



Universidade Federal do Piauí
Centro de Ciências da Natureza
Programa de Pós-Graduação em Ciência da Computação

Range-Only SLAM Baseado em Wi-Fi Utilizando Filtro de Kalman Estendido para Cellbot

Ranulfo Plutarco Bezerra Neto

Número de Ordem PPGCC: M001

Teresina-PI, Janeiro de 2018

Ranulfo Plutarco Bezerra Neto

Range-Only SLAM Baseado em Wi-Fi Utilizando Filtro de Kalman Estendido para Cellbot

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da UFPI (área de concentração: Sistemas de Computação), como parte dos requisitos necessários para a obtenção do Título de Mestre em Ciência da Computação.

Universidade Federal do Piauí – UFPI

Centro de Ciências da Natureza

Programa de Pós-Graduação em Ciência da Computação

Orientador: Prof. Dr. André Macedo Santana

Coorientador: Prof. Dr. Ricardo de Andrade Lira Rabêlo

Teresina-PI

Janeiro de 2018

Ranulfo Plutarco Bezerra Neto

Range-Only SLAM Baseado em Wi-Fi Utilizando Filtro de Kalman Estendido para Cellbot/ Ranulfo Plutarco Bezerra Neto. – Teresina-PI, Janeiro de 2018-
104 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. André Macedo Santana

Dissertação (Mestrado) – Universidade Federal do Piauí – UFPI

Centro de Ciências da Natureza

Programa de Pós-Graduação em Ciência da Computação, Janeiro de 2018.

1. Palavra-chave1. 2. Palavra-chave2. I. Orientador. II. Universidade xxx. III. Faculdade de xxx. IV. Título

CDU 02:141:005.7

*“Deixe o futuro dizer a verdade,
e avaliar cada um de acordo com
seus trabalhos e suas conquistas.”
(Nikola Tesla)*

Resumo

Com o advento de novas tecnologias e o barateamento de componentes eletrônicos, a facilidade de aquisição de um smartphone tem crescido a cada dia. Atualmente, os smartphones são detentores de processamento computacional com capacidade semelhante ou superior à alguns computadores no mercado. Diversas pesquisas demonstram a viabilidade do uso de smartphones na robótica e, como tais, este trabalho almeja aplicar uma técnica probabilística para solucionar o problema de SLAM em uma arquitetura Cellbot, onde o acionamento é realizado por um sistema embarcado e o processamento é feito em um smartphone. A proposta deste trabalho consiste em uma abordagem baseada no uso de dois receptores de sinal de Wi-Fi, usando a informação do RSS (*Received Signal Strength*) recebida pelos nós transmissores (Pontos de Acesso) para o problema de SLAM (*Simultaneous Localization and Mapping*). A solução apresentada é baseada no Filtro de Kalman Estendido utilizando apenas os RSS provindos de diferentes transmissores no cenário. O sistema em questão considera que o robô navega em um ambiente desconhecido onde recebe diferentes sinais de Wi-Fi com correspondência conhecida. A partir do uso da odometria e dos sinais Wi-Fi recebidos, o robô localiza a posição de cada um dos transmissores Wi-Fi bem como a si mesmo. Os resultados realizados por meio do simulador V-REP são apresentados para validar a proposta.

Palavras-chaves: Filtro de Kalman. Range-only SLAM. Robô Móvel.

Abstract

With the advent of new technologies and cheaper electronic components, it has become easier to acquire a smartphone. Nowadays, smartphones contain processing capability as good as or even superior to some computers in the market. Several researches show the feasibility of smartphones in robotics, and then as well, this work aims to apply a probabilistic technique to solve the SLAM problem in a Cellbot architecture, where the activation is done by an embedded system and is processed by a smartphone. The proposal in this work consists in an approach based on the utilization of Wi-Fi receptors, using RSS information received by the transmission nodes (Access Point) for the SLAM problem. The solution presented is based on extended Kalman filter using RSS coming from different transmitters in the environment. The system considers that a robot navigates in an unknown environment where it receives different Wi-Fi signals from a known source. Through the odometry and Wi-Fi signals, the robot localizes itself and the position of each Wi-Fi transmitter. The initial results executed by the V-REP show the viability of the application. The results provided by V-REP simulator are presented to validate the approach.

Keywords: Kalman Filter. Range-only SLAM. Mobile Robot.

Lista de ilustrações

Figura 1 – Robôs: (a) YuMi; (b) Da Vinci; (c) Comau; (d) Drone; (e) Row-Bot; (f) Roomba.	1
Figura 2 – Cellbot.	2
Figura 3 – Etapas de um processo de navegação robótica.	3
Figura 4 – Etapa de um processo de localização baseado em marcos.	4
Figura 5 – Etapa de um processo de mapeamento baseado em marcos.	5
Figura 6 – Problema do SLAM	6
Figura 7 – Esquema simplificado de SLAM.	7
Figura 8 – Visão Geral.	8
Figura 9 – Posicionamento resultante da integração de deslocamentos ao longo do tempo.	11
Figura 10 – Filtro de Kalman	13
Figura 11 – Visão geral do sistema proposto.	20
Figura 12 – Variáveis do modelo geométrico.	21
Figura 13 – Triangulação dos marcos.	25
Figura 14 – Triangulação dos marcos.	26
Figura 15 – Informações do Sinal Wi-Fi utilizadas no sistema.	27
Figura 16 – Função de Propagação do Sinal Wi-Fi para: (a) Receptor 1; (b) Receptor 2.	28
Figura 17 – Função de Propagação do Sinal Wi-Fi para: (a) Receptor 1; (b) Receptor 2.	29
Figura 18 – Simulação do Ruído da Odometria.	31
Figura 19 – Simulação do Ruído do Sinal Wi-Fi.	31
Figura 20 – Mapa do Cenário A, B, C e D	32
Figura 21 – Cenário com 3 APs: (a) Cobertura do Sinal Wi-Fi (b) Disponibilidade do Sinal Wi-Fi com 1 Receptor (c) Disponibilidade do Sinal Wi-Fi com 2 Receptores.	33
Figura 22 – Cenário com 4 APs: (a) Cobertura do Sinal Wi-Fi (b) Disponibilidade do Sinal Wi-Fi com 1 Receptor (c) Disponibilidade do Sinal Wi-Fi com 2 Receptores.	33
Figura 23 – Cenário com 6 APs: (a) Cobertura do Sinal Wi-Fi (b) Disponibilidade do Sinal Wi-Fi com 1 Receptor (c) Disponibilidade do Sinal Wi-Fi com 2 Receptores.	34
Figura 24 – Resultado da Posição Final dos APs: (a) 3APs; (b) 4APs; (c) 6APs - Cena A.	35

Figura 25 – Trajetória do Robô nas Coordenadas x e y: (a) 3 APs; (b) 4 APs; (c) 6 APs - Cena B.	36
Figura 26 – Resultado da Posição Final dos APs: (a) 3APs; (b) 4APs; (c) 6APs - Cena B.	37
Figura 27 – Trajetória do Robô nas Coordenadas x e y: (a) 3 APs; (b) 4 APs; (c) 6 APs - Cena C.	38
Figura 28 – Resultado do Erro da Estimativa Inicial: (a) 3 APs; (b) 4 APs; (c) 6 APs - Cena C.	39
Figura 29 – Resultado da Posição Final dos APs: (a) 3 APs; (b) 4 APs; (c) 6 APs - Cena C.	40
Figura 30 – Trajetória do Robô nas Coordenadas x e y: (a) 3 APs; (b) 4 APs; (c) 6 APs - Cena D.	41
Figura 31 – Resultado do Erro da Estimativa Inicial: (a) 3 APs; (b) 4 APs; (c) 6 APs - Cena D.	42
Figura 32 – Trajetória do AP 1 na cena D com 3 APs: (a) SLAM 1R; (b) SLAM 2R.	42
Figura 33 – Resultado da Posição Final dos APs: (a) 3APs; (b) 4APs; (c) 6APs - Cena D.	44
Figura 34 – Mapa do Cenário com Loop.	45
Figura 35 – Cenário com 8 APs: (a) Cobertura do Sinal Wi-Fi (b) Grafo de Visibilidade do Sinal Wi-Fi com 1 Receptor (c) Grafo de Visibilidade do Sinal Wi-Fi com 2 Receptores.	45
Figura 36 – Percurso do Robô com 8 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário E.	47
Figura 37 – Gráfico da Variância da Posição do Robô após 1 volta nas coordenadas x e y: (a) SLAM 1R; (b) SLAM 2R.	48
Figura 38 – Resultado da Posição Final dos 8 APs após 1 volta.	48
Figura 39 – Percurso do Robô com 4 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário E.	49
Figura 40 – Gráfico da Variância da Posição do Robô após 3 voltas nas coordenadas x e y: (a) SLAM 1R; (b) SLAM 2R.	50
Figura 41 – Resultado da Posição Final dos 8 APs após 3 volta.	51
Figura 42 – Percurso do Robô após 5 voltas: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário E.	52
Figura 43 – Gráfico da Variância da Posição do Robô após 5 voltas nas coordenadas x e y: (a) SLAM 1R; (b) SLAM 2R.	52
Figura 44 – Resultado da Posição Final dos 8 APs após 5 volta.	53
Figura 45 – Trajetória do Robô com 3 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cena A.	64

Figura 46 – Trajetória do Robô com 4 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cena A.	65
Figura 47 – Trajetória do Robô com 6 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cena A.	66
Figura 48 – Trajetória dos APs no SLAM com 1 Receptor: (a) AP 1; (b) AP 2; (c): AP 3 - Cena A.	67
Figura 49 – Trajetória dos APs no SLAM com 2 Receptores: (a) AP 1; (b) AP 2; (c): AP 3 - Cena A.	68
Figura 50 – Trajetória dos APs no SLAM com 1 Receptor: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4 - Cena A.	69
Figura 51 – Trajetória dos APs no SLAM com 2 Receptores: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4 - Cena A.	70
Figura 52 – Trajetória dos APs no SLAM com 1 Receptor: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4; (e): AP 5; (f) AP 6 - Cena A.	71
Figura 53 – Trajetória dos APs no SLAM com 2 Receptores: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4; (e): AP 5; (f) AP 6 - Cena A.	72
Figura 54 – Resultado do Erro da Estimativa Inicial: (a) 3 APs; (b) 4 APs; (c) 6 APs - Cena A.	73
Figura 55 – Trajetória do Robô com 3 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário B.	75
Figura 56 – Trajetória do Robô com 4 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário B.	76
Figura 57 – Trajetória do Robô com 6 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário B.	77
Figura 58 – Trajetória dos APs no SLAM com 1 Receptor: (a) AP 1; (b) AP 2; (c): AP 3 - Cenário B.	78
Figura 59 – Trajetória dos APs no SLAM com 2 Receptores: (a) AP 1; (b) AP 2; (c): AP 3 - Cenário B.	79
Figura 60 – Trajetória dos APs no SLAM com 1 Receptor: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4 - Cenário B.	80
Figura 61 – Trajetória dos APs no SLAM com 2 Receptores: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4 - Cenário B.	81
Figura 62 – Trajetória dos APs no SLAM com 1 Receptor: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4; (e): AP 5; (f) AP 6 - Cenário B.	82
Figura 63 – Trajetória dos APs no SLAM com 2 Receptores: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4; (e): AP 5; (f) AP 6 - Cenário B.	83
Figura 64 – Resultado do Erro da Estimativa Inicial: (a) 3 APs; (b) 4 APs; (c) 6 APs - Cena B.	84

Figura 65 – Trajetória do Robô com 3 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário C. . . .	85
Figura 66 – Trajetória do Robô com 4 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário C. . . .	86
Figura 67 – Trajetória do Robô com 6 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário C. . . .	87
Figura 68 – Percurso dos APs em C no SLAM com 1 Receptor: (a) AP 1; (b) AP 2; (c): AP 3 - Cenário C.	88
Figura 69 – Percurso dos APs em C no SLAM com 2 Receptores: (a) AP 1; (b) AP 2; (c): AP 3 - Cenário C.	89
Figura 70 – Percurso dos APs em A no SLAM com 1 Receptor: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4 - Cenário C.	90
Figura 71 – Percurso dos APs em A no SLAM com 2 Receptores: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4 - Cenário C.	91
Figura 72 – Trajetória dos APs no SLAM com 1 Receptor: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4; (e): AP 5; (f) AP 6 - Cenário C.	92
Figura 73 – Trajetória dos APs no SLAM com 2 Receptores: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4; (e): AP 5; (f) AP 6 - Cenário C.	93
Figura 74 – Trajetória do Robô com 3 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário D. . . .	95
Figura 75 – Trajetória do Robô com 4 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário D. . . .	96
Figura 76 – Percurso do Robô em D com 6 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário D.	97
Figura 77 – Trajetória do Robô com 8 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário E. . . .	98
Figura 78 – Trajetória dos APs no SLAM com 1 Receptor: (a) AP 1; (b) AP 2; (c): AP 3 - Cenário D.	99
Figura 79 – Trajetória dos APs no SLAM com 2 Receptores: (a) AP 1; (b) AP 2; (c): AP 3 - Cenário D.	100
Figura 80 – Trajetória dos APs no SLAM com 1 Receptor: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4 - Cenário D.	101
Figura 81 – Trajetória dos APs no SLAM com 2 Receptores: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4 - Cenário D.	102
Figura 82 – Trajetória dos APs no SLAM com 1 Receptor: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4; (e): AP 5; (f) AP 6 - Cenário D.	103
Figura 83 – Trajetória dos APs no SLAM com 2 Receptores: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4; (e): AP 5; (f) AP 6 - Cenário D.	104

Lista de tabelas

Tabela 1 – Porcentagem de Visibilidade por AP - 3 APs	32
Tabela 2 – Porcentagem de Visibilidade por AP - 4 APs	34
Tabela 3 – Porcentagem de Visibilidade por AP - 6 APs	35
Tabela 4 – Erro da Posição Final do Robô com Variação de APs - Cena D	41
Tabela 5 – Erro de Posição Final do Marco com Variação de APs - Cena D	43
Tabela 6 – Porcentagem de Visibilidade por AP - 8 APs	44
Tabela 7 – Erro Médio Quadrático ao Final da Trajetória do Robô após 1 Volta.	46
Tabela 8 – Distância Euclidiana Final dos APs após 1 Volta.	46
Tabela 9 – Erro Médio Quadrático ao Final da Trajetória do Robô após 3 voltas.	49
Tabela 10 – Distância Euclidiana Final dos APs após 3 voltas.	50
Tabela 11 – Erro Médio Quadrático ao Final da Trajetória do Robô após 5 voltas.	51
Tabela 12 – Distância Euclidiana Final dos APs após 5 voltas.	53

Lista de abreviaturas e siglas

SLAM	Localização e Mapeamento Simultâneos.
RSS	Força de Sinal Recebido.
dBm	Decibel.
GPS	Sistema de Posicionamento Global.
FKD	Filtro de Kalman Discreto.
FKE	Filtro de Kalman Estendido.
RO-SLAM	<i>Range-only</i> Localização e Mapeamento Simultâneos.
AP	Ponto de Acesso.
TOA	Tempo de Chagada.
TDOA	Diferença de Tempo de Chagada.
RTOF	Tempo de Vôo do Sinal.
RSP	Fase de Sinal Recebido.
RSS	Força de Sinal Recebido.

Lista de símbolos

ΔL	Deslocamento linear do robô;
$\Delta \theta$	Deslocamento angular do robô;
$\Delta \hat{L}$	Deslocamento linear real do robô;
$\Delta \hat{\theta}$	Deslocamento angular real do robô;
b	Comprimento do eixo do robô;
$\Delta \theta$	Deslocamento angular da roda do robô;
$\Delta \hat{\theta}$	Deslocamento angular real da roda do robô;
r	Raio da roda do robô;
d	Direita do robô;
e	Esquerda do robô;
R_x	Posição do robô na coordenada x.
R_y	Posição do robô na coordenada y.
$R\theta$	Ângulo do robô.
$\theta_{bússola}$	Ângulo do extraído da bússola.
M_x	Posição do marco na coordenada x.
M_y	Posição do marco na coordenada y.
S_x	Posição do smartphone na coordenada x.
S_y	Posição do smartphone na coordenada y.
E_x	Posição do sistema embarcado na coordenada x.
E_y	Posição do sistema embarcado na coordenada y.
s	Vetor de estados estimado.
z	Vetor de medições estimado.
A	Matriz de transição dos estados.

B	Matriz de coeficientes de entrada.
C	Matriz de observação.
ε	Vetor de ruídos do processo.
δ	Vetor de ruídos da medição estimado.
Σ	Matriz de covariância dos estados.
R	Matriz de covariância dos ruídos do processo.
Q	Matriz de covariância dos ruídos da medição.
K	Matriz do ganho do Filtro de Kalman.
I	Matriz identidade.
G	Matriz jacobiana do modelo de processo pelo vetor de estado.
H	Matriz jacobiana do modelo de medição pelo vetor de estado.
u	Sinais de entrada enviados para as rodas;
t	Instante de tempo t .

Sumário

1	INTRODUÇÃO	1
	Introdução	1
1.1	Cellbots	2
1.2	Navegação Autônoma	3
1.2.1	Localização	3
1.2.2	Mapeamento	5
1.2.3	SLAM	6
1.3	Visão Geral	7
1.4	Motivação	8
	Motivação	8
1.5	Objetivos	9
	Objetivos	9
2	MATERIAIS E MÉTODOS	11
2.1	Odometria Mecânica	11
2.2	Filtro de Kalman	12
2.2.1	Filtro de Kalman Estendido	12
2.3	Localização <i>indoor</i> com Wi-Fi	14
2.4	Trabalhos Relacionados	15
2.5	Considerações Finais	17
3	SISTEMA PROPOSTO	19
3.1	Modelagem do Sistema	21
3.1.1	EKF - SLAM	21
3.1.2	Sinal Wi-Fi	27
3.2	Considerações Finais	29
4	RESULTADOS E DISCUSSÃO	31
4.1	Cena A: Validação da Solução	35
4.2	Cena B: Influência do Ruído da Fase de Predição	36
4.3	Cena C: Influência do Ruído da Fase de Atualização	38
4.4	Cena D: Cenário Completo	40
4.5	Cena E: Cenário com Loop	44
4.5.1	Cenário E com 1 Volta	46
4.5.2	Cenário E com 3 Voltas	49

4.5.3	Cenário E com 5 Voltas	51
4.6	Considerações Finais	54
5	CONCLUSÕES E TRABALHOS FUTUROS	55
	Conclusão e Trabalhos Futuros	55
	REFERÊNCIAS	57
	APÊNDICES	61
	APÊNDICE A – RESULTADOS ADICIONAIS DA CENA A	63
	APÊNDICE B – RESULTADOS ADICIONAIS DA CENA B	75
	APÊNDICE C – RESULTADOS ADICIONAIS DA CENA C	85
	APÊNDICE D – RESULTADOS ADICIONAIS DA CENA D	95

1 Introdução

A área da robótica está em ascensão desde as últimas décadas em função dos novos recursos de hardware e software desenvolvidos. Em relação ao hardware, os computadores e dispositivos embarcados estão passando pelo processo de miniaturização, reduzindo seus custos e aumentando sua capacidade de processamento. Sensores como laser e câmera de vídeo se tornaram mais baratos e poderosos e, conseqüentemente, refinaram a capacidade perceptiva dos robôs. A melhora na quantidade e qualidade de informações capturadas na atuação do robô reflete diretamente na área de software, permitindo o desenvolvimento de novos algoritmos em áreas de controle, tomada de decisão, processamento de imagem, reconhecimento de voz, entre outras (ROMERO et al., 2014).

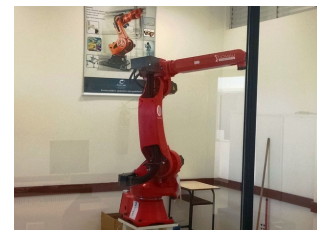
Craig (2005) classifica os robôs em duas classes: robôs manipuladores e robôs móveis. Os robôs manipuladores têm forte atuação nas indústrias, sobretudo na automobilística (Figura 1a) e na área médica, onde auxiliam tratamentos que exigem alta precisão (Figura 1b e 1c). A classe de robôs móveis é constituída por robôs que tem a capacidade de locomoção no espaço tridimensional (robôs aeroespaciais, Figura 1d e robôs aquáticos, Figura 1e) ou no espaço planar (robôs terrestres, Figura 1f).



(a)



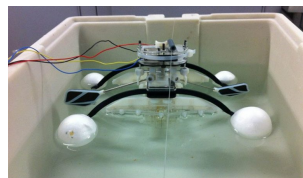
(b)



(c)



(d)



(e)



(f)

Figura 1 – Robôs: (a) YuMi; (b) Da Vinci; (c) Comau; (d) Drone; (e) Row-Bot; (f) Roomba.

Atualmente, existem robôs sendo aplicados em linhas de produção industriais (robô da ABB, YuMi), em cirurgias de alta precisão (Da Vinci), no tratamento de câncer (Comau) em sistemas de monitoramento (Drone), em limpezas aquáticas (Row-Bot), em limpezas de ambientes terrestres (aspirador de pó, Roomba) entre muitos outros. Um outro exemplo de robô terrestre é o Cellbot, o robô abordado neste trabalho.

1.1 Cellbots

No Brasil, em torno de dois terços dos robôs operando em trabalhos voltados para a educação são construídos pela instituição utilizando algum tipo de sistema embarcado. Devido ao alto custo do hardware de um robô móvel, propostas como o Cellbot e arquiteturas robóticas de baixo custo se tornam cada vez mais comuns (Bezerra Neto et al., 2015). Cellbot é uma arquitetura para sistemas robóticos móveis que faz a utilização de um telefone móvel (*smartphones*) que possibilite o desenvolvimento de aplicações como, por exemplo, uma unidade de comando para os sensores e atuadores de um robô, como descrito por Aroca et al. (2013).

Este modelo de arquitetura permite que o robô se beneficie de determinadas características dos *smartphones*, que em sua maior parte vem equipado com um sensor do sistema de posicionamento global (*global positioning system, GPS*), câmera, bússola, acelerômetro, módulo de comunicação Wi-Fi, módulo de comunicação *Bluetooth*, saídas de som, microfone e muitas outras características que conferem no uso da robótica móvel. A Figura 2 ilustra o Cellbot e seus sensores.

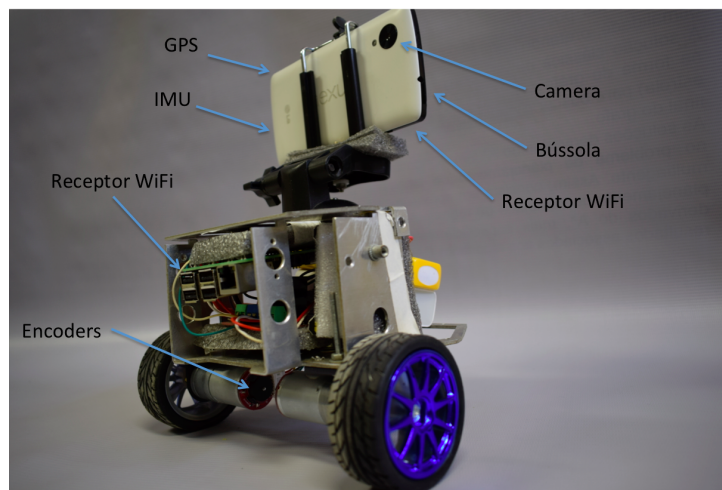


Figura 2 – Cellbot.

Santos, Tarrataca e Cardoso (2010) provaram ser viável o desenvolvimento e execução de algoritmos de navegação para robôs móveis utilizando um telefone móvel. Nessa abordagem, a implementação dos algoritmos realizada por (SANTOS; TARRATACA; CARDOSO, 2010) demonstrou alta consistência e robustez em sua análise. Considerando as vantagens citadas, optou-se pela utilização da arquitetura Cellbot neste trabalho visando solucionar duas das etapas da navegação móvel.

1.2 Navegação Autônoma

Um dos grandes desafios da robótica móvel é realizar a navegação de forma autônoma, ou seja, o conjunto de processos necessários para permitir que um robô saia de um estado inicial e alcance um estado final percorrendo um determinado caminho planejado, desviando de obstáculos de modo independente quando existirem. [Alsina et al. \(2002\)](#) divide o processo de navegação em cinco níveis hierárquicos: modelagem do ambiente, localização, planejamento de caminho, geração de trajetória e execução da trajetória. A Figura 3 ilustra as etapas do processo de navegação.



Figura 3 – Etapas de um processo de navegação robótica.

No mapeamento de ambiente, o robô utiliza sensores para coletar informações e criar um modelo com as principais características estruturais do ambiente. Durante a etapa de localização, é determinada a posição e orientação do robô no espaço com base no modelo obtido no nível de mapeamento do ambiente. No planejamento de caminho, o robô calcula a curva geométrica que o levará de sua localização inicial à final desviando dos obstáculos quando existirem. Na geração de trajetória, o caminho calculado no nível anterior é adaptado às restrições temporais impostas e a partir disso são calculadas as velocidades que devem ser aplicadas ao robô para que o mesmo possa percorrer o caminho no intervalo de tempo especificado. Por fim, na execução de trajetória os atuadores do robô são controlados de forma que a trajetória percorrida por ele seja a mais próxima possível da calculada no nível anterior.

Quando o robô não possui um mapa do ambiente onde está se locomovendo e a posição de onde se encontra, as duas primeiras etapas do processo de navegação devem ser realizadas de forma simultânea gerando o problema de SLAM (*Simultaneous Localization and Mapping*). Para facilitar o entendimento, antes de introduzir o SLAM serão comentadas as etapas de Localização e Mapeamento de forma separada.

1.2.1 Localização

Uma importante etapa no processo de navegação de robôs móveis autônomos é a localização. Nesta etapa o robô é responsável por determinar e manter a sua posição e orientação em relação ao ambiente em que se encontra, por meio da leitura de sensores, mesmo que estes apresentem erro e/ou ruído. Em outras palavras, localização consiste em determinar a pose do robô (posição e orientação) no espaço de trabalho a cada instante de tempo.

Borenstein e Feng (1996) afirmam que os métodos utilizados para a localização podem ser agrupados em dois tipos: localização com posicionamento relativo, onde a estimativa da pose do robô é feita com relação ao seu ponto de partida (por exemplo, o uso da odometria) e localização com posicionamento absoluto, onde é realizada a estimativa da pose global do robô, ou seja, a pose com relação ao ambiente em que se encontra (por exemplo, o sensor de GPS).

A localização de posicionamento relativo envolve apenas parâmetros internos do robô, portanto não depende da aquisição de informações do ambiente. Para este tipo de localização é comum o uso de odometria por meio de informações advindas de sensores acoplados às rodas do robô, de maneira a calcular o seu deslocamento. Entretanto a odometria é passível de diversos fatores que ocasionam erros neste sistema, como impressões na modelagem do sistema e irregularidades no ambiente de trabalho do robô. Outro fator que dificulta a utilização da odometria é o acúmulo de erros provindos do cálculo de cada posição do robô.

Além do uso de sensores como o GPS, é possível também calcular a pose absoluta do robô por meio de marcos, que são elementos distintos que o robô é capaz de reconhecer através de seus sensores. Os marcos geralmente são encontrados em posições fixas e conhecidas em relação ao referencial do ambiente. Sensores são utilizados para detectar esses marcos e, por meio do modelo do sensor, determinar suas posições em relação ao referencial do robô. Caso a posição dos marcos seja conhecida com relação ao referencial do robô, é possível determinar a posição do robô com relação ao referencial do ambiente (BORENSTEIN; FENG, 1996).

Como pode ser visto na Figura 4, o sistema de localização do robô móvel detecta o transmissor WiFi por meio do sinal enviado. Sabendo a posição do transmissor WiFi no sistema de coordenadas global, o sistema calcula a pose do robô neste sistema de coordenadas.

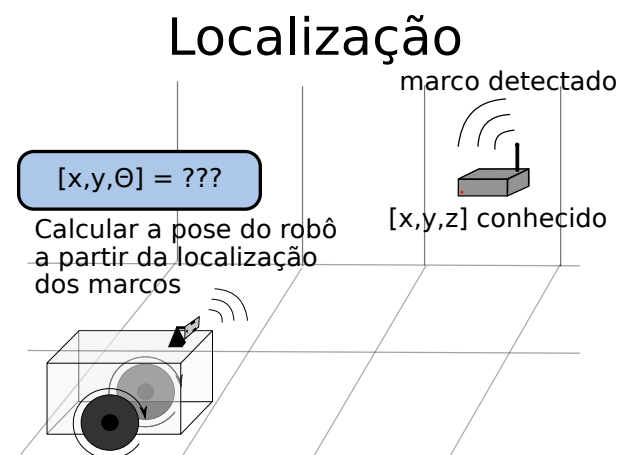


Figura 4 – Etapa de um processo de localização baseado em marcos.

É importante ressaltar que as imprecisões dos sensores do robô podem comprometer a etapa de localização. Assim, o grande desafio na solução do problema de localização está no fato de que tanto as informações recebidas pelos sensores como as informações advindas do ambiente são limitadas e imprecisas na maioria dos casos. Nesse contexto, técnicas probabilísticas têm sido amplamente utilizadas na solução desse problema (THRUN; BURGARD; FOX, 2005).

1.2.2 Mapeamento

No Mapeamento, o robô utiliza as informações obtidas pelos sensores para identificar características estruturais ao seu redor. Por meio de um processamento das informações coletadas pelos sensores do robô, um mapa do ambiente é construído. Além disso, para que um sistema robótico possa construir um mapa, é necessário que a posição e orientação do robô sejam conhecidas com relação a algum referencial fixo no mundo. Com o processo de obtenção dos dados sensoriais, o processamento das informações adquiridas pelos sensores e orientação e posição do robô, têm como objetivo a construção de um mapa denominado Mapeamento Robótico.

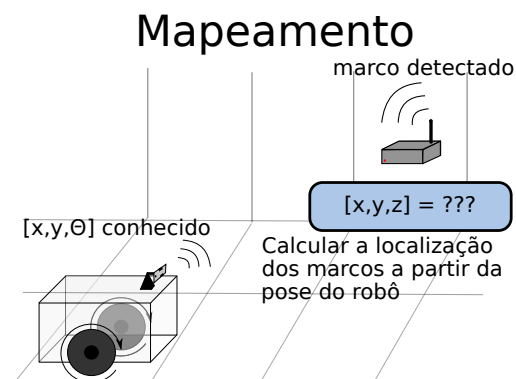


Figura 5 – Etapa de um processo de mapeamento baseado em marcos.

Na Figura 5, o sistema de mapeamento do robô móvel detecta o marco utilizando o receptor de Wi-Fi. Sabendo a posição do robô, o sistema calcula a localização do transmissor detectado e armazena esta informação de forma sistemática em uma estrutura computacional que irá representar o mapa do ambiente. Neste caso, o modelo computacional será constituído simplesmente das coordenadas dos pontos detectados.

A hipótese de que o robô tem conhecimento da própria pose global nem sempre é verdadeira, visto que, na maioria das vezes, para se obter a sua localização em relação a um referencial global é necessário um mapa a priori. Portanto, temos um conflito entre localização e mapeamento, onde para se inferir a pose do robô é preciso ter conhecimento prévio de um mapa, e para inferir a posição de marcos no ambiente é necessário um sistema de localização. Com isso, há uma interdependência entre localização e mapeamento robótico e para tratar este conflito de forma simultânea surgiram técnicas de SLAM.

1.2.3 SLAM

Localização e Mapeamento Simultâneos, referenciado pela sigla em inglês SLAM (*Simultaneous Localization and Mapping*), é um dos problemas mais determinantes da robótica. O problema de SLAM surge quando o robô não possui conhecimento prévio de mapa do ambiente, nem têm conhecimento de sua própria pose, e dado as informações adquiridas do ambiente por meio de sensores, calcula tanto sua pose quanto o mapa do ambiente de maneira simultânea. Na Figura 6 temos uma ilustração do problema.

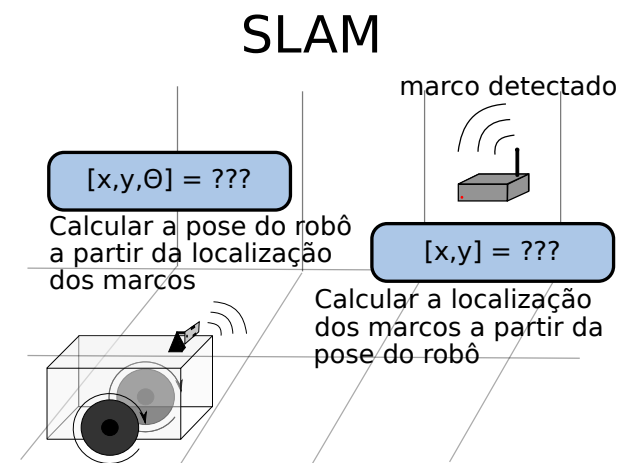


Figura 6 – Problema do SLAM

No SLAM, quando um marco é detectado em um instante posterior, o robô adquire novas informações relativas não somente sobre a posição do marco no ambiente, mas também informações relativas à sua própria pose. Desta maneira, ele consegue refinar simultaneamente ambas, tornando a localização e o mapeamento globalmente mais precisos.

Se um marco é reobservado, há então o problema de correspondência ou *Matching*. O robô deve identificar se o marco reobservado é de fato um já detectado anteriormente, ou se é um novo marco. De maneira resumida, ocorre uma correspondência quando um marco detectado já foi visto anteriormente, caso não o tenha observado a priori, não há correspondência deste marco.

Na prática, os problemas de correspondência de dados são:

- o marco pode não ser observado a cada passo;
- pode-se identificar um marco uma vez e ele não ser mais observado;
- um marco pode ser associado indevidamente a outro marco visto anteriormente;
- um marco pode ser reobservado e não reconhecido como um mesmo já visto anteriormente.

A Figura 7 mostra o esquema simplificado do SLAM. A Figura 7a mostra o robô em uma configuração inicial que possibilita a observação de um marco (transmissor WiFi). No início da navegação o robô assume sua pose conhecida ($[x,y,\theta] = [0,0,0^\circ]$) e como ele identifica pela primeira vez um marco, este é inserido diretamente no mapa de representação do ambiente. Em seguida, o robô se desloca e um marco é observado (Figura 7b). É realizada uma etapa de correspondência para verificar se o marco detectado é antigo ou novo. Caso seja detectado um novo marco, ele é inserido no mapa de representação do ambiente e então é calculado sua nova predição da pose com base na odometria e localização deste marco. Caso o marco detectado já exista no mapa, é realizada uma atualização da pose do robô e da localização do marco.

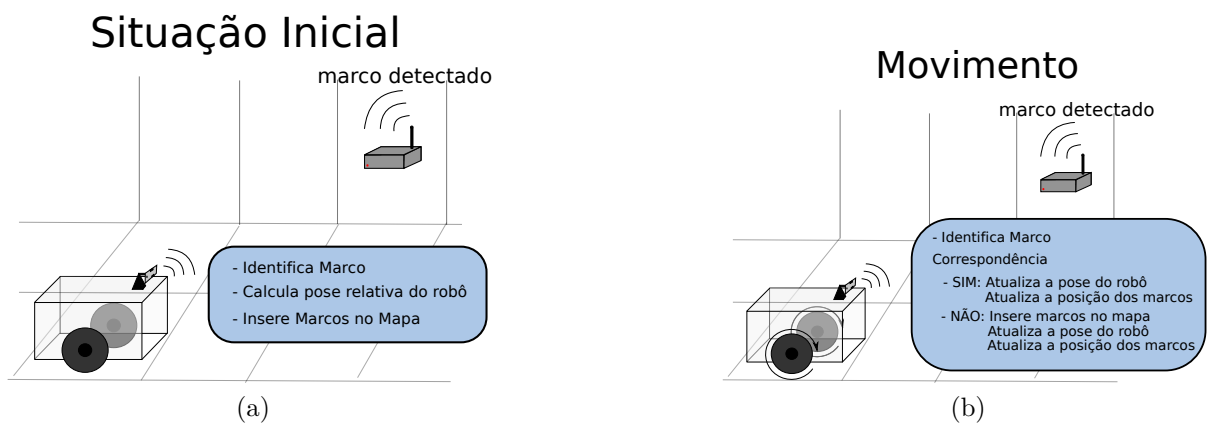


Figura 7 – Esquema simplificado de SLAM.

Os processos de predição e atualização são normalmente realizados por estimadores estatísticos como o Filtro de Kalman. Estas fases serão detalhadas nos Capítulos seguintes.

1.3 Visão Geral

Dentre os sensores presentes em um smartphone, destacamos o sensor de transmissão/recepção wireless, por onde o aparelho recebe as identidades dos transmissores de Wi-Fi presentes no seu alcance, bem como a potência de sinal registrada por cada um destes transmissores. Como este sensor não fornece um ângulo entre o robô e transmissor, ele é então denominado como um sensor de alcance (range-only sensor). Quando são utilizados sensores range-only para atualizar o SLAM, o processo é denominado range-only SLAM (RO-SLAM).

O RO-SLAM desenvolvido utiliza Filtro de Kalman Estendido (FKE) para realizar a fusão das informações sensoriais do robô. Para isso, é necessário a inicialização do filtro. Esta etapa ocorre cada vez que um novo sinal de Wi-Fi é encontrado no ambiente, onde o robô deve determinar o primeiro palpite da posição do transmissor deste novo sinal. Após a inicialização, cada potência de sinal reconhecida no ambiente é convertida em distância entre o robô e o transmissor, para então ser utilizada no FKE.

Este trabalho propõe utilizar o RO-SLAM para mapear a posição dos transmissores Wi-Fi presentes no ambiente, e ao mesmo tempo localizar o Cellbot. A Figura 8 ilustra o sistema onde, ao percorrer o ambiente, o robô pode encontrar ou não sinais Wi-Fi. Quando o robô não encontra sinais de Wi-Fi, situação observada no instante A, ele utilizará apenas a odometria, técnica de localização descrita no Capítulo 2. Quando o robô encontra sinais de Wi-Fi, como é ilustrado nos instantes B e C, o robô irá fundir a técnica de odometria com o RO-SLAM, mesmo que encontre apenas um sinal (como é o caso do instante C), para uma melhor localização do sistema.

