



**ZONA VIRTUAL DEFORMÁVEL COM FILTRO DE PARTÍCULAS
NO RASTREAMENTO DE OBSTÁCULOS EM ROBÓTICA MÓVEL**

JOHN ROBERT MUÑOZ CHAVEZ

**DISSERTAÇÃO DE MESTRADO EM SISTEMAS MECATRÔNICOS
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**FACULDADE DE TECNOLOGIA
UNIVERSIDADE DE BRASÍLIA**

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**ZONA VIRTUAL DEFORMÁVEL COM FILTRO DE PARTÍCULAS
NO RASTREAMENTO DE OBSTÁCULOS EM ROBÓTICA MÓVEL**

JOHN ROBERT MUÑOZ CHAVEZ

**DISSERTAÇÃO DE MESTRADO ACADÊMICO SUBMETIDA AO DEPARTAMENTO
DE ENGENHARIA MECÂNICA DA FACULDADE DE TECNOLOGIA DA UNI-
VERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISITOS NECESSÁRIOS
PARA A OBTENÇÃO DO GRAU DE MESTRE EM SISTEMAS MECATRÔNICOS.**

APROVADA POR:

**Prof. Dra. Carla Maria Chagas e Cavalcante Koike, CIC/UnB
(Orientador)**

**Prof. Dr. Ubirajara Franco Moreno, DAS/CTC/UFSC
Membro Externo**

**Prof. Dr. Flávio de Barros Vidal, CIC/UnB
Membro Interno**

BRASÍLIA, 27 DE OUTUBRO DE 2014.

FICHA CATALOGRÁFICA

MUÑOZ CHAVEZ, JOHN ROBERT

Zona Virtual Deformável com Filtro de Partículasno Rastreamento de Obstáculos em Robótica Móvel [Distrito Federal] 2014.

xv, 82p., 210 x 297 mm (ENM/FT/UnB, Mestre, Sistemas Mecatrônicos, 2014).

Dissertação de Mestrado – Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Mecânica

1. Zonas Virtuais Deformáveis

2. Evitar Obstáculos

3. Filtro de Partículas

4. Player/Stage

I. ENM/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

CHAVEZ, J. R. M. (2014). Zona Virtual Deformável com Filtro de Partículasno Rastreamento de Obstáculos em Robótica Móvel, Dissertação de Mestrado em Sistemas Mecatrônicos, Publicação ENM.DM-77/14, Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, DF, 82p.

CESSÃO DE DIREITOS

AUTOR: John Robert Muñoz Chavez

TÍTULO: Zona Virtual Deformável com Filtro de Partículasno Rastreamento de Obstáculos em Robótica Móvel.

GRAU: Mestre

ANO: 2014

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

John Robert Muñoz Chavez

Departamento de Eng. Mecânica (ENM) - FT

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70919-970 - Brasília - DF - Brasil

Dedicado esta dissertação aos meus pais Blanca Nubia Chavez e Dumer Muñoz, à minha esposa Clara Milena Concha.

AGRADECIMENTOS

A Deus, por ter me dado força, paciência, inteligência e ensinado o caminho certo.

Aos meus pais Blanca Nubia Chavez e Dumer Muñoz, ao meu irmão Javier Andres Muñoz, pela confiança e apoio.

À minha adorada esposa Clara Milena Concha, que sempre me apoiou na longa realização deste trabalho.

À professora Carla Koike, pela oportunidade, e orientação no desenvolvimento deste trabalho.

Ao professor Flávio Vidal, por contribuições.

Aos meus amigos pelo entusiasmo nos momentos difíceis.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pela concessão da bolsa de mestrado.

RESUMO

ZONA VIRTUAL DEFORMÁVEL COM FILTRO DE PARTÍCULAS NO RASTRE- AMENTO DE OBSTÁCULOS EM ROBÓTICA MÓVEL

Autor: John Robert Muñoz Chavez

Orientador: Prof. Dra. Carla Maria Chagas e Cavalcante Koike, CIC/UnB

Programa de Pós-graduação em Sistemas Mecatrônicos

Brasília, Outubro de 2014

Na área da robótica móvel cada vez que um robô é posto a navegar em qualquer ambiente, torna-se necessário garantir a sua integridade física. Para evitar uma colisão providenciam-se sensores cuja função é perceber o ambiente de trabalho a fim de estimar a posição dos obstáculos que possam surgir no seu caminho. Pensando nisso, foram desenvolvidas técnicas baseadas em metodologias reativas como a *Zona Virtual Deformável (ZVD)* que gera uma área de proteção ao redor do robô.

O presente trabalho investiga a situação de um robô que possui a limitação de ter um sensor *laser* a frente, o que impede a percepção do obstáculo mesmo após tê-lo evitado, o que ainda pode por em risco a sua integridade. A partir daí propõem-se o desenvolvimento de um algoritmo para estimar a posição do obstáculo a frente do robô inclusive quando ele se encontra fora do alcance do sensor. Para realizar a estimativa da posição do osbtáculo, implementou-se o filtro de partículas integrado a uma *ZVD*, que vai se deformar pela interação do robô com o ambiente. A deformação da *ZVD* gera ações de controle para minimizar sempre a deformação, fazendo com que o robô evite a colisão.

ABSTRACT

DEFORMABLE VIRTUAL ZONE WITH PARTICLE FILTER IN THE TRACKING OF OBSTACLES IN MOBILE ROBOTICS

Author: John Robert Muñoz Chavez

Supervisor: Prof. Dra. Carla Maria Chagas e Cavalcante Koike, CIC/UnB

Programa de Pós-graduação em Sistemas Mecatrônicos

In mobile robotics each time a robot is set to navigate in an environment, it is necessary to ensure their physical integrity. To avoid a collision with obstacles, sensors are employed to perceive the environment in order to estimate the position of the obstacles near the robot. Several techniques have been developed based on reactive methods such as *Deformable Virtual Zone (DVZ)* that generates a protective area around the robot.

This work assumes that the robot has the limitation of having only one laser sensor forward, which prevents the perception of obstacle after it has been avoided, but can still be near the robot. Based on this assumption, an algorithm is proposed to estimate the position of obstacles detected ahead of the robot and to track them even when they are out of the sensor range. A particle filter is proposed to estimate the obstacle pose and it is integrated into a *DVZ* around the robot, which will be deformed according to the interaction of the robot with the environment. The deformation of the *DVZ* generates control actions that minimize deformation and help the robot to avoid the collision with detected obstacles.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	OBJETIVO	2
1.2	ORGANIZAÇÃO DO TRABALHO	2
2	REVISÃO BIBLIOGRÁFICA	4
2.1	EVITANDO OBSTÁCULOS	4
2.1.1	MÉTODOS REATIVOS	4
2.1.2	MÉTODOS DELIBERATIVOS	5
2.1.3	MÉTODOS HÍBRIDOS	5
2.1.4	CAMPO POTENCIAL	5
2.1.5	TÉCNICA DE BANDAS ELÁSTICAS	7
2.1.6	ZONA VIRTUAL DEFORMÁVEL	8
2.2	DISCUSSÃO	10
3	FILTRAGEM BAYESIANA	11
3.1	CONCEITOS BÁSICOS EM PROBABILIDADE	11
3.1.1	VARIÁVEL ALEATÓRIA:	12
3.1.2	FUNÇÃO DENSIDADE DE PROBABILIDADE	12
3.1.3	DISTRIBUIÇÃO NORMAL OU DISTRIBUIÇÃO DE GAUSS	12
3.1.4	PROBABILIDADE CONDICIONAL	13
3.1.5	PROBABILIDADE CONJUNTA	13
3.1.6	PROBABILIDADE TOTAL	13
3.1.7	FUNÇÃO DISTRIBUIÇÃO ACUMULATIVA	14
3.1.8	REGRA DE BAYES	14
3.1.9	CADEIA DE MARKOV	15
3.2	FILTRAGEM ESTOCÁSTICA	15
3.2.1	FILTRO DE BAYES	16
3.3	FILTRO DE PARTÍCULAS	16
3.3.1	INICIALIZAÇÃO	17
3.3.2	PREDIÇÃO	18
3.3.3	ATUALIZAÇÃO	18
3.3.4	REAMOSTRAGEM	19
3.3.5	O ALGORITMO DO FILTRO DE PARTÍCULAS COM REAMOSTRAGEM	22
3.4	MODELO DE OBSERVAÇÃO	22
3.4.1	SENSOR LASER	23
3.4.2	MODELO DE SENSOR	23

3.4.3	OBSTÁCULO DETECTADO	24
3.4.4	OBSTÁCULO NÃO ESPERADO	25
3.4.5	FALHAS DO SENSOR	25
3.4.6	MEDIÇÃO ALEATÓRIA.....	26
3.4.7	JUNÇÃO DE DENSIDADES DE PROBABILIDADE	26
3.5	DISCUSSÃO.....	27
4	PROPOSTA E METODOLOGIA	28
4.1	MODELO DA ZONA VIRTUAL DEFORMÁVEL	29
4.1.1	DEFORMAÇÃO DEVIDO A UM OBSTÁCULO.....	30
4.2	MODELO DO SENSOR LASER	32
4.2.1	LEITURA LASER.....	32
4.2.2	MEDIDAS DO MODELO DO SENSOR	32
4.3	MODELO DE MOVIMENTO	37
4.4	FILTRO DE PARTÍCULAS	39
4.4.1	PREDIÇÃO.....	40
4.4.2	ATUALIZAÇÃO.....	41
4.4.3	REAMOSTRAGEM.....	42
4.4.4	ESTIMATIVA	43
4.5	DEFORMAÇÃO DA ZVD	45
4.6	AÇÃO DE CONTROLE PARA EVITAR O OBSTÁCULO	45
4.7	DISCUSSÃO.....	46
5	ANÁLISE DOS RESULTADOS.....	47
5.1	ROBÔ PIONEER 3AT E FERRAMENTA DE SIMULAÇÃO	47
5.2	SIMULAÇÃO DO ROBÔ EM ROTA DE COLISÃO DIRETA COM O OBSTÁCULO ..	50
5.2.1	COMPARAÇÃO DOS ERROS VARIANDO O NÚMERO DE PARTÍCULAS ...	61
5.2.2	ERRO DAS ESTIMATIVAS NA ZONA DE UMBRA	66
5.3	SIMULAÇÃO DO ROBÔ COM MOVIMENTO ALEATÓRIO DENTRO DE UMA ÁREA COM UM OBSTÁCULO	67
5.3.1	ERRO DAS ESTIMATIVAS NA ZONA DE UMBRA	73
5.4	DISCUSSÃO.....	76
6	CONCLUSÕES, CONTRIBUIÇÕES E TRABALHOS FUTUROS	77
	REFERÊNCIAS BIBLIOGRÁFICAS.....	79

LISTA DE FIGURAS

2.1	Técnica <i>Virtual Force File</i> (VFF) (modificada de [10])	6
2.2	a) Primeira forma da banda elástica e ação das forças. b) Conjunto de bolhas no caminho livre de colisão (retiradas de [14]).....	7
2.3	a) Cooperação de robôs móveis (retirada de [19]). b) Representação 3D da ZVD para o robô manipulador (retirada de [20])	8
2.4	Características da Deformação da ZVD.....	10
3.1	Variável aleatória.	12
3.2	Variável aleatória (retirada de [22]).	14
3.3	a) Varredura realizada pelo sensor <i>laser</i> . b) Sistema de coordenadas da leitura do sensor.	23
3.4	Probabilidade p_{hit} (retirada de [25]).	24
3.5	Probabilidade p_{short} (retirada de [25]).....	25
3.6	Probabilidade p_{max} (retirada de [25]).....	26
3.7	Probabilidade p_{rand} (retirada de [25]).	26
3.8	Junção das Distribuições de Probabilidade que modelam erros nas medidas do sensor (retirada de [25]).....	27
4.1	Zona Virtual Deformável: sua forma e localização em relação ao sistema cartesiano do robô.	30
4.2	Variáveis envolvidas na deformação da ZVD devido a presença de um obstáculo: d_{imax} é a medida máxima do sensor, no limite do seu alcance; S_i é o valor máximo da medida do sensor que mantém a ZVD intacta S ; θ_i é o ângulo da medida do sensor; I é o vetor de intrusão; quando $\rho_i < S_i$, o vetor Δ_i representa a deformação que altera a forma da ZVD para S_i	31
4.3	Esquema da distribuição das zonas do sensor	33
4.4	Esquema de leitura do <i>laser</i> a) detecção de um obstáculo pelo sensor <i>laser</i> em uma das zonas Z_j , b) sistema de coordenadas da leitura do sensor, c) detalhe dos feixes que atingem o obstáculo.	34
4.5	Variáveis envolvidas no movimento do robô do instante de tempo $t = k - 1$ e $t = k$ ao se aproximar um obstáculo: ρ e θ definem a posição do obstáculo em coordenadas polares nos diferentes instantes de tempo, r é o deslocamento em um intervalo de tempo que tende a zero, e φ é o ângulo auxiliar para a obtenção da orientação final do robô θ_k	38
4.6	Função de distribuição de probabilidade Gaussiana, aplicada à partícula i no instante $t = k$ próxima à posição de interesse.	42
4.7	Variáveis do controle.	45

5.1	a) Pioneer 3AT (retirada de [44]).	47
5.2	a) Modelo cliente-servidor. b) Comunicação com o <i>hardware</i> ou simulador (retiradas de [52]).	49
5.3	Trajetória percorrida pelo robô móvel na primeira simulação	51
5.4	Sequência de imagens da primeira simulação	52
5.5	Leitura ρ do obstáculo, feita pelo modelo do sensor. As letras na cor vermelha indicam as posições da Figura 5.4	53
5.6	Leitura θ do obstáculo, feita pelo modelo do sensor. As letras na cor vermelha indicam as posições da Figura 5.4	54
5.7	Distância ρ , em cm, de cada partícula	55
5.8	Orientação θ , em graus, de cada partícula	56
5.9	Peso w_i de cada partícula	57
5.10	Comparação entre a distância ρ real e as estimativas da média ponderada e da melhor partícula	57
5.11	Comparação entre a orientação θ real e as estimativas da média ponderada e da melhor partícula	58
5.12	Comparação da posição do obstáculo no sistema de coordenadas polares do robô móvel.	59
5.13	Erro relativo da estimativa ρ , da média ponderada e da melhor partícula	60
5.14	Erro relativo de θ da média ponderada e da melhor partícula	61
5.15	Postura do obstáculo no sistema de coordenadas polares do robô móvel a) com 100 partículas, b) com 500 partículas	62
5.16	Comparação do erro relativo ρ a) da média ponderada b) da melhor partícula	64
5.17	Comparação do erro relativo θ a) da média ponderada b) da melhor partícula	65
5.18	Trajetória percorrida pelo robô móvel na simulação com movimento aleatório dentro de uma área com um obstáculo.	67
5.19	Informações fornecidas pelo modelo do sensor a) função de probabilidade $P(\rho_k^i, \mathbf{X}_k)$, b) função de probabilidade e $P(\theta_k^i, \mathbf{X}_k)$	68
5.20	Distância ρ , em cm, da saída do filtro	70
5.21	Distância θ , em graus, da saída do filtro	71
5.22	Peso w_i para cada partícula	71
5.23	Comparação da distância real ρ e as estimativas do filtro	72
5.24	Comparação do valor real ρ e as estimativas do filtro	72
5.25	Comparação do erro relativo, da média ponderada e da melhor partícula na estimativa de ρ	73
5.26	Comparação do erro relativo, da média ponderada e da melhor partícula na estimativa de θ	74
5.27	Sequências das trajetórias da posição do obstáculo na zona de umbra a) trajetória entre as iterações b e c , b) trajetória entre as iterações d e e .	75

5.28 Sequências das trajetórias da posição do obstáculo na zona de umbra a) trajetória entre as iterações f e g , b) trajetória entre as iterações h e i	75
5.29 Sequências das trajetórias da posição do obstáculo na zona de umbra a) trajetória entre as iterações j e k , b) trajetória entre a iteração l até o final da simulação.	76

LISTA DE TABELAS

5.1	Média do Erro Percentual da estimativa ρ e θ desde o primeiro dado fornecido pelo sensor até o fim da primeira simulação.	66
5.2	Porcentagem de erro médio da estimativa ρ e θ da primeira simulação durante a trajetória pela zona de umbra.	66
5.3	Porcentagem de erro médio da estimativa ρ e θ desde o primeiro dado fornecido pelo sensor até o fim da segunda simulação.	73
5.4	Porcentagem de erro médio da estimativa ρ e θ da segunda simulação durante a trajetória pela umbra.	76

LISTA DE SÍMBOLOS

S	Zona Virtual Deformável não deformada
S_i	Componente da Zona Virtual Deformável
\mathbf{u}	Vetor de movimento
ν	Velocidade linear
ω	Velocidade angular
SD	Zona Virtual Deformável deformada
I	Vetor de Intrusão
Δ	Vetor de deformação
μ	Média
σ^2	Variância
$N(\mu, \sigma)$	Distribuição de probabilidade Normal ou Gaussiana com
media μ e desvio padrão σ	
$p(A)$	Probabilidade do evento A
$p(A \cap B)$	Probabilidade conjunta dos eventos A e B
$p(A B)$	Probabilidade condicional de A dado que B tenha ocorrido
Ω	Espaço amostral
$bel(x_{t-1})$	Valor esperado do estado x
η	Variável de normalização
z_t	Medições ou observações dos sensores no instante t
N_s	Número de partículas
w_i	Peso da partícula i -ésima
$Var(w_k^* i)$	Variância dos pesos das partículas no instante k
N_{eff}	Tamanho efetivo de amostragem
$N_{sensores}$	Quantidade de sensores do modelo do sensor
\mathbf{LL}_k	Leituras do Laser no tempo $t = k$
ρ_i	Distância do obstáculo fornecida pelo sensor
θ_i	Ângulo correspondente à medida da distância ρ_i
d_{imax}	Limite do alcance do sensor
Υ	Pesos totais
r	Deslocamento do robô calculado em um intervalo de tempo T
\bar{X}_{wmax}	Estimativa da posição da melhor partícula
$\bar{X}_{ponderada}$	Estimativa da posição da média ponderada
ZVD	Zona Virtual Deformável
PDF	Função de Densidade de Probabilidade
CDF	Função Distribuição Cumulativa

1 INTRODUÇÃO

O trabalho desenvolvido nesta dissertação abrange o campo da robótica móvel, que é considerada como uma área de pesquisa que lida com o controle de veículos autônomos ou semi-autônomos [1]. Uma das grandes diferenças entre a robótica móvel e outras áreas da robótica é a sua ênfase nos problemas relacionados à movimentação em ambientes complexos e que se modificam dinamicamente, compostos tanto por obstáculos estáticos como por obstáculos móveis.

Os robôs móveis, na sua grande maioria, são programados para serem autônomos e capazes de se movimentar em ambientes sem a necessidade da intervenção humana [2]. Geralmente, um robô móvel possui certo grau de inteligência em sua programação e um sistema de controle de movimentos executados para tentar cumprir seu objetivo específico de forma segura.

Sempre que um robô móvel é posto a navegar em qualquer ambiente, é preciso garantir que ele não colidirá com obstáculos. Para garantir a sua integridade física, o robô possui sensores capazes de adquirir e utilizar o conhecimento sobre seu entorno, além de ter atuadores que o tornam apto para interagir e responder em tempo real às dificuldades encontradas nesse ambiente. No entanto o robô móvel deve ter a percepção do ambiente de trabalho através dos seus vários sensores embarcados. A partir dessa percepção, o robô móvel poderá, entre outras coisas, estimar sua posição dentro desse ambiente e perceber e reconhecer obstáculos ao seu movimento. Estes são alguns dos problemas fundamentais no estudo dos robôs móveis autônomos.

Visando a garantir a segurança nas aplicações na robótica móvel, são necessários métodos eficientes para conseguir o desvio seguro do obstáculo. Dentre esses métodos, podem ser citados: Campos Potenciais, Campo de Forças Virtual (*CFV*), Histograma do Campo Virtual (*HCV*) e Zona Virtual Deformável (*ZVD*), descritos e referenciados no capítulo 2. Alguns destes métodos são chamados reativos [2], [3], pois não utilizam nenhum conhecimento prévio do meio, salvo alguma informação genérica como, por exemplo, se o ambiente é plano. A navegação nos métodos reativos baseia-se no fato de que as percepções estão diretamente relacionadas com as ações executadas pelo robô móvel, o que oferece uma grande vantagem pelo baixo custo computacional.

Frequentemente, utiliza-se um sensor *laser* em cima de um robô móvel para que este possa navegar desviando dos obstáculos que por acaso surjam em seu caminho. Contudo, alguns robôs possuem a limitação de ter somente um sensor *laser* a frente, o que impede a percepção do obstáculo após tê-lo evitado. No entanto, a trajetória do obstáculo quando já não é mais visível pode ter ainda influência sobre a movimentação do robô.

A proposta desenvolvida neste estudo objetiva permitir ao robô móvel navegar de forma

segura em um ambiente fechado (*indoors*), desviando dos obstáculos que surgirem em seu caminho utilizando somente um sensor *laser* de varredura, embarcado na frente do robô. O algoritmo aqui proposto estima a posição do obstáculo, tanto na zona visível quanto na área posterior do robô móvel, de forma que possa continuar executando os comportamentos de navegação considerando a trajetória estimada desse obstáculo.

Para propiciar uma navegação segura de um robô móvel, apresenta-se a abordagem de uma proposta de um sistema de controle que se baseia na definição de uma zona de proteção que vai se deformar pela interação robô-ambiente. Essa proposta, particularmente, terá um sensor com campo de visão limitado, porém a partir da última informação da posição do obstáculo e da movimentação do robô é possível conseguir estimar a presença de um obstáculo mesmo estando fora do alcance de visão. O método implementado na proposta terá a iteração de duas partes, uma delas será o ambiente, que tratará de maximizar as deformações na zona de proteção e a outra será o controle que tratará de minimizá-las. Nesta proposta, integrar-se-á um filtro de partículas à zona de proteção para conseguir estimar a posição do obstáculo quando ele estiver tanto dentro do alcance do sensor quanto fora dele.

1.1 OBJETIVO

O presente trabalho tem como objetivo desenvolver um algoritmo de controle reativo confiável que garanta a um robô móvel navegar de forma segura em um ambiente desconhecido, propiciando o desviando de obstáculos estando eles dentro ou fora do alcance do sensor, por que de fato representam um risco.

Para garantir que o robô móvel seja capaz de se movimentar de forma segura utilizando somente um sensor *laser* a frente, integrou-se um filtro de partículas à Zona Virtual Deformável (*ZVD*) para estimar a posição do obstáculo quando estiver na *ZVD* dentro do alcance do sensor ou fora dele. O método da *ZVD* consiste na construção de uma zona de proteção ao redor do veículo que se deformará pela intrusão de obstáculos.

1.2 ORGANIZAÇÃO DO TRABALHO

Esta dissertação está organizada da seguinte forma:

- **Capítulo 1:** neste capítulo descrevem-se, em linhas gerais, o tema desta dissertação, bem como o objetivo a ser alcançado.
- **Capítulo 2:** são apresentados os principais conceitos e as metodologias de maior relevância e que de alguma forma contribuíram para o desenvolvimento do trabalho. Dá-se maior ênfase à técnica da Zona Virtual Deformável.

- **Capítulo 3:** são expostos alguns conceitos teóricos de maior relevância referentes à probabilidade para o desenvolvimento do trabalho. Inicialmente são apresentados conceitos sobre probabilidade e filtragem estocástica. Em seguida, explica-se o filtro de partículas, bem como suas variantes.
- **Capítulo 4:** apresenta-se de forma detalhada o modelo de sensor utilizado no filtro de partículas proposto.
- **Capítulo 5:** descreve-se a metodologia utilizada. Posteriormente é apresentada a proposta, a partir da descrição dos principais componentes do algoritmo, do modelo do sensor, do modelo de movimento, do filtro de partículas, da deformação da Zona Virtual Deformável e das ações de controle, assim como a interação entre eles.
- **Capítulo 6:** inicialmente, expõem-se a ferramenta adotada para o desenvolvimento deste trabalho. São apresentados, então, os experimentos realizados, descrevendo seus objetivos, seu processo e os resultados obtidos.
- **Capítulo 7:** neste capítulo são apresentadas as conclusões do trabalho, logo após um breve resumo da proposta; são apresentadas também algumas propostas para trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo, apresentam-se algumas das principais técnicas de navegação utilizadas para resolver tanto problemas de localização quanto de desvio de obstáculos fixos e móveis, com o objetivo de tornar a navegação mais segura.

A primeira seção descreve, em linhas gerais, os métodos de classificação usados na robótica móvel para evasão de obstáculos, e apresenta algumas das técnicas que abordam esse tema. Na seção 2.1.6, descreve-se a Zona Virtual Deformável, onde se abordam os principais tópicos para a modelagem da zona de proteção. Na seção 2.2, faz-se uma discussão dos tópicos do capítulo.

2.1 EVITANDO OBSTÁCULOS

Um dos principais focos de pesquisa em robótica consiste em dotar os robôs móveis de navegação autônoma para a execução de atividades com um objetivo específico [2]. Para tanto, o robô precisa possuir um sistema de sensoriamento que forneça informações sobre a presença de objetos no ambiente, e também é preciso empregar algoritmos que forneçam ao robô certo grau de inteligência no controle de seus movimentos.

O desvio de obstáculos é uma tarefa essencial no relativo à robótica móvel, já que torna o robô capaz de chegar ao seu destino sem colisões [3]. Assim, todos os robôs apresentam algum tipo de método para evasão de obstáculos, ainda que seja um algoritmo primitivo que simplesmente interrompe a movimentação do robô ao detectar um obstáculo. Existem outras abordagens que permitem usar a informação mapeada em tempo real e, com base no sensoriamento do ambiente, planejar o melhor caminho para evitar os obstáculos.

Na robótica móvel, várias soluções têm sido propostas para o problema da detecção de obstáculos e da decisão do comportamento que o robô deve assumir para evitá-los [2]. De acordo com a abordagem adotada, os métodos utilizados podem ser classificados de forma genérica como: deliberativos, reativos ou híbridos [4].

2.1.1 Métodos reativos

Os métodos reativos não utilizam nenhum conhecimento ou planejamento prévio do ambiente no qual o robô está prestes a navegar, o que impossibilita a construção de um modelo interno [5]. A principal característica desses métodos é o fato de que todas as ações são realizadas por meio de comportamentos mapeados diretamente das entradas sensoriais para um padrão de ações motoras [2]. Assim, esses métodos conectam diretamente os sensores

e os atuadores; por esse motivo, as percepções devem estar relacionadas com as ações. Os robôs que utilizam esse tipo de método simplesmente reagem de acordo com a informação atual dos sensores, o que permite que eles respondam mais rapidamente às mudanças no seu entorno. Como resultado, obtêm-se simplicidade e menor custo computacional.

Algumas das propostas sugeridas para solucionar os problemas de detecção de obstáculos e definição de comportamentos perante eles foram desenvolvidas a partir de algoritmos de navegação visual para robôs autônomos [6], onde extrai-se informação das imagens da câmera. Em [7], por exemplo, apresentou-se um planejador de rotas para robôs móveis em ambientes com obstáculos em movimento, que tenta prever a posição futura dos obstáculos para planejar a nova rota do robô.

2.1.2 Métodos Deliberativos

Os métodos deliberativos são aqueles nos quais a navegação é baseada em um modelo do mundo ou mapa do ambiente no qual o robô está inserido [4]. Nestes métodos, incluem-se uma etapa de pré-planejamento da trajetória que será realizada a partir de um mapa previamente conhecido e das informações fornecidas pelos sensores. Para planificar e gerar as ações do robô, todos os planos de ação ou espaços de soluções possíveis são verificados até encontrar um com o qual o objetivo possa ser atingido sem colisão. O sucesso desse método baseia-se na fidelidade do modelo do mundo e da precisão dos sensores, além da necessidade de manter esse modelo o mais atualizado possível. Se o ambiente ao redor do robô mudar, o planejamento será feito com informações desatualizadas dando lugar a erros na execução. Gerar um planejamento de ações livres de obstáculos é um processamento computacionalmente caro, tanto em termos de tempo quanto de memória.

2.1.3 Métodos Híbridos

De acordo com [8], o objetivo do método híbrido é combinar o método deliberativo e o método reativo. Neste método, tem-se um módulo responsável por executar o planejamento a longo prazo e outro módulo responsável por lidar com as situações de reação imediata, tais como evitar obstáculos. Os dois módulos podem, inclusive, ser executados em paralelo, como em [4].

2.1.4 Campo Potencial

De acordo com [9], nesta técnica reativa, o obstáculo gera uma força repulsiva (negativa) ao robô, enquanto o objetivo ou posição final aplica uma força atrativa (positiva) ao robô. A soma de todas as forças gera uma força resultante que determinará a direção e a velocidade do movimento do robô. A partir do valor dessa força resultante, o sistema calcula a aceleração do robô e sua nova posição para uma iteração. O algoritmo é repetido até que o robô atinja

seu ponto de destino.

A ideia de trabalhar com forças imaginárias atuando sobre o robô foi sugerida inicialmente em [9], mas a abordagem só produziu resultados satisfatórios na proposta de Campo de Forças Vetoriais (*Virtual Force File*) ou VFF, em [10] (método deliberativo). VFF foi especialmente desenvolvida para evitar obstáculos em tempo real, e permitir um controle de movimentos rápido, contínuo e preciso em um ambiente com obstáculos desconhecidos.

Outra extensão foi proposta em [11], e é chamada de Histograma de Campo Vetorial (do inglês *Vector Field Histogram VFH*). VFH representa o ambiente mediante histograma, onde cada célula C_{ij} do histograma contém um valor de certeza associado à existência de um obstáculo naquela posição, e o caminho mais curto entre a posição do robô a do alvo é calculado com base nesses valores de certeza. Estes métodos são muito velozes e no geral, só consideram um pequeno conjunto de obstáculos próximos ao robô.

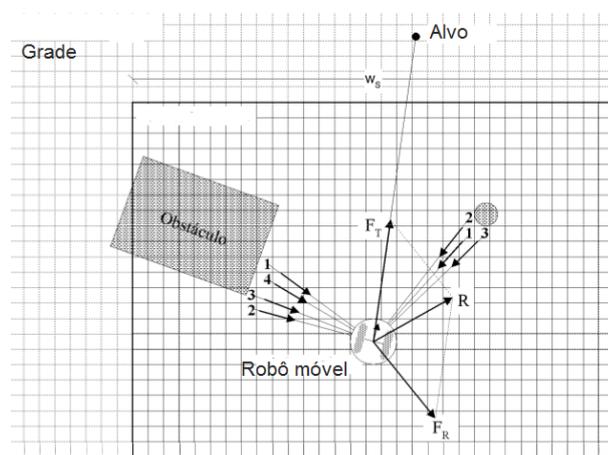


Figura 2.1: Técnica *Virtual Force File* (VFF) (modificada de [10])

Na figura 2.1, pode-se observar que as células ocupadas pelo obstáculo exercem forças repulsivas sobre o robô. A cada iteração, todas as forças repulsivas virtuais são totalizadas para se produzir a força repulsiva resultante F_R . Simultaneamente, uma força atrativa virtual F_T , de amplitude constante, é aplicada ao robô, levando-o em direção ao seu destino. A soma vetorial de F_R e F_T produz a força resultante R .

Uma nova versão desse algoritmo foi apresentada em [12], incluindo um potencial de rotação e um potencial de tarefas (em inglês *task*), onde o primeiro direciona o robô em um caminho mais retilíneo em direção ao alvo, e o segundo potencial reduz a influência dos obstáculos que não incidem no movimento do robô, mas que possam afastá-lo do alvo.

Em [13], este método foi modificado para a navegação em um ambiente desconhecido com vários robôs móveis, se preocupando com os obstáculos e os objetivos, onde a nova função potencial configura um espaço livre de mínimos locais: ou seja o espaço será configurado para que a atração do alvo se estenda pelo espaço livre.

2.1.5 Técnica de Bandas Elásticas

Esta técnica se baseia no conceito de Bolhas (*Bubbles*) inicialmente proposto em [14]. A Bolha é um espaço livre em torno do robô móvel, que pode ser explorado em qualquer direção sem perigo de colisões. Esse conceito é eficiente porque permite utilizar um modelo do ambiente próximo do robô, gerando subconjuntos locais do espaço livre. Quando o robô está longe de obstáculos, as bolhas tendem a ser grandes e podem ser distanciadas no tempo, porém, quando o robô está próximo de obstáculos, as bolhas geradas são menores e em maior quantidade para descrever uma banda elástica.

A banda elástica é um caminho livre de colisões e é também deformável, onde a primeira forma que assume a banda elástica representa o caminho livre gerado por um planejador, como mostra as imagens superiores da figura 2.2 (a).

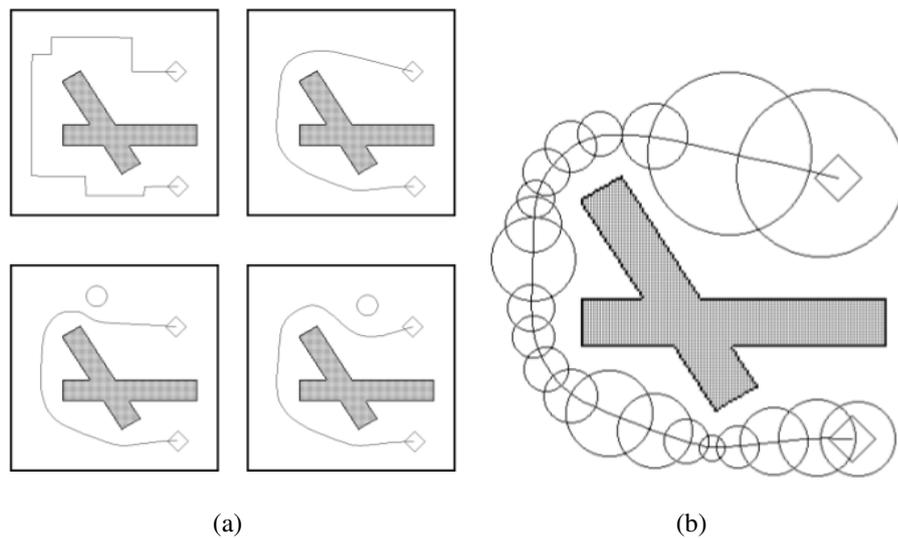


Figura 2.2: a) Primeira forma da banda elástica e ação das forças. b) Conjunto de bolhas no caminho livre de colisão (retiradas de [14]).

