humanoide real sendo guiado pelo NPFNGQI ₁₆	137
Figura 55 – Prova de conceito: predição e atualização das partíulas no NPFNGQI ₁₆ ...	138
Figura 56 – Caminhos percorridos pelo robô guiado com o NPFNGQI ₁₆	139

LISTA DE TABELAS

Tabela 1 – Principais formalismos do QSR citados neste trabalho	50
Tabela 2 – Resultados da autolocalização, utilizando também a autocalibração de Th . .	71
Tabela 3 – Porcentagem de acertos do método puramente qualitativo (PQRS em comparação com o método qualitativo- probabilístico (PQRS + filtro de Bayes)) . .	71
Tabela 4 – Características dos Robôs	113
Tabela 5 – Resultados dos experimentos realizados para demonstrar a fidelidade do simulador RoboFEI-HT quando comparado ao robô humanoide real.....	119
Tabela 6 – Sucesso em alcançar a região do objetivo.....	123
Tabela 7 – Resultados dos experimentos bem sucedidos considerando que a orientação do agente guiado era conhecida	124
Tabela 8 – Resultados dos experimentos bem sucedidos considerando que a orientação do agente guiado era desconhecida.....	125
Tabela 9 – Dados dos caminhos percorridos	131

LISTA DE ALGORITMOS

Algoritmo 1 – Forma geral do Filtro Bayesiano.....	39
Algoritmo 2 – Forma geral do Filtro de Partículas: caso específico do Filtro Bayesiano com amostragem por importância.....	41
Algoritmo 3 – Localização de Monte Carlo.	42
Algoritmo 4 – Verificando se as relações entre os pontos orientados formam uma com- posição válida da Álgebra do Ponto Orientado – <i>Oriented Point Algebra</i> (OPRA _m)	55
Algoritmo 5 – Forma geral da Navegação Guiada.	85
Algoritmo 6 – mapeamento no Filtro de Partículas Qualitativo para Navegação Guiada – <i>Qualitative Particle Filter for Guided Navigation</i> (QPFGN _m).....	88
Algoritmo 7 – predição no QPFGN _m	94
Algoritmo 8 – atualização no QPFGN _m	95
Algoritmo 9 – mapeamento no Filtro de Partículas Numérico para Navegação Guiada com Instruções Qualitativas – <i>Numerical Particle Filter for Guided Navi- gation with Qualitative Instructions</i> (NPFNGNQI _m ^t).....	102
Algoritmo 10 – triangulação – Função que realiza a triangulação.	104
Algoritmo 11 – obtençãoCoordenadasEntidades – Função para obtenção das co- ordenadas das entidades.....	105
Algoritmo 12 – predição no NPFNGNQI _m ^t	106
Algoritmo 13 – atualização no NPFNGNQI _m ^t	107

LISTA DE ABREVIATURAS

AGVs	Veículos Guiados Automaticamente – <i>Automated Guided Vehicles</i> .
CDC	Cálculo de Direções Cardeais – <i>Cardinal Direction Calculus</i> .
CSP	Problemas de Satisfação de Restrições – <i>Constraint Satisfaction Problems</i> .
DCC	Cálculo de Cruz Dupla – <i>Double Cross Calculus</i> .
DNN	Rede Neural Profunda – <i>Deep Neural Network</i> .
DRA	Cálculo Dipolo – <i>Dipole Calculus</i> .
EKF	Filtro de Kalman Estendido – <i>Extended Kalman Filter</i> .
EOPRA _m	Álgebra Relacional do Ponto Elevado Orientado – <i>Elevated Oriented Point Relation Algebra</i> .
GIS	Sistemas de Informação Geográfica – <i>Geographic Information Systems</i> .
HMM	Modelo Oculto de Markov – <i>Hidden Markov Model</i> .
IMU	Unidade de Medida Inercial – <i>Inertial Measurement Unit</i> .
JEPD	Conjuntamente Exaustivas e Mutuamente Excludentes – <i>Jointly Exhaustive and Pairwise Disjoint</i> .
KF	Filtro de Kalman – <i>Kalman Filter</i> .
LP	Programação Linear – <i>Linear Programming</i> .
MCL	Localização de Monte Carlo – <i>Monte Carlo Localization</i> .
MDP	Processo de Decisão de Markov – <i>Markov Decision Process</i> .
NPFGNQI _m ^r	Filtro de Partículas Numérico para Navegação Guiada com Instruções Qualitativas – <i>Numerical Particle Filter for Guided Navigation with Qualitative Instructions</i> .
OPRA _m	Álgebra do Ponto Orientado – <i>Oriented Point Algebra</i> .
POMDP	Processo de Decisão de Markov Parcialmente Observável – <i>Partially Observable Markov Decision Process</i> .
PQRS	Relações Qualitativas Percebíveis sobre Sombras – <i>Perceptual Qualitative Relations about Shadows</i> .
QPFGN _m	Filtro de Partículas Qualitativo para Navegação Guiada – <i>Qualitative Particle Filter for Guided Navigation</i> .
QSR	Raciocínio Espacial Qualitativo – <i>Qualitative Spatial Reasoning</i> .
RCC	Cálculo de Conexão de Regiões – <i>Region Connection Calculus</i> .
ROC	Cálculo de Oclusão de Regiões – <i>Region Occlusion Calculus</i> .
SLAM	Localização e Mapeamento Simultâneos – <i>Simultaneous Localization and Mapping</i> .
ST AR _m	Cálculo Estrela – <i>Star Calculus</i> .
SV _m	StarVars.

TPCC Cálculo de Configuração do Ponto Ternário – *Ternary Point Configuration Calculus*.

UKF Filtro de Kalman Unscented – *Unscented Kalman Filter*.

SUMÁRIO

1	INTRODUÇÃO	29
1.1	JUSTIFICATIVA	30
1.2	OBJETIVO	30
1.3	CONTRIBUIÇÕES	31
1.4	ORGANIZAÇÃO DO TRABALHO	32
2	CONCEITOS FUNDAMENTAIS	33
2.1	NAVEGAÇÃO DE AGENTES MÓVEIS AUTÔNOMOS.....	33
2.1.1	Navegação Robótica	33
2.1.1.1	Probabilidade	34
2.1.1.1.1	<i>Conceitos Básicos de Probabilidade</i>	35
2.1.1.1.2	<i>Regra de Bayes</i>	36
2.1.1.1.3	<i>Probabilidade Aplicada à Navegação Robótica</i>	37
2.1.1.2	Localização	38
2.1.1.2.1	<i>Filtro Bayesiano</i>	38
2.1.1.2.2	<i>Filtro de Partículas</i>	40
2.1.1.2.3	<i>Localização de Monte Carlo</i>	41
2.1.1.3	Mapeamento	43
2.1.1.3.1	<i>Localização e Mapeamento Simultâneos</i>	43
2.1.1.4	Planejamento de Movimento	44
2.1.1.4.1	<i>Processo de Decisão de Markov</i>	45
2.1.2	Navegação Guiada	46
2.1.3	Navegação Humana	47
2.2	RACIOCÍNIO ESPACIAL QUALITATIVO.....	48
2.2.1	Raciocínio Sobre Direção Relativa	50
2.2.1.1	Cálculos Flip-Flop e LR	51
2.2.1.2	Cálculo da Cruz Dupla – DCC	51
2.2.1.3	Cálculo Dipolo – DRA	53
2.2.1.4	A Álgebra do Ponto Orientado – OPRA_m	53
2.2.1.4.1	<i>Inconvenientes da OPRA_m</i>	56
2.2.1.5	O Cálculo StarVars_m	57
2.2.1.5.1	<i>O cálculo de direção absoluta ST AR_m</i>	57
2.2.1.5.2	<i>O StarVars_m</i>	58
2.2.1.5.3	<i>Exemplo: Inferência de Orientações Utilizando o StarVars_s</i>	60
2.2.1.6	Uso de Referências Espaciais Linguísticas nos Cálculos sobre Direções Relativas	61
2.3	COMENTÁRIOS	62
3	TRABALHOS CORRELATOS	63

3.1	NAVEGAÇÃO DE AGENTES AUTÔNOMOS POR MEIO DE ESTRATÉGIAS PURAMENTE QUALITATIVAS.....	63
3.1.1	Robô Quadrúpede Utilizando a OPRA_m	63
3.1.2	Formalizando as Regras de Prevenção de Colisões da Navegação Marítima por meio da OPRA₄	65
3.2	LOCALIZAÇÃO DE AGENTES AUTÔNOMOS COM ESTRATÉGIAS HÍBRIDAS: RACIOCÍNIO ESPACIAL QUALITATIVO PROBABILÍSTICO . .	67
3.2.1	Filtro de Bayes Aplicado às Relações Qualitativas Percebíveis sobre Sombras para Autocalibração e Autocalibração de Visão em Robôs Móveis . .	68
3.2.2	Localização Probabilística em um Mapa Qualitativo Baseado em Oclusões	72
3.2.3	Localização de Monte Carlo Baseada em Visão com Abordagem Qualitativa Durante a Fase de Atualização	74
3.3	NAVEGAÇÃO GUIADA DE UM AGENTE AUTÔNOMO	75
3.4	COMENTÁRIOS.....	76
4	NAVEGAÇÃO GUIADA COM RELAÇÕES PROJETIVAS OBTIDAS POR MÚLTIPLAS PERSPECTIVAS ESPACIAIS	79
4.1	DEFINIÇÃO DO PROBLEMA	79
4.2	BASES COMUNS ÀS PROPOSTAS	80
4.2.1	Hipótese Central e Considerações	81
4.2.2	Definições do Agente Guiado	82
4.2.2.1	<i>Regiões Espaciais Linguísticas Baseadas em Relações Projetivas</i>	82
4.2.2.2	<i>Modelo de Movimento Genérico e Parcial</i>	82
4.2.3	Definições dos Agentes Observadores	83
4.2.4	Algoritmo Geral para Navegação Guiada	84
4.2.5	Seleção das Ações: Planejamento de Movimento	86
4.3	FILTRO DE PARTÍCULAS QUALITATIVO PARA NAVEGAÇÃO GUIADA	87
4.3.1	Mapeamento e Localização	88
4.3.1.1	<i>Predição: Inserção do Modelo Qualitativo de Movimento de Translação . . .</i>	92
4.3.1.2	<i>Atualização: Inserção do Modelo de Sensores Qualitativo</i>	94
4.3.2	Definições do Objetivo a ser Alcançado pelo Agente Guiado	96
4.3.3	Lidando com a Incerteza na Observação da Condição de Parada do Agente Guiado	96
4.4	FILTRO DE PARTÍCULAS NUMÉRICO PARA NAVEGAÇÃO GUIADA COM INSTRUÇÕES QUALITATIVAS	100
4.4.1	Localização e Mapeamento	100
4.4.1.1	<i>Predição: Inserção do Modelo Aleatório de Movimento de Translação</i>	104
4.4.1.2	<i>Atualização: Inserção do Modelo de Sensores Numérico</i>	105
4.4.2	Definições do Objetivo a ser Alcançado pelo Agente Guiado	107
4.5	COMENTÁRIOS.....	108

5	EXPERIMENTOS	111
5.1	DOMÍNIO.....	111
5.1.1	RoboCup	111
5.1.2	Robôs Humanoides	112
5.2	EXPERIMENTOS SIMULADOS.....	114
5.2.1	O Simulador RoboFEI-HT	114
5.2.2	Descrição dos Experimentos	119
5.2.2.1	<i>Configuração dos Experimentos Qualitativos Puros</i>	121
5.2.2.2	<i>Configuração dos Experimentos com o QPFGN_m</i>	121
5.2.2.3	<i>Configuração dos Experimentos com o NPFGNQI_m^r</i>	122
5.2.3	Resultados dos Experimentos e Análises	122
5.3	PROVA DE CONCEITO COM ROBÔS REAIS	131
5.4	COMENTÁRIOS	132
6	CONCLUSÃO E TRABALHOS FUTUROS	141
6.1	TRABALHOS FUTUROS	142
	REFERÊNCIAS	143
	ANEXO A – Trabalhos Publicados	153
A.1	PERIÓDICOS	154
A.2	CAPÍTULOS DE LIVROS.....	154
A.3	ARTIGOS PUBLICADOS EM CONGRESSOS INTERNACIONAIS.....	154
A.4	ARTIGOS PUBLICADOS EM CONGRESSOS NACIONAIS	155
	ANEXO B – Prêmios e Títulos	157

1 INTRODUÇÃO

Navegação é a área que estuda a habilidade de um agente em se localizar e manter o trajeto de um lugar de origem até um destino. A tarefa realizada pela navegação é complexa, pois envolve diversas competências cognitivas (COLOMBO et al., 2017; HAM; CLAESSEN, 2017), como elaboração ou interpretação de mapas, localização e tomadas de decisão sobre quais ações escolher para se manter no caminho correto.

Presente na maioria dos seres vivos, a navegação pode ser encontrada desde em tarefas mais simples, como aprender o caminho de volta de algum lugar dado o caminho de ida, até tarefas mais complexas que envolvem habilidades sofisticadas de raciocínio sobre relações espaciais (NEHMZOW, 2003). De qualquer forma, a navegação é essencial para os agentes móveis autônomos, sejam humanos, animais ou robôs, pois é por meio dela que os agentes adquirem, principalmente, as capacidades de realizar tarefas relacionadas a lugares específicos e, também, de explorar o ambiente podendo voltar para um lugar previamente conhecido (MILFORD; SCHULZ, 2014).

Apesar de existirem diversos trabalhos sobre navegação nas literaturas cognitiva e robótica, pouco se estudou sobre os casos em que o agente tem que navegar sem possuir sensores para perceber o mundo e sem ter um modelo de movimento definido, que são dois requisitos necessários para se realizar navegação em agentes autônomos móveis.

Uma maneira de tornar possível a navegação de um agente sem percepção e sem modelo de movimento específico é a utilização da navegação guiada, na qual outros agentes autônomos podem, por meio da união de suas diferentes perspectivas, inferir a localização (posição e orientação), o rastreamento e as ações que o agente guiado deve tomar para realizar um trajeto de sua origem até um destino estabelecido.

Para ser melhor aproveitado e, portanto, mais eficaz, um sistema de navegação guiada deve ser capaz de guiar qualquer tipo de agente autônomo móvel, desde que estes agentes sejam capazes de seguir instruções sobre direções, como são os casos dos seres humanos e dos robôs autônomos móveis. Logo, um sistema de navegação guiada precisa conseguir trabalhar com um nível de abstração que possibilite o uso de um modelo genérico de movimento e que transmita instruções em alto nível, que possam ser interpretadas por humanos e robôs.

Neste trabalho, o sistema de navegação guiada será constituído por agentes autônomos não-humanos (robôs, máquinas inteligentes, computadores com câmeras etc), que serão os guias do agente guiado. Estes agentes não-humanos são dotados de visão computacional e cada um conhece sua própria orientação. A abstração necessária para instruir um agente guiado será alcançada por meio de instruções baseadas em preposições espaciais projetivas, como direita, esquerda, frente e atrás, que segundo Piaget e Inhelder (1967) é uma das três maneiras de se classificar relações espaciais. Dessa forma, dependendo da posição relativa do objetivo para o agente guiado, instruções do tipo “vire para direita e ande para frente” serão dadas, tornando a navegação egocêntrica para o agente que está sendo conduzido.

1.1 JUSTIFICATIVA

Em diversas ocasiões um agente autônomo móvel pode precisar ser guiado por outros agentes autônomos não-humanos. Considerando que o agente móvel a ser guiado seja um humano, isso pode acontecer, por exemplo, pelo fato do humano não poder enxergar o objetivo para onde deve ir. No caso do agente móvel ser um robô, a necessidade em receber instruções de direções pode surgir por causa de falhas no sistema perceptivo do robô ou por problemas decorrentes da complexidade das cenas a serem interpretadas, como, por exemplo, em situações com ocorrência de oclusão, em que o objeto para qual o robô deve se deslocar pode permanecer oculto de determinados pontos de vista.

A navegação guiada é de extrema importância em diversos domínios e ocasiões, como, por exemplo, em uma situação de resgate de vítimas em um desastre. Muitas linhas de pesquisa são focadas em desenvolver robôs e sistemas inteligentes que possam encontrar as vítimas e resgatá-las. Porém, a maioria dos robôs atuais não teria capacidade de prestar os primeiros socorros e, tão pouco, retirar as vítimas do meio dos escombros. Assim, parece muito mais plausível e eficiente utilizar agentes autônomos móveis, como robôs, para encontrar as vítimas e guiar, por meio de instruções baseadas em relações projetivas, um paramédico até a vítima. Conforme já citado, o paramédico do exemplo pode não conseguir enxergar a vítima e, muito provavelmente, ele, como a maioria dos seres humanos, não saberá sua própria orientação com relação a um sistema global de coordenadas.

Outra possível classe de problemas em que o uso da navegação guiada é interessante, é o domínio com múltiplos robôs que precisam cooperar para atingir um objetivo comum. Neste domínio, o robô que está mais próximo de cumprir a tarefa a que foi designado pode ficar, inesperadamente, sem percepção. Neste caso, os outros robôs podem guiar o robô sem visão até o objetivo, mesmo que ele não perceba mais o mundo ao seu redor. Um exemplo concreto desse problema pode ser encontrado no domínio da liga humanoide da *RoboCup Soccer*, no qual os robôs devem cooperar para conseguir ganhar uma partida de futebol. Para ganhar a partida, um dos objetivos dos robôs é chegar até a bola e chutá-la em direção ao gol adversário. Porém, em diversos momentos a visão do robô que está mais perto da bola pode falhar e os outros robôs poderiam guiá-lo até, pelo menos, próximo da bola.

1.2 OBJETIVO

O objetivo deste trabalho é propor, analisar e testar dois modelos híbridos, qualitativo-probabilístico, de navegação guiada de um agente autônomo por meio de instruções baseadas em relações projetivas obtidas por múltiplas perspectivas espaciais.

1.3 CONTRIBUIÇÕES

Para atingir o objetivo exposto na seção anterior, algumas etapas de pesquisa foram realizadas com a intenção de entender melhor o problema. Um ponto chave deste trabalho é o fato das instruções serem dadas ao agente guiado por meio do uso de preposições projetivas, que são objetos de estudo das áreas de Cognição Humana (PIAGET; INHELDER, 1967; HERSKOVITS, 1987; RESCORLA, 2009) e, também, da Inteligência Artificial, mais precisamente da área de Raciocínio Espacial Qualitativo (VORWERG et al., 1997; CHEN et al., 2015).

Assim, inicialmente tentou-se alcançar o objetivo deste trabalho por meio de técnicas puramente qualitativas, com foco especial nos métodos que trabalham com relações de direções relativas, por serem os que efetivamente utilizam, ou acomodam, relações projetivas como a $OPRA_m$ (MORATZ, 2006; MOSSAKOWSKI; MORATZ, 2012) e o $StarVars_m$ (LEE, 2013; LEE; RENZ; WOLTER, 2013) que serão apresentados na seção 2.

Depois de experimentos iniciais utilizando robôs reais para se realizar inferências espaciais com a $OPRA_m$, verificou-se que esse cálculo qualitativo apresenta algumas desvantagens, que serão abordadas na seção 2.2.1.4.1. Conduziu-se, então, um experimento de navegação guiada com o $StarVars_m$, que, por retornar o primeiro modelo de mundo válido encontrado, não necessariamente representa, exatamente, a configuração espacial real das entidades envolvidas, não sendo eficiente o bastante para realizar a navegação guiada, principalmente, quando não se conhece a orientação do agente guiado.

Percebeu-se então a necessidade de se incluir alguma técnica probabilística para filtrar as incertezas provenientes da visão computacional dos agentes autônomos e do próprio modelo de mundo construído pelo $StarVars_m$. Dessa forma, este trabalho propõe o uso do Filtro de Partículas utilizado no contexto de localização de agentes autônomos móveis (THRUN, 2002; THRUN; BURGARD; FOX, 2005), que é uma técnica tradicional de filtragem que trabalha em duas etapas – atualização e predição, modificado para receber, na atualização, ao invés de informações quantitativas, os dados do modelo de mundo gerados pelo $StarVars_m$ e para utilizar um modelo de movimento genérico, durante a predição, que funcionaria para qualquer tipo de agente móvel. Esse método foi nomeado Filtro de Partículas Qualitativo para Navegação Guiada – *Qualitative Particle Filter for Guided Navigation* ($QPFNG_m$).

Com o intuito de avaliar o modelo proposto, brevemente explicado no parágrafo anterior, e poder gerar comparações, outra técnica foi desenvolvida com o mesmo propósito. O segundo modelo partiu do princípio de que toda a inferência necessária para a navegação guiada poderia ser feita diretamente com dados numéricos e quantitativos como entradas para o Filtro de Partículas. E, uma vez que se obtivesse a instrução numérica para o agente guiado, a preposição projetiva seria obtida por meio de uma tabela de correlação entre graus e preposição projetiva. Esse segundo método foi nomeado Filtro de Partículas Numérico para Navegação Guiada com Instruções Qualitativas – *Numerical Particle Filter for Guided Navigation with Qualitative Instructions* ($NPFNGQI_m^c$).

Assim, as duas principais contribuições deste trabalho são estes dois modelos híbridos propostos para se realizar a navegação guiada de um agente autônomo móvel, que não pode perceber o mundo, por meio de instruções de direções baseadas em preposições projetivas, em que as inferências são alcançadas por meio da união de múltiplas perspectivas de outros agentes autônomos.

Para acomodar os dois modelos propostos, este trabalho apresenta também um algoritmo global para se realizar navegação guiada.

Outra contribuição está no fato de que, apesar de muitos trabalhos da área de Raciocínio Espacial Qualitativo utilizarem como uma das principais motivações a navegação de agentes autônomos, este trabalho demonstra, na seção de experimentos (seção 5), que, pelo menos para a navegação guiada, métodos puramente qualitativos tendem a não ser muito eficientes quando a orientação do agente guiado é conhecida e também há grande possibilidade de falha quando a orientação do agente guiado não é conhecida, pois o modelo qualitativo não trata as incertezas provenientes de informações parciais. Dessa forma, quatro comparações de métodos foram possíveis de serem realizadas: duas com o método qualitativo $StarVars_m$ e duas com os métodos híbridos propostos neste trabalho.

1.4 ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado em seções, conforme estrutura apresentada a seguir:

Seção 2 – Conceitos fundamentais: revisão dos conceitos de navegação de agentes móveis e das técnicas de Raciocínio Espacial Qualitativo, com foco nas abordagens com direções relativas;

Seção 3 – Trabalhos correlatos: apresenta três classes de problemas que são diretamente correlacionadas com as propostas deste trabalho;

Seção 4 – Navegação guiada com relações projetivas obtidas por múltiplas perspectivas espaciais: apresenta as duas propostas deste trabalho – $QPFGN_m$ e $NPFGNQI_m^r$;

Seção 5 – Experimentos: apresenta os experimentos simulados realizados com quatro modelos de navegação guiado por meio de preposições projetivas e duas provas de conceito com os modelos híbridos propostos neste trabalho que utilizaram robôs humanoides reais.

Seção 6 – Conclusão e sugestões de possíveis trabalhos futuros.

2 CONCEITOS FUNDAMENTAIS

Esta seção tem como intuito apresentar uma revisão dos conceitos fundamentais para o entendimento deste trabalho. A seção 2.1 aborda os conceitos mais tradicionais para se realizar navegação de agentes móveis, englobando robôs e humanos, além do problema de navegação guiada. Por sua vez, a seção 2.2 apresenta uma revisão sobre o Raciocínio Espacial Qualitativo, com foco nos cálculos que utilizam direções relativas.

2.1 NAVEGAÇÃO DE AGENTES MÓVEIS AUTÔNOMOS

O problema genérico da navegação pode ser brevemente definido em três perguntas: “onde estou?”, “para onde vou?” e “como chego lá?” (LEONARD; DURRANT-WHYTE, 1991). Assim, a navegação pode ser definida como a combinação de três competências fundamentais para agentes móveis: localização, construção e interpretação de mapas e planejamento de movimento (NEHMZOW, 2003; STACHNISS; BURGARD, 2014).

A construção e interpretação de mapas no contexto da navegação é qualquer representação interna, criada pelo agente, de um domínio do mundo real. Localização se refere a capacidade do agente em saber sua própria postura com relação a um sistema de referência. Por sua vez, o planejamento de movimento define o caminho a ser percorrido pelo agente depois que a localização foi realizada e que a posição do objetivo foi estabelecida com base no mesmo sistema de referência da localização (NEHMZOW, 2003).

A navegação de agentes móveis autônomos é uma área ampla de estudos, pois pode englobar estratégias de navegação em seres humanos, animais e robôs móveis. De interesse para este trabalho são as navegações realizadas por robôs móveis e humanos, portanto, a navegação em animais não será abordada.

2.1.1 Navegação Robótica

A navegação robótica depende, fortemente, do sistema sensorial do robô, uma vez que é por meio dos sensores que o robô cria o seu modelo de mapa. Além do mapa, é por meio das informações obtidas dos sensores que o robô se localiza no ambiente enquanto executa as ações que o levarão até o objetivo (MILFORD; SCHULZ, 2014).

Existem diversos tipos de sensores com diferentes características para se trabalhar em diferentes ambientes. No contexto da navegação robótica, os robôs tendem a utilizar um ou mais dos cinco tipos de sensores apresentados a seguir (MILFORD; SCHULZ, 2014):

- a) telômetros: sonares e lasers;
- b) sensores visuais: câmeras convencionais e câmeras infravermelhas;
- c) sensores de movimento: acelerômetros, giroscópios e decodificador de eixos;

- d) sensores de localização global: GPS, bússolas, sensores que detectam WiFi e Identificadores de Rádio Frequência (*Radio Frequency Identification* – RFID);
- e) outros sensores: sensores táteis, sensores de força, sensores olfativos etc.

Existem diversos métodos para navegação robótica que são tradicionais na literatura. Por exemplo, para tratar o problema da localização, uma das técnicas mais antigas, e que não é exclusiva da robótica, é o *dead reckoning*, que é o processo de se calcular a posição atual baseado na posição anterior combinada com os dados provenientes da odometria. O mapeamento, por sua vez, pode ser feito com base no reconhecimento, visual ou por sensores telêmetros, de pontos de referência naturais ou artificiais. Já para o planejamento de movimento, os métodos de campos potenciais (KHATIB, 1986), os algoritmos de busca (CHOUBEY; GUPTA, M., 2013) – Dijkstra (DIJKSTRA, 1959) e A^* (HART; NILSSON; RAPHAEL, 1968) – e o diagrama de Voronoi (CANNY, 1985; CANNY; DONALD, 1988), são bastante utilizados e todos visam encontrar o caminho mais curto até um objetivo evitando colidir com obstáculos.

O problema dos métodos apresentados no parágrafo anterior é que nenhum deles considera as complicações provocadas pela incerteza (RUSSELL; NORVIG, 2010). É sabido que é a incerteza existente no mundo físico que faz com que o problema da navegação seja tão complexo, pois, sem a incerteza, navegar seria trivial: o uso do *dead reckoning* seria suficiente para resolver o problema da localização (MILFORD; SCHULZ, 2014).

O mundo real é repleto de incertezas para um agente autônomo, que podem ser provenientes de percepções parciais e ruidosas de seu domínio de atuação. Assim, normalmente, um agente consegue, no máximo, uma avaliação probabilística da sua situação atual; isso faz com que a probabilidade seja de extrema importância para a navegação robótica. Por meio da probabilidade, o agente pode ter um grau de crença em sentenças de um determinado problema em cada instante de tempo, levando em consideração informações de uma quantidade finita de estados anteriores (hipótese de Markov) (RUSSELL; NORVIG, 2010).

2.1.1.1 Probabilidade

Agentes autônomos têm que lidar com uma quantidade enorme de incerteza que é proveniente do mundo real. Geralmente, um ambiente real tem uma grande parcela de imprevisibilidade e, mesmo a parte previsível, tem uma chance de mudar. No caso mais específico de robôs e máquinas inteligentes, além das incertezas intrínsecas do mundo real, estes agentes têm que lidar com sensores limitados e sujeitos a erros e ruídos; mais erros podem ser acrescentados por conta dos atuadores, que estão sujeitos a falhas mecânicas, folgas e controles ruidosos. Não obstante, a incerteza pode ainda ser proveniente do software, uma vez que os modelos internos costumam ser aproximações de modelos do mundo real (THRUN; BURGARD; FOX, 2005).

Por isso, o uso da probabilidade na criação de modelos que visam inserir alguma espécie de inteligência em agentes autônomos é tão importante. Por meio da probabilidade pode-se modelar as incertezas do mundo, dos sensores, dos atuadores e, até mesmo, do software, e criar

modelos mais robustos que permitam o raciocínio, a inferência ou o aprendizado em agentes inteligentes.

2.1.1.1.1 *Conceitos Básicos de Probabilidade*

Algumas das notações básicas utilizadas pela teoria da probabilidade, que são importantes para as aplicações em Inteligência Artificial, serão apresentadas nesta seção. Letras maiúsculas como X , Y e Z serão adotadas para representar variáveis aleatórias, que podem assumir múltiplos valores, enquanto letras minúsculas, x , y e z serão utilizadas para denotar um valor específico que as variáveis aleatórias podem assumir. Segundo o exemplo mencionado por Thrun, Burgard e Fox (2005), uma variável aleatória X pode ser a ação “jogar uma moeda”, sendo que X pode assumir os valores $x = cara$ ou $x = coroa$.

Na teoria da probabilidade, os graus de crença são aplicados às proposições. Por exemplo, a variável aleatória X pode assumir valor x , logo $X = x$ é uma proposição. Desta maneira, para simplificar a notação, as proposições serão representadas pelas próprias letras minúsculas x , y e z . A probabilidade associada a uma sentença é definida pela soma das probabilidades dos estados de mundo ω nas quais as proposições são válidas, conforme exibido na Equação (1) (RUSSELL; NORVIG, 2010).

$$\text{Para qualquer proposição } x, P(x) = \sum_{\omega \in x} P(\omega). \quad (1)$$

Por meio da probabilidade atribuímos valores entre **0** e **1** para cada proposição, sendo que esses valores representam o grau de crença. Assim, uma sentença com probabilidade **0** corresponde a uma crença clara de que a sentença é falsa, enquanto que probabilidade **1** significa uma crença absoluta de que a sentença em questão é verdadeira (SCHILLER; SRINIVASAN; SPIEGEL, 2008).

Desta maneira, a probabilidade proporciona o mesmo compromisso ontológico da lógica, assumindo que os fatos são ou não válidos no mundo. Contudo, os compromissos epistemológicos de agentes lógicos e probabilísticos são diferentes: um agente lógico acredita que cada sentença é verdadeira ou falsa, enquanto um agente probabilístico tem um grau numérico entre **0** e **1** de crença na veracidade da sentença (RUSSELL; NORVIG, 2010).

As probabilidades podem mudar ao passo em que mais percepções são recebidas por um agente probabilístico. Essas percepções são as evidências nas quais se baseiam as alegações de probabilidade. Antes da evidência ser recebida, o agente tem uma probabilidade a priori ou incondicional $P(x)$ (RUSSELL; NORVIG, 2010), por outro lado, depois que a evidência é obtida, o agente passa a ter uma probabilidade posterior ou condicional, $P(x|y)$, que significa “a probabilidade de x , dado que sabemos y ” (DEGROOT; SCHERVISH, 2011; THRUN; BURGARD; FOX, 2005). Na maior parte dos casos, o agente consegue obter alguma evidência por meio de suas percepções, o que torna a probabilidade posterior muito útil na modelagem de problemas reais.

As probabilidades condicionais podem ser escritas em função das probabilidades a priori: $P(x|y) = P(x \wedge y)/P(y)$. Esta equação dá origem à regra do produto, definida como $P(x \wedge y) = P(x|y)P(y)$. Devido à comutatividade da conjunção, essa regra pode ser escrita no sentido contrário, resultando em $P(x \wedge y) = P(y|x)P(x)$ (LIPSCHUTZ, 1965). A regra do produto e a propriedade da comutatividade são importantes para conceitualizar a regra de Bayes, que será abordada na seção 2.1.1.1.2.

A medida de probabilidade deve obedecer algumas propriedades básicas que são conhecidas como os axiomas de probabilidade, ou axiomas de Kolmogorov¹. São três os axiomas de probabilidade (BERTSEKAS; TSITSIKLIS, 2008):

- a) Não-negatividade: $P(x) \geq 0$, \forall proposição x .
- b) Aditividade: Considerando que duas proposições x e y são mutuamente excluídas, a probabilidade de x ou y é igual a probabilidade de x somada a probabilidade de y . Assim, $P(x \vee y) = P(x) + P(y)$. De maneira mais geral, o axioma vale para qualquer número contável de proposições mutuamente exclusivas.
- c) Normalização: A probabilidade de todo o espaço amostral Ω é igual a $\mathbf{1}$, $P(\Omega) = \mathbf{1}$.

Várias outras propriedades podem ser derivadas dos axiomas de probabilidade, como $P(x) + P(\neg x) = \mathbf{1}$ e $P(x \vee y) = P(x) + P(y) - P(x \wedge y)$ (RUSSELL; NORVIG, 2010).

2.1.1.1.2 Regra de Bayes

A regra de Bayes é fundamental em sistemas que utilizam inferências probabilísticas em geral (THRUN; BURGARD; FOX, 2005). A regra de Bayes, também conhecida como Lei de Bayes e Teorema de Bayes, é a base dos sistemas modernos de Inteligência Artificial para inferência probabilística (RUSSELL; NORVIG, 2010).

A regra de Bayes é exibida na equação (2).

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)} \quad (2)$$

É possível analisar que a regra de Bayes deriva da regra do produto $P(x \wedge y) = P(x|y)P(y) = P(y|x)P(x)$, uma vez que, para se encontrar a regra de Bayes, basta isolar um membro da equação (GELMAN et al., 2013). Pode-se notar ainda pela Equação (2) que o denominador da regra de Bayes $P(y)$ é uma probabilidade incondicional, dessa forma, $P(y)$ será o mesmo para qualquer valor de x na probabilidade posterior $P(x|y)$. Por esse motivo, $P(y)$ é normalmente escrito como um normalizador da regra de Bayes e genericamente denotado como η (THRUN; BURGARD; FOX, 2005). A regra de Bayes com normalização pode ser vista na Equação (3), na qual $\eta = \mathbf{1}/P(y) = \mathbf{1}/[P(y|x)P(x) + P(y|\neg x)P(\neg x)]$.

¹Homenagem ao matemático russo Andrei Nikolaevich Kolmogorov que mostrou como elaborar o resto da teoria da probabilidade partindo desses fundamentos básicos (RUSSELL; NORVIG, 2010).

$$P(x|y) = \eta P(y|x)P(x). \quad (3)$$

A Regra de Bayes pode também ser aplicada a problemas com múltiplas evidências. Assim, por exemplo, a proposição x pode ter a sua probabilidade condicionada a duas evidências, y e z : $P(x|y \wedge z)$ ou $P(x|y,z)$.

Desta forma, é perfeitamente aceitável utilizar a regra de Bayes para variáveis aleatórias arbitrárias, como as variáveis X , Y e Z (THRUN; BURGARD; FOX, 2005). Considerando que $X = x$, $Y = y$ e $Z = z$, a regra de Bayes para duas evidências pode ser obtida, conforme exibida na Equação (4) (RUSSELL; NORVIG, 2010).

$$P(x|y \wedge z) = \frac{P(y \wedge z|x)P(x)}{P(y \wedge z)} \quad \text{ou} \quad P(x|y, z) = \frac{P(y, z|x)P(x)}{P(y, z)}. \quad (4)$$

A forma geral da regra de Bayes para múltiplas evidências é exibida na Equação (5), adaptada de Neapolitan (2003), na qual a variável $y_{1:k}$ representa as k evidências consideradas.

$$P(x|y_{1:k}) = \frac{P(y_{1:k}|x)P(x)}{P(y_{1:k})}. \quad (5)$$

Contudo, as equações de Bayes para múltiplas evidências têm alguns problemas relacionados ao fato de que, para se encontrar a probabilidade de x dadas as evidências $y_{1:k}$, é necessário o conhecimento da probabilidade conjunta de $y_{1:k}$ e também da probabilidade posterior de $y_{1:k}$ dado x . A probabilidade conjunta de $P(y_{1:k})$ pode ser eliminada por meio da normalização η . Porém, a probabilidade posterior $P(y_{1:k}|x)$ pode ser bastante complicada de se obter.

2.1.1.1.3 Probabilidade Aplicada à Navegação Robótica

Conforme previamente citado, a navegação é normalmente vista como a combinação de três competências básicas para agentes móveis: localização, construção e interpretação de mapas e planejamento de movimento. Assim, em se tratando de máquinas inteligentes, diferentes técnicas probabilísticas foram criadas para proporcionar cada uma dessas habilidades.

Dentre estas três habilidades, a compreensão do procedimento de localização, que será apresentado na seção 2.1.1.2, é de vital importância para o completo entendimento deste trabalho, pois, apesar da proposta de navegação guiada (seção 4) contemplar também o mapeamento e o planejamento de movimento, essas duas competências não ocorrem da maneira tradicional que será brevemente apresentada nas seções 2.1.1.3 e 2.1.1.4. Por sua vez, o algoritmo principal da proposta deste trabalho foi inspirado no algoritmo da Localização de Monte Carlo, que será apresentado na seção 2.1.1.2.3.

Assim, além da localização, as próximas seções abordarão, de forma mais sucinta e em caráter de contextualização, os métodos probabilísticos mais tradicionais para se solucionar o mapeamento e o planejamento de movimento.

2.1.1.2 Localização

A localização robótica trata do problema de determinar a postura do robô com relação a um mapa dado (THRUN; BURGARD; FOX, 2005). Segundo Russell e Norvig (2010), a localização do robô pode apresentar três graus de dificuldade:

- a) Rastreamento: o problema será de rastreamento se a postura inicial do robô for conhecida, assim a localização vai tratar de estimar a postura do robô ao longo do tempo;
- b) Localização global: nesse caso a postura inicial do robô não é conhecida, logo o robô tem que estimar sua localização inicial e, depois, o problema será de rastreamento;
- c) Sequestro: nesse problema um robô autônomo já localizado é carregado para uma localização arbitrária que ele desconhece, o que faz com que ele perca o rastreamento e tenha que voltar para a condição de localização global inicial.

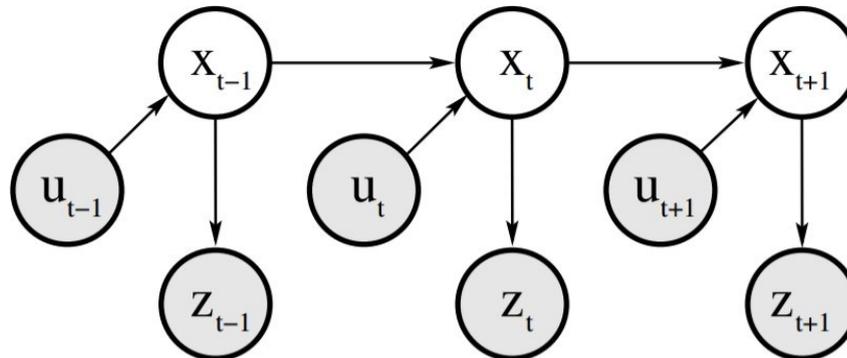
Existem algumas maneiras diferentes de se realizar a autolocalização utilizando probabilidade, como, por exemplo, a Localização de Markov (FOX; BURGARD; THRUN, 1998, 1999), o Filtro de Kalman – *Kalman Filter* (KF) (KALMAN, 1960) e suas variações, Filtro de Kalman Estendido – *Extended Kalman Filter* (EKF) (REID, 1979; SORENSON, 1985) e Filtro de Kalman Unscented – *Unscented Kalman Filter* (UKF) (JULIER; UHLMANN, 1997; WAN; MERWE, 2000), e a Localização com o uso do Filtro de Partículas, conhecida como Localização de Monte Carlo – *Monte Carlo Localization* (MCL) (FOX et al., 1999; DELLAERT et al., 1999; THRUN et al., 2000; THRUN; BURGARD; FOX, 2005). Todos os métodos de localização probabilística são variações do Filtro Bayesiano (THRUN; BURGARD; FOX, 2005), que será apresentado a seguir.

2.1.1.2.1 Filtro Bayesiano

O Filtro Bayesiano é o algoritmo mais geral utilizado para se calcular crenças sobre o estado atual. O Filtro Bayesiano calcula a probabilidade posterior *bel* a partir de todas as evidências obtidas até o presente momento t , ou seja, calcula $P(X_t | e_{1:t})$, onde X_t é o conjunto de variáveis de estado não-observáveis em t e $e_{1:t}$ é uma evidência que está sendo observada desde $t = 1$ até t . Uma vez que se conhece o conjunto de variáveis de estado e de evidências para um dado problema, o Filtro Bayesiano precisa da especificação dos modelos de transição e de observação.

O modelo de transição descreve a transição de um estado em t para o próximo estado no mundo, em $t + 1$. Se um processo de Markov de primeira ordem for considerado, no qual o estado atual depende apenas do estado anterior, a distribuição condicional deve ser dada como $P(X_t | X_{t-1})$. Caso exista alguma entrada conhecida u_t que influencie diretamente o modelo de transição, essa entrada pode ser considerada como uma nova evidência no modelo probabilís-

Figura 1 – HMM que caracteriza a evolução das ações, estados e observações



Fonte: THRUN; BURGARD; FOX, 2005

tico e a distribuição condicional passa a ser dada por $P(X_t|X_{t-1}, u_t)$ (THRUN; BURGARD; FOX, 2005). Por sua vez o modelo de observação, ou modelo de sensores, descreve como as variáveis de evidência são afetadas pelo estado atual, $P(E_t|X_t)$, onde E_t representa o conjunto de variáveis de evidência em t (RUSSELL; NORVIG, 2010).

Na robótica, por exemplo, sabe-se que as entradas u_t do modelo de transição podem ser representadas pelas ações que um robô pode executar, enquanto que, no modelo de observação, normalmente as evidências são obtidas por meio de medições Z_t efetuadas pelo robô; assim, a distribuição condicional do modelo de observação passa a ser representada por $P(Z_t|X_t)$. Dessa forma, a evolução das ações, estados e observações pode ser caracterizada como um modelo probabilístico temporal, mais especificamente, nesse caso, como um Modelo Oculto de Markov – *Hidden Markov Model* (HMM) (Figura 1), que descreve o estado por somente uma variável aleatória discreta (RUSSELL; NORVIG, 2010).

O Algoritmo 1 exibe uma forma geral do algoritmo utilizado para se realizar o Filtro Bayesiano, considerando-se que a ação u_t influencia o modelo de transição de estados, que a evidência é obtida por meio de uma medição z_t e que o algoritmo é recursivo, ou seja, $bel(x_t)$ é calculada a partir da crença $bel(x_{t-1})$.

Algoritmo 1 – Forma geral do Filtro Bayesiano (THRUN; BURGARD; FOX, 2005).

```

1 Entrada:  $bel(x_{t-1}), u_t, z_t$ 
2 Saída:  $bel(x_t)$ 
3 início
4   para todo  $x_j$  faça
5      $bel(x_t) = \int p(x_t|u_t, x_{t-1}) \cdot bel(x_{t-1}) \cdot dx_{t-1}$ 
6      $bel(x_t) = \eta \cdot p(z_t|x_t) \cdot bel(x_t)$ 
7   fim
8   retorna  $bel(x_t)$ 
9 fim

```

O Filtro Bayesiano tem duas etapas básicas: a predição $\overline{bel}(x_t)$, que é apresentada na linha **5**, em que a crença do estado atual é condicionada à ação u_t e é baseada na crença do estado anterior $bel(x_{t-1})$; e a atualização da medição $bel(x_t)$, apresentada na linha **6**, em que a probabilidade da medição z_t ter sido observada é multiplicada pela crença prevista $\overline{bel}(x_t)$ (THRUN; BURGARD; FOX, 2005).

A Localização por meio do uso do Filtro de Partículas, ou Localização de Monte Carlo – *Monte Carlo Localization* (MCL), tem fundamental importância para este trabalho, pois é dela que vem toda a base probabilística utilizada. Para tanto, as próximas seções apresentarão, respectivamente, o Filtro de Partículas e a sua variante para a localização, a MCL. Ambos são Filtros Bayesianos que tentam aproximar a crença $bel(x_t)$ por um conjunto de partículas S_t (THRUN; BURGARD; FOX, 2005).

2.1.1.2.2 Filtro de Partículas

Filtro de Partículas é o nome genérico de algumas técnicas probabilísticas cujo objetivo é computar recursivamente em cada instante de tempo t a probabilidade posterior bel por um conjunto S_t de N partículas iniciadas aleatoriamente e compostas por um peso w , que é dado conforme a sua importância. O Filtro de Partículas básico é exibido no Algoritmo 2, onde S_{t-1} representa o conjunto de partículas em $t-1$, u_t representa uma entrada que influencia o modelo de transição de estados e z_t uma medição.

Dessa forma, pode-se notar que na linha **6** (Algoritmo 2) um estado x_t^i é criado com base na partícula x_{t-1}^i e na ação u_t . O conjunto obtido após N iterações dessa linha representa a crença prevista $bel(x_t)$. Por sua vez, na linha **7**, o fator de importância w_t^i de cada partícula é calculado com base na medição z_t . De fato, w_t^i é a probabilidade de se obter a medida z_t com a partícula x_t^i . Assim, o conjunto de partículas com seus respectivos pesos representa, aproximadamente, a crença posterior $bel(x_t)$. Este filtro trabalha como um Filtro Bayesiano recursivo.

Contudo, o Filtro de Partículas conta ainda com a reamostragem ou amostragem por importância (linhas **10** a **13** do Algoritmo 2), em que o algoritmo sorteia partículas do conjunto temporário \overline{S}_t e as insere no novo conjunto S_t . A probabilidade de uma partícula ser sorteada é dada conforme o seu fator de importância w_t^i . Antes da reamostragem, as partículas estão distribuídas de acordo com \overline{bel} . Após a reamostragem, a distribuição das partículas volta a ser a crença posterior $bel(x_t)$.

Filtros de Partículas também têm um papel muito importante em problemas de localização global e rastreamento de robôs móveis (THRUN, 2002). Dentro desse contexto, o filtro é amplamente conhecido como MCL. A MCL tem fundamental importância para este trabalho, pois é dela que vem toda a base probabilística utilizada. Para tanto, a próxima seção apresentará a MCL.