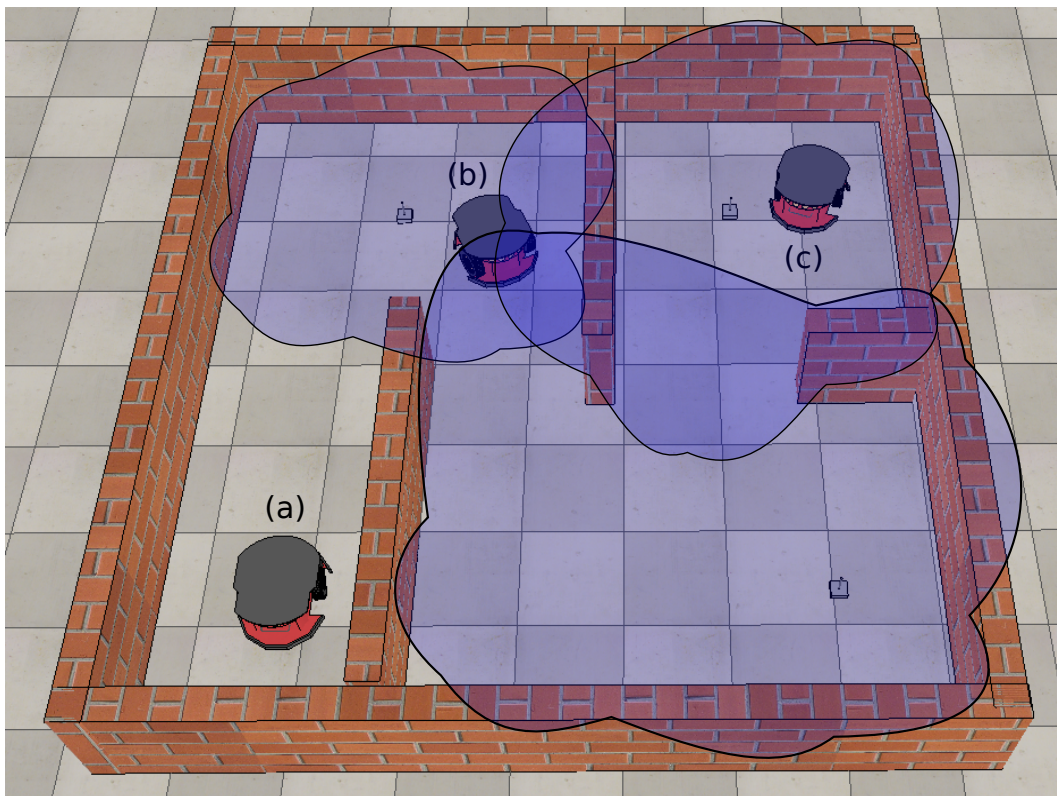


Figura 8 – Visão Geral.

Para o RO-SLAM proposto, foi aplicada a modelagem de SLAM para a atualização simultânea de dois sinais Wi-Fi para cada captura de sinal, gerando um impacto na rapidez de inicialização e precisão da atualização tanto dos marcos quanto da pose do robô. Por ser uma abordagem inovadora, não foi encontrado na literatura a utilização de uma atualização dupla de sinais Wi-Fi.

1.4 Motivação

Em um robô móvel, quanto maior a precisão na percepção do mundo, melhor seria sua autonomia. A capacidade de perceber o ambiente através de sensores torna o robô capaz de realizar tarefas úteis e importantes como alcançar um alvo, manipular objetos, entregar mercadorias, limpeza de cômodos etc.

Em relação ao SLAM, acredita-se que uma solução para este problema permite o surgimento de robôs verdadeiramente autônomos, capazes de navegar de maneira segura por ambientes desconhecidos e cumprir objetivos sem a necessidade de auxílio externo.

Em um problema de SLAM são usados muitos sensores no sistema de percepção de robôs, tais como sonares e scanner lasers. Porém, além da confiabilidade da percepção, para a aceitação generalizada de aplicações, é necessário que as tecnologias utilizadas proporcionem uma solução de baixo custo.

O Cellbot, um aparelho smartphone acoplado a um sistema embarcado, apresenta uma alta praticidade, baixo custo e qualidade devido a facilidade na obtenção de aparelhos eletrônicos moveis smarphone de ponta. O Cellbot apresenta dois receptores wi-fi, um do sistema embarcado e um do smartphone, além dos demais sensores utilizáveis do smartphone, como camera, GPS, bussola, etc.

Dessa forma, objetivou-se desenvolver uma modelagem SLAM capaz de lograr da praticidade, baixo custo, e atributos do sistema embarcado e smartphone simultaneamente. Visto que o Cellbot possui dois receptores Wi-Fi objetivou-se desenvolver uma modelagem capaz de usufruir desta peculiaridade criando um sistema SLAM para dois receptores.

1.5 Objetivos

Objetivo Geral: O objetivo deste trabalho é o desenvolvimento de uma técnica de SLAM (Localização e Mapeamento Simultâneos) adequada para ambientes que possuem transmissores Wi-Fi, de modo que permita o robô navegar no ambiente mesclando informações de odometria e de força de sinal para um robô móvel da arquitetura Cellbot utilizando dois receptores WiFi.

Objetivos Específicos:

- desenvolver um sistema de SLAM eficiente para Cellbots utilizando odometria mecânica e força de sinal Wi-Fi;
- modelar a odometria como método de predição a ser utilizado no Filtro de Kalman Estendido;
- modelar o sinal WiFi como método de atualização a ser utilizado no Filtro de Kalman Estendido;
- modelar a solução SLAM com dois receptores WiFi utilizando o Filtro de Kalman Estendido;
- avaliar a solução via simulação.

2 Materiais e Métodos

O objetivo deste Capítulo é apresentar as ferramentas utilizadas no problema de SLAM. Existem várias etapas envolvidas no SLAM e elas podem ser resolvidas usando um conjunto de diferentes técnicas. Será apresentado a odometria mecânica, os conceitos matemáticos sobre filtro de Kalman, métodos de localização por Wi-Fi e um conjunto de trabalhos que mostram a evolução da área.

2.1 Odometria Mecânica

O método da odometria mecânica consiste na determinação da posição e orientação de um robô através da integração dos deslocamentos incrementais de suas rodas, medidos a partir de um referencial fixo. Para exemplificar esse método, considere um robô deslocando-se em um ambiente seguindo uma trajetória reta, como mostrado na Figura 9. Percebe-se então que a posição atual do robô (P_2) pode ser calculada baseada no acúmulo dos deslocamentos (D_1, D_2) realizados por ele, com relação a sua posição inicial (P_0).

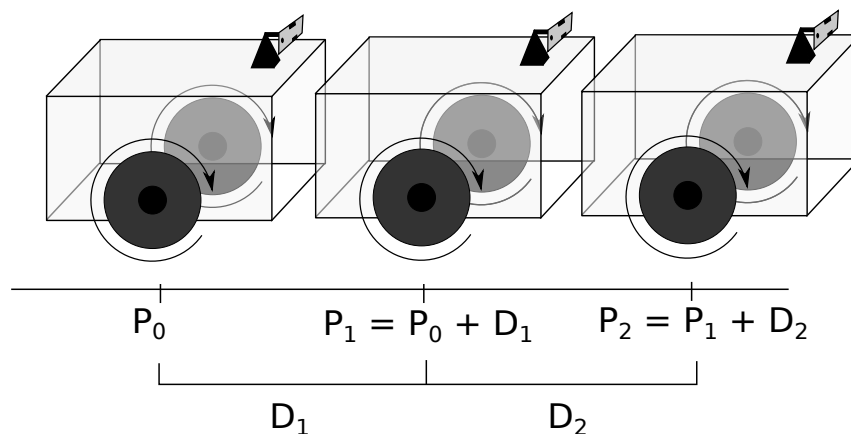


Figura 9 – Posicionamento resultante da integração de deslocamentos ao longo do tempo.

Para medir o deslocamento do robô é necessário a utilização de um sensor que possa medir a rotação das suas rodas. Em aplicações envolvendo robôs móveis, um tipo de sensor de odometria frequentemente utilizado é o *encoder* óptico. Esse tipo de sensor funciona com base na transmissão e recepção de luz através de um disco perfurado, que gira acoplado ao eixo que movimentava a roda do robô.

Mesmo os sistemas mais refinados não conseguem retratar com fidelidade o comportamento da movimentação dos robôs. A técnica de odometria está sujeita a erros que fazem com que ela forneça apenas uma estimativa da localização exata do robô em um dado instante de tempo (LATOMBE, 1991).

Como a odometria é calculada por meio do deslocamento do instante de tempo anterior, a cada instante de tempo ela está sujeita a erros. Dessa forma, à medida que ocorre deslocamento há um acúmulo dos erros de todos os instantes anteriores. Portanto ao utilizar a odometria, quanto maior o deslocamento realizado, mais impreciso será a posição final do robô.

2.2 Filtro de Kalman

O filtro de Kalman é um conjunto de equações matemáticas criado por Rudolf Kalman para filtragem e predição em sistemas gaussianos lineares. O filtro de Kalman estima o estado de um processo de modo a minimizar a média do erro quadrático. Este filtro demonstra ser bastante robusto, capaz de estimar os estados de um processo ao longo do tempo mesmo que a natureza do sistema modelado seja desconhecida (KALMAN, 1960).

No Kalman, cada valor recebido pelo sistema é interpretado como uma gaussiana, onde sua media corresponde ao valor recebido, e sua variância, o grau de certeza deste dado. Por exemplo, considerando que João está no laboratório de computação, o qual está localizado à 32° de latitude e 54° de longitude e possui área de 300 m^2 , portanto a média da gaussiana seria o ponto [32,54] e sua variância 300m, que seria o mesmo que dizer que João poderia estar a qualquer lugar a 300m, do ponto central do laboratório de computação.

Na Figura 10 temos um exemplo de funcionamento do Filtro de Kalman. Na predição, o robô se movimenta e estima sua posição atual, devido à erros ocasionados pela imprecisão dos sensores do robô, sua variância vai aumentar, ou seja, o robô vai perder a certeza de onde se encontra. Na etapa de atualização, o robô recebe informação relativa à sua posição de acordo com um sensor. É importante frisar que os sensores também possuem imprecisões, portanto a gaussiana azul representando o dado do sensor possui uma incerteza, ou seja, sua variância não é nula. Após receber a informação do sensor, ela é fundido com a incerteza da posição do robô retornando uma posição mais certa sobre onde o robô deve estar representado pela gaussiana vermelha.

2.2.1 Filtro de Kalman Estendido

O Filtro de Kalman Estendido (FKE) possui duas etapas: predição e atualização, e atua em sistemas gaussianos não-lineares. O filtro de Kalman estendido lineariza baseado na média e na covariância utilizando derivadas parciais do processo e das funções de medição para calcular a estimativa (THRUN; BURGARD; FOX, 2005).

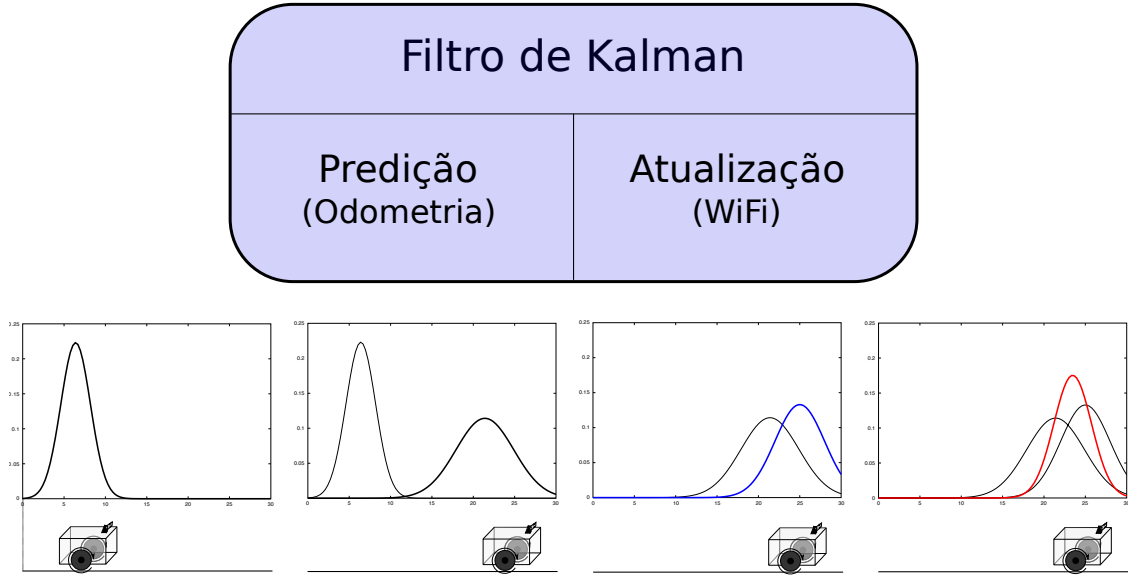


Figura 10 – Filtro de Kalman

O sistema modela o FKE seguindo a Equação 2.1,

$$\begin{cases} s_t = g(u_{t-1}, s_{t-1}) + \varepsilon_{t-1} \\ z_t = h(s_t) + \delta_t \end{cases} \quad (2.1)$$

onde $g(u_{t-1}, s_{t-1})$ representa uma função não-linear que modela o sistema, $h(s_t)$ é uma função não-linear que modela as medições, s_t é o vetor de estados, u_{t-1} são os sinais de entrada, z_t representa o vetor de observações retornadas pelo sensor, e os sinais ε_{t-1} e δ_t são ruídos gaussianos. O Sistema de Equações 2.2 representa a etapa de predição e o Sistema de Equações 2.3 representa a etapa de atualização do FKE,

$$\begin{cases} \bar{\mu}_t = g(u_{t-1}, s_{t-1}) \\ \bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t \end{cases} \quad (2.2)$$

$$\begin{cases} K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1} \\ \mu_t = \bar{\mu}_t + K_t (z_t - h_t(\bar{\mu}_t)) \\ \Sigma_t = (I - K_t H_t) \bar{\Sigma}_t \end{cases} \quad (2.3)$$

onde a matriz $G_{n \times m}$ representa o jacobiano que lineariza o sistema e a matriz $H_{l \times n}$ representa o jacobiano que lineariza o vetor das medições, K_t é a matriz denominada ganho do Kalman e representa a minimização da covariância no tempo t , R_t é a matriz de covariância do ruído do modelo de processo ε_{t-1} e Q_t é a matriz de covariância do ruído da medição δ_t . Os elementos $\bar{\mu}_t$ e $\bar{\Sigma}_t$ são, respectivamente, o vetor de estados estimado

após a correção e a covariância do vetor de estados. As matrizes G e H são definidas respectivamente nas Equações 2.4 e 2.5,

$$G_t = \frac{\partial g(u_{t-1}, s_{t-1})}{\partial s_{t-1}} \quad e \quad (2.4)$$

$$H_t = \frac{\partial h(s_t)}{\partial s_t} \quad . \quad (2.5)$$

O Algoritmo 1 demonstra uma implementação do FKE (THRUN; BURGARD; FOX, 2005).

Algoritmo 1: Filtro de Kalman Estendido

Require: $\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$

Ensure: μ_t, Σ_t

- 1: $\bar{\mu}_t = g(u_{t-1}, s_{t-1})$
 - 2: $\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$
 - 3: $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
 - 4: $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$
 - 5: $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$
-

2.3 Localização *indoor* com Wi-Fi

Com a popularização das redes sem fio Wi-Fi, muitos trabalhos na área de localização tem surgido utilizando leituras da força do sinal Wi-Fi emitido pelos pontos de acesso. Wi-Fi é uma marca registrada da *Wi-Fi Alliance* e é utilizada para certificar produtos que pertencem à classe de dispositivos de rede local sem fios (*WLAN*) baseados no padrão IEEE 802.11 (IEEE, 2016).

Muitas abordagens têm sido criadas sobre o processo de localização baseado na propagação do sinal das redes sem fio. Liu et al. (2007) citam que os métodos de localização utilizando sinais *wireless indoor* podem ser classificados baseados em duas características: quanto ao algoritmo para localizar a posição (o método que determina a localização, utilizando diversos tipos de medições como *Time Of Flight*, ângulo e força do sinal) e quanto a parte física ou infraestrutura do sensor utilizado (a tecnologia *wireless* envolvida na comunicação com os dispositivos móveis ou estáticos). Os métodos de localizar posição a partir do sinal *wireless* são divididos da seguinte maneira:

- **Análise de Cenário:** métodos que primeiro coletam características do sinal *wireless* no ambiente e depois estima a localização de um objeto através da correlação da informação do sinal atual com o sinal adquirido anteriormente. A utilização de um método baseado em *RSS* é comum na etapa de análise do cenário.

- **Trilateração (triangulação):** métodos que utilizam as propriedades geométricas do triângulo para encontrar a posição do alvo. Podem ser divididos em métodos por angulação e lateração:
 - **Angulação:** estima a posição do alvo através da interseção de vários pares de ângulos das linhas de direção, cada linha é formada pelo raio do círculo de alcance do sinal partindo da estação base ao receptor móvel.
 - **Lateração:** estima a posição do objeto através da leitura de distâncias a partir de múltiplos pontos de referência, tais como:
 - **TOA (*Time Of Arrival*):** a distância do alvo móvel até a unidade medidora é diretamente proporcional ao tempo de propagação;
 - **TDOA (*Time Difference Of Arrival*):** a ideia básica desse método é determinar a posição relativa do transmissor através da diferença de tempo que o sinal chega nas unidades receptoras;
 - **RTOF (*Reflection Time Of Flight*):** esse método utiliza o tempo de vôo do sinal, medido a partir do emissor até a chegada ao receptor, para estimar a posição;
 - **RSP (*Received Signal Phase*):** utiliza a diferença de fase no sinal das portadoras (de telefonia, por exemplo) para estimar a posição;
 - **Baseadas em RSS (*Received Signal Strength*):** como os métodos *TOA* e *TDOA* necessitam da inexistência de obstáculos entre o emissor e o receptor para uma melhor precisão do sistema, os métodos baseados em *RSS* tentam calcular a posição baseado na atenuação do sinal durante a propagação.

2.4 Trabalhos Relacionados

O problema de Range-only SLAM tem sido estudado por pesquisadores que propuseram diferentes métodos para solucionar o problema de multi-hipóteses. Para solucionar o problema de mapeamento, [Olson, Leonard e Teller \(2006\)](#) propõem uma solução baseada em grade para mapear a posição de diferentes *beacons* (sensores de distância com localização desconhecida). A solução é melhorada com o emprego de um método robusto para rejeição de *outlier* baseado em particionamento de grafo espectral.

Por outro lado, [Caballero et al. \(2008\)](#) propõem uma abordagem probabilística baseada em filtro de partículas para computar a estimativa da posição inicial 2D de cada posição de *beacon* enquanto o veículo aéreo se move através dos *beacons*. Uma vez que o filtro de partículas de cada beacon converge para uma solução gaussiana, o processo de estimação é substituído pelo FKE. Apesar dos resultados deste método serem precisos,

a principal desvantagem é a demora da inicialização do FKE, o qual perde todas as informações de distância utilizadas no filtro de partículas.

Posteriormente, os mesmos autores propuseram uma solução com multi-hipoteses enquanto o FKE estima a posição real dos *beacons* (CABALLERO; MERINO; OLLERO, 2010). Para a modelagem da informação de sinal, o método foi baseado na parametrização polar proposta em (DJUGASH; SINGH, 2008). Os resultados desse método são precisos, e em comparação ao trabalho anterior, o FKE pode ser inicializado a partir da primeira medida de distância.

Outras ferramentas convencionais de SLAM são comparadas em (KURT-YAVUZ; YAVUZ, 2012; TUNA et al., 2012), onde a abordagem Fast-SLAM é considerada a solução mais precisa e eficiente. Fast-SLAM é usado também em (YANG, 2012; WANG; MIAO; DU, 2009), onde o filtro de partículas é usado para o problema de mapeamento e localização. Duas técnicas de otimização para localizar diferentes *beacons* são aplicadas no filtro de partículas. Outra solução Fast-SLAM é proposta em (Dan Hai et al., 2010; BLANCO; GONZÁLEZ; FERNÁNDEZ-MADRIGAL, 2008), onde o filtro de partícula do problema de mapeamento é substituído por um FKE quando cada partícula converge para uma distribuição gaussiana reduzindo o custo computacional das abordagens anteriores.

Em (ZHOU, 2009), é utilizada a técnica de trilateração para computar uma estimativa dos *beacons* para inicialização do FKE, onde três medidas de distância gravadas em posições diferentes do robô são utilizadas para calcular a interseção dos três círculos cujos centros são não colineares. Apesar da simplicidade, esse método é sensível ao ruído e útil para o emprego de mais de três medidas de distância.

No contexto de técnicas de inicialização do FKE, (GENEVE; KERMORGANT; LAROCHE, 2015) realiza um comparativo das principais técnicas utilizadas para estimar os *beacons* e propõe um método que combina uma inicialização rápida do FKE com um número limitado a partir de duas hipóteses por cada beacon. O método atuou de forma mais precisa que o filtro de partículas e mais custosa que o método de trilateração, porém, mais rápida que os dois métodos para inicializar o FKE.

Fabresse et al. (2016) implementa uma técnica de percepção ativa para reduzir a incerteza do mapa estimado no RO-SLAM. O método é baseado em uma técnica de seleção de ação que usa a entropia da crença do SLAM para calcular o ganho estimado de informação resultante de cada ação possível. O conjunto de ações consideradas nesta abordagem baseia-se em uma velocidade fixa e um conjunto de ângulos de velocidade de elevação e azimute entre os quais o sistema deve tomar uma decisão para reduzir a incerteza no mapa.

2.5 Considerações Finais

Este Capítulo apresentou o conjunto de ferramentas utilizadas para solucionar o SLAM utilizado neste trabalho. Mostrou-se a odometria mecânica, o método utilizado para estimar a posição do robô através de sensores acoplados a roda. Explicou-se a definição e utilização do Filtro de Kalman, ferramenta matemática utilizada para implementar o sistema SLAM. Descreveu-se os métodos de localização indoor com Wi-Fi utilizados na literatura, bem como o método de trilateração baseada em RSS utilizado neste trabalho para inicializar a posição dos APs no Filtro de Kalman.

Por fim, foram apresentados diversos trabalhos contemplando o uso de Range-only SLAM na literatura e, dentre estes trabalhos, dois se destacam. Primeiramente o trabalho de [Zhou \(2009\)](#), pois apresenta uma modelagem similar a trabalhada nesta dissertação diferindo no quesito a utilização de duas antenas de recepção. E a seguir o [Geneve, Kermorgant e Laroche \(2015\)](#), pois utiliza uma ampla comparação de inicializações que foi aplicada para a implementação da inicialização utilizada nesta dissertação. O Capítulo seguinte explica a abordagem Range-only SLAM proposto.

3 Sistema Proposto

O sistema proposto neste trabalho apresenta uma abordagem de SLAM utilizando informação de sinais Wi-Fi recebidas por dois receptores: smartphone e sistema embarcado, para filtrar o sinal recebido e computar a posição do robô e dos transmissores Wi-Fi. Esta abordagem é adequada para ambientes *indoor* onde há pelo menos um transmissor Wi-Fi, possibilitando o robô navegar no ambiente fundindo informações de odometria e força de sinal Wi-Fi.

Inicialmente, é realizada a leitura dos sinais de Wi-Fi do ambiente. A força de sinal recebida pelo robô é composta de diversos ruídos advindos de obstáculos entre o robô e transmissor ou componentes eletrônicos. Em seguida, o sinal é transformado em metros utilizando uma função logarítmica obtida por meio da modelagem do sensor de recepção Wi-Fi. Com o sinal convertido em metros, é feita uma estimativa da posição dos marcos correspondente a cada sinal Wi-Fi recebido. Por fim, a pose do robô é corrigida juntamente com a posição dos marcos.

O algoritmo utilizado para a obtenção do mapa de características é baseado no Filtro de Kalman Estendido (FKE), cuja fase de predição é implementada usando o modelo de odometria do robô. Já na fase de atualização, os sinais de Wi-Fi são convertidos em metros e aplicados no FKE para obter novas medidas de pose do robô e dos marcos. Optou-se pelo uso do FKE em detrimento ao Filtro de Partículas devido ao seu alto custo computacional. Ressalta-se que no sistema proposto existem dois receptores, portanto ocorrem conversões diferentes do mesmo sinal recebido devido a diferente percepção de cada um dos sensores. A Figura 11 mostra uma visão geral do sistema proposto.

No sentido de obter uma solução simples e eficiente, primamos pelo uso de técnicas clássicas existentes na literatura para vencer individualmente as etapas práticas de um sistema de localização e mapeamento simultâneos. Para odometria foi utilizada a modelagem proposta em (THRUN; BURGARD; FOX, 2005), o mapeamento do sinal Wi-Fi em metros é feito pela função logarítmica (FAHEEM; VIRRANKOSKI; ELMUSRATI, 2010), para estimativa inicial foi utilizado o algoritmo de trilateração (NORRDINE, 2012), e a implementação é de um EKF-SLAM clássico (THRUN; BURGARD; FOX, 2005). Neste Capítulo será detalhada a modelagem range-only SLAM.

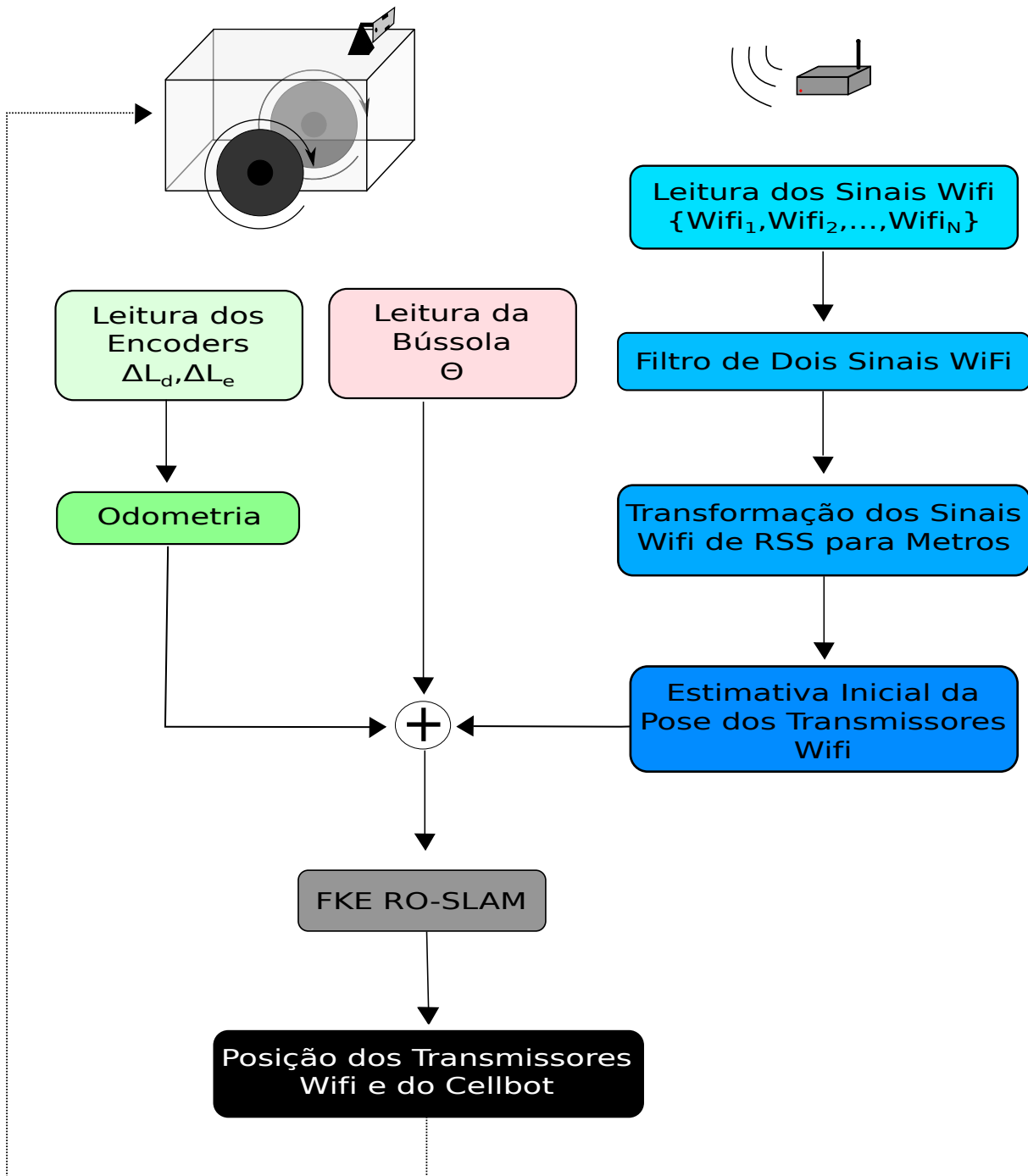


Figura 11 – Visão geral do sistema proposto.

3.1 Modelagem do Sistema

3.1.1 EKF - SLAM

O Filtro de Kalman Estendido SLAM trabalha de maneira semelhante ao FKE. Entretanto, esta solução além de estimar a pose do robô, também estima as características de todos os marcos encontrados pelo caminho. Assim, torna-se necessário incluir as poses dos marcos no vetor de estado. A cada iteração, o FKE calcula a melhor estimativa do vetor de estados com base nas informações recebidas no sensoriamento em duas fases: a predição e a atualização.

Fase de Predição: Modelo de Processo

Na fase de predição é necessário um modelo capaz de prever o comportamento do robô. Porém, variáveis como massa, momentos de inércia, atritos e atuadores tornam a modelagem do movimento do robô algo complexo, em que mesmo sistemas mais robustos são incapazes de retornar a posição exata do robô (LATOMBE, 1991).

Para a modelagem descrita neste capítulo, foi adotada a abordagem defendida por (THRUN; BURGARD; FOX, 2005) que propõe que as informações obtidas na odometria não funcionem como medidas sensoriais, e sim, que elas sejam incorporadas ao modelo de processo do robô. O modelo do processo utiliza as leituras dos sensores odométricos (*encoders*), funcionando como sinais de controle aplicados ao modelo do robô na fase de predição do filtro de Kalman.

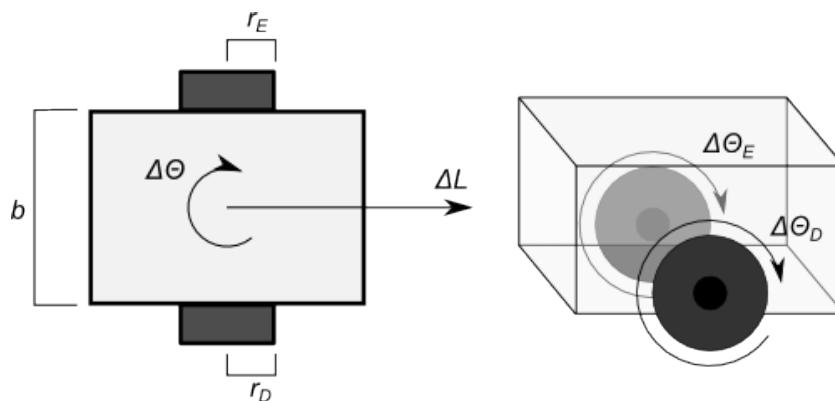


Figura 12 – Variáveis do modelo geométrico.

Assim, os sinais de controle submetidos aos atuadores do robô não são valores de tensão, mas sim, deslocamentos angulares nas rodas do robô mensurados nos *encoders* direito e esquerdo respectivamente: (θ_d e θ_e), conforme a Figura 12. Supondo que a velocidade durante os períodos de deslocamentos sejam constantes, pode-se determinar o modelo geométrico de movimentação do robô, como pode ser visto no Sistema de Equações

3.1:

$$\begin{cases} {}^R x_t = {}^R x_{t-1} + \frac{\Delta L}{\Delta \theta} [\sin(\theta_{t-1} + \Delta \theta \Delta t) - \sin(\theta_{t-1})] \\ {}^R y_t = {}^R y_{t-1} + \frac{\Delta L}{\Delta \theta} [\cos(\theta_{t-1} + \Delta \theta \Delta t) - \cos(\theta_{t-1})] \\ {}^R \theta_t = {}^R \theta_{t-1} + \Delta \theta \end{cases} \quad (3.1)$$

onde:

$$\begin{cases} \Delta L = (\Delta \theta_d r_d + \Delta \theta_e r_e) / 2 \\ \Delta \theta = (\Delta \theta_d r_d - \Delta \theta_e r_e) / b \end{cases} \quad (3.2)$$

$[{}^R x, {}^R y, {}^R \theta]$ é definido como a pose do robô, ΔL e $\Delta \theta$ são os deslocamentos linear e angular do robô; $\Delta \theta_d$ corresponde ao valor da leitura do *encoder* direito e representa o deslocamento angular aplicado à roda direita; $\Delta \theta_e$ corresponde ao valor da leitura do *encoder* esquerdo e representa o deslocamento angular aplicado à roda esquerda; b corresponde a distância do eixo das rodas do robô, e r_d e r_e representam os respectivos raios das rodas direita e esquerda do robô.

Para a modelagem da fase de predição deste trabalho, temos ainda uma informação angular ($\theta_{bússola}$) advinda do sensor de bússola do smartphone. Esta informação é então incorporada diretamente no modelo de predição como pode ser visto no Sistema de Equações 3.3:

$$\begin{cases} x_t = x_{t-1} + \frac{\Delta L}{\Delta \theta} [\sin(\theta_{t-1} + \Delta \theta \Delta t) - \sin(\theta_{t-1})] \\ y_t = y_{t-1} + \frac{\Delta L}{\Delta \theta} [\cos(\theta_{t-1} + \Delta \theta \Delta t) - \cos(\theta_{t-1})] \\ \theta_t = \theta_{bússola} \end{cases} \quad (3.3)$$

Na prática, os valores reais de deslocamento angular realizado pelas rodas do robô diferem do valor obtido na leitura dos *encoders* e a suposição de que as velocidades durante o período de deslocamento serão constantes não acontece em todos os casos. Dessa forma, essa diferença entre o valor real dos deslocamento angulares ($\Delta \theta_d$ e $\Delta \theta_e$) e os valores mensurados nos *encoders* ($\Delta \hat{\theta}_d$ e $\Delta \hat{\theta}_e$) é modelada como um ruído branco gaussiano na roda direita e esquerda (ε_d e ε_e) de acordo com o Sistema de Equações 3.4.

$$\begin{cases} \Delta \theta_d = (\Delta \hat{\theta}_d + \varepsilon_d) \\ \Delta \theta_e = (\Delta \hat{\theta}_e + \varepsilon_e) \end{cases} \quad (3.4)$$

Para o uso do FKE em SLAM incorporando os erros citados anteriormente (THRUN; BURGARD; FOX, 2005), propõe que a nova fase de predição seja dada pelo Sistema de

Equações 3.5, visto que a matriz G e R devem ser mapeadas de acordo com o novo número de estados.

$$\begin{cases} \bar{\mu}_t = \mu_{t-1} + p(u_{t-1}, \mu_{t-1}, \varepsilon) \\ \mathbf{G}_t = I + \mathbf{F}_x^T G \mathbf{F}_x \\ \bar{\Sigma}_t = \mathbf{G}_t \Sigma_{t-1} \mathbf{G}_t^T + \mathbf{F}_x^T \mathbf{V}_t \mathbf{M}_t \mathbf{V}_t^T \mathbf{F}_x \end{cases} \quad (3.5)$$

onde a matriz \mathbf{G}_t representa a matriz G mapeada no vetor de estados $2N + 3$, sendo N a quantidade de marcos incorporados no sistema, \mathbf{M} , $l \times l$, é a matriz de covariância dos ruídos dos sensores(ε), \mathbf{F} é a matriz que mapeia o estado de 3 dimensões em um estado de $2N + 3$ dimensão e \mathbf{V} , $n \times m$, é o jacobiano que mapeia o ruído dos sensores para o espaço do estado. A matriz \mathbf{V} é definida pela Equação 3.6.

$$\mathbf{V}_t = \frac{\partial p(u_{t-1}, \mu_{t-1}, \varepsilon)}{\partial u_{t-1}} \quad (3.6)$$

Portanto, utilizando os Sistemas 3.1 e 3.2 para calcular $p(\cdot)$, é possível calcular, por derivação do modelo, as matrizes G e V :

$$G = \begin{pmatrix} 0 & 0 & g_{13} \\ 0 & 0 & g_{23} \\ 0 & 0 & 0 \end{pmatrix} \quad (3.7)$$

onde:

$$g_{13} = \frac{\Delta L}{\Delta \theta} [\cos(\theta_{t-1} + \Delta \theta \Delta t) - \cos(\theta_{t-1})]$$

$$g_{23} = \frac{\Delta L}{\Delta \theta} [\sin(\theta_{t-1} + \Delta \theta \Delta t) - \sin(\theta_{t-1})]$$

$$V = \begin{pmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \\ 0 & \Delta t \end{pmatrix} \quad (3.8)$$

onde:

$$v_{11} = k1 \cos(k2) - k3[\sin(k2) - \sin(\theta_{t-1})]$$

$$v_{12} = -k1 \cos(k2) + k3[\sin(k2) - \sin(\theta_{t-1})]$$

$$v_{21} = \frac{k2 + \cos(\theta_{t-1})}{\Delta \theta}$$

$$v_{22} = -\frac{\Delta L(-k_2 + \cos(\theta_{t-1}))}{\Delta\theta^2} + \frac{k_1\Delta L\Delta t}{\Delta\theta}$$

$$M = \begin{pmatrix} \alpha_1|\Delta\theta_d| & 0 \\ 0 & \alpha_2|\Delta\theta_e| \end{pmatrix} \quad (3.9)$$

Os valores dos elementos m_{11} e m_{22} na Equação 3.9 representam o fato dos desvios padrões de ε_d e ε_e serem proporcionais ao módulo do deslocamento angular. As variáveis k_1 , k_2 e k_3 são dadas pelo Sistema de Equações 3.10.

$$\begin{cases} k_1 = \frac{2r\Delta L}{b^2\Delta\theta} \\ k_2 = \theta_{t-1} + r\Delta\theta \\ k_3 = \frac{b\Delta L}{2(r(\Delta\theta_d - \Delta\theta_e)/b)^2} \end{cases} \quad (3.10)$$

Fase de Atualização: Modelo do Sensor

Neste trabalho, as posições dos transmissores Wi-Fi foram utilizadas como marcos para realizar a atualização, entretanto, antes de iniciar a etapa de atualização do FKE, é preciso inicializar o espaço de estados, como pode ser visto na Equação 3.11. A posição inicial do robô (${}^R x_0, {}^R y_0, {}^R \theta_0$) é dada por $(0, 0, 0)$, porém para as posições dos marcos (${}^M x_1, {}^M y_1, \dots, {}^M x_n, {}^M y_n$) é necessário executar um método de localização Wi-Fi com base no RSS. Neste trabalho, o método escolhido foi o método de trilateração.

$$\mu_0 = ({}^R x_0, {}^R y_0, {}^R \theta_0, {}^M x_1, {}^M y_1, \dots, {}^M x_n, {}^M y_n)^T \quad (3.11)$$

No método de trilateração, foi utilizado a estimativa da posição do robô à medida que ele se desloca para calcular a posição do marco por meio da interseção das circunferências centradas na antena do robô em relação ao nó transmissor, como demonstrado na Figura 13. Para determinar o ponto de interseção, foi utilizado uma adaptação do método proposto por Faheem, Virrankoski e Elmusrati (2010) pela simplicidade algébrica.