As deformações na banda elástica são geradas por duas forças: a primeira força é de contração e simula a tensão na banda esticada, evitando alguma folga no caminho. A segunda força, contrária à de contração, é chamada de repulsão e é fornecida pelos obstáculos que fazem com que a banda seja repelida por eles. Assim, as duas forças permitem deformar a banda elástica com as mudanças no meio ambiente. A presença de novos obstáculos altera as forças e faz com que a banda se deforme para uma nova posição de equilíbrio, como descrito na figura 2.2 (a). Em [15], esta técnica é utilizada, e a banda elástica é formada por um conjunto de bolhas B_i , cada uma delas associada a um raio e a uma distância de mascaramento: cada bolha mascara um obstáculo ou um conjunto de obstáculos, o que deforma a banda elástica a cada mudança detectada no ambiente.

2.1.6 Zona Virtual Deformável

De acordo com [16] e [17], a *Zona Virtual Deformável*, ou *ZVD*, consiste em uma área de proteção definida ao redor do robô. Assume-se que o robô não possui um modelo do espaço ao seu redor, mas pode obter informações sobre a aproximação dos objetos a sua frente. Esta região ou zona tem a capacidade de se deformar quando são detectados obstáculos no seu interior, e assim poder recriar a realidade do espaço em torno do robô [18].

A *ZVD* foi utilizada em diversos trabalhos, sempre com o objetivo de minimizar o risco de colisão. Um exemplo é apresentado em [19], onde um conjunto de robôs móveis (um é o líder e os demais são seus seguidores) utilizam a informação da *ZVD* de cada um deles para modificar o processo de geração de trajetórias, como é possível observar na figura 2.3. Quando em tarefas de cooperação em grupo, pode reagir baseada em informações fornecidas por outros robôs. Outra aplicação foi descrita em [20], onde o princípio da *ZVD* é utilizado em três dimensões em torno de um robô manipulador. Nesta aplicação, um comando de reflexo é gerado, o que permite a emergência de um comportamento reativo, descrito na seção 2.1.1.

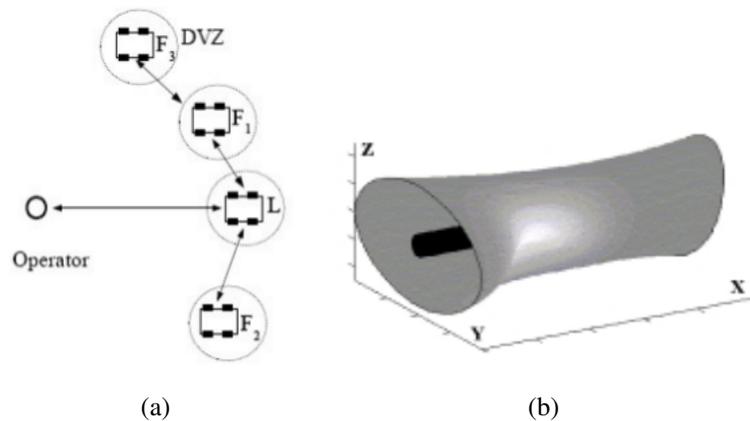


Figura 2.3: a) Cooperação de robôs móveis (retirada de [19]). b) Representação 3D da *ZVD* para o robô manipulador (retirada de [20])

- **Princípio Geral:** para controlar o comportamento reativo de um robô móvel, que se move entre obstáculos no espaço R^2 , é preciso modelar uma *ZVD* que envolva o robô. Geralmente, a *ZVD* possui a forma de uma elipse ou de um círculo, mas sua geometria poderá ser alterada em função da cinemática de movimento do robô e dos obstáculos detectados [20].

Em [20], são apresentadas algumas definições que descrevem o princípio da *ZVD* e a sua parametrização.

A *ZVD* pode ser representada por um vetor S , onde cada componente S_i corresponde a distância fornecida por cada feixe do *laser*, que representa o comprimento desde o limite da *ZVD* até o centro de referência em coordenadas polares, localizado no corpo do robô

(ou no seu centro de massa, ou ainda no centro do sensor). O tamanho do vetor S está diretamente relacionado à resolução angular utilizada no sensor *laser*:

$$S = [S_0, S_1, \dots, S_n]^T. \quad (2.1)$$

O vetor u definirá o movimento do robô e, quando a *ZVD* sofrer alguma deformação, o controle estará encarregado de levar o robô a um estado no qual a *ZVD* esteja livre de deformações. Este vetor é composto pelas variáveis controláveis do robô, como sua velocidade linear ν , e velocidade angular ω , por exemplo:

$$u = \begin{bmatrix} \nu \\ \omega \end{bmatrix}, \quad (2.2)$$

Portanto, a *ZVD* não deformada pelos obstáculos no ambiente depende unicamente do vetor que caracteriza o movimento do robô. Quando um objeto entra na zona, ocorre uma deformação que gera a *ZVD deformada*, representada pelo vetor:

$$SD = [SD_0, SD_1, \dots, SD_n]^T, \quad (2.3)$$

onde cada SD_i depende da variação da medida ρ_i fornecida pelo feixe do sensor laser nessa direção.

O vetor I , chamado de *vetor de intrusão*, é apresentado na equação 2.4, onde $I_i = d_{imax} - \rho_i$. O sensor fornece a medida $\rho_i = d_{imax}$ na ausência do obstáculo.

$$I = [I_0, I_1, \dots, I_n]^T, \quad (2.4)$$

onde cada componente I_i representa a profundidade da intrusão gerada pelo obstáculo na *ZVD* de forma polar para cada ângulo θ_i .

O *vetor de deformação* Δ representa a profundidade da deformação da *ZVD*:

$$\Delta = [\Delta_0, \Delta_1, \dots, \Delta_n]^T, \quad (2.5)$$

onde cada componente Δ_i do vetor de deformação depende do vetor de intrusão e da zona virtual não deformada. Portanto, Δ pode ser escrito como uma função na equação 2.6:

$$\Delta = \begin{cases} 0 & \text{se } \rho_i \geq S_i, \\ S_i - \rho_i & \text{se } \rho_i < S_i. \end{cases} \quad (2.6)$$

A Figura 2.4 ilustra uma situação típica, onde o robô se movimenta pelo ambiente e um obstáculo se aproxima.

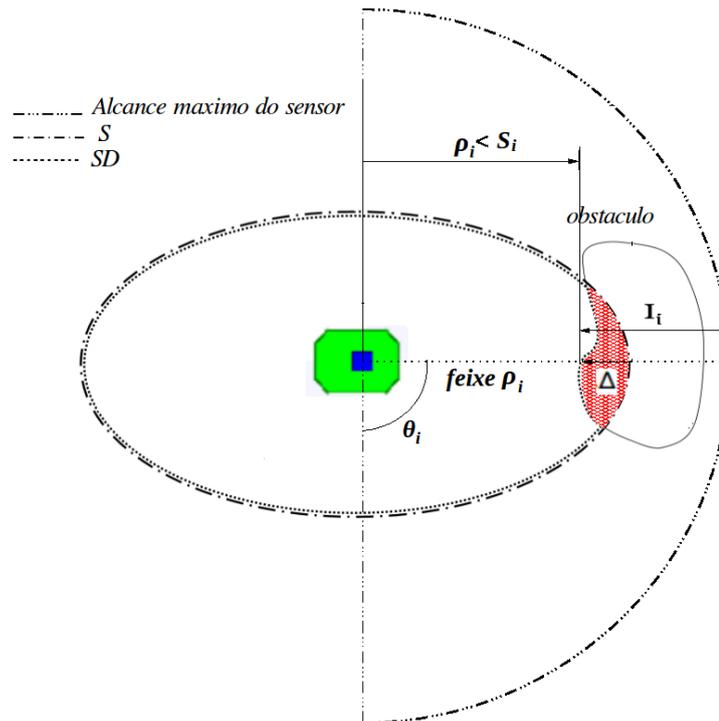


Figura 2.4: Características da Deformação da ZVD.

A *ZVD* pode retornar a sua forma não deformada pela ação de evasão do obstáculo, ou pelas ações de controle sobre a trajetória do robô, na tentativa de evitar uma colisão com o obstáculo detectado [21].

2.2 DISCUSSÃO

O uso da *ZVD* para evitar obstáculos possui características que a tornam um excelente método para navegação. Pela deformação da *ZVD*, o robô pode ter evidências da presença de obstáculos em toda a região a sua volta: caso seja necessário andar de ré, ou dar meia-volta, é possível evitar obstáculos que não estejam ainda na direção de avanço do robô.

No entanto, para que a *ZVD* deforme-se adequadamente e o robô identifique a presença de obstáculos, o vetor de Intrusão é de extrema importância. A relação entre os valores indicados pelos sensores e esse vetor definem a qualidade da deformação da *ZVD*, devido a presença de obstáculos.

Quando o sensor possui um campo de visão limitado, a *ZVD* não se deforma, pois o obstáculo não é detectado. No entanto, a última posição do obstáculo e a movimentação do robô podem ser indícios importantes de presença do obstáculos fora do campo de visão. Nessa situação, a detecção de obstáculos tem algumas semelhanças com o rastreamento de um alvo, e uma das técnicas mais utilizadas para rastreamento, com Filtro Bayesianos, é apresentada no próximo capítulo.

3 FILTRAGEM BAYESIANA

O problema do rastreamento de um alvo envolve não somente a determinação do seu caminho percorrido, mas também na previsão da sua próxima posição. Os filtros bayesianos têm como objetivo estimar o valor de uma variável no tempo, e são frequentemente utilizados para o rastreamento de alvos, apesar da sua grande complexidade computacional. O Filtro de Partículas (FP) é um tipo de filtro bayesiano e é também uma solução alternativa para problemas não lineares com ruídos não gaussianos. O FP será explicado mais adiante ainda neste capítulo, e será utilizado para a construção de uma Função de Distribuição de Probabilidade (PDF) baseada em toda a informação disponível, incluindo as últimas medidas recebidas dos sensores.

A primeira seção deste capítulo descreve os conceitos de maior relevância em probabilidade. Na seção 3.2 apresenta-se o Filtro de Bayes, tópico fundamental no conceito de filtragem estocástica. Na seção 3.3 explica-se o Filtro de Partículas com suas variantes na tentativa de obter a estimativa do obstáculo. Na seção 3.4.2 descreve-se em detalhes os principais conceitos do modelo de sensor utilizado no filtro de partículas proposto; são explicados os tipos de erros que podem se apresentar na hora da medida. Na seção 3.5 apresenta-se uma breve discussão do capítulo.

3.1 CONCEITOS BÁSICOS EM PROBABILIDADE

De acordo com [22], dependendo do fenômeno que está sendo estudado, seu modelo matemático pode ser descrito em dois tipos:

- **Modelo determinístico**, onde é possível determinar as variáveis de saída conhecendo as variáveis de entrada, e as condições nas quais o experimento foi realizado. Para os fenômenos determinísticos existe a certeza do valor da saída resultante. Na física clássica, a maioria dos fenômenos estudados são determinísticos.
- **Modelo probabilístico**, onde dadas as condições do experimento não é possível determinar o resultado final de forma exata. Neste modelo, introduz-se um componente ou variável aleatória.

Devido ao comportamento aleatório dos dados, neste trabalho empregar-se-á um modelo probabilístico. A seguir, alguns dos conceitos básicos relacionados ao uso de modelos probabilísticos serão apresentados brevemente. Neste capítulo a variável x utilizada não está diretamente relacionada com as variáveis do problema desenvolvido na metodologia deste trabalho.

3.1.1 Variável aleatória:

Sendo Ω o espaço amostral associado a um experimento, e sendo X uma função que se associe a cada elemento e que pertença a Ω , um número real $X(e)$ é denominado variável aleatória [23], como ilustrado na Figura 3.1.

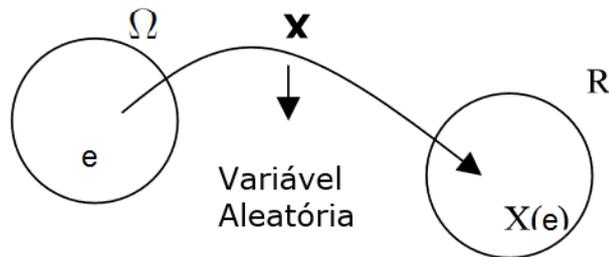


Figura 3.1: Variável aleatória.

Variáveis aleatórias são funções que transformam um espaço amostral qualquer em um espaço amostral numérico, que é um subconjunto dos números reais. As variáveis aleatórias podem ser classificadas como discretas ou contínuas [23]. São discretas todas as variáveis cujo espaço amostral é enumerável, finito ou infinito, ou seja, são um subconjunto dos inteiros. Já as variáveis aleatórias são contínuas quando o espaço amostral é infinito e não enumerável, ou seja, pode assumir qualquer valor em um intervalo $[a;b]$.

3.1.2 Função densidade de probabilidade

Uma função associa cada possível valor da variável aleatória com um valor real que representa a probabilidade da variável aleatória assumir esse possível valor: essa função é chamada *Função Densidade de Probabilidade* (PDF)[23], também pode ser descrita como a função matemática que caracteriza o possível comportamento de uma conjunto de dados. A PDF é a função que associa cada ponto do espaço amostral de uma variável aleatória, a probabilidade de ocorrência de tais pontos.

3.1.3 Distribuição normal ou distribuição de Gauss

Este tipo de distribuição é muito importante na estatística e uma das mais utilizadas, tanto em estudos teóricos quanto em modelos práticos [23]. Nesta distribuição, a maior parte das observações situam-se em torno da média, também chamada de *centro da distribuição*, e a presença de observações vai diminuindo de forma gradual e simétrica na direção dos extremos. Sua forma é definida por dois parâmetros: a média μ , que determina o centro da distribuição onde coincidem também a moda e a mediana; e a variância σ^2 , que determina a dispersão da distribuição. Os pontos de inflexão, onde a curva passa de convexa a côncava, são, portanto, definidos como $\mu - \sigma$ e $\mu + \sigma$. A distribuição normal é definida pelo uso da

notação $N(\mu, \sigma^2)$ [23]. No caso de uma única dimensão, a função é definida por:

$$p(x) = \frac{1}{2\pi\sigma^2} e^{\left\{-\frac{1}{2} \frac{(x-\mu)^2}{\sigma^2}\right\}}, \quad (3.1)$$

onde $-\infty < x < \infty$, $-\infty < \mu < \infty$ e $\sigma > 0$. Apesar do seu uso corrente, a distribuição normal é unimodal e não permite que múltiplas hipóteses possam ser expressadas por meio de seus parâmetros.

3.1.4 Probabilidade Condicional

Sendo A e B eventos do espaço amostral Ω , eles podem estar relacionados, no sentido de que o conhecimento sobre a ocorrência de um altera a probabilidade de ocorrência do outro. Portanto, a probabilidade do evento A ocorrer, dado que o evento B ocorreu, de acordo com [22] é denotada por:

$$p(A|B) = \frac{p(A \cap B)}{p(B)} \text{ se } p(B) > 0, \quad (3.2)$$

onde $p(A|B)$ é igual a probabilidade simultânea de A e B dividida pela probabilidade de ocorrência de B .

3.1.5 Probabilidade Conjunta

A distribuição da probabilidade de ocorrência simultânea de dois ou mais eventos é chamada Probabilidade Conjunta [22]. Tendo os eventos A e B , a probabilidade conjunta desses eventos é definida como $p(A \cap B)$. Se os eventos A e B são independentes, é possível definir a probabilidade conjunta como:

$$p(A \cap B) = p(A) p(B). \quad (3.3)$$

De acordo com [22], um evento A é considerado independente de outro evento B se a probabilidade de A é igual à probabilidade condicional de A dado B .

3.1.6 Probabilidade Total

De acordo com [22], sendo B_1, B_2, \dots, B_n eventos mutuamente exclusivos, e a união o espaço amostral Ω onde:

$$\Omega = \sum_{i=1}^n B_i, \quad (3.4)$$

e sendo A um evento qualquer em Ω , onde $Y = (Y \cap \Omega)$, como ilustrado na figura 3.2.

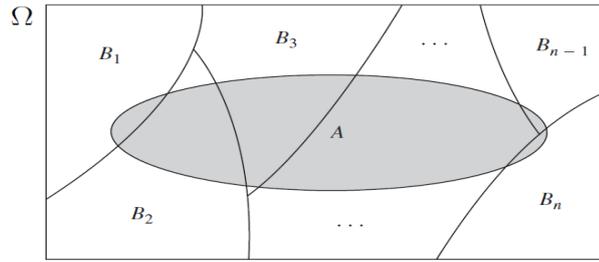


Figura 3.2: Variável aleatória (retirada de [22]).

A probabilidade total do evento A é igual a soma das probabilidades da interseção de A com cada um dos componentes do espaço amostral Ω , e é denotada como:

$$p(A) = \sum_{i=1}^n p(B_i)p(A|B_i). \quad (3.5)$$

3.1.7 Função Distribuição Acumulativa

De acordo com [22], sendo X uma variável aleatória discreta e Ω_x o seu espaço amostral, a função de distribuição, definida por $F(x)$ ou $p(X \leq x)$, é uma função que associa a cada valor de X a probabilidade $p(X \leq x)$, denotada como:

$$F(x) = P(X \leq x), \quad (3.6)$$

a função de distribuição de $F(x)$ é uma probabilidade, então satisfaz :

$$F(-\infty) = 0,$$

$$F(\infty) = 1,$$

$$0 \leq F(x) \leq 1,$$

$$\text{se } x_1 \leq x_2, \text{ então } F(x_1) \leq F(x_2).$$

3.1.8 Regra de Bayes

O teorema de Bayes permite que a distribuição da probabilidade inicial (também chamada *a priori*) de um dado evento seja atualizada à luz de novas evidências sobre um outro evento que apresenta relação de dependência com o primeiro [24]. Em outras palavras, permite calcular a probabilidade de ter acontecido o evento X_i , dado que uma de suas consequências Y_i foi observada. Matematicamente, temos:

$$p(X_i|Y) = \frac{p(X_i)p(Y|X_i)}{\sum_{i=1}^n p(X_i)p(Y|X_i)}, \quad (3.7)$$

onde $p(X_i|Y)$ é a probabilidade final (ou *a posteriori*) do evento X_i . O denominador da equação (3.7) é chamado fator de normalização, e garante que $\sum_{i=1}^N p(X_i|Y)$ seja igual a um. O termo $p(Y|X_i)$ é a probabilidade de *verossimilhança*, enquanto que $p(X_i)$ é designada como probabilidade *a priori*, ou seja, a probabilidade de X antes de observar qualquer valor de Y [24].

3.1.9 Cadeia de Markov

- **Processo Estocástico:** é uma coleção de variáveis aleatórias $X(t)$ indexadas por um parâmetro t que pertence a um conjunto T , frequentemente composto de números inteiros não negativos. $X(t)$ representa uma característica mensurável de interesse em um instante de tempo t [22]. Os valores assumidos pela variável aleatória $X(t)$ são chamados *estados*, e o conjunto de todos seus possíveis valores forma o espaço de estados do processo.
- **Processo Markoviano:** um processo estocástico $X(t)$ com $t \in T$, é um processo de Markov quando, para qualquer tempo $t_0 < t_1 < \dots < t_{n-1} < t_n$, a distribuição condicional de X_{t_n} para os valores dados de $X_{t_0}, X_{t_1}, \dots, X_{t_{n-1}}$ depende somente de $X_{t_{n-1}}$.

$$p(X_{t_n} = x | X_{t_{n-1}} = x_{n-1}, X_{t_{n-2}} = x_{n-2}, \dots, X_{t_0} = x_0) = p(X_{t_n} = x | X_{t_{n-1}} = x_{n-1}). \quad (3.8)$$

Na equação (3.8), t_0, t_1, \dots, t_{n-2} representam o passado, t_{n-1} é o instante de tempo atual, e t_n é o instante de tempo futuro imediato. A mesma equação (3.8) pode ser interpretada como: nos Processos de Markov, a distribuição da probabilidade sobre os valores dos estados no futuro próximo depende apenas dos valores dos estados no instante de tempo atual, e não dos valores passados [22].

- **Cadeia de Markov:** um processo de Markov é considerado uma cadeia de Markov quando as variáveis aleatórias $X(t)$ estão definidas em um espaço de estados discretos. Este processo estocástico tem a propriedade de que as probabilidades associadas com o processo num instante qualquer do futuro dependem somente do estado presente, portanto, são independentes dos eventos no passado. Desse modo, esse tipo de processo estocástico é também denominado processo sem memória (*memoryless process*), uma vez que o passado é esquecido.

3.2 FILTRAGEM ESTOCÁSTICA

Os processos ou sistemas estocásticos podem ser utilizados para descrever muitos fenômenos governados por leis físicas mas cujos estados em um tempo futuro são impossíveis

de determinar, mesmo quando suas condições iniciais são conhecidas. Contudo, é possível simular as incertezas que estão intrinsicamente associadas às medições desses fenômenos.

3.2.1 Filtro de Bayes

Na robótica móvel, é comum empregar um filtro de Bayes para estimar a densidade da probabilidade bel (geralmente chamada de *valor esperado*) do estado x no tempo t , definida como $bel(x_t)$, a partir de $bel(x_{t-1})$ no tempo $t - 1$. O valor esperado indica o grau de conhecimento sobre o estado de um sistema, e é representada pela probabilidade condicional:

$$bel(x_t) = p(x_t | z_t, u_{t-1}, z_{t-1}, u_{t-2}, \dots, z_0), \quad (3.9)$$

onde z_t corresponde às medições no instante de tempo t , e u_t são as ações aplicadas ao robô no instante de tempo t . É possível observar que $bel(x_t)$ depende de todo o passado observado. Considerando que o processo é Markoviano e manipulando algebricamente, é possível calcular $bel(x_t)$ recursivamente, de acordo com [25], como mostrado no algoritmo 1.

Algoritmo 1: Algoritmo do Filtro de Bayes

Entrada: $bel(x_{t-1}), u_t, z_t$

Saída: $bel(x_t)$

1 **início**

2 Para todo x_t fazer: **início**

3 $\bar{bel}(x_t) = \int p(x_t | u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1};$

4 $bel(x_t) = \eta p(z_t | x_t) \bar{bel}(x_t)$

5 **fim**

6 **fim**

O filtro de Bayes se baseia na regra de Bayes descrita na equação (3.7), e permite calcular a probabilidade *a posteriori* conhecendo a probabilidade *a priori*. No algoritmo 1, é possível observar que ele possui dois passos essenciais: a *predição* consiste no cálculo de $\bar{bel}(x_t)$ e utiliza o modelo de movimento do robô e suas ações de controle. Já a *atualização* utiliza a predição $\bar{bel}(x_t)$ e o modelo de observação (ou modelo do sensor), bem como uma variável de normalização η [25].

3.3 FILTRO DE PARTÍCULAS

O Filtro de Partículas é um método sequencial de Monte Carlo [26], que representa a probabilidade *a posteriori* por intermédio de um conjunto de amostras no espaço de estados e seus respectivos pesos. Várias propostas similares de algoritmos sequenciais de Monte

Carlo são citadas na literatura, como Algoritmo de Condensação [27], o Filtro Bootstrap [28] [29], e o Filtro de Monte Carlo ou a Amostragem Sequencial por Importância (do inglês *Sequential Importance Resampling*) [30].

Assim como o Filtro de Bayes, este algoritmo é recursivo e calculado em duas etapas: a *predição* e a *atualização*. Cada amostra desse conjunto é denominada de *partícula* e as partículas são estados possíveis do processo, representados como pontos no espaço de estados do processo. Um conjunto de N_s partículas em um filtro é definido por:

$$X_t = x_t^1, x_t^2, x_t^3, \dots, x_t^{N_s}, \quad (3.10)$$

onde cada partícula $x_t^{N_s}$ é uma hipótese do estado do mundo real no tempo t .

A densidade de probabilidade $p(X_t|z_t, U_{t-1})$ é equivalente ao valor esperado $bel(x_t)$ no filtro de Bayes (veja algoritmo 1) e representa o estado do robô no instante t , onde $Z_t = \{z_1, z_2, \dots, z_t\}$ são as observações feitas pelos sensores e $U_{t-1} = \{u_1, u_2, \dots, u_{t-1}\}$ são as ações de controle realizadas.

A seguir, as etapas de inicialização, predição e atualização no filtro de partículas são descritas em maiores detalhes.

3.3.1 Inicialização

A etapa de inicialização do filtro de partículas é executada unicamente no instante $t = 0$. Nesse instante, o conjunto de partículas é escolhido a partir da distribuição *a priori*. Quando essa informação não está disponível, as partículas são distribuídas aleatoriamente utilizando uma distribuição uniforme. Para cada uma dessas partículas é atribuído um peso inicial w_0 , que vai ser um valor normalizado entre $[0, 1]$, calculado a partir da equação (3.11):

$$w_0 = 1/N_s, \quad (3.11)$$

onde N_s é o número de partículas.

Em um instante t qualquer, o filtro de partículas se inicia com um conjunto de amostras x_{t-1}, w_{t-1}^i que caracteriza a PDF *a posteriori* $p(X_{t-1}|Z_{t-1})$ no instante $t - 1$, de forma que:

$$X_{t-1} \sim p(X_{t-1}|Z_{t-1}). \quad (3.12)$$

Resumindo: o espaço de estados é particionado e as partículas são ponderadas mediante a probabilidade da medida obtida ($p(X_0)$ ou $p(X_{t-1}|Z_{t-1})$).

3.3.2 Predição

Nesta etapa, um conjunto de novas partículas é gerado aplicando o *modelo de movimento*, $p(X_t|U_t, X_{t-1})$. Na robótica móvel, a predição é utilizada para prever a posição do robô no instante atual, a partir da função densidade de probabilidade da posição do robô no instante anterior $t - 1$, e da atuação sobre o robô. Essas novas partículas representam a predição da variável de estado:

$$X_t^{N_s} \sim p(X_t|U_t, X_{t-1}^{N_s}), \quad (3.13)$$

onde U_t são as ações aplicadas ao robô no instante de tempo t .

3.3.3 Atualização

Ao obter medidas dos sensores, um peso $w_t^{N_s}$ (também chamado de *fator de importância*) é associado a cada partícula por meio do modelo de observação ou modelo do sensor [31]. Na etapa de atualização, as partículas são ponderadas de acordo com sua proximidade da percepção obtida pelos dados dos sensores.

A probabilidade *a posteriori* do Filtro de Bayes, representada pela equação da atualização utilizada no Algoritmo 1, pode ser aproximada no filtro de partículas por:

$$p(Z_t|X_t) = \sum_{i=1}^{N_s} w_t^i \delta(X_t - X_t^i), \quad (3.14)$$

onde $w_t^{[i]}$ é o peso da partícula i , e a expressão $\delta(X_t - X_t^i)$ representa a probabilidade do estado da amostra i na distribuição do modelo do sensor quando os valores das medidas dos sensores são conhecidos [26], ou seja:

$$\delta(X_t - X_t^i) \sim p(X_t|X_t^i, Z_t). \quad (3.15)$$

A teoria de reamostragem por importância, apresentada em [30], permite que cada amostra X_t^i e os pesos sejam apresentados pelas equações (3.16) e (3.17), onde as amostras são obtidas da *densidade de importância* ($q(X_{0:t}|U_t, Z_{1:t})$), e os pesos são definidos por:

$$X_t^i \sim q(X_{0:t}^i|Z_{1:t}), \quad (3.16)$$

$$w_t^i \sim \frac{p(X_{0:t}^i|Z_{1:t})}{q(X_{0:t}^i|Z_{1:t})}, \quad (3.17)$$

e a consequente atualização dos pesos é dada por:

$$w_t^i \sim w_{t-1}^i \frac{p(Z_t|X_t^i)p(X_t^i|X_{t-1}^i)}{q(X_t^i|X_{t-1}^i, Z_t)}. \quad (3.18)$$

O algoritmo 2, a seguir, resume o funcionamento do filtro de partículas. Inicialmente, o filtro realiza o cálculo da distribuição (3.12), conforme descrito no item 3.3.1. Em seguida, em um instante t , realiza-se uma representação da função de distribuição da linha 4, expressa como x_t^i com seus pesos correspondentes w_t^i . Para representar as N_s amostras da distribuição proposta, seus pesos são normalizados na linha 8.

Algoritmo 2: Algoritmo Filtro de Partículas Genérico

Entrada: $[\{X_{t-1}^i, w_{t-1}^i\}_{i=1}^{N_s}, Z_t]$

Saída: $[\{X_t^i, w_t^i\}_{i=1}^{N_s}]$

1 **início**

2 **para** $i = 1 : N_s$ **faça**

3 Obtenha $X_t^i \sim p(X_t^i|Z_{t-1}^i)$

4 Atribua à partícula um peso w_t^i , de acordo com (3.18).

5 Calcule o peso total $\Upsilon = \sum_{i=1}^{N_s} w_t^i$

6 **fim para**

7 **para** $i = 1 : N_s$ **faça**

8 Normalize $w_t^i = \frac{w_t^i}{\Upsilon}$

9 **fim para**

10 **fim**

3.3.4 Reamostragem

A ideia central da reamostragem consiste em eliminar as partículas que têm um peso pequeno e dar importância às partículas com pesos maiores [32]. A perda de peso das partículas faz com que a contribuição na PDF seja cada vez menor.

Uma das desvantagens do filtro de partículas consiste na redução da importância dos pesos com o transcorrer do tempo: após algumas iterações, poucas partículas terão peso diferente de zero. Este fenômeno é chamado de degeneração dos pesos ou empobrecimento da amostra (em inglês *weight degeneracy* ou *sample impoverishment*) [26].

Variantes do filtro de partículas têm sido propostas para combater esse fenômeno. Em [33], as variantes resultantes foram chamadas, em inglês, de *Regularized Particle Filters (RPF)*: propõem-se que a etapa de reamostragem seja baseada em uma aproximação contínua da densidade *a posteriori*, o que, teoricamente, resultaria em um filtro mais estável no tempo.

Uma forma de medir a degeneração dos pesos é calcular o tamanho efetivo da amostra,

\hat{N}_{eff} (em inglês *Effective Sample Size* ou ESS) [34], dado por:

$$N_{eff} = \frac{N_s}{1 + Var(w_t^{*i})}, \quad (3.19)$$

onde $Var(w_k^{*i})$ é conhecida como a variância do *peso real* w_k^{*i} (em inglês *true weight*), que por sua vez é dado por:

$$w_t^{*i} = \frac{p(X_t^i | Z_{1:t})}{q(X_t^i | X_{t-1}^i, Z_t)}. \quad (3.20)$$

Na prática, este valor não pode ser calculado, mas uma aproximação pode ser obtida por:

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_s} (w_t^i)^2}, \quad (3.21)$$

onde o peso w_k^i é calculado conforme a equação (3.18). Essa aproximação é feita durante a etapa de *reamostragem*, cujo objetivo é eliminar as partículas com pequeno peso e dar maior importância as partículas com pesos maiores [35].

Quando a variância dos pesos é grande, o tamanho da amostra \hat{N}_{eff} é pequeno, indicando a necessidade de reamostragem. Porém, se por outro lado a variância é pequena, o valor de \hat{N}_{eff} é grande e a reamostragem não é necessária.

Algoritmo 3: Algoritmo de Reamostragem

Entrada: $[\{X_t^{j*}, w_t^j, i^j\}_{j=1}^{N_s}]$ **Saída:** $[\{X_t^i, w_t^i\}_{i=1}^{N_s}]$

```
1 início
2   Inicialização da Função Densidade Acumulativa (CDF)  $c_1 = 0$ 
3   para  $i = 2 : N_s$  faça
4     | obtenção da CDF:  $c_i = c_{i-1} + w_t^i$ 
5   fim para
6   Início na parte inferior da CDF:  $i = 1$ 
7   Desenho do ponto de início  $\lambda_1 \sim \Lambda[0, N_s^{-1}]$ 
8   para  $j = 1 : N_s$  faça
9     | Percorra a PDF:  $\lambda_j = \lambda_1 + N_s^{-1}(j - 1)$ 
10    | enquanto  $\lambda_j > c_i$  faça
11      |  $i = i + 1$ 
12    | fim enqto
13    | Atribua as amostras  $X_t^{j*} = X_t^i$ 
14    | Atribua os pesos  $w_t^j = N_s^{-1}$ 
15    | Atribua os parentes  $i^j = i$ 
16  | fim para
17 fim
```

O algoritmo 3 utiliza uma mínima variância, onde usa-se uma distribuição uniforme $\Lambda[0, N_s^{-1}]$ no intervalo $[0, N_s^{-1}]$, onde cada valor λ_j é sorteado a partir de $\lambda_j = \lambda_1 + N_s^{-1}(j - 1)$. As partículas com maior peso fornecerão mais cópias que aquelas que tiverem menor peso. Na linha 4, o algoritmo gera uma PDF de números aleatórios uniformemente ordenados. A partir da PDF, nas linhas 6 e 7, utiliza-se para selecionar o novo conjunto de partículas, denotado por $\{X_t^{j*}\}_{j=1}^{N_s}$, reamostrando através de substituição.

Estas novas partículas representam uma função discretizada, dada por:

$$p(X_t|Z_{1:t}) = \sum_{i=1}^{N_s} w_t^i \delta(X_t - X_t^i). \quad (3.22)$$

As novas partículas ou amostras serão, então, novamente independentes e identicamente distribuídas (i.i.d) a partir da distribuição da equação (3.22), com peso uniforme $w = 1/N_s$. A reamostragem traduz-se em um maior conjunto de amostras em torno das probabilidades mais altas, o que levaria, por exemplo, a uma melhor precisão na estimativa da posição de um robô.

3.3.5 O Algoritmo do Filtro de Partículas com Reamostragem

O algoritmo 4 mostra o Filtro de Partículas com reamostragem, conforme as partes básicas apresentadas no decorrer deste capítulo.