Algoritmo 2 – Forma geral do Filtro de Partículas: caso específico do Filtro Bayesiano com amostragem por importância (THRUN; BURGARD; FOX, 2005).

```

1 Entrada:  $S_{t-1}$ ,  $u_t$ ,  $z_t$ 
2 Saída:  $S_t$ 

3 início
4    $S_t = S_{t-1} = \emptyset$ 
5   para  $i = 0$  até  $N$  faça
6     amostre  $x_t^i \sim p(x_t | u_t, x_{t-1}^i)$ 
7      $w_t^i = p(z_t | x_t^i)$ 
8      $S_t = S_t \cup \langle x_t^i, w_t^i \rangle$ 
9   fim
10  para  $i = 0$  até  $N$  faça
11    sorteie  $m$  com probabilidade  $\propto w_t^m$ 
12    adicione  $x_t^m$  em  $S_t$ 
13  fim
14  retorna  $S_t$ 
15 fim

```

2.1.1.2.3 Localização de Monte Carlo

Na MCL, um robô pode estimar sua própria postura (x, y, θ) por meio de um conjunto S_t de N partículas, no qual cada partícula é composta por uma probabilidade, dada pelo fator de importância w , e por uma possível postura do robô.

Por ser uma versão mais específica do Filtro de Partículas, a MCL também é um Filtro Bayesiano recursivo, em que a crença da localização do robô é obtida com base na junção das informações provenientes do movimento executado pelo robô e das medições realizadas com relação aos pontos de referência de um mapa conhecido. Assim, o algoritmo da MCL também é composto pelas etapas de predição, que é onde o modelo de movimento é aplicado, e de atualização, que é onde o modelo de sensor é incorporado. Assim, o Algoritmo 3 exibe o pseudocódigo para se realizar a Localização de Monte Carlo.

A predição $\overline{bel}(x_t)$ é apresentada na linha **6**, em que a crença do estado atual é condicionada à ação u_t e é baseada na crença do estado anterior $bel(x_{t-1})$. A atualização da medição $bel(x_t)$ é apresentada na linha **7**, em que a probabilidade da medição z_t ter sido observada é multiplicada pela crença prevista $\overline{bel}(x_t)$ (THRUN; BURGARD; FOX, 2005).

A função `modeloDeMovimento` é normalmente realizada com base na cinemática do robô. A cinemática pode ser implementada de algumas maneiras, como, por exemplo, por meio de um modelo de velocidades, onde u_t especifica as velocidades de translação v_t e rotação w_t do robô. A Equação (6) exibe u_t para o modelo de velocidades.

$$u_t = \begin{bmatrix} v_t \\ w_t \end{bmatrix} \quad (6)$$

Algoritmo 3 – Localização de Monte Carlo (THRUN; BURGARD; FOX, 2005).

```

1 Entrada:  $S_{t-1}, u_t, z_t$ 
2 Saída:  $S_t$ 

3 início
4    $S_t = S_{t-1} = \emptyset$ 
5   para  $i = 0$  até  $N$  faça
6      $x_t^i = \text{modeloDeMovimento}(u_t, x_{t-1}^i)$ 
7      $w_t^i = \text{modeloDeSensor}(z_t, x_t^i)$ 
8      $S_t = S_{t-1} \cup \langle x_t^i, w_t^i \rangle$ 
9   fim
10  para  $i = 0$  até  $N$  faça
11    sorteie  $m$  com probabilidade  $\propto w_t^m$ 
12    adicione  $x_t^m$  em  $S_t$ 
13  fim
14  retorna  $S_t$ 
15 fim

```

Outro modo de se implementar a cinemática é por meio do modelo de odometria, que utiliza diretamente as medidas de odometria do robô. Assim, u_t especifica o par x_{t-1} e x_t , conforme Equação (7), que representam a postura anterior e a postura atual, respectivamente (THRUN; BURGARD; FOX, 2005).

$$u_t = \begin{matrix} - & \Sigma \\ x_{t-1} & \\ x_t & \end{matrix} . \quad (7)$$

Já a função `modeloDeSensor` é responsável por modelar o funcionamento dos sensores no mundo real. Robôs podem usar uma grande variedade de sensores, como câmeras, sonares, lasers etc. e o modelo depende, exclusivamente, do sensor que será usado, incluindo, explicitamente, a modelagem dos ruídos existentes nas medidas obtidas. Formalmente, o `modeloDeSensor` é definido por meio da probabilidade condicional $p(z_t|x_t)$, que é obtida pelo produto das probabilidades das medidas individuais z_t^k , conforme Equação (8).

$$p(z|x) = \prod_{k=1}^K p(z^k|x) . \quad (8)$$

A MCL, em sua forma original, pode resultar em estimativas de posturas incorretas, devido, principalmente, à baixa quantidade de partículas utilizadas ou por conta do problema do robô sequestrado. Para lidar com esses problemas, Lenser e Veloso (2000) propuseram o método conhecido como Inicialização por Sensor – *Sensor Resetting*. A principal ideia do método é substituir partículas com baixa probabilidade por partículas com posições calculadas por meio de triangulação de dois (ao usar estimativas de distância) ou três (quando usando estimativas de direção) pontos referenciais percebidos pelo robô. Mais recentemente, Coltin e Veloso (2013) propuseram a utilização de múltiplas observações para a realização da Inicialização por Sensor.

Uma vez que a MCL tem se mostrado um método eficiente para localização e rastreamento de agentes autônomos, um número notável de variações desta técnica pode ser encon-

trado na literatura, como, por exemplo, a MCL adaptativa (FOX et al., 1999), a MCL combinada (THRUN et al., 2000), a MCL agrupada (SANCHEZ; MILSTEIN; WILLIAMSON, 2002) e a MCL baseada em estados (HONG; ZHOU; TIAN, 2009).

2.1.1.3 Mapeamento

Conforme demonstrado na seção anterior, os algoritmos de localização precisam de um mapa previamente conhecido para funcionar. Como nem sempre o ambiente no qual o robô está atuando é conhecido, o robô pode precisar aprender o mapa. O mapeamento não é um problema trivial, uma vez que o robô não sabe onde está e não conhece o ambiente. Por isso, os problemas de mapeamento são frequentemente chamados de Localização e Mapeamento Simultâneos – *Simultaneous Localization and Mapping* (SLAM) (THRUN; BURGARD; FOX, 2005; DURRANT-WHYTE; BAILEY, 2006; BAILEY; DURRANT-WHYTE, 2006), uma vez que o robô não deve apenas construir o mapa, mas deve fazer isso enquanto se localiza.

2.1.1.3.1 Localização e Mapeamento Simultâneos

O problema de SLAM é definido por um robô se movimentando em um ambiente desconhecido, começando de um ponto inicial x_0 , que percebe o ambiente que deve mapear por meio de sensores ruidosos enquanto determina a própria posição relativa ao mapa construído (STACHNISS; LEONARD; THRUN, 2016).

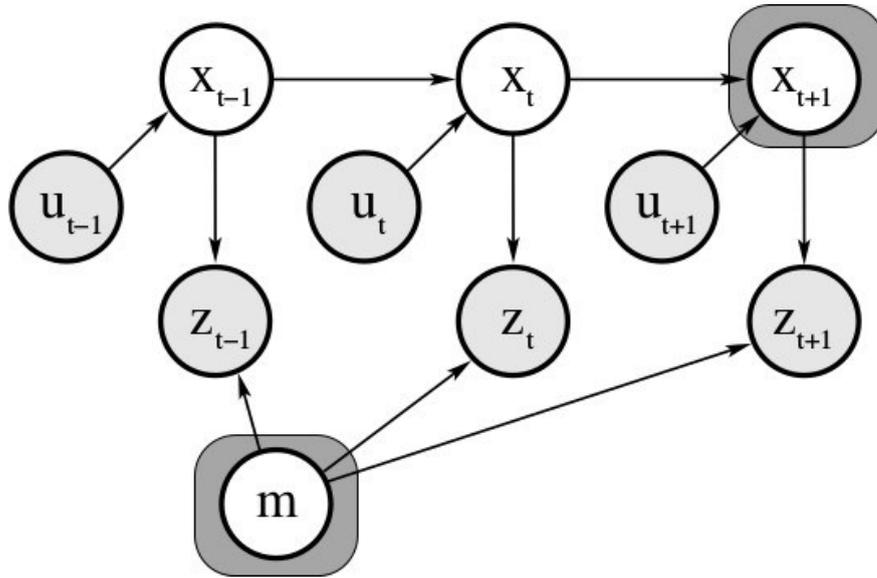
Existem dois tipos principais de SLAM na literatura: SLAM *online* e SLAM completo. O problema de SLAM *online* procura descobrir a probabilidade da localização atual do robô e também a probabilidade posterior sobre o mapa, como exibido na Equação (9). A Figura 2 exibe o diagrama das variáveis envolvidas no problema de SLAM *online*.

$$p(x_t, m | Z_t, U_t). \quad (9)$$

Já o problema de SLAM completo trata de estimar a probabilidade posterior sobre todo o caminho do robô e sobre o mapa, conforme Equação (10), onde X representa a sequência de posturas assumidas pelo robô, Z representa a sequência das medições realizadas, U representa a sequência de ações executadas pelo robô e T representa o tempo de término (T pode tender ao ∞). Tanto no problema de SLAM *online* como no completo as evidências são diretamente observáveis pelo robô (STACHNISS; LEONARD; THRUN, 2016).

$$p(X_T, m | Z_T, U_T). \quad (10)$$

Assim como na localização (2.1.1.2), existem diversos algoritmos utilizados para se resolver problemas de SLAM. Porém, existem três famílias de algoritmos que se destacam (STACHNISS; LEONARD; THRUN, 2016): Filtro de Kalman Estendido (EKF), Filtro de Partículas e os Baseados em Grafos. Os dois primeiros métodos implementam a SLAM *online*; já os ba-

Figura 2 – Modelo gráfico do problema de SLAM *online*

Fonte: THRUN; BURGARD; FOX, 2005

seados em grafos implementam a SLAM completa. O problema de SLAM formulado por meio do EKF é conhecido como EKF SLAM (SMITH; SELF; CHEESEMAN, 1988, 1990); dentre as abordagens de SLAM que utilizam o Filtro de Partículas, uma das mais conhecidas é a FastSLAM (MONTEMERLO et al., 2002); por sua vez, o método que utiliza grafos é normalmente conhecido como SLAM Baseados em Grafos (GRISSETTI et al., 2010).

Comum às três famílias de algoritmos apresentadas, e conforme apresentado nas equações (9) e (10), é a necessidade de se conhecer o modelo de sensor – medições Z – e o modelo de movimento – ações U – do agente.

2.1.1.4 Planejamento de Movimento

Planejamento de movimento é a competência que busca a escolha correta das ações para um agente móvel. Dessa forma, com o intuito de acomodar toda a incerteza já discutida previamente, algoritmos probabilísticos foram desenvolvidos com a intenção de se alcançar ações que fossem eficientes. Basicamente, são dois os tipos de incerteza que podem existir durante o planejamento de movimento de um agente: a incerteza no próprio movimento e a incerteza nas medições realizadas (THRUN; BURGARD; FOX, 2005).

A incerteza no movimento estabelece que os movimentos dos robôs não são determinísticos, dessa forma os efeitos de ações selecionadas podem não resultar no estado esperado. Isso acontece pois a natureza do próprio robô, assim como a do ambiente, é estocástica. Já a incerteza nas medições são provenientes das limitações nos sensores, que normalmente não são capazes de entregar ao robô informações completas sobre todo o ambiente (THRUN; BURGARD; FOX, 2005).

O paradigma que engloba o primeiro tipo de incerteza, a existente no movimento do robô, é conhecido como Processo de Decisão de Markov – *Markov Decision Process* (MDP) (HOWARD, 1960), que será melhor apresentada na próxima seção.

2.1.1.4.1 Processo de Decisão de Markov

Um MDP assume que o estado que representa o ambiente é completamente observável e pode ser medido durante todo o tempo. Isso significa que o modelo de sensor $p(z|x)$ é determinístico e o agente sempre sabe em que estado, ou postura no caso dos problemas de navegação, se encontra. Contudo, os efeitos das ações escolhidas podem ser determinísticos ou estocásticos. Assim, um planejamento não-probabilístico que considera somente uma sequência de ações, a partir da postura atual, não é suficiente. O planejamento deve ser feito de forma completa, em que ações são definidas para todo o conjunto de possíveis situações, o que significa gerar uma política de ações para todos os estados nos quais o robô pode estar (THRUN; BURGARD; FOX, 2005).

Dessa forma, um MDP pode ser definido pela quadrupla $\langle S, A, T, R \rangle$ (LITTMAN, 1994; KAEHLING; LITTMAN; MOORE, 1996; MITCHELL, 1997), onde:

- a) S é um conjunto finito de estados;
- b) A é um conjunto finito de ações;
- c) $R : S \times A \rightarrow \mathbf{R}$ é uma função recompensa;
- d) $T : S \times A \rightarrow \mathbf{\Pi}(S)$ é uma função de transição de estados, onde $\mathbf{\Pi}(S)$ representa o conjunto de distribuições de probabilidades discretas sobre o conjunto S .

Um exemplo de política de ações pode ser visto na Figura 3, que mostra que ao invés de uma única sequência de ações o robô calculou a melhor ação, representada pelas setas, para cada possível estado do mundo.

Conforme já citado, um MDP não trata problemas nas quais as incertezas também são provenientes dos sensores, ou seja, problemas onde o ambiente é parcialmente observável e o robô pode somente estimar o estado em que está. Para acomodar ambas as incertezas (as das ações e as dos sensores), o MDP foi aprimorado com um modelo de observação, o que resultou no Processo de Decisão de Markov Parcialmente Observável – *Partially Observable Markov Decision Process* (POMDP) (CASSANDRA; KAEHLING; LITTMAN, 1994).

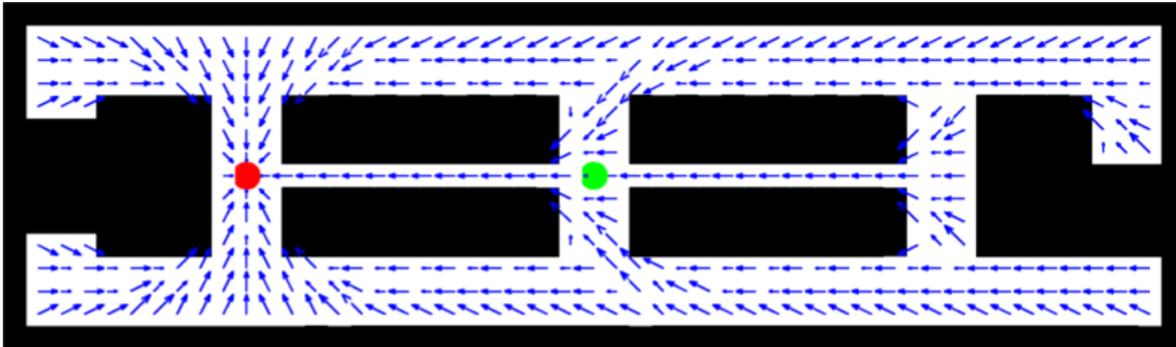
Um POMDP é definido por seis componentes $\langle S, A, T, R, \mathbf{\Omega}, O \rangle$, onde:

- a) S, A, T, R descrevem exatamente um MDP;
- b) $\mathbf{\Omega}$ é um conjunto finito de observações o ;
- c) $O : S \times A \rightarrow \mathbf{\Omega}$ é o modelo de observação $O(s^j, a, o)$, que fornece a probabilidade de se obter a observação o depois da ação a que resultou no estado s^j .

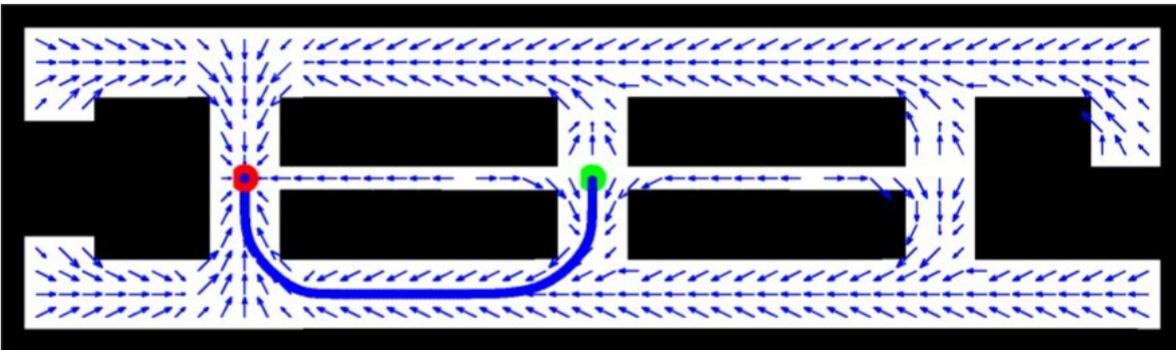
A solução para um POMDP é uma política de ações definida sobre o estado de crença do robô. Isso permite que o robô decida buscar por mais informações se estiver inseguro com relação a alguma variável de estado crítica (RUSSELL; NORVIG, 2010).

Figura 3 – Política de ações para um MDP

(a) Efeitos determinísticos das ações: o robô prefere o caminho menor



(b) Efeitos estocásticos das ações: o robô prefere um caminho maior, para reduzir o risco de colidir com a parede



Fonte: THRUN; BURGARD; FOX, 2005

2.1.2 Navegação Guiada

Uma das maneiras de um agente conseguir navegar até um objetivo é tendo outro agente autônomo o guiando até lá. Existe uma gama de robôs que são construídos especificamente para o propósito de se locomover de um ponto a outro em um ambiente industrial para realizar tarefas relacionadas à transportes. Esses robôs, conhecidos como Veículos Guiados Automaticamente – *Automated Guided Vehicles* (AGVs), são normalmente guiados por marcas artificiais inseridas no ambiente, como linhas pintadas no chão, ou são diretamente teleoperados por humanos.

Apesar de serem bastante utilizados nas indústrias, os AGVs apresentam um problema primordial: são caros para instalar e para se modificar depois de instalado, pois requerem que todo o ambiente seja adaptado para eles.

Contudo, no melhor do meu conhecimento, métodos em que agentes autônomos não-humanos guiam outros agentes autônomos (humanos ou não) são bastante difíceis de serem encontrados na literatura.

2.1.3 Navegação Humana

Até agora, a importância dos sensores para a navegação foi salientada. Contudo, pessoas não possuem, de forma natural, sensores magnéticos, como bússolas, ou lasers para medir distâncias com precisão, e, mesmo assim, podem navegar por longas distâncias e em ambientes repletos de dificuldades.

Segundo Nehmzow (2003), o princípio fundamental da navegação humana está na capacidade em reconhecer diversos pontos de referência, que são memorizados em mapas cognitivos, que, embora sejam distorcidos quando comparados com a representação espacial real, permitem a criação de relacionamentos que dão condição à navegação. Assim, mapas cognitivos desempenham um importante papel na navegação humana, uma vez que possibilitam a compreensão de ambientes externos por meio de um conjunto de sensações que foram vivenciadas e armazenadas para uso posterior em situações que precisam de tomadas de decisão (GOLLEDGE, 2005).

Por meio dos mapas cognitivos, as pessoas podem organizar os lugares, as posições e as relações entre objetos espaciais de um determinado domínio (GOLLEDGE, 2005). Esses mapas representam aspectos geométricos do ambiente por meio de relações espaciais (RESCORLA, 2009), que são normalmente classificadas, segundo a literatura cognitiva, em três categorias: topológicas, projetivas e Euclidianas (PIAGET; INHELDER, 1967).

As relações espaciais topológicas descrevem relações de vizinhança e relações de fronteira, como: dentro, fora, perto, longe, ao redor, entre etc. Já as relações projetivas são aquelas que variam conforme o ponto de projeção muda, ou seja, as relações mudam conforme muda a perspectiva do observador. Alguns exemplos de relações projetivas são: em cima, embaixo, em frente, atrás, esquerda e direita. Por sua vez as relações Euclidianas são relações métricas, portanto dependem de medidas, como distâncias e orientações, para localizar os objetos por meio de coordenadas, o que traz a necessidade de um sistema de referência externo (PORTO, M.; ABREU; PORTO, R., 2011).

Com relação ao sistema de referência, existem basicamente duas estratégias de navegação que são utilizadas pelos seres humanos: a navegação aloccêntrica, que é fundamentada nas coordenadas geográficas absolutas (norte, sul, leste, oeste etc.) e as relações são estabelecidas de objeto para objeto, não importando o ponto de vista da pessoa (COLOMBO et al., 2017); e a egocêntrica, em que a referência é baseada no próprio observador (KLATZKY, 1998). Maiores detalhes sobre os sistemas de referência, com foco nas relações projetivas, serão exibidos na seção 2.2.1.

Assim, seres humanos são capazes de raciocinar espacialmente com informações de senso comum. Além disto, humanos podem raciocinar com conhecimento incompleto (RENZ; NEBEL, 2007) e tem a habilidade de abstrair o mundo físico (MORATZ, 2006), o que torna uma abordagem qualitativa para navegação próxima da maneira como os humanos representam

e raciocinam espacialmente. Uma revisão sobre a abordagem qualitativa será apresentada na próxima seção.

2.2 RACIOCÍNIO ESPACIAL QUALITATIVO

O Raciocínio Espacial Qualitativo – *Qualitative Spatial Reasoning* (QSR) é um dos campos de pesquisa de Inteligência Artificial, dentro da subárea de Representação de Conhecimento (SANTOS, P. E., 2012), que tem como principal desafio a formalização de cálculos que permitam que um computador, possa representar a configuração espacial ou temporal de entidades pertencentes a um domínio em termos puramente qualitativos, assim como raciocinar com base nesta representação (COHN; RENZ, 2007; DYLLA, 2009).

A abordagem qualitativa busca representar o conhecimento espacial sem usar descrições numéricas precisas (CHEN et al., 2015). Este método utiliza uma quantidade limitada de palavras ou categorias qualitativas de valores numéricos para representar as relações espaciais possíveis entre entidades (RENZ; NEBEL, 2007). Segundo Renz (2002), mesmo não fazendo uso de medições quantitativas, a técnica qualitativa não deve ser inexata.

Uma das suposições básicas do QSR está no fato de que a representação espacial é realizada com base nas relações entre as entidades consideradas no espaço. Frequentemente, estas relações são binárias, isto é, entre duas entidades. Normalmente, os métodos de raciocínio qualitativo trabalham com relações que têm a propriedade de serem Conjuntamente Exaustivas e Mutuamente Excludentes – *Jointly Exhaustive and Pairwise Disjoint* (JEPD), também chamadas de relações básicas, o que quer dizer que, para o caso binário, entre quaisquer duas entidades espaciais, existe exatamente uma relação válida (COHN; RENZ, 2007; RENZ; NEBEL, 2007). Um formalismo qualitativo que tenha a propriedade de ser JEPD pode ser chamado de cálculo qualitativo.

Fundamentalmente, o QSR precisa, primeiro, criar a representação espacial qualitativa para, depois, raciocinar por meio dessa representação. Representações qualitativas são, muitas vezes, dadas na forma de restrições compostas pelas relações entre as entidades espaciais. Por isso, os formalismos de QSR podem ser vistos como Problemas de Satisfação de Restrições – *Constraint Satisfaction Problems* (CSP).

Por sua vez, um raciocínio básico deve ser capaz de decidir consistência dadas as representações espaciais (LEE; RENZ; WOLTER, 2013), onde, consistência se refere à verificação da existência de um modelo espacial que satisfaça as relações de restrição.

Uma das maneiras mais comuns de se realizar o raciocínio é por intermédio da composição entre relações, que foram a base da representação espacial. Por exemplo, considerando que existe uma relação binária rel_1 que é válida entre as entidades A e B , e que existe uma outra relação binária rel_2 , válida entre as entidades B e C , a composição de rel_1 com rel_2 deve ser capaz de restringir as possibilidades de relacionamentos entre A e C . Estas composições podem ser pré-computadas em uma tabela de composição (RENZ; NEBEL, 2007).

Cohn e Renz (2007) e Chen et al. (2015) classificam os formalismos do QSR em aspectos de mereotopologia², direção, distância, tamanho e formato das entidades espaciais. De maneira análoga, Ligozat (2013) divide os formalismos do QSR em duas principais famílias, uma que estuda as relações topológicas binárias entre regiões, sendo desenvolvida com fundamentação teórica na mereotopologia, e outra que utiliza como base entidades do espaço Euclidiano n -dimensional, como o ponto, o ponto orientado, o dipolo, os objetos geométricos no \mathbf{R}^2 e também regiões.

Por sua vez, Moratz, Nebel e Freksa (2003), Dylla (2009) e Dorr e Moratz (2014), também dividem as abordagens do QSR em dois grupos principais: raciocínio topológico sobre regiões e raciocínio posicional sobre configuração de pontos, composto por direção e distância.

Seguindo a classificação dada por Ligozat (2013), os principais métodos baseados em fundamentação mereotopológica, que trabalham com o relacionamento topológico entre as regiões, podem ser exemplificados com o Cálculo de Conexão de Regiões – *Region Connection Calculus* (RCC) (RANDELL; CUI; COHN, 1992), o Cálculo de Oclusão de Regiões – *Region Occlusion Calculus* (ROC) (RANDELL; WITKOWSKI; SHANAHAN, 2001), que é uma extensão do RCC, e o n -intersection (EGENHOFER; FRANZOSA, 1991).

Já os métodos que lidam com entidades espaciais Euclidianas são: o Cálculo de Direções Cardeais – *Cardinal Direction Calculus* (CDC) (FRANK, 1991; LIGOZAT, 1998), o DCC (FREKSA, 1992), o Cálculo Dipolo – *Dipole Calculus* (DRA) (MORATZ; RENZ; WOLTER, 2000), o Cálculo Estrela – *Star Calculus* ($STAR_m$) (RENZ; MITRA, 2004), o Cálculo de Configuração do Ponto Ternário – *Ternary Point Configuration Calculus* (TPCC) (MORATZ; NEBEL; FREKSA, 2003), a $OPRA_m$) (MORATZ, 2006; MOSSAKOWSKI; MORATZ, 2012). Podemos incluir também nessa categoria dois cálculos desenvolvidos mais recentemente: a Álgebra Relacional do Ponto Elevado Orientado – *Elevated Oriented Point Relation Algebra* ($EOPRA_m$) (MORATZ; WALLGRÜN, 2012) e o StarVars (SV_m) (LEE, 2013; LEE; RENZ; WOLTER, 2013).

A Tabela 1 exibe alguns dos principais parâmetros dos formalismos de QSR apresentados nos parágrafos anteriores.

Algumas das principais motivações para o uso do QSR estão no fato de que, muitas vezes, simples descrições espaciais podem ser suficientes para identificar objetos, seguir rotas, realizar localização e tomar decisões. Isto torna a abordagem qualitativa próxima da maneira como os humanos representam o espaço e de como raciocinam com conhecimento de senso comum. Além disto, o uso do método qualitativo possibilita o raciocínio com conhecimento incompleto (RENZ; NEBEL, 2007) e permite abstração do mundo físico (MORATZ, 2006).

Existe uma grande quantidade de possíveis aplicações para o Raciocínio Espacial Qualitativo, dentre as quais destacam-se a navegação robótica, a visão computacional de alto nível, as proposições espaciais semânticas com linguagem natural, o raciocínio com uso de senso comum

²Junção das palavras mereologia, que é o estudo das relações entre as partes e os conjuntos que elas formam, e topologia, que, na matemática, é o estudo de algumas propriedades de figuras que não variam quando suas formas geométricas são deformadas.

Tabela 1 – Principais formalismos do QSR citados neste trabalho

Nome do Cálculo	Entidade	Espaço	Aridade	Referência
StarVars _m (SV _m)	Pontos direcionados 2D	\mathbf{R}^2	Binária	Relativa
$\frac{EOPRA_m}{OPRA_m}$	Pontos direcionados 2D	$\mathbf{R}^2 \times \mathbf{R}^+$	Binária	Relativa
Dipolo	Pontos direcionados 2D	\mathbf{R}^2	Binária	Relativa
2D	Dipolos 2D	\mathbf{R}^2	Binária	Relativa
Direção Cardeal	Pontos	\mathbf{R}^2	Binária	Absoluta
Direção Cardeal	Regiões	\mathbf{R}^2	Binária	Absoluta
STAR _m	Pontos	\mathbf{R}^2	Binária	Absoluta
RCC-8	Regiões	Variável	Binária	Relativa
RCC-8 discreto	Regiões	Plano	Binária	Absoluta
ROC-20	Regiões	Variável	Binária	Relativa
Cruz Dupla	Pontos	\mathbf{R}^2	Ternária	Relativa
5-intersection	Regiões	plano/esfera	Ternária	Relativa
TPCC	Pontos	\mathbf{R}^2	Ternária	Relativa

Fonte: Autor “adaptado de” LIGOZAT, 2013

para sistemas pertencentes ao mundo real (COHN; RENZ, 2007) e em Sistemas de Informação Geográfica – *Geographic Information Systems* (GIS) (LEE; RENZ; WOLTER, 2011).

A navegação de agentes autônomos tem papel principal no desenvolvimento desse trabalho. Dessa maneira, o foco aqui será voltado aos raciocínios espaciais sobre direção relativa, uma vez que direções relativas são essenciais para agentes dotados de frente intrínseca navegam por um determinado domínio espacial (LEE, 2013; MORATZ, 2006; DYLLA; KREUTZMANN; WOLTER, 2014; DYLLA, 2009).

2.2.1 Raciocínio Sobre Direção Relativa

Os raciocínios sobre direção relativa envolvem relações de direção obtidas entre dois objetos com relação a um objeto de referência. No contexto da cognição espacial, o problema de direção relativa é também conhecido como relação projetiva (conforme demonstrado na seção 2.1.3), no qual as relações espaciais são representadas por meio das preposições direita, esquerda, frente, atrás etc. (HERSKOVITS, 1987; VORWERG et al., 1997; MORATZ; TENBRINK, 2006).

Segundo Retz-Schmidt (1988), as relações de direção podem ser classificadas em três classes, separadas de acordo com o sistema de referência que cada uma utiliza (Figura 4):

- a) intrínseco: A referência é determinada por um objeto, que determina a origem do sistema de coordenadas e a sua orientação.
- b) dístico (egocêntrico): A referência é baseada no próprio observador;
- c) extrínseco (absoluto / alocêntrico): A referência é determinada por fatores externos; tais como um objeto utilizado como um marco ou os pontos cardeais.

Figura 4 – Sistemas de referência espaciais para relações de direção

- (a) Intrínseco: a bola está na frente da cadeira (b) Dítico: a bola está a direita da cadeira (c) Extrínseco: a bola está ao sul da cadeira



Fonte: KRANJEC, 2006

Os raciocínios qualitativos sobre direção relativa mais conhecidos, normalmente são díticos ou intrínsecos, como por exemplo o Flip-Flop (LR) e o DCC que podem ser considerados díticos, enquanto o DRA, a $OPRA_m$ e o SV_m que podem ser considerados intrínsecos (LEE, 2013). Um exemplo de cálculo considerado extrínseco pode ser o $ST AR_m$, que é um cálculo de direção absoluta. Cálculos de direção absoluta não são o foco deste trabalho, contudo o $ST AR_m$ será brevemente apresentado na seção 2.2.1.5, pois ele é a base do SV_m .

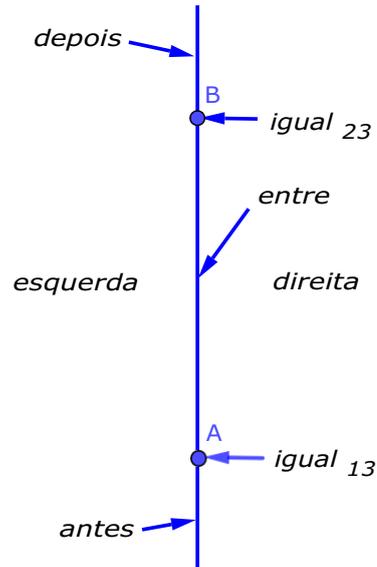
2.2.1.1 Cálculos Flip-Flop e LR

O cálculo Flip-Flop é uma das formas mais simples de se raciocinar com relações de direção (LEE, 2013). No cálculo Flip-Flop (LIGOZAT, 1993) as relações são definidas com base em um sistema gerado por uma linha que conecta dois pontos. A posição de um terceiro ponto é, então, categorizada com sendo *direita* ou *esquerda* com relação ao segmento de reta que une os dois pontos. Cinco outras categorizações são estabelecidas caso o terceiro ponto esteja posicionado na reta. São eles: *depois*, *antes*, *entre*, *igual₂₃* e *igual₁₃*. O cálculo LR (SCIVOS; NEBEL, 2005) é a extensão do Flip-Flop que considera os casos corrompidos em que a posição dos dois primeiros pontos é coincidente. A Figura 5 exhibe as relações encontradas no cálculo Flip-Flop.

2.2.1.2 Cálculo da Cruz Dupla – DCC

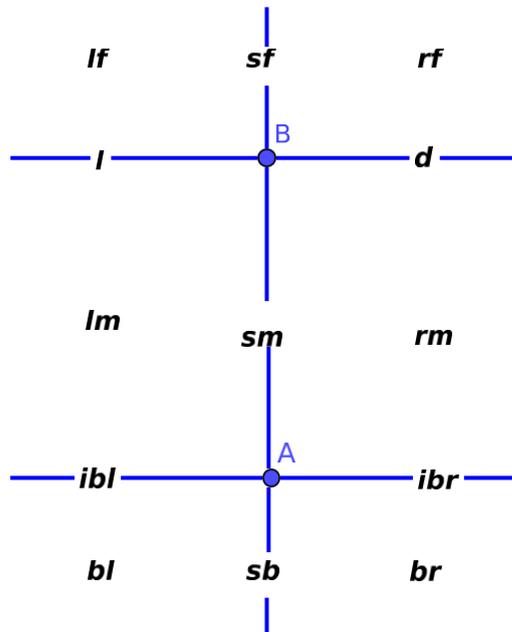
O DCC (FREKSA, 1992) pode ser considerado um refinamento do LR (LEE; RENZ; WOLTER, 2013). No DCC as relações *direita* e *esquerda* são refinadas por mais duas linhas ortogonais, conforme pode ser visto na Figura 6, onde *l,s,r,f,i,b,m* representam, respectivamente, *left* (esquerda), *straight* (reto), *right* (direita), *front* (frente), *identical* (idêntico), *back* (atrás), *middle* (meio).

Figura 5 – Relações encontradas no cálculo Flip-Flop



Fonte: Autor “adaptado de” LIGOZAT, 1993

Figura 6 – Representação do DCC

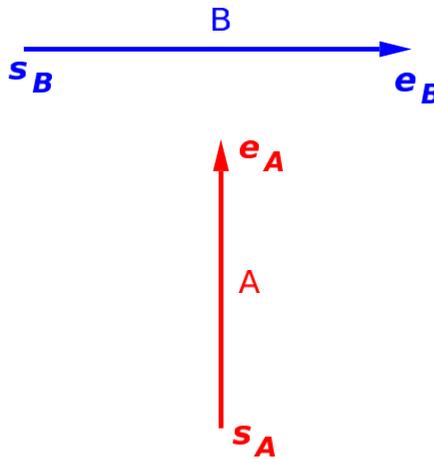


Fonte: Autor “adaptado de” FREKSA, 1992

2.2.1.3 Cálculo Dipolo – DRA

Dipolos são segmentos orientados de retas formados por dois pontos, um inicial e um final. Os dipolos são indicados por letras maiúsculas, como A, B, C, D , um ponto inicial, s_A , e um final, e_A , conforme Figura 7 (MORATZ; RENZ; WOLTER, 2000). Este cálculo também faz uso do conceito de granularidade, que pode mudar dependendo da quantidade de relações usadas entre os dipolos, por exemplo, *direita, esquerda, fim e início*. Pelo fato de trabalhar com entidades básicas de orientação, o DRA é a base da $OPRA_m$, que será apresentada na sequência.

Figura 7 – Relação de orientação entre dois dipolos

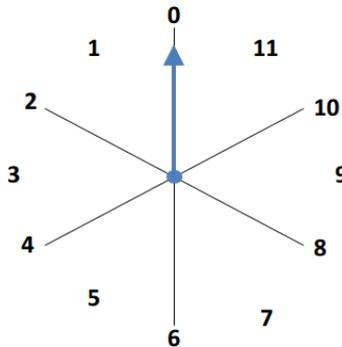
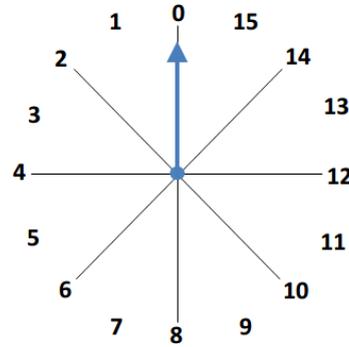


Fonte: Autor “adaptado de” MORATZ; RENZ; WOLTER, 2000

2.2.1.4 A Álgebra do Ponto Orientado – $OPRA_m$

A Álgebra do Ponto Orientado – *Oriented Point Algebra* ($OPRA_m$) é um cálculo qualitativo no qual objetos são representados como simples pontos orientados, sendo que cada ponto pode ser representado por coordenadas Cartesianas x e y e uma orientação θ . A $OPRA_m$ foi desenvolvida com base no Cálculo Dipolo (DRA) e é o cálculo qualitativo sobre direções relativas mais expressivo (LEE, 2013), podendo generalizar todos os cálculos vistos anteriormente. A $OPRA_m$ tem o parâmetro de granularidade arbitrária $m \in \mathbb{N}$, que pode ser útil quando uma distinção mais fina da orientação é necessária. A granularidade m é usada para a obtenção da resolução angular, que é igual a $\frac{2\pi}{2^m}$ (MOSSAKOWSKI; MORATZ, 2012).

A Figura 8 exibe a granularidade para um ponto orientado com m igual a 3 e 4, como exemplo. É possível verificar também pela Figura 8 que um número i é dado a cada região, seja

Figura 8 – Pontos orientados com m igual a 3 e 4(a) Ponto orientado com $m = 3$ (b) Ponto orientado com $m = 4$ 

Fonte: Autor “adaptado de” MORATZ, 2006

ela uma seção linear ou plana. A contagem começa sempre do **0**, na seção linear que representa a direção do ponto.

Assim, cada direção do ponto orientado é numerada como um intervalo de ângulos pertencentes ao grupo cíclico \mathbf{Z}_{4m} , de acordo com a Equação (11) (MOSSAKOWSKI; MORATZ, 2012).

$$[i]_m = \begin{cases} \lfloor 2\pi \frac{i-1}{4m}, 2\pi \frac{i+1}{4m} \rfloor & \text{se } i \text{ for ímpar} \\ \lfloor 2\pi \frac{i}{4m}, 2\pi \frac{i+2}{4m} \rfloor & \text{se } i \text{ for par.} \end{cases} \quad (11)$$

A $OPRA_m$ é uma álgebra relativa que não utiliza referência externa. Assim, se as coordenadas Cartesianas de dois pontos orientados, A e B , por exemplo, forem diferentes, a relação entre os pontos é representada por $A_m \angle_i^j B$ ($i, j \in \mathbf{Z}_{4m}$). Isto significa que, dada a granularidade m , a posição relativa de B com relação a A é descrita por i e a posição relativa de A com relação a B é j (MORATZ, 2006). Por exemplo, a relação entre os pontos A e B da figura 9 é representado por $A_4 \angle_{11}^1 B$.

Na $OPRA_m$, assim como em outros métodos de QSR, o raciocínio é feito por meio de uma tabela de composição. Para o caso da $OPRA_m$, a tabela é construída com o conjunto de todas as relações entre três pontos orientados, por exemplo, $A_m \angle_i^j B$, $B_m \angle_l^k C$ e $A_m \angle_t^s C$, sendo i, j, k, l, s, t as variáveis que descrevem as configurações entre os pontos orientados (MORATZ, 2006).

Desta maneira, Mossakowski e Moratz (2012) apresentaram um algoritmo que utiliza simples regras geométricas para computar a tabela de composição na $OPRA_m$. Outros algoritmos já existiam na literatura antes deste, porém, ou eles eram incompletos, como o algoritmo de Moratz (2006) ou muito complexos, como o apresentado por Frommberger et al. (2007).

O algoritmo de Mossakowski e Moratz (2012) trabalha com dois predicados principais para verificar se a composição entre três pontos orientados não coincidentes é verdadeira: *completa_volta_m* e *forma_triângulo_m*. O predicado *completa_volta_m* verifica se as relações de um ponto orientado para com os outros dois pontos podem formar voltas completas quando

Algoritmo 4 – Verificando se as relações entre os pontos orientados formam uma composição válida da $OPRA_m$ (MOSSAKOWSKI; MORATZ, 2012).

1 **início**

2 $completa_volta_m(o,p,q) \Leftrightarrow |(o + p + k + 2m) \bmod 4m - 2m|^{TM} (o \bmod 2) \times (p \bmod 2)$

3 $senal_m(q) = \begin{cases} 0, & \text{se } (q \bmod 4m = 0) \vee (q \bmod 4m = 2m) \\ 1, & \text{se } (q \bmod 4m < 2m) \\ -1, & \text{caso contrário} \end{cases}$

4 $forma_triangulo_m(u,v,w) \Leftrightarrow completa_volta(u,v,w - 2m) \wedge (u,v,w) \neq (2m, 2m, 2m) \wedge senal_m(u) = senal_m(v) = senal_m(w)$

5 $opra(m \angle i_m \angle k_m \angle s) \Leftrightarrow completa_volta_m(i,k, -s)$

6 $opra(m \angle i_m \angle k_m \angle s^t) \Leftrightarrow falso$

7 $opra(m \angle i_m \angle k_m^l \angle s) \Leftrightarrow falso$

8 $opra(m \angle i_m \angle k_m^l \angle s^t) \Leftrightarrow completa_volta_m(i,k, -s)$

9 $opra(m \angle i_m^j \angle k_m \angle s) \Leftrightarrow falso$

10 $opra(m \angle i_m^j \angle k_m \angle s^t) \Leftrightarrow completa_volta_m(t,k, -j)$

11 $opra(m \angle i_m^j \angle k_m^l \angle s) \Leftrightarrow completa_volta_m(i, -l, -s)$

12 $opra(m \angle i_m^j \angle k_m^l \angle s^t) \Leftrightarrow \exists \mathbf{0}^{TM} u,v,w < 4m. completa_volta_m(u,i, -s) \wedge completa_volta_m(v,k, -j) \wedge completa_volta_m(w,t, -l) \wedge forma_triangulo(u,v,w)$

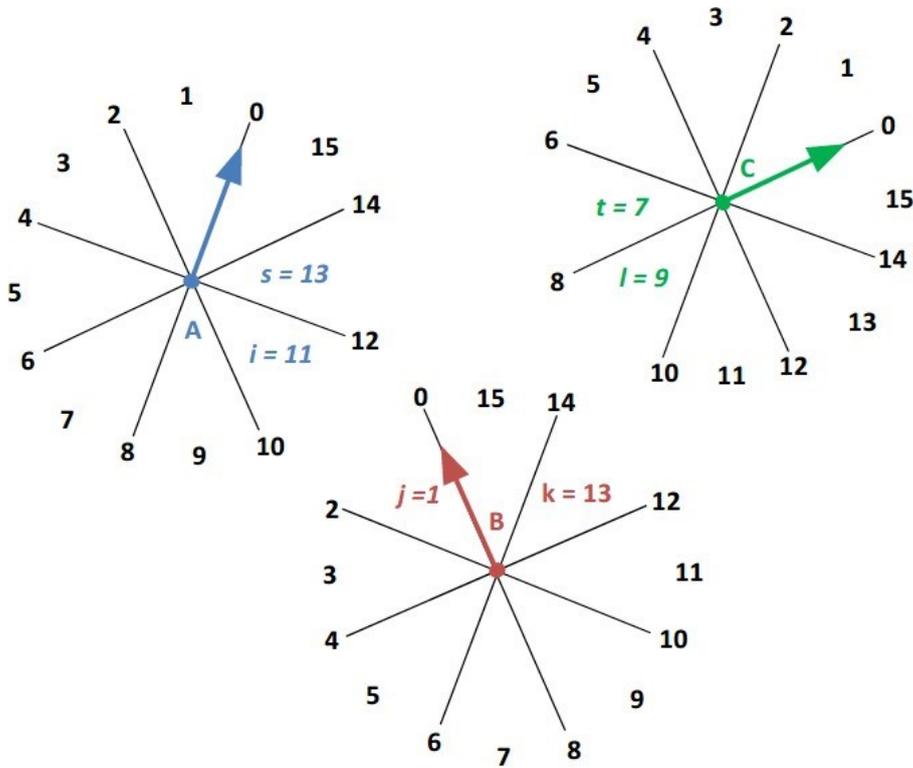
13 **fim**

somadas a um terceiro parâmetro, representado por u, v ou w , que será o ângulo, em notação abstrata baseada em m , de um dos três vértices do triângulo formado pelos pontos A, B e C , cujos ângulos internos são, respectivamente α, β e γ . Assim, u, v e $w \in \mathbf{Z}_{4m}$ de tal forma que $\alpha \in [u]_m, \beta \in [v]_m$ e $\gamma \in [w]_m$. A prova existencial de parâmetros u, v e w consistentes é feita por meio do predicado $forma_triangulo_m$ (MOSSAKOWSKI; MORATZ, 2012).

O Algoritmo 4, proposto por Mossakowski e Moratz (2012), calcula as composições na $OPRA_m$. Por ser completo, esse algoritmo considera os casos nos quais os pontos não são coincidentes, assim como os casos nos quais os pontos são coincidentes. O predicado ternário $opra$ computa as composições das relações da $OPRA_m$. Nesse algoritmo, o predicado $completa_volta_m$ foi adaptado para receber números inteiros convencionais, incluindo os negativos, e para ser diretamente implementado como uma equação, e não como uma estrutura condicional.

As linhas 5, 6, 7, 8, 9, 10 e 11 do Algoritmo 4 computam as composições para os casos de pontos orientados coincidentes. Por sua vez, a linha 12 computa as composições para os pontos não coincidentes. Esse algoritmo permite a inferência da relação existente entre dois pontos orientados por meio da composição das outras relações presentes, que devem ser conhecidas. Por exemplo, a Figura 9 exibe a composição das relações $A_4 \angle^1 B$ e $B_4 \angle^9 C$ que resulta na inferência de uma possível relação entre os pontos A e C .