De acordo com a Figura 13, temos que um número n de medidas da posição do robô e distância entre robô e marco são utilizados para determinar o posição (${}^M x, {}^M y$) do marco. Para tanto é utilizada a equação:

$$\begin{bmatrix} 1 & -2{}^R x_1 & -2{}^R y_1 \\ 1 & -2{}^R x_2 & -2{}^R y_2 \\ \vdots & \vdots & \vdots \\ 1 & -2{}^R x_n & -2{}^R y_n \end{bmatrix} \begin{bmatrix} {}^M x^2 + {}^M y^2 \\ {}^M x \\ {}^M y \end{bmatrix} = \begin{bmatrix} d_1^2 - {}^R x_1^2 - {}^R y_1^2 \\ d_2^2 - {}^R x_2^2 - {}^R y_2^2 \\ \vdots \\ d_n^2 - {}^R x_n^2 - {}^R y_n^2 \end{bmatrix} \quad (3.12)$$

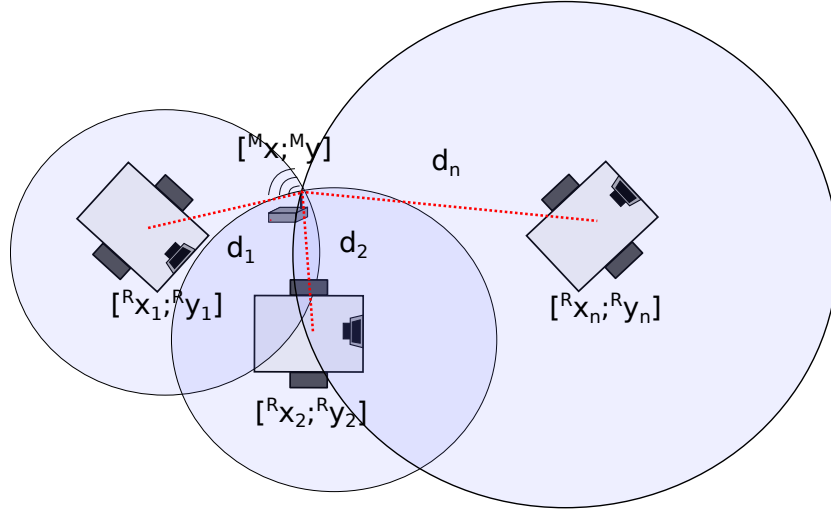


Figura 13 – Triangulação dos marcos.

A Equação 3.12 pode ser representada de maneira simplificada pela equação:

$$A \cdot \mathbf{m} = \mathbf{b} \quad (3.13)$$

A solução da Equação 3.13 utilizando o método do erro mínimo quadrado é dada por:

$$\hat{\mathbf{m}} = (A^T A)^{-1} A^T \mathbf{b} \quad (3.14)$$

Porém, as distâncias entre o robô e o marco possuem diversos ruídos que podemos representar pela matriz de covariância S , para uma solução com informação mais consistente é utilizado o método do erro mínimo quadrado ponderado mostrado na Equação 3.15:

$$\hat{\mathbf{m}} = (A^T S^{-1} A)^{-1} A^T S^{-1} \mathbf{b} \quad (3.15)$$

Uma vez que a primeira estimativa das coordenadas dos nós transmissores é obtida, a coordenada é inserida no FKE e a etapa de atualização se inicia. Cada nova medida de RSS (z_t) recebida pelo robô pode ser inferida por meio do modelo do sensor expresso na Equação 3.16.

$$h(\bar{\mu}_t) = \left[\frac{\sqrt{(Sx - Mx)^2 + (Sy - My)^2}}{\sqrt{(Ex - Mx)^2 + (Ey - My)^2}} \right] \quad (3.16)$$

onde:

- Rx_i e Ry_i correspondem à posição atual do robô;
- Mx e My a posição do marco;

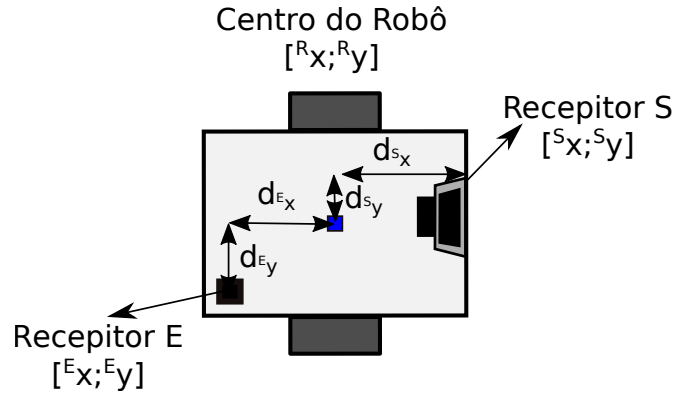


Figura 14 – Triangulação dos marcos.

- $S_x = R_{x_i} + d_{S_x}$
- $S_y = R_{y_i} + d_{S_y}$
- $E_x = R_{x_i} - d_{E_x}$
- $E_y = R_{y_i} - d_{E_y}$

Ao linearizarmos com a jacobiana obtemos:

$$H_t^T = \begin{bmatrix} \frac{S_x - M_{x_k}}{\sqrt{(S_x - M_{x_k})^2 + (S_y - M_{y_k})^2}} & \frac{E_x - M_{x_k}}{\sqrt{(E_x - M_{x_k})^2 + (E_y - M_{y_k})^2}} \\ \frac{S_y - M_{y_k}}{\sqrt{(S_x - M_{x_k})^2 + (S_y - M_{y_k})^2}} & \frac{E_y - M_{y_k}}{\sqrt{(E_x - M_{x_k})^2 + (E_y - M_{y_k})^2}} \\ 0 & 0 \\ \frac{M_{x_k} - S_x}{\sqrt{(S_x - M_{x_k})^2 + (S_y - M_{y_k})^2}} & \frac{M_{x_k} - E_x}{\sqrt{(E_x - M_{x_k})^2 + (E_y - M_{y_k})^2}} \\ \frac{M_{y_k} - S_y}{\sqrt{(S_x - M_{x_k})^2 + (S_y - M_{y_k})^2}} & \frac{M_{y_k} - E_y}{\sqrt{(E_x - M_{x_k})^2 + (E_y - M_{y_k})^2}} \\ \frac{M_{x_k} - S_x}{\sqrt{(S_x - M_{x_k})^2 + (S_y - M_{y_k})^2}} & \frac{M_{x_k} - E_x}{\sqrt{(E_x - M_{x_k})^2 + (E_y - M_{y_k})^2}} \\ \frac{M_{y_k} - S_y}{\sqrt{(S_x - M_{x_k})^2 + (S_y - M_{y_k})^2}} & \frac{M_{y_k} - E_y}{\sqrt{(E_x - M_{x_k})^2 + (E_y - M_{y_k})^2}} \end{bmatrix} \quad (3.17)$$

Com as Equações 3.16 e 3.17 podemos realizar as etapas de atualização do FKE descritas no Sistema 2.3. A modelagem do erro do sensor Q_t é descrito na Equação 3.18:

$$Q_t = \begin{bmatrix} \sigma_S & 0 \\ 0 & \sigma_E \end{bmatrix} \quad (3.18)$$

onde σ_S e σ_E é o erro do sinal relacionado respectivamente ao receptor 1 e receptor 2.

3.1.2 Sinal Wi-Fi

Dentre as informações recebidas pelo sinal Wi-Fi, neste trabalho é utilizado a força de sinal RSSI, o endereço MAC e a qualidade do sinal, de modo que o endereço MAC é utilizado para realizar a correspondência entre o sinal recebido e o marco que enviou a informação, enquanto a qualidade é usada para realizar o processo de filtragem no sinal recebido. No sistema proposto a força do sinal RSSI é empregada para inferir informação de distância entre os marcos e o robô, para tanto é necessário a modelagem da propagação do sinal para realizar o mapeamento do sinal em metros. A Figura 15 ilustra as informações do sinal utilizadas.

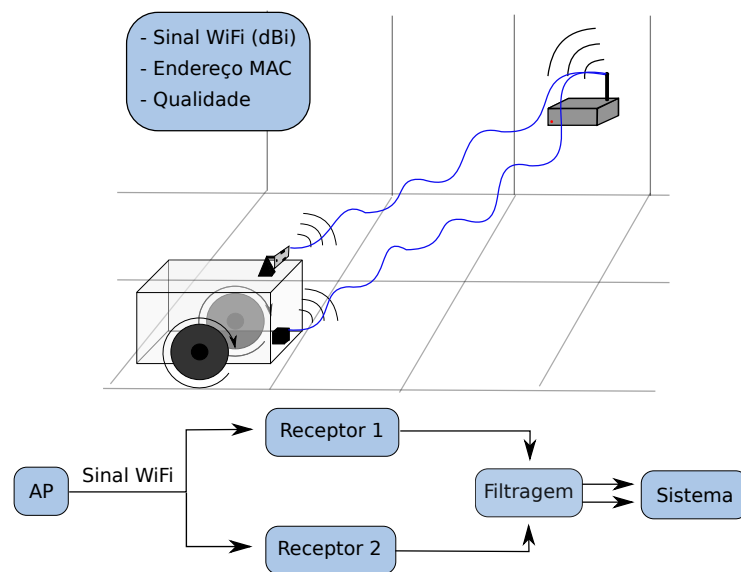


Figura 15 – Informações do Sinal Wi-Fi utilizadas no sistema.

Ao longo dos anos, um grande número de modelos para predição da propagação de sinal é encontrado na literatura para modelar típicos ambientes dotados de transmissores Wi-Fi como pequenos parques urbanos, prédios, centros de compra e hospitais. Estes modelos são principalmente baseados em medidas empíricas de distâncias diferentes para um dado raio de frequência. Portanto, para as experimentações deste trabalho, foi escolhido um modelo simplificado da propagação de sinal, definido pela Equação 3.19:

$$P = P_0 - 10\eta \log_{10}(d) \quad (3.19)$$

onde P é o sinal recebido pelo transmissor; P_0 corresponde a força de transmissão nominal à uma distância de um metro do transmissor; η é o expoente de propagação; d é a distância entre o transmissor e o receptor.

Para determinar os valores de P_0 e η de modo a obter a função de propagação do sinal foram realizados experimentos utilizando o smartphone Nexus 5 e o sistema embarcado Raspberry Pi 3, onde foi capturado a leitura RSS de 100 amostras a cada 10

centímetros por cada um dos aparelhos e aplicado técnica de ajuste de curva por regressão para obter a modelagem do sinal de cada um dos aparelhos.

Por uma questão de conveniência, vamos adotar receptor S correspondente ao smartphone Nexus 5, e receptor E ao sistema embarcado. No receptor S foram obtidos os valores -36 e 2.21, já no receptor E foram obtidos os valores -30 e 2.48, para P_0 e n , de acordo com a Equação 3.19. A Figura 16 ilustra as funções de propagação do sinal Wi-Fi para cada receptor.

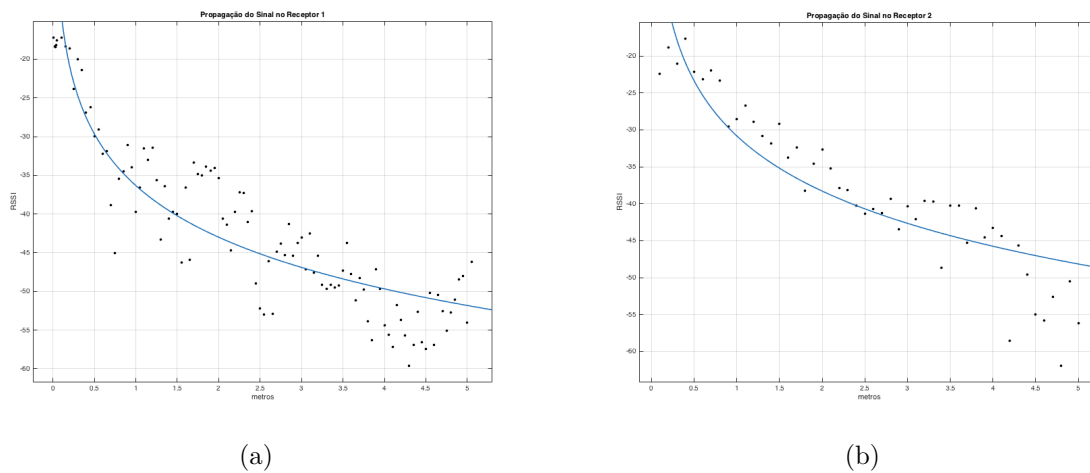


Figura 16 – Função de Propagação do Sinal Wi-Fi para: (a) Receptor 1; (b) Receptor 2.

O comportamento da variância de cada um dos receptores é dado pela Equação 3.20. Os valores das constantes foram calculados por experimentação sendo $a = 0.3256$ e $b = 0.3975$ para o receptor 1, e $a = 0.02039$ e $b = 1.156$ para o receptor 2. A Figura ?? mostra o gráfico do comportamento da variância considerando as constantes calculadas.

$$\sigma^2(r) = a.e^{b.r} \quad (3.20)$$

A variância do sinal é utilizada para simular a qualidade do sinal recebido por cada receptor. Por meio de experimentação foi determinado um threshold para cada receptor de acordo com a função de variância. Ao receber um sinal que ultrapasse o threshold definido, este sinal é descartado.

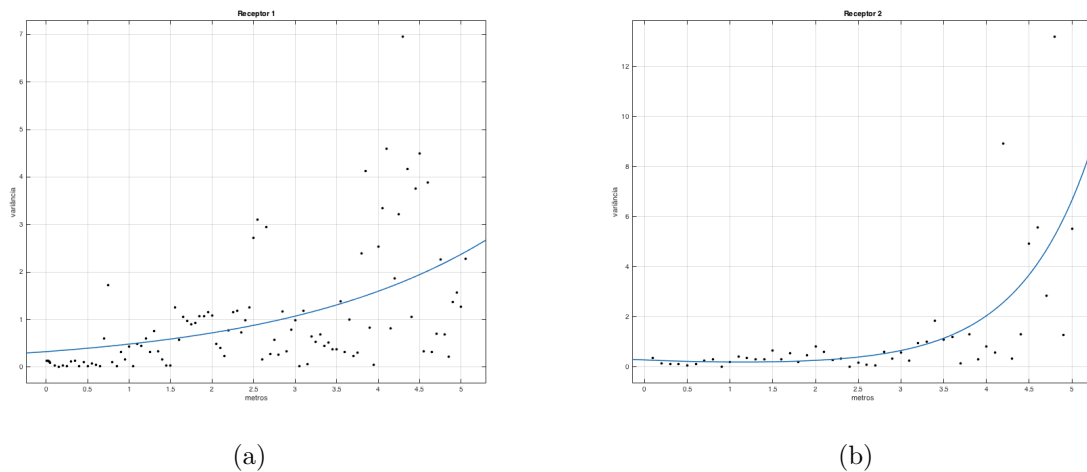


Figura 17 – Função de Propagação do Sinal Wi-Fi para: (a) Receptor 1; (b) Receptor 2.

3.2 Considerações Finais

Este Capítulo apresentou o sistema Range-only SLAM utilizado neste trabalho. Mostrou-se os aspectos gerais do sistema que tem como objetivo final a construção de um mapa de características informando as posições de transmissores Wi-Fi ao longo do percurso do robô.

Foi mencionado que a base do sistema de SLAM apresentado é o Filtro de Kalman Estendido modelado para dois receptores Wi-Fi. Apresentou-se a modelagem matemática envolvidas nessa solução e também foram mostrados alguns trabalhos de influência para esta proposta. A parte final ficou com a modelagem do sinal Wi-Fi. O Capítulo seguinte tratará os resultados simulados do Range-only SLAM proposto.

4 Resultados e Discussão

Este Capítulo apresentará os resultados do range-only SLAM descrito matematicamente no Capítulo 3. O sistema é baseado no alcance do sinal Wi-Fi e usa como marcos do SLAM a posição $[^Mx, ^My]$ do transmissor Wi-Fi. Os experimentos foram realizados pela simulação utilizando o robô *Pioneer 3dx* na plataforma de simulação V-REP (ROHMER; SINGH; FREESE, 2013). Foi simulado um *encoder* óptico utilizando a velocidade de cada roda do robô. Além disso, foram acoplados dois receptores Wi-Fi em posições diferentes no robô. Para mostrar os resultados foram utilizados cenários a fim de facilitar a discussão sobre cada experiência realizada.

O ruído da odometria foi obtido por meio de experimentação, onde o robô real foi comandado a andar 1 metro de distância. Após 20 medidas, foi calculado a média do erro proporcional a 1 metro. Visto que o simulador V-REP não possui sensor de *encoder* para simula-lo, foi necessário extrair a velocidade das rodas e transforma-la em metros e assim aplicar o ruído proporcional à distância deslocada neste instante. Após inserido o ruído, a distância é convertida novamente em metros e aplicada no sistema para o cálculo da odometria. A Figura 18 ilustra o procedimento de simulação da odometria.

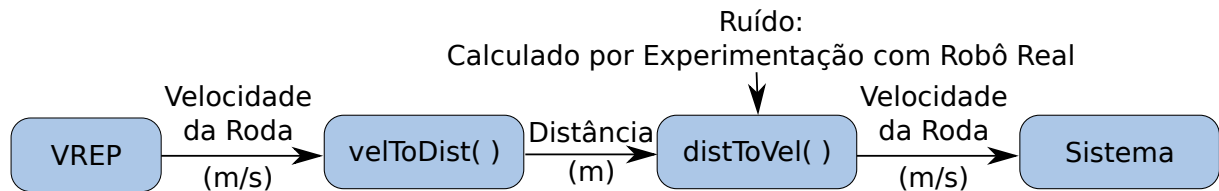


Figura 18 – Simulação do Ruído da Odometria.

Para simular o sinal Wi-Fi foi determinada por experimentação a função de variância de cada receptor do sistema. A seguir, foi extraída a distância entre os marcos e o robô e aplicado o ruído modelado para cada receptor por experimentação. Assim são obtidas duas distâncias referentes ao receptor S e receptor E utilizadas no sistema. A Figura 19 mostra o procedimento de simulação do sinal Wi-Fi.

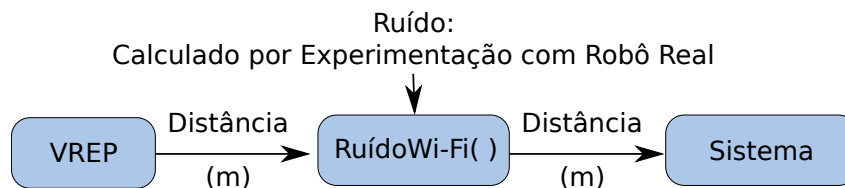


Figura 19 – Simulação do Ruído do Sinal Wi-Fi.

O robô utilizado em todos os cenários apresenta 0,52 metros de comprimento e 0,42 metros de largura. Da cena A à cena D será utilizado o mapa da Figura 20 que possui

16 metros de comprimento e 8 metros de largura. O percurso realizado pelo robô neste mapa é destacado em roxo.

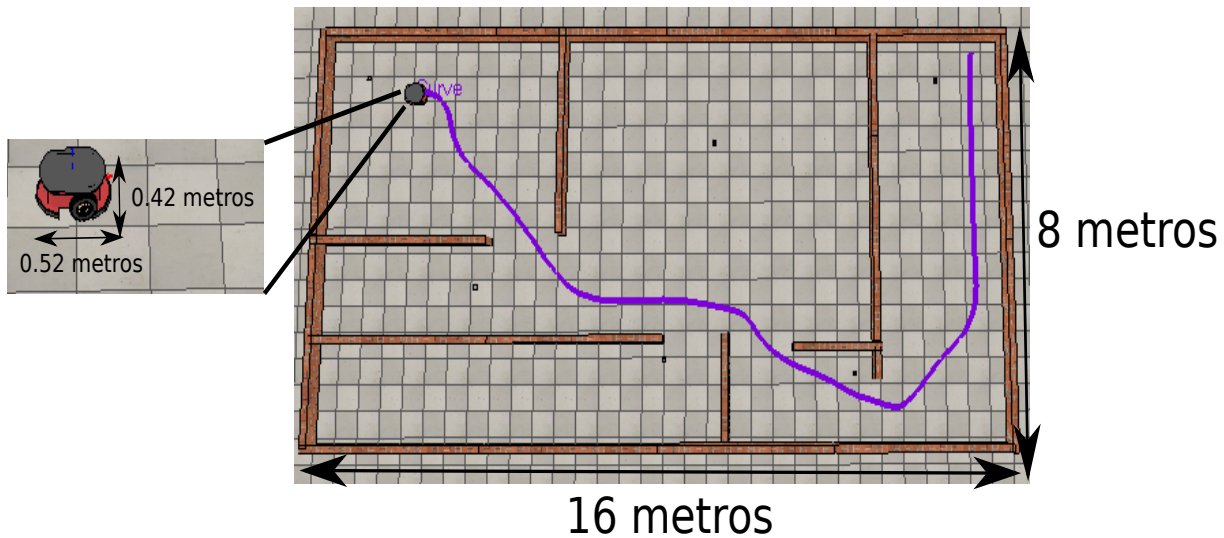


Figura 20 – Mapa do Cenário A, B, C e D

Para cada cena trabalhada no mapa da Figura 20 é mostrado resultados com 3 APs, 4 APs e 6 APs com o intuito de demonstrar a influência da cobertura dos transmissores ao longo dos experimentos. Como base de comparação para a solução de SLAM com 2 receptores deste trabalho, utilizou-se a implementação de um sistema SLAM com 1 receptor. Como previamente explanado no Capítulo 3, quando há ruído no sistema a solução realiza uma filtragem do sinal de modo a descartar sinais com ruídos. Contudo, ao realizar este descarte ocorre uma limitação do alcance do sinal recebido que tem influência direta na visibilidade dos APs ao longo do percurso. Caso não haja ruído no sistema, a solução tem o mesmo comportamento de visibilidade dos APs que o SLAM com 1 receptor. Por uma questão de conveniência, vamos adotar SLAM 1R para SLAM com 1 receptor, e SLAM 2R para SLAM com 2 receptores. Nas Figuras 21, 22 e 23 seguem as ilustrações da cobertura e visibilidade dos APs dos cenários com 3 APs, 4 APs e 6 APs, respectivamente.

A Figura 21 mostra a cobertura e visibilidade do sinal Wi-Fi com 3 APs. No SLAM 1R 4% das amostras teve visualização de 2 APs, 83% das amostras teve visualização de 1 AP e 14% das amostras não teve visualização de APs. No SLAM 2R 63% das amostras teve visualização de 1 AP e 37% das amostras não teve visualização de APs. A Tabela 1 mostra os valores de visibilidade de cada AP.

Tabela 1 – Porcentagem de Visibilidade por AP - 3 APs

	SLAM - 1 Receptor	SLAM - 2 Receptores
AP1	25%	29%
AP2	52%	51%
AP3	23%	20%

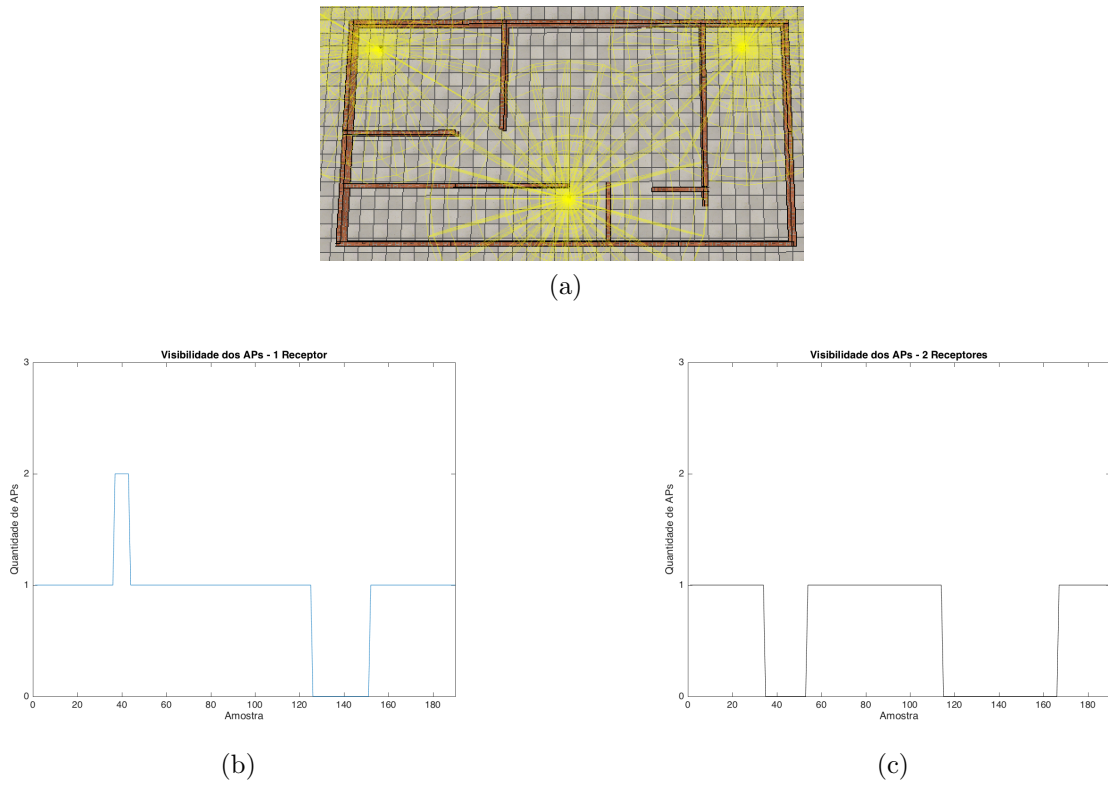


Figura 21 – Cenário com 3 APs: (a) Cobertura do Sinal Wi-Fi (b) Disponibilidade do Sinal Wi-Fi com 1 Receptor (c) Disponibilidade do Sinal Wi-Fi com 2 Receptores.

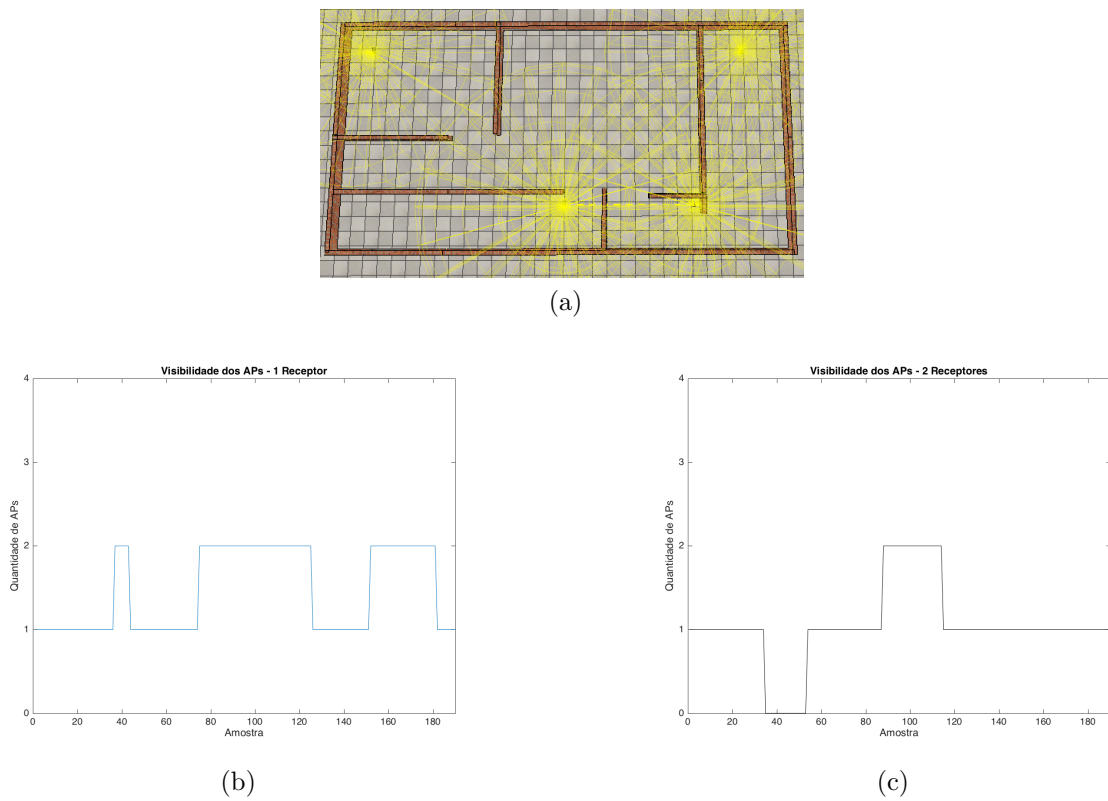


Figura 22 – Cenário com 4 APs: (a) Cobertura do Sinal Wi-Fi (b) Disponibilidade do Sinal Wi-Fi com 1 Receptor (c) Disponibilidade do Sinal Wi-Fi com 2 Receptores.

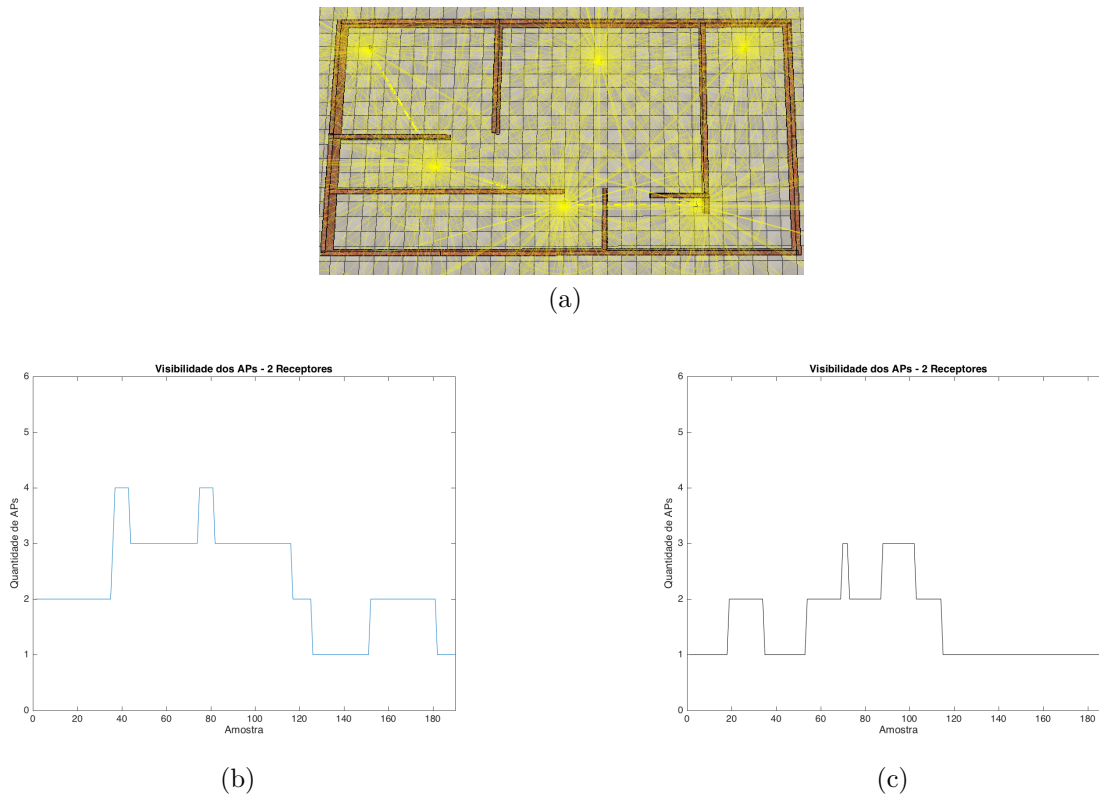


Figura 23 – Cenário com 6 APs: (a) Cobertura do Sinal Wi-Fi (b) Disponibilidade do Sinal Wi-Fi com 1 Receptor (c) Disponibilidade do Sinal Wi-Fi com 2 Receptores.

A Figura 22 mostra a cobertura e visibilidade do sinal Wi-Fi com 4 APs. No SLAM 1R 46% das amostras teve visualização de 2 APs e 54% das amostras teve visualização de 1 AP. No SLAM 2R 14% das amostras teve visualização de 2 AP, 76% das amostras teve visualização de 1 AP e 10% das amostras não teve visualização de APs. A Tabela 2 mostra os valores de visibilidade de cada AP.

Tabela 2 – Porcentagem de Visibilidade por AP - 4 APs

	SLAM - 1 Receptor	SLAM - 2 Receptores
AP1	15%	17%
AP2	32%	31%
AP3	14%	12%
AP4	38%	40%

A Figura 23 mostra a cobertura e visibilidade do sinal Wi-Fi com 6 APs. No SLAM 1R 7% das amostras teve visualização de 4 APs, 35% das amostras teve visualização de 3 APs, 39% das amostras teve visualização de 2 APs e 18% das amostras teve visualização de 1 AP. No SLAM 2R 9% das amostras teve visualização de 3 AP, 31% das amostras teve visualização de 2 AP e 59% das amostras não teve visualização de APs. A Tabela 3 mostra os valores de visibilidade de cada AP.

Tabela 3 – Porcentagem de Visibilidade por AP - 6 APs

	SLAM - 1 Receptor	SLAM - 2 Receptores
AP1	10%	12%
AP2	20%	21%
AP3	9%	8%
AP4	24%	28%
AP5	18%	19%
AP6	18%	12%

4.1 Cena A: Validação da Solução

Para validar a modelagem, nenhum tipo de ruído foi adicionado ao sistema. A Figura 24 mostra a posição final dos marcos para o SLAM 1R e SLAM 2R. Nela é possível verificar que o resultado dos dois sistemas foi bem próximo ao real, comprovando a eficácia da modelagem dos dois sistemas. Entretanto mesmo sem ruído os sistemas não foi possível obter uma distância euclidiana nula dos marcos. Isso se deve pela complexidade do problema, visto que não há informação angular, existe mais de uma hipótese ao computar a posição do marco.

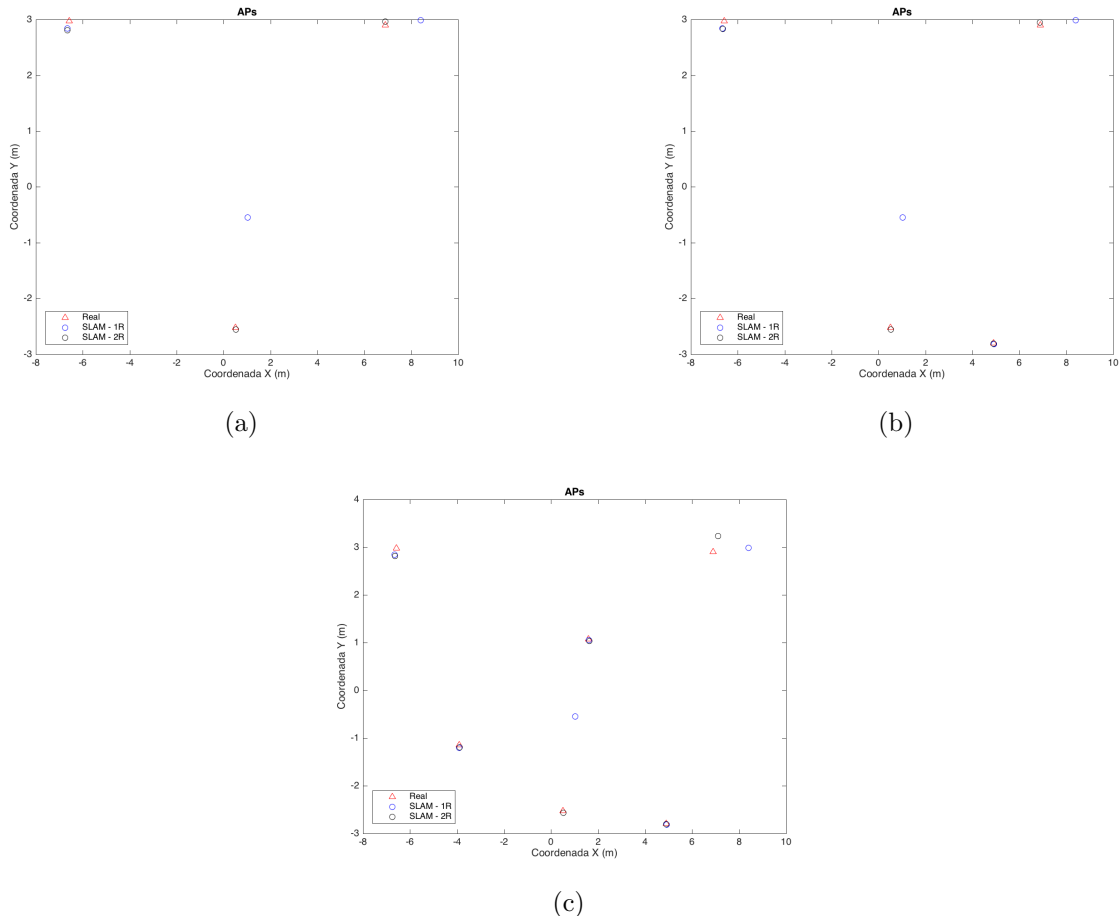


Figura 24 – Resultado da Posição Final dos APs: (a) 3APs; (b) 4APs; (c) 6APs - Cena A.

Ao comparar o SLAM com 1 receptor e o SLAM com 2 receptores, é perceptível a diferença ao observar a posição final dos APs, pois na abordagem com 2 receptores ocorre o dobro das atualizações da abordagem com 1 receptor. Além disso, devido aos receptores se apresentarem em posições diferentes, há uma melhor precisão dos resultados.

Os resultados das variações de APs mostram que mesmo diminuindo a quantidade de APs, tanto a posição do robô quanto a dos marcos mantêm-se constantes, viabilizando o funcionamento do sistema até mesmo em trechos onde não há visibilidade de APs como ocorre no cenário que contém apenas 3 APs.

4.2 Cena B: Influência do Ruído da Fase de Predição

Nesta cena o sinal Wi-Fi continua sem ruído, mas diferentemente da cena A, nesta cena o modelo de odometria possui ruído para que seja possível verificar sua influência na fase de predição. Portanto, o sistema deve ser capaz de corrigir os erros ocasionados pela odometria na etapa de atualização.