Algoritmo 4: Algoritmo Filtro de Partículas com Reamostragem

Entrada: $[\{X_{t-1}^i, w_{t-1}^i\}_{i=1}^{N_s}, Z_t]$

Saída: $[\{X_t^i, w_t^i\}_{i=1}^{N_s}]$

1 **início**

2 **para** $i = 1 : N_s$ **faça**

3 Obtenha $X_t^i \sim p(X_k | Z_{t-1}^i)$

4 Atribua à partícula um peso w_t^i , de acordo com (3.18).

5 Calcule o peso total $\Upsilon = \sum_{i=1}^{N_s} w_t^i$

6 **fim para**

7 **para** $i = 1 : N_s$ **faça**

8 Normalize $w_t^i = \frac{w_t^i}{\Upsilon}$

9 **fim para**

10 Calcule \hat{N}_{eff} usando a equação (3.21).

11 **se** $\hat{N}_{eff} < N_T$ **então**

12 Reamostragem empregando o algoritmo 3

13 **fim se**

14 **fim**

No algoritmo 4, um conjunto de amostras é inicialmente gerado, na linha 3, a partir da função de probabilidade $p(X_t | Z_{t-1}^i)$. Em seguida, o conjunto de pesos das amostras, w_t^i , é calculado na linha 4 e, posteriormente, normalizado. Após a normalização, um limiar predefinido N_T para N_{eff} for verificado na linha 11: a reamostragem é executada somente se N_{eff} é inferior à N_T [36]. Vale lembrar que, cada vez que o algoritmo de reamostragem (algoritmo 3) é aplicado, os pesos se tornam iguais: $w = 1/N_s$. É importante ressaltar que a execução deste processo produz um acréscimo no custo computacional mas não muda a classe do algoritmo: a complexidade computacional do processo de reamostragem é $O(n)$

3.4 MODELO DE OBSERVAÇÃO

Um robô móvel, mesmo quando orientado na direção de uma posição alvo, requer informações atualizadas do ambiente no qual se encontra de forma a evitar colisões com obstáculos fixos ou móveis que estejam no seu caminho. Por isso, é preciso implementar um sistema de percepção que proporcione ao robô a capacidade de se movimentar de forma segura em

um ambiente dinâmico.

3.4.1 Sensor Laser

Para obter as medidas de distância entre o robô e os obstáculos na *ZVD*, utiliza-se um sensor *laser*. Esse sensor fornece a medida da distância através da medição do tempo de voo dos pulsos dos feixes de *laser* emitidos: se um objeto for encontrado, o feixe refletido é detectado e o tempo transcorrido entre a emissão e a detecção indica a distância do objeto ao sensor.

As características do *laser* permitem diferentes configurações, com variações em parâmetros como o campo de visão e o intervalo de leituras [37]. Neste caso, o sensor *laser* realiza uma varredura em um campo de 0 a 180 graus em forma de abanador, como exibido na Figura 3.3, fornecendo 181 ou 361 leituras (leituras a cada 1 grau ou a cada meio grau, dependendo da configuração utilizada, feita através de software). A distância máxima que pode ser obtida do *laser* é, normalmente, 8 metros em ambientes fechados.

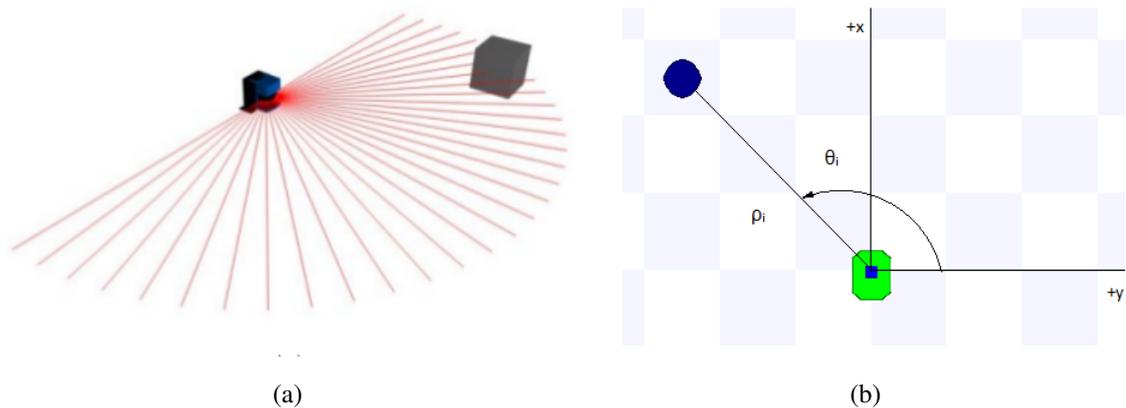


Figura 3.3: a) Varredura realizada pelo sensor *laser*. b) Sistema de coordenadas da leitura do sensor.

3.4.2 Modelo de Sensor

O modelo de sensor proporciona uma estimativa da função densidade de probabilidade $p(\mathbf{Z}|\mathbf{X}, \phi)$ que, aplicando a teoria Bayesiana, mede a verossimilhança entre as leituras do sensor $\mathbf{Z}_t = \{z_t^1, z_t^2, \dots, z_t^k\}$ e a posição do robô \mathbf{x} . O ϕ expressa os parâmetros do modelo do ambiente onde o robô está navegando. Este modelo é utilizado pela técnica de localização de Markov para obter uma aproximação da posição do robô em um ambiente não estático [38].

Assumindo que o modelo do entorno é fixo, a função de densidade se transforma em $p(\mathbf{Z}|\mathbf{X})$. Além disso, considera-se que cada sensor mede distâncias z^1, z^2, \dots, z^n , onde z^n é o valor máximo que o sensor pode alcançar.

Assim, o termo $p(\mathbf{Z}_t^k | \mathbf{X}_t)$ expressa a probabilidade de se medir uma distância Z_t^k , se o robô está localizado em X_t no tempo t . Para encontrar a função densidade de probabilidade correta para este termo em uma aplicação, o modelo do sensor típico considera quatro tipos de erros de medidas de acordo com o exposto em [25] e descritos a seguir de forma resumida.

3.4.3 Obstáculo Detectado

Quando o sensor detecta um obstáculo localizado a uma distância z_t^{k*} como descrito na figura 3.4, a função densidade de probabilidade da medida resultante desse sensor pode ser modelada como uma Gaussiana com média na distância do obstáculo:

$$p_{hit}(z_t^k | \mathbf{x}_t) = \begin{cases} \eta N(z_t^k, z_t^{k*}, \sigma_{hit}^2), & 0 \leq z_t^k \leq z_{max} \\ 0 & , \quad z_t^k > z_{max} \end{cases}, \quad (3.23)$$

onde

$$N(z_k, z_t^{k*}, \sigma_{hit}^2) = (2\pi\sigma^2)^{-\frac{1}{2}} e^{-\frac{1}{2} \frac{(z_t^k - z_t^{k*})^2}{\sigma^2}}, \quad (3.24)$$

η é uma constante de normalização definida por:

$$\eta = \frac{1}{\sum_{i=0}^{z_{max}} N(z_k, z_t^{k*}, \sigma_{hit}^2)}. \quad (3.25)$$

e o desvio padrão σ_{hit} é um parâmetro intrínseco das medidas do modelo, como é apresentado em [25]:

$$\sigma_{hit} = \sqrt{\frac{1}{|Z_{hit}|} \sum_{z_i \in Z_{hit}} (z_i - z_i^*)^2} \quad (3.26)$$

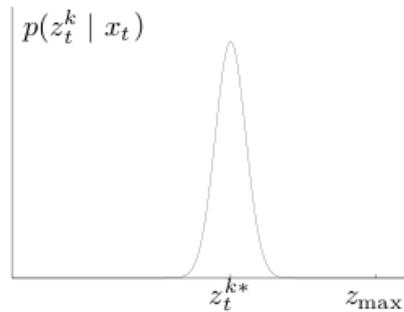


Figura 3.4: Probabilidade p_{hit} (retirada de [25]).

3.4.4 Obstáculo não esperado

Quando dois ou mais obstáculos estão no campo de visão do sensor, somente o mais próximo será detectado, ainda que o obstáculo de maior interesse seja o mais distante. Portanto, pode-se dizer que a detecção de obstáculos possui maior probabilidade quando a distância é menor. Essa situação é representada na figura 3.5 mediante distribuição exponencial, da seguinte forma:

$$p_{short}(z_t^k | \mathbf{x}_t) = \begin{cases} \eta \epsilon_{short} e^{-\epsilon_{short} z_t^k}, & 0 \leq z_t^k \leq z_t^{k*} \\ 0, & z_t^k > z_t^{k*} \end{cases}, \quad (3.27)$$

onde

$$\eta = \frac{1}{1 - e^{-\epsilon_{short} z_t^{k*}}}. \quad (3.28)$$

e

$$\epsilon_{short} = \frac{|Z_{short}|}{\sum_{z_i \in Z_{short}} z_i} \quad (3.29)$$

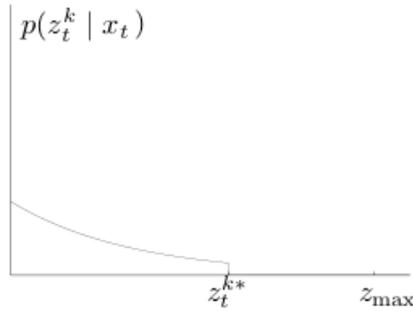


Figura 3.5: Probabilidade p_{short} (retirada de [25]).

3.4.5 Falhas do Sensor

Quando o sensor não detecta o obstáculo, ou quando este se encontra além do limite de detecção do sensor, ou ainda quando não existe obstáculo, a medida resultante do sensor será o seu valor máximo de distância, apresentado na figura 3.6. Para todos esses casos, o modelo utilizado é uma distribuição com ponto de massa centrada em z_{max} , que representa o valor máximo da medida.

$$p_{max}(z_t^k | \mathbf{x}_t) = \begin{cases} 1, & z = z_{max} \\ 0, & z \neq z_{max} \end{cases}, \quad (3.30)$$

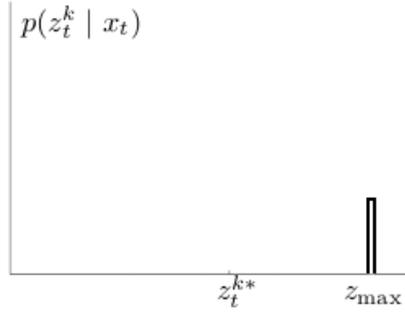


Figura 3.6: Probabilidade p_{max} (retirada de [25]).

3.4.6 Medição Aleatória

Representa as medidas com erros de causas desconhecidas ou desconsideradas como relevantes. Esse tipo de erro é modelado como uma distribuição uniforme por todo o intervalo de medida do sensor, como na figura 3.7, e calculado na equação (3.31):

$$p_{rand}(z_t^k | \mathbf{x}_t) = \begin{cases} \frac{1}{z_{max}}, & \text{se } 0 \leq z^k < z_{max} \\ 0, & z^k \neq z_{max} \end{cases}, \quad (3.31)$$

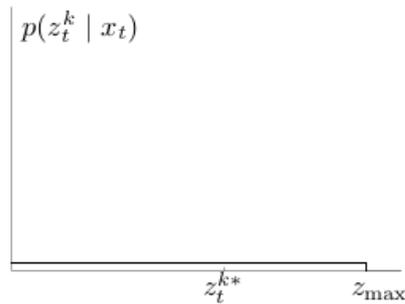


Figura 3.7: Probabilidade p_{rand} (retirada de [25]).

3.4.7 Junção de Densidades de Probabilidade

Depois de modelar os erros que podem influenciar a medida z_t^{k*} , uma função densidade de probabilidade final é definida como sendo a soma ponderada de k componentes de distribuições [39] [40] [41].

$$p(z_t^{k*} | \mathbf{x}_t) = \begin{cases} w_{hit} \cdot p_{hit}(z_t^k | \mathbf{x}_t) + \\ w_{short} \cdot p_{short}(z_t^k | \mathbf{x}_t) + \\ w_{max} \cdot p_{max}(z_t^k | \mathbf{x}_t) + \\ w_{rand} \cdot p_{rand}(z_t^k | \mathbf{x}_t). \end{cases} \quad (3.32)$$

A fusão das informações contidas nas k medidas do sensor é feita com a multiplicação das probabilidades, como dado pela equação (3.33), e posteriormente ela é normalizada. Assume-se que há independência condicional entre as medidas individuais.

$$\mathbf{P}(z_t^k | \mathbf{x}_t) \approx \prod_{j=1}^k p(z_t^{k*} | \mathbf{x}_t). \quad (3.33)$$

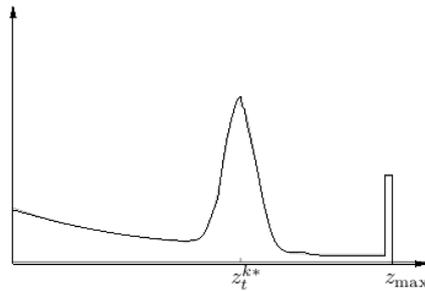


Figura 3.8: Junção das Distribuições de Probabilidade que modelam erros nas medidas do sensor (retirada de [25]).

Este modelo de observação objetiva inclui possíveis erros na medida e ruídos que fornecem incerteza à medida do sensor. A Figura 3.8 ilustra o resultado da mistura das distribuições para uma medida z_t^{k*} .

3.5 DISCUSSÃO

A técnica de Filtro de Partículas pode ser utilizada para rastrear a presença de um obstáculo dentro da *ZVD*. Além da sua capacidade de atualizar as informações sobre a posição do obstáculo a partir dos dados sensores, o filtro também pode estimar essas informações a partir da última posição observada e do movimento do robô.

Modelar um sensor com precisão pode ser uma tarefa árdua principalmente pelos complexos fenômenos físicos que se apresentam na medição. As medições feitas pelos sensores *laser* são muito robustas, no tocante à interferências externas tais como: luminosidade, variações de temperatura, e umidade. Interferências que, em alguns casos, são extremamente prejudiciais no desempenho de alguns tipos de sensores. A partir das considerações expostas neste capítulo, o modelo de observação tentará proporcionar as informações mais próximas do ambiente para gerar dados na atualização das partículas tanto dentro da *ZVD* quanto fora.

Depois de ter os conceitos básicos e as ferramentas para desenvolver o projeto, é preciso relacionar todo o conteúdo em um só, e para isso, no próximo capítulo, será descrita a metodologia empregada neste trabalho.

4 PROPOSTA E METODOLOGIA

O principal objetivo deste trabalho é fazer um robô móvel navegar de forma segura em um ambiente desconhecido e com obstáculos.

O sistema de controle proposto se baseia na definição de uma zona de proteção deformável ao redor do robô. Essa zona de proteção, chamada Zona Virtual Deformável, se baseia na interação robô-ambiente sem nenhum modelo prévio dos obstáculos. O método proposto consiste na estimativa das deformações na *ZVD*, devido a presença de obstáculos, e no controle que tenta minimizá-las.

Das propostas que abrangem os métodos reativos, uma das mais populares é a técnica baseada em campos potenciais, ainda que possua a dificuldade de desenhar um potencial sem mínimos locais indesejados [17]. A dificuldade dos mínimos locais pode ser solucionada modelando o campo potencial através das equações de Maxwell, mas com a condição de ter um conhecimento *a priori* do ambiente. Outras variações como o Campo de Forças Vetoriais (VFF) e o Histograma de Campo Vetorial (VFH) apresentam algumas desvantagens, como o volume de informação com que trabalham estes métodos requer uma grande capacidade de armazenamento de informação no robô. No caso da banda elástica, o robô móvel deve permanecer estacionário por um período de tempo, capturando uma imagem panorâmica. Outros detalhes são apresentados em [42] e [43].

Propõem-se uma extensão do método da *ZVD* para o caso de um robô provido de somente um sensor *laser* de varredura e, portanto, com regiões dentro da zona sem alcance do sensor. O método da *ZVD*, nesta proposta, garante a aproximação ideal para contornar os obstáculos. Para garantir a robustez do sistema, propõem-se a integração de um filtro de partículas à *ZVD* para que a estimativa da posição do obstáculo dentro da *ZVD* continue a ser calculada mesmo quando este está fora do alcance do sensor. O filtro é utilizado nesta proposta porque garante o rastreamento da presença do obstáculo dentro da *ZVD*, gerando uma possível localização a partir de hipóteses (amostras).

A técnica de Filtro a Partícula pode ser utilizada para rastrear a presença de um obstáculo dentro da *ZVD*. Além da sua capacidade de atualizar as informações sobre a posição do obstáculo a partir dos dados sensores, o filtro também pode estimar essas informações a partir da última posição observada e do movimento do robô.

O desenvolvimento desta proposta consistiu nas etapas de projeto, implementação e testes em simulação. Foram considerados aspectos teóricos e práticos das áreas da robótica móvel, Zona Virtual Deformável, dos modelos de filtros estocásticos, e do filtro de partículas.

A Zona Virtual Deformável (*ZVD*) foi definida na forma de elipse, e foi preciso estabelecer sua forma não deformada e seus limites, assim como a posição que a *ZVD* terá em

relação ao robô móvel, e como ela será deformada em função dos obstáculos identificados. Maiores detalhes sobre a *ZVD* encontram-se na seção 4.1.

A utilização de um filtro para estimar a posição do obstáculo requer o uso de um modelo de observação e de um modelo de movimentação do robô, que foi construído a partir da utilização do sensor Laser Sick [37], embarcado no robô Pioneer 3AT [44]. O modelo de observação é descrito detalhadamente na seção 4.2.

Já o modelo de movimentação do robô assume a hipótese de que o movimento do robô ocorre em um período de tempo suficientemente pequeno entre um ponto e outro da sua trajetória, e que ele percorre um arco de circunferência. Também considera-se que o movimento do robô seja inicialmente registrado em coordenadas polares em um sistema de referência com origem no obstáculo. A seção 4.3 mostra detalhadamente o processo de construção e utilização do modelo de movimento no filtro de partículas.

A implementação do Filtro de Partículas deu-se, inicialmente com o algoritmo proposto em [26]. Uma segunda versão foi desenvolvida; nela, o algoritmo foi modificado para realizar as etapas de predição e de atualização. Finalmente, foi implementada uma última versão com a etapa de reamostragem. A seção 4.4 descreve o projeto e implementação do filtro de partículas utilizado. Na seção 4.7 apresenta-se uma breve discussão do capítulo.

4.1 MODELO DA ZONA VIRTUAL DEFORMÁVEL

A escolha da forma geométrica da *ZVD* é arbitrária, mas a forma elíptica geralmente é escolhida por [16] e [17], sendo esta forma também a utilizada neste trabalho. No centro de referência polar da elipse está localizado o sensor *laser* e abaixo dele o robô, como pode ser visto na figura 4.1. Levando em consideração que no presente trabalho o robô possui somente um sensor *laser* embarcado, com campo de visão de 180° a frente do robô móvel, a *ZVD* é dividida em duas partes:

- **Luz:** é a porção anterior da elipse, coberta pelo sensor *laser*; e
- **Umbra:** é a porção posterior da elipse, fora do alcance do sensor.

Assumindo que o sensor *laser* forneça n leituras no instante de tempo k , essas leituras são registradas no vetor LL_k . Cada leitura é composta por medidas de proximidade (distâncias dos obstáculos) $(\rho_1, \rho_2, \rho_3, \dots, \rho_n)$, ao longo de ângulos igualmente espaçados $(\theta_1, \theta_2, \theta_3, \dots, \theta_n)$.

Considerando a descrição feita na seção 2.1.6, uma *Zona Virtual* sem deformações é representada pelos vetores na equação (4.1):

$$S = [S_0, S_1, \dots, S_n]^T, \quad (4.1)$$

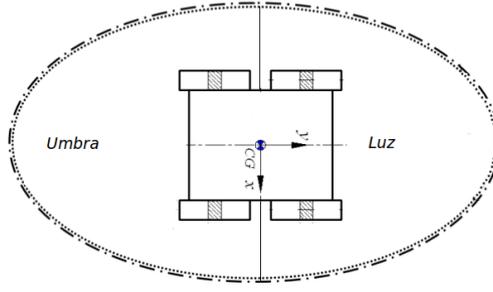


Figura 4.1: Zona Virtual Deformável: sua forma e localização em relação ao sistema cartesiano do robô.

onde cada componente S_i é um vetor que representa a distância entre o limite da ZVD e o centro de referência na origem do sistema de coordenadas fixo ao corpo do robô onde está localizado o sensor *laser*. A figura 2.4 ilustra uma situação típica, na qual o robô se movimenta pelo ambiente e um obstáculo se aproxima.

Quando um objeto ou obstáculo entra no alcance do sensor, as leituras resultantes introduzem informações da posição do obstáculo na ZVD na forma de variações nos componentes do vetor S e geram uma deformação. O vetor que representa a ZVD deformada é definido pela equação (4.2):

$$SD = \left[SD_0, SD_1, \dots, SD_n \right]^T, \quad (4.2)$$

onde cada componente SD_i representa a variação da medida fornecida pelo feixe do sensor *laser* nessa direção da deformação.

4.1.1 Deformação devido a um obstáculo

Cada leitura ρ_i , fornecida pelo sensor, identifica a distância do obstáculo detectado; seu valor máximo é d_{imax} , o limite do alcance do sensor. O *vetor de intrusão* I com componentes I_i , expressa a profundidade da intrusão daquela medida ρ_i , e é definido na equação (2.4):

$$I = \begin{bmatrix} I_1 \\ I_2 \\ \cdot \\ \cdot \\ \cdot \\ I_n \end{bmatrix} = \begin{bmatrix} d_{1max} - \rho_1 \\ d_{2max} - \rho_2 \\ \cdot \\ \cdot \\ \cdot \\ d_{nmax} - \rho_n \end{bmatrix}. \quad (4.3)$$

Também é definido o *vetor de deformação* Δ , que representa a deformação total da zona virtual, como mostrado na equação (2.5), e que possui as mesmas dimensões que o vetor I ,

$n \times 1$. Para definir esse vetor, deve-se levar em conta a distância S_i , que representa a distância de cada medida do sensor que resulta em uma zona virtual sem deformações, S .

Cada elemento Δ_i é definido pela função mostrada na equação (4.4):

$$\Delta_i = \begin{cases} 0 & , \quad \rho_i \geq S_i, \\ S_i - \rho_i, & \rho_i < S_i. \end{cases} \quad (4.4)$$

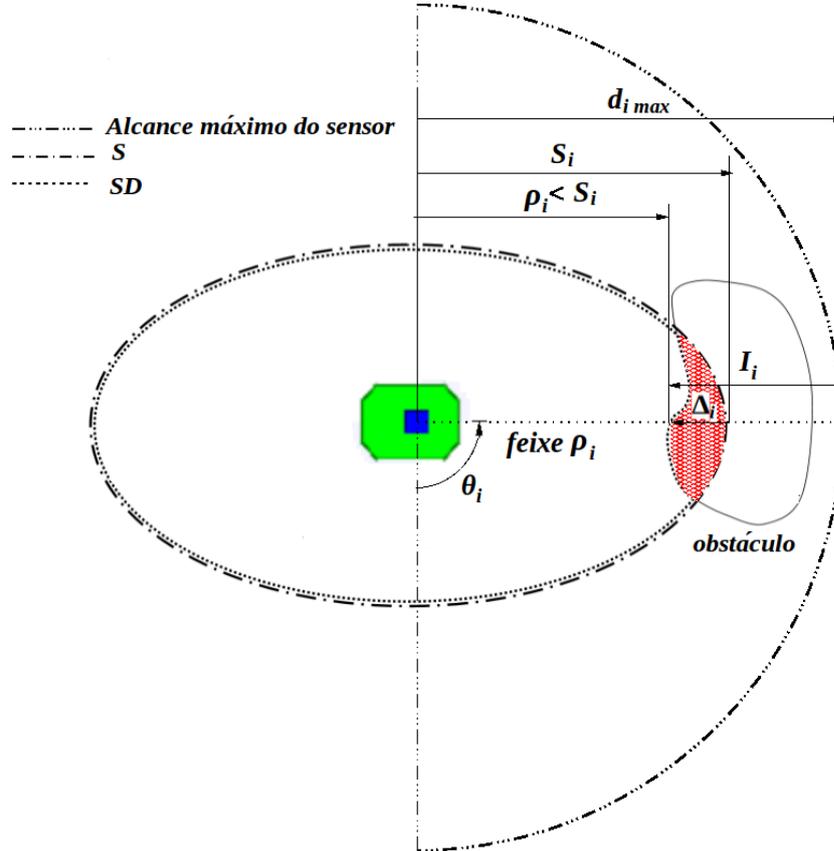


Figura 4.2: Variáveis envolvidas na deformação da ZVD devido a presença de um obstáculo: $d_{i\max}$ é a medida máxima do sensor, no limite do seu alcance; S_i é o valor máximo da medida do sensor que mantém a ZVD intacta S ; θ_i é o ângulo da medida do sensor; I é o vetor de intrusão; quando $\rho_i < S_i$, o vetor Δ_i representa a deformação que altera a forma da ZVD para S_i .

A Figura 4.2 ilustra a entrada de um obstáculo na ZVD: em vermelho está destacada a parte do obstáculo que entrou na ZVD, e que gera o vetor de deformação Δ . Também são mostradas as variáveis envolvidas na deformação e a relação entre elas, para o ângulo θ_i . Se não houvesse obstáculo nessa direção, o sensor forneceria uma medida máxima $d_{i\max}$ de acordo com o alcance máximo do sensor *laser* e, como resultado, a ZVD S permaneceria intacta. Mesmo que um obstáculo fosse detectado, caso a medida do sensor fosse maior que S_i , a ZVD S ainda permaneceria intacta. Como a medida do sensor na figura é menor que S_i , o obstáculo está próximo o suficiente para deformar a ZVD (S_i), e o elemento de deformação Δ_i , assim como o vetor de intrusão I são calculados.

4.2 MODELO DO SENSOR LASER

Para que o sensor seja utilizado no Filtro Bayesiano, é necessário um modelo probabilístico. Uma varredura completa do sensor *laser* fornece 181 medidas orientadas: quanto maior o número de medidas empregadas no modelo, maior será a sua precisão e maior será o custo computacional.

A *ZVD* ao redor do robô é formada por duas partes (luz e umbra), e o modelo do sensor somente diz respeito a obstáculos na parte da luz: caso um objeto passe da luz para a umbra, o sensor não fornece mais leituras desse obstáculo, e o modelo do sensor é conseqüentemente desativado.

4.2.1 Leitura Laser

O modelo do sensor *laser* foi construído utilizando o modelo apresentado na seção 3.4.2, mas somente três dos quatro tipos de erros de medidas apresentados foram utilizados neste trabalho: a distribuição p_{short} (chamada de erro do *obstáculo não esperado* e representada pela exponencial para medidas menores que z_k^{i*}) não foi utilizada porque aqui o robô móvel considera a presença de um obstáculo.

As medidas do sensor no instante k são salvas no vetor LL_k , constituído por N dados. Esse vetor armazena as 181 leituras (uma leitura a cada grau, entre 0 e 180 graus) no formato (ρ_i, θ_i) ,

$$LL_k = \begin{bmatrix} \rho_i \\ \theta_i \end{bmatrix}, \quad (4.5)$$

onde ρ_i representa a distância em metros entre o sensor e o obstáculo detectado, e θ_i é o ângulo (no intervalo $[0, \pi]$) onde o valor ρ_i foi obtido, e i representa a i -ésima leitura do sensor, sendo $i \in [0, N]$.

4.2.2 Medidas do Modelo do Sensor

O campo de visão do sensor *laser* foi dividido em 13 setores ou zonas, denotadas por $N_{sensores}$, levando em consideração que as duas zonas laterais são maiores que as outras. Essas zonas maiores são compostas por 22 e 23 feixes, enquanto que as zonas restantes contêm 10 feixes cada uma, como mostrado na figura 4.3. Esta escolha foi feita baseada no fato de que o obstáculo presente nessas zonas apresenta um risco menor de colisão com o robô.

Cada zona Z_j , onde $j = 1, 2, \dots, N_{sensores}$, do sensor *laser* é composta por um conjunto de feixes. Para cada zona Z_j obtém-se o menor valor de ρ_i em um conjunto de feixes desde

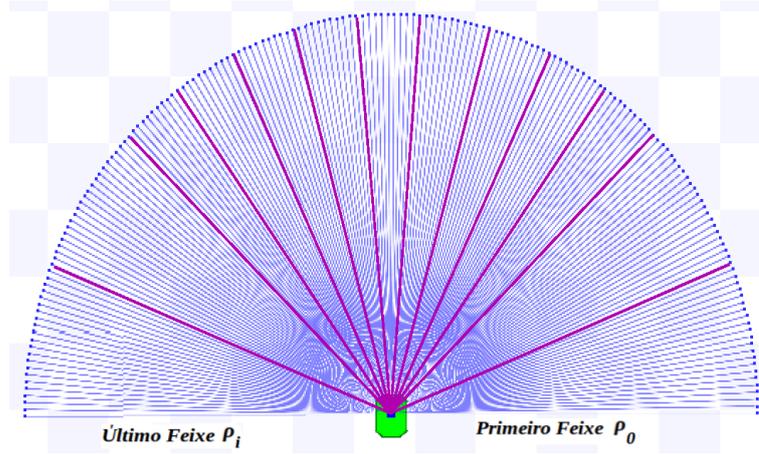


Figura 4.3: Esquema da distribuição das zonas do sensor

i_{min} até i_{max} :

$$Z_j = \min_i(\rho_i) \quad , \quad i_{min} \leq i \leq i_{max}. \quad (4.6)$$

onde i representa a i -ésima leitura do sensor em cada zona em que foi dividido, sendo $i \in [i_{min}, i_{max}]$, no instante de tempo k . A partir da equação (4.6), para cada zona Z_j do sensor laser no instante de tempo $t = k$ se obtém:

$$\begin{aligned} \rho_k^j &= \rho_{Z_j} \\ \theta_k^j &= \theta_{Z_j} \end{aligned} \quad (4.7)$$

Nas figuras 4.4 a) e b) ilustram uma situação típica quando o sensor *laser* detecta um obstáculo no sistema de coordenadas do robô móvel, e na figura c) mostra em detalhes todos os feixes que atingem a superfície do obstáculo em uma das zonas Z_j em que foi dividido o sensor laser e, na cor vermelha está representado o feixe de menor leitura obtida a partir da equação 4.6.

Inicialmente, a distribuição de probabilidade sobre a distância medida $p(\rho_k, \mathbf{X}_k)$ é obtida, e na sequência, a distribuição de probabilidade sobre o ângulo da medida $p(\theta_k, \mathbf{X}_k)$.

De acordo com as equações (3.23) e (3.24), é calculada a distribuição de probabilidade $p_{hit(\rho)}(\rho_k^i, \mathbf{X}_k)$ como uma distribuição Gaussiana com média ρ_k^j calculada nas equações (4.6) e (4.7) e desvio padrão $\sigma_{hit(\rho)}$, que indica a medida da distância em que é possível localizar o obstáculo:

$$p_{hit(\rho)}(\rho_k^i, \mathbf{X}_k) = \begin{cases} \eta N(\rho_k^i | \rho_k^j, \sigma_{hit(\rho)}^2), & 0 \leq \rho_k^i \leq \rho_{max} \\ 0 & , \quad \text{outro.} \end{cases} \quad (4.8)$$

A constante η de normalização é definida por:

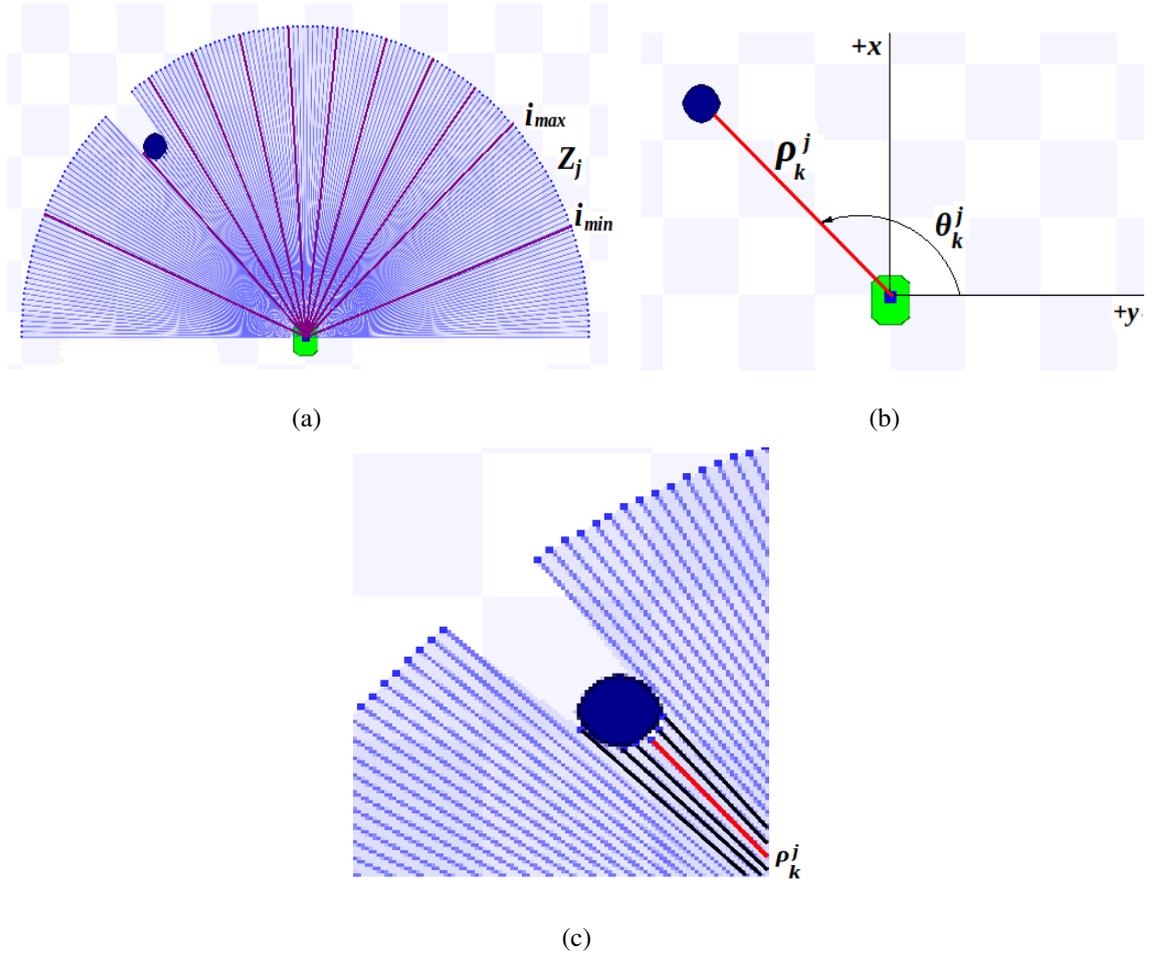


Figura 4.4: Esquema de leitura do *laser* a) detecção de um obstáculo pelo sensor *laser* em uma das zonas Z_j , b) sistema de coordenadas da leitura do sensor, c) detalhe dos feixes que atingem o obstáculo.

$$\eta = \frac{1}{\sum_{i=0}^{\rho_{max}} N(\rho_k^i | \rho_k^j, \sigma^2_{hit(\rho)})}. \quad (4.9)$$

A distribuição de probabilidade do sensor resultar em uma medida máxima erroneamente é representada pela distribuição $p_{max(\rho)}(\rho_k^i, \mathbf{X}_k)$:

$$p_{max(\rho)}(\rho_k^i, \mathbf{X}_k) = \begin{cases} 1, & \rho_k^i = \rho_{max} \\ 0, & \text{outro.} \end{cases} \quad (4.10)$$

Uma distribuição uniforme por todo o intervalo possível da medida, $p_{rand(\rho)}(\rho_k^i, \mathbf{X}_k)$, representa as leituras aleatórias de causas desconhecidas ou desconsideradas:

$$p_{rand(\rho)}(\rho_k^i, \mathbf{X}_k) = \begin{cases} \frac{1}{\rho_{max}}, & 0 \leq \rho_k^i < \rho_{max} \\ 0, & \text{outro.} \end{cases} \quad (4.11)$$

A distribuição de probabilidade $p_j(\rho_k, \mathbf{X}_k)$ de cada zona $j = 1, 2, \dots, N_{sensores}$ é formada

pela soma ponderada das distribuições descritas nas equações (4.8), (4.10) e (4.11), de acordo com [25]:

$$p_j(\boldsymbol{\rho}_k, \mathbf{X}_k) = \begin{cases} w_{hit(\rho)} \cdot p_{hit(\rho)}(\boldsymbol{\rho}_k^i, \mathbf{X}_k) + \\ w_{max(\rho)} \cdot p_{max(\rho)}(\boldsymbol{\rho}_k^i, \mathbf{X}_k) + \\ w_{rand(\rho)} \cdot p_{rand(\rho)}(\boldsymbol{\rho}_k^i, \mathbf{X}_k), \end{cases} \quad (4.12)$$

onde $w_{hit(\rho)} + w_{max(\rho)} + w_{rand(\rho)} = 1$.