Figura 9 – A composição das relações $A_4 \angle_{11}^1 B$ e $B_4 \angle_{13}^9 C$, pode ter como uma solução $A \angle_{13}^7$ na $OPRA_4$ $4 \ 13 C$



Fonte: Autor “adaptado de” MOSSAKOWSKI; MORATZ, 2012

2.2.1.4.1 Inconvenientes da $OPRA_m$

Apesar da $OPRA_m$ ser um cálculo qualitativo sobre direções relativas de extrema importância, pois é o mais expressivo (LEE, 2013), o seu uso pode não ser eficaz pois traz diversas desvantagens, como por exemplo:

- O Algoritmo 4 pode retornar uma disjunção de relações como resultado de uma determinada composição. E, dependendo do domínio, a disjunção de relações pode conter todas as possíveis relações existentes, O que pode invalidar algum tipo de decisão sobre navegação, por exemplo (PERICO et al., 2016b).
- A $OPRA_m$ não pode determinar um modelo de mundo completo (LEE, 2013).
- A $OPRA_m$ não consegue verificar a consistência da entrada de dados (LEE, 2013).
- A $OPRA_m$ não permite o raciocínio espacial com o uso de pontos de referência (LEE, 2013), embora seja possível adaptar as relações para usar algum tipo de marco, como apresentado em Perico et al. (2016b).
- O cálculo realizado por intermédio da tabela de composição da $OPRA_m$ permite o raciocínio com apenas três entidades espaciais de cada vez.

Com a finalidade de superar estes problemas, Lee (2013) propôs o cálculo ST AR com Interpretação Variável de Orientações – StarVars_m, apresentado a seguir.

2.2.1.5 O Cálculo StarVars_m

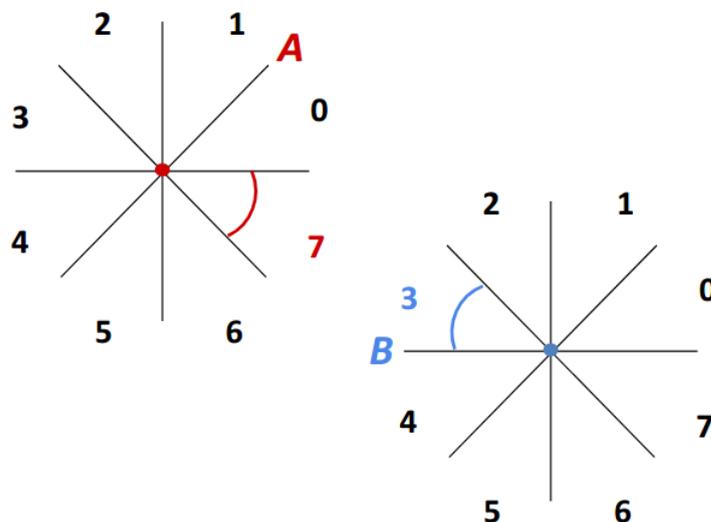
O Cálculo ST AR com Interpretação Variável de Orientações – StarVars_m (*STAR calculus with VARIable interpretation of orientation*) é um cálculo qualitativo para relações provenientes de direções relativas baseado no cálculo de direção absoluta ST AR_m (RENZ; MITRA, 2004). O StarVars_m foi desenvolvido para superar os problemas da OPRA_m, apresentados na seção 2.2.1.4.1, por isso será o método utilizado no decorrer deste trabalho.

2.2.1.5.1 O cálculo de direção absoluta ST AR_m

O cálculo ST AR_m (RENZ; MITRA, 2004) é definido no domínio \mathbf{R}^2 , é extrínseco, binário e suas entidades espaciais não possuem orientação. As direções qualitativas no ST AR_m são provenientes de setores angulares definidos com base na granularidade m ($m \in \mathbf{N}$ e $m \geq 2$), onde o tamanho angular de cada setor é definido como $360^\circ/m$. A Figura 10 ilustra as relações entre os pontos A e B no ST AR₈.

Um setor do ST AR_m é composto pela linha que o antecede e pelo próprio setor angular. As linhas são numeradas de 0 a $m - 1$, em sentido anti-horário. Dessa maneira, Lee (2013) utiliza a notação $[a, b[$ para representar o setor angular delimitado pelas linhas a e b . Esta notação permite representar também a união de setores angulares, nas quais as relações formadas são denominadas relações convexas e são válidas desde que o ângulo central resultante da união

Figura 10 – ST AR₈: $A(7)B \wedge B(3)A$



Fonte: Autor “adaptado de” LEE, 2013

dos setores seja menor ou igual a 180° . Por exemplo, considerando o ponto A da Figura 10, a relação $[6,2[$ é convexa, já uma relação $[6,3[$ não é, uma vez que a união dos setores em sentido anti-horário resulta em 225° .

O raciocínio com o cálculo $STAR_m$ pode ser realizado com base em tabelas de composição (RENZ; MITRA, 2004). Porém, Lee (2013) prova que esse mesmo raciocínio pode ser eficientemente realizado por meio de Programação Linear – *Linear Programming* (LP).

A Programação Linear (GUPTA, R. K., 2009) é um método utilizado para resolver problemas de otimização que é muito reconhecido e tem uso em diversas áreas. A LP é composta por uma função linear objetivo e por um sistema de equações e inequações lineares, denominadas restrições. O objetivo da LP é minimizar (maximizar) a função objetivo respeitando as restrições. Quando a função objetivo é constante, como é o caso utilizado no raciocínio com o cálculo $STAR_m$, resolver a LP significa somente resolver o sistema de equações e inequações lineares (LEE, 2013). A Programação Linear permite a solução de problemas em tempo polinomial (SCHRIJVER, 1986).

Dessa forma, Lee (2013) demonstra que é possível formular uma relação convexa do $STAR_m$ como um conjunto de inequações lineares. As inequações são construídas com base na álgebra vetorial, que estabelece a conexão entre a determinante de dois vetores e suas direções relativas da seguinte forma: dados dois vetores $\vec{v}, \vec{u} \in \mathbf{R}^2$ e que estes vetores formam uma matriz 2×2 com \vec{v}, \vec{u} sendo as colunas da matriz, então \vec{u} está para a esquerda de \vec{v} se, e somente se, a determinante destes dois vetores, for maior do que zero, $\det(\vec{v}, \vec{u}) > 0$. O vetor \vec{u} é paralelo a \vec{v} se, e somente se, a determinante destes dois vetores for igual a zero, $\det(\vec{v}, \vec{u}) = 0$.

Então, dado o vetor $\vec{v} = (x_2, y_2) - (x_1, y_1) \in \mathbf{R}^2$, que tem posição inicial em (x_1, y_1) e posição final em (x_2, y_2) , e que (x_1, y_1) e (x_2, y_2) são as posições dos pontos A e B em \mathbf{R}^2 , respectivamente, e um vetor unitário $u(A, s)$ que tem a mesma orientação da linha s do ponto A , isto é, $u(A, s) = (\cos(\theta_1 + s\eta), \sin(\theta_1 + s\eta))$, onde θ_1 é a orientação de A e $\eta = \frac{360^\circ}{m}$. A relação $A(s)B$ é satisfeita, se, e somente se, \vec{v} estiver para a esquerda ou paralela a $u(A, c)$ e \vec{v} estiver para a direita de $u(A, d)$, considerando a relação convexa $A[c, d]B$. Assim, $\det(u(A, c), \vec{v}) \geq 0$ e $\det(u(A, d), \vec{v}) < 0$, que é equivalente ao sistema de inequações

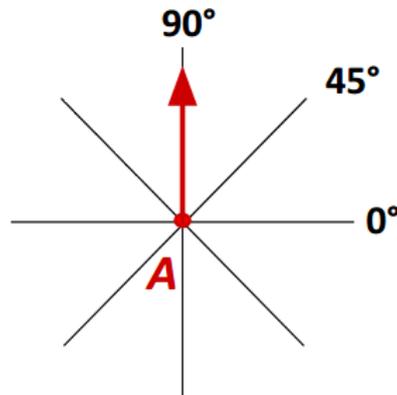
$$-\sin(c\eta)x_1 + \sin(c\eta)x_2 + \cos(c\eta)y_1 - \cos(c\eta)y_2 \geq 0, \quad (12)$$

$$\sin(d\eta)x_1 - \sin(d\eta)x_2 - \cos(d\eta)y_1 + \cos(d\eta)y_2 < 0. \quad (13)$$

Contudo, a Programação Linear utiliza somente inequações não-estritas, como é o caso da Inequação (12). Assim, a Inequação (13) precisa ser adaptada para ser menor ou igual a um valor negativo próximo de 0 , que será chamado de s , onde $s < 0$.

2.2.1.5.2 O StarVars_m

O StarVars_m é definido no domínio $\mathbf{R}^2 \times \Theta_m$, onde, assim como no $STAR_m$, m representa a granularidade ($m \in \mathbf{N}$ e $m \geq 2$) e $\Theta_m = \{0\eta, 1\eta, 2\eta, \dots, (m-1)\eta\}$ representa o

Figura 11 – StarVars₈: $\theta_A = 2\eta = 90^\circ$ 

Fonte: Autor “adaptado de” LEE, 2013

domínio da orientação com $\eta = 360^\circ/m$. Perceba que, diferentemente da $OPRA_m$, o domínio de orientação da $StarVars_m$ não está definido em $[0, 2\pi)$. Um objeto espacial $StarVars_m$ é composto por (x, y, θ) , onde $(x, y) \in \mathbf{R}^2$ determinam a posição e $\theta \in \Theta_m$ determina a orientação (LEE, 2013). A Figura 11 demonstra o ponto orientado A , como um exemplo de objeto $StarVars_8$, com $\theta_A = 2\eta = 90^\circ$.

As relações do $StarVars_m$ tem os mesmos significados do $STAR_m$. Contudo, essas relações são todas interpretadas com base nas orientações individuais de cada objeto espacial, onde a linha $\mathbf{0}$ está sempre alinhada com a orientação θ . Desta forma, o raciocínio com o $StarVars_m$ também pode ser realizado por meio da LP, com a inclusão da orientação do objeto espacial nas Inequações (12) e (13), resultando nas Inequações (14) e (15) (LEE; RENZ; WOLTER, 2013), considerando a relação convexa $[c, d]$ entre dois objetos espaciais.

$$-sen(c\eta + \theta_1)x_1 + sen(c\eta + \theta_1)x_2 + cos(c\eta + \theta_1)y_1 - cos(c\eta + \theta_1)y_2 \text{ TM } \mathbf{0}, \quad (14)$$

$$sen(d\eta + \theta_1)x_1 - sen(d\eta + \theta_1)x_2 - cos(d\eta + \theta_1)y_1 + cos(d\eta + \theta_1)y_2 \text{ TM } s. \quad (15)$$

Assim, uma vez que tenhamos como entrada a granularidade m , a quantidade de entidades espaciais n e o conjunto das relações de direção entre todas as entidades, que será chamado de φ , sendo que $\varphi = \wedge_{i,j} v_i R_{ij} v_j$, onde $i, j = 1, \dots, n, i \neq j$, pode-se utilizar as Inequações (14) e (15) para se procurar as orientações de todas as entidades espaciais do problema por meio de algum sistema de busca. A LP retorna, então, um modelo de mundo $\psi = \wedge_{i \in \{1, \dots, n\}} (x_i, y_i, \theta_i)$ assim que o primeiro conjunto de orientações que satisfaçam o sistema de inequações seja encontrado. Caso contrário, a LP continua a busca e retorna *falha* se nenhum conjunto de orientações satisfizer as inequações. Dessa forma, o $StarVars_m$ retorna um modelo completo de mundo, com x, y e θ de cada entidade espacial orientada e x e y de cada ponto de referência não orientado.

Um ponto importante do StarVars_m é que ele faz a verificação de consistência das entradas enquanto realiza o raciocínio, pois, pode-se afirmar que as entradas são consistentes se o sistema retornar um modelo de mundo qualquer. No caso do sistema não retornar nenhum modelo, pode-se dizer que as entradas são inconsistentes.

2.2.1.5.3 Exemplo: Inferência de Orientações Utilizando o StarVars_8

Esse exemplo ajuda a compreender melhor como utilizar as Inequações (14) e (15) para realizar inferências, além de elucidar as entradas e as saídas do StarVars_m .

Supondo um sistema com três entidades espaciais, onde duas têm orientações – A e B – e uma é um ponto de referência – C . Considerando $m = 8$, as relações de direção qualitativas entre as entidades é a seguinte:

$$\varphi = \{A(\mathbf{1})B \wedge B(\mathbf{0})A \wedge A(\mathbf{0})C \wedge B(\mathbf{2})C\}$$

Uma vez que se tem a relação exata entre todas as entidades, o valor entre parênteses é o c da Inequação (14). Já o d , da Inequação (15), para esse caso, é igual a $c + 1$.

Dado que $m = 8$, $\eta = 360^\circ/8 = 45^\circ$, então o domínio das orientações será definido por $\Theta_8 = \{0, 45^\circ, 90^\circ, 135^\circ, \dots, 315^\circ\}$.

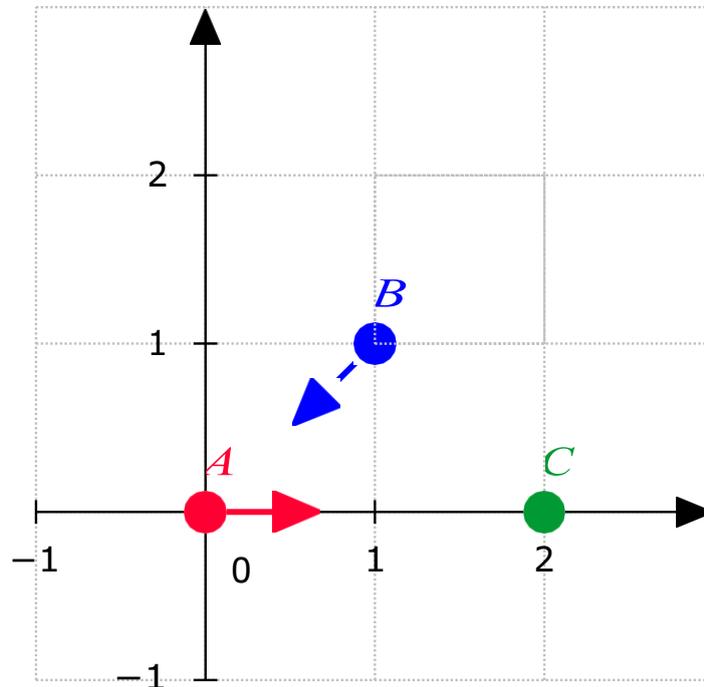
Pode-se então, com essas entradas, montar um sistema com as Inequações (14) e (15), onde as variáveis serão as orientações θ_A e θ_B , dos pontos A e B :

$$\begin{aligned} & -\text{sen}(\mathbf{1}.\eta + \theta_A)_{x_A} + \text{sen}(\mathbf{1}.\eta + \theta_A)_{x_B} + \mathbf{0}_{x_C} + \text{cos}(\mathbf{1}.\eta + \theta_A)_{y_A} - \text{cos}(\mathbf{1}.\eta + \theta_A)_{y_B} + \mathbf{0}_{y_C} \text{ TM } \mathbf{0}, \\ & \text{sen}(\mathbf{2}.\eta + \theta_A)_{x_A} - \text{sen}(\mathbf{2}.\eta + \theta_A)_{x_B} + \mathbf{0}_{x_C} - \text{cos}(\mathbf{2}.\eta + \theta_A)_{y_A} + \text{cos}(\mathbf{2}.\eta + \theta_A)_{y_B} + \mathbf{0}_{y_C} \text{ TM } s, \\ & \text{sen}(\mathbf{0}.\eta + \theta_B)_{x_A} - \text{sen}(\mathbf{0}.\eta + \theta_B)_{x_B} + \mathbf{0}_{x_C} - \text{cos}(\mathbf{0}.\eta + \theta_B)_{y_A} + \text{cos}(\mathbf{0}.\eta + \theta_B)_{y_B} + \mathbf{0}_{y_C} \text{ TM } \mathbf{0}, \\ & -\text{sen}(\mathbf{1}.\eta + \theta_B)_{x_A} + \text{sen}(\mathbf{1}.\eta + \theta_B)_{x_B} + \mathbf{0}_{x_C} + \text{cos}(\mathbf{1}.\eta + \theta_B)_{y_A} - \text{cos}(\mathbf{1}.\eta + \theta_B)_{y_B} + \mathbf{0}_{y_C} \text{ TM } s, \\ & -\text{sen}(\mathbf{0}.\eta + \theta_A)_{x_A} + \mathbf{0}_{x_B} + \text{sen}(\mathbf{0}.\eta + \theta_A)_{x_C} + \text{cos}(\mathbf{0}.\eta + \theta_A)_{y_A} + \mathbf{0}_{y_B} - \text{cos}(\mathbf{0}.\eta + \theta_A)_{y_C} \text{ TM } \mathbf{0}, \\ & \text{sen}(\mathbf{1}.\eta + \theta_A)_{x_A} + \mathbf{0}_{x_B} - \text{sen}(\mathbf{1}.\eta + \theta_A)_{x_C} - \text{cos}(\mathbf{1}.\eta + \theta_A)_{y_A} + \mathbf{0}_{y_B} + \text{cos}(\mathbf{1}.\eta + \theta_A)_{y_C} \text{ TM } s, \\ & \mathbf{0}_{x_A} - \text{sen}(\mathbf{2}.\eta + \theta_B)_{x_B} + \text{sen}(\mathbf{2}.\eta + \theta_B)_{x_C} + \mathbf{0}_{y_A} + \text{cos}(\mathbf{2}.\eta + \theta_B)_{y_B} - \text{cos}(\mathbf{2}.\eta + \theta_B)_{y_C} \text{ TM } \mathbf{0}, \\ & \mathbf{0}_{x_A} + \text{sen}(\mathbf{3}.\eta + \theta_B)_{x_B} - \text{sen}(\mathbf{3}.\eta + \theta_B)_{x_C} + \mathbf{0}_{y_A} - \text{cos}(\mathbf{3}.\eta + \theta_B)_{y_B} + \text{cos}(\mathbf{3}.\eta + \theta_B)_{y_C} \text{ TM } s. \end{aligned}$$

Então, verifica-se se o sistema pode ser satisfeito para diferentes valores de θ_A e θ_B . Assim que valores que satisfaçam o sistema forem encontrados, um modelo de mundo, com x e y de todas as entidades será retornado, além das orientações das entidades que não forem pontos de referência.

Os valores das orientações podem ser buscados por diferentes técnicas, como força bruta, onde todos os valores possíveis serão testados, até que uma resposta seja encontrada, busca por profundidade e busca hierárquica (LEE; RENZ; WOLTER, 2013), em que ambas levam em consideração a propriedade das relações convexas.

Figura 12 – Representação gráfica de uma possível resposta obtida pelo StarVars_s, dada a entrada φ



Fonte: Autor

Existem diversos métodos para se resolver problemas de LP. Dois dos mais tradicionais são o *simplex* (DANTZIG, 1990) e o ponto interior (KARMAKAR, 1984). Nesse trabalho, todas as soluções utilizarão o método *simplex*, cuja implementação existe em diversas bibliotecas de diferentes linguagens de programação. Para o problema do exemplo, utilizando o *simplex* e $s = -1$, uma possível resposta seria:

- a) $x_A = 0.0$; $y_A = 0.0$ e $\theta_A = 0^\circ$;
- b) $x_B = 1.0$; $y_B = 1.0$ e $\theta_B = 225^\circ$;
- c) $x_C = 2.0$; $y_C = 0.0$

Essa resposta pode ser representada graficamente, conforme exibido na Figura 12.

2.2.1.6 Uso de Referências Espaciais Linguísticas nos Cálculos sobre Direções Relativas

Humanos normalmente utilizam linguagem qualitativa para transmitir conhecimento espacial (MORATZ; TENBRINK, 2006) e, uma das maneiras de se fazer isso, é por meio das preposições espaciais projetivas.

Apesar de alguns cálculos qualitativos sobre direções relativas utilizarem números na nomeação de suas regiões, como é o caso da OPRA_m e do StarVars_m, muitos utilizam referências espaciais linguísticas para delimitar regiões, como o Flip-Flop, o DCC e o DRA, além de

outros cálculos, não abordados diretamente neste trabalho, como o TPCC, que trata distâncias qualitativas além de direções relativas.

O uso de referências espaciais linguísticas nos cálculos qualitativos busca melhorar a interação humano-máquina / humano-robô com relação à transmissão de conhecimento espacial (MORATZ; TENBRINK, 2006).

2.3 COMENTÁRIOS

A seção 2 descreveu os conceitos básicos da navegação de agentes móveis, por meio do uso de técnicas probabilísticas, com foco na localização, mapeamento e planejamento de movimento. Além disso, o QSR foi brevemente descrito e, de maneira mais aprofundada, algumas técnicas que utilizam direções relativas foram abordadas, onde o foco principal foi dado ao raciocínio $StarVars_m$, que é o cálculo que será usado na proposta deste trabalho.

A próxima seção relacionará os trabalhos que mais se correlacionam com a proposta que será apresentada na seção 4.

3 TRABALHOS CORRELATOS

Esta seção apresenta o estado da arte para três classes de problemas que são diretamente correlacionadas com as propostas que serão apresentadas neste trabalho:

- a) Problemas de navegação utilizando técnicas puramente qualitativas são apresentados na seção 3.1;
- b) Aplicação de técnicas híbridas – combinando métodos probabilísticos com raciocínios/representações qualitativas – para solucionar problemas de localização robótica são demonstradas na seção 3.2 e;
- c) Navegação guiada, sem o uso de técnicas qualitativas, em que um problema é exibido na seção 3.3.

3.1 NAVEGAÇÃO DE AGENTES AUTÔNOMOS POR MEIO DE ESTRATÉGIAS PURAMENTE QUALITATIVAS

Nesta seção, duas estratégias qualitativas de navegação são exibidas. As duas estratégias utilizam a $OPRA_m$ como mecanismo de raciocínio espacial, porém a granularidade m é diferente em cada uma delas.

3.1.1 Robô Quadrúpede Utilizando a $OPRA_m$

Ao propor a $OPRA_m$, Moratz (2006) demonstrou como integrar conhecimento local com a $OPRA_6$, utilizando, para isso, o robô móvel quadrúpede Aibo¹ da Sony. Durante o experimento realizado, o robô era capaz de distinguir entre cores e objetos simples por meio de visão computacional realizada com um sistema monocular. Além disso, o robô era capaz de se mover para frente e para os lados.

Apesar de fazer uso de visão computacional, o robô não teve nenhum conhecimento prévio sobre o tamanho dos objetos e também não sabia se o chão era uma superfície completamente plana. Desta forma, o robô não podia estimar as distâncias dos objetos. A única informação disponível, então, era a orientação local do robô em relação aos objetos, conforme exibido na Figura 13 (MORATZ, 2006).

Com o intuito de analisar o funcionamento do raciocínio com a $OPRA_6$, foi dada, ao robô, uma tarefa de navegação: “vá para o cubo amarelo depois do disco vermelho”. A configuração inicial do sistema pode ser vista na Figura 14, sendo $t1$ o ponto inicial. As relações iniciais percebidas pelo robô foram: $t1_d \angle^0_{23} t2_1$, $t1_d \angle^?_{23} t1$ e $t1_d \angle^0_{23} t2$ (MORATZ, 2006), onde $t1_d$ representa que em $t1$ o robô se orienta para o disco d , ou seja, a relação 0 de $t1$ aponta para d . O mesmo acontece para $t2_1$, na qual a relação 0 de $t2$ aponta para $t1$.

¹<http://www.sony-aibo.co.uk/>

Figura 13 – Aibo durante o experimento



Fonte: MORATZ, 2006

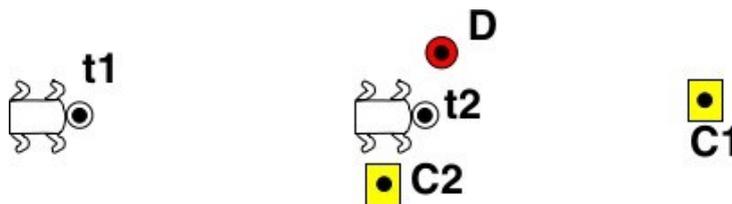
Segundo Moratz (2006), para que a tarefa dada fosse cumprida, uma das duas relações $D \stackrel{?}{\sim}_{[9,15]} C1$ ou $D \stackrel{?}{\sim}_{[9,15]} C2$ deveria ser válida.

Como o robô não conhecia os tamanhos dos objetos, o conhecimento adquirido em $t1$ não foi suficiente para ajudá-lo a distinguir as posições de ambos os cubos com relação ao disco. Então, o robô precisou se movimentar para frente, até $t2$, para adquirir mais conhecimento. Como o robô se movimentou em linha reta e para frente, a posição de $t1$ manteve-se conhecida (MORATZ, 2006).

Em $t2$ o robô pôde adquirir o seguinte conhecimento: $t2 \stackrel{0}{\sim}_{17} D_{t2}$, $t2 \stackrel{?}{\sim}_{19} C1$ e $t2 \stackrel{?}{\sim}_{11} C2$, pelo qual, por meio da tabela de composição das relações, pôde-se inferir a relação de $t1$ para $D_{t2} \Rightarrow t1 \stackrel{[19,21]}{\sim}_0 D_{t2}$.

Depois de obter a relação $t1 \stackrel{[19,21]}{\sim}_0 D_{t2}$, Moratz (2006) pôde inverter e reorientar D para $t1$, obtendo, então, a relação de D_{t1} para $t2 \Rightarrow D_{t1} \stackrel{0}{\sim}_{[3,5]} t2$. Com essa nova relação,

Figura 14 – Duas perspectivas do Aibo durante o experimento realizado: $t1$ e $t2$



Fonte: MORATZ, 2006

pôde-se inferir mais duas relações por meio da composição, D_{t1} com $C1 \Rightarrow D_{t1} \top C1$ e D_{t1} com $C2 \Rightarrow D_{t1} \text{ } \underset{[1,5]}{\text{ } } C2$.

Conforme descrito anteriormente, a relação de $C2$, descrita no parágrafo anterior, eliminou a possibilidade do cubo 2, $C2$, ser o cubo que estava depois do disco vermelho. A relação encontrada para $C1$ não permitiu nenhuma conclusão, porém ela permitiu a composição da restrição tautológica $d_{t1} \text{ } \underset{0}{\text{ } } t1_d$ com a relação $t1_d \text{ } \underset{23}{\text{ } } C1$, resultando na relação $d_{t1} \text{ } \underset{1}{\text{ } } C1$. Essa última relação permitiu a inferência de que o cubo 1, $C1$, era o único cubo com chances de ser o cubo atrás do disco.

É digno de nota esclarecer que Moratz (2006) não utilizou o Algoritmo 4, apresentado na seção 2.2.1.4, para realizar as composições do experimento apresentado nessa seção, pois, esse algoritmo, que é mais completo para composições na $OPRA_m$, foi apresentado somente em 2012 (MOSSAKOWSKI; MORATZ, 2012). De qualquer maneira, o Algoritmo 4 também resolve as composições apresentadas nesta seção.

3.1.2 Formalizando as Regras de Prevenção de Colisões da Navegação Marítima por meio da $OPRA_4$

Dylla et al. (2007) utilizaram os conceitos da $OPRA_4$ para formalizar as Regras de Prevenção de Colisões na Navegação Marítima – *International Regulations for Preventing Collisions at Sea* (ColRegs) publicadas pela Organização Marítima Internacional – *International Maritime Organization* (IMO), com o intuito de tornar a navegação mais segura.

Segundo Dylla et al. (2007), o cumprimento das regras oficiais de navegação é essencial para garantir a segurança. Assim, sistemas de suporte de tomadas de decisão qualitativos, que possam seguir essas regras, são interessantes por serem capazes de absorver conceitos de cognição humana, já que as regras são estabelecidas em linguagem natural.

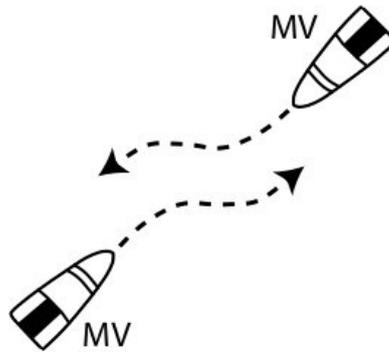
A formalização espacial das regras de navegação precisavam de uma técnica que capturasse informações direcionais, pois esse tipo de informação é de extrema importância na navegação, uma vez que a direção é necessária em diversas ocasiões, inclusive para diferenciar constelações espaciais. Nesse contexto, Dylla et al. (2007) escolheram utilizar a $OPRA_m$, devido ao fato dos barcos terem frente intrínseca e se moverem em uma determinada direção.

A $OPRA_m$ com granularidade **4** foi escolhida por ser suficientemente expressiva para traduzir as regras da navegação para a formalização qualitativa, sem comprometer o tempo de processamento nas tomadas de decisão (DYLLA et al., 2007).

Ainda segundo Dylla et al. (2007), as regras de prevenção de colisões na navegação marítima definem, para um par de embarcações, qual delas deve desviar e qual será privilegiada, podendo seguir o seu caminho. Dessa maneira, o trabalho considerou como foco os casos em que as embarcações podem se ver mutuamente.

Um exemplo é a regra 14 (DYLLA, 2009), que diz: “Quando duas embarcações a motor – *motor vessels* (MV) estão se aproximando uma da outra em cursos recíprocos ou quase

Figura 15 – Duas embarcações a motor (MVs)



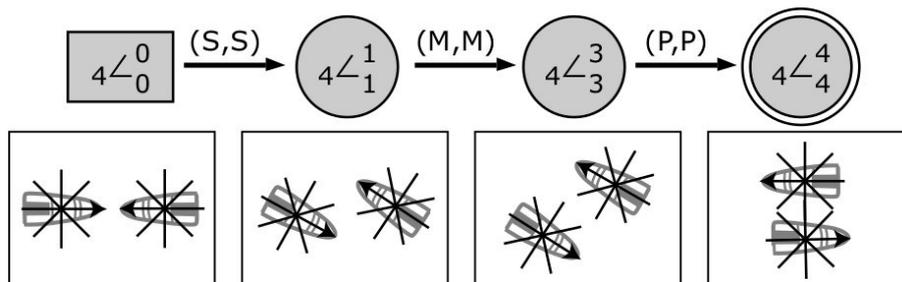
Fonte: DYLLA et al., 2007
 Legenda: ambas embarcações devem alterar os seus cursos para estibordo para passar a bombordo da outra

recíprocos (isto é, de frente ou quase de frente), de maneira que envolva algum risco de colisão, cada uma deve alterar o seu rumo para estibordo (direita), de modo que cada uma deve passar a bombordo (esquerda) da outra embarcação”. A representação gráfica dessa regra pode ser vista na Figura 15.

Para o raciocínio foi considerado que as embarcações podem executar três ações: virar para estibordo – *starboard* (S), virar para bombordo – *port* (P) e manter o curso – *midships* (M). A representação espacial foi realizada conforme exibido na Figura 16.

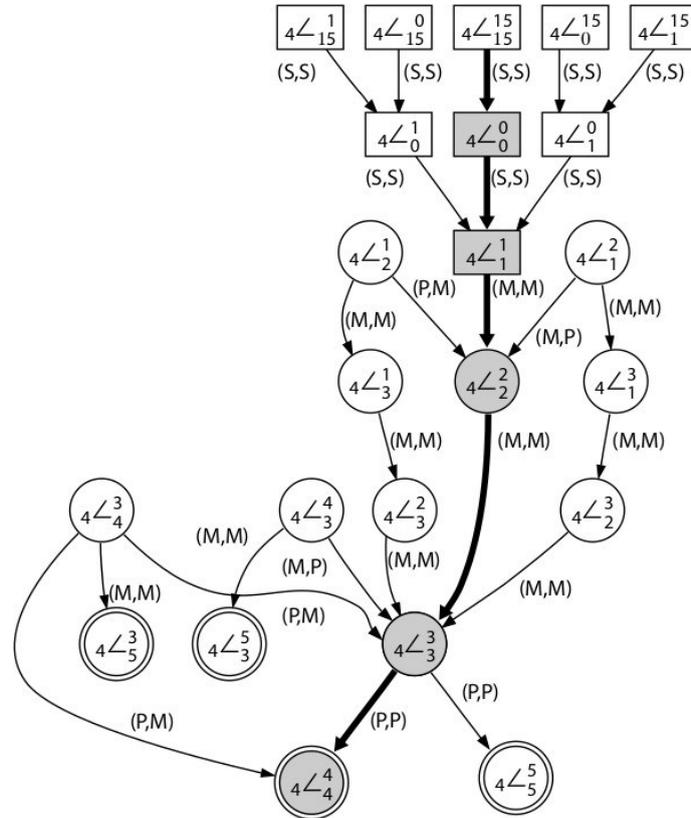
Uma vez que as ações e a representação espacial foram estabelecidas, o raciocínio foi criado por meio de um sistema de transição das regras aplicáveis em cada caso, com a intenção de formar uma rede de restrições em que a consistência corresponde a isenção de colisões. A Figura 17 exhibe o sistema de transição completo para a regra da Figura 16.

Figura 16 – Representação da $OPRA_4$ nas embarcações e o comportamento idealizado para a regra exibida na Figura 15



Fonte: DYLLA et al., 2007

Figura 17 – Sistema de transição completo para a regra exibida na Figura 16



Fonte: DYLLA et al., 2007

Outros trabalhos foram realizados depois do trabalho de Dylla et al. (2007), também com a utilização da $OPRA_m$ para formalização de regras de navegação marítima. Um exemplo é o trabalho de Dylla (2009) que segue a mesma ideia do trabalho apresentado nesta seção, porém com a utilização de uma $OPRA_m$ modificada por um parâmetro de refinamento, que foi chamada de $OPRA^v_m$ que não será abordada neste trabalho, uma vez que esse refinamento pode não ser necessário se a granularidade for aumentada. Outro exemplo, é o trabalho de Kreuzmann et al. (2013) que utiliza a $OPRA_m$ para fazer a representação espacial das embarcações, criando, então, uma formalização das regras marítimas por meio de uma lógica espaço-temporal que utiliza linguagem de fácil entendimento.

3.2 LOCALIZAÇÃO DE AGENTES AUTÔNOMOS COM ESTRATÉGIAS HÍBRIDAS: RACIOCÍNIO ESPACIAL QUALITATIVO PROBABILÍSTICO

Nesta seção, três técnicas híbridas de localização de agentes autônomos, que combinam métodos probabilísticos com raciocínios ou representações qualitativas, são exibidas.

3.2.1 Filtro de Bayes Aplicado às Relações Qualitativas Percebíveis sobre Sombras para Autocalibração e Autocalibração de Visão em Robôs Móveis

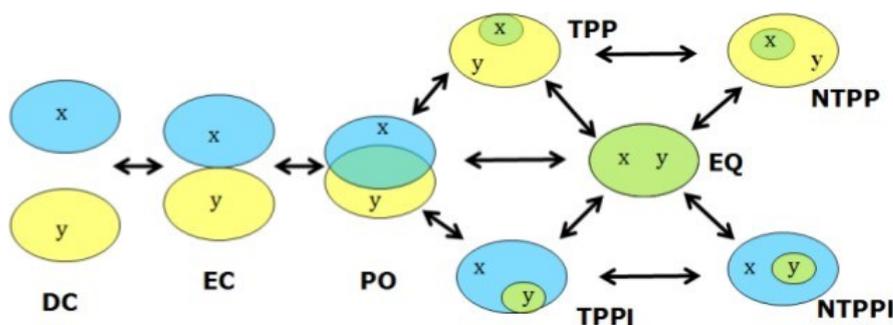
Pereira et al. (2013) propuseram um método para resolver o problema de autocalibração em robôs móveis que combina um modelo qualitativo com uma técnica probabilística. O modelo qualitativo utilizado foi o Relações Qualitativas Percebíveis sobre Sombras – *Perceptual Qualitative Relations about Shadows* (PQRS) (SANTOS, P. E.; DEE; FENELON, 2009) e a técnica probabilística foi o filtro de Bayes.

Esse método criou uma abordagem qualitativa-probabilística para localização de robôs, na qual um robô pode se localizar por meio de observações de objetos e suas sombras e que usa o filtro Bayesiano como uma maneira de lidar com incerteza (PEREIRA et al., 2013).

O PQRS permite que um agente raciocine sobre as relações entre sombras e objetos, utilizando, para isso, relações provenientes do Cálculo de Oclusão de Regiões – *Region Occlusion Calculus* (ROC) (RANDELL; WITKOWSKI; SHANAHAN, 2001). Como o ROC foi inspirado pelo Cálculo de Conexão de Regiões – *Region Connection Calculus* (RCC) (RANDELL; CUI; COHN, 1992), o conjunto de relações formalizada pelo PQRS também inclui as relações do RCC.

Resumidamente, o RCC é uma teoria sobre conexão entre regiões baseada em lógica de primeira ordem cujo objetivo principal é descrever uma relação primitiva binária de conectividade entre duas regiões. O RCC fundamental é composto por oito relações básicas principais (PEREIRA, 2014). Este conjunto é conhecido como RCC8 (RANDELL; CUI; COHN, 1992). O PQRS utiliza alguns conceitos centrais do cálculo RCC8, como o fato das transações contínuas entre as relações serem representadas pelo Diagrama Conceitual de Vizinhanças, conforme demonstrado pela Figura 18, em que x e y são regiões espaciais e DC , EC , PO , TPP , EQ , e $NTPP$ significam, respectivamente, *desconectado*, *externamente conectado*, *sobreposição parcialmente*, *parte própria tangencial*, *idêntico* e *parte própria não tangencial*.

Figura 18 – Diagrama Conceitual de Vizinhanças do RCC8



Por sua vez, o ROC é a formalização utilizada para se representar relações de oclusão entre duas entidades espaciais e é considerado um extensão do RCC8. O ROC fundamental é composto pelo conjunto de vinte relações básicas, conforme demonstrado na Figura 19.

Assim, o PQRS é composto pelo conjunto de sete relações básicas, conforme exibido na Figura 20, em que o representa um objeto e s representa sua sombra. Por meio do PQRS um robô pode inferir a sua posição qualitativa com relação à um objeto e com relação à sombra desse objeto, pois as relações expressas pelo modelo do PQRS dividem o espaço em cinco regiões distintas, conforme mapa qualitativo exibido na Figura 21. Em cada região do mapa o robô percebe uma relação diferente entre objeto-sombra (PEREIRA et al., 2013).

Dessa forma, um ponto de vista v localizado na *Região* o_1 , por exemplo, observa o topo da sombra $top(s)$ e o objeto o como $NonOccludes(o, top(s), v)$ (PEREIRA et al., 2013).

Como o PQRS é baseado no ROC e no RCC, que são construídos com lógica de primeira-ordem, o método não pode lidar com incertezas. Por outro lado, sabe-se que o sistema de visão em robôs é bastante ruidoso. Então, o trabalho de Pereira et al. (2013) propôs a combinação do PQRS com um filtro de Bayes, na qual a notação $bel(s_t) = P(s_t | e_{0:t})$ foi usada para descrever

Figura 19 – Cálculo de Oclusão de Regiões – *Region Occlusion Calculus* (ROC)

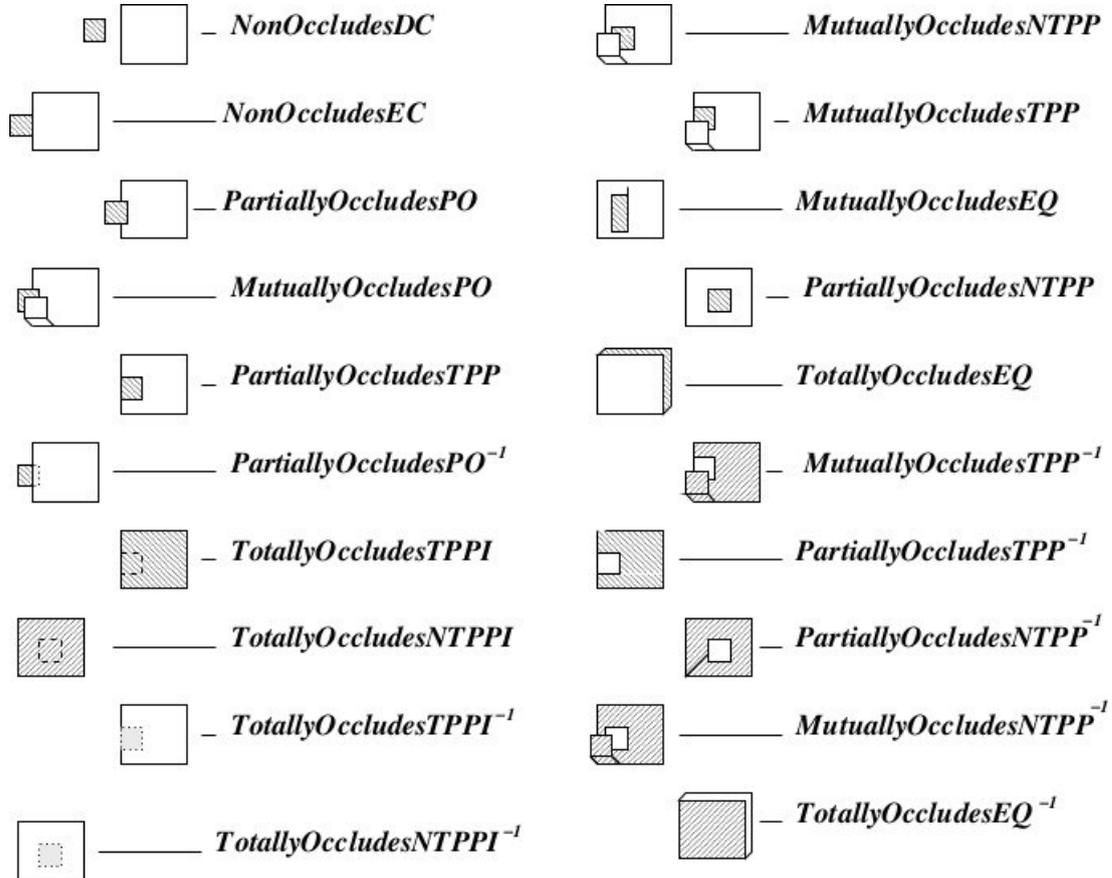
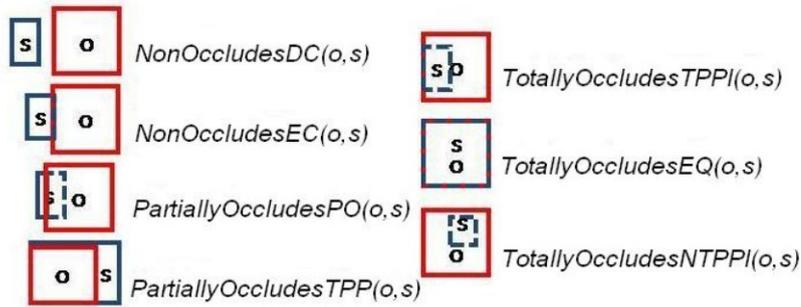


Figura 20 – Representação das relações básicas do PQRS

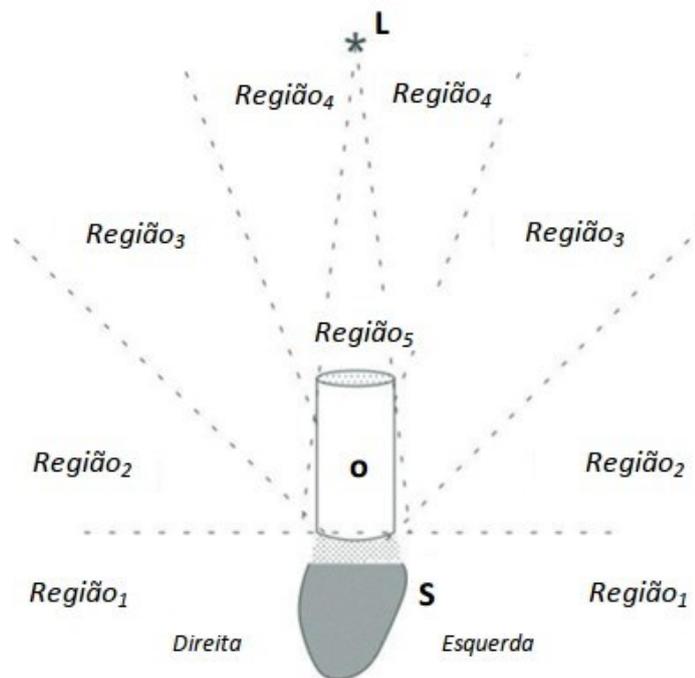


Fonte: PEREIRA et al., 2013

a crença do robô no estado s_t , dadas todas as evidências $(e_{0:t})$ até o instante t . O estado s_t representa uma região do mapa qualitativo da Figura 21 e as evidências são os termos denotados pelo PQRS.

Assim, a crença foi dada pela Equação (16), onde Th representa o limiar de intensidade de luz utilizado para extrair as sombras das imagens, uma vez que Th também era encontrado

Figura 21 – Representação do mapa qualitativo com cinco regiões distintas. O ponto fixo de luz é representado por L . As linhas entre L , o objeto e a sua sombra definem fronteiras entre as regiões



Fonte: Autor "adaptado de" PEREIRA et al., 2013

Tabela 2 – Resultados da autocalibração, utilizando também a autocalibração de Th

Regiões	Região ₁	Região ₂	Região ₃	Região ₄
Região ₁	97%	2%	0%	0%
Região ₂	25%	74%	1%	0%
Região ₃	9%	21%	66%	4%
Região ₄	6%	0%	8%	86%
Região ₅	0%	0%	35%	65%

Fonte: Autor “adaptado de” PEREIRA et al., 2013

automaticamente de forma a maximizar a crença, e \overline{bel} representa a predição, sendo dada conforme Equação (17) (PEREIRA et al., 2013).

$$bel(s_t) = \eta P(e, Th | s_t) \overline{bel}(s_t), \quad (16)$$

$$\overline{bel} = \sum_{s_{t-1}} P(s_t | s_{t-1}) \sum_{Th} P(s_{t-1}, Th | e_{0:t-1}). \quad (17)$$

Experimentos foram realizados com o método apresentado nesta seção utilizando o robô PeopleBot da Omron Adept MobileRobots, anteriormente conhecida como ActiveMedia. Para os experimentos, o robô coletou imagens em um ambiente de escritório, devendo se localizar com relação a um objeto específico e sua sombra. O objeto escolhido como referência foi um balde de lixo preto. A Tabela 2 exibe o resultado da localização em uma matriz de confusão, em que colunas representam a crença do sistema proposto em cada região, enquanto as linhas representam a posição real do robô.