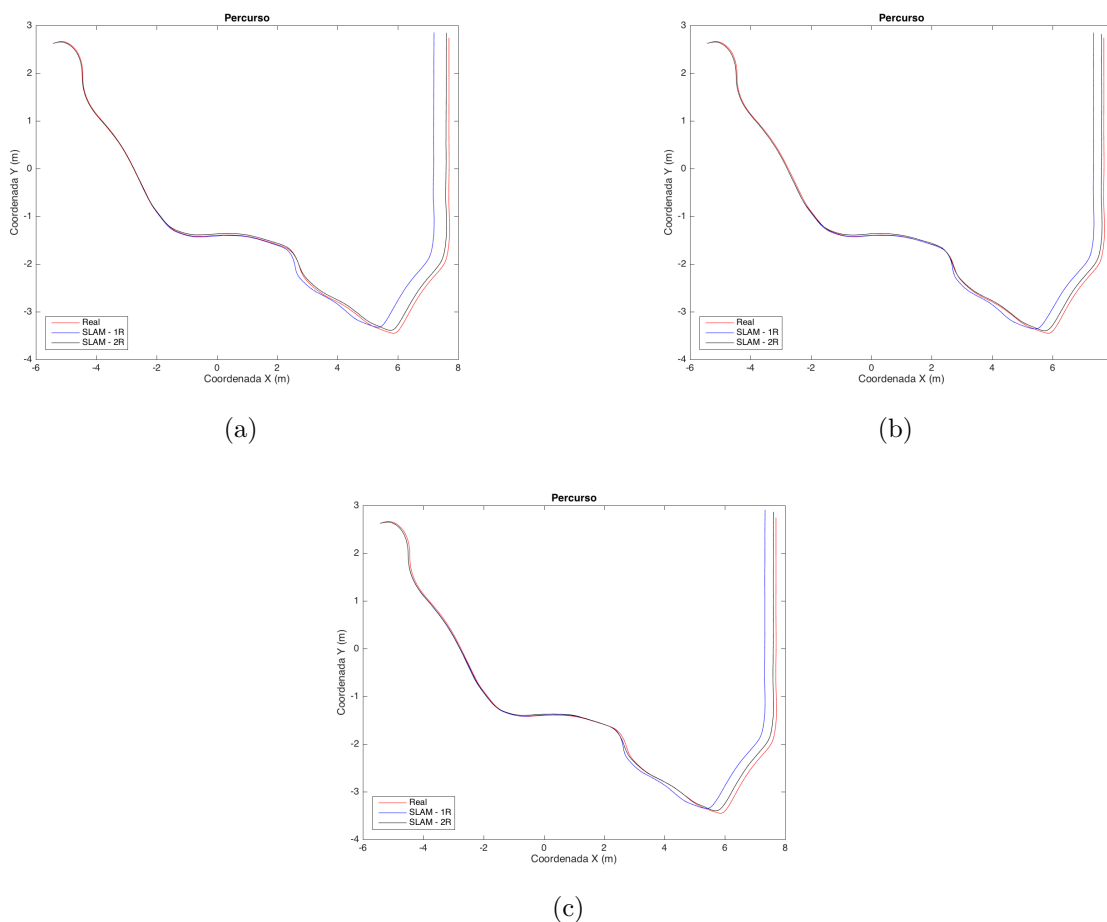


Figura 25 – Trajetória do Robô nas Coordenadas x e y: (a) 3 APs; (b) 4 APs; (c) 6 APs - Cena B.

A Figura 25 mostra um exemplo da estimativa do percurso realizado pelo robô calculado pela: odometria, SLAM 1R e SLAM 2R na Cena B com 3, 4 e 6 APs. Nela é possível observar a odometria se tornando mais distante a cada amostra devido ao ruído acumulado ao longo do tempo. Como a odometria faz parte da etapa de predição do SLAM, os dois sistemas são influenciados. Apenas a orientação do robô segue inalterada em virtude do uso da bússola nos sistemas SLAM.

Nesta cena a confiança do sistema está voltada para a etapa de atualização, ou seja, para a informação dos sinais Wi-Fi. Portanto, a precisão do percurso está diretamente relacionada a precisão dos marcos. Como o SLAM com 2 receptores possui uma boa precisão dos marcos, quanto mais marcos visíveis no sistema melhor será a trajetória percorrida. Já no SLAM com 1 receptor os valores de trajetória tornaram-se inconsistentes devido ao alto valor da variância.

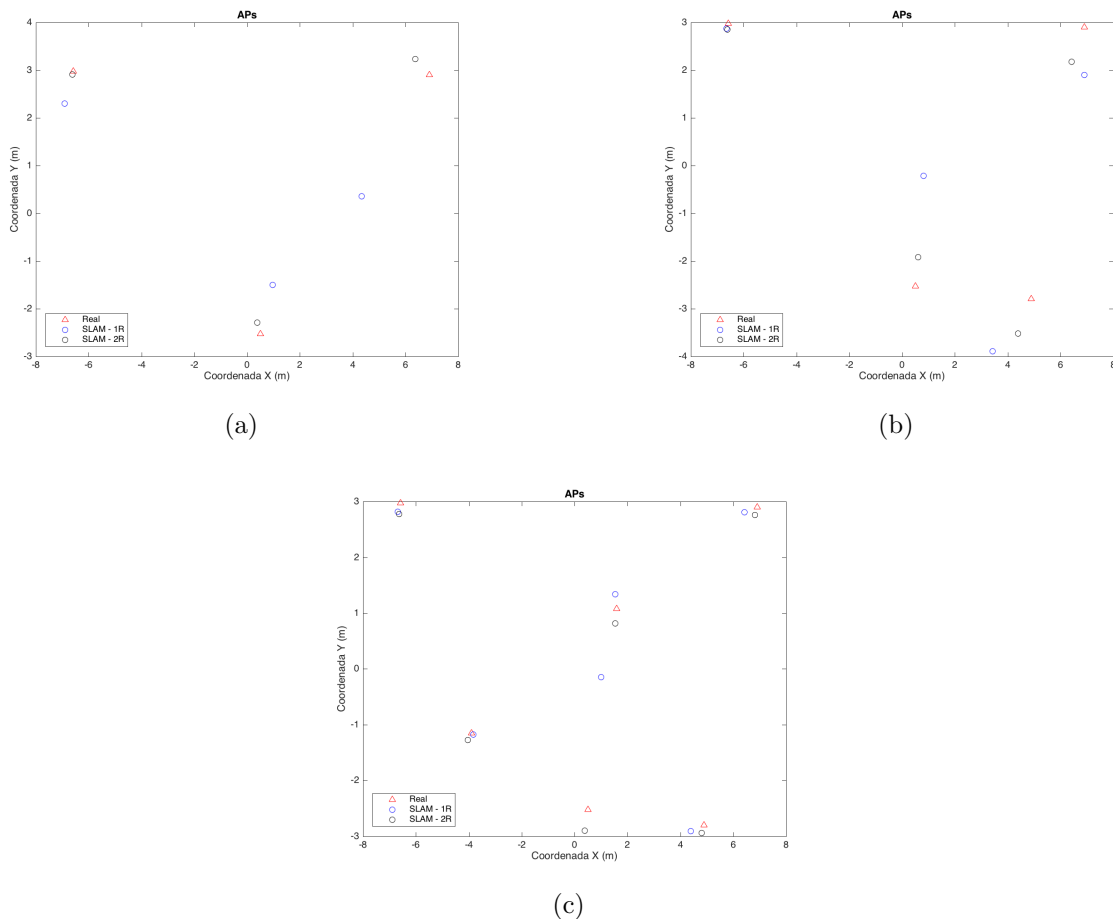


Figura 26 – Resultado da Posição Final dos APs: (a) 3APs; (b) 4APs; (c) 6APs - Cena B.

Com relação aos marcos, a Figura 26 mostra a posição final dos APs no SLAM 1R e SLAM 2R na cena B com 3, 4 e 6 APs. Nela é possível notar que houve influência da odometria para computar a posição dos APs mesmo em um sistema sem ruídos no sinal, pois os marcos se encontram em posições mais distantes do real comparado à cena A.

Ao comparar o SLAM com 1R e o SLAM com 2R, é possível visualizar de maneira mais clara a diferença das abordagens ao observar a posição final da trajetória do robô, visto que o ruído influencia diretamente a estimativa do percurso. O sistema com dois receptores realiza mais atualizações corrigindo a trajetória de maneira mais eficiente por obter informações com receptores em posições diferentes.

4.3 Cena C: Influência do Ruído da Fase de Atualização

Para verificar a influência do ruído na fase de atualização, foi aplicado ruído apenas no sinal Wi-Fi. Nesta cena, diferentemente da cena B o modelo da odometria não possui ruído, o sistema deve ser capaz de localizar os marcos e o robô mesmo com o ruído na etapa de atualização. A Figura 27 mostra um exemplo da estimativa do percurso realizado pelo robô calculado pela: odometria, SLAM 1R e SLAM 2R na Cena C com 3, 4 e 6 APs.

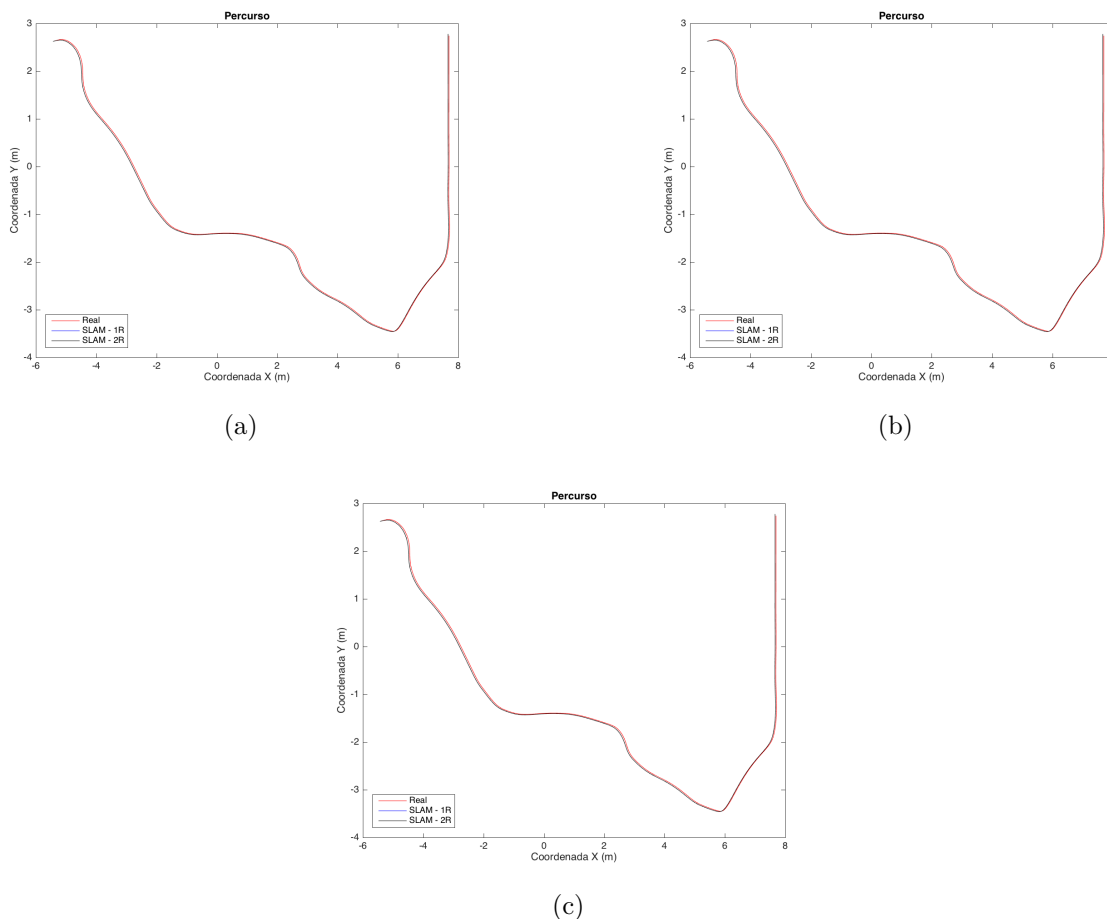


Figura 27 – Trajetória do Robô nas Coordenadas x e y: (a) 3 APs; (b) 4 APs; (c) 6 APs - Cena C.

Com relação ao percurso foram obtidos resultados próximos à trajetória real do robô para os dois sistemas SLAM em todas as variações de APs. Isso se deve a falta de ruído no modelo de predição do robô. O sistema não realiza atualizações que modifiquem o percurso do robô caso a incerteza da informação recebida na atualização seja de um grau muito elevado comparado ao modelo preditivo.

Com relação aos marcos, temos que a Figura 28 mostra um exemplo do erro e atraso de inicialização dos marcos para o SLAM 1R e SLAM 2R na cena C com 3, 4 e 6 APs. Nela é possível observar que o SLAM 2R obteve resultados mais precisos de inicialização dos marcos, entretanto por conta da filtragem do sinal o SLAM 2R possui um atraso na inicialização superior ao SLAM 1R.

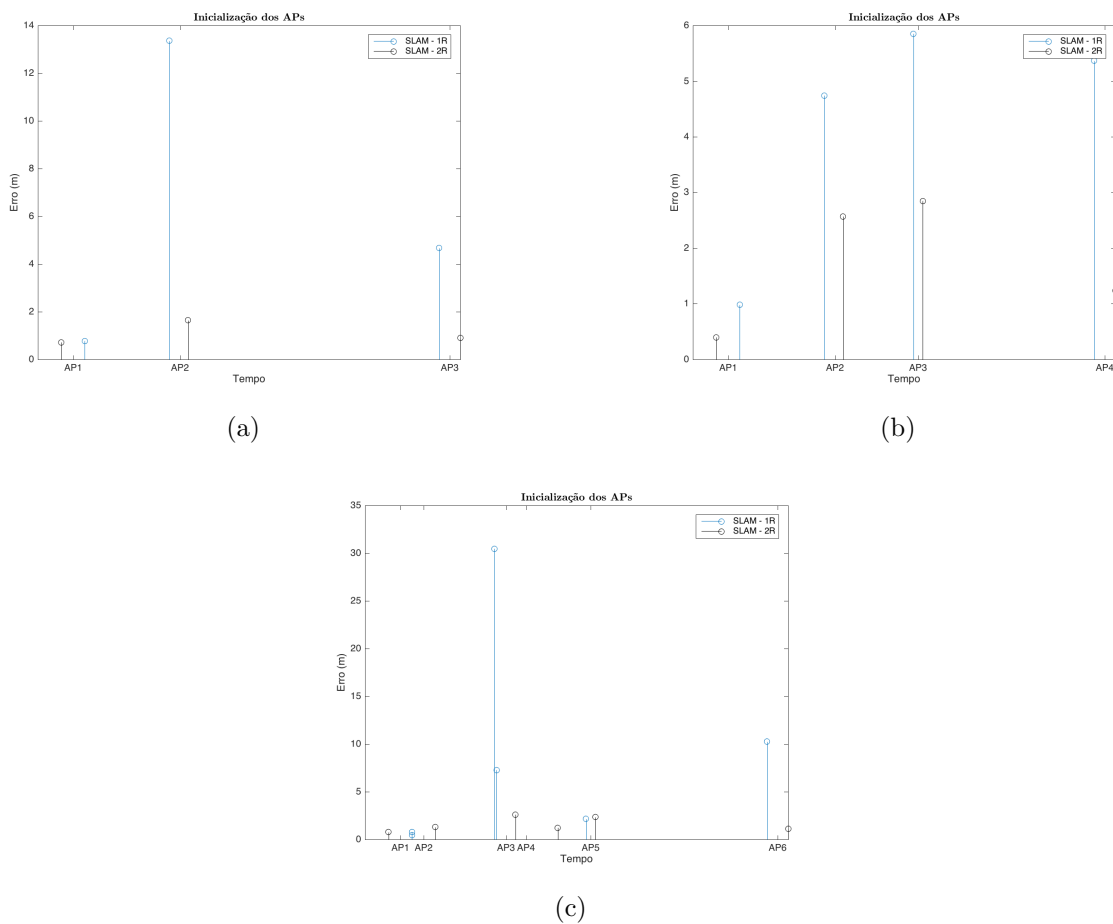


Figura 28 – Resultado do Erro da Estimativa Inicial: (a) 3 APs; (b) 4 APs; (c) 6 APs - Cena C.

Ainda com relação aos marcos, temos a Figura 29 que mostra a posição final dos APs no SLAM 1R e SLAM 2R na cena C com 3, 4 e 6 APs. Nela é possível observar que houve grande influência do ruído no sinal Wi-Fi para computar a posição dos APs mesmo em um sistema sem ruído na odometria. Ao compararmos os dois sistemas é possível observar que o SLAM 1R obteve distâncias bem mais elevadas do que nas cenas A e B.

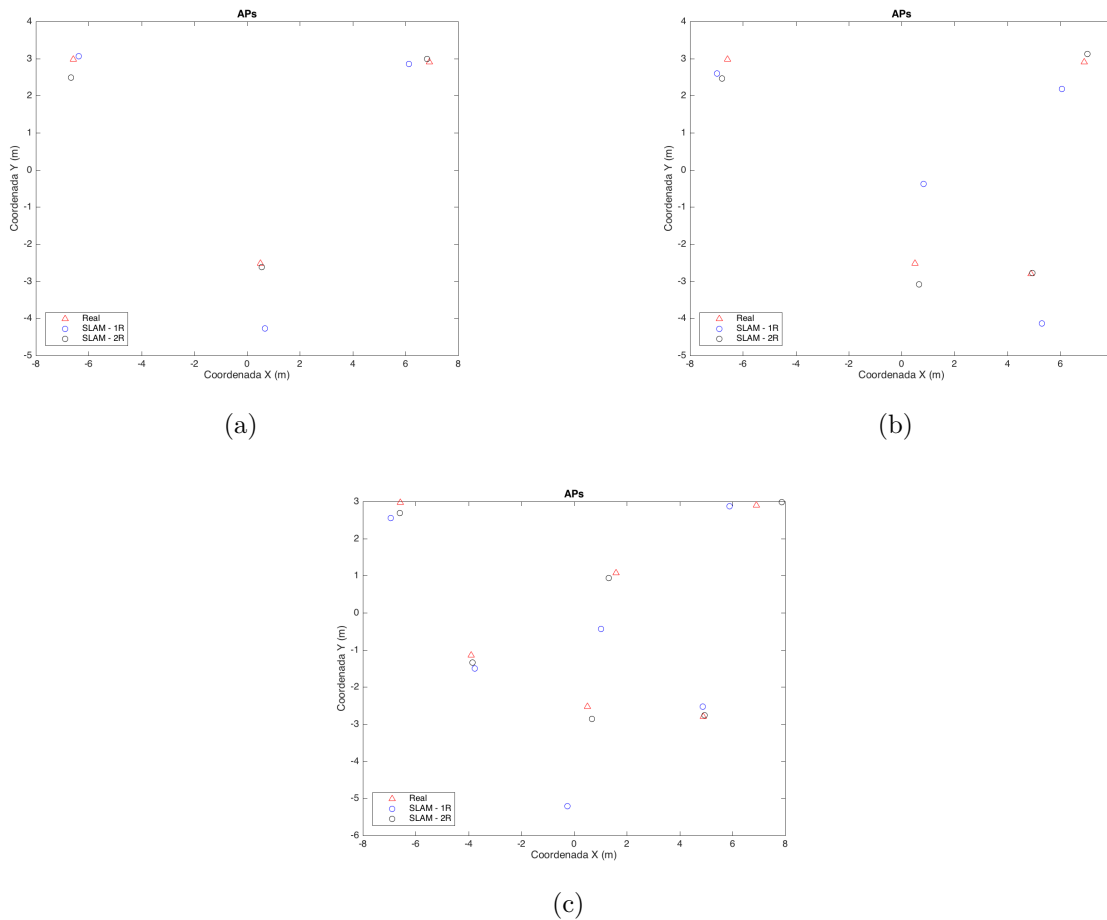


Figura 29 – Resultado da Posição Final dos APs: (a) 3 APs; (b) 4 APs; (c) 6 APs - Cena C.

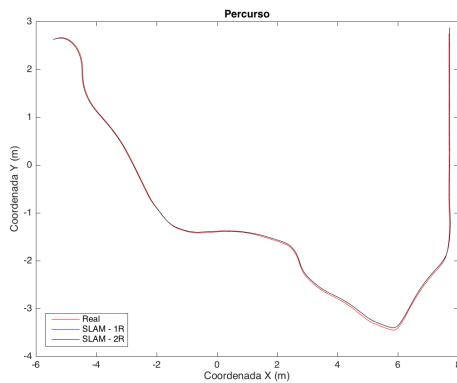
4.4 Cena D: Cenário Completo

A cena D retrata o cenário mais próximo do real, onde diferentemente das cenas anteriores, o ruído é aplicado tanto na fase de predição quanto na fase de atualização, ou seja, tanto no modelo de odometria quanto no sinal Wi-Fi. Desse modo, é possível verificar o funcionamento do sistema em condições que simulam a realidade.

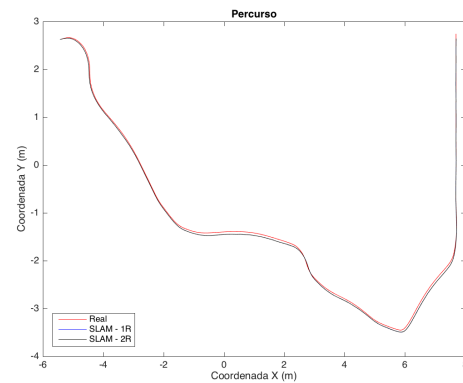
Para avaliar a localização do robô, foi calculado o erro da posição final do robô computada pela odometria, pelo SLAM com um receptor e pelo SLAM com dois receptores ao final da trajetória percorrida dez vezes. A Tabela 4 mostra a média e variância do erro médio quadrático calculado sobre a coordenada x, y e pela orientação θ do robô na cena D. Foram obtidos resultados decrescentes entre a casa de 10^{-1} metros a 10^{-2} conforme a diminuição dos APs. Isso se deve por conta do balanço de incertezas dos sensores da fase de predição e atualização. Mesmo que só haja informações com ruído, o FKE balanceia as incertezas de modo a sempre determinar a melhor estimativa dada as informações de predição e atualização. A Figura 30 mostra um exemplo da estimativa do percurso realizado pelo robô calculado pela: odometria, SLAM 1R e SLAM 2R com 3, 4 e 6 APs.

Tabela 4 – Erro da Posição Final do Robô com Variação de APs - Cena D

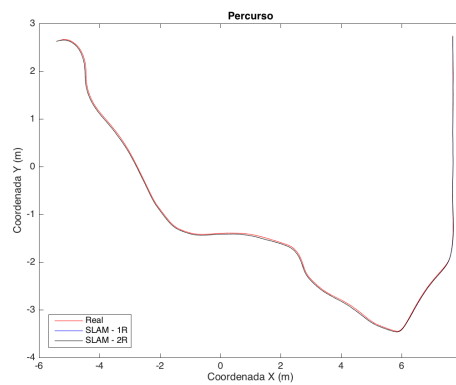
		x (m)		y (m)		θ (rad)	
		μ	σ	μ	σ	μ	σ
3 APs	Odometria	2,430	1,928	3,667	8,439	0,636	0,090
	SLAM - 1R	0,084	0,004	0,061	0,001	0,001	0,000
	SLAM - 2R	0,084	0,004	0,062	0,001	0,001	0,000
4 APs	Odometria	3,644	0,635	4,467	11,080	0,858	0,145
	SLAM - 1R	0,088	0,002	0,093	0,008	0,001	0,000
	SLAM - 2R	0,088	0,003	0,094	0,008	0,001	0,000
6 APs	Odometria	2,122	4,095	5,483	5,719	0,657	0,311
	SLAM - 1R	0,120	0,007	0,097	0,005	0,001	0,000
	SLAM - 2R	0,116	0,007	0,093	0,004	0,001	0,000



(a)



(b)



(c)

Figura 30 – Trajetória do Robô nas Coordenadas x e y: (a) 3 APs; (b) 4 APs; (c) 6 APs - Cena D.

Com relação aos marcos, temos que a Figura 31 mostra um exemplo do erro e atraso de inicialização dos marcos para o SLAM com um receptor e SLAM com dois receptores na cena D com 3, 4 e 6 APs. Nela podemos observar que os resultados de inicialização do SLAM 2R encontram-se nitidamente mais precisos que o SLAM 1R e mais discrepantes do que os resultados mostrados na cena C, pois há mais ruído nesta cena.

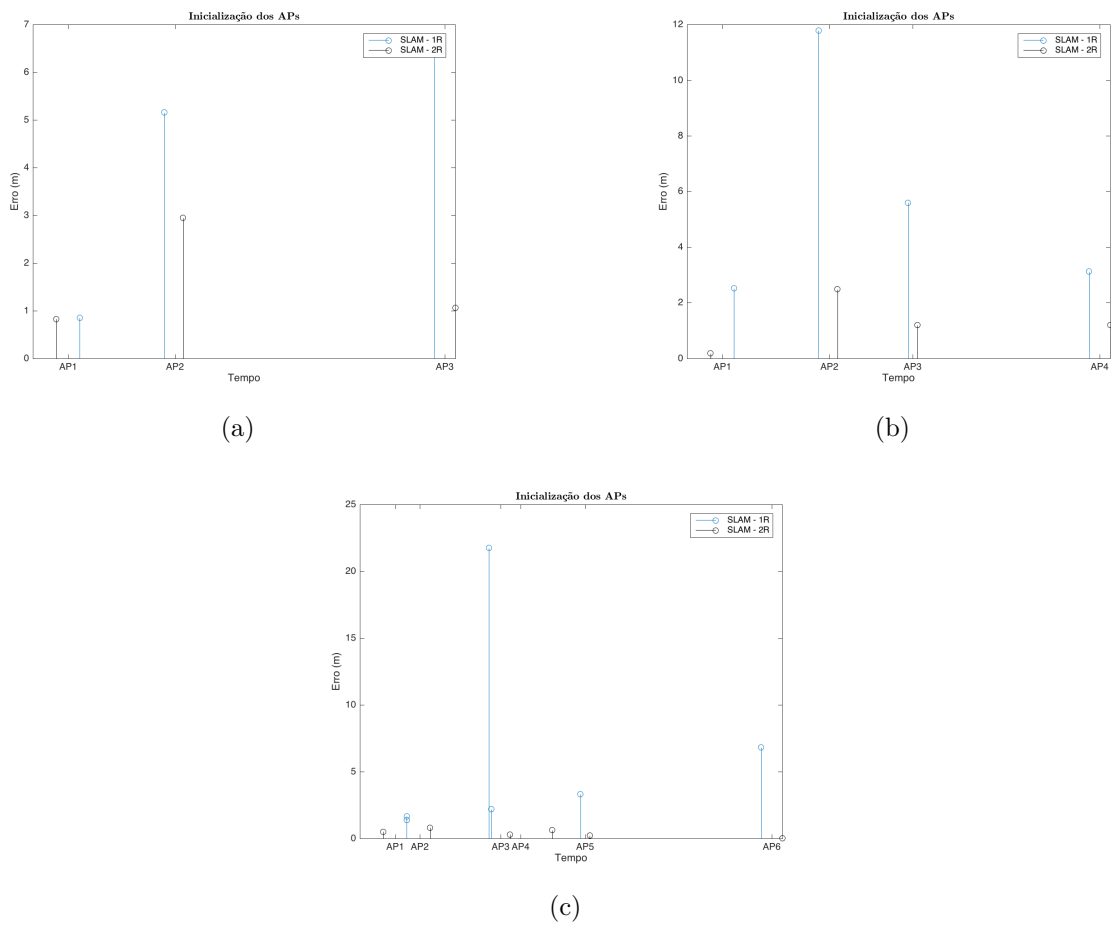


Figura 31 – Resultado do Erro da Estimativa Inicial: (a) 3 APs; (b) 4 APs; (c) 6 APs - Cena D.

A Figura 32 ilustra o comportamento de atualização da posição dos marcos. Nela é possível observar a diferença de comportamento de atualização do marco entre o SLAM 1R e SLAM 2R, onde os ruídos afetam desde a inicialização até a posição final do marco.

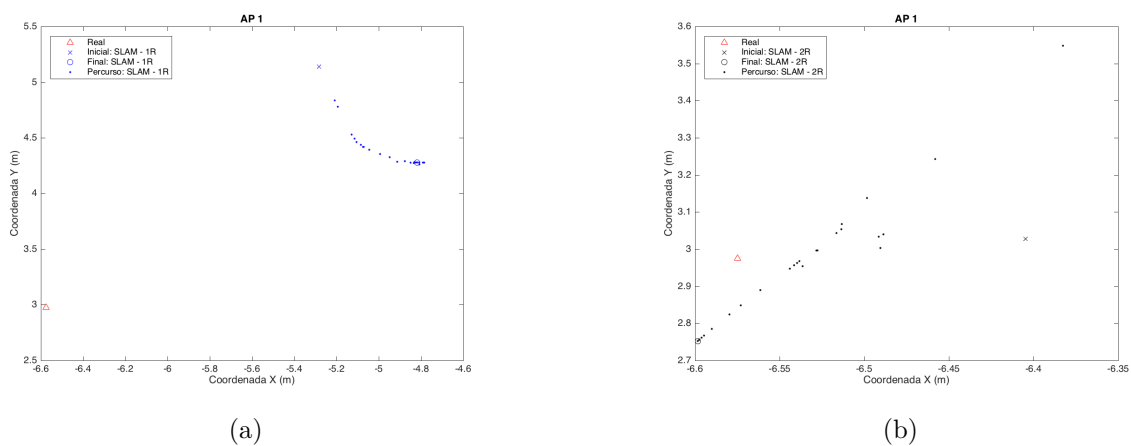


Figura 32 – Trajetória do AP 1 na cena D com 3 APs: (a) SLAM 1R; (b) SLAM 2R.

Com relação à posição final dos APs, é possível observar a influencia dos ruídos ao computar a posição dos APs. Entretanto, mesmo com ruído, tanto na predição quanto na atualização os resultados não diferiram de maneira drástica dos cenários B e C que apresentam estes ruídos de maneira separada. Isso pode ser visto com mais clareza na Tabela 5, onde foi calculada a a média e variância da distância euclidiana da posição final do marco computada pela odometria, pelo SLAM 1R e pelo SLAM 2R com a posição real do marco ao final da trajetória percorrida dez vezes pelo robô. A Figura 33 mostra um exemplo da posição final dos APs no SLAM 1R e SLAM 2R na com 3, 4 e 6 APs.

Tabela 5 – Erro de Posição Final do Marco com Variação de APs - Cena D

			AP1	AP2	AP3	AP4	AP5	AP6
3 Aps	SLAM - 1R	μ (m)	0,812	1,562	1,709			
		σ (m)	0,090	0,751	0,650			
	SLAM - 2R	μ (m)	0,305	1,353	0,547			
		σ (m)	0,047	1,652	0,079			
4 Aps	SLAM - 1R	μ (m)	1,087	1,659	1,537	0,644		
		σ (m)	0,484	0,160	0,606	0,097		
	SLAM - 2R	μ (m)	0,446	0,784	0,479	0,418		
		σ (m)	0,077	0,873	0,084	0,132		
6 Aps	SLAM - 1R	μ (m)	1,023	1,363	1,664	0,732	1,141	3,571
		σ (m)	0,534	0,663	0,536	0,049	2,095	9,591
	SLAM - 2R	μ (m)	0,349	0,547	0,422	0,461	0,174	0,812
		σ (m)	0,023	0,522	0,124	0,148	0,004	2,049

Em um sistema SLAM, o ruído da fase de atualização não se resume apenas ao ruído do sensor, mas sim a um conjunto da incerteza da posição do marco com o ruído atribuído ao sensor que tenta determinar a posição do marco. Ao visualizar um marco, o sistema tenta determinar a sua posição e ao mesmo tempo corrigir a posição do robô. Entretanto, quanto maior o ruído da fase de atualização, menor e menos precisa será a correção. Neste mapa há um curto espaço de visualização dos APs como pode ser visto nas Figuras 21, 22 e 23, e o robô não torna a vê-los. Isso faz com que a diminuição da quantidade de APs torne o sistema mais efetivo neste mapa.

Ao comparar o SLAM com 1 receptor e o SLAM com 2 receptores, é possível observar que mesmo com ruído no sinal Wi-Fi e na odometria, a distância euclidiana dos marcos no receptor 2 mantém-se na casa dos 10^{-1} metros. Já com relação ao trajeto percorrido, os dois sistemas mantém valores aproximados devido à baixa quantidade de visualizações por cada AP. Neste cenário o curto percurso do robô não foi capaz de demonstrar de maneira nítida a diferença entre os dois sistemas. Além disso, neste cenário não há re-amostragem, ou seja, o robô não visualiza o marco após deixar de visualiza-lo.

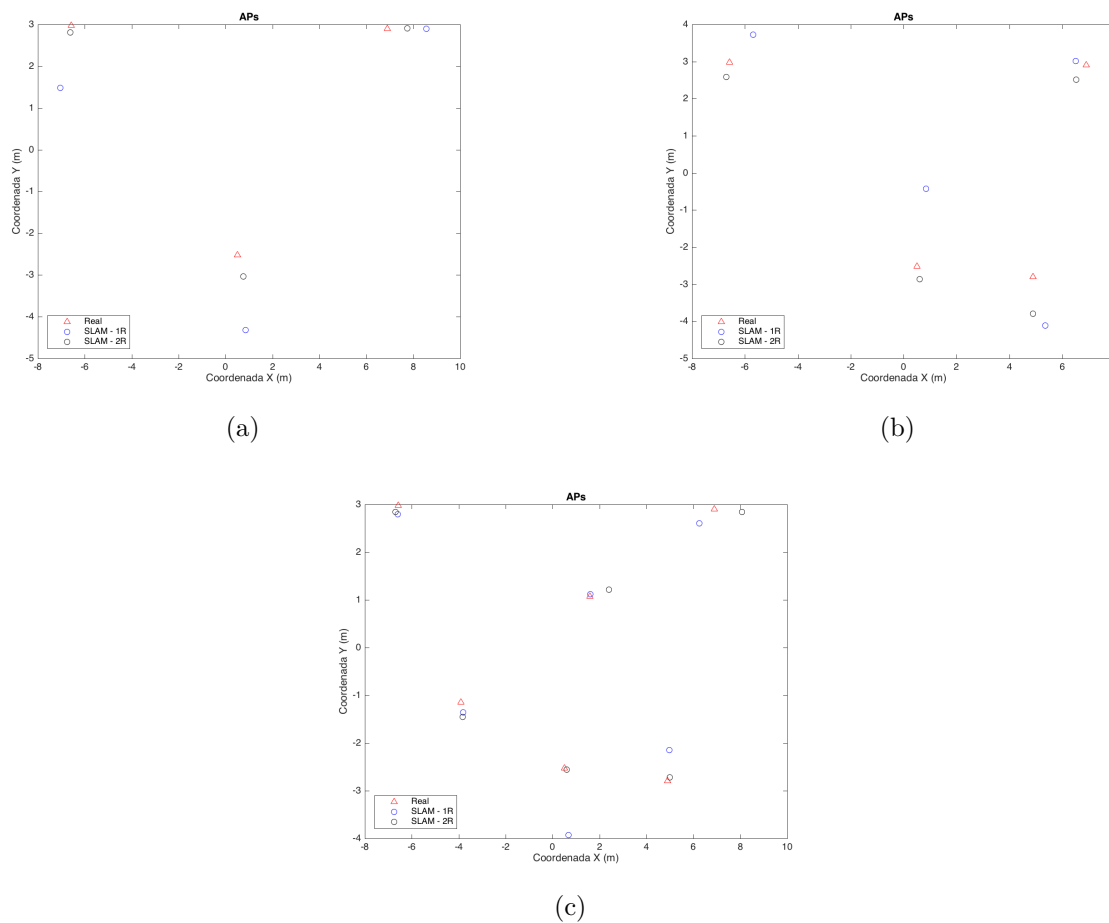


Figura 33 – Resultado da Posição Final dos APs: (a) 3APs; (b) 4APs; (c) 6APs - Cena D.

4.5 Cena E: Cenário com Loop

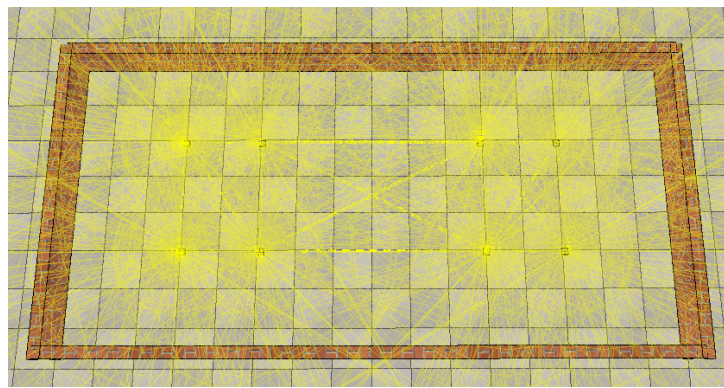
Para a cena E, é possível visualizar o mesmo robô com ruídos tanto na fase de predição quanto na atualização. A trajetória do robô é feita de maneira circular para que o mesmo reveja os APs a cada volta dada. Neste cenário, há um novo mapa de 8 metros de comprimento e 4 metros de largura, como pode ser visto na Figura 34.

Tabela 6 – Porcentagem de Visibilidade por AP - 8 APs

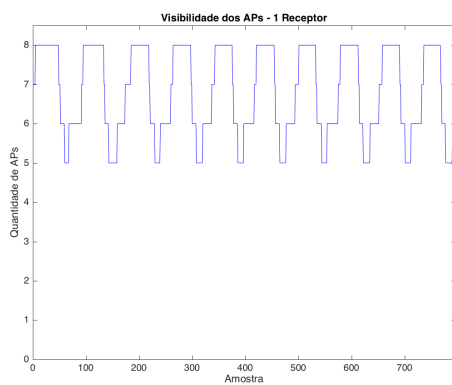
	SLAM - 1 Receptor	SLAM - 2 Receptores
AP1	15%	14%
AP2	14%	14%
AP3	13%	13%
AP4	15%	13%
AP5	11%	12%
AP6	11%	12%
AP7	11%	11%
AP8	11%	11%



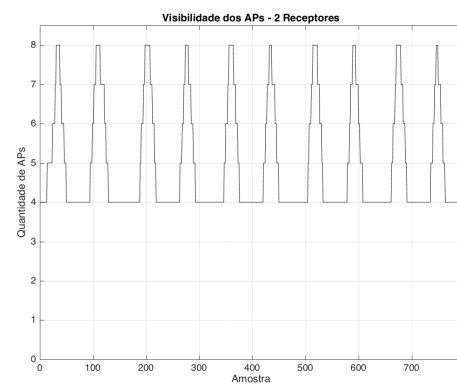
Figura 34 – Mapa do Cenário com Loop.



(a)



(b)



(c)

Figura 35 – Cenário com 8 APs: (a) Cobertura do Sinal Wi-Fi (b) Grafo de Visibilidade do Sinal Wi-Fi com 1 Receptor (c) Grafo de Visibilidade do Sinal Wi-Fi com 2 Receptores.

A Figura 35 ilustra a cobertura e visibilidade do sinal Wi-Fi com 8 APs. No SLAM 1R 44% das amostras teve visualização de 8 APs, 9% das amostras teve visualização de 7 APs, 32% das amostras teve visualização de 6 APs e 15% das amostras teve visualização de 5 AP. No SLAM 2R 9% das amostras teve visualização de 8 APs, 10% das amostras teve visualização de 7 APs, 9% das amostras teve visualização de 6 APs, 10% das amostras teve visualização de 5 AP e 62% das amostras teve visualização de 4 AP. A Tabela 6 mostra os valores de visibilidade individual de cada AP.