Como indicado na seção 3.4.2, e assumindo a independência condicional entre as leituras do sensor, a distribuição de probabilidade normalizada de ρ_k é obtida pela equação (4.13)

$$\mathbf{P}(\boldsymbol{\rho}_k, \mathbf{X}_k) = \prod_{j=1}^{N_{sensores}} p_j(\boldsymbol{\rho}_k, \mathbf{X}_k). \quad (4.13)$$

Para o modelo de leituras do ângulo θ_k do sensor, são construídas duas distribuições: $p_{hit(\theta)}(\boldsymbol{\theta}_k^i, \mathbf{X}_k)$ e $p_{rand(\theta)}(\boldsymbol{\theta}_k^i, \mathbf{X}_k)$. A distribuição que representa as falhas do sensor não foi levada em conta porque as medidas de ângulo não possuem um valor padrão para o caso de leituras errôneas.

Dessa forma, a distribuição de probabilidade que indica a medida do ângulo em que o obstáculo foi detectado é definida como uma distribuição Gaussiana com média θ_k^j calculada nas equações (4.6) e (4.7) e desvio padrão $\sigma_{hit(\theta)}$:

$$p_{hit(\theta)}(\boldsymbol{\theta}_k^i, \mathbf{X}_k) = \begin{cases} \eta N(\theta_k^i | \theta_k^j, \sigma_{hit(\theta)}^2), & 0 \leq \theta_k^i \leq \theta_{max} \\ 0, & \text{outro.} \end{cases} \quad (4.14)$$

As leituras aleatórias são representadas por uma distribuição uniforme no intervalo de i_{min} até i_{max} de cada zona Z_j onde $j = 1, 2, \dots, N_{sensores}$, como definida na equação (4.15):

$$p_{rand(\theta)}(\boldsymbol{\theta}_k^i, \mathbf{X}_k) = \begin{cases} \frac{1}{\theta_{max}}, & 0 \leq \theta_k^i < \theta_{max} \\ 0, & \text{outro.} \end{cases} \quad (4.15)$$

A distribuição de probabilidade $p_j(\boldsymbol{\theta}_k, \mathbf{X}_k)$ é definida de forma similar a distribuição $p_j(\boldsymbol{\rho}_k, \mathbf{X}_k)$ da equação (4.12). Realiza-se a soma ponderada das distribuições das equações (4.14) e (4.15), também considerando que a soma dos pesos atribuídos a cada componente deve ser um, ou seja, $w_{hit(\theta)} + w_{rand(\theta)} = 1$:

$$p_j(\boldsymbol{\theta}_k, \mathbf{X}_k) = \begin{cases} w_{hit(\theta)} \cdot p_{hit(\theta)}(\boldsymbol{\theta}_k^i, \mathbf{X}_k) + \\ w_{rand(\theta)} \cdot p_{rand(\theta)}(\boldsymbol{\theta}_k^i, \mathbf{X}_k). \end{cases} \quad (4.16)$$

A distribuição de probabilidade normalizada sobre θ_k é obtida na equação (4.17) para cada zona $j = 1, 2, \dots, N_{sensores}$:

$$\mathbf{P}(\boldsymbol{\theta}_k, \mathbf{X}_k) = \prod_{j=1}^{N_{\text{sensores}}} p_j(\boldsymbol{\theta}_k, \mathbf{X}_k). \quad (4.17)$$

As distribuições de probabilidades $\mathbf{P}(\boldsymbol{\rho}_k, \mathbf{X}_k)$ e $\mathbf{P}(\boldsymbol{\theta}_k, \mathbf{X}_k)$ permitem que cada medida ρ_k^i e θ_k^i seja devidamente utilizada na estimação da posição dos obstáculos e a consequente deformação na ZVD. O algoritmo 5 resume os passos sequenciais necessários para utilizar o modelo do sensor derivado acima.

Algoritmo 5: Algoritmo Modelo Sensor

Entrada: $LL_k = [\{\rho_k, \theta_k\}]$

Saída: PDF= $[\{\mathbf{P}(\boldsymbol{\rho}_k, \mathbf{X}_k)\}]$ PDF= $[\{\mathbf{P}(\boldsymbol{\theta}_k, \mathbf{X}_k)\}]$

1 **início**

2 **para** $j = 0 : N_{\text{sensores}}$ **faça**

3 Obter o menor valor do ρ_k^j , e seu respectivo θ_k^j de acordo com (4.6)

4 **para** $i = 0 : \rho_{max}$ **faça**

5 Obter $p_{hit(\rho)}(\rho_k^i, \mathbf{X}_k)$ de acordo com (4.8),

6 Obter $p_{max(\rho)}(\rho_k^i, \mathbf{X}_k)$ de acordo com (4.10),

7 Obter $p_{rand(\rho)}(\rho_k^i, \mathbf{X}_k)$ de acordo com (4.11)

8 **fim para**

9 **para** $i = 0 : \rho_{max}$ **faça**

10 Obter $p_j(\rho_k^i, \mathbf{X}_k)$ de acordo com (4.12)

11 **fim para**

12 Obter $\mathbf{P}(\boldsymbol{\rho}_k, \mathbf{X}_k) = \prod_{j=1}^{N_{\text{sensores}}} p_j(\rho_k^i, \mathbf{X}_k)$ de acordo com (4.13)

13 **para** $i = 0 : \theta_{max}$ **faça**

14 Obter $p_{hit(\theta)}(\theta_k^i, \mathbf{X}_k)$ de acordo com (4.14)

15 Obter $p_{rand(\theta)}(\theta_k^i, \mathbf{X}_k)$ de acordo com (4.15)

16 **fim para**

17 **para** $i = 0 : \theta_{max}$ **faça**

18 Obter $p_j(\theta_k^i, \mathbf{X}_k)$ de acordo com (4.16)

19 **fim para**

20 Obter $\mathbf{P}(\boldsymbol{\theta}_k, \mathbf{X}_k) = \prod_{j=1}^{N_{\text{sensores}}} p_j(\theta_k^i, \mathbf{X}_k)$ de acordo com (4.17)

21 **fim para**

22 **fim**

4.3 MODELO DE MOVIMENTO

O modelo de movimento representa a geometria do caminho percorrido pelo robô, Esse caminh é descrito em coordenadas polares, de acordo com o modelo matemático desenvolvido nesta seção.

Seja um sistema de referência local, fixo no corpo do robô, que tem sua origem no centro do robô e que coincide com a posição do sensor laser: o eixo $+x$ aponta para a direção do movimento do robô (a frente), e o eixo $+y$, ortogonal ao eixo $+x$, aponta para o lado direito do robô.

É necessário definir uma trajetória suave para o robô móvel, assumindo que seus movimentos são curtos em instante de tempo consecutivos e que tendem a zero. A combinação dos movimentos resulta em uma sequência de pequenas retas. Uma curva pode ser gerada pela conexão dos pontos formando linhas.

A Figura 4.5 ilustra a posição do robô em relação ao obstáculo, em dois instantes de tempo, $t = k - 1$ e $t = k$. O vetor de controle $\mathbf{u}_k = [\nu, \omega]$ define as velocidades de translação e rotação, respectivamente, com que o robô se movimenta. A partir dessas velocidades, é obtido o deslocamento r num intervalo de tempo T , este deslocamento foi ampliado intencionalmente nesta figura com o objetivo de visualizar melhor as variáveis.

Ainda de acordo com a Figura 4.5, no instante de tempo $k - 1$ os valores de θ_{k-1} e ρ_{k-1} são conhecidos quando é aplicado o vetor de controle $\mathbf{u}_k = [\nu, \omega]$.

A partir da hipótese enunciada anteriormente, onde o caminho percorrido pelo robô móvel é uma sequência de pequenas retas, pode-se calcular a nova posição do robô no tempo $t = k$ em coordenadas polares por uma relação trigonometria. A figura 4.5 ilustra a posição do robô em um instante tempo $t = k - 1$ em coordenadas polares com relação ao obstáculo, θ_{k-1} e ρ_{k-1} . O deslocamento r , é calculado em um intervalo de tempo T com velocidade de translação ν . a partir dos valores das variáveis no instante de tempo anterior e utilizando a equação (4.18) pode se calcular a nova posição do robô, θ_k .

$$\rho_k = \sqrt{r^2 + \rho_{k-1}^2 - 2 * r * \rho_{k-1} * \cos(\theta_{k-1})}. \quad (4.18)$$

Para calcular a orientação θ_k no tempo $t = k$, é necessário primeiro obter φ_k que é o ângulo interno do triângulo formado pelo deslocamento r e ρ_k , utilizando a equação (4.19):

$$\varphi_k = \arccos \frac{r^2 + \rho_k^2 - \rho_{k-1}^2}{2 * r * \rho_k}. \quad (4.19)$$

Adicionalmente, as seguintes considerações são utilizadas para calcular o ângulo de orientação do obstáculo de acordo com o sistema de coordenadas fixo no corpo do robô, como mostrado na equação (4.20).

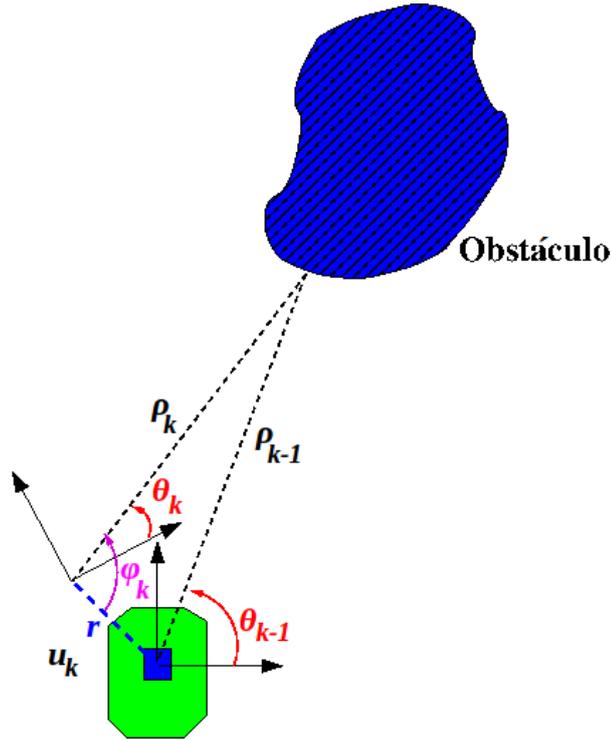


Figura 4.5: Variáveis envolvidas no movimento do robô do instante de tempo $t = k - 1$ e $t = k$ ao se aproximar um obstáculo: ρ e θ definem a posição do obstáculo em coordenadas polares nos diferentes instantes de tempo, r é o deslocamento em um intervalo de tempo que tende a zero, e φ é o ângulo auxiliar para a obtenção da orientação final do robô θ_k .

- O valor do ângulo no instante de tempo θ_{k-1} , é necessário para determinar se o obstáculo estava no eixo $+x$ ou $-x$ do sistema de referência do robô;
- Para calcular o complemento do ângulo e obter θ_k é necessário saber se o robô está se deslocando para a frente ou retrocedendo, ou seja, se a velocidade translacional é positiva, $+\nu$ ou negativa $-\nu$.

O ângulo φ_k , em violeta na Figura 4.5, é utilizado para calcular o ângulo θ_k , destacado em vermelho, e que é medido desde o eixo $+x$ do sistema de coordenadas fixo ao corpo do robô.

$$\theta_k = \begin{cases} 270^\circ - \varphi_k & \text{se } 270^\circ \leq \theta_{k-1} \leq 360^\circ \text{ ou } 0^\circ \leq \theta_{k-1} \leq 90^\circ \\ 270^\circ + \varphi_k & \text{se } 90^\circ < \theta_{k-1} < 270^\circ. \end{cases} \quad (4.20)$$

A partir deste cálculo é necessário um modelo que traduz o efeito do ruído na pose resultante. Como foi dito, qualquer movimento arbitrário pode ser obtido a partir de uma rotação seguida de uma translação.

O robô vai rodar $\Delta\theta = \theta_k - \theta_{k-1}$, onde $\Delta\theta = \arctan(\Delta y / \Delta x)$ e a translação vai ser $\rho = \sqrt{(\Delta x^2 / \Delta y^2)}$, de acordo com a equação:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + (\rho_{k-1}) \cos \theta_{k-1} \\ y_{k-1} + (\rho_{k-1}) \sin \theta_{k-1} \\ \theta_{k-1} + \Delta\theta \end{bmatrix} \quad (4.21)$$

Como em [45] quando o robô faz uma rotação, o erro de odometria E_{rot} é modelado como um ruído branco, ou seja, uma distribuição gaussiana de média $M_{rot} = 0$ e desvio padrão σ_{rot} , ou seja, $E_{rot} = N(0, 1)$.

No caso da traslação existem duas fontes de erro, a primeira se relaciona com a distância percorrida e a segunda com as mudanças de orientação [45]. No primeiro caso o erro relativo à distância percorrida E_{trans} é modelado adicionando ruído à distância percorrida, ou seja, uma distribuição gaussiana de média $M_{trans} = 0$ e desvio padrão σ_{rot} , ou seja, $E_{trans} = N(0, 1)$. E no segundo caso o erro E_{rot} consiste em adicionar o ruído na orientação de cada traslação.

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + (\Delta\rho + E_{trans}) \cos(\Delta\theta + E_{rot}) \\ y_{k-1} + (\Delta\rho + E_{trans}) \sin(\Delta\theta + E_{rot}) \\ \theta_{k-1} + \Delta\theta + E_{rot} \end{bmatrix} \quad (4.22)$$

4.4 FILTRO DE PARTÍCULAS

O Filtro de Partículas utilizado nessa proposta (SIR) [30], tem como objetivo principal estimar a posição do obstáculo tanto na zona de luz quanto na zona de umbra. O princípio básico de seu funcionamento consiste, na representação de uma distribuição de probabilidade a partir de um conjunto de amostras, cada uma associada a um peso, que representa a qualidade da estimativa dessa amostra.

Para o algoritmo empregado neste trabalho, as variáveis de interesse do filtro consistem no vetor que descreve a posição do obstáculo mais próximo, $\mathbf{X}_k = [\rho_k, \theta_k]^T$ no tempo $t=k$. No filtro de partículas, a distribuição de probabilidade deste vetor é representada por um conjunto de N_s amostras, chamadas partículas, $C_k = [\mathbf{X}_k^i, w_k^i]$, onde $i = 1 \dots N_s$ é o índice de cada partícula.

No instante $t=0$, o conjunto de N_s partículas iniciais $C_0 = [\mathbf{X}_0^i, w_0^i]$, são espalhadas por todo o campo de visão do sensor *laser*, no espaço a frente do robô. Os valores de ρ e θ , são escolhidos por meio de uma probabilidade inicial uniforme, $p(\mathbf{X}_0)$, e o peso de cada partícula também é distribuído de maneira uniforme e são calculados pela equação (4.23):

$$w_i = \frac{1}{N_s}. \quad (4.23)$$

No tempo $t=k$, a etapa de *predição* calcula a distribuição de probabilidade *a priori* a

partir da distribuição de probabilidade do instante prévio $t = k-1$, bem considerando o vetor de controle $\mathbf{u}_k = [\nu, \omega]$ e o modelo de movimento do robô, assumindo o obstáculo imóvel. Nessa etapa, a ação de controle feita sobre o robô vai repercutir na posição do obstáculo de acordo no sistema de referência fixo no corpo do robô, e esta informação é utilizada para simular o movimento na posição das partículas assim como adicionar incerteza (ruído) associado ao movimento realizado.

Uma outra etapa, chamada de *atualização*, utiliza a informação obtida do sensor laser para atualizar os pesos das partículas e estimar, com maior precisão, a posição do obstáculo.

A seguir, ambas as etapas do filtro são descritas em detalhes.

4.4.1 Predição

Para obter a função de distribuição de probabilidade da posição do objeto, após um movimento feito pelo controle do robô, é necessário conhecer o modelo desse movimento e a incerteza associada a esse modelo. Como descrito anteriormente, qualquer movimento do robô é descrito como uma combinação de uma rotação sucedida de uma translação ($\nu > 0, \omega > 0$).

A posição do obstáculo no tempo $t=k-1$ é descrita por $\mathbf{X}_{k-1} = [\rho_{k-1}, \theta_{k-1}]^T$. Primeiro, o robô realiza o movimento de rotação, onde $\Delta\theta = \hat{\theta}_k - \theta_{k-1}$, e depois, ele se movimenta a frente de uma distância r , gerando uma nova posição do obstáculo $\mathbf{X}_k = [\hat{\rho}_k, \hat{\theta}_k]^T$, no sistema de coordenadas do robô, conforme as equações (4.18), (4.19) e (4.20) adicionando o ruído na equação (4.22) do modelo de movimento.

Na etapa de predição, após calcular a trajetória gerada pelo modelo de movimento, o conjunto de partículas representam a distribuição de probabilidade da posição do obstáculo, ou seja, os valores dessas partículas são hipóteses das possíveis posições onde o obstáculo será observado durante o percurso da trajetória.

A orientação de cada partícula i , é atualizada com o valor de rotação $\bar{\theta}_k^i$ calculado para cada uma delas, mais a incerteza associada a esse movimento. A incerteza de cada movimento é modelada por uma variável aleatória na forma de uma distribuição normal, com média $M_{rot} = 0$ e desvio padrão σ_{rot} :

$$\hat{\theta}_k^i = \bar{\theta}_k^i + N(M_{rot}, \sigma_{rot}). \quad (4.24)$$

A componente $\hat{\rho}_k^i$, resultante da posição de cada partícula devido ao deslocamento do robô, é por sua vez atualizada com o valor $\bar{\rho}_k^i$ e, de forma similar a orientação da partícula, uma incerteza é adicionada na forma de uma distribuição normal, com média $M_{trans} = 0$ e desvio padrão σ_{trans} :

$$\hat{\rho}_k^i = \bar{\rho}_k^i + N(M_{trans}, \sigma_{trans}). \quad (4.25)$$

Os valores de σ_{trans} e σ_{rot} da mesma forma como é feita na seção anterior pode-se calcular como em [45]. O algoritmo 6 descreve a etapa de predição, executada independentemente de haver ou não movimento do robô, e que ao final retorna o conjunto de partículas com o efeito que aquele movimento exerce sobre seus valores.

Algoritmo 6: Algoritmo de Predição

Entrada: $\nu, \omega, \mathbf{X}_{k-1} = [\rho_{k-1}, \theta_{k-1}]^T$

Saída: $X_k^i \sim p(X_k | X_{k-1}^i)$

```

1 início
2    $r_{k-1} = \nu * T;$ 
3   para  $i = 1 : N_s$  faça
4     Obter  $\bar{\rho}_k^i$  de acordo com (4.18) ,
5     Obter  $\varphi_k$  de acordo com (4.19),
6     Cálculo do  $\bar{\theta}_k^i$  de acordo com (4.20),
7      $E_{trans} = randN(M_{trans}, \sigma_{trans});$ 
8      $E_{rot} = randN(M_{rot}, \sigma_{rot});$ 
9      $\hat{\theta}_k^i = \bar{\theta}_k^i + E_{rot};$ 
10     $\hat{\rho}_k^i = \bar{\rho}_k^i + E_{trans};$ 
11     $X_k^i \sim p(X_k | X_{k-1}^i);$ 
12  fim para
13 fim

```

4.4.2 Atualização

Nesta etapa, os pesos das partículas são atualizados utilizando a última leitura obtida pelo modelo do sensor. Para isto, são necessários tanto o conjunto das partículas resultante da etapa de predição ($\hat{\mathbf{X}}_k^i = [\hat{\rho}_k^i, \hat{\theta}_k^i]^T$), quanto as medidas do sensor *laser*, $z_k^i = (\rho_k, \theta_k)$. A posição atualizada do obstáculo é obtida empregando a distribuição de probabilidade $p(z_k^i | \mathbf{x}_k)$, apresentada na equação (3.33).

O peso de cada partícula w^i , $i = 1 \dots N_s$, é calculado comparando a posição predita $\hat{\mathbf{X}}_k^i$ com a posição dessa mesma partícula determinada pelos dados extraídos da leitura do modelo do sensor laser ρ_k^i e θ_k^i . O cálculo se faz a partir do valor das funções de distribuição de probabilidades $p(\rho_k, \mathbf{X}_k)$ e $p(\theta_k, \mathbf{X}_k)$, geradas pelas equações (4.13) e (4.17), respectivamente, utilizando a equação (4.26):

$$w_k^i = p(\rho_k^i, \mathbf{X}_k) p(\theta_k^i, \mathbf{X}_k). \quad (4.26)$$

O método é realizado individualmente para cada partícula. A Figura 4.7 ilustra como é aplicado para uma partícula i qualquer no instante k , que faz parte do conjunto de N_s partículas desenhadas na cor vermelha. A amostra em particular está localizada na posição

de interesse $\mathbf{X}_k^i = [\rho_k^i, \theta_k^i]^T$, e na figura é recuperada a medida do seu peso w_k^i na função de distribuição de probabilidade, que representa a proximidade dessa partícula com o estado verdadeiro, onde o estado mais próximo é aquele que tem a medida de probabilidade mais alta e terá o maior peso. Os pesos são normalizados desde 0 até 1, utilizando a equação (4.27), e desta forma o conjunto de pesos são uma distribuição de probabilidade discreta:

$$w_k^i = \frac{w_k^i}{\sum_{i=1}^{N_s} w_k^i}. \quad (4.27)$$

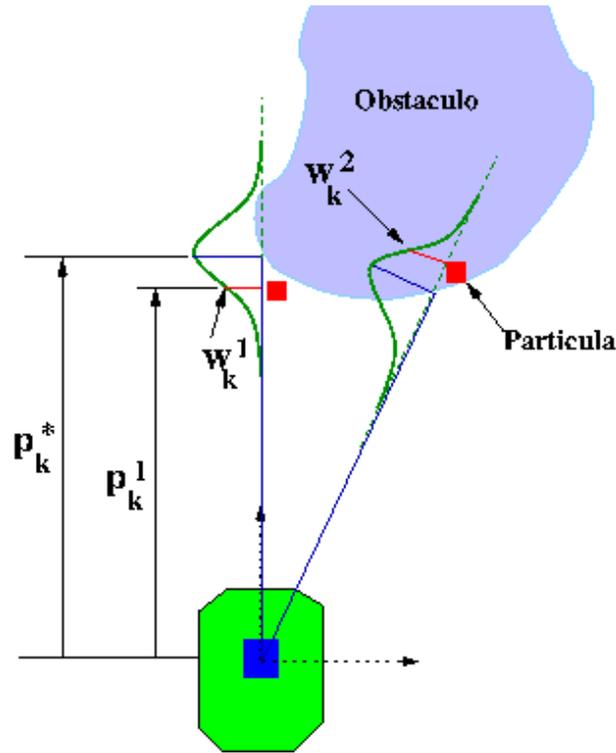


Figura 4.6: Função de distribuição de probabilidade Gaussiana, aplicada à partícula i no instante $t = k$ próxima à posição de interesse.

4.4.3 Reamostragem

Nesta etapa, é essencial para o filtro de partículas, como descrito no capítulo 3, para que consiga manter o conjunto de partículas em torno do estado mais provável, e evite o problema de degeneração.

É comum ocorrer o fenômeno de empobrecimento da amostra no filtro de partículas, depois de algumas iterações. Apenas algumas partículas tem um peso não desprezível enquanto as outras possuem um peso tão pequeno que não contribuem para a distribuição de probabilidade da posição do obstáculo.

No tempo $t=k$, o conjunto de N_s partículas $C_k = [\mathbf{X}_k^i, w_k^i]$, onde $i = 1 \dots N_s$, é o índice de cada partícula, e representa a função de distribuição de probabilidade da posição do obs-

táculo. Um novo conjunto de partículas $\hat{C}_k = [\hat{\mathbf{X}}_k^i, \hat{w}_k^i]$ é necessário, tal que $\mathbf{X}_k^i = \hat{\mathbf{X}}_k^i$ e que seus pesos sejam $\hat{w}_i = \frac{1}{N_s}$, para representar a mesma função de distribuição de probabilidade. Mas antes de realizar esta mudança, conhecida como reamostragem, é necessário estimar o número de partículas que têm o peso próximo a zero.

A medida do número de partículas com peso muito pequeno é realizada por o cálculo do tamanho efetivo da amostra, \hat{N}_{eff} , conforme a equação (3.21). Esta medida então é comparada com um limite, estabelecido como um percentual do número total de partículas N_s . No algoritmo utilizado nesse trabalho, se o valor de \hat{N}_{eff} está abaixo do limite, a etapa de re-amostragem é executada, de tal forma que as partículas com menor peso sejam eliminadas e as partículas de maior peso sejam multiplicadas. A função de distribuição de probabilidade reconstruída pela re-amostragem é muito próxima daquela que existia antes da re-amostragem.

O algoritmo 3 apresentado na seção 3.3.4 é utilizado neste trabalho, onde a entrada é um arranjo composto dos pesos normalizados das partículas e o estado atual das mesmas, e a saída é um novo conjunto de partículas $\hat{C}_k = [\hat{\mathbf{X}}_k^i, \hat{w}_k^i]$.

4.4.4 Estimativa

A valor esperado do filtro é representada por um conjunto de partículas, que vão expressar a melhor estimativa da posição, $\bar{\mathbf{X}}_k$, onde encontra-se o obstáculo. Nesta dissertação é calculada de duas formas:

- A partir do *peso máximo*, onde a partícula com maior peso no instante $t=k$ é considerada como a melhor partícula, ou seja, com a mais alta probabilidade. Esta aproximação é facilmente calculada na equação (4.28):

$$\bar{\mathbf{X}}_{wmax} = argmax(\mathbf{X}_k^i). \quad (4.28)$$

- Também pode ser estimada pela média ponderada, calculada na equação (4.29):

$$\bar{\mathbf{X}}_{ponderada} = \sum_{i=1}^{N_s} \mathbf{X}_k^i w_k^i. \quad (4.29)$$

O algoritmo 7 apresenta o pseudocódigo do Filtro de Partículas completo implementado neste trabalho para estimar a posição do obstáculo nas zona de luz e de umbra da *ZVD*. A partir do instante em que o obstáculo passa para a zona de umbra, o algoritmo do modelo do sensor não fornece mais informações (em outras palavras, ele é desligado para esse obstáculo). O filtro de partículas não atualizará os pesos pelas medidas do modelo do sensor *laser*, elas receberão uma distribuição de peso uniforme calculado na equação (4.23). O fato

de não dispor mais dos dados do modelo do sensor, faz com que a certeza na posição do obstáculo diminua, o que é percebido na simulação por que as partículas se espalham com o passar do tempo.

Algoritmo 7: Algoritmo Filtro de Partículas

Entrada: $\{\mathbf{X}_{k-1}^i\}_{i=1}^{N_s}$

Saída: $[\{\mathbf{X}_k^i, w_k^i\}_{i=1}^{N_s}]$

1 **início**

2 **se** $Ativar=1$ **então**

3 Execução do Modelo de Movimento de acordo com;

4 $\mathbf{X}_k^i \sim p(\mathbf{X}_k | \mathbf{X}_{k-1}^i)$

5 Execução do Modelo do Sensor;

6 $w_k^i = p(\rho_k^i | \mathbf{X}_k^i) \cdot p(\theta_k^i | \mathbf{X}_k^i)$

7 **fim se**

8 Calculo dos pesos totais $\Upsilon = \sum_{i=1}^{N_s} w_k^i$;

9 **para** $j = 1 : N_s$ **faça**

10 Normalizar $w_k^i = w_k^i \cdot \Upsilon^{-1}$;

11 **fim para**

12 Calculo do $\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_s} (w_k^i)^2}$;

13 **se** $\hat{N}_{eff} < N_T$ **então**

14 Reamostragem;

15 Inicialização da CDF $c_1 = 0$

16 **para** $i = 2 : N_s$ **faça**

17 construção da CDF: $c_i = c_{i-1} + w_k^i$

18 **fim para**

19 Escolha de um ponto de inicio: $\lambda_1 \sim \Lambda[0, N_s^{-1}]$

20 **para** $j = 1 : N_s$ **faça**

21 Percorrer a FDP: $\lambda_j = \lambda_1 + N_s^{-1}(j - 1)$

22 **enquanto** $\lambda_j > c_i$ **faça**

23 $i = i + 1$

24 **fim enquanto**

25 Atribua as amostras $\mathbf{X}_k^{j*} = \mathbf{X}_k^i$

26 Atribua os pesos $w_k^j = N_s^{-1}$

27 Atribua os parentes $i^j = i$

28 **fim para**

29 **fim se**

30 **fim**

4.5 DEFORMAÇÃO DA ZVD

Depois de obter o vetor de posições das partículas dado por \hat{X}_k^i , no instante $t=k$, esse vetor será comparado com o limite da ZVD sem deformações, ou seja, o vetor S apresentado na equação (4.1). Portanto o *vetor de intrusão* I , e o *vetor de deformação* Δ serão calculados pelas equações (4.3) e (4.4) respectivamente.

Cada componente da ZVD é composta por um comprimento S_i localizado num ângulo θ_i , que será comparado com a posição da partícula. O algoritmo obtém a partícula mais próxima ao robô móvel de acordo com a equação (4.30), para cada direção θ_k^i :

$$\Delta_i = \begin{cases} 0 & , \quad \rho_i \geq S_i, \\ S_i - \rho_i & , \quad \rho_i < S_i. \end{cases} \quad (4.30)$$

Este conjunto de vetores Δ_i representa a deformação total calculada como na equação (4.31), onde n , é o numero de feixes que vão conformar a ZVD arepresentada n o vetor S :

$$\Delta = \sum_{i=1}^n \Delta^i. \quad (4.31)$$

4.6 AÇÃO DE CONTROLE PARA EVITAR O OBSTÁCULO

Este controlador é responsável por orientar o robô móvel a fim de evitar a colisão com o obstáculo mais próximo que altera significativamente a ZVD, e representa uma ameaça para integridade física do robô.

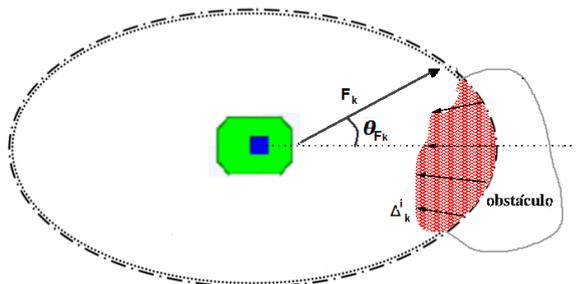


Figura 4.7: Variáveis do controle.

Na zona de luz da ZVD, um conjunto de vetores f_k^i orientados na mesma direção θ_i dos sensores é criado. O índice i representa o ângulo de direção do vetor. Esses vetores são duplicados (espelhados) na zona de umbra. Desta forma, é possível fazer o robô reagir modificando o vetor de velocidades u_k quando o obstáculo se aproxime por qualquer das

duas zonas, luz ou umbra. No tempo $t = k$, os vetores f_k^i são definidos como a diferença do vetor da deformação em dois tempos consecutivos $k - 1$ e k , como na equação (4.32):

$$\vec{f}_k^i = \begin{cases} 0 & \text{se } \Delta_k^i - \Delta_{k-1}^i < 0 \\ \Delta_k^i - \Delta_{k-1}^i & \text{se } \Delta_k^i - \Delta_{k-1}^i \geq 0. \end{cases} \quad (4.32)$$

E seja o vetor \vec{F}_k a soma de todos os vetores \vec{f}_k^i :

$$\vec{F}_k = \sum_{i=1}^{N_{\text{sensores}}} \vec{f}_k^i. \quad (4.33)$$

De acordo com [18] [46] o vetor de velocidades \mathbf{u}_k pode ser calculado pela equação (4.34) de forma a evitar uma colisão com o objeto que alterou a ZVD:

$$\mathbf{u}_k = \begin{cases} \nu_k = \nu_{k-1} + K_t ||F_k|| \cos(\theta_{F_k}) \\ \omega_k = \omega_{k-1} + K_r \text{sen}(\theta_{F_k}), \end{cases} \quad (4.34)$$

onde θ_{F_k} é o ângulo do vetor \vec{F}_k .