Tabela 3 – Porcentagem de acertos do método puramente qualitativo (PQRS em comparação com o método qualitativo-probabilístico (PQRS + filtro de Bayes))

Regiões	# imagens	PQRS (%)	PQRS + filtro de Bayes (%)
Região ₁	225	66	97
Região ₂	171	34	74
Região ₃	138	80	66
Região ₄	36	44	86
Região ₅	17	59	0
Global	587	58	80

Fonte: Autor “adaptado de” PEREIRA et al., 2013

Uma das maiores contribuições do trabalho de Pereira et al. (2013) foi a comparação do método qualitativo-probabilístico com um método simplesmente qualitativo. Em outras pa-

lavras, o trabalho apresentado nessa seção comparou a capacidade de localização de um robô móvel utilizando o PQRS combinado ao filtro Bayesiano com o PQRS puro. O resultado pode ser visto na Tabela 3, na qual “# imagens” representa o número total de imagens de cada região adquiridas pelo robô.

3.2.2 Localização Probabilística em um Mapa Qualitativo Baseado em Oclusões

O algoritmo qualitativo-probabilístico para localização de robôs, proposto por P. E. Santos et al. (2016), se baseia em um mapa topológico construído a partir de observações sobre oclusão espacial por meio do formalismo encontrado no ROC. Além do ROC, posições relativas, como esquerda (*Left*), direita (*Right*), mais perto (*Closer*) e mais distante (*Further*), entre pares de objetos também foram utilizadas para a construção das regiões qualitativas do mapa.

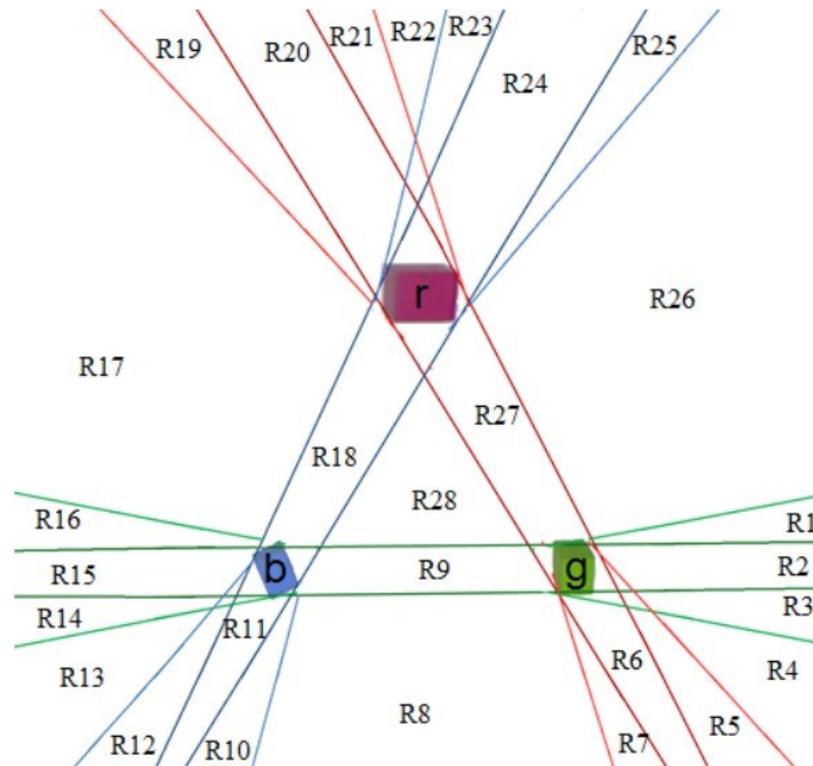
Assim, o mapa da Figura 22, composto por três objetos que são representados pelas letras r , b e g , foi construído dividindo-se o espaço por meio da combinação das relações do ROC com as posições relativas entre pares de objetos. Formalmente, a região $R1$ do mapa, por exemplo, é definida conforme fórmula (18), onde v representa o ponto de vista considerado para gerar as oclusões e as posições relativas entre os objetos. Todas as outras regiões têm também suas definições feitas por fórmulas que seguem a mesma lógica.

$$\begin{aligned}
 \text{localizado}(R1, v, b, g, r) \leftarrow & (P \text{ artiallyOccludes } T P P I(b, g, v) \wedge \\
 & \text{Right}(b, g, v) \wedge \text{Further}(b, g, v)) \vee \\
 & (\text{NonOccludesDC}(b, r, v) \wedge \text{Left}(b, r, v)) \vee \\
 & (\text{NonOccludesDC}(g, r, v) \wedge \text{Left}(g, r, v)) \wedge \\
 & \text{Closer}(g, r, v)).
 \end{aligned} \tag{18}$$

Depois que o mapa é construído, o procedimento para localização é realizado utilizando-se um Filtro Bayesiano, onde as evidências são obtidas em termos das relações de oclusão entre pares de objetos do instante 0 até t . Dessa forma, está evidência é fornecida pelas relações provenientes do ROC, conforme observadas do ponto de vista v , seguindo o exemplo da fórmula (18). O objetivo do procedimento de localização é descobrir em que região qualitativa do mapa o robô se encontra. A região com a maior crença (*bel*), é considerada a região na qual o robô está.

Assim como na seção anterior (3.2.1), experimentos com o robô PeopleBot foram realizados conforme Figura 23, utilizando o mapa exibido na Figura 22. Os resultados obtidos demonstraram que o algoritmo qualitativo-probabilístico proposto obteve melhores resultados do que a tentativa de tentar estimar a região do robô de forma puramente qualitativa, comparando-se as observações realizadas pelo robô diretamente com as fórmulas que definem cada região, sem considerar as informações que foram obtidas ao longo do tempo.

Figura 22 – Representação do mapa qualitativo composto pela divisão do espaço ao redor dos objetos r , b e g : **28** regiões distintas foram formadas.



Fonte: SANTOS, P. E. et al., 2016.

Figura 23 – Imagem do experimento realizado com o robô PeopleBot.



Fonte: SANTOS, P. E. et al., 2016.

A principal contribuição deste trabalho foi a avaliação empírica de uma teoria do QSR baseada em ponto de vista, que foi implementada dentro de um algoritmo de localização probabilística em um robô real.

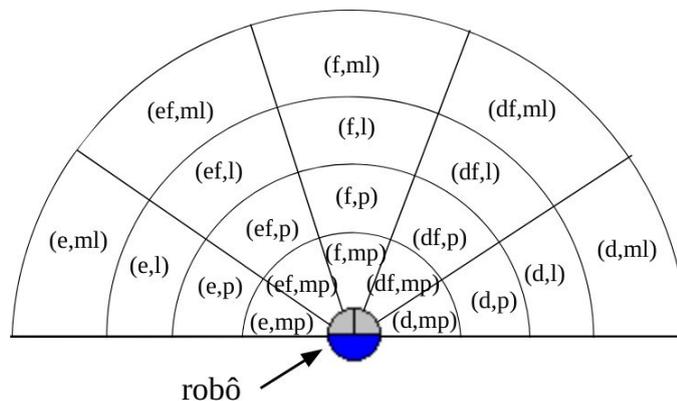
3.2.3 Localização de Monte Carlo Baseada em Visão com Abordagem Qualitativa Durante a Fase de Atualização

No decorrer deste doutorado, um estudo sobre o uso de dados qualitativos na fase de atualização da Localização de Monte Carlo foi realizado (PERICO; SANTOS, P. E.; BIANCHI, 2015). A combinação apresentada no artigo entre MCL e dados qualitativos foi chamada de Localização Híbrida e teve como principal motivação a obtenção de um nível de abstração mais próximo da categorização humana do espaço e, também, de uma maneira mais eficaz de interação entre robôs e humanos.

A proposta apresentada utilizou o conceito qualitativo de uma Egosfera (SANTOS, Paulo E.; SANTOS, D. N., 2013), que é baseada no TPCC, pelo qual o robô podia perceber o mundo por meio de relações qualitativas de direção e distância, conforme demonstrado na Figura 24, onde as seguintes abreviações *e*, *ef*, *f*, *df* e *d* representam, respectivamente, as relações de direção *esquerda*, *esquerda-frente*, *frente*, *direita-frente* e *direita* e as abreviações *ml*, *l*, *p* e *mp* representam, respectivamente, as relações de distância *muito-longe*, *longe*, *perto* e *muito-perto*.

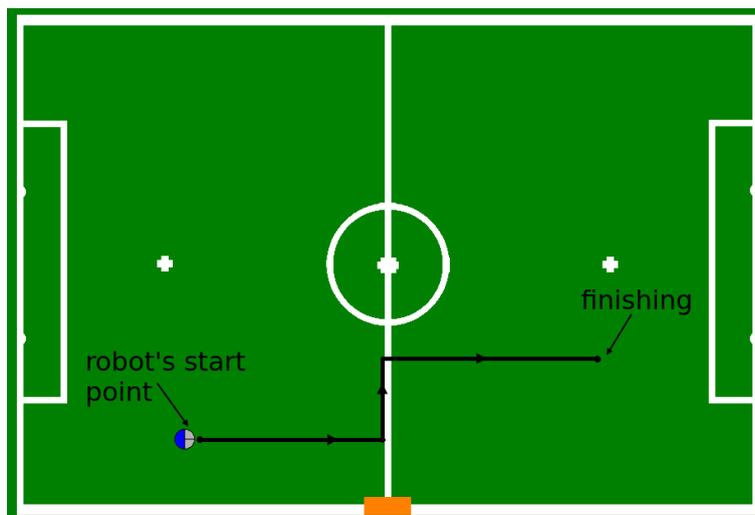
Os experimentos foram realizados em ambiente simulado com base nos desafios encontrados na liga humanoide da RoboCup, conforme Figura 25, e consistiram em comparar a robustez do abordagem proposta para um modelo MCL baseado em visão tradicional. Para avaliar o processo de localização global, o robô girava **360°** em torno de si, o que permitia que ele percebesse todos os pontos de referência do domínio. Uma vez que o robô estava globalmente localizado, um caminho foi realizado com o robô para avaliar a capacidade de rastreamento.

Figura 24 – Vista superior do corte no equador da Egosfera do robô.



Fonte: Autor "adaptado de" PERICO; SANTOS, P. E.; BIANCHI, 2015.

Figura 25 – Simulador utilizado



Fonte: PERICO; SANTOS, P. E.; BIANCHI, 2015

Legenda: o pequeno círculo no lado esquerdo representa o robô; a linha preta representa o caminho realizado pelo robô durante o experimento realizado para avaliar a capacidade de rastreamento

A maior contribuição deste trabalho foi a conclusão de que o uso de dados qualitativos podem mostrar desempenho similar quando comparado ao uso de dados numéricos na MCL baseada em visão, sendo que, dados qualitativos trazem a vantagem de estar mais próximo do modo como os seres humanos referem-se ao espaço, o que pode melhorar a comunicação entre robôs, assim como o desenvolvimento de estratégias de alto nível durante um jogo.

3.3 NAVEGAÇÃO GUIADA DE UM AGENTE AUTÔNOMO

Mueggler et al. (2014) apresentaram o trabalho em que um robô aéreo trabalha em colaboração com um robô terrestre, o guiando até um objetivo de forma mais rápida possível, em uma maquete que representa um cenário de desastre (Figura 26). Segundo os autores do trabalho, a motivação para tal desenvolvimento está no fato de que robôs têm sido constantemente usados em várias situações de desastre, contudo, em todas elas, os robôs têm sido controlados remotamente por profissionais treinados. Um dos problemas de se usar pessoas para controlar os robôs é que, normalmente, o operador precisa enviar comandos de muito baixo nível ao robô, o que torna o resgate lento, além de criar a necessidade de, em média, existirem três operadores por robô.

O modelo de resgate apresentado segue a seguinte estrutura: o robô aéreo mapeia o ambiente e procura pela vítima; então, o planejamento da missão é realizado com o objetivo de conduzir o robô terrestre até a vítima; por fim, o robô aéreo guia o robô terrestre pelo caminho computado até a vítima (MUEGGLER et al., 2014). No artigo, o robô aéreo é um quadrotor equipado com câmera monocromática, uma Unidade de Medida Inercial – *Inertial Measurement Unit* (IMU), um controlador para baixo nível e um computador Odroid-U3, utilizado,

Figura 26 – O robô aéreo e o robô terrestre na maquete de desastre.



Fonte: MUEGGLER et al., 2014.

dentre outras coisas, para fazer o processamento da imagem; já o robô terrestre é um youBot (BISCHOFF; HUGGENBERGER; PRASSLER, 2011) da KUKA, que é composto por uma base móvel e um manipulador com 5 graus de liberdade que é utilizado para remover os obstáculos móveis do caminho. Para controlar tanto a base, quanto o manipulador, um computador com processador Intel® Atom foi embarcado no robô, além disso, um laser da Hokuyo foi instalado na frente do robô para permitir que ele identificasse precisamente os obstáculos.

No artigo, o robô aéreo identifica os obstáculos, o objetivo, a origem e o robô terrestre por meio de etiquetas únicas. Para realizar o planejamento da trajetória, os algoritmos A^* e Dijkstra foram utilizados, na qual diversas possíveis missões são geradas para testar os caminhos considerando, ou não, a necessidade de remover os obstáculos móveis. Então, o robô aéreo guia o robô terrestre por meio de ações de alto nível, como “se movimente em linha reta” ou “remova um obstáculo” e o robô terrestre executa a ação até que ela seja concluída, sem nenhum retorno do robô aéreo; para realizar a navegação, o robô aéreo localiza o robô terrestre no mapa e envia para ele a instrução; o robô terrestre, então, executa o movimento utilizando dados da odometria para saber até onde se movimentar (MUEGGLER et al., 2014).

Segundo Mueggler et al. (2014), os experimentos realizados tiveram **81%** de sucesso, sendo realizados **27** vezes com início aleatório dos objetos fixos e móveis. Ainda segundo os autores do artigo, um dos maiores problemas encontrados durante o experimento foi com a comunicação sem fio entre os robôs.

3.4 COMENTÁRIOS

Conforme pôde ser notado no decorrer de toda a seção 3, o uso dos modelos puramente qualitativos normalmente tratam de problemas de navegação mais simplificados, pois

a incerteza pode acarretar diversos tipos de erros em aplicações reais. Contudo, as dificuldades provenientes das incertezas podem ser superadas, normalmente, por meio das abordagens híbridas, que tendem a manter o alto nível de abstração espacial e capacidade de interação humano-computador/robô proporcionados pelos modelos qualitativos, ao mesmo tempo que as técnicas probabilísticas tratam as incertezas existentes no domínio.

Outro ponto interessante, que pode ser observado nesta seção, é a pouca quantidade de trabalhos relacionados à navegação guiada. O único trabalho apresentado, apesar de apresentar uma comunicação com instruções de alto nível, funciona somente com 2 robôs e, necessariamente, eles precisam ser um aéreo e um terrestre. Além disso, os robôs eram dotados de diversos tipos de sensores e o ambiente era bastante controlado.

Dessa forma, a próxima seção apresenta a proposta deste trabalho, que é a navegação de um único agente, que não percebe o mundo, utilizando instruções espaciais linguísticas baseadas em relações projetivas, que são obtidas por meio da junção da perspectiva de múltiplos agentes não-humanos, que, inclusive, podem ser heterogêneos.

4 NAVEGAÇÃO GUIADA COM RELAÇÕES PROJETIVAS OBTIDAS POR MÚLTIPLAS PERSPECTIVAS ESPACIAIS

Nessa seção o problema abordado será definido e explicado e duas propostas de solução serão apresentadas: QPFGN_m e NPFNGNQI_m^r. Mais especificamente, a seção 4.1 definirá o problema em si, estabelecendo quais são as restrições impostas. A seção 4.2 abordará os fundamentos comuns às duas propostas, como o fato de que ambas consideram que o agente guiado receberá instruções de direções baseadas em preposições espaciais projetivas e de que ambas realizam inferências utilizando o filtro de partículas; a seção 4.3 apresentará o Filtro de Partículas Qualitativo para Navegação Guiada – *Qualitative Particle Filter for Guided Navigation* (QPFGN_m); e a seção 4.4 apresentará o Filtro de Partículas Numérico para Navegação Guiada com Instruções Qualitativas – *Numerical Particle Filter for Guided Navigation with Qualitative Instructions* (NPFNGNQI_m^r).

4.1 DEFINIÇÃO DO PROBLEMA

O problema que será abordado neste trabalho é o da navegação guiada, no qual um agente deverá ser guiado por um conjunto de outros agentes autônomos não-humanos, dotados de visão computacional, até um objetivo. O agente que será guiado pode ser tanto um humano, quanto um robô (terrestre ou aéreo). Este agente é desprovido de sensores que o permitam perceber o ambiente; contudo, a sua orientação pode, ou não, ser conhecida. Como o agente a ser guiado não tem uma especificação bem definida, assume-se que ele não terá dados provenientes de odometria, tão pouco de um modelo de velocidades.

Dessa forma, devido ao grau de incertezas nas características do agente a ser guiado, ele deve receber instruções mais abstratas e de alto nível, que possam ser compreensíveis por qualquer tipo de agente. Além disso, considerando que todos os agentes envolvidos podem ter qualquer tamanho e formato, trabalhar com estimativas de distâncias, obtidas por meio de visão computacional, pode ser um processo bastante inconveniente e problemático, independentemente se o sistema de visão for monocular ou estéreo.

Assim, as principais restrições do problema são dadas conforme segue:

- a) O agente guiado deve receber instruções abstratas de alto nível, que independam do agente em si;
- b) O modelo de movimento de translação do agente guiado é desconhecido; o de rotação deve ser simples o bastante para acomodar instruções abstratas de alto nível;
- c) O agente guiado não tem modelo de sensor, uma vez que não tem sensores para perceber o mundo;
- d) A orientação do agente guiado pode ou não ser conhecida;

- e) Devido ao caráter do problema, utilizar estimativas de distâncias obtidas por meio de visão computacional pode ser um processo bastante problemático.

4.2 BASES COMUNS ÀS PROPOSTAS

Como o agente guiado não tem modelo de sensor, guiá-lo utilizando navegação aloccêntrica seria bastante complicado, já que ele não percebe o domínio no qual está inserido. Dessa forma, utilizar um modelo de navegação que é centrado no agente parece ser o mais adequado, uma vez que facilita a comunicação das instruções. Assim, o modelo de navegação adotado neste trabalho será o egocêntrico com relação ao agente guiado.

Para superar o problema de estimativa de distâncias, todo o processo de navegação guiada, incluindo todas as inferências, será realizado somente com o uso de relações de direção e, como o intuito deste trabalho é o de permitir que as instruções dadas sejam compreensíveis por qualquer tipo de agente, seja humano ou robô, a navegação será realizada por meio de instruções de direção relativas baseadas em quatro preposições espaciais linguísticas projetivas: direita, esquerda, frente e atrás.

Com o intuito de superar as restrições impostas na seção 4.1, algumas propostas foram estabelecidas, conforme segue:

- a) O agente guiado deve receber instruções espaciais linguísticas baseadas em relações projetivas (seção 4.2.2.1);
- b) O modelo de movimento do agente guiado será genérico e parcial (seção 4.2.2.2);
- c) Uma vez que o agente guiado não tem modelo de sensor, os outros agentes pertencentes ao problema é que realizarão a percepção do domínio com o intuito de realizar uma navegação egocêntrica com relação ao agente guiado;
- d) Caso o agente guiado não tenha orientação conhecida, esta deve ser inferida durante a navegação (será demonstrada em cada uma das propostas – seção 4.3 e 4.4);
- e) Nenhuma informação de distância será diretamente extraída do domínio, seja ela numérica ou qualitativa.

Um ponto comum às duas propostas – $QPFGN_m$ e $NPFGNQI_m^r$ é a existência da granularidade m . Em ambos os modelos, a granularidade m representa a quantidade de setores angulares que formarão as direções qualitativas de cada agente envolvido na navegação, exatamente conforme exibido na seção 2.2.1.5. O que muda de uma proposta para outra é que no $QPFGN_m$ o modelo é todo qualitativo e todas as entidades espaciais são mapeadas com base nas relações qualitativas de direção. Já no $NPFGNQI_m^r$ as relações de direção qualitativas somente são utilizadas no agente guiado e na determinação do objetivo. A granularidade m não altera a quantidade de relações projetivas, que, durante este trabalho, serão sempre quatro. Ela altera a discretização do mundo para cada agente.

Figura 27 – Agentes autônomos utilizados durante a proposta.



Fonte: Autor

4.2.1 Hipótese Central e Considerações

A hipótese central deste trabalho é que existem modelos híbridos, qualitativo-probabilísticos, capazes de guiar um agente autônomo sem percepção do mundo com instruções de direção relativas baseadas em preposições espaciais linguísticas projetivas, que são inferidas por meio das orientações e das perspectivas recebidas de outros agentes autônomos dotados de visão computacional. Dada esta hipótese, algumas considerações foram assumidas durante o desenvolvimento de ambas as propostas que serão apresentadas nas próximas seções deste trabalho. São elas:

- a) Com exceção do agente guiado, todos os outros precisam ter informação completa do domínio;
- b) O modelo de navegação apresentado neste trabalho é independente do ambiente;
- c) Obstáculos não são considerados;
- d) O agente autônomo que será guiado pelos outros agentes autônomos não-humanos será chamado de agente guiado (Figura 27);
- e) Os agentes autônomos não-humanos responsáveis por fornecer as diferentes perspectivas espaciais que serão utilizadas para as tomadas de decisão e instrução do agente guiado serão chamados de agentes observadores (Figura 27);
- f) Os agentes observadores não se movem enquanto guiam o agente guiado.
- g) O objetivo a ser alcançado pelo agente guiado tem que ser um marco, ou ponto de referência visual, que pode ser identificado pelos agentes observadores.

4.2.2 Definições do Agente Guiado

Duas definições do agente guiado são comuns às duas propostas: a criação de regiões espaciais linguísticas baseadas em relações projetivas, que será realizada com fundamentação no StarVars_m e o modelo de movimento genérico e, portanto, com poucas restrições.

4.2.2.1 Regiões Espaciais Linguísticas Baseadas em Relações Projetivas

As direções qualitativas relativas do agente guiado são definidas com base no StarVars_m . Como a intenção é que o agente autônomo seja guiado por meio das preposições espaciais *direita*, *esquerda*, *frente* e *atrás*, as seções do StarVars_m , que são representadas por regiões numéricas, serão unidas de forma a representar estas preposições projetivas.

Para se realizar a união das seções, definiu-se que a granularidade m do StarVars_m deve ser um número múltiplo de **8**, assim $m \in \mathbf{N}$, $m \pmod{8} = 0$ e $m \geq 8$, pois a granularidade mínima **8** permite que todas as regiões formadas pelos quatro predicados projetivos tenham exatamente o mesmo tamanho angular.

A regra para se formar as regiões projetivas por meio da união das seções angulares do StarVars_m é dada conforme Equações de (19) a (22), utilizando a notação demonstrada na seção 2.2.1.5.

$$\text{esquerda} = \left[\frac{m}{8}, \frac{3m}{8} \right], \quad (19)$$

$$\text{atrás} = \left[\frac{3m}{8}, \frac{5m}{8} \right], \quad (20)$$

$$\text{direita} = \left[\frac{5m}{8}, \frac{7m}{8} \right], \quad (21)$$

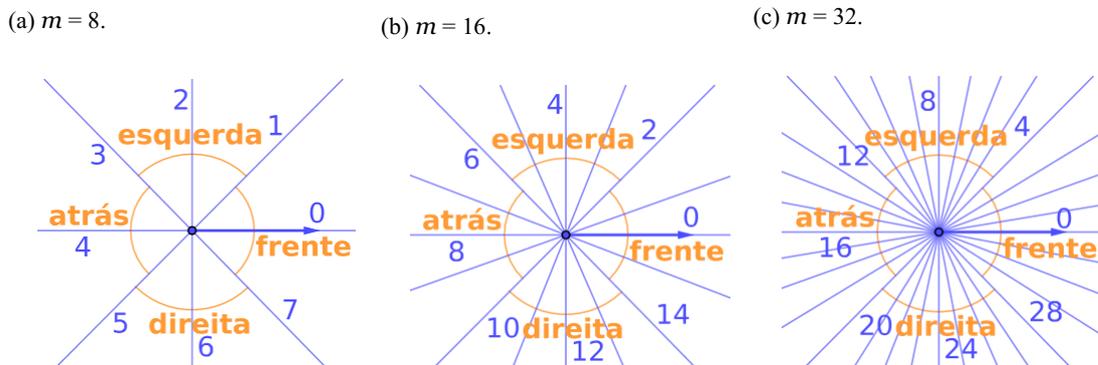
$$\text{frente} = \left[\frac{7m}{8}, \frac{m}{8} \right]. \quad (22)$$

A Figura 28 exhibe exemplos da união das seções angulares do StarVars_m com três granularidades diferentes: $m = 8, 16$ e 32 para gerar as relações projetivas. É possível notar, pela análise da Figura, que o tamanho angular das regiões projetivas não depende da granularidade m , desde que m seja um múltiplo de **8**.

4.2.2.2 Modelo de Movimento Genérico e Parcial

Para poder acomodar as ações do agente guiado de maneira genérica, o modelo de movimento proposto é bastante simples e contém poucas restrições. Nesse modelo, o agente pode agir de cinco maneiras diferentes: “ficar parado”, “movimentar-se para frente”, “virar para direita e movimentar-se para frente”, “virar para esquerda e movimentar-se para frente”, “virar para trás e movimentar-se para frente”.

Figura 28 – União das seções angulares da $StarVars_m$ para formar as regiões nomeadas com preposições projetivas.



Fonte: Autor

Para cada uma das cinco ações, o agente guiado executará o seguinte:

- Parar: não executa nenhum movimento;
- Movimentar-se para frente: o agente se movimentará em linha reta com relação a sua orientação, com qualquer velocidade;
- Virar para direita e movimentar-se para frente: o agente deverá girar $-90^\circ + \delta_t$ e repetir ação movimentar-se para frente;
- Virar para esquerda e movimentar-se para frente: o agente deverá girar $+90^\circ + \delta_t$ e repetir ação movimentar-se para frente;
- Virar para trás e movimentar-se para frente: o agente deverá girar $180^\circ + \delta_t$ e repetir ação movimentar-se para frente;

A ação movimentar-se para frente só termina quando o agente guiado receber a instrução para ficar parado. A variável δ_t representa um ruído Gaussiano com média igual a zero e variância σ , $\delta_t \sim \mathbf{N}(\mathbf{0}, \sigma)$.

O modelo será considerado parcial, pois a cinemática do agente é bastante simplificada, considerando apenas dados relacionados a rotação, sendo que nenhum dado relacionado a translação é obtido.

4.2.3 Definições dos Agentes Observadores

Com exceção do agente guiado, todos os outros agentes precisam perceber completamente o domínio por meio de visão computacional e têm suas orientações conhecidas. Ambas propostas (4.3 e 4.4) suportam n agentes observadores, onde $n \in \mathbf{N}$. Para a primeira proposta (4.3), $n \ll \mathbf{1}$, mas, no caso de $n = \mathbf{1}$, a acurácia da navegação não é muito boa, pelos motivos que serão melhor explicados na seção 4.3.2; já na segunda proposta (4.4), $n \ll \mathbf{2}$, pois existe a necessidade de se realizar triangulações.

Independentemente do número de agentes observadores, as instruções relacionadas aos movimentos, isto é, movimentar-se para frente, virar para direita, para esquerda e para trás, serão passadas ao agente guiado por somente um agente observador, que será definido como coordenador (Figura 27). Dessa forma, é o coordenador que realiza as inferências necessárias dadas as perspectivas e as orientações de todos os outros agentes observadores envolvidos. A instrução de parada é a única exceção, e pode ser dada por qualquer agente observador. O modelo que especifica a necessidade de parada do agente guiado é implementado de maneiras diferentes em cada uma das propostas que serão apresentadas nas seções 4.3 e 4.4. Qualquer agente pode ser classificado como coordenador, porém, uma vez assumida, essa função deve permanecer do início ao fim da navegação guiada.

4.2.4 Algoritmo Geral para Navegação Guiada

Esta seção apresenta o Algoritmo 5, que retrata a forma geral do procedimento da navegação guiada realizada por ambos os métodos propostos neste trabalho: QPFGN_m e NPFGNQI_{mr}^r. No Algoritmo 5, as entidades espaciais são representadas por e , os agentes observadores por o , o agente coordenador por c , o agente guiado por g e o conjunto, armazenado no agente coordenador c , composto pelas direções obtidas por cada agente observador com relação à cada entidade espacial, assim como a orientação de cada agente, por Z_c , onde $Z_c = \wedge_{i,j} \langle i, z_{ij}, j, \theta_i \rangle$. Z_c pode conter ou não a orientação do agente guiado, θ_g : isto vai depender da existência desta informação.

Todas as inferências da navegação guiada são realizadas pelo agente coordenador c , por isso, todas as funções executadas por ele recebem índice c de forma subscrita. Dentre estas funções, algumas recebem o índice g de maneira sobrescrita, pois, apesar de serem executadas pelo agente coordenador, são funções que influenciam diretamente o agente guiado, por exemplo, a função envia_c^g , na linha **16** é executada pelo agente coordenador c para enviar a instrução de movimento para o agente guiado g .

Da linha **4** até a linha **10** o procedimento para se obter as direções de cada entidade espacial com relação a cada agente observador é realizado. A função obtemDireção é a que realiza a extração dessa informação do domínio.

Na linha **15**, a função mapeamento_c é responsável pela criação da representação interna da localização de todas as entidades espaciais, ou seja, um mapa, incluindo todos os agentes e o objetivo. A função $\text{liberaPartículas}_c$, na linha **16**, gera, a partir do modelo de mundo ψ retornado pela função mapeamento_c , o conjunto S , $S = \{S_{o_{i=\{1, \dots, n_o\}}} \cup S_g \cup S_{\text{objetivo}}\}$, com todas as partículas para todas as entidades espaciais, sendo uma partícula para cada agente observador (conjunto $S_{o_{i=\{1, \dots, n_o\}}}$), várias para o agente guiado (conjunto S_g), onde a incerteza na postura é maior, e, dependendo do método, uma ou várias para o objetivo (conjunto S_{objetivo}). Assim como na MCL, cada partícula que representa um agente autônomo é definida por uma

Algoritmo 5 – Forma geral da Navegação Guiada.

```

1  Entrada:  $EE =$  Conjunto com todas as entidades espaciais  $\wedge j e_j$ , composto pelos
    agentes observadores, pelo agente guiado e pelo objetivo; número total
    de agentes observadores  $n_o$ ; e número total de entidades espaciais  $n$ ;
    granularidade  $m$ 

2  início
3   $Z_c = \emptyset$ 
4  para cada  $i = o \in \{1, \dots, n_o\}$  faça
5      para cada  $j = e \in \{1, \dots, n\}$  faça
6          se  $i = j$  então
7               $z_{ij} = \text{obtemDireção}()$ 
8               $i$  envia  $\langle i, z_{ij}, j, \theta_i \rangle$  para o agente coordenador  $c$ 
9               $Z_c = Z_c \cup \langle i, z_{ij}, j, \theta_i \rangle$ 
10             fim
11         fim
12     fim
13     se  $\exists \theta_g$  então
14          $g$  envia  $\langle g, \theta_g \rangle$  para o agente coordenador  $c$ 
15          $Z_c = Z_c \cup \langle g, \theta_g \rangle$ 
16     fim
17      $\psi = \text{mapeamento}_c(Z_c)$ 
18      $S = \text{liberaPartículas}_c(\psi)$ 
19     enquanto  $g$  não encontrar objetivo definido de acordo com  $m$  faça
20         ação =  $\text{escolheAçã}_c^s()$ 
21          $\overline{S}_g = \text{predição}_c(\text{açã}o, S_g)$ 
22         envia $_c^s(\text{açã}o)$ 
23         enquanto  $g$  não atingir a condição de parada faça
24             agente guiado  $g$  se movimenta
25             para cada  $i = r \in \{1, \dots, n_o\}$  faça
26                  $z_{ig} = \text{obtemDireção}()$ 
27             fim
28         fim
29         agente guiado  $g$  para de se movimentar
30         para cada  $i = o \in \{1, \dots, n_o\}$  faça
31              $i$  envia  $\langle i, z_{ig}, g, \theta_i \rangle$  para o agente coordenador  $c$ 
32             substitui  $\langle i, z_{ig}^j, g, \theta_i \rangle$  por  $\langle i, z_{ig}, g, \theta_i \rangle$  em  $Z_c$ 
33         fim
34         se  $\exists \theta_g$  então
35              $g$  envia  $\langle g, \theta_g \rangle$  para o agente coordenador  $c$ 
36             substitui  $\langle g, \theta_g^j \rangle$  por  $\langle g, \theta_g \rangle$  em  $Z_c$ 
37         fim
38          $S_g = \text{atualização}_c(\overline{S}_g, Z_c)$ 
39     fim
40 fim

```

postura e por um fator de importância: $((x, y, \theta), w)$. Por outro lado, cada partícula que representa o objetivo é definida somente por uma posição: (x, y) .

A função $escolheAção_c^g$ é encarregada de realizar o planejamento de movimento do agente guiado g , retornando uma *ação*, ou instrução, que utiliza relações projetivas. Essa mesma *ação* será usada pelo agente coordenador nas linhas **19** e **20** para realizar, respectivamente, a predição da postura, por meio da aplicação da *ação* às partículas, e o envio da instrução em si para o agente guiado.

Assim que o agente guiado começa a se movimentar, todos os agentes observadores mantêm atualização constante da direção relativa do agente guiado por meio da função $obtemDireção$ até que o critério de parada seja alcançado. Então, as novas direções z_{ig} obtidas pelos agentes observadores i com relação ao agente guiado g são enviadas ao agente coordenador, que substitui as direções antigas, z_{ig}^j , pelas novas dentro do conjunto Z_c (linhas **28** à **31**). Caso a orientação do agente guiado seja conhecida, esta informação também é atualizada (linhas **32** à **35**).

A atualização das partículas que representam a postura do agente guiado é, então, realizada na sequência. Na função $atualizacao_c$, o fator de importância w de cada partícula é atualizado de acordo com Z_c e a reamostragem é realizada (linha **36**).

O Algoritmo 5 contempla as três competências da navegação – localização, mapeamento e planejamento de movimento – conforme exibido na seção 2.1. A localização pode ser encontrada durante o mapeamento, mas principalmente, nas funções de $predicao_c$ e $atualizacao_c$. Já o mapeamento e o planejamento de movimento podem ser vistos diretamente nas funções $mapeamento_c$ e $escolheAcao_c^g$.

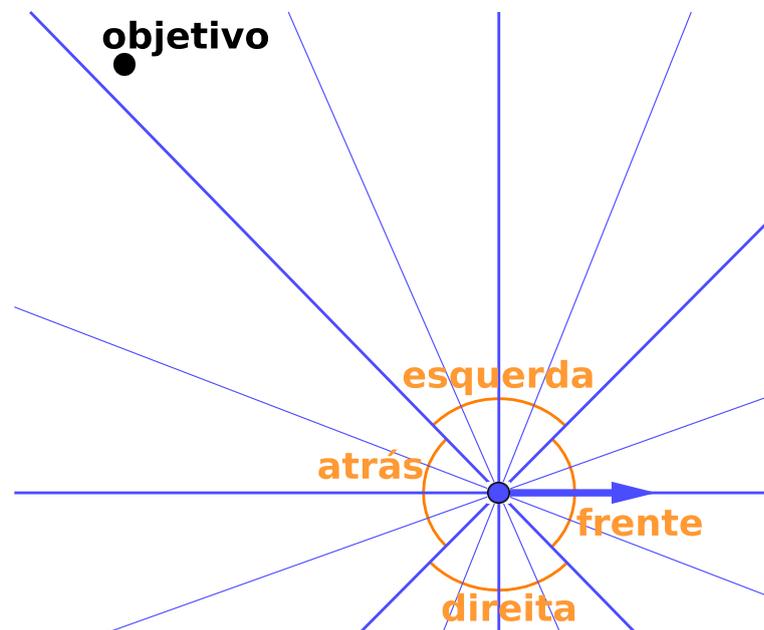
O que muda para cada um dos métodos propostos – $QPFGN_m$ e $NPFGNQI_m^r$ – é a natureza dos dados de entrada. No $QPFGN_m$ os dados são qualitativos, já no $NPFGNQI_m^r$ os dados são numéricos. Assim, as funções $obtemDireção$, $mapeamento_c$, $predicao_c$ e $atualizacao_c$ trabalham de maneiras diferentes em cada um dos métodos, pois são funções que dependem da natureza dos dados utilizados. As peculiaridades de cada modelo serão apresentadas nas seções 4.3 e 4.4, respectivamente.

Apesar das diferenças existentes entre os dois modelos propostos, ambos utilizam instruções qualitativas por meio de preposições projetivas para guiar o agente até o objetivo. Isso faz com que o procedimento para a seleção das ações do agente guiado, $escolheAcao_c^g$, seja igual para os dois métodos, conforme será exibido na próxima seção.

4.2.5 Seleção das Ações: Planejamento de Movimento

Conforme demonstrado na seção 2.1.1.4, o planejamento de movimento busca a escolha correta das ações para um agente móvel. Neste trabalho, tanto no $QPFGN_m$, quanto no $NPFGNQI_m^r$, as tomadas de decisão sobre as ações do agente guiado serão feitas com base nas partículas que serão utilizadas para representar a crença de postura desse mesmo agente. A inferência da ação que o agente guiado deve tomar é feita pelo agente coordenador, portanto esta ação é enviada para o agente guiado como uma instrução de movimento.

Figura 29 – Relacionamento entre o objetivo e uma partícula que representa uma possível postura do agente guiado



Fonte: Autor

Legenda: a ação selecionada neste caso é “virar para esquerda e se movimentar para frente”

Assim, a decisão realizada pelo agente coordenador, representada pela função $escolheAção_c$ no Algoritmo 5, sobre qual instrução dar ao agente guiado vai ser realizada comparando-se a posição relativa do objetivo com relação a cada partícula que representa uma postura do agente guiado. Lembrando que as partículas que representam o agente guiado utilizam as relações projetivas, conforme exibido na seção 4.2.2.1. A Figura 29 exibe a relação entre uma partícula e o objetivo, resultando na instrução “virar para esquerda e se movimentar para frente”.

Dessa forma, uma diferente ação ou instrução pode ser obtida para cada partícula, uma vez que ela tem posturas distintas. Como N partículas serão utilizadas, pode-se calcular a probabilidade de cada instrução. A instrução que será dada será aquela com a maior probabilidade.

Portanto, as tomadas de decisão sobre qual movimento o agente guiado deve fazer são realizadas conforme ele se movimenta pelo domínio. No fim de cada ciclo, do Filtro de Partículas, composto por predição e atualização, uma nova escolha de ação é realizada pelo agente coordenador.

4.3 FILTRO DE PARTÍCULAS QUALITATIVO PARA NAVEGAÇÃO GUIADA

Essa seção apresenta o Filtro de Partículas Qualitativo para Navegação Guiada – *Qualitative Particle Filter for Guided Navigation* (QPFGN $_m$), com granularidade m .

4.3.1 Mapeamento e Localização

A primeira etapa do QPFGN_m é a obtenção das direções relativas de cada entidade espacial para cada agente observador, como representado pela função `obtemDireção` no Algoritmo 5. Como estas direções relativas são qualitativas, todos os agentes observadores são modelados no domínio do StarVars_m, sendo $m \in \mathbf{N}$, $m \pmod{8} = 0$ e $m \neq 8$, assim como no agente guiado. A categorização espacial realizada pelo StarVars_m permite a obtenção de relações como $A(\mathbf{1})B$, o quer dizer que B está no setor $(\mathbf{1})$ de A .

Depois que todas as direções relativas qualitativas forem obtidas e enviadas ao agente coordenador, a construção de um mapa é realizada pela função `mapeamentoc`, que retorna um modelo de mundo $\psi = \bigwedge_{i \in \{1, \dots, n\}} (x_i, y_i, \theta_i)$, que contém x e y todas as entidades espaciais, além de θ dos agentes observadores. Esse mapa é elaborado pelo agente coordenador por meio das relações espaciais qualitativas e das orientações de todos os agentes observadores. O Algoritmo 6 demonstra o procedimento para se realizar o mapeamento no QPFGN_m.

A função `constróiInequações` utiliza os dados contidos em Z_c , além dos números de agentes observadores e o número total de entidades espaciais para construir as inequações utilizadas pelo StarVars_m. Então, a função `buscaModeloVálido` retorna o modelo de mundo ψ como uma solução do sistema de inequações, representado pela matriz A e pelo vetor de coeficientes b , considerando $Ax \leq b$.

A solução das inequações acontece conforme exemplo dado na seção 2.2.1.5.3, onde cada agente observador representa um ponto orientado e, tanto o agente guiado, quanto o objetivo, são considerados pontos de referência. A diferença do exemplo da seção 2.2.1.5.3 para o QPFGN_m está no fato de que, no QPFGN_m, todos os agentes observadores já tem sua orientação definida, com exceção do agente guiado, que, independentemente de ter ou não sua orientação conhecida, será tratado como uma entidade sem orientação, ou um ponto de referência, para a resolução do sistema de inequações. Logo, não é a orientação, mas sim x e y de cada entidade espacial que espera-se inferir com o StarVars_m.

Uma vez que se tem a postura de todos os agentes observadores e as posições do agente guiado e do objetivo, pode-se representar cada um deles como uma partícula por meio da função `liberaPartículasc`. Considere o seguinte exemplo: quatro entidades espaciais, A , B , C e D , onde A e B são agentes observadores, com orientações 0° e 225° , respectivamente, C é

Algoritmo 6 – mapeamento no QPFGN_m.

```

1 Entrada:  $Z_c, n_o, n$ 
2 Saída:  $\psi$ 
3 início
4    $A, b = \text{constróiInequações}(Z_c, n, n_o)$ 
5    $\psi = \text{buscaModeloVálido}(A, b)$ 
6   retorna  $\psi$ 
7 fim

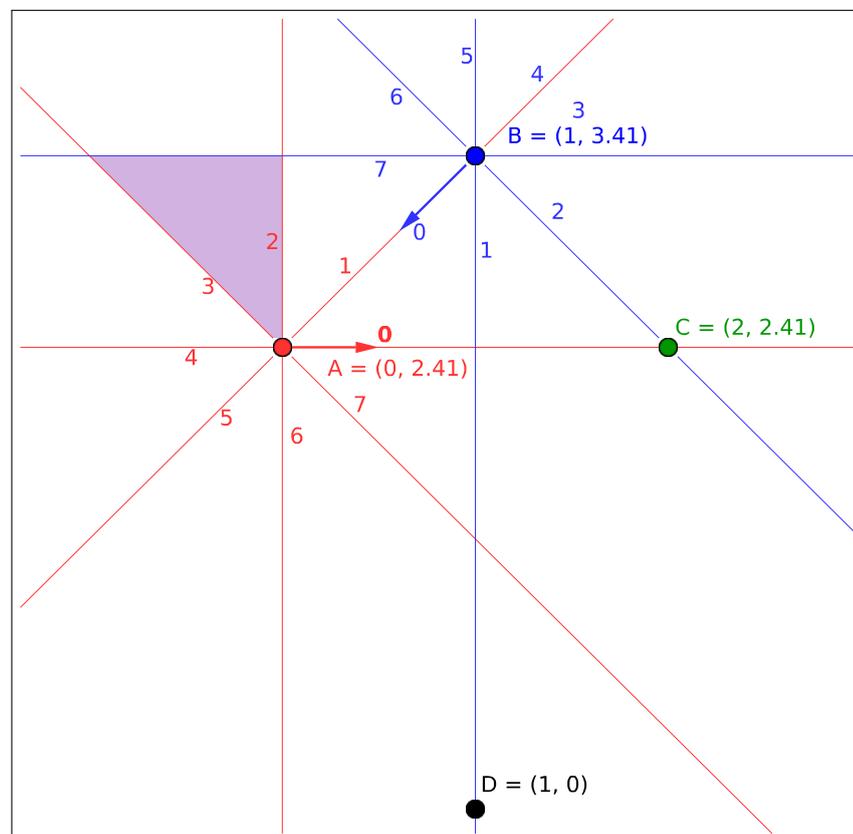
```

o agente guiado e D é o objetivo; $m = 8$; o conjunto de relações entre as entidades é $\varphi = \{A(\mathbf{1})B \wedge B(\mathbf{0})A \wedge A(\mathbf{0})C \wedge B(\mathbf{2})C \wedge A(\mathbf{6})D \wedge B(\mathbf{1})D\}$ e $s = -\mathbf{1}$. Um possível modelo de mundo retornado pelo StarVars₈, com os dados do exemplo, seria o da Figura 30.

Como cada agente observador possui direções qualitativas provenientes de setores angulares definidos com base na granularidade m , todo o domínio pode ser discretizado pelas regiões formadas pelas interseções de todos esses setores. Como exemplo, a região roxa em destaque na Figura 30 é formada pela interseção dos setores **(2)** do agente A e **(7)** do agente B . Assim, toda região formada pela interseção dos setores angulares será chamada q , sendo $q = [i_k, \dots, j_{n_o}]$, onde: $\mathbf{0} \leq i, j < m$, i é um setor do agente k , k deve ser o primeiro agente do sistema, j é um setor do último agente observador n_o do sistema. No caso do exemplo, $q = [2, 7]$. O conjunto contendo todas as regiões q será chamado $\Omega = [q_1, \dots, q_r]$, sendo r o número total de regiões formadas.