A cena E proporciona a re-observação dos marcos, além de realizar um percurso cinco vezes maior do que os cenários A, B, C e D para que haja uma noção mais nítida de diferença entre o SLAM 1R e 2 receptores, visto que o cenário D apresentou resultados próximos para ambos. Neste cenário foi analisado o comportamento do sistema em: 1 volta, 3 voltas e 5 voltas.

4.5.1 Cenário E com 1 Volta

Para avaliar a localização do robô, foi calculado o erro da posição final do robô computada pela odometria, pelo SLAM 1R e SLAM 2R ao final da trajetória percorrida pelo robô após a 1 volta no percurso. A Tabela 7 mostra o erro médio quadrático calculado sobre a coordenada x, y e pela orientação θ do robô.

Tabela 7 – Erro Médio Quadrático ao Final da Trajetória do Robô após 1 Volta.

	x	y	θ
Odometria	2,364	3,017	2,032
SLAM - 1R	0,066	0,019	0,001
SLAM - 2R	0,062	0,019	0,001

Tabela 8 – Distância Euclidiana Final dos APs após 1 Volta.

	SLAM - 1 Receptor	SLAM - 2 Receptores
AP1	0,098	0,023
AP2	0,273	0,084
AP3	0,240	0,127
AP4	0,267	0,071
AP5	0,128	0,181
AP6	0,495	0,132
AP7	1,237	0,029
AP8	1,361	0,388

A Figura 77 mostra um exemplo de percurso realizado pelo robô na cena E após 1 volta. Tanto as Figuras do trajeto do robô quando a tabela e erro da trajetória mostram que a cena E após uma volta se assemelha ao cenário D, após uma volta a cena E possui apenas as mesmas características da cena E, portanto apresenta o mesmo comportamento.

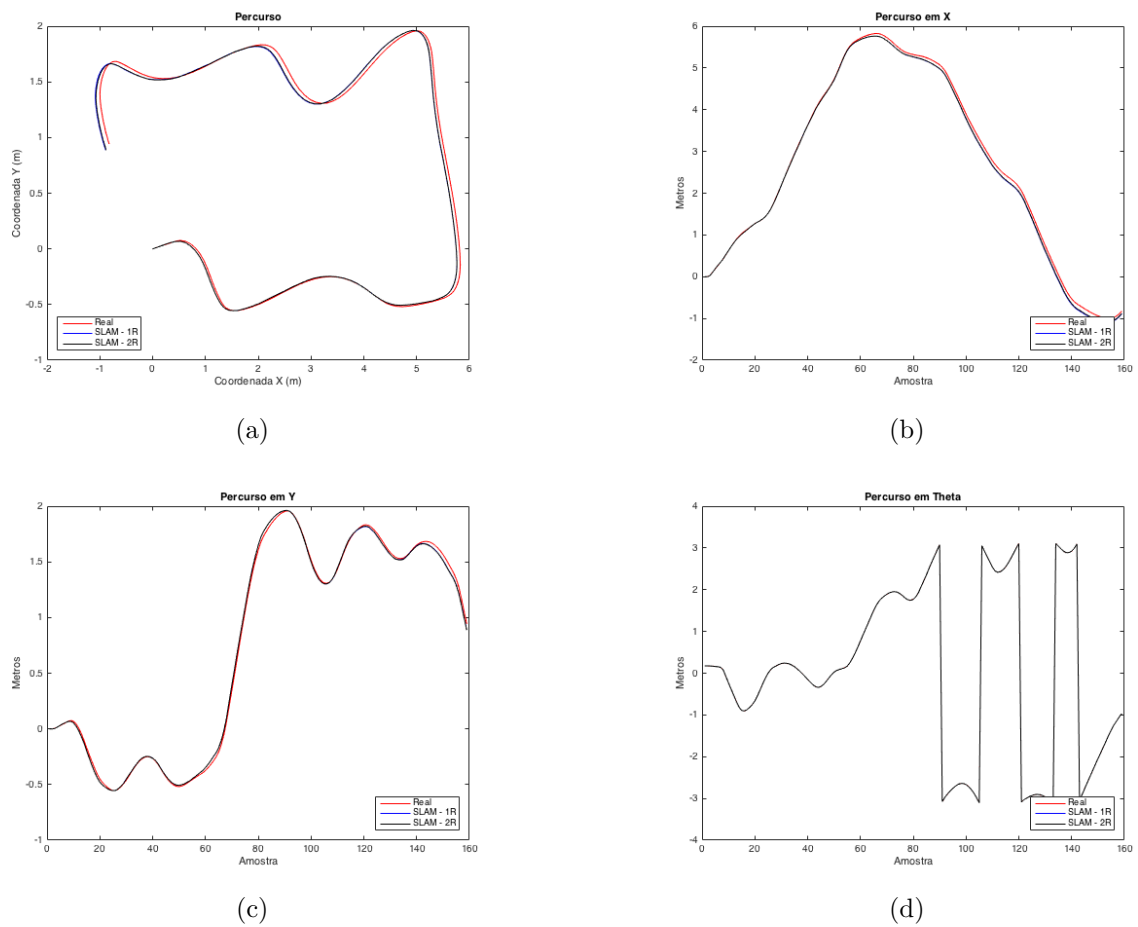


Figura 36 – Percurso do Robô com 8 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário E.

A Figura 37 mostra o comportamento da variância das coordenadas x e y para o sistema SLAM 1R (Figura 37a) e para o sistema SLAM 2R (37b). Durante a primeira volta é visto que há um crescimento constante da variância para os dois sistemas e para as duas coordenadas, com exceção de alguns períodos em que a variância se mantém constante. Estes períodos correspondem aos instantes em que o robô não está variando sua posição na determinada coordenada.

Para avaliar a localização dos marcos, foi calculada a distância euclidiana da posição final do marco computada pela odometria, pelo SLAM 1R e pelo SLAM 2R com a posição real do marco ao final da trajetória percorrida pelo robô. A Tabela 8 mostra a distância euclidiana calculado para cada AP na cena E.

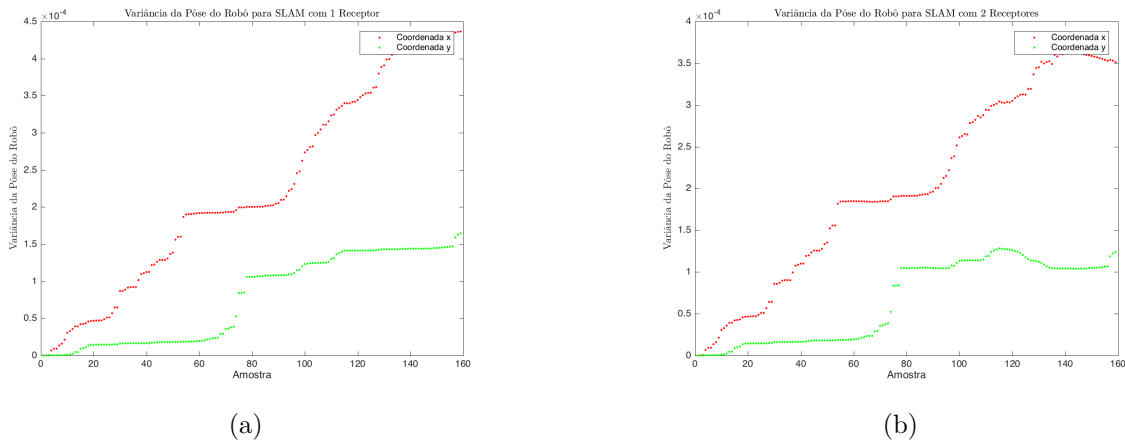


Figura 37 – Gráfico da Variância da Posição do Robô após 1 volta nas coordenadas x e y: (a) SLAM 1R; (b) SLAM 2R.

A Figura 38 mostra a posição inicial e final dos marcos após 1 volta. Neste cenário, mesmo com um trajeto de uma volta que se assemelha ao trajeto do cenário D, há uma visibilidade dos APs completamente diferente como já mostrado na Figura 35, onde no mínimo possui uma visualização de 4 APs logo há uma constante correção das posições dos APs neste cenário tornando os valores de distância euclidiana bem mais precisos do que os vistos no cenário D.

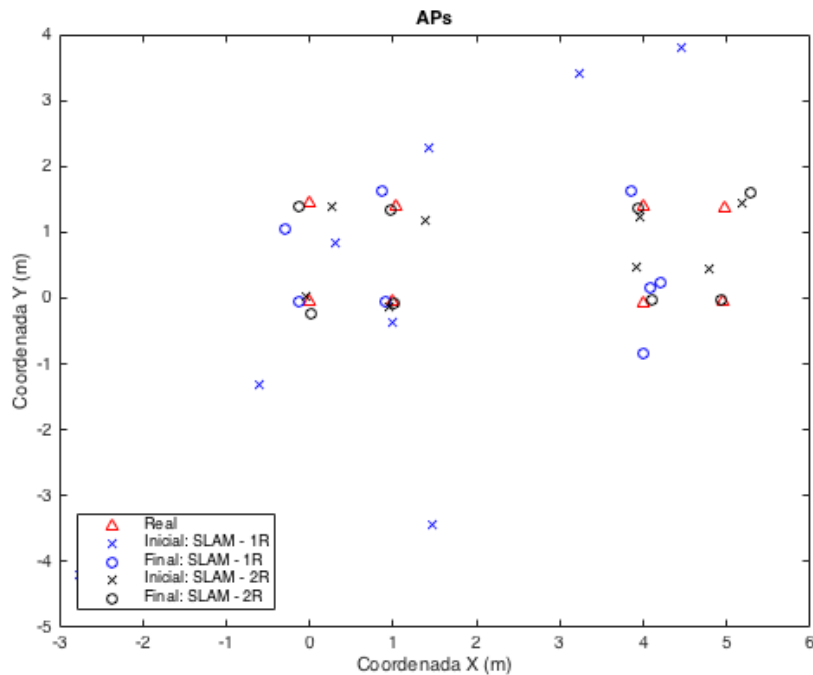


Figura 38 – Resultado da Posição Final dos 8 APs após 1 volta.

4.5.2 Cenário E com 3 Voltas

Para avaliar a localização do robô, foi calculado o erro da posição final do robô computada pela odometria, pelo SLAM 1R e pelo SLAM 2R ao final da trajetória percorrida pelo robô após a 3 voltas no percurso. A Tabela 9 mostra o erro médio quadrático calculado sobre a coordenada x, y e pela orientação θ do robô.

Tabela 9 – Erro Médio Quadrático ao Final da Trajetória do Robô após 3 voltas.

	x	y	θ
Odometria	2,364	3,017	2,032
SLAM - 1R	0,111	0,033	0,001
SLAM - 2R	0,050	0,026	0,001

É possível visualizar na Figura 39 uma divergência no percurso do SLAM 1R e com 2 receptores, onde o SLAM 2R demonstrou mais precisão em relação à duas voltas atrás, enquanto o sistema de SLAM 1R aumentou seu erro em relação a primeira volta.

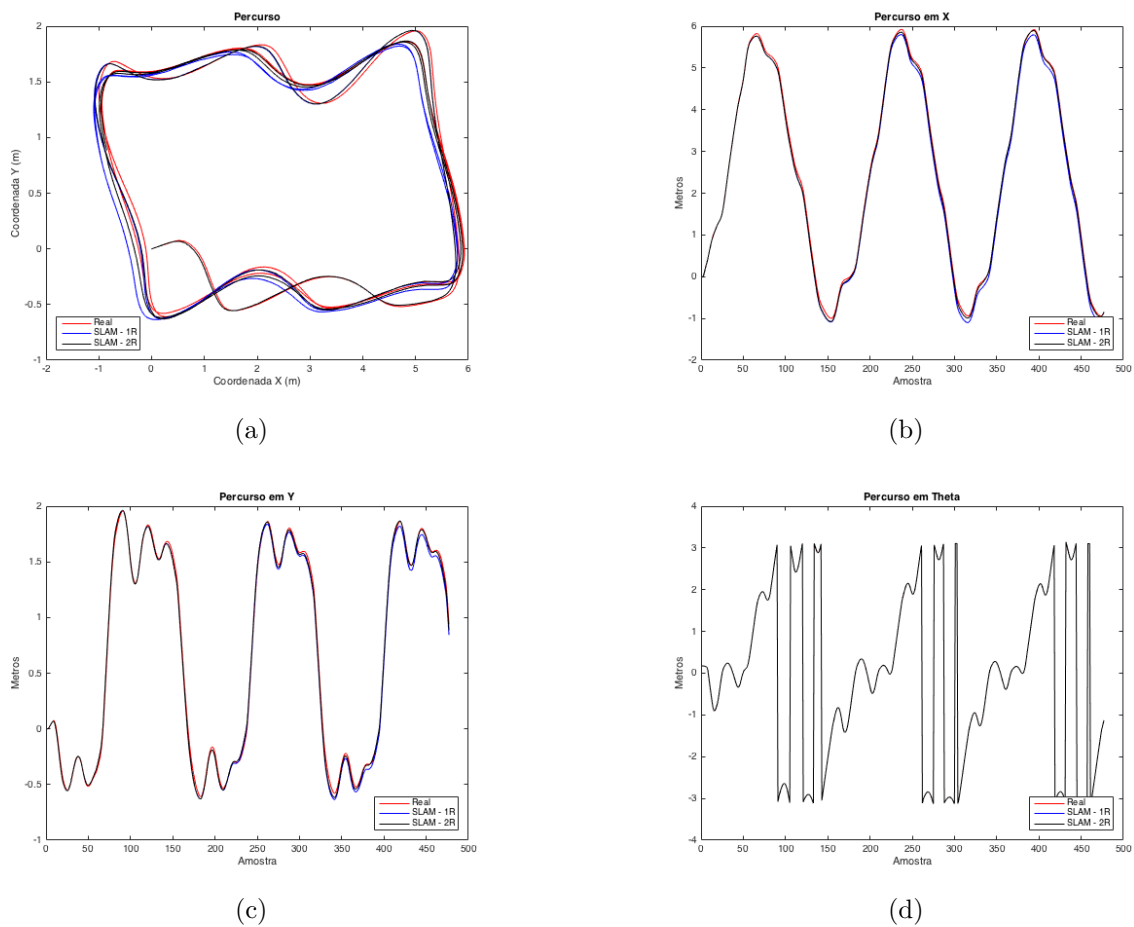


Figura 39 – Percurso do Robô com 4 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário E.

A Figura 40 mostra o comportamento da variância das coordenadas x e y para o sistema SLAM 1R (Figura 40a) e para o sistema SLAM 2R (40b). Após 3 voltas é possível

observar que há três quedas drásticas na variância do SLAM 2R. Estas quedas ocorrem quando o robô completa uma volta completa pelo mapa e re-observa os marcros. O mesmo não ocorre no SLAM 1R pois ele acredita que se encontra em um local diferente a cada volta.

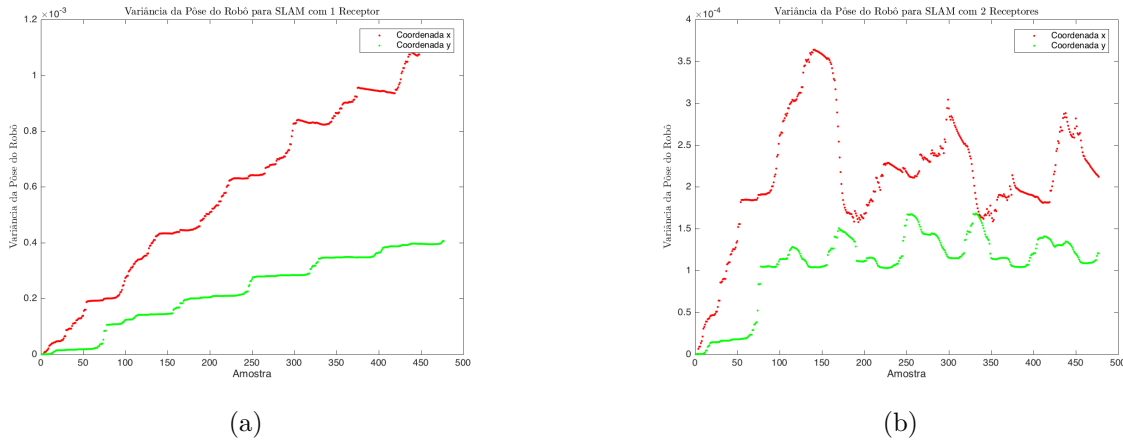


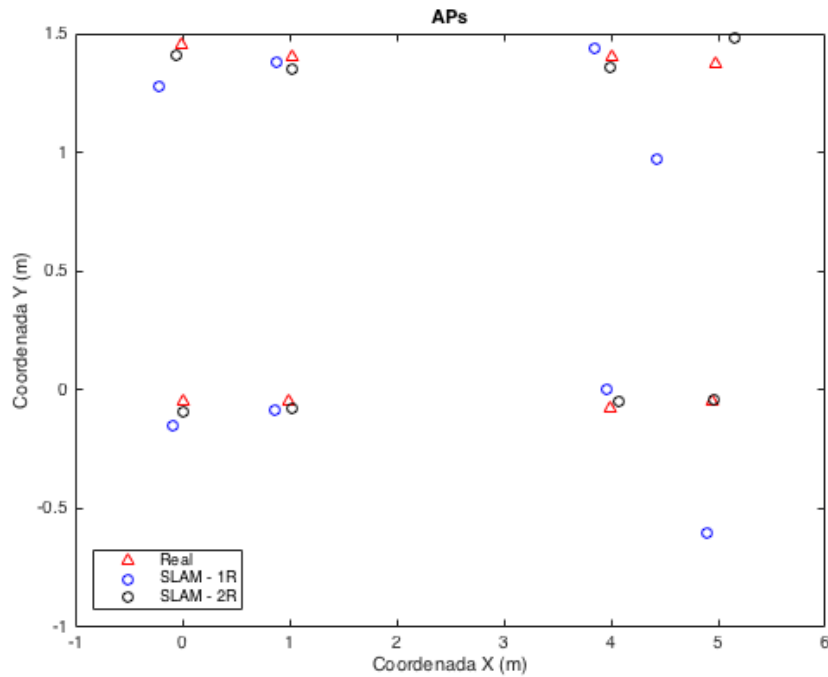
Figura 40 – Gráfico da Variância da Posição do Robô após 3 voltas nas coordenadas x e y: (a) SLAM 1R; (b) SLAM 2R.

Para avaliar a localização dos marcros, foi calculada a distância euclidiana da posição final do marco computada pela odometria, pelo SLAM 1R e pelo SLAM 2R com a posição real do marco ao final da trajetória percorrida pelo robô. A Tabela 10 mostra a distância euclidiana calculado para cada AP na cena E.

Tabela 10 – Distância Euclidiana Final dos APs após 3 voltas.

	SLAM - 1 Receptor	SLAM - 2 Receptores
AP1	0,139	0,033
AP2	0,139	0,052
AP3	0,085	0,077
AP4	0,157	0,042
AP5	0,131	0,040
AP6	0,282	0,063
AP7	0,554	0,016
AP8	0,684	0,212

A Figura 41 mostra a posição final dos marcros após 3 voltas. É possível visualizar que apenas as estimativas dos APs 7 e 8 encontram-se visivelmente distantes da posição real. Os APs 7 e 8 são os marcros mais distantes do robô quando ocorre se inicia o percurso, e são os marcros menos visíveis por todo o percurso, portanto é de se esperar que possuem os resultados inferiores aos demais marcros.



(a)

Figura 41 – Resultado da Posição Final dos 8 APs após 3 volta.

4.5.3 Cenário E com 5 Voltas

Para avaliar a localização do robô, foi calculado o erro da posição final do robô computada pela odometria, pelo SLAM 1R e pelo SLAM 2R ao final da trajetória percorrida pelo robô após a 5 voltas no percurso. A Tabela 11 mostra o erro médio quadrático calculado sobre a coordenada x , y e pela orientação θ do robô.

Tabela 11 – Erro Médio Quadrático ao Final da Trajetória do Robô após 5 voltas.

	x	y	θ
Odometria	2,364	3,017	2,032
SLAM - 1R	0,125	0,037	0,001
SLAM - 2R	0,042	0,026	0,001

A Figura 42 mostra um exemplo de percurso realizado pelo robô na cena E após 5 voltas. Podemos visualizar com nitidez a diferença de percurso realizada pelo SLAM 1R em azul claramente divergindo da trajetória real, enquanto o SLAM 2R mantém-se próximo ao real.

A Figura 43 mostra o comportamento da variância das coordenadas x e y para o sistema SLAM 1R (Figura 43a) e para o sistema SLAM 2R (43b). Após 5 voltas é possível observar que há cinco quedas drásticas na variância do SLAM 2R. Isto ocorre porque ao longo do percurso o sistema está localizando os marcos e quanto mais preciso são os marcos menor será o pico e maiores serão as quedas.

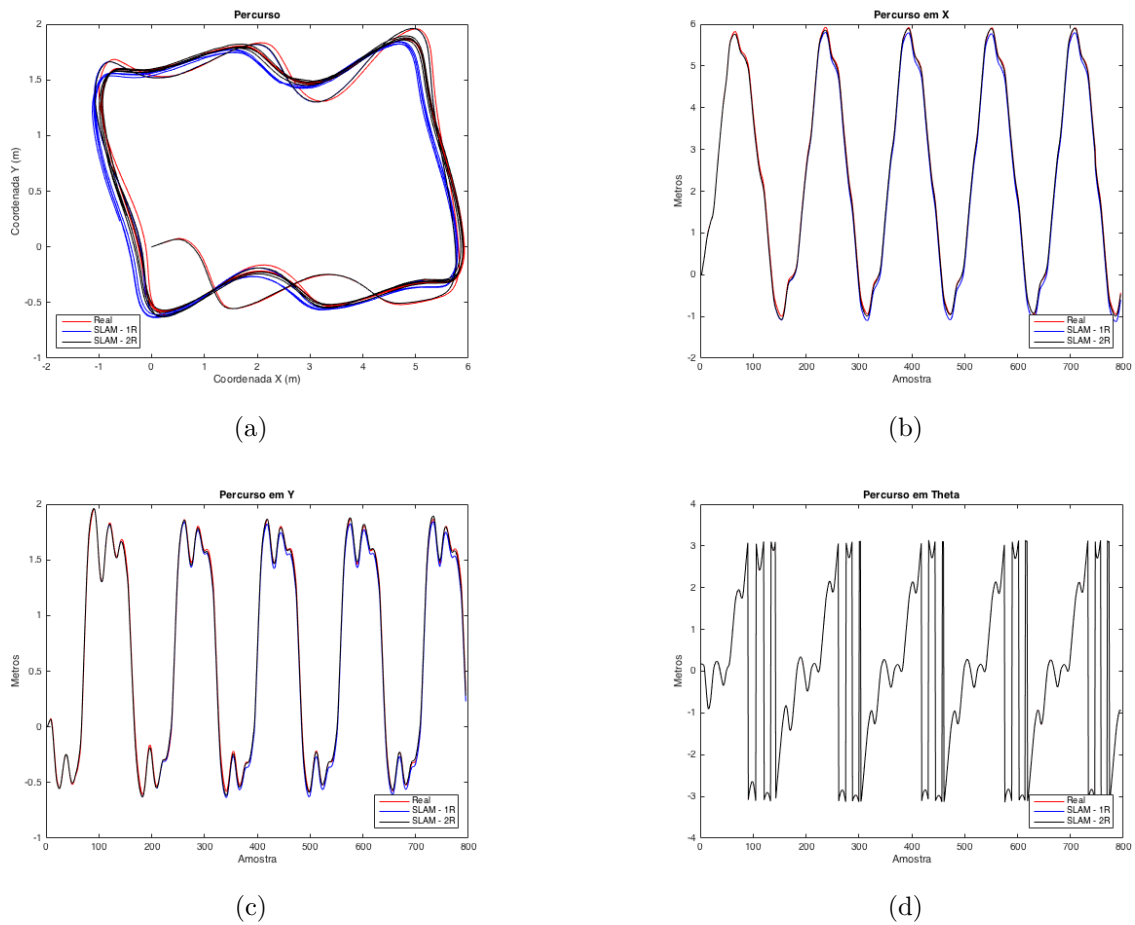


Figura 42 – Percurso do Robô após 5 voltas: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário E.

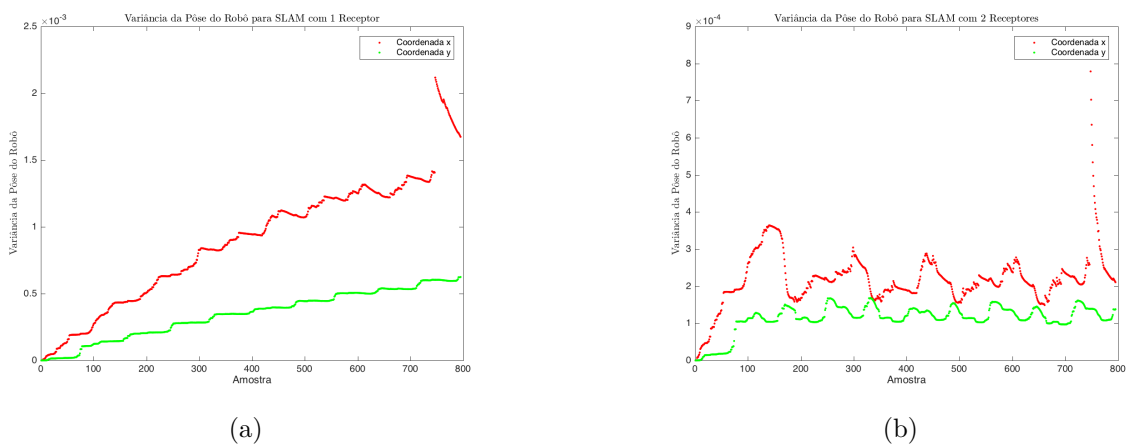


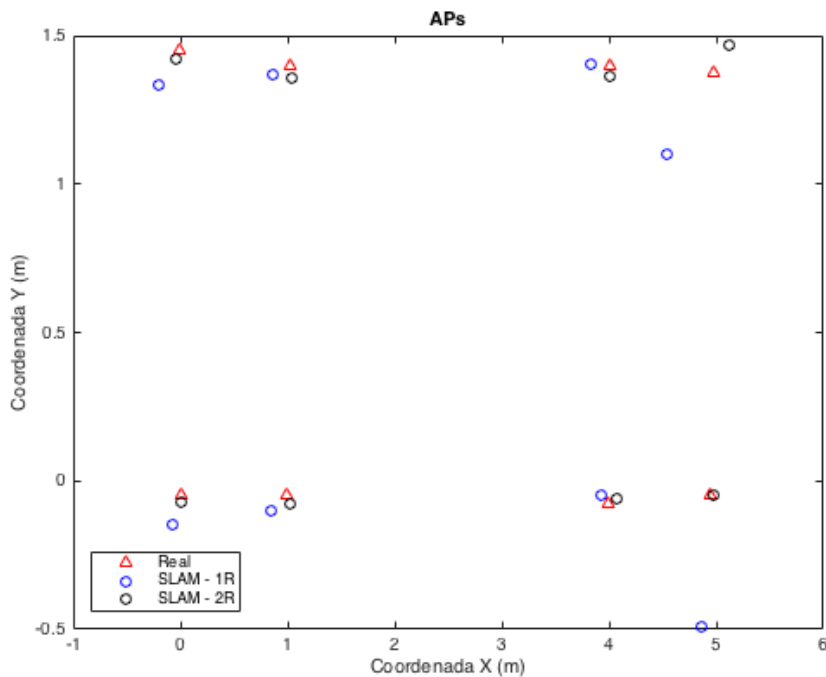
Figura 43 – Gráfico da Variância da Posição do Robô após 5 voltas nas coordenadas x e y: (a) SLAM 1R; (b) SLAM 2R.

Para avaliar a localização dos marcos, foi calculado a distância euclidiana da posição final do marco computada pela odometria, pelo SLAM 1R e pelo SLAM 2R com a posição real do marco ao final da trajetória percorrida pelo robô. A Tabela 12 mostra a distância euclidiana calculado para cada AP na cena E.

Tabela 12 – Distância Euclidiana Final dos APs após 5 voltas.

	SLAM - 1 Receptor	SLAM - 2 Receptores
AP1	0,161	0,038
AP2	0,162	0,042
AP3	0,083	0,066
AP4	0,163	0,037
AP5	0,119	0,025
AP6	0,228	0,046
AP7	0,447	0,019
AP8	0,520	0,174

A Figura 44 mostra a posição final dos marcos após 5 voltas. É obtido por fim um resultado visivelmente favorável ao SLAM de 2 receptores, onde após 5 voltas teve um erro na casa de 10^{-2} com exceção do AP 8 por razões já explicadas nos resultados da terceira volta, enquanto o SLAM 1R manteve seu erro na casa de 10^{-1} .



(a)

Figura 44 – Resultado da Posição Final dos 8 APs após 5 volta.

4.6 Considerações Finais

Dois cenários foram utilizados para exemplificar o problema tratado no SLAM. O primeiro cenário apresentou 4 cenas, sendo a primeira, cena A, onde o robô não possui ruído no sistema para exemplificar um cenário perfeito. A cena B tratou da influência do ruído no modelo de odometria do robô. A cena C mostrou a influência do ruído na etapa de atualização do SLAM e, por fim, a cena D mostrou o comportamento do sistema SLAM onde houve ruído tanto na etapa de predição quanto na etapa de atualização.

No segundo cenário foi mostrado o comportamento do robô em um percurso longo em que os APs continuam em seu campo de recepção. Neste cenário há uma diferença discrepante entre as posições dos APs e percurso do robô para o SLAM 1R e SLAM 2R.

Com os resultados mostrados neste capítulo, é possível observar que o ruído do sinal Wi-Fi é relevante para a acurácia do sistema, impactando no resultado final de um sistema real de localização e mapeamento simultâneos utilizando apenas a leitura do sinal como método atualizador.

Os resultados demonstram que, mesmo com uma odometria ruidosa, a abordagem proposta foi capaz de obter um sistema com acurácia e capaz de localizar um robô de pequeno porte enquanto mapeia os sinais de Wi-Fi de um determinado ambiente.

5 Conclusões e Trabalhos Futuros

O processo de Localização e Mapeamento Simultâneos é um dos problemas fundamentais em tarefas de navegação autônoma de robôs, e por isso vem sendo alvo de extensa pesquisa. Uma solução para esse problema pode permitir o surgimento de robôs verdadeiramente autônomos, capazes de navegar de maneira segura por ambientes desconhecidos e cumprir objetivos sem a necessidade de auxílio externo de espécie alguma.

Com base nisso, este trabalho apresentou uma solução SLAM adequada para ambientes com um ou mais transmissores Wi-Fi, de modo a permitir que o robô navegue no ambiente fundindo informações de odometria e de sinal Wi-Fi. Ressalta-se que a restrição de transmissores Wi-Fi no ambiente não é tão restritiva pois são comuns em muitos ambientes como museus, hospitais, universidades, shoppings, dentre outros.

O sistema que foi descrito faz uso de dois receptores Wi-Fi para determinar a distância entre o robô e os transmissores Wi-Fi visíveis no ambiente. Inicialmente é realizada uma trilateração para estimar a posição inicial dos transmissores no ambiente e o ponto da posição inicial é utilizado como marcos no SLAM. Em seguida é realizada uma filtragem por qualidade do sinal recebido, portanto sinais com informações com ruído acima de um threshold são eliminadas. Por fim, a pose do robô é corrigida por cada um dos sinais recebidos pelos receptores juntamente com a posição dos marcos.

Foram apresentadas várias experiências com características distintas para validar o sistema e se este executou as tarefas de forma satisfatória em todos os casos. Dentre as experiências mostradas, duas se destacam por contemplar uma simulação mais próxima da realidade, as Cenas D e E. Na Cena D foi mostrado o comportamento do robô em um percurso curto com pouca visualização de APs, e na Cena E foi apresentado o comportamento do robô em uma trajetória circular com grande visualização dos APs. Ao comparar o SLAM com 1 receptor e o SLAM com dois receptores, o uso de dois receptores obteve melhores resultados, tanto da posição final do robô quanto da posição inicial e final dos APs.

Em suma, as principais contribuições desse trabalho são:

1. sistema de range-only SLAM em tempo real adequado para ambientes com um ou mais transmissores Wi-Fi utilizando técnicas clássicas da literatura;
2. modelagem dos receptores de sinal Wi-Fi;
3. modelagem de filtro do sinal Wi-Fi;
4. modelagem da solução com FKE utilizando dois receptores;

5. maior precisão na inicialização do FKE;
6. redundância na atualização do FKE.

Como trabalhos futuros pretende-se:

1. testar a abordagem utilizando outros filtros estatísticos;
2. comparar a solução com demais soluções de range-only SLAM da literatura;
3. realizar experimentos utilizando a solução em um Cellbot

Referências

- ALSINA, P. J. et al. Navegação e controle de robôs móveis. *XIV Congresso Brasileiro de Automática*, 2002. Citado na página 3.
- AROCA, R. et al. Increasing students' interest with low-cost cellbots. *IEEE Transactions on Education*, v. 56, n. 1, p. 3–8, Feb 2013. ISSN 0018-9359. Citado na página 2.
- Bezerra Neto, R. P. et al. Robótica na Educação: Uma Revisão Sistemática dos Últimos 10 Anos. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*. [S.l.: s.n.], 2015. p. 386–393. Citado na página 2.
- BLANCO, J. L.; GONZÁLEZ, J.; FERNÁNDEZ-MADRIGAL, J. A. A pure probabilistic approach to range-only SLAM. *Proceedings - IEEE International Conference on Robotics and Automation*, p. 1436–1441, 2008. ISSN 10504729. Citado na página 16.
- BORENSTEIN, J.; FENG, L. Measurement and correction of systematic odometry errors in mobile robots. *Robotics and Automation, IEEE Transactions on*, v. 12, n. 6, p. 869–880, Dec 1996. ISSN 1042-296X. Citado na página 4.
- CABALLERO, F. et al. A particle filtering method for wireless sensor network localization with an aerial robot beacon. In: *IEEE International Conference on Robotics and Automation*. [S.l.: s.n.], 2008. p. 596–601. ISSN 1050-4729. Citado na página 15.
- CABALLERO, F.; MERINO, L.; OLLERO, A. A general Gaussian-mixture approach for range-only mapping using multiple hypotheses. In: *Proceedings - IEEE International Conference on Robotics and Automation*. IEEE, 2010. p. 4404–4409. ISBN 9781424450381. ISSN 10504729. Disponível em: <<http://ieeexplore.ieee.org/document/5509704/>>. Citado na página 16.
- CRAIG, J. *Introduction to Robotics: Mechanics and Control*. [S.l.]: Pearson/Prentice Hall, 2005. (Addison-Wesley series in electrical and computer engineering: control engineering). ISBN 9780201543612. Citado na página 1.
- Dan Hai et al. Simultaneous localization and mapping of Robot in Wireless Sensor Network. *IEEE International Conference on Intelligent Computing and Intelligent Systems*, v. 3, p. 173–178, 2010. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5658491>>. Citado na página 16.
- DJUGASH, J.; SINGH, S. A robust method of localization and mapping using only range. In: *International Symposium on Experimental Robotics*. Pittsburgh, PA: [s.n.], 2008. Citado na página 16.
- FABRESSE, F. R. et al. Active perception for 3D range-only simultaneous localization and mapping with UAVs. *2016 International Conference on Unmanned Aircraft Systems, ICUAS 2016*, p. 394–398, 2016. Citado na página 16.
- FAHEEM, A.; VIRRANKOSKI, R.; ELMUSRATI, M. Improving RSSI based distance estimation for 802.15.4 wireless sensor networks. In: *IEEE International Conference on Wireless Information Technology and Systems, ICWITS 2010*. IEEE, 2010. p. 1–4.

ISBN 9781424470914. Disponível em: <<http://ieeexplore.ieee.org/document/5611813/>>. Citado 2 vezes nas páginas 19 e 24.

GENEVE, L.; KERMORGANT, O.; LAROCHE, E. A composite beacon initialization for EKF range-only SLAM. *IEEE International Conference on Intelligent Robots and Systems*, v. 2015-Decem, p. 1342–1348, 2015. ISSN 21530866. Citado 2 vezes nas páginas 16 e 17.

IEEE. *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012) : IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Acce.* IEEE, 2016. 3534 p. ISBN 9781504436458. Disponível em: <<http://ieeexplore.ieee.org/document/7786995/>>. Citado na página 14.

KALMAN, R. E. A new approach to linear filtering and prediction problems. *ASME Journal of Basic Engineering*, 1960. Citado na página 12.

KURT-YAVUZ, Z.; YAVUZ, S. A comparison of EKF, UKF, FastSLAM2.0, and UKF-based FastSLAM algorithms. In: *INES - IEEE 16th International Conference on Intelligent Engineering Systems, Proceedings*. [S.l.: s.n.], 2012. p. 37–43. ISBN 9781467326957. ISSN 9781467326957. Citado na página 16.

LATOMBE, J.-C. *Robot Motion Planning*. Norwell, MA, USA: Kluwer Academic Publishers, 1991. ISBN 079239206X. Citado 2 vezes nas páginas 11 e 21.

LIU, H. et al. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, v. 37, n. 6, p. 1067–1080, Nov 2007. ISSN 1094-6977. Citado na página 14.

NORRDINE, A. An Algebraic Solution to the Multilateration Problem. In: *Proceedings of the 15th International Conference on Indoor Positioning and Indoor Navigation*. [S.l.: s.n.], 2012. Citado na página 19.

OLSON, E.; LEONARD, J. J.; TELLER, S. Robust range-only beacon localization. *IEEE Journal of Oceanic Engineering*, v. 31, n. 4, p. 949–958, 2006. ISSN 03649059. Citado na página 15.

ROHMER, E.; SINGH, S. P. N.; FREESE, M. V-rep: a versatile and scalable robot simulation framework. In: *Proceedings of The International Conference on Intelligent Robots and Systems (IROS)*. [S.l.: s.n.], 2013. Citado na página 31.

ROMERO, R. et al. *Robótica Móvel*. [S.l.]: LTC, 2014. ISBN 978-85-216-2303-8. Citado na página 1.

SANTOS, A.; TARRATACA, L.; CARDOSO, J. The feasibility of navigation algorithms on smartphones using j2me. *Mobile Networks and Applications*, Springer US, v. 15, n. 6, p. 819–830, 2010. ISSN 1383-469X. Disponível em: <<http://dx.doi.org/10.1007/s11036-010-0236-8>>. Citado na página 2.

THRUN, S.; BURGARD, W.; FOX, D. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. [S.l.]: The MIT Press, 2005. ISBN 0262201623. Citado 6 vezes nas páginas 5, 12, 14, 19, 21 e 22.

TUNA, G. et al. Evaluations of different Simultaneous Localization and Mapping (SLAM) algorithms. In: *IECON Proceedings (Industrial Electronics Conference)*. IEEE, 2012. p. 2693–2698. ISBN 9781467324212. ISSN 1553-572X. Disponível em: <<http://ieeexplore.ieee.org/document/6389151/>>. Citado na página 16.