Esta ação de controle é executada até que a deformação total, conforme calculada na equação (4.31), seja nula: ou seja, $\Delta = 0$.

4.7 DISCUSSÃO

Aproposta se baseia no desenvolvimento de uma arquitetura reativa, uma Zona Virtual Deformável, que obtendo dados do ambiente desde o sistema de sensoriamento. A ZVD detecta um obstáculo como uma ameaça para integridade física do robô.

O Filtro de Partículas permite estimar a posição do obstáculo ao redor do robô repassando essa informação a ZVD para sua deformação tanto na região de alcance do sensor (zona de luz) quanto na região cega (zona de umbra). O Filtro de Partículas não atualiza os pesos por ausência de informação na região cega do sensor, mas continua estimando a partir do modelo de movimento.

Nesse capítulo se propõe integrar as informações do filtro de partículas com as informações da ZVD e atingir o objetivo de que o robô navegue sem colidir com o objeto que se apresente no ambiente. Esta integração pretende tornar a navegação do robô mais segura.

5 ANÁLISE DOS RESULTADOS

Para validação da proposta apresentada nesta dissertação, o presente capítulo descreve e analisa alguns dos experimentos desenvolvidos e seus resultados. A plataforma robótica escolhida para os experimentos é o robô Pioneer 3AT, e o ambiente *Player/Stage* para a simulação do robô e prototipagem dos algoritmos a serem implementados. As simulações e as aplicações cliente foram escritas em C++ e executadas no Sistema Operacional Linux.

A primeira seção 5.1 é uma descrição detalhada do robô e do ambiente de simulação. Na seção 5.2 e a seção 5.3 descrevem em detalhe a primeira simulação e na segunda simulação respectivamente com a sua análise de resultados. Na seção 5.4 apresenta-se uma breve discussão do capítulo.

5.1 ROBÔ PIONEER 3AT E FERRAMENTA DE SIMULAÇÃO

A plataforma robótica utilizada para os testes nesse trabalho é o robô Pioneer 3AT da Mobile Robots, figura 5.1. Suas dimensões são 50 x 49 x 26 cm e seu peso é de 12 kg. Tem a capacidade de carregar uma carga de 30 kg e mover-se com uma velocidade de até 1,2 m/s [44].



Figura 5.1: a) Pioneer 3AT (retirada de [44]).

A plataforma robótica foi programada em simulação utilizando o software *Player/Stage* [47]. Esta plataforma foi desenvolvida nos laboratórios de pesquisa da Universidade de Southern Califórnia (USC Robotics Research Lab) em 1999 [47] [48], e atualmente é empregada em muitos centros de pesquisa e universidades. Entre suas características principais estão sua API baseada no princípio *cliente/servidor* com sua distribuição gratuita.

O Player é um servidor de dispositivos que permite o controle dos diversos sensores e atuadores de um robô e provê uma interface de comunicação através de uma conexão de rede IP, seguindo o protocolo cliente-servidor [48].

O programa de controle (cliente) utiliza uma interface TCP, para comunicar-se com o Player, podendo configurar os dispositivos do robô em tempo real, obter os dados dos sensores e enviar os comandos aos atuadores. O servidor Player fornece uma interface que foi desenvolvida para que qualquer programa possa solicitar ou enviar informação ao robô, independente de sua plataforma ou da linguagem de programação empregada (podendo ser C, C++, Tcl, Java ou Python) [49].

Esta ferramenta possui interfaces variadas de comunicação para câmeras, sonares e laser, e suporte adicional para que outros hardwares possam ser facilmente acrescentados. Além disso, o Player permite controlar diversas plataformas de robôs comerciais, em particular os robôs móveis Pioneer (da empresa ActiveMedia). Uma das vantagens desta forma de comunicação é poder controlar tanto o robô no mundo real ou no ambiente de simulação Stage com o mesmo código, com poucas ou nenhuma alteração [49].

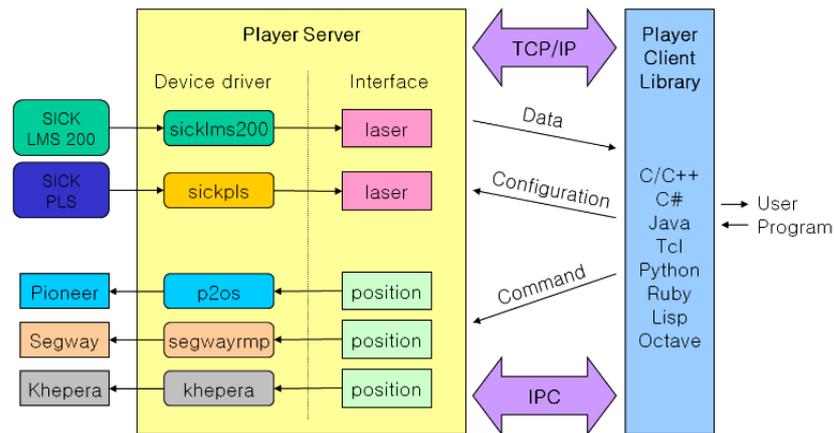
O Stage é um simulador de robôs móveis que pode ser controlado pelo Player. Basicamente, o Stage tem três componentes: o entorno, ou seja, a descrição do ambiente onde o robô simulado está inserido; o modelo dos componentes do robô simulado; e o controle que se faz através do Player [50].

O Stage foi especificamente projetado para dar suporte às pesquisas de sistemas multi-robôs, e permite simular uma grande quantidade de robôs ao mesmo tempo em um ambiente bidimensional. Seu objetivo é facilitar o desenvolvimento e os testes dos códigos antes de aplicá-los em um robô inserido em um ambiente real.

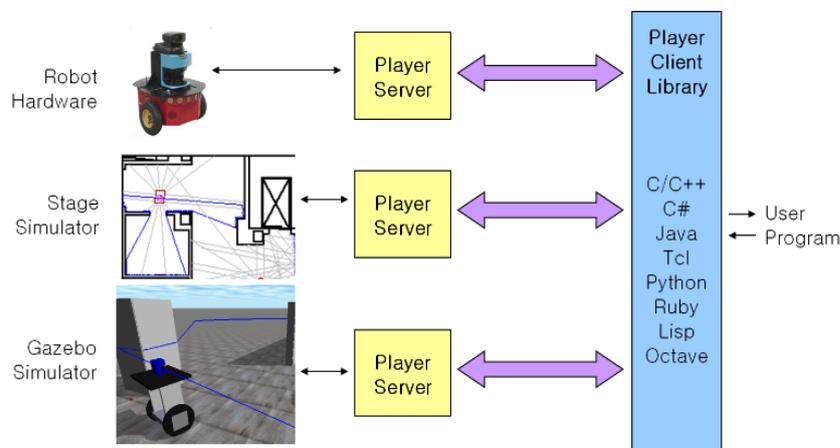
Além da interface em 2D, o *Player* possui uma outra interface em 3D chamada *Gazebo* [51]. Na Figura 5.2(a) encontra-se ilustrado o funcionamento do protocolo cliente-servidor, e apresenta a relação dos componentes físicos com os drivers do Player e com aplicação cliente num programa básico Cliente/Servidor. A estrutura de uma aplicação cliente se baseia em uma listagem de ações:

- Estabelecer a conexão ao servidor;
- Subscrever os dispositivos necessários para aplicação;
- Ler os dados dos sensores;
- Enviar comandos dos atuadores;
- Retirar a subscrição aos dispositivos antes mencionados;
- E por ultimo retirar a conexão ao servidor.

Na Figura 5.2(a) observa-se como o Player apresenta a mesma interface para diferentes fabricantes, isto permite que um mesmo programa de controle se possa utilizar em vários tipos de robôs. Esta vantagem do Player é muito útil na hora de utilizar o simulador Stage



(a)



(b)

Figura 5.2: a) Modelo cliente-servidor. b) Comunicação com o *hardware* ou simulador (retiradas de [52]).

porque os programas de controle escritos para os robôs simulados no Stage, são válidos e aplicáveis nos robôs reais [48] [52].

O programa pode ser escrito em um das linguagens de programação citadas acima para que se possa comunicar através de socket: desta forma, o programa cliente não diferencia se está interagindo com um robô real ou com um robô simulado.

Utilizando esta ferramenta de simulação foram realizadas duas simulações que tentam mostrar que é possível seguir a trajetória do obstáculo mais próximo com ajuda do filtro de partículas integrando com a ZVD, mesmo quando o obstáculo está fora do alcance do sensor. Primeiro numa situação simples com o robô em rota de colisão seção 5.2, e segundo com situações de constante colisão onde o robô permanece movimentando-se ao redor do obstáculo seção 5.3.

5.2 SIMULAÇÃO DO ROBÔ EM ROTA DE COLISÃO DIRETA COM O OBSTÁCULO

As duas simulações realizadas assumem que o obstáculo, inicialmente, entra na zona de varredura do sensor para que seja detectado pelo sensor laser e o modelo do sensor possa gerar informações. Esta hipótese é consequência do fato do robô móvel possuir somente um sensor laser e portanto detectar somente obstáculos a frente do robô, conforme descrito na seção 4.2.2.

Esta primeira simulação apresenta a situação em que o movimento do robô consiste em uma trajetória com o vetor de velocidades u_k inicialmente constante, enquanto o obstáculo está imóvel durante toda a simulação. A Figura 5.3, ilustra, em cor azul, as posições no caminho percorrido pelo robô móvel na simulação: sua posição inicial é $(x, y, \theta) = (-1, 0, 0^\circ)$, e a posição do obstáculo é $(7, -0.3, 0^\circ)$, mostrada com um ponto na cor vermelha. A trajetória inicial foi escolhida para que o robô se aproxime do obstáculo diretamente, como se fosse atingí-lo. Ao aproximar-se do obstáculo, o robô desvia para seu lado direito, e passa ao lado do obstáculo. Como consequência, o obstáculo passa para a zona de umbra, onde não é mais detectado pelo sensor. Nessa situação, o robô móvel executa uma rotação, com o objetivo de trazer o obstáculo de volta ao campo de visão do sensor e assim voltar a obter dados da sua posição. A Figura 5.3 ilustra também em letra vermelha as posições onde foram capturadas as imagens apresentadas na Figura 5.4 de situações destacadas da trajetória:

- O ponto sinalizado como (A) *iteração 1* o robô está localizado na posição inicial $(x, y, \theta) = (-1, 0, 0^\circ)$,
- No ponto (B) *iteração 55* a posição é $(x, y, \theta) = (1.7, -0.3, 0^\circ)$,
- No ponto (C) *iteração 178* a posição é $(x, y, \theta) = (5.2, -0.3, -28^\circ)$,
- No ponto (D) *iteração 287* a posição é $(x, y, \theta) = (6.6, -1.4, -28^\circ)$,
- No ponto (E) *iteração 321* a posição é $(x, y, \theta) = (8.7, -1.6, -71^\circ)$,
- E no ponto (F) *iteração 347* a posição é $(x, y, \theta) = (8.3, -2.1, -169^\circ)$.

O conjunto de Figuras 5.4 mostra a *ZVD*, nas situações destacadas na Figura 5.3. A parte cinza oval é a *ZVD*, enquanto os pontos vermelhos são as partículas do filtro: nesta simulação, o número de partículas é constante e igual a $N_s = 300$. A quantidade de partículas escolhida é devido as limitações que apresenta o computador em que foi realizada as simulações. Cada figura possui um quadro pequeno na sua parte superior esquerda que mostra a simulação do Player/Stage, onde pode ser visto a meia circunferência, na cor azul, com raio de 600 cm, e que consiste no campo de visão do sensor laser.

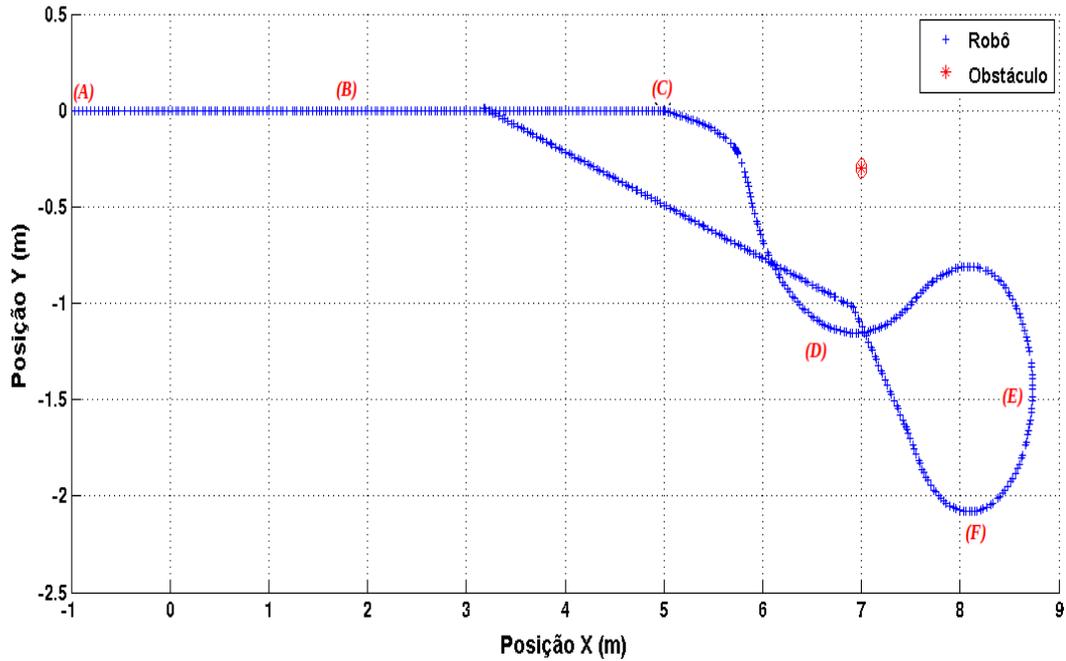


Figura 5.3: Trajetória percorrida pelo robô móvel na primeira simulação

A figura 5.4 (a) é a captura do instante inicial da simulação (ponto marcado como (A) na Figura 5.3) e as partículas encontram-se espalhadas a frente do robô móvel com peso uniforme calculado pela equação, $w_i = \frac{1}{N_s}$. O obstáculo ainda não foi detectado pelo sensor.

Na sequência, a figura 5.4-(b) refere-se ao instante em que o robô está localizado em $(x, y, \theta) = (1.7, -0.3, 0^\circ)$ marcado como (A) na figura 5.3. O obstáculo já é detectado pelo sensor, e o modelo do sensor fornece informações a respeito da distância do obstáculo ao robô. Dessa forma, as partículas começam a se localizar de acordo com a saída do filtro de partículas, $\mathbf{X}_k^i \sim p(\mathbf{X}_k | \mathbf{X}_{k-1}^i)$ do algoritmo 7.

Na figura 5.4-(c), é mostrado o instante em que o obstáculo entra na ZVD e ela começa a deformar-se: o robô está na posição $(x, y, \theta) = (5.2, -0.3, -28^\circ)$, marcada como (C) na Figura 5.3. Nesse instante, tem início as ações de controle geradas pelo algoritmo da ZVD, com o objetivo de evitar uma aproximação excessiva do obstáculo e uma consequente colisão. As ações de controle sobre o robô são calculadas pela equação 4.34 para reduzir a deformação total Δ .

Nesta simulação, as ações geradas para evitar que o robô colida com o obstáculo fazem com que o robô o contorne, e na figura 5.4-(d) pode ser observado que o obstáculo inicia a transição da zona de luz para a zona de umbra (definidas na seção 4.1) e as partículas acompanham esse movimento, localizando-se na proximidade do último valor estimado da posição. As partículas se espalham devido a perda de informação do sensor quando o obstáculo começa a aparecer na zona de umbra. Nesse instante, o robô está na posição $(x, y, \theta) = (6.6, -1.4, -28^\circ)$, que corresponde a marcação (D) na Figura 5.3.

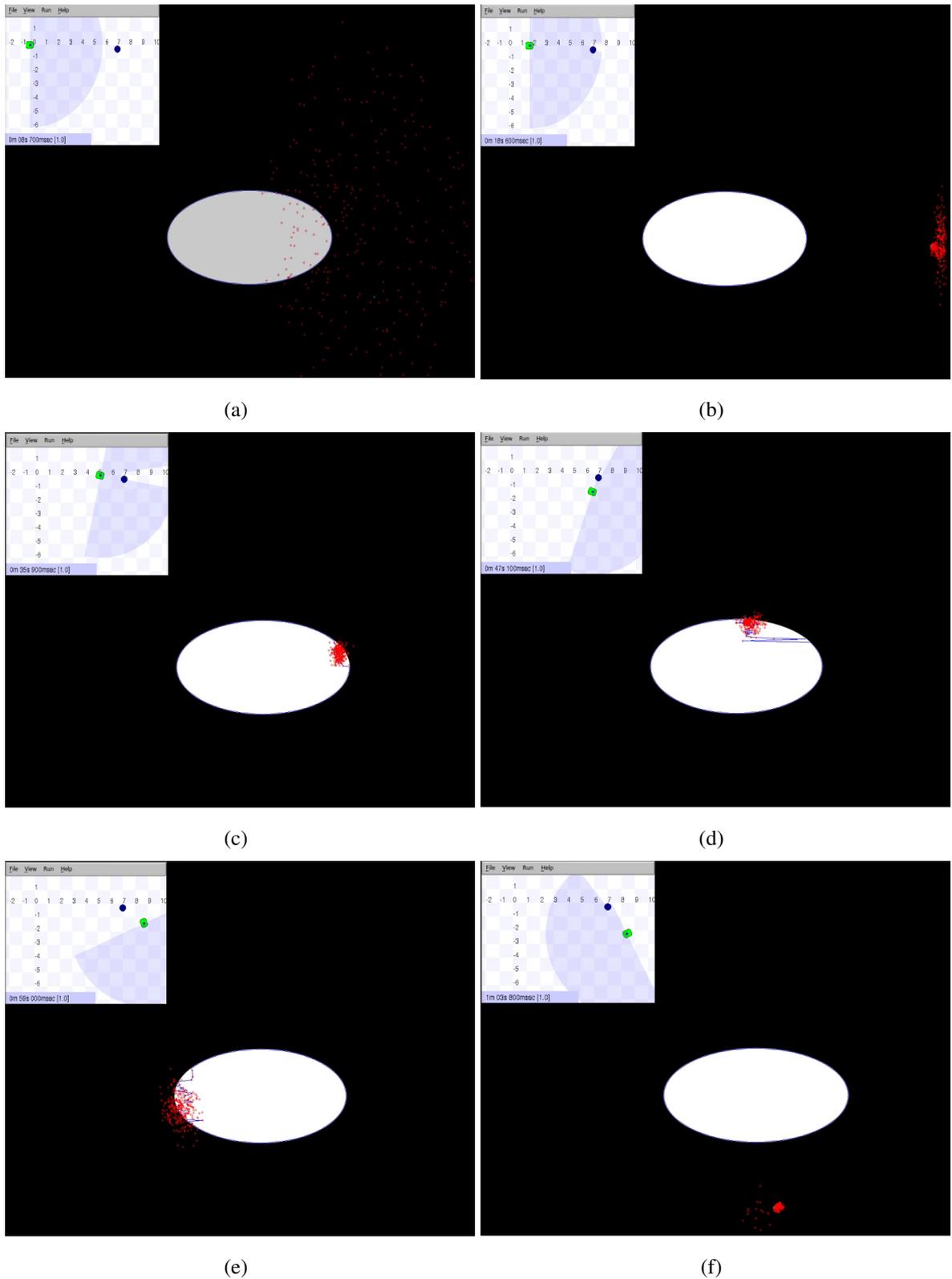


Figura 5.4: Sequência de imagens da primeira simulação

A Figura 5.4 (e) mostra a situação em que o obstáculo está totalmente posicionado na zona de sombra e pode-se observar que, pela estimativa de sua posição, ele ainda se encontra dentro da *ZVD*, deformando-a. O robô está na posição $(x, y, \theta) = (8.7, -1.6, -71^\circ)$,

marcada como (E) na Figura 5.3. Observa-se também como as partículas se espalharam ligeiramente, como na zona de umbra não existe informação do modelo do sensor, o peso das partículas é igual e dado pela equação (4.23). As partículas, então, se localizam baseadas na informação do modelo de movimento, sem realizar atualização em seus pesos.

Na Figura 5.4 (f), é mostrado o instante em que o robô está executando a ordem de rotacionar 180° graus, de forma a ter de novo o obstáculo visível pelo sensor. A posição do robô é $(x, y, \theta) = (8.3, -2.1, -169^{\circ})$, marcada como (F) na Figura 5.3. As partículas então se reagrupam quando o sensor volta a receber informação da presença do obstáculo no ambiente, e o filtro de partículas possui informação para atualizar os pesos das partículas.

Os gráficos a seguir apresentam distribuições de probabilidade relacionadas ao modelo do sensor para esta simulação. A Figura 5.5 mostra a distribuição de probabilidade $P(\rho_k^i, \mathbf{X}_k)$ em cada iteração, expressa na equação 4.13, e calculada pelo algoritmo do modelo do sensor. As probabilidades foram modeladas com desvio padrão $\sigma^2 = 20cm$. Já a função de probabilidade $P(\theta_k^i, \mathbf{X}_k)$, equação 4.17, é ilustrada na Figura 5.6, com desvio padrão $\sigma^2 = 25graus$.

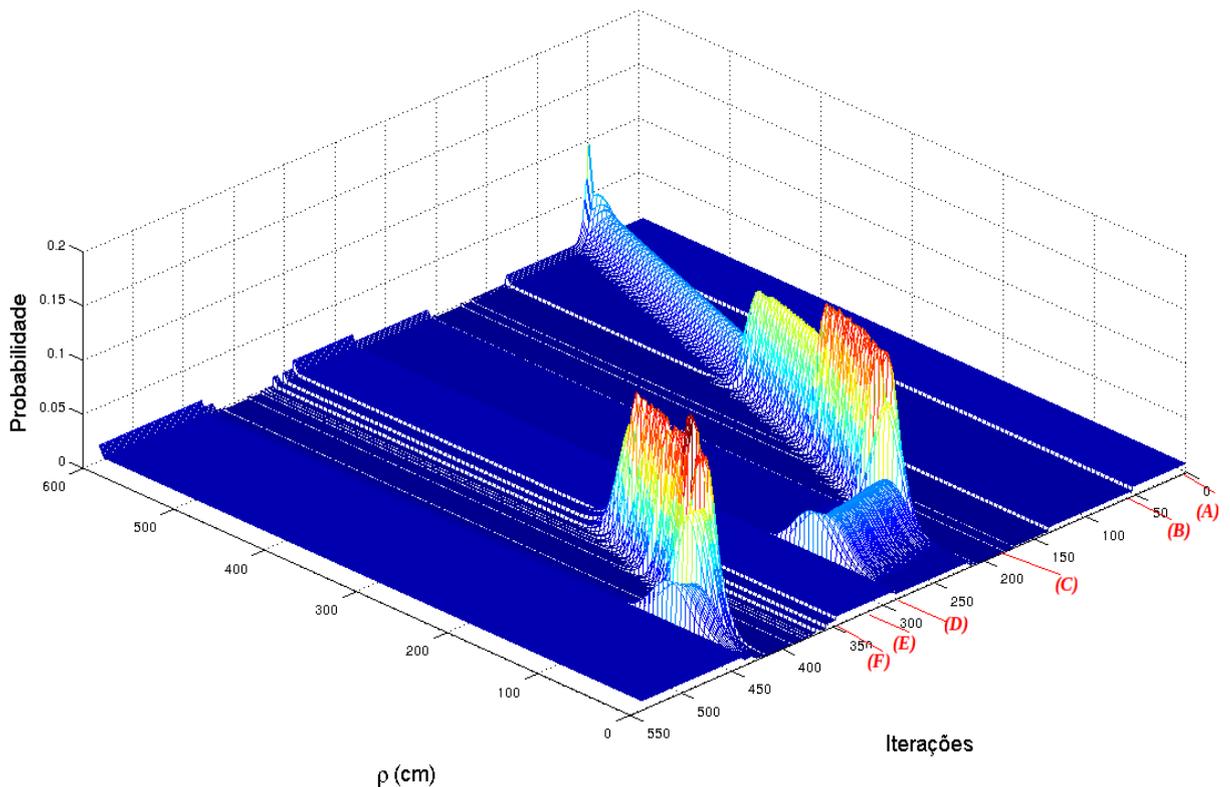


Figura 5.5: Leitura ρ do obstáculo, feita pelo modelo do sensor. As letras na cor vermelha indicam as posições da Figura 5.4

Nessas figuras, pode-se observar que, no início da simulação (primeiras iterações), ainda não existe informação da presença do obstáculo no ambiente do robô móvel. Somente após a iteração 55, quando o obstáculo está dentro do semicírculo de 600 cm, o sensor pode

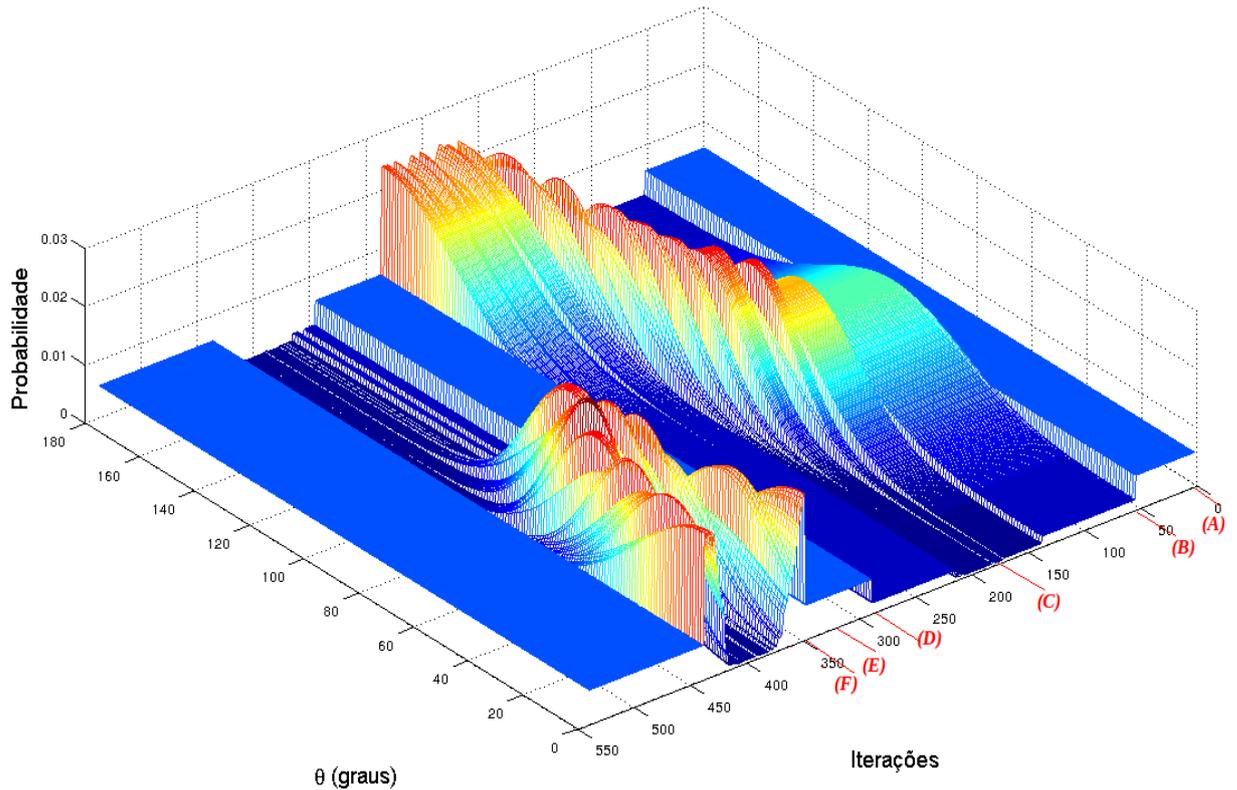


Figura 5.6: Leitura θ do obstáculo, feita pelo modelo do sensor. As letras na cor vermelha indicam as posições da Figura 5.4

detectá-lo. O robô continua aproximando-se cada vez mais do obstáculo, e um maior número de feixes do sensor fornece dados sobre sua presença, aumentando a probabilidade até seu máximo valor durante este percurso: isso ocorre próximo da iteração 211. Após esta iteração, a incerteza aumenta, tanto para ρ quanto para θ , devido ao fato de que o obstáculo se localiza na transição para a zona de umbra, ou seja, os feixes do sensor que o atingem são menos numerosos a cada iteração. Entre as iterações 288 e 346, o modelo do sensor não fornece informação sobre o obstáculo por ele encontrar-se totalmente na zona de umbra, como pode ser observado na Figura 5.4 (e).

O obstáculo entra de novo na zona de luz após a iteração 347, quando a probabilidade aumenta a medida que o obstáculo vai adentrando a área visível do sensor e os feixes do sensor que o atingem são cada vez mais numerosos. Isto ocorre até a iteração 425, quando o robô desvia e evita o obstáculo novamente. Por último, da iteração 436 até o final da simulação, o robô deixa o obstáculo para atrás e ele não é mais detectado pelo sensor.

A saída do filtro de partículas é o vetor de 300 partículas $[\{\mathbf{X}_k^i, w_k^i\}_{i=1}^{N_s}]$, calculado a cada iteração pelo algoritmo 7, e ilustrado nas Figuras 5.7, 5.8 e 5.9. A partir destas figuras até o final da serão sinalizadas as iterações onde o obstáculo permanece na zona de umbra:

- Iteração 55 sinalizada como (a) é instante em que o sensor detecta o obstáculo,

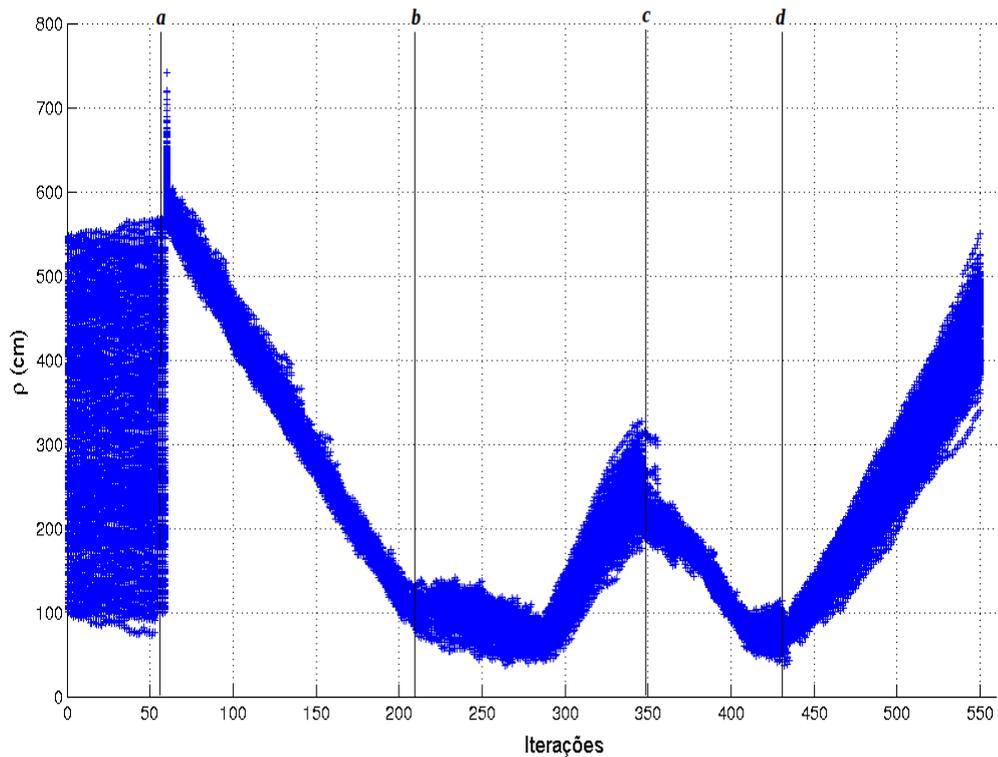


Figura 5.7: Distância ρ , em cm, de cada partícula

- A iteração 212 sinalizada como (b) onde o obstáculo começa a ficar por fora do alcance do sensor ou seja na zona de umbra, até a iteração 347 sinalizada como (c),
- Após a iteração (c) o obstáculo entra de novo na zona de luz, permanecendo até a iteração 437 sinalizada como (d), a partir de esta iteração o obstáculo permanece na zona de umbra até o final da simulação.

A Figura 5.7 e a Figura 5.8 ilustram a distância ρ e a orientação θ respectivamente, onde se localizam as partículas e a Figura 5.9 mostra o peso dessas amostras em cada iteração. As três figuras apresentam o comportamento das partículas devido as entradas do filtro, o modelo do sensor e o modelo de movimento.

Nas primeiras 50 iterações, as partículas estão distribuídas a frente do robô, a distâncias entre 70 e 570 cm, orientadas entre 10 a 170 graus, aproximadamente, e cada partícula com o mesmo peso. Como o sensor não percebe o obstáculo ainda, as partículas estão distribuídas aleatoriamente pelo ambiente a frente do robô móvel, e os pesos de todas elas é igual.