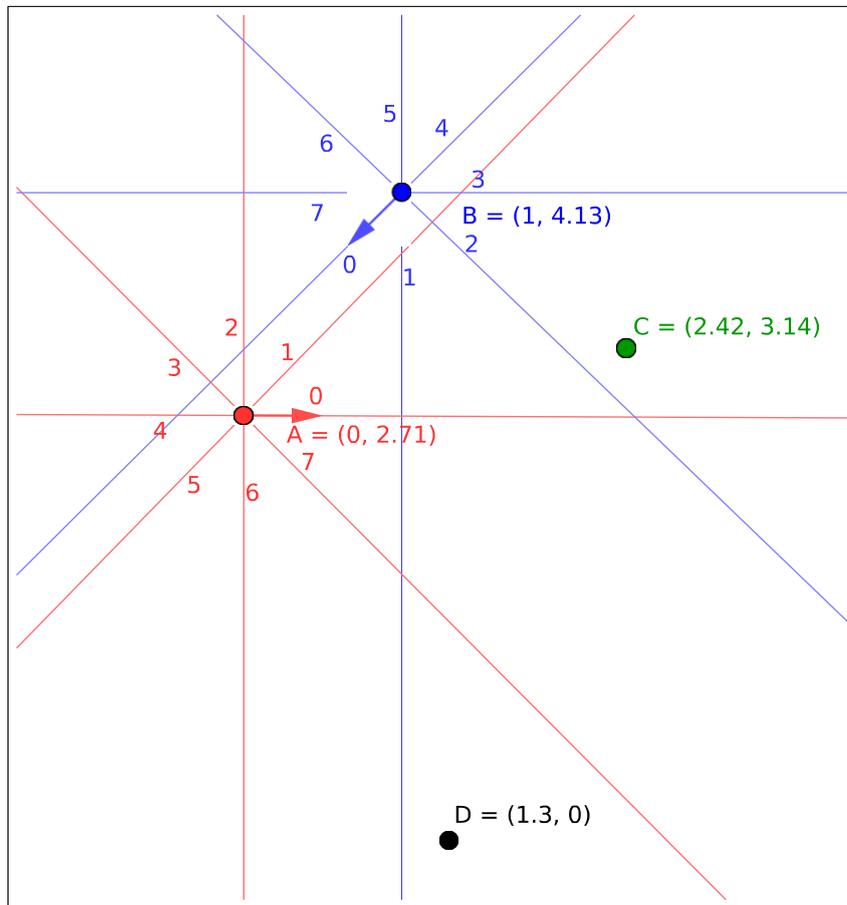
É possível notar, pela análise da Figura 30, que algumas linhas usadas para delimitar os setores angulares dos agentes A e B são coincidentes, como é o caso da linha que delimita o setor **(1)** do agente A com a linha que delimita o setor **(0)** do agente B . Dependendo do algoritmo utilizado para resolver a LP, isso pode acontecer, pois o algoritmo pode buscar valores para x e y , na Inequação (14), de forma que a resposta seja a mais próxima possível de $\mathbf{0}$, o que configura retas paralelas e, nesse caso coincidentes.

Figura 30 – Modelo de mundo retornado pelo StarVars₈



Fonte: Autor

Figura 31 – Modelo de mundo ψ retornado pelo StarVars₈ com $\varepsilon = -0.3$

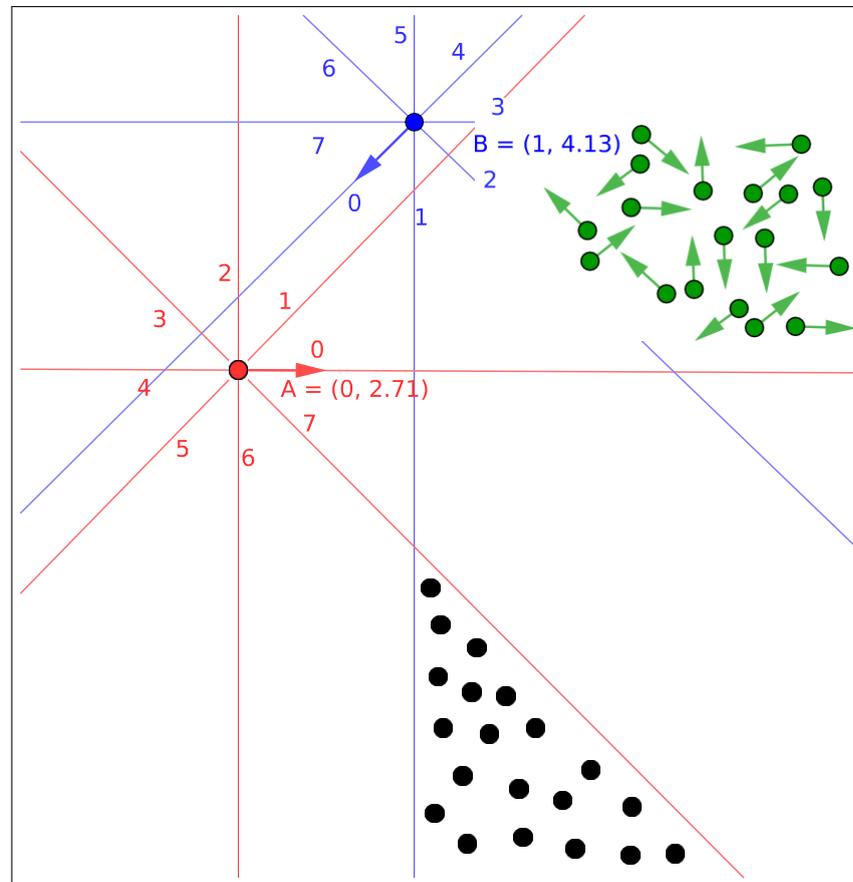


Fonte: Autor

Como a ideia é que o modelo de mundo obtido pelo StarVars_m possa representar, mesmo que de uma forma um pouco distorcida, o mapa contendo as localizações das entidades espaciais, não é desejável que a condição de linhas paralelas ocorra, pois é muito pouco provável que os agentes estejam, no mundo real, alinhados com tamanha precisão. Dessa forma, com o intuito de impedir esse alinhamento, independentemente do algoritmo utilizado para solucionar a LP, a Inequação (14) foi modificada para ser menor ou igual a um valor negativo próximo de **0**, que será chamado de ε , onde $\varepsilon < \mathbf{0}$. Isso impede o caso de igualdade com **0**, que configura sobreposição entre as retas. Essa modificação é possível, pois a restrição original continua sendo respeitada.

Com a substituição de **0** por ε na Inequação (14), o modelo de mundo retornado, passa a ter mais regiões, conforme pode ser notado na Figura 31, onde $\varepsilon = -0.3$. Assim, pode-se definir o número mínimo e máximo de regiões, formadas pela interseção dos setores angulares, que o domínio terá. O número mínimo pode ser obtido no caso em que todos os agentes observadores tem posição coincidente, logo o número mínimo será igual a m . O número máximo de regiões foi empiricamente obtido e pode ser encontrado na Equação (23), onde n_o é o número total de agentes observadores.

Figura 32 – Modelo de mundo retornado pelo StarVars₈ com partículas representando as posições localizações do objetivo e do agente guiado



Fonte: Autor

$$\text{número máximo de regiões} = 1 + \sum_{i=1}^{n_o} [(m-1) + (i-1)] \cdot \sum_{j=1}^{m/2} 2 \cdot (j-1). \quad (23)$$

É possível notar pela Figura 31 que o agente guiado C está em $q = [0,2]$ e o objetivo D está em $q = [6,1]$, da mesma forma que na Figura 30. A diferença está no fato de que algumas regiões que não existiam na Figura 30, como $q = [1,1]$, agora passaram a existir e nenhuma região ou relação inicial foi perdida ou prejudicada.

Uma das dificuldades encontradas na navegação guiada utilizando o QPFGN_m está no fato de que as regiões onde o agente guiado e o objetivo se encontram podem ser muito grandes, o que pode provocar bastante incerteza. Além disso, no caso em que o agente guiado não tem orientação conhecida, a inferência de instruções fica bastante comprometida.

Para superar esses problemas, mais partículas, distribuídas aleatoriamente, podem ser inseridas para representar as possíveis posições do objetivo e as possíveis posturas do agente guiado, dentro, somente, das regiões qualitativas a que cada um pertence. No momento da liberação das partículas que representam o agente guiado, a informação de orientação pode ser

utilizada, desde que seja conhecida. Caso o agente guiado não tenha orientação conhecida, as orientações das partículas serão aleatórias. Este procedimento é realizado dentro da função `liberaParticulas`. A Figura 32 demonstra o mapa criado pelo agente coordenador, com a inserção de N partículas para o objetivo e , também, para o agente guiado, considerando que o agente guiado não possuía orientação definida. Os conjuntos das partículas p para o objetivo e e para o agente guiado são, respectivamente, $S_{objetivo}$ e S_g , onde ambas são subconjuntos de S .

Conforme citado anteriormente, somente o agente guiado poderá se movimentar durante o processo de navegação. Assim, uma vez que todas as partículas necessárias para representar o domínio já foram criadas, a seleção da ação do agente guiado pode ser realizada, conforme demonstrado na seção 4.2.5. Depois da escolha da ação, o próximo passo do Algoritmo 5 é a predição, que, no domínio do QPFGN $_m$, deve ser realizada com os dados qualitativos presentes no problema.

4.3.1.1 *Predição: Inserção do Modelo Qualitativo de Movimento de Translação*

Normalmente, dentro do contexto de localização e rastreamento de agentes móveis, a predição é realizada com base no modelo de movimento. Conforme definido na seção 4.2.2.2, o agente guiado, no domínio deste trabalho, utiliza um modelo de movimento bastante simplificado, que é geral e parcial.

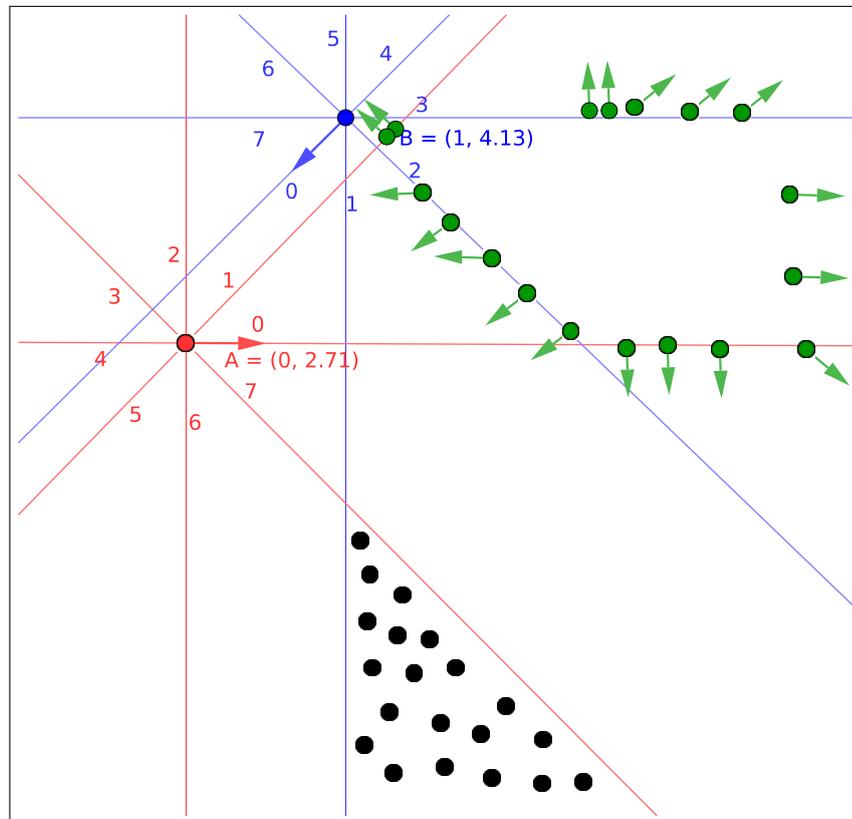
Assim, a pergunta que surge na etapa de predição é: como aplicar às partículas que representam as possíveis posturas do agente guiado um movimento semelhante ao que o agente guiado real vai executar? A resposta é trivial para os casos em que as instruções enviadas para o agente guiado são as relacionadas a algum tipo de rotação, seja para esquerda, direita ou para trás, pois, conforme seção 4.2.2.2, essas ações correspondem, respectivamente, a rotacionar **+90°**, **-90°** e **180°**, mais um ruído Gaussiano $\delta_t \sim N(\mathbf{0}, \sigma_{rotacao})$, onde $\sigma_{rotacao}$ é o desvio padrão para o movimento de rotação.

Contudo, no caso da ação “movimentar-se para frente”, a resposta à pergunta acima não é tão simples, pois, sem um modelo de velocidades ou sem um modelo de odometria, não se sabe exatamente até onde o agente guiado andará, o que torna bastante difícil a aplicação da mesma regra às partículas.

Dessa forma, a informação qualitativa das regiões pertencentes ao domínio Ω é utilizada. Assume-se que ação “movimentar-se para frente” é válida até que as partículas cruzem uma linha que delimite uma região q e o mesmo procedimento é aplicado ao agente guiado real, que deve andar para frente até que um dos agentes observadores consiga identificar que houve uma troca de região. Essa troca é a condição de parada da linha **17** do Algoritmo 5.

O agente observador que conseguir identificar a troca de região envia ao agente guiado a instrução “parar” (linha **23**). A Figura 33 exhibe a aplicação da instrução “movimentar-se para frente” para as partículas criadas na Figura 32. Portanto, para o QPFGN $_m$, a condição de parada da linha **18** do Algoritmo 5 é a troca da região $q \in \Omega$.

Figura 33 – Modelo qualitativo do movimento de translação aplicado às partículas que representam o agente guiado: andar para frente até trocar de região



Fonte: Autor

Pode-se notar, na Figura 33, que a maioria das partículas saiu da região $q = [0,2]$ e se deslocou para imediatamente o início de outras regiões adjacentes. As duas partículas cuja orientação estava em 0° , foram a exceção, pois continuaram na região $q = [0,2]$, uma vez que não era possível para elas trocar de região com o movimento “movimentar-se para frente”. Para tratar a exceção, todas as partículas estão sujeitas a uma quantidade máxima de passos para encontrar o limite da região em que se encontram, caso o limite seja atingido, elas param onde estiver dentro da mesma região que começaram.

Para lidar com alguns possíveis problemas que podem inviabilizar a parada do agente guiado real na primeira região adjacente, o que pode acontecer com regiões muito pequenas, por exemplo, uma porcentagem ζ das partículas troca de região mais uma vez. Assim, mesmo que o agente real passe por uma região e não tenha tempo hábil de parar, existirão partículas para representar a sua postura na próxima região também. O Algoritmo 7 exibe a função `prediçãoc` do $QPFGN_m$, que retorna um novo conjunto \overline{S}_g contendo as mesmas partículas que estavam em S_g , com as posturas atualizadas. A variável *dist* representa a distância que cada partícula deve percorrer em cada iteração; assim, *dist* deve assumir um valor pequeno para garantir a parada imediata na troca de região.

Algoritmo 7 – predição no QPFGN_m.

```

1 Entrada:  $a\check{c}\tilde{a}o$ ,  $S_g$ 
2 Saída:  $\overline{S_g}$ 
3 início
4    $S_g = \emptyset$ 
5   para cada partícula  $p_i \in S_g$ ,  $i = 0$  até  $N$  faça
6     se  $a\check{c}\tilde{a}o = \text{“Virar para esquerda e se movimentar para frente”}$  então
7        $\theta_i = \theta_i + 90^\circ + \delta_t$ 
8     fim
9     senão se  $a\check{c}\tilde{a}o = \text{“Virar para direita e se movimentar para frente”}$  então
10       $\theta_i = \theta_i - 90^\circ + \delta_t$ 
11    fim
12    senão se  $a\check{c}\tilde{a}o = \text{“Virar para trás e se movimentar para frente”}$  então
13       $\theta_i = \theta_i + 180^\circ + \delta_t$ 
14    fim
15    enquanto partícula  $p_i$  não mudar de região  $q$  faça
16       $x_i = x_i + \cos(\theta_i) * dist$ 
17       $y_i = y_i + \sin(\theta_i) * dist$ 
18    fim
19    sorteie  $q \sim U(0, 1)$ 
20    se  $q \stackrel{TM}{\leq} \zeta$  então
21      enquanto partícula  $p_i$  não mudar de região  $q$  faça
22         $x_i = x_i + \cos(\theta_i) * dist$ 
23         $y_i = y_i + \sin(\theta_i) * dist$ 
24      fim
25    fim
26    adicione  $p_i$  em  $\overline{S_g}$ 
27  fim
28  retorna  $\overline{S_g}$ 
29 fim

```

4.3.1.2 Atualização: Inserção do Modelo de Sensores Qualitativo

Como o agente guiado deste trabalho não percebe o domínio, a atualização acontece com base nas perspectivas dos agentes observadores. Dessa forma, depois que a predição é realizada e o agente guiado real cruza uma nova região, os agentes observadores enviam novamente as informações sobre as relações espaciais qualitativas do StarVars_m, obtidas por meio da função obtémDireção, na linha **21**, para o agente coordenador.

Assim, a função atualização_c do QPFGN_m contempla três etapas: a primeira é uma verificação da consistência dos dados de entrada, que é realizada por meio do sistema de inequações do StarVars_m, considerando, como entrada, Z_c obtido na linha **26** do Algoritmo 5. Caso a verificação retorne um modelo de mundo qualquer, os dados de entrada são consistentes e, por isso, o algoritmo continua, caso contrário, o Algoritmo geral 5 é reinicializado.

A segunda etapa é o cálculo do fator de importância w de cada partícula. Como a entrada Z_c contém a nova região q_g onde o agente guiado g está, é necessária a obtenção da região q_p onde cada partícula p se encontra, assim Ω_p representa o conjunto com q referente à cada partícula, $\Omega_p = \{q_1, \dots, q_N\}$, sendo N o número total de partículas. Tomando a Figura 33 como exemplo, $\Omega_p = \{[0, 1], [0, 1], \dots, [7, 2], [7, 2], \dots, [0, 2], \dots, [0, 3], \dots, [1, 2]\}$.

Uma vez que se tem q_g e o conjunto Ω_p , com q_p de cada partícula, obtém-se a probabilidade de cada partícula estar na mesma região do agente guiado. Ainda utilizando o exemplo da Figura 33, considerando o agente guiado C (representado pelas partículas orientadas), podemos supor que, depois de andar para frente, $q_C = [0, 1]$. Para se determinar o fator de importância w de cada partícula p , calcula-se a distância euclidiana entre q_C e q_p e, em seguida, obtém-se w conforme Equação (24).

$$w = \frac{1}{e^{-\sum_{i=1}^n (q_g^i - q^i)^2}}. \quad (24)$$

Finalmente, a última etapa da atualização é a reamostragem das partículas, que é feita com base em w . O Algoritmo 8 exibe a atualização completa.

Depois que a atualização é realizada, a partir ainda dos dados de entrada Z_c , a Inicialização por Sensor, apresentada na seção 2.1.1.2.3 é realizada para liberar uma baixa quantidade de partículas na região que foi, diretamente, obtida pela percepção dos robôs.

Algoritmo 8 – atualização no QPFGN_m.

```

1 Entrada:  $S_g, Z_c$ 
2 Saída:  $S_g$ 
3 início
4    $S_g = \emptyset$ 
5   se  $verificaConsistência(Z_c) == True$  então
6     para cada partícula  $p_i \in S_g$ ,  $i = 0$  até  $N$  faça
7        $w_i = \frac{1}{e^{-\sum_{j=1}^n (q_g^j - q_i^j)^2}}$ 
8     fim
9     para  $i = 0$  até  $N$  faça
10      sorteie  $m$  com probabilidade  $\propto w_m$ 
11      adicione  $p_m$  em  $S_g$ 
12    fim
13    retorna  $S_g$ 
14  fim
15  senão
16    recomeça Algoritmo Geral (Algoritmo 5)
17  fim
18 fim

```

4.3.2 Definições do Objetivo a ser Alcançado pelo Agente Guiado

O objetivo deve ser um marco ou ponto de referência que possa ser identificado, por meio de visão computacional, pelos agentes observadores. Não usar distâncias durante o processo de navegação guiada traz alguns inconvenientes, como, por exemplo, a definição do que pode ser considerado chegar ou não no objetivo. Como então estabelecer que o objetivo foi alcançado?

No QPFGN_m o objetivo será considerado alcançado quando o agente guiado chegar na mesma região q em que a entidade espacial que representa o objetivo se encontra. Assim, o objetivo será, na verdade, uma região qualitativa pertencente ao domínio Ω e, para guiar o agente até lá, a média das partículas que representavam a incerteza do objetivo dentro da região será utilizada.

Dessa forma, o agente coordenador conduzirá o agente guiado para, aproximadamente, o centro geométrico da região qualitativa q , na qual o objetivo se encontra.

O QPFGN_m permite a existência de somente um agente observador, o que torna uma região q igual a um setor angular deste único agente. Assim, este agente observador pode identificar em que setor qualitativo o objetivo está e em que setor o agente guiado está, e conduzir o agente guiado até a região do objetivo.

Quanto mais agentes observadores existirem no problema, mais precisa será a região do objetivo, pois será uma região q composta por mais perspectivas, tendo, então, maior probabilidade de ser uma região espacialmente menor.

4.3.3 Lidando com a Incerteza na Observação da Condição de Parada do Agente Guiado

Conforme estabelecido na seção 4.3.1.1, o agente guiado para de se movimentar quando a percepção de setor qualitativo com relação a este agente é alterada em pelo menos um dos agentes observadores. Quando um destes setores sofre alteração, a região q na qual o agente guiado se encontra é também modificada.

Dessa forma, o critério de parada estipula que o agente guiado deve parar sempre imediatamente após cruzar uma linha que define a fronteira entre dois setores qualitativos pertencentes ao mesmo agente observador. Contudo, a fronteira entre dois setores qualitativos é repleta de incerteza, uma vez que pode ser difícil para o agente observador identificar de forma absolutamente precisa em qual setor o agente guiado vai estar em um determinado instante t .

Qualquer pequeno erro na identificação correta da região pode resultar em falha de todo o processo de navegação guiada, pois os agentes observadores podem enviar instruções de parada para o agente guiado em instantes errados, o que resulta na divergência das partículas. Um exemplo será dado nos próximos parágrafos, no qual todas as linhas citadas são pertencentes ao Algoritmo 5.

O exemplo terá início na Figura 33, que representa o modelo de mundo interno criado pelo agente coordenador após a predição com a ação “movimentar-se para frente” (linha **15**). Após o recebimento da ação (linha **16**), o agente guiado vai se movimentar até cruzar uma linha de fronteira de setor. Deixe, então, a Figura 34 representar dois cenários: na coluna da esquerda o que será apresentado é uma representação da situação real; já na coluna da direita será exibido o modelo de mundo interno criado pelo agente coordenador.

Considere que na Figura 34a o agente *B* tenha enviado instrução de parada para o agente *C*, mesmo que *C*, representado pelo ponto, não tenha cruzado, de fato, do setor **(2)** para o setor **(1)** de *B*. Isso pode acontecer devido a incerteza existente nas imediações da fronteira entre dois setores. Então, a Figura 34b representa a atualização (linha **28**), que vai considerar que o agente está no setor **(1)** de *B*, e **(0)** de *A*: região $q_C = [0, 1]$. Depois da atualização, uma nova iteração é inicializada no Algoritmo 5 a partir da linha **13** e a ação escolhida pode ser, por exemplo, “virar para esquerda e se movimentar para frente”. A Figura 34c representa a predição (linha **15**) realizada pelo agente coordenador, onde as partículas executam a ação completa de virar e andar para frente. Finalmente, a Figura 34d exibe a representação do agente real *C* executando a ação recebida, contudo, desta vez, *B* tem a percepção correta da região na qual *C* está, o que faz com que a condição de parada da linha **17** seja atingida imediatamente e *C* não tem tempo de andar para frente. Isso causa uma dessincronização entre a postura do agente real *C* e das partículas que o representam. Este exemplo exibe o início do problema, que pode ficar cada vez maior conforme as iterações vão acontecendo.

Para superar o problema apresentado nos parágrafos anteriores, um modelo foi desenvolvido para desconsiderar todas as fronteiras formadas por linhas pares ou ímpares, dependendo da classificação da última fronteira cruzada, por agente observador. Este método observa se as linhas dos setores qualitativos são pares ou ímpares e utiliza sempre o oposto do que foi utilizado na condição de parada anterior para verificar a condição de parada atual. Quando as linhas pares ou ímpares são desconsideradas, regiões maiores, com a união das duas regiões que eram adjacentes são formadas, respeitando a característica da convexidade, demonstrada na seção 2.2.1.5.

Para explicar o método em detalhes, considere o exemplo da Figura 35, que vai abordar o mesmo problema que foi apresentado na Figura 34, mas, agora, com a aplicação do modelo que desconsidera as linhas pares ou ímpares.

A Figura 35a demonstra a situação em que *B* envia instrução de parada para o agente *C*, mesmo considerando que *C* não tenha, realmente, trocado de setor em *B*. Como a troca aconteceu da região **(2)** para a região **(1)**, a linha de fronteira entre estas duas regiões, que é par, é desativada, juntamente com todas as outras linhas pares do agente *B*. Assim, para o agente *B* o agente *C* está agora no setor que engloba os setores **(1)** e **(2)**: **[1,3]**. As trocas dos setores são individuais para cada agente observador, o que quer dizer que algum agente pode conter todas as linhas ativas, enquanto outros podem ter somente as linhas pares e outros somente as ímpares.

Figura 34 – Exemplo de dessincronização entre o agente guiado real e as partículas que o representam.

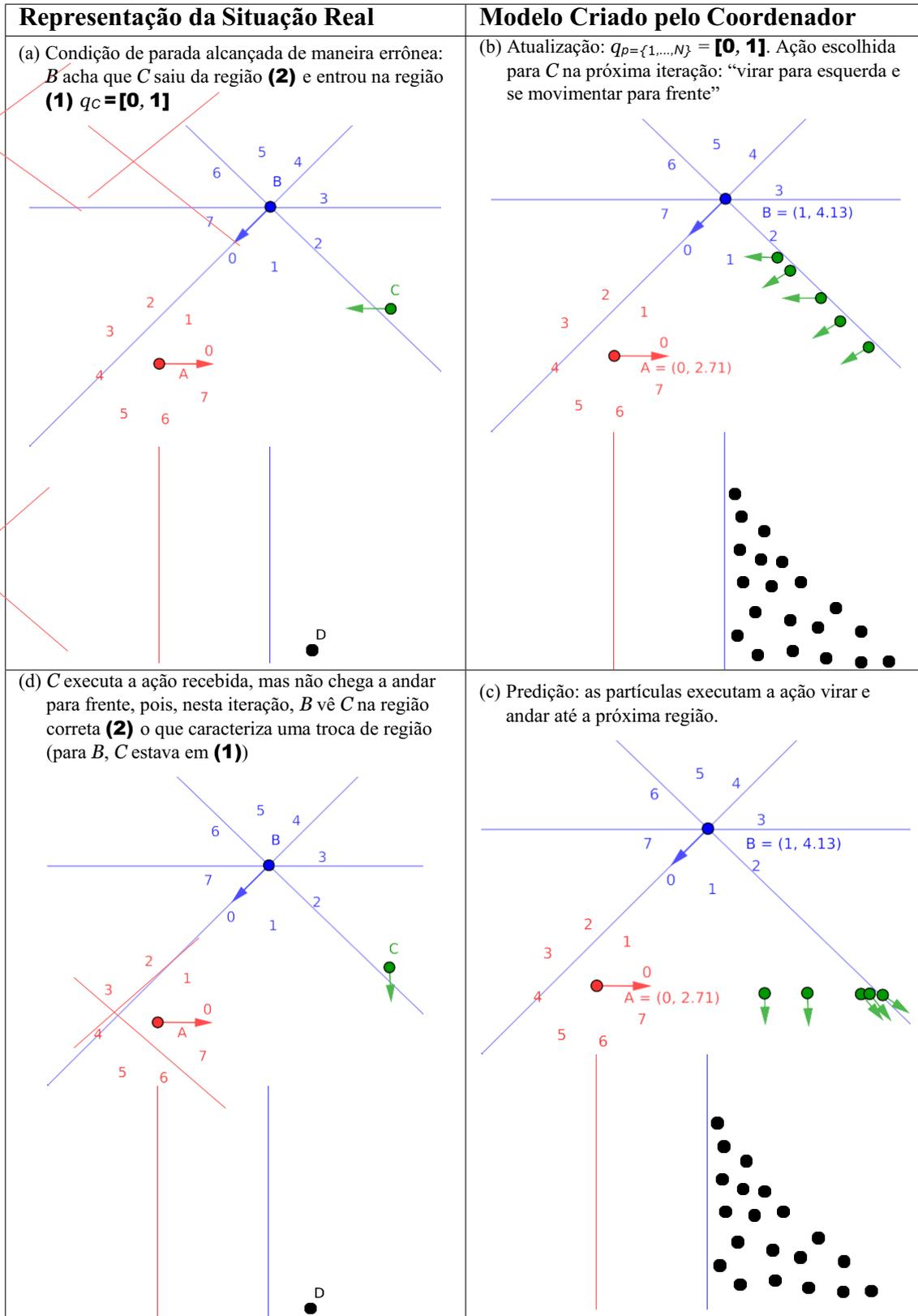
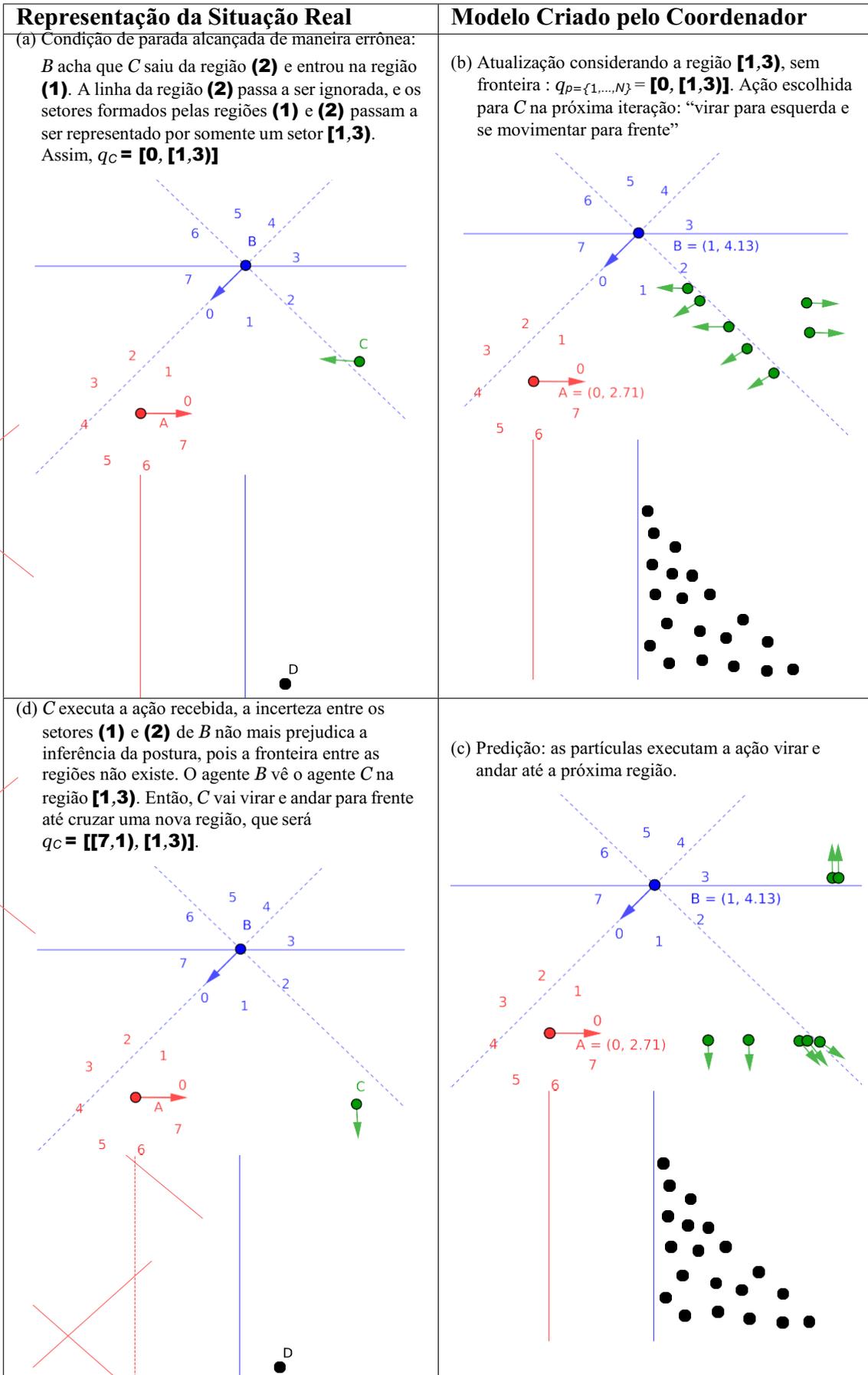


Figura 35 – Exemplo do modelo desenvolvido para desconsiderar as fronteiras pares ou ímpares em cada agente observador



Fonte: Autor

Legenda: as linhas tracejadas representam as linhas desativadas

Depois que pelo menos um dos agentes observadores reais identificou uma troca de região, a atualização das partículas deve acontecer no modelo criado pelo agente coordenador. A atualização, exibida na Figura 35b, vai também desconsiderar a linha que foi desativada, logo todas as partículas pertencentes a região $q = [0, [1, 3]]$ receberão probabilidade máxima de ser a partícula que melhor representa o agente guiado real.

Novamente a ação com maior probabilidade entre todas as partículas será a “virar para esquerda e se movimentar para frente”. Na Figura 35c, esta ação é aplicada a cada partícula. A Figura 35d exibe, então, o agente real C executando a mesma ação até cruzar a linha 0 do agente A . Assim, C passa a estar na região $q_C = [[7, 1), [1, 3]]$, que será a região utilizada para atualizar novamente as partículas e assim por diante.

No exemplo dado, depois da Figura 35d, se C cruzar alguma linha ímpar, o agente em questão passará a ignorar as fronteiras ímpares e as fronteiras pares voltam a estar ativas.

4.4 FILTRO DE PARTÍCULAS NUMÉRICO PARA NAVEGAÇÃO GUIADA COM INSTRUÇÕES QUALITATIVAS

Essa seção apresenta o Filtro de Partículas Numérico para Navegação Guiada com Instruções Qualitativas – *Numerical Particle Filter for Guided Navigation with Qualitative Instructions* (NPFNGQI $_m^r$), com granularidade m e taxa de atualização τ .

4.4.1 Localização e Mapeamento

Da mesma maneira que foi apresentado no QPFGN $_m$, a primeira etapa do NPFNGQI $_m^r$ é a obtenção das direções, conforme representado pela função `obtemDireção`. No filtro de partículas numérico, as direções são diretamente os ângulos α_{ij}^t , onde α_{ij}^t é um ângulo interpretado em $[0, 2\pi)$ obtido pelo agente i com relação à entidade espacial j em t . O agente i tem direção $\alpha_i = 0^\circ$ coincidindo com sua orientação θ_i . Dessa forma, $(z_i)_i^j = \alpha_{ij}^t$ em um instante t .

A Equação (25) exibe a regra para a obtenção de α_i .

$$\alpha_i = \begin{cases} \square & \\ \square & \alpha_i \\ \square & \alpha_i - 360 \end{cases} \quad \begin{cases} \text{se } \alpha_i < \mathbf{180}, \\ \\ \text{se } \alpha_i \geq \mathbf{180}. \end{cases} \quad (25)$$

A Figura 36 ilustra as direções numéricas α_{AB} e α_{AC} obtidas pelo agente A , que tem orientação 45° , com relação às entidades espaciais B e C .

A segunda etapa é a criação de um mapa com a localização de todas as entidades espaciais. Esse mapeamento é feito pelo agente coordenador, onde ele próprio será o centro do domínio. Dessa forma, todos os agentes observadores devem enviar as suas orientações θ_i , assim como as suas perspectivas, na forma de direções numéricas relativas α_{ij}^t , para o agente coordenador c . Além disso, caso o agente guiado conheça sua própria orientação, ele também a envia para o coordenador c .

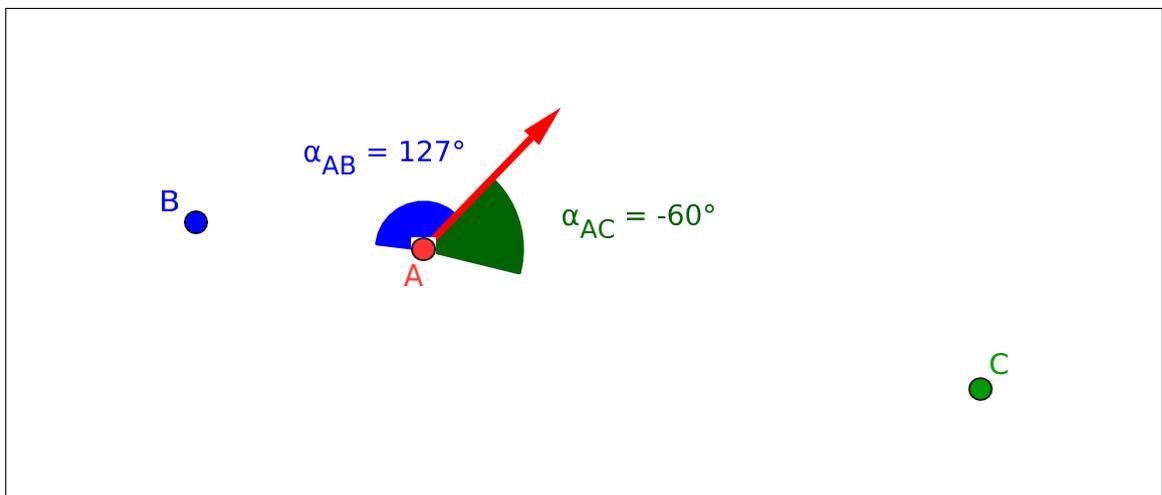
Para construir o mapa inicial de maneira numérica, duas etapas são executadas: a triangulação e a obtenção das coordenadas x e y de cada entidade espacial com base na posição do agente coordenador e em sua orientação. A triangulação é uma técnica tradicional da trigonometria que, neste trabalho, será responsável pela inferência das distâncias entre as entidades espaciais. Por sua vez, a obtenção das coordenadas x_i e y_i de uma entidade i é realizada utilizando-se o ângulo obtido pelo agente coordenador c com relação a i , α_{ci} , e a distância entre as entidades c e i .

Assim, o Algoritmo 9 demonstra o uso das funções para se construir o mapa com a localização de todas as entidades espaciais, onde um modelo de mundo ψ é retornado. O modelo de mundo ψ do NPFQI $_m^r$ segue a mesma definição da exibida na seção 4.3, $\psi = \bigwedge_{i \in \{1, \dots, m\}} (x_i, y_i, \theta_i)$, contudo θ é definido no domínio $\Theta = [0, 2\pi)$. A diferença é que ψ do QPFGN $_m$ é obtido como uma resposta do raciocínio qualitativo SV $_m$, enquanto que ψ do NPFQI $_m^r$ é obtido por meio de informações numéricas obtidas do domínio.

A função triangulação é responsável pela obtenção dos ângulos internos dos triângulos necessários à realização da localização de todas as entidades espaciais para, então, obter as distâncias, ou comprimentos, entre os vértices destes mesmos triângulos. Considere o exemplo da Figura 37, no qual as orientações de A e B são, respectivamente, 0° e 225° e as direções relativas são exibidas na própria figura. Considere também que o agente coordenador deste caso é o agente A e, como todas as outras entidades serão mapeadas com relação ao agente A , ele sempre vai precisar ser um dos vértices dos triângulos que serão formados.

O exemplo da Figura 37 permite a observação de que a obtenção dos ângulos internos dos vértices formados por agentes observadores é bastante direta: bastam algumas operações matemáticas para obtê-las. Por isso, e como estes ângulos são as únicas informações do problema, além do vértice que é formado pelo agente coordenador, um segundo vértice precisa

Figura 36 – Direções numéricas obtidas pelo agente A , que tem orientação 45° , com relação às entidades B e C



Fonte: Autor

sempre ser formado por um agente observador, pois, assim, a obtenção do terceiro ângulo, que pode ter como vértice o objetivo, o agente guiado ou outro agente observador, passa a ser simples. Dessa forma, a obtenção dos ângulos internos dos triângulos necessários para se localizar todas as entidades, fica sempre fixa em dois agentes observadores: o coordenador e um outro agente observador que será chamado de âncora.

O problema maior está na inferência das distâncias, pois, conforme foi dito na seção 4.2, nenhuma informação de distância é diretamente extraída do domínio. Assim, e considerando que para localizar todas as entidades em seus devidos lugares alguma informação de distância é necessária, uma distância de referência d é assumida entre o agente coordenador e o agente âncora. A distância d serve como um fator de escala da representação interna que será criada pelo agente coordenador.

Uma vez que existe uma distância e os ângulos internos de todos os triângulos necessários para a localização de todos os agentes, com o uso da lei dos senos todas as outras distâncias podem ser encontradas. A lei dos senos relaciona os comprimentos dos lados de um triângulo com os senos dos seus ângulos internos: $a/\text{sen}(\alpha) = b/\text{sen}(\beta) = c/\text{sen}(\gamma)$, sendo α , β e γ os ângulos internos e a, b e c os comprimentos opostos aos ângulos, respectivamente.

O Algoritmo 10 demonstra o processo para se obter todos os ângulos internos dos triângulos necessários para a criação do mapa interno do agente coordenador, assim como o processo de obtenção das distâncias entre as entidades espaciais, ou comprimentos dos lados dos triângulos, onde γ representa o agente âncora, c representa o agente coordenador, λ_i as outras i entidades espaciais, T a matriz com dimensão $e \times f$ contendo os ângulos internos, na coluna f , do triângulo e em t e D a matriz com dimensão $e \times f$ que contém as distâncias, na coluna f , entre os vértices do triângulo e em t . As variáveis p, q, r e $temp$ são todas auxiliares.

Depois que todos os triângulos tiverem sido construídos, a localização de todas as entidades espaciais deve acontecer, com base no agente coordenador, que tem os ângulos e as distâncias para todas as entidades. Assim, considerando que o agente coordenador está em $x = \mathbf{0}$ e $y = \mathbf{0}$, a função `obtençãoCoordenadasEntidades`, exibida no Algoritmo 11, demonstra como a localização das entidades espaciais é realizada.

Com o mapeamento realizado, utiliza-se a função `liberaPartículasc` para representar cada entidade espacial. Como o mapa foi construído com informações numéricas, a

Algoritmo 9 – mapeamento no NPFGNQI_m^r

1 **Entrada:** Z_c, n_o, n

2 **Saída:** ψ

3 **início**

4 $D = \text{triangulação}(Z_c, n_o)$

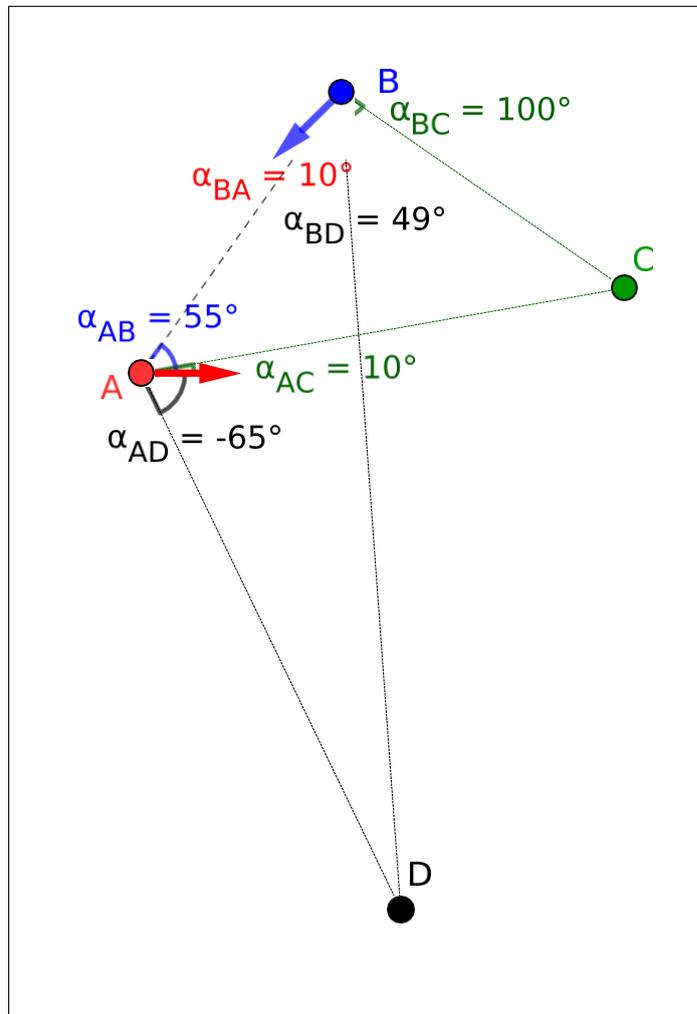
5 $X, Y = \text{obtençãoCoordenadasEntidades}(Z_c, D, n)$

6 $\psi = \langle X, Y, \Theta \rangle$

7 **retorna** ψ

8 **fim**

Figura 37 – Exemplo da obtenção dos ângulos internos dos triângulos formados pelos agentes espaciais



Fonte: Autor

incerteza das posições iniciais é menor do que no QPFGN_m. Porém, para o caso do agente guiado, a incerteza vai aumentar conforme ele se movimentar, então, N partículas são aleatoriamente distribuídas para representar suas possíveis posturas, sendo S_g o conjunto com todas essas partículas. Assim como no QPFGN_m, a liberação das partículas que representam o agente guiado g pode ser feita com ou sem a informação da orientação θ_g ; caso a informação não exista, valores aleatórios são considerados.

Depois que todas as partículas tiverem sido liberadas, a seleção da ação do agente guiado, $escolheAção_g^s$, pode ser realizada seguindo o procedimento demonstrado na seção 4.2.5. Assim que a escolha da ação for realizada, faz-se a predição, que, no domínio do NPFNGN_m^r, deve ser realizada com os dados numéricos de direção.

4.4.1.1 Predição: Inserção do Modelo Aleatório de Movimento de Translação

O problema de predição no NPFNGNQI_m^r é o mesmo do apresentado na seção 4.3.1.1: como refletir o movimento do agente guiado real nas partículas, dado que o modelo de movimento é bastante simplificado?