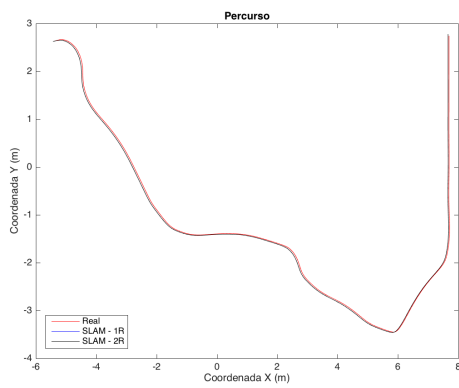
WANG, Z. M.; MIAO, D. H.; DU, Z. J. Simultaneous localization and mapping for mobile robot based on an improved particle filter algorithm. In: *International Conference on Mechatronics and Automation*. IEEE, 2009. p. 1106–1110. ISBN 978-1-4244-2692-8. Disponível em: <<http://ieeexplore.ieee.org/document/5246103/>>. Citado na página 16.

YANG, P. Efficient particle filter algorithm for ultrasonic sensor-based 2D range-only simultaneous localisation and mapping application. *Wireless Sensor Systems, IET*, v. 2, n. 4, p. 394–401, dec 2012. ISSN 2043-6386. Disponível em: <<http://digital-library.theiet.org/content/journals/10.1049/iet-wss.2011.0129>>. Citado na página 16.

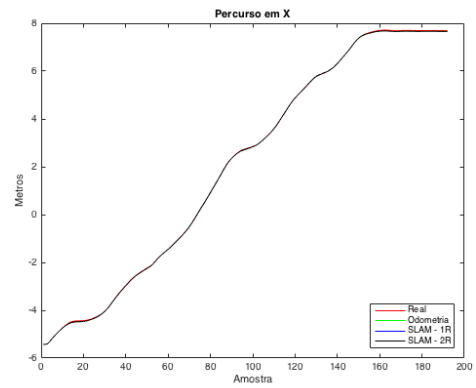
ZHOU, Y. An efficient least-squares trilateration algorithm for mobile robot localization. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*. IEEE, 2009. p. 3474–3479. ISBN 9781424438044. Disponível em: <<http://ieeexplore.ieee.org/document/5354370/>>. Citado 2 vezes nas páginas 16 e 17.

Apêndices

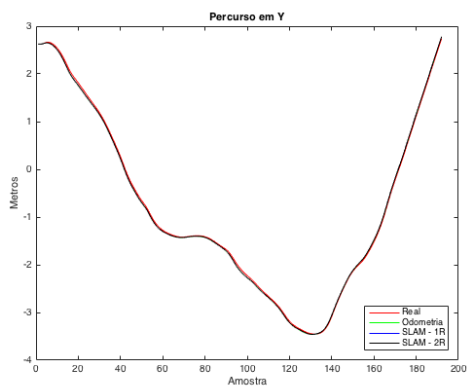
APÊNDICE A – Resultados Adicionais da Cena A



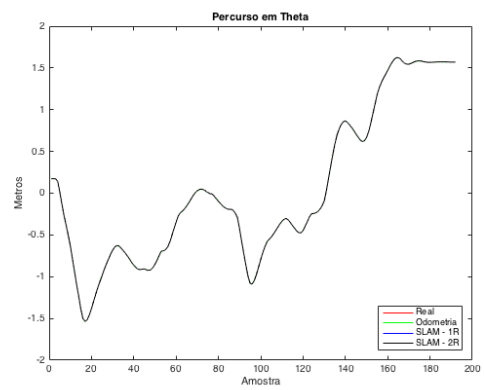
(a)



(b)

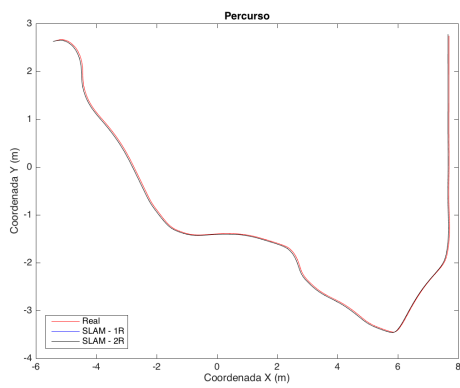


(c)

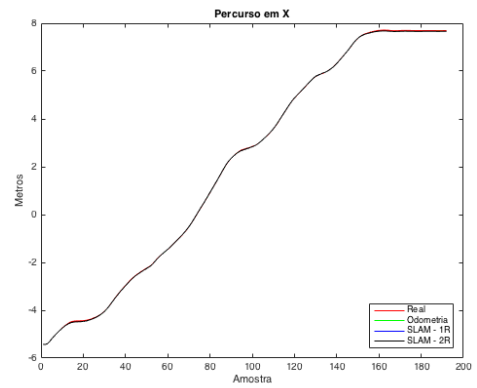


(d)

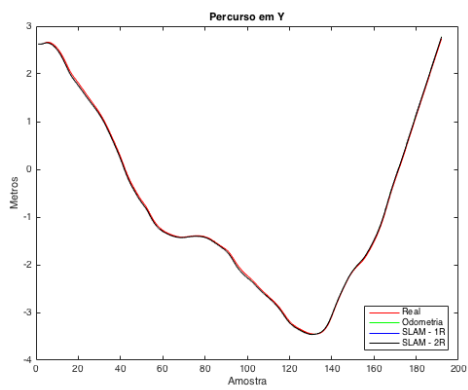
Figura 45 – Trajetória do Robô com 3 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cena A.



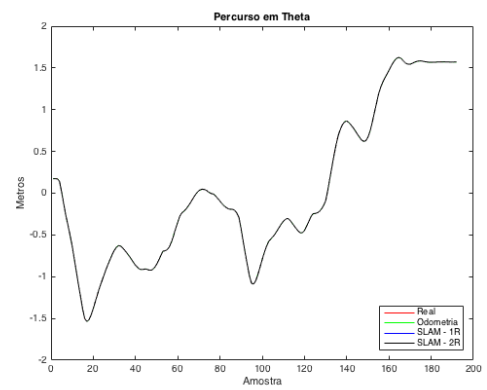
(a)



(b)

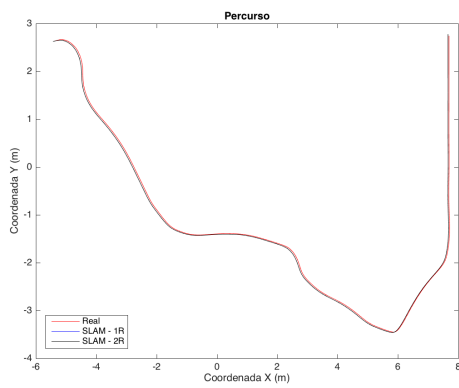


(c)

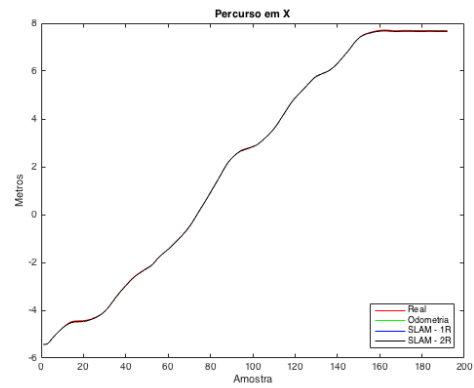


(d)

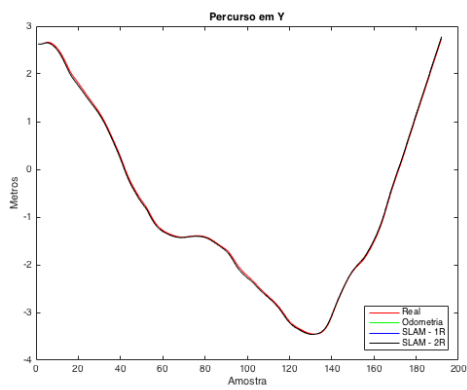
Figura 46 – Trajetória do Robô com 4 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cena A.



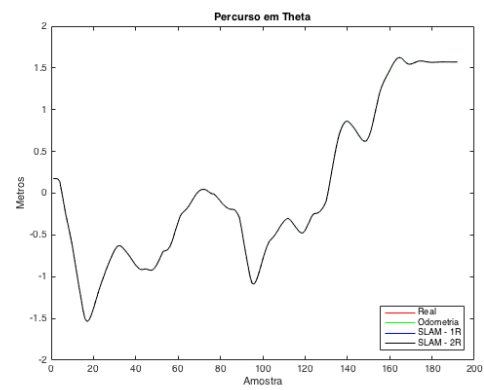
(a)



(b)

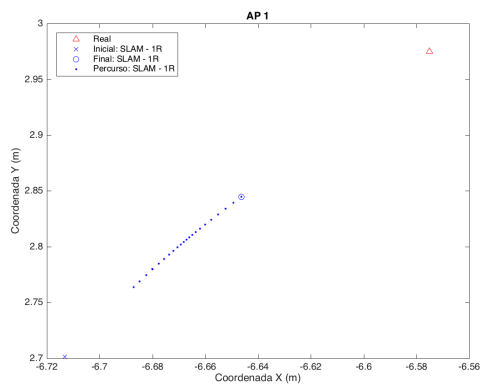


(c)

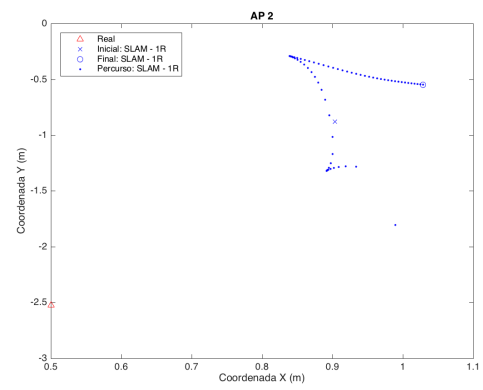


(d)

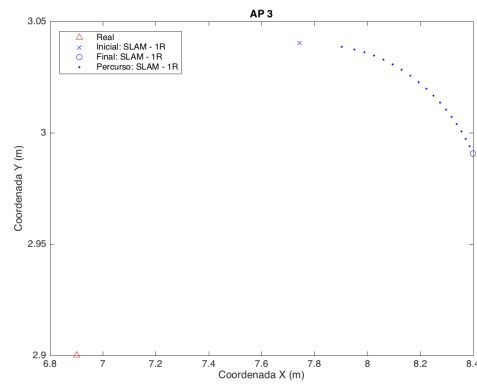
Figura 47 – Trajetória do Robô com 6 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cena A.



(a)



(b)



(c)

Figura 48 – Trajetória dos APs no SLAM com 1 Receptor: (a) AP 1; (b) AP 2; (c): AP 3 - Cena A.

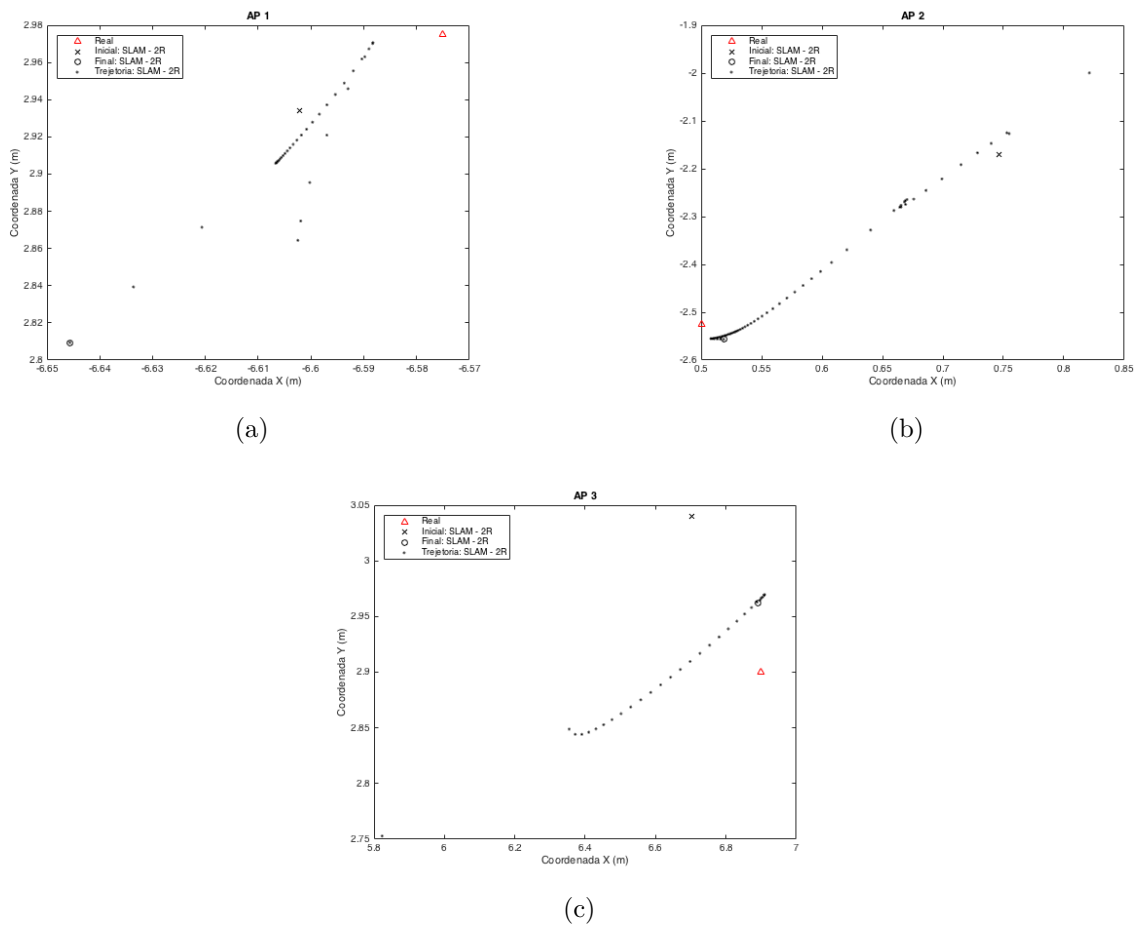
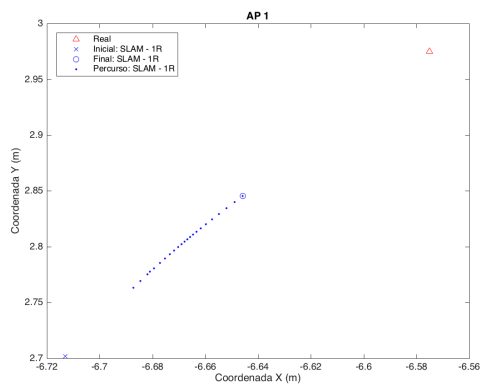
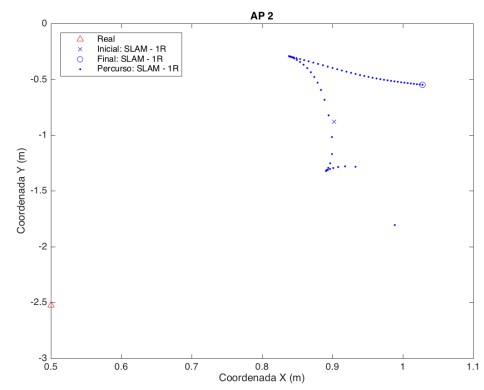


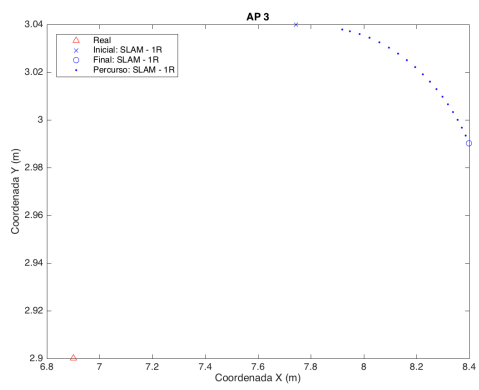
Figura 49 – Trajetória dos APs no SLAM com 2 Receptores: (a) AP 1; (b) AP 2; (c): AP 3 - Cena A.



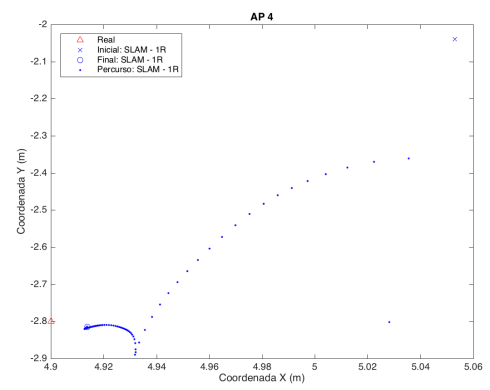
(a)



(b)



(c)



(d)

Figura 50 – Trajetória dos APs no SLAM com 1 Receptor: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4 - Cena A.

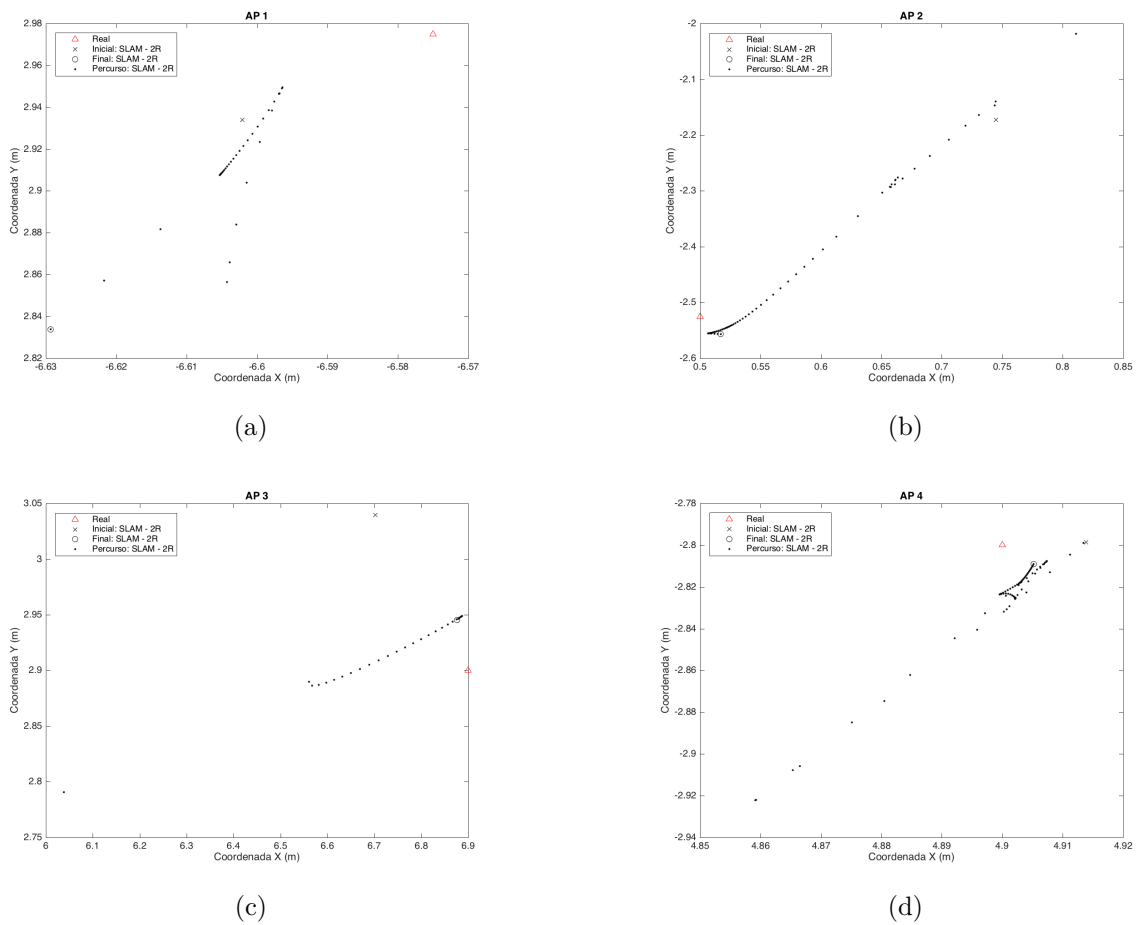
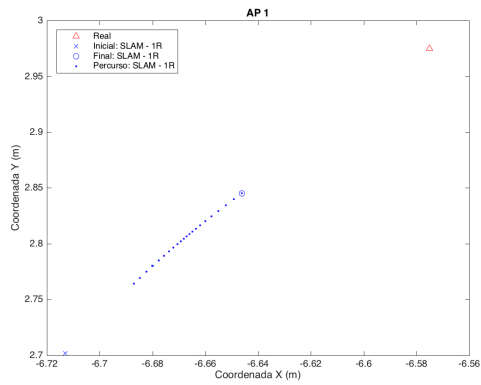
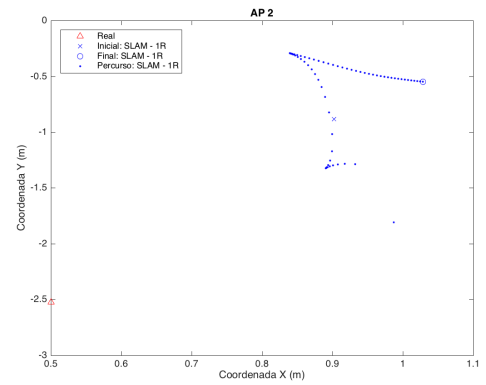


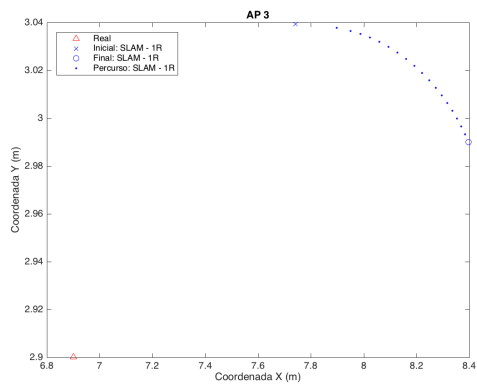
Figura 51 – Trajetória dos APs no SLAM com 2 Receptores: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4 - Cena A.



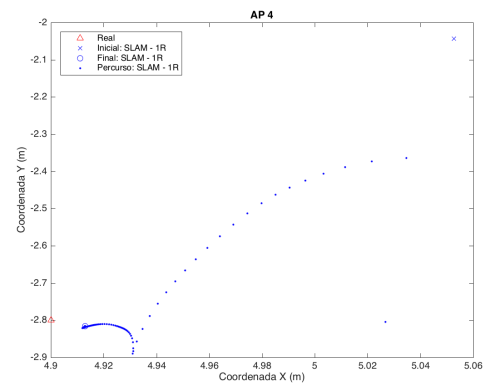
(a)



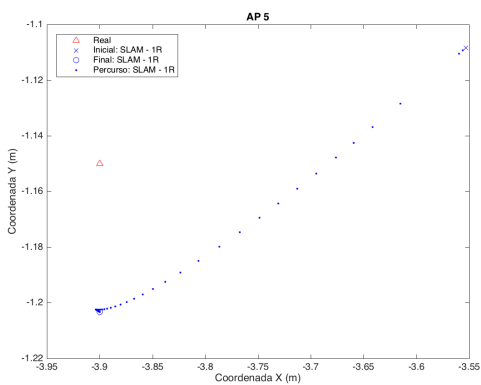
(b)



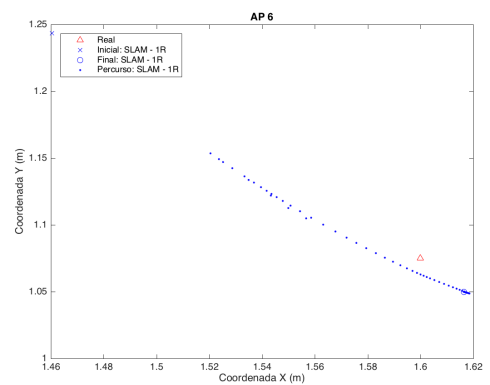
(c)



(d)



(e)



(f)

Figura 52 – Trajetória dos APs no SLAM com 1 Receptor: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4; (e): AP 5; (f) AP 6 - Cena A.

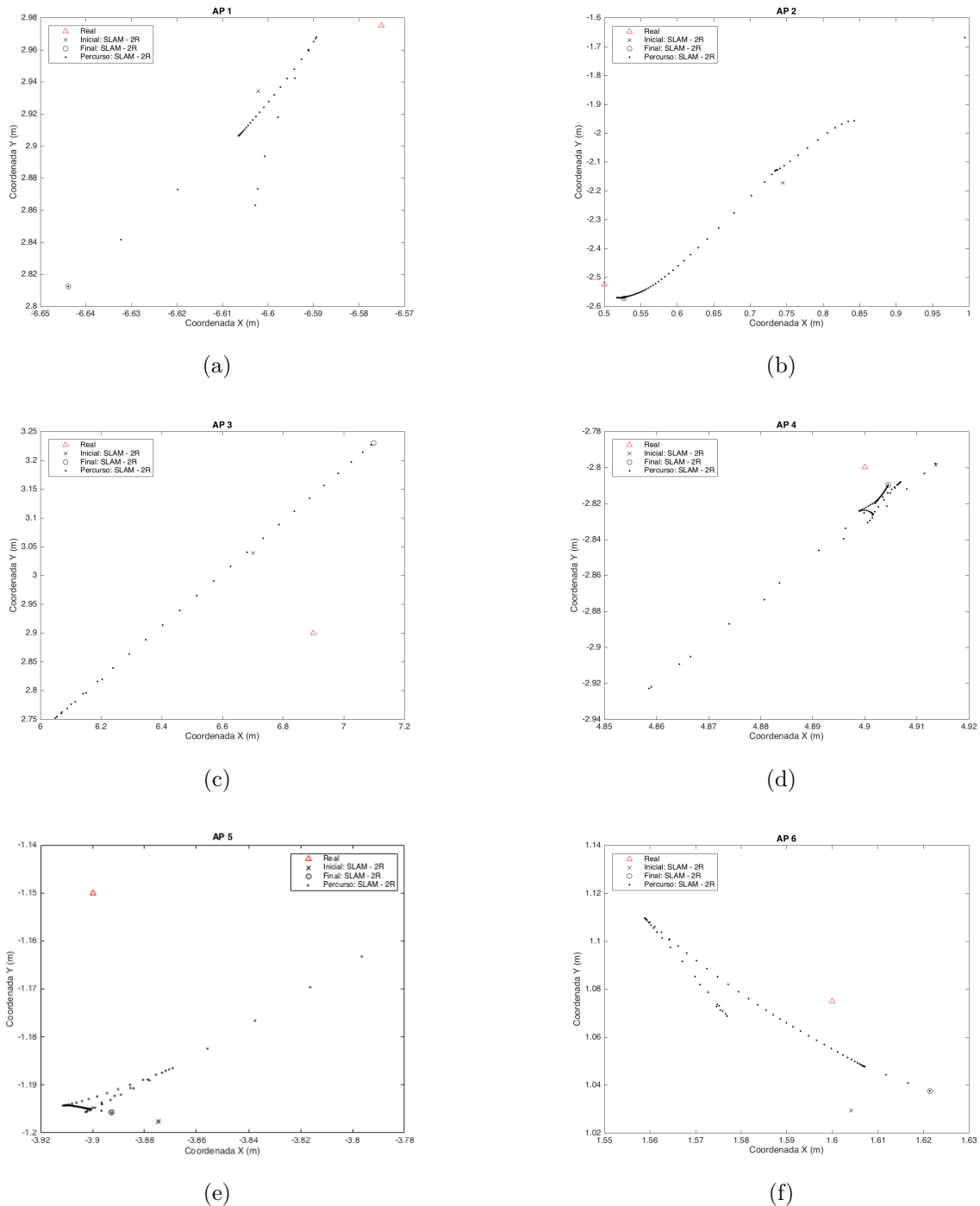
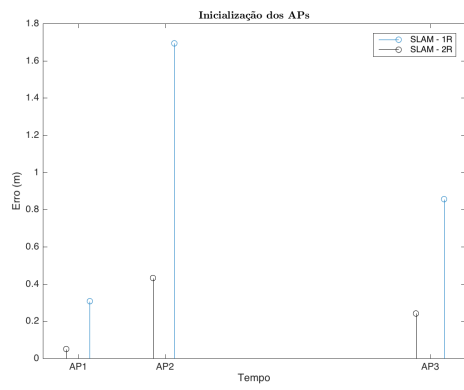
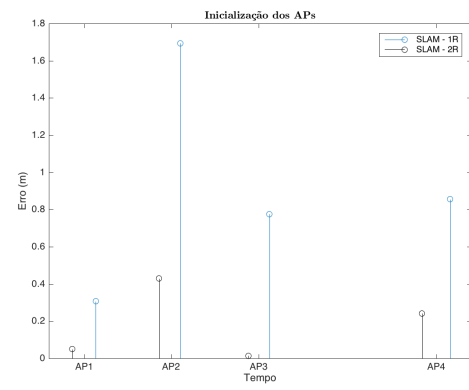


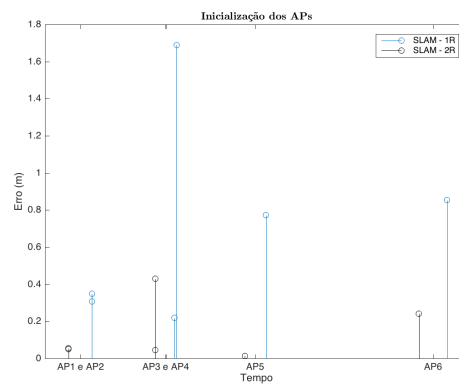
Figura 53 – Trajetória dos APs no SLAM com 2 Receptores: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4; (e): AP 5; (f) AP 6 - Cena A.



(a)



(b)



(c)

Figura 54 – Resultado do Erro da Estimativa Inicial: (a) 3 APs; (b) 4 APs; (c) 6 APs - Cena A.

APÊNDICE B – Resultados Adicionais da Cena B

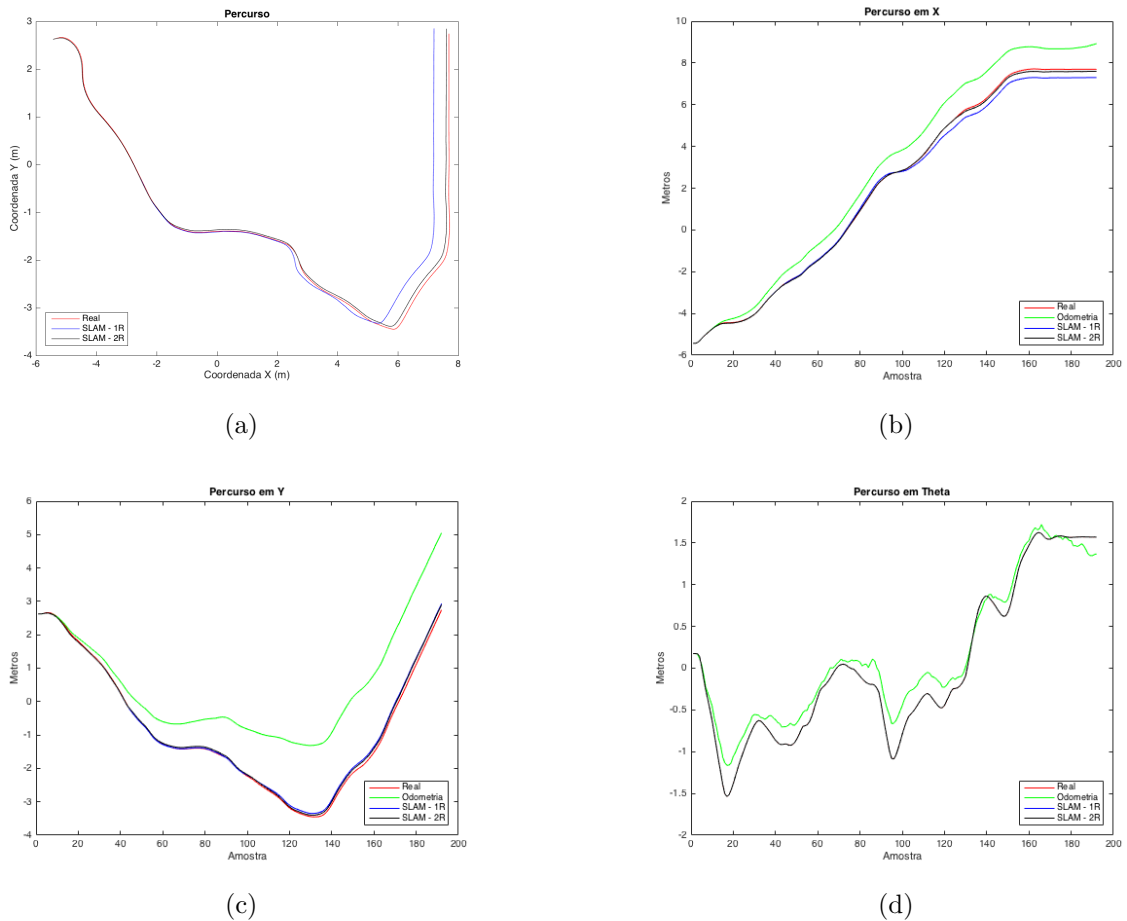
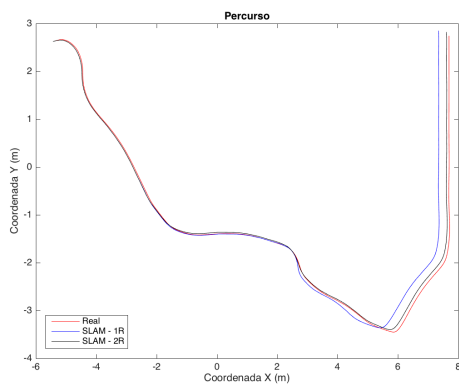
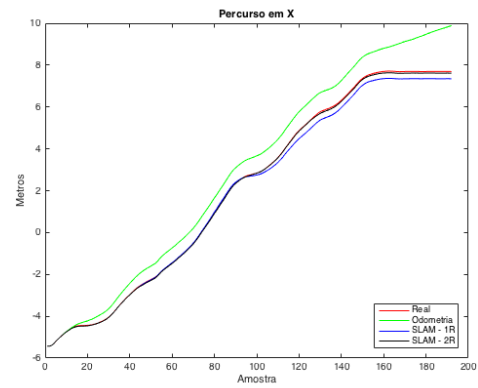


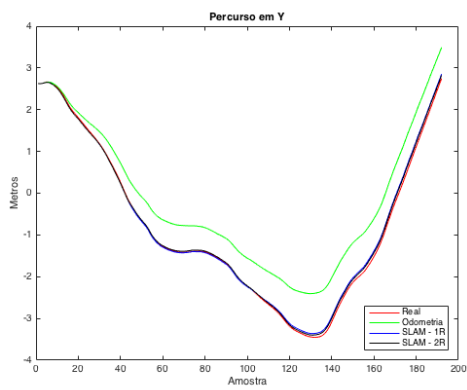
Figura 55 – Trajetória do Robô com 3 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário B.



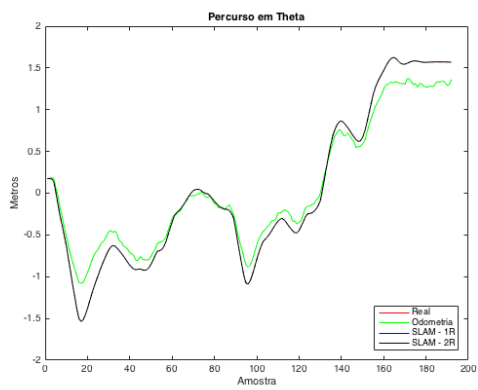
(a)



(b)

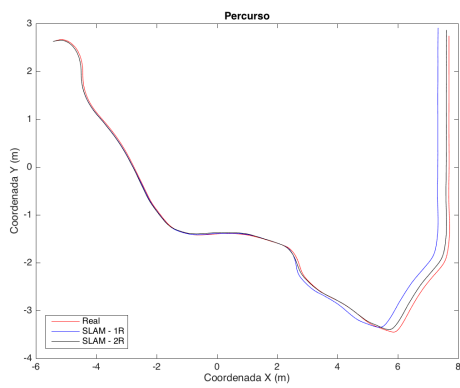


(c)

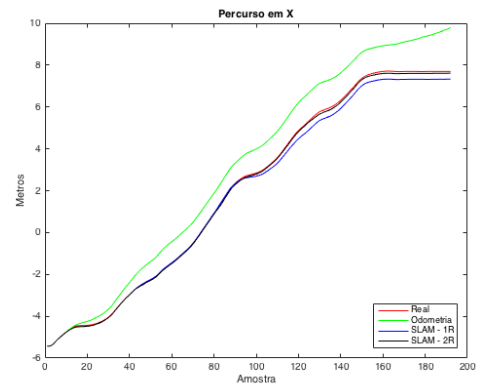


(d)

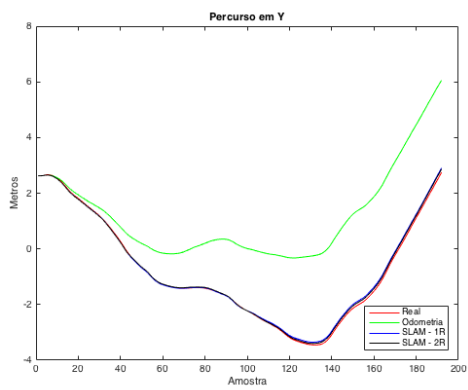
Figura 56 – Trajetória do Robô com 4 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário B.



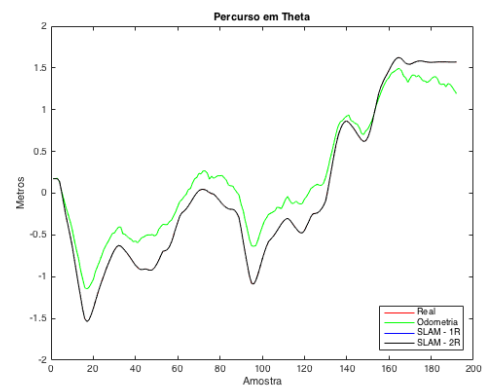
(a)



(b)



(c)



(d)

Figura 57 – Trajetória do Robô com 6 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário B.