Após a iteração 52, o obstáculo é detectado no ambiente do robô móvel, e é possível verificar que as partículas respondem a essa informação do modelo do sensor. As partículas, que antes estavam aleatoriamente distribuídas, começam a se concentrar na localização observada do obstáculo. A partir da iteração 61, todas as partículas estão agrupadas e próximas a uma mesma posição, com uma baixa dispersão, em grande parte porque a etapa de reamostragem do filtro consegue manter o conjunto de partículas em torno do estado mais provável,

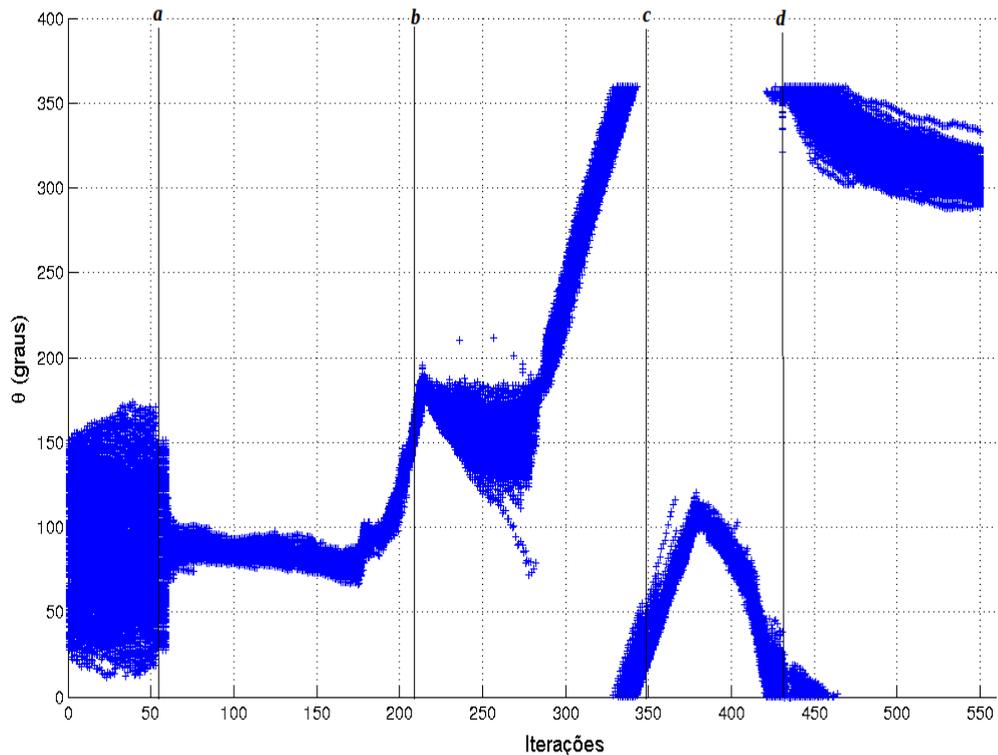


Figura 5.8: Orientação θ , em graus, de cada partícula

evitando o problema de degeneração e fornecendo uma boa aproximação da localização do obstáculo.

Quando o robô móvel executa o comportamento de rotação, em torno da iteração 211, as partículas acompanham esse movimento sem perder o alvo. Quando o obstáculo está na transição da zona de luz para zona de sombra da *ZDV*, as partículas começam a se espalhar novamente, pois elas acompanham unicamente o modelo de movimento. Isto é devido a que a partir dessa iteração o modelo do sensor fornece escassas informações da presença do obstáculo atualizando o estado de algumas partículas, e o peso das outras partículas diminui, o que vai acontecer até que nenhum feixe do sensor consiga detectar o obstáculo.

Durante a trajetória percorrida pelo robô móvel enquanto o obstáculo está totalmente na zona de sombra e não é mais visível, as partículas tornam-se cada vez mais espalhadas. O filtro de partículas permanece somente com a informação do modelo de movimento, sem efetuar a etapa de atualização e um peso uniforme é atribuído para todas as partículas, como no início da simulação. Próximos à iteração 350, pode-se observar nas três figuras que as partículas começam a se agrupar novamente, devido a presença de novas informações do sensor sobre o obstáculo.

As Figuras 5.10 e 5.11 ilustram a estimativa da distância ρ e a orientação θ , respectivamente. O valor real é representado na cor azul, que foi calculado a partir dos dados fornecidos pelo simulador Player/Stage com relação ao centro de gravidade do robô, onde também está instalado o sensor laser. Na cor vermelha, está a média ponderada de todas

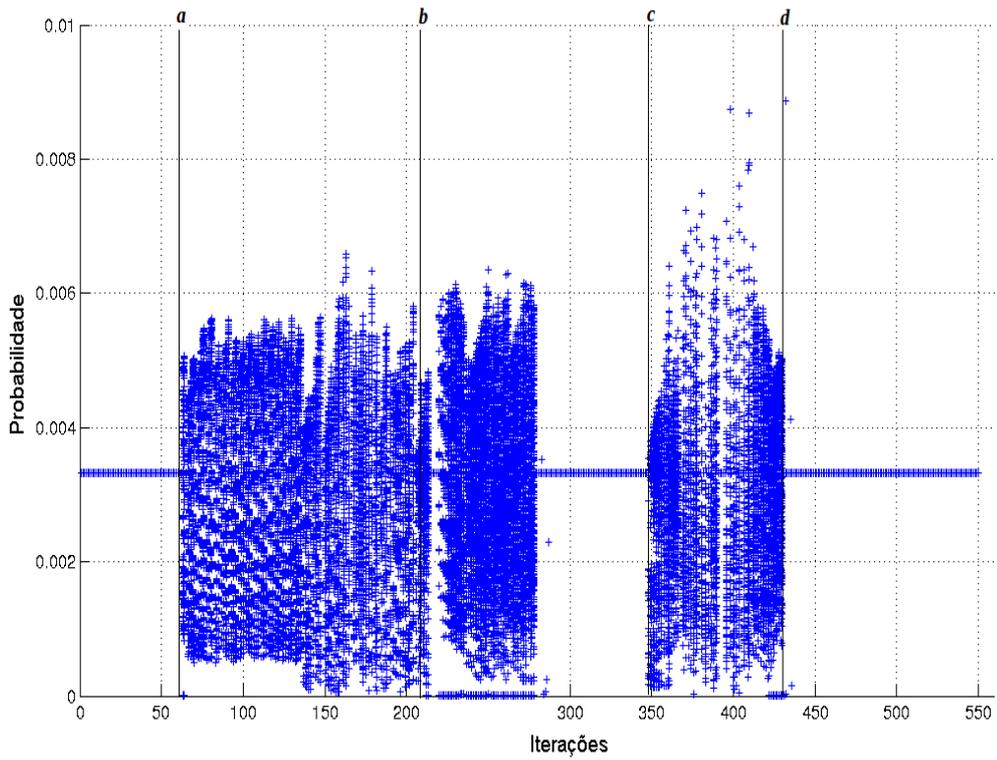


Figura 5.9: Peso w_i de cada partícula

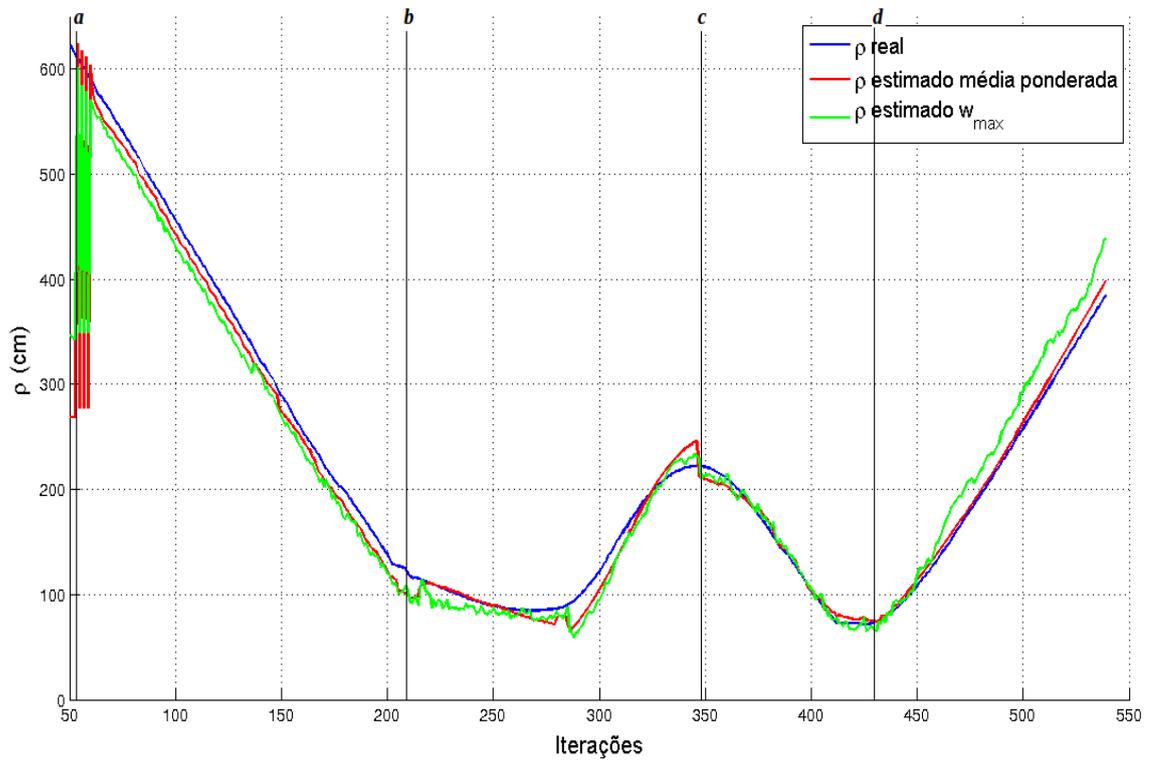


Figura 5.10: Comparação entre a distância ρ real e as estimativas da média ponderada e da melhor partícula

as partículas, calculada pela equação 4.27, e em cor verde a estimativa da melhor partícula, ou seja, aquela com maior peso w_{max} , calculada na equação 4.28. As duas figuras mostram as estimativas a partir da iteração 50, quando são observadas variações maiores nas leituras dos sensores pelo fato do obstáculo entrar no campo de detecção do sensor. Após algumas iterações, as estimativas da variável ρ acompanham de perto o valor real, mesmo quando o obstáculo está localizado na zona de umbra, em torno da iteração 250, sendo mais visível na estimativa calculada pela média ponderada.

A estimativa da orientação θ acompanha o valor real na zona de luz, mas quando o obstáculo está na transição para zona de umbra da ZDV, a estimativa de θ começa a se afastar do valor real, devido a que o modelo do sensor fornece menos informações da presença do obstáculo, permanecendo somente com o modelo de movimento. Após a iteração 350, quando o obstáculo está sendo detectado pelo sensor de novo, a estimativa tenta acompanhar o obstáculo estando completamente na zona de luz até a iteração 437.

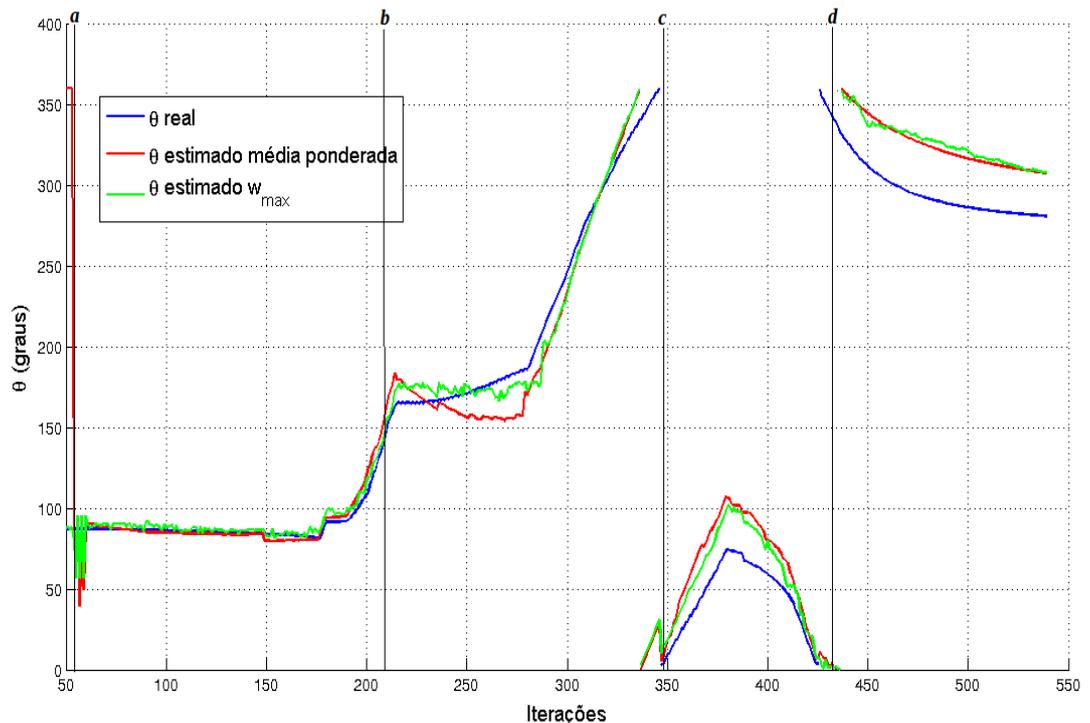


Figura 5.11: Comparação entre a orientação θ real e as estimativas da média ponderada e da melhor partícula

A Figura 5.12 ilustra a evolução do vetor de estado, \bar{X}_k , para a postura da melhor partícula e a média ponderada, no sistema de coordenadas polares, tendo como origem a posição do sensor laser (localizado a bordo do robô logo acima do seu centro de gravidade). O gráfico consiste em uma circunferência com raio de aproximadamente 800 cm, que abrange a área de alcance máximo de detecção do sensor 600 cm na frente.

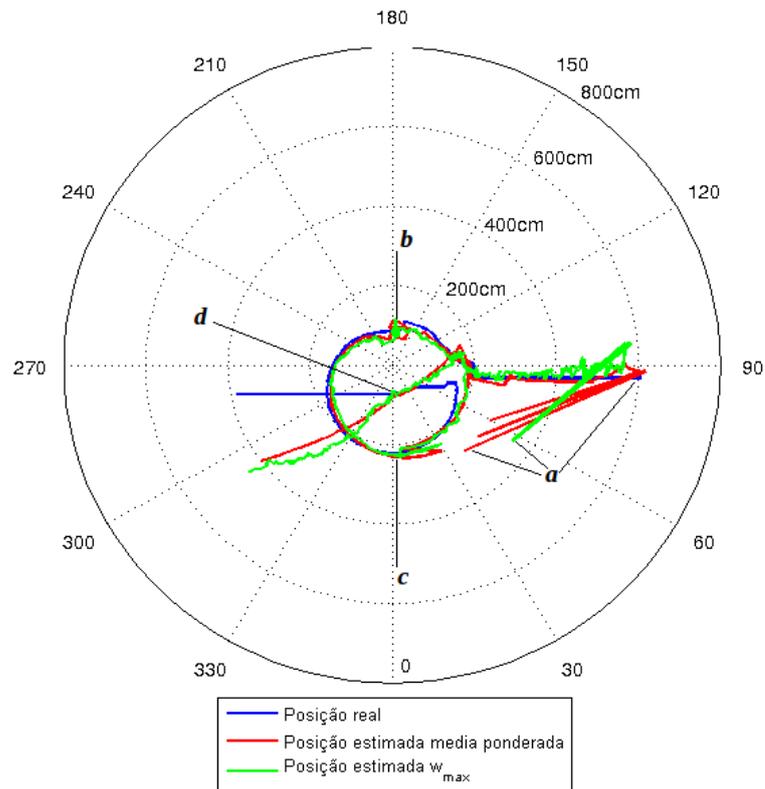


Figura 5.12: Comparação da posição do obstáculo no sistema de coordenadas polares do robô móvel.

Os dados apresentados neste gráfico correspondem aproximadamente ao instante em que a primeira leitura do sensor detecta a presença do obstáculo próximo ao robô, ou seja, na iteração 55 sinalizada como ponto (a). Nesta iteração a estimativa calculada da média ponderada na cor vermelha é no ângulo $59,58^\circ$ e da partícula mais pesada na cor verde é $57,00^\circ$ mais o valor real do ângulo em que se encontra o obstáculo é $87,16^\circ$ na cor azul. A partir desta iteração as partículas são atualizadas e após cinco iterações do algoritmo as partículas começam a se agrupar, e a estimativa da média ponderada é $88,58^\circ$ e da partícula mais pesada é $95,59^\circ$ sendo o valor real no ângulo $87,07^\circ$. É também sinalizada a iteração (b) onde o obstáculo passa para a zona de umbra pelo ângulo de 180° e as iterações (c) e (d) onde o obstáculo passa da zona da umbra para a luz e da luz para a umbra respectivamente.

É possível ver que as duas estimativas da postura do obstáculo (média ponderada na cor vermelha e a partícula mais pesada em verde) acompanham a posição real do obstáculo, mostrada na cor azul.

Tendo como base a posição real do obstáculo calculada a partir dos dados fornecidos pelo simulador Player/Stage, é possível determinar o erro relativo a cada iteração entre as posições estimadas \bar{X}_k (média ponderada e a melhor partícula) e a posição real do obstáculo, de acordo com a equação:

$$Erro\ relativo = \frac{Valor\ medido - Valor\ real}{Valor\ real}, \quad (5.1)$$

As Figuras 5.13 e 5.14 ilustram o erro relativo das estimativas de ρ e θ respectivamente. Nas duas figuras, o obstáculo primeiro se localiza na zona de luz desde a iteração (a) até a iteração sinalizada como (b), onde inicia a transição para a zona de sombra, e a cada iteração menos feixes do sensor atingem o obstáculo. Entre as iterações (b) e (c), o obstáculo se localiza totalmente na zona de sombra. Entre as iterações (c) e (d), o obstáculo volta para a zona de luz, e finalmente passa de novo para a zona de sombra desde a iteração (d) até o final da simulação.

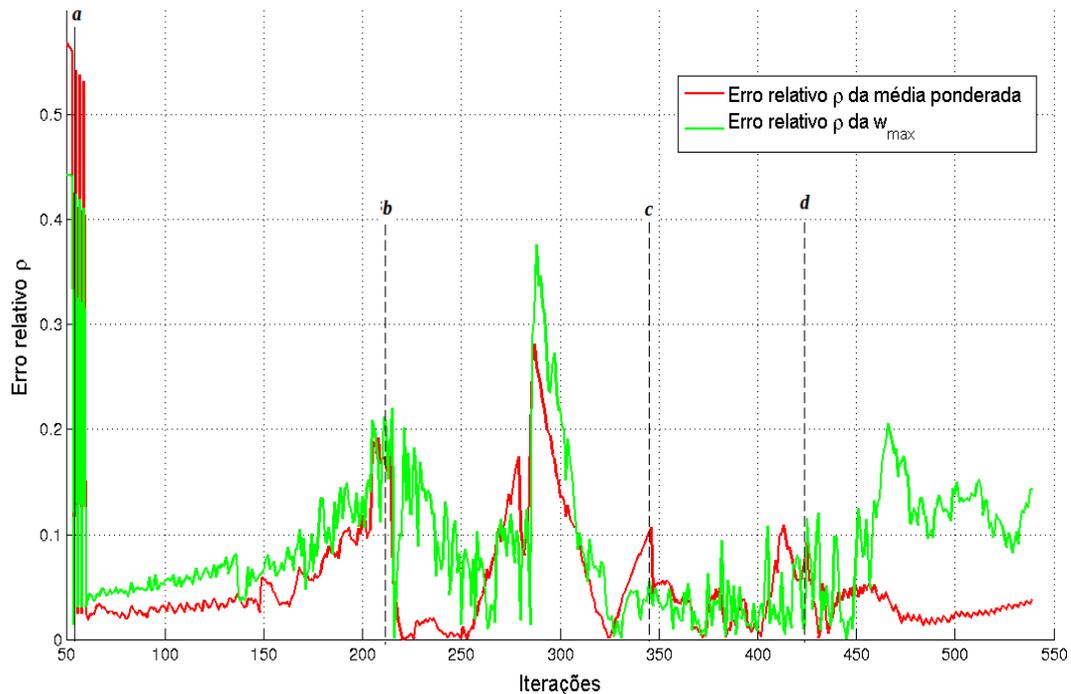


Figura 5.13: Erro relativo da estimativa ρ , da média ponderada e da melhor partícula.

A figura 5.13 compara o erro relativo das estimativas da distância ρ apresentadas na figura 5.10, onde se verifica que as estimativas do filtro apresentam grandes variações no início, até o momento em que o obstáculo é detectado pelo sensor, bastam algumas iterações para essas estimativas acompanharem o valor real da distância ρ com pequenas variações. Quando o robô executa a rotação para evitar a colisão, observa-se um aumento do erro, que torna-se ainda maior enquanto o obstáculo permanece na zona de sombra. Um exame mais atento da variação de cada um dos erros indica que o valor máximo do erro para a estimativa da melhor partícula está em torno de 38% , enquanto que para a média ponderada o erro máximo está em torno de 29%.

O erro relativo da orientação θ é apresentado na Figura 5.14, onde pode-se observar que as duas estimativas tem um erro baixo, mesmo quando o obstáculo está na zona de sombra sinalizadas entre as iterações (b) e (c), e desde a iteração (d) até o final da simulação. Entre

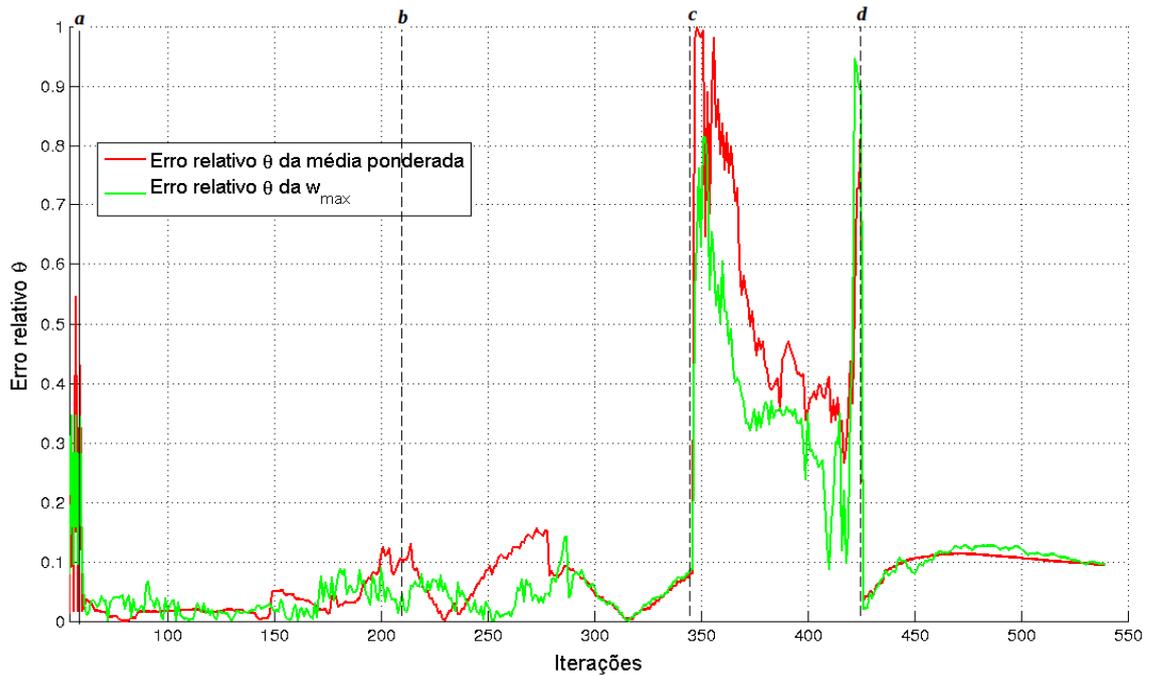


Figura 5.14: Erro relativo de θ da média ponderada e da melhor partícula

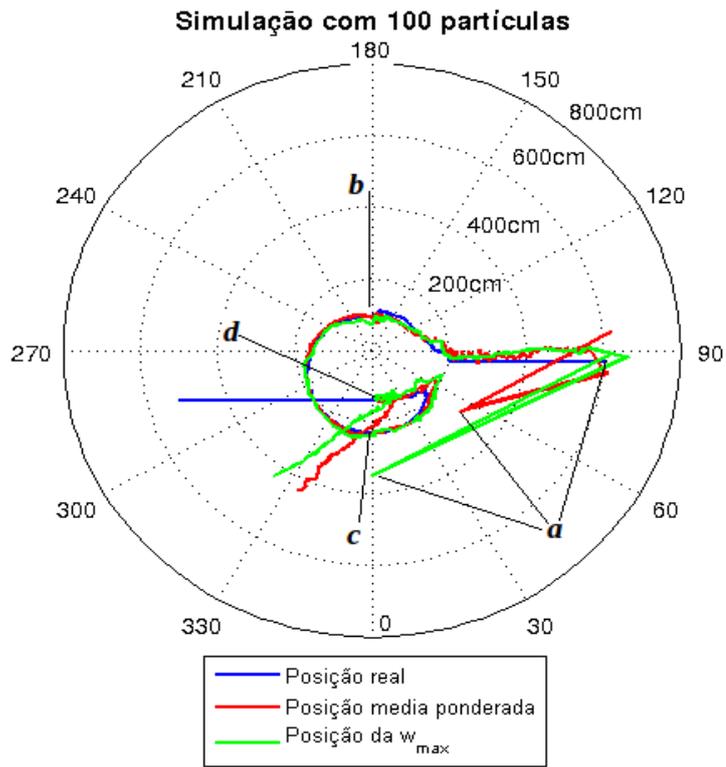
as iterações (c) e (d) o obstáculo se encontra na zona de luz mas está fazendo a transição do ângulo de 359° (graus) para o ângulo 0° (graus) e de acordo com a equação (5.1) do erro relativo este valor é alto devido a que o valor real esta entre 0° e 5° e o valor medido é menor de 360° . Por exemplo, quando o valor verdadeiro é em torno de 10 graus, o valor estimado é de 5 graus: o erro relativo é 50%, mas as estimativas calculadas na simulação acompanham os valores reais sem chegar a ser o duplo.

A partir dessa simulação é possível verificar que foi alcançado o objetivo pretendido no início desde capítulo. Baseado na análise dos dados obtidos e mostrados nos gráficos pode se verificar que é possível seguir o obstáculo com filtro de partículas integrando com a ZVD, mesmo quando o obstáculo está fora do alcance do sensor. Também é possível estimar a posição do obstáculo assumindo que ele está imóvel. Com relação aos erros da distância e do ângulo pode-se dizer que aumentam aos poucos, a medida que o filtro fica mais tempo sem leituras do sensor.

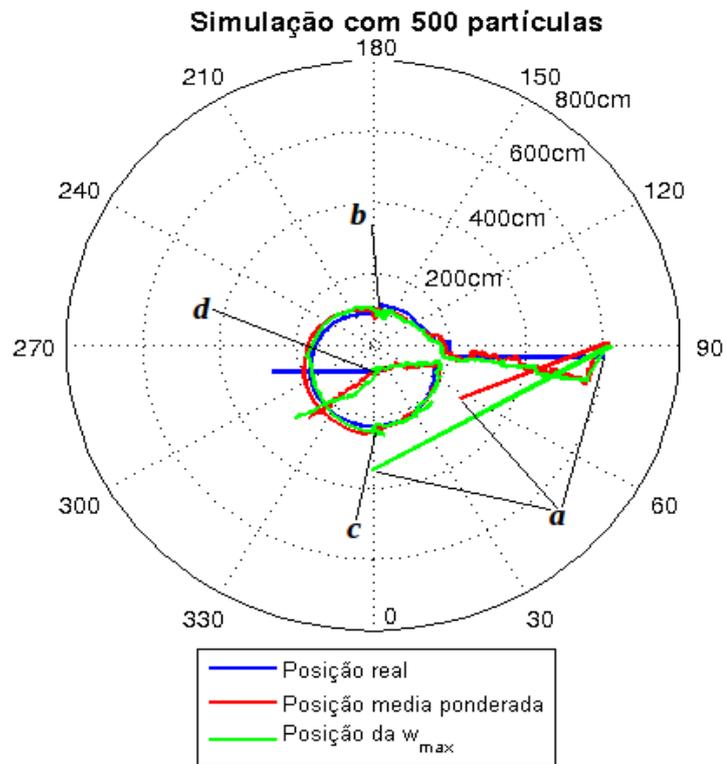
5.2.1 Comparação dos erros variando o número de partículas

A simulação descrita na Figura 5.3, foi realizada variando a quantidade de partículas, com a finalidade de observar a sua influência no comportamento do filtro no cálculo da estimativa da pose do obstáculo.

As posições plotadas nas Figuras 5.15 (a) e 5.15 (b), representam os valores estimados, em cada iteração da evolução do vetor de estado, $\bar{\mathbf{X}}_k$ da média ponderada e da melhor partícula para a simulação com $N_s = 100$ e $N_s = 500$ partículas respectivamente. Nos dois



(a)



(b)

Figura 5.15: Postura do obstáculo no sistema de coordenadas polares do robô móvel a) com 100 partículas, b) com 500 partículas

casos os gráficos iniciam no instante em que a primeira leitura do sensor detecta a presença do obstáculo próximo ao robô.

No caso da simulação com $N_s = 100$ partículas o calculo da posição do obstáculo na iteração 55 sinalizada como (a), fornecido pela média ponderada é no ângulo $53,25^\circ$ com distância $286,50cm$, e o calculo da partícula mais pesada é no ângulo $0,07^\circ$ com distância $349,99cm$ sendo o valor real no ângulo $87,17^\circ$ e distância $608,73cm$.

Da mesma forma no caso de $N_s = 500$ partículas o calculo da posição do obstáculo na iteração 55 sinalizada como (a), fornecido pela média ponderada é no ângulo 57° com distância $273,13cm$, da partícula mais pesada é no ângulo $359,76^\circ$ com distância $349cm$ sendo o valor real no ângulo $87,0^\circ$ e distância $590,76cm$.

A seguir, são apresentadas e analisadas separadamente as estimativas das variáveis que constituem o vetor de estados \bar{X}_k para diferentes valores do número de partículas. A Figura 5.16 (a) mostra o erro da média ponderada da estimativa da variável ρ nas três simulações. Observa-se um erro baixo e contínuo, mas quando as partículas começam a se espalhar na zona de umbra o erro aumenta, isso indica que esta estimativa deve ser usada preferencialmente quando a certeza é alta e todas as partículas estão juntas.

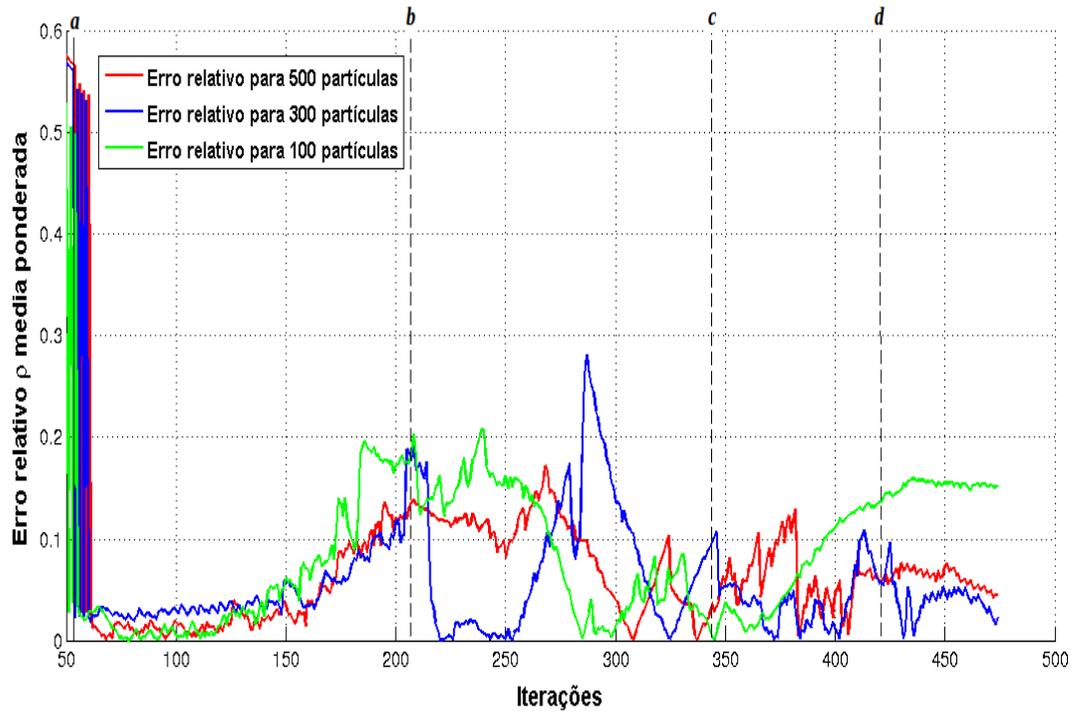
Porém, o mesmo não ocorre na Figura 5.16 (b), onde a variável ρ da melhor partícula, apresenta muitas variações durante toda a trajetória nas três simulações. Isso ocorre pelo fato de que as partículas estão movimentando-se de um lado para outro ao redor do valor real, e por ser um processo estocástico, a melhor partícula não é sempre a mesma: ou seja, a partícula de maior peso está variando em torno da postura real, fazendo uma estimativa aproximada, mas não precisa.

De forma semelhante, o erro da estimativa da variável θ é apresentado nas Figuras 5.17 (a) e 5.17-(b). Observa-se um baixo erro, que aumenta quando a robô executa a rotação, sendo maior na transição entre as zonas de umbra e de luz pelo ângulo de 359° (graus) para o ângulo 0° (graus) devido ao cálculo do erro relativo, equação (5.1) o que foi explicado na seção anterior para as Figuras 5.13 e 5.14.