Para as instruções relacionadas a qualquer tipo de rotação, a resposta segue a mesma solução apresentada no QPFNG_m (seção 4.3.1.1). Contudo, para a ação “movimentar-se para frente”, como não se sabe exatamente até onde o agente guiado deve se movimentar, cada partícula se movimenta para frente com valores aleatórios gerados a partir de uma distribuição Gaussiana.

Logo, a Equação (26) exibe a amostragem da distância $dist$ que será percorrida por cada partícula, lembrando que $dist$ é adimensional, logo não é a distância real percorrida pelo agente

Algoritmo 10 – triangulação – Função que realiza a triangulação.

```

1 Entrada:  $Z_c, n_o$ 
2 Saída:  $D$ 
3 início
4    $D = T = \emptyset$ 
5   para  $i = 0$  até  $n_o$  faça
6      $p = c$ 
7      $q = \gamma$ 
8      $r = \lambda_{i+2}$ 
9     para  $j = 0$  até  $1$  faça
10       $\hat{a}ngulo = |\alpha_{pq} - \alpha_{pr}|$ 
11      se  $\hat{a}ngulo$  “  $180$  ” então
12         $\hat{a}ngulo = |\hat{a}ngulo - 360$ 
13      fim
14      adicione  $\hat{a}ngulo$  em  $T_i$ 
15       $temp = p$ 
16       $p = q$ 
17       $q = r$ 
18       $r = temp$ 
19    fim
20     $T = T \cup (180 \tau \sum_{k=0}^1 T_{[i,k]})$ 
21    para  $j = 0$  até  $2$  faça
22      se  $j = 2$  então
23        adicione  $d \cdot \frac{sen(T_{[i,j]})}{sen(T_{[i,2]})}$  em  $D_i$ 
24      fim
25      senão
26        adicione  $d$  em  $D_i$ 
27      fim
28    fim
29  fim
30  retorna  $D$ 
31 fim

```

Algoritmo 11 – obtençãoCoordenadasEntidades – Função para obtenção das coordenadas das entidades.

```

1 Entrada:  $Z_c, D, n$ 
2 Saída:  $X, Y$ 

3 início
4    $X = Y = \emptyset$ 
5   para  $i = 1$  até  $n$  faça
6     adicione  $\cos(\alpha_{0i} + \theta_c) * D_{[i-2,1]}$  em  $X$ 
7     adicione  $\sin(\alpha_{0i} + \theta_c) * D_{[i-2,1]}$  em  $Y$ 
8   fim
9   retorna  $X, Y$ 
10 fim

```

guiado, mas sim uma distância pertencente ao modelo de mundo criado com base em d ; a média é sempre $\mathbf{0}$ e $\sigma_{translacao}$ representa o desvio padrão para os movimentos de translação. Pode-se notar que a Equação (26) utiliza módulo, pois o robô não deve se movimentar para trás sem antes virar $180^\circ + \delta_t$.

$$dist \sim |(\mathbf{N}(\mathbf{0}, \sigma_{translacao}))|. \quad (26)$$

Dessa forma, $\sigma_{translacao}$ deve ser ajustado de acordo com d para permitir uma distribuição satisfatória das partículas dada a instrução “movimentar-se para frente”. Como a média é sempre $\mathbf{0}$, a maior concentração destas partículas será próxima da localização atual.

O Algoritmo 12 exibe o procedimento para se realizar a predição no NPFNGNQI $_m^r$.

4.4.1.2 Atualização: Inserção do Modelo de Sensores Numérico

Uma vez que as partículas foram espalhadas, conforme predição apresentada na seção anterior, a atualização do fator de importância w de cada partícula é necessária para determinar a probabilidade que cada partícula tem de representar a postura real do agente guiado.

Contudo, no NPFNGNQI $_m^r$ um problema pode ser encontrado na linha **17** do Algoritmo 5: qual é a condição de parada da movimentação do agente guiado? No QPFNG $_m$, por exemplo, sabe-se que a condição de parada está vinculada com a troca de região qualitativa, porém no NPFNGNQI $_m^r$ não se tem um gatilho desse tipo.

É a condição de parada que estabelece quando a atualização deve ocorrer. Dessa forma, uma condição de parada foi definida com base no movimento de translação do agente guiado e em uma taxa de atualização τ . Resumidamente, a cada τ segundos o agente guiado deve parar de se movimentar e a atualização deve acontecer, desde que o movimento que estiver sendo realizado seja o de translação. A atualização não deve acontecer após somente um movimento de rotação, pois o conjunto Z_c não terá sofrido qualquer alteração.

A identificação do movimento de translação é feita por meio da criação de uma região, com tamanho angular h , por cada agente observador. Assim, enquanto o agente guiado conti-

Algoritmo 12 – predição no NPFQI_m^r.

```

1 Entrada:  $a\check{c}\tilde{a}o$ ,  $S_g$ 
2 Saída:  $S_g$ 
3 início
4    $S_g = \emptyset$ 
5   para cada partícula  $p_i \in S_g$ ,  $i = 0$  até  $N$  faça
6     se  $a\check{c}\tilde{a}o = \text{“Virar para esquerda e se movimentar para frente”}$  então
7        $\theta_i = \theta_i + 90^\circ + \delta_t$ 
8     fim
9     senão se  $a\check{c}\tilde{a}o = \text{“Virar para direita e se movimentar para frente”}$  então
10       $\theta_i = \theta_i - 90^\circ + \delta_t$ 
11    fim
12    senão se  $a\check{c}\tilde{a}o = \text{“Virar para trás e se movimentar para frente”}$  então
13       $\theta_i = \theta_i + 180^\circ + \delta_t$ 
14    fim
15     $dist_i \sim |(\mathbf{N}(\mathbf{0}, \sigma_{translacao})|$ 
16     $x_i = x_i + \cos(\theta_i) * dist_i$ 
17     $y_i = y_i + \sin(\theta_i) * dist_i$ 
18    adicione  $p_i$  em  $S_g$ 
19  fim
20  retorna  $S_g$ 
21 fim

```

nuar dentro da mesma região h para cada agente observador a condição de parada não é atingida. Portanto, de maneira mais específica, a linha **17** do Algoritmo 5 pode ser: **enquanto** $tempo < \tau$ **ou** $z_{ig}^{TM} h_i$ **faça**.

Quando a condição de parada for atingida, os agentes observadores devem enviar novamente as informações sobre as relações espaciais numéricas, obtidas por meio da função $obtemDireção$, para o agente coordenador. Então, a função $atualizacao_c$ será executada, assim como no QPFGN_m, em três etapas: primeiro o agente guiado deverá ser realocado no mapa inicial, depois o cálculo do fator de importância de cada partícula é realizado e, por fim, a reamostragem das partículas é feita.

A realocação do agente guiado é feita com base no mapeamento exibido no Algoritmo 9, onde um novo conjunto Z_c , com as novas direções percebidas pelos agentes observadores com relação ao agente guiado, será a entrada. Dessa forma, todas as posturas e posições retornados serão as mesmas, com exceção do agente guiado, que terá uma nova posição x_g e y_g . O Algoritmo 13 exhibe o procedimento para atualização numérica.

O cálculo do fator de importância w é feito com base em uma função Gaussiana, onde $\sigma_{atualizacao}$ representa o desvio padrão para a atualização das partículas e o valor da variável v é a distância euclidiana entre x_i e y_i de cada partícula p_i e a nova posição x_g e y_g do agente guiado.

Da mesma forma que no QPFGN_m, depois que a atualização é realizada, a partir dos dados de entrada Z_c , a Inicialização por Sensor é realizada para libe-

Algoritmo 13 – atualização no NPF_{GNQI}^r_m

```

1 Entrada:  $S_g, Z_c$ 
2 Saída:  $S_g$ 
3 início
4    $S_g = \emptyset$ 
5   para cada partícula  $p_i \in S_g, i = 0$  até  $N$  faça
6      $v = \sqrt{(x_g - x_i)^2 + (y_g - y_i)^2}$ 
7      $w_i = \frac{e^{-\frac{(v-\mu)^2}{2\sigma^2}}}{\sigma_{atualizacao} \sqrt{2\pi}}$ 
8   fim
9   para  $i = 0$  até  $N$  faça
10    sorteie  $m$  com probabilidade  $\propto w_m$ 
11    adicione  $p_m$  em  $S_g$ 
12  fim
13  retorna  $S_g$ 
14 fim

```

rar algumas partículas na posição que foi inferida pelas funções triangulação e obtençãoCoordenadasEntidades, realizadas durante a realocização do agente guiado.

4.4.2 Definições do Objetivo a ser Alcançado pelo Agente Guiado

O objetivo é considerado alcançado no NPF_{GNQI}^r_m pela mesma definição apresentada no QPFGN_m (seção 4.3.2), ou seja, quando o agente guiado chegar na mesma região q na qual a entidade espacial objetivo se encontrada. Esse método foi adotado, pois, mesmo considerando a natureza numérica do NPF_{GNQI}^r_m, não é trivial estabelecer que um objetivo é alcançado somente com base em direções numéricas, pois guiar um agente com preposições espaciais projetivas até um ponto tão específico do espaço \mathbf{R}^2 pode ser uma tarefa bastante complicada.

Para superar esse problema, um relaxamento das direções percebidas pelos agentes observadores com relação ao objetivo se faz necessário. Esse relaxamento pode ser atingido por meio da discretização espacial do StarVars_m, transformando-se o objetivo em uma região q inteira.

A diferença entre o NPF_{GNQI}^r_m e o QPFGN_m está no fato de que no NPF_{GNQI}^r_m o objetivo não é representado com um conjunto de partículas, mas por somente uma, pois a incerteza com relação ao objetivo é bem menor. Assim, o agente coordenador guiará o agente guiado com base no objetivo real e não com base no centro de massa da região qualitativa q do objetivo.

4.5 COMENTÁRIOS

A seção 4 descreveu os dois modelos híbridos, qualitativo-probabilísticos, propostos para serem utilizados em problemas de navegação guiada de um agente que não pode perceber o mundo, por meio de instruções baseadas em preposições linguísticas projetivas.

O primeiro modelo apresentado, QPFGN_m, utiliza dados qualitativos para alimentar um filtro de partículas que é responsável por rastrear o agente guiado por todo o percurso que ele deve executar até o objetivo ser alcançado. Neste modelo, tanto a predição do movimento de translação, quanto a atualização são realizadas com base nas regiões qualitativas geradas por meio da junção dos setores angulares definidos no StarVars_m. Como todo o problema se baseia na discretização do StarVars, as orientações de todos os agentes autônomos devem ser aproximadas para estarem inclusas em $\Theta_m = \{0\eta, 1\eta, 2\eta, \dots, (m - 1)\eta\}$, que representa o domínio da orientação com $\eta = 360^\circ/m$. Toda inferência de instrução que será enviada ao agente guiado é feita com base nas partículas que representam as suas possíveis posturas. A granularidade m é bastante importante no QPFGN_m, pois define as regras de atualização e de predição, nos casos de movimento de translação, além do tamanho da região do objetivo.

Já o segundo modelo, NPFGNQI_m^r utiliza dados numéricos de direção e de orientação (definidos entre $[0, 2\pi)$) para alimentar um filtro de partículas convencional e, por meio destas partículas, rastrear e inferir instruções numéricas que, na sequência, são transformadas em instruções baseadas em preposições projetivas, para o agente guiado, com o intuito de conduzi-lo até o objetivo. No NPFGNQI_m^r a predição do movimento de translação é feita com dados aleatórios e a atualização dos pesos das partículas feita com base em uma distribuição Gaussiana com desvio definido, além disso, a granularidade m só é responsável por definir o tamanho da região na qual o objetivo está incluso. A regra de atualização é feita com base na taxa de atualização τ , que, de uma forma mais simplificada, acaba fazendo a mesma função que seria feita por um modelo de movimento, pois τ precisa ser regulado de acordo com a velocidade de translação do agente para funcionar de forma a manter uma boa eficiência. Uma vantagem do NPFGNQI_m^r com relação ao QPFGN_m é o fato de que as orientações θ_i dos agentes i estão no domínio entre $[0, 2\pi)$, enquanto que no QPFGN_m as orientações estão no domínio definido por Θ_m , onde $\Theta_m = \{0\eta, 1\eta, 2\eta, \dots, (m - 1)\eta\}$ e $\eta = 360^\circ/m$.

Em ambos os modelos, os movimentos de rotação do agente guiado não são problemáticos, pois as instruções de virar para direita, para esquerda ou para trás são completamente definidas.

Os dois modelos trazem, como contribuição principal, a possibilidade de se guiar um agente autônomo por meio das perspectivas de outros agentes autônomos, que, neste trabalho, são definidos como observadores, utilizando preposições projetivas linguísticas. Isto faz com que ambos os métodos tenham alto nível de abstração espacial. Além disso, os dois modelos podem acomodar agentes guiados com ou sem orientações conhecidas, o que é importante para manter a genericidade do agente guiado, que pode ser um humano ou um robô. Por exemplo,

um humano normalmente não sabe sua própria orientação, ainda mais se ele não pode enxergar o ambiente ao redor, porém, em vários casos, robôs conseguem obter sua própria orientação de maneira independente da visão computacional, por meio de sensores inerciais.

A próxima seção trará os resultados e as análises dos experimentos realizados com robôs humanoides virtuais e de duas provas de conceito que foram realizadas com robôs humanoides reais.

5 EXPERIMENTOS

Com o objetivo de verificar e avaliar o comportamento dos modelos propostos neste trabalho, alguns experimentos foram realizados com robôs humanoides virtuais, em que quatro modelos de navegação guiada foram avaliados e comparados entre si: dois puramente qualitativos, baseados no $StarVars_m$ e os dois híbridos propostos neste trabalho – $QPFGN_m$ e $NPFGNQI_m^r$. Além dos experimentos, duas provas de conceito com robôs humanoides reais foram efetuadas: uma para o $QPFGN_8$ e uma para o $NPFGNQI_{16}^6$.

Esta seção está dividida em três partes principais: domínio, experimentos simulados e prova de conceito com robôs reais.

5.1 DOMÍNIO

Todos os experimentos e provas de conceito foram realizados com robôs humanoides que seguem as regras e restrições impostas pelo domínio encontrado na *RoboCup Soccer Humanoid League* (ROBOCUP, 2015), que é a competição mundial de futebol de robôs humanoides.

5.1.1 RoboCup

A RoboCup foi proposta por Kitano et al. (1997) com o intuito de fornecer um ambiente desafiador capaz de promover a pesquisa nas áreas da Robótica e da Inteligência Artificial. O objetivo geral da RoboCup é o de alcançar, no meio do século XXI, um time de robôs completamente autônomos, capaz de jogar, seguindo as regras oficiais da FIFA, e ganhar do time de humanos campeões da última Copa do Mundo de futebol na época.

O futebol foi escolhido como a plataforma padrão da RoboCup porque este domínio oferece uma grande quantidade de problemas de Robótica para a Inteligência Artificial, como por exemplo, problemas de navegação, sistemas multiagentes cooperativos, aprendizagem de máquina, estratégias, problemas gerais de hardware etc. (DYLLA et al., 2008).

Atualmente a RoboCup é mais do que simplesmente futebol de robôs. A competição engloba problemas de resgate em desastres, de robôs assistivos para tarefas domésticas e de robôs incluídos em ambientes de trabalho, como os industriais.

As ligas do RoboCup estão divididas da seguinte maneira:

- a) RoboCup *Soccer*
- *Small Size League*
 - *Middle Size League*
 - *Humanoid League*
 - *Standard Platform League*
 - *Soccer Simulation*

- b) RoboCup *Rescue*
- c) RoboCup@ *Home*
- d) RoboCup@ *Work*
- e) RoboCup Junior

Os experimentos deste trabalho serão relacionados com a Liga Humanoide da RoboCup *Soccer – Humanoid League*. A Liga Humanoide, que teve seu início em 2002, é a liga que mais se aproxima do objetivo geral da RoboCup. Nessa liga, robôs com características similares às humanas, como dimensões do corpo e sistemas perceptivos, devem ser capazes de jogar futebol de maneira autônoma. A liga é subdividida em 3 categorias, que variam de acordo com a altura dos robôs: *KidSize*, até 90cm, *TeenSize*, até 140cm, e *AdultSize*, até 180cm.

5.1.2 Robôs Humanoides

Isaac Asimov, a partir do seu livro “Eu, Robô” (ASIMOV, 1950), foi um dos grandes responsáveis pela difusão da ideia de robôs cuja aparência global fosse baseada na estrutura física do corpo humano, ou seja, robôs humanoides.

Os robôs humanoides da atualidade são agentes mecatrônicos, que combinam mecânica, eletrônica e computação para o desenvolvimento de sistemas autônomos inteligentes. Esses robôs têm sido desenvolvidos desde a década de 70, com a criação do Wabot-1¹ pela Universidade de Waseda, no Japão.

Mais recentemente, as empresas têm também desenvolvido os seus robôs humanoides, como o ASIMO² da Honda, que teve sua primeira versão lançada no ano 2000 e atualizada em 2011; o NAO³ da antiga Aldebaran, atual SoftBank Robotics, cujo projeto teve início em 2004 e passou pela última atualização em 2014; e o Atlas⁴ da Boston Dynamics, revelado em 2013, é um robô humanoide com 1.80m de altura desenvolvido com fundos da Agência de Projetos de Pesquisa Avançada de Defesa norte-americana – *Defense Advanced Research Projects Agency* (DARPA).

Outro importante humanoide, criado a partir de um projeto conjunto entre universidades norte-americanas e uma empresa sul-coreana, é o DARwIn-OP (HA et al., 2011). Divulgado em 2010, ele foi desenvolvido pela Universidade Virginia Tech, com a colaboração das Universidades da Pennsylvania e de Purdue e da empresa Robotis. Esse robô é importante pois tem hardware e software livres e disponíveis para download, o que permite que qualquer pessoa ou instituição possa montar um robô humanoide.

Os robôs humanoides que serão utilizados como base para os experimentos deste trabalho são todos construídos com base na mecânica do DARwIn-OP (HA et al., 2011) e foram todos desenvolvidos no Centro Universitário FEI por alunos de graduação, mestrado e douto-

¹http://www.humanoid.waseda.ac.jp/booklet/kato_2.html

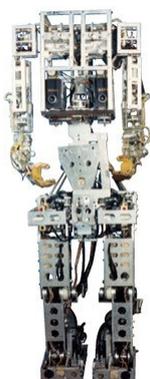
²<http://world.honda.com/ASIMO/>

³<https://www.ald.softbankrobotics.com/en/robots/nao>

⁴<https://www.bostondynamics.com/atlas>

Figura 38 – Exemplos de robôs humanoides

(a) Wabot-1



Fonte: Waseda, 1973

(b) NAO



Fonte: SoftBank, s.d.

(c) ASIMO



Fonte: Honda, s.d.

(d) DARwIn-OP



Fonte: HA et al., 2011

(e) Atlas



Fonte: DARPA Robotics Challenge, 2013

rado, para serem parte do time RoboFEI-HT. Todos os robôs são compostos por vinte graus de liberdade – *degrees of freedom* (DOF), sendo seis por perna, três por braço e dois no pescoço. A Tabela 4 expõe as principais características dos robôs e a Figura 39 exhibe uma imagem dos quatro robôs humanoides. A seção 5.3, que apresentará a prova de conceito com robôs reais, abordará ainda mais informações sobre estes humanoides.

Durante a fase de desenvolvimento desses robôs, definiu-se que a quantidade de placas eletrônicas seria reduzida, com robôs que não utilizam microcontroladores. Todos os seus processos são executados em um único computador, incluindo o controle dos motores. Assim, para lidar com a execução de todos os processos necessários – visão, localização, decisão, comunicação, planejamento e controle de movimentos – ao mesmo tempo, uma nova arquitetura híbrida de software foi proposta por Perico et al. (2014).

A Arquitetura em Cruz (PERICO et al., 2014), como foi chamada essa nova arquitetura, pode ser vista na Figura 40, na qual cada caixa da imagem representa um processo completa-

Tabela 4 – Características dos Robôs

Altura	490 mm
Peso	3.1 Kg
Velocidade do andar	10 cm/s
DOF	20 no total: 6 por perna, 3 por braço e 2 no pescoço
Motores	Dynamixel RX-28
Protocolo dos motores	RS-485
IMU	CH Robotics UM7
Câmera	Logitech Full-HD Pro Webcam C920
Central de Processamento	Intel NUC Core i5, 8GB SDRAM e 120GB SSD
Bateria	LiPo 18.5V - 2000mAh

Fonte: Autor

mente independente para o computador e n representa o número do robô. Essa arquitetura traz algumas vantagens, como por exemplo, o fato dos processos poderem trabalhar em paralelo e em linguagens diferentes de programação.

Como os processos são independentes, se fez necessário a adoção de uma estrutura de comunicação entre eles. Assim, a Arquitetura em Cruz faz uso de memória compartilhada, chamada de *blackboard*, representado por *bkb* na Figura 40, para permitir que vários processos possam acessar variáveis compartilhadas simultaneamente.

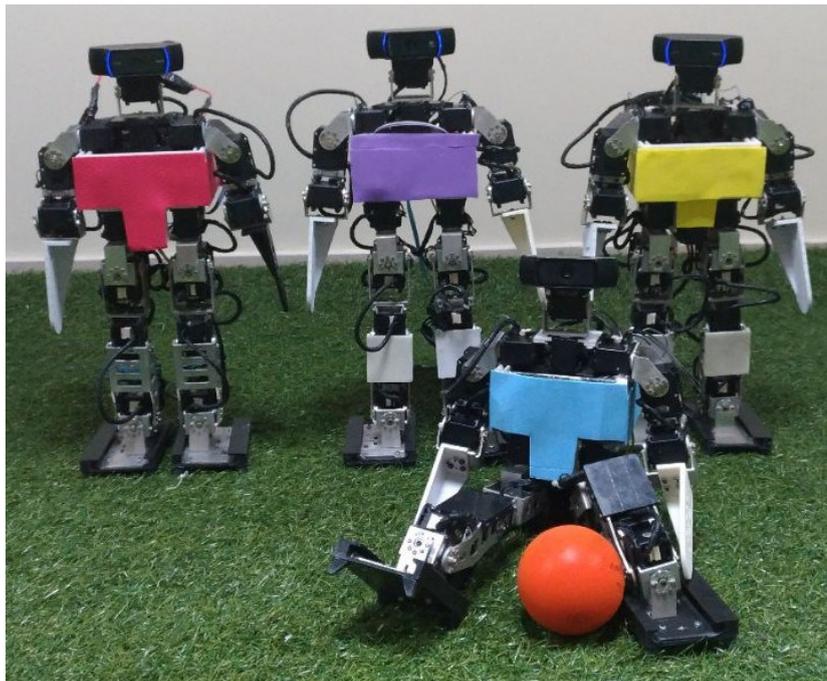
5.2 EXPERIMENTOS SIMULADOS

Nesta seção são exibidos o simulador utilizado, o detalhamento dos experimentos e os resultados obtidos.

5.2.1 O Simulador RoboFEI-HT

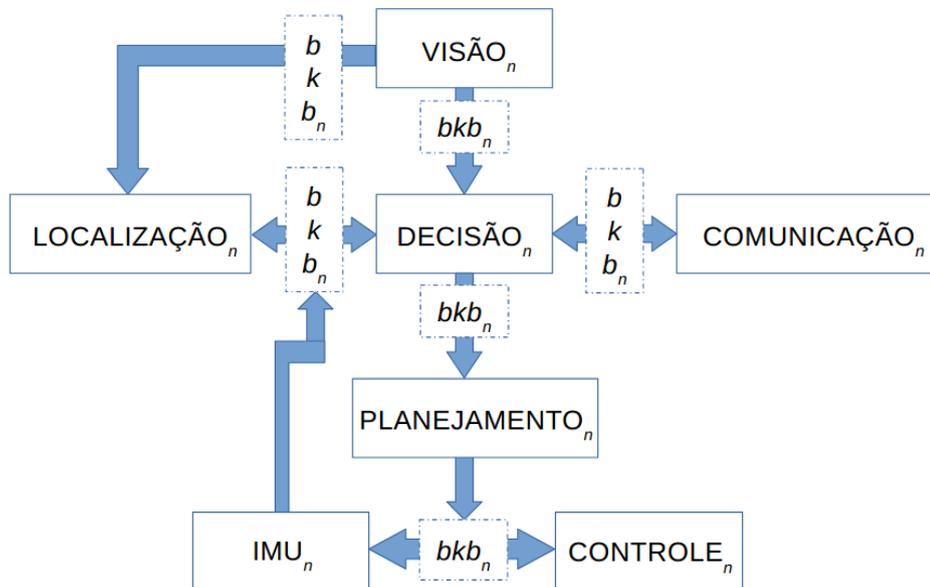
O simulador 2D utilizado para realizar os experimentos é o Simulador RoboFEI-HT (PERICO et al., 2016a), exibido na Figura 41. Esse simulador foi baseado na Arquitetura em Cruz com o intuito de permitir a portabilidade, o mais direta possível, dos algoritmos cognitivos desenvolvidos em simulação para o robô real. A ideia principal do simulador é reproduzir os

Figura 39 – Robôs humanoides do time RoboFEI-HT que foram utilizados durante a prova de conceito com suas respectivas cores



Fonte: Autor

Figura 40 – Arquitetura de software dos robôs humanoides: Arquitetura em Cruz



Fonte: Autor “adaptado de” PERICO et al., 2016

processos de Visão, Controle e IMU para n robôs. Por padrão, o simulador utiliza, como domínio, a Liga Humanoide da RoboCup *Soccer*, que tem, como cenário, o campo de futebol medindo **900 x 600 cm** (mais **70 cm** de cada lado) e como agentes móveis os robôs humanoides; porém, o domínio pode ser alterado sem influenciar diretamente no funcionamento dos robôs simulados, nem tão pouco na portabilidade dos algoritmos.

Os movimentos de um robô real são todos afetados por erros. Estes erros podem ser incluídos no robô virtual na forma de números aleatórios gerados a partir de distribuições normais. Dentre os diversos tipos de erros que podem ser inseridos no robô virtual, os mais significantes são: erro na velocidade, que provoca um andar mais rápido ou mais lento do que o previsto, erro de deslizamento lateral, que provoca movimentos laterais indesejados e o erro de rotação, que provoca rotação no robô enquanto ele se movimenta. Além dos erros de movimento, o simulador também permite a adição de erros na percepção do robô, tanto nos processos de Visão, quanto dos dados recebidos da IMU.

O Simulador RoboFEI-HT tem algumas vantagens quando comparado com os simuladores existentes, como o Webots, V-REP, Gazebo etc. Por exemplo, ele não requer um computador potente para ser executado, o que permite que a simulação seja, por exemplo, realizada no computador do robô real. Além disso, o simulador é gratuito, *open-source* e permite uma portabilidade bastante direta dos algoritmos cognitivos que nele forem desenvolvidos para robôs reais que fizerem uso da Arquitetura em Cruz.

Com o propósito de verificar a fidelidade do simulador em representar uma situação real, um experimento foi conduzido com o seguinte posicionamento inicial das entidades espaciais (Figura 43):

- a) Robô: em cima da linha da área de defesa;
- b) Bola: em cima da marca do penalty, também no campo de defesa.

Assim, um comportamento cognitivo, chamado ingênuo, que consiste em fazer o robô procurar a bola, andar até ela e chutar a bola, foi implementado no processo de decisão e testado diretamente no simulador (Figura 42a). O objetivo do robô durante o experimento era conseguir fazer um gol, porém o teste era, também, terminado se a bola fosse chutada para fora.

O mesmo algoritmo ingênuo de decisão foi transferido para o robô real (Figura 42b) para permitir a comparação do comportamento do robô em ambos os domínios: simulado e real. A avaliação foi realizada comparando-se a porcentagem de gols realizados e os caminhos percorridos por ambos robôs.

Os experimentos com o robô real foram executados **10** vezes, enquanto que os experimentos com o robô virtual foram executados **30** vezes. Assim, três gráficos de caminhos percorridos foram gerados e podem ser vistos na Figura 43. Além dos caminhos, a distância média percorrida pelos robôs nas três situações e a porcentagem de tentativas que resultaram em gols foi também contabilizada. O resultado pode ser visto na Tabela 5.

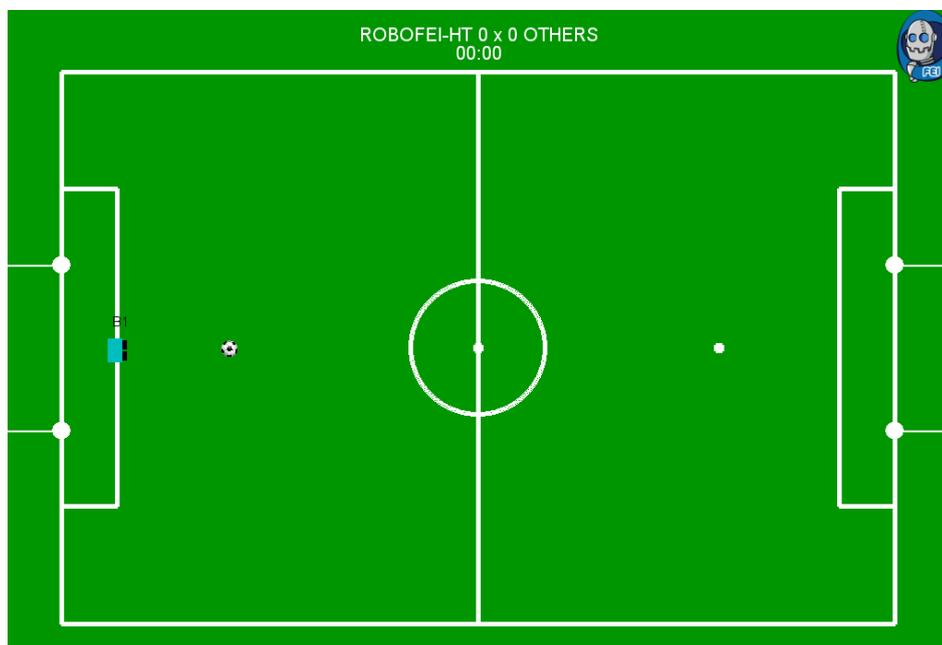
No caso do robô virtual, duas análises foram realizadas, uma sem e a outra com a inclusão de erros no robô (Figuras 43b e 43c). Os erros incluídos foram os de velocidade, de

Figura 41 – *RoboFEI-HT Simulator* exibindo três robôs de cada time.



Figura 42 – Experimento para medir a fidelidade do simulador com relação à uma situação real

(a) Posicionamento inicial para o robô virtual



(b) Posicionamento inicial para o robô real

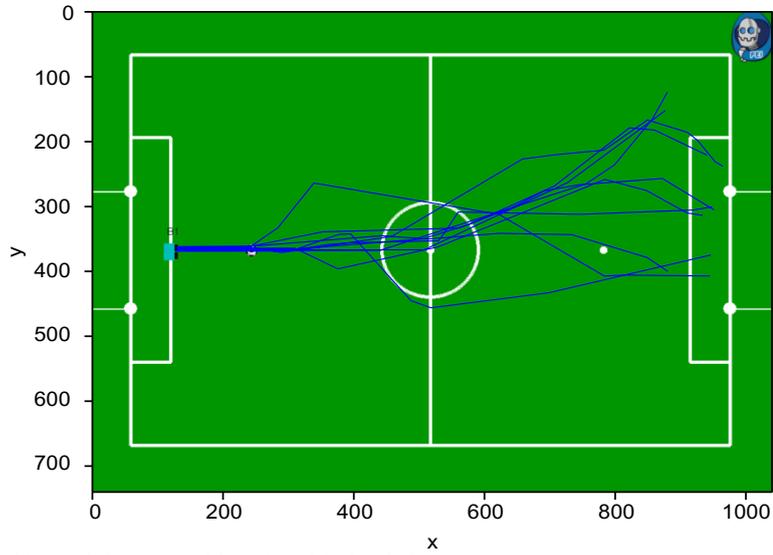


Fonte: Autor

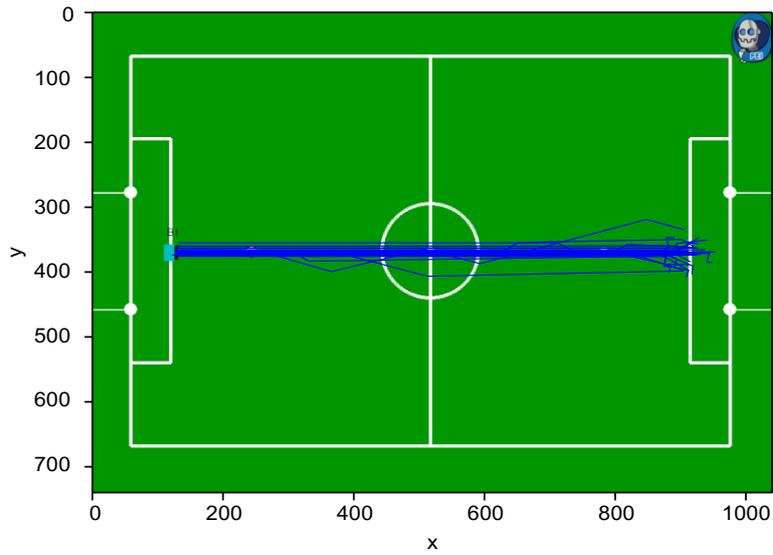
deslocamento lateral e de rotação e foram gerados aleatoriamente a partir de uma distribuição normal, considerando média zero e os seguintes desvios padrões: **0,2;0,2;0,1**. Além dos erros, um ruído Gaussiano também foi incluído nos dados provenientes da IMU. Esse ruído foi gerado com média zero e desvio padrão **0,01**. Os erros e o ruído são atualizados a cada iteração do simulador.

Figura 43 – Caminhos percorridos pelos robôs reais e virtuais (sem e com erros)

(a) Caminho percorrido pelo robô real.



(b) Caminho percorrido pelo robô simulado – sem erros.



(c) Caminho percorrido pelo robô simulado – com erros.

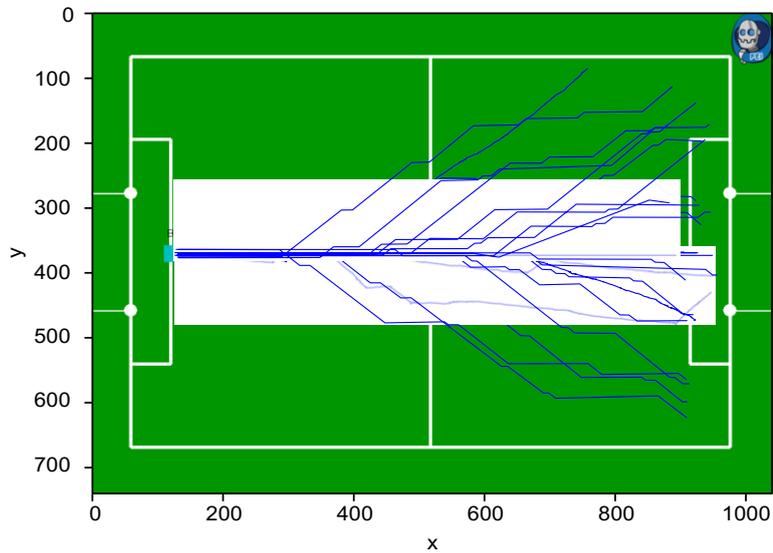


Tabela 5 – Resultados dos experimentos realizados para demonstrar a fidelidade do simulador RoboFEI-HT quando comparado ao robô humanoide real

	Distância média percorrida (cm)	Desvio (cm)	% das tentativas que resultaram em gols
Robô real	848,40	42,82	50
Robô simulado - sem erros	791,17	20,70	100
Robô simulado - com erros	879,77	101,40	53

Fonte: Autor

A avaliação com o teste t foi realizada entre as médias das distâncias percorridas nos experimentos realizados com o robô real e com o robô virtual configurado com erros. O propósito do teste t era verificar se a hipótese nula de que não existe diferença significativa entre as médias obtidas poderia ser rejeitada. Assim, o valor de t obtido foi igual a **1,37**; para um nível de confiança de **99,5%**, o valor de t crítico para o teste unicaudal foi igual a **1,69**, enquanto que t crítico para o teste bicaudal foi igual a **2,03**. Como o valor de t foi menor do que os dois valores de t crítico, pode-se concluir que não se pode rejeitar a hipótese nula.

5.2.2 Descrição dos Experimentos

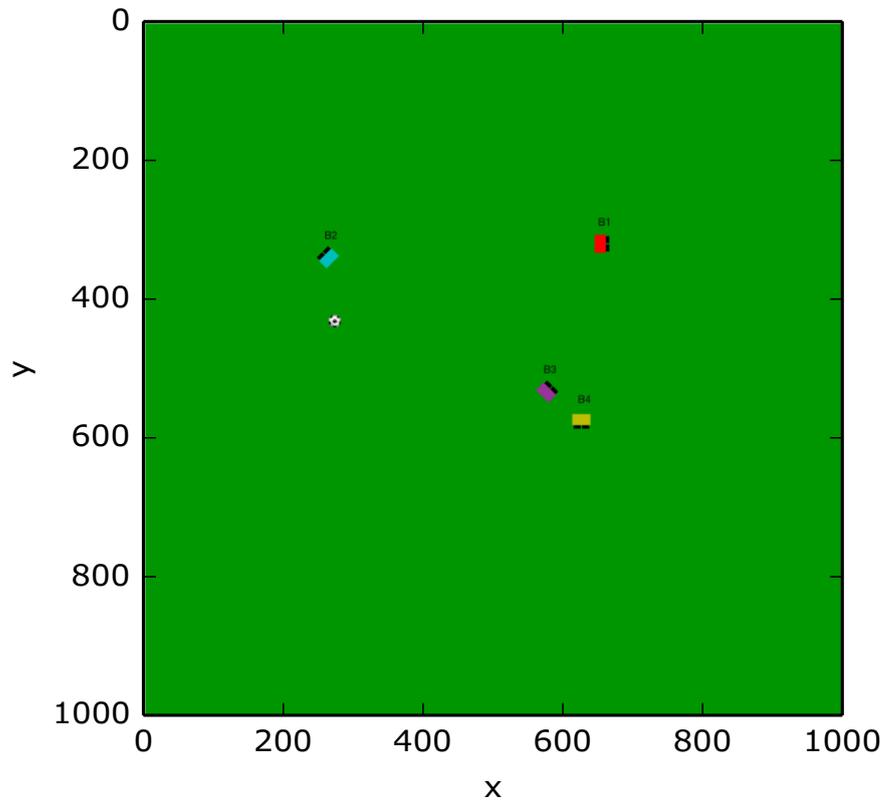
Os experimentos simulados foram conduzidos considerando robôs virtuais com os erros e o ruído exibidos na seção anterior. Além do ruído na IMU, um ruído Gaussiano com média zero e desvio padrão de **1^o** foi incluído no processo que simula a visão computacional de cada robô virtual para obtenção de direções.

Todos os experimentos foram realizados com cinco entidades espaciais: três robôs humanoides observadores, um robô humanoide guiado e um objetivo, representado pela bola; o ambiente no qual os robôs foram inseridos foi dimensionado com tamanho **1000 x 1000 cm**. Dado este domínio, quatro métodos diferentes de navegação guiada por meio de relações projetivas foram avaliados: dois métodos puramente qualitativos, o QPFGN_m e o NPFNGNQI_m^r. Nenhum destes métodos utiliza o modelo de velocidades do robô guiado, contudo, em caráter informativo, a velocidade de translação utilizada no robô durante os experimentos foi de **10,0 cm/s** e a velocidade angular foi de **0,31 rad/s**.

A Arquitetura em Cruz foi utilizada para permitir a implementação dos quatro modelos de navegação guiada nos robôs humanoides, por meio do seguinte padrão: inferências e raciocínios foram implementados no processo de decisão, enquanto que os algoritmos responsáveis pelo envio e recebimento de informações foram implementados no processo de comunicação. Toda a comunicação necessária entre os robôs foi realizada via rede, por meio do protocolo TCP/IP.

Assim, o experimento simulado consistiu em gerar um conjunto com **100** posições iniciais aleatórias de todas as entidades espaciais. Estas posições foram geradas respeitando-se **250 cm** de espaço livre com relação a cada borda do domínio. Em seguida, este mesmo conjunto de

Figura 44 – Exemplo do domínio utilizado nos experimentos



Fonte: Autor

posições iniciais foi submetido a cada um dos quatro modelos estudados, variando-se o valor de m para todos os casos e o valor de τ para o NPFQI_m^r . A navegação guiada completa, realizada em cada uma das **100** diferentes configurações espaciais iniciais, será chamada de episódio.

Para que os modelos funcionassem, os robôs precisavam ser capazes de reconhecer a identificação de cada um dos outros robôs. Para permitir tal habilidade, cada robô foi dotado de uma cor diferente, assim, os robôs vermelho, azul e roxo assumiram os papéis de robôs observadores, sendo o robô vermelho o coordenador, enquanto que o robô amarelo assumiu o papel de robô guiado. A Figura 44 exibe um exemplo de configuração inicial onde o código de cores dos robôs pode ser observado. Os resultados obtidos nos experimentos, assim como as respectivas análises, serão exibidos na seção 5.2.3. Todos os experimentos simulados foram executados em um Intel[®] NUC Core i5-4250U 1.30GHz, 8GB SDRAM e 120GB SSD com Sistema Operacional Linux, utilizando a distribuição Ubuntu 16.04. Todos os códigos foram implementados em Python, versão 2.7.

As três próximas seções (5.2.2.1, 5.2.2.2 e 5.2.2.3) exibirão as configurações mais específicas de cada modelo.

5.2.2.1 Configuração dos Experimentos Qualitativos Puros

Como o StarVars_m é o mecanismo utilizado para se realizar todas as inferências nos dois modelos puramente qualitativos, ele foi implementado no processo de decisão. Assim, durante os experimentos, o StarVars_m foi configurado com os seguintes parâmetros:

- a) $s = -\mathbf{1,0}$;
- b) $m = \mathbf{8}$ ou $\mathbf{16}$;
- c) Método para solucionar a LP: simplex (utilizando o pacote `cvxopt`⁵ do Python com o solucionador `glpk`⁶);
- d) Método de busca de um modelo válido por meio da LP: força bruta.

O primeiro método qualitativo utilizou o StarVars_m com somente uma instrução. Esse modelo foi avaliado como a primeira possibilidade para se conduzir o agente guiado até o objetivo. O modelo é a aplicação direta do StarVars_m pelo agente coordenador que, através da percepção recebida de todos os agentes observadores, consegue inferir em qual seção angular do agente guiado o objetivo se encontra. Após a inferência, o agente coordenador envia a instrução de ação para o agente guiado de acordo com a discretização das seções angulares em regiões espaciais linguísticas, conforme definição exibida na seção 4.2.2.1.

O segundo método qualitativo pode ser tratado como um método intermediário entre o modelo qualitativo descrito no parágrafo anterior e o QPFGN_m , pois toda a inferência é realizada com o StarVars_m puro, o que significa que partículas não são utilizadas, porém uma nova instrução é inferida toda vez que o agente guiado muda de região qualitativa q , onde $q \in \Omega$.

5.2.2.2 Configuração dos Experimentos com o QPFGN_m

No QPFGN_m , o StarVars_m é utilizado para se realizar a inferência sobre o modelo de mundo ψ . Após a obtenção de ψ , o agente coordenador realiza o raciocínio sobre qual instrução enviar para o agente guiado por meio do filtro de partículas qualitativo. Logo, o processo de decisão do QPFGN_m é composto pela implementação do StarVars_m e do filtro de partículas qualitativo.

A configuração do StarVars_m durante os experimentos foi a seguinte:

- a) $s = -\mathbf{1,0}$;
- b) $\varepsilon = -\mathbf{0,3}$;
- c) $m = \mathbf{8, 16}$ ou $\mathbf{32}$;
- d) Porcentagem das partículas que trocam de região mais de uma vez: $\xi = \mathbf{10\%}$;
- e) Método para solucionar a LP: simplex (utilizando o pacote `cvxopt` do Python com o solucionador `glpk`);
- f) Busca de um modelo válido por meio da LP: força bruta.

⁵<http://cvxopt.org/>

⁶<https://www.gnu.org/software/glpk/>

Por sua vez, a configuração do filtro de partículas foi definida conforme segue:

- a) Número de partículas para o agente guiado e para o objetivo: $N = 100$;
- b) $\sigma_{rotacao} = 5^\circ$;
- c) Distância percorrida por cada partícula em cada iteração: $dist = 0,01$;

O valor de $dist$ é escolhido utilizando s como referência, pois s serve como um fator de escala para o modelo de mundo interno criado pelo agente coordenador.

5.2.2.3 Configuração dos Experimentos com o NPFGNQI_m^r

O NPFGNQI_m^r utiliza somente a discretização espacial realizada pelo StarVars_m para definir a região na qual o objetivo está incluso. As inferências e raciocínios são realizados pelo filtro de partículas numérico. Dessa forma, o único parâmetro importante que é proveniente do StarVars_m é a granularidade m , que, durante os experimentos, foi igual a **8**, **16** ou **32**.

Já com relação ao filtro de partículas numérico, a configuração adotada foi a seguinte:

- a) Número de partículas para o agente guiado e para o objetivo: $N = 100$;
- b) distância de referência: $d = 1$;
- c) $\sigma_{rotacao} = 5^\circ$;
- d) $\sigma_{translacao} = 0,2$;
- e) $\sigma_{atualizacao} = 0,05$;
- f) Tamanho angular: $h = 5^\circ$;
- g) Taxa de atualização: $\tau = 3s$ ou $6s$;

Os valores de $\sigma_{translacao}$ e $\sigma_{atualizacao}$ são escolhidos tomando como base a distância de referência d , que serve como fator de escala para o modelo de mundo interno criado pelo agente coordenador.

5.2.3 Resultados dos Experimentos e Análises

O principal fator utilizado para se analisar a efetividade de cada um dos modelos estudados foi a porcentagem de sucesso atingido na tarefa de conduzir o agente guiado até a região na qual o objetivo estava inserido. A Tabela 6 exhibe o resultado alcançado após a execução dos mesmos **100** episódios para cada modelo. Duas situações diferentes foram consideradas como falhas:

- a) Número de instruções enviadas ao agente guiado maior ou igual a **50**;
- b) Encontro do agente guiado com alguma das bordas do domínio.