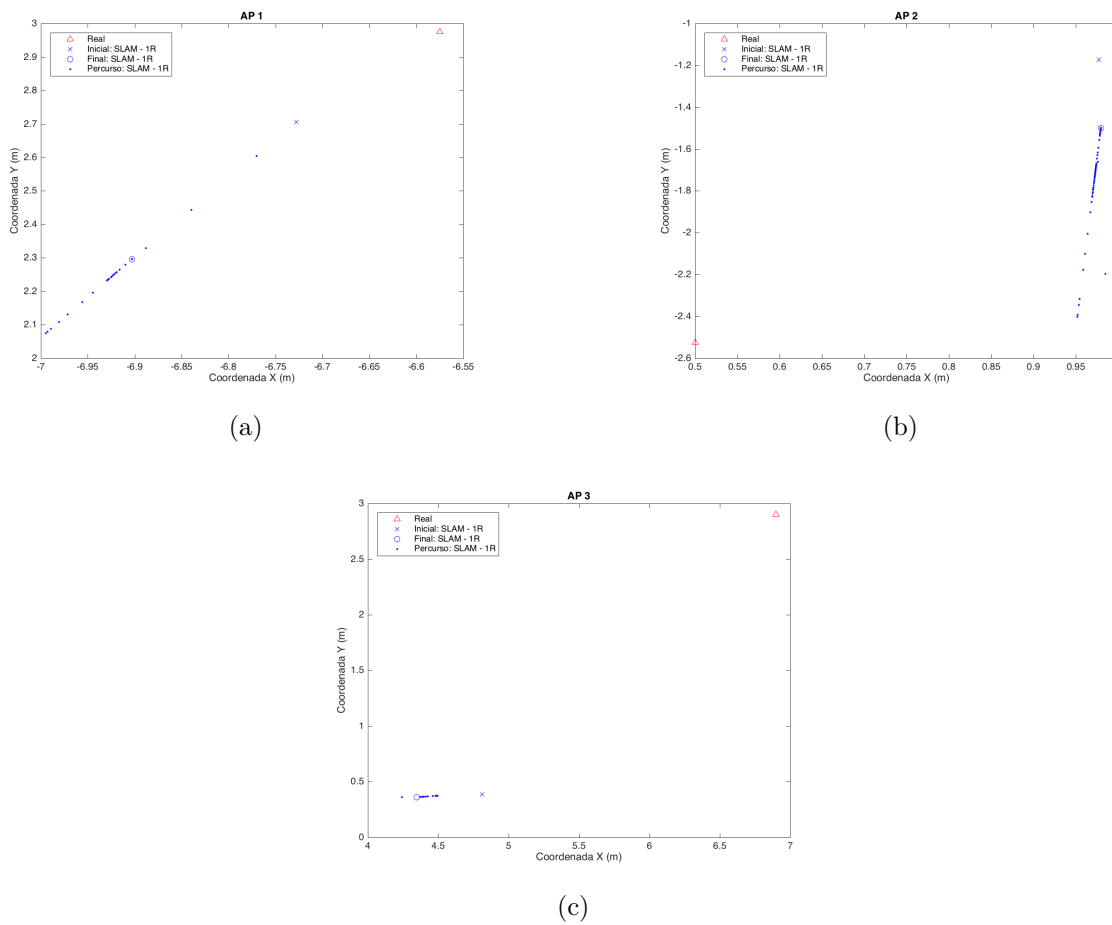
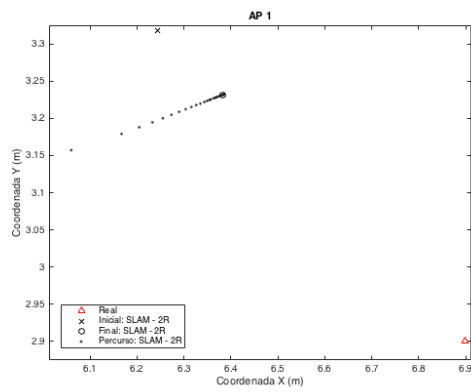
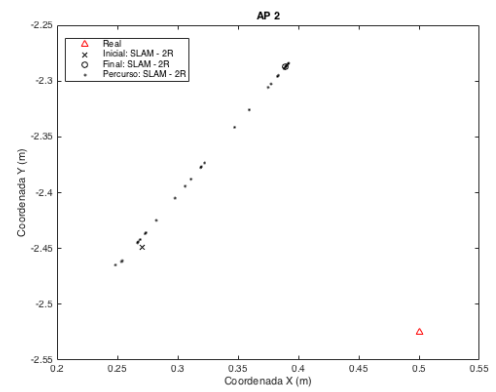


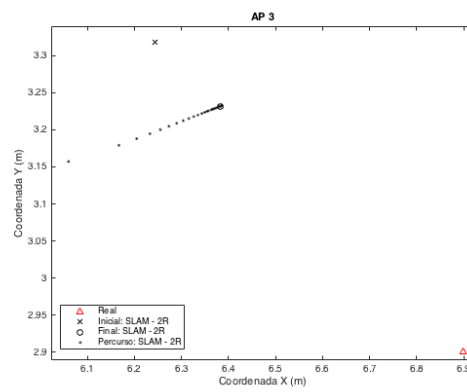
Figura 58 – Trajetória dos APs no SLAM com 1 Receptor: (a) AP 1; (b) AP 2; (c): AP 3 - Cenário B.



(a)



(b)



(c)

Figura 59 – Trajetória dos APs no SLAM com 2 Receptores: (a) AP 1; (b) AP 2; (c): AP 3 - Cenário B.

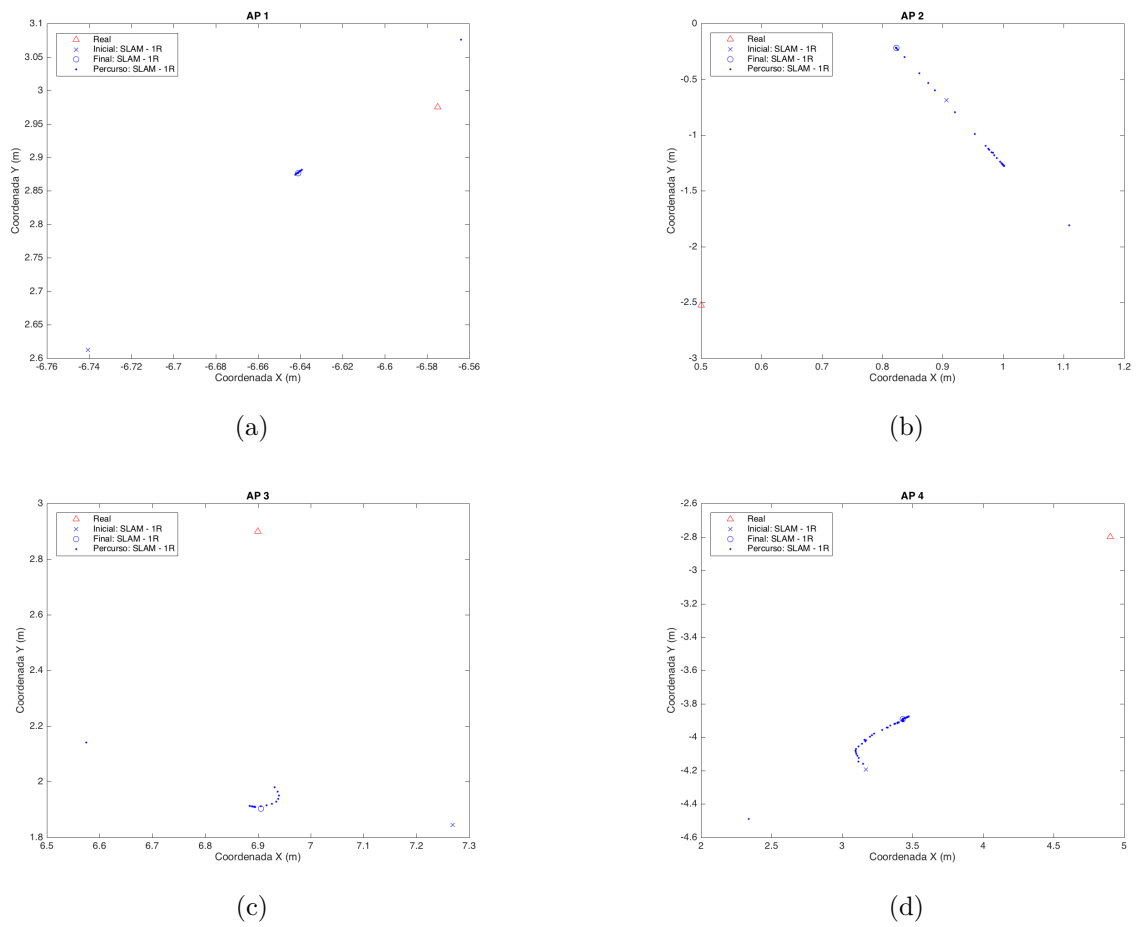
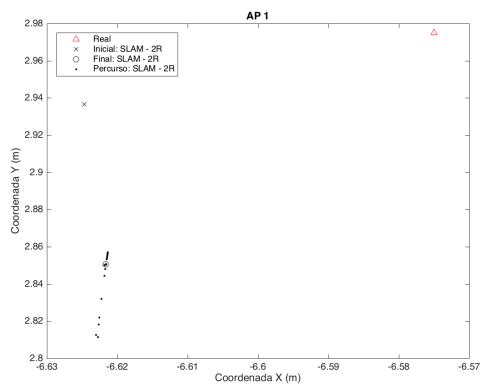
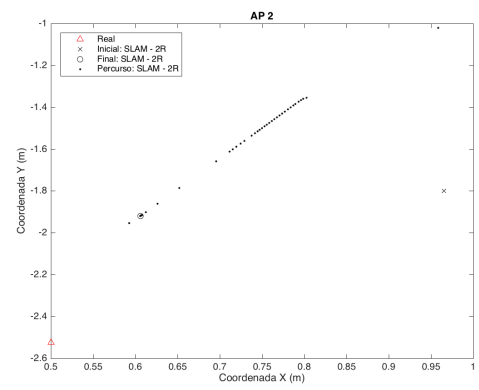


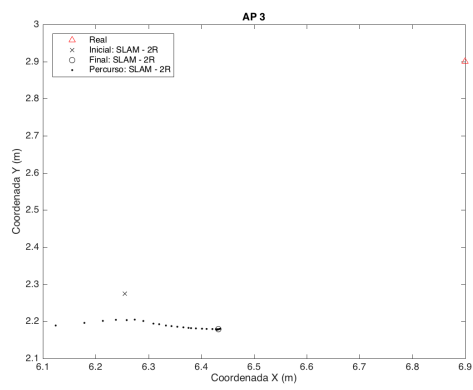
Figura 60 – Trajetória dos APs no SLAM com 1 Receptor: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4 - Cenário B.



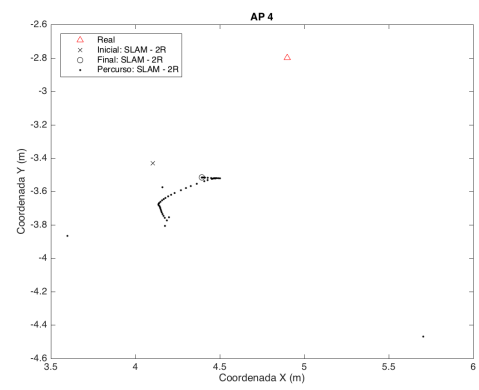
(a)



(b)

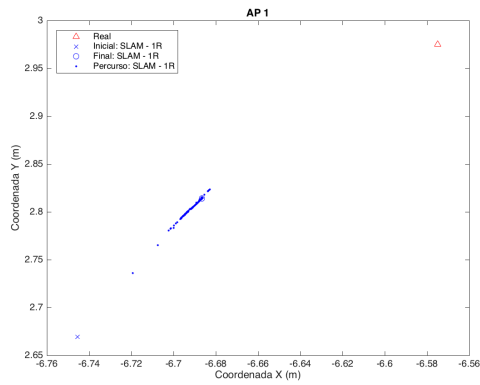


(c)

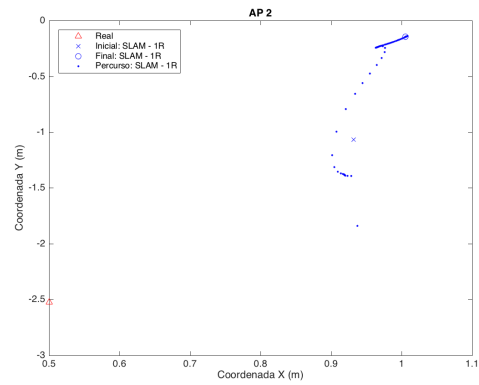


(d)

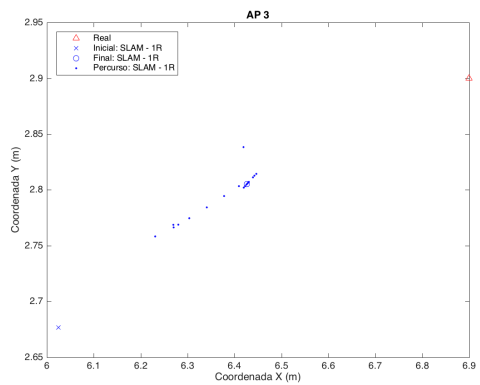
Figura 61 – Trajetória dos APs no SLAM com 2 Receptores: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4 - Cenário B.



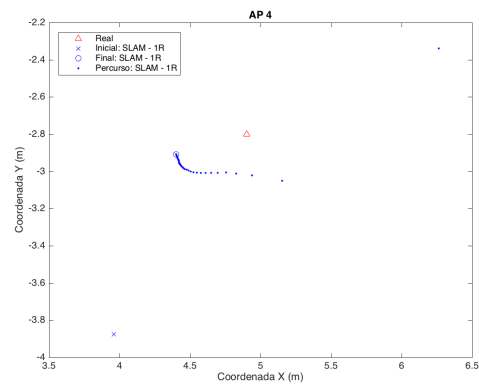
(a)



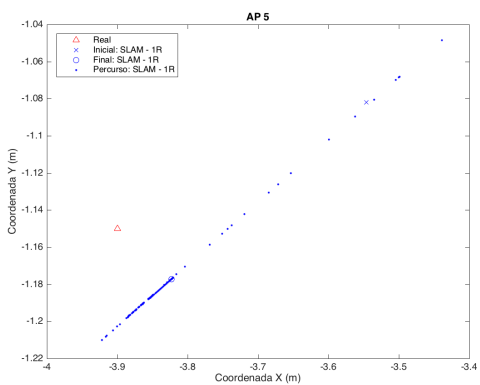
(b)



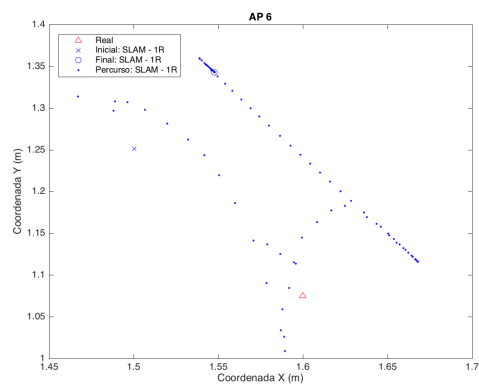
(c)



(d)

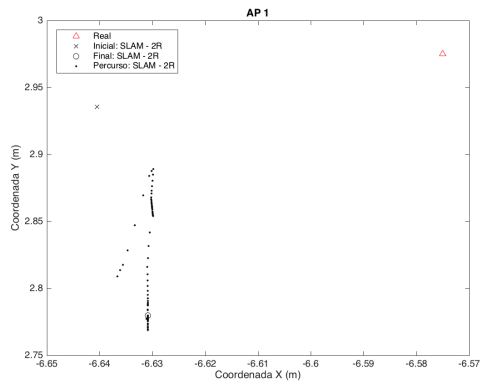


(e)

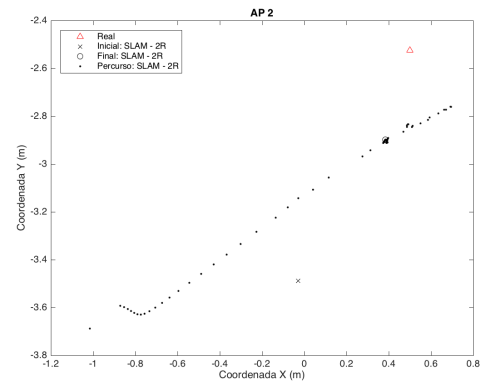


(f)

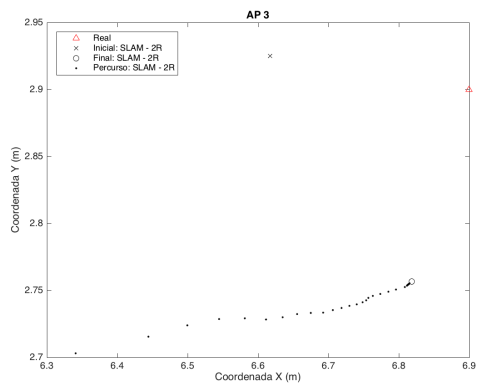
Figura 62 – Trajetória dos APs no SLAM com 1 Receptor: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4; (e): AP 5; (f) AP 6 - Cenário B.



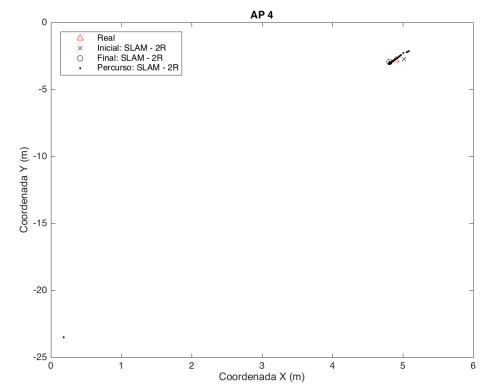
(a)



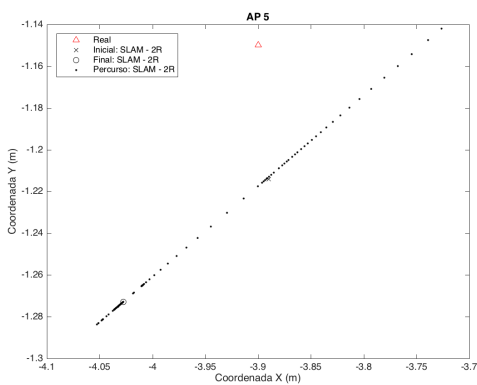
(b)



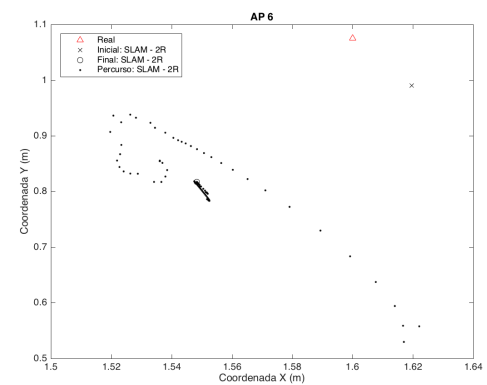
(c)



(d)

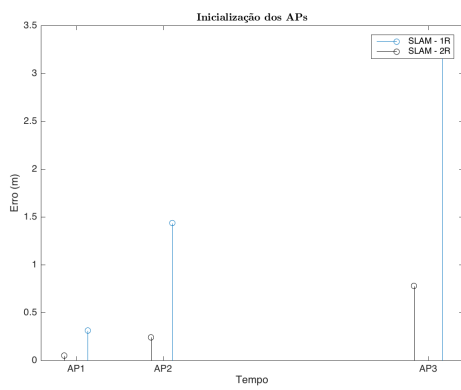


(e)

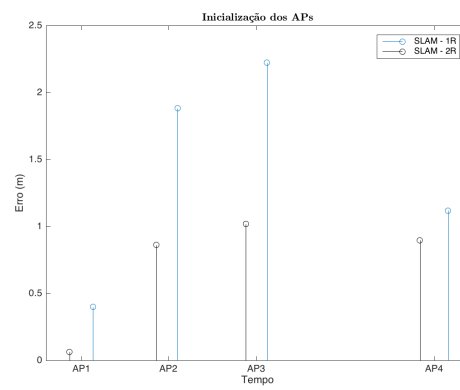


(f)

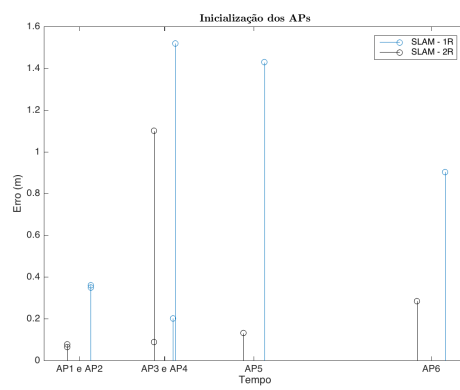
Figura 63 – Trajetória dos APs no SLAM com 2 Receptores: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4; (e): AP 5; (f) AP 6 - Cenário B.



(a)



(b)



(c)

Figura 64 – Resultado do Erro da Estimativa Inicial: (a) 3 APs; (b) 4 APs; (c) 6 APs - Cena B.

APÊNDICE C – Resultados Adicionais da Cena C

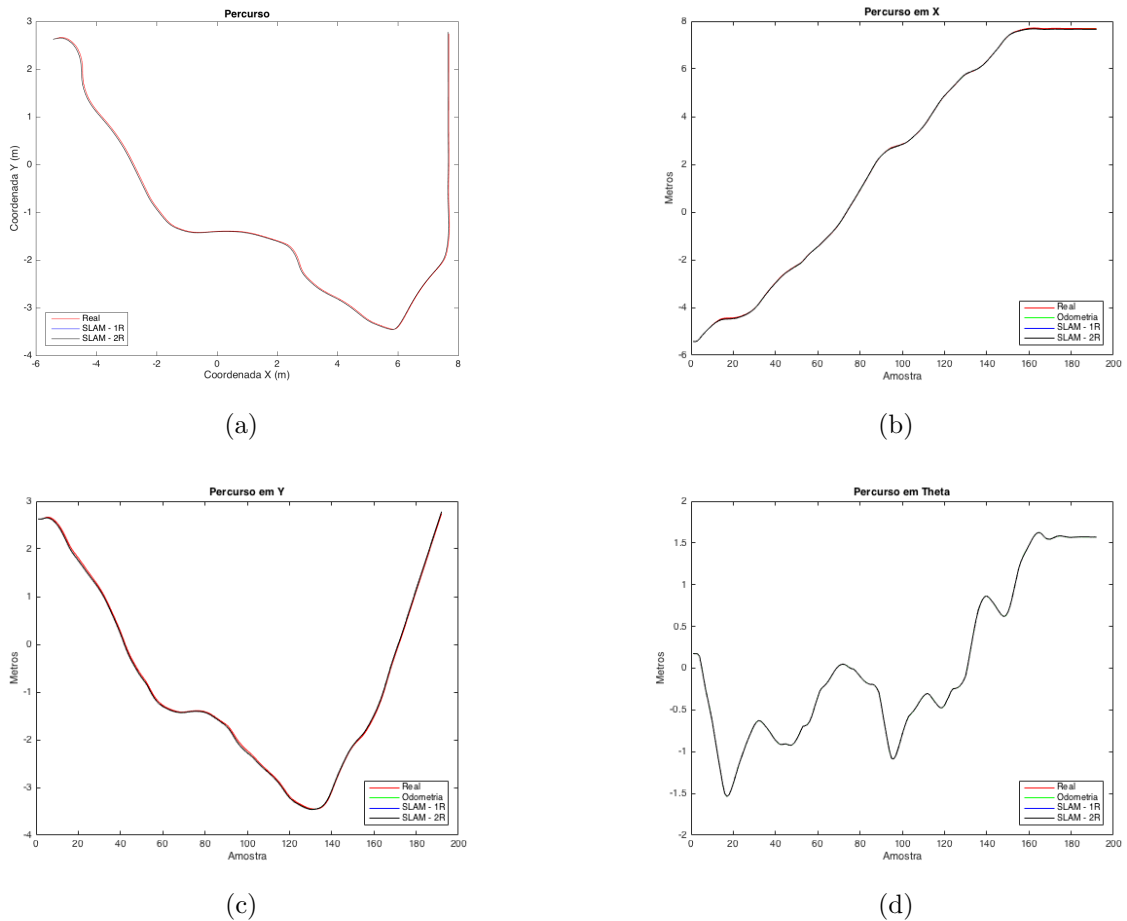
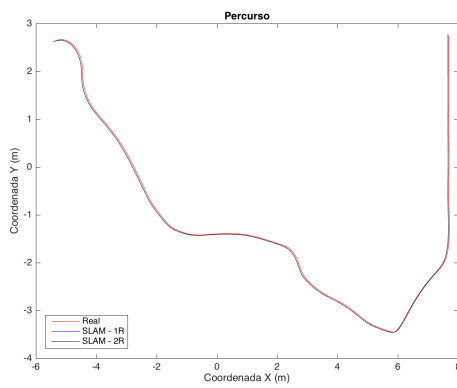
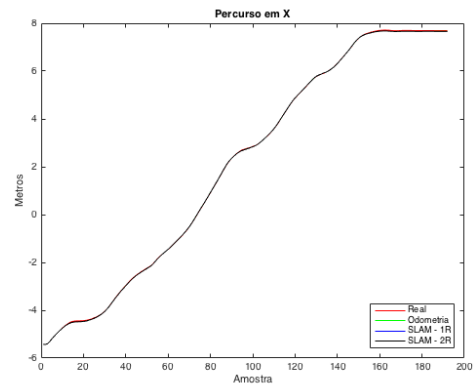


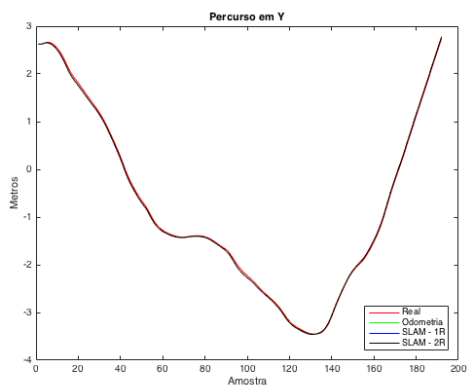
Figura 65 – Trajetória do Robô com 3 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário C.



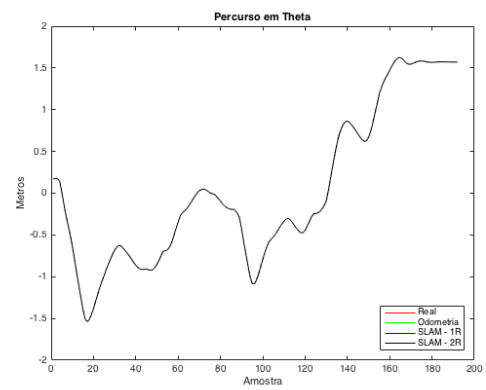
(a)



(b)

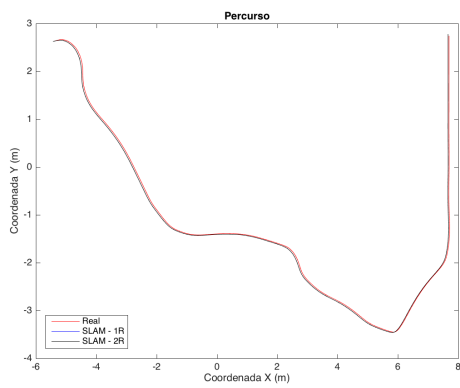


(c)

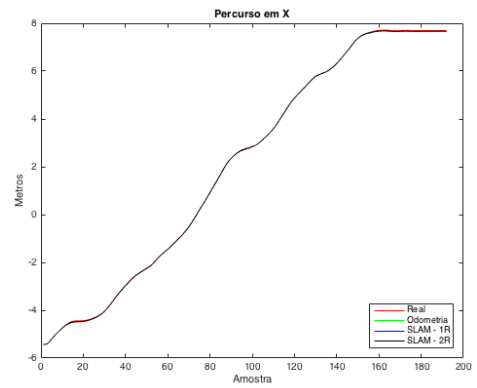


(d)

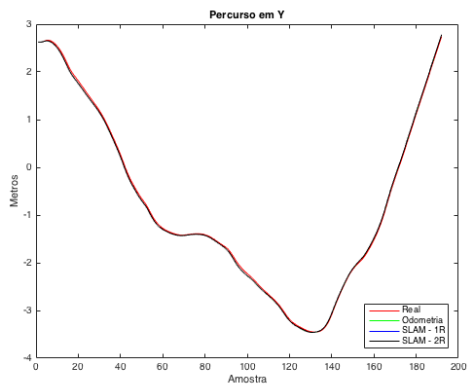
Figura 66 – Trajetória do Robô com 4 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário C.



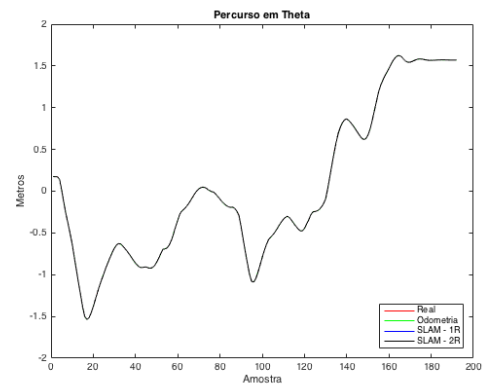
(a)



(b)



(c)



(d)

Figura 67 – Trajetória do Robô com 6 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário C.

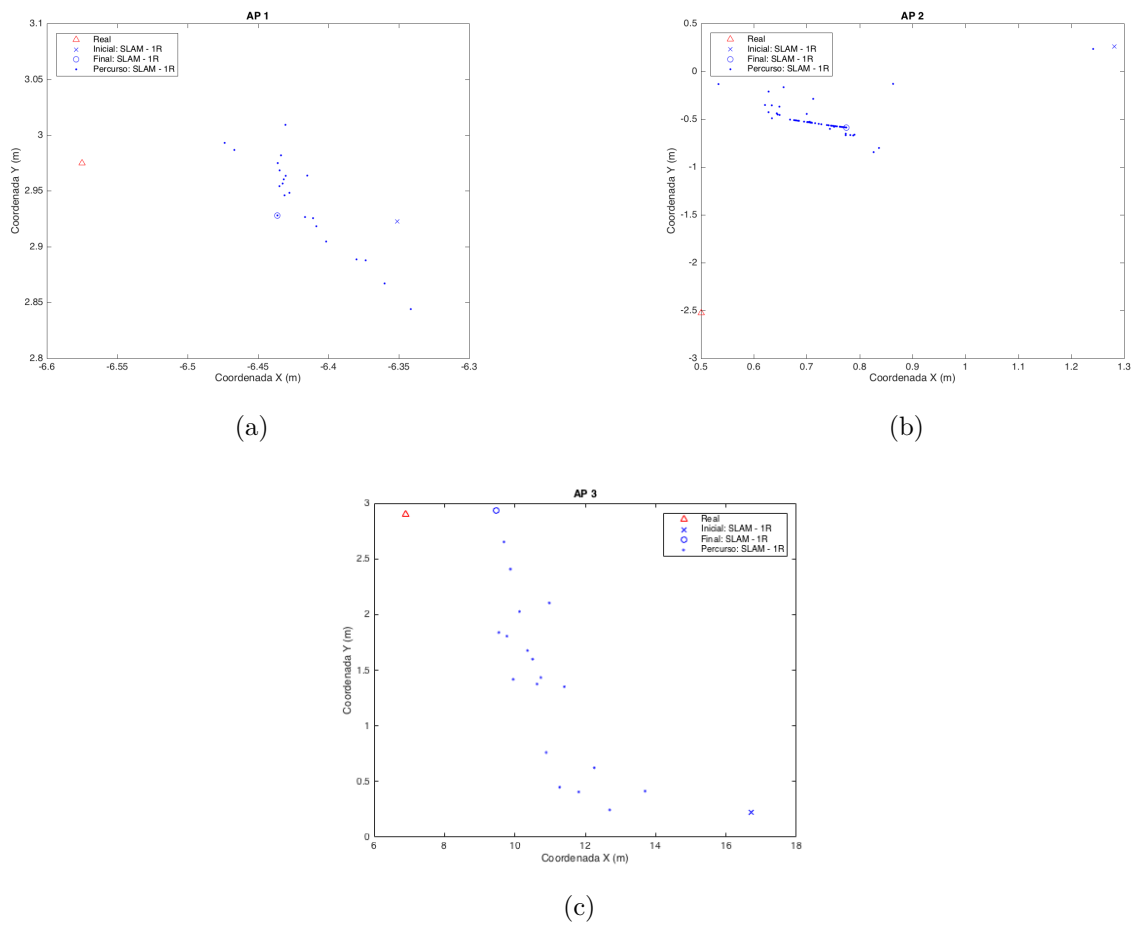
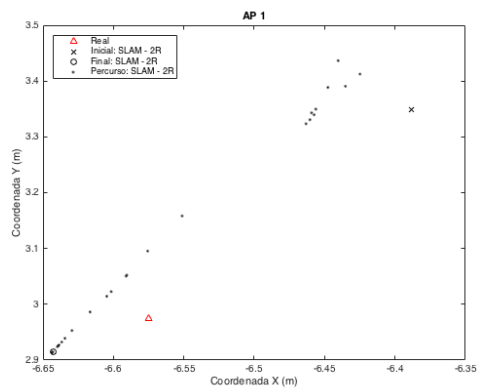
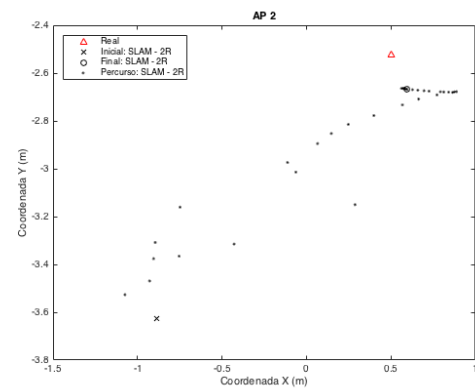


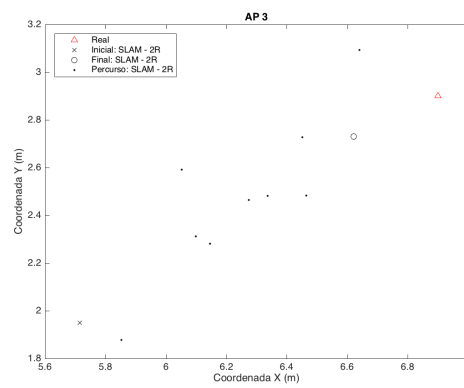
Figura 68 – Percurso dos APs em C no SLAM com 1 Receptor: (a) AP 1; (b) AP 2; (c): AP 3 - Cenário C.



(a)

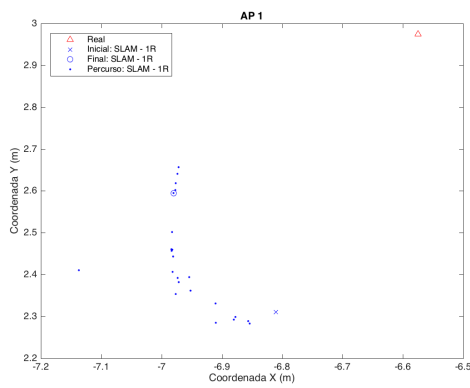


(b)

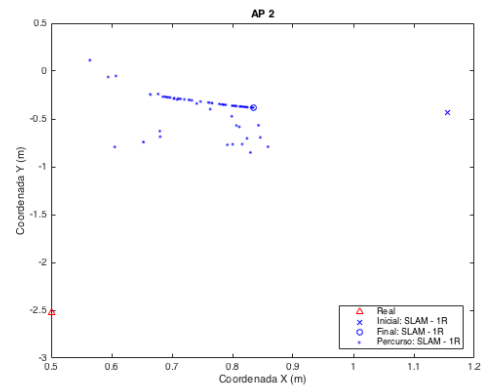


(c)

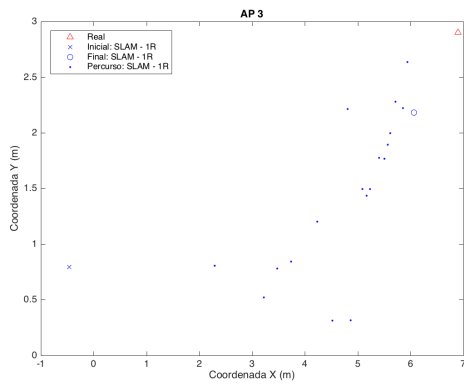
Figura 69 – Percurso dos APs em C no SLAM com 2 Receptores: (a) AP 1; (b) AP 2; (c): AP 3 - Cenário C.



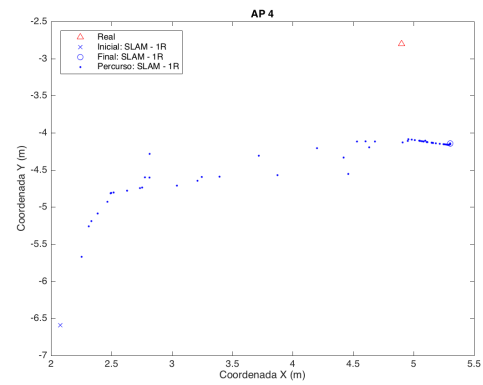
(a)



(b)



(c)



(d)

Figura 70 – Percurso dos APs em A no SLAM com 1 Receptor: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4 - Cenário C.

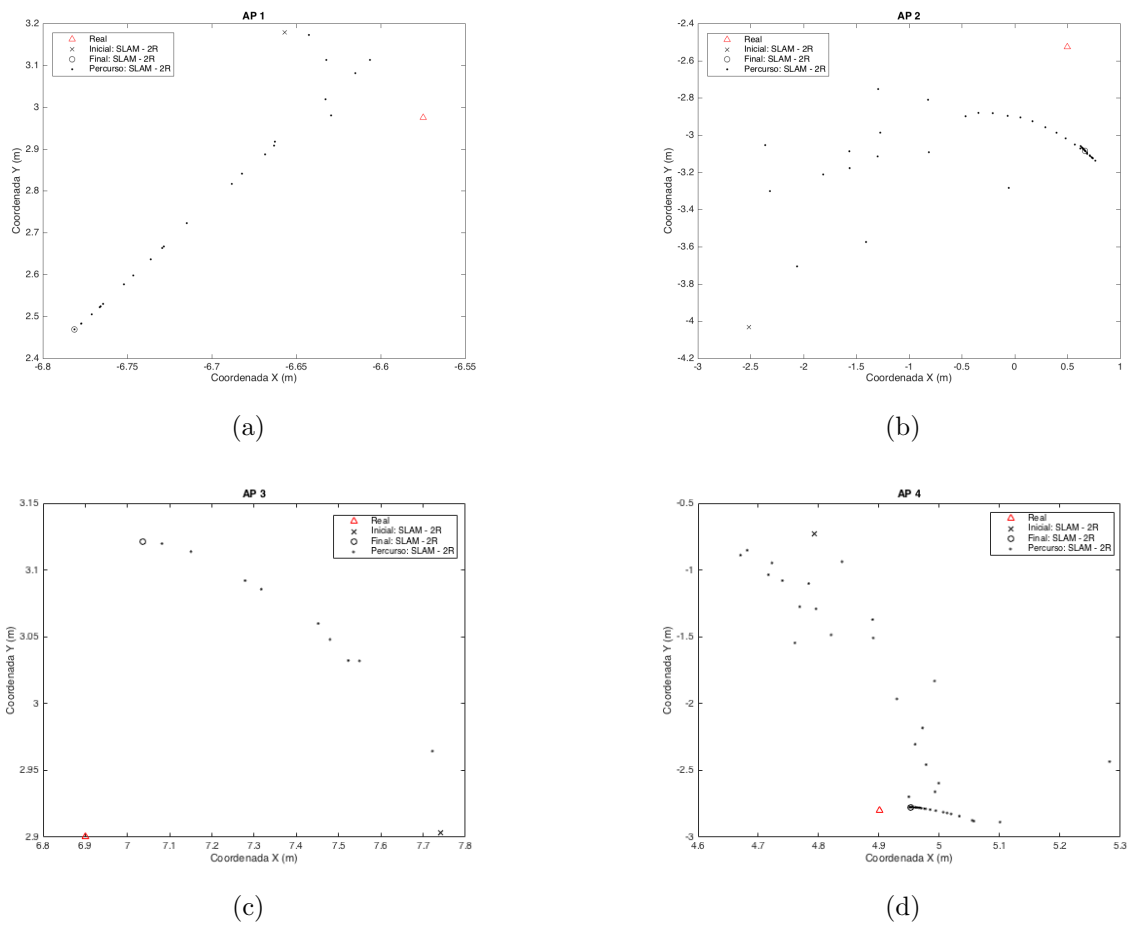


Figura 71 – Percorso dos APs em A no SLAM com 2 Receptores: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4 - Cenário C.