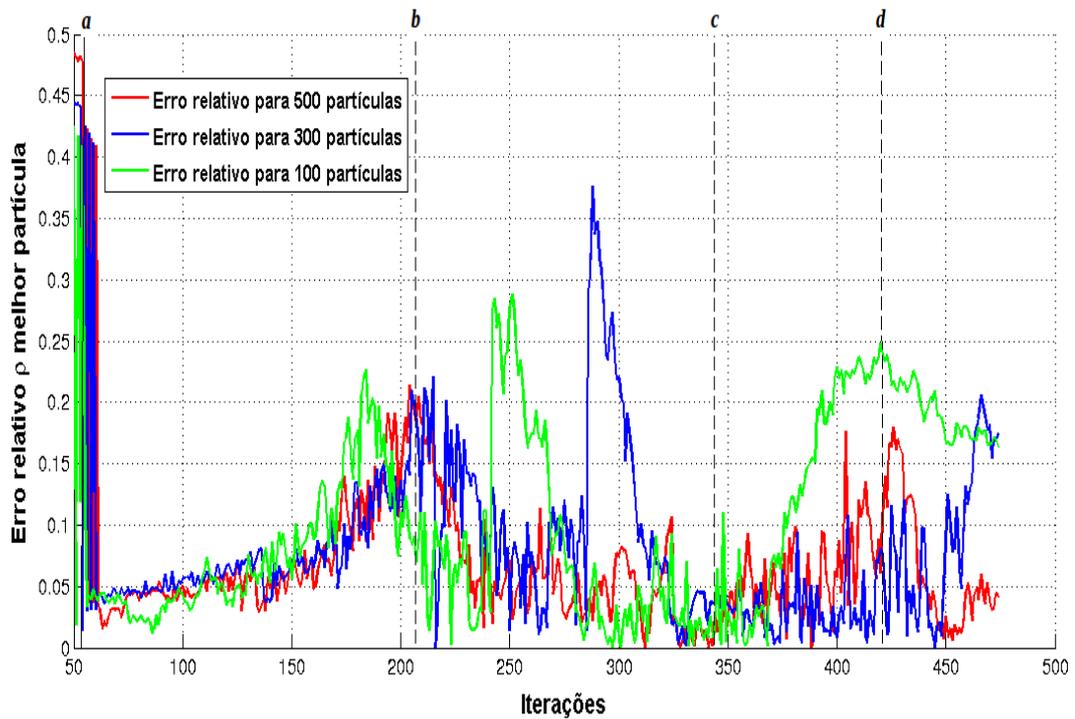
A partir dos dados anteriormente apresentados nas três simulações, é possível determinar o percentual do erro médio (%ErM) da estimativa das variáveis de interesse ρ e θ , a partir do momento em que o sensor laser fornece os dados da localização do obstáculo pelo modelo do sensor no ambiente do robô, até o final da simulação. O calculo é feito de acordo com a equação:

$$\%ErM = \frac{\sum \left(\frac{\text{Valor medido} - \text{Valor real}}{\text{Valor real}} \right)}{\text{Total de iteraes}}. \quad (5.2)$$

A tabela 5.1 apresenta o percentual da média do erro relativo das estimativas para os três casos simulados: 100, 300 e 500 partículas. Verifica-se que o erro percentual, tanto para ρ quanto para θ , é menor na simulação feita com 500 partículas, em ambos os casos



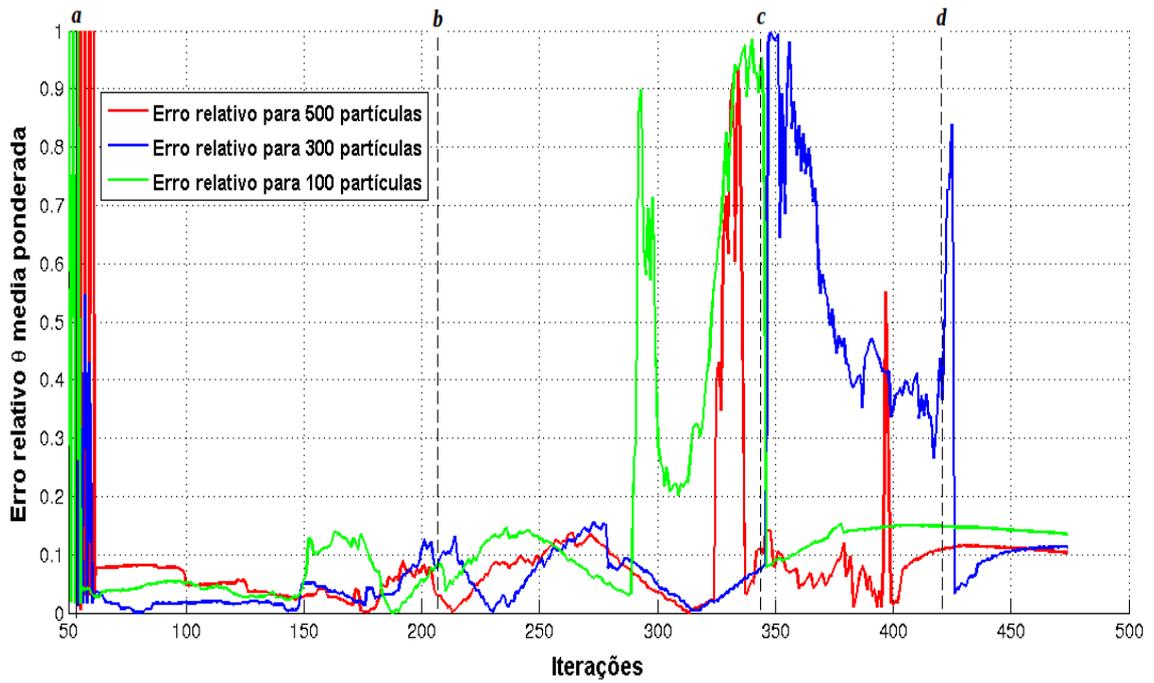
(a)



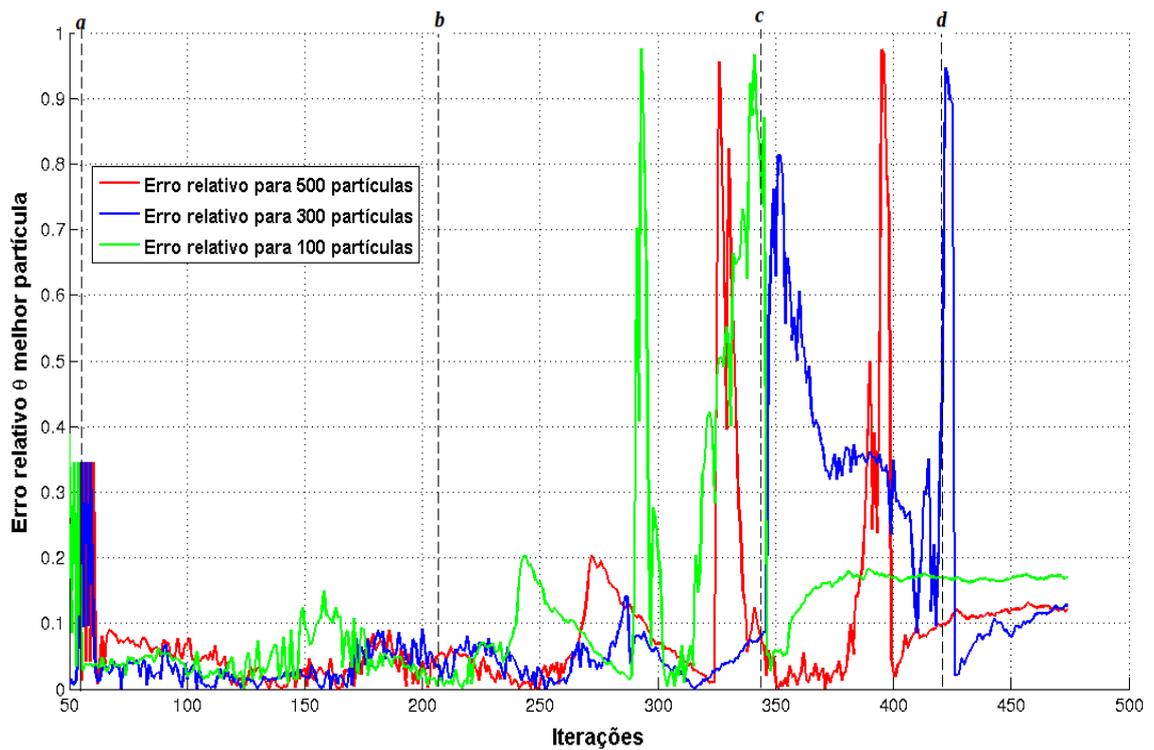
(b)

Figura 5.16: Comparação do erro relativo ρ a) da média ponderada b) da melhor partícula (estimativa da média ponderada e da melhor partícula). Também é possível identificar que para a estimativa da melhor partícula na variável θ , o valor médio do erro percentual diminui quando o número de partículas aumenta.

O erro na variável de estado θ é maior na simulação do caso de 100 partículas, tanto



(a)



(b)

Figura 5.17: Comparação do erro relativo θ a) da média ponderada b) da melhor partícula na média ponderada como na melhor partícula, devido ao fato de que o movimento de rotação possui um erro associado. Este erro fará com que as partículas se dispersem ainda mais e aumente o erro da variável estimada. No caso de 100 partículas, tem-se uma menor probabilidade de que uma das partículas esteja localizada na posição correta ou próxima a

ela.

Quantidade de Partículas		% Erro Médio Estimativa ρ	% Erro Médio Estimativa θ
$N_s = 100$	Média Ponderada	8.3%	15.6%
	Melhor Partícula	10.1%	13.2%
$N_s = 300$	Média Ponderada	5.3%	14.2%
	Melhor Partícula	8.6%	11.5%
$N_s = 500$	Média Ponderada	6.1%	8.6%
	Melhor Partícula	6.3%	8.5%

Tabela 5.1: Média do Erro Percentual da estimativa ρ e θ desde o primeiro dado fornecido pelo sensor até o fim da primeira simulação.

5.2.2 Erro das estimativas na zona de umbra

Um dos objetivos deste trabalho consiste em estimar a posição do obstáculo quando o obstáculo se situa na zona de umbra, após o robô ter evitado a colisão. É desejado manter uma boa estimativa das variáveis ρ e θ da posição desse obstáculo nesta zona. Obviamente que isto só pode acontecer após terem sido calculadas estimativas destas variáveis primeiramente na zona de luz, com base nas informações fornecidas pelo sensor a cada iteração.

A tabela 5.2 mostra a média do erro percentual das estimativas das variáveis ρ e θ somente para os instantes de simulação onde o sensor não detecta o obstáculo, ou seja, quando este está na zona de umbra. Observa-se que o erro em geral é maior que durante a trajetória completa (mostrado na tabela 5.1). Também é possível observar que o erro continua sendo menor na simulação com 500 partículas, mesmo quando o sensor não fornece informações.

Quantidade de Partículas		% Erro Médio Estimativa ρ	% Erro Médio Estimativa θ
$N_s = 100$	Média Ponderada	15.4%	20.2%
	Melhor Partícula	20.2%	21.3%
$N_s = 300$	Média Ponderada	10.3%	18.1%
	Melhor Partícula	16.0%	17,1%
$N_s = 500$	Média Ponderada	11.4%	17.8%
	Melhor Partícula	10.6%	19.4%

Tabela 5.2: Porcentagem de erro médio da estimativa ρ e θ da primeira simulação durante a trajetória pela zona de umbra.

5.3 SIMULAÇÃO DO ROBÔ COM MOVIMENTO ALEATÓRIO DENTRO DE UMA ÁREA COM UM OBSTÁCULO

Nesta seção são apresentados os resultados obtidos em um segundo experimento, onde o robô móvel navega partindo da posição inicial $(-1, 0, 0^\circ)$, com um obstáculo na posição $(7, -0.30)$. Nesta simulação, o tempo de execução é maior que na simulação anterior, pois o robô móvel aproxima-se do obstáculo e movimenta-se ao redor dele de forma aleatória, mas sem ultrapassar limites estabelecidos no sistema de coordenadas do mundo na janela do Player/Stage. Para tanto, foram colocadas restrições no algoritmo para que o movimento aleatório não afaste muito o robô do obstáculo. Quando o robô supera os valores limites de distância máxima até o obstáculo, ele executa o comportamento de rotação até aproximar-se de novo do obstáculo, e recomeça a movimentar-se de forma aleatória. Isto assemelha-se ao comportamento do robô dentro de um ambiente fechado com um obstáculo fixo no centro.

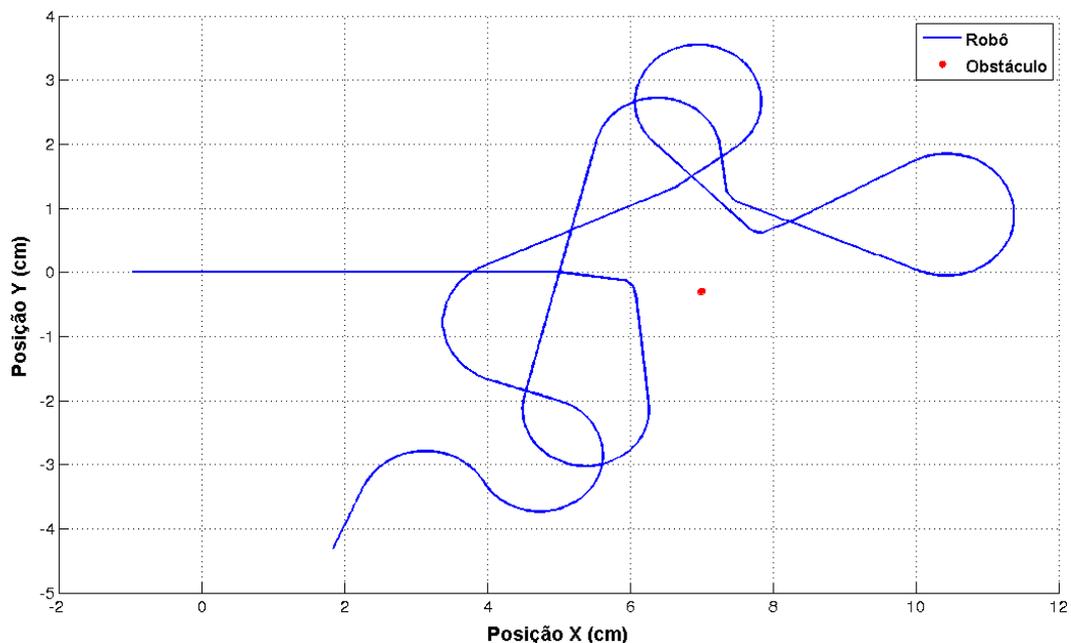
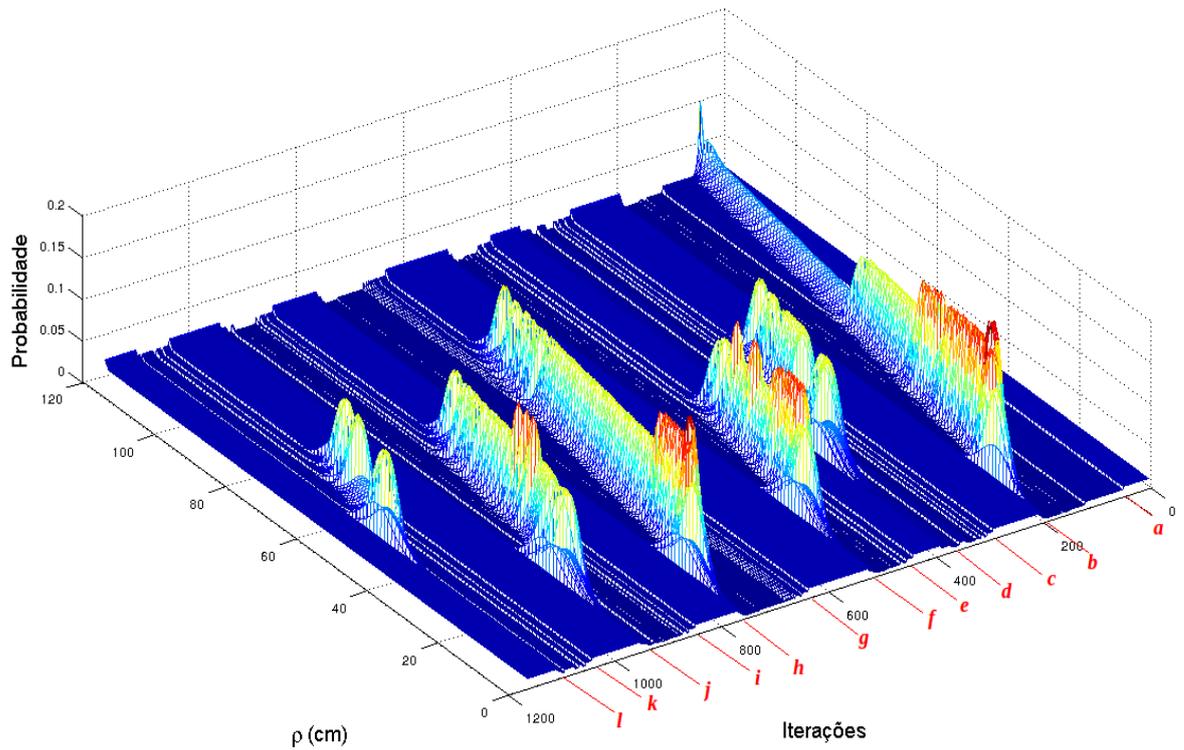


Figura 5.18: Trajetória percorrida pelo robô móvel na simulação com movimento aleatório dentro de uma área com um obstáculo.

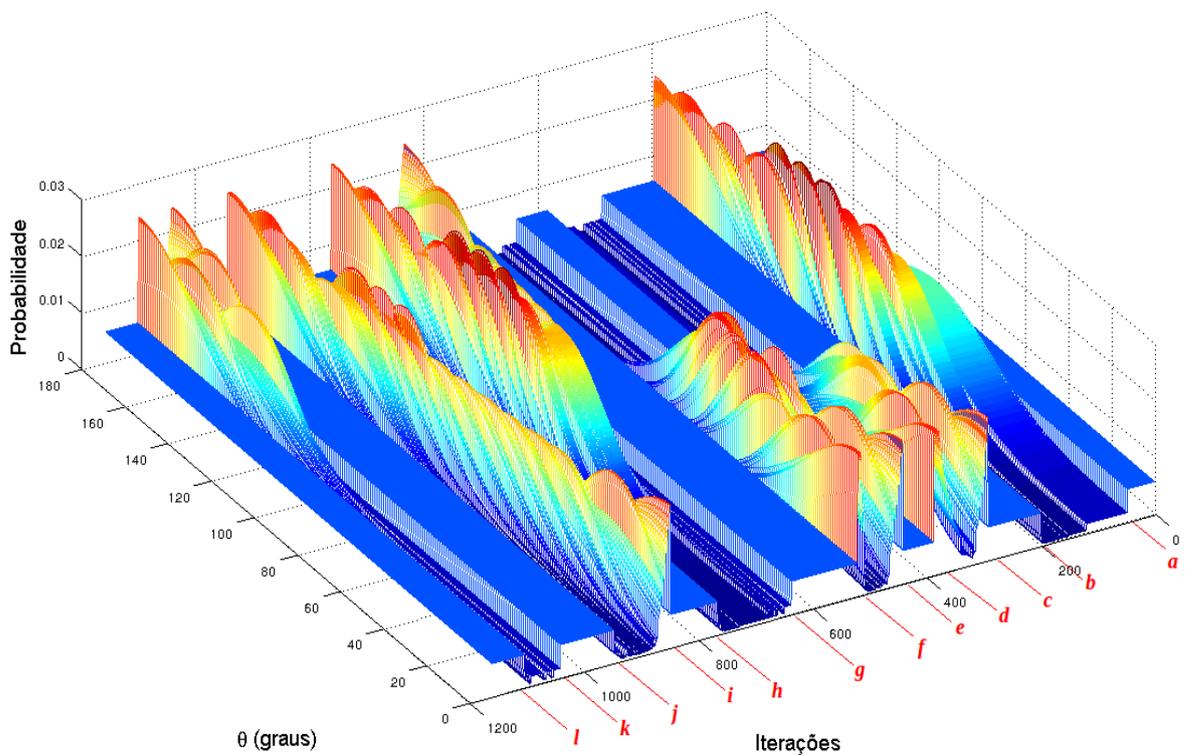
O caminho percorrido pelo robô móvel nesta segunda simulação está ilustrado na cor azul na Figura 5.18. Pode-se observar que, no início, o robô móvel avança na direção do obstáculo, passa na posição de coordenada $x=4$, $y=0$, e entra na zona em torno do obstáculo que delimita seu movimento aleatório. Verifica-se que o robô móvel saiu 6 vezes da zona limitada, e executou rotações até obter informações da localização do obstáculo pelo sensor.

As Figuras 5.19 (a) e 5.19 (b) ilustram as informações fornecidas pelo modelo do sensor da função de probabilidade $P(\rho_k^i, \mathbf{X}_k)$ e $P(\theta_k^i, \mathbf{X}_k)$, respectivamente. Observa-se que enquanto o robô movimenta-se, o modelo do sensor fornece informações da posição do obstáculo com probabilidades baixas e, com o passar das iterações, a probabilidade aumenta

devido ao fato que um maior número de feixes do sensor laser detectam o obstáculo.



(a)



(b)

Figura 5.19: Informações fornecidas pelo modelo do sensor a) função de probabilidade $P(\rho_k^i, X_k)$, b) função de probabilidade e $P(\theta_k^i, X_k)$

Este segundo experimento realiza a simulação com $N_s = 500$ partículas, devido a que com esta quantidade de partículas foram obtidos os menores erros nas estimativas. As seguintes figuras apresentam a saída do filtro de partículas $[\{\mathbf{X}_k^i, w_k^i\}_{i=1}^{N_s}]$ nesta simulação.

A partir destas figuras até o final da seção serão sinalizadas as iterações onde o obstáculo permanece na zona de umbra:

- Iteração 62 sinalizada como (a) é instante em que o sensor detecta o obstáculo,
- A iteração 198 sinalizada como (b) onde o obstáculo começa a ficar por fora do alcance do sensor, ou seja, na zona de umbra, até a iteração 295 sinalizada como (c),
- Iteração 376 sinalizada como (d),
- Iteração 447 sinalizada como (e),
- Iteração 525 sinalizada como (f),
- Iteração 638 sinalizada como (g),
- Iteração 771 sinalizada como (h),
- Iteração 846 sinalizada como (i),
- Iteração 926 sinalizada como (j),
- Iteração 1022 sinalizada como (k),
- Iteração 1107 sinalizada como (l) a partir dessa iteração o obstáculo permanece na zona de umbra até o final da simulação.

As Figuras 5.20 e 5.21 apresentam a distribuição da variável ρ e θ respectivamente da posição do robô (no sistema de coordenadas polares do robô) das partículas a cada iteração. É possível observar que durante os trajetos em que o obstáculo é detectado pelo sensor, as partículas permanecem próximas e concentradas em torno da posição real. É visível que a dispersão aumenta ao fazer a transição entre as zonas e ao localizar-se na umbra.

A probabilidade de cada partícula durante toda a trajetória é ilustrada na Figura 5.22, onde se verifica que em algumas iterações, um grupo pequeno de partículas estão com uma probabilidade muito maior que o resto, em outras palavras, estas partículas estão localizadas mais próximas da posição real do obstáculo. Destaca-se uma irregularidade nos pesos das partículas após a iteração 650, quando o robô executa o comportamento de rotação e o obstáculo é de novo localizado pelo modelo do sensor (ângulo 180° com uma distância de aproximadamente 470 cm). Embora algumas partículas apresentam alta probabilidade, a maior parte delas tem baixa probabilidade durante este deslocamento, fazendo com que o

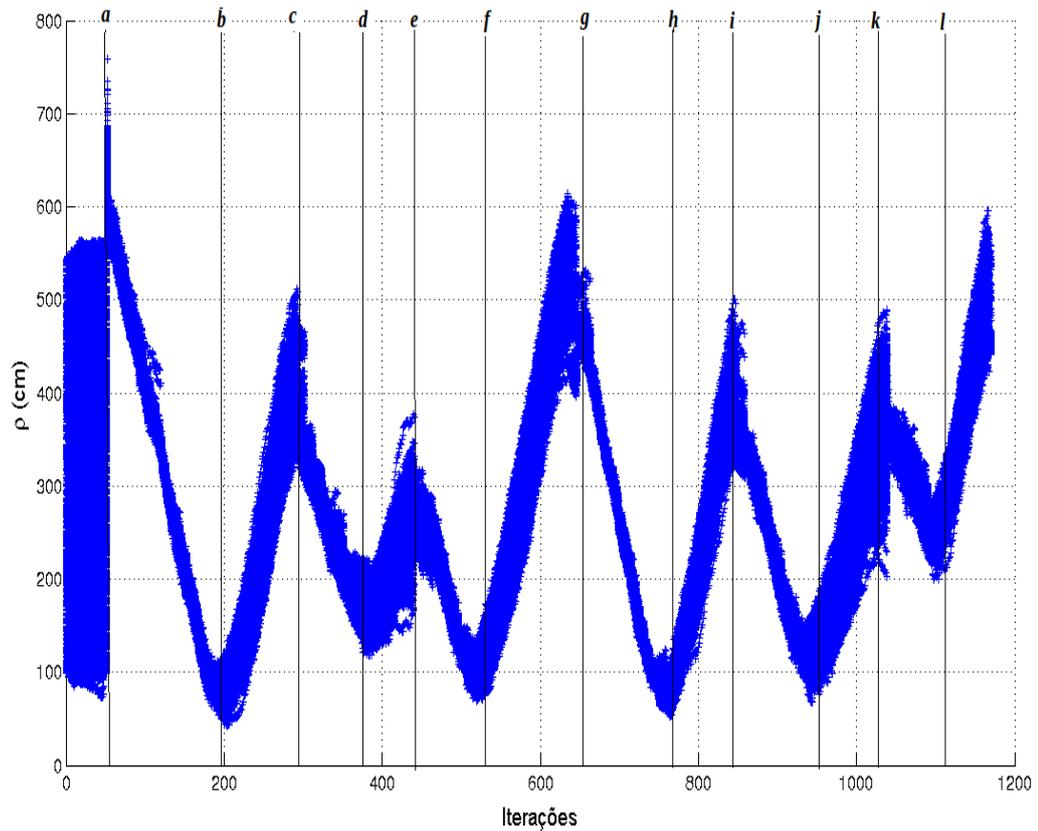


Figura 5.20: Distância ρ , em cm, da saída do filtro

cálculo da variável \hat{N}_{eff} , no algoritmo do filtro, ultrapasse o limite N_T estabelecido para a execução do algoritmo de reamostragem do algoritmo 3.

É possível observar nas Figuras 5.23 e 5.24 as posições obtidas das estimativas de ρ e θ respectivamente, tanto para a média ponderada como para a melhor partícula após a iteração 62 sinalizada por (a) onde o obstáculo é detectado pelo sensor.

Verifica-se que as duas estimativas do algoritmo acompanham com sucesso a posição real. Nas duas figuras, é possível observar a saída do algoritmo nas primeiras iterações quando o obstáculo adentra a zona de luz. A atualização da posição das partículas se realiza a cada iteração desde o momento que o sensor detecta o obstáculo.

O obstáculo permanece na zona de luz durante as iterações desde o início da iteração (a) até a iteração sinalizada como (b), entre as iterações c e d , entre e e f , entre g e h , entre i e j e entre k e l . Nas outras iterações o obstáculo se encontra na zona de umbra.

A comparação dos erros relativos das estimativas da variável ρ é ilustrada na Figura 5.25 e da variável θ na Figura 5.26. Os valores do erro relativo em relação à posição real do obstáculo para a variável ρ oscilam entre 0 e 0.43 para a melhor partícula e apresenta um valor máximo de 0.37 para a média ponderada durante toda a simulação. No erro relativo da variável θ Figura 5.26, são visíveis incrementos nas duas estimativas, devido ao fato dos trajetos onde o obstáculo estão na zona de umbra (entre 181° e 359°) e especificamente na

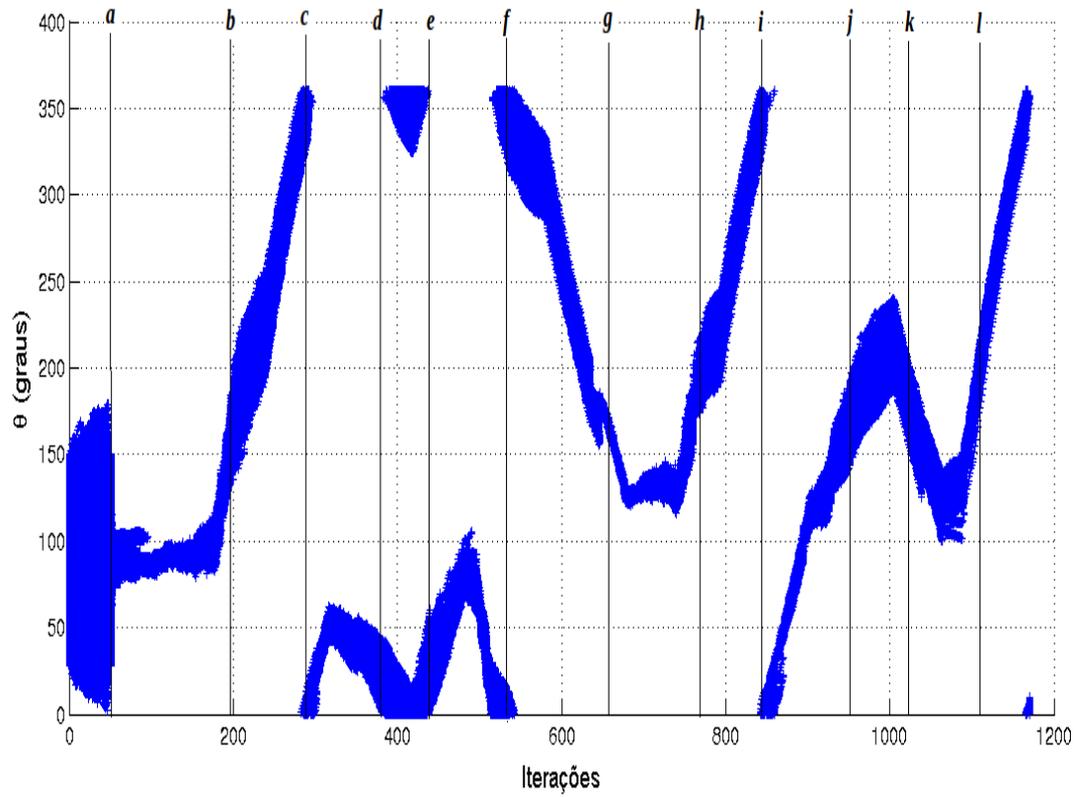


Figura 5.21: Distância θ , em graus, da saída do filtro

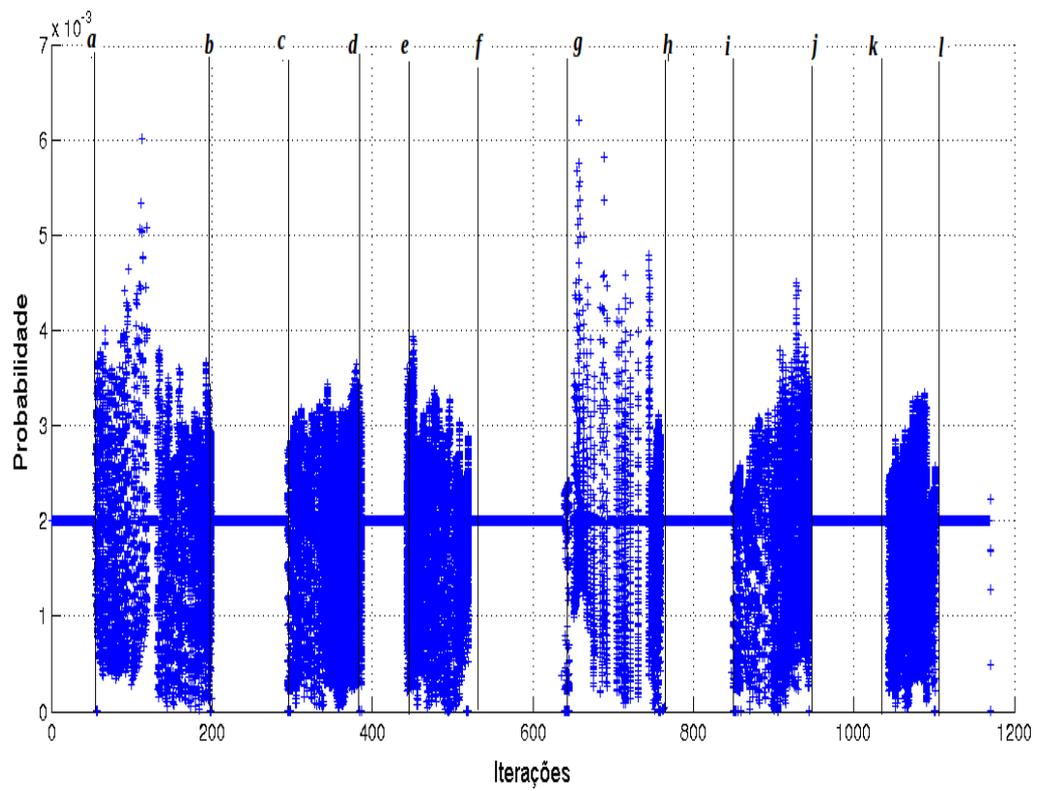


Figura 5.22: Peso w_i para cada partícula

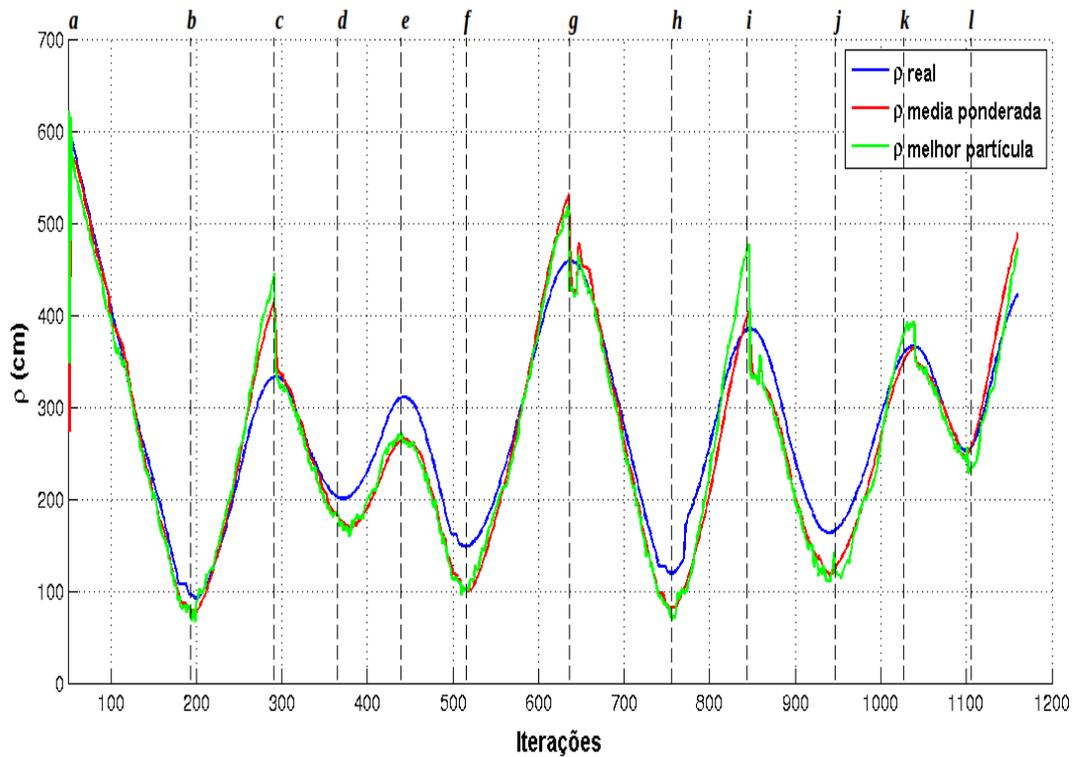


Figura 5.23: Comparação da distância real ρ e as estimativas do filtro

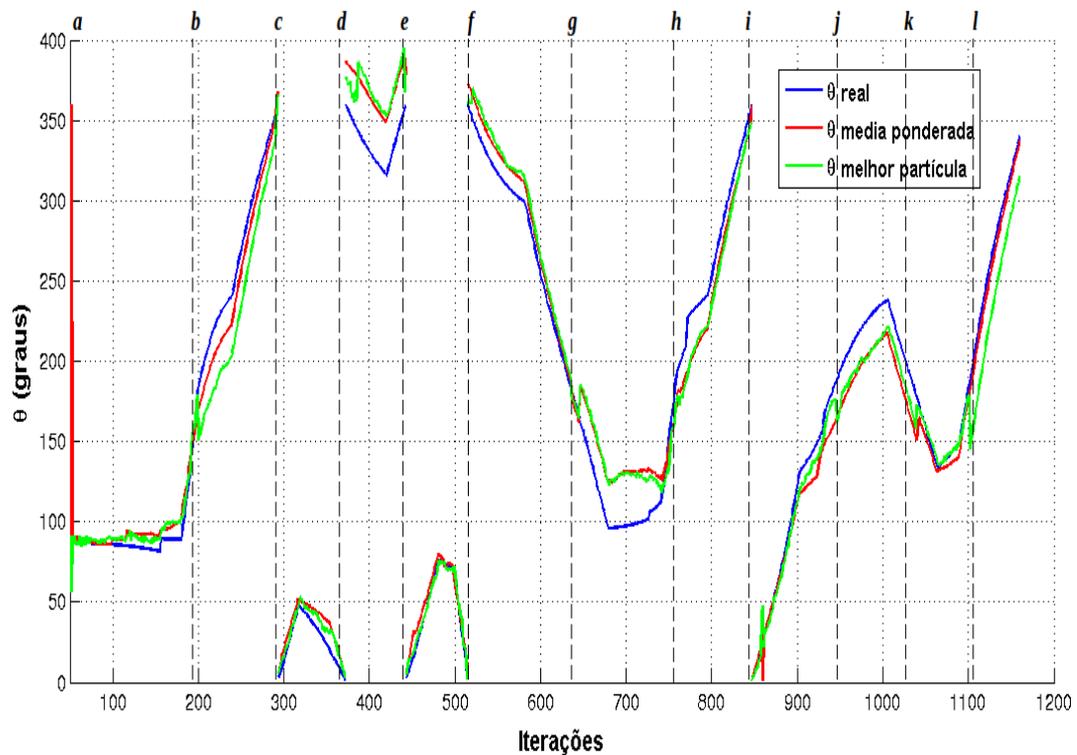


Figura 5.24: Comparação do valor real ρ e as estimativas do filtro

transição entre as zonas, pelo ângulo 359° graus para o ângulo 0° devido ao cálculo do erro relativo, explicado na seção anterior para as figuras figuras 5.13 e 5.14. Analisando os dados

de θ , sem levar em consideração os incrementos, os valores de erro médio estão entre 0 e 0.35 para a média ponderada e para a melhor partícula o valor máximo é 0.34.