Um número alto de instruções significa que o agente guiado se perdeu e, provavelmente, ficou vagueando pelo domínio. Sabe-se que **50** é um número bem maior do que a quantidade necessária para se realizar a tarefa com sucesso devido aos resultados exibidos nas Tabelas 7 e 8. Por sua vez, quando o agente guiado chega em algum dos limites do domínio, a probabilidade dele estar se afastando do objetivo se torna muito grande, logo a falha era assumida.

Tabela 6 – Sucesso em alcançar a região do objetivo

	Sucesso em alcançar a região do objetivo			
	m	τ	Com orientação	Sem orientação
StarVars _{m} (instrução única)	8	-	51%	32%
	16	-	41%	20%
StarVars _{m}	8	-	93%	28%
	16	-	93%	9%
QPFGN _{m}	8	-	92%	80%
	16	-	94%	97%
	32	-	96%	93%
NPFGNQI _{m}	8	3	91%	93%
	8	6	93%	93%
	16	3	94%	92%
	16	6	92%	88%
	32	6	94%	80%

Fonte: Autor

A análise da Tabela 6 permite concluir que, com exceção do StarVars _{m} com instrução única, os outros três modelos conseguem cumprir a tarefa de conduzir um agente sem percepção de mundo em, no mínimo, **91%** dos casos, independentemente das granularidades de m ou das taxas de atualização τ estudadas, quando a orientação do agente guiado é conhecida.

No caso em que o agente guiado não tem orientação conhecida, os métodos puramente qualitativos tendem a falhar, pois o StarVars _{m} retorna o primeiro modelo de mundo válido, que, devido a completa falta de informação proveniente do agente guiado, vai ser sempre a primeira orientação avaliada, que, devido a implementação por meio de força bruta utilizada como método de busca neste trabalho, vai ser sempre **0**^o. Por sua vez, tanto o QPFGN _{m} , quanto o NPFGNQI _{m} ^{τ} continuam conseguindo altos índices de sucesso, por conta do modelo probabilístico utilizado, que permite, dentre outras coisas, a superação da falta de informação.

Depois, tomando como base apenas os casos bem sucedidos da Tabela 6, avaliou-se, para cada modelo, a quantidade de instruções necessárias para se alcançar a região do objetivo, assim como o tempo de processamento para se inferir cada instrução, o tamanho do caminho realizado pelo agente guiado e a distância em que o agente guiado parou com relação ao objetivo no fim de cada episódio.

Como os experimentos são realizados com diversas disposições iniciais distintas, todas as grandezas que envolvem distâncias foram normalizadas com base na distância Euclidiana entre o agente guiado e o objetivo no início de cada episódio. A distância Euclidiana foi utilizada, pois, sem considerar obstáculos ou colisões, ela é o caminho ótimo entre as duas entidades espaciais em questão. Assim, considerou-se que a distância Euclidiana em cada episódio representava **100%**, e as grandezas de caminho percorrido e distância do robô até o objetivo no fim de cada episódio são feitas com base nesta porcentagem.

Tabela 7 – Resultados dos experimentos bem sucedidos considerando que a orientação do agente guiado era conhecida

	m	τ	número de instruções	tempo de processamento de uma instrução (s)	caminho percorrido (%)	distância do objetivo (%)
StarVars _m (instrução única)	8	-	1	3,52 (2,04)	69,38(42,78)	49,68 (31,81)
	16	-	1	49,60 (76,70)	80,13(26,20)	33,10 (21,57)
StarVars _m (por região)	8	-	7,86(7,29)	3,21 (1,42)	122,98 (135,66)	40,91 (32,57)
	16	-	13,14 (8,93)	35,80 (43,18)	150,79 (89,27)	26,75 (21,16)
QPFGN _m	8	-	5,44(4,72)	6,55 (4,39)	124,11 (189,35)	48,67 (49,67)
	16	-	8,78(6,01)	6,09 (4,14)	113,18 (66,50)	27,98 (20,36)
	32	-	17,94 (10,81)	6,03 (3,68)	179,80 (224,15)	19,41 (20,83)
NPFGNQI _m	8	3	16,20 (10,70)	3,06 (0,05)	93,03(59,95)	37,53 (33,59)
	8	6	8,24(5,74)	3,06 (0,08)	102,40 (56,78)	29,34 (25,46)
	16	3	15,67 (10,64)	3,07 (0,05)	124,82 (65,37)	18,81 (20,94)
	16	6	10,68 (6,54)	3,06 (0,06)	122,25 (68,83)	18,29 (15,72)
	32	6	12,98 (9,11)	3,06 (0,06)	140,63 (87,10)	13,96 (17,77)

Fonte: Autor

As Tabelas 7 e 8 exibem, respectivamente, as médias e os desvios padrões (entre parênteses) dos valores obtidos para os experimentos com e sem a orientação do robô guiado. Como cada episódio tem fim quando o robô alcança a região do objetivo, e não quando o robô alcança a entidade espacial que representa o objetivo, as porcentagens dos caminhos percorridos deveriam ser menores que **100%** nos casos em que o robô conseguisse desempenhar um caminho semelhante ao caminho ótimo representado pela distância Euclidiana. Por esse motivo, e para permitir uma melhor análise de cada situação, a distância entre a entidade espacial objetivo e o robô guiado no fim dos episódios foi também considerada.

Alguns dados precisam ser cuidadosamente avaliados nas Tabelas 7 e 8, como, por exemplo, o tamanho dos caminhos percorridos pelo robô guiado por meio das inferências realizadas pelo StarVars_m com instrução única. O resultado, com ambas as granularidades analisadas, são os melhores obtidos, contudo, o StarVars_m com instrução única somente foi bem sucedido nos casos em que a região do objetivo encontrava-se exatamente no caminho que o robô estava executando após a primeira inferência, o que, em muitos os casos, era praticamente igual a distância Euclidiana. Na grande maioria dos casos inferidos com os outros modelos avaliados, os caminhos percorridos são maiores que **100%**, pois o robô guiado não consegue executar, recebendo apenas instruções de tão alto nível, um caminho muito próximo ao que seria o caminho ótimo.

Outra análise importante envolve o StarVars_m, que é atualizado a cada troca de região, na situação em que o agente guiado tem a orientação conhecida. A taxa de sucesso com o uso deste método foi de **93%** para ambas as granularidades. Contudo, quando o caminho percorrido é verificado, pode-se notar que, para a granularidade **8**, o tamanho é similar ao do caminho

percorrido pelo robô guiado com inferências provenientes do QPFGN₈. Por outro lado, para a granularidade $m = 16$ o StarVars _{m} atualizado em cada região apresenta o pior resultado dentre os três modelos que conseguiram manter taxa de sucesso acima de **91%**.

É possível notar também que, em diversos casos, a porcentagem média dos caminhos percorridos é menor para a granularidade menor ($m = 8$), porém a distância média entre o robô guiado e a entidade espacial objetivo no fim dos episódios sempre fica maior. A granularidade menor permite que o caminho percorrido seja menor, pois a região em que o objetivo se encontra é maior e, portanto, mais facilmente acessível; por outro lado, o robô termina o episódio mais longe da entidade espacial objetivo. Além disso, pode-se perceber que a distância média entre robô e objetivo no fim de cada episódio costuma ser menor no NPFNGNQI _{m} ^r. Isso se deve ao fato de que, no NPFNGNQI _{m} ^r o robô é guiado diretamente para a entidade objetivo e, devido à natureza numérica do modelo, as incertezas são menores; no StarVars _{m} que é atualizado em cada região, o robô também é guiado diretamente para a entidade objetivo, porém, esta entidade está posicionada no x e y que foi retornado como um modelo válido pelo sistema de inequações criado com os setores angulares de cada agente, o que torna maior a incerteza; como o QPFGN _{m} é realizado com base no StarVars _{m} a incerteza também é maior do que no NPFNGNQI _{m} ^r, o que torna maior a distância entre o robô e o objetivo no fim de cada episódio.

Ainda com relação aos dados das Tabelas 7 e 8, nota-se que o número de instruções é diretamente proporcional ao valor de m nos casos do StarVars _{m} atualizado a cada mudança de região e no QPFGN _{m} , porém, no caso do NPFNGNQI _{m} ^r a quantidade de instruções é inversamente proporcional a taxa de atualização τ . Com relação ao tempo médio de processamento, nota-se que, para o QPFGN _{m} e para o NPFNGNQI _{m} ^r ele é constante, sendo independente de m

Tabela 8 – Resultados dos experimentos bem sucedidos considerando que a orientação do agente guiado era desconhecida

	m	τ	número de instruções	tempo de processamento de uma instrução (s)	caminho percorrido (%)	distância do objetivo (%)
StarVars _{m} (instrução única)	8	-	1	6,61 (9,68)	80,11 (45,51)	54,75 (48,28)
	16	-	1	21,65 (19,92)	70,09 (38,03)	65,41 (24,97)
StarVars _{m} (por região)	8	-	6,92 (9,79)	4,04 (3,95)	115,35 (149,77)	58,00 (33,36)
	16	-	5,56 (4,16)	27,32 (23,83)	132,28 (163,19)	48,33 (38,37)
QPFGN _{m}	8	-	7,53 (6,33)	6,40 (4,00)	240,75 (300,50)	46,27 (31,77)
	16	-	10,17 (5,58)	6,04 (3,84)	172,39 (141,12)	28,27 (22,30)
	32	-	18,92 (9,11)	6,11 (3,69)	171,31 (116,22)	20,30 (20,55)
NPFNGNQI _{m}	8	3	12,58 (8,07)	3,07 (0,04)	120,32 (68,01)	32,51 (29,61)
	8	6	9,99 (8,34)	3,07 (0,06)	114,27 (71,64)	32,02 (27,05)
	16	3	16,40 (8,91)	3,07 (0,02)	149,12 (67,15)	18,10 (15,20)
	16	6	11,69 (5,99)	3,07 (0,05)	135,53 (44,88)	19,93 (19,04)
	32	6	17,10 (9,64)	3,07 (0,06)	176,16 (130,04)	12,84 (16,19)

Fonte: Autor

ou τ . Porém, nos dois métodos qualitativos puros o tempo de cada inferência tende a crescer conforme a granularidade m aumenta. O método de busca por um modelo de mundo válido no StarVars_m poderia ter sido implementado de forma mais eficiente, como, por exemplo, utilizando a busca hierárquica (LEE; RENZ; WOLTER, 2013), o que reduziria o tempo em cada inferência, contudo, de qualquer maneira, o tempo ainda seria proporcional à m ; o que não acontece com o QPFGN_m que, mesmo utilizando o StarVars_m como base, realiza as inferências utilizando as partículas, o que torna o tempo proporcional ao número de partículas N , assim como no NPFNGNQI_m^r .

A Análise de Variância (ANOVA) com teste de Welch (WELCH, 1951; LIU, 2015; MCDONALD, 2014) foi realizada sobre os tamanhos dos caminhos percorridos exibidos nas Tabelas 7 (agente guiado com orientação) e 8 (agente guiado sem orientação). Para os casos em que o agente guiado tinha sua orientação conhecida, os três modelos que obtiveram taxa de sucesso maior que **90%** na Tabela 6 foram avaliados: StarVars_m , QPFGN_m e NPFNGNQI_m^6 . As avaliações foram realizadas com as granularidades **8** e **16** e todas consideraram nível de significância igual a **0,05**. Assim, para $m = 8$, o resultado da comparação entre os métodos foi que eles não geraram caminhos com tamanhos significativamente diferentes; por outro lado, para $m = 16$, a análise de variância demonstrou que existia ao menos um modelo significativamente diferente dos outros para a variável tamanho do caminho. Portanto, após a utilização do pós-teste de Games-Howell (GAMES; HOWELL, 1976), identificou-se que a diferença se encontrava no StarVars que, então, pôde ser classificado como o pior dentre os três. Não houve diferença significativa entre o QPFGN_{16} e o NPFNGNQI_{16}^6 .

Já para o caso em que o agente não conhecia sua própria orientação, os dois modelos que obtiveram taxa de sucesso maior que **80%** na Tabela 6 foram analisados: QPFGN_m e NPFNGNQI_m^6 . Então, para ambas as granularidades, o QPFGN_{16} pôde ser considerado pior do que o NPFNGNQI_{16}^6 .

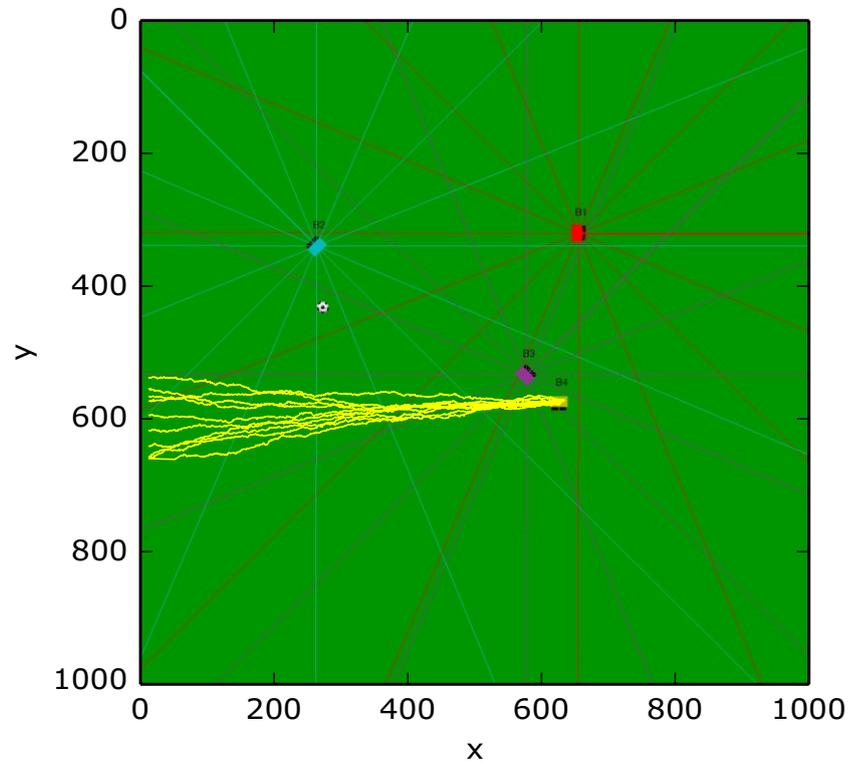
Além dos dados obtidos nas Tabelas 7 e 8, uma análise mais aprofundada foi realizada com a configuração espacial inicial exibida na Figura 44, com $m = 16$ e $\tau = 3$. Esta disposição das entidades foi escolhida, pois o agente guiado precisa, necessariamente, executar ações de virar, além das ações de andar para frente, para chegar a região do objetivo.

Assim, as Figuras de 45 a 48 exibem os caminhos percorridos pelo robô guiado, com e sem orientação, com cada um dos métodos avaliados, sendo que, para cada gráfico, **10** rodadas foram executadas, gerando, então, **10** possíveis caminhos. A Tabela 9 exhibe a porcentagem de sucesso referente a cada gráfico de caminho percorrido, assim como as médias e os desvios dos dados obtidos nos casos bem sucedidos de cada método.

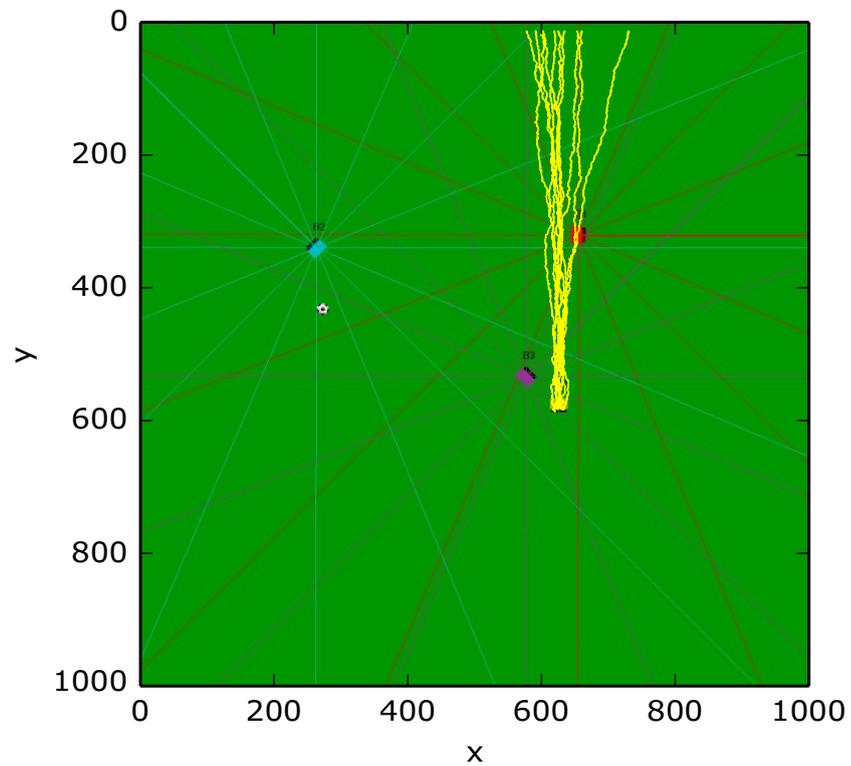
Com o intuito de se interpretar os resultados obtidos na Tabela 9, novamente a Análise de Variância (ANOVA) com teste de Welch foi realizada sobre os tamanhos dos caminhos obtidos em todos os métodos bem sucedidos (**100%** de sucesso em alcançar o objetivo), considerando os casos em que o agente guiado conhecia a sua orientação e os casos nos quais ele não conhecia. Para ambos os casos, os resultados das comparações entre os métodos foi que eles

Figura 45 – Exemplo de caminhos percorridos utilizando o StarVars₁₆ com instrução única para realizar a navegação guiada

(a) Com orientação



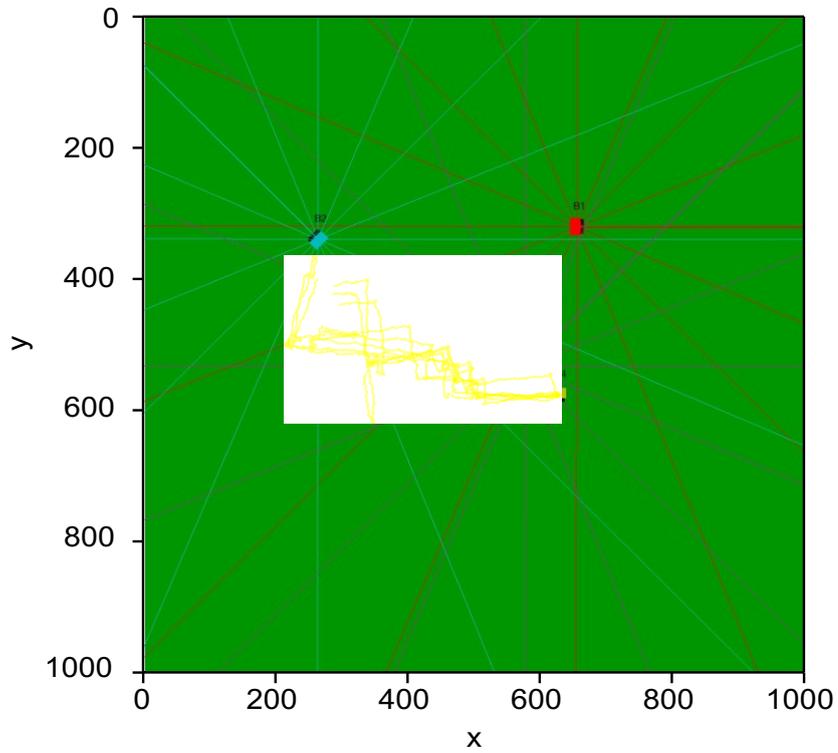
(b) Sem orientação



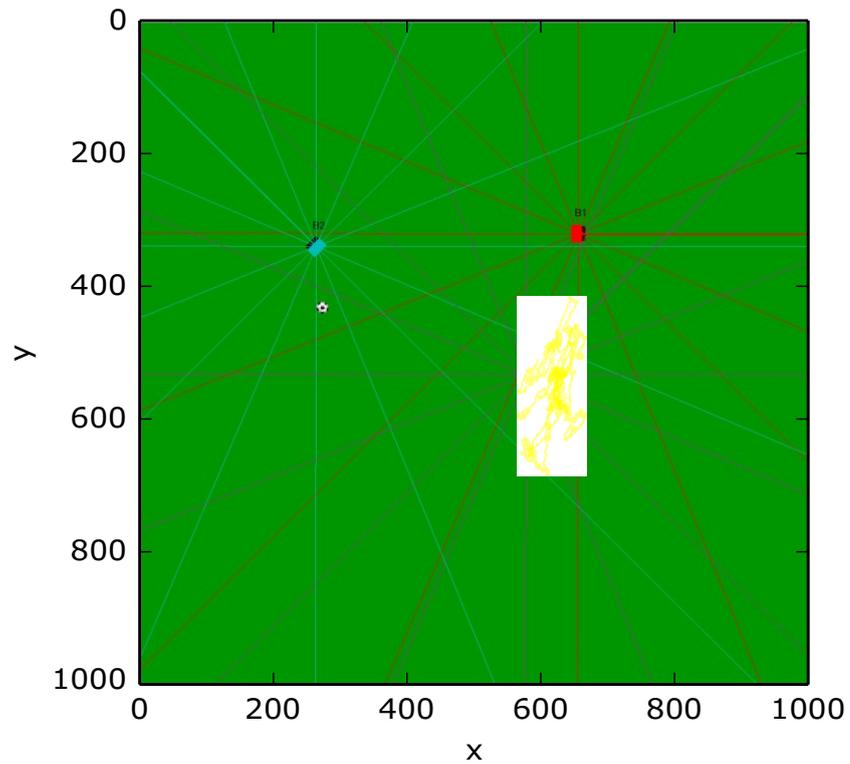
Fonte: Autor

Figura 46 – Exemplo de caminhos percorridos utilizando o StarVars₁₆ atualizado a cada troca de região para realizar a navegação guiada

(a) Com orientação



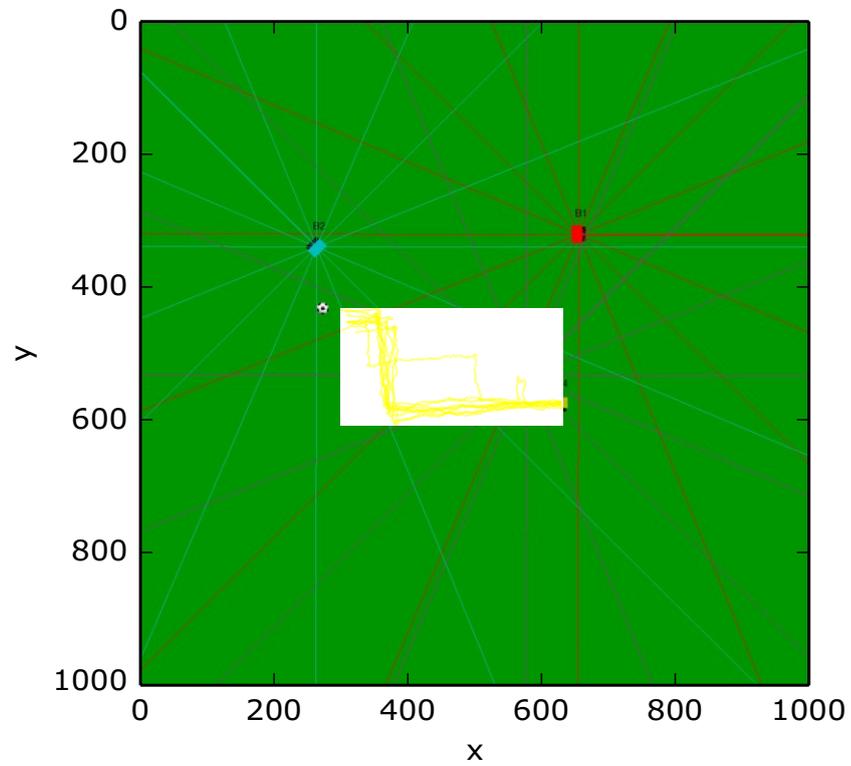
(b) Sem orientação



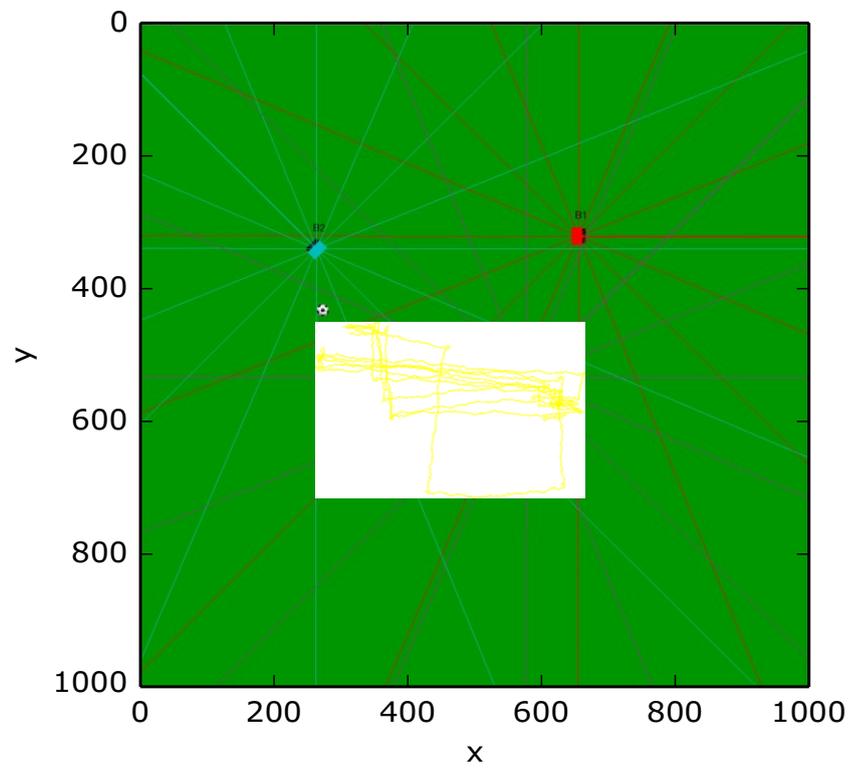
Fonte: Autor

Figura 47 – Exemplo de caminhos percorridos utilizando o QPFGN₁₆ para realizar a navegação guiada

(a) Com orientação



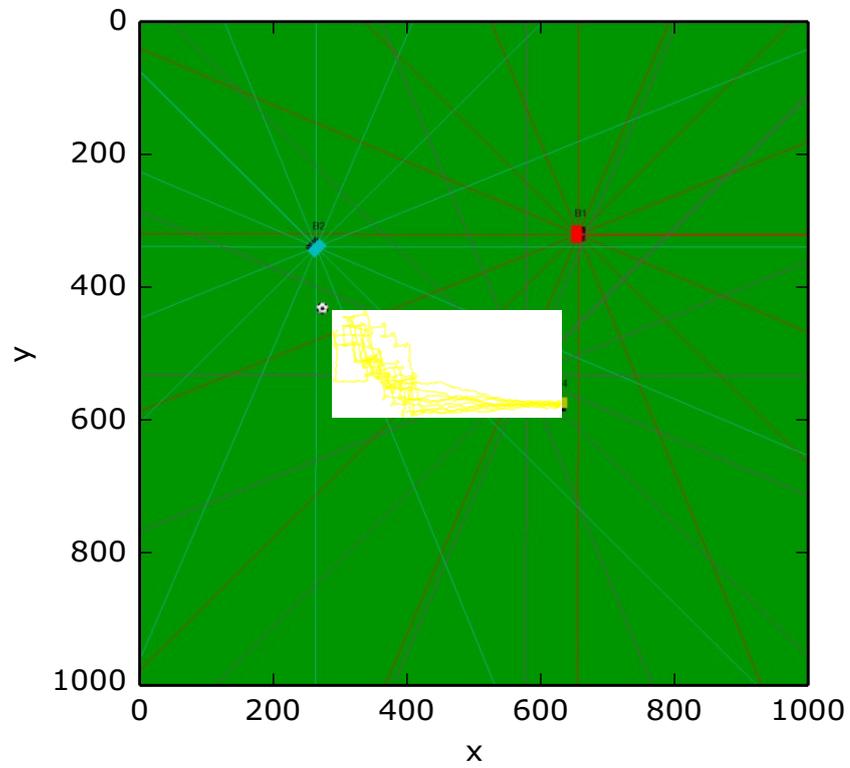
(b) Sem orientação



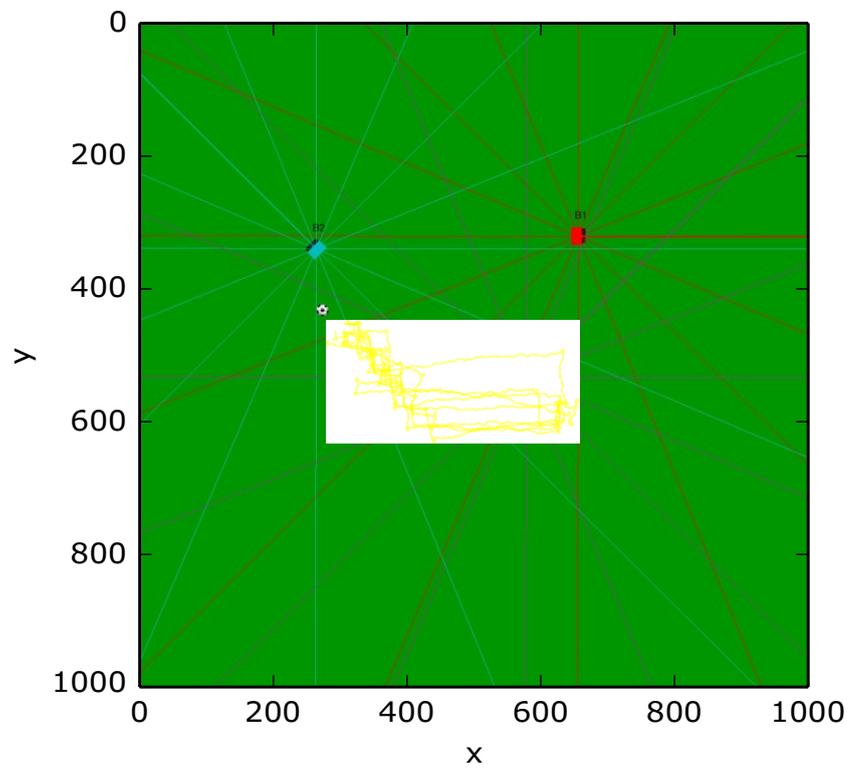
Fonte: Autor

Figura 48 – Exemplo de caminhos percorridos utilizando o NPF_{GNQI}³₁₆ para realizar a navegação guiada

(a) Com orientação



(b) Sem orientação



Fonte: Autor

Tabela 9 – Dados dos caminhos percorridos

	Orientação	Sucesso em alcançar o objetivo (%)	número de instruções	caminho percorrido (%)	distância do objetivo (%)
StarVars ₁₆ (instrução única)	com	0	1	-	-
	sem	0	1	-	-
StarVars ₁₆	com	100	27,00 (8,83)	155,25 (45,78)	9,74 (3,97)
	sem	0	-	-	-
QPFGN ₁₆	com	100	13,60 (1,43)	118,91 (8,00)	11,52 (2,48)
	sem	100	14,10 (2,56)	128,81 (27,33)	15,02 (3,26)
NPFGNQI ₁₆ ³	com	100	17,40 (1,96)	119,32 (10,09)	10,90 (1,07)
	sem	100	19,20 (3,60)	132,73 (20,72)	12,04 (1,42)

Fonte: Autor

não geraram caminhos com tamanhos significativamente diferentes para a configuração espacial inicial estudada, apesar da média e do desvio padrão obtidos pelo StarVars₁₆, no caso em que o agente conhecia sua própria orientação, terem sido maiores do que os valores obtidos pelos outros dois modelos.

5.3 PROVA DE CONCEITO COM ROBÔS REAIS

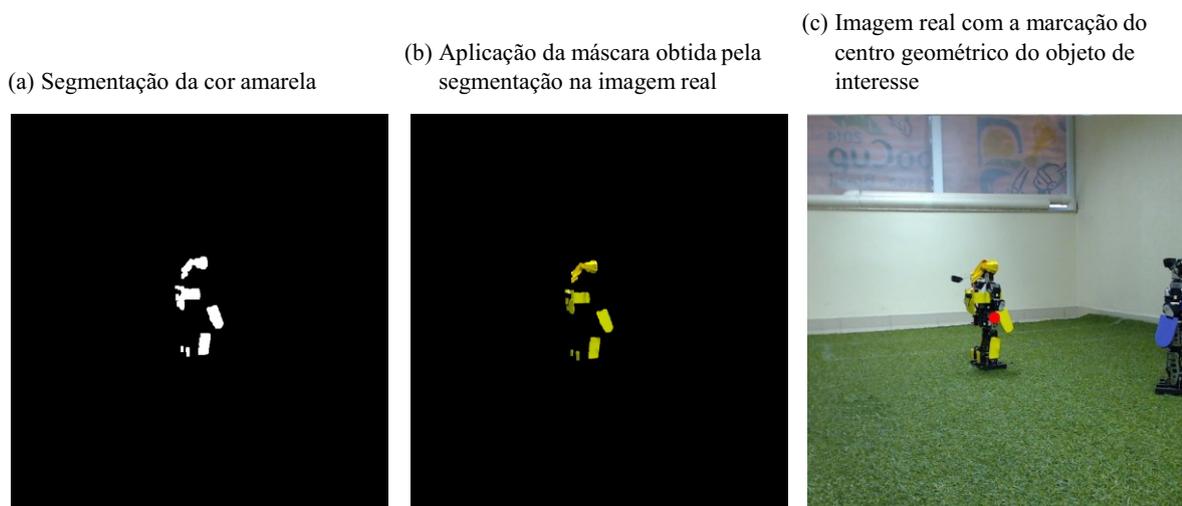
Uma vez que os diversos experimentos da seção anterior foram executados, provas de conceitos com os métodos QPFGN_m e NPFGNQI_m^r foram realizadas com robôs humanoides reais. As provas de conceitos seguiram as mesmas configurações dos experimentos simulados, contudo, para o QPFGN_m, *m* foi mantido fixo e igual a **8** e, para o NPFGNQI_m^r, *m* foi mantido fixo e igual a **16** e τ foi mantido igual a **6s**. Durante as provas de conceitos, o robô guiado sempre teve sua orientação conhecida, sendo obtida por meio da IMU.

Assim como nos experimentos simulados, os robôs, e também a bola, foram identificados com cores diferentes para permitir a identificação única de cada um. A Figura 39 exibe uma imagem dos quatro robôs humanoides e da bola com suas respectivas cores. O padrão de cores seguiu o modelo adotado no experimento simulado: robôs vermelho, azul e roxo com papéis de agentes observadores, sendo o robô vermelho o coordenador, e o robô amarelo com papel de agente guiado.

Dessa forma, o reconhecimento dos objetos foi realizado somente por meio de segmentação de cores. O processo de visão computacional foi implementado com o uso da biblioteca OpenCV⁷, em que a função `moments` foi utilizada para identificar o centro geométrico da imagem segmentada. A visão do robô humanoide é implementada para sempre centralizar a câmera no centro geométrico do objeto de interesse. Então, depois que a centralização é realizada, os robôs reais obtêm as informações de direção por meio da leitura da posição do servomotor que

⁷<https://opencv.org/>

Figura 49 – Processo de visão computacional utilizado para identificação dos objetos de interesse



Fonte: Autor

controla o movimento de pan da câmera. A Figura 49 demonstra o processo de visão utilizado durante a prova de conceito.

A configuração espacial inicial adotada durante a prova de conceito pode ser vista na Figura 50 e cada modelo de navegação foi executado **3** vezes. As Figuras de 51 até 53 exibem a prova de conceito com o QPFGN₈, sendo que a Figura 51 exibe o movimento do robô humanoide real, enquanto que a Figura 52 demonstra o raciocínio executado pelo robô coordenador para conduzir o robô guiado e a Figura 53 exibe os três caminhos realizados pelo robô até a região do objetivo.

As Figuras de 54 até 56 exibem a resolução do mesmo problema, porém com o NPFNGNQI₁₆⁶. O movimento do robô humanoide real pode ser visto na Figura 54. O raciocínio executado pelo robô coordenador é demonstrado na Figura 55, enquanto que os **3** caminhos percorridos pelo robô guiado são exibidos na Figura 56.

5.4 COMENTÁRIOS

As seções 5.2.3 e 5.3 apresentaram, respectivamente, os resultados de diversos experimentos realizados no simulador RoboFEI-HT e de duas provas de conceito, uma para cada modelo proposto, em robôs humanoides reais.

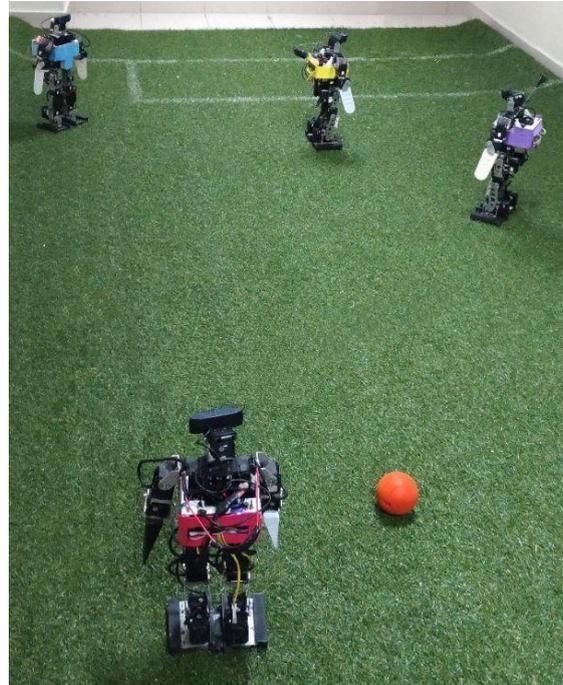
Os resultados encontrados demonstraram que, tanto o robô guiado por meio das inferências obtidas pelo QPFGN_m, quanto pelo NPFNGNQI_m^r, conseguiu chegar na região do objetivo. Os dois modelos foram bem sucedidos nos casos em que o robô conhecia, e nos casos em que o robô desconhecia, a sua própria orientação, o que não aconteceu com os modelos qualitativos puros que utilizaram somente o StarVars_m para raciocinar espacialmente e inferir a instrução correta a ser enviada para o robô guiado.

Figura 50 – Configuração espacial inicial utilizada na prova de conceito

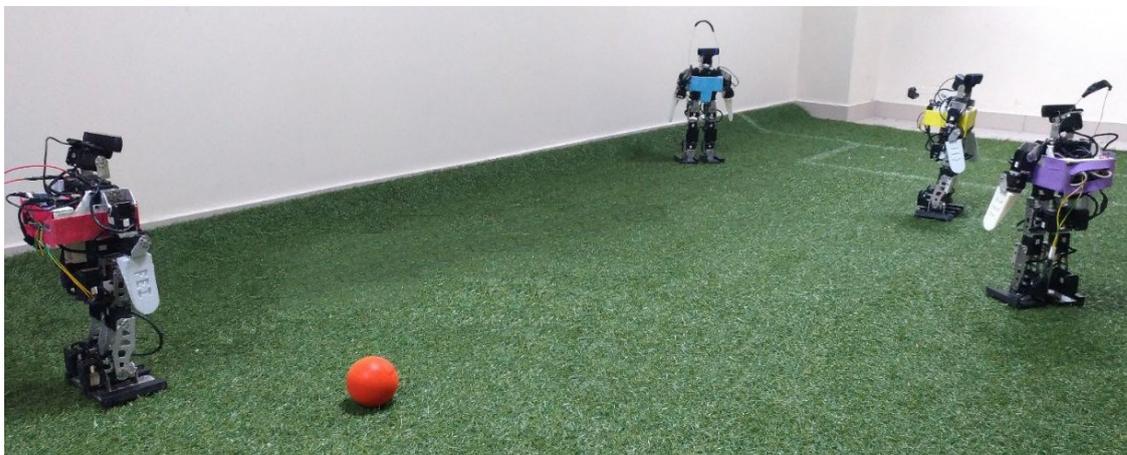
(a) Representação no simulador



(b) Imagem real



(c) Imagem real em outra perspectiva



Fonte: Autor

Uma análise mais superficial dos experimentos permite a conclusão de que o $NPFQI_m^r$ é mais eficiente, pois apresentou os menores tamanhos de caminho e foi o modelo que conseguiu fazer com que o robô chegasse mais perto da entidade espacial objetivo, mesmo considerando que os tamanhos das regiões dos objetivos foram os mesmos para todos os modelos, de acordo com a granularidade m . Contudo, pode-se notar que τ tem uma função parecida com a função que seria desempenhada pelo modelo de movimento de translação, logo é um valor que precisa ser regulado para que não fique pequeno demais, a ponto do agente não conseguir sair do lugar antes de uma nova atualização, e nem grande demais, a ponto de permitir que o agente se afaste demais do objetivo a cada movimento. Isso torna o $NPFQI_m^r$ um

Figura 51 – Movimento do robô humanoide real sendo guiado pelo QPFGN₈

(a) Posição inicial



(b) Após virar para direita e andar para frente



(c) Após andar para frente



(d) Após andar para frente



(e) Após andar para frente



(f) Região do objetivo alcançada após andar para frente



Fonte: Autor

modelo não tão abstrato quanto o QPFGN_m, que, não precisa de ajustes para funcionar, basta a escolha da granularidade desejada.

Com relação às provas de conceito, os dois modelos testados se mostraram eficazes; porém, diversos problemas foram encontrados na aplicação dos modelos com robôs reais. Os principais foram atrasos na rede e a detecção dos agentes e do objetivo por meio da visão computacional utilizando apenas segmentação de cores.