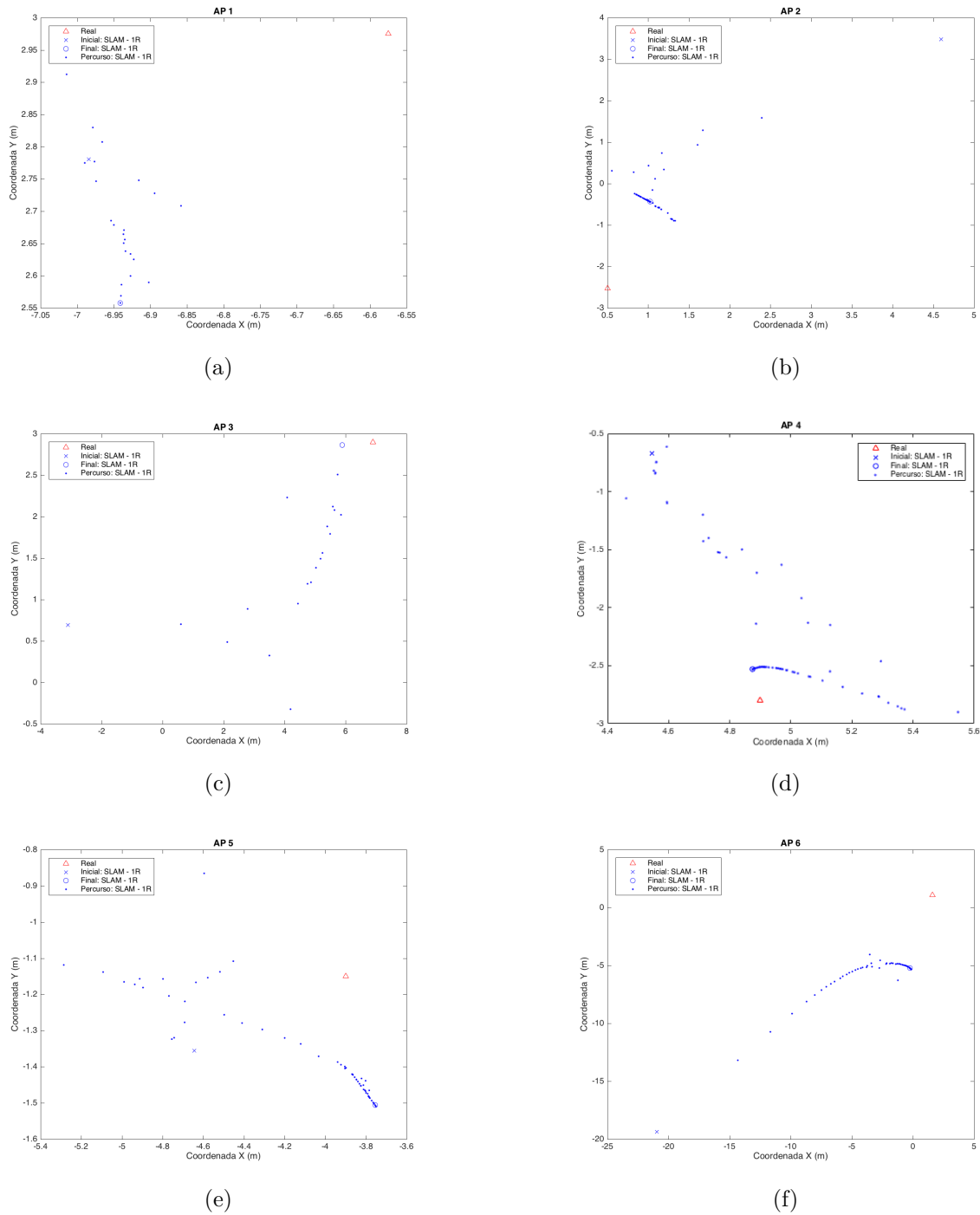
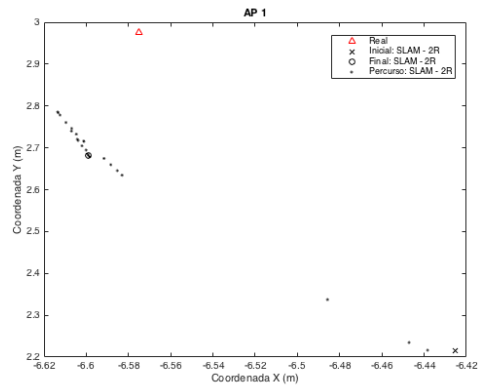
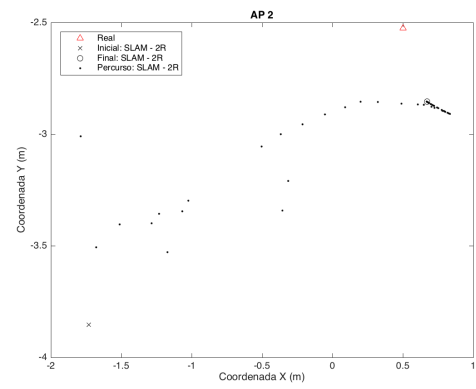


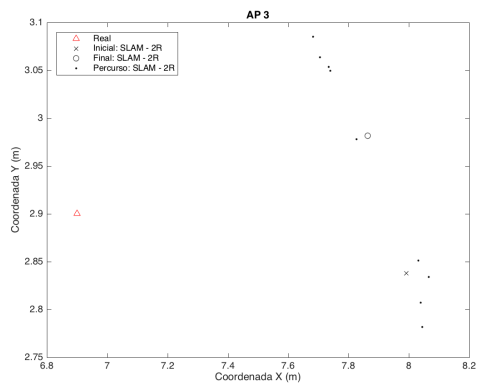
Figura 72 – Trajetória dos APs no SLAM com 1 Receptor: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4; (e): AP 5; (f) AP 6 - Cenário C.



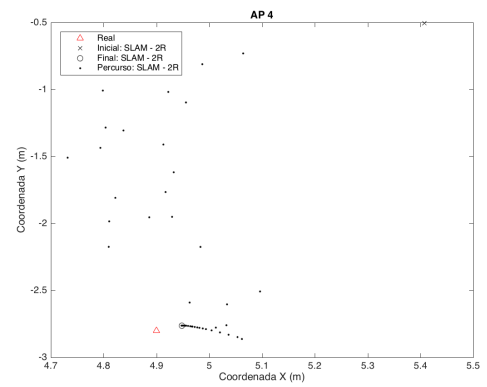
(a)



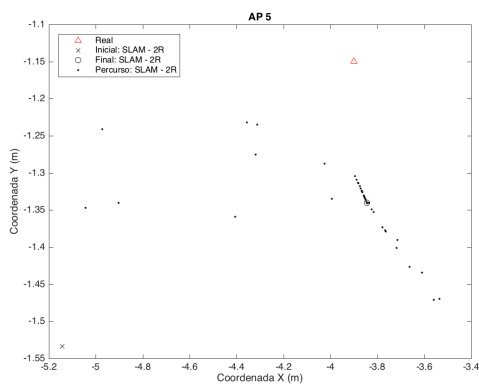
(b)



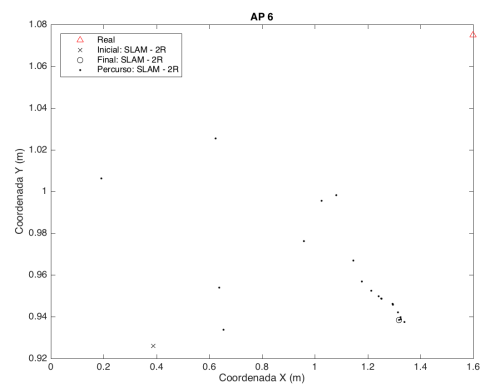
(c)



(d)



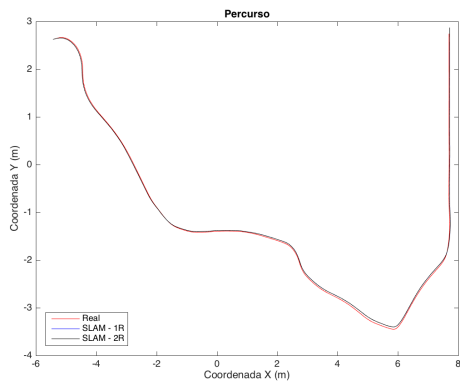
(e)



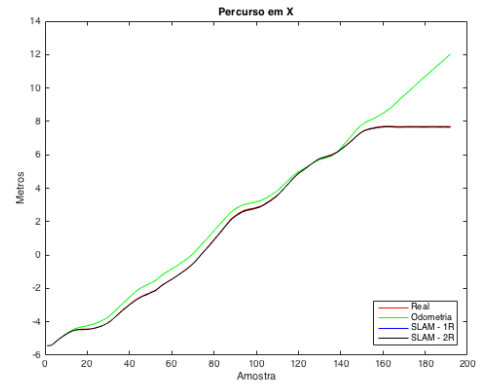
(f)

Figura 73 – Trajetória dos APs no SLAM com 2 Receptores: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4; (e): AP 5; (f) AP 6 - Cenário C.

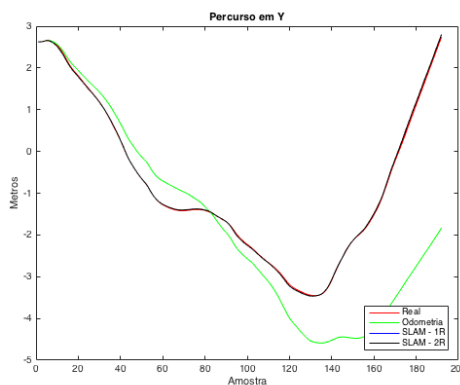
APÊNDICE D – Resultados Adicionais da Cena D



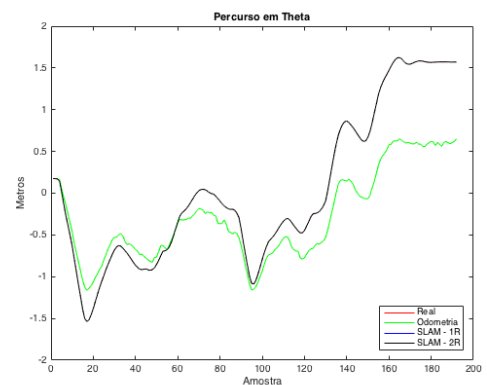
(a)



(b)

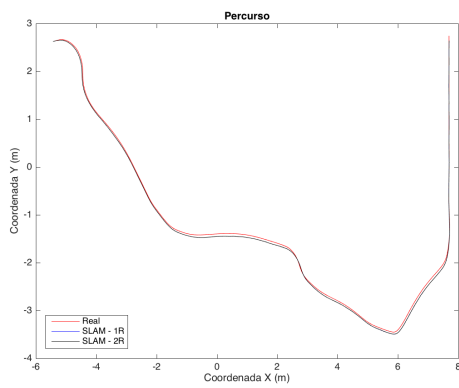


(c)

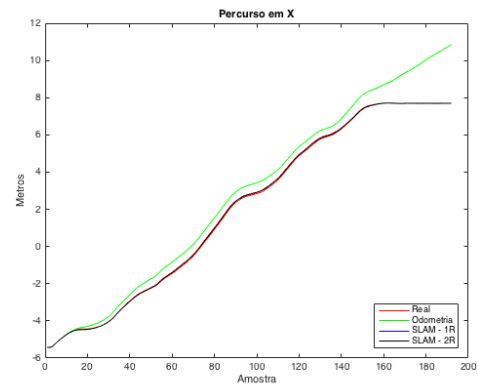


(d)

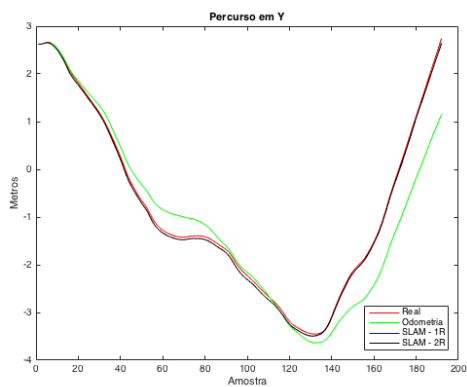
Figura 74 – Trajetória do Robô com 3 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário D.



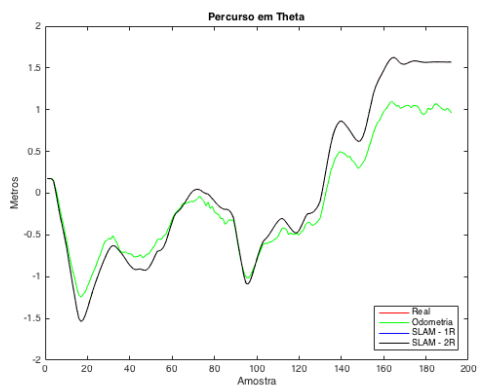
(a)



(b)

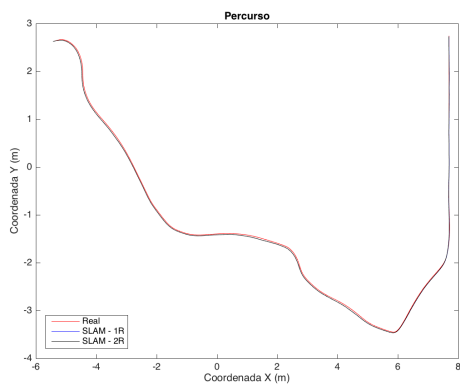


(c)

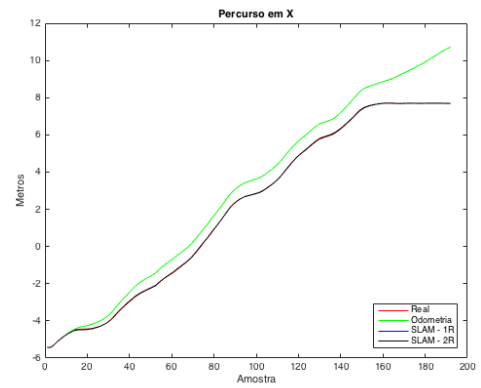


(d)

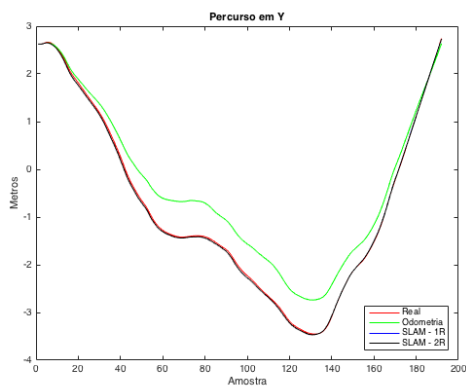
Figura 75 – Trajetória do Robô com 4 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário D.



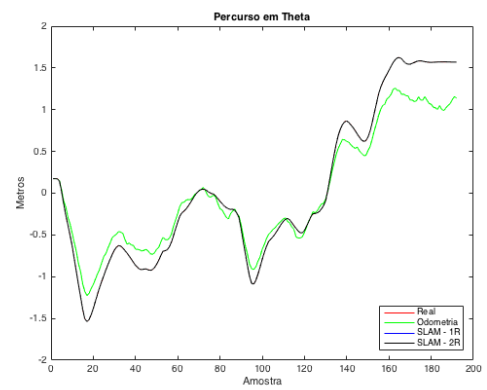
(a)



(b)



(c)



(d)

Figura 76 – Percorso do Robô em D com 6 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário D.

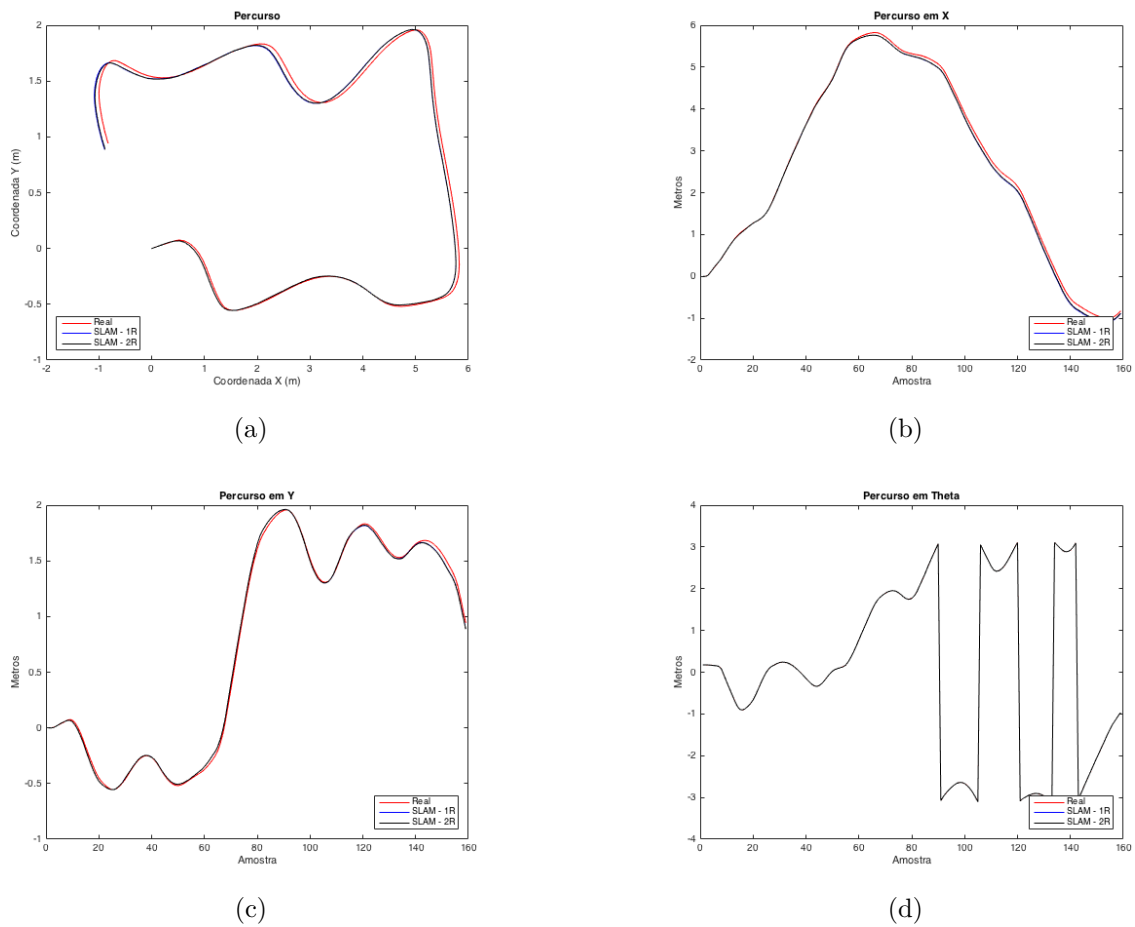
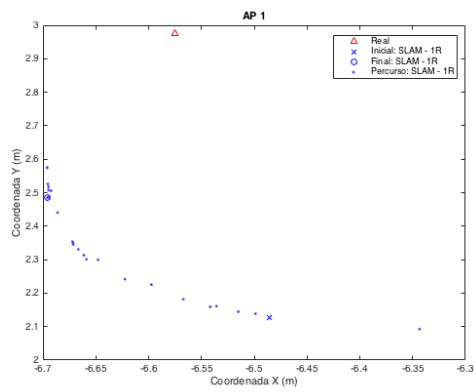
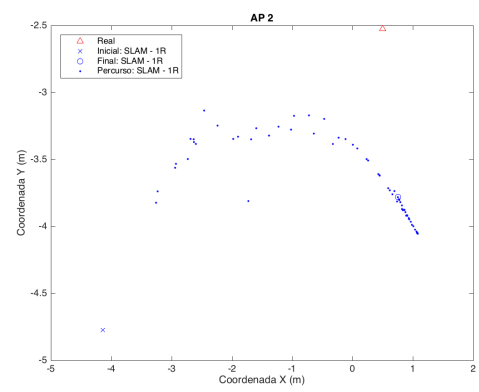


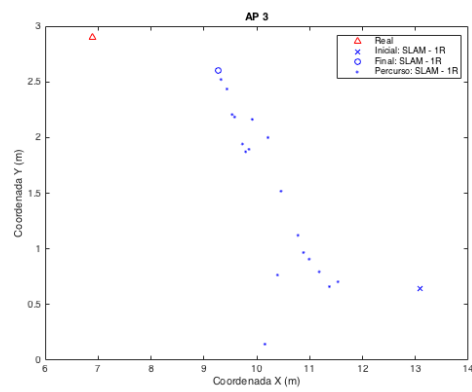
Figura 77 – Trajetória do Robô com 8 APs: (a) Coordenadas x e y; (b) Coordenada x; (c) Coordenada y; (d) Orientação do Robô (theta) - Cenário E.



(a)



(b)



(c)

Figura 78 – Trajetória dos APs no SLAM com 1 Receptor: (a) AP 1; (b) AP 2; (c): AP 3 - Cenário D.

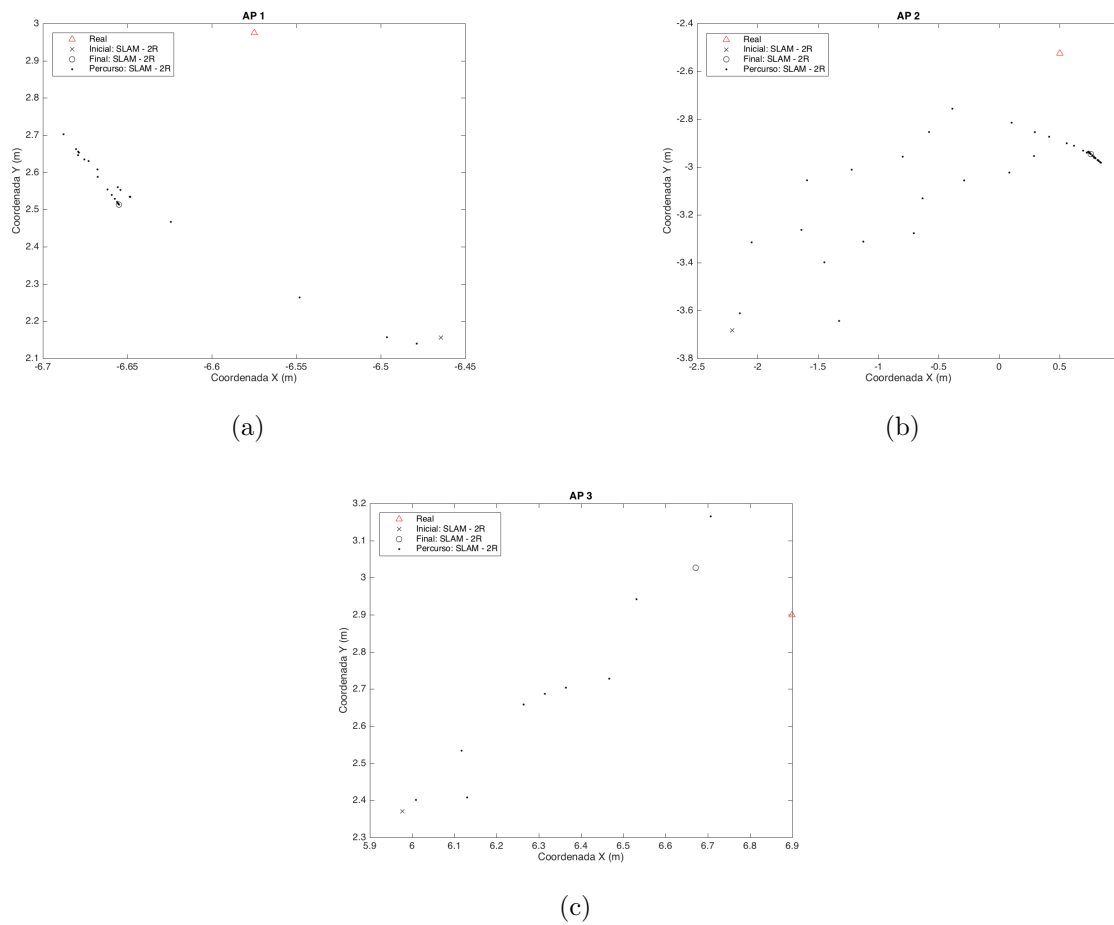
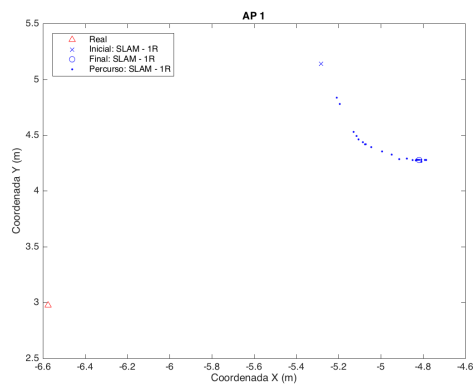
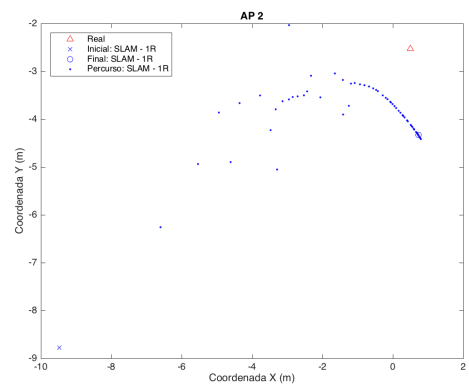


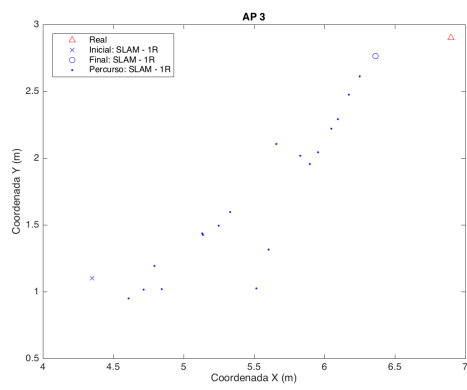
Figura 79 – Trajetória dos APs no SLAM com 2 Receptores: (a) AP 1; (b) AP 2; (c): AP 3 - Cenário D.



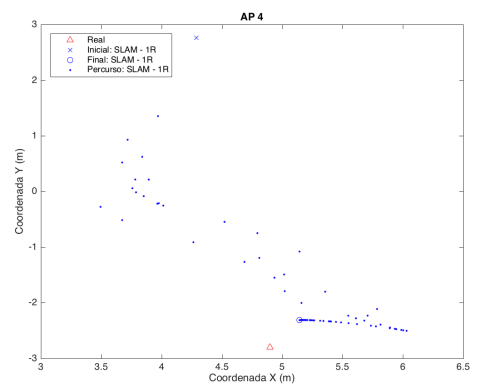
(a)



(b)



(c)



(d)

Figura 80 – Trajetória dos APs no SLAM com 1 Receptor: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4 - Cenário D.

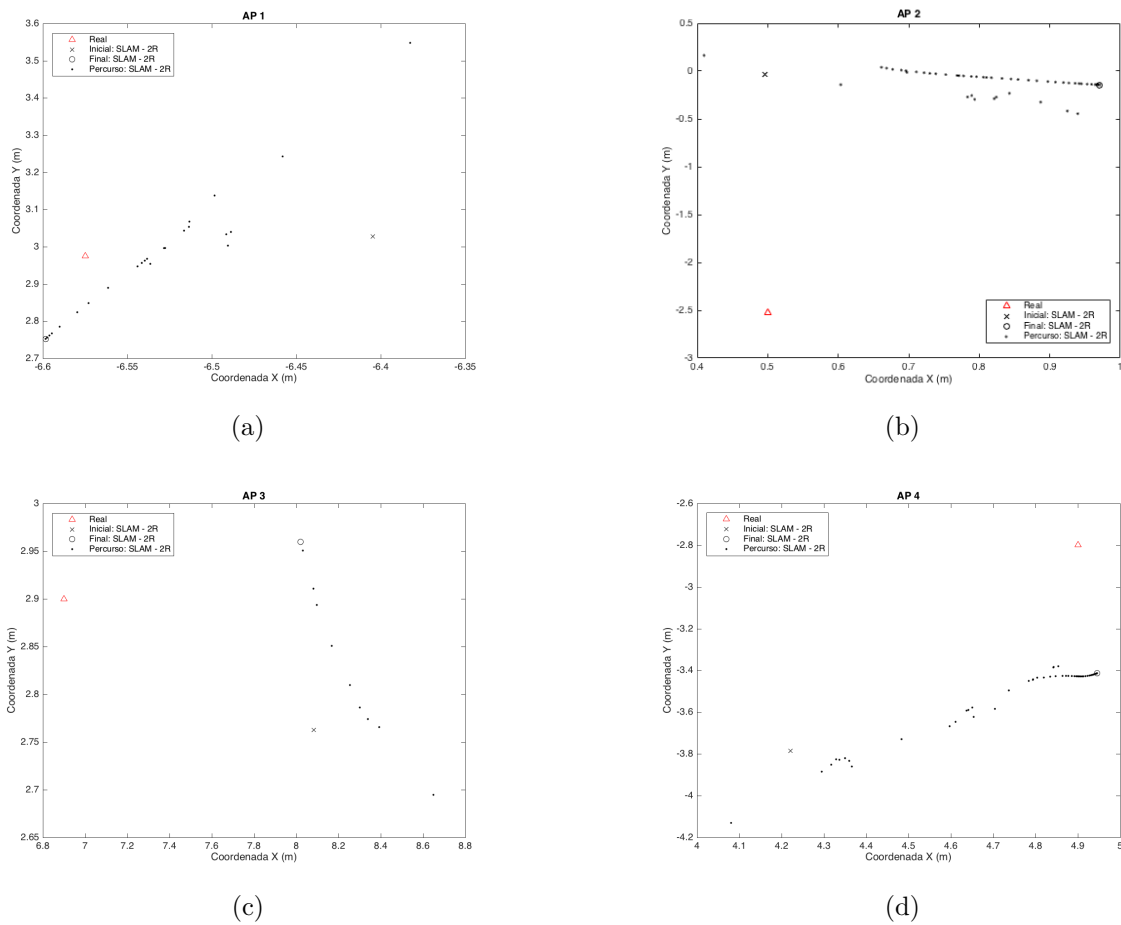
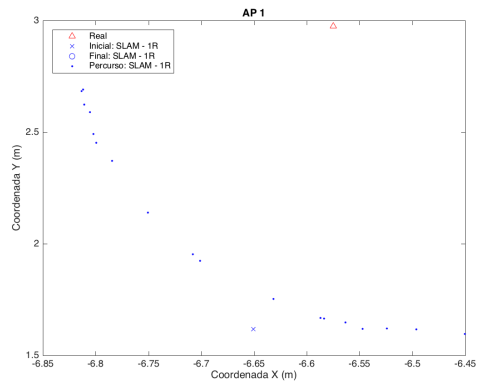
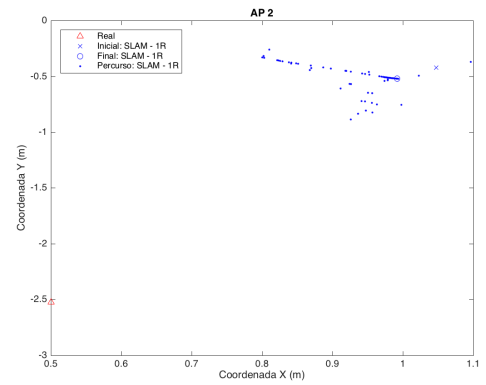


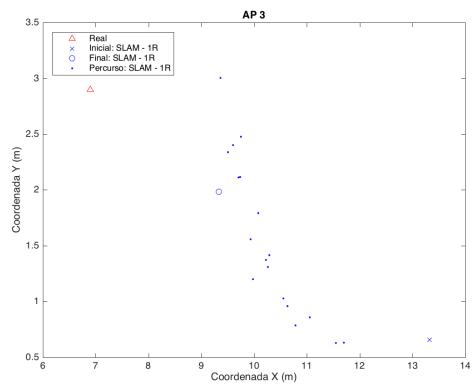
Figura 81 – Trajetória dos APs no SLAM com 2 Receptores: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4 - Cenário D.



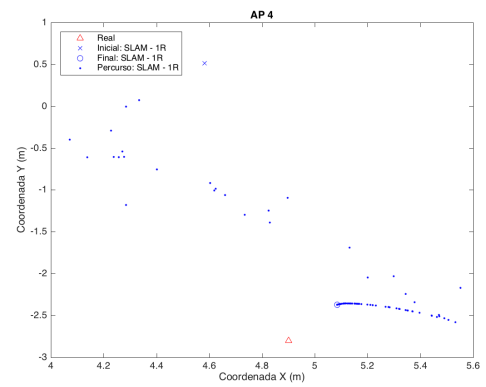
(a)



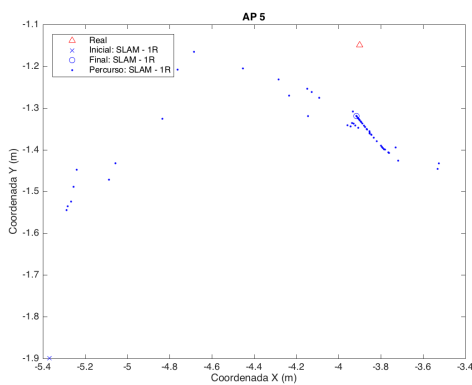
(b)



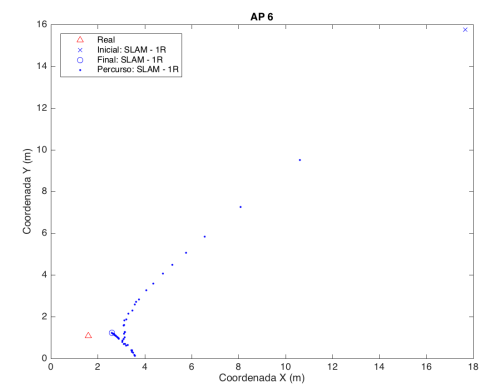
(c)



(d)



(e)



(f)

Figura 82 – Trajetória dos APs no SLAM com 1 Receptor: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4; (e): AP 5; (f) AP 6 - Cenário D.

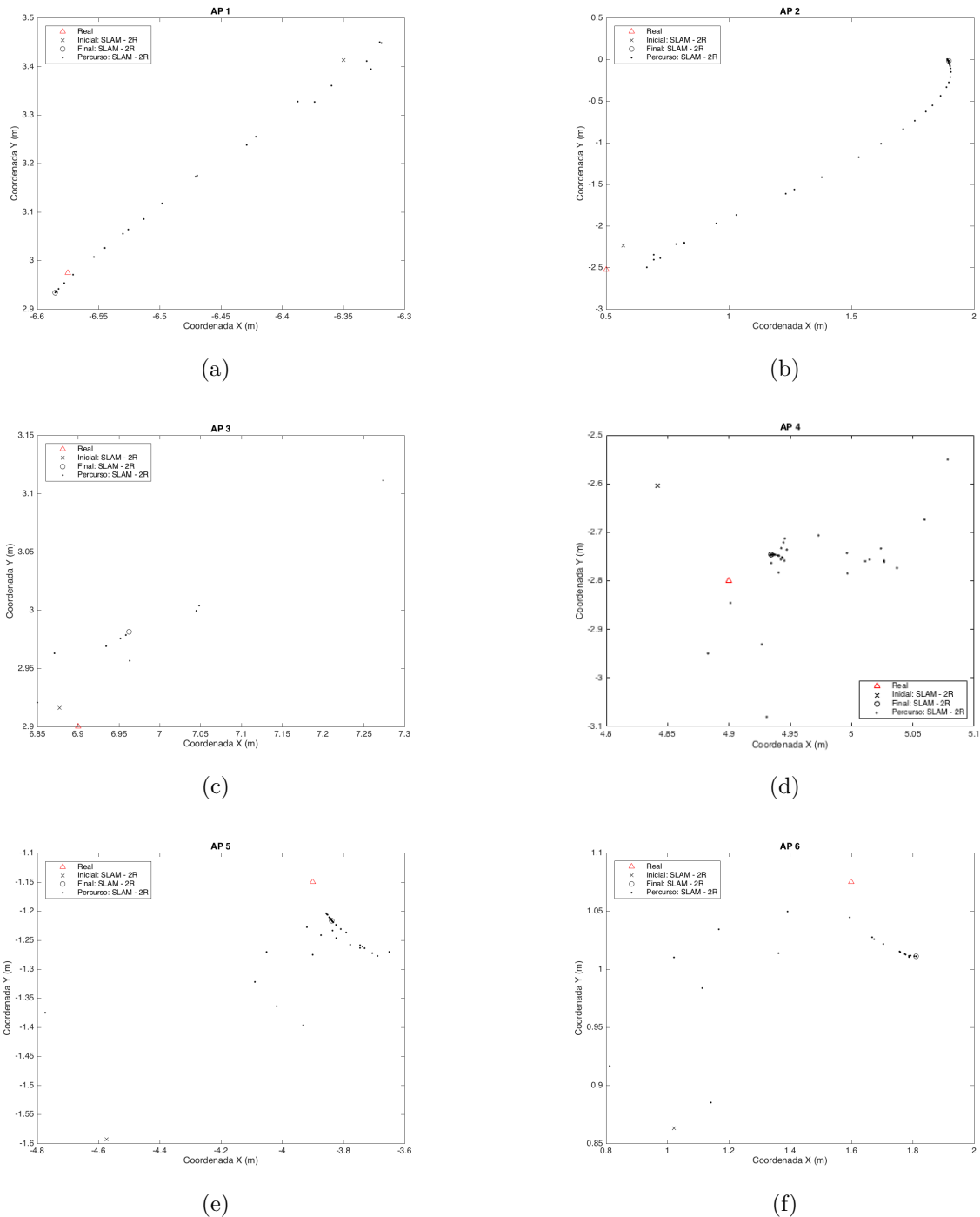


Figura 83 – Trajetória dos APs no SLAM com 2 Receptores: (a) AP 1; (b) AP 2; (c): AP 3; (d) AP 4; (e): AP 5; (f) AP 6 - Cenário D.