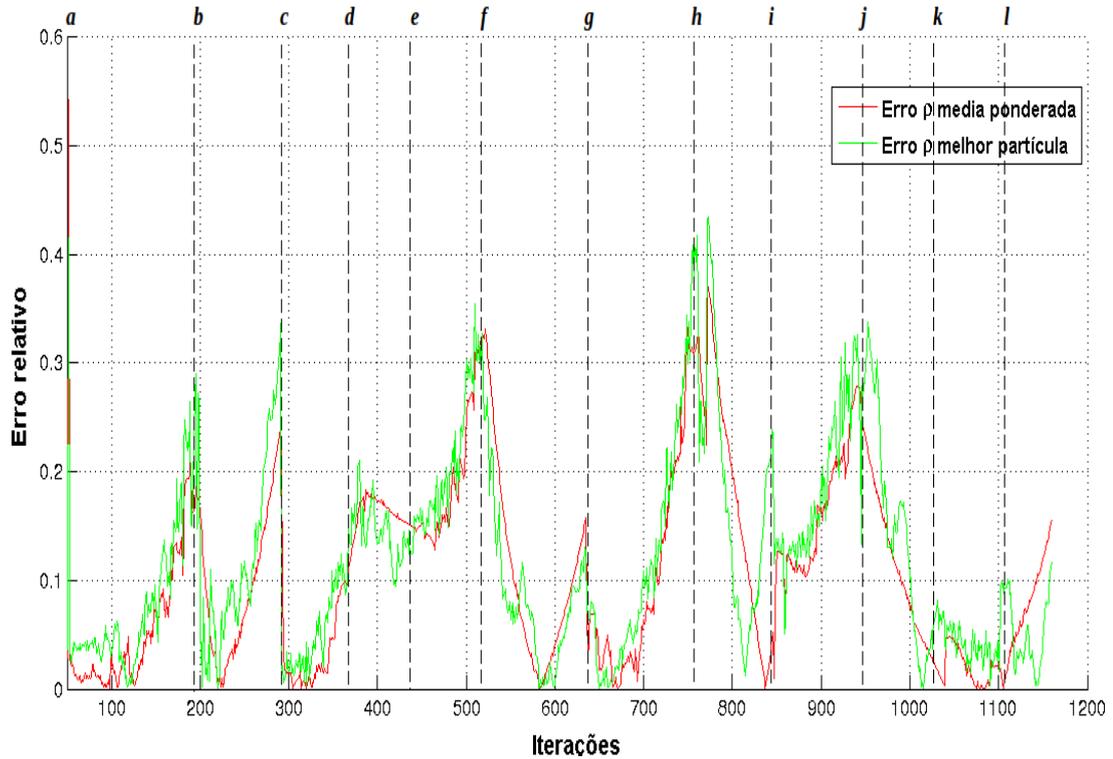


Figura 5.25: Comparação do erro relativo, da média ponderada e da melhor partícula na estimativa de ρ

Quantidade de Partículas		% Erro Médio Estimativa ρ	% Erro Médio Estimativa θ
$N_s = 500$	Média Ponderada	11.8%	14.0%
	Melhor Partícula	11.9%	11.7%

Tabela 5.3: Porcentagem de erro médio da estimativa ρ e θ desde o primeiro dado fornecido pelo sensor até o fim da segunda simulação.

Na tabela 5.3 apresenta-se a porcentagem do erro médio da estimativa durante toda a segunda simulação. Verifica-se que a estimativa da média ponderada em relação à variável θ é maior que da melhor partícula. Além do motivo apresentado na subseção 5.2.1, nesta simulação o número de iterações é maior e uma maior quantidade de comportamentos de rotação são executados, o que espalha as partículas com a redução da certeza.

5.3.1 Erro das estimativas na zona de umbra

Na análise das seis trajetórias do robô quando o obstáculo sai do campo de visão do sensor laser e fica somente na zona de umbra, é importante lembrar que o filtro de partículas para

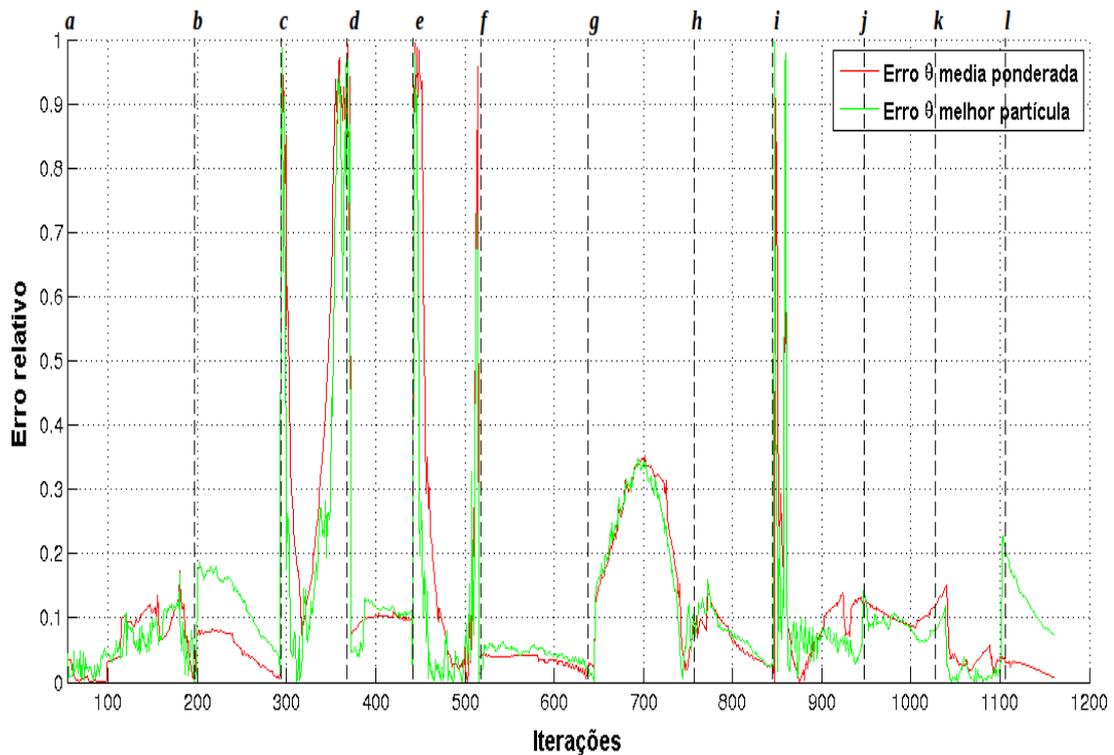


Figura 5.26: Comparação do erro relativo, da média ponderada e da melhor partícula na estimativa de θ

estas trajetórias só recebe informação do modelo de movimento. Nesta segunda simulação, devido ao fato de que o robô executa trajetórias aleatórias, o obstáculo permanece fora do campo de visão do sensor em 6 ocasiões, sinalizadas na Figura 5.24.

Verifica-se que nas Figuras 5.27 (a), 5.28 (a), 5.28 (b) e 5.29 (b), onde as trajetórias percorridas pelo robô são maiores, as estimativas geralmente acompanham a posição real com sucesso. Analisando de forma detalhada, observa-se que desde as primeiras iterações em que o obstáculo fica totalmente na umbra, as estimativas estão afastadas da posição real, devido ao fato de que a cada iteração uma menor quantidade de raios do sensor laser atingem o obstáculo. Com o decorrer do tempo, verifica-se que as estimativas começam a se afastar, de forma gradual, do valor real já que nessas trajetórias não existem informações do modelo do sensor para atualizar os pesos das partículas, e elas se espalham cada vez mais.

No conjunto de trajetórias na zona da umbra, mostradas nas Figuras 5.27 (b) e 5.29 (a), o obstáculo permanece por um menor tempo, ou seja por menos iterações, totalmente na zona da umbra. A Figura 5.27 (b) apresenta o robô executando seus movimentos entre as iterações 376 e 447, e das duas estimativas na variável θ , o maior erro percentual calculado foi de 15,5% na média ponderada, e 15,2% na melhor partícula. A variável ρ também teve seu maior erro na trajetória apresentado na Figura 5.28 (b) com valores de 17,9% e 18,9% para a média ponderada e a melhor partícula respectivamente.

Foi calculado um percentagem de erro médio total das seis trajetórias apresentado na

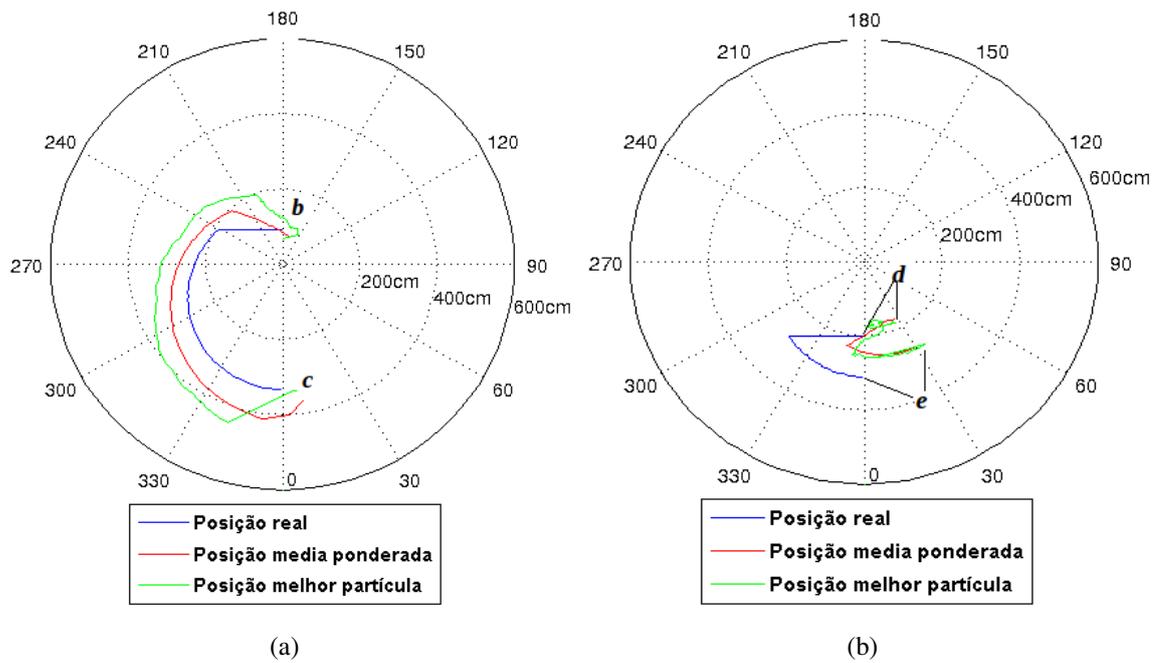


Figura 5.27: Sequências das trajetórias da posição do obstáculo na zona de umbra a) trajetória entre as iterações *b* e *c*, b) trajetória entre as iterações *d* e *e*.

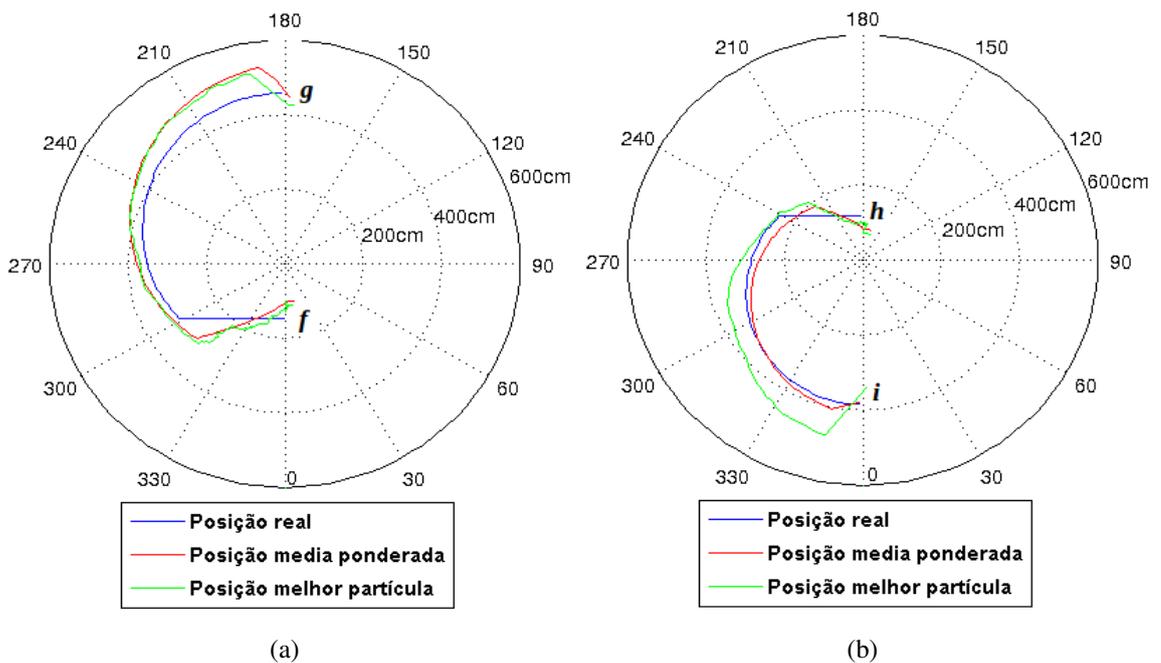


Figura 5.28: Sequências das trajetórias da posição do obstáculo na zona de umbra a) trajetória entre as iterações *f* e *g*, b) trajetória entre as iterações *h* e *i*.

tabela 5.4, levando em consideração que foi feito a partir da iteração (a) em que o sensor não consegue obter mais informações do obstáculo. Pode observar que os valores dos erros são bem próximos aos calculados na tabela 5.2 para 500 partículas da anterior simulação considerando que o obstáculo ficou somente por uma vez na zona de umbra e nesta última simulação o robô fez com que o obstáculo estivesse por mais tempo na umbra e como con-

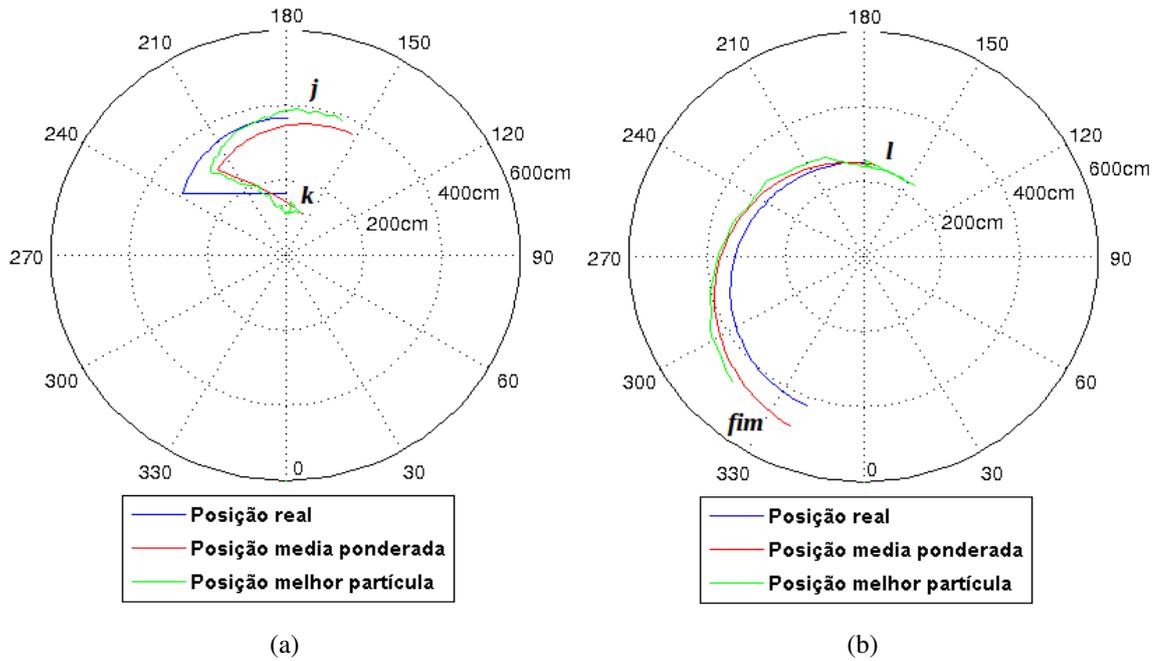


Figura 5.29: Sequências das trajetórias da posição do obstáculo na zona de umbra a) trajetória entre as iterações j e k , b) trajetória entre a iteração l até o final da simulação.

sequência o filtro fica mais tempo sem leituras do sensor.

Quantidade de Partículas		% Erro Médio Estimativa ρ	% Erro Médio Estimativa θ
$N_s = 500$	Média Ponderada	15.5%	11.8%
	Melhor Partícula	15.7%	13.7%

Tabela 5.4: Porcentagem de erro médio da estimativa ρ e θ da segunda simulação durante a trajetória pela umbra.

5.4 DISCUSSÃO

As simulações realizadas tem por objetivo mostrar que é possível seguir um obstáculo mesmo quando ele está fora do alcance do sensor, o que foi feito com a implementação do filtro de partículas integrado a uma área de proteção chamada de *Zona Virtual Deformável (ZVD)*.

As análises realizadas permitem concluir que é possível estimar a posição do obstáculo mesmo ficando sem receber leituras do sensor por algum tempo, mais considerando que a medida que o tempo passa o erro vai aumentando aos pucos.

6 CONCLUSÕES, CONTRIBUIÇÕES E TRABALHOS FUTUROS

Foi apresentado neste trabalho uma metodologia para o tratamento da evasão de um obstáculo fixo em um ambiente de aplicação para robôs móveis. Para esta abordagem foi considerado que o robô possui somente um sensor laser de varredura na orientação frontal do robô, conforme o exposto no Capítulo 4. O problema tratado consistiu em como estimar a posição de um obstáculo não mais visível aos sensores, devido a evasão do robô, mas que ainda pode apresentar um risco de colisão.

A proposta desenvolvida no algoritmo foi baseada em arquiteturas reativas, utilizando o sistema de sensoriamento para coletar as informações referentes ao ambiente no qual o robô móvel navega. Desta feita, o algoritmo proposto reagiu de forma adequada, de forma que o robô pôde continuar navegando de forma segura. O comportamento de reação para a evasão do obstáculo foi baseado na implementação de uma Zona Virtual Deformável (*ZVD*), e logo que um obstáculo é percebido e sua intrusão detectada como uma deformação na *ZVD*, este é identificado como uma ameaça para a integridade física do robô. O processo de estimação empregando o Filtro de Partículas permitiu a correta estimação do obstáculo, bem como tornando viável à aplicações reais, permitindo ainda, incorporar incertezas referentes ao modelo de sensor utilizado em sua implementação.

Foram realizadas simulações em situações controladas, a partir do uso de um ambiente de simulação computacional, modelado de forma a aproximar os resultados obtidos de situações reais de aplicação. Os resultados destas simulações foram analisadas e demonstraram que o algoritmo proposto foi capaz de extrair as informações do obstáculo, de acordo com a detecção realizada pelo sensor, estimando sua posição (objeto presente na zona visível do sensor). Com a manobra de evasão do robô, foram obtidas respostas condizentes permitindo manter estimativas aceitáveis do obstáculo, mesmo quando na zona de não detecção. Os valores obtidos para as incertezas foram condizentes com a proposta de modelo do sensor apresentado, principalmente em situações em que não existe mais informações do obstáculo provenientes do sensor utilizado (obstáculo fora da zona visível do sensor).

A estimativa na posição de cada partícula contribui de forma efetiva no auxílio da deformação da *ZVD*. Este comportamento foi observado, tanto na zona de detecção como na zona de não detecção. Esta característica possibilitou o vetor de intrusão fosse calculado e, mesmo na zona de não detecção, o robô móvel foi capaz de executar o comportamento de evasão sem colidir com obstáculo.

Dentre as diversas contribuições realizadas por este trabalho realizado, a mais relevante é a utilização de técnicas baseadas em *ZVD* utilizando filtro a partículas, técnica esta praticamente inexistente na literatura atual. Esta abordagem permite que o nível de segurança seja mantido no processo de evasão de obstáculos. Podendo então ser aplicada a sistemas

robóticos com quantidades, e qualidade, deficiente de sensores.

Como trabalhos futuros, é desejado expandir a capacidade da técnica para situações envolvendo obstáculos não fixos e ambientes dinâmicos. Adaptar a técnica desenvolvida para execução em tempo real e adaptadas para uma plataforma robótica móvel (Pioneer P3-AT) real. Outras atividades a serem desenvolvidas são definidas no âmbito da exploração da convergência do filtro de partículas utilizado, bem como em sua capacitação em aprimorar a estimativa relativas às incertezas dos obstáculos em situações de elevada complexidade, estas voltadas ao universo de uma aplicação no mundo real.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] G. Dudek and M. Jenkin., *Computational Principles of Mobile Robotics*. Cambridge University Press, 2010.
- [2] R. R. Murphy, *Introduction to AI Robotics*. The MIT Press, 2000.
- [3] A. S. Brandão, M. S. Filho, and T. F. B. Filho., “Navegação de robôs móveis com desvio de obstáculos: Implementação de desvio tangencial modificado,” *VIII Simpósio Brasileiro de Automação Inteligente VIII SBAI*, 2007.
- [4] C. R. Jung, F. S. Osório, C. R. Kelber, and F. J. Heinen, “Computação embarcada: Projeto e implementação de veículos autônomos inteligentes,” *Congresso da Sociedade Brasileira de Computação*, 2005.
- [5] A. A. Medeiros, “A survey of control architectures for autonomous mobile robots,” *JBCS - Journal of the Brazilian Computer Society, special issue on Robotics*, 1998.
- [6] K. Souhila and A. Karim, “Optical flow based robot obstacle avoidance,” *International Journal of Advanced Robotic Systems*, 2007.
- [7] Z. Hui-zhong, D. Shu-xin, and W. Tie-jun, “On-line real-time path planning of mobile robots in dynamic uncertain environment,” *Journal of Zhejiang University SCIENCE*, 2006.
- [8] D. F. Wolf, E. do Valle Simões, F. S. Osório, and O. T. Junior, “Robótica móvel inteligente: Da simulação às aplicações no mundo real,” *Mini-Curso: Jornada de Atualização em Informática (JAI), Congresso da SBC*, 2009.
- [9] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” *International Journal of Robotics Research*, vol. 5, pp. 90–98, 1986.
- [10] J. Borenstein and Y. Koren, “Real time obstacle avoidance for fast mobile robots,” *IEEE Transactions on Robotics and Automation*, 1989.
- [11] J. Borenstein and Y. KorenK, “The vector field histogram - fast obstacle avoidance for mobile robots,” *IEEE Transactions on Robotics and Automation*, 1991.
- [12] M. Khatib and R. Chatila, “An extended potential field approach for mobile robot sensor-based motions,” *In Intl. Conf. On Intelligent Autonomous Systems IAS’4*, 1995.
- [13] S. K. Pradhan, D. R. Parhi, A. K. Panda, and R. K. Behera, “Potential field method to navigate several mobile robots,” *Appl Intell*, vol. 25, pp. 321 – 333, 2006.

- [14] S. Quinlan and O. Khatib, “Elastic bands: Connecting path planning and control,” *In Proceedings of the IEEE Conference on Robotics and Automation*, 1993.
- [15] R. Philippsen and R. Siegwart, “Smooth and efficient obstacle avoidance for a tour guide robot,” *IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA '03.*, vol. 1, pp. 446–451, 2003.
- [16] L. Amouri, C. Novales, G. Poisson, M. Njah, M. Jallouli, and N. Derbel, “Dvz-based obstacle avoidance control of a wheelchair mobile robot,” *IEEE International Conference on Mechatronics*, vol. 1, pp. 911–915, 2011.
- [17] L. Lapierre, R. Zapata, and P. Lepinay, “Simultaneous path following and obstacle avoidance control of a unicycle-type robot,” *IEEE International Conference on Robotics and Automation (rCRA 2007)*, vol. 1, pp. 2617–2622, 2007.
- [18] A. Sanchez, R. Cuautle, M. A. Osorio, and R. Zapata, “Reactive motion planning for mobile robots,” *Mobile Robots Motion Planning, New Challenges*, pp. 469 – 486, 2008.
- [19] A. G. Pinto, P. Fraisse, and R. Zapata, “A decentralized adaptive trajectory planning approach for a group of mobile robots,” *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 2006.
- [20] A. Cacitti and R. Zapata, “Reactive behaviours of mobile manipulators based on the dvz approach,” *IEEE International Conference on Robotics and Automation (RCRA 2001)*, 2001.
- [21] L. Lapierre, R. Zapata, and B. Jouvencel, “Concurrent path following and obstacle avoidance control of a unicycle-type robot,” *AVCS'07: Advances in Vehicle Control and Safety*, 2007.
- [22] A. L. Garcia, *Probability, Statistics, and Random Processes for Electrical Engineering*. Pearson Prentice-Hall, 2008.
- [23] G. C. Canavos, *Probabilidad y Estadística, Aplicaciones y metodos*. McGraw-Hill-Interamericana de Mexico S.A, 1998.
- [24] P. Z. and J. Peebles, *Probability, Random Variables and Random Signal Principles*. McGraw-Hill, Inc., 1987.
- [25] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2005.
- [26] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking,” *IEEE Transactions on signal processing*, 2002.
- [27] M. Isard and A. Blake., “Condensation - conditional density propagation for visual tracking,” *International Journal of Computer Vision*, 1998.

- [28] N. Gordon, D. Salmond, and F. M. Smith., “Novel approach to non-linear and non-gaussian bayesian state estimation.” *IEEE Proceedings - F*, 1993.
- [29] A. Doucet, N. de Freitas, and N. Gordon, “Sequential monte carlo methods in practice,” *Springer-Verlag*, 2001.
- [30] A. Doucet, S. Godsill, and C. Andrieu, “On sequential monte carlo sampling methods for bayesian filtering,” *Statistics and Computing*, 2000.
- [31] D. Fox, J. Hihgtower, L. Lioa, D. Schulz, and G. Borriello., “Bayesian filters for location estimation,” *IEEE Pervasive Computing*, 2003.
- [32] J. D.Hol, “Resampling in particle filters,” Linkopings Universitet, Tech. Rep., 2004.
- [33] C. Musso, N. Oudjane, and F. L. Gland, “Improving regularised particle filters in sequential monte carlo methods in practice,” in *Sequential Monte Carlo Methods in Practice*, 2001.
- [34] A. kong, J. S. Liu, and W. H. Wong., “Sequential imputations and bayesian missing data problems,” *Journal of the American Statistical Association*, 1994.
- [35] A. Doucet, S. Godsill, and C. Andrieu, “On sequential monte carlo methods for bayesian filtering,” *Statistics and Computing*, 2000.
- [36] J. S. Liu, R. Chen, and T. Logvinenko, “A theoretical framework for sequential importance sampling with resampling,” in *Sequential Monte Carlo Methods in Practice*, 2001.
- [37] S. S. Intelligence, *Technical description, LMS200/211/221/291 Laser Measurement Systems*, SICK Sensor Intelligence, Nimbunger Strasse 11 79276 Reute Germany, 2006. [Online]. Available: www.sick.com
- [38] D. Fox, W. Burgard, and S. Thrun., “Markov localization for mobile robots in dynamics environments,” *Journal of Artificial Intelligence Research*, 1999.
- [39] C. Tomasi, “Estimating gaussian mixture densities with em-a tutorial,” 2004.
- [40] J. A. Bilmes, “A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models,” University of Berkeley, TR-97-021, Tech. Rep., 1998.
- [41] B. Resch, *Mixtures of Gaussians, A Tutorial for the Course Computational Intelligence*.
- [42] H. A. Secchi, *Uma Introdução aos Robôs Móveis*, N. de Estudos em Robotica e Automação, Ed. IFES Ú Instituto Federal de Educação, Ciencia e Tecnologia do Espírito Santo, 2012.

- [43] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*. A Bradford Book, 2008.
- [44] A. T. Inc., *Pioneer 3-AT Datasheet*, 10 Columbia Drive, Amherst, NH 03031, 2011. [Online]. Available: <http://www.mobilerobots.com/ResearchRobots/P3AT.aspx>
- [45] I. M. Rekleitis, “A particle filter tutorial for mobile robot localization tr-cim-04-02,” *Center for Intelligent Machines*, 2002.
- [46] A. Sanchez, R. Cuautle, M. A. Osorio, and R. Zapata, “A reactive lazy prm approach for nonholonomic motion planning,” *Advances in Artificial Intelligence - IBERAMIA-SBIA 2006*, 2006.
- [47] B. Gerkey and contributors. (1999-2007) Player 2.0.1 online manual. [Online]. Available: <http://playerstage.sourceforge.net/doc/Player-2.0.0/player/>
- [48] B. P. Gerkey, R. T. Vaughan, and A. Howard, “The player/stage project: Tools for multi-robot and distributed sensor system,” *In Proceeding of the 11th International Conference on Advanced Robotics (ICAR)*, 2003.
- [49] B. P. Gerkey, R. T. Vaughan, K. Story, A. Howard, G. S. Sukhatme, and M. J. Mataric, “Most valuable player: A robot device server for distributed control,” *In Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001)*, 2001.
- [50] R. T. Vaughan and contributors. (1998-2006) Stage 2.0.1 online manual. [Online]. Available: <http://playerstage.sourceforge.net/doc/stage-2.0.0/>
- [51] R. T. Vaughan and B. P. Gerkey, “Reall reusable robot code and the player/ stage project,” *Software Engineering for Experimental Robotics*, 2006.
- [52] F. S. Osório, D. F. Wolf, and E. Simões. (2010) Curso de programação de robôs móveis. São Carlos/ SP e Porto Alegre/RS. [Online]. Available: <http://www.inct-sec.org/actrep/sites/default/files/highlights/Course-MR.pdf>