Figura 52 – Prova de conceito: predição e atualização das partículas no QPFGN₈

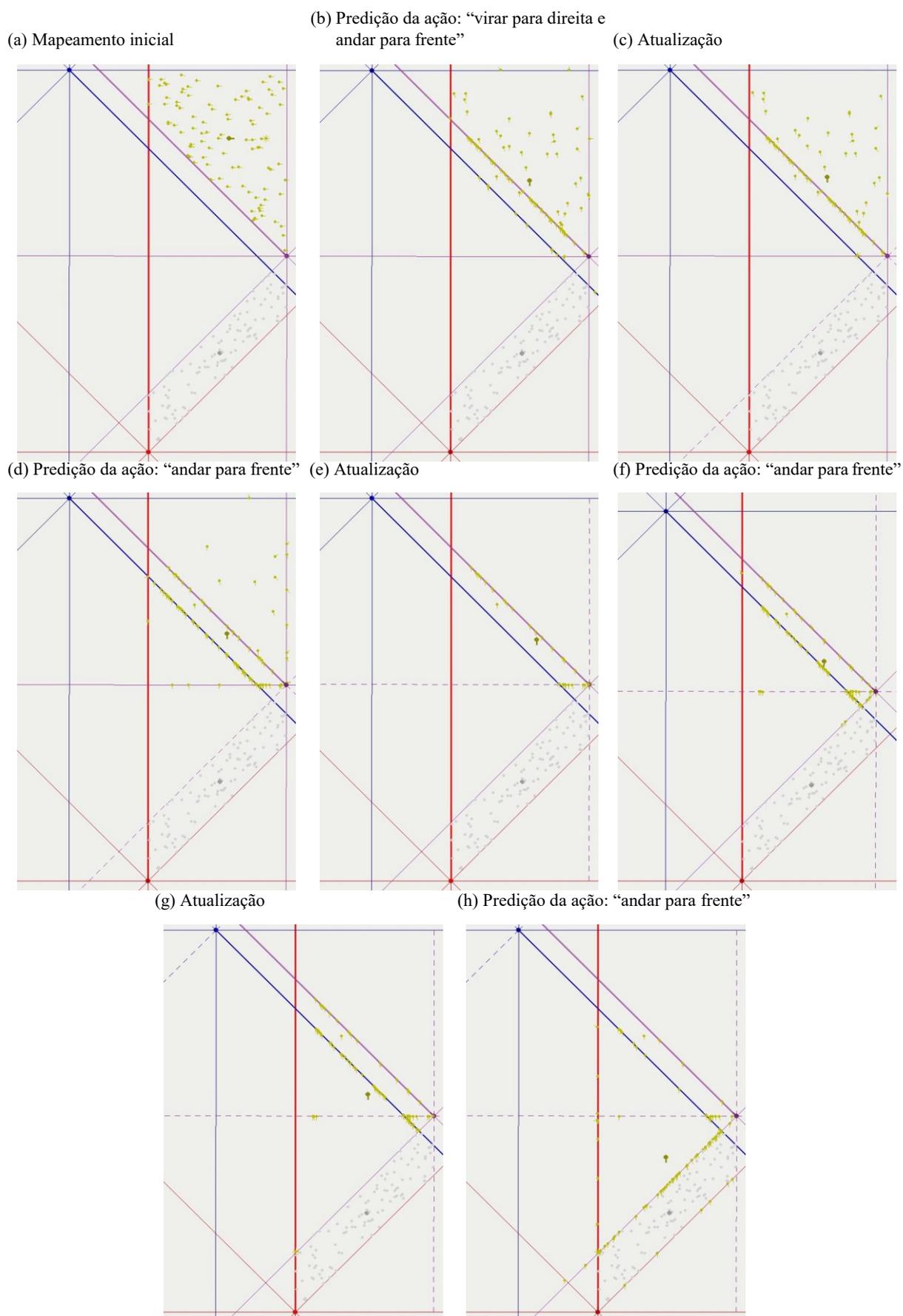
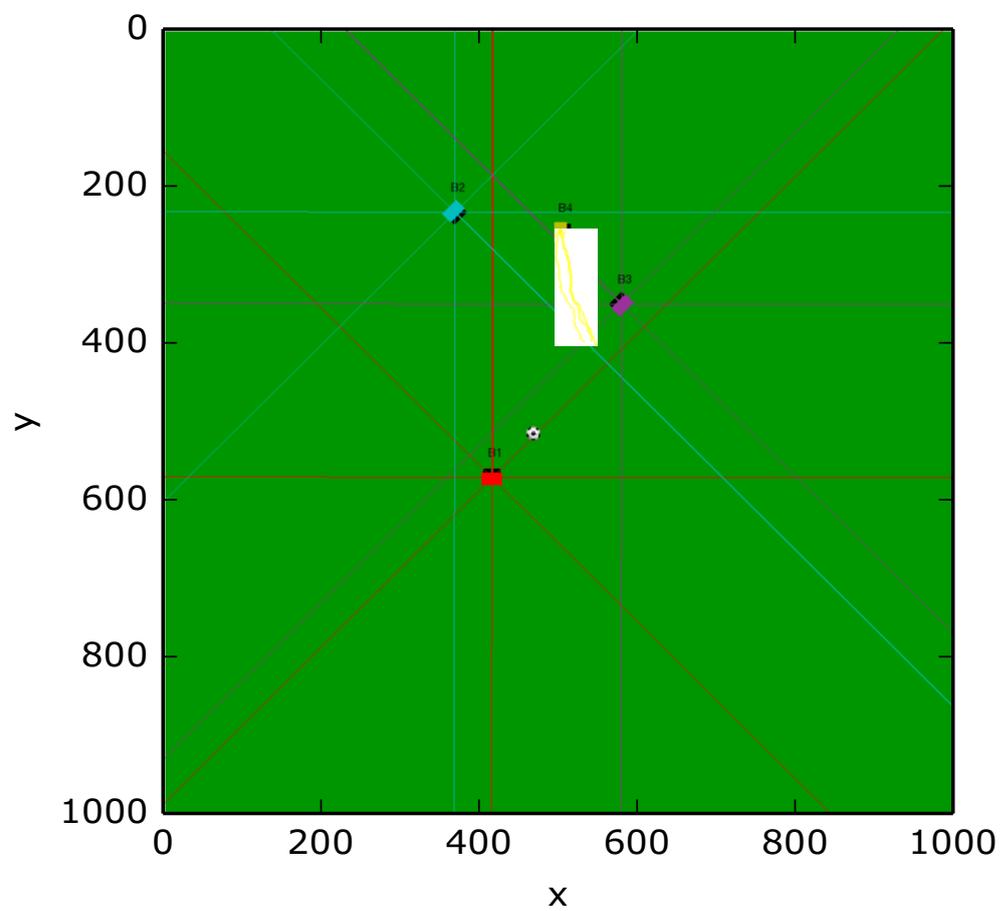


Figura 53 – Caminhos percorridos pelo robô guiado com o QPFGN₈

Fonte: Autor

Figura 54 – Prova de conceito: movimento do robô humanoide real sendo guiado pelo NPFGNQI₁₆⁶

(a) Após virar para direita e andar para frente



(b) Após andar para frente



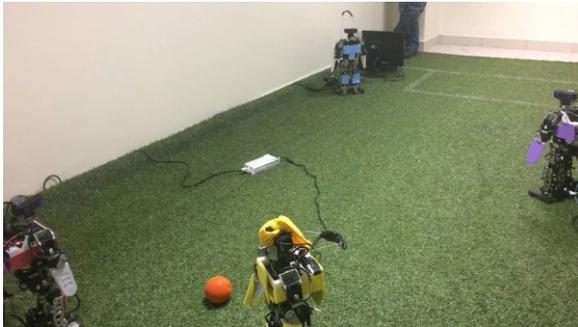
(c) Após andar para frente



(d) Após andar para frente



(e) Após virar para direita e andar para frente



(f) Após andar para frente



(g) Após virar para trás e andar para frente

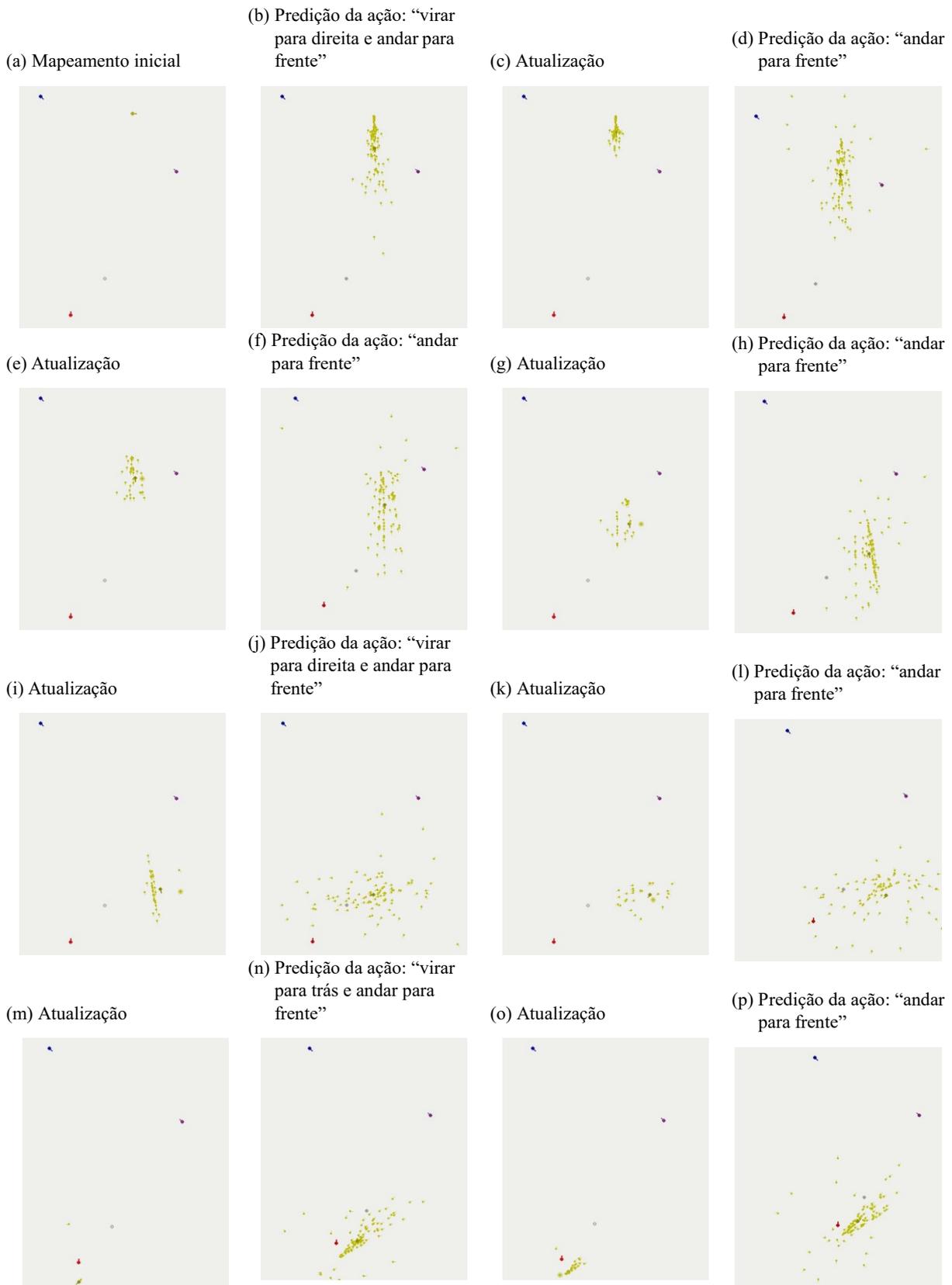


(h) Região do objetivo alcançada após andar para frente

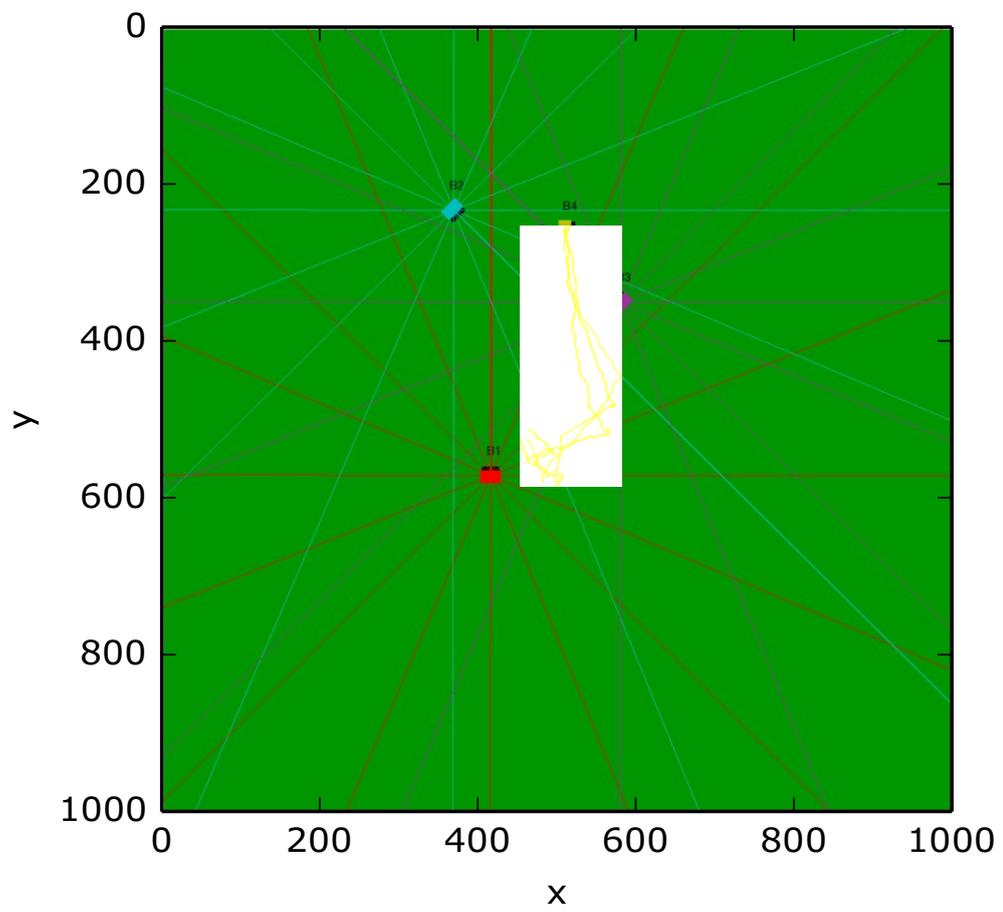


Fonte: Autor

Figura 55 – Prova de conceito: predição e atualização das partículas no NPFGNQI₁₆⁶



Fonte: Autor

Figura 56 – Caminhos percorridos pelo robô guiado com o NPFGNQI₁₆⁶

Fonte: Autor

6 CONCLUSÃO E TRABALHOS FUTUROS

O problema de navegação em agentes que não podem perceber o mundo e não tem modelo de movimento completamente definido foi, até agora, pouco estudado; contudo este tipo de navegação pode ser viabilizada por meio da navegação guiada, na qual outros agentes autônomos podem, por meio da união de suas diferentes perspectivas, inferir a localização e as instruções que o agente sem percepção precisa receber para navegar de um ponto qualquer até um objetivo determinado.

A navegação guiada pode auxiliar a resolução de uma série de problemas, como casos de resgate de vítimas em ambientes de desastre ou em outro domínio qualquer com múltiplos robôs que precisam cooperar para um objetivo comum, como o caso da liga de futebol de robôs humanoides da RoboCup.

Dessa forma, este trabalho apresentou duas propostas híbridas, qualitativo-probabilísticas, de navegação guiada de um agente autônomo por meio de instruções geradas com base em preposições projetivas obtidas por múltiplas perspectivas espaciais.

A primeira proposta foi chamada de Filtro de Partículas Qualitativo para Navegação Guiada – *Qualitative Particle Filter for Guided Navigation* (QPFGN_m), pois utiliza dados qualitativos em um filtro de partículas. No QPFGN_m, a predição do movimento de translação e a atualização dos pesos das partículas são realizadas com base nas regiões qualitativas geradas por meio da junção dos setores angulares definidos no StarVars_m. A regra de atualização é feita com base na granularidade m .

O segundo modelo proposto foi chamado de Filtro de Partículas Numérico para Navegação Guiada com Instruções Qualitativas – *Numerical Particle Filter for Guided Navigation with Qualitative Instructions* (NPFGNQI_m^τ), pois utiliza somente dados numéricos de direção e de orientação em um filtro de partículas. No NPFGNQI_m^τ a predição do movimento de translação é feita com dados gerados de maneira aleatória e a atualização dos pesos das partículas é realizada com base em uma distribuição Gaussiana. A regra de atualização é feita com base na taxa de atualização τ . Em ambos os modelos propostos, a granularidade m define o tamanho da região na qual o objetivo está incluso.

Dados os dois métodos propostos, experimentos foram conduzidos em ambiente simulado para comparar os modelos entre si e, também, para compará-los com dois métodos de navegação qualitativos que utilizaram somente o StarVars_m como mecanismo de raciocínio e inferência. Dentro destes quatro modelos avaliados, variações com os valores de m e τ foram estudadas, além do fato do agente guiado conhecer ou não sua própria orientação. Em adição aos dados obtidos pelos experimentos simulados, duas provas de conceito em robôs humanoides reais foram realizadas, uma com cada método proposto.

Os resultados exibidos na seção 5 mostram que as duas propostas apresentadas neste trabalho trazem contribuições relevantes para a área de navegação, pois uma nova classe de

problemas passa a poder ser resolvida com base no Algoritmo 5: a navegação guiada de agentes autônomos móveis por meio de instruções baseadas em preposições projetivas.

As contribuições dos dois modelos são notáveis, principalmente quando a orientação do agente guiado não é conhecida, pois, tanto as inferências obtidas pelo QPFGN_m, quanto pelo NPFNGQI_m^c foram suficientes para conduzir o agente guiado até a região do objetivo com somente o uso de cinco instruções: “movimentar-se para frente”, “virar para direita e movimentar-se para frente”, “virar para esquerda e movimentar-se para frente”, “virar para trás e movimentar-se para frente” e “ficar parado”.

Finalmente, como contribuição para a sociedade científica, as pesquisas e os estudos realizados durante este doutorado resultaram, até o presente momento, em uma publicação de artigo em periódico, três capítulos de livros, seis artigos em congressos internacionais e dois em congressos nacionais. A lista com as publicações pode ser vista no Anexo A. Além das publicações, contribuições foram dadas por meio do desenvolvimento do projeto de futebol de robôs humanoides, que resultaram em alguns prêmios e títulos, conforme exibido no Anexo B.

6.1 TRABALHOS FUTUROS

Dados os resultados obtidos, a extensão deste trabalho se mostra relevante e de grande interesse. Neste sentido, são sugeridos os seguintes trabalhos futuros:

- a) Incluir obstáculos no problema de navegação guiada;
- b) Incluir probabilidade na detecção dos setores angulares para o caso do QPFGN_m;
- c) Implementar um novo modelo que combina o critério de parada utilizado no QPFGN_m, ou seja, a troca de região, com a atualização numérica das partículas, conforme realizado no NPFNGQI_m^c;
- d) Integrar um dos modelos tradicionais de planejamento de trajetória, como campos potenciais, com ambos os métodos propostos para deixar o caminho percorrido mais próximo ao caminho ótimo. Para o QPFGN_m, o método de campos potenciais poderia ser utilizado para atrair, ou repelir, o agente guiado para determinadas regiões.
- e) Realizar mais experimentos simulados alterando o número de agentes autônomos para verificar a influência de mais ou menos agentes no raciocínio.
- f) Realizar experimentos com robôs reais de forma a obter uma quantidade maior de dados, com o intuito de se realizar estudos estatísticos mais completos. Para isso, a detecção das entidades espaciais pode ser realizada por meio de uma Rede Neural Profunda – *Deep Neural Network* (DNN) e, após a detecção inicial, a identificação única pode ser feita por meio de cores ou de QR Codes.

REFERÊNCIAS

- ASIMOV, I. I. **Robot**. [S.l.]: Bantam Books, 1950. (Robot series).
- BAILEY, T.; DURRANT-WHYTE, H. F. Simultaneous localization and mapping (SLAM): part II. **IEEE Robotics Automation Magazine**, v. 13, n. 3, p. 108–117, set. 2006.
- BERTSEKAS, Dimitri P.; TSITSIKLIS, John N. **Introduction to probability**. 2nd. [S.l.]: Athena Scientific, jul. 2008.
- BISCHOFF, R.; HUGGENBERGER, U.; PRASSLER, E. KUKA youBot - a mobile manipulator for research and education. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION. **Proceedings...** [S.l.: s.n.], mai. 2011. p. 1–4.
- CANNY, John. A Voronoi method for the piano-movers problem. In: IEEE INTL. CONF. ON ROBOTICS AND AUTOMATION (ICRA). **Proceedings...** [S.l.: s.n.], 1985. v. 2, p. 530–535.
- CANNY, John; DONALD, Bruce. Simplified Voronoi diagrams. **Discrete & Computational Geometry**, v. 3, n. 3, p. 219–236, set. 1988.
- CASSANDRA, Anthony R.; KAEHLING, L.; LITTMAN, Michael L. **Acting Optimally in Partially Observable Stochastic Domains**. Providence, RI, USA, 1994.
- CHEN, Juan et al. A survey of qualitative spatial representations. **The Knowledge Engineering Review**, v. 30, p. 106–136, 1 out. 2015.
- CHOUBEY, N; GUPTA, MBK. Analysis of working of Dijkstra and A* to obtain optimal path. **International Journal of Computer Science and Management Research**, v. 2, p. 1898–1904, 2013.
- COHN, A. G.; RENZ, Jochen. Qualitative Spatial Reasoning. In: HARMELEN, Frank van; LIFSCHITZ, Vladimir; PORTER, Bruce (Ed.). **Handbook of Knowledge Representation**. [S.l.]: Elsevier, 2007.
- COLOMBO, Desirée et al. Egocentric and allocentric spatial reference frames in aging: A systematic review. **Neuroscience & Biobehavioral Reviews**, v. 80, Supplement C, p. 605–621, 2017.
- COLTIN, Brian; VELOSO, Manuela. Multi-observation sensor resetting localization with ambiguous landmarks. **Autonomous Robots**, Springer, v. 35, n. 2-3, p. 221–237, 2013.
- DANTZIG, George B. A History of Scientific Computing. In: NASH, Stephen G. (Ed.). New York, NY, USA: ACM, 1990. Origins of the Simplex Method, p. 141–151.

DARPA ROBOTICS CHALLENGE. **Atlas Robot – IHMC Robotics**. [S.l.: s.n.], 2013.
Disponível em: <<http://archive.darpa.mil/roboticschallengetrialsarchive/node/61/index.html>>.
Acesso em: 28.mai.2015.

DEGROOT, M.H.; SCHERVISH, M.J. **Probability and Statistics**. 4. ed. [S.l.]: Pearson, 2011.

DELLAERT, F. et al. Monte Carlo Localization for Mobile Robots. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA99). **Proceedings...** Detroit, Michigan, USA: [s.n.], mai. 1999.

DIJKSTRA, Edsger W. A note on two problems in connexion with graphs. **Numerische mathematik**, Springer, v. 1, n. 1, p. 269–271, 1959.

DORR, Christopher H.; MORATZ, Reinhard. Qualitative shape representation based on the qualitative relative direction and distance calculus eOPRAM. **CoRR**, abs/1412.6649, 2014.

DURRANT-WHYTE, H. F.; BAILEY, T. Simultaneous localization and mapping: part I. **IEEE Robotics Automation Magazine**, v. 13, n. 2, p. 99–110, jun. 2006.

DYLLA, F. Qualitative Spatial Reasoning for Navigating Agents: Behavior Formalization with Qualitative Representations. In: **Behaviour Monitoring and Interpretation – BMI**. Edição: B. Gottfried e H. Aghajan. [S.l.: s.n.], 2009. v. 3. (Ambient Intelligence and Smart Environments).

DYLLA, F.; KREUTZMANN, Arne; WOLTER, D. A Qualitative Representation of Social Conventions for Application in Robotics. In: AAAI SPRING SYMPOSIUM - QUALITATIVE REPRESENTATIONS FOR ROBOTS. **Proceedings...** [S.l.: s.n.], 2014.

DYLLA, F. et al. Approaching a Formal Soccer Theory from Behaviour Specifications in Robotic Soccer. In: DABNICKI, Peter; BACA, Arnold (Ed.). **Computers in Sports**. [S.l.]: WIT Press; London / Southampton, UK; 2008. p. 161–185.

DYLLA, F. et al. SailAway: Formalizing Navigation Rules. In: AISB'07 ARTIFICIAL AND AMBIENT INTELLIGENCE SYMPOSIUM ON SPATIAL REASONING AND COMMUNICATION. **Proceedings...** [S.l.: s.n.], 2007.

EGENHOFER, M. J.; FRANZOSA, R. Point-set topological spatial relations. **International Journal of Geographical Information Systems**, Taylor & Francis, p. 161–174, 1991.

FOX, D.; BURGARD, W.; THRUN, S. Active markov localization for mobile robots. **Robotics and Autonomous Systems**, Elsevier, v. 25, n. 3-4, p. 195–207, 1998.

- _____. Markov Localization for Mobile Robots in Dynamic Environments. **Journal of Artificial Intelligence Research**, Morgan Kaufmann Publishers, v. 11, p. 391–427, jul. 1999.
- FOX, D. et al. Monte Carlo Localization: Efficient Position Estimation for Mobile Robots. In: AAAI NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE. **Proceedings...** [S.l.: s.n.], 1999. p. 343–349.
- FRANK, Andrew. Qualitative Spatial Reasoning about Cardinal Directions. In: AMERICAN CONGRESS ON SURVEYING AND MAPPING (ACSM-ASPRS). **Proceedings...** [S.l.: s.n.], 1991. p. 148–167.
- FREKSA, Christian. Using orientation information for qualitative spatial reasoning. In: FRANK, Andrew U.; CAMPARI, Irene; FORMENTINI, Ubaldo (Ed.). **Theories and methods of spatio-temporal reasoning in geographic space**. [S.l.]: Springer, Berlin, 1992. v. 639. (Lecture Notes in Computer Science). p. 162–178.
- FROMMBERGER, Lutz et al. **Composition in OPRAM**. [S.l.], 2007.
- GAMES, Paul A.; HOWELL, John F. Pairwise Multiple Comparison Procedures with Unequal N's and/or Variances: A Monte Carlo Study. **Journal of Educational Statistics**, v. 1, n. 2, p. 113–125, 1976.
- GELMAN, A. et al. **Bayesian Data Analysis**. 3. ed. [S.l.]: Taylor & Francis, 2013.
- GOLLEDGE, Reginald G. Cognitive Maps. In: KEMPF-LEONARD, Kimberly (Ed.). **Encyclopedia of Social Measurement**. New York: Elsevier, 2005. p. 329–339.
- GRISSETTI, Giorgio et al. A Tutorial on Graph-Based SLAM. **IEEE Intelligent Transportation Systems Magazine**, IEEE, v. 2, n. 4, p. 31–43, 2010.
- GUPTA, R. K. **Linear Programming**. [S.l.]: Krishna Prakashan, 2009.
- HA, Inyong et al. Development of Open Humanoid Platform DARwIn-OP. In: SOCIETY OF INSTRUMENT AND CONTROL ENGINEERS (SICE) ANNUAL CONFERENCE. **Proceedings...** Tokyo, Japan: IEEE, 2011. p. 2178–2181.
- HAM, Ineke J.M. van der; CLAESSEN, Michiel H.G. Chapter 8 - Navigation Ability. In: POSTMA, Albert; HAM, Ineke J.M. van der (Ed.). **Neuropsychology of Space**. San Diego: Academic Press, 2017. p. 267–308.
- HART, Peter E; NILSSON, Nils J; RAPHAEL, Bertram. A formal basis for the heuristic determination of minimum cost paths. **IEEE transactions on Systems Science and Cybernetics**, IEEE, v. 4, n. 2, p. 100–107, 1968.

HERSKOVITS, Annette. **Language and Spatial Cognition**. New York: Cambridge University Press, 1987.

HONDA. **ASIMO**. [S.l.: s.n.]. Disponível em:

<http://asimo.honda.com/ASIMO_DCTM/News/images/highres/Meet_ASIMO.jpg>. Acesso em: 23.jul.2015.

HONG, Wei; ZHOU, Changjiu; TIAN, Yantao. Robust Monte Carlo Localization for humanoid soccer robot. In: IEEE/ASME INTL. CONF. ON ADVANCED INTELLIGENT MECHATRONICS. **Proceedings...** [S.l.]: IEEE, 2009.

HOWARD, R.A. **Dynamic Programming and Markov Processes**. [S.l.]: The MIT Press, 1960.

JULIER, Simon J.; UHLMANN, Jeffrey K. A New Extension of the Kalman Filter to Nonlinear Systems. In: p. 182–193.

KAELBLING, L.; LITTMAN, Michael L.; MOORE, Andrew W. Reinforcement learning: A survey. **Journal of Artificial Intelligence Research**, USA, v. 4, p. 237–285, 1996.

KALMAN, R. E. A New Approach to Linear Filtering and Prediction Problems. **Transactions of the ASME – Journal of Basic Engineering**, 82 (Series D), p. 35–45, 1960.

KARMAKAR, N. A new polynomial-time algorithm for linear programming. In: SIXTEENTH ANNUAL ACM SYMPOSIUM ON THEORY OF COMPUTING. **Proceedings...** New York: ACM, 1984. p. 302–311.

KHATIB, O. Real-time obstacle avoidance for manipulators and mobile robots. **International Journal of Robotics Research**, Sage Publications, Thousand Oaks, v. 5, n. 1, p. 90–98, abr. 1986.

KITANO, H. et al. RoboCup: A challenge Problem for AI. **AI Magazine**, v. 18, n. 1, p. 73–85, 1997.

KLATZKY, Roberta L. Allocentric and egocentric spatial representations: definitions, distinctions, and interconnections. In: **Spatial Cognition: An Interdisciplinary Approach to Representing and Processing Spatial Knowledge**. Edição: Christian Freksa, Christopher Habel e Karl F. Wender. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998. p. 1–17.

KRANJEC, Alexander. Extending Spatial Frames of Reference to Temporal Concepts. In: ANNUAL CONFERENCE OF THE COGNITIVE SCIENCE SOCIETY. **Proceedings...** [S.l.: s.n.], 2006.

KREUTZMANN, Arne et al. Towards Safe Navigation by Formalizing Navigation Rules. **TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation**, Gdynia Maritime University, Faculty of Navigation, v. 7, n. 2, p. 161–168, 2013.

LEE, Jae Hee. **Qualitative Reasoning About Relative Directions**. 2013. f. 98. Doctorate in Mathematics and Informatics – University of Bremen, Bremen, Germany.

LEE, Jae Hee; RENZ, Jochen; WOLTER, D. Integration of Qualitative Spatial Reasoning into GIS - An Example with SparQ. In: GEOINFORMATIK-2011 GEOCHANGE. **Proceedings...** Münster, Germany: [s.n.], 2011.

_____. StarVars – Effective Reasoning about Relative Directions. In: INTL. JOINT CONF. ON ARTIFICIAL INTELLIGENCE (IJCAI). **Proceedings...** Beijing, China: [s.n.], 2013. p. 976–982.

LENSER, Scott; VELOSO, Manuela. Sensor Resetting Localization for Poorly Modelled Mobile Robots. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA). **Proceedings...** [S.l.]: IEEE, 2000.

LEONARD, J. J.; DURRANT-WHYTE, H. F. Mobile Robot Localization by Tracking Geometric Beacons. **IEEE Trans Robotics and Automation**, v. 7, p. 376–382, 1991.

LIGOZAT, G. E. **Qualitative Spatial and Temporal Reasoning**. [S.l.]: Wiley, 2013. (ISTE).

_____. Qualitative triangulation for spatial reasoning. In: EUROPEAN CONFERENCE ON SPATIAL INFORMATION THEORY (COSIT'93) - SPATIAL INFORMATION THEORY: A THEORETICAL BASIS FOR GIS. **Proceedings...** Italy: Springer Berlin Heidelberg, 1993. p. 54–68.

_____. Reasoning about Cardinal Directions. **Journal of Visual Languages and Computing**, v. 9, p. 23–44, 1998.

LIPSCHUTZ, S. **Theory and problems of probability**. [S.l.]: McGraw-Hill, 1965.

LITTMAN, Michael L. Markov games as a framework for multi-agent reinforcement learning. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING. **Proceedings...** New Jersey, USA: [s.n.], 1994. p. 157–163.

LIU, Hangcheng. **Comparing Welch ANOVA, a Kruskal-Wallis test, and traditional ANOVA in case of heterogeneity of Variance**. 2015. Diss. (Mestrado) – Faculty of Virginia Commonwealth University. Master of Science Degrees in Biostatistics.

MCDONALD, J.H. **Handbook of Biological Statistics**. 3. ed. Baltimore, Maryland: Sparky House Publishing, 2014.

MILFORD, Michael; SCHULZ, Ruth. Principles of goal-directed spatial robot navigation in biomimetic models. **Philosophical Transactions of the Royal Society of London B: Biological Sciences**, The Royal Society, v. 369, n. 1655, 2014.

MITCHELL, T. **Machine Learning**. New York: McGraw Hill, 1997.

MONTEMERLO, Michael et al. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In: AAAI NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE. **Proceedings...** [S.l.]: AAAI, 2002. p. 593–598.

MORATZ, Reinhard. Representing Relative Direction as a Binary Relation of Oriented Points. In: ECAI 2006 Proceedings of the 17th European Conference on Artificial Intelligence. [S.l.: s.n.], 2006.

MORATZ, Reinhard; NEBEL, Bernhard; FREKSA, Christian. Qualitative Spatial Reasoning about Relative Position. In: SPATIAL COGNITION III. **Proceedings...** Berlin, Heidelberg: Springer Berlin Heidelberg, 2003. p. 385–400.

MORATZ, Reinhard; RENZ, Jochen; WOLTER, D. Qualitative Spatial Reasoning about Line Segments. In: EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE (ECAI). **Proceedings...** [S.l.]: IOS Press, 2000. p. 234–238.

MORATZ, Reinhard; TENBRINK, Thora. Spatial reference in linguistic human-robot interaction: Iterative, empirically supported development of a model of projective relations. **Spatial Cognition & Computation**, Taylor & Francis, v. 6, n. 1, p. 63–107, 2006.

MORATZ, Reinhard; WALLGRÜN, Jan Oliver. Spatial reasoning with augmented points: Extending cardinal directions with local distances. **Journal of Spatial Information Science**, v. 5, n. 1, p. 1–30, 2012.

MOSSAKOWSKI, Till; MORATZ, Reinhard. Qualitative Reasoning About Relative Direction of Oriented Points. **Artificial Intelligence**, Elsevier Science Publishers Ltd., Essex, UK, v. 180-181, p. 34–45, abr. 2012.

MUEGGLER, E. et al. Aerial-guided navigation of a ground robot among movable obstacles. In: IEEE INTERNATIONAL SYMPOSIUM ON SAFETY, SECURITY, AND RESCUE ROBOTICS. **Proceedings...** [S.l.: s.n.], out. 2014. p. 1–8.

NEAPOLITAN, Richard E. **Learning Bayesian Networks**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2003.

NEHMZOW, Ulrich. **Mobile Robotics: A Practical Introduction**. 2. ed. Berlin, Heidelberg: Springer-Verlag, 2003.

PEREIRA, V. F. **Interpretação de Imagens com Raciocínio Espacial Qualitativo Probabilístico**. 2014. f. 174. Tese (Doutorado) – Universidade de São Paulo, São Paulo, SP.

PEREIRA, V.F. et al. A Qualitative-Probabilistic Approach to Autonomous Mobile Robot Self Localisation and Self Vision Calibration. In: BRAZILIAN CONFERENCE ON INTELLIGENT SYSTEMS (BRACIS). **Proceedings...** Fortaleza: [s.n.], 2013. p. 157–162.

PERICO, D. H.; SANTOS, P. E.; BIANCHI, R. A. C. Vision-Based Monte Carlo Localization without Measurement: A Qualitative Approach during Update Phase. In: LATIN AMERICAN ROBOTICS SYMPOSIUM AND BRAZILIAN SYMPOSIUM ON ROBOTICS (LARS-SBR). **Proceedings...** Uberlândia: [s.n.], out. 2015. p. 216–221.

PERICO, D. H. et al. A Robot Simulator Based on the Cross Architecture for the Development of Cognitive Robotics. In: LATIN AMERICAN ROBOTICS SYMPOSIUM AND BRAZILIAN SYMPOSIUM ON ROBOTICS (LARS-SBR). **Proceedings...** Recife: [s.n.], out. 2016a. p. 317–322.

PERICO, D. H. et al. Collaborative communication of qualitative spatial perceptions for multi-robot systems. In: 11. INTERNATIONAL WORKSHOP ON QUALITATIVE REASONING @ IJCAI-16. **Proceedings...** New York: [s.n.], 2016b. v. 4, p. 11.

PERICO, D. H. et al. Hardware and Software Aspects of the Design and Assembly of a New Humanoid Robot for RoboCup Soccer. In: JOINT CONFERENCE ON ROBOTICS: SBR-LARS ROBOTICS SYMPOSIUM AND ROBOCONTROL (SBR LARS ROBOCONTROL). **Proceedings...** São Carlos: [s.n.], out. 2014. p. 73–78.

PESSOA, F. **Poesias**. [S.l.]: L&PM, 1997. Seleção de Sueli Barros Cassal.

PIAGET, J.; INHELDER, B. **The Child's Conception of Space**. New York: W.W. Norton & Company, 1967.

PORTO, M.; ABREU, J.; PORTO, R. Human-computer interaction: A prototype for data and spatial relations representation and interpretation in WebGIS. In: INTERNATIONAL CONFERENCE ON GEOINFORMATICS. **Proceedings...** [S.l.: s.n.], jun. 2011. p. 1–5.

RANDELL, D.; CUI, Zhan; COHN, A. G. A Spatial Logic Based on Regions and Connection. In: INTERNATIONAL CONFERENCE ON PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING. **Proceedings...** Cambridge, MA: Morgan Kaufmann Publishers, 1992. (KR'92), p. 165–176.

RANDELL, D.; WITKOWSKI, Mark; SHANAHAN, Murray. From Images to Bodies: Modelling and Exploiting Spatial Occlusion and Motion Parallax. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE (IJCAI). Seattle: Morgan Kaufmann Publishers, 2001. p. 57–63.

REID, Donald B. An Algorithm for Tracking Multiple Targets. **IEEE Transactions on Automatic Control**, v. 24, p. 843–854, 1979.

RENZ, Jochen. **Qualitative Spatial Reasoning with Topological Information**. Berlin, Heidelberg: Springer-Verlag, 2002.

RENZ, Jochen; MITRA, Debasis. Qualitative Direction Calculi with Arbitrary Granularity. In: PACIFIC RIM INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE (PRICAI). **Proceedings...** Auckland: Springer, 2004. p. 65–74.

RENZ, Jochen; NEBEL, Bernhard. Qualitative Spatial Reasoning Using Constraint Calculi. In: AIELLO, Marco; PRATT-HARTMANN, Ian; VAN BENTHEM, Johan (Ed.). **Handbook of Spatial Logics**. [S.l.]: Springer Netherlands, 2007. p. 161–215.

RESCORLA, Michael. Cognitive Maps and the Language of Thought. **The British Journal for the Philosophy of Science**, v. 60, n. 2, p. 377–407, 2009.

RETZ-SCHMIDT, Gudula. Various Views on Spatial Prepositions. **AI Magazine**, p. 95–105, 1988.

ROBOCUP. **RoboCup.org**. 2015. Disponível em: <<http://www.robocup.org/>>.

RUSSELL, Stuart; NORVIG, Peter. **Artificial Intelligence: A Modern Approach**. 3. ed. [S.l.]: Prentice Hall, 2010.

SANCHEZ, Javier Nicolas; MILSTEIN, Adam; WILLIAMSON, Evan Tang. Robust Global Localization Using Clustered Particle Filtering. **CoRR**, 2002.

SANTOS, P. E. Raciocínio e Percepção Espacial: Uma Abordagem Lógica. [S.l.], 2012. Disponível em: <<http://fei.edu.br/~psantos/qsr-tutorial.pdf>>.

SANTOS, P. E.; DEE, H.M.; FENELON, V. Qualitative robot localisation using information from cast shadows. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA). **Proceedings...** [S.l.: s.n.], mai. 2009. p. 220–225.

SANTOS, P. E. et al. Probabilistic self-localisation on a qualitative map based on occlusions. **Journal of Experimental & Theoretical Artificial Intelligence**, Taylor & Francis, v. 28, n. 5, p. 781–799, 2016.

- SANTOS, Paulo E.; SANTOS, Danilo Nunes. Towards an image understanding system for multiple viewpoints. In: BRAZILIAN SYMPOSIUM ON INTELLIGENT AUTOMATION. **Proceedings...** Fortaleza: [s.n.], 2013.
- SCHILLER, J.; SRINIVASAN, R.A.; SPIEGEL, M. **Schaum's Outline of Probability and Statistics**. 3. ed. [S.l.]: McGraw-Hill Education, 2008. (Schaum's Outline Series).
- SCHRIJVER, Alexander. **Theory of Linear and Integer Programming**. New York: John Wiley & Sons, 1986.
- SCIVOS, Alexander; NEBEL, Bernhard. The Finest of its Class: The Natural Point-Based Ternary Calculus LR for Qualitative Spatial Reasoning. In: SPATIAL COGNITION IV. REASONING, ACTION, INTERACTION. **Proceedings...** Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 283–303.
- SMITH, R.; SELF, M.; CHEESEMAN, P. A stochastic map for uncertain spatial relationships. In: INTERNATIONAL SYMPOSIUM ON ROBOTICS RESEARCH. **Proceedings...** Univ. of California, Santa Clara: MIT Press, 1988. p. 467–474.
- _____. Estimating uncertain spatial relationships in robotics. In: COX, I. J.; WILFONG, G. T. (Ed.). **Autonomous Robot Vehicles**. [S.l.: s.n.], 1990. v. 8. p. 167–193.
- SOFTBANK. **NAO Robot**. [S.l.: s.n.]. Disponível em: <<https://www.ald.softbankrobotics.com/en/robots/nao/find-out-more-about-nao>>. Acesso em: 23.set.2017.
- SORENSEN, H.W. **Kalman Filtering: Theory and Application**. [S.l.]: IEEE Press, 1985. (IEEE Press selected reprint series).
- STACHNISS, Cyrill; BURGARD, W. Particle Filters for Robot Navigation. **Found. Trends Robot**, Now Publishers Inc., Hanover, MA, USA, v. 3, n. 4, p. 211–282, jan. 2014.
- STACHNISS, Cyrill; LEONARD, J. J.; THRUN, S. Simultaneous Localization and Mapping. In: **Springer Handbook of Robotics**. Edição: Bruno Siciliano e Oussama Khatib. Cham: Springer International Publishing, 2016. p. 1153–1176.
- THRUN, S. Particle Filters in Robotics. In: ANNUAL CONFERENCE ON UNCERTAINTY IN AI (UAI). **Proceedings...** [S.l.: s.n.], 2002.
- THRUN, S.; BURGARD, W.; FOX, D. **Probabilistic Robotics**. [S.l.]: The MIT Press, 2005.
- THRUN, S. et al. Robust monte carlo localization for mobile robots. **Artificial Intelligence**, v. 128, p. 99–141, 2000.

VORWERG, Constanze et al. Projective relations for 3D space: Computational model, application, and psychological evaluation. In: NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE (AAAI). **Proceedings...** Providence: [s.n.], 1997. p. 159–164.

WAN, Eric A.; MERWE, Rudolph Van Der. The unscented Kalman filter for nonlinear estimation. In: ADAPTIVE SYSTEMS FOR SIGNAL PROCESSING, COMMUNICATIONS, AND CONTROL SYMPOSIUM. **Proceedings...** [S.l.: s.n.], 2000. p. 153–158.

WASEDA. **Wabot-1**. [S.l.: s.n.], 1973. Disponível em:
<http://www.humanoid.waseda.ac.jp/booklet/kato_2.html>. Acesso em: 22.mai.2014.

WELCH, B. L. On the Comparison of Several Mean Values: An Alternative Approach. **Biometrika**, v. 38, n. 3-4, p. 330–336, 1951.

ANEXO A – TRABALHOS PUBLICADOS

Todos os trabalhos publicados durante o doutorado encontram-se abaixo, em ordem cronológica, respeitando a seguinte sequência: periódicos, capítulos de livros, artigos em congressos internacionais e artigos em congressos nacionais.

A.1 PERIÓDICOS

Retrieving and reusing qualitative cases: An application in humanoid-robot soccer.

HOMEM, T. P. D.; PERICO, D. H.; SANTOS, P. E.; BIANCHI, R. A. C.; DE MANTARAS, R. L. *AI COMMUNICATIONS*, v. 30, p. 251-265, 2017.

A.2 CAPÍTULOS DE LIVROS

Humanoid Robot Gait on Sloping Floors Using Reinforcement Learning.

SILVA, I. J.; PERICO, D. H.; HOMEM, T. P. D.; VILÃO, C. O.; TONIDANDEL, F.; BIANCHI, R. A. C. In: Fernando Santos Osório, Rogério Sales Gonçalves. (Org.). *Communications in Computer and Information Science*. 1ed.: Springer International Publishing, 2016, v. 619, p. 228-246.

Qualitative Case-Based Reasoning for Humanoid Robot Soccer: A New Retrieval and Reuse Algorithm. HOMEM, T. P. D.; PERICO, D. H.; SANTOS, P. E.; BIANCHI, R. A. C.; DE MANTARAS, R. L. In: Ashok Goel, M Belén Díaz-Agudo, Thomas Roth-Berghofer. (Org.). *Lecture Notes in Computer Science*. 1ed.: Springer International Publishing, 2016, v. 9969, p. 170-185.

Newton: A High Level Control Humanoid Robot for the RoboCup Soccer KidSize League. PERICO, D. H.; SILVA, I. J.; VILÃO Jr., C. O.; HOMEM, T. P. D.; DESTRO, R. C.; TONIDANDEL, F.; BIANCHI, R. A. C. In: Fernando S. Osório; Denis Fernando Wolf; Kalinka Castelo Branco; Valdir Grassi Jr.; Marcelo Becker; Roseli A. Francelin Romero. (Org.). *Communications in Computer and Information Science*. 1ed.: Springer Berlin Heidelberg, 2015, v. 507, p. 53-73.

A.3 ARTIGOS PUBLICADOS EM CONGRESSOS INTERNACIONAIS

Collaborative Communication of Qualitative Spatial Perceptions for Multi-Robot Systems. PERICO, D. H.; SANTOS, P. E.; BIANCHI, R. A. C.; MANTARAS, R. L. In: International Workshop on Qualitative Reasoning @ IJCAI-16, 2016, New York. 29th International Workshop on Qualitative Reasoning @ IJCAI-16, 2016. v. 29. p. 77-84.

A Robot Simulator Based on the Cross Architecture for the Development of Cognitive Robotics. PERICO, D. H.; HOMEM, T. P. D.; ALMEIDA, A. C.; SILVA, I. J.;

VILÃO, C. O.; FERREIRA, V. N.; BIANCHI, R. A. C. In: 2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR), 2016, Recife. 2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR). p. 317-322.

Using Reinforcement Learning to Improve the Stability of a Humanoid Robot: Walking on Sloped Terrain. SILVA, I. J.; PERICO, D. H.; HOMEM, T. P. D.; VILÃO, C. O.; TONIDANDEL, F. ; BIANCHI, R. A. C. In: 2015 12th Latin American Robotics Symposium (LARS) and 2015 3rd Brazilian Symposium on Robotics (LARSSBR), 2015, Uberlândia. 2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR). p. 210.

Vision-Based Monte Carlo Localization without Measurement: A Qualitative Approach during Update Phase. PERICO, D. H.; SANTOS, P. E.; BIANCHI, R. A. C. In: 2015 12th Latin American Robotics Symposium (LARS) and 2015 3rd Brazilian Symposium on Robotics (LARSSBR), 2015, Uberlândia. 2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR). p. 216.

Hardware and Software Aspects of the Design and Assembly of a New Humanoid Robot for RoboCup Soccer. PERICO, D. H.; SILVA, I. J.; VILÃO, C. O.; HOMEM, T. P. D.; DESTRO, R. C.; TONIDANDEL, F.; BIANCHI, R. A. C. In: 2014 Joint Conference on Robotics: SBRLARS Robotics Symposium and Robocontrol (SBR LARS Robocontrol), 2014, São Carlos. 2014 Joint Conference on Robotics: SBR-LARS Robotics Symposium and Robocontrol. p. 73.

A Single Camera Vision System for a Humanoid Robot. VILÃO, C. O.; PERICO, D. H.; SILVA, I. J.; HOMEM, T. P. D.; TONIDANDEL, F.; BIANCHI, R. A. C. In: 2014 Joint Conference on Robotics: SBRLARS Robotics Symposium and Robocontrol (SBRLARS Robocontrol), 2014, São Carlos. 2014 Joint Conference on Robotics: SBR-LARS Robotics Symposium and Robocontrol. p. 181.

A.4 ARTIGOS PUBLICADOS EM CONGRESSOS NACIONAIS

Using Reinforcement Learning to Optimize Gait Generation Parameters of a Humanoid Robot. SILVA, I. J.; PERICO, D. H.; COSTA A. H. R.; BIANCHI, R. A. C. In: XIII Simpósio Brasileiro de Automação Inteligente, 2017, Porto Alegre. p. 288.

Improving Reinforcement Learning results with Qualitative Spatial Representation. HOMEM, T. P. D.; PERICO, D. H.; SANTOS, P. E.; COSTA A. H. R.; BIANCHI, R. A. C. Brazilian Conference on Intelligent Systems, 2017, Uberlândia. p. 151.

ANEXO B – PRÊMIOS E TÍTULOS

Todos os prêmios e títulos recebidos durante o doutorado encontram-se abaixo em ordem cronológica.

Best Student Video Award for the video: Qualitative Case-Based Reasoning for Humanoid Robot Soccer, International Conference on Case-Based Reasoning - Video Competition (ICCBR-VC). 2017.

Campeão Latino-Americano e Brasileiro da RoboCup Humanoid Kid Size League com o time RoboFEI-HT, Latin American Robotics Competition / Competição Brasileira de Robótica (LARC / CBR). 2016.

Destaque acadêmico no ano de 2014 pelo projeto de futebol de robôs humanoides - indicação feita pelos departamentos de Ciência da Computação e Engenharia de Automação e Controle, Centro Universitário da FEI. 2015.

Campeão Latino-Americano e Brasileiro da RoboCup Humanoid Kid Size League com o time RoboFEI-HT, Latin American Robotics Competition / Competição Brasileira de Robótica (LARC / CBR). 2014.